Sicherheit biometrischer Systeme

Analyse der Sicherheit und Rückführbarkeit eines biometrischen Hash Algorithmus für die dynamische Handschrift

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Karl F. Kümmel, M.Sc.

geb. am 13.12.1978 in Berlin

Gutachterinnen/Gutachter

Prof. Dr.-Ing. Jana Dittmann, Otto-von-Guericke-Universität Magdeburg
Prof. Dr.-Ing. Claus Vielhauer, Technische Hochschule Brandenburg
Prof. Dr. Rüdiger Grimm, Universität Koblenz – Landau
Univ.-Prof. Dr. Andreas Uhl, Paris Lodron Universität Salzburg

Magdeburg, den 11.04.2021

Zusammenfassung

Wie selbstverständlich verschaffen sich Millionen von Menschen mehrmals täglich mit ihren körperlichen Merkmalen Zugang zu ihrem Smartphone. Mittels Gesichtsscan oder Fingerabdruck, die durch geeignete Sensoren erfasst und durch spezielle Algorithmen verarbeitete werden, erlaubt das Smartphone den bequemen Zugang zu Kontaktdaten, sozialen Medien und mobilen Bezahlsystemen. Doch wie sicher sind diese biometrischen Verifikationssysteme? Mit diesen und anderen Themen befasst sich die IT-Sicherheit, speziell die Biometrie bzw. die Benutzerverifikation mit biometrischen Daten. In dieser wissenschaftlichen Arbeit wird die Sicherheit eines handschriftenbasierten biometrischen Verifikationsverfahrens analysiert. In verschiedenen Bereichen der Arbeit werden Schwachstellen identifiziert und am Ende Verbesserungsvorschläge gegeben, die potentiell auch auf andere Verifikationsverfahren übertragen werden können.

Im ersten Bereich der Arbeit werden u.a. ausgesuchte bekannte Angriffsverfahren untersucht, die biometrische Verifikationssysteme versuchen zu kompromittieren. Dabei sind die Angriffsverfahren in der Regel auf bestimmte biometrische Modalitäten (Fingerabdruck, Iris, Handschrift, Stimme etc.) konzipiert und optimiert worden. Es besteht jedoch die Gefahr, dass einige Angriffe adaptiert werden und auf eine andere biometrische Modalität angewendet werden können. Im ersten Teil dieser Arbeit wird daher ein neues Klassifikationsverfahren vorgestellt, welches es potentiell ermöglicht, solche Angriffsverfahren leichter zu identifizieren. Dieses Klassifikationsverfahren wird exemplarisch an ausgewählten Angriffsverfahren angewendet. Anschließend wird eine klassifizierte und als gefährlich identifizierte Angriffstechnik so adaptiert, dass es auf das handschriftenbasierten Verifikationssystem angewendet werden kann. Dieser Angriff ermöglicht es, künstliche biometrische Daten auf Basis von biometrischen Referenzdaten zu erzeugen. Diese künstlich erzeugten biometrischen Daten können eingesetzt werden, um unberechtigten Zugang zu einem Verifikationssystem zu erlangen. In experimentellen Tests wird u.a. diese potentielle Gefahr des Angriffs aufgezeigt. Anschließend werden Gegenmaßnahmen vorgeschlagen, welche die Durchführung des Angriffs unter Ausschluss bestimmter biometrischer Merkmale verhindern. Weitere Tests ohne diese Merkmale zeigen, dass sich die Verifikationsperformanz in vier der fünf getesteten Semantikklassen (Schreibinhalten) sogar verbessert hat. Die Designvorschläge können auch auf andere (handschriftenbasierte) Verifikationssysteme übertragen werden.

Im zweiten Teil der Arbeit wird eine neue Methode zur Generierung von künstlichen biometrischen Handschriftendaten vorgestellt. Diese können eingesetzt werden, um beispielsweise Testdatenbanken zu erstellen oder Testangriffe auf biometrische Systeme u.a. auch auf das handschriftenbasierte Verifikationssystem, durchzuführen. Daher soll dieses Verfahren unter anderem künstliche Schreibindividuen erzeugen, die wiederum beliebig viele künstliche Schreibsignale generieren können. Eine weitere Eigenschaft des Verfahrens ist die freie Wahl des Schreibinhalts der künstlich erzeugten Schreibsignale. Erste experimentelle Ergebnisse zeigen, dass die erzeugten Handschriftensignale sich teilweise wie reale Handschriftensignale verhalten, das gilt insbesondere für die Reproduktionsraten. Die Verifikationsperformanz der künstlich erzeugten Handschriftensignale zeigt hingegen ein unnatürliches Verhalten. So sind beispielsweise bei der Bestimmung der Verifikationsperformanz von ausschließlich künstlich erzeugten Schreibindividuen untereinander sehr geringe Gleichfehlerraten (EER) von unter 1% gemessen worden. Der

Schreibinhalt ist teils gut erkennbar, jedoch wirken die Handschriftensignale bei genauer Betrachtung unnatürlich.
Im letzten Teil der Arbeit wird die bekannte Angriffsform Hill-Climbing und deren Gefahrenpotential auf das biometrische Handschriftenverifikationsverfahren hin untersucht. Dabei werden zunächst die Punkte eines biometrischen Verifikationssystems identifiziert, an denen die benötigten Daten für den Hill-Climbing-Angriff eingespielt bzw. ausgelesen werden können. Anschließend wird ein bekanntes Hill-Climbing-Angriffsverfahren adaptiert und auf das handschriftenbasierten Verifikationsverfahren angewendet sowie evaluiert. Die experimentellen Ergebnisse zeigen eine teils sehr hohe Erfolgsrate und offenbaren die potentielle Gefahr die von solchen Angriffen ausgehen können. Aufbauend auf diesen Ergebnissen werden Designvorschläge zur Verbesserung des Schutzes vor Hill-Climbing-Angriffen für das handschriftenbasierten Verifikationsverfahren gegeben.
Mit dieser wissenschaftlichen Arbeit können in allen durchgeführten Bereichen, Designvorschläge zum Schutz des handschriftenbasierten Verifikationsalgorithmus gegeben werden. Diese können zum Teil auch auf andere Systeme übertragen werden und tragen zum Schutz dieser Systeme bei.

1 Einführung und Motivation

In diesem Kapitel wird die Motivation zur Erstellung der Arbeit erläutert. Weiterhin werden einführende Worte in die Thematik gegeben, die Einordnung und Abgrenzung der Arbeit dargelegt sowie das Ziel der Arbeit aufgezeigt.

An dieser Stelle wird darauf hingewiesen, dass diese Arbeit auf bereits veröffentlichte Arbeiten des Autors mit weiteren Co-Autoren (siehe Abschnitt 10 Publikationen des Autors) aufbaut, welche in dieser Arbeit an den entsprechenden Stellen referenziert werden. Anregungen zu Ideen und Konzepten dieser Arbeit entstanden u.a. in und aus Gesprächen und Diskussionen an der Technischen Hochschule Brandenburg sowie der Otto-von-Guericke Universität Magdeburg.

1.1 Einführung

Die Notwendigkeit der Authentifizierung spielt in vielen Bereichen des alltäglichen Lebens eine große Rolle. Formell betrachtet ist sie die Verifizierung einer bestimmten Behauptung einer Authentizität. Das heißt, die Authentifizierung soll prinzipiell sicherstellen, dass es sich um ein Original handelt, wobei sich eine Authentifizierung nicht nur auf Menschen, sondern auf beliebige materielle oder immaterielle Gegenstände beziehen kann. Diese Gegenstände können beispielsweise Kunstgegenstände, Vertragsdokumente oder elektronische Dokumente sein. Der Nachweis eines originalen authentischen Dokuments ist bspw. im Vertragswesen oder im Kunsthandel essentiell. Niemand möchte ein falsches Vertragsdokument unterschreiben (bewilligen), dessen Inhalt nicht die beabsichtigten und ausgehandelten Vertragsklauseln enthält. Ebenso wenig möchte ein Kunsthändler für ein vermeintlich originales Kunstwerk viel Geld bezahlen, welches sich später ggf. als Fälschung entpuppt. Die beiden Bespiele sind im Kontext der Authentifizierung sicherlich interessant, sollen jedoch nicht Teil dieser Arbeit sein.
In dieser Arbeit soll ausschließlich die Authentifizierung einer oder mehrere Personen betrachtet werden. Diese personenbezogene Authentifizierung dient beispielsweise der Zugangsbeschränkung zu schützenswerten Räumlichkeiten oder Anlagen wie z.B. militärischen Einrichtungen, Forschungslaboren und Serverräumen. Auch private Wohnungen und Häuser sind in der Regel mit zugangsbeschränkenden Maßnahmen (z.B. abschließbare Eingangstür) ausgestattet, um unberechtigten Personen den Zugang zu verwehren. Es existieren derzeit drei mögliche Methoden der Authentifizierung einer Person.
Zum einen kann sich eine Person auf Grund einer nur ihre übertragene geheime Information autorisieren, dies kann unter anderem ein Passwort oder eine PIN sein. Zum anderen kann sich eine Person mit einem bestimmten Besitztum authentifizieren. Ein solcher Besitz kann beispielsweise ein Schlüssel oder Token darstellen. Neben dem Besitz oder geheimem Wissen kann die Authentifizierung einer Person auch mit Hilfe der Biometrie erfolgen. Hierbei werden bestimmte biometrische Charakteristiken einer Person erfasst und mit einer hinterlegten Referenz verglichen. Nachfolgend sind die drei existierenden Authentifizierungsmöglichkeiten zur Übersicht aufgelistet (siehe dazu auch [ScSa12]).

1. **Geheimes Wissen:** Eine Person autorisiert sich auf Grund einer nur ihr übertragenen geheimen Information. Dies kann unter anderem ein Passwort oder eine PIN sein.

2. **Besitz:** Mit Hilfe eines bestimmten Besitztums wird eine Authentifizierung durchgeführt. Dieser Besitz kann beispielsweise ein Schlüssel oder Token sein.
3. **Biometrie:** Eine Person autorisiert sich auf Grund einer bestimmten biometrischen Charakteristik. Biometrische Charakteristiken für eine Authentifizierung können beispielsweise die Form des Fingerabdruckes oder die Struktur der Iris sein.

Die hier genannten Authentifizierungsmöglichkeiten sind beliebig kombinierbar. So kann beispielsweise der Zugang zu einem Raum nur mit dem richtigen Türschlüssel und gültiger PIN erfolgen. Es sind auch mehrere Kombinationen aus einer Methode denkbar z.B. Zugangsbeschränkung durch PIN und Passwort oder Authentifizierung am Smartphone mittels Fingerabdruck und Gesichtserkennung. Weiterhin sind auch Authentifizierungen mit mindestens einer Methode denkbar. Wenn beispielsweise ein Auto mit einem gültigen Schlüssel oder einem authentischen Fingerabdruck geöffnet werden kann. Hier ist eine Fülle an Kombinationsmöglichkeiten denkbar, die an dieser Stelle nicht weiter ausgeführt werden sollen. Vielmehr soll auf die Methode der biometrischen Authentifizierung näher eingegangen werden.
Derzeit existiert eine Vielzahl verschiedener sogenannter biometrischer Modalitäten, die für den Zweck der Authentifizierung herangezogen werden können, eine Übersicht dazu gibt Jain et al in [JaNN08]. Der Fingerabdruck ist wohl einer der bekanntesten biometrischen Modalitäten. Weitere Modalitäten sind unter anderem Sprache, Gesicht, Handvenen und Handschrift. Die Handschrift hat gegenüber anderen biometrischen Modalitäten einige Vorteile. So ist in der heutigen Gesellschaft die Unterschrift als ein Akt der aktiven Willensbekundung, z.B. die Zeichnung von Verträgen oder Vollmachten, weitestgehend akzeptiert. Entsprechend sind Menschen weniger abgeneigt, eine Unterschrift auf einem Handschriftensensor zu tätigen als beispielsweise bei der Paketannahme in ein Mikrofon zu sprechen oder einen Fingerabdruck zu präsentieren. Die Handschrift als biometrische Modalität soll daher in dieser Arbeit besonders betrachtet werden. Insbesondere soll auf die Sicherheit von Authentifizierungsmechanismen für die Handschrift eingegangen werden. Diese Mechanismen sollen u.a. sicherstellen, dass die biometrischen Referenzdaten geschützt werden und verschiedene Personen innerhalb eines Systems voneinander unterscheidbar sind.

Die Authentifizierung einer Person bzw. einer behaupteten Identität steht oftmals am Anfang einer Autorisierungskette. Nach einer erfolgreichen Authentifizierung erfolgt in der Regel die Autorisierung. Hier werden der Person die ihm bzw. ihr zustehenden Zugangsberechtigungen zugewiesen. Meistens nimmt eine Person die beiden Schritte (Authentifizierung und Autorisierung) im Einzelnen nicht wahr, sie verschmelzen in ihren Augen. Der Anwender registriert lediglich den Prozess der "Anmeldung am System" siehe auch [Balz11].
Mit der ständig wachsenden Anzahl webbasierten Anwendungen wie Soziale Netzwerke, Kurznachrichtendienste, Online Versandhäuser, Bewerbungsportale usw. erhöht sich somit auch die Anzahl der täglichen Authentifizierungen. Viele Menschen verwenden derzeit mehrere zugangsgeschützte Anwendungen und besitzen oftmals auch verschiedene Zugangsdaten und Passwörter. Erlangt ein potentieller Angreifer Zugangsdaten einer Person, kann er buchstäblich in die Rolle der Person schlüpfen und dies zu seinen Gunsten ausnutzen. So kann in etwa ein finanzieller Schaden oder eine Rufbeschädigung erfolgen, um nur zwei Beispiele zu nennen.

Es ist demnach essentiell, dass eine Authentifizierung nicht nur technisch sicher abläuft, sondern auch sichergestellt ist, dass sich die berechtigte Person vor Ort befindet und nicht ein Angreifer, der sich unberechtigt die Zugangsdaten angeeignet hat. Hierbei kann die Biometrie einen wichtigen Beitrag leisten, denn anders als bei geheimem Wissen oder Besitz, können biometrische Daten nicht ohne weiteres weitergegeben werden.
Darüber hinaus bietet die biometrische Authentifizierung noch einige weitere Vorteile. Die Tatsache, dass man seine biometrischen Merkmale in der Regel stets bei sich trägt verhindert, dass sie gestohlen oder verlegt werden können. Weitere Aspekte der Biometrie werden in den folgenden Abschnitten beleuchtet.
In den nachfolgenden Abschnitten wird ausführlicher auf die Biometrie, biometrische Modalitäten und Charakteristiken sowie den Unterschied beider eingegangen.

1.2 Einordnung und Abgrenzung

Eines der großen Ziele der IT-Sicherheit neben vielen anderen ist es, sichere Authentifizierungsmechanismen und -techniken bereitzustellen. Das heißt, eine Person A kann sich mit seinen Zugangsdaten (Besitz, Geheimes Wissen und/oder Biometrie) z.B. an einem System mit Zugangsbeschränkung anmelden, wobei eine andere potentiell unerwünschte Person keinen Zugang zum selben System erhält. Wie bereits im Abschnitt 1.1 erläutert, bietet die Authentifizierung mittels biometrischer Charakteristiken einige Vorteile gegenüber den klassischen Methoden (Besitz und geheimes Wissen). Es gilt, die biometrischen Verifikations- und Authentifizierungssysteme so zu gestalten, dass unberechtigte Personen keinen Zugang zum System erhalten, wenn Sie Ihre biometrischen Daten am Sensor präsentieren. Ähnlich wie ein Sicherheitsschloss darf nur ein bestimmter Schlüssel in der Lage sein, die entsprechende Tür zu öffnen, auch wenn ein Schlüssel mit ähnlichem Schlüsselbart verwendet wird. Ein biometrischer Authentifizierungsalgorithmus muss u.a. in der Lage sein, ähnliche biometrische Charakteristiken von verschiedenen Personen zu unterscheiden. Gleichzeitig muss der Algorithmus die natürlichen Schwankungen der Charakteristiken einer Person erkennen und der entsprechenden Person zuordnen können.
In den vergangenen Jahren wurden zahlreiche Verfahren zur Authentifizierung von Personen auf der Basis biometrischer Charakteristiken vorgestellt, wie z.B. hier von Riaz et al. in [RiRK18] zusammengetragen wurde. Ziel dieser dort vorgestellten Arbeiten ist es, Personen anhand ihrer biometrischen Charakteristiken, wie Beispielsweise Fingerabdrücke, Iris (Auge) oder Gesicht, eindeutig zu verifizieren. Hierfür müssen die biometrischen Verifikationssysteme mit einem entsprechenden Verfahren ausgestattet sein, welches es ermöglicht, eine Person anhand der verwendeten biometrischen Charakteristik wiederzuerkennen und gleichzeitig von anderen Personen zu Unterscheiden. Im Grunde genommen ist ein biometrisches System ein Klassifikationssystem, welches biometrische Charakteristiken einer bestimmten Person (Klasse) zuordnet bzw. nicht zuordnet. In [JaRP04] wird dieses Thema ausführlicher behandelt.

Biometrische Charakteristiken oder hier auch Modalitäten genannt, unterlaufen jedoch immer einer gewissen Unschärfe. Das heißt, jede biometrische Modalität schwankt und je nach physischer und/oder psychischer Verfassung einer Person (z.B. Krankheit, Gemütszustand etc.) kann diese Schwankung mitunter größer ausfallen.
Diese Unschärfe der biometrischen Charakteristiken bringt u.a. zwei direkte Herausforderungen bei der Verarbeitung der Daten mit sich. Zum einen kann es vorkommen, dass die biometrischen Charakteristiken der ursprünglichen Person während des

Authentifizierungsprozesses nicht mehr korrekt zugeordnet werden können. Dieses Ereignis ist die Folge einer sogenannten *Intraklassen-Variabilität* und kann unter Umständen dazu führen, dass eine Person mehrmals die Authentifizierungsprozedur durchlaufen muss, bevor sie korrekt erkannt wurde. Zum anderen kann es vorkommen, dass die biometrischen Charakteristiken einer Person dem einer anderen sehr ähneln und ggf. falsch zugeordnet werden, Jain et al. verdeutlichen dies in ihrer Arbeit [JaRP04]. Das kann je nach Beschaffenheit des Authentifizierungssystems zur Folge haben, dass Person A als Person B verifiziert wurde oder umgekehrt. Dieses Phänomen wird als *Interklassen-Ähnlichkeit* (auch *Interklassen-Variabilität*) bezeichnet (siehe Abschnitt 4.1.4 Biometrische Fehlerraten).
Eine Schwankung biometrischer Charakteristiken gilt es in einem biometrischen Verifikationssystem zu kompensieren. Die Kompensation der natürlichen Schwankungen einer Person, die es erst ermöglicht, eine automatische Benutzererkennung durchzuführen, ist zugleich eine Schwachstelle eines biometrischen Erkennungssystems und eine große Herausforderung in der Biometrie. Sie lässt Spielraum für die Eingabemenge zu und eröffnet somit potentiellen Angreifern Chancen, diese zu nutzen. Sei es durch einen nicht zwingend gewollten zufälligen Treffer (Interklassen-Ähnlichkeit) oder beispielsweise durch das absichtliche Nachahmen einer biometrischen Eigenschaft (z.B. Sprache, Gangart oder Handschrift) oder aber auch das gezielte Klonen (Gummifingerabdruck, Gesichtsmaske usw.). Die nächste Steigerung ist ein systematischer Angriff auf ein spezielles Erkennungssystem. Hierbei werden nicht nur die Besonderheiten biometrischer Charakteristiken ausgenutzt (Unschärfe) sondern ggf. Schwachstellen im Verifikationsalgorithmus und/oder Verifikationssystems ausgenutzt, um den Authentifizierungsablauf zu umgehen/hintergehen.

In einer Vielzahl von wissenschaftlichen Arbeiten werden verschiedene Angriffstechniken auf biometrische Verifikationssysteme und Verifikationsverfahren vorgestellt (u.a. [ToMa15], [BaCr14], [GGTF12]). Hierbei gibt es Angriffstechniken, die auf viele Verifikationsverfahren, mit teils unterschiedlich verwendeten biometrischen Charakteristiken, angewendet werden können, wie z.B. in [MFA+06] beschrieben. Weiterhin existieren Angriffsmethoden, welche auf spezifische Schwachstellen eines Verifikationssystems ausgerichtet sind und auf dem ersten Blick nicht auf andere Systeme übertragbar sind. Es ist jedoch durchaus denkbar, dass ein spezielles Angriffsverfahren auch auf andere Verifikationsverfahren angewendet werden kann, auch wenn dies ursprünglich nicht vom Entwickler des Angriffsverfahrens vorgesehen war. Diese, auf andere Verfahren potentiell übertragbaren Angriffstechniken, stellen unter anderem eine potentielle Gefahr dar. Mögliche Angreifer könnten diese Angriffstechnik anpassen und auf andere Systeme anwenden. Einige biometrische Verifikationssysteme sind unter Umständen gegen mehr potentielle Angriffstechniken anfällig als bisher bekannt ist bzw. angenommen wurde.
Um die Gefahr einer Angriffsmethode für andere biometrische Verifikationssysteme besser einschätzen zu können, ist es hilfreich, einen Angriff hinsichtlich der Angriffsstrategie bzw. Angriffstechnik zu klassifizieren.
In [RaCB01] werden zum Beispiel acht mögliche Angriffspunkte auf ein klassisches biometrisches Erkennungssystem definiert. Die hier gezeigten möglichen Angriffe setzen unter anderem auf den Schnittstellen der jeweiligen Komponenten (Sensor, Merkmalsextraktor, Vergleicher etc.) auf, wo ggf. gefälschte Daten eingespielt und/oder abgefangen werden können (siehe Abbildung 1).

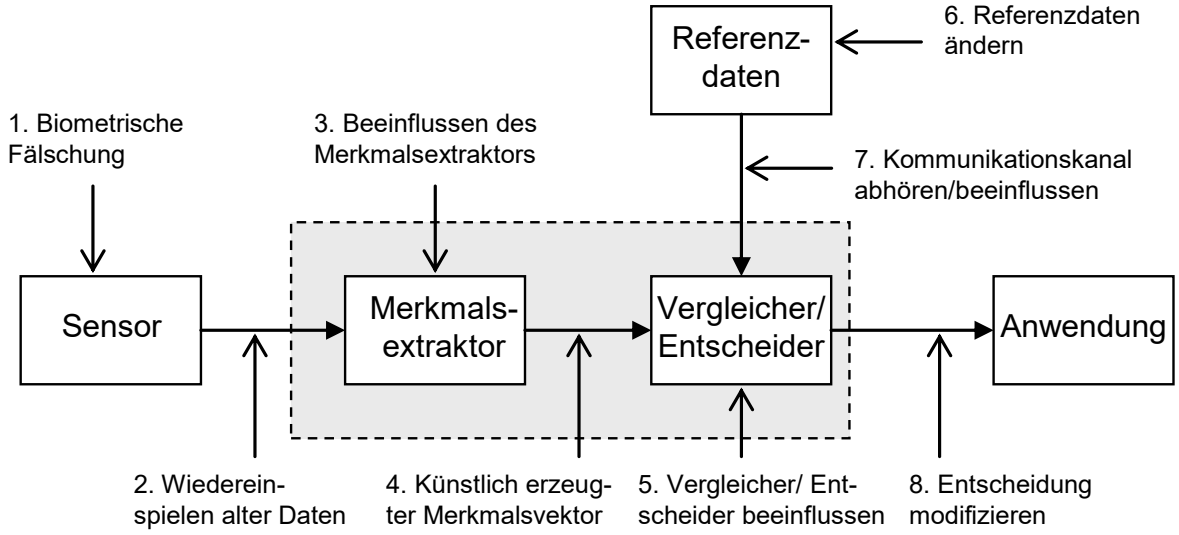


Abbildung 1 Mögliche Angriffspunkte auf ein biometrisches Erkennungssystem, adaptiert von [RaCB01]; In den Abschnitten 2.3 und 5.1.2 wird u.a. detailliert auf die Angriffspunkte eingegangen

Ein Angriff kann beispielsweise einem oder mehreren dieser acht Angriffspunkte zugeordnet werden. Diese einfache Zuordnung ermöglicht bereits eine grobe Einschätzung der auszugehenden Gefahr eines Angriffes. So kann beispielsweise ein Angriff, welcher Angriffspunkt sieben ausnutzt und auf einem Datenbankkommunikationskanal ansetzt, mitunter auf andere Systeme adaptiert werden, die ein ähnliches Datenbanksystem verwenden.

Eine wesentliche Forschungsaufgabe (Forschungsaufgabe 1 - FA1) dieser Dissertation besteht darin zu untersuchen, inwieweit bereits existierende Angriffstechniken auf andere biometrische Verifikationssysteme adaptiert werden können. Mit dem Hintergrund der Vielzahl an existierenden Angriffstechniken und Verifikationssystemen wird diese Untersuchung jedoch etwas eingeschränkt. Es werden ausschließlich ausgewählte Angriffstechniken dahingehend untersucht, inwiefern sie auf handschriftenbasierte Verifikationssysteme im Allgemeinen und auf einen speziellen Algorithmus [Viel06] angewendet werden können. Ein Angriff mit hohem Potential wird auf dem in [Viel06] vorgestellten handschriftenbasierten Verifikationssystem praktisch getestet und evaluiert. Die dabei gewonnen Erkenntnisse fließen unter anderen in Designvorschläge zur Verbesserung der Sicherheit des Algorithmus ein. Weiterhin wird diskutiert, inwieweit diese Designvorschläge auch auf handschriftenbasierte Verifikationssysteme im Allgemein anwendbar sind.

Zusätzlich wird in dieser Arbeit besonderes Augenmerk auf die Angriffspunkte 1 und 2 der in Abbildung 1 dargestellten Angriffspunkte gelegt. Ein potentieller Angreifer könnte beispielsweise gefälschte biometrische Daten erzeugen, um sich so Zugang zu einem System zu verschaffen (siehe Abbildung 1: Angriffspunkt 1). Hierfür könnte er z.B. eine Kopie (Fälschung) der biometrischen Modalität oder künstlich erzeugte biometrische Daten am Sensor präsentieren. Letzteres kann nicht ohne weiteres für jede biometrische Modalität durchgeführt werden, jedoch ist es möglich den Sensor zu umgehen, um gefälschte Daten direkt in das System einzuspielen (Abbildung 1 Angriffspunkt 2). So können entweder vorher abgefangene Daten in das System eingespielt werden (Replay Attacke) oder, wenn diese dem Angreifer nicht zur Verfügung stehen, gefälschte Daten verwendet werden. Solche gefälschten Daten können unter anderem künstlich erzeugt werden. Wie in einigen

wissenschaftlichen Beiträgen bereits gezeigt wurde z.B. in [CMLM07], besitzen synthetisch erzeugte biometrische Daten durchaus das Potential, sich unberechtigten Zugang zu einem System zu verschaffen.
Aus diesem Grund wird sich diese Arbeit näher mit Methoden zur Erzeugung von synthetischen Handschriftendaten befassen. Unter der Berücksichtigung existierender Verfahren zur Generierung von künstlichen Handschriftendaten wird ein neues Verfahren als eine weitere Forschungsaufgabe (FA2) in dieser Arbeit vorgestellt. Ein wichtiges Kriterium der künstlich zu erzeugenden Handschriftendaten soll sein, dass sie nicht künstlich wirken, sondern ein natürliches Aussehen besitzen. Weiterhin sollen zur Erzeugung der künstlichen Handschriften möglichst wenige Handschriftendaten von realen Personen verwendet werden. Die Gründe dafür sind offensichtlich, möchte man eine große Menge von künstlichen biometrischen Daten erzeugen z.B. für Performanztests, ist man unter Umständen von einer großen Menge Handschriftendaten realer Personen abhängig. Die Beschaffung solcher Daten birgt nicht nur datenschutzrechtliche Schwierigkeiten, sondern auch zeitlichen Aufwand. Die Daten müssen teils von unterschiedlichen Personen aufgenommen und verarbeitet werden, was wiederum hohen finanziellen Aufwand mit sich bringt.

Ein weiterer Teilbereich (FA3) der in dieser Arbeit behandelt wird, sind Hill-Climbing-Angriffe auf biometrische Systeme. In verschieden wissenschaftlichen Beiträgen (z.B. [GFOG11] und [GaFO07] wurden bereits Hill-Climbing-Angriffe sehr erfolgreich auf handschriftenbasierte Verifikationssystem durchgeführt. Diese Angriffe sollen analysiert und erstmals auf einen bestimmten handschriftenbasierten Verifikationsalgorithmus [Viel06] durchgeführt werden. Die gewonnen Ergebnisse fließen unter anderen in Designvorschläge zur Verbesserung der Sicherheit des Algorithmus ein. Es sollen in dieser Arbeit keine neuen Hill-Climbing-Verfahren oder Hill-Climbing-Angriffe entwickelt und evaluiert werden. Lediglich existierende Algorithmen werden diskutiert und zu Evaluationszwecken implementiert.
In dieser Arbeit werden ausschließlich handschriftenbasierte Verifikationsalgorithmen betrachtet, keine Verfahren zur Handschriftenerkennung (Zeichen- bzw. Texterkennung), welche zum Ziel haben geschriebenen Text in digitale Textsymbole (z.B. ASCII Zeichen) zu transformieren. Die Graphologie oder auch Schriftpsychologie wird in dieser Arbeit ebenfalls nicht betrachtet. Hierbei handelt es sich um die Analyse der Handschrift von Individuen zum Zwecke der psychologischen Diagnose. Auch die in der Kriminalistik verwendete Handschriftenuntersuchung bzw. Schriftvergleichung ist nicht Bestandteil dieser Arbeit.

2 Stand der Technik

Im Abschnitt 1.2 sind die Einordnung und Abgrenzung der Arbeit erläutert. Ferner werden die drei wesentlichen Teilbereiche dieser Arbeit kurz vorgestellt. Bevor eine detaillierte Definition der Ziele bzw. Forschungsaufgaben dieser Arbeit in Abschnitt 3 vorgestellt wird, soll der Stand der Technik in diesem Bereich erläutert werden. Anschließend wird mit Hilfe dieser Erläuterungen eine klare Definition der eigenen Forschungsaufgaben gegeben.
Im Abschnitt 2.1 werden grundlegende Sicherheitsmechanismen zum Schutz biometrischer Referenzdaten vorgestellt (Abschnitt 2.1.1) und weiterhin handschriftenbasierte Verifikationssysteme auf Basis der dynamischen und statischen Handschrift erläutert (Abschnitt 2.1.2). Abschnitt 2.2 zeigt typische Angriffspunkte auf biometrische Systeme im Allgemeinen. Im nachfolgenden Abschnitt 2.3 werden Angriffsmethoden und Angriffstechniken näher betrachtet. Potentielle Gegenmaßnahmen werden im Abschnitt 2.4 behandelt.

2.1 Biometrische Erkennungssysteme

Jain et al. stellen in [JaNN08] Verifikationsverfahren vor mit unterschiedlichsten biometrischen Modalitäten als Verifizierungsgrundlage. In diesem Beitrag wird unter anderem Wert auf die Beschreibung der verschiedenen Techniken zum Schutz der biometrischen Referenzdaten gelegt. Diese Sicherheitsmechanismen werden klassifiziert, verglichen und deren Vor- und Nachteile beschrieben.
In Abschnitt 2.1.1 werden diese Schutzmechanismen näher betrachtet und erläutert. Abschnitt 2.1.2 gibt einen Überblick über handschriftenbasierte Verifikationssysteme.

2.1.1 Schutzmechanismen biometrischer Referenzdaten

Biometrische Referenzdaten werden in der Regel in einem System zum Abgleich mit den aktuell präsentierten biometrischen Daten einer Person verwendet. Sie können entweder zusammen mit Referenzdaten anderer Personen zentral in einer Datenbank gespeichert werden oder dezentral z.B. auf einer Speicherkarte. In beiden Fällen sollten die Referenzdaten nicht im Klartext gespeichert werden. Klartext soll hier, in Anlehnung an die Kryptographie, für die Rohform von Daten stehen. Ähnlich wie bei Passwörtern, die in der Regel nicht als Klartext, sondern in einer verschlüsselten Form abgelegt werden.
Der Schutz von biometrischen Daten ist unter anderem so wichtig, da es sich um personenbeziehbare Daten handelt. So können anhand biometrischer Daten, je nach Modalität, z.B. das Geschlecht, die ethnische Herkunft, das Alter und/oder möglich Krankheiten abgelesen werden, siehe z.B. in [WhDV18]. Des Weiteren können mit biometrischen Rohdaten einfacher Kopien erstellt werden (z.B. Gummifingerabdruck [BCG+18]), als mit z.B. verschlüsselten Repräsentationen der biometrischen Daten. Weiterhin können unverschlüsselte biometrische Daten leichter für eine einfache Identifikation herangezogen werden ohne Zustimmung der betroffenen Person (z.B. Gesichtserkennung). Zusätzlich verpflichten einige Regularien und gesetzliche Bestimmungen (z.B. Datenschutz-Grundverordnung - DSGVO) den Schutz von biometrischen Daten. Mitunter ist ein Verstoß der gesetzlichen Bestimmungen bzw. der Missbrauch biometrischer Daten strafbar. Dies sind mitunter Gründe, weswegen es nötig ist, biometrische Daten zu schützen.
In den letzten Jahren wurden verschiedenen Mechanismen zum Schutz der Referenzdaten vorgestellt. In [JaNN08] werden einige wesentliche Schutzmechanismen erläutert, die hier kurz vorgestellt werden sollen.

Im Englischen spricht man auch von Template Protection, wenn es um den Schutz der biometrischen Referenzdaten geht. Maltoni et al. beschreiben in [MMJP03] vier Eigenschaften, welche ein Schutzmechanismus für ein ideales biometrisches Erkennungssystem besitzen sollte. Nachfolgend werden diese Eigenschaften beschrieben.

- *Vielfalt (Diversity)*: Es darf nicht möglich sein, dass ein geschütztes biometrisches Datum für verschiedene Referenzdatenbanken als Kreuzprobe verwendet werden kann (cross matching).

- *Widerrufbarkeit (Revocability)*: Es muss möglich sein, ein kompromittiertes biometrisches Referenzdatum jederzeit unkompliziert widerrufen zu können. Weiterhin sollte ein neues Referenzdatum auf Basis der gleichen biometrischen Daten erstellt werden können.

- *Sicherheit (Security)*: Es muss rechnerisch nahezu unmöglich sein, biometrische Rohdaten anhand des biometrischen Templates zu ermitteln. Auf diese Weise soll verhindert werden, dass ein potentieller Angreifer eine Fälschung auf Basis eines gestohlenen Templates erstellt.

- *Erkennungsperformanz (Performance)*: Der Schutzmechanismus sollte die Erkennungsperformanz (FAR und FRR, siehe Abschnitt 4.1.4) nicht negativ beeinflussen.

Die Herausforderung in der Entwicklung eines Schutzmechanismus liegt in der Umsetzung aller vier genannten Eigenschaften. Um diese Eigenschaften zu erfüllen, muss unter anderem die Unschärfe der biometrischen Daten einer Person (Intraklassen-Variabilität) berücksichtigt werden. Die Intraklassen-Variabilität ist auch der Grund, warum herkömmlich kryptographische Methoden nicht ohne weiteres zum Schutz der biometrischen Referenzdaten verwendet werden können. Eine kryptographische Hashfunktion bildet eine Menge *A* von Daten auf einer Menge *B* von Daten mit fester Länge ab. Im idealen Fall können keine inhaltlichen Informationen mehr von einem kryptographischen Hash auf das Original zurückgeführt werden, siehe hierzu auch [Buch10]. Es handelt sich demnach um eine so genannte Einwegfunktion. Des Weiteren hat ein kryptographischer Hash die Anforderung, dass bei geringer Änderung der Eingangsdaten eine große Änderung des Hashwertes vollzogen werden soll [Schm09]. Dies ist einer der Hauptgründe, warum kryptographische Hashfunktionen nicht zum Schutz biometrischer Referenzdaten eingesetzt werden können. Bei einer geringen Änderung eines erfassten biometrischen Merkmals einer Person soll bei biometrischen Verifikationssystemen die Änderung im Hash-Raum auch sehr gering sein, im idealen Fall sogar keine Veränderung auftreten. Um diesen besonderen Eigenschaften zu begegnen, wurden biometrische Hashfunktionen eingeführt. In Abschnitt 4.4 wird detaillierter auf die unterschiedlichen Anforderungen an einen kryptographischen und einen biometrischen Hash eingegangen.

Die in der Literatur vorgestellten biometrischen Schutzmechanismen können laut Jain et al. [JaNN08] grundsätzlich in zwei Kategorien unterteilt werden, zum einen in Merkmals Transformationen (Feature Transformation) und zum anderen in Biometrische Kryptosysteme (Biometric Cryptosystems). In Abbildung 2 werden diese beiden Kategorien inklusive weiterer Einteilungen gezeigt und nachfolgen erläutert.

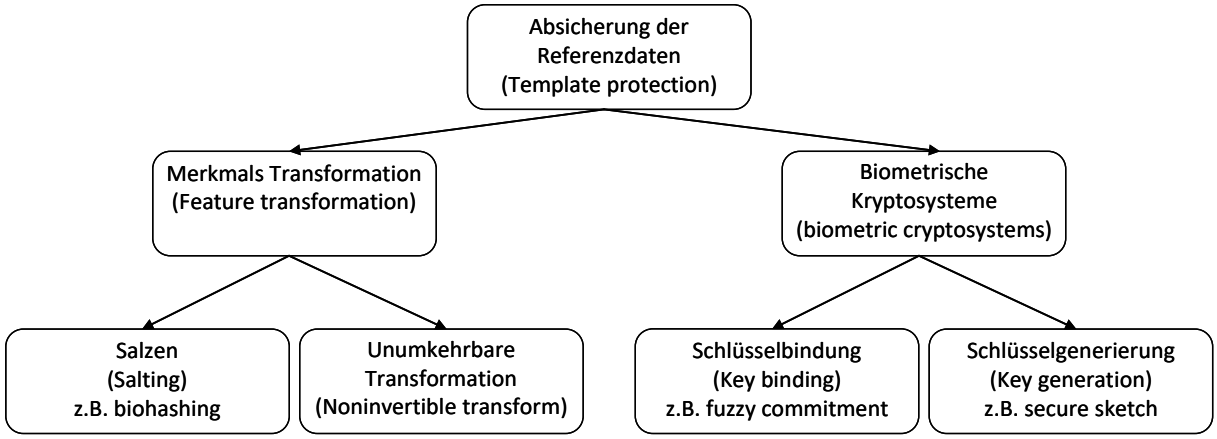


Abbildung 2 Einordnung verschiedener Template Protection Schemes nach [JaNN08]

Merkmals Transformation

Bei den Verfahren, welche Merkmalstransformationen einsetzen, transformiert eine Funktion *(F)* ein biometrisches Referenzdatum *(T)*, wobei nur dieses transformierte Datum *(F(T; K)) z.B.* in einer Datenbank gespeichert wird. Die Parameter für die Transformationsfunktion werden typischerweise von einem zufälligen Schlüssel *(K)* oder einem Passwort abgeleitet. Die gleiche Transformationsfunktion wird auch auf das aktuelle, zu verifizierende biometrische Datum *(Q)* angewendet *(F(Q; K))* und mit dem transformierten Referenzdatum *(F(T; K))* verglichen.

Je nach Eigenschaft der Transformationsfunktion *F*, kann die Merkmalstransformation in zwei weitere Kategorien unterteilt werden, dem so genannten Einstreuen bzw. Salzen (salting) und der unumkehrbaren Transformation (noninvertible transform), siehe Abbildung 2. Beim Verfahren des Einstreuens ist die Transformationsfunktion *(F)* umkehrbar, das heißt, ist ein potentieller Angreifer im Besitz des Schlüssels *(K)* und dem transformierten Template *(F(T; K)*, so kann das originale biometrische Referenzdatum wiederhergestellt werden bzw. eine sehr gute Annäherung davon. Somit basiert die Sicherheit des salting- Schutzmechanismus auf dem Schlüssel bzw. dem Passwort. Bei unumkehrbaren Transformationen ist, wie der Name schon andeutet, die Transformation nicht umkehrbar. Typischerweise wird eine Einwegfunktion auf das Referenzdatum angewendet, welches die Rückrechnung der Transformation sehr schwer macht, auch wenn der Schlüssel bekannt ist.

Biometrische Kryptosysteme

Biometrische Kryptosysteme wurden ursprünglich entwickelt, um zum einen kryptographische Schlüssel mit Hilfe von biometrischen Charakteristiken zu sichern und zum anderen um einen kryptographischen Schlüssel direkt von einer biometrischen Charakteristik abzuleiten. Allerdings können sie auch als Schutzmechanismen für biometrische Referenzdaten eingesetzt werden.

In biometrischen Kryptosystemen wird ein Teil des Referenzdatums öffentlich gespeichert, dieser öffentliche Teil wird auch als Helper-Data bezeichnet. Helper-Data gibt bzw. soll keine relevanten Informationen über das originale Referenzdatum preisgeben. Es wird beim Vergleichsprozess verwendet, um vom aktuell präsentierten biometrischen Datum einen kryptographischen Schlüssel zu extrahieren. Der Vergleich der biometrischen Daten wird indirekt durchgeführt, indem die Gültigkeit des extrahierten Schlüssels geprüft wird.

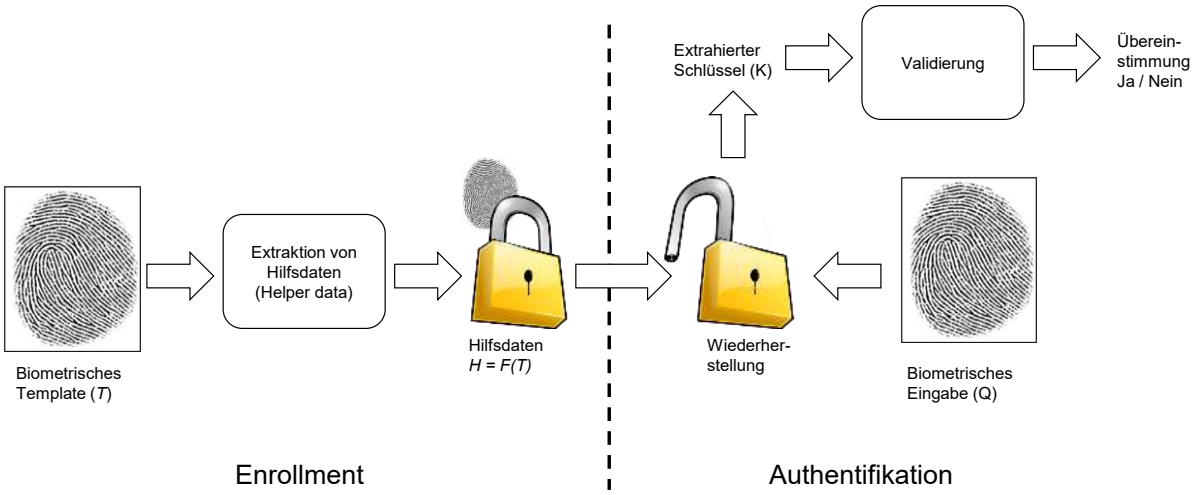


Abbildung 3 Authentifizierungsmechanismus mittels biometrischen Kryptosystem (Schlüsselgenerierend) nach [JaNN08]

Biometrische Kryptosysteme können wiederum in Schlüsselbindende- (key-binding) und Schlüsselgenerierende (key generation) Systeme unterteilt werden. Dies hängt von der Art und Weise ab, wie Helper-Data von den biometrischen Daten ermittelt bzw. abgeleitet wurde [JaNN08].
Wird Helper-Data generiert, indem ein Schlüssel am Referenzdatum gebunden wird, spricht man von einem Schlüsselbindenden System. In Abbildung 3 wird der Authentifizierungsmechanismus eines biometrischen Kryptosystems anhand des Schlüsselbindenden Verfahrens dargestellt. An dieser Stelle sei erwähnt, dass ein potentieller Angreifer mit Helper-Data alleine, den Schlüssel bzw. das biometrische Referenzdatum nicht ohne weiteres ermitteln kann. Bei einem Vergleich von aktuellen Daten und Referenzdaten in einem Schlüsselgebundenen System wird stets der Schlüssel mittels Helper-Data vom biometrischen Datum extrahiert.

Tabelle 1 Zusammenfassung aller hier vorgestellten biometrischen Schutzmechanismen: *T* repräsentiert das biometrische Template, *Q* steht für die Anfrage und *K* ist der verwendete Schlüssel zum Schutz des Templates. Beim Salzen (salting) und der unumkehrbaren Transformation repräsentiert *F* die Transformationsfunktion und *M* den Vergleicher (Matcher) im transformierten Bereich. In den biometrischen Kryptosystemen beschreibt *F* die Funktion zur Extraktion der Helper-Data und *M* die Fehlerkorrekturfunktion, welche die Rekonstruktion des Schlüssels *K* erlaubt. [JaNN08]

Methode	**Wie wird der Schutz des Referenzdatums gewährleistet?**	**Welche Teile werden gespeichert/hinterlegt?**	**Wie wird Intraklassen-Variabilität behandelt?**
Salzen (salting)	Geheimhaltung von *Schlüssel K*	Jedem zugänglich: Template *F(T;K)* Geheim: *Schlüssel K*	Quantisierung und Vergleich im transformierten Bereich *M(F(T;K),F(Q;K))*
Unumkehrbare Transformationen (noninvertible transform)	Unumkehrbarkeit der Transformationsfunktion *F*	Jedem zugänglich: transformiertes Template *F(T;K)*, Schlüssel *K*	Vergleich im transformierten Bereich *M(F(T;K),F(Q;K))*
Schlüsselbindendes biometrisches Kryptosystem	Sicherheitsniveau abhängig von den Informationen, welche die Helper-Data *H* preisgibt	Jedem zugänglich: Helper-Data *H = F(T;K)*	Fehlerkorrektur und Personenspezifische Quantisierung *K = M(F(T;K),Q)*
Schlüsselgenerierendes biometrisches Kryptosystem	Sicherheitsniveau abhängig von den Informationen, welche die Helper-Data *H* preisgibt	Jedem zugänglich: Helper-Data *H = F(T)*	Fehlerkorrektur und Personenspezifische Quantisierung *K = M(F(T),Q)*

Wird Helper-Data ausschließlich vom biometrischen Referenzdatum gewonnen und der kryptographische Schlüssel direkt von Helper-Data und den aktuell präsentierten biometrischen Daten generiert, spricht man von einem Schlüsselgenerierenden Schutzverfahren. Bei einem Schlüsselgenerierenden Schutzverfahren wird ein Schlüssel auf Basis der Helper-Data und dem Referenzdatum erzeugt, wohingegen bei einem Schlüsselbindenden Verfahren ein existierender Schlüssel am Referenzdatum gebunden wird, um Helper-Data zu erzeugen. Demnach besteht die Herausforderung bei einem Schlüsselgenerierenden Schutzverfahren darin geeignete Helper-Data zu extrahieren.
In Tabelle 1 werden alle oben beschriebenen biometrischen Schutzmechanismen zusammengefasst dargestellt. Eine weiterführende Beschreibung der hier kurz vorgestellten Schutzmechanismen wird in [JaNN08] gegeben. Des Weiteren geben Cavoukian und Stoianov in [CaSt07] einen Überblick über Anwendung von biometrischen Kryptosystemen und deren positive Auswirkungen unter anderem auf die Verifikation von Identitäten und den Schutz der Privatsphäre. Die in Tabelle 1 aufgelisteten Methoden und deren Vor- und Nachteile sollen hier kurz beschrieben werden. Insbesondere werden die "Biometrischen Kryptosysteme" und "Merkmals Transformationen" gegenübergestellt.

Der Vorteil der "Merkmals Transformation" liegt u.a. darin, dass von einer Person mehrere biometrische Templates zu einer biometrischen Modalität generiert werden können. Das wird mittels personenspezifischer Schlüssel (Salting) oder anwendungsspezifische Transformationsfunktion (Unumkehrbare Transformation) realisiert. Zusätzlich können bei Bedarf kompromittierte Templates zurückgezogen werden (revocable). Dies wird ebenfalls durch personenspezifische Schlüssel (Salting) als auch personenspezifischer Transformationsfunktion (Unumkehrbare Transformation) gewährleistet. Das Zurückrechnen eines originalen biometrischen Referenzdatums anhand der Templates ist schwierig (Unumkehrbarkeit), jedoch beim Salting Verfahren ist die Rückrechnung möglich, sollten der Schlüssel und das Template bekannt sein. Siehe weiterführend dazu auch [JaNN08].
Da der Vergleich bei Transformationen im transformierten Raum/Bereich stattfindet, muss nach Jain et al. in [JaNN08] z.B. der Salting Mechanismus derart gestaltet sein, dass die Verifikationsperformanz nicht negativ beeinträchtigt wird. Ähnlich verhält es sich bei der Gestaltung der Transformationsfunktion (Unumkehrbarkeit), die ein Kompromiss zwischen der Unterscheidbarkeit und der Unumkehrbarkeit nach sich zieht. Die Unterscheidbarkeit beschreibt das Verhalten biometrischer Merkmale einer Person, welche sich im normalen als auch im transformierten Bereich nur gering unterscheiden, wohingegen biometrische Merkmale anderer Personen sich stark von diesen unterscheiden sollen. Das Entwerfen einer Transformationsfunktion die beide Eigenschaften berücksichtigt, ist schwierig. Die Optimierung einer Ausprägung führt in der Regel zur Verschlechterung der anderen [JaNN08].

Vorteilhaft bei biometrischen Kryptosystemen ist die Kompensation der Intraklassen-Variabilität durch personenbezogene Quantifizierung und Fehlerkorrekturmaßnahmen (Schlüsselbindendes Kryptosystem). Weiterhin sind generierte Schlüssel, die auf biometrischen Daten basieren, recht interessant (Schlüsselgenerierendes System). Dies ist insbesondere dann der Fall, wenn sie innerhalb kryptografischer Anwendungen eingesetzt werden sollen.

Jedoch ist es schwierig einen Schlüssel zu generieren, der stabil ist und einen hohen Informationsgehalt (Entropie) bietet. Des Weiteren wirkt sich auf Schlüsselbindende Kryptosysteme der Vergleich negativ aus. Der Vergleich muss mittels eines Fehlerkorrekturverfahrens durchgeführt werden, dies kann zu einer Beeinträchtigung der Vergleichsgenauigkeit des Systems führen. Auch ist das Zurückziehen kompromittierter Templates und das Erstellen mehrerer Templates pro Person nicht vorgesehen. Jedoch können beide Punkte durch das Einbinden eines Salting Verfahren realisiert werden. Ein weiterer Nachteil des Schlüsselgebundenen biometrischen Kryptosystems besteht in den Hilfsdaten (Helper Data). Diese müssen sorgfältig gewählt werden, da sie Informationen des originalen biometrischen Referenzdatums enthalten, um die Intraklassen-Variabilität zu kompensieren und dürfen gleichzeitig keine signifikanten biometrischen Informationen der Nutzer offenbaren [JaNN08].

Zusätzlich zu den oben genannten Schutzmechanismen bieten Methoden der *homomorphen Verschlüsselung* Techniken, die den Schutz biometrischer Daten gewährleisten können.

Homomorphe Verschlüsselung
Eine weitere Methode biometrische Referenzdaten zu schützen, ist die Verwendung der sogenannten „Homomorphen Verschlüsselung“. Dieses Verfahren erlaubt es, Operationen auf verschlüsselte Daten zu tätigen ohne diese dabei zu entschlüsseln [BrGV12] (siehe Abschnitt 4.3.3). Dementsprechend können biometrische Referenzdaten beispielsweise im verschlüsselten Zustand verglichen werden. So kann die Sicherheit biometrischer Referenzdaten erhöht werden, da diese stets im verschlüsselten Zustand im System vorliegen. Derzeit sind Verfahren der homomorphen Verschlüsselung relativ rechenintensiv und demnach zeitaufwändig. So ist beispielsweise die zu verarbeitenden Datenmenge des von Troncoso-Pastoriza et al. in [TrGP13] präsentierten biometrischen Verifikationsverfahrens, basierend auf der Modalität Gesicht, für einen Vergleich ca. 390 MiByte groß und die Berechnungszeit auf dem Server liegt zwischen 60 und 120 Sekunden.
Für kleine Datenmengen (wenige Bytes – KiBytes) ist der Rechenaufwand jedoch überschaubar. So konnte wenige Jahre später Boddeti in [Bodd18], ebenfalls für ein auf das Gesicht basierenden optimierten biometrischen Verifikationsverfahren die Datenmenge auf 66 KiByte reduzieren und entsprechend die Vergleichsberechnung auf 0,01 Sekunden minimieren. Diese Optimierung hat aber zur Folge, dass die Erkennungsperformanz des Systems leicht gesunken ist. Sind die zu schützenden Referenzdaten eines biometrischen Systems wenige KiBytes groß ist ein praktikabler Einsatz von homomorpher Verschlüsselung möglich, siehe auch [YaSK14].
In den letzten Jahren haben weitere Autoren homomorphe Verschlüsselungsmechanismen eingesetzt um biometrische Daten und dazugehörige Verifikationsprozess zu schützen. Nachfolgend werden beispielhaft einige Arbeiten kurz vorgestellt.

Wie bereits erwähnt, zeigt Boddeti ([Bodd18]) in einem Verifikationsverfahren basierend auf der Modalität Gesicht, dass ein angepasster Verifikationsalgorithmus (Modalität Gesicht) in Kombination mit einer homomorphen Verschlüsselung durchaus für den praktischen Gebrauch, akzeptable Verarbeitungszeiten liefert. So konnte Boddeti die Referenzdatengröße auf 16 KiByte und die Zeit für die Vergleichsberechnung entsprechend auf

unter 0,01 Sekunden senken. Diese Optimierung hat aber zur Folge, dass die Erkennungsperformanz des Systems leicht gesunken ist.
Cheon et al. haben in [CCKL16] ein auf der Modalität Iris basierendes Verfahren vorgestellt, welches mit einer optimierten "somewhat" homomorphen Verschlüsselung (siehe Abschnitt 4.3.3) die Referenzdaten schützt. Dabei verwenden sie 2400 Bit große biometrische Referenzdaten und erreichen somit eine Verarbeitungszeit für einen Vergleich von ca. 0,5 Sekunden.
Ein multimodales biometrisches Verifikationsverfahren haben Gomez-Barrero et al. in [GMG+17] vorgestellt. Sie verwenden das von Paillier in [Pail99] entwickelte partielle homomorphe Verschlüsselungssystem und haben es hierfür modifiziert. Es kann mit beliebigen biometrischen Modalitäten betrieben werden. Evaluiert wurde das multimodale System mit den biometrischen Modalitäten Handschrift und Fingerabdruck. Mit der vorgestellten Technik konnten Gomez-Barrero et al. eine Referenzdatengröße von ca. 200 KiByte erzeugen und eine Erkennungsperformanz (Equal Error Rate) von 0.12% (siehe Abschnitt 4.1.4 Biometrische Fehlerraten).
Ferner haben Failla in [Fail11] und Karabat et al. in [KKES14] Möglichkeiten vorgestellt, biometrische Referenzdaten mittels homomorpher Verschlüsselung zu prozessieren. Sie beschreiben praktische Einlern-(Enrollment) und Verifikationsprotokolle zur Verarbeitung der biometrischen Daten.

2.1.2 Handschriften Biometrie

In diesem Abschnitt wird ein Überblick über bestehende Verifikationsverfahren auf Basis der Handschrift gegeben. Es werden nur ausgesuchte Verfahren betrachtet, die für die Verifikation einer Person auf Basis der Handschrift geeignet sind. Sie sollen stellvertretend für die Vielzahl von Verifikationsverfahren stehen, die bis dato vorgestellt wurden.

Im Allgemeinen können handschriftenbasierte Verifikationsverfahren in zwei Kategorien unterteilt werden, statische und dynamische Verfahren. Bei einem statischen Verfahren werden biometrische Merkmale auf Basis einer z.B. auf einem Papier geschriebenen Handschrift extrahiert. In der Regel wird ein Schriftzug digitalisiert und in einem computergestützten System weiterverarbeitet (Signalaufbereitung, Merkmalsextraktion, etc.).
Bei einem dynamischen handschriftenbasierten Verifikationssystem werden die Handschriftendaten mit Hilfe eines Sensors zur Erfassung des Schreibvorgangs (z.B. Signaturtablett, Spezialstifte etc.) direkt aufgezeichnet und können sogleich am Computer verarbeitet werden. Da die Handschriftendaten während des Schreibvorgangs aufgezeichnet werden, spricht man hierbei auch von einem Online Verfahren, wohingegen das statische Verfahren auch als Offline Verfahren bezeichnet wird. Da in dieser Arbeit der Fokus primär auf dynamische handschriftenbasierte Verifikationsverfahren gerichtet ist, werden nur wenige gängige Methoden der Offline Verfahren im Überblick vorgestellt.

Statische (Offline) Verfahren
In den letzten Jahren wurde eine Vielzahl von verschiedenen Verifikationsmethoden auf Basis der statischen Handschrift vorgestellt. Chavan et al. [ChVi18] als auch Pal et al. [PaBP11] stellen in ihren Arbeiten die bekanntesten Offline-Methoden zur automatischen Handschriften- bzw. Unterschriftenerkennung vor und vergleichen die Erkennungsperformanz von ausgewählten Verfahren. Einige dort vorgestellte Verfahren sollen hier kurz erläutert werden.

Justino et al. stellen in [JuBS01] ein Offline Verifikationsverfahren basierend auf der Analyse mittels Hidden Markov Model (HMM) vor. Ziel hierbei ist es, mit Hilfe des HMMs zufällige, einfache und geübte Handschriftenfälschungen zu detektieren. Die Handschriftendaten werden hierfür auf ein computergeneriertes Gitternetz gelegt und für jede einzelne Zelle des Gitters werden jeweils drei Merkmale extrahiert: Punktdichte, Punktverteilung und Schriftneigung. Weitere Verfahren, welche Hidden Markov Modelle zur Verifikation nutzen, sind unter anderem in [FFFG07], [BaGs09], [McTr08], [BaAS09], [DFES17] und [FCD+18] beschrieben.
In [OzSK05] stellen Ozgunduz et al. eine Methode zur Detektion von zufälligen und geübten Angriffen mittels Support Vector Machinen (SVM) vor. Dabei werden globale geometrische Merkmale, Richtungsmerkmale und Gitternetzmerkmale von einer Unterschrift extrahiert. In ihrer Arbeit vergleichen sie die Klassifikationsperformanz der Support Vector Machine mit einem künstlichen neuronalen Netzwerk als Klassifikator. In diesen Beitrag fallen die Erkennungsraten des mittels neuronalen Netzwerks klassifizierenden Verfahrens schlechter aus, als bei dem SVM basierten Verfahren.
Ein weiteres SVM basiertes Verfahren wird von Kisku et al. in [KiGS09] präsentiert. Hierbei werden verschiedene Vergleichsverfahren mit Hilfe der SVM miteinander kombiniert/ fusioniert. Um die Erkennungsperformanz zu steigern, werden einige Vorverarbeitungsschritte ausgeführt. Dabei werden Operationen der Bildverarbeitung zur Verbesserung der Bildqualität auf dem Rohdatensatz (Handschrift/Unterschrift) durchgeführt. Das von Kuski et al. vorgestellte System verwendet drei verschiedene statistische Ähnlichkeitsmaße (Euklidischer Abstand, Mahalanobis-Distanz und Normalverteilungsregel), welche auf die extrahierten Merkmalsdatensätze separat angewendet werden. Anschließend wird mittels SVM ein Vergleichswert auf Basis der drei Ähnlichkeitsmaße ermittelt. Bei der Merkmalsextraktion werden globale Merkmale und lokale geometrische Gitternetzmerkmale bestimmt. Mit Hilfe der SVM konnte eine Fusion der verschiedenen Vergleichsmaße durchgeführt werden, welche zu einer signifikanten Steigerung der Erkennungsperformanz gegenüber der einzelnen Ähnlichkeitsmaße führte. Verfahren, welche einen ähnlichen Ansatz verfolgen, sind in [JaPR04], [MMJP09] und [LWWZ05] beschrieben.
Zusätzlich zu den bereits erwähnten Verfahren wurden in den letzten Jahren vermehrt Methoden des maschinellen Lernens auf Basis neuronaler Netze (NN) verwendet, um Handschriftenverifikationsverfahren zu entwickeln/optimieren [DFI+19]. So stellte beispielsweise Kumar et al. in [KuSC2012] ein Verfahren auf Basis „einfacher“ neuronaler Netze vor. Bei diesen Verfahren werden von einer binären Handschriftenrepräsentation (schwarz/weiß Bild) Umgebungsmerkmale der Handschrift ermittelt. Somit sollen die Eigenschaften der Handschrift, wie Form und Textur, berücksichtigt werden. Die Klassifikation wurde mittels NN durchgeführt. Die erzielten Verifikationsergebnisse waren vergleichbar bzw. leicht besser als die eines SVM basierten Klassifikators. Neben der Arbeit von Kumar et al., wurden weitere Arbeiten vorgestellt, welche ebenfalls neuronale Netzwerke bzw. varianten dieser als Klassifikator in handschriftenbasierte Verifikationssysteme einsetzen. So wurden u.a. Verifikationssysteme auf Basis von Convolutional Neural Networks (CNNs) in [AlSB16], [DDT+17], [HaSO16], [HaSO17], [KhMT12], Deep Neural Networks (DNN) in [RaYM16], Deep Multitask Metric Learning (DMML) in [SoAF16] und Deep Convolutional Generative Adversarial Network (DCGAN) in [ZhLC16] vorgestellt. Die Aufzählung ist nicht abschließend, zeigt jedoch bereits, dass in den letzten Jahren viel in diesem Bereich mit neuronalen Netzen geforscht wurde.

Weitere Verfahren auf Basis der statischen Handschrift werden in der Arbeit von Pal et al. in [PaBP11] und Diaz et al. in [DFI+19] erwähnt. Im nächsten Absatz werden einige Verfahren auf Basis der dynamischen Handschrift erläutert.

Dynamische (Online) Verfahren

Erste Verifikationssysteme auf Basis der dynamischen Handschriften wurden bereits vor mehreren Jahrzenten vorgestellt. Plamondon und Lorette stellen in [PlLo89] bis dato alle relevanten automatischen Offline und Online Verifikationsverfahren vor. In einem circa fünf Jahre später veröffentlichten Beitrag [PlLe04] stellten Leclerc und Plamondon bis dato neu erschienene Verfahren vor. Hier werden mitunter Verfahren präsentiert, welche Neuronale Netze zur Klassifizierung von biometrischen Daten nutzen. In einer relativ umfassenden Arbeit [DFI+19] fassen Diaz et al. etwas später verschiedene Handschriftenverfahren zusammen. Hier werden neben den Verfahren zur Verifikation von westlichen Handschriftenzeichen auch solche aufgelistet, die chinesische, japanische, indische und persische Schriftzeichen verwenden. Gupta et al. stellen in [GuMc97] ebenfalls veröffentlichte Verifikationsverfahren vor. Die in diesen Beiträgen vorgestellten Verfahren wurden in zwei wesentliche Kategorien eingeteilt, zum einen in funktionsbasierte und zum anderen in parameterbasierte Verfahren. Bei einem funktionsbasierten Verfahren werden die aufgezeichneten zeitabhängigen Schreibsignale (x(t), y(t), p(t), azi(t), alt(t)) jeweils einzeln als mathematische Zeitfunktion betrachtet und sind direkt mit einem Merkmalssatz verbunden. Der Vorteil dieser Methode liegt in der relativ leichten Bestimmung (Ermittlung) der Merkmale, welche jedoch zum Teil sehr rechenintensiv ausfallen können. Die Herausforderung der funktionsbasierten Verfahren liegt beim Vergleich (Matching) zweier Datensätze. Bei einem parameterbasierten Ansatz werden *m* verschiedene (statistische) Parameter vom Schreibsignal berechnet, welche die Merkmale repräsentieren. Der Vorteil des Verfahrens liegt bei dem sehr schnellen Vergleich zweier Datensätze (z.B. Merkmalsvektoren). Jedoch liegt hier die Herausforderung bei der Selektion aussagekräftiger Merkmale, welche eine Klassifizierung und somit die Erkennungsperformanz begünstigen. Gupta erläutert diese beiden Kategorien in [Gupt06] detailliert und zeigt darüber hinaus Ergebnisse bezüglich der Erkennungsperformanz.

Nachfolgend werden einige ausgewählte Online Handschriftenverifikationsverfahren der beiden Kategorien (funktions- und parameterbasierend) kurz vorgestellt.

In [DeBr85] stellt De Bruyne ein Verfahren basierend auf 18 globalen Merkmalen vor, welche sechs dynamische Merkmale und weitere statische Merkmale beinhaltet. Die dynamischen Merkmale beinhalten die gesamte Schreibdauer, die Anzahl der Stiftabsetzer, die Schreibzeit und die Zeit, in der der Stift nicht aufgesetzt war. Hierbei setzt sich die gesamte Schreibdauer aus der Addition der Schreibzeit und der Zeit in der der Stift abgesetzt war zusammen. Weiterhin wird die maximale Geschwindigkeit berechnet und zusätzlich zu welcher Zeit diese aufgetreten ist. Die statischen Merkmale setzten sich zusammen aus dem Flächeninhalt, den Proportionen, der Standartabweichung von x und y Werten und das Verhältnis der gesamten Verschiebungen in x und y Richtung. Innerhalb der Evaluation wurden die Referenzdaten auf Basis von zehn Handschriftendaten berechnet. Die Referenzdaten wurden anschließend mit den Testdaten und einigen Angriffsdaten verglichen. Eine Falschrückweisungsrate von 3% und eine Falschakzeptanzrate von 2% konnte in den Tests, welche aus lediglich elf Personen generiert wurde, erreicht werden.

Nelson et al. stellen in [NeTH94] ein Verfahren basierend auf 25 Merkmalen vor. Davon sind zwei dieser Merkmale zeitabhängige, sechs geschwindigkeits- sowie beschleunigungsabhängige und vier formbezogene Merkmale. Weiterhin werden verschiedene Dichteverteilungen bezüglich des Pfades ermittelt. Bei Ihren Tests verwenden Sie drei verschiedene Methoden, um den Abstand der Referenz- und Testdaten zu ermitteln, diese sind die Euklidische Abstandsmessung, Mahalanobis Abstandsmessung sowie die quadratische Diskriminanzanalyse. Des Weiteren stellen Sie eine einfache Merkmalsselektion vor in der die Standartabweichung zum Mittelwert für jedes einzelne Merkmal berechnet und auf dieser Basis ein Merkmalsranking durchgeführt wird. In der Evaluation werden verschiedene Testreihen der besten 8, 10, 12 und 14 der insgesamt 25 Merkmale durchgeführt. Die Ergebnisse der Testreihen sind ähnlich, wobei die Testreihen mit den besten acht bzw. zehn Merkmalen eine Falschakzeptanzrate (FAR) von nahezu null aufweisen. Unter Einsatz des euklidischen Abstandsmaßes und der Verwendung der besten zehn Merkmale konnte eine FRR von 0,5 % und einer Falschakzeptanzrate (FAR) von 14% erreicht werden.

Kholmatov und Yanikoglu stellen in [KhYa05] eine auf Dynamic Time Warping (DTW) arbeitende Methode vor. Der unter anderem in der Spracherkennung eingesetzte Algorithmus erstellt auf Basis der Referenzdaten und der aktuell präsentierten Testdaten einen dreidimensionalen Merkmalsvektor. Hierbei werden in der Einlernphase (Enrollment) alle Referenzdaten einer Person paarweise miteinander verglichen. Mit Hilfe des DTW Algorithmus wird die Unähnlichkeit der beiden Handschriftendaten ermittelt. Das Handschriftensample mit dem geringsten Abstand zu den übrigen gilt als Referenzsample. In der Trainingsphase werden Handschriftensamples einer Person mit den Referenzdaten der selbigen vergleichen, dabei wird die Unähnlichkeit zu allen Referenzdaten der Person paarweise bestimmt. Die Abstandswerte zu den am weitesten entfernten, am nächsten liegenden und zu dem Referenzdatensatz werden in einem dreidimensionalen Merkmalsvektor gespeichert. Für jedes Testsample wird ein Merkmalsvektor berechnet und gespeichert. Gleiches gilt für Angriffs bzw. Fälschungsdaten. Mit Hilfe von verschiedenen Klassifikatoren: Bayes, Support Vektor Maschine (SVM) und eine Hauptkomponentenanalyse (PCA), werden die echten (authentischen) von den unechten Handschriftendaten "getrennt". Innerhalb der präsentierten Evaluationsdaten wurde mit Hilfe der PCA eine FRR von 1,64% und eine FAR von 1,28% erreicht. An dieser Stelle sei erwähnt, dass dieses Verifikationsverfahren bei dem "Ersten Internationalen Handschriften Verifikationswettbewerb (First International Signature Verification Competition)" eingereicht wurde. Dort hat dieses Verfahren den ersten Platz in den zwei Teildisziplinen des Wettbewerbs gewonnen. Für die genauen Testbedingungen und weitere Ergebnisse anderer Verfahren sei auf den Beitrag von Yeung et al. in [YCX+04] verwiesen.

Jain et al. stellen in [JaGC02] ein Verfahren vor, welches dynamische als auch statische Merkmale einer Handschrift in den Verifikationsprozess mit einfließen lässt. Innerhalb einer Verifikation werden alle Merkmale von einem Testdatensatz extrahiert und mit dem Referenzdatum der behaupteten Identität verglichen. Hierbei wird der Testdatensatz mit allen hinterlegten Referenzdaten der behaupteten Identität verglichen. Mit Hilfe eines String-Matching-Algorithmus (Dynamic Time Warping) wird die Ähnlichkeit bzw. Unähnlichkeit zweier Strings (Handschriftendaten) ermittelt. Jain et al. testen zwei verschiedene Schwellenwertverfahren innerhalb des Entscheidungsprozesses. Zum einen wird ein

globaler Schwellenwert, welcher für alle registrierten Personen einer Datenbank gilt, getestet und zum anderen ein individueller Schwellenwert. Evaluationsergebnisse zeigen, dass die mit Hilfe von individuellen Schwellenwerten für jede Person eine FRR von 2,8% und eine FAR von 1,6% erzielt worden ist. Mit einem globalen Schwellenwert konnte in den Tests eine FAR von 3,3% bzw. eine FRR von 2,7 % erreicht werden.

Fierrez-Aguilar et al. beschreiben in [FNL+05] ein auf 100 Merkmalen basierendes Verifikationsverfahren. Bei diesem Verfahren werden zwei verschiedene Verifikationssysteme auf Matcher bzw. Entscheidungsebene miteinander fusioniert. Eines der Systeme extrahiert globale Merkmale mittels Parzen-Fenster-Methode, wohingegen das andere System lokale Informationen als zeitabhängige Funktion von dynamischen Eigenschaften unter Zuhilfenahme des Hidden Markov Models ermittelt.
Die Fusion der beiden Systeme geschieht auf der Entscheidungsebene, wo die Ähnlichkeitsmaße beider Systeme miteinander fusioniert werden. Weiterhin werden durch eine Merkmalsselektion die 40 prägnantesten Merkmale auf Basis der verwendeten Testdatenbank (MCYT) ermittelt. Hierbei wurde für jedes Merkmal der Mahalanobis Abstand zwischen den Durchschnittswert der Trainingsdaten einer Person und aller anderen Personen ermittelt. Durch Fusion der beiden Systeme konnte unter Verwendung der 40 besten Merkmale eine Gleichfehlerrate (FAR = FRR) von 1,7 % ermittelt werden, wobei die FAR mittels geübten Fälschungsdaten bestimmt wurde.

Elahen et al. haben in [ElMo09] ein Verifikationssystem basierend auf fünf Merkmalen und einem Adaptiven Neuro-Fuzzy-Inferenzsystem (ANFIS) als Klassifikator vorgestellt. Eines der fünf Merkmale wird mittels Fraktaler Dimension, im speziellen "Boxcounting-Methode", ermittelt. Merkmal zwei ermittelt die gesamte Schreibdauer, Merkmal drei und vier beschreiben die positiven Geschwindigkeiten im X respektive im Y Signal und Merkmal fünf repräsentiert den normalisierten Druck an der Stiftspitze beim Schreibvorgang. Letzteres wird in der ersten Phase des Systems verwendet, um eine Vorauswahl bezüglich der Echtheit des Handschriftensignals einer Person zu treffen. In der zweiten und letzten Phase des Systems wird ein adaptive Neuro-Fuzzy-Inferenzsystem (ANFIS) verwendet, um eine Entscheidung zu treffen. Das ANFIS besteht aus fünf Ebenen und ist eine Kombination der Prinzipien aus der Welt der Neuronalen Netze und der Fuzzylogik. Die vier verbleibenden extrahierten Merkmale dienen hierbei als Eingabewerte für das ANFIS. Mit Hilfe von Trainingsdaten wird das System angelernt und ein Schwellenwert zur Trennung der echten und unechten Handschriftendaten bestimmt. Bei ihren Test verwenden Elahen et al. zwei verschiedene Testdatenbanken, eine freiverfügbare SUBCORPUS-100-MCYT ([OFS+03]) Datenbank und eine selbsterstellte "persische" Datenbank und vergleichen ihre Ergebnisse mit Arbeiten, welche die gleiche Datenbasis verwenden z.B. [FaFM08]. Bei geübten Fälschungen ermittelten sie einen Gleichfehlerrate (Equal Error Rate - EER) von 7,16% (MCYT DB) bzw. 3,95 % (erstellte DB) und bei zufälligen Fälschungen 3,3% (MCYT) und 2,29% (erstellte DB). Diese Ergebnisse sind in Anbetracht der geringen Anzahl von verwendeten Merkmalen beachtlich gut.

Verfahren auf Basis von sogenannten Datenhandschuhen werden unter anderem in [KaSE08], [SaKB09] und [SaKB09a] beschrieben. Hierbei werden die Daten mit einem Handschuh, welcher mit verschiedenen Sensoren bestückt ist, aufgenommen. Anhand dieser Handschuhdaten, die Bewegungen der Hand in einem virtuellen Raum abbilden

soll, werden Handschriftendaten ermittelt. Kamel konnte in [KaSE08] somit eine Gleichfehlerrate von 2,37% erreichen. Diese Verfahren werden in dieser Arbeit nicht weiter betrachtet, sollten aber auf Grund der Vollständigkeit hier erwähnt werden.

In den letzten Jahren wurde, ähnlich wie bei den statischen Verifikationsverfahren, ebenfalls Forschungsarbeit auf dem Gebiet der Neuronalen Netzte als Klassifikator für handschriftenbasierte Verifikationssysteme aufgewendet [DFI+19].

So haben beispielsweise Tolosana et al. in [TVFO18] eine neue Methode vorgestellt, bei der eine „Siamese Network" – Architektur als Klassifikator eingesetzt wird. Bei einer „Siames Network" oder auch „Twin neural Network" genannten Architektur, handelt es sich um zwei parallellaufende Neuronale Netze mit gleicher Wichtung, die bei unterschiedlichen Eingangsdaten einen vergleichbaren Ausgabevektor erzeugen. Bei der Arbeit von Tolosana et al. sind die Eingabedaten entsprechend die Merkmalsvektoren zweier unterschiedlicher Handschriftensignale. Am Ende der Klassifikation steht ein Wert, welcher den Unterschied der beiden Merkmalsvektoren repräsentiert. Je größer der Ausgabewert, desto mehr unterscheiden sich die beiden Merkmalsvektoren bzw. die ursprünglichen Handschriftensignale. Ziel von Tolosana et al. in ihrer Arbeit ist es u.a., mit Hilfe des Neuronalen Netzes den Unterschied der beiden Eingangsvektoren zu erlernen. Vorteil dieser Architektur ist die relativ geringe Menge an benötigten Trainingsdaten für das Neuronale Netz. In ihren experimentellen Versuchen verwendeten sie die BiosecurID Datenbank [FGO+10], wobei vier Handschriftensignale einer Person für das Anlernen (Training) des NN und zwölf weitere Handschriftensignal für die Verifikationsversuche eingesetzt werden. Zusätzlich verwenden sie zwölf, für die jeweilige Person angefertigte, Angriffsdaten um professionelle Nachahmungsangriffe zu simulieren. Die experimentellen Ergebnisse haben bessere Verifikationsergebnisse erzielt, als herkömmliche Klassifikationsverfahren. So konnte unter Verwendung der qualifizierten Angriffsdaten eine EER von 5,5 % erzielt werden, wohingegen eine „klassische" DTW (Dynamic Time Warping) Methode bei gleichen Testdaten eine EER von 7,75 % erreichte.
In [AhBa17] und [Lain09] werden ebenfalls Neuronal Netze als Klassifikator eingesetzt, wobei hier Rückgekoppelte Neuronale Netze (RNN) eingesetzt werden. In [YaGh17] stellen Yahyatabar et al. einen Klassifikator vor, der auf einem Convolutional Neural Networks (CNNs) basiert.

Es existieren noch viele weitere Verifikationsverfahren auf Basis der dynamischen Handschrift die hier jedoch nicht weiter erwähnt werden sollen. Das für diese Arbeit relevante Verfahren von Vielhauer [Viel06] wird ausführlich in Abschnitt 4.2 beschrieben.

Werden im Bereich der Biometrie neue Verifikationsverfahren vorgestellt, präsentieren die Autoren in der Regel Daten bezüglich der Erkennungsperformanz des neuen Systems. Hierfür verwenden sie biometrische Testdaten aus öffentlich verfügbaren Datenbanken und/oder sammeln hierfür explizit biometrische Daten von Probanden. Die Datenbanken beinhalten, in einem handschriftenbasierten System, in der Regel authentische und nicht authentische Handschriftendaten. Authentische Daten repräsentieren die biometrischen Daten von Individuen, die in einer Datenbank als Referenzdaten hinterlegt sind und solche die von diesen Individuen zum Zwecke der Verifikation präsentiert werden. Nicht

authentische Daten repräsentieren hier gefälschte Handschriftendaten von potentiellen Angreifern bzw. Daten anderer Personen.
Somit betrachten die meisten Autoren nur Angriffe auf das System, welche durch den Angriffspunkt 1 (siehe Abbildung 1) repräsentiert werden. Bei der Vorstellung einer neuen Methode ist das durchaus legitim, jedoch werden oftmals keine weiteren Sicherheitsrisiken neuer Verifikationsverfahren betrachtet. Dies erschwert teilweise einen direkten Vergleich der Testergebnisse und somit auch den Vergleich verschiedener Verifikationsalgorithmen bzgl. ihrer Sicherheit.

2.2 Allgemeine Schwachstellen/Angriffspunkte biometrischer Systeme

In diesem Abschnitt werden allgemeine Schwachstellen und potentielle Angriffspunkte von biometrischen Systemen beschrieben. Aus verschiedenen ausgewählten Beiträgen wird eine Übersicht über diverse jeweils dort vorgestellte Angriffsmodelle gegeben.
In Abbildung 1 wurden bereits acht mögliche Angriffspunkte von Ratha et al. gezeigt. Wayman stellt ein etwas detaillierteres Modell eines biometrischen Verifikationssystems in [Waym1999] vor. Es ermöglicht eine Makro- und Mikrosichtweise auf ein biometrisches Verifikationssystem. Das vorgeschlagene Modell besteht aus fünf Subsystemen mit zum Teil mehreren Prozessen, die wiederum potentielle Angriffspunkte darstellen können, siehe Abbildung 4. *Datenerhebung* beinhaltet die Aufnahme der biometrischen Modalität am Sensor. *Datenübertragung* repräsentiert den Kommunikationskanal, welcher eine Komprimierung bzw. Dekomprimierung enthalten kann. *Signalverarbeitung* umfasst die Merkmalsextraktion, Qualitätskontrolle und den Mustervergleich. *Speicherung* beschreibt die Art der Speicherung der biometrischen Referenzdaten, z.B. Datenbank oder Smartcard. Und im Subsystem *Entscheidung* wird ein boolescher Ausdruck (Ja/Nein) generiert, der bestimmt, ob eine Verifikation erfolgreich abgelaufen ist oder nicht. Alle acht Angriffspunkte aus Abbildung 1 können auf dieses Modell portiert werden mit dem Vorteil, dass die Angriffspunkte detaillierter beschrieben werden können. Somit kann eine genauere Einordnung einer ermittelten bzw. bekannten Schwachstelle eines biometrischen Systems exakter beschrieben und analysiert werden. Des Weiteren kann die potentielle Gefahr, welche von einem Angriffsverfahren ausgehen kann, besser abgeschätzt werden.

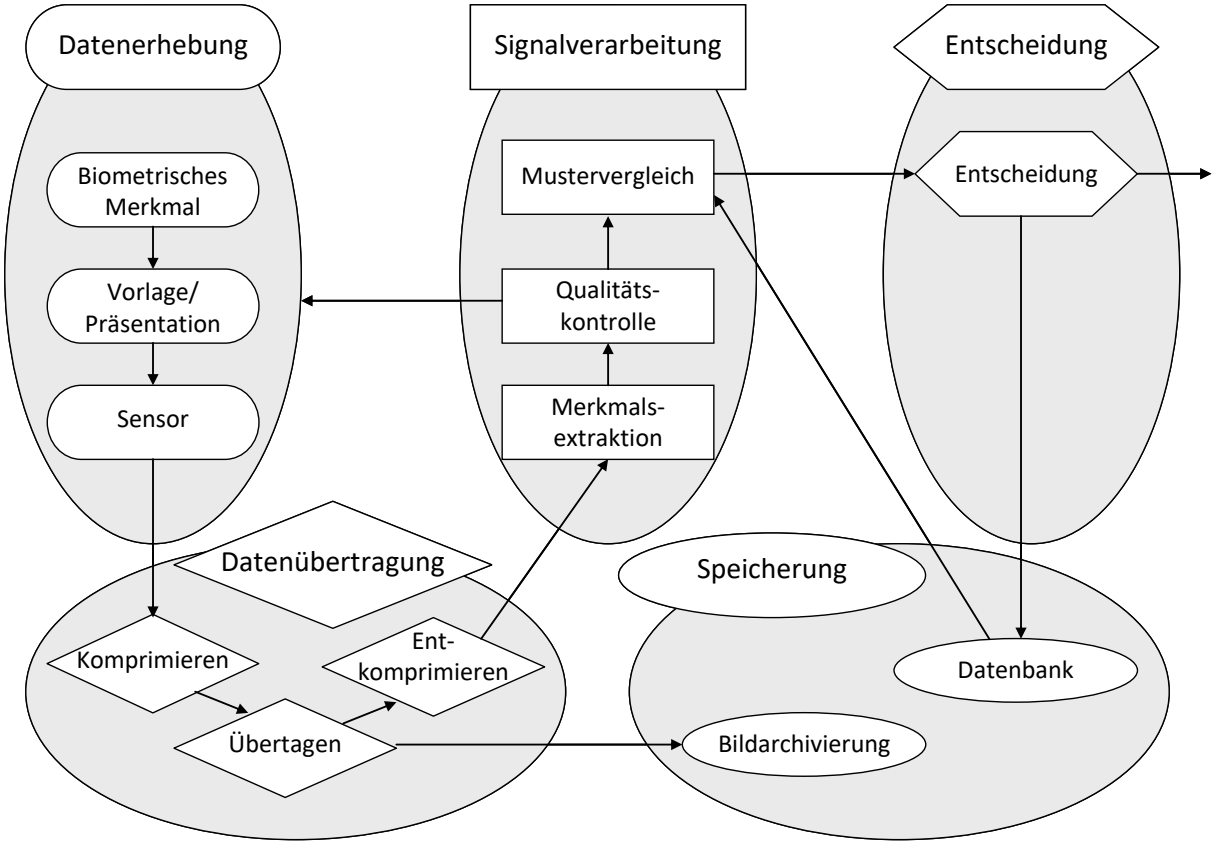


Abbildung 4 Detailliertere Sichtweise eines biometrischen Verifikationssystems, übersetzt von [Waym1999]

Obied hingegen erweitert das von Ratha et al. in [RaCB01] vorgeschlagene Angriffsmodell. Er ergänzt das Modell um einen weiteren, neunten, Angriffspunkt auf der Anwendungsebene (Application Device) in [Obie06]. Abbildung 5 zeigt alle neun Angriffspunkte des Vorgeschlagenen erweiterten Modells.

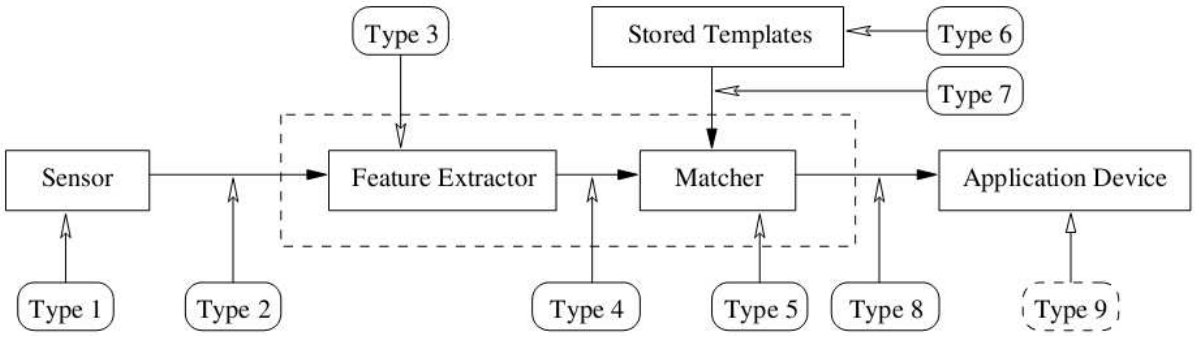


Abbildung 5 Erweitertes Modell der Angriffspunkte auf ein biometrisches System, übersetzt von [Obie06]

Bartlow und Cukic gehen in [BaCu05] und [CuBa05] noch einen Schritt weiter und kombinieren bereits bekannte Komponenten aus Waymann [Waym1999] und fügen dem Modell drei weitere Komponenten hinzu. Diese drei Komponenten bestehen aus der Administrativen Ebene, der IT-Umgebung und der Token-Präsentation. Insgesamt ermitteln Bartlow und Cukic 20 mögliche Angriffspunkte/-verfahren und 22 Schwachstellen innerhalb eines biometrischen Verifikationssystems, welche in Abbildung 6 präsentiert werden. Die von Bartlow und Cukic vorgestellten Angriffspunkte, siehe Abbildung 6, zeigen die Vielzahl möglicher Gefahrenstellen innerhalb eines biometrischen Verifikationssystems.

Sicherlich besitzt nicht jedes biometrische System alle dort gezeigten Schwachstellen, jedoch kann dieses Modell eingesetzt werden, um die potentiellen Schwachstellen oder Angriffspunkte eines Systems auf diesem abzubilden.

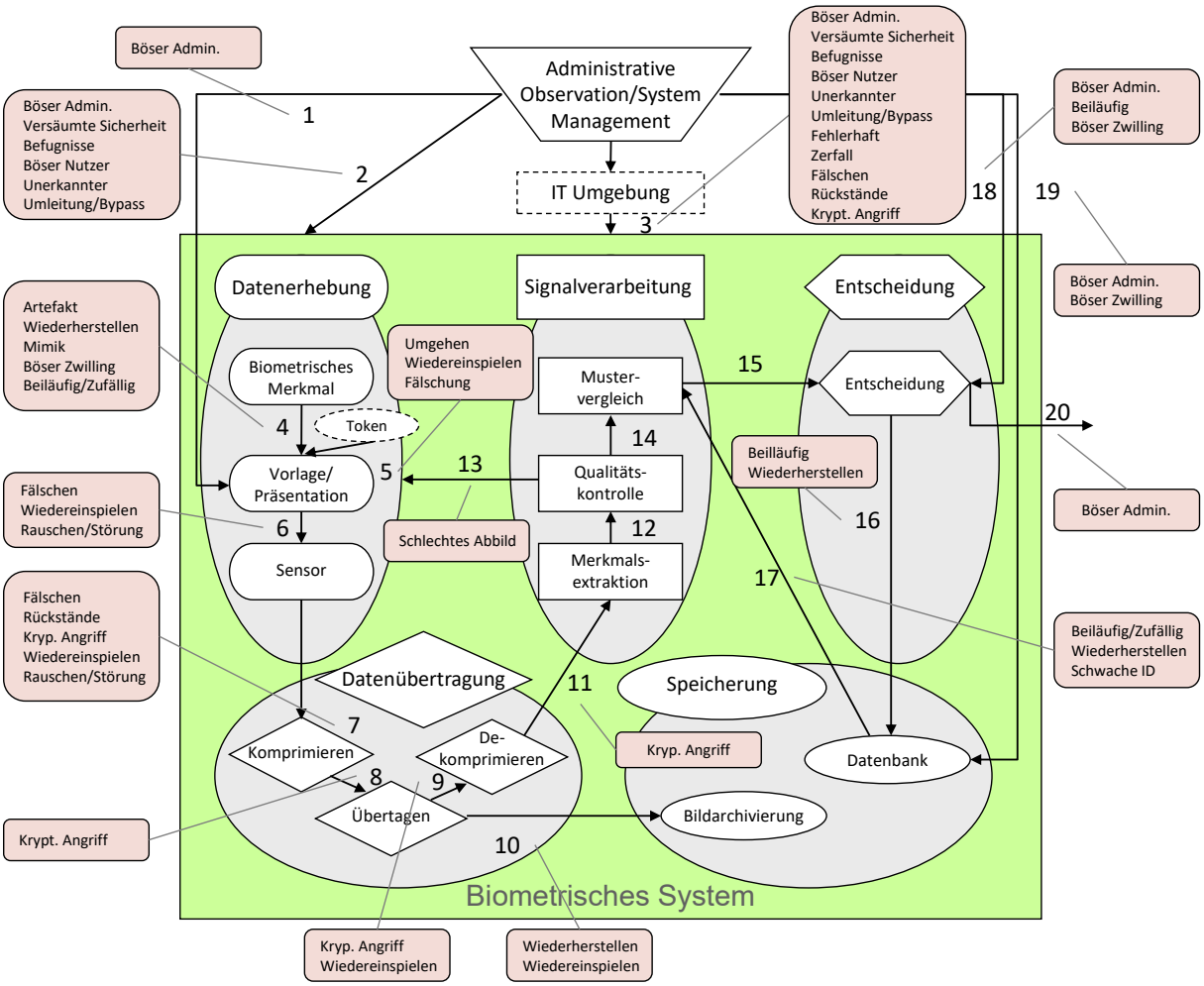


Abbildung 6 Potentielle Angriffspunkte und Schwachstellen, übersetzt von [BaCu05]

Diese Angriffsmodelle werden in Abschnitt 5 nebeneinandergestellt und bilden u.a. die Grundlage für eine in dieser Arbeit vorgestellte Forschungsaufgabe.

2.3 Angriffsmethoden/-techniken

Wie bereits im Abschnitt 1.1 erläutert wird, dienen biometrische Verifikationssysteme unter anderem dazu, nur berechtigten Personen Zugang zu einem bestimmten System, einer Ressource oder beispielsweise Räumlichkeit zu geben. Potentielle Angreifer, welche sich unberechtigt Zugriff zu einem System verschaffen wollen, entwickeln hierfür verschiedenen Techniken, um dies umzusetzen. Der Grund für das Aufbringen an krimineller Energie einer Person oder Personengruppe kann verschiedenen Ursachen haben. Angreifer wollen sich beispielsweise finanziell bereichern, indem sie sich vertrauliche Informationen widerrechtlich aneignen und weiterverkaufen oder damit erpressen. Angriffe können aber auch politischer oder persönlicher Natur sein oder gar aus einer Langenweile heraus entstehen. Der Grund für die Durchführung von Angriffen ist sehr vielfältig und soll hier an dieser Stelle nicht weiter vertieft werden. In Abschnitt 2.3.5 wird im Überblick auf die

CERT Taxonomy eingegangen, welche unter anderem potentielle Gründe für die Durchführung von Angriffen benennt.
In den folgenden Abschnitten werden einige ausgewählte Angriffstechniken auf biometrische Systeme in den jeweiligen Unterkapiteln erläutert.

2.3.1 Angriffsarten

Nach Jain et al. werden in der Literatur generell zwei verschiedene Arten von Angriffen auf biometrische Verifikationssysteme beschrieben [JaRP06], das sind zum einen Brute-Force Angriffe und zum anderen die sogenannten Gegenspielerangriffe (Adversary attacks), siehe Abbildung 7. Nachfolgend werden beide Angriffsarten nach Galbally in [Galb09] kurz beschrieben:

Brute-Force Angriff
Die Gefahr die von einem Brute-Force Angriff, auch bekannt als *zero-effort* Angriff oder *intrinsic failure*, ausgeht ist nach Jain et al. in [JaNN08] omnipräsent. Bei biometrischen Systemen existiert eine Eintrittswahrscheinlichkeit, die nicht bei null liegt, dass biometrische Daten zweier Personen sich genügend ähneln, um eine positive Verifikation zu veranlassen. Diese Eintrittswahrscheinlichkeit ist im Prinzip vergleichbar mit der Eintrittswahrscheinlichkeit, dass ein Passwort oder eine PIN erraten werden kann (siehe z.B. [KhYa08]). Bei einem Brute-Force Angriff nutzt der Angreifer das Verifikationssystem im konventionellen Sinne, ohne es zu manipulieren. Bei klassischen Passwortsystemen gibt der Angreifer beispielsweise verschiedene Buchstaben- und Zahlenkombinationen ein und probiert entsprechend alle möglichen Kombinationen systematisch durch. Dass dieser Ansatz sehr langwierig sein kann, liegt auf der Hand, erhöht sich zusätzlich mit der Komplexität (Länge des Passworts, Kombination aus Buchstaben, Zahlen und Sonderzeichen etc.) des zu "knackenden" Passworts.
Bei einem biometrischen System kann man nicht so ohne weiteres alle Kombinationen einer biometrischen Charakteristik durchprobieren. Ein Brute-Force Angriffe in der Biometrie stellt vielmehr eine statistisch mögliche Eintrittswahrscheinlichkeit dar.

Gegenspielerangriff
Bei einem Gegenspielerangriff manipuliert ein potentieller Angreifer das System. Hierbei kann der Angreifer im System registriert sein oder nicht. Er interagiert mit dem System in einer Art und Weise, wie es ursprünglich nicht gedacht war. So kann ein Angreifer beispielsweise versuchen, das System zu umgehen oder zu manipulieren, um Daten anderer Personen zu erhalten. Dafür kann der Angreifer verschiedene Werkzeuge und Hilfsmittel einsetzen wie beispielsweise Trojanische Pferde. Der Spieler ist in diesem Sinne der "normale" bzw. berechtigte Benutzer eines Systems und der Gegenspieler ist ein Angreifer.

Da ein Brute-Force Angriff auf einer statistischen Eigenart von biometrischen Systemen beruht, fokussiert sich die wissenschaftliche und technische Arbeit innerhalb der biometrischen Gemeinschaft auf die Gegenspielerangriffe, so Galbally in [Galb09].
Galbally hat in [Galb09] die Gegenspielerangriffe in direkte und indirekte Angriffsverfahren klassifiziert, siehe Abbildung 7. Dabei beschreibt er die beiden Arten wie folgt:

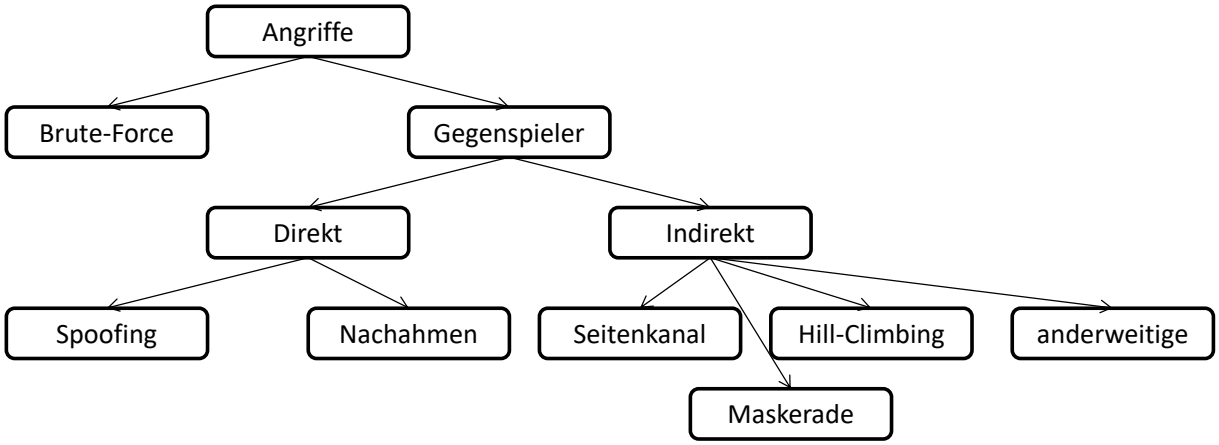


Abbildung 7 Klassifizierung von Angriffen auf biometrische Systeme, übersetzt von [Galb09]

Direkte Angriffe

Bei einem solchen Angriff versucht der Angreifer direkt am biometrischen Sensor Zugang zum System zu erlangen, indem er eine reale Person imitiert. Bei physiologisch basierten biometrischen Modalitäten (siehe Abschnitt 4.1.1), wie Fingerprint, Gesicht oder Iris, wird diese Art von Angriffen auch als Spoofing bezeichnet [Schu02]. Dabei präsentiert der Angreifer eine biometrische Fälschung (z.B. Gummifinger, Iris- oder Gesichtsbild) direkt am Sensor, um das System zu täuschen. Bei verhaltensbasierten biometrischen Modalitäten z.B. Handschrift, Gangart oder Tippverhalten (siehe Abschnitt 4.1.1), werden solche Angriffe auch als Nachahmungsangriffe oder Nachahmungen bezeichnet. Hier versucht der Angreifer, mit sogenannten "Geschickten Fälschungen" sich Zugang zum System zu verschaffen. In Bezug auf Abbildung 1 werden indirekte Angriffe auf den ersten Angriffspunkt durchgeführt. An dieser Stelle sei erwähnt, dass bei dieser Art von Angriffen keine besonderen Kenntnisse des Angreifers über die Funktionsweise des Systems (Merkmalsextraktion Vergleicher usw.) nötig sind. Des Weiteren wird der direkte Angriff im analogen Bereich ausgeführt, nicht im digitalen Bereich des Systems, wo eventuell Schutzmechanismen wie digitale Signatur zum Einsatz kommen können.

Indirekte Angriffe

Bei indirekten Angriffen wird das System von einem Angreifer mit Hilfe von verschiedenen Werkzeugen (z.B. Trojanisches Pferd) manipuliert. So kann mitunter der Merkmalsextraktor oder der Vergleicher mittels eines Trojanischen Pferds gänzlich ausgetauscht werden, um ein komplett anderes Verifikationsergebnis zu erhalten. Weiterhin können Kommunikationskanäle manipuliert werden, um Daten abzufangen, zu ändern und/oder einzuspielen. Angriffspunkte zwei bis acht in Abbildung 1 sind demnach indirekte Angriffe, siehe auch [UlJa04]. Folglich muss ein potentieller Angreifer Informationen über die Arbeitsweise des Verifikationssystems besitzen und in den meisten Fällen zusätzlich physischen Zugang zu bestimmten Komponenten haben (Merkmalsextraktor, Datenbank, usw.).
Indirekte Angriffe können, nach Ratha et al. in [RaCB01a], wiederum in weitere Kategorien eingeordnet werden. Hierbei wird die Angriffstechnik (Angriffsverfahren) für die Klassifizierung der Angriffe verwendet. Nachfolgend werden diese indirekten Angriffsverfahren nach [RaCB01a] kurz erläutert.

Replay Attacks sind Angriffe bei denen ein bereits eingespielter biometrischer Datensatz erneut in das System eingespielt wird (Abbildung 1 Angriffspunkt 2), es kann sich hierbei auch um künstlich erzeugte Daten handeln.
Ein Angriff wird als *Maskerade* bezeichnet, wenn ein auf dem biometrischen Template rekonstruierter Datensatz erstellt und in das System eingespielt wird (Abbildung 1 Angriffspunkt 2).
Tampering (zu Deutsch - einmischen) bezeichnet übergreifend Angriffe, bei denen eine Komponente des Systems manipuliert wird, um Daten zu ändern (Merkmalsvektor, Vergleichsergebnis, etc.). Für die Umsetzung eines Tampering Angriffs sind spezielle Voraussetzungen nötig. Zum einen muss ein Angreifer in der Regel Zugriff zu den jeweiligen Komponenten (z.B. physisch oder per Fernzugriff via Netzwerk) und zum anderen zusätzlich detaillierte Kenntnisse über die Funktionsweise des Systems bzw. der Komponente besitzen.
Ein *Seitenkanal-Angriff* beinhaltet die Auswertung von zusätzlichen Informationen eines Sicherheitssystems, um ggf. Korrelationen zwischen diesen zusätzlichen Informationen aus dem Seitenkanal und den sicherheitsrelevanten Daten (Chiffrat, Schlüssel usw.) zu generieren. Seitenkanalinformationen können beispielsweise Stromverbrauch, Rechenzeit oder elektromagnetische Strahlung eines Systems während der Verarbeitung von sicherheitsrelevanten Daten sein.
Hill-Climbing- bzw. Bergsteiger-Algorithmen sind einfache heuristische Optimierungsverfahren und gehören zu den lokalen Suchverfahren. Der Ausdruck leitet sich aus der Vorgehensweise ab, welches als Ziel hat, lokale Minima oder lokale Maxima zu finden. Dabei wird von einer gegebenen Startlösung ausgehend solange in der Nachbarschaft "gewandert", bis der nächstbeste Punkt erreicht wird. Da Hill-Climbing-Algorithmen auf Grund ihrer Arbeitsweise sehr oft in lokalen Maxima bzw. Minima stecken bleiben, werden sie oft mit zufällig ausgewählten Startpunkten wiederholt, um ggf. eine noch bessere Lösung zu erhalten. Weiterführende Informationen zu Hill-Climbing Algorithmen können in [Schw77] nachgelesen werden. Hill-Climbing-Algorithmen können auch eingesetzt werden, um beispielsweise künstliche biometrische Daten zu erzeugen.
Bei *Substitution*- bzw. *Stellvertreter*-Angriffen werden die im System hinterlegten biometrischen Templates durch die eines Angreifers ersetzt. Ein Angreifer kann sich demnach mit seinen biometrischen Merkmalen von einem System als eine andere im System registrierte Person authentifizieren lassen.
Des Weiteren werden in der Literatur von Ratha et al. [RaCB01a] noch so genannte *overriding response* Angriffe (Ergebnisüberschreibende Angriffe) beschrieben, bei denen beispielsweise das Verifikationsergebnis zugunsten des Angreifers überschrieben wird (Abbildung 1 Angriffspunkt 8). Das Überschreiben von Zwischenergebnissen fällt auch unter diese Kategorie von Angriffen und ist somit nicht nur auf das Verändern des Endergebnisses beschränkt.
Bei indirekten Angriffen benötigt der Angreifer Wissen über die Arbeitsweise des Verifikationssystems bzw. über die einzelnen Komponenten, um einen solchen Angriff durchführen zu können.

