h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt

- Fachbereich Informatik -

Template Protection für biometrische Sprecherverifikation nach ISO/IEC 24745

Abschlussarbeit zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

vorgelegt von

Stefan Billeb

Referent:	Prof. Dr. Michael Braun
Korreferent:	Dr. Christian Rathgeb

Ausgabedatum:	21.11.2013
Abgabedatum:	21.02.2014

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, den 21. Februar 2014

Stefan Billeb

Abstract

Ausgefeilte Sicherheitskonzepte sind heutzutage gefragter denn je. In den letzten Jahren stieg die Anzahl gemeldeter Angriffe auf Benutzerauthentifizierungsdaten kontinuierlich an. Angreifer, die einen erfolgreichen und unbemerkten Angriff durchführen und dabei in Besitz von Passwörtern oder Smartcards gelangen, können sich unbemerkt mit diesen im System anmelden, da keine Prüfung der Identität vorgenommen werden kann.

Biometrische Systeme verwenden individuelle oder hinreichend unterscheidbare Merkmale, um die Identität der sich anmeldenden Person zu überprüfen. Die menschliche Stimme verfügt über solche Merkmale und bildet die Grundlage für Sprecherverifikationssysteme. Der ISO Standard ISO/IEC 24745 Biometric Information Protection beschreibt sowohl Anforderungen an biometrische Systeme zum Schutz der biometrischen Daten, als auch an das im System hinterlegte biometrische Template. Die wichtigsten Anforderungen sind hierbei die Erneuerbarkeit und Widerrufbarkeit, Unumkehrbarkeit und Unverkettbarkeit des Templates. Passwörter und Smartcard-Schlüssel können durch Einwegfunktionen unumkehrbar und durch das Ersetzen mit einem neuen Passwort oder Schlüssel erneuert werden. Da biometrische Daten einer natürlichen Varianz unterliegen, können diese nicht durch herkömmliche Einwegfunktionen, beispielsweise SHA2-256 oder andere Hashfunktionen, transformiert werden. Ein nachfolgender Vergleich wäre nicht eindeutig verifizierbar. Die biometrischen Daten müssen daher durch die eingesetzten Algorithmen in eine andere Datenrepräsentation transformiert werden, die zum einen eine Vergleichbarkeit und zum anderen den Schutz der sehr sensiblen Daten gewährleistet.

In dieser Arbeit wurde ein Template Protection System für biometrische Sprecherverifikation konstruiert, welches die Anforderungen des ISO Standards erfüllt. Es wurden die Anforderungen des ISO Standards bezüglich Sicherheit und Datenschutz erläutert und anschließend basierend auf einem Fuzzy Commitment Template Protection Algorithmus für Online-Signatur eine Anpassung und Implementierung für die Sprecherverifikation durchgeführt. Die erzielte biometrische Performanz wird durch Angabe der Equal-Error-Rate (EER) von 5.7% als praktikabel eingestuft. Für die hierbei erzielte Schlüssellänge von 14 Bit werden Optimierungsansätze untersucht und im Ausblick vorgeschlagen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Ausgefeilte Sicherheitskonzepte sind heutzutage gefragter denn je. In Zeiten der nahezu totalen Vernetzung werden Menschen täglich mit einer zunehmenden Anzahl von datenverarbeitenden Systemen konfrontiert. Viele dieser Systeme verarbeiten personenbezogene oder sensible Daten (z.B. Geldautomaten, Computerzugang). Um die auf den Systemen liegenden Daten vor unautorisiertem Zugriff zu schützen, gibt es drei gängige Verfahren zur Authentifizierung: wissensbasierte, besitzbasierte und biometrische Verfahren.

Bei wissens- und besitzbasierten Verfahren ist es möglich, eine Erneuerung und Widerrufung durchzuführen. So kann beispielsweise ein altes Passwort durch ein neues ersetzt oder eine Smartcard ausgetauscht werden. Dennoch kann mit wissens- und besitzbasierten Authentifizierungsverfahren keine Aussage über die tatsächliche Identität der angemeldeten Person getroffen werden. Ist ein Angreifer einmal in Besitz von Login-ID und Passwort, stehen ihm alle Möglichkeiten offen. Selbst wenn zusätzliche Sicherheitstokens (bspw. RSA-SecureID-Token[1]) eingesetzt werden, kann die angemeldete Person auch ein Angreifer sein. Erschwerend kommt hinzu, dass Wissen vergessen, Gegenstände verloren und beides ungewollt oder unautorisiert auch an Dritte weitergegeben werden können.

An diesem Punkt setzen biometrische Verfahren [bio12, S.49ff] an. Sie verwenden individuelle oder hinreichend unterscheidbare Merkmale der registrierten Person, um eine Aussage zu treffen, ob die sich anmeldende Person tatsächlich diejenige ist, die sie vorgibt zu sein. Die menschliche Stimme ist eine Charakteristik, deren Einzigartigkeit sich für den Einsatz in biometrischen Verfahren eignet. Der Bereich der Spracherkennung und Sprecherverifikationverifikation ([DPK08], [BBF+04]) wird schon seit langer Zeit erforscht. Die Möglichkeiten zur Sprecherverifikation sind heutzutage sehr ausgereift und liefern immer bessere Ergebnisse.

Durch die wachsenden Bedrohungen und die steigenden Anforderungen an die Sicherheit müssen auch biometrische Systeme zuverlässig gegenüber Angriffen auf das System, aber vor allem gegenüber Angriffen auf die sehr sensiblen biometrischen Daten, geschützt

[1] Ein spezielles Produkt der Firma RSA, `http://germany.emc.com/security/rsa-securid.htm`

werden. Darüber hinaus ist es wichtig, dass auch diese Systeme die Möglichkeit der Erneuerbarkeit und Widerrufbarkeit aufweisen, ohne die eingesetzte Charakteristik zu ändern. So müssen Sprecherverifikationssysteme ein erneutes Registrieren ermöglichen, wobei sich die Referenz im System ändert. Dieser Vorgang ist vergleichbar mit dem Ändern eines Passworts. An dieser Stelle setzt der ISO Standard ISO/IEC 24745 [ISO11] an. Er beschreibt Anforderungen an biometrische Systeme zum Schutz der biometrischen Daten sowie allgemeine Anforderungen an das System. Darüber hinaus stellt er Anforderungen an die im System hinterlegte biometrische Referenz im Hinblick auf Widerrufbarkeit einer Registrierung sowie deren Erneuerbarkeit. Die Ergebnisse dieser Arbeit sollen die Voxguard Software ISO/IEC 24745 standardkonform erweitern. Die atip GmbH betreibt mit der Software atip Voxguard[2] ein Sprecherverifikationssystem, welches ohne Template Protection ([JNN08], [RU11]) eine Verifikation durchführt. Hierfür werden Gaussian Mixture Model (GMM) und Universal Background Model (UBM) ([RQD00]) in den Algorithmen eingesetzt.

In dieser Arbeit werden die Anforderungen des ISO Standards 24745 an die biometrischen Daten und an das Gesamtsystem vorgestellt. Basierend auf einer ausgewählten Architekturempfehlung des Standards wurde das Konzept von Argones Rùa et. al [ARMACC12] zur Template Protection von Online-Signaturen auf die Verwendung von GMM und UBM in der Sprecherverifikation angepasst und implementiert. Das Verfahren binarisiert die biometrischen Merkmale und schützt das Template über ein Fuzzy Commitment Scheme [JW99]. Im Anschluss wurde eine Evaluation der erzielten biometrischen Performanz sowie die Erfüllung der Sicherheitsanforderungen durchgeführt. Das System konnte in der besten Konfiguration eine False Match Rate von 5.8% bei einer False Non-Match Rate von 5.6% erzielen.

Die Arbeit gliedert sich in fünf Teile: Zu Beginn werden die theoretischen Grundlagen zur Biometrie, biometrischen Systemen und den eingesetzten Verfahren, Hashfunktionen und fehlerkorrigierender Code erläutert. Anschließend werden die Anforderungen des Standards erläutert sowie die Anpassung des ausgewählten Algorithmus auf die Architektur vorgestellt. Hierauf folgt die Vorstellung der Implementierung und die detaillierte Erklärung der Verarbeitungsschritte des Systems. Den Abschluss bildet die Evaluation zur biometrischen Performanz und den Anforderungen an das System sowie eine Zusammenfassung der Ergebnisse mit einem Ausblick.

[2]Ein Produkt der atip GmbH, `http://www.atip.de/leistungen/produkte/`

2 Theoretische Grundlagen

Gesprochene Sprache ist eines der wichtigsten Kommunikationsmittel. Neben der Übertragung von Informationen sind auch ihre individuellen Merkmale und Ausprägungen für Menschen erkenn- und unterscheidbar. Dieser biometrische Aspekt ist für die Mensch-Maschine-Kommunikation schon seit längerer Zeit Gegenstand von Forschungen verschiedener Richtungen. Neben der Spracherkennung ist die Sprecherverifikation zur Anmeldung an Systemen ein sich stärker entwickelnder Bereich.

Um ein Sprachsignal informationstechnisch zu verarbeiten, sind mehrere Schritte notwendig. Nach der Sprachsignalcodierung, der Analog-Digital-Wandlung (A/D-Wandlung), werden Informationen aus dem Audiosignal gewonnen. Die gewonnenen Informationen werden anschließend mit einem Verfahren verarbeitet, das individuelle und vergleichbare Merkmale des Sprechers herausarbeitet. Dieses Kapitel erläutert die theoretischen Grundlagen der Stimmbildung und gibt eine Einführung in das Forschungsgebiet der Biometrie. Es werden die Bestandteile eines biometrischen Systems und die Beurteilung der biometrischen Performanz eines solchen Systems dargestellt. Den Abschluss bilden Erläuterungen zu den in dieser Arbeit eingesetzten Techniken, fehlerkorrigierender Code und Hashfunktion.

2.1 Stimmbildung

Die Stimme ist eine Summe von Schallwellen, die von einem Menschen durch eine Folge artikulatorischer Bewegungen durch seinen Sprachapparat (vgl. Abb. 2.1) erzeugt werden. Die Schallwellen werden beim Hörer durch das Ohr wahrgenommen und im Anschluss vom Gehirn verarbeitet. Eine wichtige Aufgabe beim Artikulationsprozess nimmt die Stimmritze (Glottis) ein. Sie besteht aus den Stimmlippen und den Stellknorpeln. Grundsätzlich unterscheidet man beim Sprechprozess zwischen Schallproduktion und Klangformung [DPK08, S.12].

Während der Schallproduktion wird aus der Lunge Luft ausgestoßen. Hierbei werden die aneinander liegenden Stimmlippen in Schwingung versetzt. Das Ergebnis ist ein pe-

riodisch unterbrochener Luftstrom, ein akustisches Signal. Schlagen dabei die Stimmlippen periodisch aneinander, ist das entstehende Signal stark oberwellenhaltig. Liegen die Stimmlippen beim Luftausstoß nicht aneinander, entweicht die Luft gleichmäßig durch Mund und Nase. Behindert man allerdings den Luftstrom durch Engstellen (z.B. das Anstellen der Zunge), entstehen Turbulenzen, die als zischendes Geräusch wahrgenommen werden [DPK08, S.12f].

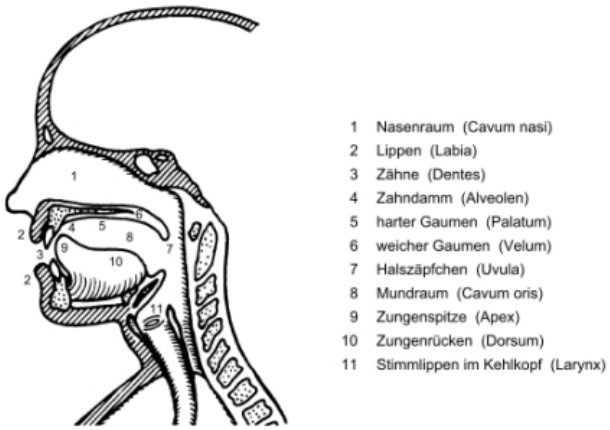

Abbildung 2.1: Mittelschnitt durch den menschlichen Sprechapparat, Quelle: [DPK08]

Die genannten akustischen Signale sind großteils sprecherunabhängig. Sie werden jedoch durch den Vokaltrakt (hierzu zählen Rachen-, Mund- und Nasenraum) klanglich verändert. Der Vokaltrakt bildet dadurch einen akustischen Filter, der sich bei jedem Sprecher durch seine individuell ausgeprägten Organe verändert. Entscheidenden Einfluss haben hierbei die Bewegung der Artikulatoren (Zunge, Lippen, Kiefer und Gaumensegel). Durch die Bewegung dieser Artikulatoren verändern sich die Übertragungsfunktion und die Resonanzfrequenzen. Es entstehen verschiedene Klänge, die man als Laute bezeichnet [DPK08, S.13].