Biometrische Systeme sind nicht nur durch die speziellen Angriffsarten, welche eigens für selbige entwickelt wurden, gefährdet. Sie sind auch verwundbar gegen herkömmliche Angriffsmethoden auf computergestützte Systeme, wie sie von Howard in [Howa97] vorgestellt wurden. In den meisten Fällen ist ein biometrisches System auf üblicher

Rechentechnik (PC, Smart Phone, Tablett, Server, Mikrocontroller etc.) mit den dazugehörigen Komponenten (Netzwerk, Datenbanken, Betriebssystem etc.) integriert. Sind Schwachstellen und ggf. Angriffsverfahren auf eines oder mehrere Komponenten (z.B. Betriebssystem) bekannt, kann dies der erste Schritt zum Kompromittieren eines biometrischen Systems, welches auf diesem System integriert ist, sein. Als Beispiel sei hier ein mittels SSL/TLS verschlüsselter Kommunikationskanal zwischen Sensor und Eingang eines biometrischen Systems genannt. Werden nun Schwachstellen des SSL/TLS Verfahrens bekannt, die es erlauben verschlüsselte Daten zu entschlüsseln, können die Übertragenen biometrischen Daten abgefangen und für einen Angriff (z.B. Wiedereinspielen) verwendet werden. Solche und ähnliche Angriffsverfahren werden in dieser Arbeit nicht weiter betrachtet, sondern auf die spezifischen Angriffstechniken auf biometrische Systeme eingegangen. In den nachfolgenden Abschnitten werden einige Beispiele für die oben beschriebenen direkten und indirekten Angriffstechniken kurz beschrieben.

2.3.2 Direkte Angriffsverfahren

Wie bereits im vorherigen Abschnitt 2.3.1 erläutert, werden direkte Angriffe auf biometrische Systeme unmittelbar am Sensor durchgeführt.
Putte et al. beschreiben in [PuKe00] ein Verfahren zur Erzeugung von künstlichen Fingerspitzen. Dabei unterscheiden sie, ob eine Person freiwillig bei der Erzeugung mitwirkt oder ob eine Person nicht daran beteiligt ist. Ist eine Person beteiligt, so kann innerhalb von wenigen Stunden ein Gummifinger erstellt werden. Die Person legt dabei ihren Finger in eine aus Modellier-Wachs geformte Schüssel. Anschließend wird mittels hochwertiger Plasteline (z.B. aus dem zahnmedizinischen Bereich oder zum Modellieren von Figuren) der hinterlassene Fingerabdruck überdeckt. Die ausgehärtete Plasteline besitzt nun den Fingerabdruck und kann wie ein Stempel eingesetzt werden. Putte et al. beschreiben in den nächsten Schritten, wie damit ein sehr dünner (1mm) Gummifinger aus Silikon erstellt wird. Ist eine Person nicht beteiligt, wird der Fingerabdruck von einer Oberfläche, z.B. Glas, mittels feinen Pulvers, z.B. Oxidpulver, sichtbar gemacht. Anschließend wird ein Foto von diesem Fingerabdruck erstellt und in verschiedenen Schritten ein Fingerabdruckstempel erzeugt. Ob mit oder ohne Hilfe der betroffenen Person können somit dünne Silikonfingerspitzen hergestellt werden. Diese können dann beispielsweise auf einem Finger einer anderen Person befestigt (z.B. kleben) und am Fingersensor präsentiert werden. Dabei konnten die Autoren in ihren Tests zeigen, dass alle untersuchten Fingerabdrucksensoren getäuscht werden konnten.
Matsumoto et al. haben in [MMYH02] ebenfalls eine Methode zur Erstellung von künstlichen Fingerkuppen beschrieben. Sie erwähnen auch die Möglichkeit, dass ein Finger von einer Person abgetrennt werden kann, um am Sensor präsentiert zu werden. In ihrer Arbeit haben sie ihre künstlichen Fingerabdrücke an elf verschiedenen Sensoren erfolgreich getestet und gezeigt, dass nicht nur Silikon sondern auch Gelatine geeignet ist, um künstliche Fingerspitzen herzustellen. Es existieren zusätzlich weitere Beiträge ([BCG+18], [KLK+03], [ThKZ02], [WSOS04]), welche sich mit der Problematik der Gummifingerspitzen und ihre Wirkung auf Verifikationssysteme befassen.
Möglich sind auch direkte Angriffe auf andere biometrische Modalitäten; so haben Thalheim et al. in [ThKZ02] gezeigt, wie man nicht nur Fingerabdrucksensoren, sondern auch Gesichtserkennungsverfahren und Irisscanner täuschen kann. Beim Testen der Gesichtserkennungssoftware, welche eine normale Internetkamera als Sensor verwendet, werden ein aufgezeichnetes Video bzw. Bilder einer Person vor dem Sensor präsentiert

(Notebook dient als Abspielgerät). So konnte die Verifikationssoftware relativ einfach getäuscht werden. In weiteren Tests konnten sie zeigen, wie ein Irisscanner mit Hilfe eines hochauflösenden Bildes eines Auges getäuscht werden konnte. Hierfür musste in der Mitte des Auges, wo sich die Pupille befindet, ein Loch in das Foto geschnitten werden, um den Sensor zu täuschen. Ein Angreifer hielt sich dann das Foto vor seinem Auge und simulierte somit erfolgreich die zu verifizierende Person.
Im Bereich von handschriftenbasierten Verifikationssystemen haben Hennebert et al. in [HLHI07] ein mögliches Angriffsszenario gezeigt. Hierbei wird von einer Person statische Handschriftenaufzeichnung auf Papier verwendet, um dynamische Handschriftendaten zu gewinnen. Diese werden dann in das System eingespielt. Zwar werden die generierten dynamischen Handschriftendaten am Sensor vorbei in das System eingespielt, können jedoch genauso gut mit einem Schreibroboter am Sensor geschrieben werden. Das Szenario stellt auf Grund der Verwendung von handschriftlichen Aufzeichnungen einer Person ein relativ realistisches Angriffsszenario dar. Jede Person hinterlässt in der Regel eine Vielzahl von handschriftlichen Dokumenten (Einkaufsliste, Notizen, Adresse auf einem Briefumschlag usw.). Ein potentieller Angreifer kann sich beispielsweise durch Sozial Engineering (schriftliche Wegbeschreibung erfragen oder den Mülleimer der Zielperson durchsuchen) Zugriff zu handgeschriebenen Dokumenten verschaffen.
Zusätzlich haben sich Alonso-Fernandez et al. mit der Fälschung von Handschriftendaten in [AFG+09] beschäftigt. Hier wurde unter anderem versucht ein statisches und dynamisches Verifikationssystem mit Fälschungen zu täuschen. Sie haben in ihrer Arbeit gezeigt, dass Handschriftensysteme auf Basis von statischen Handschriften anfälliger gegen geübte Fälschungen sind.

2.3.3 Indirekte Angriffsverfahren

Nachfolgend werden indirekte Angriffsverfahren kurz beschrieben, wobei nicht aus allen in Abschnitt 2.3.1 beschrieben Angriffstypen jeweils ein Beispiel erläutert wird. Die in diesem Abschnitt beschrieben Angriffsverfahren sind teilweise aus anderen Bereichen der Sicherheitstechnik adaptiert, z.B. der Kryptoanalyse, und für biometrische Authentifizierungssystem angepasst worden. Somit kann die Arbeitsweise einzelner Verfahren leicht von Angriffsverfahren, aus denen sie ursprünglich hervorgingen, abweichen. Das Grundprinzip der Angriffsverfahren ist jedoch weitestgehend gleich.

Hill-Climbing
Uludag et al. beschreiben in [UlJa04] ein Hill-Climbing-Angriff auf ein fingerabdruckbasierendes Verifikationssystem. Dieses System ermittelt bestimmte Eigenschaften der Minutien eines Fingerabdrucks die für einen Vergleich (Referenzdaten mit aktuell präsentierten Daten) herangezogen werden. Minutien sind Endungen bzw. Verzweigungen von Papillarleisten eines Fingerabdrucks. In einem Verifikationssystem wird in der Regel die Lage aller Minutien und der Winkel einer Minutie zu zughörigen Papillarleisten bestimmt und gespeichert. Diese gespeicherten Minutiendaten werden als Referenz zur Verifikation herangezogen. Auf Basis dieser Daten ist es nicht mehr möglich, einen kompletten Fingerabdruck zu rekonstruieren. Uludag et al. umgehen den Merkmalsextraktor und schleusen künstlich erzeugte Minutiendaten in das System ein (Abbildung 1 Angriffspunkt 4). Weiterhin besitzt in ihrem Angriffsszenario der Angreifer Zugang zu den Ergebnissen des Vergleiches seines aktuellen künstlichen Minutiendatensatzes und des Referenzdatensatzes (Abbildung 1 Angriffspunkt 8). Nun werden mittels Hill-Climbing-Verfahren

Minutiendaten erzeugt, in das System eingespielt und der Vergleichswert herangezogen, um zu ermitteln, inwieweit dieser künstliche Datensatz mit dem Referenzdatensatz übereinstimmt. Im Durchschnitt werden 271 Iterationen benötigt, um ein künstlichen Minutiendatensatz zu erzeugen, welcher genügend Ähnlichkeit mit den Referenzdatensatz aufweist.
Ein ähnliches Verfahren stellen Martinez-Diaz in [MFA+06] vor. Hierbei wird der Hill-Climbing-Angriff auf ein Matching-On-Card (MoC) System und ein NFIS Referenzsystem [GWM+01] angewendet. Das MoC System verwendet weniger Minutieninformation als das Referenzsystem und ist, laut Autoren, aus diesem Grund anfälliger gegen einen Hill-Climbing-Angriff.

Adler stellt in [Adle03] einen Hill-Climbing-Angriff auf ein Erkennungssystem basierend auf der Modalität Gesicht vor. Der Angriff setzt auf den zweiten Angriffspunkt (siehe Abbildung 1) auf. Hierbei werden künstlich erzeugte Gesichtsbilder in das System eingespielt. Anhand des Vergleichswerts zu den Referenzdaten wird das künstliche Gesicht ein wenig verändert und erneut eingespielt. Dieser Vorgang wird so lange wiederholt, bis ein bestimmter Vergleichswert erreicht wird. Adler konnte innerhalb seiner Tests zeigen, dass nach ca. 4000 Iterationen ein ausreichend guter Vergleichswert erzielt wurde, welcher eine erfolgreiche Verifikation zur Folge hätte. Ein ähnliches Verfahren wird von Soutar in [Sout02] präsentiert, jedoch auf ein weniger komplex arbeitendes Gesichtserkennungsverfahren.

Ein weiteres Hill-Climbing-Angriffsverfahren auf ein handschriftenbasiertes Erkennungssystem stellen Galbally et al. in [GaFO07] vor. Der Angreifer ist in der Lage einen Merkmalsvektor in das System einzuspielen (siehe Abbildung 1 Angriffspunkt 4). Zusätzlich hat er in diesem Szenario Zugriff auf den Vergleichswert (Abstandsmaß) des aktuellen Merkmalsvektors zum Referenzmerkmalsvektor. Mit Hilfe eines Hill-Climbing-Algorithmus werden nun Merkmalsvektoren erstellt, in das System eingespielt und mit dem Referenzmerkmalsvektor verglichen. Der Vergleichswert wird als Richtwert für den Hill-Climbing-Algorithmus verwendet. Dieses Verfahren wurde von Marta Gomez-Barrero et al. in [GFOG11] optimiert, wobei ein anderer Hill-Climbing-Algorithmus (Uphill simplex anstelle von Bayesian) und ein erweitertes Testfeld verwendet wurde. Beide Verfahren erzielten eine erfolgreiche Angriffsrate von über 90 Prozent. Das Verfahren arbeitet mit Merkmalsvektoren fester Länge. Laut den Autoren kann dieses Angriffsverfahren somit auf alle biometrischen Verifikationsverfahren angewendet werden, welche eine fixe Länge des Merkmalsvektors vorschreiben. Damit ist dieses Angriffsverfahren, in Anbetracht der potentiellen Übertragbarkeit auf andere biometrische Systeme, recht interessant und soll in dieser Arbeit genauer betrachtet werden (siehe Abschnitt 7.1.3).

Seitenkanal
Galbally et al. haben ein auf Fingerabdruck basierendes biometrisches Verifikationssystem (NFIS2 von NIST) einer Verarbeitungszeitanalyse unterzogen. Hierbei wird die Verarbeitungszeit für die Berechnung eines Vergleichswertes (Matching) gemessen und mit dem erzielten Vergleichswert (Matching-Score) verglichen. Dabei wurde festgestellt: je höher die Verarbeitungszeit ist, desto höher der Vergleichswert. Es konnte eine klare Korrelation der beiden Werte ermittelt werden. Diesen Zusammenhang können potentielle Angreifer nutzen, um beispielsweise effizienter Angriffsdaten zu erzeugen.

Substitution/Stellvertreter
In der Arbeit von Biggio et al. [BDFR13] wird ein Angriff vorgestellt, welcher die Eigenschaft von adaptiven biometrischen Systemen ausnutzt. Adaptive biometrische Systeme verwenden Mechanismen, um biometrische Referenzdaten regelmäßig zu aktualisieren. Auf diese Weise wird der Alterung von biometrischen Daten entgegengewirkt, welche eine der Herausforderungen biometrischer Erkennungssysteme darstellt (siehe auch Abschnitt 4.1.6 Herausforderungen in der Biometrie). Biggio et al. haben in ihrer Arbeit eine Methode entwickelt, die hinterlegten biometrischen Templates einer Gesichts-Datenbank durch eigene biometrische Templates zu ersetzen, um sich so Zugang zum System zu verschaffen.

Maskerade
Wie bereits am Anfang des Abschnitts erläutert, sind bei indirekten Angriffen Information bezüglich der Arbeitsweise und/oder der technischen Umsetzung notwendig. So auch in der vorgestellten Angriffsmethode von Mai et al. in [MCYJ18]. In ihrer Arbeit greifen sie ein Gesichtserkennungssystem an, welches auf eine *Convolution Neural Network* (CNN) basiert. Hierbei hat der potentielle Angreifer Zugriff auf die Referenzdaten, welche in einer Datenbank hinterlegt sind. Anhand der Referenzdaten konstruieren sie Bilder (Gesichter) mittels eines angepassten *De-Convolutional Neural Network* (D-CNN). Im Schnitt erzielten sie eine positive Verifikationsrate mit ihren rekonstruierten Gesichtern von ca. 95%. Der Angreifer muss in diesem Angriffsszenario u.a. die Arbeitsweise des CNN kennen, um das entsprechende D-CNN zu konstruieren. Weitere Beispiele für Maskerade Angriffe werden u.a. in [GRG+13] und [VeSa11] vorgestellt. Darüberhinaus sind dem Autor, trotz intensiver Literaturrecherche, keine weiteren Forschungsarbeiten bekannt, die zwischen den Jahren 2013 und 2020 veröffentlicht wurden.

Die unter anderem vom Autor der vorliegenden Arbeit entwickelten und vorgestellten Angriffstechniken in [KVS+10], [KüVi10] und [KüVi10a] benötigen ebenfalls Information über die Arbeitsweise des Verifikationsalgorithmus und die verwendeten Merkmale. So wird in den drei Beiträgen eine Schwachstelle des Handschriftenverifikationsalgorithmus ausgenutzt und ein Merkmalsvektor auf Basis der Referenzdaten konstruiert. Mit Hilfe des Merkmalsvektors ist eine erfolgreiche Verifikation möglich, sobald dieser am Angriffspunkt 4 (siehe Abbildung 1) eingespielt wird. In den genannten Arbeiten des Autors werden jedoch mittels Merkmalsvektors und den darin enthaltenen statistischen Informationen der Referenzhandschrift künstliche Handschriftendaten erzeugt. So kann ein Angriff direkt am Sensor (Angriffspunkt 1) mittels Schreibroboter oder am Angriffspunkt 2 durchgeführt werden.
In [KüVi10a] werden so aus dem zurückgerechneten bzw. konstruierten Merkmalsvektor Informationen über die Anzahl der Maxima und Minima des vertikalen und horizontalen Schreibsignals zur Generierung von künstlichen Handschriften verwendet. Dabei werden Stützpunkte generiert und mittels einer Spline-Interpolation das vertikale und horizontale Schreibsignal erzeugt. In Abschnitt 4.2.4 wird u.a. die Konstruktion bzw. Rückrechnung des Merkmalsvektors genauer beschrieben.
In [KVS+10] werden künstliche Handschriftendaten ebenfalls auf Basis der statistischen Merkmale des zurückgerechneten Merkmalsvektors erstellt. Hier kommt zusätzlich ein evolutionärer Algorithmus zum Einsatz, welcher das "fitteste" künstliche Handschriftendatum ermittelt. Fittest heißt in diesen Zusammenhang, dass eine Handschrift

(Individuum) den besten Vergleichswert zur Referenzhandschrift aufweist. Der Vergleichswert wird ermittelt, indem ein sogenannter biometrischer Hash aus dem Handschriftendatum erzeugt wird, welcher mit dem biometrischen Referenz Hash verglichen wird. In den Arbeiten konnten künstliche Handschriftendaten erzeugt werden, die dem angegriffenen Referenz-Hash bis zu knapp 60% ([KüVi10a]) respektive 70% ([KVS+10]) ähneln.
Künstlich erzeugte Handschriftendaten können, wie bereits weiter oben erläutert, potentiell mittels Schreibroboter direkt am Sensor eingespielt werden. Ein möglicher Angreifer benötigt somit keinen direkten Zugriff auf andere Komponenten des Systems. Besitzt der Angreifer teilweise Zugriff auf bestimmte Komponenten dann kann er ggf. Handschriftendaten am Sensor vorbei in das System einspielen. Prinzipiell können Algorithmen zur Erzeugung künstlicher Handschriftendaten ein mächtiges Werkzeug für potentielle Angreifer darstellen. Aus diesem Grund soll das Erzeugen von künstlichen Handschriftendaten im nächsten Abschnitt genauer betrachtet werden.

2.3.4 Verfahren zur Erzeugung künstlicher Handschriftendaten

Nachdem in den vorherigen Abschnitten verschiedene Angriffsarten auf biometrische Systeme im Allgemeinen vorgestellt wurden, befasst sich dieser Abschnitt speziell mit Angriffsverfahren für handschriftenbasierte Verifikationssysteme, im speziellen die Erzeugung von künstlichen Handschriften.
In der Literatur werden Verfahren zur Erzeugung künstlicher Handschriftendaten je nach Arbeitsweise in drei verschiedenen Kategorien unterschieden, siehe dazu auch [GPF+12]. Im Allgemeinen unterscheiden sich die Kategorien anhand der Ausgangsdaten, welche als Basis für die Erzeugung dienen.

Duplizierte Handschriftendaten
In diesem Fall startet der Generierungsprozess mit einem oder mehreren realen Handschriftendaten einer Person. Innerhalb mehrerer Transformationsschritte werden künstliche bzw. duplizierte Handschriftendaten passend zu dieser Person generiert. Mit Hilfe dieser Methode können bereits bestehende biometrische Datensätze künstlich vergrößert werden. Sie eignen sich jedoch nicht, um beispielsweise komplette Datensätze einer „neuen“ Person zu erzeugen. Vielmehr dienen sie dazu, die Anzahl der Enrollment -Datensätze von Identifikations- und Verifikationssystemen zu erhöhen, siehe dazu [VaBu03], [AlMA08] und [GFM+09].

Aneinanderreihen von Daten
In vielen Verfahren, z.B. vorgestellt in [BaLM07], [BLML06], [Guyo96] und [LoRa05], werden reale Handschriftendaten aneinandergereiht, um neue künstliche Handschriftendaten zu erzeugen. Hierbei kommen beispielsweise einzelne Buchstaben, Glyphen oder Wörter zum Einsatz, die mittels bestimmter Verfahren zu einem neuen Handschriftendatum verbunden werden. Ähnlich wie beim Erzeugen von duplizierten Handschriftendaten, können mit dieser Methode lediglich künstliche Handschriftendaten einer Person, nämlich der Person, von denen die realen Handschriftenteile stammen, erzeugt werden. Sicherlich können Handschriftensätze von verschiedenen Personen aneinandergereiht werden, um neue Handschriftendaten zu erzeugen. Jedoch könnten beispielsweise potentielle Angreifer damit weniger Chancen haben, die Handschrift einer bestimmten Person künstlich nachzuahmen.

Künstliche Individuen

Bei der Erzeugung künstlicher Individuen bzw. synthetischer Daten, werden im Vorfeld biometrische Daten realer Individuen untersucht. Dabei werden bestimmte Parameter aus den realen Daten ermittelt, welche typisch für diese biometrische Modalität sind und für die Konstruktion eines Modells zur Generierung künstlicher Individuen herangezogen werden können. In einem nachfolgenden Schritt können auf Basis dieser künstlichen Individuen synthetische Daten passend zum Individuum erzeugt werden. Unter Verwendung dieses Prinzips wurden in verschiedenen biometrischen Modalitäten (Iris [ZuSC07], [CWH+04], [ShRo06]; Fingerabdruck [Capp03] und Sprache [PiCL89], [Klat80]) bereits einige Verfahren vorgestellt. Im Bereich der Handschriftenbiometrie wurden ebenfalls einige Modelle vorgestellt, welche den charakteristischen Prozess des Schreibens versuchen abzubilden. Dazu zählen unter anderem das Sigma Lognormal Modell [DjPl09], das Oszilatory Motion Modell [Holl81] und das Beta Elliptic Modell [BeKA07].

Der Vorteil dieses Verfahrens zur Erzeugung künstlicher Individuen besteht darin, dass beliebig viele Individuen "geschaffen" werden können und zu jedem Individuum die dazu passenden synthetischen Daten in beliebiger Menge. Dadurch lassen sich Testdatenbanken für verschiedene Zwecke (z.B. Testen neuer Merkmalsextraktoren, Performanz-Teste, etc.) erstellen.

In [GPF+12] beschreiben Galbally et al. ein solches Verfahren zur Erzeugung künstlicher Individuen inklusive synthetischer Handschriftendaten. Dabei kombinieren sie eine Spektralanalyse von Handschriftendaten realer Personen mit der kinematischen Theorie schneller Bewegungen (engl. kinematic theory of rapid human movements), um eine Datenbank mit künstlichen Individuen und den dazugehörigen synthetischen Handschriftendaten zu generieren. Die Evaluation dieses Verfahrens wird in einem separaten Beitrag [GPF+12a] beschrieben. Dabei werden verschiedene Testszenarien definiert und die Ergebnisse (synthetische Handschriftendaten) mit realen Handschriftendaten verglichen. Weiterhin stellen die Autoren eine Datenbank mit künstlichen Individuen und den dazugehörigen synthetischen Handschriften unentgeltlich zur Verfügung.

2.3.5 Einordnung von Angriffen mittels CERT Taxonomy

Das Computer Emergency Response Team (CERT) ist eine Organisation bzw. ein spezielles Team von IT- Sicherheitsfachleuten, welches sich mit aktuellen Sicherheitsvorfällen in der IT- Welt befasst. CERT gibt unter anderen Warnungen vor Sicherheitslücken (z.B. in Betriebssystemen) heraus und bietet Lösungsvorschläge hierzu an. Die Bekanntmachung von neuartigen Virenverbreitungen oder Angriffstaktiken zählt ebenfalls dazu. Es existieren mehrere Computer Emergency Response Teams, die teils Branchenspezifisch agieren. Bekannte CERTs in Deutschland sind unter anderem das CERT-Bund (CERT der Bundesverwaltung beim BSI), S-CERT (Sparkassen-Finanzgruppe) oder das SIEMENS-CERT. Um Angriffe (Vorfälle) besser beschreiben zu können, wurde eine Klassifizierung eingeführt, welche hier kurz anhand der Abbildung 8 erläutert werden soll, siehe auch [HoLo98].

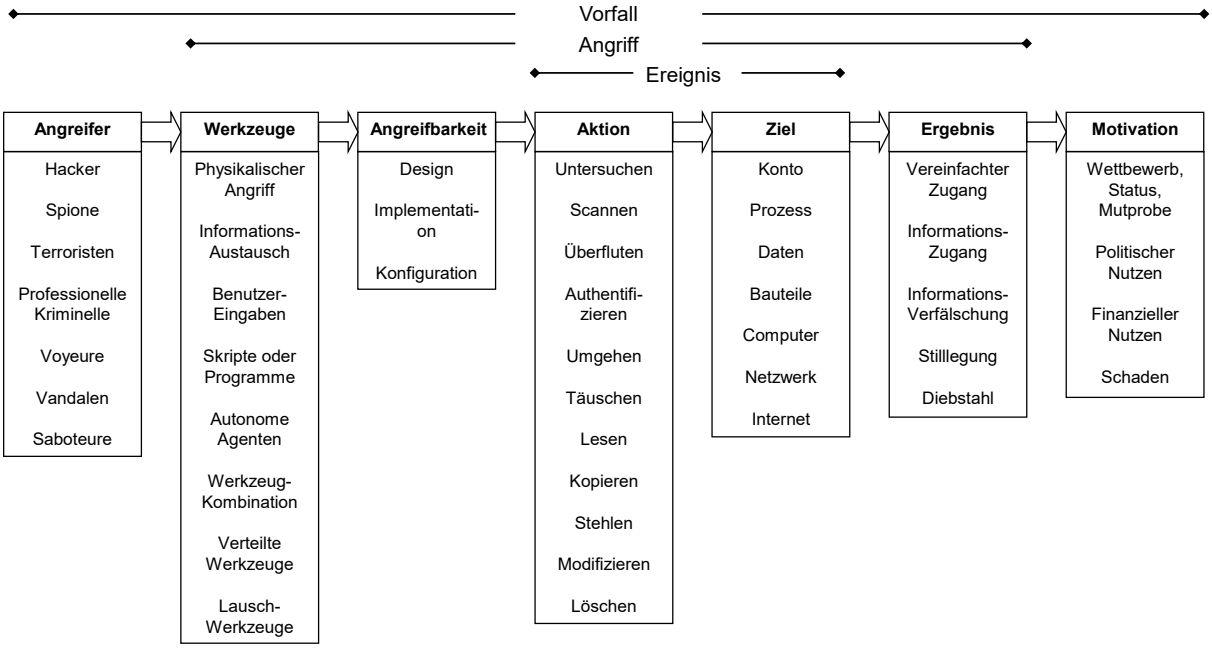


Abbildung 8 CERT Klassifizierung für Sicherheitsvorfällenach [Howa97], übersetzt für Forschung und Lehre durch [AMSL20]

Ein Angriff ist in der CERT-Klassifizierung ein Teil einer Störung, welche in mehrere Schritte unterteilt werden kann. Ein *Angreifer* versucht, mit bestimmten *Werkzeugen* eine *Schwachstelle* auszunutzen (Angreifbarkeit), um eine *Aktion* auszuführen. Der Angreifer hat ein bestimmtes *Angriffsziel*. Nach der Ausführung der Aktion auf dieses Ziel stellt sich ein gewolltes *Ergebnis* ein, welches den gewünschten *Auftrag* erfüllt (siehe Abbildung 8). Mit Hilfe dieser Taxonomie kann man beliebige Vorfälle in der IT- Sicherheit klassifizieren, um eine gute, einheitliche und schnelle Sicherheitseinschätzung treffen zu können.

2.4 Gegenmaßnahmen

In den Abschnitten 2.2 und 2.3 wurden Schwachstellen und Angriffstechniken auf biometrische Verifikationssysteme beschrieben. Um solche biometrischen Angriffe zu verhindern bzw. deren Erfolgschancen zu minimieren müssen Gegenmaßnahmen getroffen werden. Gegenmaßnahmen sind zusätzliche Funktionen die in der Regel nicht für die eigentliche Authentifizierung benötigt werden. Sie dienen dazu, das biometrische System vor Angriffsversuchen zu schützen.
Genau wie bei der Unterscheidung zwischen direkten und indirekten Angriffsverfahren, kann man auch die Gegenmaßnahmen in direkte und indirekte unterteilen. Hierbei bezieht sich das Direkte und Indirekte jedoch auf den passenden Angriffstyp, den man versucht zu vermeiden und nicht auf die Arbeitsweise der Gegenmaßnahme selbst. Bei den direkten Gegenmaßnahmen wird unter anderen geprüft, inwieweit die präsentierten biometrischen Daten von einer realen Person stammen. In der Regel geschieht dies durch die Feststellung, ob es sich um eine "lebende" Repräsentation oder ggf. um eine künstlich erzeugte oder kopierte Abbildung der biometrischen Modalität handelt. Sie sind so ausgelegt, direkte Angriffe (Abbildung 1 Angriffspunkt 1) zu vermeiden bzw. die Gefahr durch solche zu minimieren.
Indirekte Gegenmaßnahmen richten sich entsprechend gegen Angriffe auf Angriffspunkt zwei bis acht (siehe Abbildung 1). Im Gegensatz zu direkten Angriffen muss in diesen Fällen

ein Angreifer Informationen über die Arbeitsweise und den Aufbau des biometrischen Systems besitzen, um einen erfolgreichen Angriff durchzuführen. Jedoch können die für einen indirekten Angriff benötigten digitalen Daten schneller vervielfältigt und modifiziert werden, als die für einen direkten Angriff benötigte Fälschungen am Sensor. Indirekte Gegenmaßnahmen beinhalten unter anderem Techniken, einzelne Komponenten (Merkmalsextraktor, Vergleicher, Datenbank etc.) und deren Kommunikationskanäle vor Angriffen zu schützen. In den folgenden Abschnitten 2.4.1 und 2.4.2 werden einige in der Literatur beschriebenen Techniken zur Abwehr von biometrischen Angriffen vorgestellt.

2.4.1 Maßnahmen gegen direkte Angriffe

Um direkte Angriffe auf biometrische Systeme zu detektieren und zu verhindern, müssen Vorbereitungen zur Erkennung und Abwehr getroffen werden. Dabei können zum Beispiel Mechanismen eingesetzt werden, welche die präsentierten biometrischen Daten am Sensor auf "Lebendigkeit" prüfen. So kann zum Beispiel die Temperatur oder Oberflächenspannung der Haut am Fingerabdrucksensor direkt gemessen werden, um ggf. Gummifingerabdrücke zu detektieren. Derakhshani et al. beschreiben in [DSHG03] eine weitere Methode, gefälschte Fingerabdrücke zu detektieren. Dieses Verfahren nimmt ein kurzes Video (ca. fünf Sekunden) des Fingerabdrucks auf, während er am Sensor präsentiert wird und wertet die Schweißbildung entlang der Schweißporen auf den Papillarlinien aus. Dabei stützen sich die Autoren auf die Annahme, dass Gummifingerabdrücke keine Schweißbildungen der Haut nachempfinden können bzw. auf der Oberfläche keine Schweißbildung eintritt.
In [PaCz06] stellen Pacut et al. Lebendigkeitsprüfungen für Verifikationssysteme basierend auf der Modalität Iris vor. Hier werden Pupillenbewegungen und Augenzwinkern (Lidschläge) detektiert, welche bei Fälschungen mittels Bilder bzw. Fotos nicht auftreten.
Li et al. zeigen in [LHW+18] eine Methode zur Lebendigkeitsprüfung für die Modalität Gesicht. Dabei verwenden sie ein künstliches neuronales Netzwerk, genauer ein 3D Convolutional Neural Network (3D CNN), um Fälschungen zu erkennen. Mit dieser Methode konnten die Autoren Gesichtsbilder und Videos, die vor einer Videokamera präsentiert wurden, von realen Gesichtern unterscheiden.
Des Weiteren können die digitalisierten Repräsentationen von biometrischen Daten hinsichtlich typischer Fälschungseigenschaften untersucht werden. Bei der Modalität dynamische Handschrift kann beispielsweise die Schreibgeschwindigkeit oder Beschleunigung ermittelt werden. Weichen diese Eigenschaften zu stark von realen Handschriften ab, handelt es sich ggf. um ein künstlich erzeugtes Handschriftendatum.

2.4.2 Maßnahmen gegen indirekte Angriffe

Hill-Climbing-Angriffe sind die in der Literatur am meisten beschriebenen indirekten Angriffe. Sie basieren in der Regel auf ein Unterscheidungsmaß (Matching-Score), welches der Vergleicher während eines Vergleiches zwischen den biometrischen Referenzwert und den aktuell präsentierten Daten ermittelt. Eine einfache, jedoch nicht für alle Systeme umsetzbare Gegenmaßnahme ist die, auf einen Vergleichswert zu verzichten und eine direkte binäre Aussage zu treffen (wahr oder falsch bzw. echt oder unecht). Dort, wo beispielsweise eine biometrische Fusion eingesetzt wird, muss jedoch ggf. das Unterscheidungsmaß ermittelt und weiterverarbeitet werden, siehe dazu u.a. [UlJa04].
Eine ebenfalls einfache und recht effektive Methode ist, laut Uludag und Jain in [UlJa04], die Anzahl der Falscherkennungen pro Person und Tag zu beschränken. Insbesondere

wenn ein auf Hill-Climbing-Verfahren basierender Angriff mehrere tausend Iterationen benötigt und die Beschränkung auf zwanzig Falscherkennungen pro Tag reduziert ist, ist diese Maßnahme durchaus sinnvoll.
Ratha et al. stellen in [RaCB01a] ein Challenge Response Verfahren vor, um Angriffe vom Typ 2 (siehe Abbildung 1 Angriffspunkt 2) zu verhindern. Challenge Response Verfahren fordern in der Regel zusätzliche Informationen von einer Person oder von einem System an. So ist beispielsweise die Frage nach dem Geburtsnamen der Mutter, während einer Passwortabfrage, eine solche zusätzliche Information. Ratha et al. schlagen ein Challenge Response Verfahren basierend auf Bildern für die Modalität Fingerabdruck vor. Dabei ist idealerweise der Sensor mit einem Mikroprozessor direkt verbunden, womit eine sichere Kommunikation zwischen den beiden Komponenten gewährleistet ist. Der Prozessor kann nun bestimmte Funktionen auf die am Sensor präsentierten Daten, durchführen z.B. die Graustufenwerte einer bestimmten Pixelreihenfolge ermitteln. Anschließend wird das Bild des Fingerabdrucks inklusive des Ergebnisses der Funktion an das biometrische System, z.B. Merkmalsextraktor, weitergeleitet. Der Merkmalsextraktor kann nun die gleiche Funktion ausführen (Graustufenwerte von den gleichen Bildpunkten ermitteln) und die Ergebnisse miteinander vergleichen. Erst wenn diese Ergebnisse übereinstimmen, werden die Bilddaten weiterverarbeitet und die eigentliche Merkmalsextraktion gestartet. Ratha et al. schlagen in diesem Beitrag zusätzlich weitere Funktionen (Checksummenberechnung etc.) vor, welche auf das Sensorbild angewendet werden können. Mit einem solchen Challenge Response Verfahren wird es einen Angreifer erheblich erschwert, Angriffsdaten vom Typ 2 in das System einzuspielen.
Die Forschung im Bereich der Gegenmaßnahmen wird ständig weiterverfolgt und ähnelt dem Katz-und-Maus-Spiel von Schadsoftware und Herstellern von Antiviren-Software. In einigen Bereich existieren entsprechende Forschungslücken. So versuchen beispielsweise Ferrer et al. in [FMV+12] ein Verfahren zu entwickeln, welches die Identität eines Fälschers bestimmen soll. Der Fälscher ist registrierter Nutzer einer biometrischen Handschriftendatenbank und versucht sich per Nachahmung als anderer Nutzer positiv zu verifizieren. Ihr beschriebenes und getestetes Verfahren hat leider nicht zum Erfolg geführt. Die Autoren beschreiben dieses Feld als ein offenes Forschungsthema. In dieser Arbeit soll dieses Forschungsthema nicht behandelt werden, sondern soll beispielhaft dafür sein, dass das Feld der Gegenmaßnahmen noch einige offene Forschungsfelder besitzt. Im nachfolgenden Abschnitt sollen die offenen Forschungsfragen und Aufgaben dieser Arbeit näher betrachtet werden.

3 Ziel der Arbeit

Im Abschnitt 1.2 sind die Einordnung und Abgrenzung der Arbeit erläutert. Ferner werden die drei wesentlichen Teilbereiche dieser Arbeit vorgestellt. In Abschnitt 3 werden diese drei Bereiche aufgegriffen und entsprechende Forschungsziele aufgestellt und definiert.

In den nachfolgenden drei Abschnitten werden diese Teilbereiche als Forschungsaufgaben (FA1, FA2 und FA3) entsprechend beschrieben. Zu Beginn eines jeden Abschnittes wird das jeweilige Ziel in einem Absatz formuliert. Zum besseren Verständnis werden die entsprechenden Teilaufgaben der einzelnen Forschungsaufgaben anschließend nochmals aufgelistet.

Nach einer Zusammenfassung der Grundlagen im Abschnitt 4 werden in den Abschnitten 5, 6 und 7 die jeweiligen Forschungsaufgaben behandelt. Dazu werden die Vorgehensweisen, die Durchführung und entsprechende experimentelle Tests der Forschungsaufgaben vorgestellt und diskutiert. Dabei wird der aktuelle Forschungsstand innerhalb der jeweiligen Thematiken berücksichtigt und eingearbeitet.

3.1 Klassifizieren von Angriffen (FA1)

Innerhalb der Forschungsaufgabe 1 sollen ausgewählte biometrische Angriffsverfahren klassifiziert werden. Hierfür werden die Verfahren in direkte und indirekte Verfahren eingeordnet sowie bestimmt, auf welche Angriffspunkte (Abbildung 1) die jeweiligen Verfahren ansetzten. Des Weiteren soll eine Bewertung der Angriffe bezogen auf die Adaptierbarkeit und somit potentielle Gefahr für andere biometrische Modalitäten abgegeben werden. Zudem wird ein ausgewähltes Verfahren auf einen speziellen handschriftenbasierenden Verifikationsalgorithmus [Viel06] adaptiert und das Angriffspotential evaluiert. Auf Basis dieses Ergebnisses sollen Designvorschläge (Gegenmaßnahmen) zur Verbesserung der Sicherheit des Handschriftenalgorithmus gegeben werden. Weiterhin sollen allgemeine modalitätenübergreifende Designvorschläge definiert werden, um Gefahren, die von bestimmten Angriffen ausgehen, zu minimieren.

Teilaufgabe A:
Klassifizieren ausgewählter Angriffsverfahren auf biometrische Verifikationssysteme.

Teilaufgabe B:
Bewerten von klassifizierten Angriffsverfahren bezogen auf ihre Adaptierbarkeit.

Teilaufgabe C:
Adaptieren und evaluieren eines ausgewählten Angriffsverfahrens auf den in [Viel06] vorgestellten handschriftenbasierenden Verifikationsalgorithmus.

3.2 Erzeugen künstlicher Handschriftendaten (FA2)

Synthetisch erzeugte Handschriftendaten stehen im Vordergrund der Forschungsaufgabe 2. Dabei soll ein Verfahren eruiert und evaluiert werden, welches real wirkende Handschriftendaten auf der Basis bestimmter Parameter, ähnlich wie in [GPF+12], generiert. Dabei sollen vorhandene biometrische Daten realer Personen nicht direkt zu neuen Handschriftendaten zusammengesetzt werden, wie in Abschnitt 2.3.4 beschrieben ist. Des

Weiteren sollen diese künstlichen Daten sich an einem Verifikationssystem ähnlich verhalten (Fehlerraten) wie reale Handschriftendaten. Dabei soll jedoch das real wirkende Erscheinungsbild der künstlichen Handschriftendaten nicht negativ beeinflusst werden. Hierbei soll ebenfalls untersucht werden, inwiefern diese künstlich erzeugten Handschriftendaten als Angriffsdaten auf reale Handschriften verwendet werden können. Darauf aufbauend sollen Designvorschläge zur Generierung von künstlichen Handschriften formuliert werden, welche dazu beitragen, diese wie reale Handschriftendaten an einem Verifikationssystem erscheinen zu lassen.

Teilaufgabe A:
Entwickeln und evaluieren eines Verfahrens zur Generierung von künstlichen Handschriften.

Teilaufgabe B:
Bestimmen geeigneter Parameter, welche die Eigenschaften der künstlichen Handschriftendaten so anpassen, dass diese den Eigenschaften realer Handschriftendaten ähneln.

Teilaufgabe C:
Untersuchen, inwieweit die künstlich erzeugten Handschriften für Angriffe auf reale Handschriftendaten eingesetzt werden können.

3.3 Hill-Climbing-Verfahren adaptieren (FA3)

Die in Abschnitt 2.3.3 vorgestellten Angriffsverfahren, insbesondere Hill-Climbing-Verfahren, sollen innerhalb der Forschungsaufgabe 3 genauer behandelt werden. Zunächst soll untersucht werden, inwieweit Hill-Climbing-Verfahren auf das handschriftenbasierte Verifikationsverfahren [Viel06] adaptiert werden können. Darauf aufbauend soll ein bekannter ausgewählter Hill-Climbing-Algorithmus [GaFO07] implementiert und evaluiert werden. Weiterhin werden aufbauend auf den Ergebnissen der Evaluation Verbesserungsvorschläge für den Verifikationsalgorithmus gegeben.

Teilaufgabe A:
Untersuchen, inwieweit Hill-Climbing-Angriffsverfahren auf den handschriftenbasierten Verifikationsalgorithmus [Viel06] adaptiert werden können.

Teilaufgabe B:
Formulieren von Verbesserungsvorschlägen für den handschriftenbasierten Verifikationsalgorithmus [Viel06] zur Vorbeugung Hill-Climbing basierter Angriffsverfahren.

4 Grundlagen

Zur Schaffung einer soliden Grundlage und zum besseren Verständnis der weiterführenden Arbeit werden nachfolgend relevante Grundlagen im Bereich der Biometrie, Kryptologie sowie wesentliche Unterschiede zwischen kryptographischen und biometrischen Hash vermittelt. Einige Themenbereiche werden im späteren Teil der Arbeit herangezogen, um angefertigte Designvorschläge zu diskutieren. Des Weiteren wird der in dieser Arbeit verwendete Verifikationsalgorithmus für die dynamische Handschrift und dessen Funktionsweise in Abschnitt 4.2 detailliert beschrieben.

Die folgenden Abschnitte (4.1 bis 4.4) dieser Arbeit vermitteln Grundlagenwissen, welches im Bereich der Informatik und der IT-Sicherheit bereits als allgemeines Wissen bekannt ist. Die dargestellten Sachverhalte werden teilweise nahe am Originaltext der referenzierten Quellen erläutert. Der Autor erhebt hierbei keinen Anspruch darauf, Urheber dieses geistigen bzw. intellektuellen Eigentums/Wissens zu sein. Zu den hier vorgestellten Themenbereichen existiert eine Vielzahl an weiterer Fachliteratur, es wird bewusst nur ein Auszug dargestellt, um dem geneigten Leser eine kurze Einführung zu geben. Es wird kein Anspruch auf Vollständigkeit erhoben. Die Auswahl der hier aufgeführten Literatur ist u.a. dadurch begründet, dass der Autor Zugang zu dieser Literatur hatte und/oder während der fachlichen Ausbildung (z.B. in Vorlesungen) sowie im Laufe des wissenschaftlichen Arbeitens (z.B. Konferenzen) diese Quellen verwendet wurden.

4.1 Biometrie

Der Begriff Biometrie setzt sich aus den zwei griechischen Wörtern bios (Leben) und metron (Maß) zusammen. Biometrie beschäftigt sich demnach mit der Messung an Lebewesen. Das Wissensgebiet der Biometrie wird in zwei große Teile gegliedert, zum einen in die biometrische Statistik und zum anderen in biometrische Erkennungsverfahren, siehe auch [BeRo01]. Diese Arbeit beschäftigt sich ausschließlich mit Letzterem.
In der Biometrie beschreiben biometrische Daten physikalisch-körperliche Eigenschaften von Personen wie zum Beispiel die Körpergröße, Unterarmlänge und Haarfarbe. Alle körperlichen Eigenschaften einer Person die physikalisch erfasst (gemessen) werden können, sind biometrische Eigenschaften bzw. Charakteristiken. Menschen und Lebewesen im Allgemeinen unterscheiden sich durch die Ausprägung ihrer biometrischen Charakteristik voneinander [BeRo01].

4.1.1 Biometrische Modalitäten

Biometrische Modalitäten sind Körper- oder Verhaltenscharakteristika, die sich unterscheiden lassen in aktive (verhaltensbasiert/dynamisch) oder passive (physiologiebasiert/statisch) Merkmale. Zu den langfristig stabilen verhaltensbasierten Merkmalen zählen beispielsweise die Stimme, die Hand- oder Unterschrift, Tippverhalten beim Schreiben auf der Tastatur und die Gangdynamik. Wohingegen der Fingerabdruck, die Iris (Auge) oder die Handgeometrie beispielsweise zu den langfristig stabilen physiologischen Merkmalen zählen [Viel06]. In Abbildung 9 wird die Einteilung der verschiedenen biometrischen Modalitäten dargestellt.

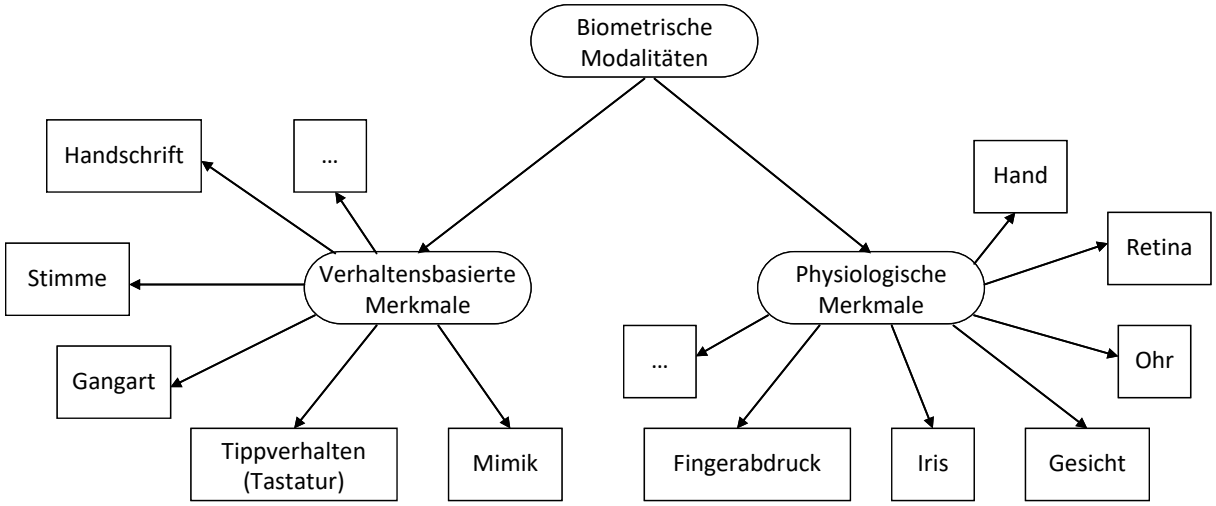


Abbildung 9 Biometrische Modalitäten, übersetzt und adaptiert von [Viel06]

Für eine zuverlässige Erkennung einzelner Individuen sollten nach Behrens und Roth ([BeRo01]) biometrische Modalitäten folgende Kriterien erfüllen:

- *Konstanz*: Alter und Zeitpunkt der Messung beeinflussen den Messwert nicht.
- *Messbarkeit*: Eine gut definierte Messgröße sollte existieren, welche von einem geeigneten Sensor erfasst werden kann.
- *Universalität*: Das Merkmal kommt bei möglichst vielen Personen vor.
- *Einmaligkeit*: Das Merkmal ist für möglichst alle Personen unterschiedlich.
- *Akzeptanz*: Das Merkmal sollte von möglichst vielen Personen als Authentifizierungsmerkmal anerkannt und akzeptiert sein.

4.1.2 Biometrisches Erkennungssystem

Ein biometrisches Erkennungssystem ist nach Jain et al. ([JaFR08]) im Wesentlichen ein Mustererkennungssystem, welches (1) biometrische Daten eines Individuums erfasst, (2) hervorstechende Merkmale aus den biometrischen Daten extrahiert, (3, 4) diese Merkmale mit Referenzmerkmalen in einer Datenbank vergleicht/abgleicht und auf Basis der Entscheidung eine Aktion anstößt bzw. auslöst. Demnach kann ein biometrisches System in vier Basismodule (Sensor, Merkmalsextraktor, Vergleicher/Entscheider und Datenbank) eingeteilt bzw. aufgebaut werden. Nachfolgend werden die jeweiligen Module im Überblick nach Jain et al. beschrieben [JaFR08].

Sensor

Der Sensor erfasst die biometrische Modalität einer Person und wandelt diese in elektrische Signale (biometrische Rohdaten) um. Welche Art Sensor eingesetzt wird, hängt stark von der verwendeten biometrischen Modalität ab. Bei der Spracherkennung wäre ein Mikrofon beispielsweise das Sensorelement, welches akustische Signale (Schallwellen) in elektrische Signale umwandelt (Analog/Digital-Wandlung). Der Sensor bildet die Schnittstelle zwischen Mensch und Maschine und spielt eine Schlüsselrolle in der Performanz eines biometrischen Erkennungssystems. Schlecht gewählte oder falsch eingesetzte Sensoren können zu einer schlechten Aufnahme der biometrischen Modalität und ggf. zu einer wiederholten Eingabe bzw. Präsentation der Modalität führen. Somit sinkt in Konsequenz auch die Akzeptanz eines Erkennungssystems.

Merkmalsextraktor
Merkmalsextraktoren besitzen in der Regel eine Signalvorverarbeitung um Störsignale oder für die Modalität nicht benötigte Informationen im Vorfeld herauszufiltern. Hierfür sind oftmals komplexe Algorithmen notwendig, welche die biometrischen Rohdaten für die eigentliche Merkmalsextraktion aufbereiten. Bei der Spracherkennung können zum Beispiel durch Frequenzfilter Hintergrundgeräusche minimiert oder gar komplett eliminiert werden. Bei der eigentlichen Merkmalsextraktion werden für die biometrische Modalität charakteristische Merkmale extrahiert. Bei dem bereits erwähnten Beispiel der Spracherkennungen wäre die maximale oder minimale Frequenz beispielsweise ein solches Merkmal. In der Regel werden mehrere Merkmale aus den Rohdaten extrahiert und in einem Datensatz zusammengefasst (z.B. Merkmalsvektor). Dieser Datensatz wird in der Einlernphase (siehe weiter unten) in einer Datenbank hinterlegt und dient als Referenzdatensatz (biometrisches Template).

Vergleicher/Entscheider
Nach der Merkmalsextraktion wird bei einer Verifikation der aktuelle Datensatz (z.B. Merkmalsvektor) mit dem Referenzdatensatz verglichen. Das Resultat ist ein sogenannter Vergleichswert (Matching-Score) zwischen aktuellem und gespeichertem Datensatz. In Abhängigkeit eines festgelegten Schwellenwertes wird anschließend eine Entscheidung getroffen, Verifikation erfolgreich oder nicht erfolgreich. Das Entscheidungsmodul ist in der Regel in den Vergleich mit eingebettet, sie bilden zusammen eine Einheit/Modul.

Datenbank (Datenspeicherung)
Das Datenbankmodul ist der Aufbewahrungsort des biometrischen Referenzdatensatzes. Während der Einlernphase werden die Referenzdaten in die Datenbank, zusammen mit weiteren Daten (Name, Adresse, PIN usw.), gespeichert. Bei der Verifikation werden die gespeicherten Referenzdaten abgerufen. Referenzdaten müssen nicht zwangsläufig in einer Datenbank gespeichert werden, sondern können beispielsweise auch auf einer Smart-Card oder einem mobilen Endgerät (Smartphone) abgelegt werden (dezentrale Speicherung).

Abbildung 10 zeigt den Aufbau eines einfachen biometrischen Erkennungssystems mit den vier Grundmodulen Sensor, Signalvorverarbeitung, Merkmalsextraktor, Vergleicher/Entscheider und Datenbank.

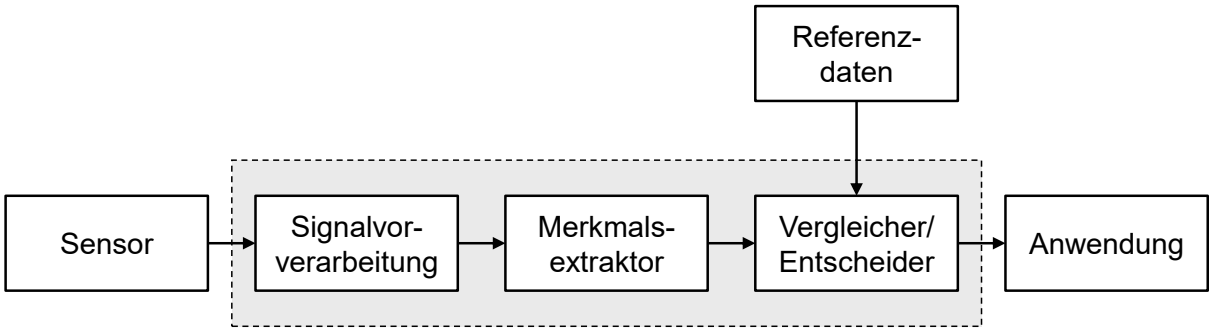


Abbildung 10 Grundkomponenten eines biometrischen Erkennungssystems, übersetzt und adaptiert von [Viel06]

Ziel eines modernen biometrischen Erkennungssystems ist es, Personen bzw. Individuen zweifelsfrei anhand ihrer biometrischen Charakteristiken zu identifizieren. Dabei werden in einer Einlernphase (Enrollment) ein oder mehrere Merkmale einer Person erfasst und mit weiteren personenspezifischen Daten gespeichert. Diese gespeicherten Merkmale werden auch als biometrisches Template bezeichnet. Bei einem Authentifizierungsprozess werden die aktuellen biometrischen Merkmale einer Person erfasst und mit dem Template (Referenzdatum) verglichen. In Abbildung 11 werden die Einlernphase und die Authentifizierungsphase dargestellt.

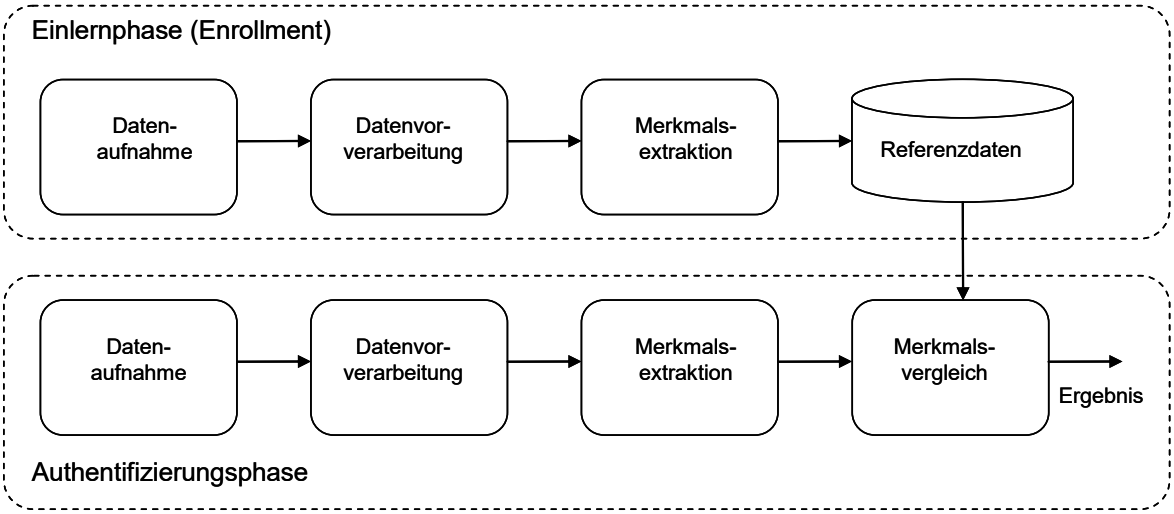


Abbildung 11 Ablaufprozess in biometrischen Erkennungssystemen, übersetzt und adaptiert von [JaNN08]

4.1.3 Verifikation und Identifikation

Im Ablaufprozess der Biometrie bzw. biometrischer Erkennungssysteme existiert neben der Einlernphase die Authentifizierungsphase, welche sich generell in zwei Authentifizierungsmodi unterscheiden lässt. Zum einen die Verifikation und zum anderen die Identifikation. Diese beiden Modi lassen sich wie folgt beschreiben:

Identifikation
Bei einer Identifikation wird der aktuell präsentierte Datensatz gegen alle in der biometrischen Datenbank vorhandenen Referenzdaten verglichen. Hierbei spricht man auch von einem „eins zu alle“ Vergleich (1:N). Das Resultat eines solchen Vergleichs ist eine Menge von Datensätzen, mit denen der aktuelle Datensatz Ähnlichkeiten aufweist (abhängig vom gewählten Schwellenwert). So ist die Suche nach einer bestimmten Person anhand eines Fingerabdrucks in einer kriminalistischen Datenbank ein typisches Beispiel für eine Identifikation. Hier wird vom System verlangt, dass alle Personen gefunden werden, die eine hohe Übereinstimmung zu diesem Fingerabdruck aufweisen. Im idealen Fall würde nur eine Person für den Fingerabdruck ermittelt werden, siehe dazu u.a. [ChBl03].

Verifikation
Bei der Verifikation wird der aktuelle Datensatz nur mit einem Referenzdatensatz verglichen. Ein solcher Vergleich wird auch als „eins zu eins“-Vergleich bezeichnet (1:1). Das Ergebnis hierbei ist entweder eine Ja/Nein – Antwort oder ein Vergleichswert, dies ist abhängig von der Konfiguration des Systems. Es wird also nur der eine Referenzdatensatz

einer Person zum Vergleich mit dem aktuellen Datensatz der vermeintlich befugten Person herangezogen, siehe [ChBl03].

4.1.4 Biometrische Fehlerraten

Wie im Abschnitt 1.2 erläutert, unterliegen biometrische Daten einer Unschärfe u.a. aufgrund natürlicher physiologischer Schwankungen und gesundheitlicher Verfassung einer Person. Die biometrische Modalität und deren spezifische Eigenschaften ändern sich bei einem Individuum, hierbei spricht man auch von der Intraklassen-Variabilität. Jain et al. beschreiben in ihrer Arbeit [JaRP04], dass sich bestimmte Eigenschaften einer biometrischen Modalität einer Person denen einer anderen Person ähneln können. Letzteres beschreibt die Interklassen-Variabilität (Interklassen-Ähnlichkeit). In Abbildung 12 wird beispielhafte die Häufigkeitsverteilungen einer bestimmten Eigenschaft (biometrisches Merkmal) einer Modalität für eine Person und deren Überlappung mit denen anderer Person dargestellt. Der linke Graf zeigt beispielhaft die Wahrscheinlichkeitsverteilung eines Merkmalswertes des Merkmals n_i einer authentischen Person (Intra-Klasse). Wohingegen der rechte Graf exemplarisch die hypothetische Verteilung des Merkmalswertes des gleichen Merkmals für alle übrigen, nicht-authentischen Personen darstellt (Inter-Klasse). Im Bereich der Überlappung kann eine Falschklassifizierung des Merkmals stattfinden, welches das charakteristische Klassifizierungsproblem in der Biometrie beschreibt.

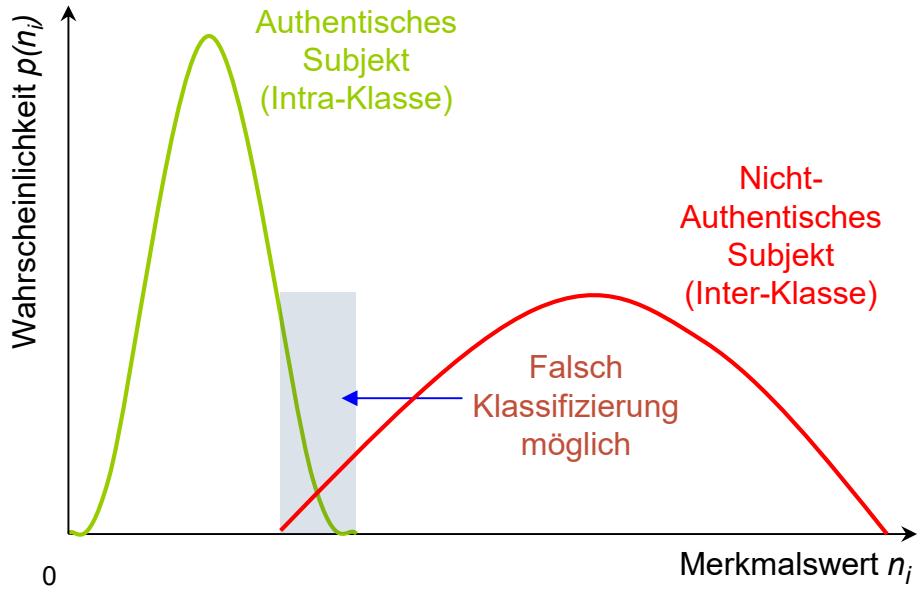


Abbildung 12 Variabilität und Trennschärfe eines biometrischen Merkmals, übersetzt von [Viel06]

Aufgrund der möglichen Falschklassifizierung können Falschakzeptanzen und Falschrückweisungen entstehen. Um die Arbeitsweise biometrischer Systeme besser optimieren und vergleichen zu können, bedarf es bestimmter Messgrößen. In der Regel werden die Erkennungs- und Zurückweisungsraten eines Systems ermittelt, um auf Basis dieser Werte eine Qualitätsaussage treffen zu können. Hierfür wird die sogenannte Falschakzeptanzrate (FAR) und Falschrückweisungsrate (FRR) eines Systems für diese Zwecke ermittelt, wobei die FAR die relative Häufigkeit bzw. Wahrscheinlichkeit angibt, mit der ein Zugang zum System gewährt wurde, obwohl diese Person keine Zugangsberechtigung hat (siehe Formel 1).

$$FAR = \frac{Anzahl\ fälschlicher\ Akzeptanzen}{Gesamtanzahl\ unberechtigter\ Zutrittsversuche}$$ **Formel 1**

Die FRR beschreibt die Häufigkeit, mit der eine Person irrtümlicherweise vom System zurückgewiesen wurde, siehe Formel 2. Die FRR wird auch als Komfortmerkmal eines biometrischen Systems bezeichnet, da eine irrtümliche Zurückweisung lediglich lästig, aber nicht sicherheitsrelevant ist.

$$FRR = \frac{Anzahl\ fälschlicher\ Rückweisungen}{Gesamtanzahl\ berechtigter\ Zutrittsversuche}$$ **Formel 2**

Weitere Messgrößen, die bei der Beschreibung biometrischer Systeme verwendet werden, sind die *false non match rate* (FNMR) und *false match rate* (FMR). Ähnlich wie bei der FRR gibt die FNMR die Häufigkeit an, mit der eine Person vom System irrtümlicherweise zurückgewiesen wurde. Jedoch werden die fehlerhaften Rückweisungen aufgrund schlechter Aufnahmedaten (z.B. minderwertige Bildqualität) nicht berücksichtigt. Eine geringe Qualität der Aufnahmedaten kann zum Beispiel dadurch entstehen, dass ein Sensor nicht korrekt arbeitet oder aber auch durch schlechte Aufnahmebedingungen (z.B. schlechte Ton- oder Lichtverhältnisse). Gleiches gilt für die FMR, welche ähnlich wie die FAR auch die Wahrscheinlichkeit angibt, mit der ein Zugang zu einem System gewährt wurde, obwohl keine Zugangsberechtigung vorlag (siehe dazu auch [Waym1999]).

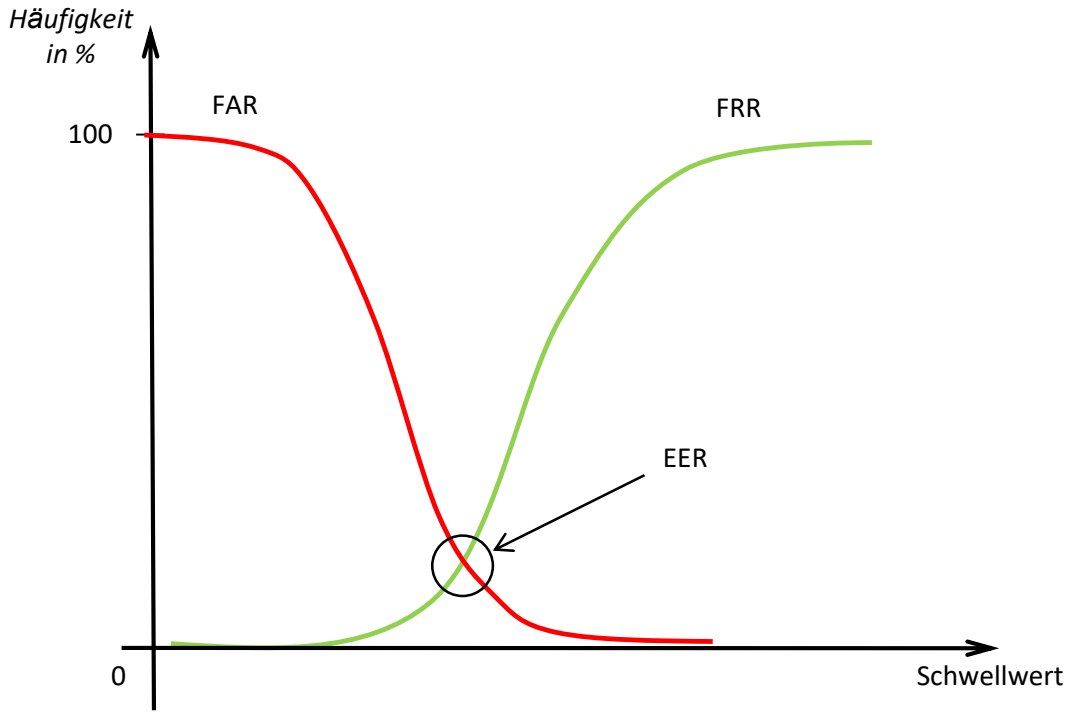


Abbildung 13 FAR - FRR-Diagramm, beispielhafte Darstellung

FRR und FAR hängen grundsätzlich von einem Schwellenwert ab, welcher angibt, wie groß der Unterschied zu einem zu verifizierenden Muster und einem gespeicherten Referenzmuster sein darf, damit dieses akzeptiert bzw. zurückgewiesen wird. Je höher dieser Schwellenwert (Sensibilitätswert) gesetzt wird, desto mehr Muster werden akzeptiert und die FAR steigt bzw. die FRR sinkt. Setzt man einen niedrigen Schwellenwert, sinkt die FAR

und die FRR steigt (Trade-off-Problem). Wählt man einen Schwellenwert so, dass die Werte von FAR und FRR gleich sind, so erhält man die Gleichfehlerrate oder auch equal error rate (EER). Die EER wird oft als aussagekräftiges Maß für die Güte eines biometrischen Systems angesehen. Es beschreibt jedoch nicht den optimalen Arbeitspunkt eines biometrischen Systems. In dieser Arbeit werden die FAR, FRR und EER verwendet, welche auch in der Biometrie weitestgehend gebräuchlich sind. In Abbildung 13 wird beispielhaft grafisch veranschaulicht, wie FAR, FRR und EER zusammenhängen.
Die FAR und FRR beschreiben zwei gegenläufige voneinander abhängige Fehlerraten. Das heißt, die Optimierung der einen führt zur Verschlechterung der andern und umgekehrt. Diese Eigenschaft der biometrischen Fehlerraten wird in Abschnitt 4.1.6 (Herausforderungen in der Biometrie) genauer betrachtet.

Des Weiteren stellt die ROC (Receiver Operating Characteristic) – Kurve eine in der Biometrie weit verbreitete Möglichkeit zur Bewertung biometrischer Erkennungssysteme dar, so Jain et al. in [JaRP04]. Hier werden im Gegensatz zum FAR – FRR-Diagramm, die Richtig-Positiv-Rate (1-FRR) auf der Ordinate und die Falschakzeptanzrate (FAR) auf der Abszisse für alle gemessenen Schwellenwerte aufgetragen. In Abbildung 14 wird der typische Verlauf einer ROC – Kurve für ein biometrisches Erkennungssystem veranschaulicht. Wobei in der Regel ein Kurvenverlauf angestrebt wird, der beginnend bei [0,0] möglichst steil senkrecht ansteigt und nahe der 100 Prozent Marke (Richtig-Positiv-Rate) waagerecht nach rechts [100, 100] verläuft. Die ROC – Kurve findet nicht nur in der Biometrie Verwendung, sondern wird unter anderem auch bei Zwei-Klassen-Klassifizierungsproblem im Allgemeinen eingesetzt.

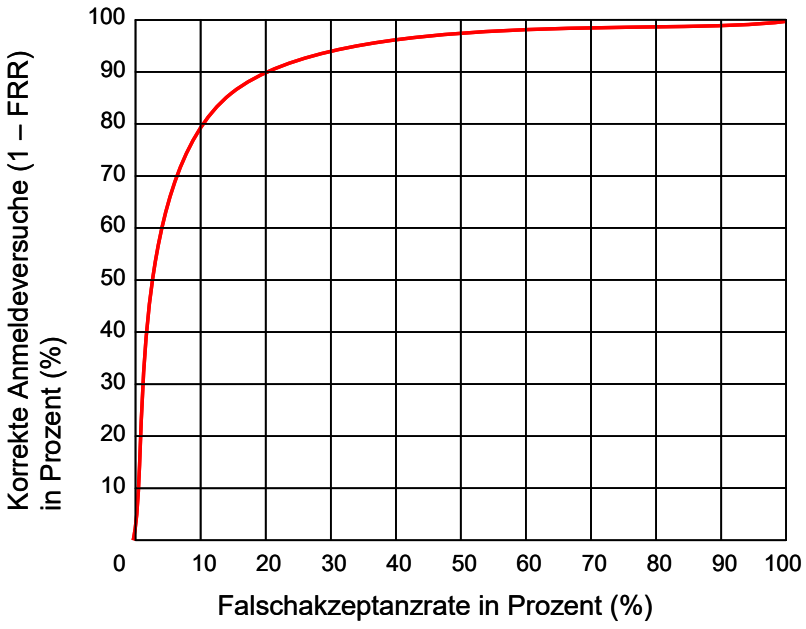


Abbildung 14 Receiver Operating Characteristic (ROC) - Kurve, beispielhafte Darstellung

Die Fehlerraten eines biometrischen Systems können nicht berechnet, sondern nur experimentell über Messungen bestimmt werden. Dabei gilt, je mehr Testwerte innerhalb einer Messung verwendet wurden, desto aussagekräftiger sind die Fehlerraten bezüglich der Güte eines Erkennungssystems.

Kollisions- und Reproduktionsrate
Für biometrische Erkennungsalgorithmen, die sogenannte biometrische Hashwerte (siehe Abschnitt 4.4) erzeugen, können weitere Fehlerraten zur Performanzbestimmung herangezogen werden. Zusätzlich zu den Fehlerraten FAR, FRR und EER sind die Reproduktions- und Kollisionsrate (RR/CR) nützliche Indikatoren für biometrische Erkennungssysteme. Sie geben an, wie häufig identische biometrische Hashwerte reproduziert werden konnten, wobei die Kollisionen durch Angriffsdaten und Reproduktionen durch authentische Daten verursacht werden. Da sich beide Raten wechselseitig beeinflussen, Optimieren der einen führt in der Regel zum Verschlechtern der anderen, kann eine Kollisionsreproduktionsrate (CRR) als zusätzliches Qualitätsmaß zum besseren Vergleich angegeben werden. In der Arbeit von Scheidat et al. [ScVD08] werden die Kollisions-, Reproduktions- und Kollisionsreproduktionsrate detailliert beschrieben. Die Kollisionsreproduktionsrate kann nach Ermittlung der Reproduktions- und Kollisionsraten wie in Formel 3 dargestellt berechnet werden [ScVD08]:

$$CRR = \frac{1}{2}(CR + (1 - RR)) \qquad \textbf{Formel 3}$$

4.1.5 Doddingtons Zoo

In Abschnitt 1.2 wird bereits erläutert, dass biometrische Charakteristiken schwanken und somit biometrische Daten einer gewissen Unschärfe unterliegen. Doddington et al. haben in [DLM+98] vier Personengruppen eines biometrischen Erkennungssystems identifiziert, welche in Abhängigkeit ihrer biometrischen Schwankungen unterschiedlich Auswirkungen zeigen. Diese Erkenntnisse haben sie in Zusammenhang mit der Sprechererkennung erlangt, sie können jedoch auch auf weitere biometrische Modalitäten angewendet werden. Die verschiedenen Personengruppen eines biometrischen Systems, auch bekannt als Doddingtons Zoo, sind wie folgt definiert, sieh auch [JaFR08]:

Schaf
Schafe repräsentieren Individuen mit sehr markanten und unverwechselbaren biometrischen Merkmalen und zeigen eine geringe Intraklassen-Variabilität. Dementsprechend weisen Schafe eine geringe Falschakzeptanz- und Falschrückweisungsrate auf.

Ziege
Individuen, welche häufig Falschrückweisungen hervorrufen, werden von Doddington et al. als Ziegen bezeichnet. Biometrische Merkmale solcher Individuen weisen ein hohes Maß an Intraklassen-Variabilität auf.

Lamm
Die biometrischen Merkmale eines Lamms überschneiden sich ausgiebig mit biometrischen Merkmalen anderer Individuen. Dementsprechend weisen die biometrischen Merkmale solcher Individuen eine hohe Interklassen-Variabilität auf. Sie sind somit anfälliger für Falschakzeptanzen als andere Individuen.

Wolf
Individuen, die ihre biometrischen Charakteristiken manipulieren können, um andere Individuen des Systems zu imitieren, werden als Wölfe bezeichnet. Dies gilt speziell für dynamische biometrische Charakteristiken wie z.B. die Sprache. Wölfe steigern demzufolge die Falschakzeptanzrate eines Erkennungssystems.

Die in der Arbeit von Doddington et al. vorgestellten Personengruppen beschreiben verschiedenen Typen von Individuen, die in einem biometrischen System auftreten können. Wenn die Individuen eines biometrischen Systems einer der Personengruppen zugeordnet werden, können ggf. Maßnahmen getroffen werden, die Auswirkungen auf die Erkennungsperformanz des Systems zu verbessern. So könnten bspw. Schwellenwerte für bestimmte Individuen angepasst werden, um deren Erkennungsperformanz zu steigern. Das Erreichen einer guten Erkennungsperformanz eines biometrischen Systems stellt u.a. eine Herausforderung in der Biometrie dar. Im nachfolgenden Abschnitt 4.1.6 werden diese und weitere Herausforderungen der Biometrie zusammengefasst.

4.1.6 Herausforderungen in der Biometrie

Die im Abschnitt 2.1.1 von Maltoni et al. in [MMJP03] vorgestellten vier Anforderungen an ein ideales biometrisches Erkennungssystem stellen unter anderem auch die Herausforderungen in der Biometrie dar. Diese vier oben genannten Anforderungen sind Vielfalt (Diversity), Widerrufbarkeit (Revocability), Sicherheit (Security) und Erkennungsperformanz (Performance). Nachfolgend werden die jeweiligen Anforderungen kurz beschrieben und mögliche Techniken/Verfahren vorgestellt, welche diese Ansprüche gerecht werden können.

Vielfalt (Diversity)

Anforderung:	Es darf nicht möglich sein, dass ein geschütztes biometrisches Datum für verschiedene Referenzdatenbanken als Kreuzprobe verwendet werden kann (cross matching).
Verfahren:	Die Speicherung von biometrischen Referenzdaten innerhalb einer sicheren Umgebung wie beispielsweise einer Smartcard senkt den Missbrauch der Referenzdaten alleine schon durch den Umstand, dass die Daten nicht ohne weiteres aus einer Datenbank gestohlen werden können. Eine dezentrale Datenhaltung kann dementsprechend dazu beitragen, *cross matching* zu verhindern. Des Weiteren können auch Matching-on-Card (MoC) Systeme, wie beispielsweise in [StSc02], [HeFr04] und [CAP+06] vorgestellt, eingesetzt werden, um den Vergleich der biometrischen Daten auf einer Smartcard durchzuführen. So wird das *cross matching* ebenfalls erheblich erschwert. Verfahren, die es erlauben, eine Vielzahl von biometrischen Referenzdaten auf Basis einer biometrischen Modalität eines Individuums zu generieren, können verwendet werden, um für jede Anwendung ein spezielles Referenzdatum zu erstellen, siehe auch *Widerrufbarkeit.* So kann die Verwendung einer Kreuzprobe ebenfalls ausgeschlossen werden.

Widerrufbarkeit (Revocability)

Anforderung:	Es muss möglich sein, ein kompromittiertes biometrisches Referenzdatum jederzeit unkompliziert widerrufen zu können. Weiterhin sollte ein neues Referenzdatum auf Basis der gleichen biometrischen Daten erstellt werden können.
Verfahren:	Die in Abschnitt 2.1.1 erwähnten Verfahren zur Sicherung von biometrischen Referenzdaten mittels Merkmalstransformation (Salzen und unumkehrbare Transformation) eignen sich, um beliebig viele biometrische Referenzdaten anhand einer Modalität eines Individuums zu erzeugen. So kann ein kompromittiertes Referenzdatum widerrufen und ein neues Datum auf Basis derselben biometrischen Modalität erzeugt werden. Weiterhin sind biometrische Kryptosysteme (Schlüsselbindung und Schlüsselgenerierung) in der Lage den Anforderungen der Widerrufbarkeit gerecht zu werden.

Sicherheit (Security)

Anforderung:	Es muss rechnerisch nahezu unmöglich sein, biometrische Daten anhand des biometrischen Templates zu ermitteln. Auf diese Weise soll verhindert werden, dass ein potentieller Angreifer eine Fälschung auf Basis eines gestohlenen Templates erstellt oder personenbezogene Informationen daraus ableitet.
Verfahren:	Die in Abschnitt 2.1.1 vorgestellten Mechanismen zum Schutz der Referenzdaten (Biometrische Kryposysteme/Merkmals Transformation) bieten entsprechende Möglichkeiten, den Anforderungen der Sicherheit zu entsprechen und der potentiellen Rückführbarkeit von biometrischen Daten zu minimieren bzw. zu verhindern.

Erkennungsperformanz (Performance)

Anforderung:	Der Schutzmechanismus (Template Protection) sollte die Erkennungsperformanz (FAR und FRR) nicht negativ beeinflussen.
Verfahren:	Einer der wichtigsten Punkte ist die Beibehaltung bzw. Verbesserung der Erkennungsperformanz mit Einführung von Schutzmechanismen. Eine Möglichkeit, die Performanz zu steigern, ist der Einsatz von Multimodalen biometrischen Verfahren, also die Kombination von zwei oder mehreren biometrischen Modalitäten, zum Beispiel Fingerabdruck und Iris [NaJa07] oder Gesicht, Stimme und Körpergewicht [SBD+09]. Biometrische Fusion kann nicht nur auf der Ebene verschiedener Modalitäten durchgeführt werden. Die Fusion verschiedener Verifikationsmechanismen auf einer Modalität ist ein weiteres Beispiel für die biometrische Fusionen. In [JaFR08] (Seite 271 ff.) beschreiben Jain et al. mögliche Fusionsmöglichkeiten im Allgemein.

Zusätzlich zu den oben genannten Anforderungen an ein ideales biometrisches Erkennungssystem existieren weitere Herausforderungen in der Biometrie. Nachfolgend werden weitere ausgewählte Probleme kurz beschrieben:

Trade-Off Problem

Das in Abschnitt 4.1.4 beschriebene Trade-Off Problem beschreibt das Zusammenspiel der Falschakzeptanz- und Falschrückweisungsrate. Wird durch technische, algorithmische oder sonstige Maßnahmen versucht, eine der Fehlerraten zu minimieren, führt dies in der Regel zur Steigerung der anderen Fehlerrate. Das ultimative Ziel von Entwicklern biometrischer Erkennungssysteme ist es, eine Gleichfehlerrate (EER) von Null zu erzielen, um somit eine klare Trennung aller registrierten Individuen eines Systems zu erreichen.

In [TeNG04] wurden von Teoh et al. eine personenspezifische Authentifizierungsmethode vorgestellt (BioHashing), welche biometrische Merkmale mit sogenannten Tokenized (pseudo-) random numbers (TRN) kombiniert. Aufbauend auf dieser Arbeit wurden von derselben Forschungsgruppe für die Modalitäten Fingerabdruck ([TeNG04]), Gesicht ([NgTG04], [TeNG04a], [TeNG04b], [TeNg05]) und Handflächenabdruck ([CTGN04], [PaTN04], [CTGN05]) Authentifizierungsmethoden vorgestellt, welche das in [TeNG04] eingeführte Verfahren verwenden und teilweise eine Gleichfehlerrate (EER) von Null erreichen. Kong et al. bzw. Cheung et al. haben wiederum in [KCZ+06] respektive [CKZ+06] wissenschaftliche Ergebnisse veröffentlicht, welche die oben genannten Arbeiten der Forschungsgruppe um Andrew Teoh und deren Ergebnisse widerlegen. Es wird vor allem darauf hingewiesen, dass das Verfahren auf versteckten bzw. unpraktischen Voraussetzungen und Annahmen beruht. Ferner wird von Kong et al. gezeigt, dass die oben genannten BioHashing Methoden keine Möglichkeit besitzen, kompromittierte Referenzdaten zu widerrufen und zu ersetzen.

Dieses kleine Beispiel innerhalb der Biometrie Gemeinschaft zeigt, dass Verfahren zur eindeutigen Klassifizierung nur sehr schwer, wenn nicht sogar unmöglich zu realisieren sind.

Sammeln biometrischer Testdaten

Eine Herausforderung, die nicht unmittelbar im Betrieb, jedoch in der Entwicklung von biometrischen Erkennungssystemen auftritt, ist das Sammeln von Testdaten. Testdatenbanken werden speziell bei einem Entwurf und bei einem Vergleich mit anderen Systemen herangezogen. Bei der Sammlung von biometrischen Daten treten nicht nur logistische Schwierigkeiten auf. Die Anonymisierung der Daten inklusive Mechanismen zur individuellen Löschung sind genauso zu berücksichtigen, wie das kontrollierte Aufzeichnen der Daten. Des Weiteren sollte für jede Modalität ermittelt werden, welche zusätzlichen Daten (Geschlecht, Alter, Größe, Gewicht, Augenfarbe, trägt oder trägt keine Kontaktlinsen, Rechts-/Linkshänder usw.) der Individuen aufgenommen werden sollen. Die Anzahl der Testsamples und der zeitliche Abstand der Datenaufzeichnung sind je nach Anforderung ebenfalls entscheidend für eine Testdatenbank. In [JaFR08] (Seite 529 ff.) wird von Jain et al. genauer spezifiziert, welche Anforderungen eine biometrische Testdatenbank haben sollte. Das Anlegen einer biometrischen Testdatenbank ist unter Verwendung von Hilfestellungen, wie sie in [JaFR08] dargestellt werden, grundsätzlich realisierbar. Dennoch sind vor allem rechtliche Gegebenheiten (Privatsphäre, Datenschutz etc.), speziell auf nationaler Ebene zu prüfen. So sind beispielsweise mit der Einführung der Datenschutz-Grundverordnung im Jahr 2018 (DSGVO) u.a. einheitliche Vorgaben bzgl. der Verarbeitung personenbezogener Daten EU-weit eingeführt worden. In der Arbeit [WhDV18] von Whiskerd et al. sind u.a. die Auswirkungen der DSGVO auf die Verarbeitung von biometrischen Daten adressiert worden.

Alterung biometrischer Referenzdaten
Die Erkennungsperformanz eines biometrischen Systems hängt nicht nur von der aktuellen körperlichen Verfassung einer Person (Krankheit, klamme Finger, usw.), der Interaktion mit dem Sensor (z.B. unübliche Mimik) oder den technischen Gegebenheiten (fehlerhafter Sensor, unzureichende Beleuchtung, etc.) ab, sondern unter anderem auch von den Alterungserscheinungen. Wobei sich die natürliche Alterung auf einige Modalitäten mehr auswirkt, als auf andere. So sind beispielsweise die Modalitäten Gesicht, Gangart und Unterschrift alterungsbedingten Änderungen mehr ausgesetzt als zum Beispiel Iris, Fingerabdruck und Retina. In der Biometrie wird zwischen kurzzeitigen (wenige Monate bis unter einem Jahr) und langzeitigen (mehr als ein Jahr) Auswirkungen auf biometrische Modalitäten unterschieden. Auswirkungen auf die Erkennungsperformanz verschiedener Modalitäten wie Fingerabdruck ([MoEl06], [MoEH07]), Gesicht ([RaCh06], [LSRJ07]), Stimme ([DHAZ11], [HMHL12]), Unterschrift/Handschrift ([ScKV12], [GaMF13]) und Gangart ([VeNC05]) wurden bereits in einigen wissenschaftlichen Arbeiten analysiert. Zudem sind auch Lösungsvorschläge in der Biometrie-Gemeinschaft vorgestellt und teilweise getestet worden. Hier sind vor allem Strategien zur automatischen Erneuerung der Referenzdaten präsentiert worden ([Carl09], [RFMR09], [ScMV07]). Trotz dieser Lösungsvorschläge stellt die Alterung biometrischer Modalitäten ein Problem dar, da nur wenige oder keine Langzeitdatenbanken für einige Modalitäten existieren und somit eine genaue Analyse der Auswirkungen mangels Testdaten nicht durchgeführt werden kann.

4.2 Biometrischer Hash Algorithmus für Handschrift

Der in dieser Arbeit verwendete Verifikationsalgorithmus basiert auf der biometrischen Modalität „Handschrift“, wurde von Vielhauer et al. in [ViSM02] eingeführt und unter anderem in [Viel06] erweitert bzw. weiterentwickelt. Der dort beschrieben Biometric Hash Algorithmus für die dynamische Handschrift soll in diesem Abschnitt genauer beschrieben werden, da er in dieser Arbeit innerhalb der Forschungsaufgaben FA1, FA2 und FA3 behandelt wird. Bevor der Biometric Hash Algorithmus detailliert betrachtet wird, werden Grundlagen der Arbeitsweise von Handschriftenaufzeichnungsgeräten erläutert.

4.2.1 Arbeitsweise von Handschriftaufzeichnungsgeräten

Während der Aufzeichnung einer Handschrift auf einem elektronischen Gerät, wie zum Beispiel Digitalisiertablett oder Signaturtablett, können je nach Gerät und Ausstattung mehrere Parameter zeitlich erfasst werden. Bei allen Geräten werden die X- und Y-Koordinaten des Stiftes auf dem Schreibfeld ermittelt. Zusätzlich verfügen einige Modelle über Drucksensoren. Diese Drucksensoren ermitteln den Druck, den der Stift auf einer bestimmten Koordinate ausübt. Dabei kann die Auflösung der Drucksensibilität variieren. Es gibt außerdem Handschriftenaufzeichnungsgeräte, welche die Lage des Stiftes (Azimut und Altitude) im Bezug zum Schreibfeld bestimmen können.
Mit lpi (lines per inch) oder dpi (dots per inch) gibt man die Auflösung eines Digitalisiertabletts an. Die Möglichkeit, die Lage eines Stiftes relativ zum Schreibfeld zu bestimmen, wird mit der Genauigkeit des Winkelwertes angegeben. Ist ein Digitalisiertablett mit einem Drucksensor ausgestattet, so wird der aktuelle Druckwert bestimmt. Dieser hängt von der Auflösung des Sensors ab. So gibt es zum Beispiel Drucksensoren mit einem Wertebereich von 0-127, oder aber auch Sensoren die einen Wertebereich von 0-1023 aufweisen. Einige Digitalisiertabletts registrieren lediglich, ob der Stift die Schreiboberfläche berührt oder nicht.

Analog Digital Wandlung

Unabhängig davon, wie viele verschiedene Parameter ein Aufzeichnungsgerät bestimmen kann, eines haben alle Geräte gemeinsam: die Umsetzung der realen analogen Werte in die für Rechnersysteme verarbeitbaren eingeschränkten digitalen Werte. Ein so genannter Analog-Digital-Wandler oder Analog-Digital-Umsetzer (ADU) bildet den analogen Wertebereich in einem digitalen eingeschränkten Wertebereich ab. Eine Analog-Digital-Wandlung ist auf Grund des begrenzten Wertebereiches immer verlustbehaftet. Eine A/D Wandlung besteht aus drei nacheinander ablaufenden Schritten: aus der *Abtastung*, der *Quantisierung* und der *Codierung* des analogen Eingangssignals, siehe [Wern06].

Bei der Abtastung wird das kontinuierliche (analoge) Signal in ein zeitdiskretes Signal umgewandelt. Die Abtastrate (engl. *sampling rate*) bestimmt, wie oft ein Signal pro Zeiteinheit abgetastet werden soll. Der Abstand zwischen den einzelnen Abtastzeitpunkten wird in der Regel in Millisekunden angegeben und als Abtastintervall bezeichnet. Die Abtastrate (Abtastfrequenz) muss mindestens doppelt so groß sein, wie die Frequenz des abzutastenden Signals, ansonsten treten Fehler bei der Digitalisierung auf (z.B. Alias-Effekt[1]), siehe dazu [Shan49].

Bei der Quantisierung wird die Amplitude des gerade ermittelten Abtastwertes einem bestimmten Wert des eingeschränkten digitalen Wertebereiches zugeordnet (siehe Abbildung 15). Dabei können Rundungsfehler auftreten, wenn der Amplitudenwert zwischen zwei Quantisierungsstufen liegt (Quantisierungsfehler). Die Genauigkeit eines Analog-Digital-Umsetzers hängt größtenteils von den möglichen Quantisierungsstufen ab. Ein 10 Bit ADU zum Beispiel besitzt $(2^{10}) -1$, also *1023* Stufen, wohingegen ein 8 Bit ADU nur *255* Quantisierungsstufen aufweisen kann. Dem 8 Bit ADU steht somit ein geringerer Wertebereich für die Abbildung des analogen Signals zur Verfügung.

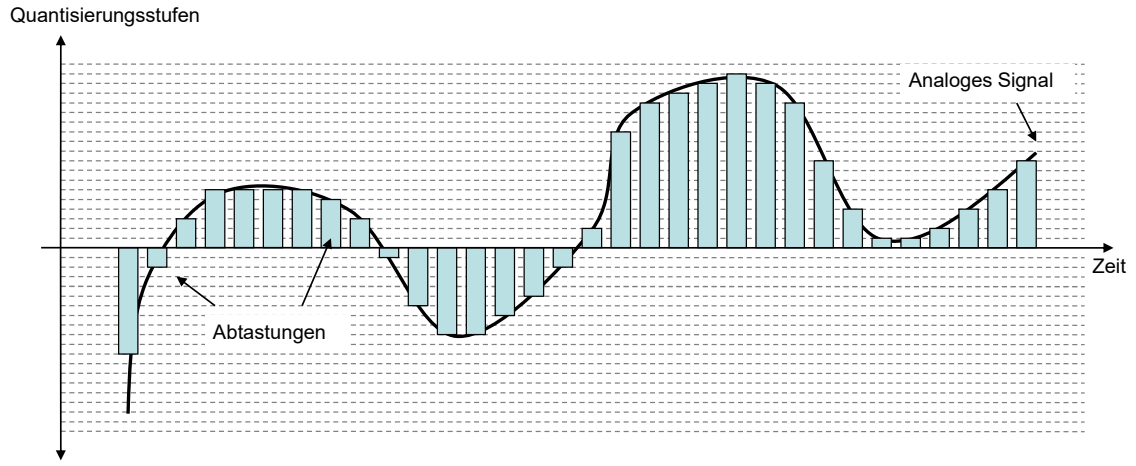


Abbildung 15 Analog/Digital – Wandlung (Digitalisierung), beispielhafte Darstellung

Nach der Quantisierung wird der Wert einer bestimmten Bitfolge zugeordnet. Die Bitfolge kann durch Codetypen repräsentiert werden, die verschiedene Eigenschaften besitzen können (Hamming-Distanz, Redundanz, etc.). Als Beispiele seien hier der Gray-Code und BCD Code genannt.

[1] Alias-Effekte treten in der Signalverarbeitung beim Digitalisieren von analogen Signalen auf. Wird das Abtasttheorem verletzt, werden Frequenzanteile, die höher als die Abtastfrequenz sind, als niedrigere Frequenzen interpretiert.

Handschriftaufzeichnungsgeräte besitzen auch einen ADU und somit auch eine Abtastrate. Diese liegt in der Regel zwischen 80 und 400 Hz, kann jedoch bei einigen Geräten auch höher sein.

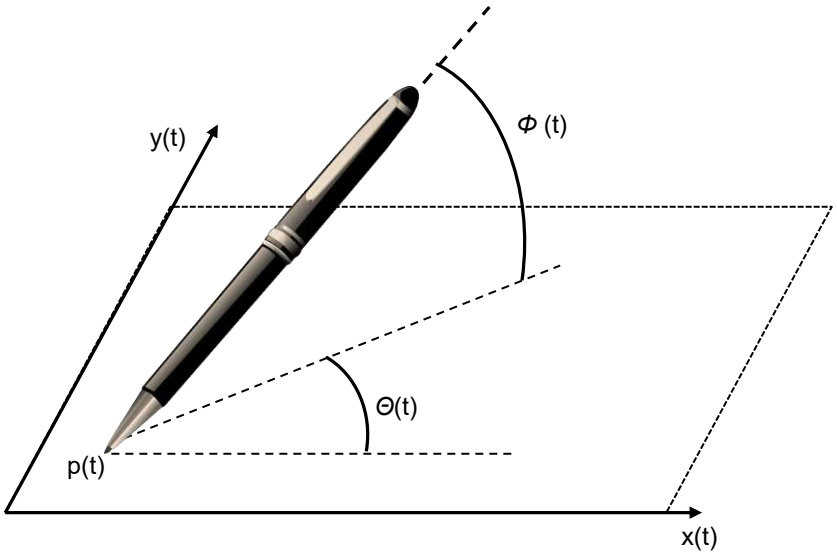


Abbildung 16 Signale, die von Signaturtabletts aufgenommen werden können, adaptiert von [Sche15]

In Abbildung 16 werden noch einmal alle typischen Signale dargestellt, welche von Signaturtabletts aufgenommen werden können. Dabei werden alle Signale zu einem bestimmten Zeitpunkt *t* erfasst und digitalisiert. Die Größe des Abstandes vom Zeitpunkt t_1 zum darauffolgendem Zeitpunkt t_2 hängt von der Abtastrate des Signaturtabletts ab. Demnach werden folgende Signale zu einem Zeitpunkt *t* erfasst: X-Koordinate *x(t)*, Y-Koordinate *y(t)*, Druck *p(t)* an der Stiftspitze, Seitenwinkel (Azimut) *Θ(t)* und Höhenwinkel (Altitude) *Φ(t)*.

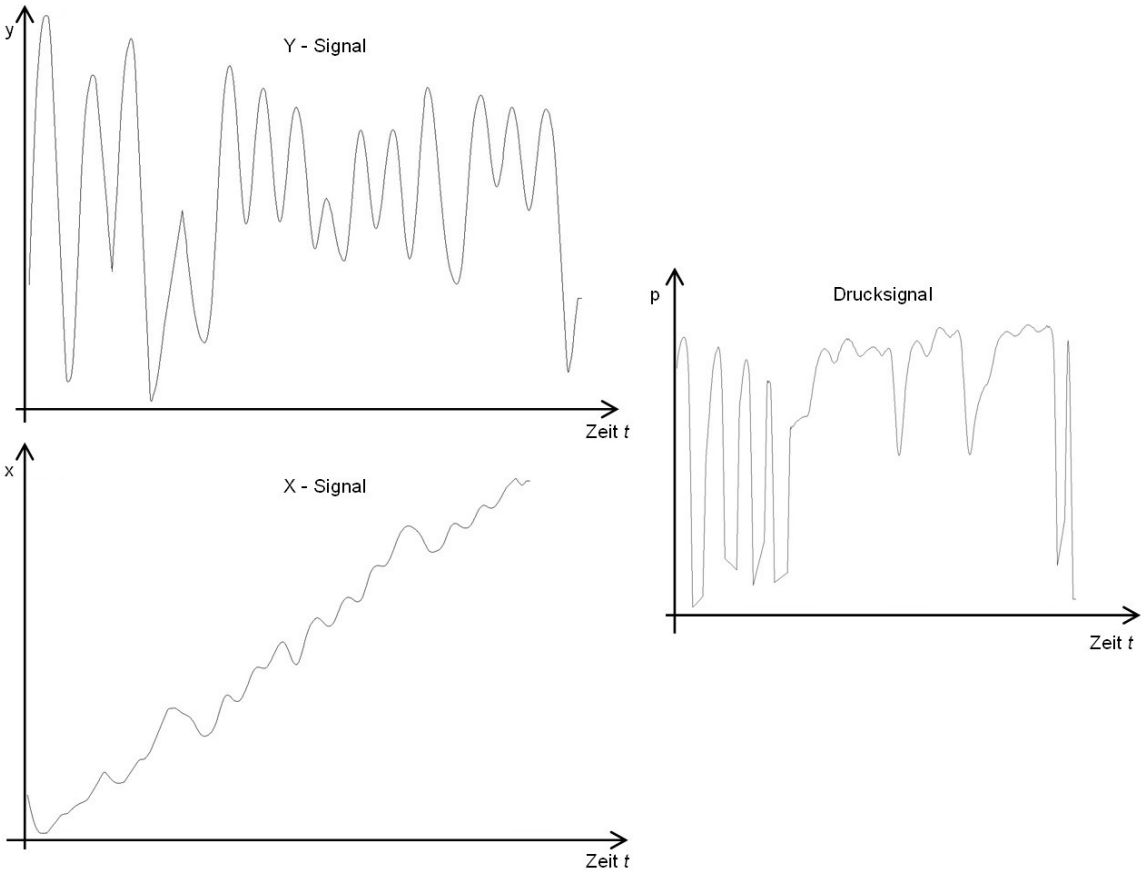


Abbildung 17 X, Y, und P Signal eines digitalisierten Schriftzuges, beispielhafte Darstellung

Betrachtet man sich nun einige Signale in Abhängigkeit der Zeit *t* so kann man erkennen, dass *y(t)* einen Kurvenverlauf zeigt, der relativ gleichmäßig auf und ab schwingt. Wohingegen *x(t)* einen zusätzlichen stetigen Anstieg zeigt. In Abbildung 17 sind beispielhaft typische Signalverläufe (von links nach rechts schreibend) von *x(t)*, *y(t)* und *p(t)* dargestellt.

4.2.2 Beschreibung des Algorithmus

Die Methode zur Generierung von Hashwerten auf Basis der biometrischen Modalität Handschrift wird von Vielhauer in [Viel06] ausführlich beschrieben. In diesem Abschnitt werden Teile des Verfahrens aus [Viel06] dargestellt, welche insbesondere für den späteren Verlauf der Arbeit relevant sind. Hierfür wurden teilweise Abbildungen und Formeln aus der Arbeit von Vielhauer übersetzt und übernommen. Der allgemeine Verlauf der Bio-Hash-Generierung wird in Abbildung 18 dargestellt, wobei der obere Teil (Intervallmatrix-Bestimmung) als Enrollment und der untere Teil (Hash-Generierung) als Verifikation interpretiert werden kann. Nach einer kurzen verbalen Beschreibung des Verfahrens werden anschließend die einzelnen Schritte etwas genauer betrachtet.

Kurze Beschreibung des Verfahrens

Der Biometric Hash Algorithmus berechnet einen statistischen Merkmalsvektor mit *k* Elementen, welche durch eine so genannte Interval Mapping Funktion in einen Hash-Raum transformiert werden. Die statistischen Merkmale basieren dabei auf den fünf Signalen *x(t)*, *y(t)*, *p(t)*, *Θ(t)* und *Φ(t)*. Die Grundlage des Mappings ist die Intervallmatrix, welche ein statistisches Model basierend auf Informationen der individuellen Nutzer oder Nutzergruppen darstellt. Die Intervallmatrix wird während des Enrollmentprozesses erstellt.

Der Hash-Vektor, den das Verfahren erzeugt, besitzt eine Dimensionalität von *k*. Das Ziel ist es, unter Verwendung des Hash-Vektors einen kryptografischen Schlüssel abzuleiten oder ihn zur nicht-umkehrbaren Repräsentation der biometrischen Daten in Authentifikationssystemen zu verwenden.