Das komplexe Zusammenspiel verursacht jedoch auch Varianzen, die in den Schallsignalen erkenn- und messbar sind. Abbildung 2.2 zeigt zwei zeitlich unabhängige Aufnahmen der Aussprache des Wortes „mein“ desselben Sprechers. Dargestellt ist hierbei der zeitliche Verlauf der Aussprache (horizontale Ausbreitung) und die zum Zeitpunkt t gemessene Amplitude (vertikale Ausbreitung).

Zu erkennen ist die auftretende Varianz des Sprachsignals sowohl im horizontalen als auch vertikalen Verlauf der Aussprache. Durch das erlernte Sprechverhalten und die indi-

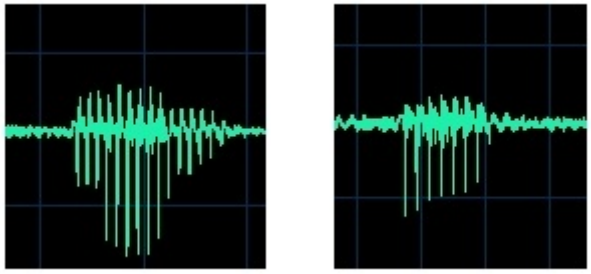

Abbildung 2.2: Zwei Aussprachen des Wortes 'mein' desselben Sprechers

viduell physiologischen Ausbildungen des Sprechapparates besitzt jeder Mensch eine für uns Menschen unterscheidbare und erkennbare Stimme.

2.2 Biometrie

Die Biometrie[3] untersucht die Messungen an Lebewesen sowie die entsprechenden Mess- und Auswerteverfahren. Die Charakteristiken können dabei in zwei Kategorien unterteilt werden (vgl. Abb. 2.3):, die physiologischen und die verhaltensabhängigen Charakteristiken [Bun, S.1].

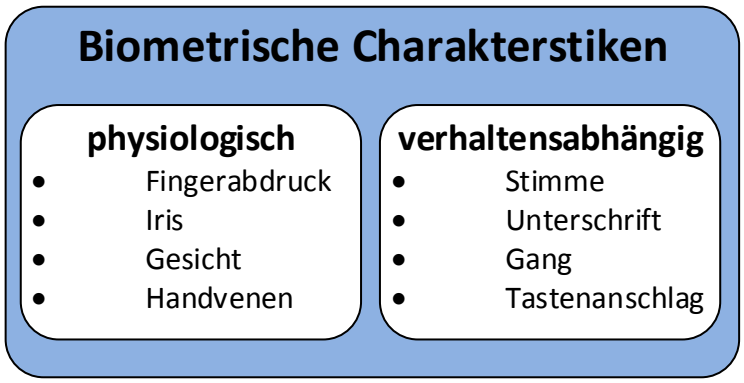

Abbildung 2.3: Übersicht und Einteilung einer Auswahl verschiedener biometrischer Charakteristiken

[3]Begriffliche Herleitung aus dem altgriechischen *bíos* – *Leben* und *métron* – *Maßstab*.

2.2.1 Eigenschaften eines biometrischen Charakteristikums

Damit sich ein biometrisches Charakteristikum zur Unterscheidung von Individuen eignet, muss es bestimmte Eigenschaften besitzen. Zur Bewertung dieser Eignung werden sieben Faktoren herangezogen (vgl. [Eck09, S.470], [Bun, S.4]) :

- **Universalität:** Es muss bei möglichst vielen Personen vorkommen, sodass sich nahezu jeder im System registrieren kann.
- **Einzigartigkeit:** Es muss bei verschiedenen Personen hinreichend unterschiedlich sein.
- **Performanz:** Benötigte Ressourcen und Erkennungsleistung sollten der Anwendung angemessen sein.
- **Messbarkeit:** Es muss mess- und digitalisierbar sein. Die gemessenen Werte müssen reproduzierbar sein.
- **Beständigkeit:** Es sollte so lange wie möglich stabil sein und sich nicht oder nur kaum durch Alterungsprozesse verändern.
- **Akzeptanz:** Das System muss vom Nutzer akzeptiert und gleichzeitig in einem akzeptablen Kostenrahmen betreibbar sein.
- **Überlistung:** Es sollte gar nicht oder nur sehr schwer möglich sein, das Charakteristikum zu fälschen und unerlaubten Zutritt zum System zu erlangen.

Abbildung 2.4 zeigt die Beurteilung verschiedener Charakteristiken anhand der genannten sieben Faktoren.

2.2.2 Stimme als biometrisches Charakteristikum

Sprecherverifikationssysteme verwenden das biometrische Charakteristikum „Stimme“, welches den verhaltensabhängigen Charakteristiken zugeordnet wird (vgl. [YG08]). Darüber hinaus wird die Stimme aber auch durch die Anatomie des Individuums beeinflusst.

Wie in Abbildung 2.4 zu sehen, existiert eine grundlegende Eignung der Stimme als biometrisches Charakteristikum. Hierbei sticht vor allem die hohe Nutzerakzeptanz hervor. Zusätzlich zu erwähnen ist die besonders kostengünstige und sehr weit verbreitete Sensortechnik. Mikrofone und Soundkarten sind heutzutage in nahezu jedem Computer

Merkmal	*Universalität*	*Einzigartigkeit*	*Beständigkeit*	*Messbarkeit*	*Leistung*	*Akzeptanz*	*Resistenz**
Fingerbild	mittel	**hoch**	**hoch**	mittel	**hoch**	mittel	**hoch**
Handgeometrie	mittel	mittel	mittel	**hoch**	mittel	mittel	mittel
Iris	**hoch**	**hoch**	**hoch**	mittel	**hoch**	*gering*	**hoch**
Retina	**hoch**	**hoch**	mittel	*gering*	**hoch**	*gering*	**hoch**
Gesicht	**hoch**	*gering*	mittel	**hoch**	*gering*	**hoch**	*gering*
+Thermogramm	**hoch**	**hoch**	*gering*	**hoch**	mittel	**hoch**	**hoch**
Unterschrift	gering	*gering*	*gering*	**hoch**	*gering*	**hoch**	*gering*
Stimme	mittel	*gering*	*gering*	mittel	*gering*	**hoch**	*gering*
Handvenen	mittel	mittel	mittel	mittel	mittel	mittel	**hoch**
Tastenanschlag	*gering*	*gering*	*gering*	mittel	*gering*	mittel	mittel

Abbildung 2.4: Bewertung biometrischer Verfahren, Quelle: [PS02]

vorhanden. Telefone und Smartphones sind darüber hinaus seit langer Zeit als Kommunikationsmittel in der Gesellschaft akzeptiert und mit der notwendigen Hardware zur Aufnahme ausgestattet.

Auch Siegert beschreibt in einer neueren Beurteilung die Stimme als Charakterstikum mit Einzigartigkeit, Universalität, hoher Akzeptanz und einfacher Messbarkeit (vgl. [Sie09, S.12]). Die Beständigkeit wird als problematisch angegeben, da sich die Stimme durch Alterungsprozesse, aber z.B. auch durch starken Tabak- und Alkoholkonsum verändern kann. Des Weiteren wird die erreichbare Performanz als praktikabel eingeschätzt.

Bei biometrischen Systemen unterscheidet man in der Anwendung weiterhin die Verifikation und die Identifikation. Während bei der Identifikation eine 1:n-Entscheidung durchgeführt wird, ist bei der Verifikation eine 1:1-Prüfung durchzuführen.

Die Verarbeitung der gemessenen biometrischen Daten wird in einem biometrischen System durchgeführt, welches im folgenden thematisiert wird.

2.3 Biometrisches System

In einem biometrischen System werden Signale eines biometrischen Charakteristikums in ein biometrisches Sample umgewandelt. Die Systeme führen eine automatische Erkennung von Individuen basierend auf Merkmalen physiologischer und/oder verhaltensabhängiger Charakteristiken durch, welche bestimmte Eigenschaften besitzen müssen, um hierfür geeignet zu sein ([Kna09, S.26f]).

2.3.1 Prozesse eines biometrischen Systems

Ein biometrisches System besteht üblicherweise aus fünf Teilsystemen (vgl. [ISO12], [ISO06, S.8]), die nicht zwangsläufig auf ein und derselben Maschine ausgeführt werden müssen. Abbildung 2.5 zeigt das Zusammenwirken der Teilsysteme für den Identifikations- bzw. Verifikationsvorgang.

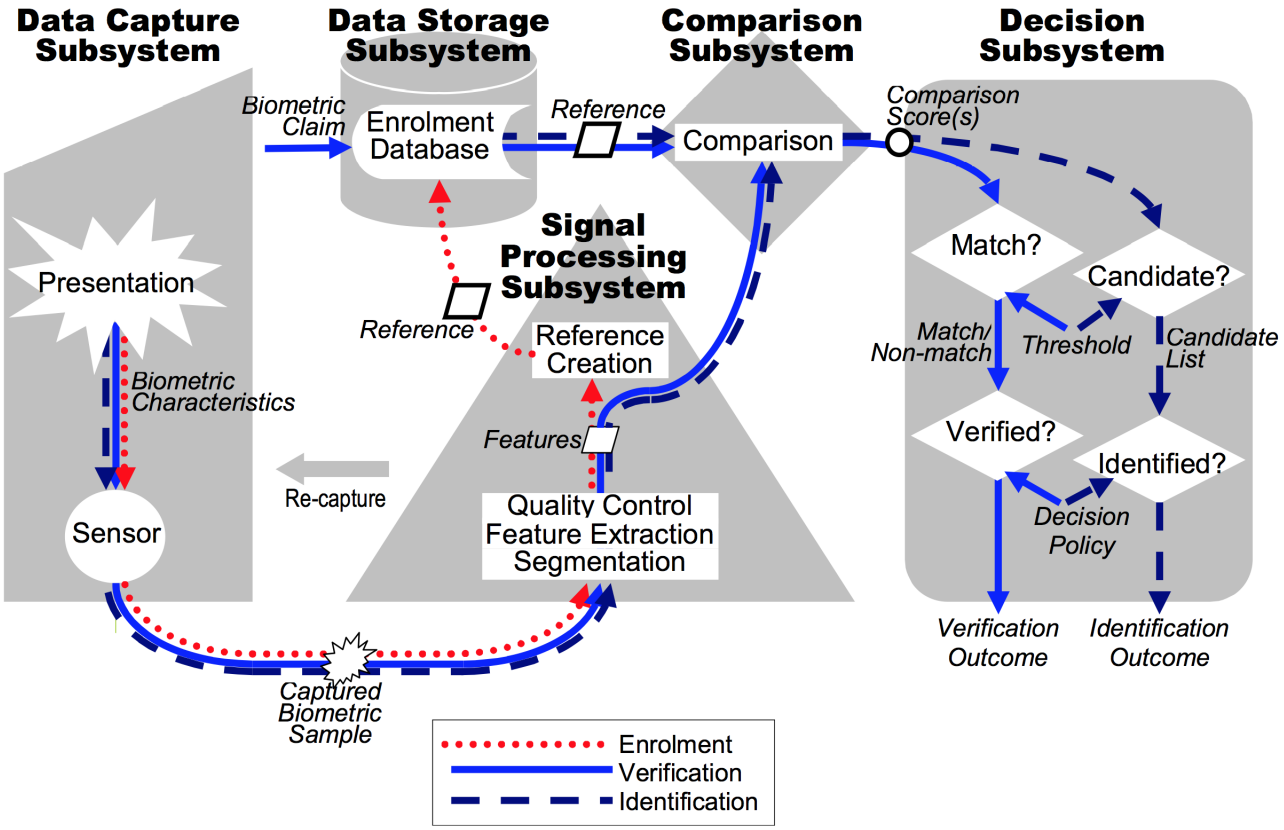

Abbildung 2.5: Architektur eines biometrischen Systems, Quelle: ISO/IEC SC37 [ISO12]

- Im ***Data Capture Subsystem*** misst ein Sensor die Signale. Im Kontext dieser Arbeit ist dies die Aufnahme eines Sprachsignals.
- Im ***Signal Processing Subystem*** werden aus dem Signal Merkmale (Features) extrahiert.
- Im ***Data Storage Subsystem*** ist die Referenz eines Users hinterlegt.

- Im ***Comparison Subsystem*** wird das präsentierte Sample mit der Referenz verglichen.
- Im ***Decision Subsystem*** wird anhand des Ergebnisses aus dem Comparison Subsystem eine Entscheidung über die Authentifizierung durchgeführt.

Damit die Teilsysteme diese Schritte durchführen können, muss das System folgende Funktionen bereitstellen:

- **Enrollment**: Das Erstellen und Abspeichern einer Referenz für einen sich registrierenden Benutzer.
- **Verifikation**: Das präsentierte Sample (die biometrische Probe) wird mit der vorhandenen Referenz verglichen.

2.3.2 Biometrische Referenz und Template

Eine biometrische Referenz besteht aus einem oder mehreren persistenten biometrischen Samples, biometrischen Templates oder biometrischen Modellen. Sie werden einer Person zugeordnet, sodass sie zu einem biometrischen Vergleich herangezogen werden können (vgl. [ISO11, S.10], [Bus12]). Ein biometrisches Template ist eine Menge oder ein Vektor gespeicherter biometrischer Merkmale. Mit ihnen muss, ggf. nach einer Verarbeitung, ein Vergleich mit biometrischen Merkmalen einer biometrischen Probe möglich sein.

2.3.3 Merkmalsextraktion

Die Merkmalsextraktion hat das Ziel, aus den gemessenen biometrischen Daten Merkmale, sogenannte Features, zu extrahieren.

Bevor ein Sprachsignal in einem Computer verarbeitet werden kann, muss das Sprachsignal digitalisiert werden. Hierzu wird eine Abtastfrequenz (Samplingrate) und Codierung festgelegt. Für Telefonapplikationen wird meist eine Samplingrate von 8 kHz und Pulse-Code-Modulation (PCM) ([Wag95]) verwendet. Zu beachten ist hierbei, dass nach dem Nyquist-Shannon-Abtasttheorem ([DPK08, S.40]) die Abtastfrequenz f_{abtast} mindestens doppelt so groß wie die maximal auftretende Frequenz f_{max} (bzw. die maximal zu messende Frequenz) im Signal sein muss:

$$f_{abtast} > 2 * f_{max} \tag{2.1}$$

Nach der Digitalisierung besteht das Sprachsignal aus einer Menge von quantisierten Amplitudenwerten für jeden abgetasteten Zeitpunkt. Aus dieser Zeitdarstellung können noch keine aussagekräftigen Merkmale extrahiert werden. Das Sprachsignal wird deshalb anschließend mittels Fast-Fourier-Transformation (FFT) in den Frequenzbereich überführt. Die FFT liefert für periodische Signale eine korrekte Transformation in den auftretenden Frequenzbereich [DPK08, Kapitel 4].

Durch die Nichtperiodizität von Sprachsignalen wird die FFT so berechnet, dass das Sprachsignal mittels einer Fensterfunktion in mehrere kurze Abschnitte aufgeteilt wird. Innerhalb dieser Fenster kann das Sprachsignal als periodisch angesehen werden, wodurch eine genauere Analyse möglich ist ([ST95, S.48ff]). Als adäquate Fenstergrößen werden in der Literatur Fenstergrößen von 5 - 30 ms angegeben [Sie09, S. 27].

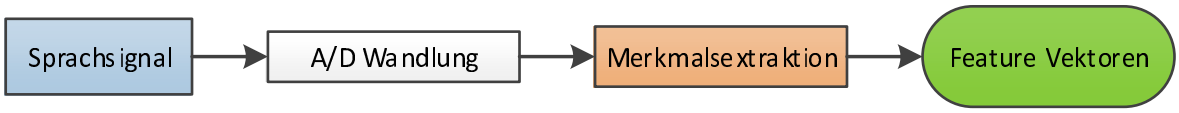

Abbildung 2.6: Verarbeitungsschritte des Sprachsignals zu Feature-Vektoren

Aus dem Frequenzbereich können nun Features extrahiert werden, die eine Repräsentation der Sprechercharakteristik ermöglichen. In der Praxis hat sich die Verwendung von Mel-Frequency Cepstral Coefficients[4] für Sprecherverifikation bewährt (vgl. [KL10]). Bevor die Features extrahiert werden, werden aus dem Sprachsignal Anteile mit Stille (Teile des Sprachsignals in denen keine Laute enthalten sind) entfernt, um nur die Bereiche mit Informationen zu erhalten.

Anschließend erhält man für jedes Fenster einen Feature-Vektor, der die extrahierten Features für diesen Zeitabschnitt beinhaltet. In dieser Arbeit werden für ein Fenster zwölf Mel-Koeffizienten, der Energiewert, sowie die Delta-Koeffizienten (erste Ableitung) und Delta-Delta-Koeffizienten (zweite Ableitung) extrahiert. Somit erhält man einen Feature-Vektor mit $(12+1)*3 = 39$ Koeffizienten. Die Gesamtzahl gewonnener Feature-Vektoren ist abhängig von der zeitlichen Länge des Sprachsignals. Die Verknüpfung der beschriebenen Phasen ist schematisch in Abbildung 2.6 abgebildet und zeigt die Verarbeitungsschritte vom Sprachsignal bis zu den gewonnenen Feature-Vektoren.

[4]Weiterführende Literatur: [ST95, S.60ff], [DPK08, S. 296].

2.4 Sprecherverifikation

Bei der Sprecherverifikation wird mit dem Charakterstikum Stimme eine Prüfung eines Sprechers gegen eine abgespeicherte Referenz durchgeführt. Hierzu muss zunächst für einen Benutzer das Enrollment durchgeführt werden, um anschließend gegen eine Referenz eine Verifikation zu ermöglichen.

Eine allgemeine Veranschaulichung einer Sprecherverifikation ist in Abb. 2.7 zu sehen. Nachdem ein Benutzer den Prozess des Enrollments durchlaufen hat, kann seine Referenz gegen eine abgegebene Probe verifiziert werden. Klassische Sprecherverifikationssysteme berechnen einen Score, welcher eine Aussage darüber trifft wie ähnlich das präsentierte Sample zur Referenz ist. Die Verifikation kann dann anhand eines festgesetzten Schwellwertes den Verifikationsversuch als akzeptiert oder abgewiesen klassifizieren.

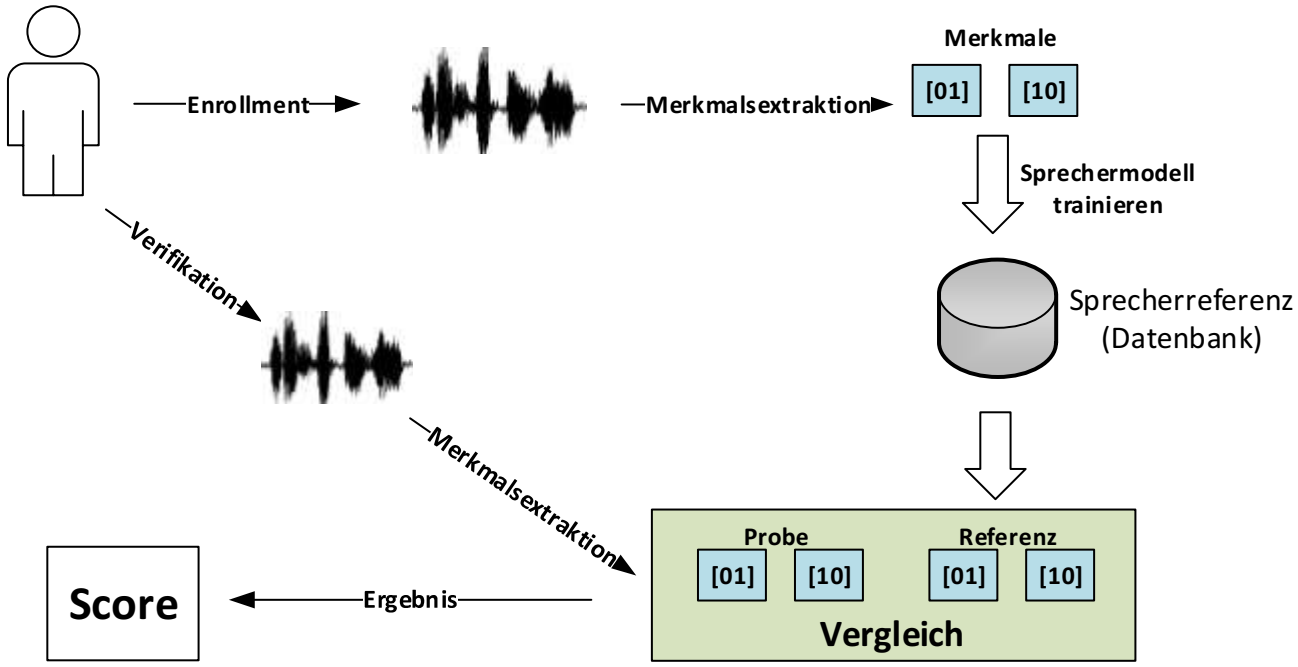

Abbildung 2.7: Allgemeiner Ablauf einer klassischen Sprecherverifikation

Eine Sprecherverifikation kann sowohl textabhängig als auch textunabhängig durchgeführt werden. Bei der textabhängigen Verifikation wird für das Enrollment und die Probe derselbe Text gesprochen. Bei einer textunabhängigen Verifikation sind die Texte des Enrollments unabhängig von der Verifikation. Die textunabhängige Verifikation kann daher zum Beispiel auch während eines laufenden Telefongespräches (vgl. [Kun11]) durchgeführt werden oder das System gibt zur Absicherung gegenüber replay-Attacken[5] zufällige Texte vor.

2.5 Biometrische Performanz

Der letzte Schritt eines jeden biometrischen Systems ist die Entscheidungsfindung. Dabei muss ein Verifikationssystem entscheiden, ob das präsentierte Sample als akzeptiert – eine gültige Verifikation – oder abgewiesen – ein Einbruchsversuch – klassifiziert wird. Für einen Verifikationsversuch eines Sprechers bei seiner eigenen Referenz wird dieser als Genuine, bei einer nicht-zugehörigen Referenz als als Impostor bezeichnet.

Für Systeme, deren Vergleichsergebnis ein Score S ist, muss ein Schwellwertparameter (Threshold) t gewählt werden, anhand dessen der Score klassifiziert wird. Diesen Schwellwert zu kalibrieren, führt in der Praxis nicht immer zu optimalen Ergebnissen. Stattdessen ist er eine Kalibrierung der beiden Fehlerraten False Match Rate (FMR) und False Non-Match Rate (FNMR) ([ISO06], [Eck09, S.475]). Die FMR beschreibt die Rate akzeptierter Impostor während die FNMR die Rate abgewiesener Genuines angibt.

In Abbildung 2.8 ist ein Histogramm abgebildet, welches das Auftreten aller Genuine-Scores und Impostor-Scores darstellt. Wählt man einen spezifischen Threshold t, so ergeben sich die Fehlerraten FMR und FNMR aus den Flächen unter den Verteilungsdichtefunktionen. Die FMR entspricht dabei der Fläche unter der Impostor-Verteilung (pdf_i) rechts des Schwellwertes, da alle Scores $S \geq t$ akzeptiert werden. Die FNMR entspricht der Fläche unterhalb der Genuine-Verteilung (pdf_g). Wird der Schwellwert so gewählt, dass FMR und FNMR gleich sind, spricht man von einer Gleichfehlerrate, der Equal Error Rate (EER).

[5] Eine replay-Attacke ist das Wiedereinspielen aufgezeichneter Daten. Sie dient oft zum Vortäuschen einer fremden Identität. Weiterführende Literatur [BWS06, S.95], [Eck09, S.120].