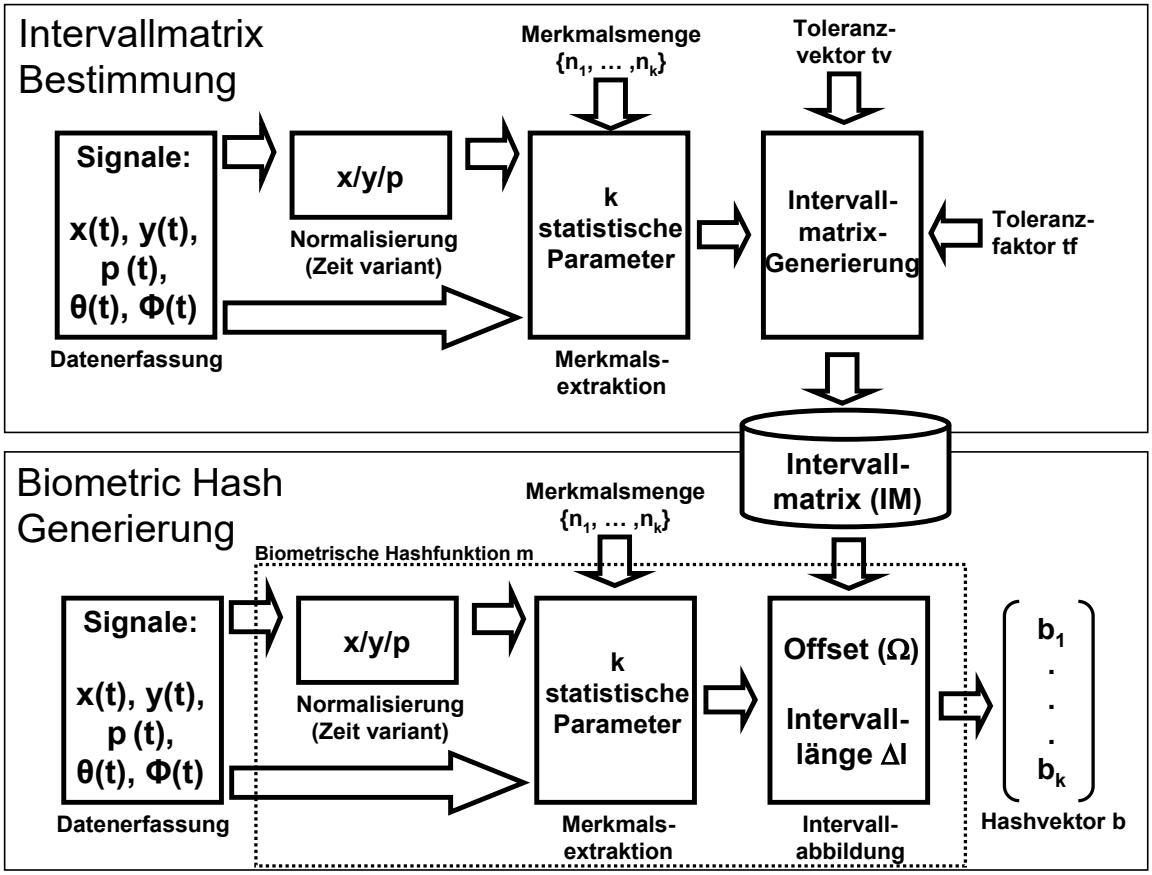


Abbildung 18 Überblick über die BioHash-Generierung, übersetzt aus [Viel06]

Enrollment (Intervallmatrix Bestimmung)

Das Ziel innerhalb des Enrollments ist es, eine Intervallmatrix (IM) zu bestimmen, welche später für die BioHash-Berechnung verwendet wird. Während der Datenerfassung werden die Signale, die von einem Signaturtablett stammen, erfasst. Die in 4.2.1 erläuterten Signale werden während der Datenerfassung ermittelt: X-Koordinate *x(t)*, Y-Koordinate *y(t)*, Druck *p(t)* an der Stiftspitze und die Winkelwerte Azimut *Θ(t)* und Altitude *Φ(t)*. Basierend auf diesen fünf Signalen werden statistische Merkmale anhand einer Merkmalsmenge $\{n_1, \ldots, n_k\}$ extrahiert, wobei *k* die Anzahl der verwendeten Merkmale beschreibt. Beispiele für solche Merkmale sind unter anderem die gesamte Schreibdauer (*Ttotal*), Anzahl der Samplepunkte (*SampleCount*) oder aber die Anzahl der Absetzpunkte (*SegmentCount*). Diese und weitere statistische Merkmale eines Handschriftsamples werden in einen *k*-dimensionalen Merkmalsvektor geschrieben. In der originalen Version des Algorithmus wurden 69 Merkmale aus den Rohdaten bestimmt, jedoch gab es mehrere Erweiterungen, wodurch aktuell 131 Merkmale für die Bestimmung des Merkmalsvektors verwendet werden. Eine komplette Liste mit einer Kurzbeschreibung der einzelnen Merkmale ist im Anhang Anlage 1 zu finden. Während des Enrollments wird für jedes Handschriftensample (Enrollment Sample es_e) ein Merkmalsvektor (Feature Vector $n^{(e)}$)

bestimmt, wobei *e* als Index für das aktuelle Enrollment Sample steht. In Abbildung 19 wird dieser Prozess bildlich dargestellt.

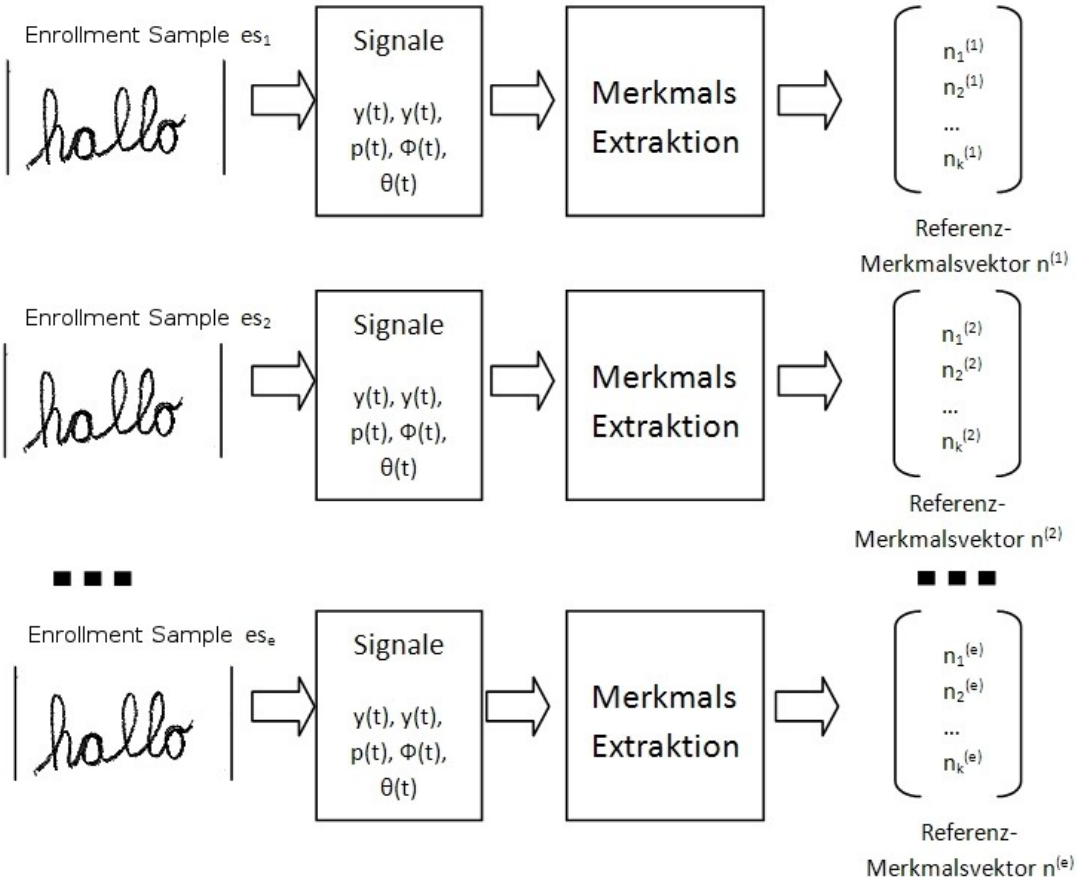


Abbildung 19 Berechnung der Merkmalsvektoren während des Enrollment Prozesses übersetzt und adaptiert aus [Viel06]

Intervallmatrix Berechnung

Für die Erläuterung des nächsten Schrittes (Berechnung der Intervallmatrix) wird folgende Notation eingeführt, um den Algorithmus detaillierter erklären zu können.

ES sei die Menge der Enrollment Samples eines Benutzers, wobei $ES=\{es_1, \ldots, es_e\}$. Die statistischen Merkmalsvektoren von *ES* seien $\{n(1), \ldots, n(e)\}$, wobei jeder Vektor dieselbe Anzahl an Merkmalen *k* enthält. Für die BioHash-Berechnung wird ein Enrollment Sample aus der Menge der Enrollment Samples ES eines Benutzers entnommen. Dieses Enrollment Sample es_{ref} wird ausschließlich für die BioHash-Berechnung verwendet.

MIN und *MAX* seien Funktionen, die bei Eingabe von mehreren *k*-dimensionalen Vektoren einen Vektor der Dimension *k* zurückgeben, wobei einer alle minimalen Werte und der andere alle maximalen Werte der jeweiligen Vektorkomponenten enthalten.

$I_{InitLow}$, $I_{InitHigh}$, I_{Low}, I_{High} und I_{Init} seien k-dimensionale Hilfsdaten, die für die Berechnung als Zwischenvariablen genutzt werden.

tv sei ein Toleranzvektor mit *k* Elementen: $tv=(tv_1, \ldots, tv_k)$, wobei jedes Element dieses Vektors als Faktor für die Intervallbreite eines jeden Merkmals (lokal) steht.

tf sei ein Toleranzfaktor, welcher global für alle Merkmale eingesetzt wird, um die Intervallbreite zu strecken bzw. zu kürzen.

MOD sei eine komponentenweise Modulo-Operation zwischen zwei *k*-dimensionalen Vektoren v_1 und v_2, wobei ein *k*-dimensionaler Vektor entsteht, bei dem jedes Element einzeln das Modulo-Ergebnis zwischen $v_1[i]$ *MOD* $v_2[i]$ $(i=1…k)$ ist.
Mit Hilfe dieser Notation kann die IM-Berechnung wie folgt definiert werden:

$$IM = (\Delta I, \Omega)$$ **Formel 4**

Wobei ΔI definiert wird mit

$$\Delta I = \begin{pmatrix} \Delta I_1 \\ \dots \\ \Delta I_k \end{pmatrix} = I_{High} - I_{Low} = \begin{pmatrix} \Delta I_{High,1} \\ \dots \\ \Delta I_{High,k} \end{pmatrix} - \begin{pmatrix} \Delta I_{Low,1} \\ \dots \\ \Delta I_{Low,k} \end{pmatrix}$$ **Formel 5**

ΔI ist ein Vektor, welcher aus Ganzzahlen besteht und die Intervallbreite für jedes einzelne Merkmal an der Stelle *i* enthält. I_{High} und I_{Low} sind dabei folgendermaßen gegeben:

$$I_{High} = \begin{pmatrix} \lceil I_{InitHigh,1} + tv_1 * \Delta I_{Init,1} * tf \rceil \\ \dots \\ \lceil I_{InitHigh,k} + tv_k * \Delta I_{Init,k} * tf \rceil \end{pmatrix}$$ **Formel 6**

$$I_{Low} = \begin{pmatrix} \lfloor I_{InitLow,1} + tv_1 * \Delta I_{Init,1} * tf \rfloor wenn \ (\lfloor I_{InitLow,1} + tv_1 * \Delta I_{Init,1} * tf \rfloor) > 0, sonst \ 0 \\ \dots \\ \lfloor I_{InitLow,k} + tv_k * \Delta I_{Init,k} * tf \rfloor wenn \ (\lfloor I_{InitLow,k} + tv_k * \Delta I_{Init,k} * tf \rfloor) > 0, sonst \ 0 \end{pmatrix}$$ **Formel 7**

$I_{InitLow}$ und $I_{InitHigh}$ werden mittels dem oben beschrieben Funktionen *MIN* und *MAX* wie folgt ermittelt (Formel 8 und Formel 9):

$$I_{InitHigh} = MAX(ES)$$ **Formel 8**

$$I_{InitLow} = MIN(ES)$$ **Formel 9**

I_{Init} wiederum wird auf Basis von $I_{InitLow}$ und $I_{InitHigh}$ berechnet (siehe Formel 10)

$$I_{Init} = I_{InitHigh} - I_{InitLow}$$ **Formel 10**

Der Offset Ω einer Intervallmatrix an der Stelle *i*, wird wie folgt berechnet:

$$\Omega_i = I_{Low,i} \ MOD \ \Delta I_i$$ **Formel 11**

Die Intervallmatrix beschreibt die Intervallbreite und den Intervallversatz für jedes Element eines Merkmalsvektors, der sich ergibt, wenn mehrere Enrollment Samples (mit jeweiliger interpersoneller Variabilität) aufgenommen werden. Die Intervallmatrix wird nur während des Enrollments berechnet und für die Verifikation im System hinterlegt.

BioHash-Berechnung
Von dem separierten Enrollment Sample es_{ref} zur BioHash-Berechnung wird ebenfalls der Merkmalsvektor berechnet. Mit Hilfe der IM wird anschließend dieser Merkmalsvektor in

einen Referenz-BioHash b_{ref} abgebildet und ebenfalls im System gespeichert. Er dient als Referenzwert bei einer späteren Verifikation, wobei die Generierung eines BioHash während der Verifikation identisch abläuft. Die Berechnung eines Wertes an der Stelle *i* sieht dann wie folgt aus:

$$b_i = m_{Skalar}(n_i, \Delta I_i, \Omega_i) = \left\lfloor \frac{(n_i - \Omega_i)}{\Delta I_i} \right\rfloor, \qquad \textbf{Formel 12}$$

wobei m_{Skalar} eine skalare biometrische Hashfunktion ist, welche einen positiven ganzzahligen Wert zurückgibt und b_i ein BioHash-Wert an der Stelle *i* ist. Dabei ist $i \epsilon \{1 \dots k\}$ und steht für die Anzahl der Elemente des Merkmalsvektors. Der BioHash-Vektor *b* besitzt entsprechend *k* Elemente $b=(b_1, \dots b_k)$, für jedes der Elemente wird demzufolge der BioHash-Wert an der Stelle i nach Formel 12 berechnet. Der formelle Ausdruck für die Berechnung des gesamten BioHashs sieht dann wie folgt aus:

$$b = m(n, IM) = m(n, \Delta I, \Omega) = \begin{pmatrix} m_{Skalar}(n_1, \Delta I_1, \Omega_1) \\ \dots \\ m_{Skalar}(n_k, \Delta I_k, \Omega_k) \end{pmatrix} = \begin{pmatrix} \left\lfloor \frac{(n_1 - \Omega_1)}{\Delta I_1} \right\rfloor \\ \dots \\ \left\lfloor \frac{(n_k - \Omega_k)}{\Delta I_k} \right\rfloor \end{pmatrix} \qquad \textbf{Formel 13}$$

Anhand der formellen Beschreibung wird klar, dass der BioHash-Vektor (BioHash) eine elementweise Verrechnung des gegebenen Merkmalsvektors mit vorher bestimmter Intervallmatrix ist. Von jedem Element des Merkmalsvektors wird der Intervallversatz subtrahiert, das Ergebnis durch die Intervallbreite dividiert und anschließend abgerundet. Als Ergebnis dieser Abbildung entsteht ein *k*-dimensionaler BioHash-Vektor mit ganzzahlig positiven Elementen.
Die beiden Faktoren *tf* und *tv* spielen bei der BioHash-Generierung eine wesentliche Rolle. Sie dienen dazu, die Werte der Intervallmatrix dahingehend zu verändern, dass die Intervallbreite verlängert oder gekürzt werden kann. Dies geschieht über den Toleranzvektor *tv* lokal für jedes Merkmal, währenddessen der Toleranzfaktor *tf* global auf alle Merkmale in gleicher Weise Einfluss nimmt. Abbildung 20 zeigt den Einfluss des *tv* beispielhaft auf ein Merkmal während der IM-Bestimmung. Es ist zu erkennen, dass die Intervallbreite in a) nach der IM Bestimmung für dieses Merkmal *16* ($\Delta I_i=16$) ist und der Offset $\Omega_i=10$ beträgt, beides für den Fall $tv_i=0$.

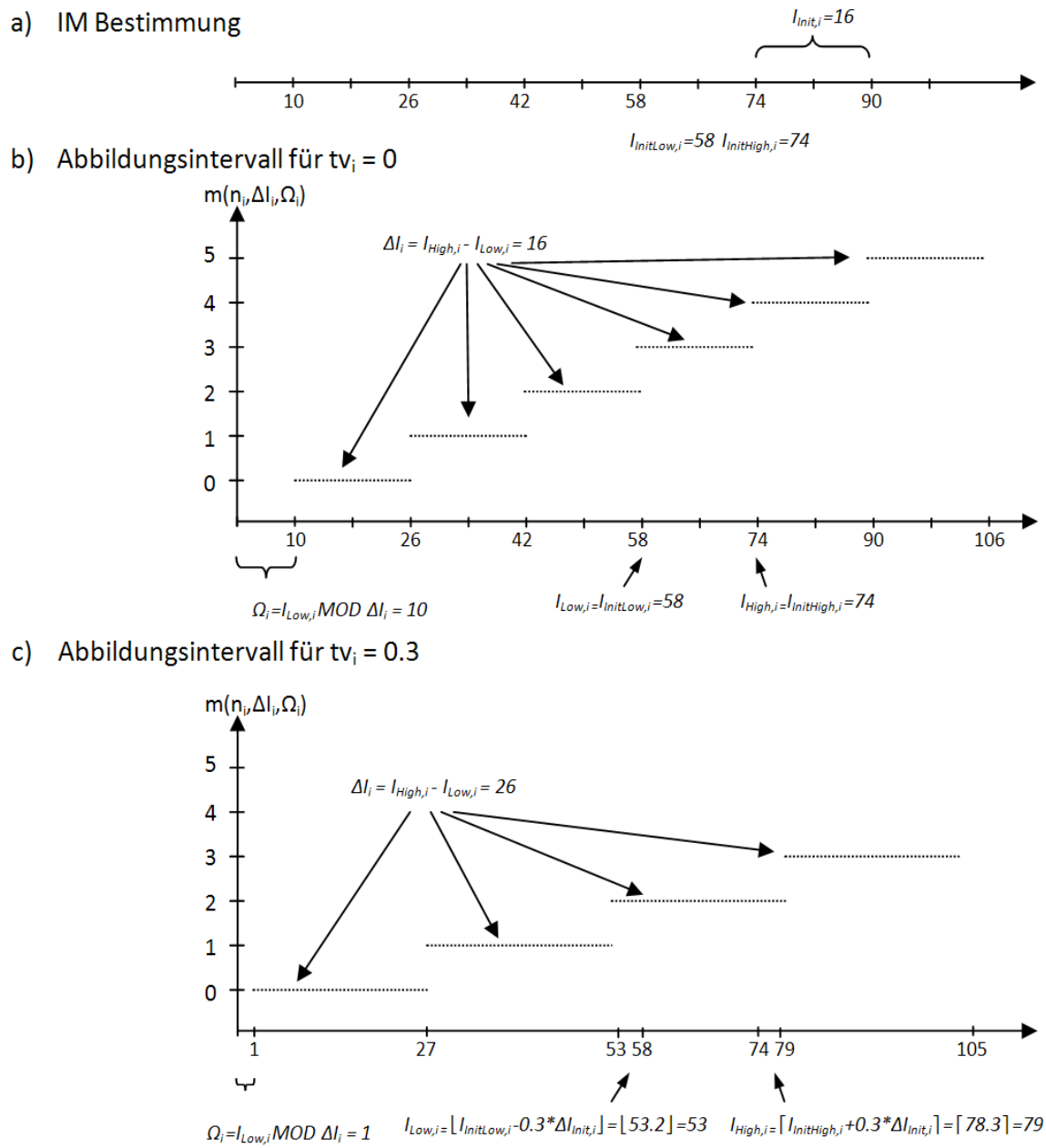


Abbildung 20 Einfluss eines Toleranzvektors bei der IM-Bestimmung übersetzt und adaptiert aus [Viel06]

Wird der Toleranzvektor an der Stelle *i* verändert (tv_i=0.3), so wird die Intervallbreite (ΔI_i=26) größer und der Intervallversatz (Ω_i=1) kleiner. Das System ist nun toleranter in Bezug auf dieses eine Merkmal (lokal). Wird die Intervallbreite für alle Merkmale vergrößert, so verringert sich die FRR des Systems und wird aus Sicht eines Benutzers komfortabler. Ziel ist es, mit Hilfe dieser Faktoren ein Erkennungssystem besser anpassen zu können, um es komfortabler oder sicherer zu gestalten.

Um zwei BioHash-Vektoren zu vergleichen und anschließend eine Entscheidung darüber treffen zu können, wie ähnlich sie sich sind, werden Funktionen benötigt, die den Abstand (Unterschied) zwischen diesen Vektoren ermitteln. Im nächsten Abschnitt werden derartige Funktionen vorgestellt.

4.2.3 Distanzfunktionen

Die hier vorgestellten Distanzfunktionen sind bereits bekannte Distanzfunktionen aus der Literatur. Sie werden in [Viel06] ebenfalls eingesetzt und sollen in dieser Arbeit auch eingesetzt werden. Diese Distanzfunktionen werden allgemein verwendet, um den Abstand zwischen zwei Vektoren oder Zeichenketten gleicher Länge zu bestimmen. Da ein BioHash auch ein Vektor ist, können diese Funktionen auch eingesetzt werden, um den Abstand zweier BioHashs zu bestimmen.
Es existieren verschiedene Möglichkeiten, zwei *k*-dimensionale Vektoren miteinander zu vergleichen. An dieser Stelle werden beispielhaft drei Methoden gezeigt, welche ein Maß für die Ähnlichkeit bzw. Unähnlichkeit zweier Vektoren ermitteln. Das sind die Hamming-Distanz [Ham50], Canberra-Distanz (siehe auch [VanL04]) und Euklidische-Distanz (siehe auch [VanL04]).

Hamming-Distanz
Bei der Hamming-Distanz werden die Elemente x_i und y_i der beiden Vektoren *x* und *y*, die sich jeweils an der *i*-ten Position befinden, miteinander verglichen. Sind sie identisch, ist das Ergebnis des Vergleichs *0*, ansonsten *1*. Im Anschluss werden die Einzelergebnisse summiert [Ham50]. Der Vorteil dieses Verfahrens liegt darin, dass der minimal bzw. maximal mögliche Abstandswert bekannt ist. Das Ergebnis kann Werte von *0* bis zur Anzahl *k* der untersuchten Merkmale reichen. In diesem Fall gilt also:

$$0 \le hd(x,y) \le k. \qquad \textbf{Formel 14}$$

Es ist unter anderem auch üblich, das Ergebnis der Hamming-Distanz zu normalisieren. Dazu wird das Ergebnis der Hamming-Distanz durch die Anzahl der verwendeten Merkmale dividiert. Das Ergebnis hierbei ist ein Wert zwischen *0* und *1*. In diesem Fall gilt demnach:

$$0 \le hd(x,y) \le 1. \qquad \textbf{Formel 15}$$

Ein Vorteil der Normierung besteht darin, dass Systeme mit einer unterschiedlichen Anzahl an verwendeten Merkmalen besser miteinander verglichen werden können.

Canberra-Distanz
Die Canberra-Distanz [VanL04] beschreibt nicht nur den Abstand zweier Punkte, sondern auch deren Lage zum Koordinatenursprung. Auch wenn je zwei Vektoren geometrisch den gleichen Abstand haben, ist der Canberra-Abstand unterschiedlich. In einem solchen Fall ist die Canberra-Distanz der beiden Vektoren, die näher am Ursprung liegen, kleiner als die zwischen den beiden anderen Vektoren. Die Canberra-Distanz ist definiert durch:

$$cd(x,y) = \sum_{i=1}^{k} \frac{|x_i - y_i|}{|x_i| + |y_i|} \qquad \textbf{Formel 16}$$

Genau wie bei der Hamming-Distanz liegt der Wertebereich des Ergebnisses einer Canberra Distanz zwischen *0* und *k*. Auch hier bietet es sich an, eine Normierung durchzuführen, um auf einen Wertebereich zwischen *0* und *1* zu gelangen.

Euklidische Distanz
Die Euklidische Distanz [VanL04] ist die mathematische Beschreibung des direkten Abstandes zwischen zwei Vektoren. Der Euklidische Abstand wird definiert durch:

$$ed(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$ **Formel 17**

Im Gegensatz zu den beiden zuvor erwähnten Methoden kann man keine Aussage darüber treffen, in welchem Wertebereich das Ergebnis des Vergleiches liegt. Hier ist es auf jeden Fall sinnvoll, eine Normierung durchzuführen, möchte man unterschiedliche Systeme oder Distanzfunktionen miteinander vergleichen.

4.2.4 Bekannte Schwachstellen des BioHash-Algorithmus

In diesem Abschnitt soll dargestellt werden, welche Schwachstellen der BioHash-Algorithmus aufweist, die sich ein potentieller Angreifer zunutze machen kann. Es wird davon ausgegangen, dass ein Angreifer sich Zugang zu einem System verschaffen möchte, zu dem er keine Zugangsberechtigung hat. Die Authentifizierung erfolgt mit Hilfe des BioHash-Algorithmus.
Potentielle Schwachstellen können in direkte und indirekte Schwachstellen gegliedert werden, wobei indirekte Schwachstellen sich nicht auf die Arbeitsweise des Authentifizierungsalgorithmus beziehen, sondern auf allgemeine Schwachpunkte von Sicherheitssystemen. Direkte Schwachstellen beziehen sich auf die Arbeitsweise und das Design des Authentifizierungsalgorithmus und die damit potentiell einhergehenden Sicherheitslücken (siehe Abschnitt 2.3.1 Angriffsarten).

Indirekte Schwachstellen
Wie jedes Sicherheitssystem, welches auf Authentifizierung von Nutzern basiert, hat auch der BioHash-Algorithmus Schwachstellen aufgrund der Arbeitsweise des gesamten Authentifizierungssystems (z.B. Zusammenspiel von Hard- und Softwarekomponenten). Ein Angreifer kann den Datenstrom zwischen Sensor und Rechnersystem mitschneiden, später kann er diese Daten wieder in das System einspielen, um sich so einen Zugang zum System zu verschaffen. Es ist außerdem möglich, dass ein Angreifer eine Videokamera installiert, um den Authentifizierungsvorgang aufzuzeichnen. Mit diesem Verfahren ist es möglich, zumindest das geschriebene Wort, die Zeichenkette oder das Symbol zu bestimmen, welches eine Person verwendet. Druckwerte und Winkel können über dieses Verfahren nicht bestimmt werden, jedoch können die gewonnen Informationen in einem *Brute-Force* Angriff als Zusatzinformation (Seiteninformation) einfließen. Dadurch kann der Eingabewerteraum erheblich eingeschränkt werden, was wiederum zu einer verkürzten Angriffszeit führt.
Um beispielsweise Schwachstellen in der Architektur (siehe Angriffspunkte in Abbildung 1), Hardware und/oder Betriebssystemen (Windows, Linux, MacOS, Android, etc.), auf denen Authentifizierungsalgorithmen implementiert werden können zu vermeiden, sind Portierungsmöglichkeiten in verschieden wissenschaftlichen Arbeiten vorgestellt worden. Kreditkartengroße Kryptokarten können beispielsweise eingesetzt werden, um bestimmte Arbeitsschritte eines biometrischen Erkennungsalgorithmus durchzuführen. Das Speichern (Store-On-Card) und Vergleichen zweier biometrischer Templates (Matching-On-Card), wie in [StSc02], [HeFr04] und [CAP+06] gezeigt, seien hier als Beispiel genannt.

Des Weiteren können mitunter alle Prozesse eines biometrischen Erkennungssystems in einer sicheren Umgebung portiert werden (Biometric-System-On-Card). Letzteres hat der Autor zusammen mit Vielhauer in den Arbeiten [KüVi11] und [KüVi11a] vorgestellt, wobei eine Java Card als sichere Umgebung und der Biometric Hash Algorithmus als Authentifizierungsalgorithmus verwendet wurde. In solchen sicheren Umgebungen kann das Angriffspotential drastisch minimiert werden.

Direkte Schwachstellen

Eine direkte Schwachstelle des Algorithmus liegt in der Verwendung der statistischen Merkmale, deren Berechnungsgrundlage und deren Beziehung untereinander. Betrachtet man alle verwendeten Merkmale zur BioHash-Generierung (aktuell 131), so erkennt man Abhängigkeiten der Merkmale untereinander. Franke hat in seiner Arbeit [Fran09] diese Abhängigkeiten genauer untersucht und konnte die Merkmale in unterschiedliche Klassen einteilen.

Zuerst hat er alle Merkmale in folgende Kategorien unterteilt: *samplepunktabhängige* Merkmale, *positionsabhängige* Merkmale, *druckabhängige* Merkmale, *höhenwinkelabhängige* Merkmale und *rotationswinkelabhängige* Merkmale. Eine Korrelation zwischen diesen Kategorien wird in der Arbeit als nicht existent angenommen, jedoch können Abhängigkeiten innerhalb einer Kategorie auftreten. Aufgrund dieser Erkenntnis hat Franke weitere Einteilungen vorgenommen. Merkmale, die unabhängig von anderen Merkmalen sind, bezeichnet er als *Basismerkmale* und jene, die von anderen Merkmalen abhängig sind, als *abhängige* Merkmale. Des Weiteren wurde die Komplexität (gering, mittel, schwer) der einzelnen Merkmale in Bezug auf ihre Reproduzierbarkeit abgeschätzt. Dieses Wissen kann dafür verwendet werden, Rohdaten zu einem bestimmten Merkmalsvektor zu generieren.

Eine weitere Schwachstelle des BioHash-Algorithmus, wurde unter anderem vom Autor und weiteren Co-Autoren in der Arbeit [KVS+10] aufgedeckt. Sie basieren teilweise auf Erkenntnissen der bereits oben dargelegten Schwachstellen aus [Fran09] und befassen sich mit der Rückrechnung des Merkmalsvektors und der Interpolation von Signalen anhand bestimmter Merkmale.

Rückrechnung des Merkmalsvektors

Im folgenden Szenario hat ein Angreifer ein Sicherheitssystem kompromittiert und Zugang zur Intervallmatrix, BioHash und Name einer oder mehrerer Personen erhalten. Es ist nun möglich, anhand einer Intervallmatrix und dem zugehörigen BioHash einen für die Verifizierung gültigen Merkmalsvektor zu erzeugen. Hierfür bedarf es lediglich einer Rückrechnung, welche sich aus der Formel zur BioHash-Berechnung ableiten lässt. Im Abschnitt 4.2.2 wird in Formel 12 beschrieben, wie der BioHash an der Stelle *i* berechnet wird. Stellt man nun diese Formel nach n_i um, erhält man aufgrund der Abrundung den unteren Wert der Intervallbreite. Um jedoch einen besseren Wert bezüglich der Reproduzierbarkeit zu erhalten, bietet es sich an, die halbe Intervallbreite zu addieren. Das Resultat wurde bereits in [KüVi10a] präsentiert und sieht wie folgt aus:

$$n_i = b_i \cdot \Delta I_i + \Omega_i + \frac{\Delta I_i}{2} \qquad \textbf{Formel 18}$$

Rohdatengenerierung auf Basis bestimmter Merkmale
Schaut man sich Basismerkmale und solche mit geringer Komplexität an, so kann man erkennen, dass sich einige Merkmale sofort in Rohdaten umsetzen lassen. Die Merkmale n_1 (gesamte Schreibdauer) oder n_2 (Anzahl der Samplepunkte) können beispielsweise relativ einfach umgesetzt werden. Hierfür muss einfach ein pseudozufälliges Signal generiert werden mit der Gesamtdauer n_1 und mit einer Anzahl von Samplepunkte n_2. Andere Merkmale können für mathematische Operationen verwendet werden, um Signalkurven zu generieren. Merkmale, welche Maxima und Minima in X sowie Y Richtung zählen, wie n_{70}, n_{71}, n_{72} und n_{73}, lassen sich gut für eine Spline-Interpolation verwenden. Dabei wird eine Reihe von Stützpunkten pseudozufällig generiert (Anzahl der Maxima und Minima) und auf Basis dieser Stützpunkte ein Signal mittels Spline-Interpolation generiert, dies wurde bereits in [KüVi10a] gezeigt.

Ist ein Angreifer im Besitz eines gültigen Merkmalsvektors, kann er unter Zuhilfenahme weiterer Angriffsmethoden gegebenenfalls Rohdaten erzeugen, welche zu einer positiven Verifizierung an einem System führt. Ein Brute-Force Angriff sei hier nur als eine mögliche Methode genannt.
Die in diesem Abschnitt bereits bekannten und genannten Schwachstellen stellen unter bestimmten Bedingungen eine potenzielle Bedrohung dar. Angreifer könnten sie nutzen, um z.B. einen Identitätsdiebstahl durchzuführen.

4.3 Kryptologie

Der Begriff Kryptologie leitet sich ab vom griechischen *kryptoin* (verborgen) [Dude20] und *logos* (Wissenschaft) [Dude20a]. Sie beschäftigte sich ursprünglich mit der Verschlüsselung von Klartexten mit der Verhinderung des Zugriffs (lesen) durch unbefugte Personen. Außerdem behandelt die Kryptologie das Übertragen von Nachrichten auf sicheren Kanälen, damit nur der rechtmäßige Empfänger diese Nachricht entschlüsseln und somit lesen kann. Mit der Entwicklung kryptografischer Verfahren wurden, neben vielen anderen, auch Hash-Funktionen eingeführt die u.a. Verfahren für die digitale Signatur ermöglichten (siehe z.B. auch [FHKS14]). Das Themenfeld der Kryptologie ist sehr weitreichend und umfassend, weshalb an dieser Stelle nur ausgesuchte Einblicke gegeben werden sollen, die für diese Arbeit relevant sind. Interessierte Leser finden zum Thema Kryptologie eine Vielzahl von literarischen Werken, die einen detaillierten und umfangreichen Einblick in dieses Feld geben z.B. [FHKS14] oder [Beut15].
Die Kryptologie wird in weitere Themengebiete klassifiziert, zum einen die Kryptographie mit dem Gegenstück Kryptoanalyse und der Steganographie mit dem Gegenstück Stegoanalyse, siehe Abbildung 21.

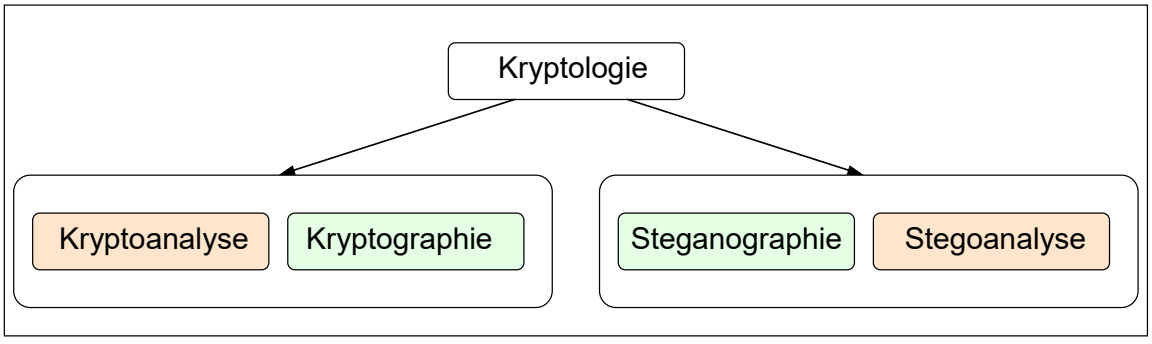


Abbildung 21 Kryptologie und deren Teilgebiete, adaptiert bei [AmRa13] und [Schm09]

Der Teilbereich der Steganographie befasst sich mit der Übermittlung von Nachrichten in verdeckten Kanälen. Demnach hat die Steganographie das Ziel mittels geeigneter Methoden geheime Nachrichten in z.B. anderen Nachrichten oder Bildern zu verstecken. Die Wissenschaft der Stegoanalyse hingegen beschäftigt sich mit Verfahren, um versteckte Nachrichten zu entdecken und sichtbar bzw. lesbar zu machen, weswegen es auch als Gegenstück der Steganographie bezeichnet wird (siehe dazu auch [FrPf98] und [JoJa98]). Die Teilgebiete Steganographie und Stegoanalyse sollen hier nicht weiter beleuchtet werden. Da sich die Arbeit zum Teil mit Methoden der Kryptographie bzw. Kryptoanalyse beschäftigt, werden diese Teilgebiete in den nachfolgenden zwei Abschnitten im Überblick betrachtet. Weiterhin wird im anschließenden Abschnitt die Idee *homomorpher Verschlüsselung* dargestellt, die sich grundsätzlich auch für den Schutz und der Verarbeitung biometrischer Referenzdaten einsetzen lässt.

4.3.1 Kryptographie

Die Wissenschaft der Kryptographie befasst sich unter anderem mit Techniken und Methoden, Nachrichten vor Unbefugten zu schützen, indem beispielsweise Verschlüsselungstechniken eingesetzt werden. In der Kryptographie sieht man einer verschlüsselten Nachricht auch an, dass sie verschlüsselten Inhalt enthält. Hier wird im Gegensatz zur Steganographie kein vermeintlich harmloser Text als Trägermedium für eine geheime Nachricht zu dessen Verschleierung verwendet. Primär geht es darum Informationen nur befugten Personen zugänglich zu machen. Die Vertraulichkeit ist unter anderem ein Sicherheitsaspekt, welcher von der Kryptographie adressiert wird. Weitere durch die Kryptographie adressierte Sicherheitsaspekte sind u.a. Integrität, Authentizität und Verbindlichkeit, siehe dazu auch [Schm09] und [Schn96].
Über die Zeit haben sich verschiedene Methoden zur Verschlüsselung von Nachrichten innerhalb der Kryptographie entwickelt. Bei der *Transposition* werden Buchstaben einer Nachricht in einer anderen Reihenfolge angeordnet (z.B. Skytale Verfahren). Wohingegen bei der *Substitution* die Buchstaben der Botschaft durch andere Buchstaben bzw. Symbole ersetzt werden (z.B. Caesar- oder Vigenére-Verschlüsselung). Moderne kryptographische Verfahren manipulieren Nachrichten nicht auf der Buchstabenebene, sondern auf der Bit-Ebene. Dies ermöglicht das Verschlüsseln von Textnachrichten, als auch beliebiger digitaler Daten (z.B. Bilder, Videos usw.).
Hierfür wird die zu verschlüsselnde Nachricht m (Klartext) mit Hilfe einer Verschlüsselungsfunktion E und eines Schlüssels k_1 in einen Schlüsseltext (Chiffrat) c umgewandelt. Dieses Chiffrat c kann dann mit Hilfe einer Entschlüsselungsfunktion D und einem Schlüssel k_2 wieder in den ursprünglichen Klartext m transformiert werden. Formal betrachtet und unter Verwendung der einzelnen Definitionen sieht eine Ver- und Entschlüsselung folgendermaßen aus, siehe z.B. [Schm09]:
Verschlüsselung: $c = E(m, k_1)$
Entschlüsselung: $m = D(c, k_2)$.

Symmetrische und asymmetrische Verschlüsselung
Des Weiteren werden moderne kryptographische Methoden in symmetrische und asymmetrische Verschlüsselung unterschieden. Bei einer symmetrischen Verschlüsselung wird ein und derselbe Schlüssel ($k_1 = k_2 = k_s$) für die Ver- und Entschlüsselung verwendet (siehe Abbildung 22).

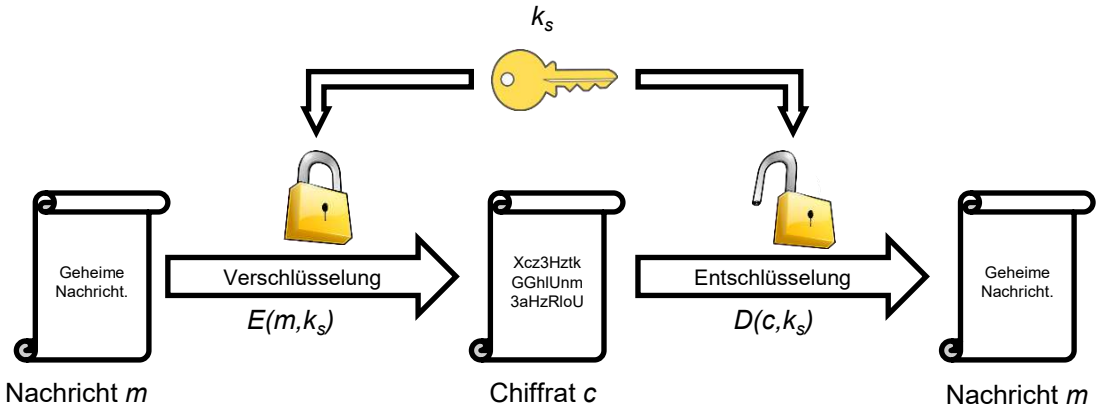


Abbildung 22 Symmetrische Ver- und Entschlüsselung, übersetzt und adaptiert von [Schn96]

Bei der asymmetrischen Verschlüsselung verwendet jeder Nutzer bzw. jede Anwendung ein Schlüsselpaar. Ein privater (geheimer) Schlüssel für das Entschlüsseln und ein öffentlicher Schlüssel für das Verschlüsseln von Nachrichten (siehe Abbildung 23).

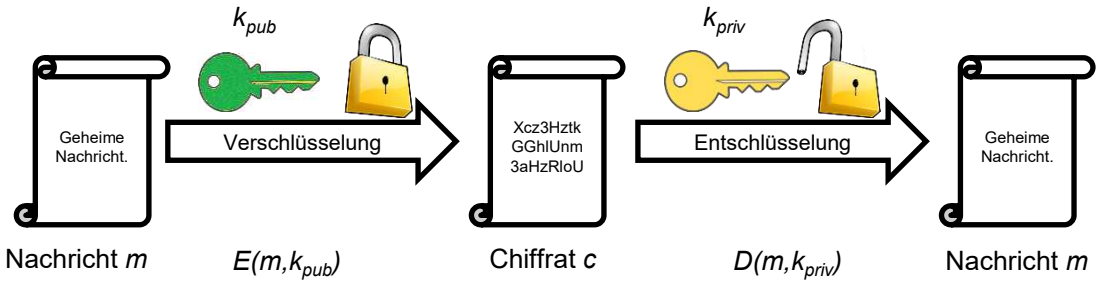


Abbildung 23 Asymmetrische Ver- und Entschlüsselung, übersetzt und adaptiert von [Schn96]

Die Kombination von symmetrischer und asymmetrischer Verschlüsselung (hybride Verschlüsselung) wird unter anderem im Bereich der Kommunikation im Internet eingesetzt. Hierbei werden asymmetrische Verfahren verwendet, um einen sicheren Kommunikationskanal aufzubauen. Anschließend werden symmetrische Schlüssel, sogenannte Session-Keys, für die eigentliche Verschlüsselung der Daten ausgetauscht. So werden die Vorteile der asymmetrischen Verschlüsselung (einfache Schlüsselverwaltung) und der symmetrischen Verschlüsselung (schnelle Verarbeitung auch bei großen Datenmengen) genutzt [Schn96].

Kryptographischer Hash

Die Kryptographie beschäftigt sich auch mit der Generierung von Hashwerten und dem digitalen Signieren von elektronischen Dokumenten. Wobei innerhalb des Prozesses zur Erzeugung einer digitalen Signatur eine Hashwertgenerierung stattfinden kann. Bei der Hashwertgenerierung wird im Allgemein eine Datenmenge beliebiger Länge auf eine Datenmenge mit fester Länge abgebildet. Hashfunktionen sind Einwegfunktionen und demnach unumkehrbar, siehe z.B. auch [Schm09]. Sie werden beispielsweise eingesetzt, um Passwörter nicht im Klartext in einer Datenbank zu speichern; es wird entsprechend nur der Hashwert des Passwortes gespeichert. Bei der Verifizierung einer Passworteingabe wird der Hashwert der aktuellen Eingabe gebildet und anschließend mit dem Hashwert in der Datenbank verglichen. Außerdem werden Hashwerte eingesetzt, um die Integrität einer Datei, welche beispielsweise von einer Webapplikation zur Verfügung gestellt wird,

zu verifizieren. Die Webapplikation bietet neben der eigentlichen Datei auch den dazugehörigen generierten Hashwert an. Nachdem die Datei heruntergeladen wurde, kann ein Hashwert der Datei gebildet und anschließend mit dem Hashwert der Webapplikation verglichen werden. Anforderungen die an einer Hashfunktion bzw. an einem kryptographischen Hashwert gestellt werden, können wie folgt kurz formuliert werden (siehe dazu auch z.B. [BMC+05]):

Reproduzierbarkeit:	Sind zwei Eingabewerte *a* und *a'* identisch, so sollen auch die beiden durch ein und dieselbe kryptographische Hashfunktion *H* berechneten Werte *H(a)* und *H(a')* identisch sein.
Kollisionsresistenz:	Sind zwei Eingabewerte *a* und *a'* ungleich, dann müssen auch die durch eine kryptographische Hashfunktion *H* berechneten Hashwerte *H(a)* und *H(a')* ungleich sein.
Unumkehrbarkeit:	Es sollte rechnerisch nicht möglich sein, aus dem durch die kryptographische Hashfunktion *H* erzeugten Hashwert *H(a)* wieder den Ausgangswert *a* zu bestimmen.
Bitsensitivität:	Kleine Änderungen der Eingabedaten *a* sollten zu möglichst großen Veränderungen der Ausgabedaten *H(a)* führen.

Digitale Signatur

Digitale Signaturen bieten die Möglichkeit, die Sicherheitsaspekte *Authentizität* und *Verbindlichkeit* mittels kryptographischer Methoden umzusetzen. Als Bespiel sei hier die Authentizität eines Senders einer bestimmten Nachricht gegeben. Möchte Alice eine Nachricht an Bob schicken und Bob will sicherstellen, dass die empfangene Nachricht wirklich von Alice verfasst wurde, kann er dies mit Hilfe der digitalen Signatur bewerkstelligen. Hierzu kann z.B. das Signieren mittels Public-Key und Hashfunktionen wie folgt eingesetzt werden [Schn96].

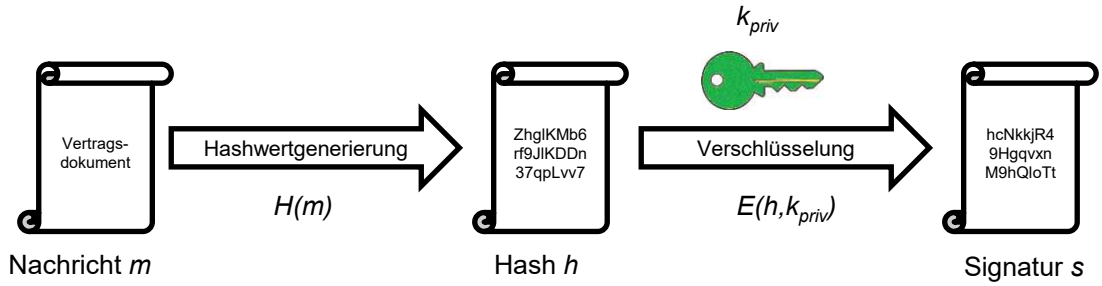


Abbildung 24 Signieren eines Dokumentes, übersetzt und adaptiert von [Schn96]

1. Alice generiert einen Einweg-Hash (kryptographischen Hash) eines Dokumentes (Nachricht).
2. Alice verschlüsselt den Hash mit ihrem privaten Schlüssel (signieren des Dokumentes).
3. Alice sendet das Dokument und den verschlüsselten Hash an Bob.
4. Bob generiert einen Einweg-Hash (kryptographischen Hash) des Dokumentes, welches Alice gesendet hat. Anschließend entschlüsselt er mit Hilfe des digitalen Signieralgorithmus und des öffentlichen Schlüssels von Alice den gesendeten Hash von Alice. Sind beide Hashwerte (Bobs und Alice) gleich so ist die Signatur gültig.

Das Signieren der Nachricht wird in Abbildung 24 und das Prüfen einer Signatur in Abbildung 25 grafisch dargestellt.

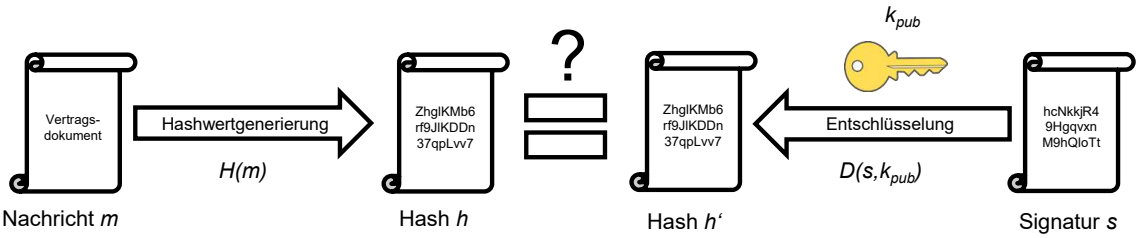


Abbildung 25 Prüfen einer digitalen Signatur, übersetzt und adaptiert von [Schn96]

Anwendungsgebiete der Kryptographie

Kryptographische Techniken und Methoden haben sich derzeit unter anderem fest im Bereich der Telekommunikation und der digitalen Informationssysteme im Allgemein etabliert, siehe [Schn96]. Internetanwendungen mit Einwahlmöglichkeiten für den Nutzer, wie beispielsweise soziale Netzwerke, Verkaufsplattformen und sogenannte Emailwebclients, bieten die Möglichkeiten eines sicheren Logins via SSL (HTTPS)-Verschlüsselung. Die Webanwendung und der Client etablieren hierfür eine verschlüsselte Kommunikation, somit wird die Übertragung der Login-Daten z.B. Name und Passwort, chiffriert durchgeführt [Schn96].

Mit Hilfe von sogenannten Virtuellen Privaten Netzwerken (VPN) können zum Beispiel zwei räumlich getrennte Netzwerke über das Internet sicher miteinander verbunden werden [FeHu98]. Diese Form der sicheren Kommunikation findet nicht nur im universitären, geschäftlichen und behördlichen Umfeld statt, sondern wird auch im privaten Umfeld immer öfter eingesetzt. Auch der Einsatz von verschlüsseltem Emailverkehr wird mittels kryptographischer Verfahren durchgeführt, hier sei PGP (Pretty Good Privacy) als Beispiel genannt. Kryptographische Einwegfunktionen kommen oftmals dort zum Einsatz, wo beispielsweise Passwörter in einer Datenbank zur späteren Authentifikation hinterlegt werden müssen. Speziell die Häufigkeit der Verwendung verschlüsselter Kommunikation ist seit der Bekanntmachung von E. Snowden bzgl. der Mithöraktivitäten amerikanischer Geheimdienst drastisch gestiegen, siehe dazu auch [West17].

4.3.2 Kryptoanalyse

Wie bereits im Abschnitt 4.3 erläutert, bildet die Kryptoanalyse das Gegenstück zur Kryptographie. Sie beschäftigt sich demnach damit, kryptographische Verfahren zu brechen oder zu umgehen, um so beispielsweise verschlüsselte Nachrichten im Klartext darzustellen. Hierbei ist jedoch nicht immer von einer bösen Absicht auszugehen, sondern die Kryptoanalyse wird in der Regel angewendet, um die Sicherheit von kryptographischen Verfahren und Algorithmen zu testen und ggf. zu bestätigen, siehe dazu auch [FHKS14]. In dieser Arbeit sollen nur beispielhaft Auszüge aus dem Themenfeld der Kryptoanalyse präsentiert werden. So soll u.a. die Vielzahl existierender Angriffs- bzw. Analysetechniken dargestellt werden, die zum Teil auch auf biometrische Systeme adaptiert werden können (z.B. Seitenkanal Angriffe). Um ein Verschlüsselungsverfahren grundsätzlich sicher zu gestalten, sollte die Sicherheit nicht von der Geheimhaltung des Verschlüsselungsverfahrens abhängen, sondern nur von der Geheimhaltung des Schlüssels (Prinzip von Kerkhoff). Diese Aussage gilt insbesondere für symmetrische Verfahren, wo ein und derselbe Schlüssel für die Ver- und Entschlüsselung verwendet wird ($k_1 = k_2$).

Shannon definierte 1949 den Begriff der „Perfekten Sicherheit“ mit der sinngemäßen Aussage: Eine Verschlüsselungsfunktion ist sicher, wenn Klartext und Chiffrat statistisch unabhängig sind, das heißt das Chiffrat keine Informationen über den Klartext gibt

[Shan49a]. Nun ist es schwer, eine solche statistische Unabhängigkeit umzusetzen und anschließend beweisen zu können [Shan48]. Um dennoch eine Verschlüsselungsfunktion besser einschätzen und als sicher einstufen zu können, muss sie Angriffen (statistische Analyse) aus dem Bereich der Kryptoanalyse standhalten können, siehe z.B. dazu [Schm09]. Diese Angriffe sind zum Beispiel *Ciphertext-Only-Attacken*, *Known-Plaintext-Attacken*, *Choosen-Plaintext-Attacken* und *Chosen-Ciphertext-Attacken*.
Es sollte weiterhin nicht möglich sein, einen Schlüssel durch systematisches Probieren aller möglichen Kombinationen (Brute-Force) in einer bestimmten Zeit ermitteln zu können. In vielen Kryptosystemen bestimmt die Zeichenlänge des Schlüssels den Suchraum für einen solchen Brute-Force-Angriff; je größer die Schlüssellänge, desto größer der Suchraum.
Die hier erwähnten Angriffsmethoden können auch in modifizierter Form auftreten, wenn beispielsweise zusätzliche Informationen in einen Angriff mit einfließen. Zusätzlich ist eine Kombination der aufgezählten Angriffsmethoden möglich.
Die oben erwähnten Angriffsmethoden aus dem Bereich der Kryptoanalyse können in einigen Fällen modifiziert werden. So ist es vorstellbar, dass ein Angreifer zu den oben genannten Möglichkeiten noch weitere Informationen besitzt. Ist dies der Fall, werden solche Angriffe auch *Seitenkanal Angriffe* genannt. Demnach hat ein Angreifer über einen anderen Kanal zusätzliche Information erhalten, die es ihm ermöglichen, noch effizienter einen Angriff durchzuführen. Die Beschaffung zusätzlicher Informationen über einen Seitenkanal kann dabei völlig unterschiedlich aussehen. So ist es möglich, dass der Angreifer über Diebstahl, Spionage oder Social Engineering an zusätzliche Informationen gelangt ist, um nur einige Beispiele zu nennen.

Anwendungsgebiete der Kryptoanalyse
Neben dem eigentlichen Zweck, kryptographische Verfahren zu testen, werden die Angriffsmethoden der Kryptoanalyse auch „praktisch“ eingesetzt. Das Aufdecken von vertraulichen Informationen wie beispielsweise Passwörter oder PINs haben für Personen mit gewisser krimineller Energie einen großen Nutzen. Vertrauliche Firmengeheimnisse sind sicherlich für konkurrierende Firmen und/oder Finanz- und Wirtschaftsspekulanten von ungeheurem Interesse. Auch staatliche Einrichtungen wie Nachrichtendienste und polizeiliche Ermittlungsbehörden haben ein Interesse, ggf. verschlüsselte Informationen für ihre Zwecke zu verwerten. Die Bandbreite für die Verwendung von Methoden und Werkzeugen der Kryptoanalyse ist groß.

Angreifermodelle
Bei Sicherheitsanalysen von Kryptosystemen wird ein bestimmtes Angreifermodell vorausgesetzt, welches das Verhalten und die Fähigkeiten eines Angreifers bzw. Gegenspielers (engl. Adversary) in einem bestimmten Szenario definiert [Cane20].

- Honest-but-curios Adversary (ehrlich aber neugierig) beschreibt das Szenario, in dem sich zwei oder mehrere Parteien an das vorgesehene Protokoll halten (ehrlich sind) jedoch versuchen, durch das Mitlesen der empfangenen Daten zusätzlich Informationen zu rekonstruieren, die nicht für sie bestimmt sind (neugierig).
- Malicios Adversary (bösartig), beschreibt einen manipulierenden Angreifer, der bösartig ist. Im Gegensatz zu einem lesenden Angreifer (Honest-but-curios Adversary) kann nicht vorausgesetzt werden, dass er sich an das Protokoll hält. Dadurch

kann es passieren, dass das Protokoll verzögert abgearbeitet bzw. vorzeitig abgebrochen wird.

4.3.3 Homomorphe Verschlüsselung

Homomorphe Verschlüsselung ist eine Art der Verschlüsselung, die es erlaubt, Operationen auf verschlüsselte Daten durchzuführen, ohne diese vorher zu entschlüsseln [BrGV12]. In dieser Arbeit wird in den Ergebnisdiskussionen auf homomorphe Verschlüsselung eingegangen, weswegen sie an diese Stelle kurz beschrieben werden soll.
Bei einer Datenverarbeitung mit „normal" verschlüsselten Daten werden die entsprechenden Daten vor der Verarbeitung entschlüsselt, danach verarbeitet und anschließend wieder verschlüsselt. Diese Art der Datenverarbeitung hat u.a. den Nachteil, dass die Daten im Klartext verarbeitet werden müssen und dies ein Sicherheitsrisiko darstellen kann. Das Ziel der *homomorphen Verschlüsselung* ist es u.a., dieses Sicherheitsrisiko zu minimieren und die Datenverarbeitung im verschlüsselten Zustand durchzuführen.
Im Laufe der letzten Jahre wurden verschiedene homomorphe Verschlüsselungstechniken und -methoden entwickelt. Nachfolgend werden einige Meilensteine der Entwicklung homomorpher Verschlüsselungssysteme dargestellt.
Partielle homomorphe Verschlüsselung
Bei der partiellen homomorphen Verschlüsselung handelt es sich um eine Klasse von homomorphen Verschlüsselungsmethoden die lediglich die Additionsoperation auf verschlüsselte Daten erlauben. Als Beispiel seien hier die Arbeiten von Cramer et al. [CrSS97] und Paillier [Pail99] genannt. In einer Arbeit von Barni et al. [BaDL15] wird beispielsweise das von Paillier vorgestellte System eingesetzt, um ein biometrisches Verifikationssystem auf Basis der Modalität Fingerabdruck zu schützen.

„Somewhat" homomorphe Verschlüsselung
Der Name „Somewhat homomorphe Verschlüsselung" (ein „bisschen" homomorphe Verschlüsselung) bezieht sich auf die in der wissenschaftlichen Gemeinschaft etablierte Bezeichnung „Somewhat homomorphic encryption (SHE)". Es bezieht sich auf die funktionalen Eigenschaften der Klasse von homomorphen Verschlüsselungsmethoden. Bei dieser Klasse können Additions- und Multiplikationsoperation auf verschlüsselten Daten vorgenommen werden, jedoch ist die Anzahl der Operationen begrenzt. Aus diesem Grund bezeichnet man diese Klasse auch „Somewhat" (etwas, ein bisschen oder ein wenig) homomorphe Verschlüsselung.
Eine der ersten *„Somewhat" homomorphen Verschlüsselungsmethoden* wurde von Boneh et al. in [BoGN05] vorgestellt. Diese Methode erlaubt es eine gewisse Anzahl von Additionen jedoch nur eine Multiplikation auf den verschlüsselten Daten durchzuführen. Nachdem Gentry in [Gent09] und [Gent09a] eine Methode der *Vollständigen homomorphen Verschlüsselung* vorstellte, wurden einige weitere „somewhat" homomorphe Verschlüsselungsmethoden vorgestellt, welche als Grundsteine weiterer vollständiger homomorpher Verschlüsselungsmethoden verwendet werden konnten. Als Beispiel seien an dieser Stelle die Arbeiten von Brakerski et al. [BrVa11], [BrVa11a], [BrGv12], Coron et al. [CMNT11] sowie van Dijk et al. [VanD10] genannt. Im Gegensatz zu der oben genannten Methode von Boneh et al. erlauben diese Methoden mehr als eine Multiplikationsoperation auf verschlüsselte Daten.

Vollständige homomorphe Verschlüsselung

Die *Vollständige homomorphe Verschlüsselung* (VHV) unterstützt unbegrenzte Additions- und Multiplikationsoperationen auf verschlüsselte Daten. Gentry hat erstmals im Jahr 2009 in [Gent09] und [Gent09a] eine Methode zur *Vollständigen homomorphen Verschlüsselung* (VHV) vorgestellt. Ein Nachteil dieser ersten Methode waren jedoch die relative hohe Berechnungszeit und die steigende Größe der verschlüsselten Daten.

In den folgenden Jahren wurden weitere vollständig homomorphe Verschlüsselungs-methoden vorgestellt (siehe u.a. [BrVa11], [BrVa11a], [BrGv12], [VanD10] und [CMNT11]), welche jedoch alle auf dem von Gentry entwickelten ersten System basieren. Die weiter entwickelten VHV Methoden waren allerdings effizienter konzipiert, wodurch sich die Berechnungszeit und die Datengröße der verschlüsselten Daten verringerten.

So konnten beispielsweise Lucas und Micciancio in [DuMi15] ein Verfahren zeigen, welches die Berechnung in weniger als einer Sekunde absolvierte. Aufbauend auf dieser Arbeit haben Chillotti et al. in [CGGI16] eine Methode vorgestellt, welche die Berechnung in weniger als 0,1 Sekunden durchführt. Auf Basis dieser Berechnungszeiten lassen sich u.a. praktische Anwendungen erstellen.

4.4 Kryptographischer Hash vs. Biometrischer Hash

In diesen Abschnitt werden Gemeinsamkeiten und Unterschiede zwischen kryptographischem und biometrischem Hash beschrieben. Diese Unterschiede sind bereits bekannt und wurden vom Autor und Vielhauer in [KüVi10a] vorgestellt.

Um zwei Funktionen miteinander zu vergleichen bietet es sich an, die Anforderungen der beiden Funktionen gegenüberzustellen. In Abschnitt 4.3.1 werden die Anforderungen (Reproduzierbarkeit, Kollisionsresistenz, Unumkehrbarkeit und Bitsensitivität) für den kryptographischen Hash erläutert. Für einen übersichtlichen Vergleich werden in Tabelle 2 die Anforderungen eines kryptographischen als auch eines biometrischen Hashs beschrieben.

Reproduzierbarkeit

Die Anforderungen bei einem kryptographischen Hash bezüglich der Reproduzierbarkeit beziehen sich auf den konkreten Eingabewert (Eingabemenge). Hier soll der Hashwert bei gleicher Eingabemenge stets den gleichen Wert haben. Bei einem biometrischen Hash hingegen bezieht sich die Reproduzierbarkeit auf die Eingabedaten einer bestimmten Person. Der Hashwert soll entsprechend der Person den gleichen Wert besitzen.

Kollisionsresistenz

Für kryptographische Hashfunktionen besteht weiterhin die Anforderung, dass zwei unterschiedliche Eingabemengen auch zwei voneinander verschiedene Ausgabemengen generieren sollen. Für biometrische Hashfunktionen verhält es sich ein wenig anders. Hier soll sich, ähnlich wie bei der Reproduzierbarkeit, die Ausgabemenge nicht auf Basis der Eingabedaten ändern, sondern in Abhängigkeit der Personen. Dies bedeutet, dass Eingabedaten von zwei unterschiedlichen Personen auch zu zwei verschiedenen Ausgabedaten führen.

Unumkehrbarkeit
Der kryptographische als auch der biometrische Hash besitzen für die Unumkehrbarkeit die gleichen Anforderungen. Diese drückt aus, dass es rechnerisch nicht möglich sein soll die Eingabedaten auf Basis der Hashwerte zurückzurechnen bzw. zu bestimmen.

Bitsensitivität
Bei einer kryptographische Hashfunktion sollte eine kleine Änderung der Eingabedaten eine möglichst große Änderung der Ausgabedaten aufweisen. Bei einer biometrischen Hashfunktion sollte eine kleine Änderung der Eingabedaten, wenn diese von derselben Person stammt, möglichst keine Änderung in der Ausgabemenge hervorrufen.

In Tabelle 2 sind die Anforderungen der beiden Funktionen im Überblick nochmals gegenübergestellt, wobei diese für beide Verfahren nicht immer erreicht werden. So kann es in der Kryptografie zumindest theoretisch zu Kollisionen kommen, wenn beispielsweise Eingabedaten einer nicht festgelegten Größe auf Ausgabedaten mit fixer Länge abgebildet werden sollen. Die kryptographische Hasherzeugung unterscheidet sich von der Generierung stabiler Hashwerte aus biometrischen Daten in den Eigenschaften (a) Reproduzierbarkeit, (b) Kollisionsfreiheit und (d) Bitsensibilität (siehe Tabelle 2).

Tabelle 2 Anforderungen an kryptographische und biometrische Hashfunktionen [KüVi10a]

Anforderung	**Kryptographische Hashfunktion**	**Biometrische Hashfunktion**
a) Reproduzierbarkeit	Sind zwei Eingabewerte *a* und *a'* identisch, so sollen auch die beiden durch ein und dieselbe kryptographische Hashfunktion *H* berechneten Werte *H(a)* und *H(a')* identisch sein. *H(a) = H(a'), wenn a = a'*	Stammen zwei biometrische Eingabedaten *a* und *a'* von derselben Person *P*, so sollen auch die beiden durch ein und dieselbe biometrische Hashfunktion *B* berechneten Werte *B(a)* und *B(a')* identisch sein. *B(a) = B(a'), wenn P(a) = P(a')*
b) Kollisionsresistenz	Sind zwei Eingabewerte *a* und *a'* ungleich, dann müssen auch die durch eine kryptographische Hashfunktion *H* berechneten Hashwerte *H(a)* und *H(a')* ungleich sein. *H(a) ≠ H(a'), wenn a ≠ a'*	Stammen zwei biometrische Eingabedaten *a* und *a'* von zwei unterschiedlichen Personen *P* und *P'*, dann müssen auch die durch eine biometrische Hashfunktion *B* berechneten Hashwerte *B(a)* und *B(a')* ungleich sein. *B(a) ≠ B(a'), wenn P(a) ≠ P'(a')*
c) Unumkehrbarkeit	Es sollte rechnerisch nicht möglich sein, aus dem durch die kryptographische Hashfunktion *H* erzeugten Hashwert *H(a)* wieder den Ausgangswert *a* zu bestimmen.	Es sollte rechnerisch nicht möglich sein, aus dem durch die biometrische Hashfunktion *B* erzeugten Hashwert *B(a)* einen Ausgangswert *a'* zu bestimmen, für den gilt *B(a)=B(a')*.
d) Bit-sensitivität	Kleine Änderungen der Eingabedaten *a* sollten zu möglichst großen Veränderungen der Ausgabedaten *H(a)* führen.	Änderungen der Eingabedaten *a* sollen nur dann keinen Einfluss auf die Ausgabedaten *B(a)* haben, wenn diese von ein und derselben Person stammen.

Während in der Kryptographie die Anforderung an eine Hashfunktion darin besteht nur aus identischen Eingabewerten auch identische Hashwerte zu erzeugen, besteht das Problem in der Biometrie in unscharfen Eingabedaten. Dies resultiert daraus, dass

biometrische Daten einer Person von einem Aufnahmezeitpunkt zum nächsten variieren (Intraklassen-Variabilität) bzw. Daten verschiedener Personen sehr ähnlich sein können (Interklassen-Ähnlichkeit), siehe hierzu Abschnitt 4.1.4. Eine biometrische Hashfunktion muss demzufolge einerseits in der Lage sein, identische Hashwerte aus variierenden Daten einer einzelnen Person zu generieren (Abweichung der Bitsensibilität und Kollisionsresistenz bezüglich einer Person, um Reproduzierbarkeit für diese Person zu erreichen). Auf der anderen Seite muss sie aber auch unterschiedliche Hashwerte für unterschiedliche Personen erzeugen können (Kollisionsresistenz bei verschiedenen Personen) [KüVi10a].

5 Klassifizieren von Angriffen (FA1)

Die Forschungsaufgabe 1 (FA1), welche in Abschnitt 3.1 formuliert ist, soll in diesem Abschnitt behandelt werden. Hierfür wird zunächst in Abschnitt 5.1 ein neues Klassifikationsverfahren aufbauend auf bestehende Verfahren beschrieben. Anschließend werden im Abschnitt 5.2 ausgewählte Angriffsverfahren mittels der neu entwickelten Klassifikationsmethode exemplarisch klassifiziert. Ferner wird eine ausgewählte Angriffsmethode adaptiert und auf den biometrischen Verifikationsalgorithmus [Viel06] angewendet. Im letzten Abschnitt dieses Kapitels werden die experimentellen Tests dieses adaptierten Angriffsverfahrens präsentiert und die Ergebnisse entsprechend bewertet.

5.1 Vorgehensweise und Methodik

Wie in Abschnitt 3.1 beschrieben, sollen innerhalb der Forschungsaufgabe 1 bekannte Angriffsverfahren auf biometrische Erkennungssysteme hinsichtlich ihrer potentiellen Gefahr eingeordnet bzw. klassifiziert werden. Somit kann ein Angriffsverfahren hinsichtlich des Gefahrenpotentials und mögliche Adaptierbarkeit auf andere biometrische Verfahren besser eingeschätzt werden. In Abschnitt 5.1.1 wird hierfür ein neues geeignetes Klassifizierungsverfahren ermittelt unter Berücksichtigung der in Abschnitt 2.3 vorgestellten Methoden. Anschließend wird im Abschnitt 5.1.2 die Vorgehensweise der Klassifikation verschiedener Angriffsverfahren beschrieben. Zusätzlich wird ein bekanntes Angriffsverfahren ausgewählt, welches auf einen speziellen handschriftenbasierten Verifikationssystem [Viel06] adaptiert und evaluiert werden sollen.

5.1.1 Methoden zur Klassifikation von Angriffen

Die in Abschnitt 2.2 vorgestellten Modelle zur Beschreibung der Angriffspunkte sollen für eine erste Einordnung der Angriffsverfahren herangezogen werden. Jedes auf ein biometrisches System angepasstes oder entwickeltes Angriffsverfahren kann einer der Angriffspunkte innerhalb der vorgestellten Modelle zugeordnet werden. Unter Anbetracht der in Abschnitt 2.2 vorgestellten Modelle können drei verschiedene Modelle als Grundlage für die Einordnung herangezogen werden. Nachfolgend werden nochmals die genannten Modelle kurz beschrieben.

[RaCB01] bzw. [Obie06]
Das von Ratha et al. in [RaCB01] beschriebene Modell besitzt acht Angriffspunkte, welche auf die Hauptkomponenten eines biometrischen Systems aufsetzen. Dieses wurde von Obied in [Obie06] um einen weiteren Angriffspunkt, welcher auf die nachgeschaltete Anwendung zielt, erweitert.

[BaCu05] und [CuBa05]
Bartlow und Cukic fügen in ihren Arbeiten [BaCu05] und [CuBa05] drei weitere Komponenten hinzu (Administrative Ebene, IT-Umgebung und Token Präsentation). In Ihren Arbeiten definieren sie insgesamt 20 Angriffspunkte und 22 Schwachstellen, welche innerhalb eines biometrischen Systems existieren bzw. auftreten können.

[Waym1999]
Das von Wayman in [Waym1999] vorgestellte Modell ermöglicht eine Makro- und Mikrosichtweise eines biometrischen Verifikationssystems. Es besteht aus fünf Subsystemen mit zum Teil mehreren Prozessen, welche potentielle Angriffspunkte darstellen können.

Die zuletzt genannten Modelle von Bartlow und Cukic sowie von Waymen ermöglichen eine detailliertere Einordnung der Angriffspunkte als das von Ratha et al. vorgeschlagene Modell. Jedoch sind acht Angriffspunkte für eine grobe Einordnung biometrischer Angriffsverfahren aus Sicht des Autors dieser Arbeit hinreichend. Eine weitere Einteilung kann sicherlich auf Basis der von Bartlow und Cukic sowie von Waymen entwickelten Modelle durchgeführt werden, würde jedoch den Klassifizierungsbaum[2] erheblich aufblähen und unübersichtlich gestalten. Die weiterführende Einteilung des von Ratha et al. eingeführten Modells in einen neunten Angriffspunkt auf die nachfolgende Anwendung [Obie06] wird nicht betrachtet, da in dieser Arbeit speziell die Angriffe auf biometrische Systeme betrachtet werden sollen und nicht die nachgeschaltete Anwendung. Für die Klassifikation biometrischer Angriffe werden demzufolge die acht Angriffspunkte aus [RaCB01] verwendet und stellen die erste Stufe der Einordnung dar.

5.1.2 Klassifikation von Angriffen

In Abschnitt 2.3.1 wurden unter anderem direkte und indirekte Angriffsarten definiert. Wobei die direkten Angriffsarten Angriffspunkt 1 (AP 1) und die indirekten Angriffsarten Angriffspunkt zwei bis acht (AP 2-8) zugeordnet werden können. Die Einordnung der Angriffspunkte in direkte und indirekte Angriffsarten stellt die zweite Stufe der Klassifizierung biometrischer Angriffe dar. Weiterhin können indirekte Angriffsarten, wie im Abschnitt 2.3.3 beschrieben, in mindestens sechs weitere Kategorien unterteilt werden (Seitenkanal, Hill-Climbing, Replay, Tampering, Overriding Response und Maskerade). Diese Unterteilung indirekter Angriffe stellt die dritte und letzte Stufe der Klassifizierung dar.

Tabelle 3 Angriffsmethoden und Angriffstechniken

Angriffsmethode	Angriffstechnik	Angriffsnummer
Direkt	Spoofing	1
	Nachahmen	2
Indirekt	Seitenkanal	1
	Hill-Climbing	2
	Replay	3
	Tampering	4
	Ergebnis Überschreiben	5
	Maskerade	6
	Stellvertreter	7
	Andere	8

Diese drei Stufen der Klassifizierung stellen eine neue Methode zur Einordnung von Angriffsverfahren dar. In Abbildung 26 werden die verschiedenen Klassen der Angriffe mittels eines Entscheidungsbaums dargestellt. Am Ende der Entscheidung / Klassifizierung

[2] Ein Klassifizierungsbaum bzw. Entscheidungsbaum wird u.a. verwendet, um hierarchisch aufeinanderfolgende Entscheidungen mit Hilfe eines Baumdiagramms zu visualisieren siehe z.B. auch [KaJS18].

wird ein jeweiliger Angriff in eine Angriffsklasse (AK) eingeordnet. Wobei die nachfolgenden Ziffern für die jeweilige Angriffsart (direkt/indirekt), Angriffstechnik (Spoofing, Nachahmen, Seitenkanal, Hill-Climbing, Maskerade usw.) und die Angriffspunkte (AP 1 bis AP 8) stehen. Die Angriffsmethoden und Angriffstechniken sind mit der jeweiligen Nummerierung in Tabelle 3 dargestellt.
Da direkte Angriffe nur am Angriffspunkt (AP 1) auftreten, wird explizit nicht mehr der jeweilige Angriffspunkt angegeben. Wird beispielsweise ein Angriff auf Angriffspunkt (AP 1) durchgeführt und mittels Nachahmung versucht das biometrische System zu täuschen, handelt es sich um einen Angriff der Angriffsklasse AK 1.2. Indirekte Angriffe werden in Seitenkanal-, Hill-Climbing-, Replay-, Tampering, Ergebnisüberschreibende, Maskerade und Stellvertreter Angriffe unterschieden. Zusätzlich werden Angriffe, die nicht unter diese Kategorien fallen in „Andere Angriffe“ zusammengefasst. Beispielsweise fällt ein Brute-Force-Angriff unter diese Kategorie.

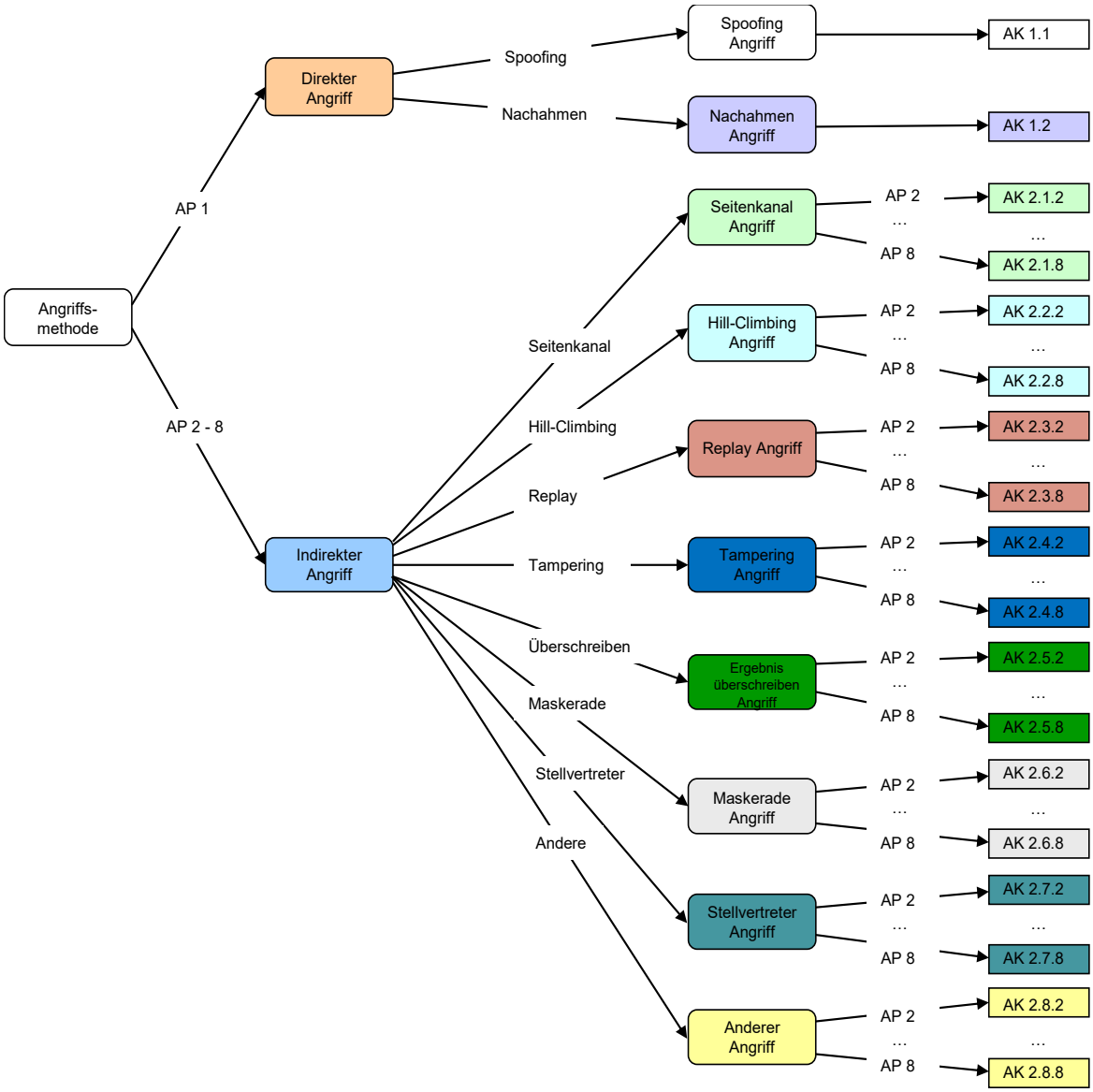


Abbildung 26 Neu eingeführter Entscheidungsbaum zur Klassifizierung der Angriffsarten

Tabelle 4 Tabellarische Auflistung der neuen Angriffsklassen

	Angriffsart	Angriffstechnik	Angriffsklasse	Angriffspunkt (AP)
Angriffsmethode	Direkter Angriff	Spoofing Angriff	AK 1.1	AP 1
		Nachahmen Angriff	AK 1.2	AP 1
	Indirekter Angriff	Seitenkanal Angriff	AK 2.1.2	AP 2
			AK 2.1.3	AP 3
			AK 2.1.4	AP 4
			AK 2.1.5	AP 5
			AK 2.1.6	AP 6
			AK 2.1.7	AP 7
			AK 2.1.8	AP 8
		Hill-Climbing-Angriff	AK 2.2.2	AP 2
			AK 2.2.3	AP 3
			AK 2.2.4	AP 4
			AK 2.2.5	AP 5
			AK 2.2.6	AP 6
			AK 2.2.7	AP 7
			AK 2.2.8	AP 8
		Replay Angriff	AK 2.3.2	AP 2
			AK 2.3.3	AP 3
			AK 2.3.4	AP 4
			AK 2.3.5	AP 5
			AK 2.3.6	AP 6
			AK 2.3.7	AP 7
			AK 2.3.8	AP 8
		Tampering Angriff	AK 2.4.2	AP 2
			AK 2.4.3	AP 3
			AK 2.4.4	AP 4
			AK 2.4.5	AP 5
			AK 2.4.6	AP 6
			AK 2.4.7	AP 7
			AK 2.4.8	AP 8
		Ergebnis überschreiben Angriff	AK 2.5.2	AP 2
			AK 2.5.3	AP 3
			AK 2.5.4	AP 4
			AK 2.5.5	AP 5
			AK 2.5.6	AP 6
			AK 2.5.7	AP 7
			AK 2.5.8	AP 8
		Maskerade Angriff	AK 2.6.2	AP 2
			AK 2.6.3	AP 3
			AK 2.6.4	AP 4
			AK 2.6.5	AP 5
			AK 2.6.6	AP 6
			AK 2.6.7	AP 7
			AK 2.6.8	AP 8
		Stellvertreter Angriff	AK 2.7.2	AP 2
			AK 2.7.3	AP 3
			AK 2.7.4	AP 4
			AK 2.7.5	AP 5
			AK 2.7.6	AP 6
			AK 2.7.7	AP 7
			AK 2.7.8	AP 8
		Andere Angriffe	AK 2.8.2	AP 2
			AK 2.8.3	AP 3
			AK 2.8.4	AP 4
			AK 2.8.5	AP 5
			AK 2.8.6	AP 6
			AK 2.8.7	AP 7
			AK 2.8.8	AP 8

Dementsprechend können nach dieser Definition Angriffe in 58 verschiedene einfache Angriffsklassen unterschieden werden. In Tabelle 4 werden alle Angriffsklassen zusammenfassend aufgelistet.

Mit Hilfe des neu eingeführten Entscheidungsbaums für Angriffe steht ein leicht anzuwendendes Hilfsmittel für eine einfache Klassifizierung bereit. Ähnlich wie bei der in Abschnitt 2.3.5 vorgestellten CERT-Taxonomy zur Einordnung von Angriffen kann diese neue Klassifizierung eingesetzt werden, um z.B. Informationen über neue Angriffstechniken auszutauschen, wobei diese neue Klassifizierung speziell für biometrische Systeme ausgelegt ist und sich auf Angriffspunkt und Angriffstechnik beschränkt. Ist ein neuer Angriff eingeordnet, können geeignete Gegenmaßnahmen schneller umgesetzt werden. Gegebenenfalls existieren für bestimmte Angriffspunkte bereits Gegenmaßnahmen anderer biometrischer Systeme die übernommen und/oder adaptiert werden können. Eine Einordnung ausgewählter Angriffsmethoden mittels des neu eingeführten Entscheidungsbaums wird im folgenden Abschnitt erläutert und in Abschnitt 5.2.2 exemplarisch durchgeführt.

Zusammenfassung
In diesem Abschnitt wurde eine neue Methode zur Klassifikation von Angriffen auf biometrische Systeme vorgestellt. Die Klassifikation wird bei dieser Methode in drei Stufen gegliedert. Zum einen wird in direkte und indirekte Angriffe unterschieden (Stufe 1). Anschließend wird die verwendete Angriffstechnik (Spoofing, Replay, Hill-Climbing, Seitenkanal etc.) des Angriffes bestimmt (Stufe 2). Im letzten Schritt (Stufe 3) der Klassifikation wird der Angriffspunkt (AP1 bis AP8) ermittelt. Um potentielle Angriffe zu klassifizieren, wurde ein Klassifikationsbaum erstellt.

5.2 Durchführung

Das im Abschnitt 5.1.2 vorgestellte neue Klassifikationsverfahren soll hier erstmals exemplarisch angewendet werden. Hierfür werden bekannte biometrische Angriffsverfahren ausgewählt und klassifiziert. Im nachfolgenden Abschnitt 5.2.1 werden die Kriterien für die Wahl der jeweiligen Angriffsverfahren erläutert. Anschließend werden im Abschnitt 5.2.2 die ausgewählten Angriffe kurz beschrieben, klassifiziert und deren Gefahrenpotential für andere biometrische Systeme eingeschätzt.
Eine weitere Teilaufgabe der Forschungsaufgabe 1 (FA1) sieht vor, dass eine ausgewählte bekannte Angriffstechnik auf den im Abschnitt 4.2 beschriebenen handschriftenbasierten Verifikationsalgorithmus adaptiert und angewendet werden soll. Die Beschreibung des Angriffes und deren Adaption wird in Abschnitt 5.2.3 dargestellt.

5.2.1 Auswahl der Angriffe

In den letzten Jahren wurden viele Schwachstellen und Angriffsverfahren auf biometrische Systeme aufgedeckt bzw. vorgestellt, siehe Abschnitt 2.3. Jede einzelne Angriffsmethode zu klassifizieren würde den Rahmen dieser Arbeit sprengen. Aus diesem Grund werden nur einige ausgewählte Angriffsverfahren für eine beispielhafte Klassifizierung herangezogen. Bei der Auswahl der Angriffsmethoden wurden folgende drei Bedingungen berücksichtigt, um eine möglichst große Vielfalt zu präsentieren:

- Für nahezu alle gängigen biometrischen Modalitäten (Fingerabdruck, Gesicht, Iris, Handabdruck, Venenverlauf einer Hand, Gangart, Handschrift/Unterschrift,

Stimme, Tippverhalten) soll mindestens eine Angriffsmethode zur Klassifizierung herangezogen werden.
- Des Weiteren sollen für die Angriffstechniken Spoofing, Nachahmen, Tampering, Seitenkanal, Hill-Climbing und Maskerade) mindestens ein Vertreter klassifiziert werden. Angriffsarten zu Replay, Ergebnisüberschreibende- und Stellvertreterangriffe werden nicht klassifiziert, da dem Autor zum Zeitpunkt der Erstellung der Arbeit keine dieser Angriffsarten auf biometrische Verifikationssysteme bekannt waren.
- Zusätzlich sollen für die Angriffspunkte AP 1, AP 2, AP 4, AP 5, AP 7 und AP 8 mindestens ein Angriffsverfahren exemplarisch ausgewählt werden. Es konnten keine aktuellen Angriffsmethoden ermittelt werden, die auf Angriffspunkt 3 (Merkmalsextraktor modifizieren) und Angriffspunkt 6 (Referenzdaten/ Referenzdatenbank ändern) zielen. Das liegt ggf. an der Art der Technik, die für eine solche Angriffsmethode eingesetzt werden müsste. Datenbanken oder Systemkomponenten manipulieren sind Themenfelder der "klassischen" IT-Sicherheit, die nicht erst mit der Einführung von biometrischen Erkennungssystemen bekannt wurden. Entsprechend sind diese Angriffspunkte für Wissenschaftler aus dem Bereich der Biometrie wohl eher unattraktiv. Letzteres sind persönliche Einschätzungen und Annahmen des Autors.

Weitere Auswahlkriterien, die an dieser Stelle jedoch nicht bei der Auswahl berücksichtigt wurden, sind u.a. die Aktualität des Angriffsverfahrens, die Auswirkung des Angriffs auf einen oder mehrere Nutzerdaten (ggf. zusätzliche kompromittierte personenbezogenen Daten) und ob bei der Vorstellung/Evaluation der Angriffsmethode eine frei verfügbare biometrische Testdatenbank verwendet wurde.

Des Weiteren kann es durchaus vorkommen, dass für einige Erkennungssysteme bestimmter biometrischer Modalitäten keine Angriffsmethoden existieren bzw. nicht bekannt sind und deshalb nicht mit aufgenommen worden sind.

Direkte Angriffsverfahren wie Spoofing und Nachahmen sind in der Regel nur für die bestimmte biometrische Modalität verwendbar und somit nicht auf andere übertragbar. So kann eine Technik für die Erzeugung eines künstlichen Fingerabdruckes nicht für die Erzeugung von künstlichen Handschriften angewendet werden. Mit Blick auf die Forschungsaufgabe 1 (FA1) werden direkte Angriffsmethoden von nicht handschriftenbasierten Verifikationssystemen zwar in Abschnitt 5.2.2 betrachtet, jedoch nicht in die Auswahl der in Abschnitt 5.2.3 adaptierten Verfahren aufgenommen.

5.2.2 Klassifikation von Ausgewählten Angriffsverfahren

Nachfolgend werden beispielhaft einige Angriffstechniken mittels im Abschnitt 5.1.2 eingeführten Klassifikationsverfahren eingeordnet und anschließend bewertet, inwieweit diese auf handschriftenbasierte Verifikationssysteme adaptiert werden können. Hierfür werden allgemeine Informationen, Klassifikationsbeschreibung und Adaptierbarkeit der jeweiligen Angriffsverfahren in einem Steckbrief (Tabelle) unter folgenden Punkt zusammengefasst:

1. **Angriffsnummer:** unabhängige fortlaufende Nummer der Angriffsverfahren
2. **Biometrische Modalität:** beschreibt die biometrische Modalität(en), die vom angegriffenen biometrischen Verifikationsverfahren verwendet wird bzw. werden
3. **Autoren/Quelle:** Gibt den Hauptautor und die Quelle der (wissenschaftlichen) Publikation an, in der das Angriffsverfahren vorgestellt und beschrieben wird
4. **Kurzbeschreibung:** gibt einen knappen Überblick über das Angriffsverfahren
5. **Klassifikation:** beschreibt, wie der Angriff mittels Entscheidungsbaum klassifiziert wird
6. **Adaptierbarkeit:** erläutert, inwiefern das Angriffsverfahren auf ein handschriftenbasiertes Verifikationssystem und im Allgemein auf andere biometrische Verifikationssysteme adaptiert werden kann
7. **Angriffsklasse:** gibt das Ergebnis der Klassifikation in Form der ermittelten Angriffsklasse aus Tabelle 4 an
8. **Gefahrenpotential Handschrift:** gibt einen Gefahrenwert zwischen 1 und 5 an, welcher bestimmt, inwieweit sich die Angriffstechnik auf ein dynamisches handschriftenbasiertes Verifikationssystem adaptieren lässt und somit ein gewisses Gefahrenpotential beschreibt (Gefahrenpotential in fünf Stufen: 1 - gering, 2 – mäßig, 3 - erheblich, 4 – hoch, 5 sehr hoch).
9. **Allg. Gefahrenpotential:** gibt einen Gefahrenwert zwischen 1 und 5 an, welcher bestimmt, inwieweit sich die Angriffstechnik auf andere biometrische Verifikationssystem im Allgemeinen adaptieren lässt und entsprechend ein gewisses Gefahrenpotential beschreibt (Gefahrenpotential in fünf Stufen: 1 - gering, 2 – mäßig, 3 - erheblich, 4 – hoch, 5 sehr hoch).

Unter Punkt 8 werden fünf Stufen für Gefahrenpotentiale für handschriftenbasierte Verifikationssysteme benannt. In Tabelle 5 wird beschrieben, unter welchen Bedingungen diese Klassifikation eines Angriffsverfahrens durchgeführt wird.

Tabelle 5 Gefahrenpotentiale für handschriftenbasierte Verifikationssysteme

Gefahrenpotential	Klassifizierungsvoraussetzung
Stufe 1 (gering)	Das Angriffsverfahren ist nicht für die dynamische Handschrift anwendbar.
Stufe 2 (mäßig)	Die Angriffstechnik ist potentiell auf die dynamische Handschrift anwendbar jedoch nicht direkt und muss entsprechend adaptiert werden.
Stufe 3 (erheblich)	Das Angriffsverfahren ist potentiell direkt auf die dynamische Handschrift anwendbar.
Stufe 4 (hoch)	Die Angriffstechnik ist für die dynamische Handschrift konzipiert worden und kann auch auf andere Handschriftenverifikationsalgorithmen indirekt übertragen werden.
Stufe 5 (sehr hoch)	Die Angriffstechnik ist für die dynamische Handschrift konzipiert worden und kann direkt auf andere handschriftenbasierte Verifikationsalgorithmen angewendet werden.

Die unter Punkt 9 gelisteten fünf Gefahrenpotentiale und deren Anwendung bei der Klassifizierung können wie in Tabelle 6 dargestellt beschreiben werden.

Tabelle 6 Gefahrenpotentiale für andere biometrische Verifikationssysteme

Gefahrenpotential	Klassifizierungsvoraussetzung
Stufe 1 (gering)	Das Angriffsverfahren ist nur für eine biometrische Modalität und einen bestimmten Verifikationsalgorithmus anwendbar.
Stufe 2 (mäßig)	Die Angriffstechnik ist nur für eine biometrische Modalität einsetzbar kann jedoch auch auf andere Verifikationsalgorithmen/-verfahren angewendet werden. Die Adaption der Angriffstechnik auf andere Verifikationsverfahren ist jedoch schwierig.
Stufe 3 (erheblich)	(a) Die Angriffstechnik kann nur auf eine biometrische Modalität angewendet werden, jedoch ist die Adaption der Angriffstechnik auf andere Verifikationsverfahren relativ leicht. (b) Andere biometrische Modalitäten und Verifikationsalgorithmen können mit dieser Angriffstechnik ebenfalls kompromittiert werden. Die Adaption der Angriffstechnik auf andere Verfahren und Modalitäten ist jedoch schwierig.
Stufe 4 (hoch)	Das Angriffsverfahren kann auf andere biometrische Modalitäten und Verifikationsverfahren angewendet werden, wobei die Adaption auf andere Modalitäten und Verfahren relativ einfach erscheint.
Stufe 5 (sehr hoch)	Die Angriffstechnik ist auf nahezu alle biometrischen Modalitäten und Verifikationssystem anwendbar und zum Teil leicht zu implementieren.

Die Klassifikation der Gefahrenpotentiale in Tabelle 5 und Tabelle 6 sind explizit hierfür erstellt worden und sollen beschreiben, inwiefern eine Angriffsmethode auf ein handschriftenbasiertes Verifikationssystem bzw. ein anderes biometrisches Verifikationssystem adaptiert werden kann.
Aus Gründen der Lesbarkeit dieses Abschnittes werden beispielhaft zwei Steckbriefe dargestellt. Eine vollständige Liste der ausgewählten und klassifizierten Angriffe ist im Anhang der Arbeit in Anlage 2 zu finden.

1	**Angriff Nr.**	1						
2	**Biometrische Modalität**	Gesicht						
3	**Autoren/Quelle**	Thalheim et al / [ThKZ02]						
4	**Kurzbeschreibung**	Bei der Datenerfassung der Gesichtserkennungssoftware, welche eine normale Internetkamera als Sensor verwendet, werden ein aufgezeichnetes Video bzw. Bilder einer Person vor dem Sensor präsentiert (Notebook dient als Abspielgerät). Die verwendete Verifikationssoftware konnte mit dieser Methode relativ einfach getäuscht werden.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff. Er verwendet eine einfache Spoofing Methode und kann somit der Angriffsklasse AK 1.1 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis eines optischen Sensors basiert, kann es nicht auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett, technische Arbeitsweise siehe Abschnitt 4.2.1, für die Datenaufzeichnung verwendet. Jedoch ist es möglich, ein statisches handschriftenbasiertes Verifikationssystem mittels dieser Technik zu täuschen. Bei solch einen Verfahren werden Bilder von Handschriftendaten für eine Verifikation herangezogen. Da für die dynamische handschriftenbasierte Verifikation nicht anwendbar, wird das Gefahrenpotential als sehr gering eingestuft (Stufe 1). Die Angriffsmethode ist jedoch recht einfach auf alle biometrischen Verifikationssysteme anwendbar, welche einen optischen Sensor (CMOS Kamera o.ä.) verwenden wie beispielsweise bei der Aufnahme der Iris oder von der Kontur der Hand. Weshalb der Angriff im Allgemein in Stufe 3 eingestuft werden muss, da er teils recht einfach umzusetzen ist, solange keine Gegenmaßnahmen (Lebenderkennung etc.) getroffen wurden.						
7	**Angriffsklasse**	AK 1.1	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	12						
2	**Biometrische Modalität**	Fingerabdruck						
3	**Autoren/Quelle**	Cappelli et al. / [CMLM07]						
4	**Kurzbeschreibung**	In Ihrer Arbeit beschreiben die Autoren, wie sie auf Basis von Referenzdaten künstliche Fingerabdrücke erzeugen. Dabei verwenden sie relativ wenige Informationen, welche in den Referenzdaten hinterlegt sind, um einen künstlichen Fingerabdruck zu erzeugen. Ziel ist es, ein Fingerabdruckbild zu erzeugen welcher den originalen ursprünglichen Abdruck ähnelt und die gleichen Referenzdaten generiert, um eine positive Verifikation am System zu bewirken.						
5	**Klassifikation**	Die Angriffsdaten werden auf Basis von biometrischen Referenzdaten erzeugt (Maskerade) und am Angriffspunkt 2 eingespielt. Demnach kann dieser Angriff der Angriffsklasse AK 2.6.2 zugeordnet werden.						
6	**Adaptierbarkeit**	Die Methode zur Erzeugung von künstlichen Fingerabdrücken kann nicht direkt auf ein handschriftenbasiertes Verifikationssystem angewendet werden. Jedoch ist die Strategie, künstlich erzeugte biometrische Daten auf Basis weniger Daten der Referenzdaten zu generieren (Maskerade), interessant. Außerdem wurde dieser Angriff erfolgreich auf neun verschiedenen Verifikationssystemen für die Modalität Fingerabdruck durchgeführt und zeigt die Gefährlichkeit für diese. Die Idee charakteristische Merkmale (Positionen der Minutien) zu verwenden um Fingerabdrücke zu rekonstruieren kann potentiell auf andere biometrische Verfahren adaptiert werden. Voraussetzung hierfür ist jedoch, dass diese Informationen in den Referenzdaten enthalten sind bzw. aus denen ermittelt werden können. Aus diesem Grund wird das Gefahrenpotential dieser Methode als erheblich (Stufe 3) für Verifikationssysteme im Allgemeinen als auch für handschriftenbasierte Verifikationssystem eingeschätzt.						
7	**Angriffsklasse**	AK 2.6.2	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

Die Angriffsmethode mit der Angriffsnummer 12 von Cappelli et al. soll exemplarisch adaptiert und auf den dynamischen Handschriftenverifikationsalgorithmus [Viel06] angewendet werden. Der Grund für die Wahl ist u.a. die Tatsache, dass nur wenige Informationen aus den Referenzdaten genügten, um gefälschte Fingerabdruckbilder zu erzeugen, welche den ursprünglichen originalen Fingerabdruckbildern sehr ähneln.
Das zeigt, dass in einigen Fällen wenige Informationen ausreichen, um akzeptable Replikationen biometrischer Daten zu erzeugen, die ein aktuelles Verifikationssystem täuschen können.
Im nachfolgenden Abschnitt 5.2.3 wird die adaptierte Angriffstechnik auf den handschriftenbasierten Verifikationsalgorithmus beschrieben.

5.2.3 Adaptieren einer ausgewählten Angriffstechnik

In diesem Abschnitt wird zuerst das Verfahren zur Erzeugung künstlicher Fingerabdrücke von Cappelli et al. in [CMLM07] und deren jeweilige Schritte beschrieben. Anschließend wird auf Basis dieser Methode die Angriffstechnik beschrieben, welche der Autor einsetzt um künstliche Handschriftensignale zu generieren. Diese Angriffstechnik wurde bereits vom Autor und Vielhauer in [KüVi10a] vorgestellt. Hier soll die Herangehensweise der Adaption erstmals im Detail vorgestellt werden.

Ausgewählte Angriffsmethodik von Cappelli et al. [CMLM07]

Zunächst soll dargestellt werden, wie der Angriff von Cappelli et al. [CMLM07] im Grundsatz funktioniert. Grundlage für die Konstruktion der Fingerabdruckbilder bilden die Referenzdaten, welche in ISO/IEC 19794-2 Format vorliegen. Das ISO/IEC 19794-2 Format wurde eingeführt, um einen Standard für den Austausch von biometrischen Daten zu schaffen [ISOI05]. Somit können beispielsweise Referenzdaten auf einer Smartcard transportiert und von Verifikationssystemen unterschiedlicher Hersteller verwendet werden. Cappelli et al. beziehen sich auf die ISO/IEC 19794-2 aus dem Jahr 2005 [ISOI05].
Die im ISO/IEC 19794-2 beschriebenen kompakten Datenformate beinhalten wesentliche Informationen der Fingerabdruckreferenzdaten und konzentrieren sich auf eine möglichst kompakte Kodierung der für den Merkmalsvergleich notwendigen mathematischen Größen. Für die Erzeugung synthetischer Fingerabdruckbilder verwenden sie in ihrer Angriffsmethode folgende Information der Referenzdaten:

- Vertikale und horizontale Bildgröße (in Anzahl der Pixel)
- Vertikale und horizontale Auflösung (in Pixel pro Zentimeter), bei üblichen Fingerabdrucksensoren sind horizontale und vertikale Auflösung in den meisten Fällen identisch,
- Anzahl der Minutien (0 - 255),
- Minutientyp (Gabelung, Terminierung oder andere), Position (x, y) und Ausrichtung für jeden Minutien und
- Kern und Delta Position (wenn vorhanden).

Die Vorgehensweise des Angriffes besteht aus einer Abfolge verschiedener Schritte.
Im ersten Schritt wird der Fingerabdruckbereich des Bildes gebildet. Dieser Bereich stellt den Bereich dar, der vom Sensor erfasst wurde. Im zweiten Schritt wird das Ausrichtungsbild (Orientation Image) auf Basis der Ausrichtungen aller Minutien gebildet. Es beschreibt die lokale Ausrichtung der Papillarleisten (charakteristischen Linien in der Haut) des Fingerabdrucks. Gleichzeitig werden die Minutien in das Ausrichtungsbild mit eingebettet. Im dritten Schritt wird auf Basis des Ausrichtungsbildes die Struktur (Pattern) des Fingerabdrucks, also der Papillarleisten gebildet. Um aus dem "perfekt" aussehenden Papillarleistenbild eine realistisch aussehende Rekonstruktion eines Fingerabdruckbildes zu gestalten, werden im vierten Schritt dem Bild "Störungen" hinzugefügt. Im fünften Schritt werden das Abbild des Fingerabdrucks und der Bereich des Fingerabdrucks (aus Schritt 1) zusammengeführt. In Abbildung 27 sind die jeweiligen Schritte des Ablaufs dargestellt.