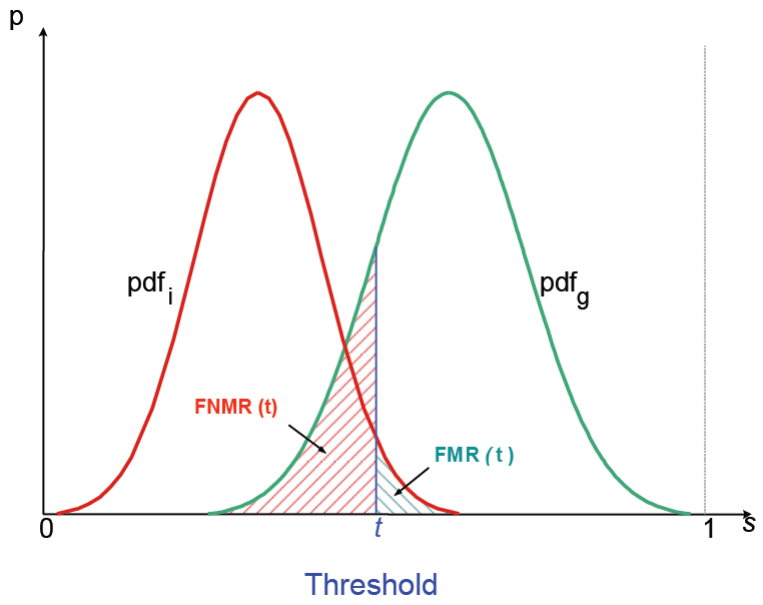

Abbildung 2.8: Beispielhafte Verteilungsdichtefunktionen für Genuine- und Impostor-Scores mit False-Match-Rate und False-Non-Match-Rate, Quelle: [Bus11]

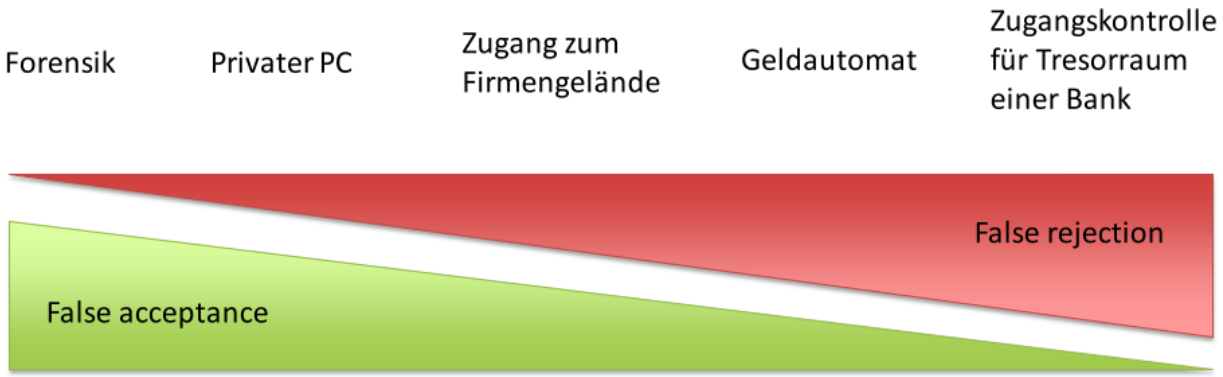

Abbildung 2.9: Subjektive Anforderungen verschiedener Einsatzbereiche an biometrische Systeme, Quelle: [PR13]

Die Einstellung des Schwellwertes führt auch zu einer Einstellung im Verhältnis Sicherheit zu Nutzerakzeptanz. Wählt man den Schwellwert so, dass das System eine höhere Nutzerakzeptanz besitzt (eine kleinere FNMR), führt dies gleichzeitig zu einer geringeren Sicherheit, da die FMR steigt. Eine niedrigere FMR kann jedoch auch zu einer hohen

Falschabweisung mit hoher FNMR führen. Abbildung 2.9 zeigt eine Einteilung verschiedener Anforderungen an das Verhalten eines biometrischen Systems.

Beispielsweise wird für forensische Analysen die Anzahl falscher Rückweisungen als besonders gering eingestuft (geringe FNMR). Durch das allgemeine Nichterreichen einer EER von 0% wird jedoch auch eine höhere Falschakzeptanz akzeptiert. Im Zweifelsfall sollen in diesem Kontext mehr Kandidaten zur Identifikation herangezogen werden. Im Gegensatz hierzu steht der Zugriff auf einen Banktresor, bei dem man eine FMR von 0% erhalten möchte, wodurch sich jedoch eine höhere Falschrückweisung ergibt. Insgesamt ist es also wichtig, die Trennung von Genuine und Impostor im System so gut wie möglich zu gestalten, um beide Fehlerraten möglichst gering zu halten.

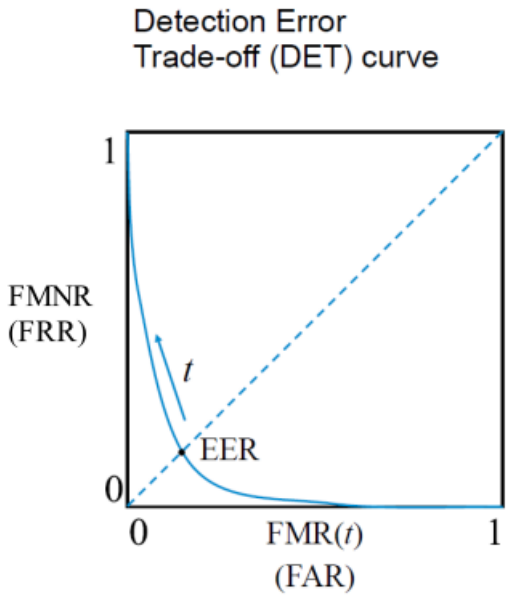

Abbildung 2.10: Exemplarisches DET-Diagramm, Quelle: [PR13]

Die in einem System erreichten Fehlerraten können in ein Detection-Error-Tradeoff (DET)-Diagramm ([ISO06, S.35], [MDK+97]) eingetragen werden. Hierbei wird die FMR gegen die FNMR eingetragen. Abbildung 2.10 zeigt ein exemplarisches DET-Diagramm. Auf der x-Achse wird hierbei der Bereich der FMR, auf der y-Achse der Bereich FNMR aufgetragen. Durch den Schwellwert t zeigt sich beim Auftragen der zugehörigen Fehlerraten eine Abhängigkeit zwischen FNMR und FMR. Gleichzeitig lässt sich in dieser Darstellung die EER, welche die Winkelhalbierende der beiden Achsen bildet, direkt ablesen. Sie entspricht dem Schnittpunkt zwischen der Winkelhalbierenden und dem eingetragenen

Graphen. Je näher sich dieser Graph am Koordinatenursprung befindet, desto biometrisch performanter ist das System.

Die Fehlerraten FMR und FNMR bestimmen die algorithmischen Fehlerraten. Ihnen gegenüber stehen die False Acceptance Rate (FAR) und False Rejection Rate (FRR), die die Fehlerraten unter Einbeziehung von Systemfehlern[6] berechnen. Da in dieser Arbeit der Schwerpunkt auf der Sicherheit der Templates und die dabei erreichbare biometrische Performanz gelegt wird, werden die FAR und FRR nicht weiter betrachtet.

2.6 Hashfunktion

Hashfunktionen sind vielseitig einsetzbare Funktionen, die in verschiedenen Teilgebieten der Informatik Anwendung finden. Neben der Anwendung in der Kryptographie (z.B. dem Message Authentication Code (MAC)-Verfahren [Eck09, S.367]) werden sie beispielsweise auch als Suchverfahren in Datenbanken eingesetzt ([Eck09, S.356]). Eine Hashfunktion H ist eine Einwegfunktion[7], die eine beliebig lange Eingabe m auf eine Ausgabe h (den Hashwert) fester Länge abbildet:

$$H(m) = h \tag{2.2}$$

Hashfunktionen werden in schwache und starke Hashfunktionen (vgl. [Eck09, S.357f]) eingeteilt. Eine schwache Hashfunktion besitzt die Eigenschaften:

1. H ist eine Einwegfunktion
2. Der Hashwert h ist für eine Eingabe m leicht zu berechnen
3. Für ein gegebenes h für die Nachricht m ist es praktisch unmöglich, eine Nachricht $m^* \neq m$ zu finden, sodass sie denselben Hashwert bilden: $H(m) = H(m^*)$

Für den Fall, dass $m^* \neq m$ und $H(m) = H(m^*)$ ist, spricht man von einer Kollision. Eigenschaft (3) beschreibt, dass zu einer vorgegebenen Nachricht m jedoch kein effizientes Verfahren existiert, um ein m^* zu konstruieren, sodass eine Kollision auftritt. Da die Eingabelänge wesentlich größer als die Ausgabelänge sein kann, können grundsätzlich zwei unterschiedliche Nachrichten $m \neq m^*$ existieren, sodass eine Kollision auftritt.

[6] Systemfehler sind zum Beispiel Failure to Capture. Mögliche Fehler werden unter [ISO06] beschrieben.
[7] Definition einer Einwegfunktion unter [Eck09, S.329 Definition 7.8].

Dies führt zur Definition einer starken Hashfunktion. Eine starke Hashfunktion ist eine schwache Hashfunktion mit der zusätzlichen Eigenschaft, dass es praktisch nicht möglich ist, zwei Eingabewerte $m \neq m^*$ zu finden, sodass beide Eingabewerte denselben Hashwert bilden.

Hieraus folgt, dass bei einer starken Hashfunktion

$$H(m) = H(m^*) \ , \ nur\ für\ m = m^* \tag{2.3}$$

derselbe Hashwert gebildet wird. Eine Empfehlung für starke Hashfunktionen wird von Eckert unter [Eck09, S. 359, S.364] sowie der Europäischen Agentur für Netz- und Informationssicherheit unter [Inf13] gegeben.

2.7 Fehlerkorrigierender Code

Fehlerkorrigierende Codes sind Algorithmen der Codierungstheorie, deren Ziel die Erhöhung der Sicherheit von übertragenen Nachrichten in störungsanfälligen Kommunikationssystemen ist. Sie ermöglichen, dass Fehler in der Übertragung erkannt und bis zu einer gewissen Grenze korrigiert werden können ([BFK+98, S.1]). Sie bilden hierfür eine Nachricht m der Länge k in einen Vektor c der Länge n ab. Dieser Vorgang, bei der den Nachrichten Redundanzen hinzugefügt werden, nennt sich Codierung.

Ein solcher Code C wird auch linearer (n, k)-Code genannt, wobei k die Dimension und n die Länge des Codes genannt wird und $n > k$ ist. Die bei der Abbildung entstehenden Vektoren nennt man Codewörter.

Um die in einem empfangenen Codewort enthaltene Nachricht zu erhalten, muss das Codewort vom Empfänger decodiert werden. Bei der Decodierung unterscheidet man zwischen der Fehlererkennung und der Fehlerkorrektur.

Die Leistungsfähigkeit zur Fehlerkorrektur wird durch die Minimaldistanz d der Codewörter des Codes bestimmt. Die Minimaldistanz beschreibt den kleinsten Abstand, der zwischen zwei Codewörtern c, c' eines Codes C existiert. Ein solcher Code wird als (n, k, d)-Code bezeichnet. Die Minimaldistanz kann über die Hamming-Distanz[8] HD zwischen allen Codewörtern bestimmt werden:

$$d = min\left\{HD(c, c')|c, c' \in C, c \neq c'\right\} \tag{2.4}$$

[8]Die Hamming-Distanz gibt an, an wie vielen Stellen sich zwei binäre Vektoren unterscheiden ([BFK+98, S.3]).

Treten bei der Übertragung eines Codeworts c^* mindestens ein Fehler, aber höchstens $d-1$ Fehler auf, ist der empfangene Vektor kein Codewort. Es können daher maximal $d-1$ Fehler erkannt werden, da bei genau d Fehlern das übertragene Wort wieder ein Codewort des Codes ist. Für $d \geq 2t+1$ können bis zu t aufgetretene Fehler garantiert korrigiert werden, da die Zuordnung eindeutig ist. Darüber hinaus können im Allgemeinen auch Fehler korrigiert werden, jedoch existiert keine Garantie, dass das zurückgegebene Codewort auch das ursprünglich gesendete Codewort ist. Die Distanz kann zu zwei oder mehreren Codewörtern gleich groß sein, wodurch sich keine Eindeutigkeit ergibt.

Das in dieser Arbeit eingesetzte Decodierungsverfahren ist die Maximum-Likelihood-Decodierung ([BFK+98, S.7]). Sie berechnet zwischen dem empfangen Wort c^* und allen Codewörtern $c \in C$ die Hamming-Distanz und wählt das Codewort mit geringster Distanz aus. Neben der Maximum-Likelihood-Decodierung gibt es weiterhin die Maximum-a-posteriori-Decodierung ([Dam10, S.9]) und Syndrom-Decodierung ([BFK+98, S.10]). Diese Decodierungsmethoden werden jedoch in dieser Arbeit nicht implementiert.

3 Architektur eines Template Protection Systems nach ISO/IEC 24745

Für das Betreiben eines biometrischen Systems müssen sich die Betreiber bewusst sein, dass der Schutz biometrischer Daten sowie dessen Sicherheitsmechanismen nicht nur für die Einhaltung gesetzlicher Bestimmungen notwendig ist, sondern dass die Systeme einen Beitrag für die Akzeptanz in der Gesellschaft bilden.[9] Der ISO/IEC 24745 Standard [ISO11] beschreibt die Anforderungen hinsichtlich der Sicherheit und des Datenschutzes durch Schutzziele, die zu erfüllen sind, sowie Architekturmodelle und Bedrohungsszenarien und deren Gegenmaßnahmen. Die Anforderungen werden nachfolgend beschrieben und beziehen sich auf den Standard.

3.1 Sicherheitsanforderungen

Die Sicherheitsanforderungen des Standards sind eng an die Schutzziele kryptographischer Verfahren angelehnt (vgl. [BWS06, Kapitel 1], [Eck09, S.6ff]). Im Nachfolgenden werden die Schutzziele und eine Empfehlung zur Erreichung selbiger aus dem ISO-Standard erläutert.

3.1.1 Vertraulichkeit

Vertraulichkeit ist die Eigenschaft, dass Informationen gegen unautorisierten Zugriff oder Offenlegung geschützt sind (vgl. [ISO11, S.12]).

Die in biometrischen Systemen gespeicherte Referenz (bzw. das Template) muss auch zum Zeitpunkt der Nutzung, das heißt in der Verarbeitung eines Verifikationsprozesses, jederzeit durch Mechanismen vor unerlaubtem Zugriff geschützt sein. Dies kann durch den

[9] Eine Studie zur Akzeptanz biometrischer Systeme findet sich unter [KRB13].

Einsatz von Verschlüsselungsmechanismen – z.B. Advanced Encryption Standard (AES) ([DR02], [Bun13, S.21]) oder Rivest, Shamir und Adleman Verfahren (RSA) ([Bun13, S.34], [BWS06, S.17]) – und Zugriffskontrollen erreicht werden.

3.1.2 Integrität

Integrität stellt die Vollständigkeit und Fehlerfreiheit von Daten und deren Ursprung sicher (vgl. [ISO11, S.12]). Diese Eigenschaft ist für biometrische Daten und Templates fundamental. Ist die biometrische Referenz oder das präsentierte Sample nicht vertrauenswürdig und gesichert, kann das Ergebnis der Verifikation auch nicht vertrauenswürdig sein. Fehlerhafte Daten können auf Hard-/Softwarefehler, aber auch durch Angriffe auf das System, entstehen. Die Daten dürfen nicht unautorisiert und unbemerkt manipuliert werden. Durch den Einsatz kryptographischer Verfahren wie MAC ([Bun13, S.38], [Eck09, S.367]) oder digitale Signatur ([Eck09, S.371]) kann das Schutzziel Integrität erreicht werden. Die Daten können auch mittels Timestamping ([Eck09, S.416]) gegen das Wiedereinbringen gestohlener Daten und zur Verhinderung von replay-Attacken gesichert werden.

3.1.3 Erneuerbarkeit und Widerrufbarkeit

Während klassische Authentifizierungsarten wie Passwörter und Smartcards problemlos widerrufen und erneuert werden können, ist dies bei biometrischen Systemen nicht jederzeit gegeben. So haben Fingerabdruck-Systeme – bedingt durch die Anatomie des Menschen – spätestens bei der elften Erneuerung keine Möglichkeit mehr erneuert zu werden. Beim Einsatz von Stimme als Charakterstikum ist sogar gar keine Erneuerbarkeit gegeben.

Kompromittiert ein Angreifer die Referenz, muss diese widerrufbar sein. Damit der Benutzer das System weiterhin sicher nutzen kann, muss die Referenz erneuerbar sein (ähnlich dem Ändern eines Passworts). Dabei ist zu beachten, dass die Erneuerung nicht mit einer anderen Charakteristik durchgeführt werden muss. Das heißt, für die Erneuerung der Referenz muss wieder Stimme als Charakteristikum verwendet werden können.

Um den Sicherheitsanforderungen heutzutage gerecht zu werden, ist es unabdingbar, das System so zu konzipieren, dass es den Benutzern die Möglichkeit des Widerrufs und Erneuerns ihres Templates zur Verfügung stellt.

3.2 Datenschutzanforderungen

Die Nutzung eines Systems ist oft an Rollen gekoppelt. An diese werden Rechte gebunden (z.B. um auf eine bestimmte Datei zuzugreifen), die entweder durch Gruppen oder direkte Bindung an ein Benutzerprofil verknüpft werden. Die Benutzerprofile beinhalten darüber hinaus private und teils sensible Informationen. Durch die Verknüpfung von biometrischen Daten mit einer virtuellen Identität entsteht eine weitere Kopplung sehr sensibler Informationen. Diese Kopplung in Systemen ist aber auch teilweise notwendig, um die in Abb. 2.5 dargestellte Verifikation durchzuführen.

Der Schutz biometrischer Daten soll verhindern, dass die Daten für andere Zwecke als für den ursprünglich bestimmten Zweck verarbeitet werden. Dazu zählt die Verhinderung der Verkettung biometrischer Daten über verschiedene Systeme, Datenbanken und Applikationen hinweg und der Schutz vor Analyse der Daten, die nicht für die Verifikation notwendig sind. Dazu zählen beispielsweise Analysen zur Ermittlung der ethnischen Herkunft, medizinischer Informationen oder zur Gesundheit.

Beispiele für Analysen zur Gewinnung medizinischer Informationen:

- Aus einem Handabdruck lässt sich erkennen, ob die Person am Down-Syndrom leidet, vgl. [KJW+];
- Sprechstörungen bei der Parkinson-Erkrankung, vgl. [DF].

3.2.1 Unumkehrbarkeit

Um eine Nutzung der Daten vor einem nicht bestimmungsgemäßen Zweck zu schützen, sollten die Daten durch eine nicht umkehrbare Transformation vor der Speicherung verarbeitet werden. Eine Möglichkeit bieten die Feature-Extraktionsverfahren, durch die es bereits in der Repräsentation schwer wird, medizinische oder ethnische Daten zu extrahieren. Beispielsweise können aus einer Extraktion der Minutien ([Eck09, S.476]) eines Handabdruck-Scans nicht mehr die typischen Merkmale des Down-Syndroms erkannt werden. Diese Eigenschaft gilt jedoch nicht pauschal für alle Extraktionsverfahren bzw. möglichen Merkmale. Neben einer klassischen Verschlüsselung können die Daten auch in die Form eines Pseudonymous Identifier (PI) und Auxiliary Data (AD) überführt werden. Die bloße Verschlüsselung der biometrischen Daten ist hierbei jedoch kein PI.

Ein PI ist Teil einer erneuerbaren biometrischen Referenz, welche die Identität eines Nutzers nach einer nicht umkehrbaren Transformationen an den biometrischen Daten darstellt. Über diese Transformationen kann unter Zuhilfenahme ggf. vorhandener AD auch mit präsentierten biometrischen Daten eine Verifikation durchgeführt werden. AD sind benutzerabhängige Daten, die notwendig sind, um den PI während der Verifikation zu generieren. Bei Verwendung von PI muss eine AD nicht erstellt und verwendet werden. Wird sie jedoch erstellt, so ist sie Teil der biometrischen Referenz (vgl. [ISO11, S. 1f] und [ISO11, S. 4f]). Eine Veranschaulichung und nähere Erläuterung zu PI und AD erfolgt unter Kapitel 3.5 und in den Abbildungen 3.1–3.5.

3.2.2 Unverkettbarkeit

Die Verkettung von biometrischen Daten beschreibt die Nutzung von Referenzen und biometrischen Daten über verschiedene unabhängige Systeme und Datenbanken hinweg. Um die Verkettung von biometrischen Referenzen über unterschiedliche Applikationen und Datenbanken hinweg zu verhindern, gibt es verschiedene Ansätze:

- Verschlüsselung der Referenz mittels unabhängiger und systemspezifischer Schlüssel;
- Verwendung von Pseudonymous Identifiers;
- Verteilte Speicherung von Identitäten und biometrischen Referenzen bzw. verteilte Speicherung von PI und AD bei erneuerbaren Referenzen.

3.2.3 Personenbezogene Information

Wenn bei biometrischen Systemen ein Sample als Rohdaten[10] hinterlegt wird, entsteht aus diesen biometrischen Daten eine personenbezogene Information (Personally Identifiable Information (PII)). Darüber hinaus lässt sich aus diesen Daten auch eine Unique Universal Identifier (UUID) erstellen, mittels derer über verschiedene Systeme hinweg ein User identifiziert und weitere Daten an diese Referenz gelinkt werden können. Der Standard ISO/IEC 24745 empfiehlt, eine biometrische Referenz niemals als UUID einzusetzen (vgl. [ISO11, S.11]).

[10] Rohdaten können beispielsweise Features, eine Aufnahme des Charakterstikums oder Ähnliches sein.

3.2.4 Vertraulichkeit

Die Vertraulichkeit der Daten spielt auch im Datenschutz eine Rolle. Um einen nicht-autorisierten Zugriff zu verhindern, können die Daten verschlüsselt oder auf zusätzlichen Tokens (z.B. Smartcard) abgelegt werden. Ein zusätzliches Token erhöht die Sicherheit für den Fall eines Zugriffs auf die Datenbank, da sich die Daten aus der Datenbank nur in Kombination mit den Daten des Tokens verwenden lassen. Die Daten auf dem Token müssen in jedem Fall verschlüsselt abgelegt werden, um vor einem unautorisierten Zugriff von außen gesichert zu sein.

3.3 Datenschutz-Management

Um biometrische Daten zu erheben, sollte zu jeder Zeit die Einwilligung des Benutzers vorliegen, solange keine gesetzlichen Vorgaben diese Einschränkung aufheben. Neben den gesetzlichen Bestimmungen zum Datenschutz (z.B. Bundesdatenschutzgesetz in Deutschland [Der10]) muss beim Betreiben eines biometrischen Systems ein Prozess existieren, um den Benutzer über seine Rechte sowie die Verarbeitung der Daten zu informieren. Dieser Prozess kann in die Bereiche Datenerhebung, Übertragung, Verwendung, Speicherung, Archivierung und Löschung unterteilt werden.

3.3.1 Datenerhebung

Der Nutzer muss über die Erhebung der Daten umfänglich informiert werden, sodass seine Einwilligung unmissverständlich vorliegt. Die folgende (unvollständige) Liste gibt eine Übersicht über Informationen, die der Benutzer erhalten sollte:

- Art und Anzahl zu messender biometrischer Daten;
- alternative Prozesse, falls der Benutzer nicht registriert werden will oder kann;
- den Grund der Datenerhebung, sowie die Dauer der Aufbewahrung;
- Übersicht über die Verarbeitung der Daten im System;
- Informationen für die zuständige Person des Betreibers für das System inklusive Name, Position, Kontaktmöglichkeiten.

Die gesetzlichen Bestimmungen zur Informationsbereitstellung und Datenerhebung müssen in jedem Fall erfüllt werden.