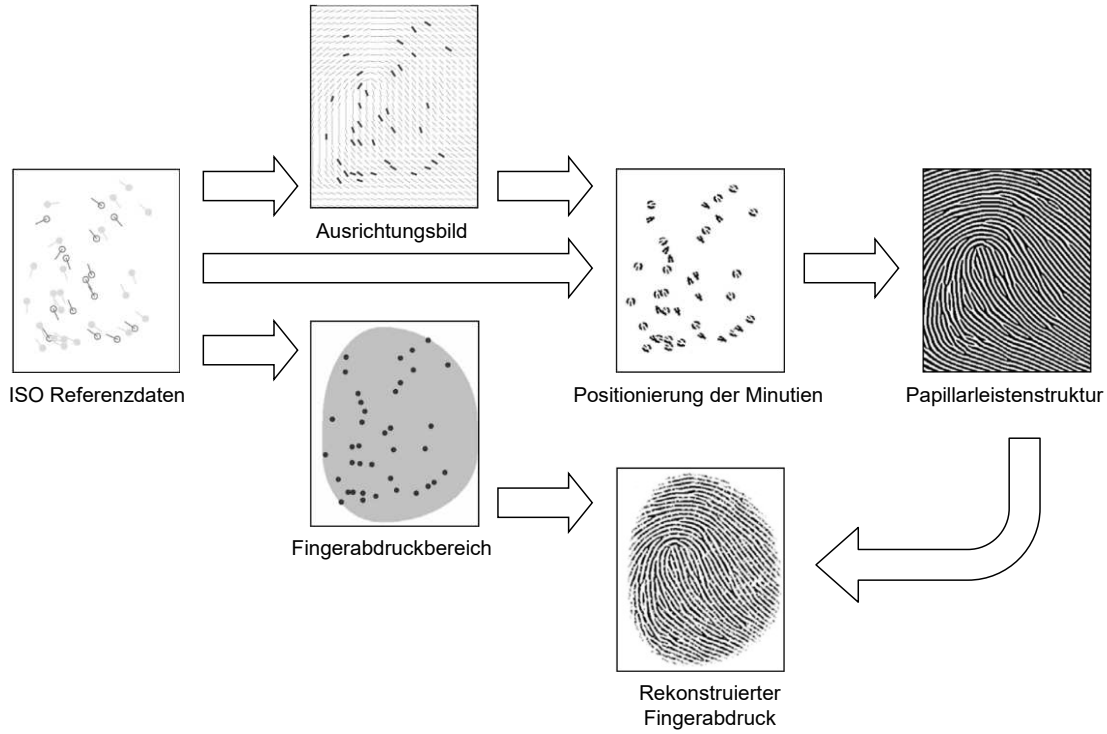


Abbildung 27 Ablauf der Rekonstruktion von Fingerabdrücken übersetzt aus [CMLM07]

In ihrer Arbeit evaluieren Cappelli et al. dieses Verfahren an neun Verifikationssystemen. Sie konnten unter Verwendung verschiedener Sicherheitseinstellungen (security levels; FMR=1,0%, FMR 0,1% und FMR=0,0%) im Durchschnitt eine sehr hohe erfolgreiche Angriffsrate zwischen 79,3% und 99,63% erzielen. Weiterhin beschreiben die Autoren, dass es zwar möglich ist, ein Verifikationssystem mittels dieser Angriffstechnik zu täuschen, jedoch sind die Fingerabdruckbilder nicht in der Lage, einen Experten aus dem Bereich der Daktyloskopie zu täuschen. Beispielsweise sind die Form bzw. die Gestalt der lokalen Minutien in den Referenzdaten nicht hinterlegt und können dementsprechend nicht rekonstruiert werden. Bei einem direkten Vergleich mit dem originalen Fingerabdruck könnte ein Experte leicht feststellen, dass diese Minutienformen nicht übereinstimmen.

Die vorgestellten Voraussetzungen und Arbeitsschritte können wie folgt zusammengefasst werden (Voraussetzungen/Arbeitsschritte in Klammern)

1. Als Basis dienen biometrische Referenzdaten in einem bestimmten Format (Referenzdaten im ISO/IEC 19794-2 Format).
2. Anhand der Referenzdaten können Rückschlüsse auf die originalen biometrischen Daten gezogen werden (Merkmale der Referenzdaten).
3. Auf Basis der Rückschlüsse wird eine erste grobe künstliche Struktur eines rekonstruierten biometrischen Datums erzeugt (Schritt 1 Fingerabdruckbereich erstellen).
4. In einem oder mehreren anschließenden Schritten werden weitere Modifikationen durchgeführt, um das künstliche biometrische Datum realer bzw. echter wirken zu lassen (Schritt 2, 3, und 4).

Die vier Arbeitsschritte bzw. Voraussetzungen der Angriffsmethode können auf den Bio-Hash-Algorithmus und den bekannten Schwachstellen (siehe Abschnitt 4.2.4) übertragen

werden. In der Tabelle 7 sind die Voraussetzungen und Arbeitsschritte der beiden Verfahren nebeneinandergestellt, welche vom Autor so noch nicht gezeigt wurden.

Tabelle 7 Arbeitsschritte und Voraussetzungen beider Angriffsmethode

Vorrausetzung bzw. Arbeitsschritt	[CMLM07]	Adaptiert
Biometrische Referenzdaten liegen in einem bestimmten bekannten Format vor.	Referenzdaten im ISO/IEC 19794-2 Format	Intervallmatrix und BioHash-Verktor
Schwachstelle(n) des Verfahrens / der Referenzdaten, welche genutzt werden, um Rückschlüsse auf die originalen biometrischen Daten zu ziehen.	Merkmale enthalten verwertbare Informationen	Merkmalsvektor aus Intervallmatrix und BioHash berechenbar
Erzeugen einer ersten groben künstlichen Struktur eines rekonstruierten biometrischen Datums.	Schritt 1 Fingerabdruckbereich erstellen	Schritt 2 Rohdatengrundstruktur erstellen
In einem oder mehreren weiteren Schritten werden Modifikationen durchgeführt, um dem originalen biometrischen Datum näher zu kommen bzw. das künstliche biometrische Datum "realer/echter" wirken zu lassen.	Schritt 2, 3, und 4	Schritt 3 Weiterer Merkmale integrieren

Adaptierte Angriffsmethodik
Nachfolgend werden die drei Schritte des adaptierten Angriffsverfahrens detaillierter beschrieben, welche bereits vom Autor in [KüVi10a] vorgestellt wurden.

Aufgrund einer bekannten Schwachstelle des BioHash-Algorithmus (siehe Abschnitt 4.2.4) kann anhand einer Intervallmatrix und dem zugehörigen BioHash ein für die Verifizierung gültiger Merkmalsvektor erzeugt werden (siehe Formel 18). Das kann jedoch nur geschehen, solange ein potentieller Angreifer im Besitz der Intervallmatrix und zugehörigem BioHash ist. Der Merkmalsvektor besitzt Merkmale des ursprünglichen Handschriftensignals, die es ermöglichen, daraus ein künstliches Handschriftensignal zu erzeugen. Ein potentieller Angreifer muss außerdem den Kontext der jeweiligen Merkmale des Merkmalsvektors verstehen, um ein Rohdatensignal zu erzeugen.

Tabelle 8 Klassifizierung der verwendeten Merkmale zur Konstruktion eines Handschriftenbildes [KüVi10a]

Merkmalsklasse	Beschreibung	Zugehörige Merkmale
Basismerkmale (n_{basis})	Merkmale, die für die Erstellung einer Rohdatengrundstruktur verwendet werden	$n_{basis} = \{ n_1, n_2, n_4, n_5, n_{28}, n_{70}, n_{71}, n_{72}, n_{73} \}$
Erweiterte Basismerkmale (n_{ext})	Unabhängige Merkmale, die in eine Rohdatengrundstruktur eingebettet werden	$n_{ext} = \{ n_6, n_{14}, n_{19}, n_{20}, n_{21}, n_{22}, n_{23}, n_{24}, n_{25} \}$
Übrige Merkmale (n_{rest})	Merkmale, die weder *Basis*- noch *erweiterte Basismerkmale* sind	$n_{rest} = n_{all} \setminus n_{basis} \setminus n_{ext}$
Alle Merkmale (n_{all})	Alle Merkmale, die während der BioHash-Generierung verwendet werden	$n_{all} = \{ n_1, ..., n_i \}$

In Tabelle 8 sind alle Informationen (Merkmale) des Merkmalsvektors aufgelistet, die für die Konstruktion eines Handschriftensignals verwendet werden. Dabei sind die Merkmale in zwei unterschiedliche Merkmalsklassen eingeteilt. Die Basismerkmale n_{basis} werden für die Konstruktion einer Rohdatengrundstruktur verwendet und die erweiterten Basismerkmale n_{ext} in einem darauffolgenden Schritt in die Rohdatengrundstruktur eingebettet. Tabelle 9 listet diese Merkmale zusammen mit einer kurzen Beschreibung auf.

Tabelle 9 Basis- und erweiterte Basismerkmale [KüVi10a]

Merkmalsklasse	**Merkmalsnummer n_i**	**Merkmalsbeschreibung**
Basismerkmale (n_{basis})	n_1	Dauer der Aufzeichnung
	n_2	Anzahl an Samplepunkten
	n_4	Durchschnittsgeschwindigkeit in X * 1000 Pixel / ms
	n_5	Durchschnittsgeschwindigkeit in Y * 1000 Pixel / ms
	n_{28}	Dauer der gesamten Absetzzeit des Stiftes
	n_{70}	Anzahl an Maxima in X Richtung
	n_{71}	Anzahl an Minima in X Richtung
	n_{72}	Anzahl an Maxima in Y Richtung
	n_{73}	Anzahl an Minima in Y Richtung
Erweiterte Basismerkmale (n_{ext})	n_6	Anzahl an Zeiträumen, in denen der Stift aufgesetzt ist
	n_{14}	Maximal aufgezeichneter Druck
	n_{19}	Maximaler Wert für den Höhenwinkel des Stiftes
	n_{20}	Minimaler Wert für den Höhenwinkel des Stiftes
	n_{21}	Maximaler Wert für den Richtungswinkel des Stiftes
	n_{22}	Minimaler Wert für den Richtungswinkel des Stiftes
	n_{23}	Durchschnittsdruck relativ zum Maximaldruck * 1000
	n_{24}	Durchschnittlicher Richtungswinkel des Stiftes
	n_{25}	Durchschnittlicher Höhenwinkel des Stiftes

Nachdem im ersten Schritt der Merkmalsvektor berechnet wurde, wird im nachfolgenden zweiten Schritt die Rohdatengrundstruktur erzeugt. Hierfür werden jedoch noch weitere Parameter benötigt, die aus Merkmalen des Merkmalsvektors berechnet werden. Beispielsweise können mit Hilfe von Merkmal n_4 (Durchschnittsgeschwindigkeit in X-Richtung), n_5 (Durchschnittsgeschwindigkeit in Y-Richtung) und n_1 (gesamte Schreibdauer) die maximalen Werte von *x(t)*: x_{max} und y(t): y_{max} bestimmt werden. Diese sind im weiteren Verlauf der Rohdatengenerierung von großer Wichtigkeit, wie im folgenden Schritt ersichtlich wird. Außerdem kann die Steigung des *X*-Signals (Linearanteil) x_{slope} mit Hilfe von x_{max} und n_1 (gesamte Schreibdauer) ermittelt werden. Eine komplette Liste der verwendeten Parameter und deren Berechnung ist in Tabelle 10 angegeben.
Mit Hilfe der Basismerkmale n_{basis} und den berechneten Parametern kann eine Rohdatengrundstruktur mit Hilfe einer Spline-Interpolation für das horizontale Schreibsignal x(t) und das vertikale Schreibsignal y(t) erzeugt werden. Hierfür werden Position und Wert (Höhe) aller Stützpunkte (Splines) benötigt. Mit Hilfe des Merkmalsvektors kann jedoch lediglich die Anzahl der Stützpunkte für *x(t)* und *y(t)* in Form von Minima und Maxima ermittelt werden. Die *Basismerkmale* n_{70}, n_{71}, n_{72} und n_{73} enthalten diese Informationen

(Anzahl der Minima und Maxima von x(t) bzw. y(t)). Außerdem kann neben der Anzahl der Stützpunkte die ermittelten Parameter x_{max} und y_{max} für die Spline-Interpolation verwendet werden. Fehlende Koordinaten für die Platzierung der Interpolationsstützpunkte werden mit einem Pseudozufallszahlengenerator (Pseudo Random Number Generator - PRNG) generiert. So können verschiedene Varianten von Rohdaten erzeugt werden, die nicht nur die jeweiligen Rekonstruktionsmerkmale, sondern potentiell auch die aller übrigen Merkmale beinhalten.

Tabelle 10 Berechnete Parameter auf Basis verschiedener Merkmale [KüVi10a]

Parameter	Beschreibung	Benötigte Merkmale	Berechnung
x_{max}	Maximaler Wert des X-Signals	n_1 und n_4	$x_{\max} = \frac{n_1 * n_4}{1000}$ **Formel 19**
y_{max}	Maximaler Wert des Y-Signals	n_1 und n_5	$y_{\max} = \frac{n_1 * n_5}{1000}$ **Formel 20**
x_{slope}	Steigung des X-Signals (Linearanteil)	n_1	$x_{slope} = \frac{x_{\max}}{n_1}$ **Formel 21**
$t_{s,calc}$	Berechnete Abtastrate (Samplerate) des Signaturtabletts	n_1, n_{28} und n_2	$t_{s,calc} = \frac{n_1 - n_{23}}{n_2}$ **Formel 22**
p_{avr}	Durchschnittlicher Druck während der gesamten Schreibdauer	n_{14} und n_{23}	$p_{avr} = \frac{n_{14} * n_{23}}{1000}$ **Formel 23**
tn_penup	Durchschnittliche Dauer, in der der Stift abgesetzt ist	n_6 und n_{28}	$tn\_penup = \frac{n_{28}}{n_6}$ **Formel 24**

Im dritten und letzten Schritt werden die erweiterten Basismerkmale n_{ext} in die Rohdatengrundstruktur eingebettet. Dies sind winkelbasierte und druckbasierte Merkmale, welche unabhängig vom horizontalen und vertikalen Schreibsignal eingebettet werden können. Hier ein Beispiel für die Einbettung der Merkmale n_{19} (Maximaler Höhenwinkel), n_{20} (Minimaler Höhenwinkel) und n_{25} (Durchschnittlicher Höhenwinkel) in das zu generierende Signal: Zunächst werden alle Höhenwinkelwerte in der gesamten Rohdatengrundstruktur auf den Wert von n_{25} gesetzt. Anschließend wird der maximale Höhenwinkelwert (n_{19}) einem zufällig gewählten Samplepunkt zugeordnet und umliegende Höhenwinkelwerte so geändert (verringert), dass der durchschnittliche Höhenwinkelwert nicht beeinflusst wird. Ähnliches wird bei der Umsetzung des minimalen Höhenwinkelwertes durchgeführt, es dürfen jedoch hier nur Höhenwinkelwerte geändert werden, die den durchschnittlichen Winkelwert besitzen. So wird verhindert, dass der bereits integrierte maximale Wert nicht verändert wird. Diese Prozedur wird analog für Seitenwinkel und Druckwerte durchgeführt. Im Anhang Anlage 5 ist beispielhaft die Umsetzung des maximalen Seitenwinkels in einer Matlab/Octave Funktion (Quellcode) dargestellt und erläutert. In Abbildung 28 werden die jeweiligen Schritte der Rohdatengenerierung dargestellt.

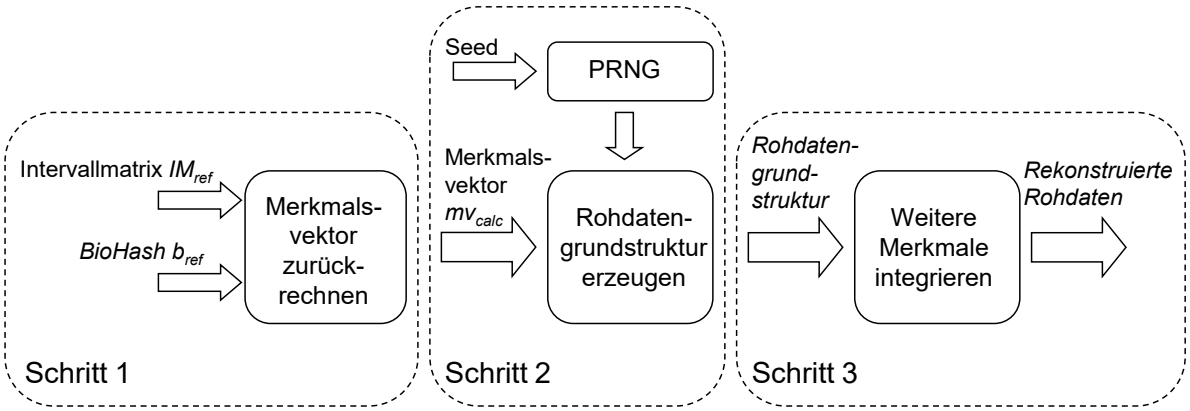


Abbildung 28 Blockdiagramm der Arbeitsschritte zur Erzeugung von Rohdaten, [KüVi10a]

Erste Evaluationsergebnisse in [KüVi10a] haben bereits gezeigt, dass diese Methode geeignet ist Handschriftensignale zu erzeugen, die den angegriffenen Referenz-BioHash bis zu knapp 60% ähneln. Die Ergebnisse basieren auf 103 verwendeten statistischen Merkmalen welche vom Schreibsignal erhoben wurden. In der Arbeit von Scheidat [Sche15] wurden u.a. weitere neue 28 statistische Merkmale eingeführt und getestet. Aus diesem Grund soll dieses Verfahren auf den aktuell verwendeten 131 Merkmalen des Algorithmus evaluiert werden. Zusätzlich sollen verschiedene Betriebsmodi des Verifikationssystems gewählt und deren Auswirkung auf die Erfolgschance des Angriffs evaluiert werden.
Ein Teil der Forschungsaufgabe FA1 besteht darin, Designvorschläge für den BioHash-Algorithmus zu unterbreiten, welche die Möglichkeit eines solchen Angriffs auf Erfolg minimiert. Diesbezüglich scheint ein offensichtlicher Änderungsvorschlag am Verifikationsalgorithmus, die Nutzung der Basismerkmale zu unterlassen. So kann unterbunden werden, dass eine Rohdatengrundstruktur auf Basis dieser Merkmale erzeugt werden kann. Deshalb soll evaluiert werden, inwieweit sich die Verifikationsperformanz des Algorithmus auf das Weglassen der Basismerkmale auswirkt.

5.3 Experimentelle Tests

Nachdem die Angriffstechnik in Abschnitt 5.2.3 vorgestellt wurde, werden in den folgenden Abschnitten die Messmethodik und der Evaluationsaufbau der Experimente (Abschnitt 5.3.1) sowie die erzielten experimentellen Ergebnisse vorgestellt und bewertet (Abschnitt 5.3.2).

5.3.1 Messmethodik und Evaluationsaufbau

Für die experimentellen Tests werden verschiedene Schreibinhalte, sogenannte Schreibsemantiken, verwendet. Vielhauer hat beispielsweise in [Viel06] bereits verschiedene Schreibinhalte für die Evaluation verwendet. In einigen Vorarbeiten des Autors mit weiteren Ko-Autoren (z.B. [KSVD12], [KSVD12a], [KSAV11] oder [KüVi11a]) wurden ebenfalls unterschiedliche Schreibinhalte eingesetzt. Die Verwendung verschiedener Schreibsemantiken ermöglicht es, u.a. einen Eindruck über die Verifikationsperformanz unterschiedlicher Schreibinhalte zu bekommen. So können beispielsweise potentielle Unterschiede zwischen PIN-basierten und Symbol-basierten Schreibinhalten innerhalb eines Verifikationsverfahrens ermittelt werden. Nachfolgend werden, die in dieser Forschungsaufgabe verwendeten Schreibinhalte, kurz beschrieben und warum sie eingesetzt werden.

gegebene PIN
Allen Testpersonen ist eine vorgegebene PIN als Schreibinhalt vorgegeben. Somit wird sichergestellt, dass sich der Unterschied zwischen den einzelnen Schriftproben nicht durch den Schreibinhalt, sondern durch die individuelle Art zu schreiben bestimmt.

geheime PIN (PIN)
Bei dieser Semantik kann eine Person eine fünfstellige Ziffernkombination aus einem Grundzeichensatz (Ziffern von 0-9) frei wählen. Theoretisch ergeben sich dadurch 100.000 (10^5) unterschiedliche Kombinationen, wodurch sich die Wahrscheinlichkeit gleicher Kombination unterschiedlicher Testpersonen (auch potentielle Angreifer) minimiert.

Pseudonym
Der Schreiber kann sich einen Namen frei wählen und diesen üben, erst dann wird anschließend mit der Datenaufzeichnung begonnen. Das Pseudonym wird hier u.a. gewählt, weil Testpersonen Bedenken geäußert haben ihre "richtige" Signatur zu verwenden.

Symbol
Für diese Semantik wird der Schreiber gebeten, ein Symbol oder eine einfache Zeichnung als Schriftprobe zu wählen. Mit der freien Wahl des Symbols wird nicht nur die Vertraulichkeit des Schreibinhalts zu einem gewissen Maße berücksichtigt, sondern auch die individuelle Schreibweise, also mit welchem Segment des Symbols begonnen wird (Reihenfolge).

Woher
Der Inhalt der Semantikklasse Woher ist die Antwort auf die Frage "Woher kommen Sie?". Der Schreiber verfügt über gewisse Einschränkungen aber auch Freiheiten bei der Wahl der Antwort. Beobachtungen bei der Aufzeichnung haben u.a. gezeigt, dass die Probanden oft mit dem Wohnort oder -land beziehungsweise ihrem Geburtsort oder -land antworteten. Einige Schreiber haben auch den Ort genannt, an dem sie sich vor der Datenaufzeichnung aufgehalten haben.

Ferner sollen für diesen Test die Toleranzfaktoren bestimmt werden. Wie im Abschnitt 4.2.2 bereits beschrieben, kann mittels Toleranzfaktor *tf* die Intervallbreite innerhalb des Verifikationsalgorithmus für jedes Merkmal global beeinflusst werden. Somit lässt sich die Verifikationsleistung des Handschriftenalgorithmus für einen bestimmten Datensatz optimieren, siehe z.B. auch die Arbeit von Scheidat in [Sche15].
Für den jeweiligen Datensatz (Handschriftendaten einer Semantikklasse) werden entsprechend die Equal Error Rate (EER), Reproduktionsrate (RR), Kollisionsrate (CR) und die Kollisionsreproduktionsrate (CRR) bestimmt. Der Toleranzfaktor wird, beginnend bei null, in Schritten der Größe 0,25 jeweils erhöht, bis er den Wert zehn erreicht (siehe Abbildung 29). Anschließend können die Toleranzfaktoren für den jeweils gewünschten Arbeitspunkt (geringste EER, niedrigste CRR etc.) ausgewählt werden, siehe dazu [Sche15]. Im abgebildeten Beispiel (Abbildung 29) kann entsprechend der Toleranzfaktor mit dem Wert 1,75 für die geringste EER abgelesen werden.

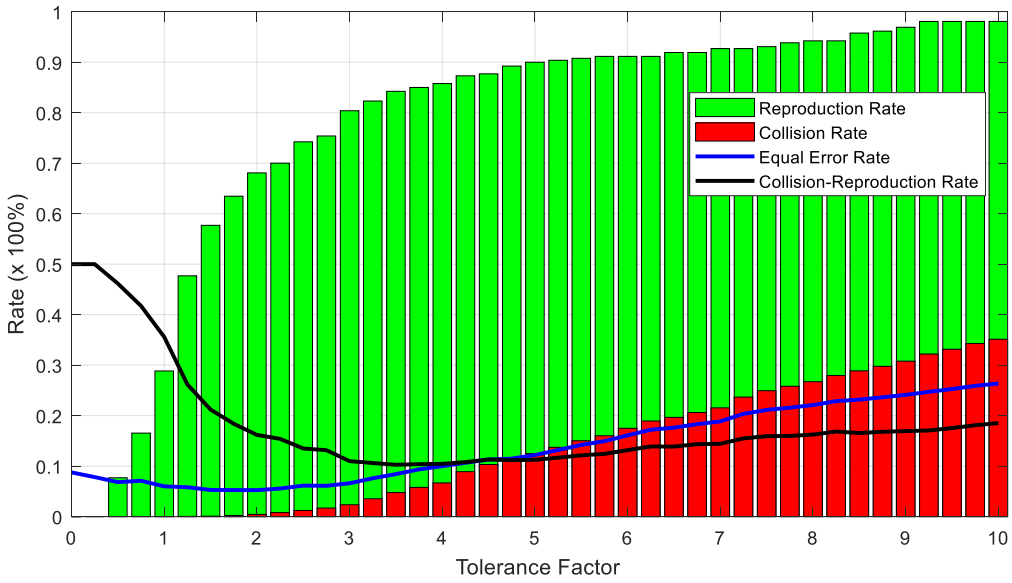


Abbildung 29 Darstellung der Verifikationsperformanz zur Bestimmung der Toleranzfaktoren(Beispielhaft)

Folgendes Angriffsszenario wird für den experimentellen Test angenommen. Ein Angreifer möchte sich unberechtigten Zugriff zu einem geschützten System verschaffen. Dieses System wird durch ein handschriftenbasiertes Verifikationssystem geschützt. Der von Vielhauer in [Viel06] vorgestellte Verifikationsalgorithmus kommt hierbei als Verifikationssystem zum Einsatz. Der Angreifer hat die Möglichkeiten, die Referenzdaten (BioHash und zugehörige Intervallmatrix) einer oder mehrerer Personen aus der Datenbank zu extrahieren. Des Weiteren ist es dem Angreifer möglich Handschriftensignale am Sensor vorbei in das System einzuspielen (Angriffspunkt AP 2). Der Angreifer berechnet auf Basis der extrahierten Referenzdaten einen Merkmalsvektor (siehe Formel 18) und erzeugt mit Hilfe des Merkmalsvektors und der in Abschnitt 5.2.3 beschrieben Methode künstliche Handschriftensignale. Ziel des Angreifers ist es, Handschriftensignale zu generieren, welche Merkmalsvektoren erzeugen, die dem des zurückgerechneten Merkmalsvektors ähneln und entsprechend zu BioHash-Werten führt, die sich vom Referenz-BioHash kaum unterscheiden.
In diesen experimentellen Tests soll u.a. bestimmt werden, wie die Erfolgsaussichten des Angreifers unter bestimmten Arbeitsmodi des Verifikationsalgorithmus sind. Die Arbeitsmodi wurden bereits vom Autor in Zusammenarbeit mit Ko-Autoren u.a. in [KSVA11] und [KSVD12a] bzw. von Makrushin et al. in [MaSV11] erfolgreich angewendet. Folgende Arbeitsmodi wurden für diese Evaluation gewählt:

EER Arbeitsmodus
Das Verifikationssystem wurde mittels Toleranzfaktor so optimiert, dass für die hinterlegten Testdaten die geringste Equal Error Rate (EER) in den jeweiligen Semantikklassen erzielt wird. Mit diesem Arbeitsmodus soll ein System nachgestellt werden, welches einen Kompromiss zwischen Komfort und Sicherheit aus Sicht des Benutzers darstellt.

CRR Arbeitsmodus
Die Kollisions-Reproduktions-Rate beschreibt einen Arbeitsmodus, in dem eine hohe Reproduktionsrate bei geringer Kollisionsrate mittels optimierten Toleranzfaktors erzielt wird. Der Arbeitsmodus soll ein System simulieren, welches Wert auf eine hohe

Reproduktionsrate legt, beispielsweise als Grundlage zur Generierung von kryptografischen Hashwerten.

Nicht optimiertes System (noS)
Dieser Arbeitsmodus beschreibt ein nicht optimiertes System. Der Toleranzfaktor wurde für diesen Arbeitsmodus auf eins (keine Optimierung) gesetzt.

Die Toleranzfaktoren werden für die jeweiligen Arbeitsmodi EER (geringste EER) und CRR (geringste CRR), wie in Abbildung 29 dargestellt, für alle Semantiken bestimmt.
Für die Evaluation wird eine Datenbank verwendet, welche die Handschriftendaten von 34 Personen in fünf verschiedenen Semantikklassen beinhaltet. Pro Person und Semantik wurden zehn Schreibsignale aufgezeichnet, somit ergibt sich eine Gesamtzahl von 1700 Handschriftendaten. In folgenden fünf Semantiken liegen die Handschriftensignale in der Testdatenbank vor:

- gegebene PIN (77993)
- geheime PIN (PIN)
- Pseudonym
- Symbol
- Woher

Für die Berechnung der Referenzdaten werden die ersten vier Handschriftensignale einer Person in der entsprechenden Semantikklasse verwendet. Das fünfte, sechste und siebte Handschriftensignal wird für die Parameterbestimmung (Toleranzfaktor) verwendet und die restlichen drei Handschriftensignale für die eigentliche Verifikation.

Wie im Abschnitt 5.2.3 beschrieben, soll ein Verfahren zur Erzeugung künstlicher Handschriftensignale evaluiert werden, welches auf einer Spline-Interpolation basiert. Dazu werden zu jeder Person und entsprechender Semantik 100 künstliche Handschriftensignale (aber durch Zufallskomponenten doch unterschiedlich) mittels dieser Methode generiert. Insgesamt werden so 17.000 künstliche Handschriftensignale erzeugt.
Nachdem die künstlichen Handschriftensignale erzeugt wurden, werden die Verifikationstests mit diesen Daten durchgeführt. Hierfür werden die entsprechenden Falschakzeptanzraten (FAR) für die jeweiligen Arbeitsmodi und Semantikklassen berechnet. Die Ergebnisse können dann anschließend miteinander verglichen werden. Je nachdem, wie einige Messwerte ausfallen, können Aussagen über den Erfolg des Angriffs innerhalb eines bestimmten Arbeitsmodus getroffen werden. So sollen insbesondere die EER, die CRR und die Reproduktionsrate der originalen mit den der künstlich erzeugten Handschriftensignale verglichen werden.
Weiterhin soll die Verifikationsperformanz des Systems unter Ausschluss der Basismerkmale ermittelt werden. Diese Basismerkmale sind für die Erstellung der künstlichen Handschriftensignale, so wie im Abschnitt 5.2.3 vorgestellt, notwendig. Mit diesem Test soll untersucht werden, inwieweit die Basismerkmale Einfluss auf die Verifikationsperformanz des Systems haben. Hierfür werden die Basismerkmale bei der Berechnung der Fehlerraten nicht berücksichtigt. Anschließend sollen die EER, CRR, CR, und RR beider Tests (mit und ohne Basismerkmale) verglichen werden. Sind die Auswirkungen auf die

Verifikationsperformanz vernachlässigbar, wird empfohlen, diese Merkmale nicht zu verwenden (Designvorschlag), um einen solchen Angriff zu unterbinden.
In Tabelle 11 werden die wesentlichen Schritte der Evaluation zusammenfassend dargestellt. Hier werden die Anzahl der verwendeten Merkmale (131 oder 122) als auch die Arbeitsmodi (EER, CRR, noS) angegeben. Der Arbeitsmodus noS (nicht optimiertes System) verwendet einen Toleranzfaktor von tf=1.

Tabelle 11 Auflistung der wesentlichen Evaluationsschritte

Schritt	Beschreibung	Arbeits-modi	Anz. verwendeter Merkmale
1	Berechnung der Optimierungsparameter (Toleranzfaktoren) für alle Semantikklassen und Arbeitsmodi unter Verwendung aller 131 Merkmale	EER, CRR	131
2	Bestimmen der Fehlerraten (EER, RR, CRR) für die authentischen und künstlich erzeugten Handschriftendaten aller Semantikklassen	ERR, CRR, noS	131
3	Vergleichen der Fehlerraten aus Schritt 2, um den Erfolg des Angriffes zu bestimmen.	EER, CRR, noS	131
4	Berechnung der Optimierungsparameter (Toleranzfaktoren) für alle Semantikklassen und Arbeitsmodi unter Verwendung von 122 Merkmalen (ohne sensible Basismerkmale)	ERR, CRR	122
5	Bestimmen der Fehlerraten (EER, RR, CRR) für die authentischen Handschriftendaten, hier keine künstlich erzeugten Handschriften da Angriff ohne Basismerkmale nicht durchführbar ist	EER, CRR, noS	122
6	Vergleichen der Fehlerraten aus Schritt 2 mit denen aus Schritt 5 (nur die Fehlerraten der authentischen Handschriftendaten vergleichen), um Auswirkung der 9 weggelassenen Basismerkmale zu bestimmen.	EER, CRR, noS	122, 131

Für die Erfassung der Handschriftendaten wurde das naturaSign Pad Classic der Firma StepOver GmbH verwendet. In Tabelle 12 sind einige ausgewählte technische Eigenschaften des Gerätes aufgelistet.

Tabelle 12 Technische Eigenschaften des verwendeten Aufzeichnungsgerätes "naturaSign Pad Classic"

Komponente	Eigenschaft
Display	320 x 85 Pixel
Sensor	1000 DPI (nicht interpoliert)
Abtastrate	500 Samples pro Sekunde
Sample	x und y Koordinate, Druck und Zeitangabe
Druckstufen	512

5.3.2 Präsentation und Bewertung der Ergebnisse

Nachfolgend werden die Ergebnisse der Toleranzfaktorermittlung für die jeweiligen Arbeitsmodi und entsprechenden Semantiken präsentiert. Die Abbildung 30 zeigt exemplarisch die Verifikationsperformanz für die Semantik Symbol. Für alle weiteren Semantiken sind die Abbildungen der Verifikationsperformanz im Anhang Anlage 3 zu finden. Entsprechend des Arbeitsmodus werden die Toleranzfaktoren abgelesen, niedrigste EER bzw. geringste CRR für EER Modus respektive CRR Modus.

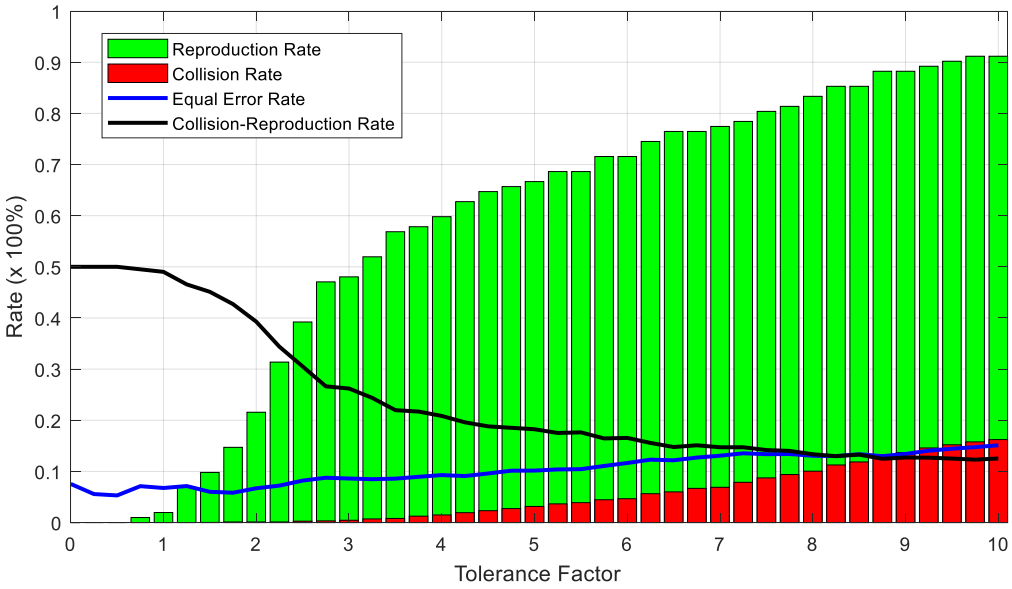


Abbildung 30 Exemplarische Darstellung der Verifikationsperformanz der Semantikklasse Symbol

In Tabelle 13 sind alle ermittelten Toleranzfaktoren entsprechenden aufgelistet. Der Toleranzfaktor für das nicht optimierte System beträgt für jede Semantikklasse jeweils *tf=1*.

Tabelle 13 Ermittelte Toleranzfaktoren der jeweiligen Arbeitsmodi und Semantikklasse

Arbeitsmodi	Semantikklasse	Toleranzfaktor
EER	77993	0,75
	PIN	1,25
	Pseudonym	1
	Symbol	0,5
	Woher	0,5
CRR	77993	7
	PIN	9
	Pseudonym	6,25
	Symbol	8,75
	Woher	8,75

Im nächsten Schritt wurden pro Person und Semantik 100 künstliche Handschriftensignale erstellt. Mit diesen Handschriftensignalen wurde entsprechend der Person und Semantik ein Verifikationsversuch durchgeführt. Die Ergebnisse werden in Form einer FAR_{re} zusammen mit FRR und FAR (siehe auch Abschnitt 4.1.4 Biometrische Fehlerraten) in Abbildung 31 exemplarisch für die Semantik 77993 im Arbeitsmodus EER gezeigt. Die FAR_{re} beschreibt hier die relative Häufigkeit mit der es künstlich erzeugten Handschriftendaten

gelungen ist, ein Zugang zum System zu erlangen. Die Fehlerratendiagramme aller übrigen Semantiken sind im Anhang Anlage 4 zu finden.

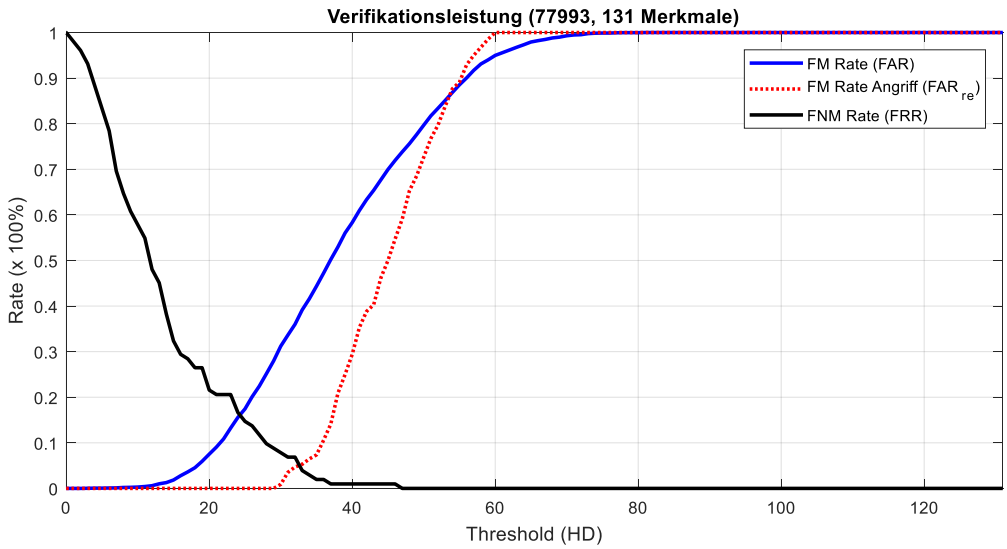


Abbildung 31 Fehlerraten der Semantik 77993 mit FAR_{re} der Angriffsdaten (rot gepunktet)

In der Tabelle 14 und Tabelle 15 sind die Fehlerraten (EER und EER_{re}) und Reproduktionsraten (CR, CR_{re}, RR, CRR und CRR_{re}) für die jeweiligen Arbeitsmodi (EER respektive CRR Modus) zusammengefasst dargestellt. CR_{re} beschreibt die Kollisionsrate der Angriffsdaten gegenüber den entsprechenden Referenzdaten. Stimmen ein BioHash, basierend auf einem künstlichen Handschriftensignal, komplett mit dem jeweiligen Referenz-BioHash überein, so wurde eine Kollision erzielt. Die CRR_{re} beschreibt das Verhältnis von Kollisionsrate CR_{re} gegenüber der Reproduktionsrate RR der "echten" Handschriftendaten. Die Angabe HD_re_best beschreibt die geringste Hamming-Distanz, die innerhalb einer Semantikklasse aufgetreten ist. Sie beschreibt den Abstand zwischen dem Referenz-BioHash und einem BioHash, der auf Basis eines künstlich erzeugten Handschriftensignals generiert wurde. Wie bereits in Abschnitt 4.1.4 erläutert, werden im Arbeitsmodus CRR die Systemparameter so gewählt, dass eine möglichst geringe CRR erzielt wird. Im Arbeitsmodus EER wird hingegen versucht, eine möglichst geringe EER zu erreichen. Die Optimierung in einem Arbeitsmodus kann jedoch dazu führen, dass sich der jeweils andere Wert zum Schlechteren ändert. So ist im CRR Arbeitsmodus die EER höher und im EER Arbeitsmodus die CRR hoch. Das ist u.a. in den nachfolgenden Tabellen (Tabelle 14 und Tabelle 15) für die authentischen als auch für die künstlich erzeugten Angriffsdaten zu erkennen.

Tabelle 14 Erzielte Fehlerraten der Angriffsdaten im Arbeitsmodus EER

Angabe	Semantikklasse				
	77993	PIN	Pseudonym	Symbol	Woher
CRR in %	50,00	49,01	49,50	50,00	50,00
CRR_{re} in %	50,00	49,01	49,50	50,00	50,00
RR in %	0,00	1,96	0,98	0,00	0,00
CR in %	0,00	0,00	0,00	0,00	0,00
CR_{re} in %	0,00	0,00	0,00	0,00	0,00
EER in %	16,10	11,37	9,52	7,45	6,07
EER_{re} in %	5,06	5,88	3,86	6,56	1,60

Angabe	Semantikklasse				
	77993	PIN	Pseudonym	Symbol	Woher
HD_re_mean	37,00	28,20	33,60	31,60	40,50
HD_re_best	30,00	18,00	22,00	22,00	35,00

Die Angabe in HD_re_mean beschreibt den durchschnittlich erzielten Abstand (Hamming-Distanz) der innerhalb einer Semantikklasse aufgetreten ist. Auch hier ist der Abstand zwischen Referenz-BioHash und dem eines BioHash-Wertes, der durch ein künstlich erzeugtes Handschriftensignal generiert wurde, gemeint.

Tabelle 15 Erzielte Fehlerraten der Angriffsdaten im Arbeitsmodus CRR

Angabe	Semantikklasse				
	77993	PIN	Pseudonym	Symbol	Woher
CRR in %	18,22	19,57	20,60	14,05	18,03
CRR_{re} in %	12,30	11,17	19,55	10,78	12,74
RR in %	77,45	79,41	64,70	84,31	74,50
CR in %	13,90	18,56	5,91	12,41	10,57
CR_{re} in %	2,05	1,76	3,82	5,88	0,00
EER in %	18,71	19,70	14,90	14,07	16,33
EER_{re} in %	7,12	8,82	9,44	9,89	2,50
HD_re_mean	2,00	2,00	3,00	1,00	3,00
HD_re_best	0,00	0,00	0,00	0,00	1,00

Die in Tabelle 16 dargestellten Fehlerraten (EER und EER_{re}) und Reproduktionsraten (CR, CR_{re}, RR, CRR und CRR_{re}) sind im nicht optimierten System ermittelt worden.

Tabelle 16 Erzielte Fehlerraten der Angriffsdaten im nicht optimierten System

Angabe	Semantikklasse				
	77993	PIN	Pseudonym	Symbol	Woher
CRR in %	49,50	50,00	49,50	48,52	50,00
CRR_{re} in %	49,50	50,00	49,50	48,52	50,00
RR in %	0,98	0,00	0,98	2,94	0,00
CR in %	0,00	0,00	0,00	0,00	0,00
CR_{re} in %	0,00	0,00	0,00	0,00	0,00
EER in %	17,24	11,91	9,52	8,60	5,94
EER_{re} in %	4,90	5,88	3,86	7,87	0,98
HD_re_mean	31,90	34,10	33,60	23,20	30,90
HD_re_best	26,00	19,00	22,00	14,00	23,00

Im nachfolgenden Teil werden die Ergebnisse der Untersuchung der Verifikationsleistung präsentiert. Innerhalb dieser experimentellen Versuche wurden die Basismerkmale (insgesamt neun Merkmale) vom Verifikationsalgorithmus nicht berücksichtigt.
Zunächst wurden die Toleranzfaktoren aller Semantikklasse für die Arbeitsmodi EER und CRR bestimmt. Die Abbildung 32 zeigt exemplarisch die Verifikationsperformanz für die Semantik Symbol unter Verwendung von 122 Merkmalen. Im Anhang Anlage 3 befinden sich die Diagramme der übrigen Semantikklassen.

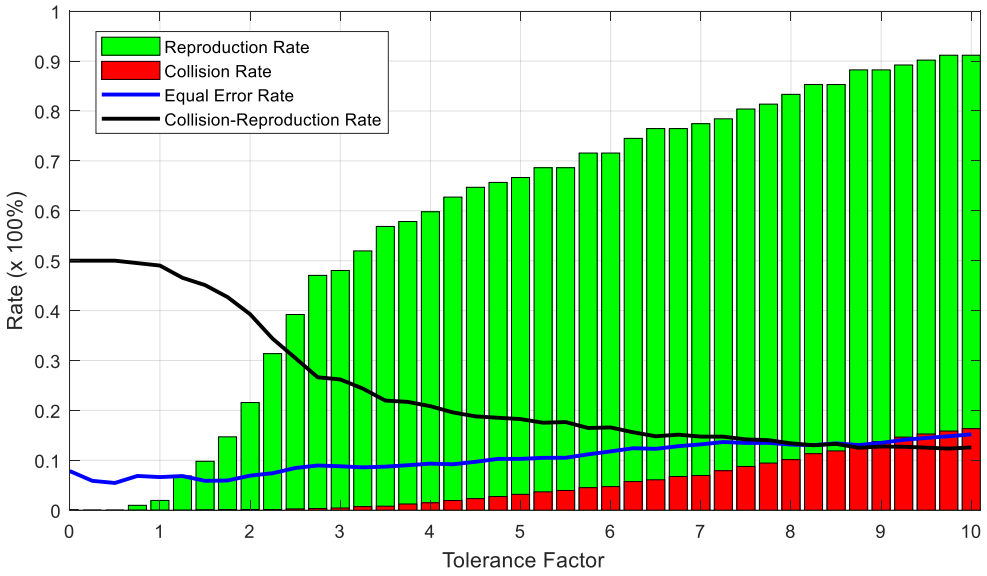


Abbildung 32 Exemplarische Darstellung der Verifikationsperformanz der Semantikklasse Symbol (122 Merkmale)

Anschließend wurden die Fehlerraten des Verifikationssystems unter Verwendung der 122 Merkmale (ohne sensible Basismerkmale) durchgeführt. Die Ergebnisse sind in Tabelle 17 dargestellt und zum besseren Vergleich neben die Fehlerraten, welche unter Verwendung aller 131 Merkmale erzielt wurden, gestellt. Der Arbeitsmodus *tf=1* beschreibt das nicht optimierte Verifikationssystem, bei dem der Toleranzfaktor für alle Semantiken mit dem Wert eins gesetzt ist.

Tabelle 17 Fehlerraten des Systems unter Verwendung von 122 und 131 Merkmalen

Semantik	Rate	Arbeitsmodus (122)			Arbeitsmodus (131)		
		EER	CRR	tf=1	EER	CRR	tf=1
77993	CRR in %	48,52	22,17	49,50	50,00	**18,22**	49,50
	RR in %	2,94	60,78	0,98	0,00	77,45	0,98
	CR in %	0,00	5,13	0,00	0,00	13,90	0,00
	EER in %	**14,94**	16,52	17,83	16,10	18,71	17,24
PIN	CRR in %	50,00	19,75	50,00	49,01	**19,57**	50,00
	RR in %	0,00	66,67	0,00	1,96	79,41	0,00
	CR in %	0,00	6,17	0,00	0,00	18,56	0,00
	EER in %	11,40	17,25	11,35	**11,37**	19,70	11,91
Pseudonym	CRR in %	49,50	22,57	49,50	49,50	**20,60**	49,50
	RR in %	0,98	59,80	0,98	0,98	64,70	0,98
	CR in %	0,00	4,96	0,00	0,00	5,91	0,00
	EER in %	**7,19**	14,44	9,80	9,52	14,90	9,52
Symbol	CRR in %	50,00	17,95	48,52	50	**14,05**	48,52
	RR in %	0,00	68,62	2,94	0,00	84,31	2,94
	CR in %	0,00	4,54	0,00	0,00	12,41	0,00
	EER in %	**7,14**	10,67	7,98	7,45	14,073	8,60
Woher	CRR in %	43,62	19,38	50,00	50	**18,03**	50,00
	RR in %	12,74	64,70	0,00	0,00	74,50	0,00
	CR in %	0,00	3,47	0,00	0,00	10,57	0,00
	EER in %	**5,24**	10,45	6,24	6,07	16,33	5,94

In Tabelle 17 ist zu erkennen, dass sich die EER im ERR-Modus bei 122 verwendeten Merkmalen, mit Ausnahme der Semantik PIN, verbessert hat. Die ermittelte EER der Semantik PIN ist nur geringfügig schlechter (0,03 Prozentpunkte). Die Kollisionsreproduktionsrate fällt bei Verwendung von nur 122 Merkmalen höher und somit schlechter aus. Die Unterschiede belaufen sich bei den Semantiken 77993 und Symbol auf ca. vier Prozentpunkte. Bei der Semantik Woher sind es ca. 1,3 Prozentpunkte und bei der Semantik PIN ist der Unterschied mit 0,18 Prozentpunkten vernachlässigbar gering. Unter Berücksichtigung der evaluierten geringen Unterschiede in der Verifikationsperformanz zwischen 122 und 131 verwendeten Merkmalen empfiehlt der Autor, die neun sensiblen Basismerkmale nicht zu verwenden. Die nur wenig schlechteren Ergebnisse der Kollisionsreproduktionsrate (0,18 - 3,95 Prozentpunkte bzw. 2,83 Prozentpunkte im Durchschnitt) rechtfertigen nicht die Verwendung der Basismerkmale. Mit diesem Designvorschlag für den biometrischen Hashalgorithmus kann somit die Gefahr eines solchen, in dieser Arbeit vorgestellten und evaluierten, Angriffs verhindert werden.

Designvorschläge
Prinzipiell empfiehlt der Autor für alle biometrischen (Handschriften-) Verifikationsverfahren, dass biometrische Merkmale, welche direkt aus den Referenzdaten extrahiert werden können und dazu beitragen eine Basisstruktur für biometrische Angriffsdaten zu schaffen, nicht verwendet werden sollten. Dies gilt auch für Merkmale, die anscheinend durch Schutzmechanismen nicht direkt ermittelt werden können. Schutzmechanismen können unter Umständen ausgehebelt oder umgangen werden. Sollte ein Schutzmechanismus ausgehebelt werden, kann somit entsprechend verhindert werden, dass ein potentieller Angreifer auf einfache Weise Angriffsdaten produzieren kann. Beispiele für solche Basismerkmale sind u.a. globale Merkmale wie die Aufzeichnungsdauer, Abtastrate, Höhen- und Seitenverhältnisse sowie absolute Maximal- bzw. Minimalwerte.

6 Erzeugen künstlicher Handschriften (FA2)

Innerhalb der Forschungsaufgabe 2 (FA2), soll ein neues Verfahren zur Erzeugung künstlicher Handschriftendaten entwickelt und entsprechend evaluiert werden. Hierfür werden im Abschnitt 6.1 aus der Literatur bekannte Verfahren vorgestellt und deren Vor- und Nachteile diskutiert. Anschließend wird in Abschnitt 6.2 ein neues Verfahren präsentiert, welches im letzten Abschnitt des Kapitels mittels experimenteller Tests evaluiert werden soll.

6.1 Vorgehensweise und Methodik

Wie im Abschnitt 2.3.4 bereits kurz erläutert, existieren unterschiedliche Verfahren zur Erzeugung künstlicher Handschriftendaten. Diese unterscheiden sich u.a. in der Art der verwendeten Eingangs- bzw. Ausgangsdaten, in der Zielsetzung (was soll mit den Handschriftendaten geschehen/erreicht werden) und dem verwendeten Verfahren/Algorithmus zur Generierung der Handschriftendaten.
Prinzipiell können in diesem Zusammenhang die Ein- und Ausgangsdaten in online und offline Daten unterschieden werden. Wobei online Daten dynamische Handschriftendaten sind, welche mit einem speziellen Gerät aufgezeichnet wurden (z.B. Signaturtablett) und in zeitdiskreten Koordinaten mit ggf. entsprechenden Druck- und Winkelwerten versehen sind, siehe Abschnitt 4.2.1. Offline oder statische Handschriftendaten hingegen sind Abbildungen von Handschriftendaten (z.B. digitalisiertes Foto einer Handschrift), sie enthalten keine zeitdiskreten Informationen. Merkmale wie beispielsweise der ausgeübte Druck auf der Stiftspitze während des Schreibens müssen indirekt ermittelt werden z.B. durch Analyse der Linienstärke (Breite) an unterschiedlichen Stellen der Buchstaben und Wörter.
Auch Diaz unterteilt in seiner Arbeit [Diaz16] die verschiedenen Verfahren zur Erzeugung von Handschriften hinsichtlich ihrer Form der Eingangs- und Ausgangsdaten d.h. liegen die Daten in statischer (offline) oder dynamischer (online) Form vor. So existieren Verfahren (z.B. [RaGF08] und [GuHH14]), welche dynamische (online) Handschriftendaten verwenden, um aufbauend auf diesen ein statisches (offline) Handschriftenbild zu erzeugen (Online-zu-Offline bzw. On-2-Off). Des Weiteren existieren Methoden (z.B. [OKBS97] und [FrSV06]), welche auf Basis von offline Handschriftendaten neue statische Handschriftendaten erzeugen (Off-2-Off). Zusätzlich werden On-2-On-Verfahren (z.B. [SoSu14] und [RaGF07]) von Diaz in seiner Arbeit kategorisiert und kurz vorgestellt. Einzig bei der Erzeugung von künstlichen Handschriftendaten im Bereich Off-2-On gibt es noch keine bekannten Verfahren. Dies scheint derzeit ein offenes Forschungsfeld zu sein, soll jedoch in dieser Arbeit nicht behandelt werden. In der Tabelle 18 werden alle von Diaz vorgestellten Verfahren aufgelistet. Allen gemein ist die Notwendigkeit, reale Handschriftendaten (Muster) für die Generierung künstlicher Handschriftendaten zu verwenden.
Im folgenden Abschnitt 6.1.1 sollen einige Verfahren und deren Vor- bzw. Nachteile kurz vorgestellt werden, um anschließend die Gründe für die Einführung eines neuen Verfahrens zu diskutieren. Dieses neue Verfahren wird im Abschnitt 6.1.2 ausgiebiger erläutert und soll im Rahmen dieser Arbeit entsprechend evaluiert werden.

Tabelle 18 Arbeiten mit Bezug zur Generierung von Handschriftenduplikaten [Diaz16]

Konvertierung	Autoren	Methode/Verfahren	Muster[3]	Zielsetzung
On-2-On	Munich et al. 2003 in [MuPe03]	Affine Scale / geometrische Transformation	> 1	Statistisch aussagekräftige Beurteilung/ Bewertung
On-2-On	Rabasse et al. in [RaGF07]	Affine Scale / geometrische Transformation	2	Gleiche Verifikationsperformanz wie Ausgangsdaten
On-2-On	Galbally et al. in [GFM+09]	Affine Scale / geometrische Transformation	1	Steigerung der Verifikationsperformanz
On-2-On	Song et al. in [SoSu14]	Selektieren von Klonen	> 1	Steigerung der Verifikationsperformanz
On-2-Off	Rabasse et al. in [RaGF08]	Affine Scale / geometrische Transformation	2	Gleiche Verifikationsperformanz wie Ausgangsdaten
On-2-Off	Guest et al. in [GuHH14]	Interpolationsmethoden	1	Gleiche Verifikationsperformanz wie Ausgangsdaten
On-2-Off	Galbally et al. in [GDF+15]	Farbablagerung	1	Gleiche Verifikationsperformanz wie Ausgangsdaten
Off-2-Off	Oliveira et al. in [OKBS97]	Faltung auf Basis von Polynomen und Signaldarstellungen	1	Testdatenbank vergrößern
Off-2-Off	Huang et al. in [HuYa97]	Affine Scale / geometrische Transformation	1	Steigerung der Verifikationsperformanz
Off-2-Off	Fang et al. in [FLT+02]	Elastische Anpassung	2	Steigerung der Verifikationsperformanz
Off-2-Off	Frias et al. in [FrSV06]	Affine Scale / geometrische Transformation	1	Testdatenbank vergrößern
Off-2-On	Offener Forschungs-punkt	-	-	-

6.1.1 Potentielle Möglichkeiten zur Erstellung künstl. Handschriftendaten

In der Arbeit von Rabasse et al. [RaGF07] wird ein Verfahren vorgestellt, welches online Handschriftendaten auf Basis von zwei Handschriftensamples einer realen Person erzeugt. Hierfür werden die realen Handschriften eingesetzt, um mittels einer dynamischen Zeitverschiebungsmethode (Dynamic Time Warping – DTW) ein neues Abbild zu formen. Dieses neue Abbild stellt dementsprechend eine Streuung der beiden realen Handschriften dar und dient als Grundlage zur Generierung der künstlichen Handschriftendaten. Rabasse et al. sind mit ihren Verfahren in der Lage, beliebig viele künstliche Handschriftendaten auf Basis von zwei realen Handschriftensamples (seeds) zu erzeugen. Die gleiche Methode haben die Autoren auch eingesetzt, um künstliche Offline Handschriftendaten mittels realen Offline Handschriftendaten zu erzeugen.

Vorteil: Es können viele künstliche Handschriftendaten erzeugt werden, welche sich bzgl. der Erkennungsperformanz untereinander wie reale Handschriftendaten verhalten. Der

[3] Anzahl der Handschriftenmuster die mindestens benötigt werden, um künstliche Handschriftendaten zu erzeugen.

Schreibinhalt ist erkennbar, wenn die zugrundeliegenden realen Handschriftendaten auch „lesbar“ sind.
Nachteil: Für die Generierung werden zwei reale Handschriftensamples einer Person benötig. Künstliche Individuen können mit dieser Methode nicht erzeugt werden, sondern lediglich Klone bzw. Kopien der originalen Schreibsignale.

Song et al. nutzen in [SoSu14] einen Klon-Algorithmus, um künstliche Handschriftendaten zu erzeugen. Dabei fokussieren sie sich nicht auf die Qualität eines einzelnen künstlich erzeugten Handschriftensignals im Vergleich zum originalen Muster, sondern auf die Erkennungsperformanz des gesamten Systems. Die Erkennungsperformanz aller im System hinterlegten Handschriftensignale soll somit verbessert werden.
Vorteil: Die Erkennungsperformanz einer vorhandenen Handschriftendatenbank kann potentiell verbessert werden. Dies hängt aber u.a. vom verwendeten Verifikationsverfahren ab.
Nachteil: Der Schreibinhalt der erzeugten künstlichen Handschriftensignale ist nicht mehr lesbar. Es können keine künstlichen Individuen erzeugt werden. Es sind mehrere originale Handschriftenmuster für die Generierung erforderlich.

Im Gegensatz zu den zwei zuvor genannten Verfahren verwenden Haines et al. in [HaAB16] offline Handschriftendaten. In ihrer Arbeit beschreiben sie ein Verfahren, historische Handschriften nachzuahmen und neue Inhalte auf Basis dieser historischen Handschriften zu erzeugen. Die Prozedur ist relativ aufwändig und erfordert u.a. ein manuelles Aufbereiten der Eingangsdaten, bevor ein nachfolgender automatischer Prozess die Daten weiterverarbeiten kann.
Vorteil: Die generierten Handschriftendaten sehen sehr realistisch aus. Es können beliebige Schreibinhalte produziert werden.
Nachteil: Für die Generierung sind umfangreiche Handschriftenmuster erforderlich. Die Verarbeitungsschritte sind aufwändig und teils manuell. Des Weiteren sind die produzierten Handschriftendaten nicht für handschriftenbasierte Verifikationssysteme geeignet, welche Online-Handschriftendaten verwenden.

Weiterhin wurden Verfahren in [Grav13] und [CHJO16] vorgestellt, welche Neuronale Netze verwenden, um künstliche Handschriftendaten zu erzeugen. Hier wurden Handschriftendaten realer Personen als Trainingsmenge verwendet und das System kann entsprechende Handschriftensignale dieser Testperson erzeugen.
Vorteil: Die von diesem System generierten Handschriftendaten sehen sehr realistisch aus. Weiterhin können beliebige Schreibinhalte produziert werden (Abhängig von der Eingabe - bzw. Trainingsmenge).
Nachteil: Das System benötigt Handschriftendaten realer Personen. Weiterhin sind die produzierten Handschriftendaten nicht für handschriftenbasierte Verifikationssysteme geeignet, welche Online-Handschriftendaten verwenden.

In den letzten Jahren wurden auch Arbeiten vorgestellt, welche künstliche Handschriftensignale generieren ohne reale Handschriftenmuster direkt zu verwenden. In Tabelle 19 sind diese Arbeiten und deren verwendete Methoden/Techniken dargestellt. Der Vorteil der beiden in Tabelle 19 gelistet Methoden ist u.a., dass keine realen Handschriftendaten für die Generierung direkt nötig sind.

Galbally et al. verwenden in [GPF+12] zwar Handschriftendaten realer Personen, aber nicht zur direkten Generierung künstlicher Handschriften. Sie haben bestimmte Merkmale realer Handschriftendaten verschiedenen Personen in ein Parametermodell übertragen. Dieses Modell verwenden die Autoren, um künstliche Schreibindividuen zu erzeugen. Auf Basis dieser Individuen können sie dann entsprechende künstliche Handschriftensignale erzeugen.

Vorteil: Sie nutzen keine Handschriftendaten direkt zur Erzeugung. Es können künstliche Individuen mit entsprechenden Handschriftensignalen erzeugt werden. Die erzeugten Handschriftendaten wirken größtenteils real.
Nachteil: Die erzeugten Handschriftendaten wirken teils wie echte Unterschriften, lassen jedoch keinen Schreibinhalt erkennen. Sie erinnern an Unterschriften von Personen, bei denen man den Namen nicht entziffern kann (verschnörkelte Unterschrift).

Tabelle 19 Arbeiten mit Bezug zur Generierung von komplett künstlichen Handschriftendaten [Diaz16]

Konvertierung	Autoren	Methode/Verfahren	Schrift	Art der Handschriften
On-line	Popel in [Pope07]	Visuelle Handschriftencharakteristiken aus dem Zeitbereich extrahiert	westliche verschnörkelte Handschrift	Unterschrift
On-line	Galbally et al. in [GPF+12] und [GPF+12a]	Spektralanalyse von Handschriftendaten realer Personen in Kombination mit der kinematischen Theorie schneller Handbewegungen	westliche verschnörkelte Handschrift	Unterschrift

Das von Diaz in [Diaz16] vorgestellte Verfahren wird verwendet, um eine komplette Datenbank mit synthetischen Handschriftendaten zu erzeugen und künstliche Handschriftenduplikate realer Person (Handschrift) nachzubilden. Um diese Ziele zu erreichen, setzt Diaz auf Verfahren, welche menschliche Bewegungsabläufe von der kognitiven Verarbeitung bis hin zur neuromotorischen Kontrolle (das Schreiben) mathematisch formulieren/beschreiben. So werden im ersten Schritt bestehende Theorien verwendet, welche versuchen zu beschreiben, wie ein Handschriftenabbild in einem menschlichen Gehirn gespeichert wird. Im zweiten Schritt werden Verfahren verwendet, welche das Zusammenspiel der Muskelbewegungen beschreiben, die beim Schreiben verwendet werden. Diese beiden Schritte werden kombiniert, um den komplexen Prozess des Schreibens nachzubilden. Um die Intraklassen-Variabilität zu simulieren werden Filter angewendet, welche das Handschriftenabbild verzerren.
Vorteil: Die erzeugten künstlichen Handschriftendaten wirken sehr real und erhalten Schreibinhalt, solange das Original Schreibinhalt enthält bzw. erkennen lässt. Das vorgestellte Verfahren kann für Offline- als auch Online-Handschriftendaten angewendet werden.
Nachteil: Es werden keine künstlichen Individuen erzeugt, lediglich auf Basis eines Handschriftensamples werden Duplikate generiert.

Die in diesem Abschnitt vorgestellten Verfahren zur Generierung von Handschriftendaten besitzen unterschiedliche Herangehensweisen, um die Ziele (Handschriftengenerierung)

zu erreichen. Bis auf das Verfahren von Haines et al. in [HaAB16] berücksichtigen die Verfahren den Schreibinhalt nicht bzw. nur passiv. Ist in den Ursprungsdaten Schreibinhalt vorhanden, werden diese teils von den Verfahren beim "Klonen" übernommen. Das Verfahren von Haines et al., welches explizit den Schreibinhalt darstellen möchte, ist jedoch nicht für biometrische Verifikationsalgorithmen ausgelegt. Hier wird insbesondere auf das äußere Aussehen der Handschrift geachtet. Ziel ist es, Texte von beispielsweise verstorbenen historischen Persönlichkeiten nachzustellen. Die zeitdiskreten Informationen zur Verarbeitung in Online-Verifikationssystemen sind hierbei nicht berücksichtigt worden. Sie könnten jedoch bei Offline-Verfahren eingesetzt werden. In der Arbeit von Guyon [Guyo96] werden ähnliche Ziele verfolgt, jedoch liegt der Fokus nicht darin, diese Handschriftendaten einen biometrischen Verifikationssystem zuzuführen. Hier sollen lediglich handschriftlich Kurznotizen nachgebildet werden.

Aus diesem Grund soll in dieser Arbeit ein Verfahren vorgestellt werden, welches künstliche Handschriftendaten inklusive Schreibinhalt erzeugt, die von einem Online-Verifikationsverfahren verarbeitet werden können. Warum ist der Schreibinhalt für die Verifikation so wichtig?

Die Sicherheit aber auch Akzeptanz von handschriftenbasierten Verifikationssystemen kann unter der Verwendung von Schreibsemantiken (Schreibinhalte) erhöht werden. Zum einen sind Menschen teils misstrauisch, wenn sie an einem Verifikationssystem ihre Unterschrift preisgeben sollen, welche sie in der Regel zur aktiven Willensbekundung bei Urkunden, Dokumenten und Zahlungsverkehr leisten. Der Gedanke, die in digitaler Form geleistete Unterschrift könnte für potentielle kriminelle Zwecke verwenden werden, lässt die Akzeptanz eines solchen Systems sinken. Zum anderen sind zusätzliche, z.B. geheime, Schreibinhalte ein zusätzlicher Schutz vor beispielsweise Nachahmungsattacken. Weiterhin kann die Verifikationsperformanz eines handschriftenbasierten Verifikationssystems gesteigert werden, wenn die registrierten Personen unterschiedlichen Schreibinhalt leisten. So unterscheiden sie sich nicht nur in der Art und Weise wie sie schreiben, sondern zusätzlich durch den Inhalt.

In dieser Arbeit sollen aus diesem Grund künstliche Handschriftendaten erzeugt werden, (1) deren Schreibinhalt potentiell lesbar ist, (2) deren Schreibinhalt aus Buchstaben und Ziffern gewählt werden kann, (3) die diese Handschriftendaten realistisch anmuten bzw. nicht künstlich wirken lassen und (4) mit den künstliche Individuen erstellt werden können, welche sich in Ihrer Handschrift voneinander unterscheiden.

Um die oben genannten Punkte zu erfüllen, werden u.a. Zeichensätze benötigt. Diese Zeichensätze sind die Basis für das Erstellen der Schreibinhalte und der eigentlichen Handschriftendaten der künstlichen Individuen. Wie in Abschnitt 2.3.4 bereits beschrieben, können Schreibinhalte, also Wörter oder Sätze, mittels verschiedener Techniken erstellt werden. Beispielsweise können Glyphen zu Wörtern und Sätzen aneinandergereiht werden, dies hat u.a. den Vorteil, dass die Buchstabenverbindungen realistischer aussehen, da sie zum Teil von den Glyphen gebildet werden. Nachteil dieser Methode ist jedoch u.a., dass diese Glyphen aufgezeichnet (aufgenommen) werden müssen. In [Guyo96] beispielsweise werden 970 Glyphen verwendet die als Wörterbuch zum Erstellen von Handschriftennotizen dienen. Hierfür müssen alle Glyphen handschriftlich erfasst werden, was einen enormen Zeitaufwand bedeutet, würde man dies für eine Vielzahl von Personen tun. In Abbildung 33 sind alle in [Guyo96] verwendeten Glyphen (u.a. Buchstaben, Zahlen, Sonderzeichen und Unterschrift) abgebildet.

Aus diesem Grund sollen nur Klein- und Großbuchstaben sowie Ziffern von Null (0) bis Neun (9) aufgezeichnet werden. Die Anzahl der aufzuzeichnenden Handschriftensignale ist somit viel geringer als beispielsweise die von Guyon in [Guyo96]. Jedoch sinkt im Gegensatz dazu das natürlich anmutende Aussehen der Handschrift (Wörter) die erzeugt werden. Die Buchstabenübergänge sind somit nicht mehr ein Teil der Glyphen oder komplett hinterlegten Wortmuster, sondern müssen künstlich erstellt bzw. weggelassen werden.

Zusammenfassung
Die meisten in der Literatur bekannten Verfahren zur Erzeugung künstlicher Handschriftendaten berücksichtigen den Schreibinhalt nicht. Die wenigen Verfahren, die den Schreibinhalt berücksichtigen, sind nicht für die Verwendung von biometrischen Erkennungsmethoden optimiert/ausgelegt. Dies stellt ein offenes Forschungsfeld dar, welches mit der Vorstellung eines neuen Verfahrens in dieser Arbeit adressiert werden soll. Dabei sollen folgende vier Punkte vom neuen Verfahren umgesetzt werden. (1) Der Schreibinhalt der künstlichen Handschriftendaten ist potentiell lesbar, (2) der Schreibinhalt kann aus Buchstaben und Ziffern gewählt werden, (3) die Handschriftendaten sollen realistisch anmuten bzw. nicht künstlich wirken und (4) künstliche Schreibindividuen sollen erstellt werden können, die sich in Ihrer Handschrift voneinander unterscheiden.

6.1.2 Verwendetes Verfahren zu Erzeugung künstlicher Handschriftendaten

Die u.a. in Abschnitt 6.1.1 dargestellten Verfahren [GPF+12], [Diaz16], [SoSu14] und [RaGF07] zur Erzeugung künstlicher Handschriftendaten fokussieren sich u.a. auf die Verbesserung der Erkennungsperformanz des Verifikationsalgorithmus, indem die Datenbasis (biometrische Daten) optimiert wird. Das geschieht zum einen mittels der Erzeugung künstlicher Individuen und entsprechender künstlicher Handschriftendaten und zum anderen werden reale Handschriftendaten dupliziert und entsprechend modifiziert. Dabei wird jedoch nicht auf den Erhalt des Schreibinhalts geachtet.
Arbeiten wie die von Guyon in [Guyo96] oder [HaAB16], welche sich mit der Erstellung von realwirkenden Handschriftenaufzeichnungen (z.B. Notizen, Briefe etc.) einer Person beschäftigen, haben nicht den Fokus, diese Handschriftendaten einem biometrischen Verifikationssystem zuzuführen. Hier zählen in erster Linie das optische Bild des erstellten künstlichen Schriftstücks und deren Schreibinhalt. Hierfür werden jedoch zum Teil eine Menge von Handschriftensamples (Buchstaben, Glyphen und/oder Wörter) benötigt, siehe Abbildung 33.
In dieser Arbeit soll ein Verfahren vorgestellt werden, welches zum einen den Schreibinhalt berücksichtigt und zum anderen sich für die Verifikation an einem handschriftenbasierten Verifikationssystem eignen soll.

sign (first name signature), Sign (formal signature), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, ?, . (dot), >, <, ` (back quote), (tilde), , (coma), ' (quote), " (double quote), ; (semi-colon), : (colon),], [, g, f, | (vertical bar), =, +, (underscore), - (minus),), (, *, &, ^(caret), %, $, #, , !, e, E, t, T, a, A, o, O, i, I, n, N, s, S, r, R, h, H, d, D, l, L, c, C, th, he, u, U, f, F, m, M, p, P, in, the, g, G, w, W, er, re, an, y, Y, on, b, B, ed, at, en, or, es, ti, v, V, to, te, nd, st, nt, ng, ar, it, of, ing, al, ha, as, ou, and, io, is, co, de, se, ve, ion, le, ai, ea, ro, me, id, k, K, ce, ent, li, ne, ll, sa, ri, ic, om, ra, fo, hi, il, tio, tion, si, pe, ec, pr, for, ta, la, be, ma, rs, di, ho, ut, aid, po, sai, said, tr, ca, ns, ur, ch, el, we, wa, wi, mi, ati, ts, ot, ee, un, no, rt, ac, ter, nc, ad, tha, ct, wh, ate, fi, ge, et, us, ol, hat, ie, her, pa, mo, ere, ers, ss, that, em, atio, ation, lo, na, ted, so, ly, con, ci, res, ni, os, ig, men, wo, ver, ow, ld, ir, ill, ul, su, vi, sh, ke, was, pl, pro, op, ment, ear, iv, ons, im, com, est, ov, ay, all, wit, fr, ther, with, ith, bu, ag, ia, bo, sp, mp, ev, his, are, nce, gh, av, rn, sta, fe, per, ep, ove, ab, am, x, X, ff, out, ort, gr, ry, tin, fro, bl, der, rom, by, oun, ive, do, tt, cl, ide, ef, from, rc, iti, tu, ex, go, ave, ect, ei, ty, tra, ga, cr, wer, ess, ls, int, red, eve, tor, rea, oul, ap, ting, pp, ba, cou, rat, j, J, over, nte, rm, cen, thi, cu, igh, not, str, up, ions, ye, uld, ist, man, oo, ore, ice, cti, ck, pre, lli, cent, ome, da, p or, ies, hav, ial, ain, era, ine, one, fa, du, rd, par, uc, ey, have, din, art, oc, ern, mil, our, port, nde, ght, nts, eg, but, den, were, od, ding, has, off, sti, und, oth, if, end, au, ont, sed, inc, lea, ew, tat, fic, als, sin, nti, who, ast, led, age, rr, lin, ste, rce, ki, rin, af, use, min, ant, bi, illi, lat, ht, sc, ity, ight, eas, cal, rec, nat, rep, enc, ug, een, yea, year, ned, gi, abo, ud, whi, mm, here, pla, ak, whe, pu, ru, rou, cha, han, cia, omp, othe, sio, sion, lit, comp, ue, other, pi, nal, q, Q, act, ft, qu, hey, ead, ica, wou, rg, lio, ring, ffi, lion, wor, sid, wil, ose, hou, stat, ici, tur, ffic, ces, lle, lu, will, tes, ssi, ral, ele, mon, ure, dent, rk, ade, ctio, ction, cial, ds, tri, side, ded, pri, eme, eat, ntr, tic, bou, ree, ny, anc, eco, pres, ind, had, lic, dr, its, mor, part, um, new, erc, oli, any, unt, comm, nter, tw, ple, nati, offi, ase, sho, offic, sto, va, abou, tho, omm, bout, tim, nin, ins, pos, hin, tiv, pol, mb, thr, oi, rit, dis, ndi, own, inte, iden, cc, rie, ost, eo, ua, ble, nv, hr, tte, she, tate, bee, ub, fu, ona, ning, app, hei, been, spe, fir, cont, lan, mu, thei, kin, att, fte, hea, aft, this, ten, hic, nm, erce, ass, z, Z, poli, ren, emen, perc, ran, ated, ust, les, nn, ser, ack, ugh, wn, mpa, lt, oug, ough, ys, old, rati, tal, ju, ement, nme, nmen, rcent, rcen, ile, rl, ok, ake, br, ever, las, nment, xp, ili, afte, exp, vern, eral, orm, fter, ui, king, ire, nu, gov, gove, gover, owe, hen, ence, tive, eri, ount, ina, whic, more, vo, ote, esi, ident, onal, ini, ong, epo, ound, rv, pen, tl, eir, tro, ob, har, rte, iona, day, mill, ime, coun, ling, nst, sen, ous, gu, mar, yo, bil, wee, milli, gn, ord, ign, esti, form, cre, ance, epor, eport, ich, ven, nf, ner, ars, uri, som, lly, gre, repo, some, eop, isi, ves, sec, ope, erv, ard, air, che, fici, spo, two, ip, peo, ical, oll, ene, arg, gen, rge, ame, cte, nge, tar, sing, med, rnm, itio, sur, ople, opl, ition, duc, ita, thin, aw, fore, icia, count, enti, can, cted, ms, pec, acc, og, resi, inter, nes, ori, oin, eal, rate, iz, tie, say, cons, ks, sm, pt, chi, abl, ors, mer, ell, gra, last, eca, low, work, cau, ens, lar, nder, rad, ses, time, tan, iss, when, ib, than, uni, plan, bec, sse, you, arr, ffici, ang, oa, eli, pan, ied, hos, unde, cat, icial, gro, ric, ges, ary, rel, mat, rke, ompa, ace, sit, yi, ficia, late, ber, lie, vic, ters, nk, ecti, uti, ict, ties, omi, aus, ered, inv, ws, cit, mbe, ppo, esid, dec, ans, tel, ears, fer, eside, des, rest, ite, ese, ying, yin, edi, siden, now, ork, olic, ppe, cor, fl, hing, ture, serv, mber, los, ress, leg, nsi, ates, how, call, ilit, ret, lled, ause, arge, war, gai, gain, mit, nta, ann, lso, ivi, econ, hu, ond, hil, bli, ark, pea, rti, itt, inst, ecu, ffe, eek, vis, fou, arl, usi, eac, son, hel, ved, mis, fin, eci, ali, reat, alle, lati, ease, gs, bill, hre, clu, mpl, sts, even, car, ubl, ked, ris, tem, rme, vin, irs, rre, ail, rac, uct, sl, beca, dit, under, orte, ece, able, try, rst, sup, rem, acti, rov, jo, cur, mpan, lead, firs, ries, nis, mos, ays, ssion, ssio, ange, erat, week, nds, ced, ster, unc, eed, rvi, aga, stra, imp, win, tors, mark, ncl, tak, ders, cto, ctor, ach, tre, prov, ise, ains, rop, lec, irst, ani, ari, ged.

Abbildung 33 Buchstaben, Glyphen und Wörtern verwendet von Guyon in [Guyo96]

Hierfür werden auf Basis realer Handschriftendaten künstliche Individuen erzeugt. Diese Individuen repräsentieren ein Alphabet aus Buchstaben und Zahlen. Mittels dieses Alphabets können beliebige Buchstaben zu Wörtern aneinandergereiht werden. Um die Intraklassen-Variabilität der Individuen zu adressieren, können die erzeugten künstlichen Wörter entsprechend manipuliert werden. Diese Wörter bzw. künstlichen Handschriftendaten können somit für verschiedenen Zweck eingesetzt werden. Sei es beispielsweise für die Erweiterung einer bestehenden Datenbasis, um die Verifikationsperformanz zu testen oder für die Simulation von Angriffen, wo beispielsweise der Schreibinhalt bekannt ist.
Die Interklassen-Variabilität wird künstlich erzeugt, indem das Alphabet buchstaben- bzw. ziffernweise verändert wird. So können beispielsweise die Schriftneigung, das Höhen/Seitenverhältnis und die Schreibdauer aller Buchstaben und Zahlen eines Individuums mittels Parameter modifiziert werden. Jeder Parametersatz repräsentiert entsprechend ein Schreibindividuum. In Abbildung 34 wird der Prozess grob dargestellt. Im ersten Schritt werden die Handschriftendaten des Alphabets von realen Personen aufgezeichnet und statistische Merkmale extrahiert, aus denen die Schreibsignale wieder generiert werden können (Rekonstruktionsmerkmale). Die Rekonstruktionsmerkmale aller Buchstaben bilden das Alphabet. Die Modifikation des Alphabets findet im zweiten Schritt statt, wobei hier spezielle Modifikationsparameter verwendet werden. Auf Basis des modifizierten

Alphabets (Rekonstruktionsmerkmale aller Buchstaben), welches ein Schreibindividuum darstellt, werden im dritten Schritt anschließend die Handschriftendaten generiert. Im letzten Schritt werden Modifikationsparameter verwendet, welche die Möglichkeit geben, die Intraklassen-Variabilität realer Handschriften einer Person zu simulieren.

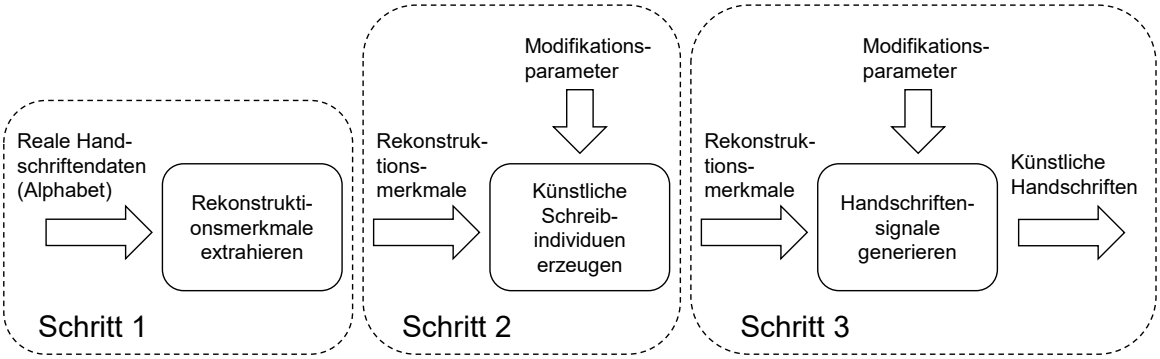


Abbildung 34 Grober Prozessablauf der Handschriftengenerierung

Eine detaillierte Beschreibung des Verfahrens wird im nachfolgenden Abschnitt 6.2 dargestellt.

6.2 Durchführung

Das in Abschnitt 6.1.2 ausgewählte Verfahren zur Erzeugung künstlicher Handschriften soll in diesem Abschnitt näher beschrieben werden. Die in dieser Arbeit erstellten Handschriftensignale basieren nicht auf Glyphen, sondern aus einem Alphabet Σ, welches eine bestimmte Anzahl von Symbolen (Ziffern und/oder Buchstaben) besitzt, z.B. $\Sigma = \{a, A, b, B, ..., z, Z\}$. Dies soll u.a. den Aufzeichnungsaufwand minimieren. Die Buchstaben werden entsprechend des gewählten Schreibinhalts aneinandergereiht. In der Arbeit von Hasselberg et al. in [HZKS+13], an der auch der Autor beteiligt war, wurde diese Vorgehensweise bereits erfolgreich getestet. Jedoch sollen in dieser Arbeit zusätzlich künstliche Schreibindividuen erzeugt werden. Jedes Schreibindividuum besitzt ein entsprechendes individuelles Basisalphabet (Ziffern und Buchstaben). Diese Methode wurde so noch nicht vorgestellt und stellt entsprechend eine Neuerung dar.
Zum besseren Verständnis wird zunächst in Abschnitt 6.2.1 eine Kurzbeschreibung des neuen Verfahrens präsentiert. Im Anschluss daran wird in Abschnitt 6.2.2 die Methode zur künstlichen Handschriftenerzeugung näher beschreiben.

6.2.1 Kurzbeschreibung des Verfahrens

Die Methode zur Erzeugung künstlicher Handschriften kann in zwei Phasen eingeteilt werden. In Phase 1 werden die künstlichen Individuen erzeugt und in Phase 2 auf Basis dieser künstlichen Individuen die künstlichen Handschriften generiert.

Phase 1: Künstliche Individuen erzeugen

Zur Erzeugung künstlicher Individuen werden abgeleitete Informationen von Handschriftendaten realer Personen verwendet. Dabei werden zunächst Buchstaben und Ziffern (Alphabet) von einer Person aufgezeichnet. Anschließend werden von diesen Handschriftensignalen statistische Merkmale abgeleitet (siehe Tabelle 20). Mit diesen Merkmalen ist es möglich, den jeweiligen Buchstaben oder die jeweilige Ziffer des Alphabets zu

rekonstruieren (Spline-Interpolationsverfahren). Mittels verschiedene Modifikationsparameter (siehe Tabelle 21) werden die statistischen Merkmale der Buchstaben bzw. Ziffern verändert und in einer Datenbank für jedes künstliche Schreibindividuum gespeichert. Ein Satz von Modifikationsparameter MP_1 bildet somit ein neues Basisalphabet BA_1 eines entsprechenden neuen künstlichen Schreibindividuums. Modifikationsparameter MP_2, welches ungleich ist von MP_1, erzeugt ein Basisalphabet BA_2, welches ungleich ist von BA_1. So können beliebig viele Basisalphabete erzeugt werden.

Phase 2: Künstliche Handschriften generieren

In der zweiten Phase werden auf Grundlage der Basisalphabete Handschriftensignale erzeugt. Zunächst werden ein Schreibindividuum und der Schreibinhalt gewählt. Anschließend wird das entsprechende Basisalphabet des Schreibindividuums geladen. Die Buchstaben werden entsprechend des Schreibinhalts mittels eines Spline-Interpolationsverfahrens erzeugt und aneinandergereiht. Vor der Erzeugung des Schreibsignals werden verschiedene Modifikationen durchgeführt, um eine "reale" Intraklassen-Variabilität zu simulieren. In Tabelle 22 werden die Modifikationsparameter für diesen letzten Schritt dargestellt. Die Handschriftendaten können anschließend z.B. in einer Verifikationsdatenbank gespeichert werden.

6.2.2 Ausführliche Beschreibung des Verfahrens

Innerhalb dieses Abschnitts soll die neue Methode zur Generierung von künstlichen Schreibindividuen und Handschriftensignalen genauer beschrieben werden.
Wie in Abschnitt 6.2.1 bereits erwähnt, kann das neue Verfahren in zwei Phasen eingeteilt werden. Innerhalb der ersten Phase werden auf Basis von abgeleiteten Merkmalen realer Handschriftendaten künstliche Basisalphabete generiert. In der zweiten Phase werden auf Basis dieser künstlichen Individuen Handschriftensignale erzeugt. In Abbildung 35 werden die beiden Phasen und die entsprechenden Schritte veranschaulicht.

Phase 1: Künstliche Individuen erzeugen

Im ersten Schritt muss ein Alphabet Σ definiert werden, welches den kompletten Zeichensatz bzw. alle Symbole enthält, auf deren Basis später in Phase 2 die künstlichen Handschriftensignale (Wörter) generiert werden sollen. Anschließend müssen alle Symbole des Alphabets Σ von einer realen Person einzeln aufgezeichnet bzw. digitalisiert werden. Die Aufzeichnung der analogen Signale kann beispielsweise mittels digitalem Signaturtablett durchgeführt werden.
Im nächsten Schritt werden bestimmte statistische Merkmale (Rekonstruktionsmerkmale *rm*) von jedem einzelnen Buchstaben (Symbol) extrahiert und gespeichert. Die aufgezeichneten Handschriftendaten der realen Person werden nach diesem Schritt nicht weiter benötigt und können gelöscht werden. Die extrahierten Merkmale, welche auf Basis der originalen Handschriftendaten extrahiert werden sind in Tabelle 20 aufgelistet. Sie ähneln denen der in der ISO/IEC 19794-11 beschriebenen Merkmale zur kompakten Speicherung von Handschriftendaten. In der Arbeit von Guest et al. [GuHH14], wird die Norm tiefgehend diskutiert und evaluiert.

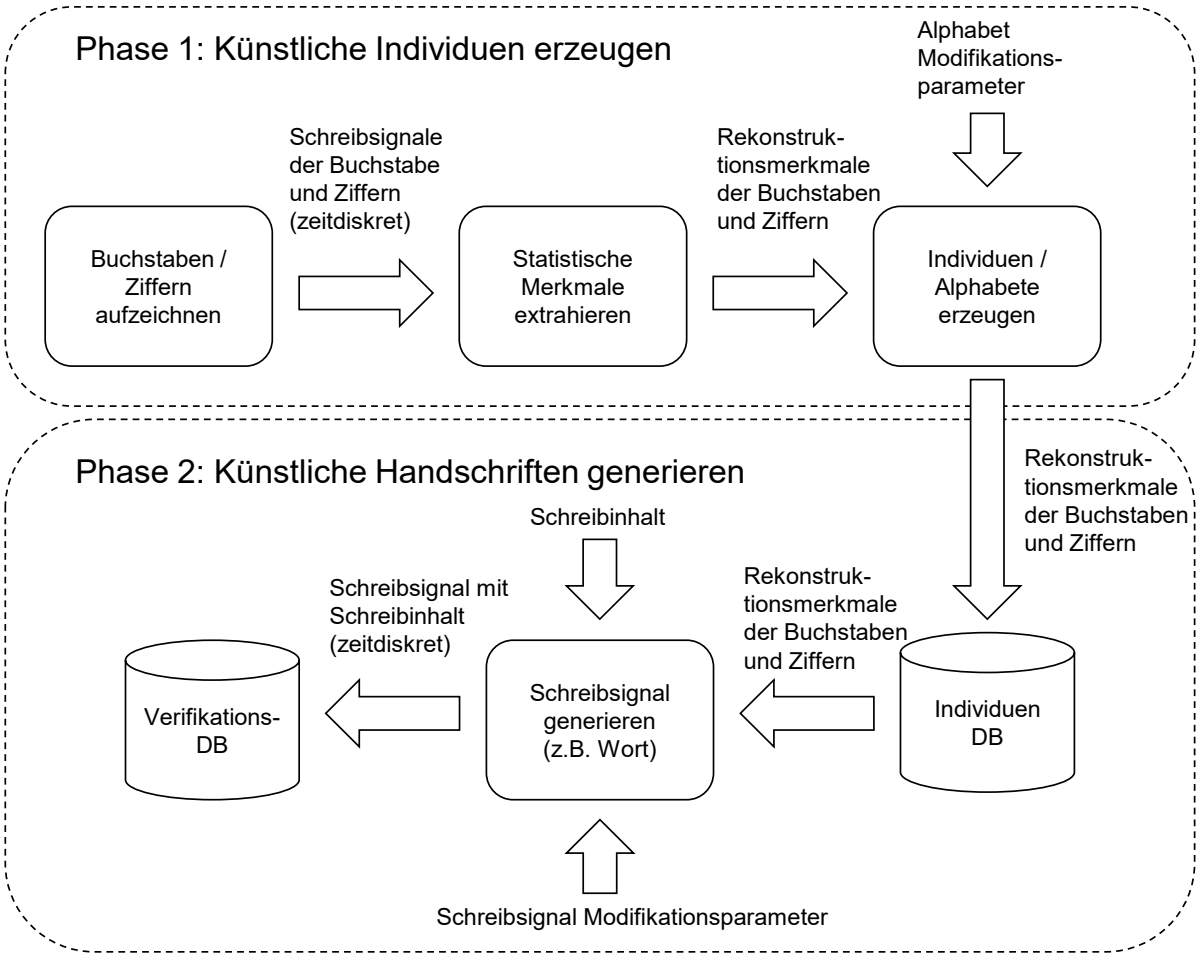


Abbildung 35 Ablauf der Generierung von künstlichen Handschriftendaten

Die wichtigsten Merkmale zur Rekonstruktion der jeweiligen Symbole (Buchstaben) bilden rm_1, rm_2 und rm_3. Sie beinhalten die Position und den Wert der Maxima und Minima des X-, Y- und P-Signals. Anhand dieser und weiterer Rekonstruktionsmerkmale $rm_{4\text{-}7}$ können mittels Interpolationsverfahren die jeweiligen Symbole (Buchstaben) des Alphabets Σ rekonstruiert werden. Die extrahierten Rekonstruktionsmerkmale rm aller Symbole des Alphabets Σ bilden somit das Basisalphabet BA. Die Rekonstruktionsmerkmale rm_1, rm_2, rm_3 und rm_4 sind Merkmalsmengen, wohingegen rm_5, rm_6 und rm_7 Skalare sind.

Tabelle 20 Extrahierte Merkmale die zur Generierung des Basisalphabets verwendet werden

Merkmal Nr.	Bezeichnung	Beschreibung
rm_1	ValMaximaX	Wert und Position aller Minima und Maxima des X Signals (alle Werte Normalisiert [0...1])
rm_2	ValMaximaY	Wert und Position aller Minima und Maxima des Y Signals (alle Werte Normalisiert [0...1])
rm_3	ValMaximaP	Wert und Position aller Maxima und Minima des P Signals (alle Werte Normalisiert [0...1])
rm_4	PosPenUp	Zeitliche Position aller Stiftabsetzer (Normalisiert [0...1])
rm_5	ttotal	Dauer der Aufzeichnung
rm_6	XYRatio	Verhältnis von maximalen X Wert zu maximalen Y Wert
rm_7	MaxPressure	Maximal aufgezeichneter Druck

Für die Erstellung der künstlichen Basisalphabete bzw. Schreibindividuen werden im nächsten Schritt der Phase 1 die Rekonstruktionsmerkmale *rm* mittels Modifikationsparameter *mp* modifiziert. Dabei werden u.a. die Positionen und der Werte der Stützpunkte (Splines) geändert. Des Weiteren können das Höhen- und Seitenverhältnis angepasst und die Schriftneigung geändert werden. Alle Modifikationsparameter sind Skalare und in Tabelle 21 aufgelistet, in der letzten Spalte der Tabelle sind beispielhaft Werte angegeben, wie sie auch teilweise in der Evaluation verwendet wurden. Die ersten drei Modifikationsparameter ModTtotal, ModXYRatio und ModMaxPressure sind einfache Faktoren. Sie werden mit den entsprechenden Merkmalen multipliziert. Die nächsten Modifikationsparameter der Tabelle ModMaximaXVal, ModMaximaXTime, ModMaximaYVal, ModMaximaYTime, ModMaximaPVal und ModMaximaPTime werden mit einer jeweils zufällig erzeugten Zahl (RNG), welche zwischen 0 und 1 liegt, multipliziert und anschließend mit dem entsprechenden Merkmal multipliziert. Somit bestimmen sie welches Gewicht/Einfluss die zufällig erzeugte Zahl auf das Merkmal hat. Der letzte Modifikationsparameter ModFontSlam repräsentiert den Scherungsfaktor und wird eingesetzt um die Schriftneigung zu bestimmen.
Am Ende dieses Schrittes steht die Menge der geänderten Rekonstruktionsmerkmale aller Buchstaben und bildet das Basisalphabet eines Schreibindividuums *BA*. Unter Verwendung anderer Modifikationsparameterwerte mp_x können entsprechend weitere Basisalphabete BA_x und somit weitere Schreibindividuen generiert werden.

Tabelle 21 Modifikationsparameter zur Generierung neuer Basisalphabete(Schreibindividuen)

Parameter	Beschreibung	Beispiel
ModTtotal	Modifikation der gesamten Schreibdauer	0.8
ModXYRatio	X/Y - Verhältnis anpassen	1.2
ModMaxPressure	Maximalen Druckpunkt ändern	0.85
ModMaximaXVal	Spline Stützpunkt des X Signals modifizieren (Höhe)	0.01
ModMaximaXTime	Spline Stützpunkt des X Signals modifizieren (Position)	0.01
ModMaximaYVal	Spline Stützpunkt des Y Signals modifizieren (Höhe)	0.015
ModMaximaYTime	Spline Stützpunkt des Y Signals modifizieren (Position)	0.015
ModMaximaPVal	Spline Stützpunkt des P Signals modifizieren (Höhe)	0.25
ModMaximaPTime	Spline Stützpunkt des P Signals modifizieren (Position)	0.25
ModFontSlam	Schriftneigung ändern	0.45

Phase 2: Künstliche Handschriften generieren

Im ersten Schritt der Phase 2 wird ein aus Phase 1 generiertes Basisalphabet ausgewählt. Zusätzlich wird ein Schreibinhalt *si* definiert (beliebige Verkettung von Symbolen des Alphabets *Σ*). Nun werden die Rekonstruktionsmerkmale der benötigten Buchstaben zur Konstruktion des Schreibinhalts *si* verwendet, um die entsprechenden Schreibsignal mittels einer Interpolationsmethode zu erzeugen. Um die Intraklassen-Variabilität realer Handschriftendaten zu adressieren, werden vor der Signalgenerierung die Rekonstruktionsmerkmale der jeweiligen Buchstaben leicht modifiziert. Bei diesem Schritt wird das Basisalphabet nicht geändert, sondern lediglich die verwendeten Rekonstruktionsmerkmale der jeweiligen Buchstaben temporär modifiziert. Nach der Rekonstruktion der Buchstaben werden die Änderungen wieder verworfen. Die erzeugten bzw. rekonstruierten Buchstabensignale werden nun aneinandergereiht und zu einem Schreibsignal zusammengefasst. Da Buchstaben im Verhältnis zueinander unterschiedlich groß (z.B. A und i) und im Verhältnis zur Schriftlinie (Basislinie) anders positioniert sein können (z.B. a und g), werden sie in dieser Arbeit in Buchstabenklassen eingeteilt. Diese Buchstabenklassen werden dann entsprechend ihrer Eigenschaften aneinandergereiht. In Abbildung 36 werden Beispiele für Buchstaben und Buchstabenklassen gegeben und wie diese im Verhältnis zu den Typografie-Linien stehen. Dabei bilden die Buchstaben der Klasse 1 alle Buchstaben, welche sich von der p-Line bis zur Mittellinie erstrecken. Buchstaben der Klasse 2 erstrecken sich von der Grundlinie bis zur Mittellinie. Buchstabenklasse 3 beinhaltet alle Buchstaben, welche von der Grundlinie bis zur h-Linie bzw. k-Linie reichen. Alle Buchstaben, die sich von der p-Linie bis zur h/k-Linie erstrecken, werden der Buchstabenklasse 4 zugeordnet. In dieser Arbeit wird zur Vereinfachung kein Unterschied zwischen der in der DIN-Norm 16507-2 [DIN19] erwähnten h- und k-Linie gemacht.

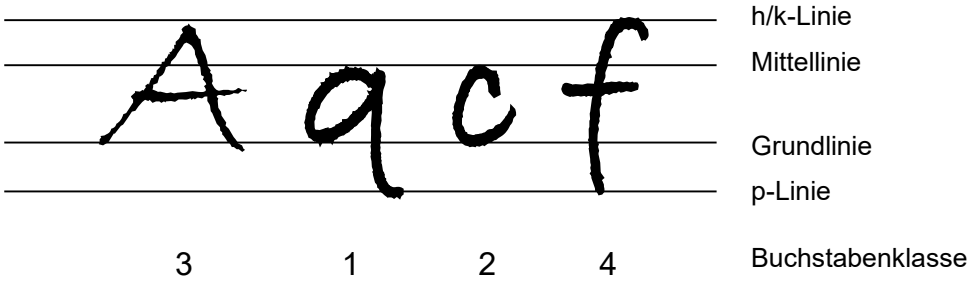


Abbildung 36 Typografie-Linien und Buchstabenklassen

Beim Prozess der Aneinanderreihung der Buchstaben können weitere Parameter, wie z.B. der Buchstabenabstand, definiert werden. Alle Modifikationsparameter *mpS*, welche für die Generierung des künstlichen Schreibsignals verwendet werden, sind in Tabelle 22 dargestellt. Mit dieser Modifikation des künstlichen Schreibsignals soll eine „reale“ Intraklassen-Variabilität nachgeahmt werden.

Für das vorgestellte neue Verfahren zur Generierung von künstlichen Schreibindividuen und Handschriftensignalen soll in dieser Arbeit u.a. untersucht werden, inwieweit sich diese Schreibindividuen für die Erstellung einer Testdatenbank zur Evaluierung handschriftenbasierter Verifikationssysteme eignen. Insbesondere soll untersucht werden, inwieweit sich die Individuen bzw. dessen künstliche Schreibsignale voneinander unterscheiden und ob diese Unterscheidung ähnlich „groß“ ist, wie die realer Schreibsignale untereinander.

Tabelle 22 Modifikationsparameter *mpS* zur Generierung des Handschriftensignals

Parameter	Beschreibung
InterpolMethode	Verwendete Interpolationsmethode z.B. Linear Interpolation (linear), Interpolation mit Polynomen dritten Grades (cubic), nächster Nachbar Interpolation (nearest)
ModMaximaXVal	Spline Stützpunkt des X Signals modifizieren (Höhe)
ModMaximaXTime	Spline Stützpunkt des X Signals modifizieren (Position)
ModMaximaYVal	Spline Stützpunkt des Y Signals modifizieren (Höhe)
ModMaximaYTime	Spline Stützpunkt des Y Signals modifizieren (Position)
ModMaximaPVal	Spline Stützpunkt des P Signals modifizieren (Höhe)
ModMaximaPTime	Spline Stützpunkt des P Signals modifizieren (Position)
ModSampleRate	Abtastrate modifizieren (zeitlicher Abstand zwischen zwei Signalpunkten)
ModLetterDistance	Abstand zwischen den Buchstaben/Ziffern modifizieren

Ferner soll geprüft werden, ob diese Schreibindividuen eingesetzt werden können, um Angriffsdaten zu erzeugen, welche für einen Angriff auf ein Verifikationssystem verwendet werden können. Zusätzlich soll untersucht werden, inwieweit die Modifikationsparameter Einfluss auf das Aussehen (real wirkend) der künstlich erzeugten Handschriftensignale haben.
Für diese Untersuchungen wird der Verifikationsalgorithmus von Vielhauer [Viel06] als Verifikationssystem fungieren. Des Weiteren soll der gleiche Verifikationsalgorithmus verwendet werden, um das Angriffspotential der Schreibindividuen zu evaluieren. Die Ergebnisse sollen dazu beitragen, potentielle Designvorschläge für die Generierung künstlicher Schreibsignale zu liefern. So soll beispielsweise ermittelt werden, welche Parameter Einfluss auf das Aussehen und die Fehlerraten der künstlichen Handschriftensignale haben.

6.3 Experimentelle Tests

Die ersten durchgeführten experimentellen Teste des neuen Verfahrens sollen in diesem Abschnitt erläutert werden. Dazu werden im folgenden Abschnitt 6.3.1 der Evaluationsaufbau und die Messmethodik erläutert. Im Abschnitt 6.3.2 werden die Ergebnisse präsentiert und bewertet.

6.3.1 Messmethodik und Evaluationsaufbau

Die Rahmenbedingungen für die Evaluation des neuen Verfahrens zur Generierung von künstlichen Schreibindividuen und künstlichen Handschriftendaten wird in diesem Abschnitt präsentiert.
Das Verfahren soll in zwei verschiedene Einsatzszenarien evaluiert werden. Zum einen soll die Verifikationsperformanz getestet werden und zum anderen die Angriffsperformanz.