3.3.2 Übertragung

In manchen Einsatzszenarien ist es notwendig, die biometrischen Daten an ein anderes System oder Unternehmen für die Verarbeitung oder andere Aufgaben zu übertragen. Aus Sicht des Benutzers ist die Übertragung dieser Informationen an ein anderes System gleichzusetzen mit dem direkten Präsentieren von biometrischen Daten an dieses System. Aus diesem Grund sollte jede dritte Stelle vertraglich gebunden werden, die Informationen entsprechend zu schützen. Die Übertragung der Daten sollte ausschließlich nach Zustimmung des Benutzers durchgeführt werden, solange der angebotene Dienst dies nicht zwangsläufig erfordert oder gesetzliche Vorschriften dies vorschreiben.

Die dem Benutzer zugänglichen Informationen zur Übertragung der Daten sollten mindestens umfassen:

- Informationen an wen Daten übertragen werden, inkl. Name, Kontaktmöglichkeit etc;
- Art und Inhalt der zu übertragenden Daten;
- Grund der Datenübertragung sowie die Dauer der Aufbewahrung.

3.3.3 Verwendung

Verwendung umfasst den Zugriff, die Verarbeitungsschritte sowie die Modifikation der Daten innerhalb des Systems. Ergibt sich ein neuer Kontext im System, so ist der Nutzer erneut über die Verarbeitung und Speicherung der Daten im neuen Kontext zu informieren. Ohne seine Einwilligung sind seine Daten zunächst für diesen Kontext zu sperren.

3.3.4 Speicherung

Biometrische Daten werden wie andere virtuelle Identitäten und Rollen zur Verarbeitung gespeichert. Um größtmöglichen Schutz zu gewährleisten, sind biometrische Daten als sensible private Informationen zu betrachten und zu speichern. Hierzu gehört auch die Separierung anderer, zu dieser Person gehörender, sensibler Informationen. Die Daten sind so zu speichern, dass die Schutzziele Vertraulichkeit und Integrität erfüllt werden.

3.3.5 Archivierung und Backup

Die Archivierung biometrischer Daten für Langzeitanwendungen darf nur mit der Einwilligung des Benutzers geschehen. Wurde in den Nutzungsbedingungen ein Gültigkeitszeit-

raum definiert, darf dieser durch Archivierung nicht außer Kraft gesetzt werden. Backup und Archivierungen sind vor nicht-autorisierten Zugriffen zu schützen und gemäß den Schutzzielen Vertraulichkeit und Integrität zu sichern. Das Wiedereinspielen eines Backups darf die Vertrauenswürdigkeit des Systems nicht beeinflussen.

3.3.6 Löschung

Die erhobenen Daten sind unwiderruflich zu löschen, wenn

- der Grund der Datenerhebung beendet oder für nicht mehr notwendig befunden wird;
- die Dauer der Aufbewahrung überschritten wurde;
- der Benutzer seine Einwilligung widerruft.

Bei verteilten Systemen ist darauf zu achten, dass alle Bestandteile und Verbindungen der erhobenen Daten gelöscht werden. Dies beinhaltet auch die Löschung der Daten aus dem Backup.

3.3.7 Verantwortlichkeit des Systembetreibers

Die Erfüllung der Sicherheits- und Datenschutzanforderungen liegen in der Zuständigkeit des Systembetreibers. Werden Dritte in die Prozesse eingebunden, so muss der Systembetreiber sicherstellen, dass auch Dritte die Anforderungen in ihrer Zuständigkeit erfüllen. Es ist Aufgabe des Betreibers, dem Benutzer folgende Rechte einzuräumen:

- Der Benutzer behält die Rechte an seinen Daten und Informationen für den Zeitraum der Systemnutzung;
- Der Benutzer muss die Möglichkeit haben, seine Einwilligung zu jeder Zeit widerrufen zu können, solange gesetzliche Vorgaben dies nicht verhindern;
- Der Benutzer hat das Recht, Einsicht in seine Daten zu erhalten sowie Anfragen zur Art der Verarbeitung zu stellen;
- Der Betreiber muss den Benutzer bei Einbrüchen, Diebstahl, Verlust oder Schaden an den Daten sofort und umfänglich informieren.

3.4 Modell nach ISO/IEC 24745

Ein biometrisches System kann durch die Art und Weise, an welcher Stelle Daten abgelegt, verarbeitet und verglichen werden, unterschieden werden. Die Verteilung der Daten kann Vor- und Nachteile haben. Der Standard ISO/IEC 24745 schlägt für verschiedene Kombinationen der Verteilung der Prozesse und Datenspeicherung verschiedene Architekturmodelle vor. Das Modell mit geringster Komplexität speichert und verarbeitet die Daten auf demselben Server und wird nachfolgend vorgestellt. Im Standard wird dieses Modell als *Modell A* bezeichnet [ISO11, S. 26]. Durch die geringe Komplexität kann das Modell für eine Referenzimplementierung und Testumgebung schnell und ohne großen Ressourcenaufwand eingesetzt werden. Das Modell bildet die Architektur für das in dieser Arbeit implementierte System.

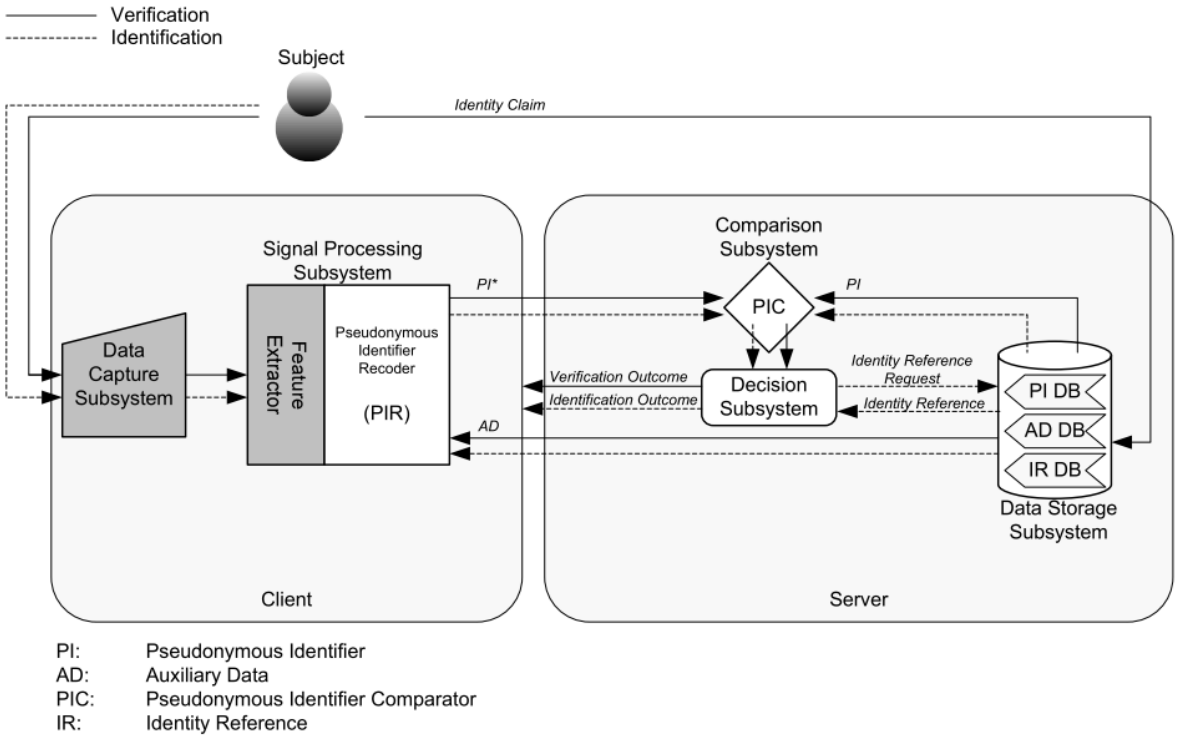

Abbildung 3.1: Architekturmodell – Speicherung und Verarbeitung auf dem Server mit erneuerbaren biometrischen Referenzen, Quelle: [ISO11, S. 25]

Das Modell A des Standards ISO/IEC 24745, zu sehen in Abb. 3.1, teilt die Architektur in zwei Komponenten auf. Die erste Komponente, links in der Abbildung, behandelt die

Prozesse des „Data Capture Subsystem“ und „Signal Processing Subsystem“ (vgl. 2.3.1). Diese Komponente führt die Messung biometrischer Daten und Feature-Extraktion durch.

In der zweiten Komponente, rechts in der Abbildung, befindet sich das „Comparison Subsystem“ und das „Decision Subsystem“ sowie das „Data Storage Subsystem“. Bei der Verifikation wird hier nach der Feature-Extraktion das präsentierte Sample gegen die Referenz aus dem Speicher verglichen. Der Benutzer wird entsprechend des Verifikationsergebnisses akzeptiert oder abgewiesen.

Die Datenbank beinhaltet die erneuerbare Referenz bestehend aus PI und AD und erfüllt formal die Anforderungen zum Schutz der biometrischen Daten (Unumkehrbarkeit, Widerrufbarkeit und Erneuerbarkeit). Die Datenbank kann durch Verschlüsselung (z.B. AES) das Schutzziel Vertraulichkeit erreichen.

3.5 Module

Wie in Abb. 3.1 zu sehen, beinhalten die Komponenten weitere Module, die Funktionen bereitstellen, um die Anforderungen des Standards zu erfüllen. Die Funktionen dieser Module werden nachfolgend beschrieben und werden im Standard ISO/IEC 24745 [ISO11, Annex C] definiert.[11]

Die Module bilden den Kern zur Verwendung von PI im System. Abbildung 3.2 (vgl. [BBGK08, Fig.1] und [ISO11, S.41 C1]) zeigt die Phasen des Lebenszyklus zur Verwendung von PI in einem biometrischen System.

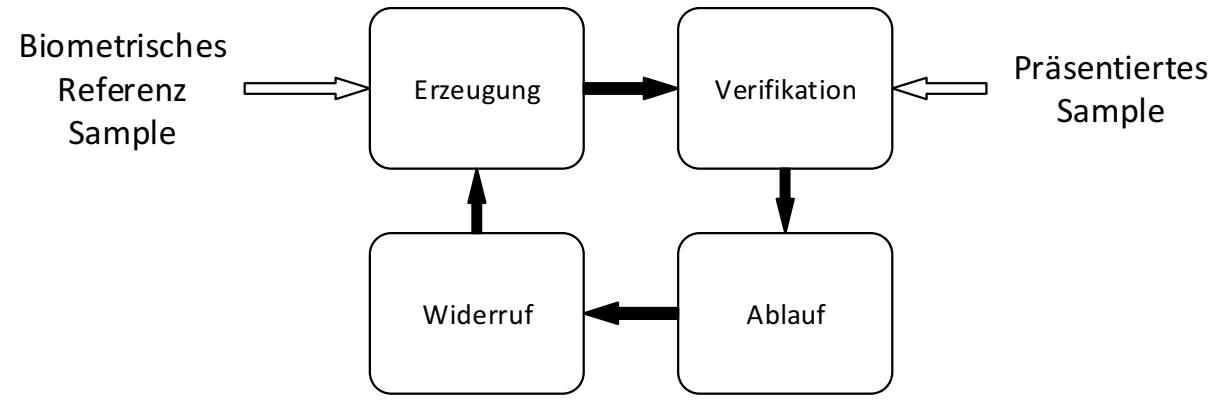

Abbildung 3.2: Lebenszyklus der PI

[11] Einen weiteren Überblick über die Funktion und den Einsatz findet sich in der Ausarbeitung von Breebaart et al. [BBGK08].

Der gesamte Lebenszyklus wird durch die Prozesse in den Modulen ermöglicht und unterteilt sich während der Verwendung in die Erzeugung und die Verifikation. Je nach Bedarf und Notwendigkeit läuft die Referenz ab oder wird widerrufen und muss erneuert werden. Hierdurch schließt sich der Zyklus.

3.5.1 Feature Extractor Modul

Das Feature Extractor Modul erhält als Eingabe ein oder mehrere biometrische Samples und extrahiert aus diesen die Features. Die Merkmalsextraktion der Mel-Frequency Cepstral Coefficients (MFCC) ist unter 2.3.3 beschrieben.