Verifikationsperformanz
In diesem Szenario soll untersucht werden, inwieweit die erzeugten künstlichen Handschriftendaten sich als Testdaten für biometrische Verifikationssysteme eignen. Dabei soll u.a. geprüft werden, ob sich die künstlichen Handschriftendaten wie reale Handschriftendaten hinsichtlich ihrer Verifikationseigenschaften verhalten. Insbesondere die Interklassen-Variabilität und Intraklassen-Variabilität sollen näher betrachtet werden.

Angriffsperformanz
Das neue Verfahren soll zusätzlich dahingehend untersucht werden, ob es sich für die Durchführung eines Angriffs eignet. Dabei sollen künstliche Handschriftendaten verwendet werden, um unberechtigten Zugriff auf ein Verifikationssystem zu erlangen. Das neue Verfahren soll insbesondere bei Angriffsszenarien hinsichtlich der Erfolgsrate evaluiert werden, bei denen der Schreibinhalt der anzugreifenden biometrischen Handschriftendaten bekannt ist.

Als Datenbasis beider Szenarien (Verifikationsperformanz / Angriffsperformanz) dienen u.a. Handschriftensignale (Alphabete) von elf Personen. Die Auswahl von lediglich elf Personen ist für erste Tests ausreichend, da die biometrischen Daten nicht direkt für die Erzeugung verwendet werden. Die Alphabete bestehen aus den Groß- und Kleinbuchstaben des modernen lateinischen Alphabets $\Sigma_{lat} = \{a, A, b, B, ..., z, Z\}$ und aus den arabischen Zahlen $\Sigma_{ziff} = \{0, 1, ..., 9\}$. Somit ergibt sich ein Zeichensatz bzw. Alphabet mit $\Sigma = \Sigma_{lat} \cup \Sigma_{ziff}$ für diese Evaluation. Dabei wurden von jeder Person fünf Handschriftensignale eines Symbols des Alphabets aufgezeichnet. Somit werden für das Alphabet 26x5 Kleinbuchstaben, 26x5 Großbuchstaben und 10x5 Ziffern pro Person aufgezeichnet, insgesamt 3410 Handschriftensignale. Zusätzlich wurden von diesen elf Personen je zehn Handschriftensignale der folgenden Semantiken (Schreibinhalte) aufgezeichnet:

- **Seife:** wurde gewählt, da diese Semantik Buchstaben der Klasse 2, 3, und 4 enthält (siehe Abschnitt 6.2.2)
- **arbeiten:** wurde gewählt, weil kein Großbuchstabe enthalten ist und ein Buchstabe (e) doppelt vorkommt
- **Iraq:** die englische Schreibweise des Landes Irak wurde gewählt, weil es relativ wenig Buchstaben enthält und diese der Klasse 1, 2 und 3 zugeordnet werden können (siehe Abschnitt 6.2.2)
- **2759:** die vier Ziffern sollen stellvertretend für eine feste PIN stehen, sie wurden zufällig gewählt

Für diese Evaluation wurden andere Semantiken verwendet als im Abschnitt 5.3. Gründe hierfür sind u.a. der vorgegebene Schreibinhalt, der bekannt sein muss, um diesen entsprechend künstlich reproduzieren zu können. Die in Abschnitt 5.3 verwendeten Semantiken sind außer der vorgegeben PIN "geheim" bzw. können frei gewählt werden. Weiterhin sind die Schreibinhalte in diesem Abschnitt so gewählt, dass sie zusammen alle Buchstabenklassen (siehe Abbildung 36) abdecken.
Mit diesen 440 Handschriften-Signalen und den Alphabet-Handschriftendaten (3410) werden insgesamt 3850 Handschriftensignale (350 pro Person) für diese Evaluation verwendet.
Um die Verifikationsperformanz des Verfahrens zu evaluieren, werden von jeder Person die Alphabet-Handschriftendaten optimiert/modifiziert (Phase 1 siehe Abschnitt 6.2). Dabei wird von jedem Buchstaben im Alphabet ein sogenannter Mittelwertbuchstabe gebildet. Das geschieht wie folgt: Von jedem der fünf Handschriftendaten eines Buchstabens werden die Rekonstruktionsmerkmale extrahiert (siehe Tabelle 20). Anschließend werden die Mittelwerte der Rekonstruktionsmerkmale errechnet. Bei den Rekonstruktionsmerkmalen rm_5, rm_6 und rm_7 ist dieser Schritt eindeutig, da hier nur ein Wert pro Merkmal und Buchstabe ermittelt wird. Bei den Merkmalen rm_1 bis rm_4 können jedoch mehrere Werte

pro Merkmal ermittelt werden. Das Rekonstruktionsmerkmal rm_4 (zeitliche Position der Stiftabsetzer) kann aus einem Vektor bestehen, wobei die Anzahl der Elemente des Vektors die Anzahl der Stiftabsetzer und die Elemente die zeitliche Position (normiert [0 ... 1]) der Stiftabsetzer abbildet. Um hier den Mittelwert der Stiftabsetzer zu bestimmen, wird die Länge jedes Vektors ermittelt, diese können unterschiedlich sein. Die Vektorlänge, welche am häufigsten auftritt, wird verwendet. Anschließend wird von allen Vektoren mit dieser Länge der arithmetische Mittelwert der jeweiligen Elemente gebildet.
Bei den Rekonstruktionsmerkmalen rm_1, rm_2 und rm_3 (Position und Wert der Maxima/Minima für X-Signal (rm_1), Y-Signal (rm_2) und P-Signal (rm_3)) besteht die Berechnung der Merkmale je aus einer Matrix mit zwei Spalten. Die Anzahl der Zeilen *i* dieser Matrix beschreibt die Summe der auftretenden Maxima und Minima. In der ersten Spalte der Matrix werden die zeitlichen Positionen der Maxima bzw. Minima hinterlegt, wohingegen in der zweiten Spalte die dazugehörigen Werte (Höhe) der Maxima und Minima gespeichert werden. Die Werte jeder Spalte sind normiert [0 ... 1], entsprechend beträgt der Minimalwerte 0 und der Maximalwert 1 für die jeweilige Spalte. Für die Berechnung des Mittelwertes $\bar{A}$ dieses Merkmals (Matrix) aller fünf Buchstaben-Handschriftensignale wird die Anzahl der Zeilen *i* jeder Matrix bestimmt. Die am häufigsten auftretende Anzahl von Zeilen *i* wird gewählt. Alle Matrizen mit dieser Anzahl von *i* Zeilen werden für die weitere Berechnung verwendet. Da die Matrizen die gleiche Länge haben, können sie relativ einfach mit einer Matrizenaddition zu einer Matrix A_{add} mit $A_{add} = A_1+A_2+\ldots+A_k$ zusammengefasst werden, wobei *k* die Anzahl der Matrizen mit der Länge *i* repräsentiert. Anschließend wird jedes Element der Matrix A_{add} mit $\frac{1}{k}$ multipliziert (Skalarmultiplikation). Die Ermittlung des „arithmetischen Mittelwertes“ für alle Matrizen A_1 - A_k mit gleicher Länge *i* kann wie in Formel 25 dargestellt zusammengefasst werden.

$$\bar{A} = (A_1 + A_2 + \cdots A_k) * \frac{1}{k} \qquad \textbf{Formel 25}$$

Besitzen die fünf Matrizen eines Merkmals alle eine unterschiedliche Länge *i*, so wird die Matrix als Mittelwert gewählt, welche die drittgrößte Länge *i* besitzt. So sollen potentiell auftretende extreme Werte nach oben bzw. nach unten kompensiert werden. Hier sind auch andere Strategien möglich, so kann beispielsweise auch die Matrix mit der größten oder mit der kleinsten Länge *i* als Mittelwert gewählt werden.
Am Ende dieses Prozesses wird von jedem Buchstaben und jeder Ziffer des Alphabets ein Satz von Rekonstruktionsmerkmalen stehen. Diese Rekonstruktionsmerkmale bilden das Mittelwertalphabet, es ist mit Σ_{mean} definiert. Auf Basis dieses Mittelwertalphabets werden weitere künstliche Basisalphabete erzeugt (siehe Abschnitt 6.2). Hierfür werden Modifikationsparameter eingesetzt. Ein Satz von Modifikationsparametern bestimmt ein neues künstliches Basisalphabet. Geeignete Modifikationsparameter *mp,* für die Erzeugung künstlicher Individuen, und Modifikationsparameter *mpS,* zur Erzeugung der Handschriften, sollen innerhalb der Evaluation ermittelt werden.
Hierfür wird zunächst ein beliebiger Modifikationsparametersatz *mp* gewählt (siehe Tabelle 23). Basierend auf diesen Parametern werden künstliche Handschriftenindividuen erzeugt, indem die Parameter auf die Mittelwertalphabete Σ_{mean1} bis Σ_{mean11} angewendet werden. Die Modifikationsparameter sind Faktoren, die mit dem entsprechenden Merkmal multipliziert werden. Danach werden künstliche Handschriftendaten erzeugt, wobei die in Tabelle 25 gelisteten Parameter *mpS* verwendet werden, um diese leicht zu

modifizieren. Anschließend werden die künstlich erzeugten Handschriftensignale optisch untersucht, inwieweit diese realwirkend erscheinen. Weiterhin werden die Fehlerraten (FRR, FAR, EER, CRR, RR und CR) der künstlichen Signale bestimmt und geprüft, ob diese sich ähnlich verhalten wie Fehlerraten realer Handschriftendaten. Nach der Prüfung der Fehlerraten und der optischen Eigenschaften der künstlichen Handschriftendaten, werden die Parameter *mp* und *mpS* so geändert, dass sich die optische Erscheinung und die Fehlerraten denen realer Handschriftendaten annähern. Anschließend beginnt der Prozess wieder von vorne bis geeignete Modifikationsparametersätze (*mp / mpS*) ermittelt wurden. Diese ermittelten Parameter werden für die weiteren experimentellen Tests verwendet.

Tabelle 23 Initiale Modifikationsparameter zur Erstellung der künstlichen Schreibalphabete

Mod. Parameter	**Künstliches Basisalphabet (Schreibindividuum)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ModTtotal	0,8	1,2	1,6	0,8	1,2	1,6	1,4	2	2	2
ModXYRatio	1,2	1,2	1,8	1,2	1,2	1,8	1,3	1	1	1
ModMaxPressure	0,85	0,85	0,85	0,85	0,85	0,85	0,9	1,3	1,3	1,3
ModMaximaXVal	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModMaximaXTime	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModMaximaYVal	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModMaximaYTime	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModMaximaPVal	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModMaximaPTime	0,01	0,015	0,05	0,01	0,015	0,05	0,05	0,1	0,25	0,4
ModFontSlam	0,45	0,9	1,35	-0,45	-0,9	-1,35	-1,10	0	0	0

Weiterhin werden in den nachfolgenden Untersuchungen vier Buchstabenklassen verwendet, die bei der Aneinanderreihung der entsprechenden Buchstaben berücksichtigt werden. In Tabelle 24 wird die Zuordnung der Buchstaben und Zahlen in die entsprechende Klasse dargestellt. Der Schritt zur Erzeugung von künstlichen Basisalphabeten wird für alle elf Personenhandschriftendaten durchgeführt. Insgesamt werden so 110 künstliche Basisalphabete erzeugt.

Tabelle 24 Einordnung der Zeichen in Buchstabenklassen

Buchstabenklasse	**Zeichen der jeweiligen Klasse**
1	g, j, p, q, y
2	a, c, e, m, n, o, r, s, u, v, w, x, z
3	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, b, d, h, i, k, l, t, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
4	f

Im nächsten Schritt (Phase 2) werden auf Basis der künstlichen Alphabete für vier verschiedene Schreibsemantiken (Seife, arbeiten, Iraq und 2759) je zehn Handschriftendaten generiert. Um die Intraklassen-Variabilität zu simulieren, werden bei der Generierung der Handschriftensignale die ermittelten Modifikationsparameter *mpS* verwendet, die im vorherigen Test bestimmt wurden. Die zehn verschiedenen Parametersätze werden für alle 110 künstlichen Basisalphabete zur Generierung der vier Semantiken verwendet. Somit ergeben sich 4400 künstliche Handschriftensignale für die Evaluation. Der

Modifikationsparameter InterpolMethode (siehe Tabelle 25) beschreibt die Art der verwendeten Interpolationsmethode, neben den in dieser Evaluation verwendeten Methode (pchip) sind u.a. die Lineare Interpolation oder "Nearest Neighbor" Interpolation ebenfalls möglich. Die für die Modifikationsparameter benötigte Pseudozufallszahl besitzt einen Wertebereich zwischen null und eins [0,1].

Tabelle 25 Initiale Modifikationsparameter der künstlichen Handschriftensignale

Mod. Parameter	Künstliches Handschriftensample	Hinweis / Erklärung
	1 - 10	
InterpolMethode	pchip	Piecewise Cubic Hermite Interpolating Polynomial (Kubisch Hermitescher Spline)
ModMaximaXVal	0,07	Eine Pseudozufallszahl wird für jeden Stützpunkt (Spline-Wert) erzeugt. Danach wird die Zufallszahl mit dem angegeben Faktor (hier 0,07) und dem Spline-Wert multipliziert. Das Ergebnis wird anschließend mit dem Spline-Wert addiert.
ModMaximaXTime	0,07	
ModMaximaYVal	0,07	
ModMaximaYTime	0,07	
ModMaximaPVal	0,07	
ModMaximaPTime	0,07	
ModSampleRate	10	10 Millisekunden (ms) Abtastrate
ModLetterDistance	10	10 Pixel Abstand zwischen zwei Zeichen

Für die Aufzeichnung der Handschriftendaten der elf Personen wurde das Grafiktablett (Pen Display) Cintiq 21UX der Firma Wacom verwendet. In Tabelle 26 sind einige ausgewählte technische Eigenschaften des Geräts für den interessierten Leser gelistet.

Tabelle 26 Technische Daten des Aufnahmegerätes Wacom Cintiq 21UX

Komponente	Eigenschaft
Display	1600x1200 Pixel(UXGA), 21,3“ Bilddiagonale
Sensorauflösung	0,005 mm pro Punkt (5080 Linien pro Zoll)
Abtastrate	200 Samplepunkte pro Sekunde
Sample	x und y Koordinate, Druck und Zeitangabe
Druckstufen	1024

Für die Evaluation wird der Verifikationsalgorithmus von Vielhauer [Viel06] verwendet. Bei der Evaluation der Verifikationsperformanz sollen folgende Aspekte betrachtet werden. Zum einen soll die Intraklassen-Variabilität der künstlichen Individuen und zum anderen die Interklassen-Variabilität untersucht werden. Für die Untersuchung werden für jede Semantik die Fehlerraten FAR, FRR und EER ermittelt. Zusätzlich werden die Reproduktionsrate (RR) und Kollisionsrate (CR) der künstlichen Handschriftendaten pro Semantik (Seife, arbeiten, Iraq und 2759) berechnet. Dabei werden für das Enrollment die ersten fünf der zehn künstlichen Handschriftensignale verwendet und für die Verifikation die restlichen fünf Handschriftensignale. Um eine Aussage über die Intraklassen-Variabilität treffen zu können, werden die Falschrückweisungsraten (FRR) und Reproduktionsraten (RR) mit denen von realen Handschriftensignalen vergleichen. Um die Interklassen-Variabilität der künstlichen Handschriftensignale zu ermitteln, werden Kollisionsrate (CR) und

Falschakzeptanzrate (FAR) der künstlichen mit denen der realen Handschriftensingale verglichen.
Bei der Ermittlung der Fehlerraten für die realen Handschriftensignale der elf Personen werden die fünf Schreibsemantiken (Seife, arbeiten, Iraq und 2759) verwendet. Auch hier werden die ersten fünf Schreibsignale für das Enrollment und die restlichen fünf für die Verifikation verwendet. Die Fehlerraten (FAR, FRR und EER) bzw. Kollisions- und Reproduktionsrate (CR/RR) der künstlichen Handschriftensignale sollen pro Semantik mit denen der realen Handschriftensignale vergleichen werden. Sind die Werte ähnlich, ist das ein erstes Zeichen dafür, dass die künstlichen Handschriftendaten ein ähnliches Verifikationsverhalten besitzen wie die realen Handschriftendaten.

Für die Evaluation der Angriffsperformanz der künstlichen Handschriftendaten, werden die künstlichen Handschriftensignale verwendet, um reale Handschriftensignale zu imitieren. Für diese Evaluation werden die ersten fünf Handschriftensignale der realen Handschriften (Semantik: Seife, Iraq, arbeiten und 2759) für das Enrollment verwendet. Die fünf restlichen Handschriftensignale dienen der Verifikation, um die Fehlerraten bzw. Reproduktionsraten zu bestimmen. Anschließend werden auf Basis der künstlichen Individuen Angriffshandschriftensignale des Schreibinhalts der entsprechenden Semantiken erzeugt. Dabei sollen jedoch die künstlichen Individuen nicht die biometrischen Referenzdaten "angreifen", von denen sie ursprünglich abstammen bzw. gebildet wurden. Somit soll verhindert werden, dass potentielle übertragene biometrische Handschrifteninformationen bei der Bildung der Individuen die Evaluation verfälschen. Es wird in den ersten Tests je zehn künstliche Handschriftensignale und anschließend je 100 Handschriftensignale von zufällig gewählten Individuen pro Semantik erzeugt und für die Verifikation verwendet. Somit wird jede reale Person pro Semantik von einem zufällig gewählten Schreibindividuum angegriffen. Die so entstanden FAR_{att} soll mit der FAR_{org} verglichen werden. Ziel aus Sicht eines Angreifers ist es, die FAR in die Höhe zu treiben, also die FAR_{att} höher ausfällt als die FAR_{org}.

6.3.2 Präsentation und Bewertung der Ergebnisse

Für die Bestimmung der Verifikationsperformanz als auch für die Bestimmung der Angriffsperformanz werden die Erkennungs- bzw. Fehlerraten der originalen Handschriftendaten zum Vergleich benötigt. Hierfür werden nachfolgend die ermittelten Toleranzfaktoren für den jeweiligen Arbeitsmodus beschrieben und die entsprechenden Fehlerraten bestimmt.
In der Abbildung 37 wird für die Bestimmung des Toleranzfaktors der jeweiligen Arbeitsmodi (EER und CRR) die Verifikationsperformanz der Semantik 2759 beispielhaft dargestellt. Die Verifikationsperformanz aller übrigen Semantiken sind im Anhang Anlage 6 zu finden.

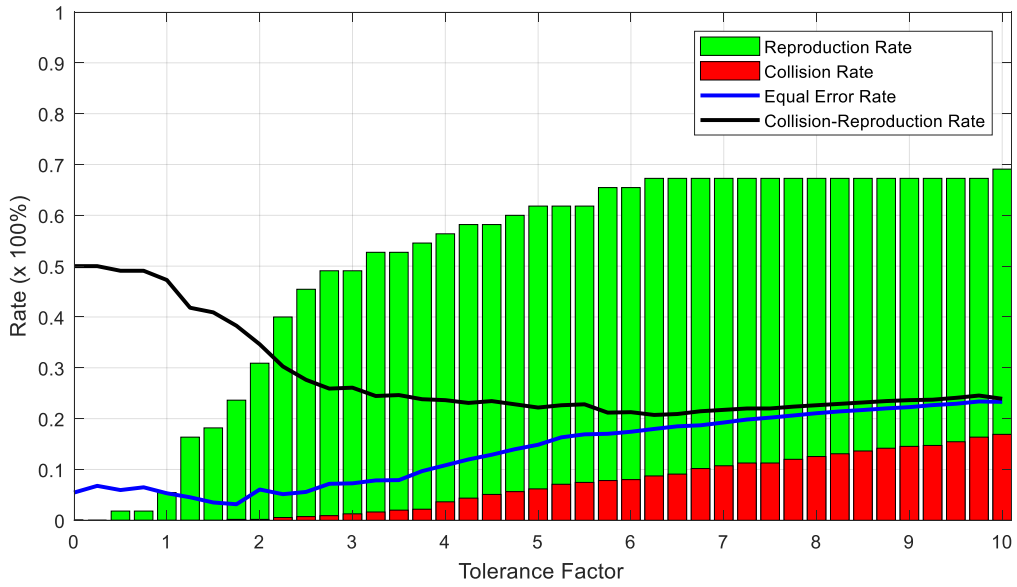


Abbildung 37 Bestimmung des Toleranzfaktors hier beispielhaft für die Semantik 2759

Beim EER Arbeitsmodus wird der Toleranzfaktor so gewählt, dass die geringste EER erzielt wird. Während beim CRR Arbeitsmodus die Systemparameter (Toleranzfaktor) so gewählt wird, dass die CRR am geringsten ist.
Die Ergebnisse der Toleranzfaktorbestimmung aller Semantiken sind in Tabelle 27 gelistet. Die ermittelten Toleranzfaktoren werden für die nachfolgende Evaluation der Fehlerraten (FAR, FRR und EER) verwendet.

Tabelle 27 Ermittelte Toleranzfaktoren für die Semantik "2759", "arbeiten", "Iraq"und "Seife"

Arbeitsmodi	Semantikklasse	Toleranzfaktor
EER	2759	1,75
	arbeiten	0
	Iraq	0,75
	Seife	0,5
CRR	2759	6,25
	arbeiten	5
	Iraq	9,25
	Seife	6,25

In Abbildung 38 werden für die echten Handschriftendaten die Fehlerraten FAR und FRR als auch die Equal Error Rate (EER) für die Semantik "Iraq" im Arbeitsmodus EER exemplarisch abgebildet. In Anhang Anlage 7 sind die Kurvenverläufe der Fehlerraten aller Semantiken in den jeweiligen Arbeitsmodi dargestellt. Zusätzlich sind in Tabelle 28, Tabelle 29 und Tabelle 30 alle ermittelten Fehlerraten zusammengefasst.

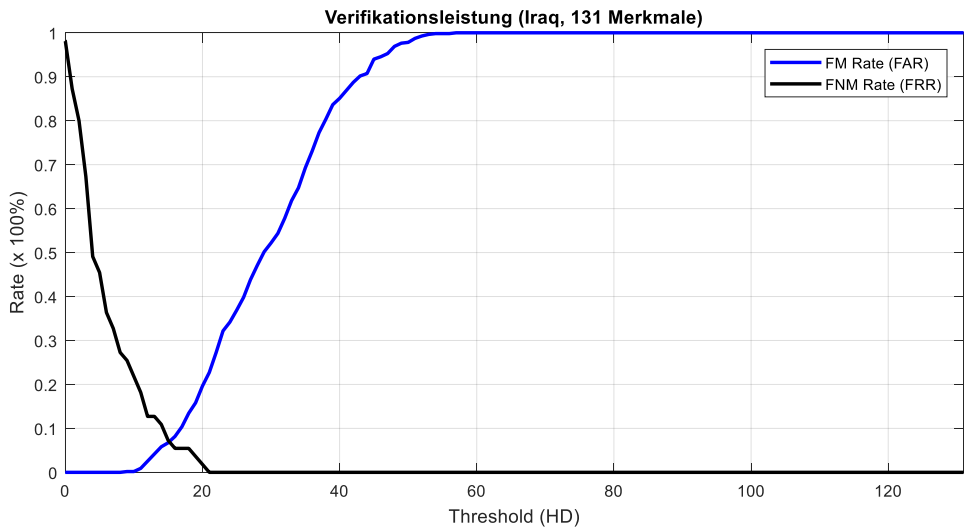


Abbildung 38 Fehlerraten der Semantik "Iraq"

In Tabelle 28 sind die Fehlerraten des nicht optimierten Systems dargestellt. Wobei die Fehlerraten im nicht optimierten Modus (Tabelle 28), für die niedrigste EER (Tabelle 29) und niedrigste CRR (Tabelle 30) bestimmt wurden. Die "EER Th." beschreibt in den jeweiligen Tabellen den Schwellenwert (Threshold), bei welchem die entsprechende EER aufgetreten ist.

Tabelle 28 Fehlerraten aller vier Semantiken (nicht optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	47,27	5,45	0	5,31	10,03
arbeiten	48,18	3,63	0	5,45	13,16
Iraq	47,27	5,45	0	8,15	11,51
Seife	44,54	10,90	0	2,18	9,80

Die EER-optimierten Fehlerraten sind in Tabelle 29 aufgezeigt. Es ist zu erkennen, dass sich die Gleichfehlerrate (EER) im Vergleich zum nicht optimierten System entsprechend verbessert hat.

Tabelle 29 Fehlerraten aller vier Semantiken (EER optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	38,27	23,63	0,18	3,15	4,13
arbeiten	50,00	0	0	1,33	47,26
Iraq	49,09	1,81	0	6,96	15,16
Seife	49,09	1,81	0	1,81	18,80

In Tabelle 30 sind die Fehlerraten für den CRR-optimierten Betrieb dargestellt. Es ist zu erkennen, dass die Reproduktionsrate sich gegenüber dem nicht optimierten System stark verbessert hat und wie erwartet, die Kollisions-Reproduktionsrate verringert werden konnte.

Tabelle 30 Fehlerraten aller vier Semantiken (CRR optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	20,72	67,27	8,72	17,97	0,54
arbeiten	19,45	63,63	2,54	11,49	0,91
Iraq	23,81	60,00	7,63	19,47	0,86
Seife	29,54	45,45	4,54	17,01	0,82

Bestimmung der Modifikationsparameter

Die Ergebnisse für die Bestimmung der geeigneten Modifikationsparameter werden nachfolgend präsentiert. In Tabelle 31 werden die Fehlerraten für die Initialen Modifikationsparameter *mp* dargestellt. Hier ist zu erkennen, dass die Gleichfehlerraten (EER) gegenüber realer Fehlerraten (siehe z.B. Tabelle 28) relativ gering sind. Weiterhin treten bei den künstlichen Handschriftendaten keine Kollisionen auf (Kollisionsraten CR = 0%).

Tabelle 31 Fehlerraten künstlicher Handschriftensignale zur Bestimmung geeigneterParameter

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	19,00	62,00	0	0,0385	7,78
arbeiten	21,18	57,63	0	0,0251	6,86
Iraq	16,18	67,63	0	0,0316	11,82
Seife	22,27	55,45	0	0,1558	13,14

Die optischen Eigenschaften der künstlichen Handschriftensignale sind beispielhaft in Abbildung 39 dargestellt. In der ersten Spalte A sind drei Signale mit dem Schreibinhalt „Iraq“ dargestellt. Die Spalten B und C repräsentieren Schreibsignale mit den Schreibinhalten „arbeiten“ bzw. „Seife“. Bei den beispielhaften Darstellungen ist zu erkennen, dass die Schreibinhalte zum Teil nicht mehr lesbar sind und die Signale entsprechend unnatürlich wirken. Die hier gezeigten künstlichen Schreibsignale wurden auf Basis der künstlichen Schreibindividuen 8, 9 und 10 (siehe Modifikationsparameter Tabelle 23) generiert. Für diese Schreibindividuen sind die Parameter, welche das X- und Y- Signal modifizieren, am größten. Damit ist der Einfluss auf diese Werte entsprechend grösser, als bei den übrigen Schreibindividuen.

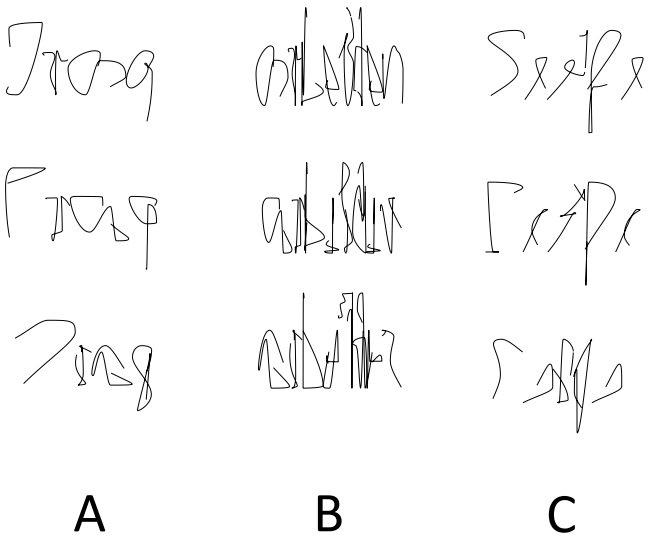


Abbildung 39 Beispielhafte Darstellung künstlicher Handschriftensignale (Initiale Parameter)

Die Modifikationsparameter wurden für die Schreibindividuen 8, 9 und 10 entsprechend angepasst und verringert (fett markiert). In Tabelle 32 sind die neuen Parameter dargestellt, wobei auch die übrigen Schreibindividuen minimal angepasst wurden.

Tabelle 32 Modifikationsparameter zur Erzeugung künstl. Schreibindividuen (2. Iteration)

Mod. Parameter	**Künstliches Basisalphabet (Schreibindividuum)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ModTtotal	0.8	1.2	1.6	0.8	1.2	1.6	2	2	2	0.8
ModXYRatio	1,10	1,20	1,30	1,40	1,50	1,60	1,10	1,20	1,30	1,40
ModMaxPressure	0,5	0,7	0,9	1	1,1	1,2	0,5	0,7	0,9	2
ModMaximaXVal	0.01	0.015	0.02	0	0	0	0.01	**0.015**	**0.02**	**0.05**
ModMaximaXTime	0.01	0.015	0.02	0	0	0	0.01	**0.015**	**0.02**	**0.05**
ModMaximaYVal	0	0	0	0.01	0.015	0.02	0.01	**0.015**	**0.02**	**0.05**
ModMaximaYTime	0	0	0	0.01	0.015	0.02	0.01	**0.015**	**0.02**	**0.05**
ModMaximaPVal	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ModMaximaPTime	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ModFontSlam	0.15	0.25	0.35	0.45	0.55	-0,15	-0,25	-0,35	-0,45	-0,55

Die Fehlerraten der künstlichen Schreibindividuen für die 2. Iteration sind in Tabelle 33 dargestellt. Es ist zu erkennen, dass die Gleichfehlerraten (EER) noch geringer sind als die der ersten initialen Erstellung (siehe Tabelle 31).

Tabelle 33 Fehlerraten künstlicher Handschriften (2. Iteration)

Semantik	**CRR in %**	**RR in %**	**CR in %**	**EER in %**	**EER Th.**
2759	45,27	9,45	0	0	22,50
arbeiten	43,00	14,00	0	0	29,00
Iraq	41,00	18,00	0	0,0574	35,84
Seife	43,45	13,09	0	0	25,50

Abbildung 40 zeigt beispielhaft die künstlichen Handschriftensignale der Schreibindividuen 8, 9 und 10 (von oben nach unten). Es ist zu erkennen, dass sich nach der Änderung der Modifikationsparameter der optische Eindruck verbessert hat. Der Schreibinhalt die Schreibindividuen ist besser zu erkennen (A:„Iraq“, B:„arbeiten“ und C:„Seife“). Im nächsten Schritt sollen die Modifikationsparameter *mpS* zur Erstellung der Schreibsignale angepasst werden, um die Intraklassen-Variabilität realistischer zu gestalten.

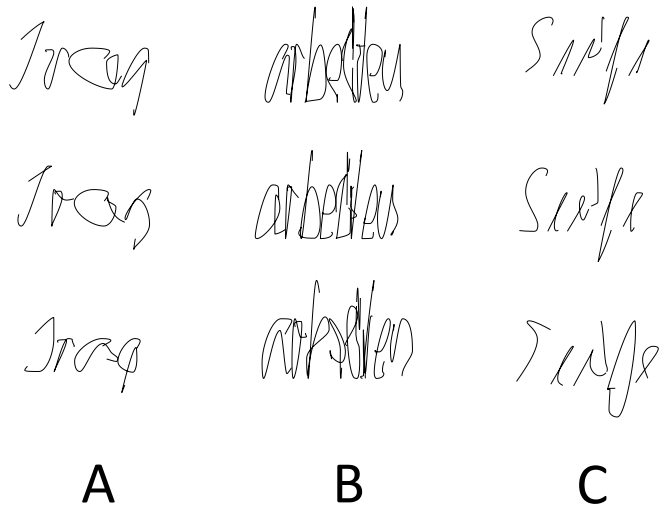


Abbildung 40 Beispielhafte Darstellung künstlicher Schreibsignale (2. Iteration)

In der Tabelle 34 sind die Modifikationsparameter entsprechend dargestellt. Die Werte haben sich gegenüber den initialen Parametern nicht geändert, jedoch das Vorzeichen. Hier werden zufällig negative oder positive Vorzeichen für die entsprechenden Parameter (in der Tabelle mit +/- markiert) erzeugt. Negative Vorzeichen bedeuten, dass sich die entsprechenden Werte der Stützpunkte (Spline-Werte) auch verringern können.

Tabelle 34 Modifikationsparameter zur Erzeugung der künstlichen Handschriftendaten (3. It.)

Mod. Parameter	**Künstliches Handschriftensample**	**Hinweis / Erklärung**
	1 - 10	
InterpolMethode	pchip	Piecewise Cubic Hermite Interpolating Polynomial (Kubisch Hermitescher Spline)
ModMaximaXVal	(+/-) 0,07	Eine Pseudozufallszahl wird für jeden Stützpunkt (Spline-Wert) erzeugt. Danach wird die Zufallszahl mit dem angegeben Faktor (hier 0,07) und dem Spline-Wert multipliziert. Das Ergebnis der Multiplikation wird anschließend mit dem Spline-Wert addiert.
ModMaximaXTime	(+/-) 0,07	
ModMaximaYVal	(+/-) 0,07	
ModMaximaYTime	(+/-) 0,07	
ModMaximaPVal	(+/-) 0,07	
ModMaximaPTime	(+/-) 0,07	
ModSampleRate	10	10 Millisekunden (ms) Abtastrate
ModLetterDistance	10	10 Pixel Abstand zwischen zwei Zeichen

Die Fehlerraten für die 3. Iteration zur Bestimmung geeigneter Modifikationsparameter sind in Tabelle 35 aufgelistet. Die Gleichfehlerraten sind gegenüber der 2. Iteration wieder gestiegen, jedoch noch immer geringer als die realer Handschriftendaten.

Tabelle 35 Fehlerraten künstlicher Handschriften (3. Iteration.)

Semantik	**CRR in %**	**RR in %**	**CR in %**	**EER in %**	**EER Th.**
2759	46,00	8,00	0	0,0425	16,76
arbeiten	45,72	8,54	0	0	17,00
Iraq	42,90	14,18	0	0,1818	21,77
Seife	45,27	9,45	0	0,0873	21,52

Der optische Eindruck der 3. Generation von künstlichen Handschriftendaten hat sich nicht wesentlich geändert, da die Modifikationsparameter *mp* zur Erzeugung der künstlichen Schreibindividuen nicht geändert wurden.
Im Anhang Anlage 8 sind, aus Gründen der Übersichtlichkeit dieses Abschnittes, die übrigen Iterationsstufen veranschaulicht dargestellt. Das Ergebnis der iterativen Bestimmung geeigneter Modifikationsparameter ist in Tabelle 32 und Tabelle 50 dargestellt. Die hier ermittelten Modifikationsparameter bilden optisch real wirkende künstliche Handschriftensignale. Weiterhin führen sie zu künstlichen Schreibindividuen (Mittelwertalphabete), welche sich optisch voneinander unterscheiden. Die Fehlerraten sind im Vergleich zu realen Handschriftensignalen geringer. Die Tests haben gezeigt, dass die Fehlerraten zwar erhöht werden können, um denen realer Handschriftendaten zu ähneln, sich dies jedoch auf das Erscheinungsbild der künstlichen Handschriftensignale negativ auswirkt. Die hier gewählten Modifikationsparameter stellen dementsprechend einen Kompromiss aus dem real wirkenden optischen Eindruck und realen Fehlerraten dar. In den nachfolgenden experimentellen Untersuchungen werden diese Parameter verwendet.

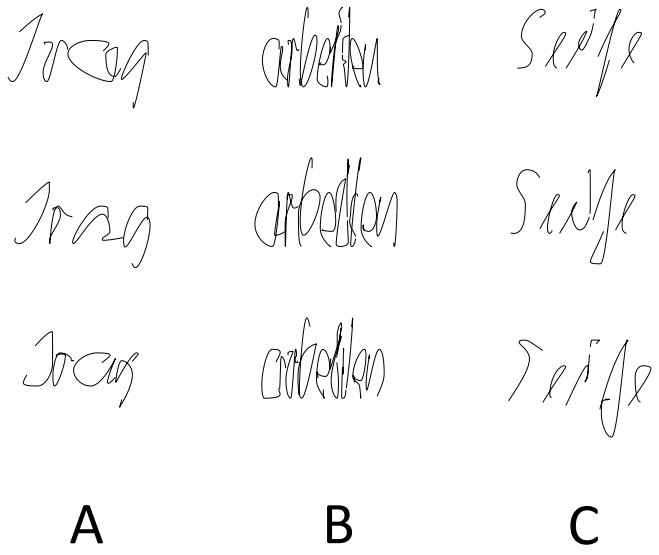


Abbildung 41 Beispielhafte Darstellung künstlicher Schreibsignale (3. Iteration)

Erzeugen künstlicher Schreibindividuen und Schreibsignale
Aufbauend auf den Schreibsignalen der realen Personen (Buchstaben und Zahlen) und den ermittelten Modifikationsparametern sind Mittelwertalphabete generiert worden. In Abbildung 42 sind Beispiele einer solchen Generierung dargestellt. Die jeweils links dargestellten fünf Buchstaben bzw. Zahlen sind reale Handschriftensignale, die im roten Kästchen markierten Buchstaben sind die generierten Mittelwertbuchstaben. Es kann festgestellt werden, dass einige Mittelwertbuchstaben unleserlich sind und kaum noch als Buchstabe erkannt werden können (siehe Buchstabe A bei User 3 und User 5). Ein Grund hierfür könnten die Stiftabsetzer zum Schreiben des waagerechten Striches beim Buchstabe A der originalen Handschriftendaten sein. Die Zeit, bei dem der Stiftabsetzer beim Schreibprozess durchgeführt wird, wird bei der Berechnung des Mittelwertbuchstabens ggf. zu einem falschen Zeitpunkt gesetzt. Das würde auch erklären, warum das A bei User 1 solche Schreibartefakte nicht besitzt, hier wurde das A in einem Zug ohne Stiftabsetzer geschrieben.

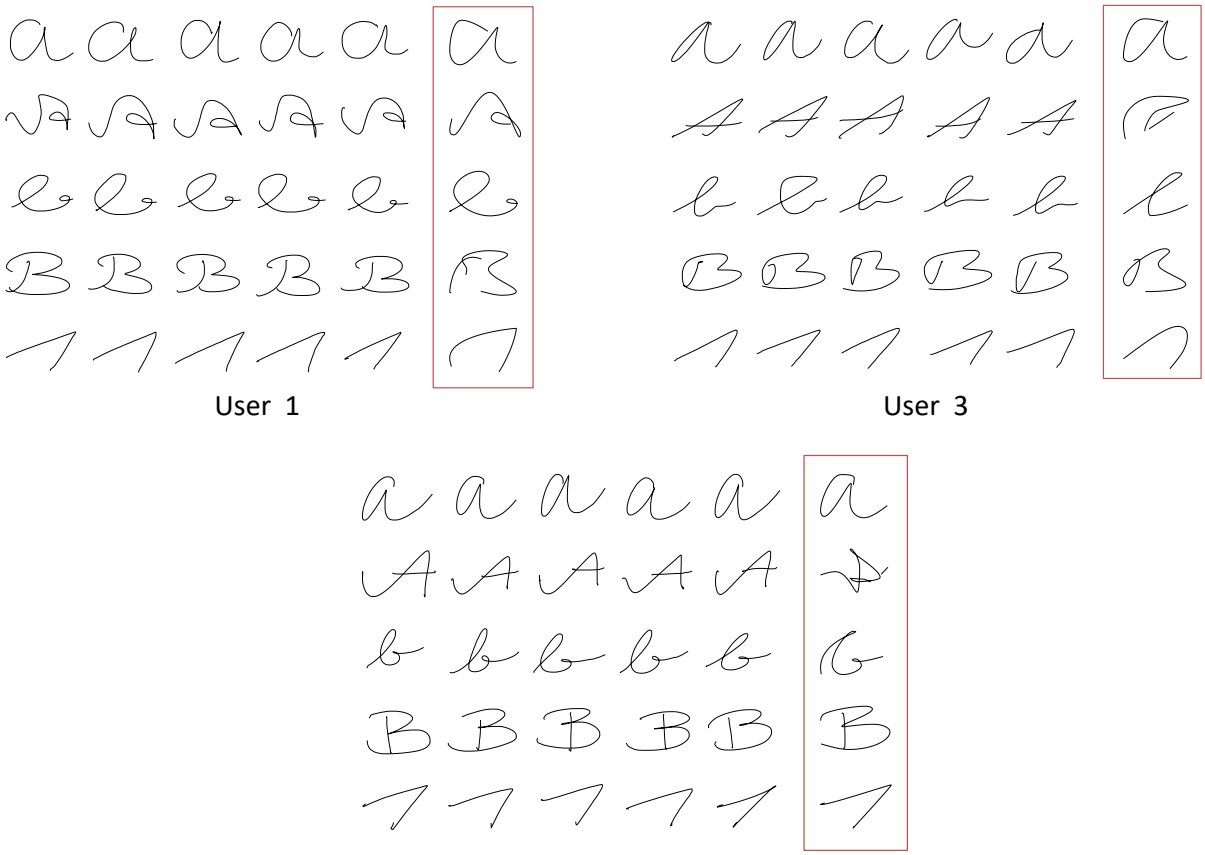
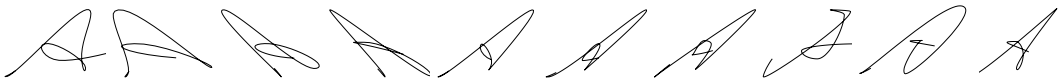


Abbildung 42 Beispiele für die Generierung von Mittelwertbuchstaben (rot markiert)

Anschließend wurden künstliche Alphabete auf Basis dieser Mittelwertalphabete gebildet. In Abbildung 43 sind Beispiele für die Buchstaben S, e, t und die Ziffer 3 gegeben. Jeder Buchstabe gehört entsprechend zu einem künstlichen Schreibindividuum. Die angegebenen User in der Abbildung geben die Herkunft des Basisalphabets an, auf welchem die künstlichen Buchstaben gebildet wurden. Es ist klar zu sehen, dass der Großteil der hier beispielhaft dargestellten Buchstaben lesbar ist.

User 11 (Buchstabe S)

User 5 (Buchstabe e)



User 4 (Buchstabe t)

User 2 (Ziffer 3)

Abbildung 43 Beispiele für künstl. Buchstaben auf Basis der jeweiligen Mittelwertalphabete

Auch sind von allen generierten künstlichen Schreibindividuen Schreibsignale der vier Semantiken erzeugt worden. In Abbildung 44 sind beispielhaft für die Semantik "Seife" zehn Schreibsignale eines Individuums in einer Zeile dargestellt. Insgesamt sind Schreibsignale von zehn Schreibindividuen in dieser Abbildung gezeigt. Alle gezeigten Schreibindividuen basieren auf dem Basisalphabet einer realen Person, in diesem Fall von "User 5". Im Anhang Anlage 9 sind für die Semantiken "2759", "Iraq" und "arbeiten" ebenfalls Schreibsignale der künstlichen Schreibindividuen beispielhaft dargestellt.
Es kann festgestellt werden, dass die Handschriftensignale zwar lesbar, teils jedoch nicht natürlich wirken. So sind bspw. die i-Punkte im Wort "Seife" teils nicht mehr zu identifizieren. Weiterhin ist bei der Ziffer sieben (Semantik „2759“), teilweise der mittlere waagerechte Strich ebenfalls nicht erkennbar. Zusammenfassend kann festgestellt werden, dass die künstlichen Handschriftendaten auf den ersten Blick teils echt wirken, bei genauerer Betrachtung, auch für einen Laien, aber unnatürlich wirken können. Das liegt u.a. auch daran, dass für die ersten Versuche keine geeignete Technik angewendet wurde, die Buchstaben miteinander zu verbinden.

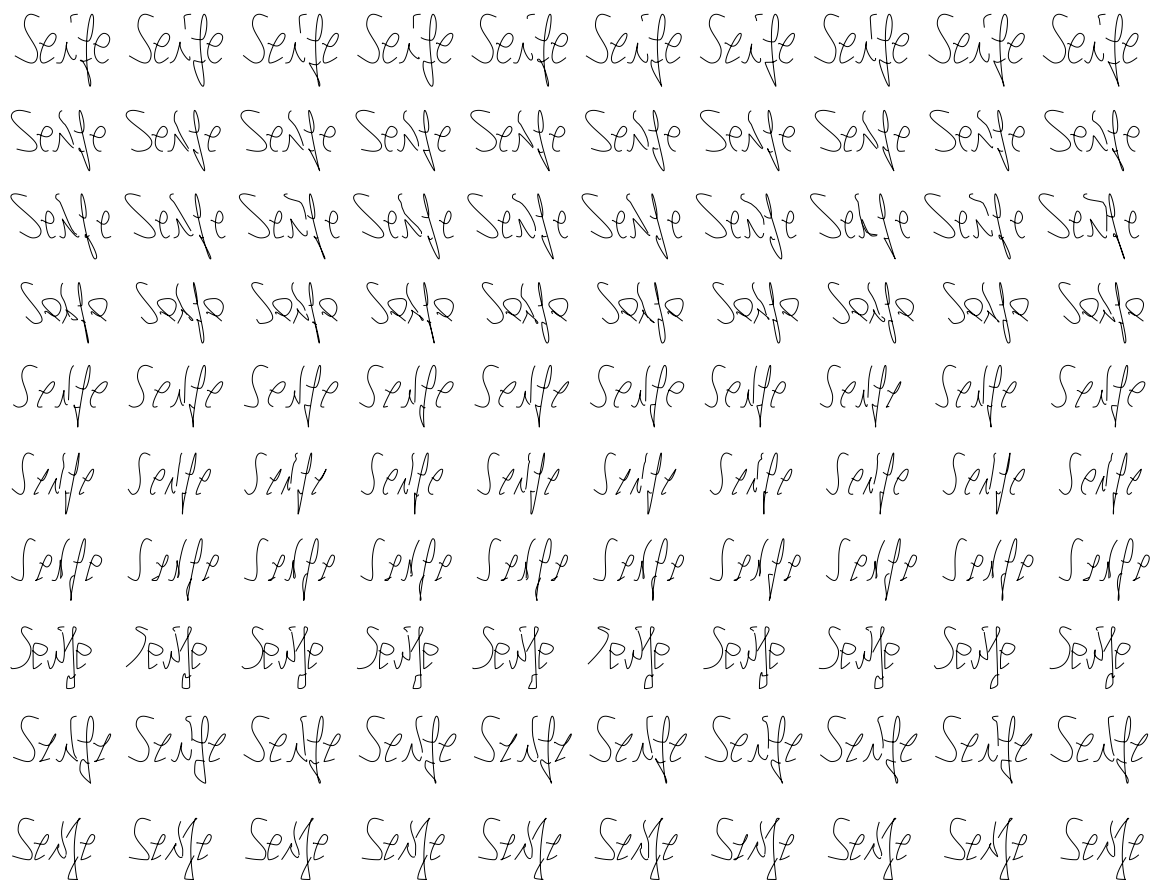


Abbildung 44 Schreibsignale von zehn künstlichen Schreibindividuen der Semantik "Seife"

Fehlerratenbestimmung der künstlichen Handschriften

Nachdem alle künstlichen Schreibindividuen und entsprechende Handschriftensignale für alle vier Semantiken erzeugt wurden, sind die Fehlerraten der ausschließlich künstlich erzeugten Handschriftendaten ermittelt worden. In Abbildung 45 sind exemplarisch die Kurvenverläufe der Fehlerraten (FAR und FRR) für die Semantik "Iraq" dargestellt. Sie zeigt einen, für ein handschriftenbasiertes Verifikationssystem vergleichsweise guten Kurvenverlauf. Die FRR und FAR liegen relativ weit auseinander und die EER ist mit ca. 0.7% recht gering. Für die übrigen Semantiken verhält es sich mit den Fehlerkurven ähnlich. Im Anhang Anlage 10 sind die FAR/FRR Diagramme aller Semantik abgebildet.

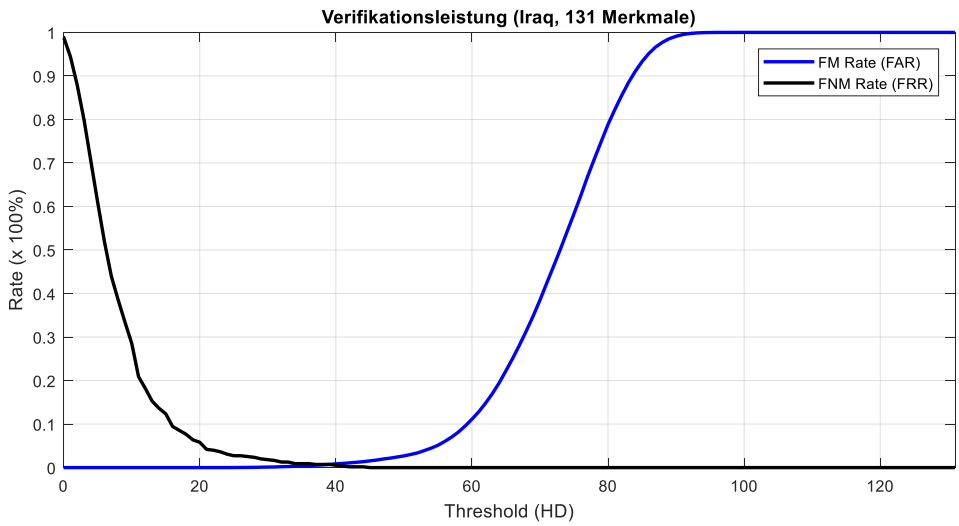


Abbildung 45 Verifikationsperformanz künstlicher Handschriftensignale der Semantik "Iraq"

In Tabelle 36 sind zusammenfassend die erreichten Fehlerraten entsprechend der Semantik dargestellt. Bei der Bestimmung der Fehlerraten wurden, wie in Abschnitt 6.3.1 angegeben, ausschließlich künstliche Schreibindividuen verwendet.

Tabelle 36 Fehlerraten der künstlichen Schreibindividuen (nicht optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	45,81	8,36	0	0,1818	20,04
arbeiten	46,54	6,90	0	0,0517	24,71
Iraq	42,81	14,36	0	0,7273	29,11
Seife	45,36	9,27	0	0,0522	19,71

Die Bestimmung der optimalen Toleranzfaktoren (EER und CRR Modus) ist in Abbildung 46 für die Semantik "2759" beispielhaft dargestellt. Auch hier ist zu erkennen, dass die EER untypisch gering ausfällt und nur sehr kleine Kollisionsraten ermittelt werden können. Die Diagramme der übrigen Semantiken sind im Anhang Anlage 11 einzusehen. Auch diese zeigen sehr geringe EER-Verläufe und Kollisionsraten.

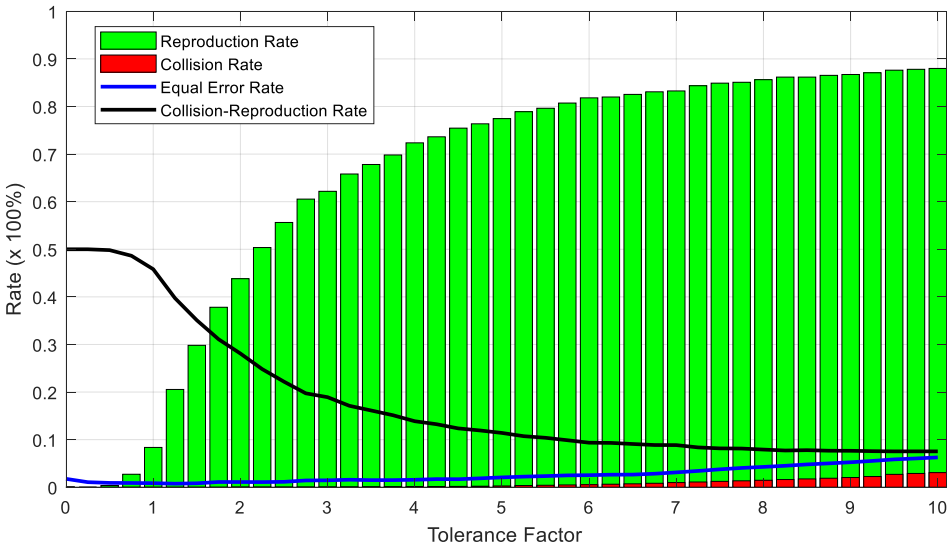


Abbildung 46 Bestimmung des Toleranzfaktors künstlicher Signale (2759)

Die Evaluation der Verifikationsperformanz der künstlich generierten Handschriftensignale zeigt einen besseren Fehlerkurvenverlauf, als vergleichbare originale Handschriftendaten (siehe Abbildung 38). Dies sind u.a. Anzeichen dafür, dass die Interklassen-Variabilität der künstlichen Schreibindividuen entsprechend geringer ist, als die von realen Personen (niedrige ERR). Auch die Intraklassen-Variabilität scheint gegenüber originalen Handschriftensignalen geringer zu sein, die Gleichfehlerrate (EER) ereignet sich bei einen höheren Schwellenwert (ca. 20) als bei den realen Handschriften (Schwellenwert ca. 10). Weiterhin sind die Reproduktionsraten der künstlichen Signale im nicht optimierten System überwiegend höher, als die der originalen (nicht optimierten) Handschriften, vgl. Tabelle 28. Dies sind im Vergleich zu den realen Handschriftensignalen eher untypische Fehlerraten und Kurvenverläufe.

Tabelle 37 Ermittelt Toleranzfaktoren der künstlichen Handschriftensignale

Arbeitsmodi	Semantikklasse	Toleranzfaktor
EER	2759	1,25
	arbeiten	0,5
	Iraq	0,5
	Seife	1,25
CRR	2759	9.75
	arbeiten	8,5
	Iraq	9,5
	Seife	9,0

Auf Basis der ermittelten Toleranzfaktoren (siehe Tabelle 37) der jeweiligen Arbeitsmodi (EER und CRR) sind die entsprechenden Fehlerraten für die künstlichen Handschriftensignale bestimmt worden. In Tabelle 38 sind die Fehlerraten im EER Modus dargestellt, wohingegen die Fehlerraten des CRR Modus in Tabelle 39 gelistet sind.

Tabelle 38 Fehlerraten der künstlichen Handschriftensignale (EER optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	41,72	16,54	0	0,1517	17,16
arbeiten	49,90	0,18	0	0,1403	36,22
Iraq	49,45	1,09	0	0,7263	39,00
Seife	42,00	16,00	0	0,0541	17,70

Auch die in den Tabellen dargestellten Fehlerraten bestätigen die untypischen Werte für Handschriftensignale. So ist die EER im CRR-optimierten System vergleichsweise gering und es sind keine Kollisionsraten ermittelbar, was angesichts der Diagramme zur Bestimmung der Toleranzfaktoren (siehe Abbildung 46) bereits vermutet werden konnte.

Tabelle 39 Fehlerrasten der künstlichen Handschriftensignale (CRR optimiert)

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	18,27	63,45	0	0	3,50
arbeiten	17,18	65,63	0	0	5,50
Iraq	19,81	60,36	0	0,1452	8,60
Seife	19,00	62,00	0	0	6,00

Angriffsperformanz der künstlichen Handschriftensignale

In Abbildung 47 ist beispielhaft die Angriffsperformanz der künstlichen Handschriftendaten in Form der Fehlerratenkurven für die Semantik "Iraq" dargestellt. Dabei ist klar zu erkennen, dass die FMR der künstlich erzeugten Angriffsdaten (rot gestrichelte Linie) die FNMR Kurve nicht schneidet und somit keine EER bestimmt werden kann. Ziel eines Angreifers in diesem Szenario ist es jedoch, eine höhere EER zu erzielen und somit, bei entsprechend gewähltem Systemtoleranz, eine höhere Chance einer positiven Verifikation der Angriffsdaten besteht. In Abbildung 48 sind zum Vergleich die Fehlerkurven unter Verwendung von 100 Angriffsdaten pro User dargestellt. Hier ist erkennbar, dass der Kurvenverlauf (FMR Angriff, rot gestrichelt) etwas steiler verläuft und sich der FMR etwas annähert. Jedoch ist auch unter Verwendung von 100 Angriffssamples keine EER ermittelbar.

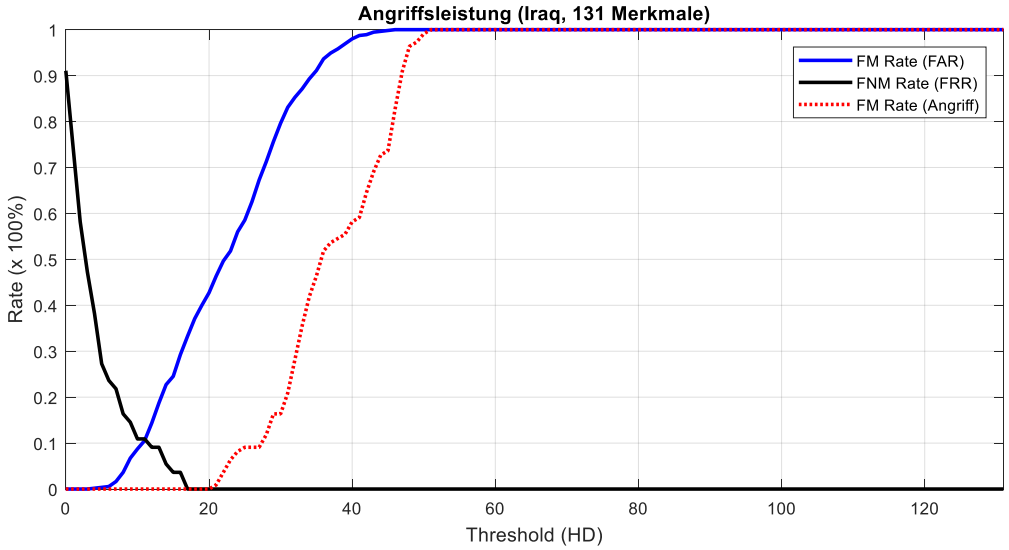


Abbildung 47 Angriffsperformanz für die Semantik Iraq (10 Angriffssamples)

Im Anhang Anlage 12 können die Fehlerkurven aller Semantiken betrachtet werden. Auch bei den übrigen Semantiken ist der Verlauf der Fehlerkurven ähnlich. Bei keiner der Semantiken konnte eine EER der Angriffsdaten verzeichnet werden.

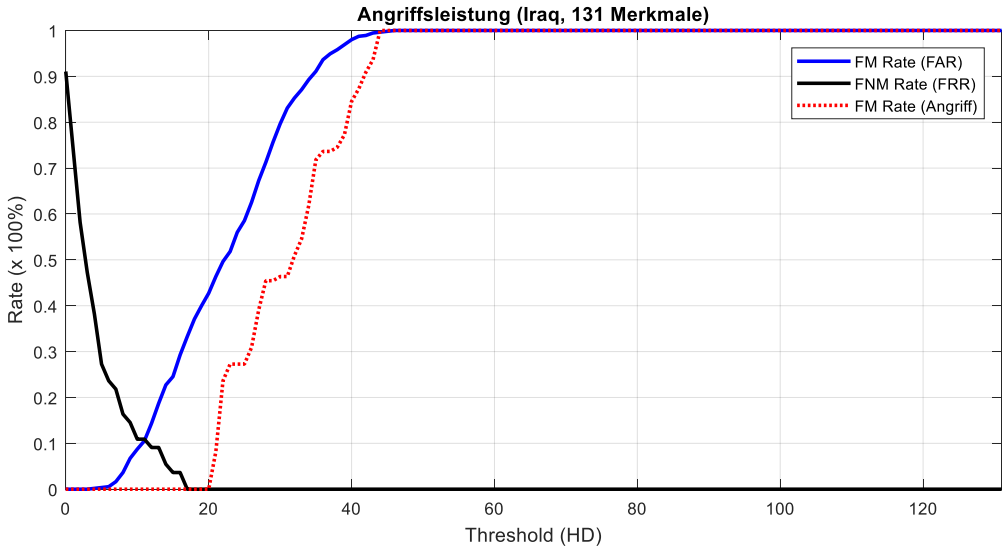


Abbildung 48 Angriffsperformanz für die Semantik Iraq (100 Angriffssamples)

Unter den getesteten Evaluationsparametern konnte keine hinreichende Angriffsperformanz erzielt werden. Es scheint, als besitzen die künstlichen Handschriftensignale nicht genügend Ähnlichkeit (Interklassen-Variabilität) mit originalen Handschriftensignalen. Gründe hierfür könnten u.a. sein, dass die aneinandergereihten Buchstaben nicht verbunden sind und somit mehr Stiftabsetzer generiert werden, als bei den originalen Handschriftendaten (Merkmal n_6). Des Weiteren kann die Schreibdauer für zusammengesetzte Buchstaben eines Wortes größer sein, als die Schreibdauer eines am Stück geschriebenen Wortes. Dies hat Auswirkungen auf verschiedene Merkmale z.B. gesamte Schreibdauer (n_1), Durchschnittsgeschwindigkeit (n_4 und n_5), maximale und minimale Geschwindigkeiten (n_7 - n_{10}), normalisierte Durchschnittsgeschwindigkeit (n_{26} und n_{27}), Dauer der Absetzpunkte (n_{28}) und Geschwindigkeit beim Wendepunkt (n_{130}).

Designvorschläge

Visuelle Designvorschläge

Künstlich erzeugte Handschriften wie reale Handschriften wirken zu lassen, insbesondere wenn der Schreibinhalt noch erkennbar sein soll, scheint nicht trivial zu sein. Verschärft wird das Vorhaben zusätzlich, wenn sich diese künstlichen Handschriften an einem Verifikationssystem ähnlich verhalten sollen (Fehlerraten) wie authentische Handschriften.

In dieser Arbeit konnten einige Punkte identifiziert werden, welche potentiell Auswirkung auf die Eigenschaften haben können und sollen an dieser Stelle als Designvorschläge zur Generierung künstlicher Handschriftendaten formuliert werden. Insbesondere bei der Verwendung von Mittelwertalphabeten zur Generierung von künstlichen Individuen, wie in dieser Arbeit, sollten diese Vorschläge berücksichtigt werden.

Bei der Aneinanderreihung von Buchstaben ist es von Vorteil, ein geeignetes Konzept für die Verbindung der Buchstaben untereinander zu haben. Solch eine Verbindung ist, wenn es um die Lesbarkeit des Schreibinhaltes geht, nicht unbedingt notwendig. Dadurch erhöht sich jedoch u.a. das äußere Erscheinungsbild. Buchstaben in Schreibschriftform aneinandergereiht ohne Verbindungslinien sehen unnatürlich aus. Bei Buchstaben in Druckschrift wiederum sind Verbindungslinien eher untypisch. Buchstaben sollten demnach, je nach Schreibtyp mit Linien verbunden werden.

Ein weiterer Punkt der in dieser Arbeit beobachtet werden konnte, ist die Auswirkung des korrekten Setzens der Stiftabsetzer. Buchstaben wirken unrealistisch, wenn Lücken auftreten wo i.d.R. ohne Abzusetzen geschrieben wird oder wenn bspw. der Querstrich der Sieben an untypischen Stellen beginnt bzw. endet. Bei der Erstellung eines Mittelwertalphabetes sollten die zeitlichen Positionen der Stiftabsetzer unbedingt berücksichtigt werden.

Ebenso sollten Schriftneigungen bei der Erstellung von Schreibindividuen eingesetzt werden. Diese lassen das Handschriftenbild natürlicher wirken, hier sollte der Schriftneigungswinkel jedoch nicht zu hoch gewählt werden.

Designvorschläge zur Anpassung der Fehlerraten

In dieser Arbeit wurden Modifikationsparameter *mpS* verwendet, um die Falschrückweisungsrate (FRR) zu steigern mit dem Ziel die Intra-Klassenvariabilität realer Handschriften nachzubilden. Hier hat sich u.a. gezeigt, dass die Fehlerraten sich tatsächlich realen Handschriften genähert haben, jedoch auch das optische Erscheinungsbild

negativ beeinflusst haben. Hier sollte der Fokus der Modifikation nicht allein auf das X- und Y- Signal des Schriftzugs gelegt werden, sondern auf Merkmale die das Erscheinungsbild nicht beeinflussen. Das Drucksignal, der Seitenwinkel oder der Höhenwinkel könnten hier beispielsweise verwendet werden. Zusätzlich können die Abstände zwischen den Buchstaben oder aber auch das Verhältnis zwischen Höhe und Breite eines Buchstabens variiert werden. Bei entsprechender Wahl, können die Veränderungen minimale Auswirkungen auf das Schreibbild haben, die Intraklassen-Ähnlichkeit jedoch beeinflussen.

Um die Falschakzeptanzrate (FAR) zu steigern, damit eine realistische Interklassen-Variabilität auftritt, wurden die Modifikationsparameter *mp* für die Generierung der Schreibindividuen entsprechend angepasst. Die Tests haben gezeigt, dass die FAR bzw. EER angestiegen ist, wenn die Modifikationsparameter ähnlich Werte besaßen. Das hatte jedoch zur Folge, dass die Erscheinungsbilder der Schreibindividuen relativ ähnlich waren. Eine Unterscheidung der erzeugten Schreibsignale war ohne weiteres nicht mehr möglich. In Abbildung 94 (siehe Anlage 8 im Anhang) ist dieser Prozess beispielhaft veranschaulicht. Hier ist klar zu erkennen, dass bei der 6. Iteration die Handschriften nahezu identisch wirken. Die Modifikationsparameter *mp* können eingesetzt werden, um die FAR zu erhöhen, jedoch sollten die Auswirkungen auf das Erscheinungsbild berücksichtigt werden.

7 Hill-Climbing-Verfahren (FA3)

Der Einsatz von Hill-Climbing-Verfahren als Angriffstechnik auf biometrische Verifikationsverfahren ist Bestandteil der Untersuchung in Forschungsaufgabe 3 (FA3) und soll hier betrachtet werden. Dafür werden im folgenden Abschnitt 7.1 der potentielle Einsatz von Hill-Climbing-Verfahren diskutiert und ein ausgewähltes HC-Verfahren vorgestellt. Im darauffolgenden Abschnitt 7.2 wird erläutert, wie die ausgewählte und bereits bekannte HC-Angriffstechnik auf den handschriftenbasierten Verifikationsalgorithmus von Vielhauer [Viel06] angewendet wird. Die experimentellen Tests dieses HC-Angriffs auf den Verifikationsalgorithmus werden in Abschnitt 7.3 präsentiert.

7.1 Vorgehensweise und Methodik

Wie bereits in Abschnitt 2.3.1 erläutert, sind Hill-Climbing- bzw. Bergsteiger- Algorithmen einfache heuristische Optimierungsverfahren. Sie gehören zu den lokalen Suchverfahren. Diese Art von Suchalgorithmen versucht, ein lokales Minimum oder lokales Maximum zu finden. Dabei wird, von einer gegebenen Startlösung ausgehend, solange in der Nachbarschaft "gewandert", bis der nächstbeste Punkt erreicht wird, siehe dazu u.a. [Schw77].
Anwendungsgebiete, in denen Hill-Climbing-Algorithmen zum Lösen von Optimierungsproblemen eingesetzt werden, sind unter anderem in der Wirtschaftsmathematik, Physik, Klimaforschung, Statistik (Data Mining) oder Spieltheorie zu finden. Hill-Climbing-Algorithmen können des Weiteren eingesetzt werden, um künstliche biometrische Daten zu optimieren.
In den folgenden Teilabschnitten werden die Voraussetzungen für die Umsetzung eines Hill-Climbing-Algorithmus als Angriffsmethode auf ein biometrisches Erkennungssystem beschrieben.

7.1.1 Hill-Climbing-Verfahren als Angriffstechnik auf biometrische Systeme

Ein Hill-Climbing-Algorithmus benötigt immer einen Vergleichswert („Güte Maß“), um festzustellen, wie gut das aktuell ermittelte Datum ist. Je nachdem, wie gut oder schlecht dieser Wert ist, wird der nächste Schritt (neues Datum) berechnet.
Betrachtet man den generellen Aufbau eines biometrischen Erkennungssystems in Bezug auf einen Angriff mittels HC-Algorithmus, können verschiedene Einspiel- bzw. Entnahmepunkte identifiziert werden. Wobei mit Entnahmepunkt der Ort gemeint ist, wo der „Güte Wert“ zur Verarbeitung des HC-Algorithmus entnommen wird. Der Einspielpunkt bezeichnet den Ort, wo die durch den HC-Algorithmus generierten Daten (Angriffsdaten) eingespielt werden, um z.B. einen unautorisierten Zugriff zu erlangen. In Abbildung 49 werden die potentiellen Entnahmepunkte (AP 2, AP 4, AP 7) und Einspielpunkte (AP 4, AP 7, AP 8) eines biometrischen Erkennungssystems dargestellt.

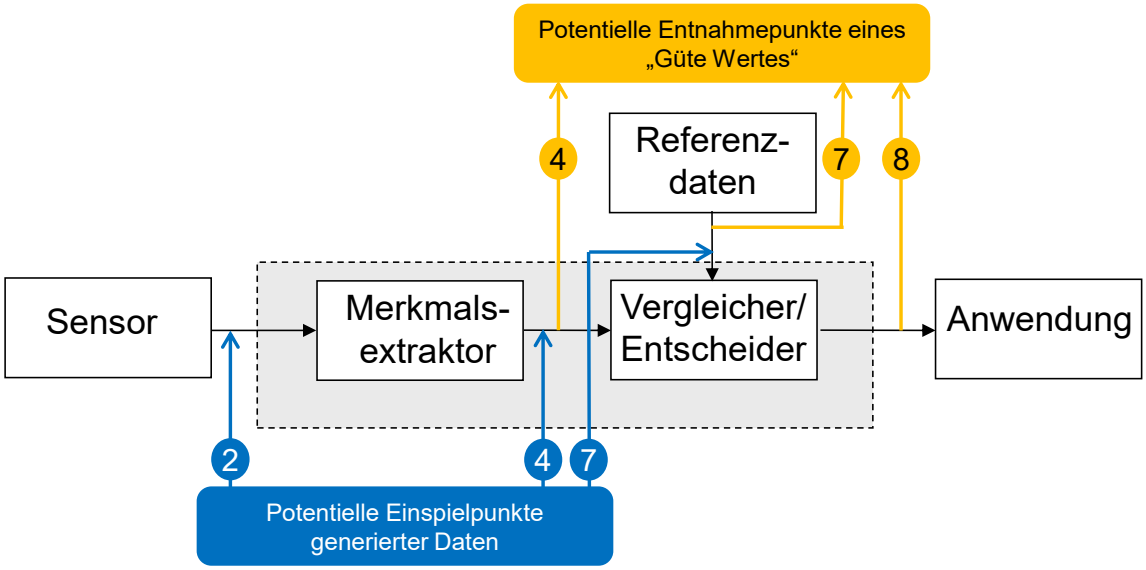


Abbildung 49 Potentielle Hill-Climbing Angriffspunkte auf ein biometrisches Erkennungssystem (Erweiterung des Modells nach [RaCB01]; vgl. Abbildung 1

Die am Angriffspunkt 4 (AP 4) entnommen Daten (z.B. Merkmalsvektor) können herangezogen werden, um mittels HC-Algorithmus Rohdaten zu generieren. Weiterhin können die am Angriffspunkt 7 (AP 7) entnommen Referenzdaten für eine Rohdatengenerierung verwendet werden. Dabei müssen die an AP 4 und AP 7 entnommenen Daten nicht unbedingt im gleichen Format vorliegen. Template Protection Mechanismen haben beispielsweise die Referenzdaten in verschlüsselter oder transformierter Form gebracht. Die am Angriffspunkt 8 (AP 8) entnommen Daten können beispielsweise in Form eines Vergleichswertes (Matching-Score) oder einer Ja/Nein- Entscheidung vorliegen.
Die vom HC-Algorithmus generierten Daten können am Angriffspunkt 2 (AP 2) in Form von Rohdaten, am AP 4 in Form von z.B. Merkmalsvektoren und am AP 7 in Form von Referenzdaten eingebracht werden. Prinzipiell ist das Einspielen von generierten Daten am AP 8 auch möglich. Dies soll jedoch nicht betrachtet werden, da hier lediglich das Umgehen des gesamten biometrischen Erkennungssystems beschrieben wird.
Im nachfolgenden Abschnitt wird erläutert, an welchen Punkten ein Hill-Climbing-Algorithmus auf den BioHash-Algorithmus (siehe Abschnitt 4.2) angesetzt werden kann.

7.1.2 Biometrischer Hash Algorithmus und Hill-Climbing-Angriffe

Angriffsdaten aus einem Hill-Climbing-Angriff können prinzipiell an vier Punkten beim biometrischen Hash Algorithmus eingespielt werden (siehe Abbildung 50):

(1) hinter dem Sensor
(2) nach der Merkmalsextraktion
(3) nach der BioHash-Generierung und
(4) hinter dem Vergleicher

Am Punkt (1) können Handschriftenrohdaten, am Punkt (2) Merkmalsvektoren, am Punkt (3) BioHash-Vektoren und am Punkt (4) Vergleichswerte in das System eingespielt werden.

Der für einen Hill-Climbing-Algorithmus notwendige Güte Wert, welcher angibt, inwieweit das aktuelle Datum für den weiteren Verlauf geeignet oder nicht geeignet ist, kann an folgenden Punkten des BioHash-Algorithmus entnommen werden:

(a) nach dem Entscheider (Ja/Nein bzw. True/False)
(b) hinter dem Vergleicher (Vergleichswert, z.B. Hamming-Distanz)
(c) nach der BioHash-Generierung (BioHash-Vektor)
(d) hinter der Merkmalsextraktion (Merkmalsvektor)

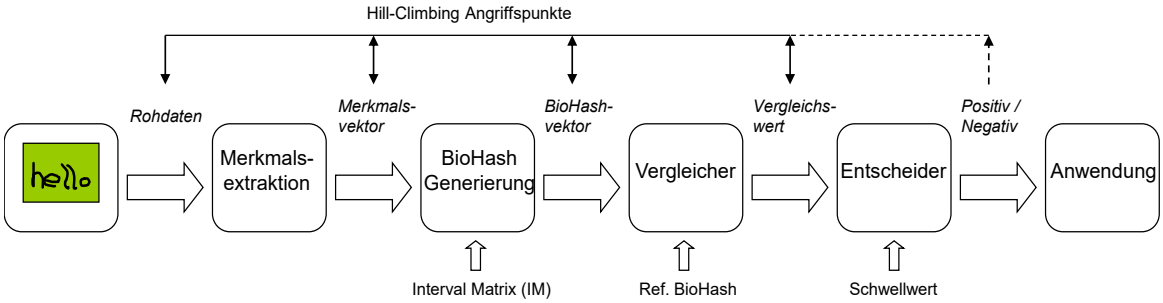


Abbildung 50 Potentielle Hill-Climbing-Angriffspunkte auf den Biometrischen Hash Algorithmus übersetzt und adaptiert von [KVS+10]

Für die Punkte (a) und (b) sind die Entnahmepunkte für einen Hill-Climbing-Algorithmus leicht nachzuvollziehen. Ein potentieller Angreifer kann jedoch auch einen Merkmalsvektor (Punkt (d)) verwenden, um gezielt mittels Hill-Climbing-Algorithmus einen bestimmten Rohdatensatz zu finden. Weiterhin ist es möglich, einen bestimmten Merkmalsvektor auf Basis eines BioHash-Vektors (Punkt (c)) mittels HC- Algorithmus zu bestimmen. In Tabelle 40 werden alle möglichen Kombinationen (mit 'x' gekennzeichnet) aus Entnahmepunkt und generierbaren Daten (Rohdaten, Merkmalsvektor, etc.) dargestellt.

Tabelle 40 Entnahmepunkte und mit HC-Algorithmus mögliche generierbare Datensätze

	Entnahmepunkte des „Güte Wertes“ am BioHash-Algorithmus			
Generierbare Datensätze	Merkmalsvektor	BioHash-Vektor	Vergleichswert	Ja/Nein
Rohdaten	x	x	x	x
Merkmalsvektor		x	x	x
BioHash-Vektor			x	x
Vergleichswert				x

Die Chancen, potentiell gute Datensätze mittels HC-Algorithmus zu generieren, hängen stark vom „Güte Wert“ ab. Rohdaten auf Basis einer Ja/Nein Antwort des Entscheiders zu generieren ist wesentlich schwieriger, als auf Basis eines Merkmalsvektors, welcher viel präzisere Informationen über einen Rohdatenschriftzug enthält. Je mehr Information ein „Güte Wert“ enthält, desto größer sind die Erfolgschancen. Des Weiteren ist für die Erfolgschance die Anzahl der Verarbeitungsstufen zwischen Entnahmepunkt des „Güte Werts“ und den zu generierenden Daten entscheidend. Je mehr Verarbeitungsschritte und ggf. Transformationen der Daten innerhalb dieser Prozesse stattfinden, desto schwieriger die Umsetzung des HC- Algorithmus. Bezogen auf Tabelle 40 kann die Aussage getroffen werden, dass die Erfolgschancen steigen, je näher der Entnahmepunkt an den zu generierenden Punkt der Daten steht.

7.1.3 Bayesian Hill-Climbing-Angriff

Das in Abschnitt 2.3.3 kurz vorgestellte direkte Angriffsverfahren von Galbally et al. [GaFO07] auf einen handschriftenbasiertes Verifikationsalgorithmus soll hier näher beschrieben werden. Ziel ist es das Verfahren auf den in Abschnitt 4.2 vorgestellten handschriftenbasierten Verifikationssystem anzuwenden und zu evaluieren. Das Angriffsverfahren von Galbally et al. wird gewählt, weil es ebenfalls für ein handschriftenbasierendes Verifikationsverfahren angewendet wird. So können die experimentellen Ergebnisse besser miteinander verglichen werden.
Um sich die Motivation und Arbeitsweise des eingesetzten Algorithmus zu verdeutlichen, wird folgendes Szenario vorausgesetzt. Ein Angreifer ist in der Lage, einen Merkmalsvektor in das System einzuspielen (siehe Abbildung 49 Angriffspunkt 4). Zusätzlich hat er in diesem Szenario Zugriff auf den Vergleichswert (Abstandsmaß) des aktuellen Merkmalsvektors zum Referenzmerkmalsvektor. Des Weiteren besitzt der Angreifer eine Menge von Merkmalsvektoren verschiedener Personen, die mit dem Merkmalsextraktor des anzugreifenden biometrischen Verifikationssystems erstellt wurden. Mit Hilfe des Hill-Climbing-Algorithmus sollen nun Merkmalsvektoren erstellt, in das System eingespielt und mit den Referenzmerkmalsvektor verglichen werden. Der Vergleichswert wird als Richtwert für den Hill-Climbing-Algorithmus verwendet. Die nachfolgende Beschreibung des Algorithmus beruht auf dem von Galbally et al. in [GaFO07] beschrieben Algorithmus.
Betrachtet wird das Problem, einen K- dimensionalen Vektor *y* zu finden, der im Vergleich zu einem unbekannten Vektor *C* einen Vergleichswert liefert, welcher einen bestimmten Schwellenwert δ, unter Verwendung der Vergleichsfunktion *J*, überschreitet. Wobei die Vergleichsfunktion *J* wie folgt definiert werden kann:

$$J(C, y) > \delta \qquad \textbf{Formel 26}$$

Der unbekannte Vektor *C* kann ein *K*-dimensionaler Vektor oder ein generatives Modell von *K*-dimensionalen Vektoren sein. Ferner sei *G* ein statistisches Modell, welches eine mehrdimensionale bzw. multivariate Normalverteilung darstellt. Wobei *G* aus dem Erwartungsvektor (Mittelwertvektor) μ_G und der diagonalen Kovarianzmatrix $\sum_G$ besteht, mit

$$\sigma_G^2 = diag(\Sigma_G). \qquad \textbf{Formel 27}$$

In dem oben eingeführten Szenario besitzt der Angreifer Zugang zur Vergleichsfunktion (siehe Formel 26), um den K-dimensionalen Vektor y mit dem unbekannten Template Vektor C zu vergleichen. Weiterhin wird davon ausgegangen, dass G mittels einer Datenbasis erzeugt wurde, welche in einigen Fällen mit dem unbekannten Vektor C übereinstimmt. G steht demnach dem Angreifer zur Verfügung. Um das Problem zu lösen, wird die globale mehrdimensionale Normalverteilung *G* in einer Weise adaptiert, dass lokale Besonderheiten des unbekannten Vektors C mittels folgender iterativer Strategie berücksichtigt werden:

1. Nimm *N* Muster (Merkmalsvektoren) y_i der globalen mehrdimensionalen Normalverteilung *G* und berechne den Vergleichswert $J(C, y_i)$, mit $i = 1, \ldots, N$.
2. Wähle *M* Muster (mit $M < N$) mit dem höchsten Vergleichswert (Ähnlichkeitswert).
3. Berechne die lokale Normalverteilung $L(\mu_L, \sigma_L)$ basierend auf den M gewählten Mustern.

4. Berechne eine adaptierte Normalverteilung A (μ_A, σ_A), welche eine Kompromiss zu der globalen Normalverteilung *G(μ_G, σ_G)* und der lokalen Normalverteilung von *L(μ_L, σ_L)* darstellt. Die Berechnung wird durchgeführt, indem die hinreichenden Statistiken wie folgt adaptiert werden:

$$\mu_A = \alpha\mu_L + (1-\alpha)\mu_G \qquad \textbf{Formel 28}$$

$$\sigma_A^2 = \alpha(\sigma_L^2 + \mu_L^2) + (1-\alpha)(\sigma_G^2 + \mu_G^2) - \mu_A^2 \qquad \textbf{Formel 29}$$

5. Definiere *G* neu mit *G = A* und fahre fort mit Schritt 1.

In Formel 28 und Formel 29 ist μ^2 wie folgt definiert: $\mu^2 = diag(\mu\mu^T)$. Weiterhin ist α ein Anpassungskoeffizient im Bereich von [0, 1]. Der Algorithmus endet entweder, wenn ein Muster *N* einen Vergleichswert besitzt der über den Schwellenwert δ liegt, oder die maximale Anzahl der vorgegebenen Iterationen erreicht ist.

Betrachtet man die Arbeitsweise des von Galbally et al. in [GaFO07] beschrieben Hill-Climbing-Algorithmus, können Analogien zu evolutionären Algorithmen festgestellt werden. So existiert mit *G* eine Startmenge von Individuen (Start Population) aus denen die „Besten" bzw. „Fittesten" Individuen separiert werden, siehe Schritt 2. Anschließend bilden diese „Besten" Individuen eine neue Population und die Prozedur beginnt wieder von vorne (Schritt 3 bis 5).

Im Abschnitt 7.2 wird beschrieben, wie der von Galbally et al. vorgestellte HC-Algorithmus angepasst bzw. adaptiert wird, um einen Angriff auf den im Abschnitt 4.2 beschrieben handschriftenbasierten Verifikationsalgorithmus durchzuführen und zu evaluieren.

Zusammenfassung
In diesem Abschnitt wurde gezeigt, wie ein Hill-Climbing-Algorithmus prinzipiell arbeitet. Es wurde gezeigt, an welchen der acht Angriffspunkte eines biometrischen Verifikationssystems die benötigten Daten hierfür entnommen bzw. eingespielt werden können. Anschließend ist dargestellt worden, welche Punkte dies speziell beim handschriftenbasierten Verifikationssystem [Viel06] sind und welche Konstellationen (Entnahme-/Einspielpunkt) möglich sind. Letztendlich ist ein Hill-Climbing-Algorithmus aus der Literatur [GaFO07] detailliert vorgestellt worden, welcher auf den handschriftenbasierten Verifikationsalgorithmus angewendet werden soll.

7.2 Durchführung

Im Abschnitt 7.1.3 wurde die Arbeitsweise des von Galbally et al. in [GaFO07] vorgestellten HC-Algorithmus erläutert. Im Folgenden wird beschrieben, wie dieser HC-Algorithmus auf den handschriftenbasierten Verifikationsalgorithmus angewendet wird, um einen Angriff mittels Hill-Climbing-Verfahren durchzuführen.

In Abbildung 51 sind der Entnahmepunkt des Güte Wertes sowie der Einspeisepunkt für die generierten Daten des HC-Algorithmus am Prozessablauf des handschriftenbasierten

Verifikationsalgorithmus abgebildet. Am Entnahmepunkt wird der Vergleichswert ausgelesen. Dieser Wert ist das Resultat eines Vergleiches zwischen dem Referenz-BioHash B_{Ref}, welcher im System hinterlegt ist, sowie dem aktuell präsentierten BioHash B_{Akt} (siehe Abschnitt 4.2.2). Der Vergleich wird mittels Berechnung der Hamming-Distanz (siehe Abschnitt 4.2.3) durchgeführt (an dieser Stelle sei erwähnt, dass auch eine andere Vergleichsfunktion z.B. Canberra oder Euklidische Distanz verwendet werden kann). Der aktuell präsentierte BioHash B_{Akt} wird generiert mittels hinterlegter Interval Matrix (*IM*) und dem aktuellen Merkmalsvektor, welcher in diesem Fall vom HC-Algorithmus erzeugt wird. Der Einspeisepunkt liegt, wie in Abbildung 51 dargestellt, vor der BioHash-Generierung. An diesem Punkt werden die vom HC-Algorithmus generierten Merkmalsvektoren eingespielt. Anschließend wird der BioHash generiert und der Vergleichswert erzeugt. Dieser wird vom HC-ausgelesen und als Güte Wert verarbeitet.

Der verwendete Hill-Climbing-Algorithmus benötigt in diesem Szenario eine Startmenge von Merkmalsvektoren, um zu arbeiten. Wie in Abschnitt 7.1.3 beschrieben, wird die Gauß-Verteilung oder auch „Gaußsche Normalverteilung“ von einer Menge von Merkmalsvektoren bestimmt. Diese Merkmalsvektoren wurden auf Basis realer Handschriftendaten von Personen erzeugt. So wird sichergestellt, dass die potentiell möglichen Werte innerhalb eines Merkmalsvektors nicht unrealistische Werte annehmen, wie sie bspw. bei per Zufall generierten Werten auftreten könnten. Es ist potentiell möglich, dass die Startmenge von Merkmalsvektoren keine geeigneten Merkmalsvektoren enthält, um zum Ziel zu kommen. Also einen BioHash zu generieren der den Referenz-BioHash gleicht bzw. sehr ähnelt und somit aus Sicht des Angreifers zu einer positiven Entscheidung (Verifikation) führt.
In der späteren Evaluation (siehe Abschnitt 7.3.2) sind die Personen, auf Basis derer die Startmenge gebildet wird, ungleich der Personen die mittels HC-Verfahren angegriffen werden.

In dieser Forschungsaufgabe (FA3) wurde sich für die Generierung von Merkmalsvektoren entschieden, welche vom HC-Algorithmus generiert werden sollen. Nachfolgend werden die Gründe für diese Auswahl benannt:

- Der Abstand zwischen Abnahmepunkt des Güte Wertes und Einspeisepunkt der generierten Daten sollte nicht zu nahe beieinander liegen, um das Problem nicht zu einfach zu gestalten, da hier mindestens ein Verarbeitungsschritt (BioHash-Generierung) und ein Vergleichsschritt (Vergleicher) dazwischen liegen.
- Es ist einfacher, die hier erlangten Ergebnisse mit denen von Galbally et al. in [GaFO07] zu vergleichen, da in der Arbeit von Galbally et al. ebenfalls Merkmalsvektoren vom HC-Algorithmus generiert werden.

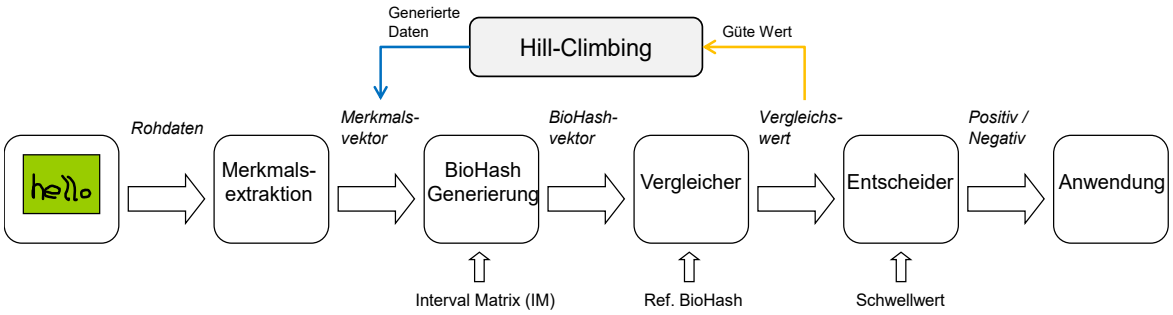


Abbildung 51 Entnahmepunkt für Güte Wert und Einspeisepunkt für generierte Daten des HC-Algorithmus übersetzt und adaptiert von [KVS+10]

Ziel ist es zu bestimmen, wie viele der durch den HC-Algorithmus generierten Merkmalsvektoren eine positive Verifikationsentscheidung hervorrufen würden, bei der Verwendung eines bestimmten Schwellenwerts (siehe Evaluationsaufbau und Messmethodik im Abschnitt 7.3.1).

7.3 Experimentelle Tests

Um die Performanz des HC-Verfahrens bestimmen zu können werden in diesem Abschnitt die Messmethodik und der Aufbau der Evaluation beschrieben. Dabei werden im Abschnitt 7.3.1 die Rahmenbedingungen und verwendeten Daten beschrieben und im Abschnitt 7.3.2 die entsprechenden Ergebnisse präsentiert und bewertet.

7.3.1 Messmethodik und Evaluationsaufbau

In diesem experimentellen Test wird folgendes Angriffsszenario angenommen.
Ein Angreifer möchte sich unbefugten Zugang zu einem geschützten System verschaffen. Die Zugangskontrolle zu diesem System wird mit Hilfe eines biometrischen Verifikationssystems sichergestellt. Das biometrische System arbeitet auf Basis der Modalität *dynamische Handschrift* und verwendet den Verifikationsalgorithmus nach [Viel06].
Ferner hat der Angreifer eingeschränkten Zugang zum Verifikationssystem und ist in der Lage, Daten (Merkmalsvektoren) vor den Prozess der BioHash-Generierung einzuspielen sowie Daten (Vergleichswert) nach dem Vergleichsprozess zu entnehmen.
Der Angreifer möchte sich Zugang zum System verschaffen, indem er einen Merkmalsvektor erzeugt, der eine positive Verifikation am System erzeugt. Hierfür verwendet der Angreifer einen Hill-Climbing-Algorithmus (HC-Algorithmus) nach Galbally et al. in [GaFO07], um diesen Merkmalsvektor zu ermitteln. In Abbildung 51 sind Entnahmepunkt und Einspielpunkt der Daten am Prozessablauf des BioHash-Algorithmus nach [Viel06] dargestellt. Zusätzlich besitzt der Angreifer Handschriftendaten von beliebigen realen Personen, wobei diese ungleich der Personen sind, welcher er angreifen möchte. Aus diesen Daten bildet der Angreifer u.a. die Startmenge des HC-Algorithmus.
Ziel soll es sein, zu bestimmen, inwieweit der Angriff erfolgreich für den Angreifer verläuft. Die Ergebnisse werden mit den Ergebnissen von Galbally et al. in [GaFO07] verglichen. Anschließend soll bewertet werden, ob die Adaption des Verfahrens erfolgreich war.

Für die Evaluation des adaptierten Hill-Climbing-Angriffs wurde eine Handschriftendatenbank verwendet, welche Handschriftendaten von 53 Personen beinhaltet. Jede Person hat zehn Handschriftensamples pro Semantik zu zwei verschieden Zeitpunkten (Sessions) aufgezeichnet Dabei wurden insgesamt fünf verschiedene Semantiken verwendet. Der

zeitliche Abstand zwischen der ersten und zweiten Session betrug 30 Tage. Semantiken bezeichnen verschiedene, teils vorgegebene Schreibinhalte. Somit stehen für diese Evaluation 5300 Handschriftensamples zur Verfügung. Folgende fünf Schreibsemantiken wurden bei der Aufzeichnung der Daten verwendet (siehe auch die Beschreibung der Semantiken im Abschnitt 5.3.1):

- gegebene PIN (77993)
- geheime PIN (PIN)
- Pseudonym
- Symbol
- Woher

Da der beschriebene HC-Algorithmus Merkmalsvektoren von realen Handschriften benötigt, um eine Startmenge auf Basis der globalen Normalverteilung zu ermitteln, werden die Testdaten unterteilt. Ein Teil der Testdaten repräsentiert die Referenzdaten eines Verifikationssystems und der andere Teil der Testdaten wird für den HC-Algorithmus verwendet, um die Angriffsdaten zu erzeugen. Anschließend werden die Testdaten genau andersherum verwendet (zweifache Kreuzvalidierung). Die Testdaten werden anhand der Identifikationsnummern (Personen-ID) der Testdatenbank eingeteilt. Ungerade Personen-IDs repräsentieren die eine Gruppe und gerade Personen-IDs die andere Testgruppe. Diese Form der Gruppeneinteilung wurde gewählt, da sie sich leicht umsetzen lässt. Hier wären auch andere Strategien denkbar, wie beispielsweise eine zufällige Wahl der Nutzer für die entsprechenden Testgruppen.
Zu Beginn der Evaluation werden die Toleranzfaktoren der jeweiligen Semantikklassen und entsprechende Kreuzvalidierung (gerade/ungerade Personen-IDs) separat bestimmt, um während der Verifikation die Verifikationsperformanz des Systems zu erhöhen (siehe Abschnitt 5.3.1).
Der Hill-Climbing-Algorithmus benötigt verschiedene Startwerte und Parameter, welche zum Teil mittels experimenteller Versuche ermittelt werden. Für einen besseren Vergleich der Ergebnisse mit denen aus der Arbeit von Galbally et al. in [GaFO07], werden die identischen experimentellen Untersuchungen für die Parameterbestimmung durchgeführt. Die Abbruchbedingungen (Anzahl der Iterationen und Schwellenwert) für den HC-Algorithmus werden entsprechend übernommen. Der Schwellenwert legt fest, ab wann ein Angriff als erfolgreich angesehen wird, also der maximale Unterschied der zwischen den Referenz Merkmalsvektor und des vom HC-Algorithmus generierten Merkmalsvektors auftreten darf. In diesen Versuchen wird der Unterschied der Merkmalsvektoren mittels Hamming-Distanz (HD) bestimmt. Die Schwellenwerte (HD-Werte) werden anhand der Falschrückweisungsrate der Testdaten bestimmt. Hierfür werden die Daten der ersten Session verwendet, um die Referenzdaten (BioHash und Intervallmatrix) zu erzeugen und die Handschriftendaten der zweiten Session zur Verifikation eingesetzt. Nun werden, genau wie in [GaFO07], die Schwellenwerte (Thresholds) bei einer Falschrückweisungsrate von FRR = 20%, 30% und 40% abgelesen und als eines der Abbruchkriterien festgelegt.
Ein weiteres Abbruchkriterium des HC-Algorithmus stellt die Anzahl der zulässigen Iterationen dar. Diese soll an die Größe der Startmenge *N* gekoppelt werden, um die gesamte Anzahl der potentiellen Vergleiche zwischen dem Referenz-Merkmalsvektor und dem HC generierten Merkmalsvektor bei unterschiedlichen Startmengen gleich zu halten. Galbally et al. wählen diesen Weg, um einen besseren Vergleich zu einem potentiellen Brute-

Force-Verfahren herzustellen, welcher nach einer gewissen Anzahl zu einem möglichen positiven Angriff führt.
Die Startmenge *N* und die Anzahl der besten Samples *M*, welche für die Optimierung der nächsten Stufe gewählt werden, richteten sich bis auf eine kleine Änderung nach denen von Galbally. Die Änderung beinhaltet die zusätzliche Einführung der Startmenge *N* mit 300. In Tabelle 41 werden alle *N* und *M* Werte angegeben, anhand derer die Erfolgsraten für einen positiven Angriff und die dazu benötigten durchschnittlichen Iterationen bestimmt werden sollen. Unter der Anzahl der Startmenge *N* ist in Klammern die maximal erlaubte Anzahl an Iterationen pro Versuch angegeben. Bei diesen Tests sind der Parameter α = 0.5 und die Falschrückweisungsrate FRR = 30% entsprechend fixiert. Da M eine Teilmenge von N ist (die besten/fittesten Individuen von N), können bestimmte Konstellationen entsprechend nicht bestimmt werden. Diese sind in der Tabelle 41 entsprechend dunkel hinterlegt.
Die Parameter *N* und *M* werden entsprechend der höchsten Erfolgsrate und der niedrigsten durchschnittlichen Anzahl an Versuchen gewählt.

Tabelle 41 Einstellung von N und M, bei denen die Erfolgsrate/Iterationen bestimmt werden soll

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3						
	5						
	10						
	25						
	50						
	100						

Ein weiterer HC-Parameter, welcher experimentell ermittelt werden soll, ist der Parameter α. Der Wert α bestimmt, inwieweit die globale Normalverteilung G bzw. die lokale Normalverteilung L Einfluss auf die Bestimmung der Samples (Merkmalsvektoren) für die nächste Stufe haben. Ist der Wert von α = 0, wird nur die globale Normalverteilung G berücksichtig. Wird hingegen der Wert von α = 1 gesetzt, beeinflusst ausschließlich die lokale Normalverteilung L die Berechnung der Samples (Merkmalsvektoren) für die nächste Stufe, siehe dazu auch Formel 28 und Formel 29. Der Parameter α kann demnach für die Optimierung des HC-Algorithmus verwendet werden und einen Wert zwischen 0 und 1 annehmen [0,1].
Der Parameter α soll entsprechend so gewählt werden, dass die Erfolgsrate des Angriffs auf alle angegriffenen Merkmalswerte erhöht wird und gleichzeitig so wenig Versuche (Iterationen) wie möglich dabei durchgeführt werden müssen. Für die Ermittlung eines geeigneten α-Wertes, wird dieser beginnend bei null in einem Versuch schrittweise um 0,1 erhöht, bis er den Wert eins angenommen hat. Für die Optimierung sind *N* (Anzahl der zu generierenden Merkmalsvektoren auf Basis der globalen Normalverteilung) und *M* (Anzahl der besten Merkmalsvektoren die in die nächste Stufe mitgenommen werden) konstant bei den jeweiligen bestimmten optimalen M, N Wertepaaren (siehe Tabelle 41).

Für einen besseren Vergleich mit der von Galbally et al. in [GaFO07] vorgestellten Methode werden die gleichen Abbruchbedingungen für den HC-Algorithmus gewählt. Hierfür

werden die Schwellenwerte (Threshold) bei den Falschrückweisungsraten FRR = 20%, 30% und 40% bestimmt. Dies wird jeweils für die ungeraden und geraden Personen-IDs und den entsprechenden Semantikklassen durchgeführt. In Abbildung 52 wird ein typischer Verlauf einer Falschakzeptanzrate sowie Falschrückweisungsrate dargestellt, anhand der FRR werden die zugehörigen Schwellenwerte für die Arbeitsmodi FRR=20%, FRR=30% und FRR=40% entsprechend abgelesen.

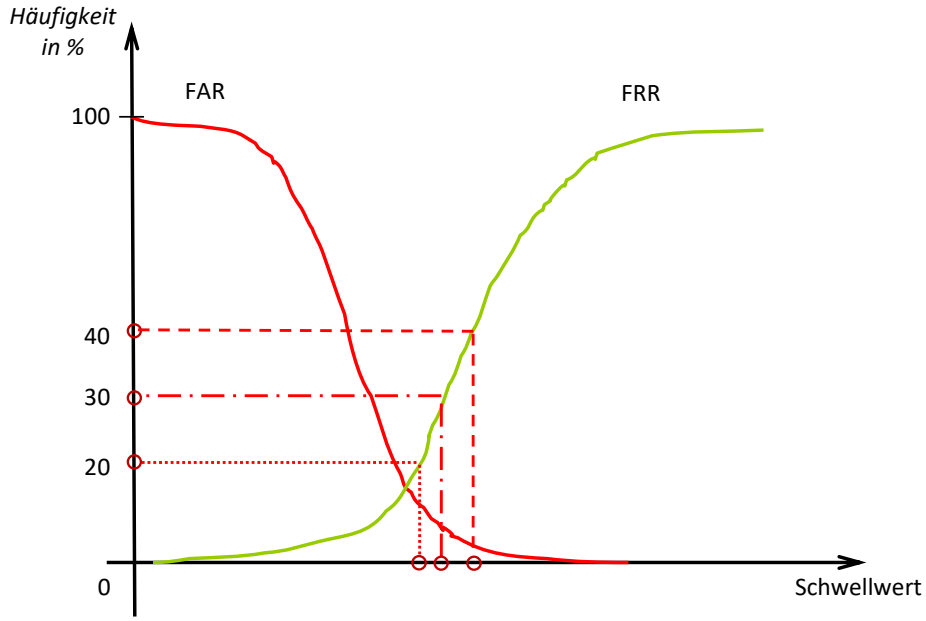


Abbildung 52 Schwellenwertbestimmung anhand der Falschrückweisungsrate (FRR) nach [GaFO07]

Im letzten Schritt werden alle Erfolgsraten für die jeweiligen Arbeitsbereiche (FRR=20%, FRR=30% und FRR=40) für alle Semantikklassen und geraden bzw. ungeraden Personen-IDs berechnet und mit denen von Galbally et al .aus [GaFO07] verglichen.

7.3.2 Präsentation und Bewertung der Ergebnisse

Im ersten Schritt der experimentellen Untersuchungen wurden die jeweiligen Toleranzfaktoren für die entsprechende niedrigste EER bestimmt. In Abbildung 53 wird das Ergebnis Beispielhaft an der Semantikklasse Pseudonym für alle ungeraden Personen-IDs gezeigt. Der Toleranzfaktor wird an der Stelle abgelesen, wo die EER am geringsten ist. Im Beispiel der Semantikklasse Pseudonym (Abbildung 53) kann entsprechend der Toleranzfaktor mit den Wert "1" für die geringste EER abgelesen werden. Eine vollständige Auflistung aller Diagramme zur Bestimmung der jeweiligen Toleranzfaktoren befindet sich im Anhang Anlage 14 dieser Arbeit.

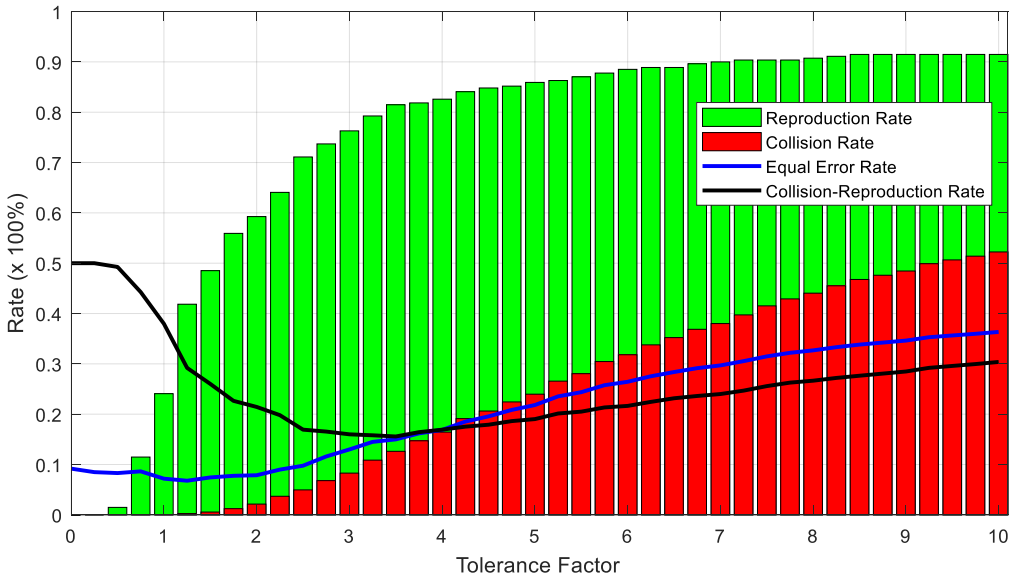


Abbildung 53 Bestimmung des Toleranzfaktors der Semantik Pseudonym aller ung. Personen-IDs

Neben den Toleranzfaktoren werden auch die Schwellenwerte für die Arbeitsmodi FRR=20%, FRR=30% und FRR=40% bestimmt. Hierfür werden die entsprechenden FRR berechnet und der jeweilige Schwellenwert abgelesen. In Abbildung 54 wird die Verifikationsleistung des Systems anhand der ungeraden Personen-IDs für die Semantik *„Symbol“* gezeigt. Anhand des Verlaufs der Falschrückweisungsrate (FRR) werden an den Punkten (FRR=20, FRR=30 und FRR=40) die jeweiligen Schwellenwerte (Threshold) auf der X-Achse abgelesen. In diesem Beispiel betragen die Schwellenwerte für die jeweiligen Falschrückweisungsraten $th_{20} = 13$, $th_{30} = 11$ und $th_{40} = 8$. Alle übrigen Diagramme zur Bestimmung der jeweiligen Schwellenwerte sind im Anhang Anlage 13 zu finden.

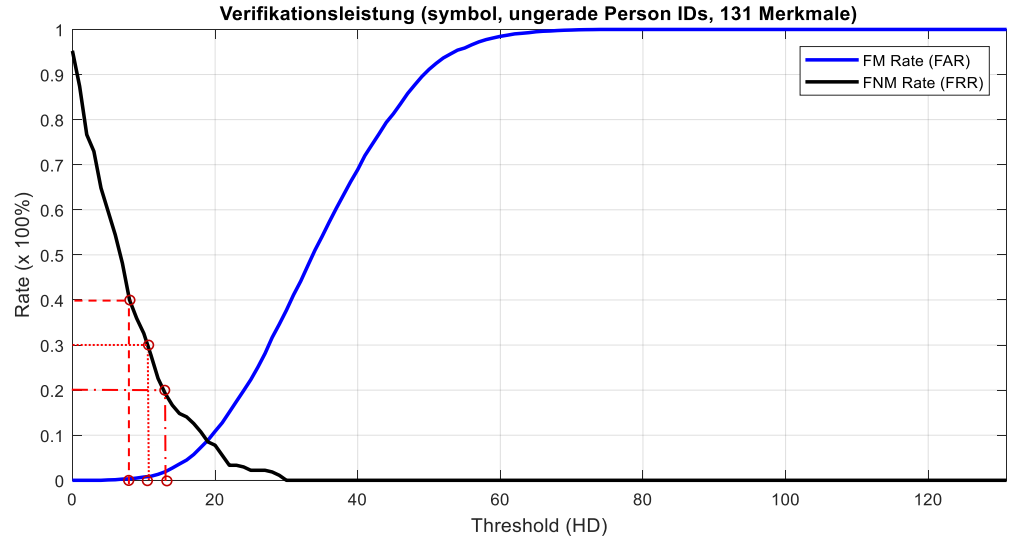


Abbildung 54 Verifikationsleistung für alle ung. Personen-IDs der Semantikklasse Symbol

In Tabelle 42 und Tabelle 43 sind die Ergebnisse der Schwellenwert- und Toleranzfaktorbestimmung für die geraden bzw. ungeraden Personen-IDs sowie entsprechender Semantikklasse aufgelistet.

Tabelle 42 Ermittelte Schwellenwerte und Toleranzfaktoren für alle ungeraden Personen-IDs

Semantikklasse	Schwellenwert (th_{FRR})für die jeweiligen Falschrückweisungsraten (20%, 30% und 40%)			Toleranzfaktor
	th_{20}	th_{30}	th_{40}	
77993	14	12	8	0.5
Pin	12	10	8	0.5
Pseudonym	5	3	2	1
Symbol	13	11	8	0.5
Woher	11	8	7	0.5

Nach der Bestimmung der Toleranzfaktoren und Schwellenwerte, werden die Werte M und N bestimmt. Hierfür wurden die Erfolgsraten und die Anzahl der durchschnittlichen Versuche für die jeweiligen N, M Wertepaare ermittelt. In Tabelle 41 sind die N, M Konstellationen vorgegeben, zu denen die entsprechenden Erfolgsraten inkl. Iterationen (Versuche) ermittelt werden sollen.

Tabelle 43 Ermittelte Schwellenwerte und Toleranzfaktoren für alle geraden Personen-IDs

Semantikklasse	Schwellenwert (th_{FRR})für die jeweiligen Falschrückweisungsraten (20%, 30% und 40%)			Toleranzfaktor
	th_{20}	th_{30}	th_{40}	
77993	12	9	7	0.5
Pin	10	8	6	0.5
Pseudonym	13	11	9	0.5
Symbol	8	5	3	0.75
Woher	23	20	17	0.25

In Tabelle 44 sind die Ergebnisse für die Semantik 77993 und alle geraden Personen-IDs beispielhaft aufgelistet. Hier wurden die Daten der ungeraden Personen-IDs als Basis für den HC-Algorithmus verwendet und die Testdaten der geraden Personen-IDs angegriffen.

Tabelle 44 Erfolgsrate und Anzahl an Iterationen der Semantik 77993 (geraden Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	7,69 % (241)	30,77 % (234,63)	11,54 % (75,67)	23,08 % (67,67)	23,08 % (65,83)
	5	0	53,84 % (534,21)	42,41 % (200,91)	61,54 % (135,81)	46,15 % (75,17)	42,31 % (58,45)
	10		34,62 % (342,22)	65,38 % (205,59)	65,38 % (104,06)	80,77 % (76,71)	57,69 % (56,4)
	25			69,23 % (259,44)	84,62 % (125,05)	**84,62 % (75,32)**	73,08 % (58,37)
	50				61,54 % (153,88)	61,54 % (86,19)	53,85 % (63,36)
	100					26,92 % (100,14)	23,08 % (67)

Unter der Erfolgsrate ist in Klammern die Anzahl der durchschnittlichen Versuche dargestellt. Erfolgsrate (success rate - SR) in diesem Zusammenhang beschreibt die Falschakzeptanzrate (FAR). Also die Anzahl der durch den Angreifer (HC-Algorithmus) verursachten Falschakzeptanzen bezogen auf die Referenzdaten. Im Anhang Anlage 15 werden alle Ergebnistabellen für die N, M-Wertepaarbestimmung dargestellt. Das N, M-Wertepaar, welches die höchste Erfolgsrate bei geringster Anzahl an versuchen erzielt, wird für die weiteren Tests (α-Bestimmung) verwendet. Das gewählte Wertepaar ist in den Tabellen jeweils farblich markiert (siehe beispielhaft Tabelle 44).

Wie oben bereits beschrieben, werden die M- und N-Werte, welche die höchste Erfolgsrate generieren und die niedrigste durchschnittliche Anzahl von Versuchen benötigten, für die Bestimmung des optimierten α-Wertes ausgewählt. Für die Semantikklasse Symbol (gerade Personen-IDs) als auch für die Semantikklasse Pseudonym (ungeraden Person IDs) konnte kein positiver Angriff festgestellt und demnach auch keine Erfolgsrate berechnet werden. Für diese beiden Semantiken wurden die Wertepaare der jeweils anderen Personen-ID-Gruppe gewählt, siehe Tabelle 45 (kursiv markiert). Damit soll geprüft werden, ob es möglich ist, noch einen positiven Angriff zu erzielen.

Tabelle 45 Ausgewählte [N,M] Wertepaare für die α Bestimmung

Semantik	**Gerade Personen-IDs** [N,M]	**Ungerade Person IDs** [N,M]
77993	[200,25]	[50,10]
PIN	[100,10]	[300,10]
Pseudonym	[300,10]	*[300,10]*
Symbol	*[200,10]*	[200,10]
Woher	[200,10]	[300,10]

In Tabelle 46 und Tabelle 47 werden die Ergebnisse der Bestimmung des optimalen α-Wertes aufgelistet. Zur jeweiligen Semantik werden die Erfolgsrate (success rate - SR) und dazugehörige durchschnittliche Anzahl an Versuchen (Iterationen - itt) angegeben.

Tabelle 46 Bestimmung des optimalen α-Wertes auf Basis der Erfolgsraten (ung. Personen-IDs)

α	**77993**		**PIN**		**Pseudonym**		**Symbol**		**Woher**	
	itt	**SR in %**	**itt**	**SR in %**	**itt**	**SR in %**	**itt**	**SR in %**	**itt**	**SR in %**
0	500	0	83	0	500	0	125	0	83	0
0,1	359	22,22	83	0	500	0	125	0	83	0
0,2	284,52	**100**	83	7,41	500	0	125	0	83	0
0,3	177,81	96,3	67	59,26	500	0	106,33	11,11	81	3,7
0,4	118,69	96,3	57,05	70,37	500	0	94,13	29,63	74,5	14,81
0,5	92,17	88,89	47	**85,19**	500	0	83,1	**37,04**	68,67	**33,33**
0,6	85,1	74,07	41,48	77,78	500	0	73,67	33,33	53,25	14,81
0,7	59,67	77,78	37,95	81,48	500	0	61,29	25,93	63	11,11
0,8	72,17	44,44	32,54	48,15	500	0	56	7,41	49	3,7
0,9	88,27	40,74	26,67	55,56	500	0	58	3,7	29,5	7,41
1	33	22,22	27,57	25,93	500	0	125	0	83	0

Die Erfolgsrate SR gibt an, bei wie viel der angegriffenen Referenzdaten eine positive Falschakzeptanz generiert werden konnte. Eine SR von 100 % heißt demnach, dass alle Referenzdaten erfolgreich angegriffen werden konnten. Die durchschnittliche Anzahl der Versuche für einen solchen Angriff wird durch den Wert itt (Iterationen) bestimmt.

Tabelle 47 Bestimmung des optimalen α-Wertes auf Basis der Erfolgsraten (gerade Personen-IDs)

α	77993		PIN		Pseudonym		Symbol		Woher	
	itt	SR in %	itt	SR in %	itt	SR in %	itt	SR in %	Itt	SR in %
0	125	0	250	0	83	0	500	0	125	0
0,1	125	0	250	0	83	0	500	0	120	3,85
0,2	111	7,69	223,5	15,38	83	0	500	0	95,5	38,46
0,3	99,67	46,15	172,1	38,46	74,71	26,92	500	0	86,84	96,15
0,4	90,68	73,08	158,14	**53,85**	64,36	53,85	500	0	65,12	100
0,5	75,19	80,77	133,55	42,31	55,24	65,38	500	0	50,88	100
0,6	65,48	**80,77**	88,43	26,92	48,22	**69,23**	500	0	43,69	**100**
0,7	56,75	76,92	112	7,69	40,08	50	500	0	36,92	96,15
0,8	51,35	76,92	116,5	7,69	30,9	38,46	500	0	32,46	92,31
0,9	39,36	53,85	250	0	28,64	42,31	500	0	24,52	80,77
1	35,44	34,62	250	0	35,5	23,08	500	0	28,55	76,92

Die Erfolgsraten für die Arbeitspunkte FRR=20%, FRR=30% und FRR=40% wurden mittels optimierten α-Wertes berechnet. In Tabelle 48 und Tabelle 49 sind die Ergebnisse im Vergleich zu den Ergebnissen von Galbally et al. aus [GaFO07] dargestellt.

Tabelle 48 Erfolgsraten des HC-Algorithmus für alle ungeraden Personen-IDs im Vergleich zu [GaFO07]

	Arbeitspunkt (in %)		
	FRR=20	FRR=30	FRR=40
Success rate (in %): [GaFO07]	98.12	96.60	94.90
Success rate (in %): 77993	100	100	40,74
Success rate (in %): PIN	88,89	85,19	51,85
Success rate (in %): Pseudonym	0	0	0
Success rate (in %): Symbol	44,44	37,04	11,11
Success rate (in %): Woher	70,37	33,33	7,41

Die erzielten experimentellen Ergebnisse in Tabelle 48 und Tabelle 49 zeigen, dass für den Arbeitspunkt FRR=20% die erzielten Erfolgsraten in einigen Semantikklassen denen von Galbally et al. ähneln. Die Semantiken Symbol und Pseudonym (ungerade Personen-IDs) sowie Symbol (gerade Personen-IDs) zeigen starke Abweichungen. Zuletzt genannte erreicht lediglich ein Erfolgsrate von SR=11,54 %, für die anderen Arbeitsbereiche werden keine erfolgreichen Angriffe verzeichnet.

Tabelle 49 Erfolgsraten des HC-Algorithmus für alle geraden Personen-IDs im Vergleich zu [GaFO07]

	Arbeitspunkt (in %)		
	FRR=20	**FRR=30**	**FRR=40**
Success rate (in %): [GaFO07]	98.12	96.60	94.90
Success rate (in %): 77993	96,15	80,77	11,54
Success rate (in %): PIN	96.15	46,15	3.85
Success rate (in %): Pseudonym	80,77	57,69	42,31
Success rate (in %): Symbol	11,54	0	0
Success rate (in %): Woher	100	100	100

Des Weiteren kann beobachtet werden, dass in der Semantik Pseudonym (ungerade Personen-IDs) für keinen der drei Arbeitspunkte einen positiven Angriffsverlauf generiert werden kann. Auch innerhalb des Arbeitspunktes FRR=30% können für einige Semantiken teils ähnlich Erfolgsraten (vgl. zu Galbally et al.) generiert werden. Für den Arbeitspunkt FRR=40% kann nur für die Semantik Woher (gerade Personen-IDs) ein ähnliches Ergebnis erreicht werden. Diese Semantik ist auch die einzige Klasse, welche für alle Arbeitsbereiche eine 100% Erfolgsrate verzeichnete. Alle übrigen Semantikklassen kommen über eine Erfolgsrate von rd. 52% nicht hinaus. Eine Ursache für diesen Unterschied könnte die Verwendung von 131 Merkmalen anstelle der 40 Merkmale sein, welche der Handschriftenalgorithmus von Galbally et al. verwendet. Es scheint demnach für den HC-Algorithmus schwerer zu sein, einen passenden Merkmalsvektor zu finden.
Schaut man sich die Schwellenwerte (Hamming-Distanzen) jener Semantiken an, welche die schlechtesten Erfolgsraten (SR) erzielten, können Gemeinsamkeiten festgestellt werden (Tabelle 42 und Tabelle 43). Für die Semantik Pseudonym (ungerade Personen-IDs) sind die Schwellenwerte (Hamming-Distanzen) am geringsten, dicht gefolgt von den Werten der Semantik Symbol (gerade Personen-IDs). Diese beiden Semantikklassen erzielen auch die schlechtesten Erfolgsraten. Die Semantikklasse Woher (gerade Personen-IDs), welche in allen Arbeitsmodi stets eine 100% Erfolgsrate erreicht, besitzt hingegen die größten Schwellenwerte (Hamming-Distanz). Hier kann eine Korrelation zwischen den Schwellenwerten und den Erfolgsraten festgestellt werden, die durchaus zu erwarten war und eine Erklärung für das schlechte Abschneiden einiger Semantikklassen liefert. In Abbildung 55 sind die maximal erzielten Erfolgsraten (gerade und ungerade Personen-ID) bezogen auf den Schwellenwert für alle Semantiken abgebildet. Es ist klar zu erkennen, dass der HC-Algorithmus bei einem Schwellenwert (Hamming-Distanz) von fünf oder kleiner keine Erfolgsraten (Falschakzeptanzen) mehr generieren kann. Um diese neue Erkenntnis zu verifizieren, wurde ein zusätzlicher Test durchgeführt. Der Test soll zeigen, ob diese Eigenschaft bei allen Semantikklassen auftritt. Hierfür wurden für alle Semantikklassen die in Tabelle 41 gelisteten N, M Wertepaare bei einem festen Schwellenwert von fünf und einem α-Wert von 0,5 berechnet. Das Ergebnis zeigte bei keiner der getesteten Semantikklassen ein positives Ergebnis, die Erfolgsrate lag durchweg bei 0%. Die beschriebene Eigenschaft tritt demnach bei allen Semantikklassen auf.

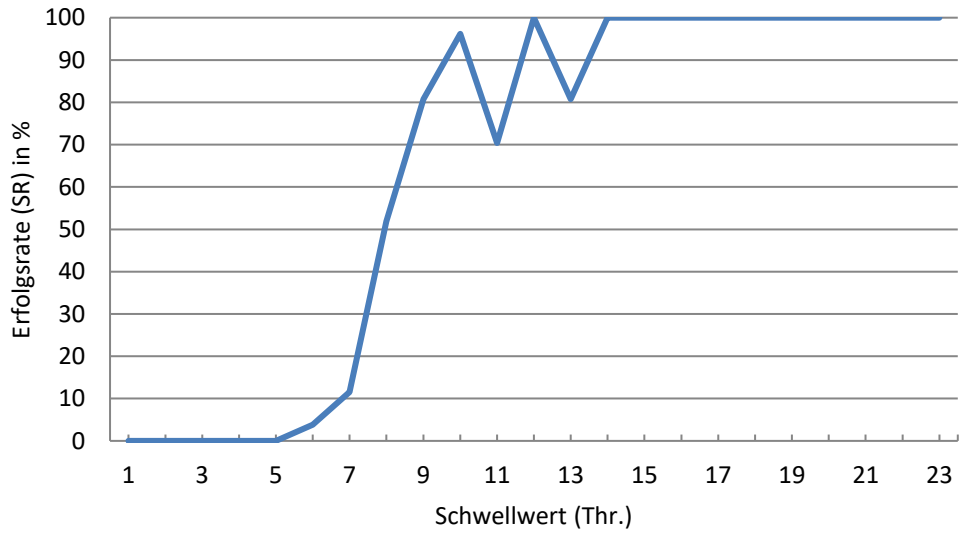


Abbildung 55 Erfolgsrate des HC-Algorithmus in Abhängigkeit zum Schwellenwert

In dieser Evaluation stößt der HC-Algorithmus offensichtlich an seine Grenzen. Er kann somit, unabhängig von der verwendeten Semantik, nicht mehr als 96% Übereinstimmung (131 Merkmale maximal 5 Merkmale unterschiedlich) zwischen Angriffs-BioHash und Referenz-BioHash erzielen.
Wie im Versuchsaufbau bereits erwähnt, dienen die Handschriftendaten (Merkmalsvektoren) der einen Testgruppe als Basis für eine Startmenge G, um Angriffsdaten zu erzeugen, die wiederum auf die andere Testgruppe angewendet werden. Der Erfolg eines Angriffs hängt demnach stark von der gewählten Startmenge ab. Im Gegensatz zu einem evolutionären Algorithmus ändert der HC-Algorithmus die Individuen (Merkmalsvektoren) nicht, sondern sucht aus der Startmenge lediglich die Merkmalsvektoren heraus, die eine möglichst hohe Übereinstimmung mit den anzugreifenden Referenzdaten haben. Besitzt die Startmenge keine Individuen (Merkmalsvektoren), die denen der gesuchten Merkmalsvektoren hinreichend ähneln, so wird der HC-Algorithmus entsprechend keine passenden finden und somit auch keine erfolgreichen Angriffe erzielen können.
Diese Eigenschafft des HC-Algorithmus scheint auch der Grund für die unterschiedlichen Ergebnisse der Semantik Pseudonym in den jeweiligen Testgruppen (ungerade/gerade Personen-ID) zu sein. In der ungeraden Testgruppe Pseudonym sind für die jeweiligen Schwellenwerte th_{20}=5, th_{30}=3 und th_{40}=2 (siehe Tabelle 42) ermittelt worden. Diese wurde auf Basis der FRR der entsprechenden Testgruppe, Pseudonym (ung. Personen-ID) siehe Anlage 13 im Anhang, ermittelt. Hier zeigt sich, dass die FRR – Kurve steil nach unten abfällt gegenüber der FRR der anderen Testgruppe (gerade Personen-ID) und somit sehr niedrige Schwellenwerte für die drei Arbeitspunkte entstehen. Diese führt wiederum zu den schlechten Erfolgsraten des HC gegenüber der anderen Testgruppe, wo die Schwellenwerte höher sind (th_{20}=13, th_{30}=11, th_{40}=9). Gleiches zeigt sich bei der Semantik "Symbol". Hier treten die niedrigen Schwellenwerte (th_{20}=8, th_{30}=5, th_{40}=3) bei der Testgruppe mit geraden Personen IDs auf, wohingegen die Testgruppe mit ungeraden Personen-IDs höhere Schwellenwerte aufweist (th_{20}=13, th_{30}=11, th_{40}=8). Entsprechend wirkt sich das auch auf die Erfolgsraten des HC-Algorithmus aus, was zu den unterschiedlichen Ergebnissen der Testgruppen in dieser Semantik führt.

Für die erzielten und teils unterschiedlichen Erfolgsraten, innerhalb der Semantikklassen der hier durchgeführten Evaluation, können u.a. zwei Ursachen bestimmt werden. Zum einen sind die Schwellenwerte für die drei Arbeitsmodi (FRR=20%, FRR=30% und FRR=40) teils recht unterschiedlich, trotz gleicher Semantikklasse. Und zum anderen, dass die Erfolgschancen des HC-Algorithmus einen passenden Treffer (Match) zu erzielen stark von der gewählten Startmenge G abhängt. Innerhalb der hier verwendeten Testdaten konnte der HC-Algorithmus keinen Treffer mehr bei einem Schwellenwert von fünf oder weniger erzielen.
Des Weiteren kann festgestellt werden, dass die erzielten Ergebnisse des HC-Algorithmus für jeden Durchlauf variieren. So wird beispielsweise bei der Bestimmung der optimalen N, M Wertepaare für die Semantik 77993 (gerade Personen-IDs) eine Erfolgsrate von SR=84,62% erzielt bei N=20, M=25, α=0,5 und FRR=30% (siehe Tabelle 44 farblich markiert). Bei der Prozedur zur Bestimmung des optimalen Wertes von α wird hingegen nur ein Wert von 80,77% bei gleichen Parametern (N=20, M=25, α=0,5 und FRR=30%) erzielt, siehe Tabelle 47. Diese Variation kann u.a. damit begründet werden, dass im ersten Schritt des HC-Algorithmus eine beliebige Anzahl von Mustern (Merkmalsvektoren) aus der globalen mehrdimensionalen Normalverteilung *G* entnommen werden soll, siehe Abschnitt 7.1.3. Die Anzahl der entnommenen Muster wird mit dem Wert N bestimmt, jedoch sind die Muster, die entnommen werden, nicht immer dieselben, sondern können mit jedem neuen Start des HC-Algorithmus variieren. Die Unterschiede in den Ergebnissen (Erfolgsrate und Anzahl der durchschnittlichen Versuche) die im Rahmen dieser experimentellen Ergebnisse beobachtet wurden, sind jedoch nicht gravierend. Galbally et al. gehen auf diese Variation in ihrer Arbeit [GaFO07] nicht gesondert ein.

Designvorschläge
Um einen, wie oben beschrieben, HC-Angriff zu unterbinden, sind verschiedene Maßnahmen denkbar. Eine recht einfache Maßnahme ist das Herabsetzen des Schwellenwertes auf maximal fünf oder weniger. Ob dieser "Grenzschwellenwert" auch für andere Handschriftendatenbanken gilt, müsste in fortführenden Tests evaluiert werden. Prinzipiell kann anhand der Testergebnisse abgelesen werden, dass je geringer der Schwellenwert gewählt wird, desto schwieriger wird es für den HC-Algorithmus passende Angriffsdaten zu finden. Ein niedriger Schwellenwert führt jedoch auch zu hohen Falschrückweisungsraten (FRR) und macht das System entsprechend unkomfortabler für die Nutzer.
Eine weitere Methode könnte der Einsatz von homomorpher Verschlüsselung sein. So könnten Merkmalsvektor, Intervallmatrix, BioHash-Werte, Vergleichswert und der Schwellenwert mit einer homomorphen Verschlüsselungstechnik verschlüsselt werden. Die Prozesse der BioHash-Generierung, des Vergleichens und der Entscheidung könnten im verschlüsselten Raum durchgeführt werden. So wären die Daten während der Verarbeitung geschützt. Selbst wenn z.B. der Angreifer im Besitz passender verschlüsselter Merkmalsvektoren wäre, um eine Startmenge für den HC-Algorithmus zu bilden, könnte der eingesetzte HC-Algorithmus in dem hier betrachteten Angriffsszenario den verschlüsselten Vergleichswert nicht mehr nutzen. Der Verifikationsalgorithmus bzw. deren Komponenten (z.B. Merkmalsextraktor) müssen für den Einsatz einer homomorphen Verschlüsselung jedoch vorbereitet werden. Ggf. können nicht alle Funktionen ohne spezielle Anpassungen direkt durch homomorphe Funktionen (Addition/Multiplikation) abgebildet werden.

Weiterhin sind traditionelle Schutzmechanismen denkbar. So könnten beispielsweise die Kommunikationswege der jeweiligen Komponenten des BioHash Algorithmus kryptografisch verschlüsselt werden. Das Einspielen der Merkmalsvektoren und das Auslesen des Gütewertes (Vergleichswert) wären dann nicht mehr ohne weiteres möglich. Zusätzlich wäre es denkbar, die Laufumgebung auf dem der Algorithmus ausgeführt wird und die dazugehörigen Komponenten bzw. Schnittstellen sicherheitstechnisch zu härten. Auch ist eine Ausführung einzelner oder aller Komponenten auf einer geschützten Umgebung z.B. Smartcard (Secure Memory Card) vorstellbar.
Die zuletzt genannten Designvorschläge (traditionelle Schutzmechanismen) gelten für biometrische Verifikationssysteme im Allgemeinen und können je nach Beschaffenheit des Systems in der Regel relativ leicht implementiert werden. Die Umsetzung homomorpher Verschlüsselung und die entsprechende Datenverarbeitung im verschlüsselten Raum kann indes nicht so einfach auf andere Verifikationssysteme übertragen werden. Einige Operationen (z.B. Vergleiche) können nicht ohne Anpassungen mittels homomorpher Verschlüsselung umgesetzt werden.

8 Zusammenfassung

Im letzten Kapitel werden die Ergebnisse der Arbeit im Abschnitt 8.1 zusammengefasst dargestellt. Außerdem werden in Abschnitt 8.2 die Schwierigkeit und Herausforderung erläutert, die bei der Bearbeitung der jeweiligen Forschungsaufgaben aufgetreten sind. Im letzten Abschnitt sind weiterführende Aufgaben beschrieben, die als Ansatzpunkt für potentielle Forschungsfragen in diesem Bereich herangezogen werden können.

8.1 Fazit

In dieser Arbeit wurde u.a. die Sicherheit biometrischer Verifikationssysteme insbesondere die des BioHash-Algorithmus von Vielhauer [Viel06] beleuchtet. Im ersten Teil der Arbeit (FA1) wurden bekannte Angriffspunkte und Schwachstellen von biometrischen Verifikationssystemen vorgestellt. Außerdem wurden bekannte Klassifikationsverfahren von biometrischen Angriffen (direkte und indirekte Angriffe) aufgezeigt. Aufbauend auf den bekannten Angriffspunkten und Klassifikationsverfahren konnte ein neues Klassifikationsverfahren erstellt und vorgestellt werden, welches beide Aspekte zusammenführt. So können biometrische Angriffsverfahren detaillierter klassifiziert und deren Gefahrenpotential entsprechend besser eingeschätzt werden. Exemplarisch wurden zwölf bekannte biometrische Verifikationsverfahren nach dieser neuen Methode klassifiziert und deren Gefahrenpotential auf dynamische Handschriftenverifikationsverfahren und biometrische Verifikationssysteme im Allgemeinen bewertet bzw. eingeschätzt. Des Weiteren ist ein bekanntes Angriffsverfahren ausgewählt und zur Anwendung auf den BioHash-Algorithmus adaptiert worden. Hier wurden vorherige Arbeiten des Autors verwendet, entsprechend modifiziert und weiterentwickelt sowie mit neuen Testdaten evaluiert. Anschließend sind Designvorschläge für den BioHash-Algorithmus definiert worden, die empfehlen, dass bestimmte Merkmale nicht verwendet werden sollten. So wird sichergestellt, dass diese spezielle modifizierte Angriffstechnik auf den BioHash-Algorithmus nicht mehr angewendet werden kann. Eine Evaluation ohne Verwendung dieser kritischen Merkmale zeigt, dass die Verifikationsperformanz des BioHash-Algorithmus nicht signifikant beeinträchtigt wird, in vier von fünf der getesteten Semantiken ist sie sogar gestiegen. Lediglich die Kollisionsreproduktionsrate (CRR) hat sich im Durchschnitt um 2,83 Prozentpunkte verschlechtert. Auf Grundlage dieser Ergebnisse wird empfohlen, diese Merkmale in potentieller Produktivumgebung nicht zu verwenden. Die so geschlossene Schwachstelle und deren positiver Effekt auf die Sicherheit der Referenzdaten wiegt die nur unwesentlich geringere Kollisionsreproduktionsrate aus Sicht des Autors auf.

Zusammenfassung FA1

Die Forschungsaufgabe FA1 und deren Teilaufgaben A - C konnten erfolgreich bearbeitet werden.
Teilaufgabe A: Klassifizieren ausgewählter Angriffsverfahren auf biometrische Verifikationssysteme.
Es wurde ein neues Klassifikationsverfahren eingeführt sowie zwölf bereits bekannte Angriffsverfahren ausgewählt und basierend auf dem neuen Verfahren exemplarisch klassifiziert.

Teilaufgabe B: Bewerten von klassifizierte Angriffsverfahren, bezogen auf ihre Adaptierbarkeit.
Die in Teilaufgabe A klassifizierten Angriffstechniken sind hinsichtlich ihres Gefahrenpotentials und ihrer Adaptierbarkeit auf andere biometrische Verifikationssysteme eingeordnet worden.

Teilaufgabe C: Adaptieren und evaluieren eines ausgewählten Angriffsverfahrens auf den in [Viel06] vorgestellten handschriftenbasierenden Verifikationsalgorithmus.
Eine ausgewählte Angriffstechnik wurde entsprechend adaptiert und evaluiert. Darauf aufbauend sind Designvorschläge getätigt und erneut evaluiert worden. Es wird empfohlen die Designvorschläge auch auf andere biometrische Verifikationssysteme anzuwenden.

Im Zweiten Teil der Arbeit wurde ein neues Verfahren zur Generierung von künstlichen Handschriftendaten vorgestellt. Dieses Verfahren extrahiert bestimmte Parameter von Buchstaben und Ziffern realer Handschriftendaten und generiert Schreibalphabete. Diese Alphabete werden mittels bestimmter Parameter modifiziert und bilden neue Alphabete und somit neue Schreibindividuen. Für die Erzeugung künstlicher Handschriftendaten werden nun die Buchstaben der künstlichen Schreibindividuen aneinandergereiht. Mit dieser Methode können eine beliebige Anzahl von Schreibindividuen und entsprechende Handschriftensignale generiert werden. Der Schreibinhalt kann für die künstlichen Signale entsprechend gewählt werden. Zur Ermittlung des Verifikationsverhaltens gegenüber realen Handschriftendaten wurden 110 künstliche Schreibindividuen erzeugt und in vier verschiedenen Semantikklassen evaluiert. Die ersten Evaluationsergebnisse zeigen, dass die Interklassen-Variabilität der künstlichen Handschriftendaten bzw. der Schreibindividuen sich von realen Handschriftendaten bzw. Personen durchaus unterscheidet. So sind die Werte für die EER relativ gering (im Durchschnitt unter 0,11% gegenüber 3,3% realer Handschriften) und es konnten keine Kollisionsraten festgestellt werden. Das wären hervorragende Werte für das Verhalten realer Handschriften innerhalb eines handschriftenbasierten Verifikationssystems. Die 0% Kollisionsrate in allen getesteten Schreibsemantiken untermauert das untypische Verhalten der künstlichen Handschriftendaten während der Evaluation. Die Reproduktionsraten der künstlichen Individuen spiegeln im Vergleich zu den originalen Handschriftendaten ein natürlicheres Verhalten wider. So kann im Durchschnitt eine Reproduktionsrate von 9,7% im nicht optimierten Arbeitsmodus und eine Reproduktionsrate von 62,8% im CRR optimierten Arbeitsmodus ermittelt werden. Bei den realen Handschriftendaten sind es im Vergleich dazu 6% im nicht optimierten, respektive 59% im optimierten Arbeitsmodus. Die ersten Evaluationsergebnisse zeigen, dass für die gewählten Modifikationsparameter das Verfahren zur Generierung von künstlichen Handschriftendaten nur bedingt geeignet ist. Die erzeugten Daten verhalten sich bezüglich der Interklassen-Variabilität nicht wie reale Handschriftendaten. Die Kompensation der Interklassen-Variabilität biometrischer Daten ist jedoch eines der großen Herausforderungen eines biometrischen Erkennungssystems. Diese künstlichen Handschriftendaten können aus Sicht des Autors dementsprechend nicht als Testdaten für die Optimierung eines handschriftenbasierten Verifikationssystems verwendet werden. Für die Optimierung eines biometrischen Systems hinsichtlich der Verarbeitungsperformanz (Verarbeitungszeit) z.B. für Merkmalsextraktion etc. können die künstlich erzeugten

Schreibindividuen und Handschriftendaten aus Sicht des Autors hingegen eingesetzt werden.
Der äußere Eindruck der künstlichen Handschriftensignale wirkt bei genauerer Betrachtung teils unnatürlich. Das liegt u.a. daran, dass keine spezielle Verbindungstechnik zur Aneinanderreihung der Buchstaben angewendet wurde. Weiterhin wirken einige Buchstaben zum Teil unnatürlich, so sind beispielsweise die i-Punkte der Semantik "Seife" bisweilen nicht mehr identifizierbar.
Weiterhin zeigen die Evaluationsergebnisse, dass die Ähnlichkeit der künstlichen Daten zu den realen Handschriftendaten nicht ausreichend hoch ist, um positive Angriffsdaten zu produzieren.
Gleichzeitig bietet das Verfahren jedoch noch Verbesserungspotential. So können beispielsweise Verbindungslinien zwischen den Buchstaben erzeugt werden oder die Stiftabsetzpunkte genauer positioniert werden. Weiterhin ist eine andere Kombination oder die Wahl neuer Modifikationsparameter hilfreich, die Eigenschaften künstlicher Handschriften zu steigern. So können das Verifikationsverhalten als auch das Erscheinungsbild der künstlichen Handschriften potentiell verbessert werden. Damit würde sich ggf. auch die Eignung der künstlichen Handschriftendaten als Angriffsdaten erhöhen. Dies sind nur einige Ansatzpunkte für weitere Studien in diesem Bereich.

Zusammenfassung FA2

Die Forschungsaufgabe FA2 und deren Teilbereiche A bis C konnten überwiegend erfolgreich bearbeitet werden.
Teilaufgabe A: Entwickeln und evaluieren eines Verfahrens zur Generierung von künstlichen Handschriften
Es konnte ein Verfahren zur Generierung künstlicher Handschriftensignale entwickelt werden. Es ist mit diesem Verfahren möglich, künstliche Schreibindividuen zu erzeugen und den Schreibinhalt frei zu wählen. Äußerlich wirken die künstlichen Signale bei genauerer Betrachtung jedoch unnatürlich. Des Weiteren ähnelt das Verhalten der künstlichen Handschriften dem Verhalten (Fehlerraten/Reproduktionsraten) realer Handschriften nur zum Teil.

Teilaufgabe B: Bestimmen geeigneter Parameter, welche die Eigenschaften der künstlichen Handschriftendaten so anpassen, dass diese den Eigenschaften realer Handschriftendaten ähneln.
Es konnten einige Modifikationsparameter bestimmt werden, welche die Verifikationsperformanz beeinflussen können. Diese können sich jedoch auch negativ auf das Erscheinungsbild der künstlichen Handschriften auswirken. Ein Kompromiss aus realistischem Erscheinungsbild und vergleichbaren Fehlerraten zu authentischen Handschriften konnte in diesem ersten Versuch nicht gefunden werden.

Teilaufgabe C: Untersuchen, inwieweit die künstlich erzeugten Handschriften für Angriffe auf reale Handschriftendaten eingesetzt werden können.
Die Qualität der künstlichen Handschriftendaten erreicht innerhalb der experimentellen Tests nicht die Qualität, um erfolgreiche Angriffe auf reale Handschriften durchzuführen.

Der dritte Bereich der Arbeit widmete sich bekannten Hill-Climbing-Verfahren und wie diese auf biometrische Verifikationssysteme angewendet werden können. Insbesondere die Adaption eines ausgewählten HC-Verfahrens auf den BioHash-Algorithmus [Viel06] stand im Vordergrund. Der adaptierte HC-Algorithmus sucht aus einer Menge von Merkmalsvektoren (Startmenge des Angreifers) diejenigen heraus, die für den Angriff auf einen Referenz-BioHash am geeignetsten erscheinen. Hierfür verwendete er den Vergleichswert zwischen dem Referenz-BioHash und dem BioHash, welcher auf Basis dieser Merkmalsvektoren erzeugt wurde. Für die Evaluation des HC-Algorithmus werden innerhalb des Verifikationsalgorithmus 131 statistische Merkmale eingesetzt, dementsprechend besitzen die gesuchten Merkmalsvektoren 131 Werte. Die Erfolgsraten sind teils ähnlich gut, wie die des originalen HC-Algorithmus, vorgestellt von Galbally et al. in [GaFO07]. Auf dem ersten Blick scheint es so, als würden bestimmte Semantiken resistenter gegen diesen Angriff sein als andere. Bei genauerer Betrachtung ist jedoch zu erkennen, dass ab einem bestimmten Vergleichswert (Schwellenwert kleiner gleich fünf) keine positiven Treffer mehr erzielt werden konnten. So erzielte z.B. die Semantik "Pseudonym" für alle drei Arbeitspunkte bei sehr geringen Schwellenwerten (th_{20}=5, th_{30}=3 und th_{40}=2) keine positiven Treffer, wohingegen die gleiche Semantik bei einer anderen Startmenge mit höheren Schwellenwerten (th_{20}=13, th_{30}=11 und th_{40}=8) für die jeweiligen drei Arbeitspunkte eine Erfolgsrate von 80%, 58% bzw. 42% erreicht werden konnte. Bei der Semantik „Woher“ konnte sogar in allen drei Arbeitspunkten eine Erfolgsrate von 100% festgestellt werden. Hier liegen die dafür ermittelten Schwellenwerte vergleichsweise hoch (th_{20}=23, th_{30}=20 und th_{40}=17).
Zusätzlich hängt der Erfolg des HC-Algorithmus von der zugrundeliegenden Startmenge ab, welche als „Datenpool“ für die Angriffsdaten dient. Generell kann der Unterschied der Erfolgsraten zwischen denen in dieser Arbeit und denen von Galbally et al. in [GaFO07] durch die unterschiedliche Anzahl der verwendeten biometrischen Merkmale vermutet werden. So wurden in dieser Arbeit 131 Merkmale und in der von Galbally lediglich 40 Merkmale verwendet. Der Merkmalsraum ist somit dreimal Größer als in [GaFO07]. Entsprechend sind die Erfolgschancen geringer. Des Weiteren konnten Designvorschläge formuliert werden, welche potentielle HC-Angriffsverfahren neutralisieren können. Diese Designvorschläge beziehen sich nicht nur auf die direkte Arbeitsweise des Verifikationsalgorithmus, sondern zielen u.a. auf die Laufumgebung des Algorithmus ab. So wird beispielsweise empfohlen, die Schnittstellen und Kommunikationsstrecken des Systems zu härten und/oder wenn möglich alle Komponenten in einer geschützten Umgebung auszuführen. Auch ist der Einsatz homomorpher Verschlüsselungsmethoden denkbar, um sensible Daten (z.B. Merkmalsvektor, Vergleichswert) im verschlüsselten Raum zu prozessieren. Hierfür müssen jedoch entsprechend die Komponenten des Verifikationssystems (BioHash-Erzeugung, Vergleicher usw.) dahingehend untersucht und ggf. angepasst werden. Alle benötigten Operationen können unter Umständen nicht mit den bereitgestellten Operationen homomorpher Verfahren eins zu eins umgesetzt werden.

Zusammenfassung FA3

Die Forschungsaufgabe FA3 und deren Teilaufgaben konnte erfolgreich abgearbeitet werden.

Teilaufgabe A: Untersuchen, inwieweit Hill-Climbing-Angriffsverfahren auf den handschriftenbasierten Verifikationsalgorithmus [Viel06] adaptiert werden können.
Ein ausgewählter HC-Algorithmus konnte adaptiert und erfolgreich eingesetzt werden.

Teilaufgabe B: Formulieren von Verbesserungsvorschlägen für den handschriften-basierten Verifikationsalgorithmus [Viel06] zur Vorbeugung Hill-Climbing basierter Angriffsverfahren.
Es sind Designvorschläge formuliert worden, welche für die Laufumgebung des Verifikationsalgorithmus angewendet werden können.

8.2 Herausforderungen

In diesem Abschnitt sollen kurz die Herausforderung beschrieben werden, denen sich der Autor während der Bearbeitung der vorliegenden Arbeit gegenübersah.
Bei der Recherche von potentiellen biometrischen Angriffsverfahren und Techniken war es teilweise schwierig diese in Forschungsbeiträgen zu finden. In einigen Fällen haben die zuständigen Autoren ihre neuen Techniken verständlicherweise auch nicht als Angriffsverfahren deklariert. Ein neues Verfahren zur Generierung von biometrischen Handschriftendaten stellt nicht unmittelbar eine Angriffstechnik dar. Es könnte jedoch als Bestandteil eines potentiellen Angriffsverfahrens dazu eingesetzt werden. Des Weiteren sind einige Angriffsverfahren innerhalb der wissenschaftlichen Publikationen nur unzureichend und unpräzise beschrieben worden. Die Klassifizierung der Angriffe und die Einschätzung der potentiellen Gefahr, die von ihnen ausgeht, stellten sich als schwierig dar.
Für die Evaluation der in dieser Arbeit behandelten Verfahren (FA1, FA2 und FA3) wurden biometrische Daten benötig. Je mehr biometrische Testdaten zur Verfügung stehen, desto aussagekräftiger sind die erzielten Fehlerraten und somit auch die Qualität der jeweiligen Verfahren. Diese biometrischen Daten zu erfassen stellt, wie in vielen anderen biometrischen Modalitäten auch, eine zeitliche und logistische Herausforderung dar. Für die Aufzeichnung der Buchstaben und Ziffern einer Person inklusive der Handschriftensignale der vier Semantikklassen (siehe FA2) wurden im Durchschnitt 18 Minuten benötigt. Dabei musste das Aufzeichnungsequipment (Signaturtablett und Notebook) teilweise zu den Personen vor Ort gebracht werden. Anschließend sind die Daten hinsichtlich ihrer Quantität (z.B. korrekte Anzahl der Klein- und Großbuchstaben bei der Alphabetaufzeichnung) manuell geprüft worden.
Eine weitere Herausforderung war die teils lange Berechnungszeit in allen durchgeführten experimentellen Untersuchungen. Das lag u.a. an den verwendeten Matlab Skripten, welche nicht auf maximale Performanz ausgelegt sind. So sind beispielsweise einige Hill-Climbing-Tests mehrere Stunden gelaufen. Auch die Merkmalsextraktion von mehreren tausend künstlichen Handschriftensignalen führte zu langen Wartezeiten.

8.3 Zukünftige Arbeiten

Aufbauend auf den in dieser Arbeit erzielten Ergebnissen können weiterführende Arbeiten definiert werden. Nachfolgend sollen einige dieser zukünftigen Forschungsfragen bzw. Forschungsaufgaben grob formuliert werden.
Bezogen auf die in FA1 eingeführten Angriffsklassen und deren Bezug zu den Angriffspunkten eines biometrischen Verifikationssystems kann künftig untersucht werden, wie Angriffe eingeordnet werden, die mehrere Angriffspunkte ausnutzen. Das in dieser Arbeit vorgestellte Klassifikationsverfahren verwendet zur Klassifizierung lediglich einen

Angriffspunkt. Zusätzlich könnte ein Katalog mit bekannten Gegenmaßnahmen erstellt werden, der sich an die Angriffsklassen des Klassifikationsverfahrens anlehnt. Des Weiteren kann innerhalb der FA1 eine potentielle Optimierung des Angriffsverfahrens identifiziert werden. Das Angriffsverfahren generiert künstliche Handschriftensignale auf Basis zurückgerechneter Merkmalsvektoren (siehe Abschnitt 4.2.4) realer Handschriftendaten. Hierfür werden u.a. die Anzahl der Maximas des X- und Y-Signals verwendet, um mittels einer Spline-Interpolation Schreibsignale zu generieren. In einer künftigen Arbeit könnten die Merkmale „Anzahl der Minimas des X- und Y-Signals“ bei der Spline-Interpolation ebenfalls verwendet und entsprechend evaluiert werden.
Für den Bereich der Forschungsaufgabe FA2 können ebenfalls weiterführende Arbeiten identifiziert werden. Insbesondere die ungewöhnlichen Fehlerraten der künstlichen Handschriften gegenüber den realen Handschriftendaten könnten genauer untersucht werden. Im ersten Schritt sollte untersucht werden, welche Merkmale der künstlichen Handschriftendaten nie oder nur selten während des Verifikationsprozesses reproduziert werden können. Außerdem sollten die Merkmale identifiziert werden, welche immer reproduziert werden können. Aufbauend auf diesem ersten Schritt sollte geprüft werden, welche Modifikationsparameter Einfluss auf diese Merkmale haben. Diese Parameter könnten entsprechend ihrer Auswirkung auf die Interklassen- bzw. Intraklassen-Variabilität gewählt werden. So könnte ggf. das Verifikationsverhalten von realen Handschriften besser reproduziert werden. Zusätzlich ist eine Kombination des Angriffsverfahrens und des Verfahrens zur Generierung von künstlichen Handschriften interessant. So könnten die Informationen des zurückgerechneten Merkmalsvektors verwendet werden, um das Verfahren zur Erzeugung von künstlichen Schreibindividuen zu optimieren. Eine weitere potentielle zukünftige Aufgabe ist das Formbild der künstlichen Handschriftendaten. Hier können Methoden angewendet werden, welche die aneinandergereihten Buchstaben miteinander verbindet (z.B. wie in [VaKB05]). So können ggf. nicht nur das Aussehen, sondern die Interklassen- bzw. Intraklassen-Variabilität von realen Handschriften besser imitiert werden. Weiterhin können Neuronale Netze bzw. Methoden des Maschinellen Lernens, wie z.B. vorgestellt in [Grav13] und [CHJO16], dahingehend evaluiert werden, inwieweit sie im Kontext dieser Arbeit zur Erzeugung künstlicher Handschriftensignale bzw. Schreibindividuen herangezogen werden können.
Mit Blick auf die Ergebnisse der Forschungsaufgabe FA3 können zwei weiterführende Forschungsaufgaben beschrieben werden. Zum einen kann der HC-Angriff auf den BioHash-Algorithmus wiederholt werden, wobei bei dieser Untersuchung weniger statistische Merkmale bei der Berechnung des BioHash-Wertes verwendet werden sollen. In der Arbeit von Scheidat ([Sche15]) wurden statistische Merkmale identifiziert, die für eine Handschriftenverifikation besser geeignet sind als andere. Diese Merkmale könnten in einer künftigen Evaluation der HC-Angriffstechnik verwendet werden. Die Erwartungshaltung, dass die Erfolgsraten zunehmen sobald weniger statistische Merkmale verwendet werden, könnte somit bestätigt werden. Außerdem ist die Adaption des Verfahrens auf einen weiteren handschriftenbasierten Verifikationsalgorithmus (z.B. den Secure Sketch-Algorithmus [SLM07], angepasst für die Handschrift in [ScVD09]) erstrebenswert, um die erzielten Ergebnisse zu untermauern.
Eine weitere potentielle künftige Aufgabe wäre es, den Verifikationsalgorithmus [Viel06] und deren Komponenten so zu adaptieren, dass Teile oder komplette Funktionen mittels homomorphen Verschlüsselungsmethoden berechnet bzw. bearbeitet werden können. Auch könnte das Verfahren von Gomez-Barrero et al. [GMG+17] (siehe Abschnitt 2.1.1

unter homomorphe Verschlüsselung), welches unabhängig von der biometrischen Modalität arbeitet, auf den Verifikationsalgorithmus [Viel06] adaptiert werden.
Die hier formulierte Liste künftiger Forschungsaufgaben ist sicherlich nicht als abschließend zu betrachten und könnte noch erweitert werden. Die hier beschrieben Punkte können jedoch als Einstiegspunkt künftiger Arbeiten betrachtet werden. Das Forschungsgebiet „Biometrie“ ist auf Grund der Vielzahl verschiedener biometrischer Modalitäten weitreichend. Auch die Fülle an Anwendungen und Einsatzgebiete der Biometrie als Authentifikationsmethode bietet ausreichend Forschungsspielraum für künftige Arbeiten und macht sie deshalb auch, aus Sicht des Autors, so interessant.

9 Literaturverzeichnis

[Adle03] Adler, A.: Sample images can be independently restored from face recognition templates. In: Proc. Canadian Conference on Electrical and Computer Engineering (CCECE), vol. 2, pp. 1163–1166, (2003)

[AFG+09] Alonso-Fernandez, F., Fierrez, J., Gilperez, A., Galbally, J. und Ortega-Garcia, J.: Robustness of Signature Verification Systems to Imitators with Increasing Skills. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition (ICDAR '09). IEEE Computer Society, pp. 728-732, (2009)

[AhBa17] Ahrabian, K. und Babaali, B.: On usage of autoencoders and siamese networks for online handwritten signature verification, arXiv:1712.02781, (2017)

[AlMA08] Almaksour, A., Mouchere, H. und Anquetil, E.: Fast online incremental learning with few examples for online handwritten character recognition, In: Pro-ceedings of the International Conference on Frontiers in Handwriting Recognition ICFHR, pp. 623–628, (2008)

[AlSB16] Alvarez, G., Sheffer, B. und Bryant, M.: Offline Signature Verification with Convolutional Neural Networks, Technical Report, Stanford University, Stanford (2016)

[AmRa13] Amirtharajan, R. und Rayappan, J.B.B.: Steganography-Time to Time: A Review, Research Journal of Information Technology 5, pp. 53-66, (2013)

[AMSL20] Advanced Multimedia and Securityy Lab - AMSL, Arbeitsgruppe an der Otto-von-Guericke Universität Magdeburg, https://omen.cs.uni-magdeburg.de/itiamsl/deutsch/home/index.html, aufgerufen am 29.01.20, (2020)

[BaAS09] Balbed, M.A.M., Ahmad, S.M.S. und Shakil, A.: ANOVA-Based Feature Analysis and Selection in HMM-Based Offline Signature Verification System‖, 2009 Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA-2009, pp. 66-69, (2009)

[BaCr14] Baroughi, A. F. und Craver, S.: Additive attacks on speaker recognition, SPIE 9028, Media Watermarking, Security, and Forensics 2014, (2014)

[BaCu05] Bartlow, N. und Cukic, B.: The Vulnerabilities of Biometric Systems - An Integrated Look and Old and New Ideas, Technical report, W.V. University, (2005)

[BaDL15] Barni, M., Droandi, G. und Lazzeretti, R.: Privacy Protection in Biometric-Based Recognition Systems: A marriage between cryptography and signal processing, IEEE Signal Processing Magazine, vol. 32, no. 5, pp. 66-76, (2015)

[BaGS09] Batista, L., Granger, E. und Sabourin, R.: A Multi-Hypothesis Approach for Off-Line Signature Verification with HMMs‖, 10th International Conference on Document Analysis and Recognition, ICDAR-2009, pp. 1315-1319, (2009)

[BaLM07] Ballard, L., Lopresti, D. und Monrose, F.: Forgery quality and its implications for behavioral biometric security, IEEE Transactions on Systems, Man, and Cybernetics, pp. 1107-1118, (2007)

[Balz11] Balzert H.: Authentifizierung und Autorisierung, In: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. Spektrum Akademischer Verlag, pp. 153-175, (2011)

[BCG+18] Blanco-Gonzalo, R., Corsetti, B., Goicoechea-Telleria, I., Husseis, A., Liu-Jimenez, J., Sanchez-Reillo, R., Eglitis, T., Ellavarason, E., Guest, R., Lunerti, C., Azimi, M., Nourmohammadi K. J., Ezennaya-Gomez, S., Whiskerd, N., Salih, R.

und Okoh, E.: Attacking a Smartphone Biometric Fingerprint System: A Novice's Approach, (2018)

[BDFR13] Biggio, B., Didaci, Fumera, L.G. und Roli, F.: Poisoning Attacks to Compromise Face Templates, International Conference on Biometrics (ICB), pp.1-7, Madrid, (2013)

[BeKA07] Bezine, H., Kefi, M. und Alimi, M.: On the beta-elliptic model for the control of the human arm movement, International Journal of Pattern Recognition 21, pp. 5–19, (2007)

[BeRo01] Behrens, M. und Roth, R.: Biometrische Identifikation: Grundlagen, Verfahren, Perspektiven, Vieweg Verlag Braunschweig/Wiesbaden (2001)

[Beut15] Beutelspacher, A.: Kryptologie, Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen, 10. Auflage, Springer Spektrum, (2015)

[BLML06] Ballard, L., Lopresti, D., Monrose, F. und Lorette, G. (Ed.) Evaluating the Security of Handwriting Biometrics, Tenth International Workshop on Frontiers in Handwriting Recognition, Suvisoft, (2006)

[BMC+05] Bless, R., Mink, S., Conrad, M., Kutzner, K., Blaß, E.-O., Hof, H.-J. und Schöller, M.: Grundlagen zur Kryptographie, In: Sichere Netzwerkkommunikation, X.systems.press, Springer, Berlin, Heidelberg, (2005)

[Bodd18] Boddeti, V. N.: Secure Face Matching Using Fully Homomorphic Encryption, Computer Vision and Pattern Recognition, BTAS 2018, (2018)

[BoGN05] Boneh, D., Goh, E.J. und Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts, Theory of Cryptography (TCC 2005), LNCS 3378, Springer Verlag, Berlin, Heidelberg, (2005)

[BrGV12] Brakerski, Z., Gentry, C. und Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping, In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS '12), ACM, New York, USA, pp. 309-325, (2012)

[BrVa11] Brakerski, Z. und Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE, IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, pp. 97-106, (2011)

[BrVa11a] Brakerski Z. und Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages, In Proceedings of the 31st annual conference on Advances in cryptology (CRYPTO'11), Springer-Verlag, Berlin, Heidelberg, pp. 505-524, (2011)

[Buch10] Buchmann J.: Kryptographische Hashfunktionen. In: Einführung in die Kryptographie, pp. 193-204, Springer-Lehrbuch, Berlin, Heidelberg, (2010)

[Cane20] Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols, In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science, Newport Beach, CA, USA, 2001, pp. 136-145, aktualisierte Fassung von Februar 2020 abgerufen unter: ia.cr/2000/067, (2020)

[CAP+06] Choi, W.Y., Ahn, D., Pan, S.B., Chung, K.I., Chung, Y. und Chung, S.H.: SVM-based speaker verification system for match-on-card and its hardware implementation. Electronics and Telecommunications Research Institute Journal 28(3), 320-328, (2006)

[Capp03] Cappelli, R.: Synthetic fingerprint generation, Handbook of Fingerprint Recognition, pp. 203–231, Springer, (2003)

[Capp04] Cappelli, R.: SFinGe: an Approach to Synthetic Fingerprint Generation, International Workshop on Biometric Technologies, (2004)

[Carl09] Carls, J. W.: A framework for analyzing biometric template aging and renewal prediction, Ph.D. thesis, US Air Force Institute of Technology, (2009)

[CaSt07] Cavoukian, A. und Stoianov, A.: Biometric Encryption: A Positive-Sum Technology that Achieves Strong Authentication, Security AND Privacy, Technischer Report, Office of the Information and Privacy Commissioner of Ontario, Toronto, (2007)

[CCKL16] Cheon, J.H., Chung, H., Kim, M. und Lee, K.W.: Ghostshell: Secure Biometric Authentication using Integrity-based Homomorphic Evaluations, IACR Cryptology ePrint Archive 2016, 484, (2016).

[CGGI16] Chillotti I., Gama N., Georgieva M. und Izabachène M.: Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds, Advances in Cryptology, ASIACRYPT 2016, LNCS 10031, Springer Verlag, Berlin, (2016)

[ChBl03] Chirillo, J. und Blaul, S.: Implementing Biometric Security, Wiley Publishing, Inc., Indianapolis, Indiana, USA, (2003)

[CHJO16] Carter, S., Ha, D., Johnson, I. und Olah, C.: Experiments in Handwriting with a Neural Network, Distill, http://doi.org/10.23915/distill.00004, aufgerufen am 03.01.2020, (2016)

[ChVi18] Chavan, H. G. und Vikhar, P. A.: A Survey: Offline Handwritten Signature Recognitionsystem, In: Multidisciplinary Journal of Research in Engineering and Technology (MJRET2018), Volume 5, Issue3&4, pp 08-15, (2018)

[CKZ+06] Cheung, K.-H., Kong, A., Zhang, D., Kamel, M. und You, J.: Revealing the secret of facehashing. In Proceedings of the 2006 international conference on Advances in Biometrics (ICB'06), David Zhang and Anil K. Jain (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 106-112, (2006)

[CMLM07] Cappelli, R., Maio, D., Lumini, A. und Maltoni, D.: Fingerprint Image Reconstruction from Standard Templates, IEEE Transaction on Pattern Analysis and Machine Intelligence 29, pp. 1489-1503, (2007)

[CMNT11] Coron, JS., Mandal, A., Naccache, D. und Tibouchi, M.: Fully Homomorphic Encryption over the Integers with Shorter Public Keys, In: Advances in Cryptology CRYPTO 2011, LNCS 6841, Springer, Berlin, Heidelberg, (2011)

[CrSS97] Cramer, R., Shoup, V. und Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme, In: Advances in Cryptology-EUROCRYPT 1997, Springer LNCS, pp. 103-118, (1997)

[CTNG04] Connie, T., Teoh, A., Goh, M. und Ngo, D.: “PalmHashing: A Novel Approach for Dual-Factor Authentication”, Pattern Analysis and Application, vol 7, no. 3, pp. 255-268, (2004)

[CTNG05] Connie, T., Teoh, A., Goh, M. und Ngo, D.: PalmHashing: a novel approach to cancelable biometrics, Information Processing Letter, vol. 93, no. 1, pp. 1-5, (2005)

[CuBa05] Cukic, B. und Bartlow, N.: Biomentric System Threats and Countermeasures: A Risk-Based Approach, Biometric Consortium Conference, September (2005)

[CWH+04] Cui, J., Wang, Y., Huang, J., Tan, T. und Sun, Z.: An iris image synthesis method based on pca and super-resolution, In: Proceedings of the IAPR International Conference on Pattern Recognition (ICPR), pp. 471-474, (2004)

[DDT+17] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J. und Pal, U.: SigNet: Convolutional Siamese network for writer independent offline signature verification, arXiv:1707.02131, (2017)

[DeBr85] De Bruyne, P.: Signature Verification using Holistic Measures, Comp Security, 4, pp. 309-315, (1985),

[DFES17] Diaz, M., Ferrer, M.A., Eskander, G.S. und Sabourin, R.: Generation of duplicated off-line signature images for verification systems, In: IEEE Transactions on Pattern Analysis and Machine Intelligence 39, pp. 951–964, (2017)

[DFI+19] Diaz, M., Ferrer, M. A., Impedovo, D., Malik, M. I., Pirlo, G. und Plamondon, R.: A Perspective Analysis of Handwritten Signature Technology, In: ACM Comput. Surv. 51, Artikelnr. 117, (2019)

[DHAZ11] Dobry, G., Hecht, R., Avigal, M. und Zigel, Y.: Supervector dimension reduction for efficient speaker age estimation based on the acoustic speech signal, IEEE Trans on Audio, Speech and Language Processing 19, pp. 1975-1985, (2011)

[Diaz16] Diaz, M.: Synthetic Signature Generation for Automatic Signature Verification, Dissertation, Universidad de Las Palmas de Gran Canaria, (2016)

[DIN19] DIN 16507-2:2019-09, Schriften - Schriftgrößen - Teil 2: Textverarbeitung, Mediengestaltung und verwandte Techniken (2019)

[DjPl09] Djioua, M. und Plamondon, R.: A new algorithm and system for the characterization of handwriting strokes with delta-lognormal parameters, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (11), pp. 2060-2072, (2009)

[DLM+98] Doddington, G., Liggett, W., Martin, A., Przybocki, M. und Reynolds, D.: Sheep, Goats, Lambs and Wolves: A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation. In CD-ROM Proceedings of the Fifth International Conference on Spoken Language Processing (ICSLP), Sydney, Australien, (1998)

[DSHG03] Derakhshani, R., Schuckers, S.A.C., Hornak, L.A. und Gorman, L.O.: Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners, Pattern Recognition, vol. 36, pp. 383-396, (2003)

[Dude20] Dudenredaktion (o. J.): „krypto“ auf Duden online, URL: https://www. duden.de/node/150139/revision/150175, Abrufdatum: 11.03.2020, (2020)

[Dude20a] Dudenredaktion (o. J.): „-logie“ auf Duden online, https://www. duden.de/node/129846/revision/129882, Abrufdatum: 11.03.2020, (2020)

[DuMi15] Ducas L. und Micciancio D.: FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second, Advances in Cryptology, EUROCRYPT 2015, LNCS 9056, Springer Verlag, Berlin, Heidelberg, (2015)

[ElMo09] Elahen, D. und Mohsen, E.M.: On-line signature verification using ANFIS. Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, Sept. 4-6, Dubrovnik, Croatia, pp: 546-549, (2009)

[FaFM08] Fard, M.A.M., Fard, M.E.M. und Mozayani, N.A.: A new on-line signature verification by spatio- temporal neural network, IEEE International Conference on Intelligence and Security Informatics, pp.233-235, (2008)

[Fail11] Failla, P.: Privacy-Preserving Processing of Biometric Template by Homomorphic Encryption, Phd Thesis, University of Siena, (2011)

[FCD+18] Ferrer, M.A., Chanda, S., Diaz, M., Banerjee, C.K., Majumdar, A., Carmona-Duarte, C., Acharya, P. und Pal, U.: Static and dynamic synthesis of Bengali and

Devanagari signatures, In: IEEE Transactions on Cybernetics 48, pp. 2896–2907, (2018)

[FeHu98] Ferguson, P. und Huston, G.: What is a VPN?, Cisco Systems, Tech. Rep., (1998)

[FFFG07] Fernandez, F. A., Fairhurst, M. C., Fierrez, J. und Garcia, J. O.: Impact of Signature Legibility and Signature type in Off-line Signature Verification, Biometrics Symposium, pp. 1-6, (2007)

[FGO+10] Fierrez, J., Galbally, J., Ortega-Garcia, J., Freire, M.R., Alonso-Fernandez, F., Ramos, D., Toledano, D. T., Gonzalez-Rodriguez, J., Siguenza, J. A., Garrido-Salas, J., Anguiano, E., Gonzalez-de-Rivera, G., Ribalda, R., Faundez-Zanuy, M., Ortega, J. A., Cardenoso-Payo, V., Viloria, A., Vivaracho, C. E., Moro, Q. I., Igarza, J. J., Sanchez, J., Hernaez, I., Orrite-Urunuela, C., Martinez Conteras, F. und Gracia-Roche, J. J.: BiosecurID: a multimodal biometric database, In: Pattern Analysis and Applications, 13(2), pp. 235-246, (2010)

[FHKS14] Freiermuth K., Hromkovič J., Keller L., Steffen B.: Die Suche nach Sicherheit und modulares Rechnen, In: Einführung in die Kryptologie, pp. 29-71, Springer Vieweg, Wiesbaden, (2014)

[FLT+02] Fang, B., Leung, C. H., Tang, Y. Y., Kwok, P. C. K., Tse, K. W. und Wong, Y. K.: Offline signature verification with generated training samples, IEE Proceedings - Vision, Image and Signal Processing, pp. 85–90, (2002)

[FMV+12] Ferrer, M., Morales, A., Vargas-Bonilla, J., Lemos, I. und Quintero, M.: Is It Possible to Automatically Identify Who Has Forged My Signature?: Approaching to the Identification of a Static Signature Forger, In 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 175-179, 10.1109/DAS.2012.47, (2012)

[FNL+05] Fierrez-Aguilar, J., Nanni, L., Lopez-Penalba, J., Ortega-Garcia, J. und Maltoni, D.: An on-line signature verification system based on fusion of local and global information. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546. Springer, Heidelberg, (2005)

[Fran09] Franke, D.: Diplomarbeit: Analyse der Interklassenreproduzierbarkeit von Handschriften Merkmalen eines BioHashes, Otto-von-Guericke-Universität Magdeburg: Fakultät für Informatik, Institut für Technische und Betriebliche Informationssystem, Arbeitsgruppe Multimedia and Security, (2009)

[FrPf98] Franz, E. und Pfitzmann, A.: Einführung in die Steganographie und Ableitung eines neuen Stegoparadigmas, Informatik-Spektrum 21, (1998)

[FrSV06] Frias-Martinez, E., Sanchez, A. und Velez, J.: Support vector machines versus multilayer perceptrons for efficient off-line signature recognition, Engineering Applications of Artificial Intelligence, pp. 693 – 704, (2006)

[GaFO07] Galbally, J., Fierrez, J. und Ortega-Garcia, J.: Bayesian hill-climbing attack and its application to signature verification. In: Proc. IAPR International Conference on Biometrics, ICB Bd. 4642, Springer, pp. 386-395, (2007)

[Galb09] Galbally, J.: Vulnerabilities and Attack Protection in Security Systems Based on Biometric Recognition, Phd Thesis, Universidad Autonoma de Madrid, (2009)

[GaMF13] Galbally J., Martinez-Diaz M. und Fierrez J.: Aging in Biometrics: An Experimental Analysis on On-Line Signature. PLoS ONE 8(7): e69897, (2013)

[GCFO09] Galbally, J., Carballo, S., Fierrez, J. und Ortega-Garcia, J.: Vulnerability Assessment of Fingerprint Matching Based on Time Analysis, in Proc. Biometric ID

Management and Multimodal Communication, BioID, Springer LNCS-5707, pp. 285-292, Madrid, (2009)

[GDF+15] Galbally, J., Diaz-Cabrera, M., Ferrer, M. A., Gomez-Barrero, M., Morales, A., und Fierrez, J.: On-line signature recognition through the combination of real dynamic data and synthetically generated static data. Pattern Recognition, pp. 2921 – 2934, (2015)

[Gent09] Gentry, C.: A fully homomorphic encryption scheme, Phd Thesis, Stanford University, (2009)

[Gent09a] Gentry, C.: Fully homomorphic encryption using ideal lattices, In Symposium of Theory of Computing-STOC, ACM, pp. 169-178, (2009)

[GFM+09] Galbally, J., Fierrez, J., Martinez-Diaz, M. und Ortega-Garcia, J.: Improving the enrollment in dynamic signature verification with synthetic samples, In: Proceedings of the IAPR International Conference on Document Analysis and Recognition (ICDAR), (2009)

[GFOG11] Gomez-Barrero, M., Fierrez, J., Ortega-Garcia, J. und Galbally, J.: Hill-Climbing Attack Based on the Uphill Simplex Algorithm and its Application to Signature Verification. In: European Workshop on Biometrics and Identity Management, Springer Verlag, LNCS-6583, pp. 83-94, (2011)

[GGTF12] Gomez-Barrero M., Galbally J., Tome P. und Fierrez J.: On the Vulnerability of Iris-Based Systems to a Software Attack Based on a Genetic Algorithm, In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, CIARP 2012, Lecture Notes in Computer Science, vol 7441. Springer, Berlin, Heidelberg, pp 114-121, (2012)

[GMG+17] Gomez-Barrero, M., Maiorana, E., Galbally, J., Campisi, P. und Fierrez, J.: Multibiometric template protection based on Homomorphic Encryption, Pattern Recognition, Volume 67, pp. 149-163, (2017)

[GPF+12] Galbally, J., Plamondon, R., Fierrez, J. und Ortega-Garcia, J.: Synthetic on-line signature generation. Part I: Methodology and algorithms, Pattern Recognition, (2012)

[GPF+12a] Galbally, J., Plamondon, R., Fierrez, J. und Ortega-Garcia, J.: Synthetic on-line signature generation. Part II: Experimental validation, Pattern Recognition, (2012)

[Grav13] Graves, A.: Generating Sequences with Recurrent Neural Networks, arXiv preprint arXiv:1308.0850v5, (2013)

[GRG+13] Galbally, J., Ross, A., Gomez-Barrero, M., Fierrez, J. und Ortega-Garcia, J.: Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms, In: Computer Vision and Image Understanding, Volume 117, Issue 10, pp. 1512-1525, (2013)

[GuHH14] Guest, R. M., Hurtado, O. M. und Henniger, O.: Assessment of methods for image recreation from signature time-series data, *IET Biometrics*, vol. 3, no. 3, pp. 159-166, (2014)

[GuMc97] Gupta, G. und McCabe, A.: A Review of Dynamic Handwritten Signature Verification, James Cook University, Australia, (1997)

[Gupt06] Gupta, G. K.: The State of the Art in On-line Handwritten Signature Verification, Monash University, Australia, (2006)

[Guyo96] Guyon, I: Handwriting Synthesis from Handwritten Glyphs, In Proceedings of the Fifth International Workshop on Frontiers of Handwriting Recognition, pp. 309-312, (1996)

[GWM+01] Garris, M.D., Watson, C.I., McCabe, R.M. und Wilson, C.L.: User's Guide to Nist Fingerprint Image Software (nfis). Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, (2001)

[HaAB16] Haines, T., Aodha, O. und Brostow, G.: My Text in Your Handwriting. ACM Transactions on Graphics, (2016)

[Ham50] Hamming, R. W.: Error detecting and error correcting codes. In Bell System Technical Journal 29 (2): pp. 147–160, (1950)

[HaSO16] Hafemann, L.G., Sabourin, R. und Oliveira, L. S.: Writer-independent feature learning for offline signature verification using deep convolutional neural networks, In: Int. Joint Conf. on Neural Networks (IJCNN'16), IEEE, pp. 2576-2583, (2016)

[HaSO17] Hafemann, L.G., Sabourin, R. und Oliveira, L. S.: Learning features for offline handwritten signature verification using deep convolutional neural networks, In: Pattern Recognition 70, pp. 163–176, (2017)

[HeFr04] Henniger, O. und Franke, K.: Biometric User Authentication on Smart Cards by Means of Handwritten Signatures. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 547-554. Springer, Heidelberg, (2004)

[HGBN13] Hadid A., Ghahramani M., Bustard J. und Nixon M.: Improving Gait Biometrics under Spoofing Attacks, Image Analysis and Processing, ICIAP 2013, LNCS 8157, Springer Verlag, Berlin, Heidelberg, (2013)

[HLHI07] Hennebert, J., Loeffel, R., Humm, A. und Ingold, R.: A new forgery scenario based on regaining dynamics of signature. In Proceedings of the 2007 international conference on Advances in Biometrics (ICB'07), Springer-Verlag, Berlin, Heidelberg, pp. 366-375, (2007)

[HMHL12] Hasan, M., McLaren, M., Hamme, H. V. und Leeuwen, D. V.: Age estimation from telephone speech using i-vectors, In: Proc. of InterSpeech, pp. 1-4, (2012)

[Holl81] Hollerbach, J.M.: An oscillation theory of handwriting, Biological Cybernetics 39, pp. 139-156, (1981)

[HoLo98] Howard, J.D. und Longstaff, T.A.: A common language for computer security incidents, (1998)

[Howa97] Howard, J.D.: An Analysis of Security Incidents on the Internet, Doctorial Thesis, Carnegie Mellon University, Pittsburg, Pennsylvania, USA, (1997)

[HuYa97] Huang, K. und Yan, H.: Off-line signature verification based on geometric feature extraction and neural network classification, Pattern Recognition, pp. 9 - 17, (1997)

[HZKS+13] Hasselberg, A., Zimmermann, R., Kraetzer, C., Scheidat, T., Vielhauer, C. und Kümmel, K.: Security of Features Describing the Visual Appearance of Handwriting Samples Using the Bio-hash Algorithm of Vielhauer against an Evolutionary Algorithm Attack, Communications and Multimedia Security. CMS 2013. LNCS 8099, Springer, Berlin, Heidelberg (2013)

[ISOI05] ISO/IEC 19794-2:2005, Information Technology—Biometric Data Interchange Formats—Part 2: Finger Minutiae Data, (2005)

[JaFR08] Jain, A.K., Flynn, P. und Ross, A. A.: Handbook of biometrics, Springer Science+Business Media, New York, USA, (2008)

[JaGC02] Jain, A.K., Griess, F.D. und Connell, S.D.: On-line signature verification, In: Pattern Recognition 2002, pp. 2963 – 2972, (2002)

[JaNN08] Jain, A.K., Nandakumar, K. und Nagar, A.: Biometric Template Security. In: EURASIP Journal on Advances in Signal Processing, Article ID 579416, (2008)

[JaRP04] Jain, A.K., Ross, A. und Prabhakar, S.: An Introduction to Biometric Recognition, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, vol. 14, no. 1, pp. 4-20, (2004)

[JaRP06] Jain, A.K., Ross, A. und Prabhakar, S.: Biometrics: a tool for information security. In: IEEE Transactions on Information Forensics and Security, pp. 125-143, (2006)

[JoJa98] Johnson, N. F. und Jajodia, S.: Exploring Steganographiy: Seeing the Unseen, IEEE, Computing Practices, (1998)

[JuBS01] Justino, E., Bortolozzi, E. und Saburin, R.: Off-line Signature Verification Using HMM for Random, Simple and Skilled Forgeries, In: ICDAR 2001, vol.1, pp. 105-110, (2001)

[KaJS17] Kamiński B., Jakubczyk M. und Szufel P.: A Framework for Sensitivity Aanalysis of Decision Trees, In: Central European journal of operations research vol. 26, pp. 135-159, (2018)

[KaSE08] Kamel, N.S., Sayeed, S. und Ellis, G.A.: Glove-Based Approach to Online Signature Verification, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 6, pp. 1109-1113, (2008)

[KCZ+06] Kong, A., Cheung, K.-H., Zhang, D., Kamel, M. und You, J.: An analysis of BioHashing and its variants, Pattern Recognition, Volume 39, Issue 7, pp. 1359-1368, (2006)

[KhMT12] Khalajzadeh, H., Mansouri, M. und Teshnehlabm M.: Persian signature verification using convolutional neural networks, In: International Journal of Engineering Research and Technology 1, pp. 7-12, (2012)

[KhYa05] Kholmatov, A. und Yanikoglu, B.: Identity authentication using improved online signature verification method, In Pattern Recognition Letter 26, pp. 2400-2408, (2005).

[KhYa08] Kholmatov, A. und Yanikoglu, B.: An Individuality Model for Inline Signatures Using Global Fourier Descriptors, (2008)

[KiGS09] Kisku, D.R., Gupta, P. und Sing, J.K.: Fusion of Multiple Matchers using SVM for Offline Signature Identification, Communications in Computer and Information Science, Volume 58, pp. 201-208, (2009)

[KKES14] Karabat, C., Kiraz, M. S., Erdogan, H. und Savas, E.: THRIVE: Threshold Homomorphic Encryption Based Secure and Privacy Preserving Biometric Verification System, EURASIP Journal on Advances in Signal Processing, (2015)

[Klat80] Klatt, D.H.: Software for a cascade/parallel formant synthesizer, Journal Acoustic Society of America 67, pp. 971–995, (1980)

[KLK+03] Kang, H., Lee, B., Kim, H., Shin, D. und Kim, J.: A Study on Performance Evaluation of the Liveness Detection for Various Fingerprint Sensor Modules, In proceeding of: Knowledge-Based Intelligent Information and Engineering Systems, 7th International Conference, KES 2003, pp. 1245-1253, (2003)

[KoZK05] Kong A., Zhang D. und Kamel M.: A Study of Brute-Force Break-ins of a Palmprint Verification System, Audio- and Video-Based Biometric Person Authentication, AVBPA 2005, LNCS 3546, Springer, Berlin, Heidelberg, (2005)

[KSAV11] Kümmel, K.; Scheidat, T., Arndt, C. und Vielhauer, C.: Feature Selection by User Specific Feature Mask on a Biometric Hash Algorithm for Dynamic Handwriting, In 12th Joint IFIP TC6 and TC12 Conference on Communications and Multimedia Security, (2011)

[KuSC12] Kumar, R., Sharma, J. D. und Chanda, B.: Writer-independent off-line signature verification using surroundedness feature, In: Pattern Recognition Letters 33, pp. 301-308, (2012)

[KüVi10] Kümmel, K. und Vielhauer, C. Reverse-engineer Methods on a Biometric Hash Algorithm for Dynamic Handwriting, MM&Sec '10: Proceedings of the 12th ACM workshop on Multimedia and security, ACM, pp. 67-72, (2010)

[KüVi10a] Kümmel, K. und Vielhauer, C. Potentielle Rückführbarkeit eines biometrischen Hashes für Handschriften, D-A-CH Security 2010, pp. 66-77, (2010)

[KüVi11] Kümmel, K. und Vielhauer, C.: Biometric Hash Algorithm for Dynamic Handwriting Embedded on a Java Card, In: Third European Workshop on Biometrics and Smart Card (BioID2011), Springer LNCS, 2011, pp. 61-72, (2011)

[KüVi11a] Kümmel, K. und Vielhauer, C.: Experimentelle Machbarkeitsstudie eines Bio-Hash Algorithmus auf einer Java Card, D-A-CH Security 2011, pp. 445-456, (2011)

[KVS+10] Kümmel, K.; Claus Vielhauer; Tobias Scheidat; Dirk Franke und Dittmann, J. Handwriting Biometric Hash Attack: A Genetic Algorithm with User Interaction for Raw Data Reconstruction, 11th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security, pp. 178-190, (2010)

[Lani09] Lanitis, A.: A survey of the effects of aging on biometric identity verification, In: International Journal of Biometrics, pp. 34-52, (2009)

[LHW+18] Li, H., He, P., Wang, S., Rocha, A., Jiang, X. und Kot, A. C.: Learning generalized deep feature representation for face anti-spoofing, In: IEEE Transactions on Information Forensics and Security, (2018)

[LoRa05] Lopresti, D. P. und Raim, J. D.: The effectiveness of generative attacks on an online handwriting biometric, In Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication, pp. 1090-1099, (2005)

[LSRJ07] Ling, H., Soatto, S., Ramanathan, N. und Jacobs, D. W.: A study of face recognition as people age, In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), 18, (2007)

[LWWZ05] Lv, H., Wang, W., Wang, C. und Zhuo, Q.: Offline Chinese Signature Verification based on Support Vector Machines, Pattern Recognition Letters, vol. 26, no. 15, pp. 2390-2399, (2005)

[MaSV11] Makrushin A., Scheidat T. und Vielhauer C.: Handwriting Biometrics: Feature Selection Based Improvements in Authentication and Hash Generation Accuracy. In: Biometrics and ID Management, BioID 2011, Lecture Notes in Computer Science, vol 6583, Springer, Berlin, Heidelberg, (2011)

[McTr08] McCabe, A. und Trevathan, J.: Markov Model-based Handwritten Signature Verification‖, IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp. 173-179, (2008)

[MCYJ18] Mai, G., Cao, K., YUEN, P. C. und Jain, A. K.: On the Reconstruction of Face Images from Deep Face Templates, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2018)

[MFA+06] Martinez-Diaz, M., Fierrez-Aguilar, J., Alonso-Fernandez, F., Ortega-Garcia, J. und Siguenza, J.A.: "Hill-climbing and brute-force attacks on biometric systems: A case study in Match-on-Card fingerprint verification", In Proc. IEEE Intl. Carnahan Conf. on Security Technology, ICCST, pp. 151-159, (2006)

[MMJP03] Maltoni, D., Maio, D., Jain, A.K. und Prabhakar, S.: Handbook of Fingerprint Recognition, Springer, Berlin, Germany, (2003)

[MMJP09] Maltoni, D., Maio, D., Jain, A.K. und Prabhakar, S.: Handbook of Fingerprint Recognition‖, Second Edition, Springer, (2009)

[MMYH02] Matsumoto, T., Matsumoto, H., Yamada, K. und Hoshino, S.: Impact of Artificial "Gummy" Fingers on Fingerprint Systems, Optical Security and Counterfeit Deterrence Techniques IV, Proc. of SPIE Vol. 4677, pp. 275-289, (2002)

[MoEH07] Modi, S. K., Elliott S. J. und Hakil, K.: Impact of age groups on fingerprint recognition performance. In: Proc. IEEE Workshop on Automatic Identification Advanced Technologies (AutoID), pp. 1923, (2007)

[MoEl06] Modi, S. K. und Elliott, S. J.: Impact of image quality on performance: age comparison of young and elderly fingerprints. In: Proc. Int. Conf. on Recent Advances in Soft Computing (ICRASC), pp. 10-12, (2006)

[MuPe03] Munich, M. und Perona, P.: Visual identification by signature tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, pp. 200 - 217, (2003)

[NaJa07] Nandakumar, K. und Jain, A.K.: Multibiometric Template Security Using Fuzzy Vault, Biometrics: Theory, Applications and Systems, BTAS 2008, 2nd IEEE International Conference, pp. 1-6, (2008)

[NeTH94] Nelson, W., Turin, W. und Hastie, T.: Statistical Methods for On-line Signature Verification. International Journal of Pattern Recognition and Artificial Intelligence, pp 749-770, (1994)

[NgTG04] Ngo, D.C.L., Teoh, A.B.J. und Goh, A.: “Eigenspace-based face hashing”, in Proc. Of International Conference on Biometric Authentication (ICBA), pp. 195-199, Hong Kong, (2004)

[Obie06] Obied, A.: How to attack biometric systems in your spare time, Department of Computer Science, University of Calgary, November 2006 Internet: http:// ahmed.obied.net/research/papers/biometric.pdf, Unveröffentlicht, (2006)

[OFS+03] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.-J., Vivaracho, C., Escudero D. und Moro, Q.-I.: MCYT baseline corpus: a bimodal biometric database. IEE Proc. Vis. Image Signal Process. Vol. 150, 395–401, (2003)

[OKBS97] de Oliveira, C., A Kaestner, C., Bortolozzi, F. und Sabourin, R.: Advances in Document Image Analysis: First Brazilian Symposium, BSDIA’97 Curitiba, Brazil, pp 283–298, Springer Berlin Heidelberg, Berlin, Heidelberg (1997)

[OzSK05] Ozgunduz, E., Senturk, T. und Karsligil, E.: Off-line signature verification and recognition by support vector machine‖, Paper presented at the European Signal Processing Conference, (2005)

[PaBP11] Pal, S., Blumenstein, M. und Pal U.: Automatic off-Line Signature Verification Systems: A Review. In: IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET), pp. 20-27, (2011)

[PaCz06] Pacut, A. und Czajka, A.: Aliveness detection for iris biometrics. Proc. IEEE Intl. Carnahan Conf. on Security Technology, ICCST 2006, pp. 122-129, (2006)

[Pail99] Paillier, P.: Public-key cryptosystems based on composite degree residousity classes. In: Advances in Cryptology-EUROCRYPT 1999, Springer LNCS, pp.223-238, (1999)

[PaTN04] Pang, Y.H., Teoh, A.B.J und Ngo, D.C.L.: Palmprint based cancelable biometric authentication system, International Journal of Signal Processing, vol. 1, no. 2, pp. 98-104, (2004)

[PiCL89] Pinto, N.B., Childers, D.G.und Lalwani, A.L.: Formant speech synthesis: improving production quality, IEEE Transactions on Acoustics, Speech and Signal Processing 37, pp. 1870–1887, (1989)

[PlLe04] Leclerc, F. und Plamondon, R.: Automatic Signature Verification: The State Of the Art - 1989 - 1993, International Journal of Pattern Recognition and Artificial Intelligence, pp. 643-660, (1994)

[PlLo89] Plamondon, R. und Lorette, G.: Automatic signature verification and writer identification - the state of the art, In: Pattern Recognition, pp. 107-131, (1989)

[Pope07] Popel, D. V.: Signature analysis, verification and synthesis in pervasive environments, Kapitel: Synthesis and Analysis in Biometrics, World Scientific, pp. 31 – 64, (2007)

[PuKe00] Van der Putte, T. und Keuning, J.: Biometrical Fingerprint Recognition: Don't get your Fingers Burned, IFIP TC8/WG8.8, Fourth Working Conference on Smart Card Research and Advanced Applications, pp. 289-303, (2000)

[RaCB01] Ratha, N. K., Connell, J. und Bolle, R. M.: An analysis of minutiae matching strength Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'01), Halmstad, Schweden, pp. 223-228, (2001)

[RaCB01a] Ratha, N. K., Connell, J. und Bolle, R. M.: Enhancing security and privacy in biometrics-based authentication systems, IBM Systems Journal, pp. 614-634, (2001)

[RaCh06] Ramanathan, N. und Chellappa, R.: Face verification across age progression, IEEE Trans on Image Processing 15, pp. 3349-3361, (2006)

[RaGF07] Rabasse, C., Guest, R. und Fairhurst, M.: A method for the synthesis of dynamic biometric signature data. Ninth International Conference on Document Analysis and Recognition (ICDAR), volume 1, pp 168–172, (2007)

[RaGF08] Rabasse, C., Guest, R. und Fairhurst, M.: A new method for the synthesis of signature data with natural variability. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, pp. 691–699, (2008)

[RaYM16] Rantzsch, H., Yang, H. und Meinel, C.: Signature embedding: Writer independent offline signature verification with deep metric learning, In: Int. Symposium on Visual Computing, Springer, pp. 616-625, (2016)

[RFMR09] Rattani, A., Freni, B., Marcialis, G. L. und Roli, F.: Template update methods in adaptive biometric systems: a critical review, In: Proc. IAPR/IEEE Int. Conf. on Biometrics (ICB), Springer LNCS-5558, pp. 847-856, (2009)

[RiRK18] Riaz, N., Riaz, A. und Khan, S.: Biometric template security: an overview, Sensor Review, Vol. 38 No. 1, pp. 120-127, (2018)

[RTA+08] Ruiz-Albacete V., Tome-Gonzalez P., Alonso-Fernandez F., Galbally J., Fierrez J. und Ortega-Garcia J.: Direct Attacks Using Fake Images in Iris Verification,

Biometrics and Identity Management, BioID 2008, LNCS 5372, Springer Verlag, Berlin, Heidelberg, (2008)

[SaBK06] Sayeed, S., Besar, R. und Kamel, N.S.: Dynamic signature verification using sensor based data glove, Proceedings of 8th International Conference on Signal Processing, IEEE Press, pp. 2387-2390, (2006)

[SaKB09] Sayeed, S., Kamel N.S. und Besar, R.: A sensor-based approach for dynamic signature verification using data glove. Signal Process. International Journal pp. 1-10, (2009)

[SaKB09a] Sayeed, S., Kamel N.S. und Besar, R.: A novel approach to dynamic signature verification using sensor-based data glove. American Journal Applied Sciences, Volume 6, Issue 2, pp. 233-240, (2009)

[SBD+09] Scheidat, T., Biermann, M., Dittmann, J., Vielhauer, C. und Kümmel K.: Multibiometric Fusion for Driver Authentication on the Example of Speech and Face, In: BioID_MultiComm'09 Proceedings of the 2009 joint COST 2101 and 2102 international conference on Biometric ID management and multimodal communication, Springer LNCS, pp. 220-227,(2009)

[Sche15] Scheidat, T.: Optimierung biometrischer Hash-Algorithmen für die dynamische Handschrift, Dissertation, Otto-von-Guericke-Universität Magdeburg, (2015)

[Schm09] Schmeh, K.: Kryptografie: Verfahren, Protokolle, Infrastrukturen, 4. Auflage, dpunkt- Verlag, Heidelberg, (2009)

[Schn96] Schneier, B.: Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. Wiley Computer Publishing, (1996)

[Schu02] Schuckers, S.: Spoofing and anti-Spoofing measures. Information Security Technical Report, pp. 56-62, (2002)

[Schw77] Schwefel, H.: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie. Birkhäuser Verlag Basel -ISBN: 3-7643-0876-1, (1977)

[ScKV12] Scheidat T., Kümmel K. und Vielhauer C.: Short term template aging effects on biometric dynamic handwriting authentication performance. In: Proc. Int. Conf on Communications and Multimedia Security. Springer LNCS 7394, pp 107-116, (2012)

[ScMV07] Scheidat, T., Makrushin, A. und Vielhauer, C.: Automatic Template Update Strategies for Biometrics, Technical Report, Otto-von-Guericke University Magdeburg, (2007)

[ScSa12] Schlöglhofer, R. und Sametinger, J.: Secure and usable authentication on mobile devices. In: ACM International Conference Proceeding Series. 10.1145/2428955.2429004, (2012)

[ScVD08] Scheidat, T., Vielhauer, C. und Dittmann, J.: Advanced Studies on Reproducibility of Biometric Hashes. In: Proceedings of First Workshop on Biometrics and Identity Management (BIOID 2008), pp. 150-159, Roskilde University, Denmark, (2008)

[ScVD09] Scheidat, T., Vielhauer, C. und Dittmann, J.: Biometric Hash Generation and User Authentication based on Handwriting using Secure Sketches. In: Proceedings of 6th International Symposium on Image and Signal Processing and Analysis (ISPA), (2009)

[Shan48] Shannon, C. E.: A Mathematical Theory of Communication. In: Bell System Technical Journal. Short Hills N.J. 27, pp. 379–423, 623–656, (1948)

[Shan49] Shannon, C. E.: Communication in the Presence of Noise. In: Proc. IRE, Vol. 37, No. 1, (1949)

[Shan49a] Shannon, C. E.: Communication Theory of Secrecy Systems. In: The Bell System Technical Journal, Vol. 28, No. 4, pp. 656-715, (1949)

[ShRo06] Shah, S. und Ross, A.: Generating synthetic irises by feature agglomeration, In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 317–320, (2006)

[SoAF16] Soleimani, A., Araabi, B. N. und Fouladi, K.: Deep multitask metric learning for offline signature verification, In: Pattern Recognition Letters 80, pp. 84-90, (2016)

[SoSu14] Song, M. und Sun, Z.: An immune clonal selection algorithm for synthetic signature generation, Mathematical Problems in Engineering, (2014)

[Sout02] Soutar, C.: "Biometric System Security," Secure, vol. 5, pp. 46-49, (2002)

[StSc02] Struif, B. und Scheuermann, D.: Smartcards with Biometric User Verification. In: Proceedings of IEEE International Conference on Multimedia and Expo 2002, vol. 2, pp. 589-592. Swiss Federal Institute of Technology, Lausanne, (2002)

[StYa10] Stefan, D. und Yao, D.: Keystroke-dynamics authentication against synthetic forgeries, 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 1-8, (2010)

[SuLM07] Sutcu, Y., Li, Q. und Memon, N.D.: Protecting Biometric Templates with Sketch: Theory and Practice. In: IEEE Transactions on Information Forensics and Security 2, pp. 503–512, (2007)

[TeNG04] Teoh, A.B.J., Ngo, D.C.L und Goh, A.: BioHashing: two factor authentication featuring fingerprint data and tokenised random number, Pattern Recognition, vol. 37, pp. 2245-2255, (2004)

[TeNG04a] Teoh, A.B.J., Ngo, D.C.L und Goh, A.: An integrated dual factor authenticator based on the face data and tokenised random number, in Proc. of International Conference on Biometric Authentication (ICBA), pp. 117-123, Hong Kong, (2004)

[TeNG04b] Teoh, A.B.J., Ngo, D.C.L und Goh, A.: "Personalised cryptographic key generation based on FaceHashing", Computers and Security Journal, vol. 23, no. 7, pp. 606-614, (2004)

[TeNg05] Teoh, A.B.J. und Ngo, D.C.L.: Cancellable biometrics featuring with tokenized random number, Pattern Recognition Letter Volume 26, Issue 10, pp. 1454-1460, (2005)

[ThKZ02] Thalheim, L., Krissler, J. und Ziegler, P.-M.: Body Check: Biometric Access Protection Devices and their Programs Put to the Test, c't Zeitschrift, pp. 114-121, November (2002)

[ToMa15] Tome, P. und Marcel, S.: On the vulnerability of palm vein recognition to spoofing attacks, International Conference on Biometrics, ICB 2015, pp. 319-325, (2015)

[TrGP13] Troncoso-Pastoriza, J. R., González-Jiménez, D. und Pérez-González, F.: Fully Private Noninteractive Face Verification, IEEE Transactions on Information Forensics and Security, vol. 8, no. 7, pp. 1101-1114, (2013)

[TVFO18] Tolosana, R., Vera-Rodriguez, R., Fierrez, J. und Ortega-Garcia, J.: Exploring recurrent neural networks for online handwritten signature biometrics, In: IEEE Access, pp. 1-11, (2018)

[UlJa04] Uludag, U. und Jain, A.: Attacks on biometric systems: a case study in fingerprints. In Proceedings of SPIE Seganography and Watermarking of Multimedia Contents VI, vol. 5306, pp. 622-633, (2004)

[VaBu03] Varga, T. und Bunke, H.: Generation of synthetic training data for an HMM-based handwriting recognition system. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 618–622, (2008)

[VaKB05] Varga, T., Kilchhofer, D. und Bunke, H.: Template-based synthetic handwriting generation for the training of recognition systems, In: Proceedings of the International Graphonomics Society, (2005)

[VanD10] Van Dijk, M., Gentry, C., Halevi, S. und Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers, In: Advances in Cryptology, EUROCRYPT 2010, LNCS 6110, Springer, Berlin, Heidelberg, (2010)

[VanL04] Van Laerhoven, K.: Basic Statistics and Metrics for Sensor Analysis; http://www.comp.lancs.ac.uk/~kristof/research/notes/basicstats/, aufgerufen am 16.07.2014, (2014)

[VeNC05] Veres, G., Nixon, M. und Carter, J.: Model-based approaches for predicting gait changes over time, In: Proc. IEEE Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 325-330, (2005)

[VeSa11] Venugopalan, S. und Savvides, M.: How to generate spoofed irises from an iris code template, In: IEEE Trans. on Information Forensics and Security 6, pp. 385-394, (2011)

[Viel06] Vielhauer, C.: Biometric User Authentication for IT Security: From Fundamentals to Handwriting, Springer, New York, (2006)

[ViSM02] Vielhauer, C., Steinmetz, R. und Mayerhöfer, A.: Biometric Hash based on Statistical Features of Online Signatures, In: Proc. of the IEEE International Conference on Pattern Recognition (ICPR), (2002)

[Waym1999] Wayman, J.L.: Technical Testing and Evaluation of Biometric Identification Devices, In A.K. Jain et al. (Eds.), Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, Boston, MA, USA, pp.345-368, (1999)

[Wern06] Werner, M.: Nachrichtenübertragungstechnik: Analoge und digitale Verfahren mit modernen Anwendungen, Vieweg Verlag, Wiesbaden, (2006)

[West17] Westernhagen, O.: Google Transparenzbericht: HTTPS-Traffic nimmt weltweit zu, heise Verlag, heise security, https://heise.de/-3870427, aufgerufen am 25.10.2018, (2017)

[WhDV18] Whiskerd, N., Dittmann, J. und Vielhauer, C.: A Requirement Analysis for Privacy Preserving Biometrics in View of Universal Human Rights and Data Protection Regulation, In: 26th European Signal Processing Conference (EUSIPCO), Rome, pp. 548-552, (2018)
doi: 10.23919/EUSIPCO.2018.8553045

[WSOS04] Wiehe, A., Søndrol, T., Olsen, O.K. und Skarderud, F.: Attacking Fingerprint Sensors, Technical Report, NISLAB Authentication Laboratory, Gjovik University College, (2004)

[YaGh17] Yahyatabar, M. E. und Ghasemi, J.: Online signature verification using double-stage feature extraction modelled by dynamic feature stability experiment, In: IET Biometrics 6, pp. 393-401, (2017)

[YaSK14] Yasuda, M., Shimoyama, T. und Kogure, J.: Secret computation of purchase history data using somewhat homomorphic encryption, Pacific Journal of Mathematics for Industry, (2014)

[YCX+04] Yeung, D., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T. und Rigoll, G.: SVC2004: First international signature verification competition, In: Proc. International Conference on Biometric Authentication. pp. 16–22, (2004)

[YNTK05] Yamazaki, Y., Nakashima, A., Tasaka, K. und Komatsu, N.: A study on vulnerability in on-line writer verification system. In: Document Analysis and Recognition, Eighth International Conference on, Vol. 2, pp. 640- 644, (2005)

[ZhLC16] Zhang, Z., Liu, X. und Cui, Y.: Multi-phase offline signature verification system using deep convolutional generative adversarial networks, In: 9th International Symposium on Computational Intelligence and Design (ISCID'16), Vol. 2. pp. 103-107, (2016)

[ZuSC07] Zuo, J., Schmid, Z., und Chen, X.: On generation and analysis of synthetic iris images, IEEE Transactions on Information Forensics and Security 2, pp. 77-90, (2007)

10 Publikationen des Autors

[HZKS+13] Hasselberg, A., Zimmermann, R., Kraetzer, C., Scheidat, T., Vielhauer, C. und Kümmel, K.: Security of Features Describing the Visual Appearance of Handwriting Samples Using the Bio-hash Algorithm of Vielhauer against an Evolutionary Algorithm Attack, Communications and Multimedia Security. CMS 2013. LNCS 8099, Springer, Berlin, Heidelberg (2013)

[KSAV11] Kümmel, K.; Scheidat, T., Arndt, C. und Vielhauer, C.: Feature Selection by User Specific Feature Mask on a Biometric Hash Algorithm for Dynamic Handwriting, In 12th Joint IFIP TC6 and TC12 Conference on Communications and Multimedia Security, (2011)

[KSVD12] Kümmel, K., Scheidat, T., Vielhauer, C. und Dittmann, J.: Reverse Engineering als Werkzeug zur biometrischen Sicherheitsanalyse, D-A-CH Security, (2012)

[KSVD12a] Kümmel, K., Scheidat, T., Vielhauer, C. und Dittmann, J.: Feature Selection on Handwriting Biometrics: Security Aspects of Artificial Forgeries, In 13th Joint IFIP TC6 and TC12 Conference on Communications and Multimedia Security, pp. 16-25, (2012)

[KüVi10] Kümmel, K. und Vielhauer, C. Reverse-engineer Methods on a Biometric Hash Algorithm for Dynamic Handwriting, MM&Sec '10: Proceedings of the 12th ACM workshop on Multimedia and security, ACM, pp. 67-72, (2010)

[KüVi10a] Kümmel, K. und Vielhauer, C. Potentielle Rückführbarkeit eines biometrischen Hashes für Handschriften, D-A-CH Security 2010, pp. 66-77, (2010)

[KüVi11] Kümmel, K. und Vielhauer, C.: Biometric Hash Algorithm for Dynamic Handwriting Embedded on a Java Card, In: Third European Workshop on Biometrics and Smart Card (BioID2011), Springer LNCS, 2011, pp. 61-72, (2011)

[KüVi11a] Kümmel, K. und Vielhauer, C.: Experimentelle Machbarkeitsstudie eines Bio-Hash Algorithmus auf einer Java Card, D-A-CH Security 2011, pp. 445-456, (2011)

[KVS+10] Kümmel, K.; Claus Vielhauer; Tobias Scheidat; Dirk Franke und Dittmann, J. Handwriting Biometric Hash Attack: A Genetic Algorithm with User Interaction for Raw Data Reconstruction, 11th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security, pp. 178-190, (2010)

[SBD+09] Scheidat, T., Biermann, M., Dittmann, J., Vielhauer, C. und Kümmel K.: Multibiometric Fusion for Driver Authentication on the Example of Speech and Face, In: BioID_MultiComm'09 Proceedings of the 2009 joint COST 2101 and 2102 international conference on Biometric ID management and multimodal communication, Springer LNCS, pp. 220-227, (2009)

[ScKV12] Scheidat T., Kümmel K. und Vielhauer C.: Short term template aging effects on biometric dynamic handwriting authentication performance. In: Proc. Int. Conf on Communications and Multimedia Security. Springer LNCS 7394, pp 107-116, (2012)