3.5.2 Erzeugungsmodul PIE

Das Erzeugungsmodul Pseudonymous Identifier Encoder (PIE), siehe Abb. 3.3, erhält als Eingabe die extrahierten Features. Aufgabe des Moduls ist die Durchführung des Enrollments und somit die Erzeugung einer erneuerbaren biometrischen Referenz im System ,bestehend aus PI und AD.

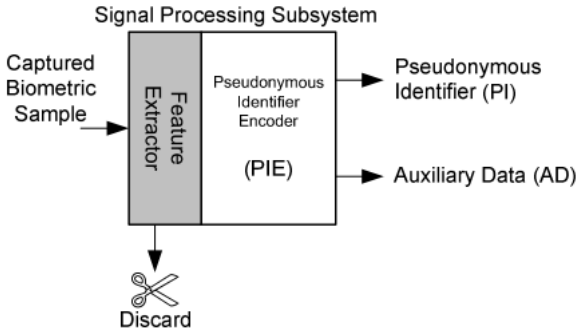

Abbildung 3.3: Erzeugungsmodul PIE zur Erstellung von PI und AD, Quelle: [ISO11, S. 41]

Die Erzeugung von AD ist je nach verwendetem Algorithmus zur Erzeugung nicht bindend. Wird AD jedoch erzeugt, kann es folgende Zwecke erfüllen:

- Erzeugung eines PI aus einem präsentierten Sample zur Vorbereitung eines Vergleichs
- Erzeugung unabhängiger PIs zum Erfüllen der Anforderung Erneuerung
- Parameterisierbarer Bestandteil in der Verifikation, um die biometrische Performanz zu optimieren

PI und AD werden nach dem Enrollment abgespeichert, während das biometrische Sample vollständig zerstört wird.

3.5.3 Vergleichsmodule PIR und PIC

Das Pseudonymous Identifier Recorder (PIR) und Pseudonymous Identifier Comparator (PIC) Modul bilden in diesem Modell das Vergleichsmodul, welches die Verifikation durchführt.

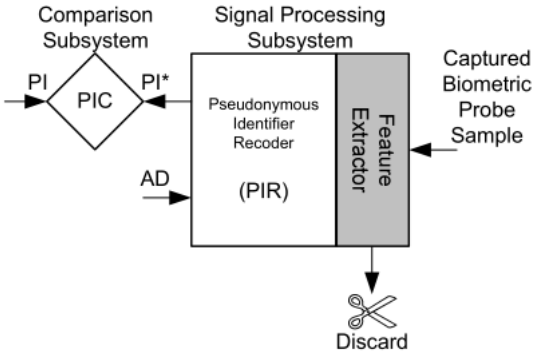

Abbildung 3.4: Vergleichsmodul bestehend aus PIR und PIC, Quelle: [ISO11, S.42]

Das PIR-Modul generiert einen PI* aus den extrahierten Features und der AD des Benutzers. Das PIC-Modul führt anschließend die Verifikation durch, indem PI und PI* verglichen werden. Der Vergleich kann durch eine harte Entscheidung durch Übereinstimmung (Ja/Nein) oder basierend auf einem Score-Wert und einem systemweiten Schwellwertparameter erfolgen.

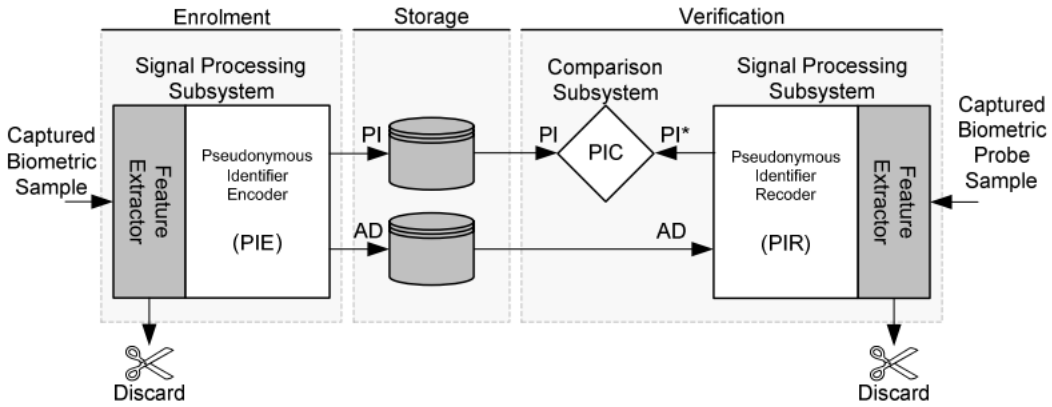

Abbildung 3.5: Module in den Prozessen Enrollment, Speicherung und Verifikation, Quelle: [ISO11, S.43]

Abbildung 3.5 zeigt die Zusammenfassung der vorgestellten Module und das Zusammenspiel in den Prozessen Enrollment, Speicherung und Verifikation. Der vom PIE-Modul generierte PI und AD werden nach dem Enrollment abgespeichert. Zur Verifikation wird die gespeicherte AD verwendet, um mit dem PIR-Modul einen PI zu generieren, welcher anschließend mit dem abgespeicherten PI im PIC-Modul verglichen wird.

4 Template Protection basierend auf UBM und Fuzzy Commitment

Im Folgenden Kapitel wird die Implementierung des Template Protection Systems erläutert. Dabei werden die Bedingungen zur Wahl eines geeigneten Algorithmus und anschließend die zwei Funktionen des Systems, „Enrollment“ und „Verifikation“, beschrieben. Das Enrollment wird hierbei vom PIE Modul und die Verifikation vom PIR und PIC Modul durchgeführt (vgl. Kapitel 3.5.2 - 3.5.3).

Die konkrete Implementierung des Systems basiert auf der Idee eines Template Protection Systems für Online-Signatur von Argones-Rùa et al. [ARMACC12]. Die Verwendung von Hidden Markov Model (HMM)[12] wurde in dieser Arbeit auf den Einsatz von Gaussian Mixture Model (GMM) und Universal Background Model (UBM) angepasst, um aus den Sprachdaten einen Binärvektor zu generieren. Dieser Binärvektor b wird anschließend in ein Fuzzy Commitment Scheme ([JW99]) überführt. Hierfür wird während des Enrollments ein Zufallswert r generiert, der anschließend mit einem fehlerkorrigierenden Code codiert wird. Der Binärvektor wird daraufhin zur Verschleierung des Codeworts c eingesetzt, indem das Codewort bitweise XOR mit dem Binärvektor verknüpft wird und so die AD δ im System bildet. Aus dem generierten Zufallswert wird mittels einer Hashfunktion ein Hash h berechnet und der Zufallswert anschließend vernichtet. Im System werden nach dem Enrollment nur der Hash und das verschleierte Codewort abgespeichert.

Bei der Verifikation wird aus der Probe ein Binärvektor b^*generiert, der zur Entschleierung des Codeworts bitweise XOR mit der AD δ verknüpft wird. Anschließend wird das Codewort mit dem fehlerkorrigierenden Code decodiert. Die hieraus resultierende Nachricht $r*$ wird mit derselben Hashfunktion gehasht und mit dem abgespeicherten Hash verglichen. Stimmen diese Hashes überein, ist der Benutzer verifiziert.

[12] Weiterführende Literatur [Rab89], [ST95, S.121f].

4.1 Anforderungen an das Template Protection System

Die Ergebnisse dieser Arbeit sollen eine standardkonforme Erweiterung für die atip Software Voxguard ermöglichen. Voxguard ist eine von der atip GmbH entwickelte Software zur textunabhängigen Sprecherverifikation. Sie ist Bestandteil der atip Sprachplattform. Die Sprechermodelle werden mittels GMM und UBM modelliert und verifiziert. Sie bieten eine hohe biometrische Performanz[13]. Um den Funktionsumfang der Voxguard Software mit der Fähigkeit von Template Protection zu erweitern, muss das System auf der Verwendung von GMM und UBM aufsetzen.

4.1.1 Gaussian Mixture Model (GMM)

Ein Gaussian Mixture Model λ ist ein gewichtetes, stochastisches Modell bestehend aus einer festen Anzahl I multivariater Normalverteilungen. Die Normalverteilungen werden bei GMMs auch als Komponenten bezeichnet. Eine Komponente ist somit definiert als

$$P(x|\lambda) = \sum_{i=1}^{I} w_i \mathcal{N}(x|\mu_i, \Sigma_i) \tag{4.1}$$

wobei w_i das Gewicht, μ_i der Mittelwert und Σ_i die Kovarianzmatrix der Komponente i ist. $\mathcal{N}(x|\mu_i, \Sigma_i)$ bezeichnet eine multivariate Normalverteilung. Der Ausdruck aus Gleichung 4.1 definiert das Wahrscheinlichkeitsmaß, zu dem x vom GMM λ abgebildet werden kann. Für einen Sprecher u und das zugehörige, sprecherspezifisch trainierte GMM $\lambda^{(u)}$ kann somit eine Aussage über die Ähnlichkeit einer abgegebenen Probe x zu diesem Sprechermodell gemacht werden.

4.1.2 Universal Background Model (UBM)

Ein Universal Background Model Λ ist ein systemspezifisches, sprecherunabhängiges GMM. Während ein GMM einen spezifischen Sprecher und seinen Sprecherraum modelliert, hat das UBM das Ziel, einen universalen Sprecherraum abzubilden, weshalb es auch als Weltmodell bezeichnet wird. Für eine abgegebene Probe x kann somit analog zu Gleichung 4.1 eine Aussage über die Ähnlichkeit zum UBM gemacht werden.

[13] Basierend auf internen Versuchsreihen. Vergleichbare Evaluationen zur Leistungsfähigkeit von GMM in der Sprecherverifikation finden sich bspw. unter [RQD00] oder [KL10].

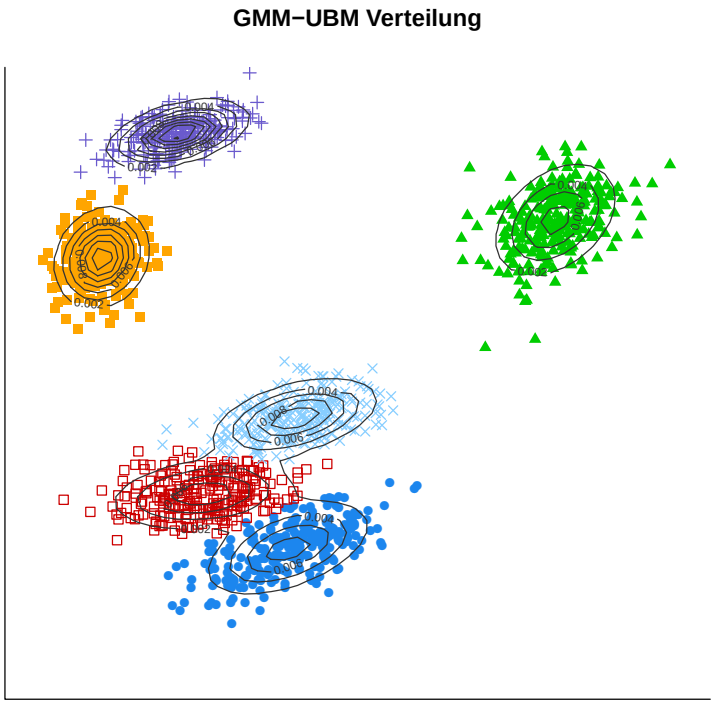

Abbildung 4.1: Beispielhafte Darstellung eines UBM im zweidimensionalen Raum mit sechs Komponenten, Quelle: [Heg13]

Die Abbildung 4.1 zeigt eine beispielhafte Darstellung eines UBM im zweidimensionalen Raum mit sechs Komponenten.

4.2 Enrollment

Um einen Sprecher im System zu registrieren, muss dieser den Prozess des Enrollments durchlaufen. Dabei werden eine Menge von Trainingsdaten vom Sprecher erhoben, indem Sprachsamples aufgezeichnet werden.

Die erhobenen Trainingsdaten des Sprechers u werden verwendet, um vom UBM Λ ein sprecherspezifisches GMM λ zu adaptieren. Das eingesetzte Verfahren ist hierbei das Maximum A Posteriori (MAP)-Verfahren [Rey].