11 Abbildungsverzeichnis

12 Tabellenverzeichnis

13 Anhang

Anlage 1 Liste aller verwendeten Merkmale im Biometrischen Hash Algorithmus für die dynamische Handschrift [Sche15]

ID	Merkmalsname	Beschreibung
n_1	Ttotal	Dauer der Aufzeichnung
n_2	SampleCount	Anzahl an Samplepunkten
n_3	AspectRatio	Höhe * 1000 DIV Breite
n_4	VxAbsolute	Durchschnittsgeschwindigkeit in X * 1000 Pixel / ms
n_5	VyAbsolute	Durchschnittsgeschwindigkeit in Y * 1000 Pixel / ms
n_6	SegmentCount	Anzahl an Zeiträumen, in denen der Stift aufgesetzt ist
n_7	VxMin	Minimum an Geschwindigkeit in X Richtung
n_8	VxMax	Maximum an Geschwindigkeit in X Richtung
n_9	VyMin	Minimum an Geschwindigkeit in Y Richtung
n_{10}	VyMax	Maximum an Geschwindigkeit in Y Richtung
n_{11}	CentroidX	Horizontaler Massepunkt der Stiftpositionen
n_{12}	CentroidY	Vertikaler Massepunkt der Stiftpositionen
n_{13}	CentroidDist	Distanz des Massepunkts vom Ursprung
n_{14}	MaxPressure	Maximal aufgezeichneter Druck (falls nicht vorhanden -1)
n_{15}	CentroidX_SN	Horizontaler Massepunkt mit normalisierter Hüllboxbreite * 1000
n_{16}	CentroidY_SN	Vertikaler Massepunkt mit normalisierter Hüllboxbreite * 1000
n_{17}	CentroidDist_SN	Distanz des Massepunkts vom Ursprung normalisiert mit Hüllboxdurchmesser * 1000
n_{18}	CentroidAzimuth_SN	Horizontaler Richtungswinkel zum Masseursprung normalisiert mit PI/2 * 1000
n_{19}	MaxAltitude	Maximaler Wert für den Höhenwinkel des Stiftes
n_{20}	MinAltitude	Minimaler Wert für den Höhenwinkel des Stiftes
n_{21}	MaxAzimuth	Maximaler Wert für den Richtungswinkel des Stiftes

ID	Merkmalsname	Beschreibung
n_{22}	MinAzimuth	Minimaler Wert für den Richtungswinkel des Stiftes
n_{23}	AvgPressure	Durchschnittsdruck relativ zum Maximaldruck * 1000
n_{24}	AvgAzimuth	Durchschnittlicher Richtungswinkel des Stiftes
n_{25}	AvgAltitude	Durchschnittlicher Höhenwinkel des Stiftes
n_{26}	Vx_TN	Normalisierte Durchschnittsgeschwindigkeit in X Richtung / Maximal Geschwindigkeit in X Richtung
n_{27}	Vy_TN	Normalisierte Durchschnittsgeschwindigkeit in Y Richtung / MaximalGeschwindigkeit in Y Richtung
n_{28}	TpenUP	Dauer der gesamten Absetzzeit des Stiftes
n_{29}	RatioTPenUpEnDown	Verhältnis zwischen TpenUP und Gesamtzeit * 1000
n_{30}	NoSamples	Anzahl von Abtastpunkten (Samples)
n_{31}	PathLength	Länge des Weges der Aufzeichnung in Pixeln
n_{32}	PixelCountR1C1	Anzahl an Samplepunkten, Reihe 1, Zeile 1
n_{33}	PixelCountR1C2	Anzahl an Samplepunkten, Reihe 2, Zeile 1
n_{34}	PixelCountR1C3	Anzahl an Samplepunkten, Reihe 3, Zeile 1
n_{35}	PixelCountR1C4	Anzahl an Samplepunkten, Reihe 4, Zeile 1
n_{36}	PixelCountR2C1	Anzahl an Samplepunkten, Reihe 1, Zeile 2
n_{37}	PixelCountR2C2	Anzahl an Samplepunkten, Reihe 2, Zeile 2
n_{38}	PixelCountR2C3	Anzahl an Samplepunkten, Reihe 3, Zeile 2
n_{39}	PixelCountR2C4	Anzahl an Samplepunkten, Reihe 4, Zeile 2
n_{40}	PixelCountR3C1	Anzahl an Samplepunkten, Reihe 1, Zeile 3
n_{41}	PixelCountR3C2	Anzahl an Samplepunkten, Reihe 2, Zeile 3
n_{42}	PixelCountR3C3	Anzahl an Samplepunkten, Reihe 3, Zeile 3
n_{43}	PixelCountR3C4	Anzahl an Samplepunkten, Reihe 4, Zeile 3
n_{44}	IntegralX	Numerisches Integral der normalisierten X Werte
n_{45}	IntegralY	Numerisches Integral der normalisierten Y Werte

ID	Merkmalsname	Beschreibung
n_{46}	AreaX1	Numerisches Integral der normalisierten X Werte im 1. Zeitabschnitt
n_{47}	AreaX2	Numerisches Integral der normalisierten X Werte im 2. Zeitabschnitt
n_{48}	AreaX3	Numerisches Integral der normalisierten X Werte im 3. Zeitabschnitt
n_{49}	AreaX4	Numerisches Integral der normalisierten X Werte im 4. Zeitabschnitt
n_{50}	AreaX5	Numerisches Integral der normalisierten X Werte im 5. Zeitabschnitt
n_{51}	AreaY1	Numerisches Integral der normalisierten Y Werte im 1. Zeitabschnitt
n_{51}	AreaY2	Numerisches Integral der normalisierten Y Werte im 2. Zeitabschnitt
n_{53}	AreaY3	Numerisches Integral der normalisierten Y Werte im 3. Zeitabschnitt
n_{54}	AreaY4	Numerisches Integral der normalisierten Y Werte im 4. Zeitabschnitt
n_{5}	AreaY5	Numerisches Integral der normalisierten Y Werte im 5. Zeitabschnitt
n_{56}	PenDPress	Durchschnittlicher normalisierter Druck während Stift aufgesetzt ist
n_{57}	PenUPress	Durchschnittlicher normalisierter Druck während Stift abgesetzt ist
n_{58}	BaselineAngle	Anstiegswinkel der Grundlinie des Samples
n_{59}	HistYZone1	Histogramm für Y Zone 1 in % * 100
n_{60}	HistYZone2	Histogramm für Y Zone 2 in % * 100
n_{61}	HistYZone3	Histogramm für Y Zone 3 in % * 100
n_{62}	AreaRatio1	Flächenverhältnis Konvexe Hülle zu Hüllbox * 1000
n_{63}	AreaRatio2	Flächenverhältnis Konvexer Hüllensegmente zu gesamter Konvexer Hülle * 1000
n_{64}	AreaRatio3	Flächenverhältnis Konvexer Hüllensegmente zu Hüllbox * 1000
n_{65}	PathRatio1	Verhältnis Umfang Konvexer Hülle zu Hüllbox * 1000
n_{66}	PathRatio2	Verhältnis Umfang Konvexer Hüllensegmente zu gesamter Konvexer Hülle * 1000
n_{67}	PathRatio3	Verhältnis Umfang Konvexer Hüllensegmente zu Hüllbox * 1000
n_{68}	HistXLeft	Histogramm für X im linken Bildbereich % * 100
n_{69}	HistXRight	Histogramm für X im rechten Bildbereich % * 100

ID	Merkmalsname	Beschreibung
n_{70}	NoMaximusX	Anzahl an Maxima in X Richtung
n_{71}	NoMinimusX	Anzahl an Minima in X Richtung
n_{72}	NoMaximusY	Anzahl an Maxima in Y Richtung
n_{73}	NoMinimusY	Anzahl an Minima in Y Richtung
n_{74}	NoMaximusRatio	Verhältnis Maxima in X Richtung vs. Maxima in Y Richtung * 1000
n_{75}	NoMinimusRatio	Verhältnis Minima in X Richtung vs. Minima in Y Richtung * 1000
n_{76}	NoIntersections	Anzahl an Kreuzungspunkten
n_{77}	NoIntersections_X1	Anzahl an Schnittpunkten mit Linie im 1. Viertel in X
n_{78}	NoIntersections_X2	Anzahl an Schnittpunkten mit Linie im 2. Viertel in X
n_{79}	NoIntersections_X3	Anzahl an Schnittpunkten mit Linie im 3. Viertel in X
n_{80}	NoIntersections_X4	Anzahl an Schnittpunkten mit Linie im 4. Viertel in X
n_{81}	NoIntersections_Y1	Anzahl an Schnittpunkten mit Linie im 1. Drittel in Y
n_{82}	NoIntersections_Y2	Anzahl an Schnittpunkten mit Linie im 2. Drittel in Y
n_{83}	NoIntersections_Y3	Anzahl an Schnittpunkten mit Linie im 3. Drittel in Y
n_{84}	NoIntersections_D1	Anzahl an Schnittpunkten mit Diagonale links oben zu rechts unten
n_{85}	NoIntersections_D2	Anzahl an Schnittpunkten mit Diagonale rechts oben zu links unten
n_{86}	StartEndRatio	Verhältnis Distanz Start/Endpunkt zu Pfadlänge * 1000
n_{87}	XmaxXminRatio	Verhältnis Distanz MinX/MaxX zu Pfadlänge * 1000
n_{88}	YmaxYminRatio	Verhältnis Distanz MinY/MaxY zu Pfadlänge * 1000
n_{89}	StartCentroidEndRatio	Verhältnis Distanz Start-/Masseschwerpunkt zu End-/Masseschwerpunkt * 1000
n_{90}	FmapX	Abbildung der Maxima/Minima in X auf eine Zahl
n_{91}	FmapY	Abbildung der Maxima/Minima in Y auf eine Zahl
n_{92}	FmapP	Abbildung der Maxima/Minima in P auf eine Zahl
n_{93}	FmapA	Abbildung der Maxima/Minima in A auf eine Zahl

ID	Merkmalsname	Beschreibung
n_{94}	IntRange	Abstand aller Absetzpunkte
n_{95}	DotCountMin	Anzahl an Punkten im Radius 1/3*Hüllboxfläche zum Punkt mit wenigsten Nachbarpunkten
n_{96}	DotCountMax	Anzahl an Punkten im Radius 1/3*Hüllboxfläche zum Punkt mit meisten Nachbarpunkten
n_{97}	DotCountAvg	Anzahl an Punkten im Radius 1/3*Hüllboxfläche zum Punkt mit durchschnittlichsten Nachbarpunkten
n_{98}	Angle0_30	Durchschnittlicher Winkel für alle Schnittpunktwinkel zwischen 0-30°
n_{99}	Angle31_60	Durchschnittlicher Winkel für alle Schnittpunktwinkel zwischen 31-60°
n_{100}	Angle61_90	Durchschnittlicher Winkel für alle Schnittpunktwinkel zwischen 61-90°
n_{101}	AngleCount30	Anzahl an Winkeln für alle Schnittpunktwinkel zwischen 0-30°
n_{102}	AngleCount60	Anzahl an Winkeln für alle Schnittpunktwinkel zwischen 31-60°
n_{103}	AngleCount90	Anzahl an Winkeln für alle Schnittpunktwinkel zwischen 61-90°
n_{104}	MaxPenUpDist	Maximaler Abstand zwischen zwei Schriftzügen (Stift hoch genommen)
n_{105}	MinPenUpDist	Minimaler Abstand zwischen zwei Schriftzügen (Stift hoch genommen)
n_{106}	AvgPenUpDist	Durchschnittlicher Abstand zwischen zwei Schriftzügen (Stift hoch genommen)
n_{107}	areas	Abgebildete Größe der eingeschlossenen Flächen
n_{108}	Cluster0X	Normalisiert X - Koordinate des Clusters 0
n_{109}	Cluster0Y	Normalisiert Y - Koordinate des Clusters 0
n_{110}	Cluster1X	Normalisiert X - Koordinate des Clusters 1
n_{111}	Cluster1Y	Normalisiert Y - Koordinate des Clusters 1
n_{112}	Cluster2X	Normalisiert X - Koordinate des Clusters 2
n_{113}	Cluster2Y	Normalisiert Y - Koordinate des Clusters 2
n_{114}	Cluster3X	Normalisiert X - Koordinate des Clusters 3
n_{115}	Cluster3Y	Normalisiert Y - Koordinate des Clusters 3
n_{116}	Cluster4X	Normalisiert X - Koordinate des Clusters 4
n_{117}	Cluster4Y	Normalisiert Y - Koordinate des Clusters 4

ID	Merkmalsname	Beschreibung
n_{118}	Cluster5X	Normalisiert X - Koordinate des Clusters 5
n_{119}	Cluster5Y	Normalisiert Y - Koordinate des Clusters 5
n_{120}	Cluster0XYAvgPrs	Durchschnittlicher Druck von xy Cluster 0
n_{121}	Cluster1XYAvgPrs	Durchschnittlicher Druck von xy Cluster 1
n_{122}	Cluster2XYAvgPrs	Durchschnittlicher Druck von xy Cluster 2
n_{123}	Cluster3XYAvgPrs	Durchschnittlicher Druck von xy Cluster 3
n_{124}	Cluster4XYAvgPrs	Durchschnittlicher Druck von xy Cluster 4
n_{125}	Cluster5XYAvgPrs	Durchschnittlicher Druck von xy Cluster 5
n_{126}	Cluster0Alti	Durchschnittlicher Höhenwinkel des Druck Clusters 0
n_{127}	Cluster1Alti	Durchschnittlicher Höhenwinkel des Druck Clusters 1
n_{128}	Cluster2Alti	Durchschnittlicher Höhenwinkel des Druck Clusters 2
n_{129}	Cluster3Alti	Durchschnittlicher Höhenwinkel des Druck Clusters 3
n_{130}	InflectionPointSpeed	Abgebildete Geschwindigkeit beim Wendepunkt eines Samples
n_{131}	PressureDeviation	Standardabweichung des Drucks

Anlage 2 Steckbriefe: Angriffsverfahren

1	**Angriff Nr.**	1						
2	**Biometrische Modalität**	Gesicht						
3	**Autoren/Quelle**	Thalheim et al / [ThKZ02]						
4	**Kurzbeschreibung**	Bei der Datenerfassung der Gesichtserkennungssoftware, welche eine normale Internetkamera als Sensor verwendet, werden ein aufgezeichnetes Video bzw. Bilder einer Person vor dem Sensor präsentiert (Notebook dient als Abspielgerät). Die verwendete Verifikationssoftware konnte mit dieser Methode relativ einfach getäuscht werden.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff. Er verwendet eine einfache Spoofing Methode und kann somit der Angriffsklasse AK 1.1 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis eines optischen Sensors basiert, kann es nicht auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett, technische Arbeitsweise siehe Abschnitt 4.2.1, für die Datenaufzeichnung verwendet. Jedoch ist es möglich, ein statisches handschriftenbasiertes Verifikationssystem mittels dieser Technik zu täuschen. Bei solch einem Verfahren werden Bilder von Handschriftendaten für eine Verifikation herangezogen. Da für dynamisch handschriftenbasierte Verifikation nicht anwendbar, wird das Gefahrenpotential als sehr gering (Stufe 1) eingestuft. Die Angriffsmethode ist jedoch recht einfach auf alle biometrischen Verifikationssysteme anwendbar, welche einen optischen Sensor (CMOS Kamera oder ähnliches) verwenden wie beispielsweise bei der Aufnahme der Iris oder von den Konturen der Hand. Weshalb der Angriff im Allgemein in Stufe 3 eingestuft werden muss, da er teils recht einfach umzusetzen ist, solange keine Gegenmaßnahmen (Lebenderkennung etc.) getroffen wurden.						
7	**Angriffsklasse**	AK 1.1	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	2						
2	**Biometrische Modalität**	Tippverhalten						
3	**Autoren/Quelle**	Deian Stefan und Danfeng (Daphne) Yao / [StYa10]						
4	**Kurzbeschreibung**	Stefan et al. haben zwei einfache Bots (GaussianBot and NoiseBot) entwickelt, welche das Tippverhalten verschiedener Personen simulieren können. Die Bots beziehen Informationen (Tastaturanschlagszeiten, Pause zwischen den Buchstaben etc.) von Personen, die in einer Userdatenbank registriert sind. Diese Daten stammen jedoch nicht von der anzugreifenden Person. Das eingegebene Wort bzw. die Zeichenfolge ist dem Bot bekannt, jedoch nicht das Eingabemuster (Tippverhalten). Dieses wird vom Bot anhand der Daten der anderen Nutzer ermittelt.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet hinter dem Sensor (Angriffspunkt 2) statt und ist dementsprechend ein indirekter Angriff, siehe Entscheidungsbaum. Diese Angriffstechnik basiert auf einem modifizierten Tampering. Hierbei wird nicht das biometrische Template der anzugreifenden Person manipuliert, sondern die biometrischen Templates anderer in der Datenbank registrierten Personen werden für die Erzeugung künstlicher biometrischer Daten herangezogen. Anschließend werden diese generierten Daten in das System am Angriffspunkt 2 eingespielt. Der Angriff wird entsprechend der Angriffsklasse AK 2.4.2 zugeordnet.						
6	**Adaptierbarkeit**	Das Verfahren setzt einen Algorithmus (Bot) ein, welcher auf Basis von biometrischen Daten realer Personen künstliche biometrische Daten generiert. Diese Methode zur Erzeugung von biometrischen Daten ist prinzipiell für alle biometrischen Verifikationssysteme als Angriffsstrategie vorstellbar. Somit auch auf ein handschriftenbasiertes Verifikationssystem. Die in diesem Angriff beschriebenen spezifischen Angriffstechniken (GaussianBot und NoisBot) sind zwar nicht auf jede biometrische Modalität direkt anwendbar, gleichwohl bildet die Strategie ein erhebliches Angriffspotential. Aus diesem Grund wird der Angriff sowohl für das Gefahrenpotential Handschrift als auch für das Allgemeine Gefahrenpotential als hoch (Stufe 4) eingeordnet.						
7	**Angriffsklasse**	AK 2.4.2	**8**	**Gefahrenpotential Handschrift**	4	**9**	**Allg. Gefahrenpotential**	4

1	**Angriff Nr.**	3						
2	**Biometrische Modalität**	Fingerabdruck						
3	**Autoren/Quelle**	Galbally et al. / [GCFO09]						
4	**Kurzbeschreibung**	Ein auf Fingerabdruck basierendes biometrisches Verifikationssystem (NFIS2 von NIST) wird einer Verarbeitungszeitanalyse unterzogen. Hierbei wird die Verarbeitungszeit für die Berechnung eines Vergleichswertes gemessen und mit dem erzielten Vergleichswert (Matching-Score) verglichen. Dabei wurde festgestellt, dass je höher die Verarbeitungszeit ist, desto höher fällt auch der Vergleichswert aus. Es konnte eine klare Korrelation der beiden Werte ermittelt werden.						
5	**Klassifikation**	Die Angriffstechnik ist eine indirekte Angriffsmethode und setzt auf Angriffspunkt 5 auf. Da die Verarbeitungszeit des Matchers/Entscheiders gemessen wird (AP5) basiert dieser Angriff auf Seitenkanal – Information. Entsprechend kann dieser Angriff der Angriffsklasse AK 2.1.5 zugeordnet werden.						
6	**Adaptierbarkeit**	Im Allgemein ist diese Angriffsmethode auf jedes Verifikationssystem anwendbar. Hierfür muss jedoch teils Zugang zu einigen Systemkomponenten bestehen, um beispielsweise den Vergleichswert (Matching-Score) auszulesen. Eine Korrelation zwischen der Verarbeitungszeit zur Berechnung des Vergleichswerts und der Vergleichswert an sich besteht nicht unbedingt für jedes Verifikationsverfahren und muss evaluiert werden. Aus diesem Grund wird das Gefahrenpotential für Handschriften und im Allgemein als erheblich (Stufe 3) eingestuft						
7	**Angriffsklasse**	AK 2.1.5	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	4						
2	**Biometrische Modalität**	Handschrift						
3	**Autoren/Quelle**	Galbally et al. / [GaFO07]						
4	**Kurzbeschreibung**	Ein Angreifer hat Zugriff auf ein auf Handschriften basierendes biometrisches Verifikationssystem und ist in der Lage, Merkmalsvektoren in das System einzuspielen. Des Weiteren hat der Angreifer Zugriff auf das Ergebnis des Vergleichers (Matchingscore). Unter Verwendung dieser Daten und eines Hill-Climbing-Algorithmus generiert der Angreifer Merkmalsvektoren, um so Zugang zum System zu erhalten.						
5	**Klassifikation**	Die Angriffstechnik ist eine indirekte Angriffsmethode und setzt auf Angriffspunkt 4 und 8 auf. Die Merkmalsvektoren werden in Punkt 4 eingespielt und die Ergebnisse des Entscheiders am Punkt 8 entnommen. Entsprechend kann dieser Angriff den Angriffsklassen AK 2.2.2 und AK 2.2.8 zugeordnet werden.						
6	**Adaptierbarkeit**	Prinzipiell ist diese Angriffsmethode auf jedes Verifikationssystem anwendbar. Jedoch müssen hierfür Zugänge zu zwei Punkten im System bestehen, um diesen Angriff erfolgreich durchführen zu können. Aus diesem Grund wird das Gefahrenpotential für Handschriften und im Allgemein als erheblich (Stufe 3) eingestuft						
7	**Angriffsklasse**	AK 2.2.2 AK 2.2.8	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	5						
2	**Biometrische Modalität**	Gangart						
3	**Autoren/Quelle**	Hadid et al / [HGBN13]						
4	**Kurzbeschreibung**	Bei diesem Nachahmungsangriff versuchen Angreifer, die Gangart einer Person zu imitieren. Hierbei ist den Angreifern bekannt, welche Kleidung die Person bei der Aufzeichnung der Referenzdaten getragen hat. Die Angreifer kleiden sich entsprechend den Personen, welche sie imitieren möchten.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff, siehe Entscheidungsbaum. Er verwendet eine einfache Nachahmungsmethode mit zusätzlichen Informationen (Kleidung) und kann somit der Angriffsklasse AK 1.2 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis mehrerer optischer Sensoren basiert, kann es nicht direkt auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett für die Datenaufzeichnung verwendet. Jedoch ist es prinzipiell möglich auch eine Handschrift nachzuahmen, um somit ein System zu täuschen. Da jedoch die dynamische Handschrift eine sehr komplexe feinmotorische Handlung darstellt, kann sie nicht ohne weiteres imitiert werden. Aus diesem Grund wird das Gefahrenpotential auf die dynamische Handschrift als sehr gering (Stufe 1) eingestuft. Auch für die übrigen biometrischen Systeme kann dieses doch sehr spezielle Angriffsverfahren nicht direkt auf andere Verfahren angewendet werden. Da in der Regel biometrische Verifikationssysteme schon bei der Wahl der biometrischen Modalität u.a. die Eventualität des Nachahmens im Design berücksichtigen sollten. Aus diesem Grund wird das Gefahrenpotential des Angriffsverfahrens im Allgemeinen als sehr gering (Stufe 1) eingestuft.						
7	**Angriffsklasse**	AK 1.2	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	1

1	**Angriff Nr.**	6						
2	**Biometrische Modalität**	Iris						
3	**Autoren/Quelle**	Ruiz-Albacete et al / [RTA+08]						
4	**Kurzbeschreibung**	Ein auf der biometrischen Modalität Iris basierendes Verifikationssystem wird mittels gefälschter Bilder angegriffen. Die Irisbilder stammen dabei aus verschiedenen handelsüblichen Druckern. Die Arbeit von Ruiz-Albacete et al. zeigt, dass Verifikationssysteme ohne Lebenderkennung relativ einfach zu täuschen sind.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff. Er verwendet eine einfache Spoofing Methode und kann somit der Angriffsklasse AK 1.1 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis einen optischen Sensor basiert, kann es nicht direkt auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett für die Datenaufzeichnung verwendet. Aus diesem Grund wird das Gefahrenpotential auf Handschriftenbasierte Verifikationsverfahren als sehr gering (Stufe 1) eingeschätzt. Die Angriffsmethode ist jedoch recht einfach auf alle biometrischen Verifikationssystem anwendbar, welche einen optischen Sensor (CMOS-Kamera oder ähnliches) verwenden wie beispielsweise bei der Aufnahme von Gesichtern oder von den Konturen der Hand. Weshalb der Angriff im Allgemeinen in Stufe 3 eingestuft werden muss, da er teils recht einfach umzusetzen ist, solange keine Gegenmaßnahmen (Lebenderkennung etc.) getroffen wurden.						
7	**Angriffsklasse**	AK 1.1	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	7						
2	**Biometrische Modalität**	Venenverlauf einer Hand						
3	**Autoren/Quelle**	Tome und Marcel / [ToMa15]						
4	**Kurzbeschreibung**	Bei diesem Angriff verwenden die Autoren Kopien eines Venenverlaufs der Handfläche von 50 Personen (Angriffsdatenbank). Sie haben hierfür einen handelsüblichen Drucker verwendet und nachgewiesen, dass die Tinte durch den Infrarotscanner reflektiert wird und entsprechend für den Angriff verwendet werden kann. Mit den erstellten Kopien konnten Sie eine Falschakzeptanzrate von 65% erzeugen.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff. Er verwendet eine einfache Spoofing Methode und kann somit der Angriffsklasse AK 1.1 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis eines Infrarotsensors funktioniert, kann es nicht direkt auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett für die Datenaufzeichnung verwendet. Aus diesem Grund wird das Gefahrenpotential auf handschriftenbasierte Verifikationsverfahren als sehr gering (Stufe 1) eingeschätzt. Die Angriffsmethode ist jedoch recht einfach auf alle biometrischen Verifikationssystem anwendbar, welche einen Infrarotsensor verwenden. Weshalb der Angriff im Allgemein in Stufe 3 eingestuft werden muss, da er teils recht einfach umzusetzen ist, solange keine Gegenmaßnahmen (Lebenderkennung etc.) getroffen wurden.						
7	**Angriffsklasse**	AK 1.1	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	8						
2	**Biometrische Modalität**	Stimme						
3	**Autoren/Quelle**	Baroughi und Craver / [BaCr14]						
4	**Kurzbeschreibung**	In den von den Autoren vorgestellten Angriff werden künstlich erzeugte Geräusche im Hintergrund abgespielt, während ein Angreifer in das Mikrofon des Verifikationssystems spricht. Die künstlich erzeugten Geräusche wurden auf Basis der Arbeitsweise des Algorithmus erzeugt. Das System konnte in 95% der Fälle somit getäuscht werden.						
5	**Klassifikation**	Der hier durchgeführte Angriff findet am Sensor (Angriffspunkt 1) statt und ist dementsprechend ein direkter Angriff. Er verwendet eine Spoofing Methode und kann somit der Angriffsklasse AK 1.1 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren auf Basis eines Mikrofons funktioniert, kann es nicht direkt auf ein dynamisches handschriftenbasiertes Verifikationssystem angewendet werden, welches in der Regel ein spezielles Handschriftensignaturtablett für die Datenaufzeichnung verwendet. Aus diesem Grund wird das Gefahrenpotential auf handschriftenbasierte Verifikationsverfahren als sehr gering (Stufe 1) eingeschätzt. Die Angriffsmethode ist auf andere biometrische Verifikationssysteme nicht ohne weiteres anwendbar, da die erzeugten Angriffsgeräusche speziell für den in der Arbeit vorgestellten Verifikationsalgorithmus produziert wurden. Weshalb der Angriff im Allgemein in Stufe 2 eingestuft werden muss, da er nicht ohne weiteres umzusetzen ist, es sei denn der Verifikationsalgorithmus wird von diesem System verwendet.						
7	**Angriffsklasse**	AK 1.1	**8**	**Gefahrenpotential Handschrift**	1	**9**	**Allg. Gefahrenpotential**	2

1	**Angriff Nr.**	9						
2	**Biometrische Modalität**	Handabdruck (Palmprint)						
3	**Autoren/Quelle**	Kong et al. / [KoZK05]						
4	**Kurzbeschreibung**	Ein Handabdruckerkennungssystem wird mittels Brute-Force-Verfahren angegriffen. Dabei werden die Brute-Force-Datensätze vor dem Vergleicher eingespielt. Je nachdem, wie der Schwellenwert für das Erkennungssystem eingestellt ist, kann das System mittels Brute- Force-Angriff zwischen 1 Tag und 31 Jahren gebrochen werden.						
5	**Klassifikation**	Die Angriffsdaten werden vor dem Vergleicher am Angriffspunkt 4 eingespielt. Der Angriff ist dementsprechend ein indirekter Angriff. Er verwendet eine Brute-Force-Methode (Andere Angriffe) und kann somit der Angriffsklasse AK 2.8.4 zugeordnet werden.						
6	**Adaptierbarkeit**	Da dieses Verfahren eine Brute-Force-Angriffsmethode verwendet kann es prinzipiell auf alle biometrischen Erkennungssysteme am Punkt 4 angewendet werden. Voraussetzung hierfür ist der Zugriff auf diesen Angriffspunkt. Das Gefahrenpotential auf Handschriftenbasierte Verifikationsverfahren und auf andere biometrische Erkennungssysteme wird als erheblich (Stufe 3) eingeschätzt. Die Durchführung von Brute-Force-Methoden ist zwar relativ einfach durchzuführen, jedoch sind sie verhältnismäßig einfach zu detektieren und können entsprechend vermieden bzw. Gegenmaßnahmen getroffen werden. So kann beispielsweise eine Wartezeit nach drei negativen Verifikationsversuchen gesetzt werden, um den Brut-Force-Angriff ineffektiv zu gestalten. Die Möglichkeit, einen Brute-Force Angriff auf andere biometrische Verifikationssysteme zu implementieren, wird als hoch eingestuft, jedoch wird die Erfolgschance je nach Modalität und Verifikationsverfahren als eher gering eingeschätzt.						
7	**Angriffsklasse**	AK 2.8.4	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	10						
2	**Biometrische Modalität**	Gesicht						
3	**Autoren/Quelle**	Mai et al. / [MCYJ18]						
4	**Kurzbeschreibung**	In ihrer Arbeit greifen die Autoren ein Gesichtserkennungssystem an, welches auf eine *Convolution Neural Network* (CNN) basiert. Hierbei hat der potentielle Angreifer Zugriff auf die Referenzdaten, welche in einer Datenbank hinterlegt sind. Anhand der Referenzdaten konstruieren sie Bilder (Gesichter) mittels eines angepassten *De-Convolutional Neural Network* (D-CNN). Im Schnitt erzielten sie eine positive Verifikationsrate mit ihren rekonstruierten Gesichtern von ca. 95%. Der Angreifer muss in diesem Angriffsszenario u.a. die Arbeitsweise des CNN kennen, um das entsprechende D-CNN zu konstruieren.						
5	**Klassifikation**	Die Angriffsdaten werden auf Basis der Referenzdaten rekonstruiert, entsprechend kann dieser Angriff als ein Maskerade-Angriff klassifiziert werden. Die Daten werden aus der Referenzdatenbank gelesen (Angriffspunkt 7) und die rekonstruierten Angriffsdaten am Angriffspunkt 2 eingespielt. Der Angriff kann entsprechend den Angriffsklassen AK 2.6.2 und 2.6.7 zugeordnet werden.						
6	**Adaptierbarkeit**	Dieser Angriff kann auf ein handschriftenbasiertes Verifikationssystem angewendet werden, solange es ein CNN verwendet. Dementsprechend wird das Gefahrenpotential als erheblich (Stufe 3) eingeschätzt. CNNs werden u.a. in der Signalverarbeitung eingesetzt, somit auch in der Bildverarbeitung, hier insbesondere zur Detektion von Objekten/Strukturen auf digitalen Bildern. Entsprechend können alle biometrischen Verifikationssysteme, die auf die Verarbeitung von digitalen Bildern basieren ein CNN zur Klassifikation verwenden kompromittiert werden. Entsprechend wären solche CNN basierten Verifikationssysteme für dieses Angriffsverfahren anfällig. Aus diesem Grund wird das Gefahrenpotential im Allgemeinen als erheblich (Stufe 3) eingeschätzt.						
7	**Angriffsklasse**	AK 2.6.2 AK 2.6.7	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	11						
2	**Biometrische Modalität**	Fingerabdruck						
3	**Autoren/Quelle**	Cappelli et al. / [Capp03]						
4	**Kurzbeschreibung**	In seiner Arbeit beschreibt der Autor wie künstliche Fingerabdrücke erzeugt werden können. Primär bestand die Intention des Autors darin, künstliche Fingerabdrücke zu erzeugen, um Verifikationssysteme zu testen, da das Sammeln und Aufzeichnen von echten biometrischen Daten ein zeit- und kostenintensiver Prozess darstellen. Diese Daten können jedoch auch verwendet werden, um einen Angriff auf ein Verifikationssystem durchzuführen. Daher soll diese Technik zur Erzeugung von künstlichen Daten hier betrachtet werden.						
5	**Klassifikation**	Die Angriffsdaten werden komplett künstlich erzeugt und benötigen keine echten biometrischen Daten als Basis. In der Arbeit werden die Daten nicht für einen Angriff verwendet, könnten jedoch bei einem Verifikationssystem am Angriffspunkt 2 eingespielt werden, um einen Brute-Force-Angriff durchzuführen. Der Angriff könnte entsprechend der Angriffsklasse AK 2.8.2 (Andere Angriffstechnik) zugeordnet werden.						
6	**Adaptierbarkeit**	Die Methode zur Erzeugung von künstlichen Fingerabdrücken kann nicht direkt auf ein handschriftenbasiertes Verifikationssystem angewendet werden. Jedoch ist die Strategie, künstlich erzeugte biometrische Daten für einen potentiellen Angriff zu verwenden interessant und auf andere biometrische Verifikationssysteme übertragbar. Aus diesem Grund wird das Gefahrenpotential dieser Methode als erheblich (Stufe 3) für andere biometrische Verifikationssysteme eingeschätzt. Da es ebenfalls auch indirekt auf die dynamische Handschrift übertragen werden kann, wird es für diesen Fall als mäßig (Stufe 2) eingestuft.						
7	**Angriffsklasse**	AK 2.8.2	**8**	**Gefahrenpotential Handschrift**	2	**9**	**Allg. Gefahrenpotential**	3

1	**Angriff Nr.**	12						
2	**Biometrische Modalität**	Fingerabdruck						
3	**Autoren/Quelle**	Cappelli et al. / [CMLM07]						
4	**Kurzbeschreibung**	In Ihrer Arbeit beschreiben die Autoren, wie sie auf Basis von Referenzdaten künstliche Fingerabdrücke erzeugen. Dabei verwenden sie relativ wenige Daten, welche in den Referenzdaten hinterlegt sind, um einen künstlichen Fingerabdruck zu erzeugen. Ziel ist es, ein Fingerabdruckbild zu erzeugen welches dem originalen ursprünglichen Abdruck ähnelt und die gleichen Referenzdaten generiert, um eine positive Verifikation am System zu bewirken.						
5	**Klassifikation**	Die Angriffsdaten werden auf Basis von biometrischen Referenzdaten erzeugt (Maskerade) und am Angriffspunkt 2 eingespielt. Demnach kann dieser Angriff der Angriffsklasse AK 2.6.2 zugeordnet werden.						
6	**Adaptierbarkeit**	Die Methode zur Erzeugung von künstlichen Fingerabdrücken kann nicht direkt auf ein handschriftenbasiertes Verifikationssystem angewendet werden. Jedoch ist die Strategie, künstlich erzeugte biometrische Daten auf Basis weniger Referenzdaten zu generieren (Maskerade) interessant. Außerdem wurde dieser Angriff erfolgreich auf neun verschiedenen Verifikationssystemen für die Modalität Fingerabdruck durchgeführt und zeigt die Gefährlichkeit für diese. Die Idee, charakteristische Merkmale (Positionen der Minutien) zu verwenden, um Fingerabdrücke zu rekonstruieren kann potentiell auf andere biometrische Verfahren adaptiert werden. Voraussetzung hierfür ist jedoch, dass diese Informationen in den Referenzdaten enthalten sind bzw. aus diesen ermittelt werden können. Aus diesem Grund wird das Gefahrenpotential dieser Methode als erheblich (Stufe 3) für Verifikationssysteme im Allgemeinen als auch für handschriftenbasierte Verifikationssystem eingeschätzt.						
7	**Angriffsklasse**	AK 2.6.2	**8**	**Gefahrenpotential Handschrift**	3	**9**	**Allg. Gefahrenpotential**	3

Anlage 3 Bestimmung der Toleranzfaktoren für die jeweiligen Semantiken (FA1)

Nachfolgend werden alle Diagramme angegeben, welche für die Bestimmung des globalen Optimierungsfaktors (Toleranzfaktor) verwendet wurden. Hierbei wurde jeweils der Toleranzfaktorwert gewählt, welcher die niedrigste Equal Error Rate (EER) bewirkt für den EER Arbeitsmodus bzw. die geringste Kollisions-Reproduktions-Rate (CRR) für den CRR Arbeitsmodus.

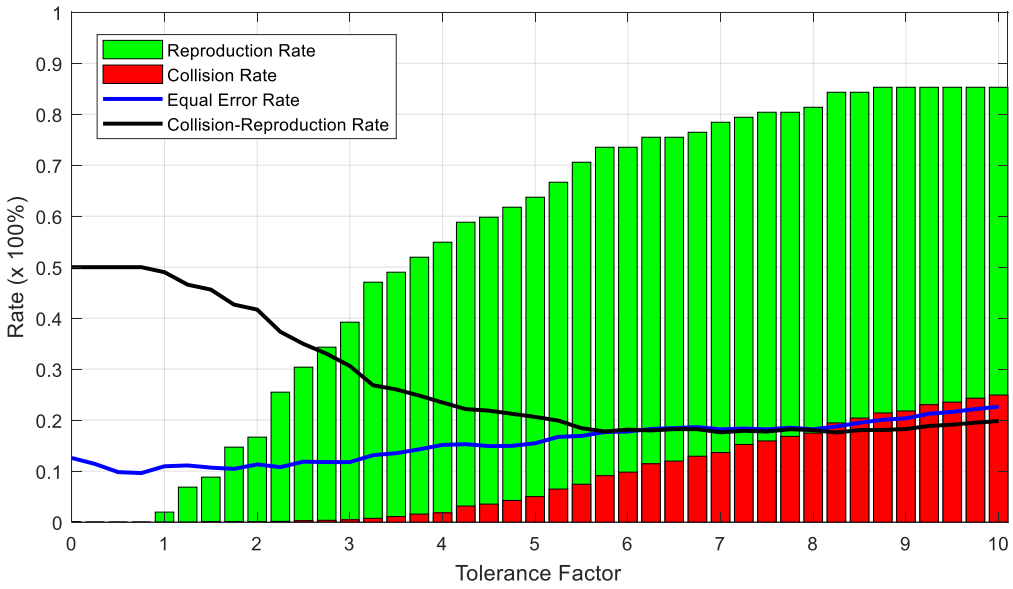


Abbildung 56 Bestimmung des Toleranzfaktors der Semantik 77993 (131 Merkmale)

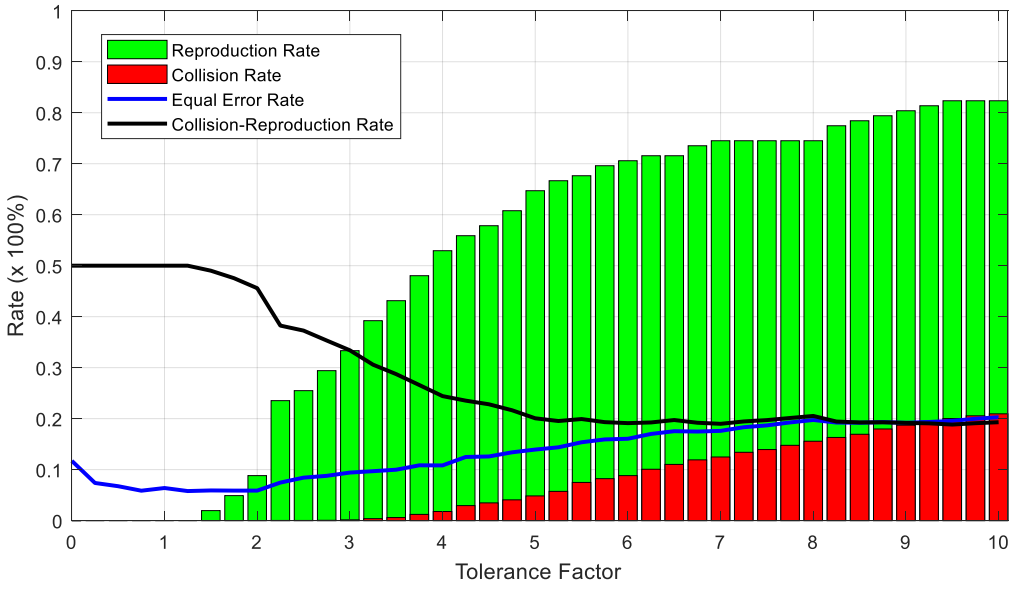


Abbildung 57 Bestimmung des Toleranzfaktors der Semantik PIN (131 Merkmale)

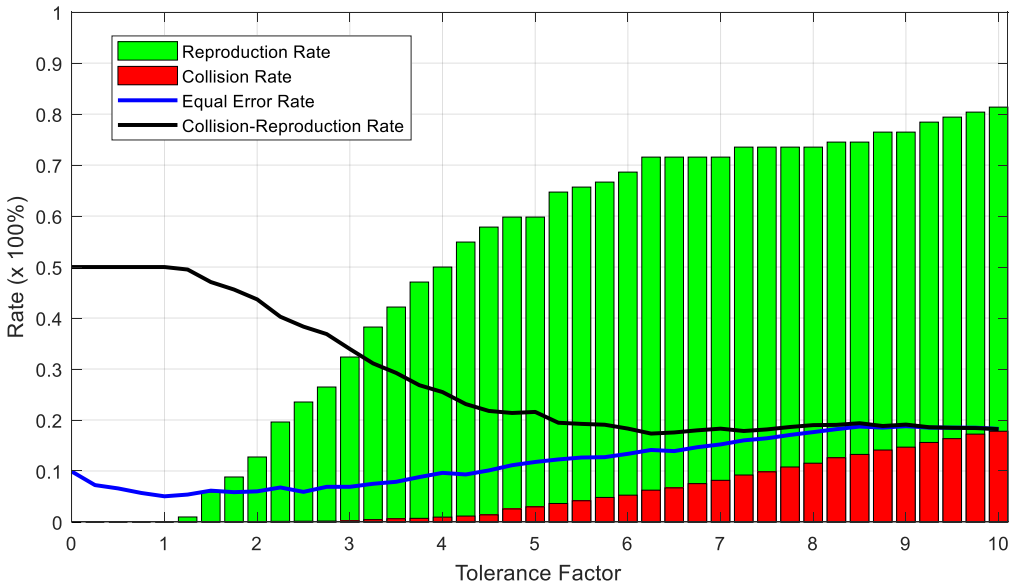


Abbildung 58 Bestimmung des Toleranzfaktors der Semantik Pseudonym (131 Merkmale)

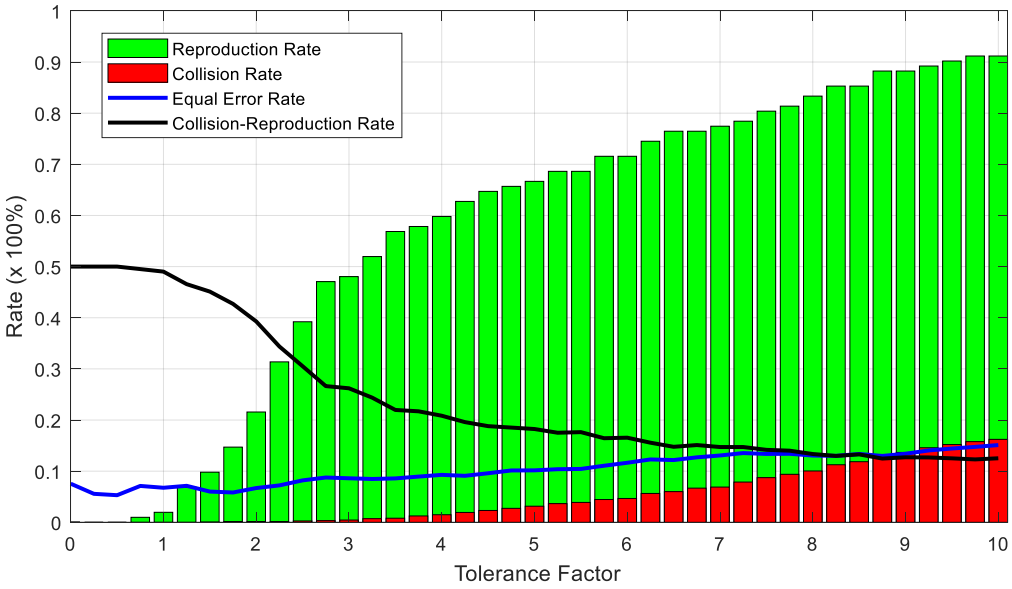


Abbildung 59 Bestimmung des Toleranzfaktors der Semantik Symbol (131 Merkmale)

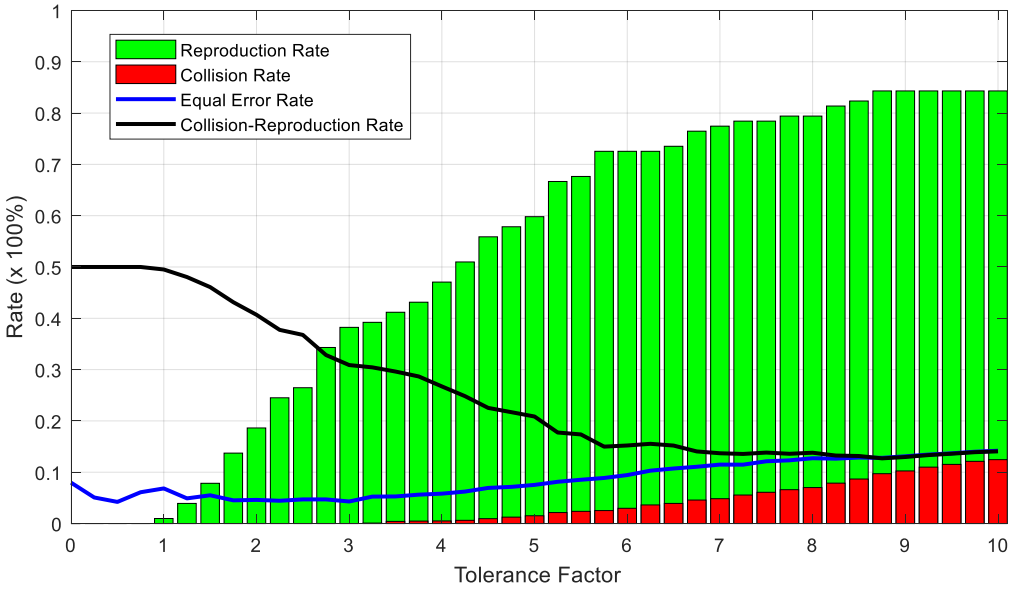


Abbildung 60 Bestimmung des Toleranzfaktors der Semantik Woher (131 Merkmale)

Nachfolgend werden alle Diagramme dargestellt, welche ohne die Basismerkmale ermittelt wurden und zur Bestimmung des jeweiligen Toleranzfaktors für den entsprechenden Arbeitsmodus dienen.

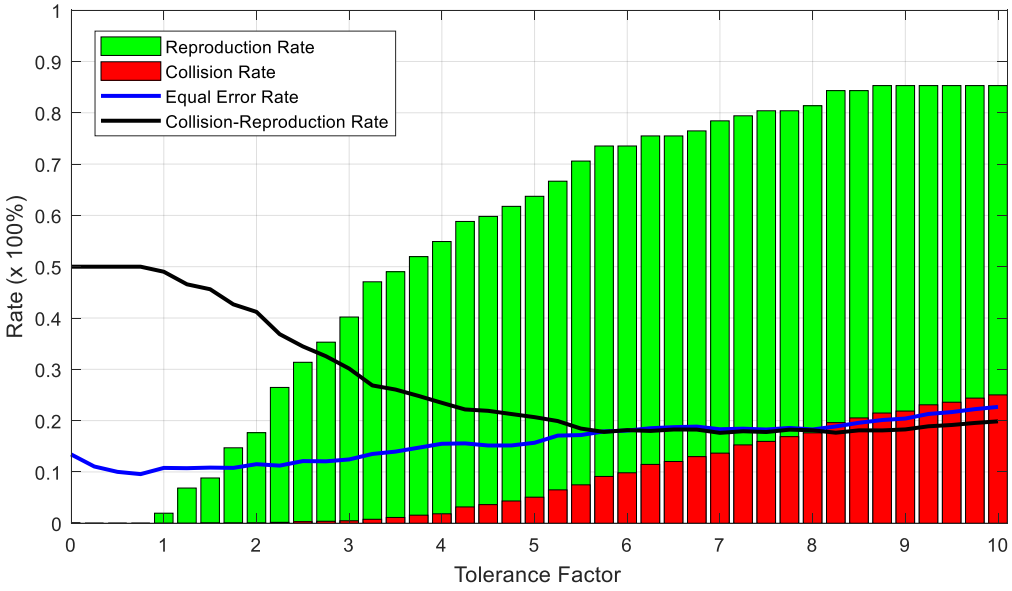


Abbildung 61 Bestimmung des Toleranzfaktors der Semantik 77993 (122 Merkmale)

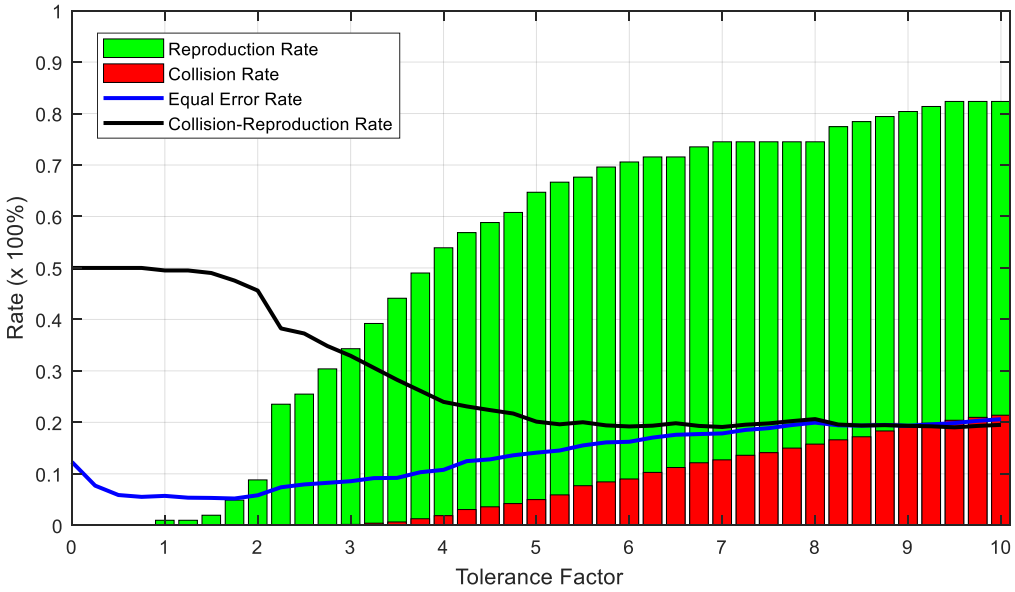


Abbildung 62 Bestimmung des Toleranzfaktors der Semantik PIN (122 Merkmale)

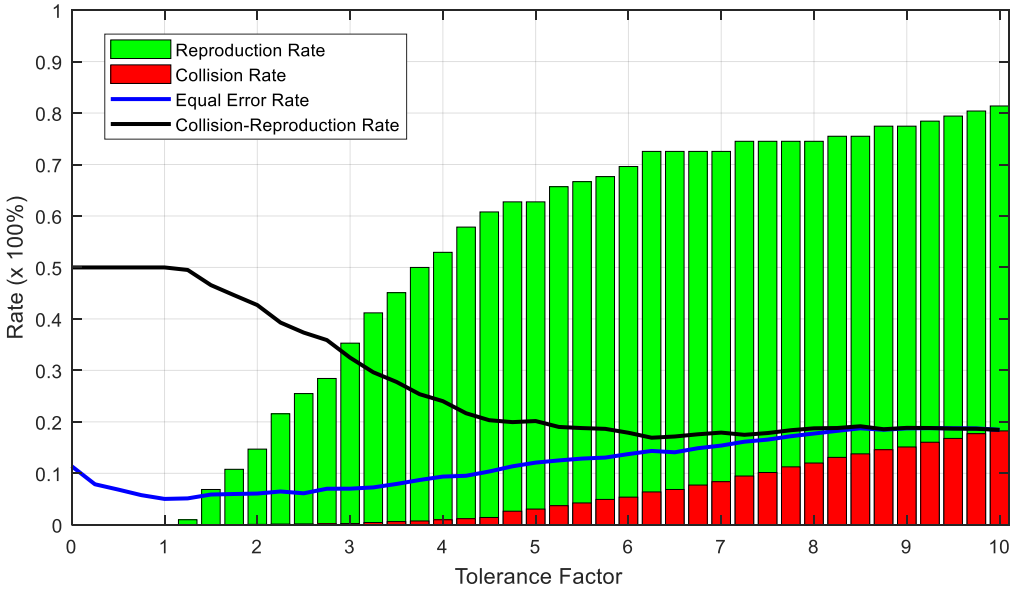


Abbildung 63 Bestimmung des Toleranzfaktors der Semantik Pseudonym (122 Merkmale)

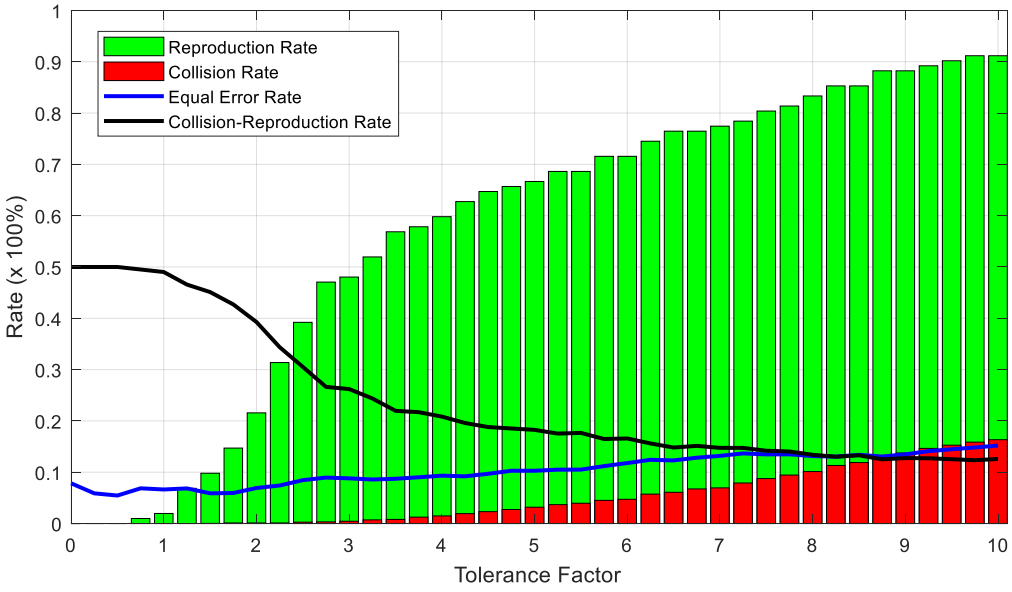


Abbildung 64 Bestimmung des Toleranzfaktors der Semantik Symbol (122 Merkmale)

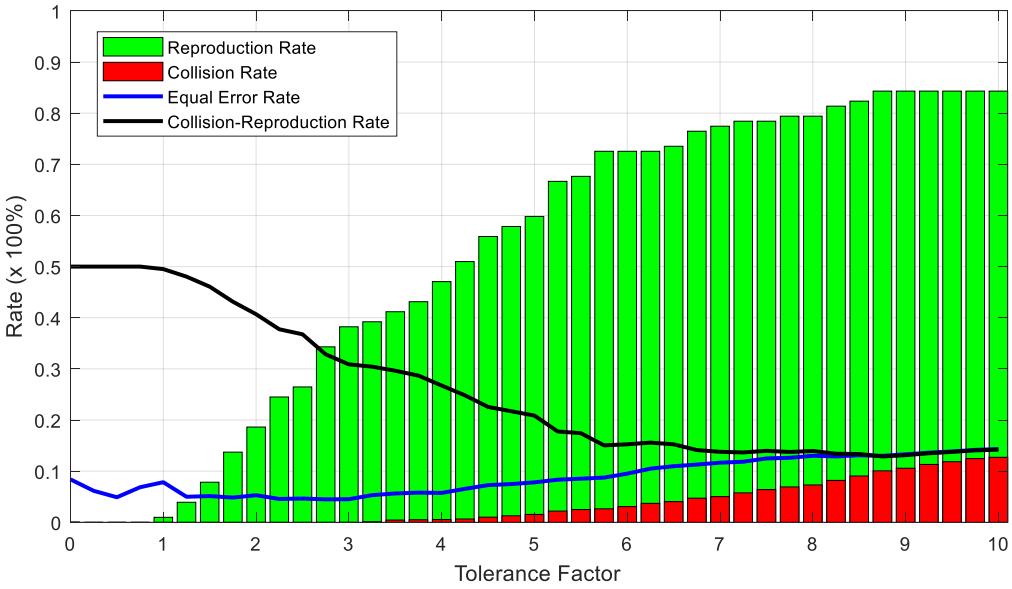


Abbildung 65 Bestimmung des Toleranzfaktors der Semantik Woher (122 Merkmale)

Anlage 4 Fehlerraten für die Bestimmung der Erfolgschancen eines Angriffs

Nachfolgend werden aller ermittelten Fehlerraten für die entsprechenden Semantiken im Arbeitsmodus EER dargestellt. FRR und FAR werden anhand der originalen Daten ermittelt und FAR_{re} mittels künstlich erzeugten Handschriftensignalen.

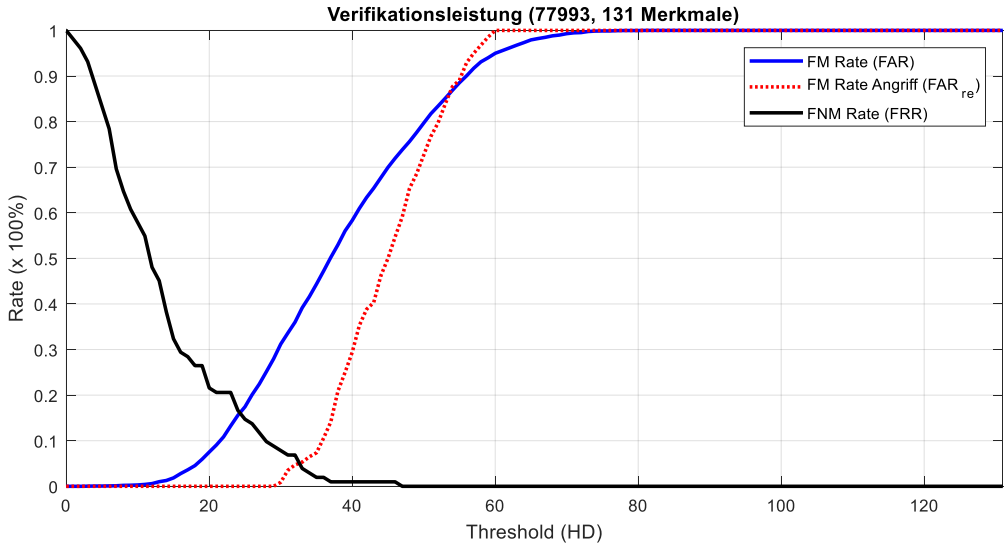


Abbildung 66 Fehlerraten der Semantikklasse 77993 im Arbeitsmodus EER

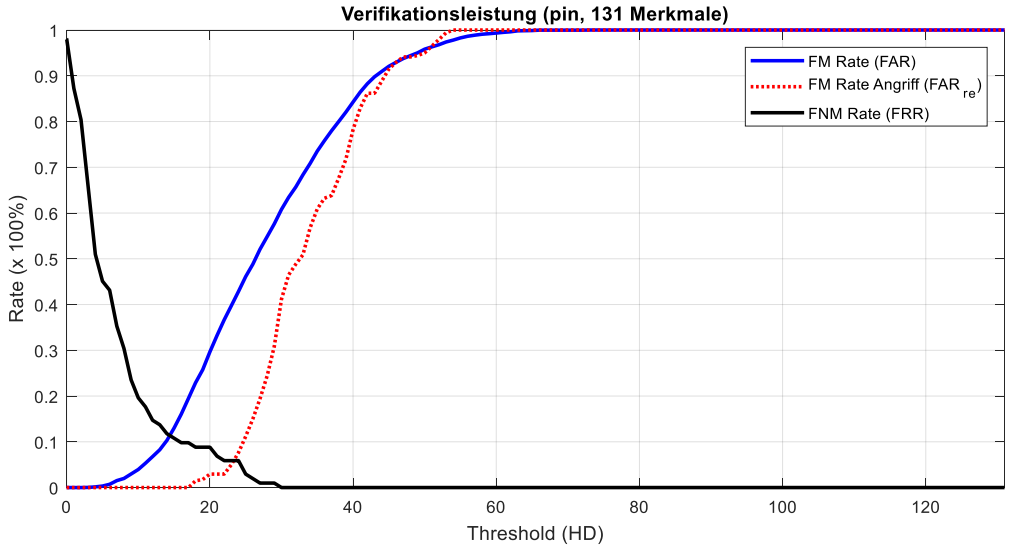


Abbildung 67 Fehlerraten der Semantikklasse PIN im Arbeitsmodus EER

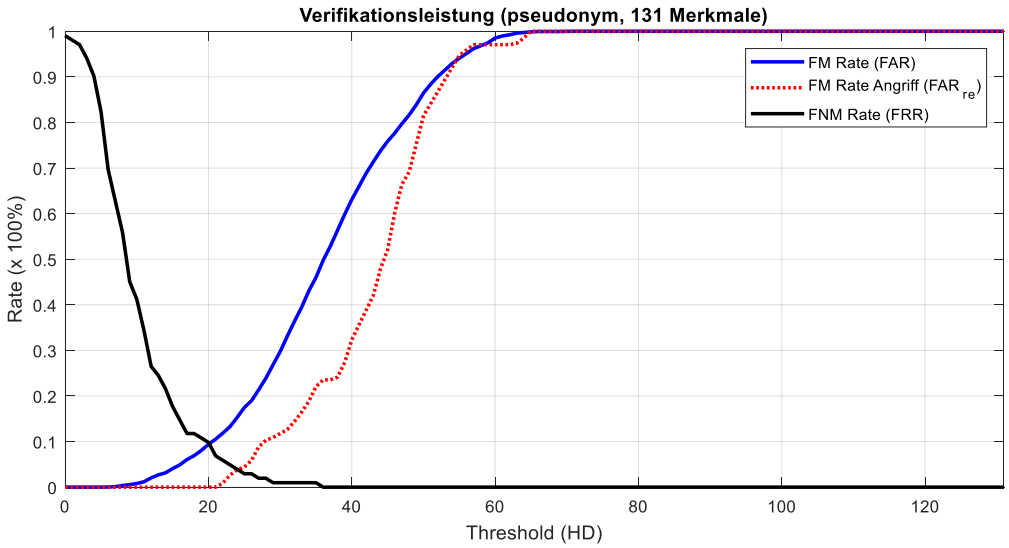


Abbildung 68 Fehlerraten der Semantikklasse Pseudonym im Arbeitsmodus EER

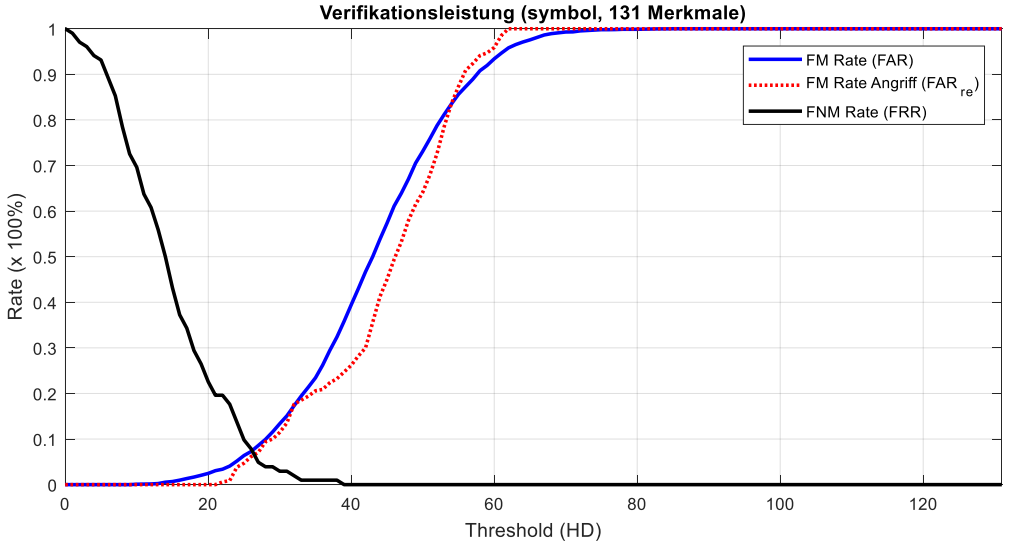


Abbildung 69 Fehlerraten der Semantikklasse Symbol im Arbeitsmodus EER

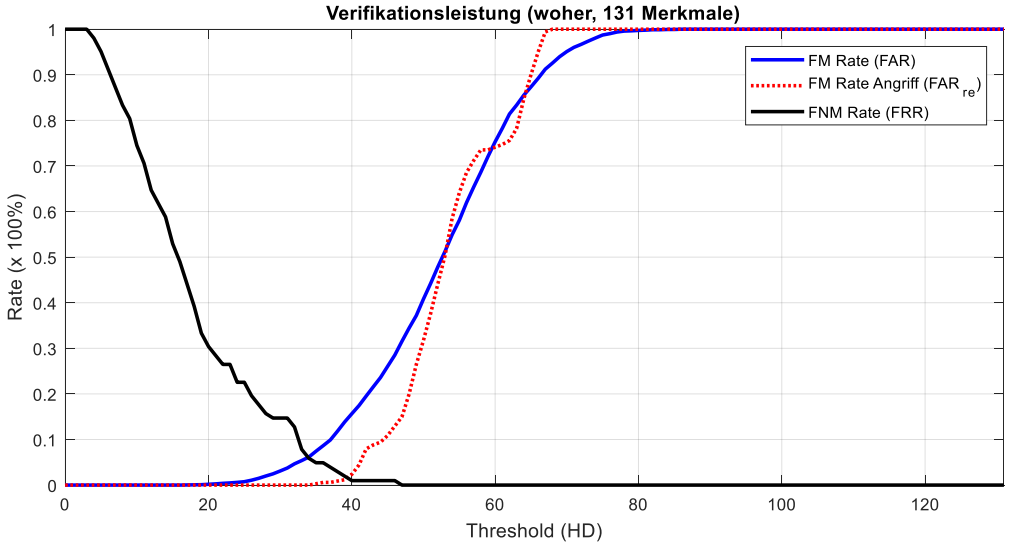


Abbildung 70 Fehlerraten der Semantikklasse Woher im Arbeitsmodus EER

Nachfolgend werden die Fehlerraten des nicht optimierten Systems in den jeweiligen Semantikklassen dargestellt.

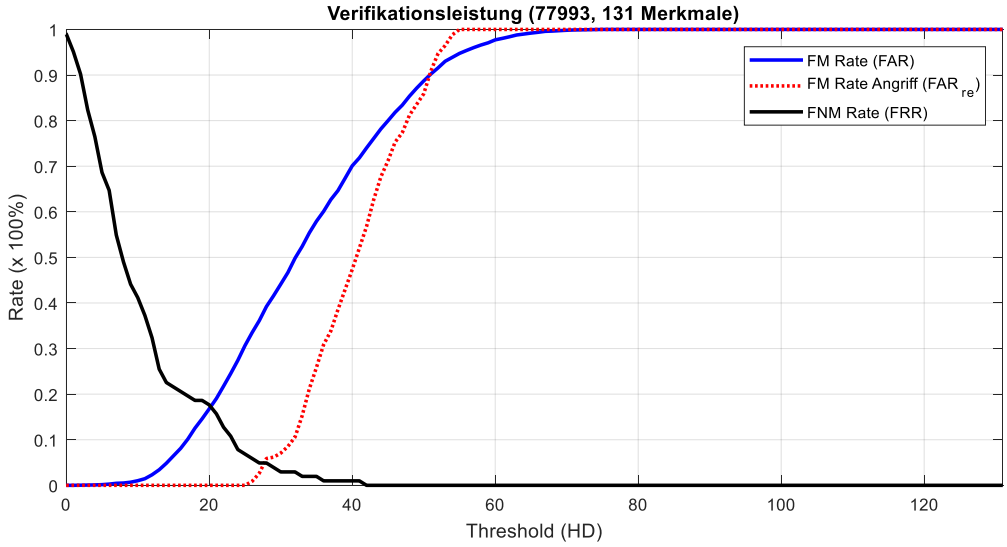


Abbildung 71 Fehlerraten der Semantikklasse 77993 im nicht optimierten System

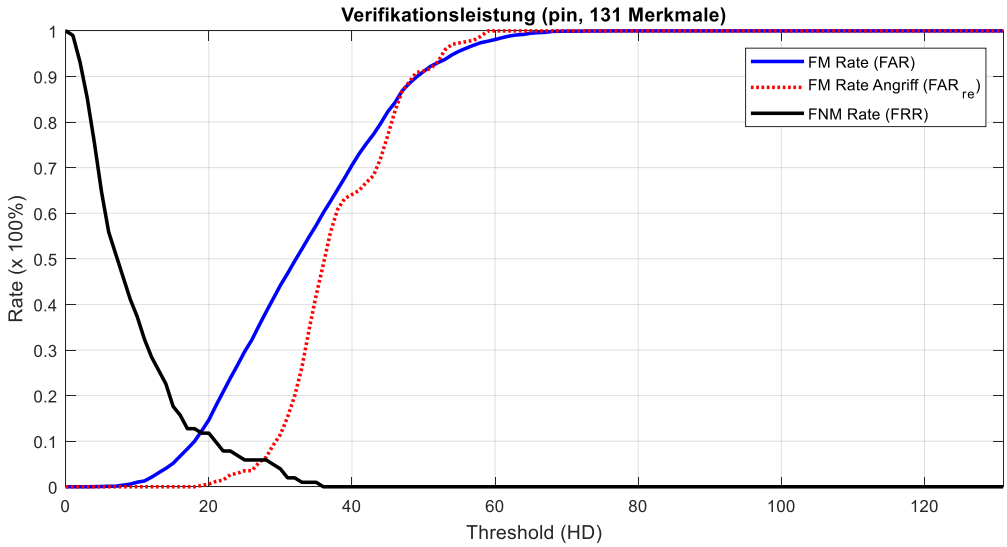


Abbildung 72 Fehlerraten der Semantikklasse PIN im nicht optimierten System

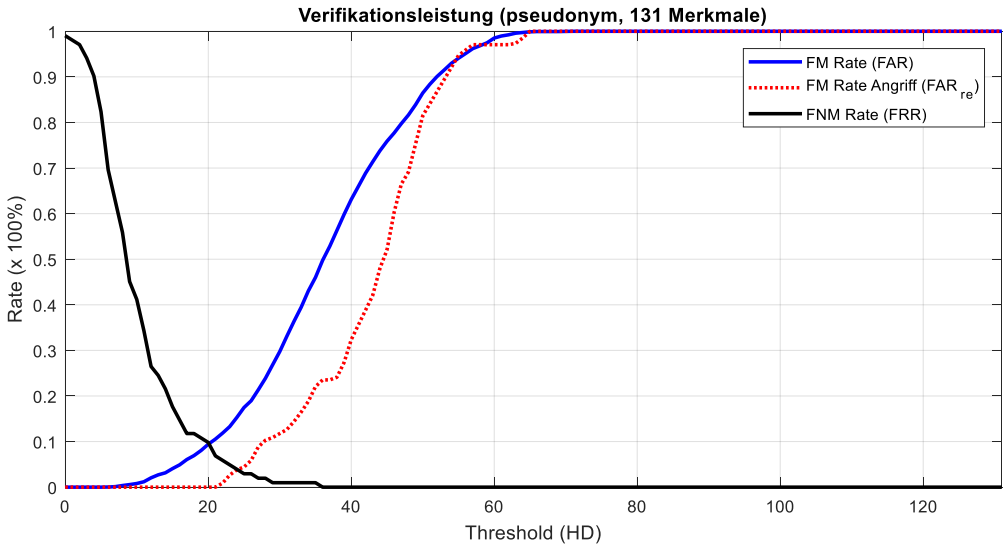


Abbildung 73 Fehlerraten der Semantikklasse Pseudonym im nicht optimierten System

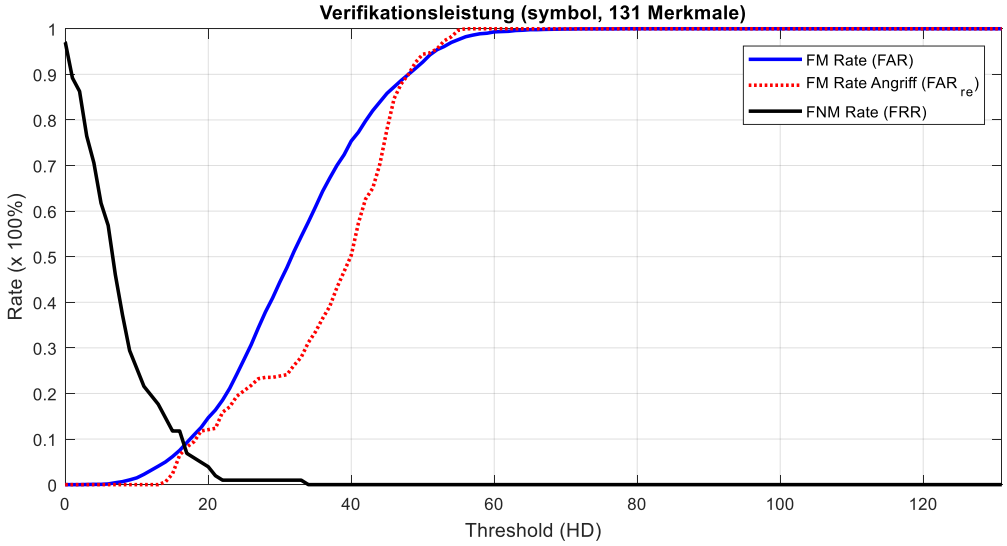


Abbildung 74 Fehlerraten der Semantikklasse Symbol im nicht optimierten System

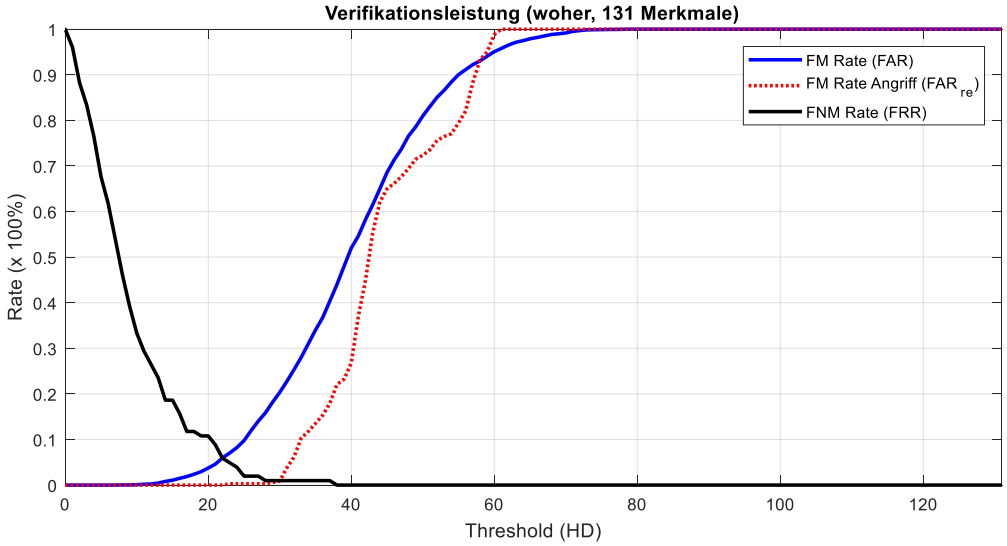


Abbildung 75 Fehlerraten der Semantikklasse Woher im nicht optimierten System

Nachfolgend werden aller ermittelten Fehlerraten für die entsprechenden Semantiken im Arbeitsmodus CRR dargestellt.

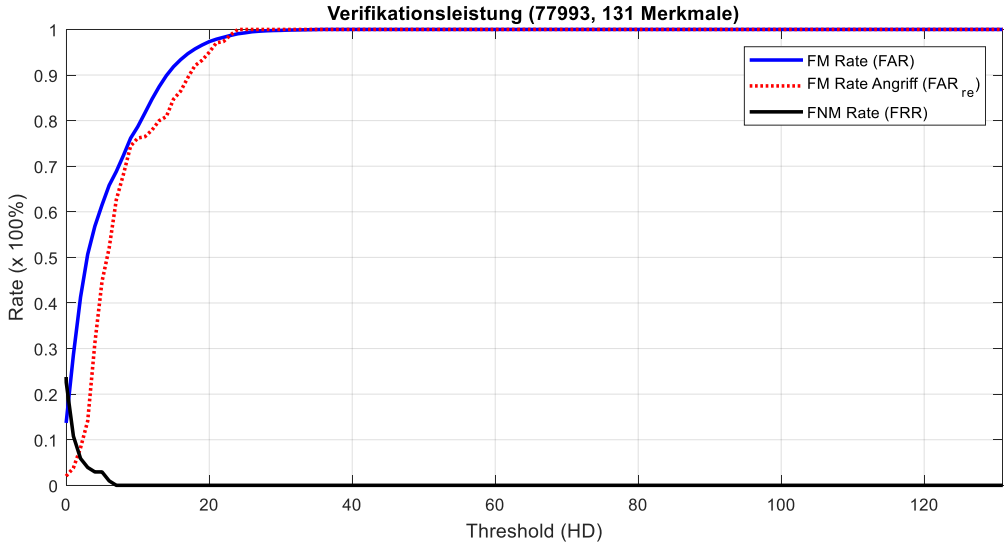


Abbildung 76 Fehlerraten der Semantikklasse 77993 im Arbeitsmodus CRR

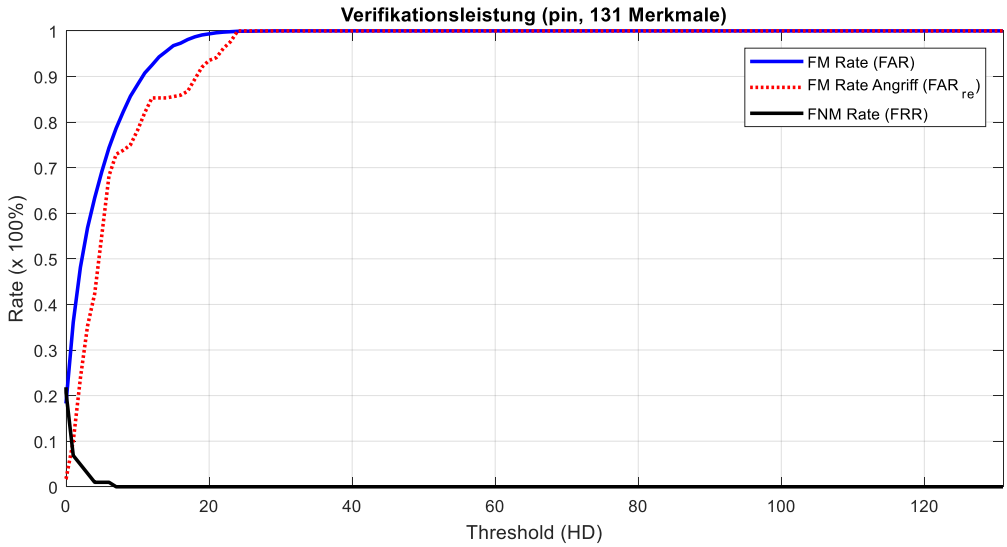


Abbildung 77 Fehlerraten der Semantikklasse PIN im Arbeitsmodus CRR

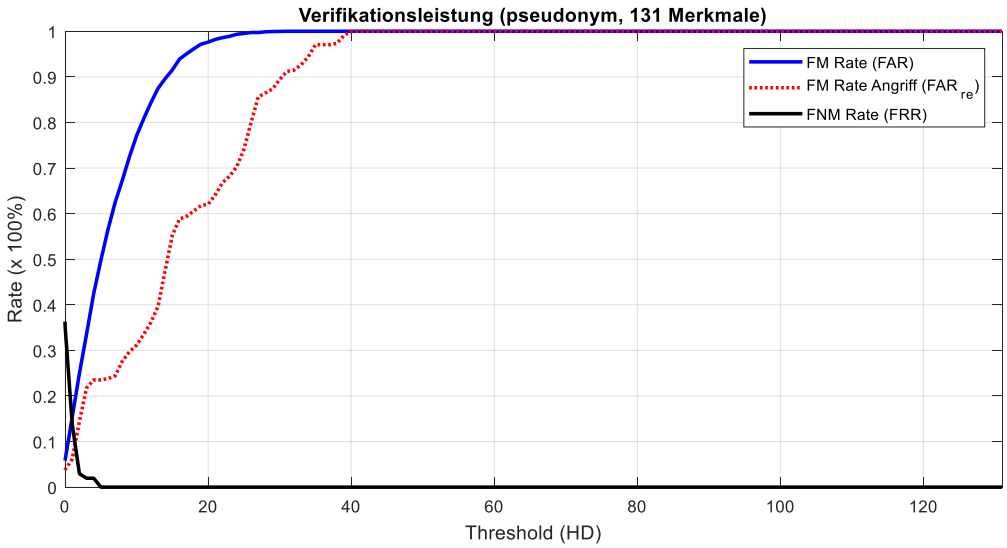


Abbildung 78 Fehlerraten der Semantikklasse Pseudonym im Arbeitsmodus CRR

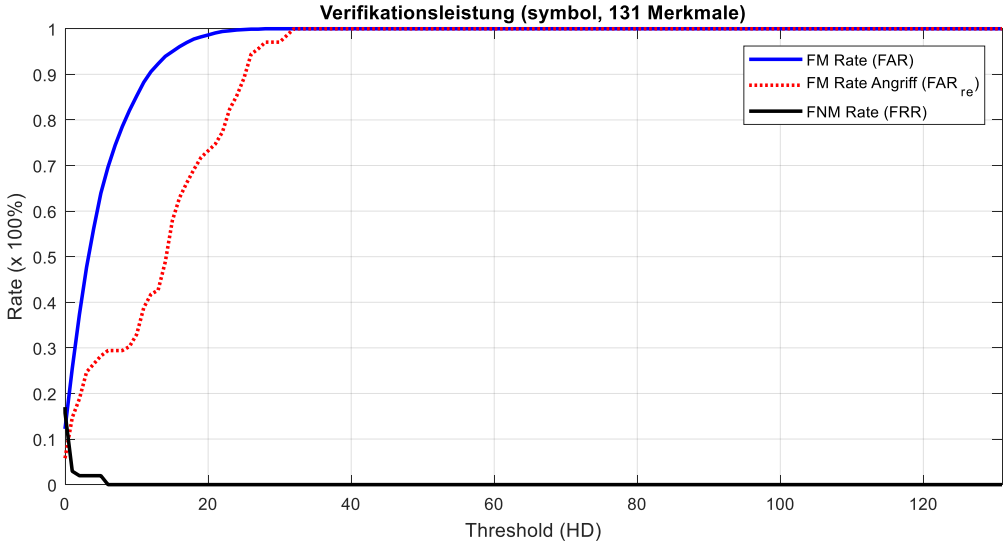


Abbildung 79 Fehlerraten der Semantikklasse Symbol im Arbeitsmodus CRR

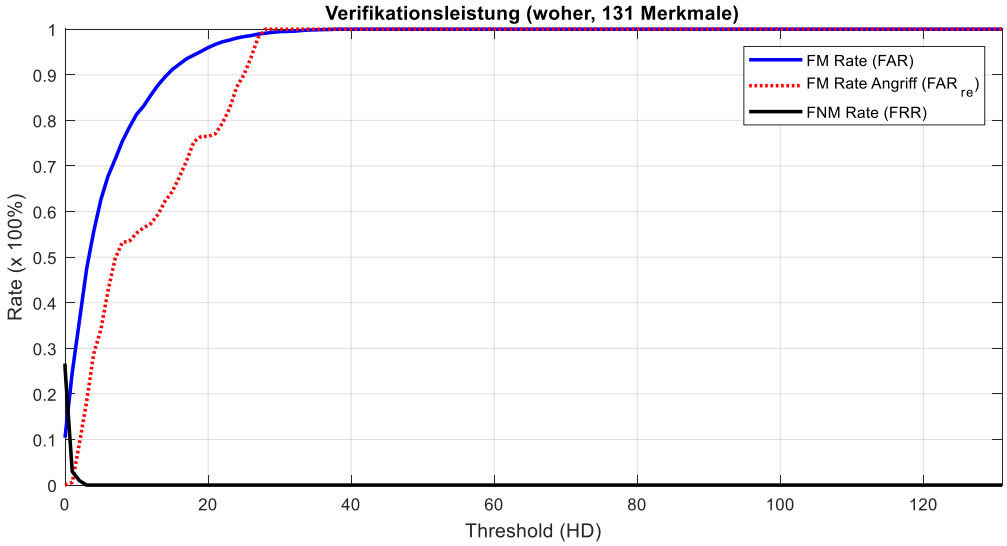


Abbildung 80 Fehlerraten der Semantikklasse Woher im Arbeitsmodus CRR

Anlage 5 Beispiel Quellcode für die Umsetzung von Merkmalen (FA2)

Nachfolgend wird in Abbildung 81 beispielhaft anhand eines Matlab/Octave Quellcodes gezeigt, wie der maximale Seitenwinkelwert (Azimut) in ein Rohdatensignal integriert werden kann. Das Rohdatensignal besteht aus mehreren Samplepunkten, wobei ein Samplepunkt neun Werte (Spalten) besitzt. Diese Werte sind (1) laufende Nummer, (2) x-Wert, (3) y-Wert, (4) x-Wert(alt), (5) y-Wert(alt), (6) Zeit, (7) Druck, (8) Höhenwinkel und (9) Seitenwinkel.
Die Funktion berücksichtigt bei der Integration des maximalen Seitenwinkels den bereits integrierten minimalen Seitenwinkel und den Durchschnitts-Seitenwinkel.

```
% Funktion integriert den maximalen Azimut-Winkelwert in die Rohdaten.
% Wobei der minimal Azimut-Winkelwert und der durchschnittliche Azimut-Winkelwert
nicht
% beeinflusst werden
% Input: rawData (Matrix; m-Zeilen, 9 Spalten) in Spalte 9 befindet sich der
% Azimut-Winkelwert des Samplepunktes
% Input: azi_min (minimaler Azimut-Winkelwert)
% Input: azi_avr (durchschnittlicher Azimut-Winkelwert)
% Input: azi_max (maximaler Azimut-Winkelwert)
% Output: rawData --> Matrix (Rohdaten mit integrierten maximalen Azimut Winkel-
wert)
function[rawData] = insert_max_azimut(rawData, azi_min, azi_avr, azi_max)
 % Anzahl der Samplepunkte (Zeilen) bestimmen
 [rawData_row_count rawData_col_count] = size(rawData);
 % zu kompensierenden Maximalwert gegenüber Durchschnittswert
 toCompensate = azi max - azi avr;
 alreadyComp = toCompensate; % bereits kompensierter Wert
 % bestimmen des maximalen Subtrahneds, damit Minimalwert nicht "unterboten" wird +
1
 MaxSubProSample = azi_avr - (azi_min + 1);
 % wenn 1 dann wurde der maximal Azimut-Winkelwert bereits bei einem Samplepunkt
gesetzt
 max_is_set = 0;
 for i=1:1:rawData_row_count % Zeile für Zeile alle Samplepunkte durchlaufen
  if(rawData(i, 7) ~= -1) % wenn aktueller Samplepunkt kein Absetzpunkt ist
   % wenn max. Azimut gesetzt und noch zu kompensierende Werte übrig
   if( (max_is_set == 1) && (toCompensate > 0) )
    % wenn zu kompensierender Wert in ein Samplepunkt untergebracht werden kann
    % ohne min./avg. Azimut zu beeinflussen
    if(((alreadyComp-MaxSubProSample) < alreadyComp)&&(azi_avr - already-
Comp)>=azi_min+1)
     rawData(i,8) = azi_avr - alreadyComp; % schreibe noch zu kompensierenden Wert
     toCompensate = 0;
    else % auf mehrer Samplepunkte verteilen
     rawData(i,8) = (azi_min+1);
     alreadyComp = alreadyComp - MaxSubProSample;
    end
   else
    if (max_is_set == 0)
     rawData(i,8) = azi_max; %Maximalwert einmal setzen
     max_is_set = 1;
    end
   end
  else % Wenn Absetzpunkt dann mache nichts
  end
 end % Ende For-Schleife
end % Ende der Funktion
```

Abbildung 81 Funktion zur Integration des Seitenwinkels (Matlab/Octave)

In Abbildung 82 sind die jeweiligen Seitenwinkel-Winkelwerte und der zu kompensierende Anteil grafisch dargestellt. Der zu kompensierende Anteil des maximalen Wertes wird auf die nachfolgenden Samplepunkte (14 und 15) verteilt bzw. abgezogen. Weiterhin ist in der Abbildung zu erkennen, dass der minimale Seitenwinkel durch die angepassten Winkelwerte nicht unterboten wird.

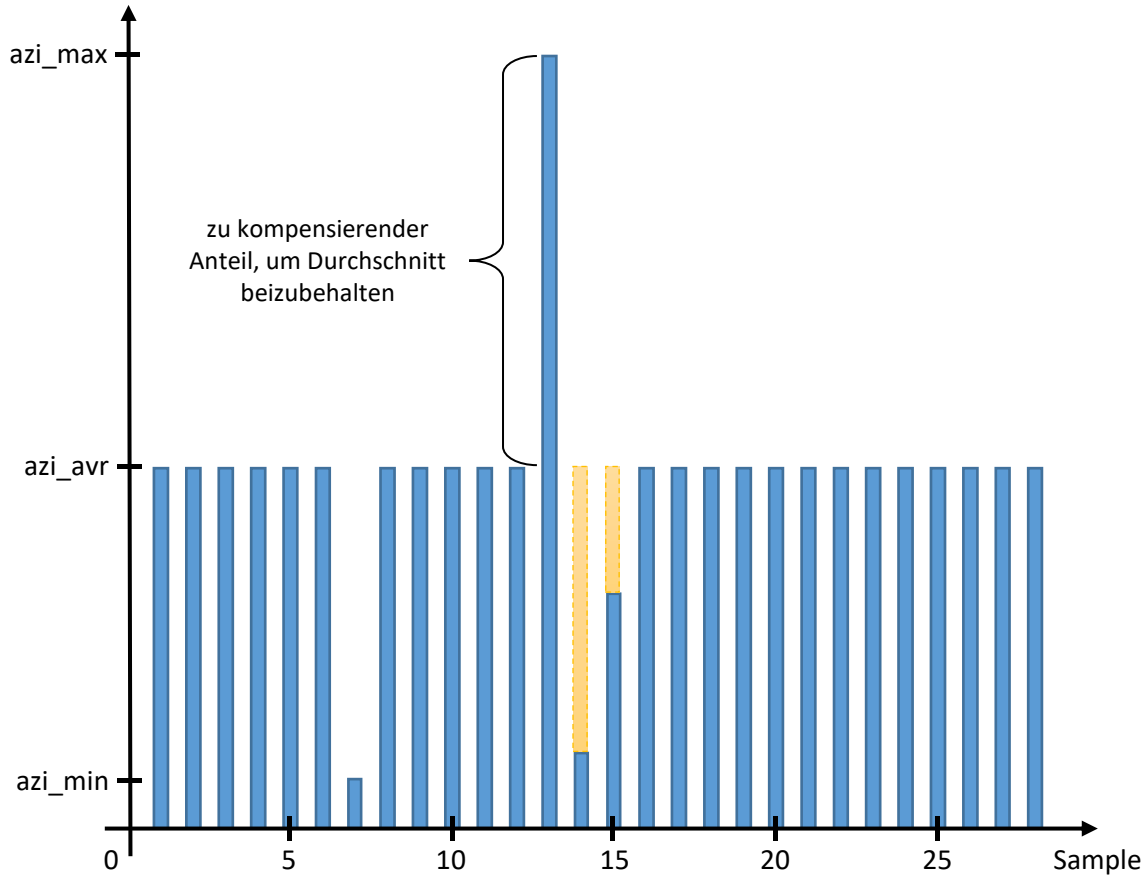


Abbildung 82 Azimut-Winkelwerte und zu kompensierender Anteil

Anlage 6 Bestimmung der Toleranzfaktoren für die jeweiligen Semantiken (FA2)

Nachfolgend werden alle Diagramme angegeben, welche für die Bestimmung des globalen Optimierungsfaktors (Toleranzfaktor) verwendet wurden. Hierbei wurde jeweils der Toleranzfaktorwert gewählt, welcher die niedrigste Equal Error Rate (EER) bewirkt.

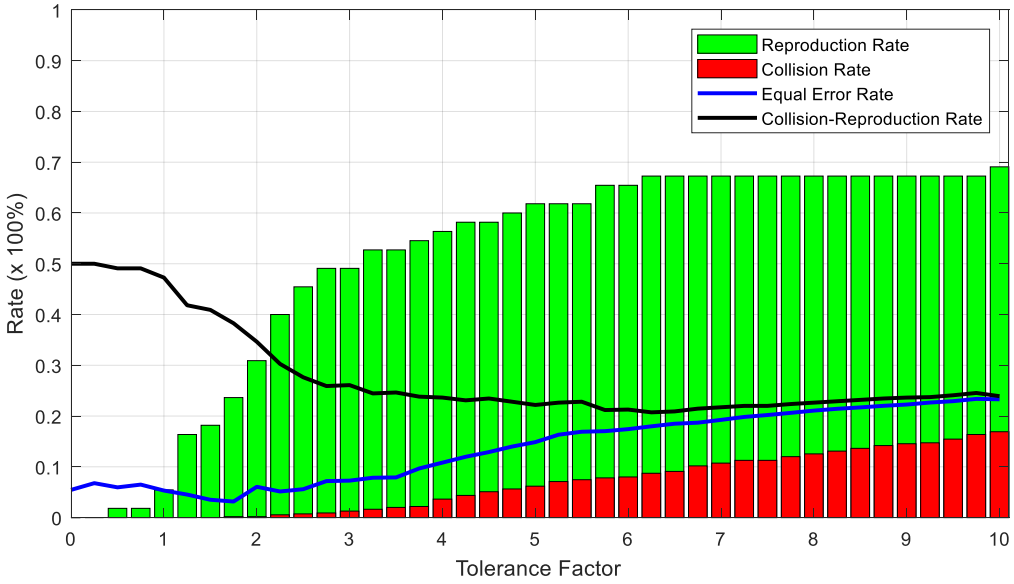


Abbildung 83 Bestimmung des Toleranzfaktors der Semantik 2759

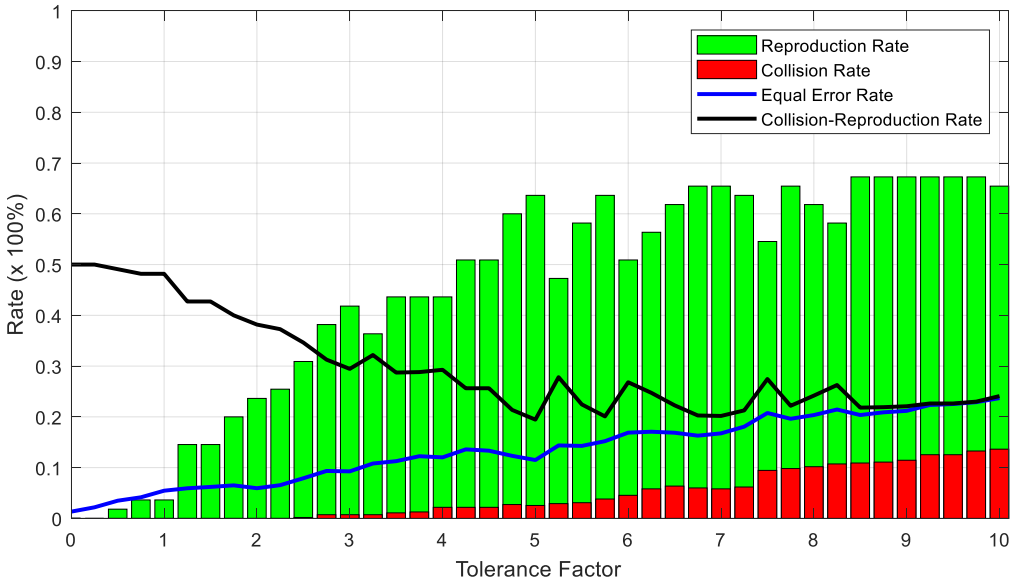


Abbildung 84 Bestimmung des Toleranzfaktors der Semantik arbeiten

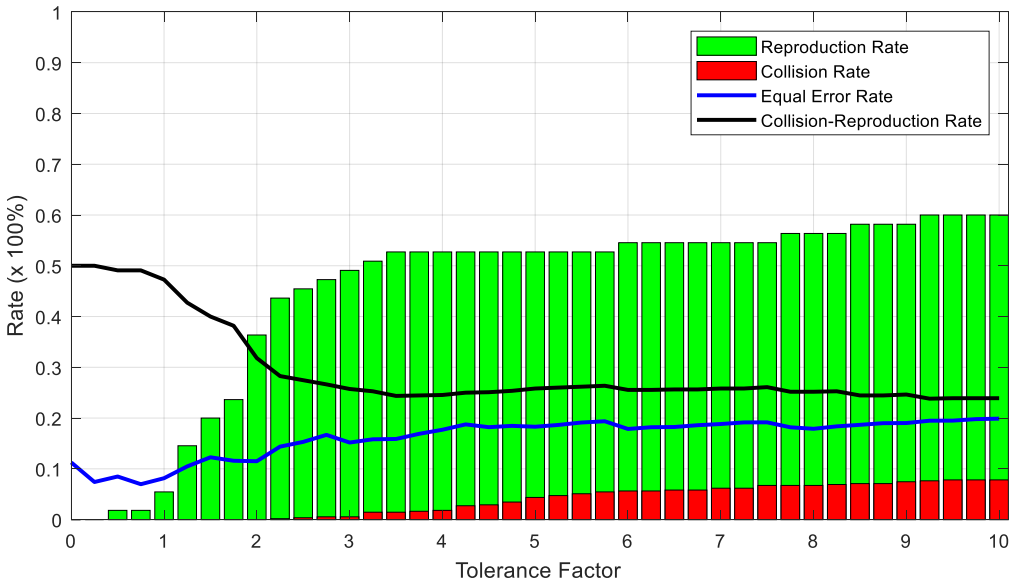


Abbildung 85 Bestimmung des Toleranzfaktors der Semantik Iraq

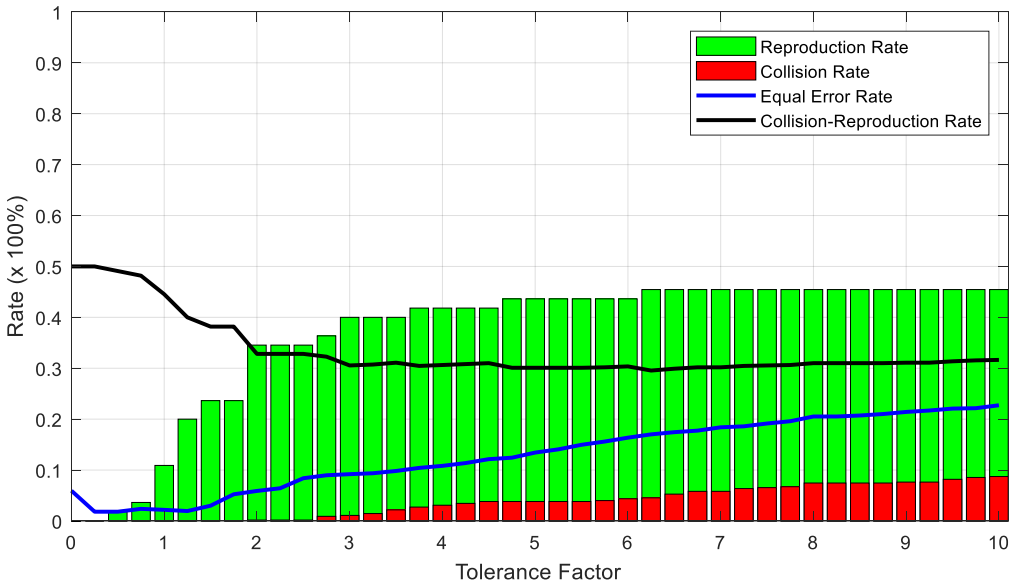


Abbildung 86 Bestimmung des Toleranzfaktors der Semantik Seife

Anlage 7 Kurvenverlauf der Fehlerraten der originalen Handschriften (FA2)

Nachfolgend werden alle Diagramme dargestellt, welche den Kurvenverlauf der Fehlerraten (FAR, FRR und EER) aller Semantiken zeigen.

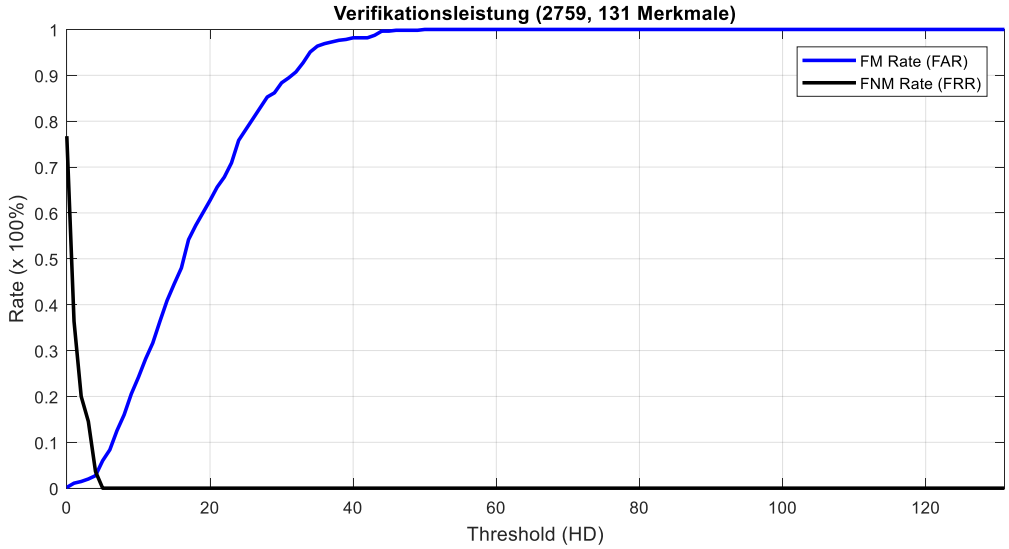


Abbildung 87 Verifikationsleistung der Semantik 2759

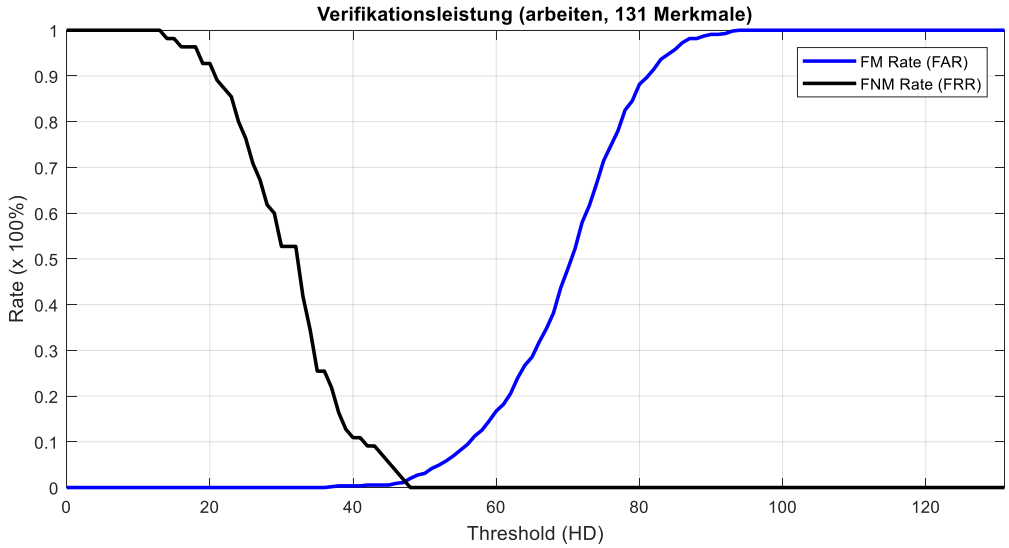


Abbildung 88 Verifikationsleistung der Semantik arbeiten

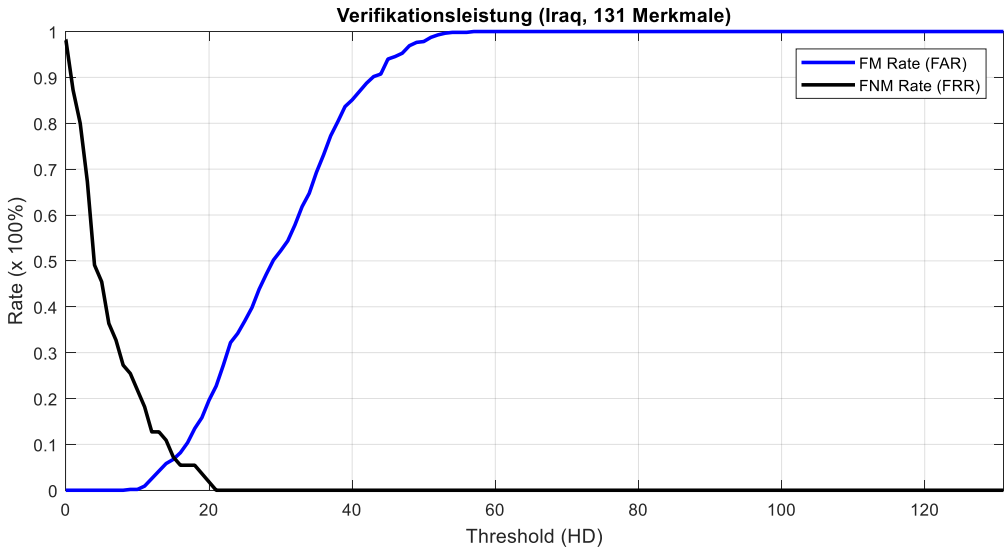


Abbildung 89 Verifikationsleistung der Semantik Iraq

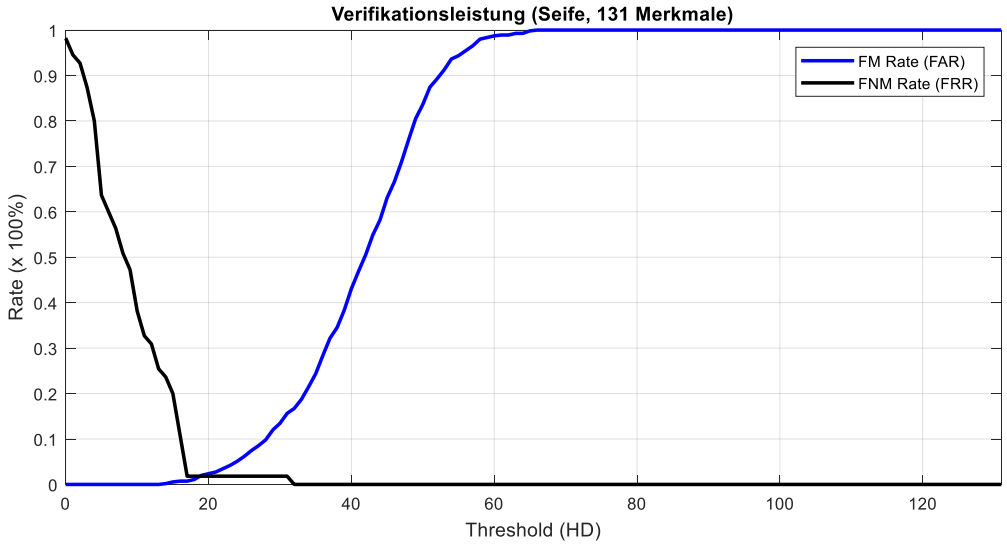


Abbildung 90 Verifikationsleistung der Semantik Seife

Anlage 8 Bestimmung geeigneter Modifikationsparameter (FA2)

Nachfolgend werden die Modifikationsparameter, Fehlerraten und beispielhafte künstliche Handschriftensignale präsentiert, die zur Bestimmung geeigneter Modifikationsparameter geführt haben.
Bei der 4. Iteration wurden nicht die Modifikationsparameter *mp* für die Erzeugung der Schreibindividuen geändert, sondern lediglich die Modifikationsparameter *mpS* zur Erzeugung der Handschriftendaten (siehe Tabelle 50). Ziel hierbei ist die Erhöhung der Gleichfehler- und Kollisionsraten.

Tabelle 50 Modifikationsparameter zur Erzeugung der künstlichen Handschriftendaten (4. It.)

Mod. Parameter	Künstliches Handschriftensample	Hinweis / Erklärung
	1 - 10	
InterpolMethode	pchip	Piecewise Cubic Hermite Interpolating Polynomial (Kubisch Hermitescher Spline)
ModMaximaXVal	0,25	Eine Pseudozufallszahl wird für jeden Stützpunkt (Spline-Wert) erzeugt. Danach wird die Zufallszahl mit dem angegeben Faktor (hier 0,25) und dem Spline-Wert multipliziert. Das Ergebnis der Multiplikation wird anschließend mit dem Spline-Wert addiert.
ModMaximaXTime	0,25	
ModMaximaYVal	0,25	
ModMaximaYTime	0,25	
ModMaximaPVal	0,25	
ModMaximaPTime	0,25	
ModSampleRate	10	10 Millisekunden (ms) Abtastrate
ModLetterDistance	10	10 Pixel Abstand zwischen zwei Zeichen

In Tabelle 51 sind die Resultate der Modifikation dargestellt. Die Gleichfehlerrate konnte entsprechend erhöht werden, ist jedoch gegenüber realer Handschriftendaten immer noch sehr gering.

Tabelle 51 Fehlerraten der 4. Iteration künstlicher Handschriftendaten

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	45,81	8,36	0	0,1818	20,04
arbeiten	46,54	6,90	0	0,0517	24,71
Iraq	42,81	14,36	0	0,7273	29,11
Seife	45,36	9,27	0	0,0522	19,71

Äußerlich unterscheiden sich die künstlichen Handschriftensignale der 4. Iteration (siehe Abbildung 91) nicht sonderlich von denen der 3. Iteration, da sich die Modifikationsparameter *mp* der Schreibindividuen nicht geändert haben.

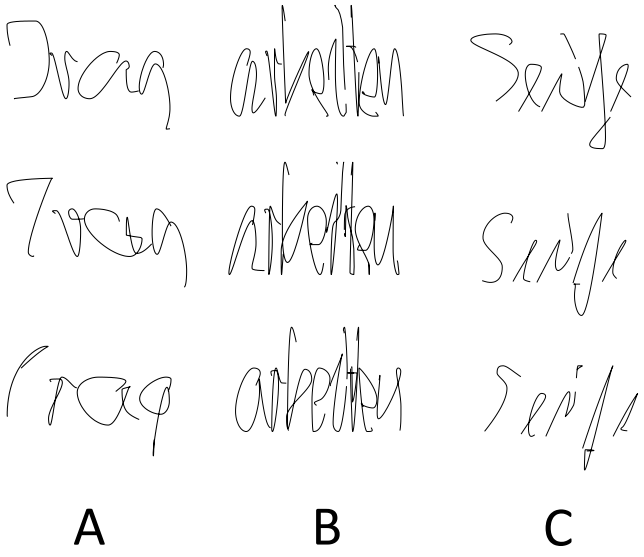


Abbildung 91 Beispielhafte Darstellung künstlicher Schreibsignale (4. Iteration)

Ziel der nächsten Iteration ist es, optisch bessere künstliche Schreibsignale bei gleichbleibenden oder besseren Fehlerraten zu erzielen. Hierfür wurden die Fehlerraten für die nächste Iteration leicht angepasst. Die Werte der Modifikationsparameter *mp* (Tabelle 52) der künstlichen Schreibindividuen nähern sich entsprechend an. Zusätzlich wurden die Parameter *mpS* geändert (siehe Tabelle 53), damit sich die künstlichen Schreibsignale eines Individuums optisch nicht so stark voneinander unterscheiden.

Tabelle 52 Modifikationsparameter zur Erzeugung der künstlichen Handschriftendaten (5. It.)

Mod. Parameter	**Künstliches Basisalphabet (Schreibindividuum)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ModTtotal	0,85	0,9	1	1,05	1,1	1,05	1	0,95	0,9	0,85
ModXYRatio	1,10	1,10	1,15	1,20	1,20	1,15	1,10	1,20	1,10	1,15
ModMaxPressure	0,5	0,6	0,7	1	1,1	1,2	0,5	0,7	0,6	1,5
ModMaximaXVal	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModMaximaXTime	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModMaximaYVal	0	0	0	0,01	0,015	0,02	0,01	0,015	0,02	0,03
ModMaximaYTime	0	0	0	0,01	0,015	0,02	0,01	0,015	0,02	0,03
ModMaximaPVal	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModMaximaPTime	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModFontSlam	0	0	0	0	0	0	0	0	0	0

Tabelle 53 Modifikationsparameter zur Erzeugung der künstlichen Handschriftendaten (5.It.)

Mod. Parameter	Künstliches Handschriftensample	Hinweis / Erklärung
	1 – 10	
InterpolMethode	Pchip	Piecewise Cubic Hermite Interpolating Polynomial (Kubisch Hermitescher Spline)
ModMaximaXVal	0,07	Eine Pseudozufallszahl wird für jeden Stützpunkt (Spline-Wert) erzeugt. Danach wird die Zufallszahl mit dem angegeben Faktor (hier 0,07) und dem Spline-Wert multipliziert. Das Ergebnis der Multiplikation wird anschließend mit dem Spline-Wert addiert.
ModMaximaXTime	0,07	
ModMaximaYVal	0,07	
ModMaximaYTime	0,07	
ModMaximaPVal	0,07	
ModMaximaPTime	0,07	
ModSampleRate	10	10 Millisekunden (ms) Abtastrate
ModLetterDistance	10	10 Pixel Abstand zwischen zwei Zeichen

Tabelle 54 zeigt die Fehlerraten der 5. Iteration. Die Gleichfehlerrate konnte bei drei Schreibsemantiken verbessert werden, bei der Semantik „Iraq“ gelang dies nicht. Die Reproduktionsraten konnten jedoch in allen Semantiken leicht verbessert werden.

Tabelle 54 Fehlerraten der 5. Iteration künstlicher Handschriftendaten

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	44,00	12,00	0	0,365	21,99
arbeiten	43,27	13,45	0	0,1818	20,00
Iraq	42,18	15,63	0	0,5455	27,59
Seife	44,63	10,72	0	0,3214	23,23

Das Erscheinungsbild der künstlichen Handschriftendaten ist in Abbildung 92 dargestellt. Es ist zu erkennen, dass sich die Schreibindividuen nicht mehr signifikant voneinander unterscheiden.

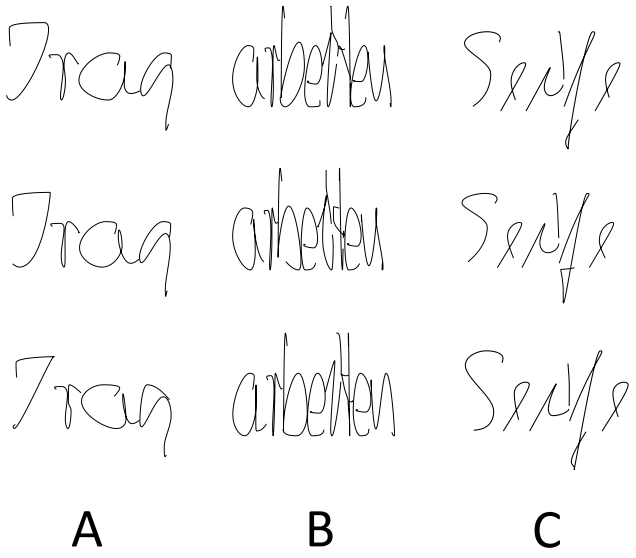


Abbildung 92 Beispielhafte Darstellung künstlicher Schreibsignale (5. Iteration)

Um den Einfluss der Modifikationsparameter *mp* auf die Fehlerraten weiter zu untersuchen, werden diese nochmals angepasst. Die Werte der Parameter nähern sich gegenüber der vorherigen Iteration noch weiter an (siehe Tabelle 55). Die Modifikationsparameter *mpS* bleiben bei dieser Iteration gleich.

Tabelle 55 Modifikationsparameter zur Erzeugung der künstlichen Handschriftendaten (6. It.)

Mod. Parameter	Künstliches Basisalphabet (Schreibindividuum)									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ModTtotal	0,99	0,98	1,0	1,01	1,02	1,05	0,97	0,95	0,96	1,03
ModXYRatio	1,10	1,10	1,15	1,20	1,20	1,15	1,10	1,20	1,10	1,15
ModMaxPressure	0,7	0,8	0,9	1	1,1	1,2	1,1	1	0,9	0,8
ModMaximaXVal	0,01	0,015	0,015	0	0	0	0,01	0,015	0,015	0,01
ModMaximaXTime	0,01	0,015	0,015	0	0	0	0,01	0,015	0,015	0,01
ModMaximaYVal	0	0	0	0,01	0,015	0,007	0,01	0,015	0,005	0,005
ModMaximaYTime	0	0	0	0,01	0,015	0,007	0,01	0,015	0,005	0,005
ModMaximaPVal	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModMaximaPTime	0,01	0,015	0,02	0	0	0	0,01	0,015	0,02	0,03
ModFontSlam	0	0	0	0	0	0	0	0	0	0

Die Werte der erzielten Fehlerraten für die 6. Iteration sind in Tabelle 56 abgebildet. Diese haben sich gegenüber der vorherigen nicht merklich verändert. Einige Gleichfehlerraten konnten gesteigert werden („2759“, „Iraq“) andere hingegen sind gefallen („arbeiten“, „Seife“).

Tabelle 56 Fehlerraten der 6. Iteration künstlicher Handschriftendaten

Semantik	CRR in %	RR in %	CR in %	EER in %	EER Th.
2759	45,00	10,00	0	0,44	19,53
arbeiten	43,81	12,36	0	0,06	19,61
Iraq	42,36	15,27	0	1,09	25,56
Seife	44,45	11,09	0	0,30	21,32

In Abbildung 93 sind beispielhaft künstliche Schreibsignale der 6. Iteration abgebildet. Der Unterschied zwischen den jeweiligen künstlichen Schreibindividuen (von oben nach unten) ist nicht mehr erkennbar. Das Erscheinungsbild der künstlichen Handschriftensignale ist jedoch gut.

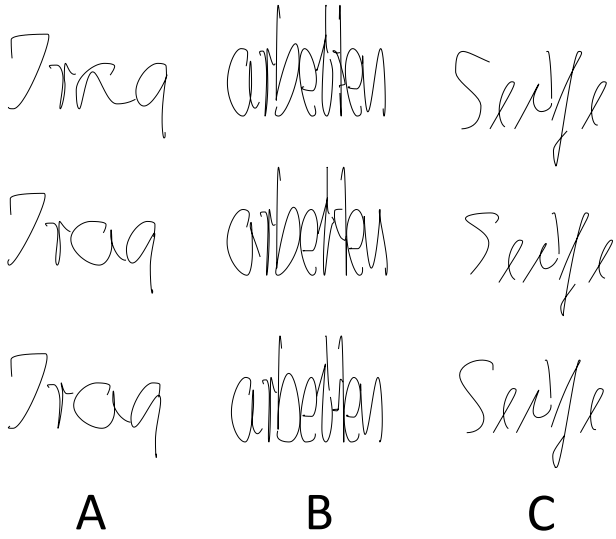


Abbildung 93 Beispielhafte Darstellung künstlicher Schreibsignale (6. Iteration)

Schlussendlich sollen die Modifikationsparameter der 4. Iteration für den weiteren Verlauf der Evaluation verwendet werden. Die Fehlerraten sind zwar geringer als die realer vergleichbarer Handschriftendaten, jedoch ist ein optischer Unterschied zwischen den jeweiligen künstlichen Schreibindividuen noch zu erkennen. In Abbildung 94 sind die Veränderungen über die Iterationen zu erkennen. Die Schreibindividuen (Ind. 1 bis Ind. 3) unterscheiden sich noch in der 3. Iteration (1. Spalte) recht gut, wohingegen in der 6. Iteration (letzte Spalte) eine Unterscheidung nicht ohne weiteres möglich ist.

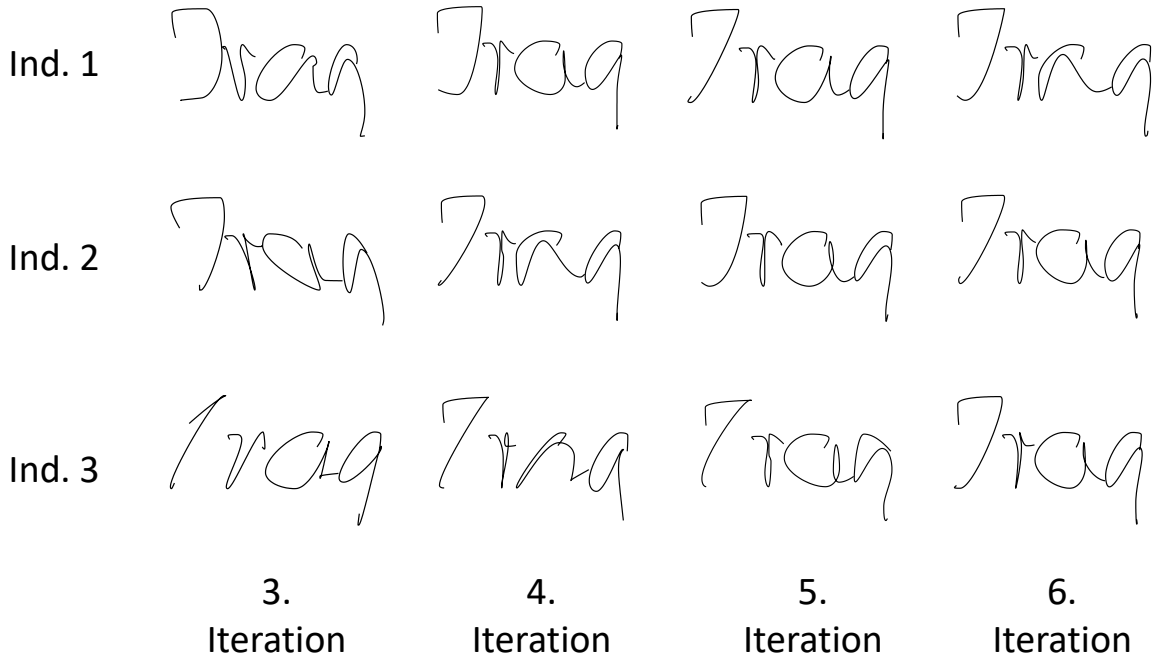


Abbildung 94 Auswirkungen der Modifikationsparameter *mp* auf das Erscheinungsbild

Anlage 9 Beispielhafte Darstellung künstlicher Handschriftensignale

Nachfolgend werden künstliche Schreibsignale in verschiedenen Semantiken dargestellt. Eine Zeile repräsentiert zehn Schreibsignale eines künstlichen Schreibindividuums. Die Angabe des "users" zeigt, auf welchen realen Handschriftendaten die künstlichen Schreibindividuen gebildet wurden.

Abbildung 95 Schreibsignale künstlicher Schreibindividuen der Semantik "Iraq" (User 3)

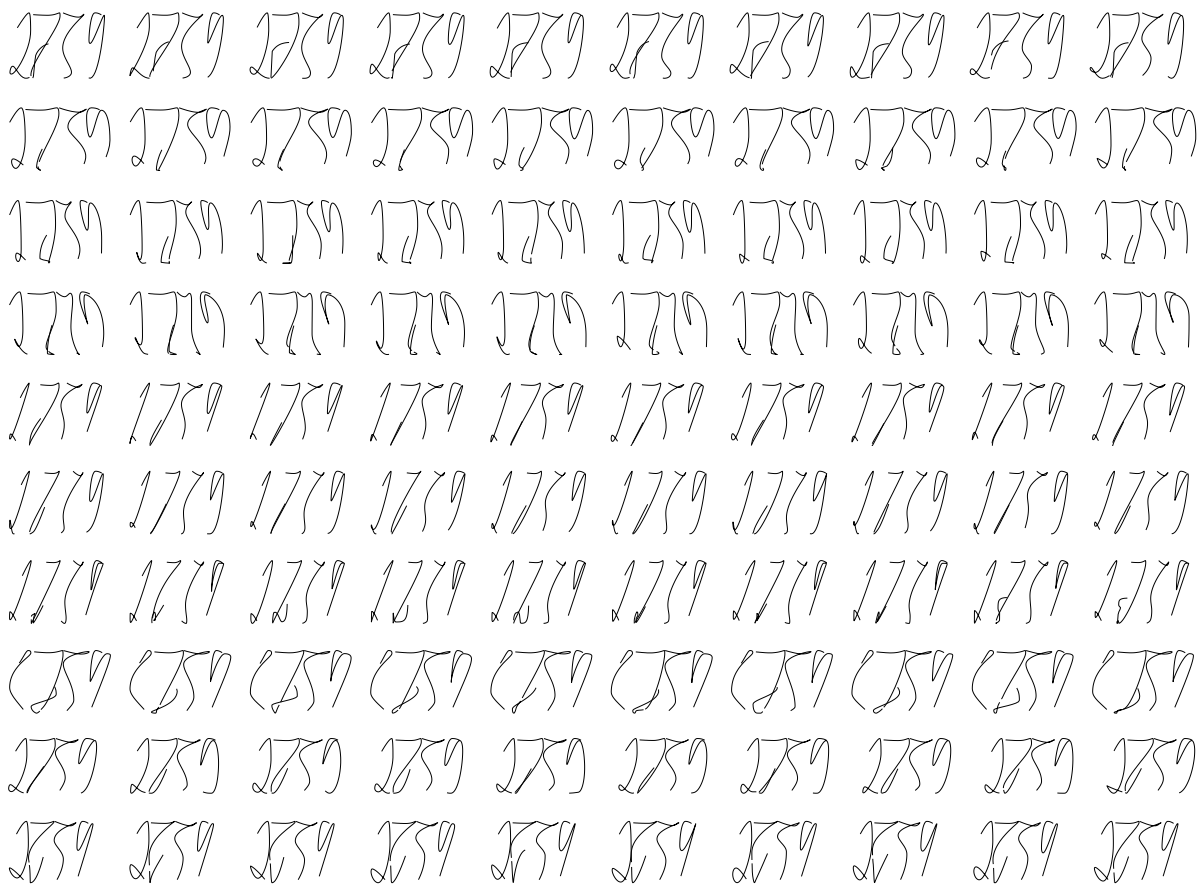


Abbildung 96 Schreibsignale künstlicher Schreibindividuen der Semantik "2759" (User 1)

Abbildung 97 Schreibsignale künstlicher Schreibindividuen der Semantik "arbeiten" (User 11)

Anlage 10 Verifikationsperformanz der künstlich generierten Handschriftendaten

Nachfolgend werden die Kurvenverläufe der FAR und FRR zur Visualisierung der Verifikationsperformanz dargestellt. Die verwendete Semantik wird in den Abbildungen selbst und in der jeweiligen Abbildungsbeschriftung in Klammern angegeben.

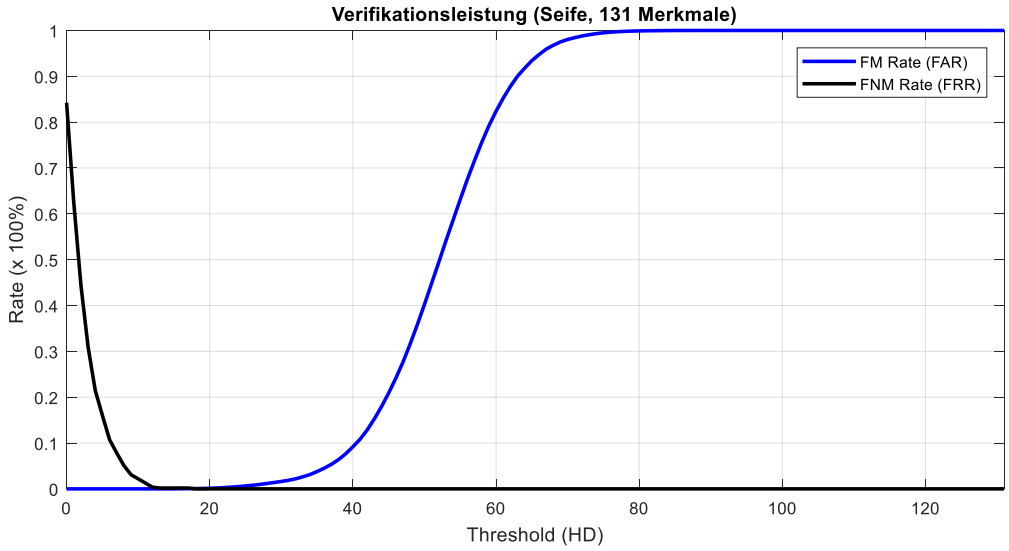


Abbildung 98 Verifikationsperformanz künstlicher Handschriftensignale (Seife)

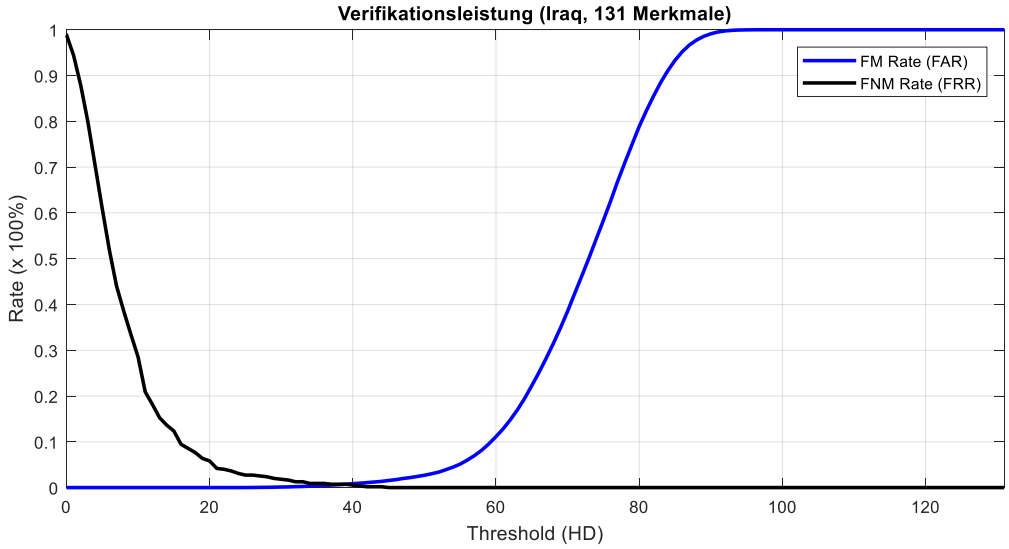


Abbildung 99 Verifikationsperformanz künstlicher Handschriftensignale (Iraq)

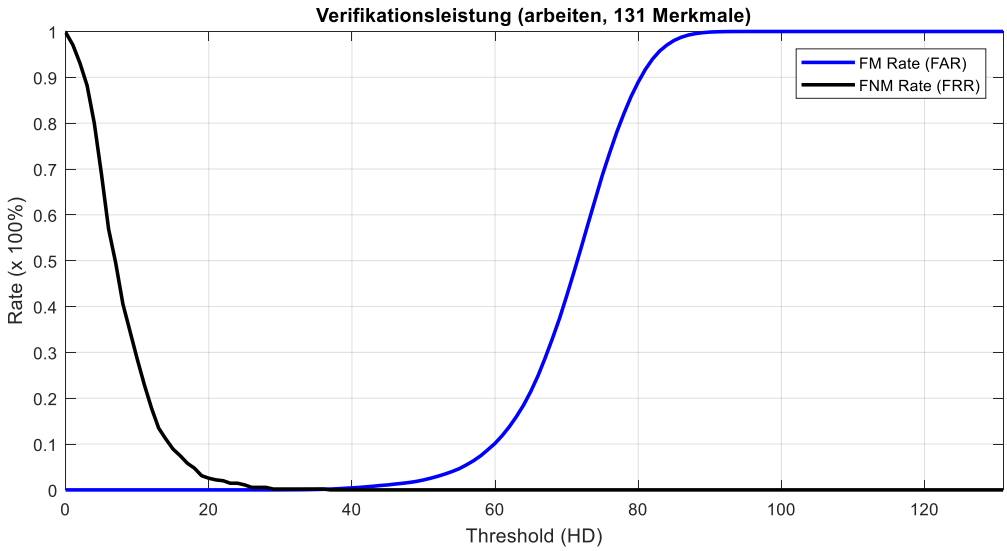


Abbildung 100 Verifikationsperformanz künstlicher Handschriftensignale (arbeiten)

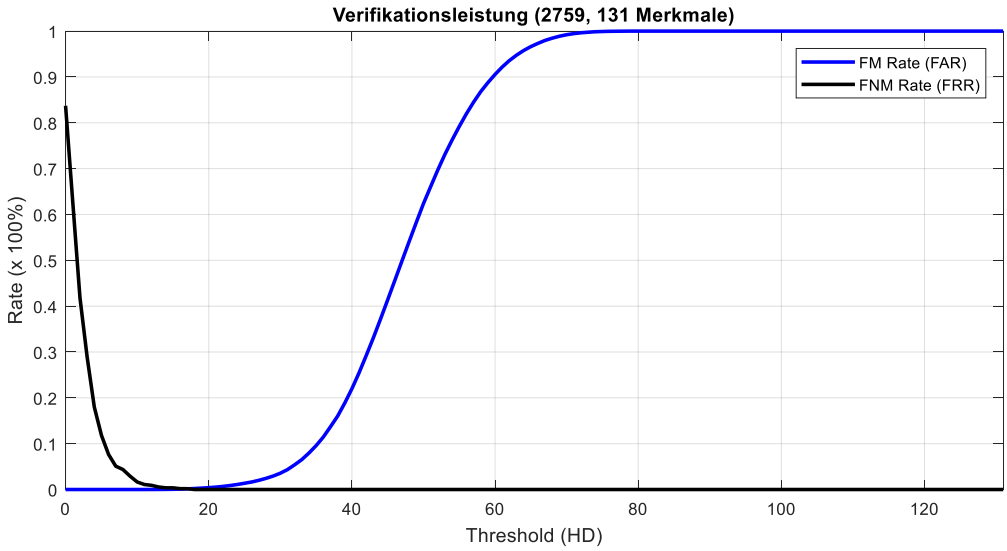


Abbildung 101 Verifikationsperformanz künstlicher Handschriftensignale (2759)

Anlage 11 Bestimmung der Toleranzfaktoren künstlicher Handschriftensignale

Nachfolgend werden die Diagramme zur Bestimmung des optimalen Toleranzfaktors für die Arbeitsmodi (EER und CRR) der künstlichen Handschriftensignale dargestellt.

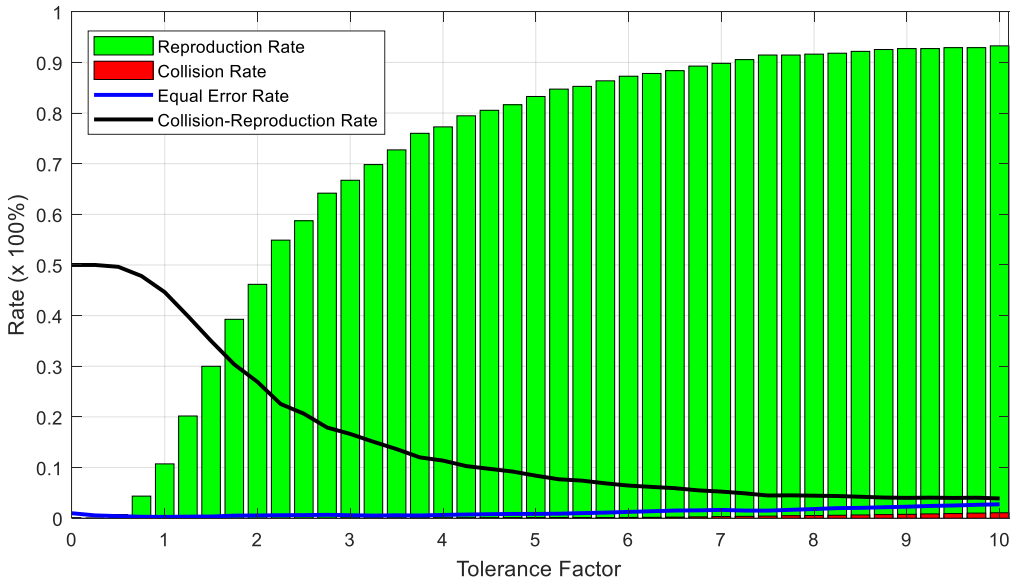


Abbildung 102 Bestimmung des Toleranzfaktors künstlicher Signale (Seife)

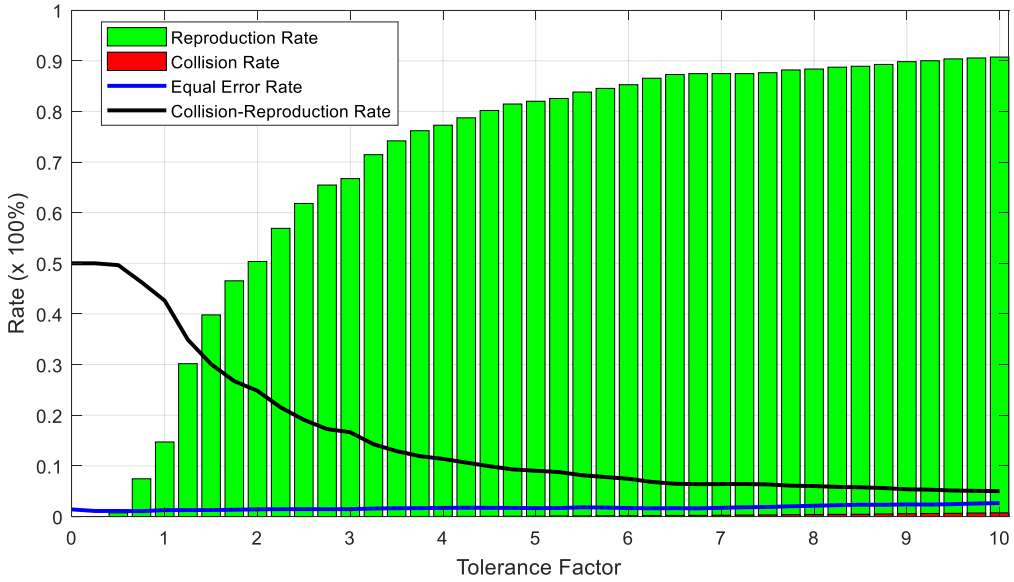


Abbildung 103 Bestimmung des Toleranzfaktors künstlicher Signale (Iraq)

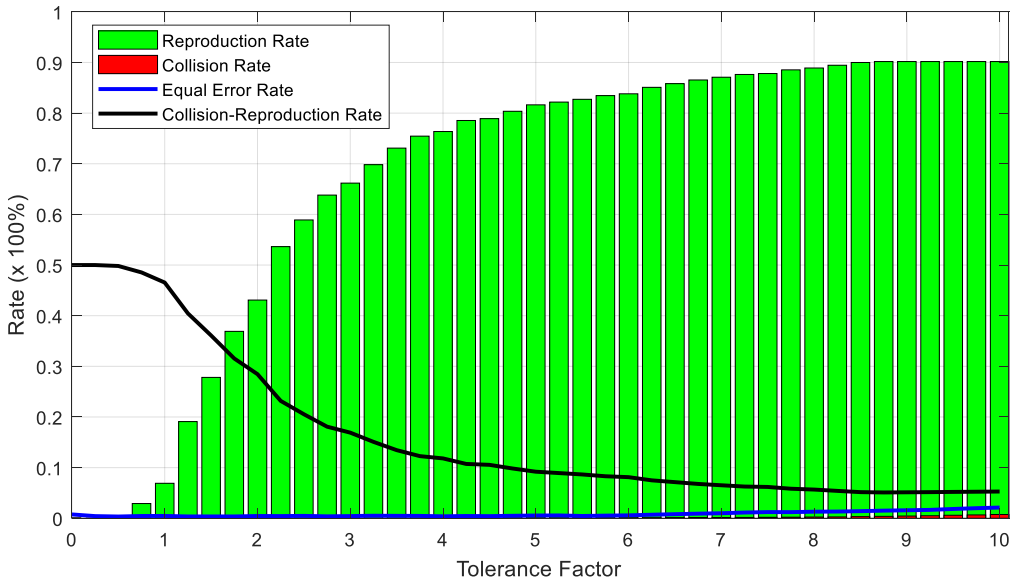


Abbildung 104 Bestimmung des Toleranzfaktors künstlicher Signale (arbeiten)

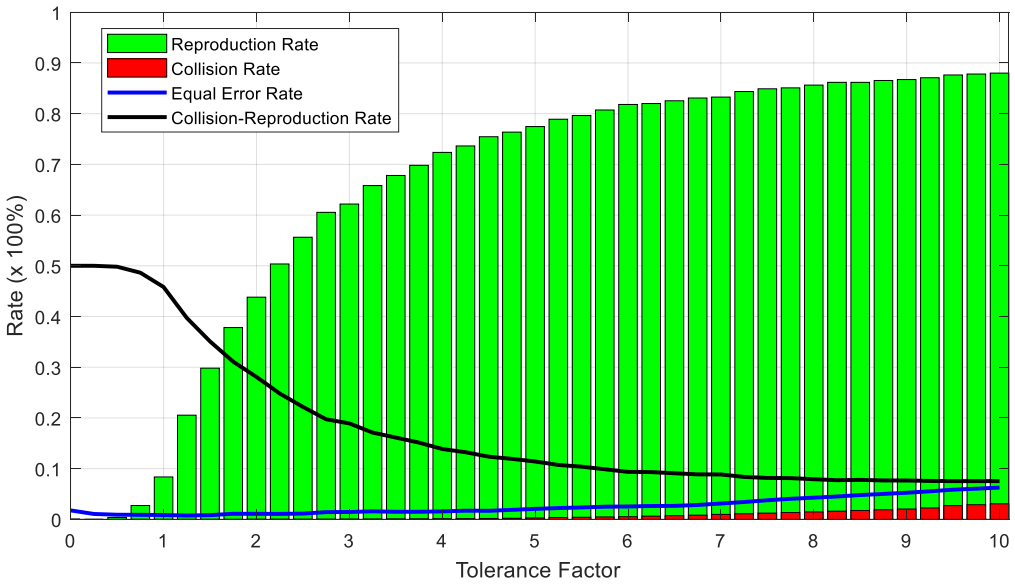


Abbildung 105 Bestimmung des Toleranzfaktors künstlicher Signale (2759)

Anlage 12 Kurvenverlauf der Fehlerraten zur Bestimmung der Angriffsperformanz (FA2)

Zur Bestimmung der Angriffsperformanz werden die Fehlerratenkurzen entsprechend der Semantike dargestellt. Im ersten Teil sind die Fehlerraten aufgelistet, welche mit zehn Angriffsdaten pro User und entsprechender Semantik durchgeführt wurden. Im darauffolgenden Teil sind die Fehlerkurven dargestellt, die 100 Angriffssample pro User und Semantik verwendeten.

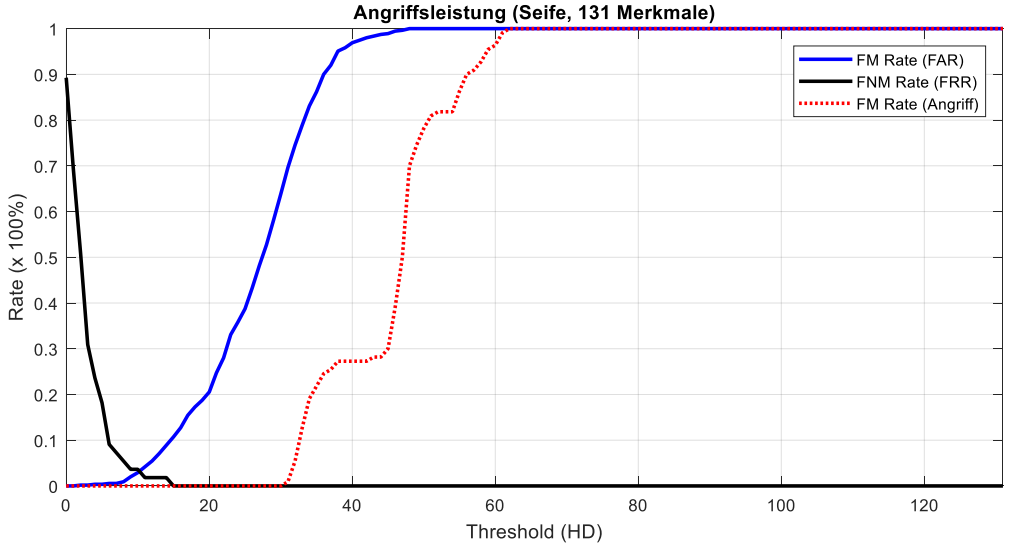


Abbildung 106 Angriffsperformanz für die Semantik Seife (10 Angriffssamples)

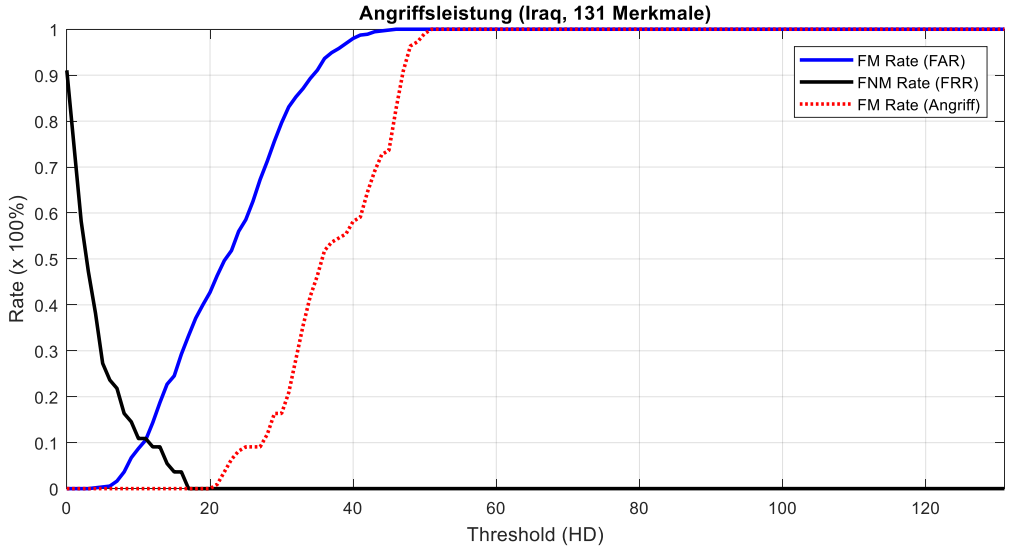


Abbildung 107 Angriffsperformanz für die Semantik Iraq (10 Angriffssamples)

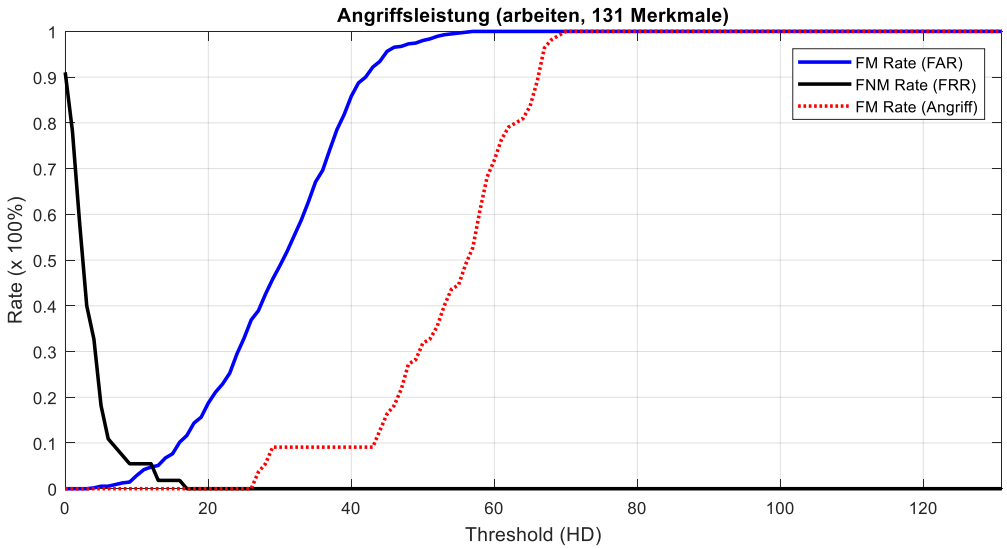


Abbildung 108 Angriffsperformanz für die Semantik arbeiten (10 Angriffssamples)

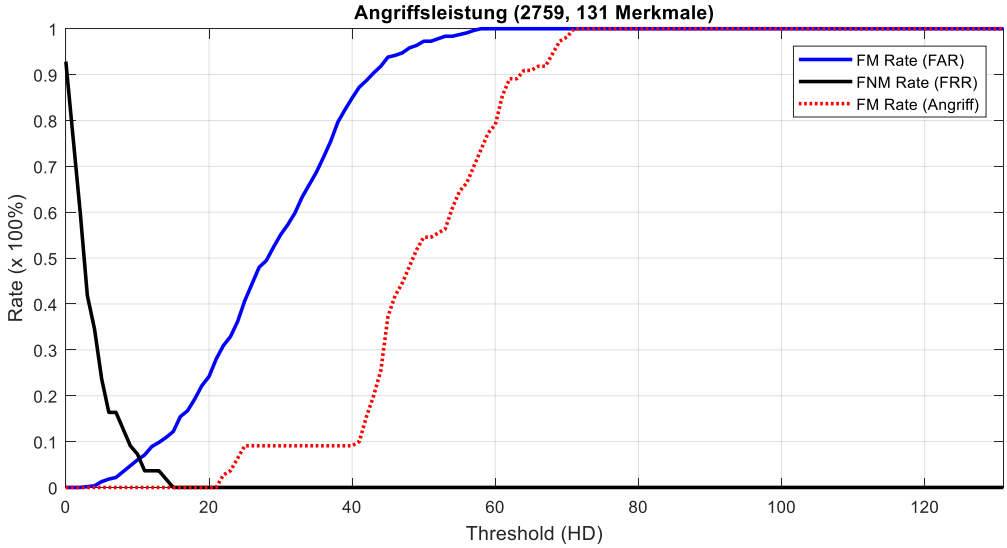


Abbildung 109 Angriffsperformanz für die Semantik 2759 (10 Angriffssamples)

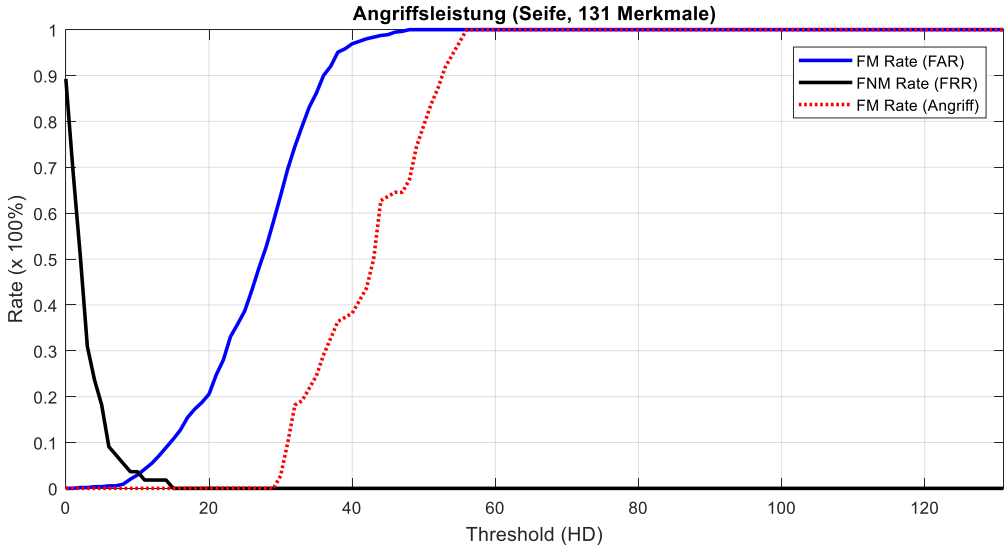


Abbildung 110 Angriffsperformanz für die Semantik Seife (100 Angriffssamples)

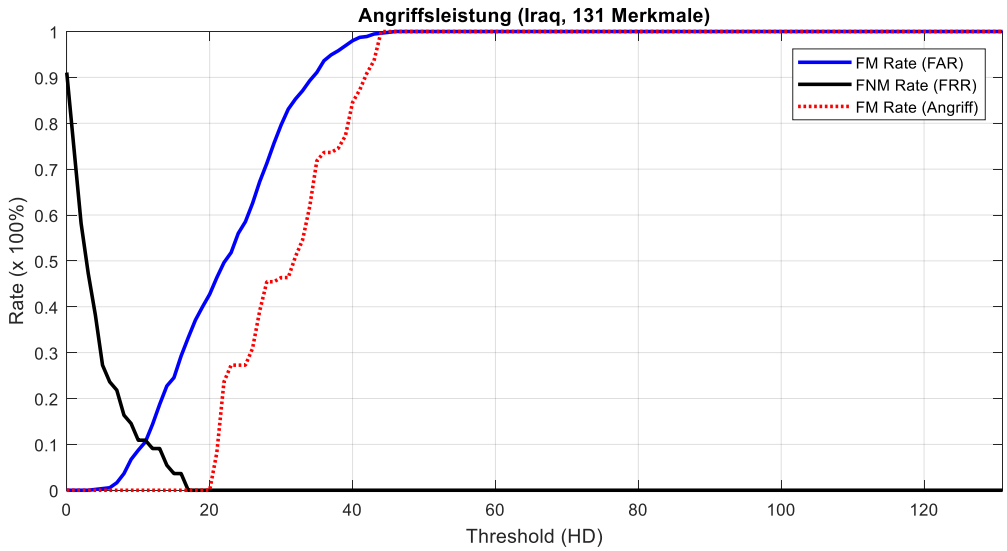


Abbildung 111 Angriffsperformanz für die Semantik Iraq (100 Angriffssamples)

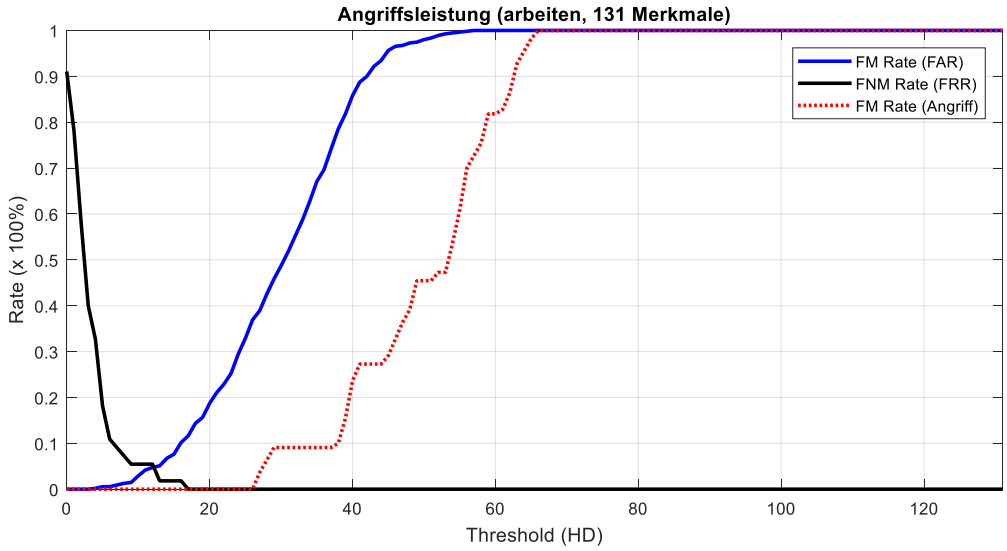


Abbildung 112 Angriffsperformanz für die Semantik arbeiten (100 Angriffssamples)

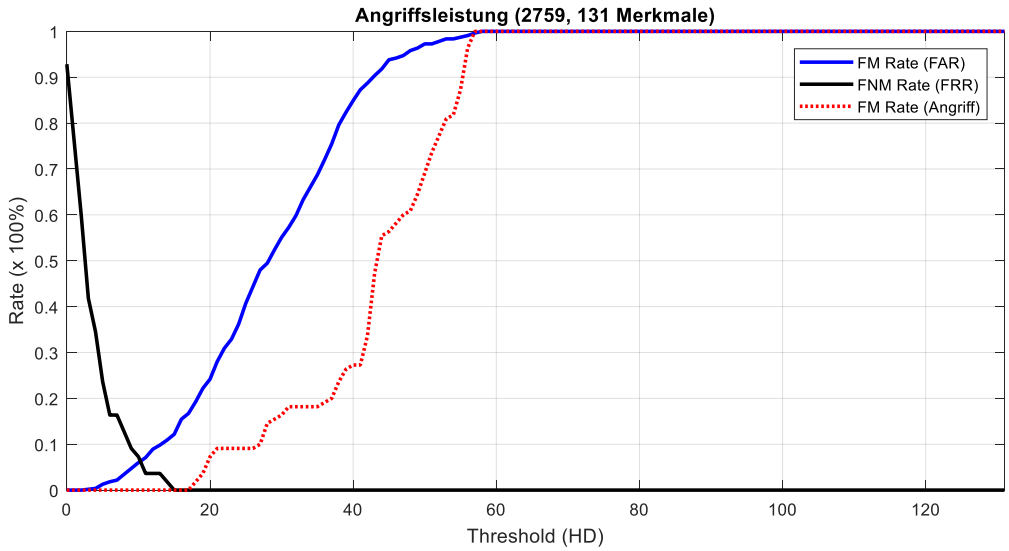


Abbildung 113 Angriffsperformanz für die Semantik 2759 (100 Angriffssamples)

Anlage 13 Bestimmung der Schwellenwerte für die jeweiligen Arbeitsbereiche des HC-Algorithmus (FA3)

Nachfolgend werden alle Diagramme dargestellt die für die Bestimmung der Schwellenwerte (Threshold) und somit eines der Abbruchkriterien für den HC-Algorithmus verwendet wurden.

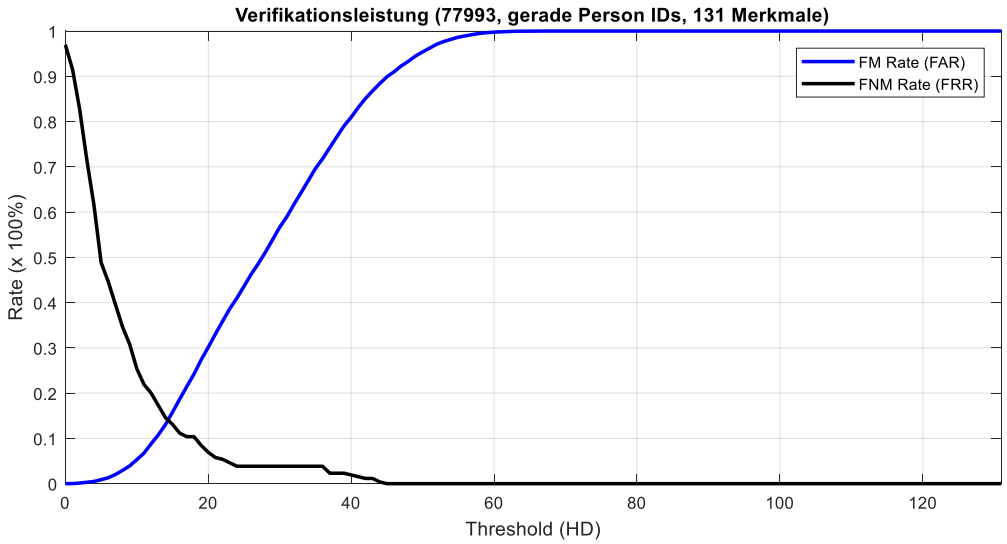


Abbildung 114 Verifikationsleistung aller geraden Personen-IDs der Semantikklasse 77993

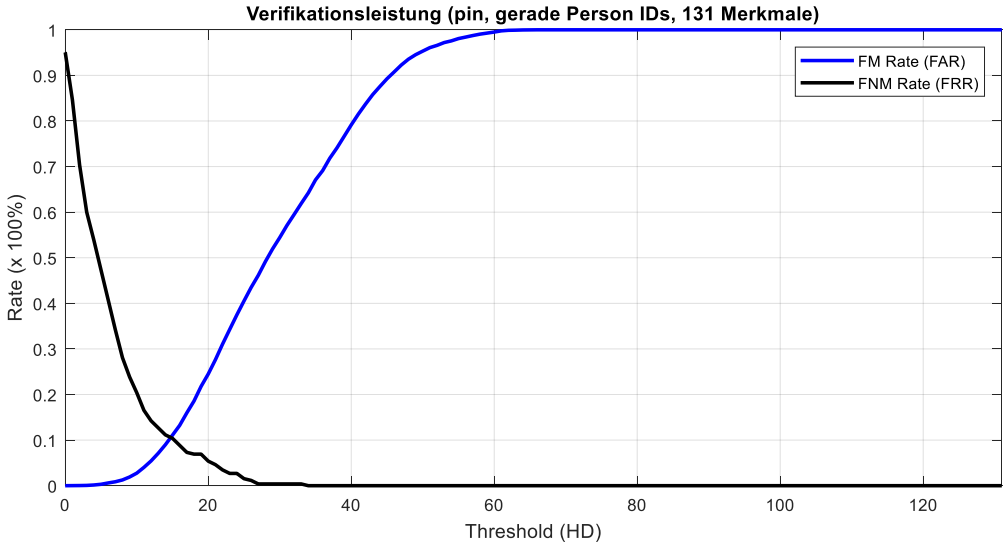


Abbildung 115 Verifikationsleistung aller geraden Personen-IDs der Semantikklasse feste PIN

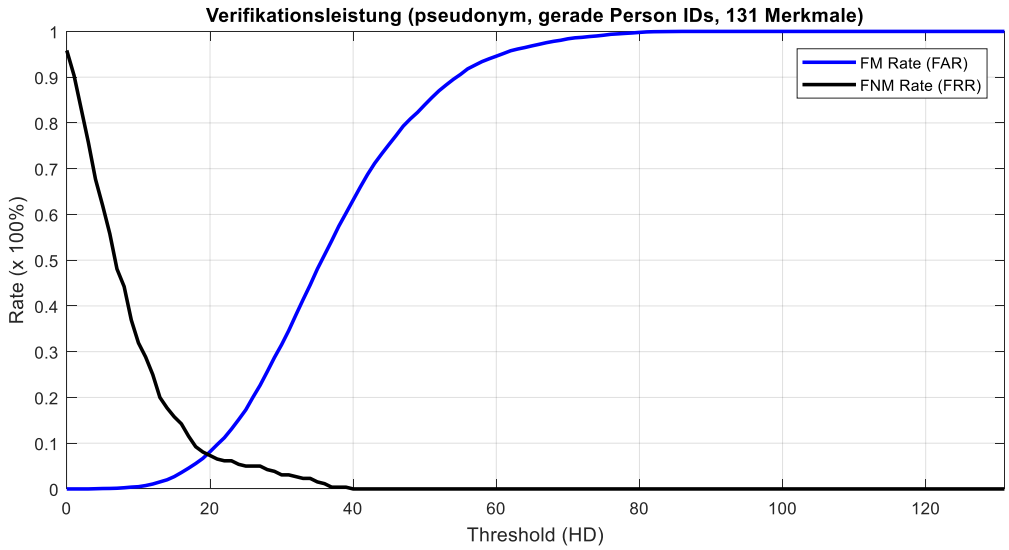


Abbildung 116 Verifikationsleistung aller geraden Personen-IDs der Semantikklasse Pseudonym

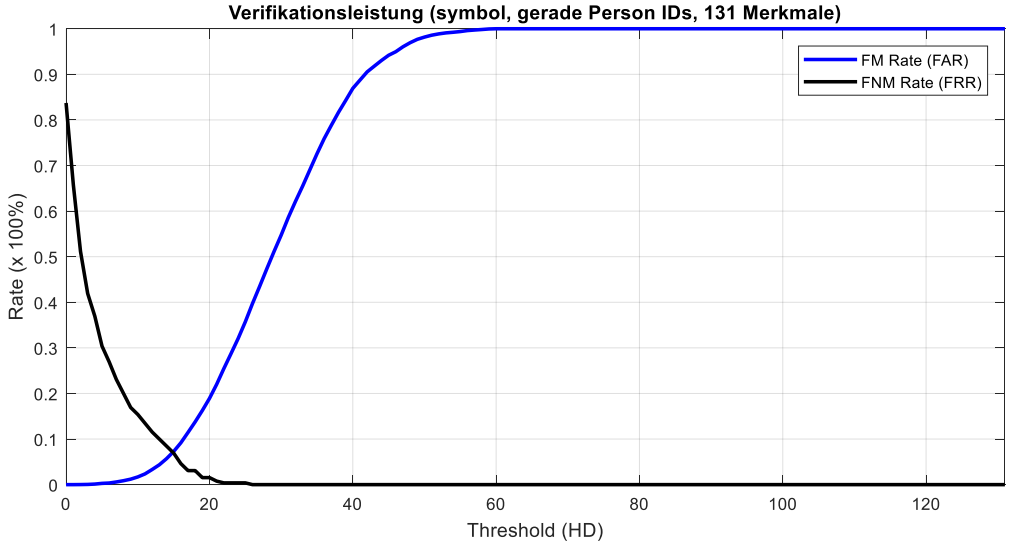


Abbildung 117 Verifikationsleistung aller geraden Personen-IDs der Semantikklasse Symbol

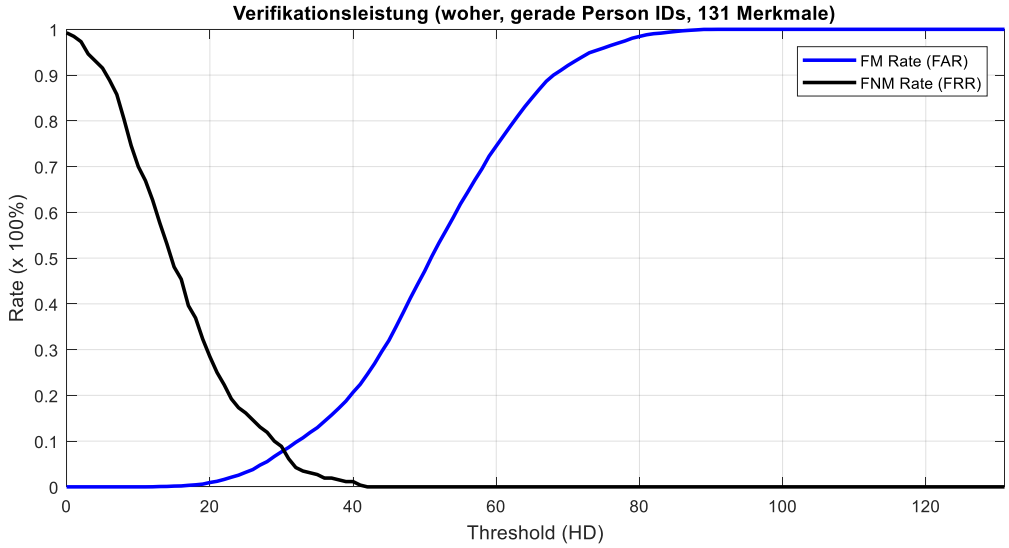


Abbildung 118 Verifikationsleistung aller geraden Personen-IDs der Semantikklasse Woher

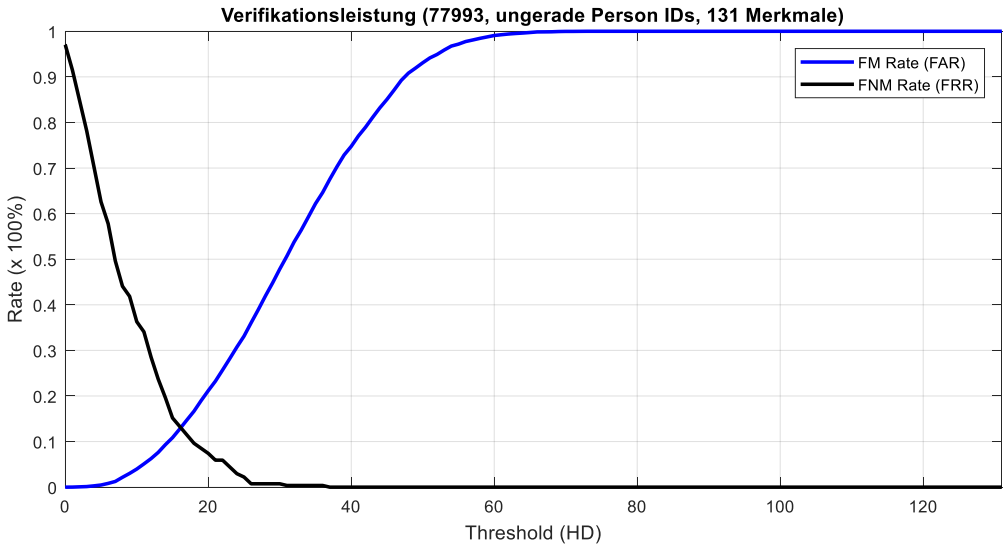


Abbildung 119 Verifikationsleistung aller ung. Personen-IDs der Semantikklasse 77993

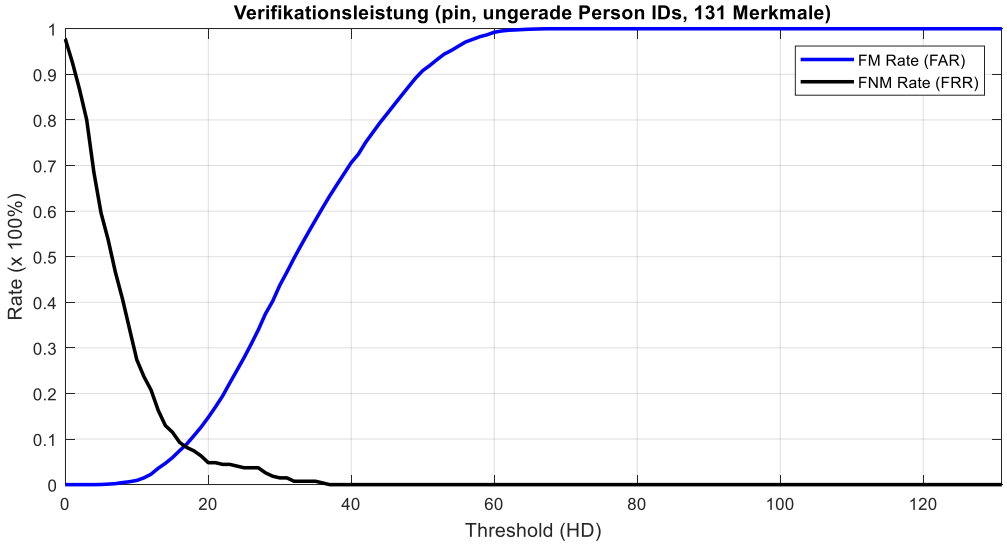


Abbildung 120 Verifikationsleistung aller ung. Personen-IDs der Semantikklasse feste PIN

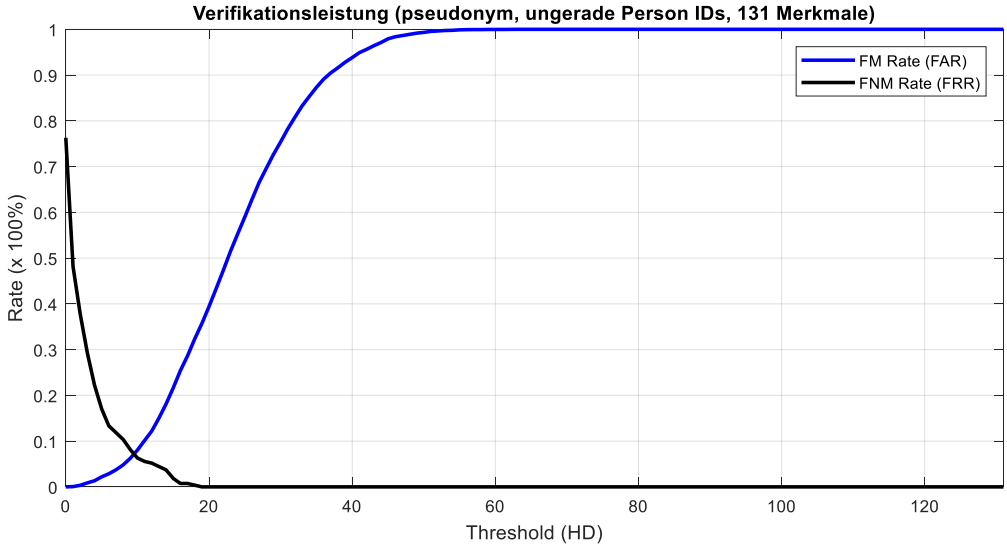


Abbildung 121 Verifikationsleistung aller ung. Personen-IDs der Semantikklasse Pseudonym

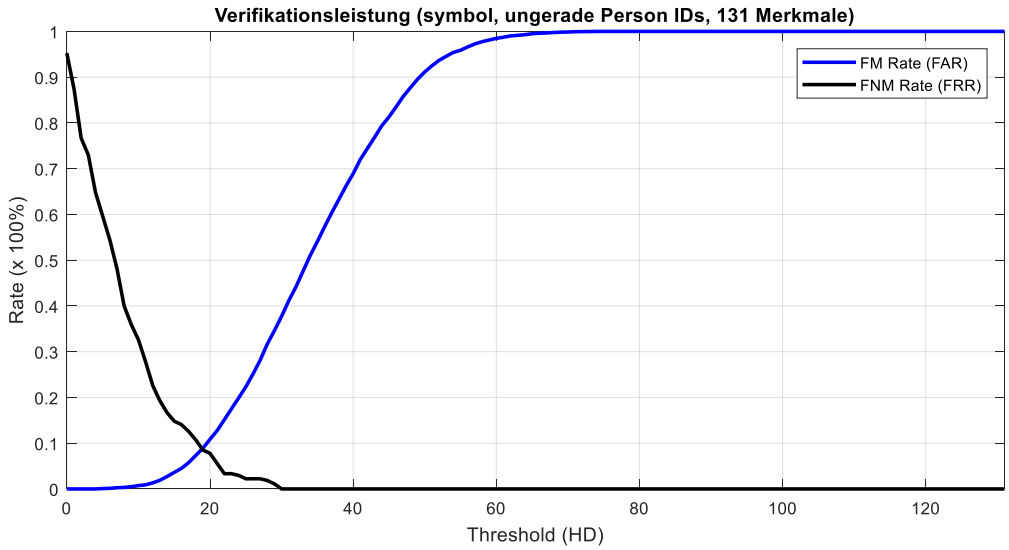


Abbildung 122 Verifikationsleistung aller ung. Personen-IDs der Semantikklasse Symbol

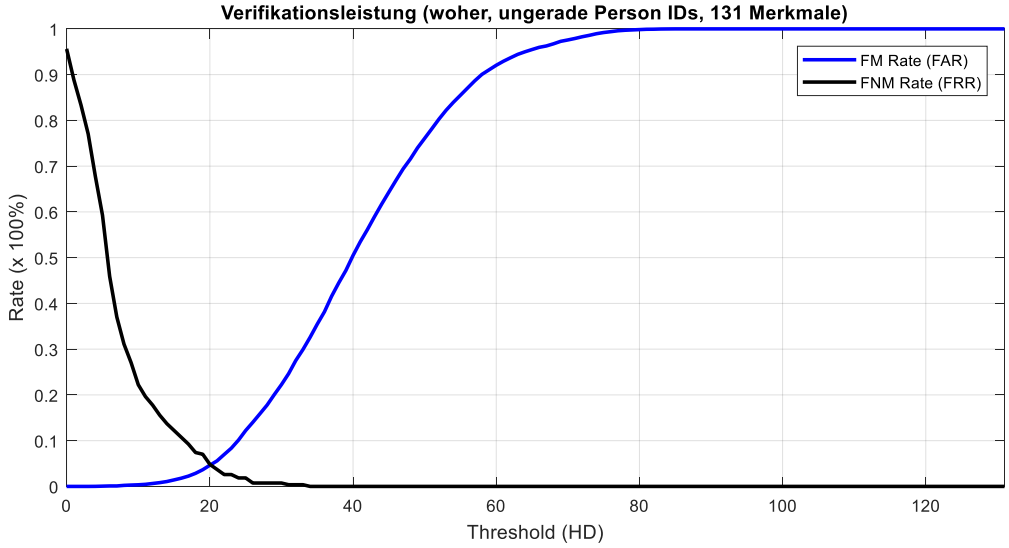


Abbildung 123 Verifikationsleistung aller ung. Personen-IDs der Semantikklasse Woher

Anlage 14 Bestimmung der Toleranzfaktoren für die jeweiligen Semantiken (FA3)

Nachfolgend werden alle Diagramme angegeben, welche für die Bestimmung des globalen Optimierungsfaktors (Toleranzfaktor) verwendet wurden. Hierbei wurde jeweils der Toleranzfaktorwert gewählt, welcher die niedrigste Equal Error Rate (EER) bewirkt.

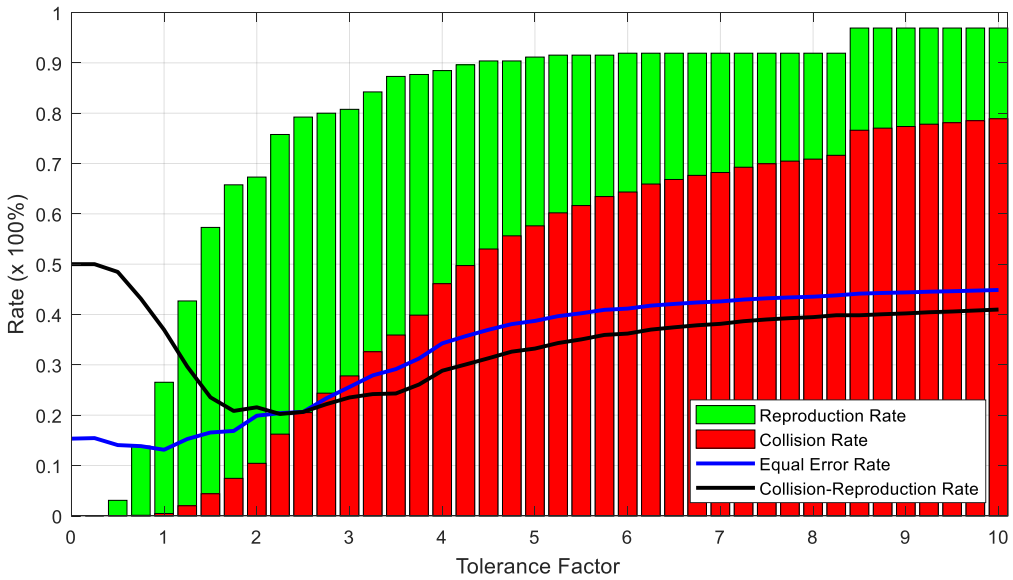


Abbildung 124 Bestimmung des Toleranzfaktors der Semantik 77993 (gerade Personen-IDs)

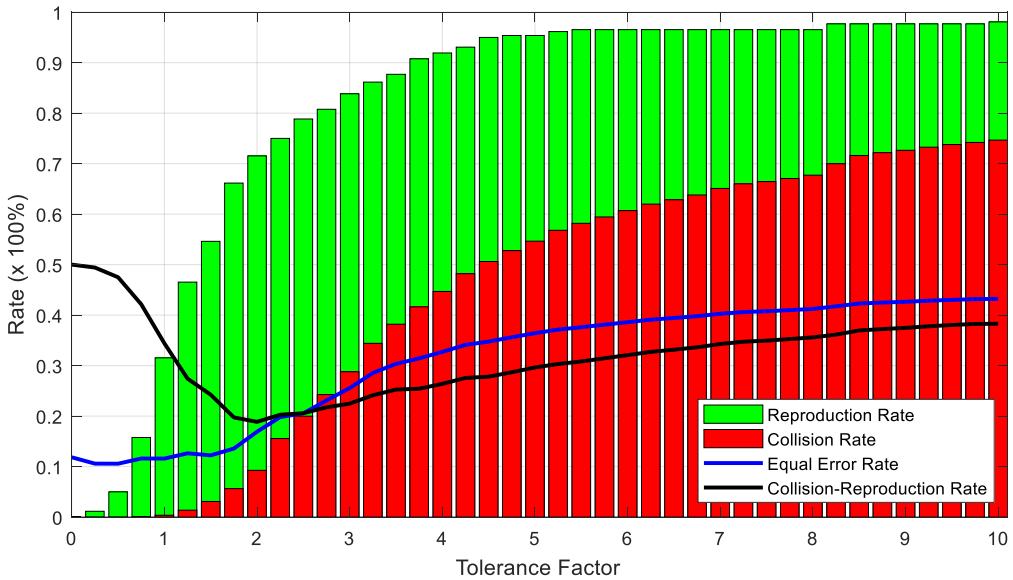


Abbildung 125 Bestimmung des Toleranzfaktors der Semantik feste PIN (gerade Personen-IDs)

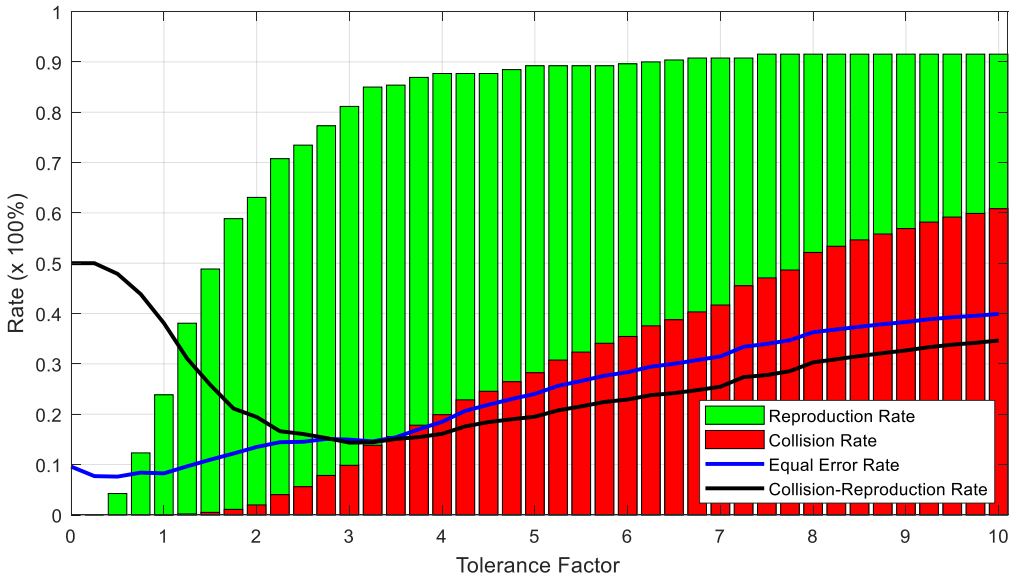


Abbildung 126 Bestimmung des Toleranzfaktors der Semantik Pseudonym (gerade Personen-IDs)

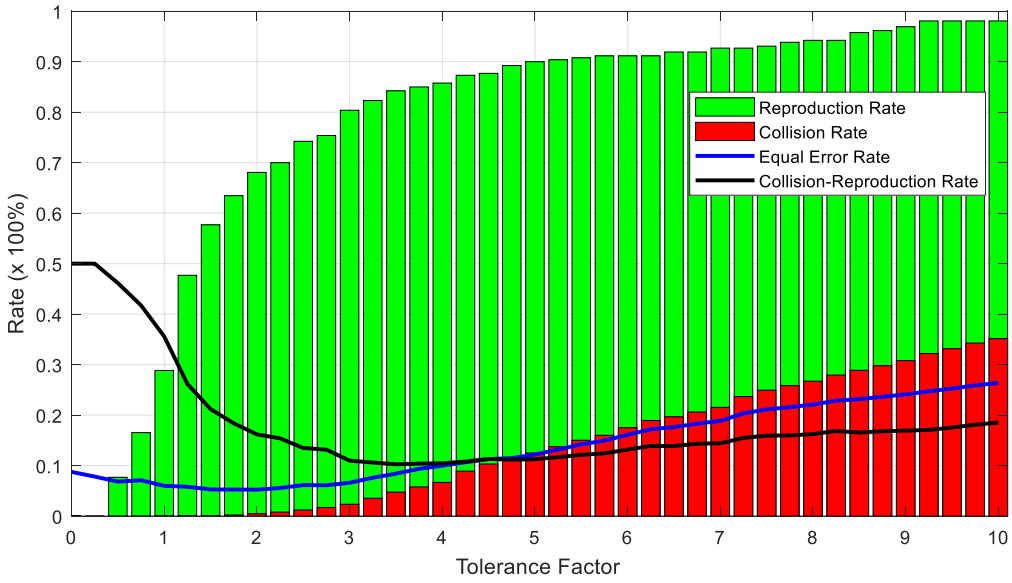


Abbildung 127 Bestimmung des Toleranzfaktors der Semantik Symbol (gerade Personen-IDs)

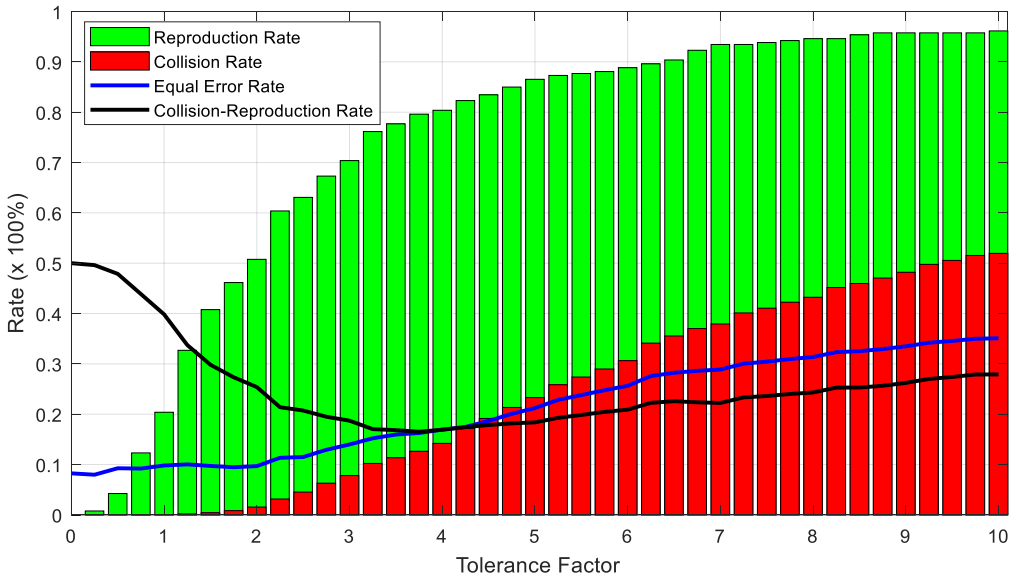


Abbildung 128 Bestimmung des Toleranzfaktors der Semantik Woher (gerade Personen-IDs)

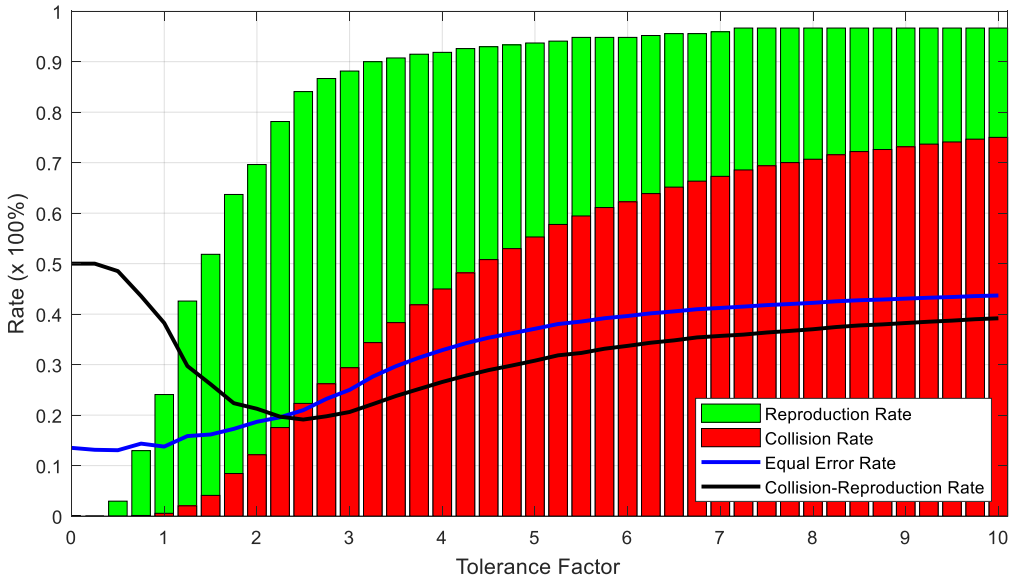


Abbildung 129 Bestimmung des Toleranzfaktors der Semantik 77993 (ungerade Personen-IDs)

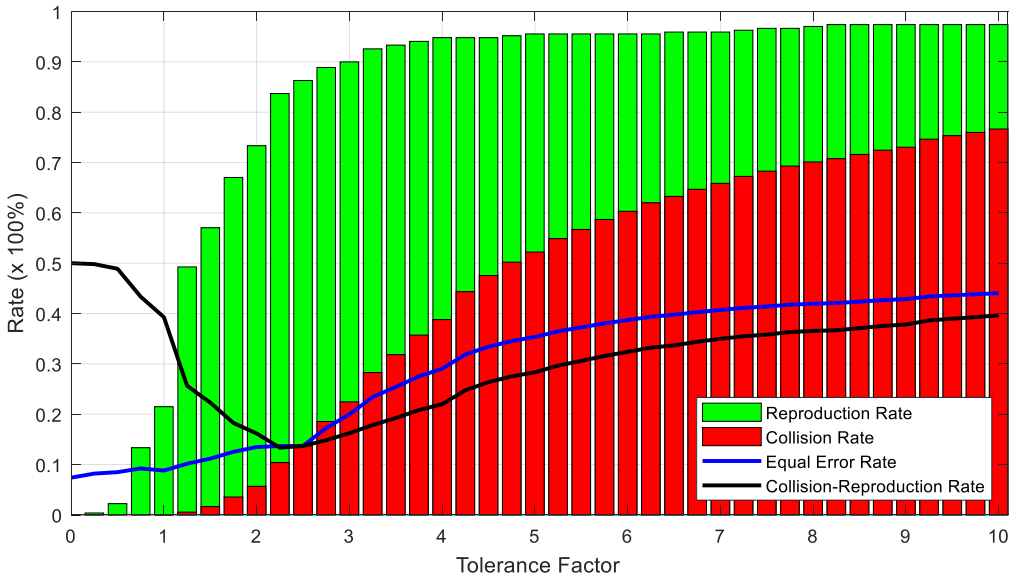


Abbildung 130 Bestimmung des Toleranzfaktors der Semantik feste PIN (ung. Personen-IDs)

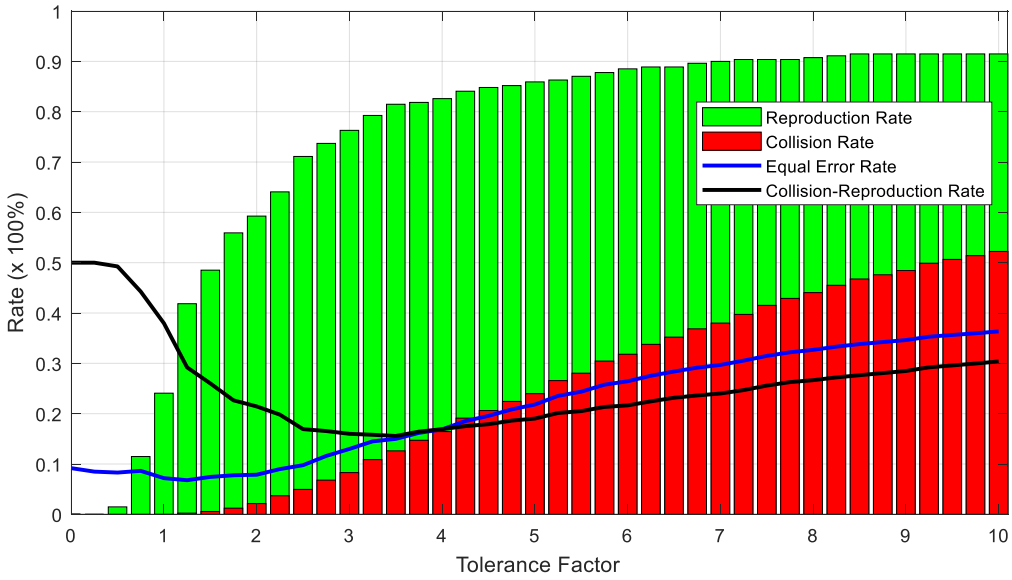


Abbildung 131 Bestimmung des Toleranzfaktors der Semantik Pseudonym (ung. Personen-IDs)

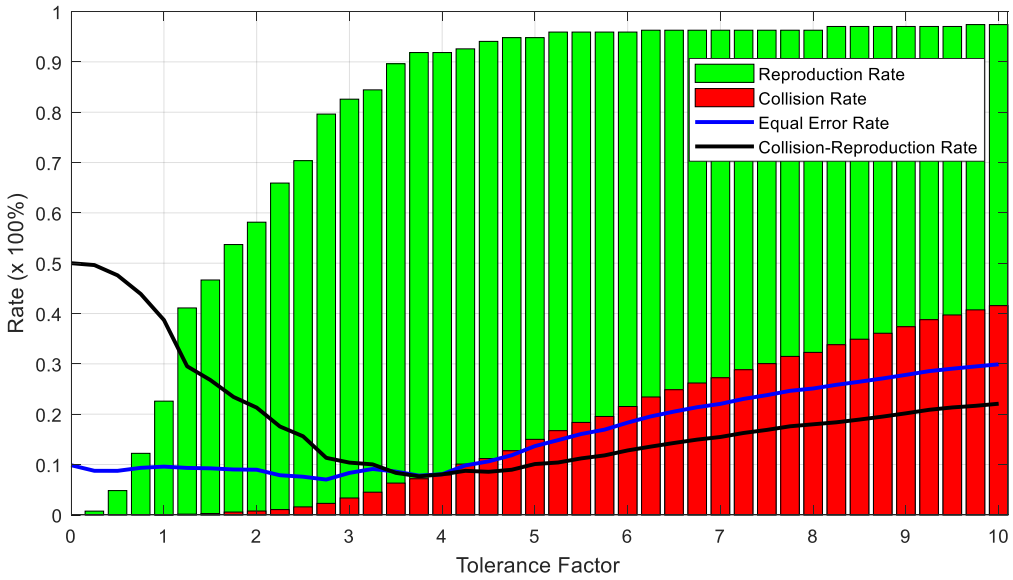


Abbildung 132 Bestimmung des Toleranzfaktors der Semantik Symbol (ung. Personen-IDs)

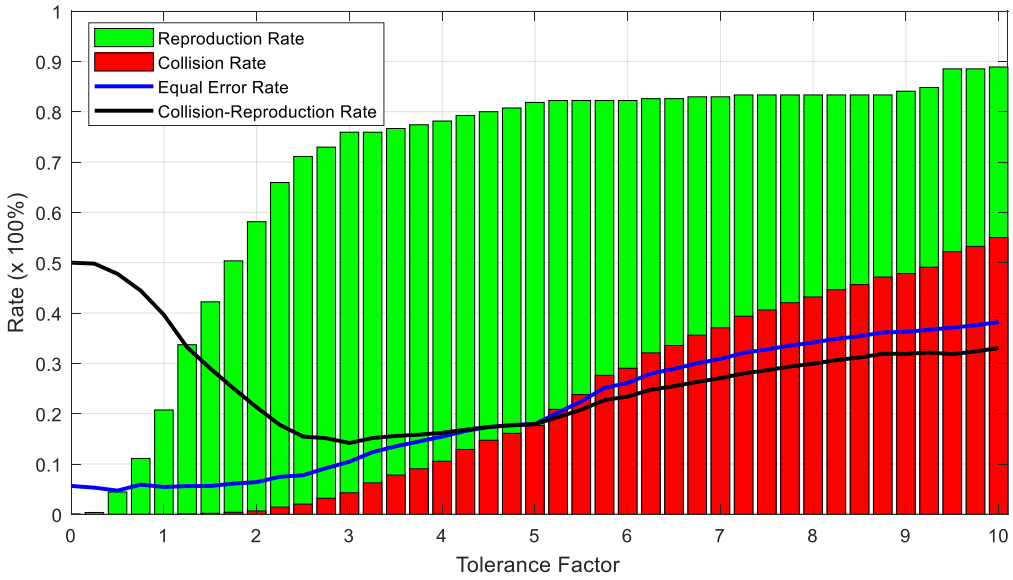


Abbildung 133 Bestimmung des Toleranzfaktors der Semantik Woher (ung. Personen-IDs)

Anlage 15 Bestimmung der N,M Wertepaare auf Basis der Erfolgsrate und Iterationen

Nachfolgend werden die Erfolgsraten und die dazugehörige Anzahl durchschnittlich benötigter Versuche (Iteration) für alle Semantikklassen dargestellt.

Tabelle 57 Erfolgsrate und Anzahl an Iterationen für die Semantik PIN (gerade Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	0	7,69 % (266,5)	11,54 % (152)	7,69 % (85)	7,69 % (64,5)
	5	0	11,54 % (584,33)	23,08 % (332,5)	26,92 % (152,43)	15,38 % (86,5)	15,38 % (70,75)
	10		11,54 % (436)	46,15 % (220,5)	**57,69 % (132,6)**	50 % (89,85)	26,92 % (60,14)
	25			19,23 % (296,2)	46,15 % (137,83)	46,15 % (90)	34,62 % (60,78)
	50				26,92 % (169,43)	38,46 % (92,7)	30,77 % (70,25)
	100					0	15,38 % (79)

Tabelle 58 Erfolgsrate und Anzahl an Iterationen für die Semantik Pseudonym(gerade Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	3,85 % (1503)	15,38 % (310,5)	30,77 % (170,13)	34,62 % (118,11)	23,08 % (66)	26,92 % (58,14)
	5	11,54 % (1596)	38,46 % (341,5)	57,69 % (210,67)	69,23 % (122)	61,54 % (73,81)	53,85 % (56,12)
	10		50 % (417)	76,92 % (163,65)	69,23 % (94,22)	73,08 % (65,37)	**76,92 % (59,05)**
	25			53,85 % (192,71)	84,62 % (116,05)	80,77 % (75,95)	65,38 % (57,24)
	50				65,38 % (157,71)	69,23 % (86,17)	50 % (60,69)
	100					30,77 % (103,25)	30,77 % (74,38)

Tabelle 59 Erfolgsrate und Anzahl an Iterationen für die Semantik Symbol (gerade Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	0	0	0	0	0
	5	0	0	0	0	0	0
	10		0	0	0	0	0
	25			0	0	0	0
	50				0	0	0
	100					0	0

Tabelle 60 Erfolgsrate und Anzahl an Iterationen für die Semantik Woher (gerade Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	38,46 % (711,9)	84,62 % (293,09)	92,31 % (115,88)	100 % (92,27)	92,31 % (62,58)	84,62 % (52,18)
	5	53,85 % (920,64)	100 % (238,19)	100 % (96,35)	100 % (68,62)	100 % (53,92)	96,15 % (46,36)
	10		96,15 % (206,28)	100 % (93,54)	100 % (67,08)	**100 % (50,38)**	100 % (45.85)
	25			100 % (154,54)	100 % (80,23)	100 % (56,27)	96,15 % (47,04)
	50				96,15 % (121,92)	96,15 % (67,96)	96,15 % (55,92)
	100					50 % (81,54)	46,15 % (56,08)

Tabelle 61 Erfolgsrate und Anzahl an Iterationen für die Semantik 77993 (ung. Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	7,14 % (1897)	51,85 % (292,79)	66,67 % (174,33)	77,78 % (104)	74,07 % (58,75)	70,37 % (50,16)
	5	25,93 % (1339,43)	85,19 % (278,52)	92,59 % (126,68)	88,89 % (75,13)	88,89 % (52,46)	88,89 % (44,63)
	10		92,59 % (243,68)	**92,59 % (106,4)**	88,89 % (60,25)	88,89 % (44,42)	88,89 % (39,67)
	25			81,48 % (147,82)	88,89 % (68,21)	88,89 % (46,58)	88,89 % (40,54)
	50				88,89 % (106,63)	88,89 % (59,96)	88,89 % (48,04)
	100					74,07 % (86,85)	74,07 % (60,50)

Tabelle 62 Erfolgsrate und Anzahl an Iterationen für die Semantik PIN (ung. Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	25,93 % (491,57)	40,74 % (235,36)	55,56 % (118,33)	62,96 % (83,59)	44,44 % (57,92)
	5	14,81 % (1712,75)	40,74 % (291,09)	74,07 % (210)	77,78 % (87,14)	66,67 % (62,28)	74,07 % (52,6)
	10		66,67 % (363,5)	81,48 % (143,55)	85,19 % (92,96)	88,89 % (54,17)	**88,89 % (45,83)**
	25			70,37 % (186,84)	88,89 % (79,63)	88,89 % (57,42)	88,89 % (50)
	50				85,19 % (137,83)	88,89 % (72,63)	85,19 % (56,22)
	100					74,07 % (86,85)	48,15 % (66,54)

Tabelle 63 Erfolgsrate und Anzahl an Iterationen für die Semantik Pseudonym (ung. Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	0	0	0	0	0
	5	0	0	0	0	0	0
	10		0	0	0	0	0
	25			0	0	0	0
	50				0	0	0
	100					0	0

Tabelle 64 Erfolgsrate und Anzahl an Iterationen für die Semantik Symbol (ung. Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	3,7 % (844)	3,7 % (293)	7,41 % (198)	0	3,7 % (65)
	5	0	7,41 % (399,5)	14,81 % (287)	29,63 % (157,13)	18,52 % (86,6)	22,22 % (67,83)
	10		18,52 % (712,4)	22,22 % (193,83)	33,33 % (125,56)	**37,04 % (77,1)**	25,93 % (69,71)
	25			18,52 % (277,2)	29,63 % (124,63)	33,33 % (84,89)	22,22 % (64)
	50				18,52 % (200,20)	22,22 % (97,33)	18,52 % (74,4)
	100					0	0

Tabelle 65 Erfolgsrate und Anzahl an Iterationen für die Semantik Woher(ung. Personen-IDs)

		N					
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)	300 (83)
M	3	0	0	7,41 % (299)	0	3,7 % (57)	0
	5	0	3,7 % (423)	3,7 % (438)	14,81 % (136)	18,52 % (90,4)	11,11 % (57,33)
	10		7,41 % 300,5)	18,52 % (277,2)	33,33 % (143,22)	25,93 % (81,86)	**33,33 % (70)**
	25			18,52 % (286,2)	25,93 % (156,14)	29,63 % (89,88)	18,52 % (64,6)
	50				18,52 % (204)	22,22 % (98,5)	11,11 % (74)
	100					0	3,7 % (77)