Die Abbildung 4.2 zeigt das beispielhafte UBM aus Abbildung 4.1 mit sprecherspezifischen eingezeichneten Trainingsdaten. Mithilfe dieser Daten kann das UBM adaptiert werden, um das sprechersprezifische GMM $\lambda^{(u)}$ zu erhalten

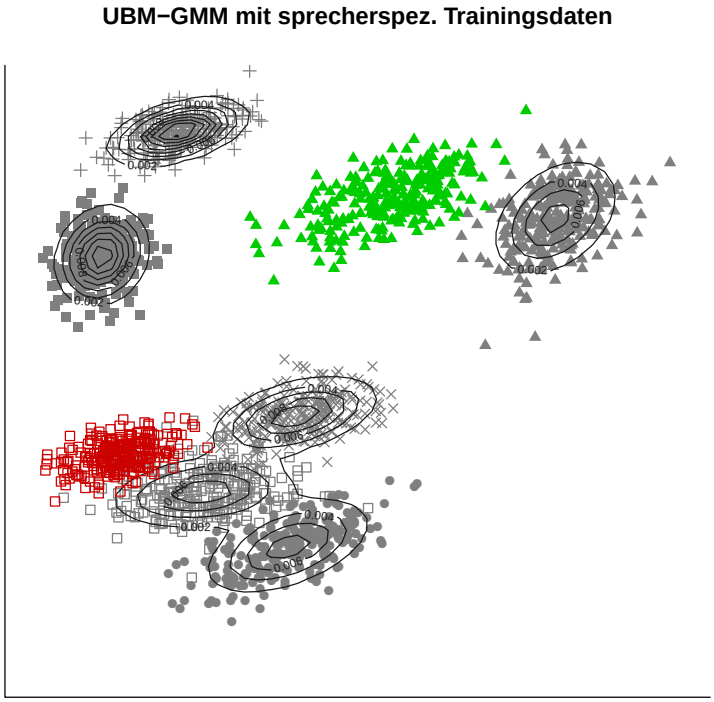

Abbildung 4.2: UBM mit eingetragenen sprecherspezifischen Trainingsdaten, Quelle: [Heg13]

Im Nachfolgenden wird der Prozess der Adaption des Sprechermodells und somit des Enrollmentprozesses für einen beispielhaften Sprecher u erläutert.

4.2.1 Adaption des Sprechermodells mittels Maximum A Posteriori

Aus den erhobenen Traininsdaten werden die Feature-Vektoren für jedes Sample (vgl. Kapitel 2.3.3) extrahiert. Die Menge an E Samples wird als Beobachtung $O^{(u)e}$ mit $e = 1, \ldots, E$ bezeichnet. Die Menge an T extrahierten Feature-Vektoren bezeichnen wir als Trainingsvektoren $\mathbf{X} = \{x_t, \ldots, x_T\}$ mit $t = 1, \ldots, T$.

Hierzu wird die a posteriori Wahrscheinlichkeit für eine Komponente i berechnet:

$$P(i|\mathbf{X}, \mathbf{\Lambda}) = \frac{w_i \mathcal{N}(\mathbf{X}|\mu_i, \sum_i)}{\sum_{j=1}^{I} w_j \mathcal{N}(\mathbf{X}|\mu_j, \sum_j)} \tag{4.2}$$

Mit dieser Wahrscheinlichkeit wird anschließend eine Maximum-Likelihood Schätzung[14] durchgeführt:

$$E_i(\mathbf{X}) = \frac{1}{n_i} \sum_{t=1}^{T} Pr\left(i|x_t, \mathbf{\Lambda}\right) x_t \tag{4.3}$$

wobei n_i die relative Anzahl zugehöriger Trainingsvektoren beschreibt:

$$n_i = \sum_{t=1}^{T} Pr\left(i|x_t, \mathbf{\Lambda}\right) \tag{4.4}$$

Der neue Mittelwertsvektor $\widehat{\mu}_i$ einer Komponente wird anschließend ermittelt:

$$\widehat{\mu}_i(\mathbf{X}) = \boldsymbol{\alpha}_i E_i(\mathbf{X}) + (1 - \alpha_i)\mu_i \tag{4.5}$$

wobei α_i ein datenabhängigen Faktor ([RQD00, S. 28], [Rey]) ist. Die Anpassung wird solange neu berechnet, bis keine hinreichenden Änderungen mehr bei der Adaption erreicht werden. $\widehat{\boldsymbol{\mu}}_i$ bildet anschließend den neuen Mittelwertsvektor für die Komponente i im sprecherspezifischen Modell λ. Alle Mittelwertsvektoren für die Komponenten können zusammengefasst als Supervektor bezeichnet werden:

$$\boldsymbol{\xi}^{(u)} = [\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_I] \tag{4.6}$$

Die Gewichtung der Komponenten ist für die weiteren Verarbeitungsschritte nicht relevant, weshalb diese verworfen werden.

In Abbildung 4.3 sieht man den Endzustand des beispielhaft adaptierten GMM aus dem UBM aus Abbildung 4.1.

Grundsätzlich wird die MAP Adaption für die Gesamtheit aller vorliegenden Audios durchgeführt. Das bedeutet, dass aus allen extrahierten Feature-Vektoren aller E Beobachtungen ein einzelnes GMM adaptiert wird.[15]

Die MAP Adaption kann jedoch auch für eine einzelne Beobachtung, d.h. $E = 1$, durchgeführt werden. Dadurch kann bei der Verifikation aus einer einzelnen Sprachäußerung ein Supervektor generiert werden. Diese Eigenschaft ermöglicht es, für jedes Sample einen Supervektor zu berechnen, zu binarisieren und darauf aufbauend eine Verarbeitung einzuleiten.

[14] Weiterführende Literatur zur Maximum-Likelihood Schätzung unter [Wun01].
[15] Dieser Schritt wird in der herkömmlichen GMM/UBM Verifikation durchgeführt.

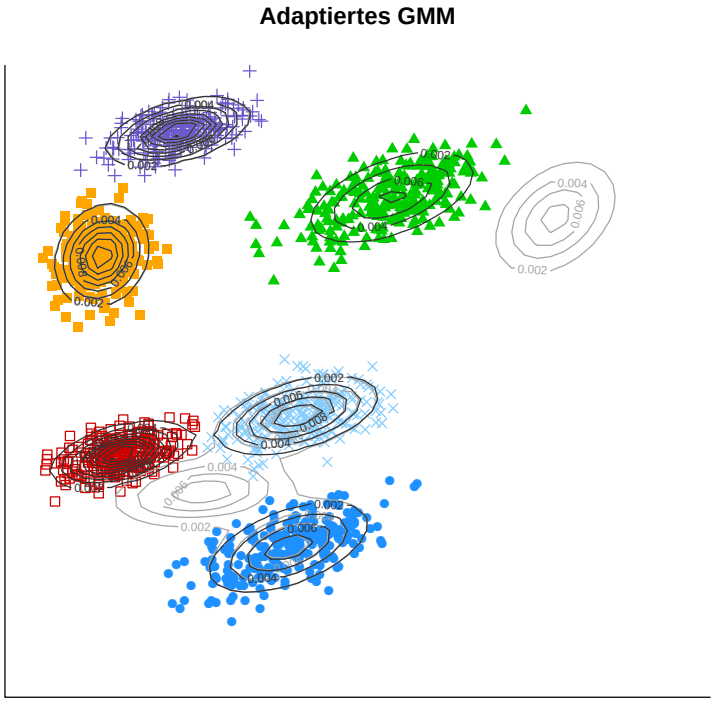

Abbildung 4.3: Adaptiertes GMM aus UBM, Quelle: [Heg13]

Für das Enrollment werden E Supervektoren berechnet, indem jede Beobachtung $O^{(u)e}$ zur Adaption eines Modells herangezogen wird.

4.2.2 Binarisierung

Um aus den generierten Supervektoren einen Binärvektor zu generieren, wird während des Enrollments zuerst der komponentenweise Mittelwert aller Supervektoren gebildet:

$$d^{(u)} = \frac{1}{E} \sum_{e=1}^{E} \boldsymbol{\xi}^{(u)e} \tag{4.7}$$

Aus diesem gemittelten Supervektor wird nun der Binärvektor generiert, indem er komponentenweise gegen eine vorab für das System trainierte Grundgesamtheit[16] verglichen wird:

$$\mathbf{b}^{(u)}[z] = \begin{cases} 0, \text{ if } d^{(u)}[z] < \overline{\mu}[z] \\ 1, \text{ if } d^{(u)}[z] \geq \overline{\mu}[z] \end{cases} \quad z = 1, \ldots, Z \tag{4.8}$$

Z entspricht hierbei der Anzahl aller Komponenten aus dem Supervektor und ergibt sich aus der Anzahl der Komponenten des UBM sowie der Anzahl Komponenten des Featurevektors $Z = I * 39$. Für ein UBM mit $I = 128$ Komponenten ergibt sich somit ein Binärvektor der Länge $Z = 128 * 39 = 4992$ Bit. Die Auswahl der Komponentenanzahl des UBM hat somit direkte Auswirkung auf die maximale Länge des generierten Binärvektors.

4.2.3 Optimierung zur Diskriminanzsteigerung des Binärvektors

Zur Optimierung der Trennfähigkeit von Binärvektoren zwischen Genuine und Impostor, aber auch zur Anpassung an Fehlerkorrekturverfahren, sollen die generierten Binärvektoren – im Optimalfall – nur die Bits beinhalten, welche die höchstmögliche Diskriminanz ermöglichen.

Zur Beurteilung der Diskriminanz wird hierfür das Diskriminanzmaß φ eingeführt (vgl. [ARMACC12, III A.]). Das Maß ist definiert als ein angepasster statistischer Z-Score[17]

$$\varphi^{(u)}[z] = \frac{|d^{(u)}[z] - \overline{\mu}[z]|}{\sigma^{(u)}[z]} \tag{4.9}$$

wobei $\sigma^{(u)}[z]$ die Standardabweichung der Features aus $d^{(u)}$ ist. Der Unterschied zur Berechnung eines Z-Scores besteht in der Betragsbildung des Zählers. Dies führt dazu, dass keine negativen Werte entstehen. Das Maß ermöglicht es, die Bitstellen zu identifizieren, deren Feature-Wert des Vektors d einen größeren Abstand zur Population $\overline{\mu}$ besitzen und somit mit größerer Wahrscheinlichkeit bei jedem Verifikationsversuch eines Sprechers die gleiche Bitwertigkeit generieren.

[16] Die Grundgesamtheit stellt hierbei ein einmalig für das System trainiertes Modell aus einer Stichprobe dar. Dabei werden aus Äußerungen unterschiedlichster Sprecher Supervektoren wie in Kapitel 4.2 generiert und anschließend der Mittelwertssupervektor $\overline{\mu}$ aus allen Supervektoren berechnet.

[17] Der Z-Score ist eine statistische Berechnungsmethode um standardisierte Daten aus unterschiedlichen Ausgangsgrößen zu berechnen. Als Ergebnis erhält man für einen Wert X eine Zahl im Bereich -2 bis +2. Diese Zahl beschreibt den Abstand von X zum Mittelwert seiner Messreihe in der Einheit Standardabweichung. Weiterführende Informationen finden sich unter [CGU].

Der Vektor $\varphi^{(u)}$ wird hierfür in absteigender Reihenfolge sortiert und die Indizes der n größten Werte im Vektor $RP^{(u)}$ abgespeichert. $RP^{(u)}$ beinhaltet somit die relevanten Indizes eines spezifischen Sprechers.

Anschließend werden die nicht-relevanten Positionen der Bits aus $b^{(u)}$ – das sind die Indizes, die nicht in $RP^{(u)}$ enthalten sind – eliminiert. Der daraus resultierende Binärvektor der Länge n soll somit für Genuines weniger unterschiedlich und gegenüber Impostors diskriminativer sein. Eine Evaluation hierzu erfolgt unter Kapitel 5.3.

4.2.4 Generierung der erneuerbaren biometrischen Referenz mit dem Fuzzy Commitment Scheme

Nachdem der binäre Enrollmentvektor generiert wurde, muss nun die biometrische Referenz erzeugt werden. Hierzu wird mittels eines Zufallszahlengenerators ([Bun13, S. 54]) ein Schlüssel r mit k Bits generiert, wobei die Entropie[18] des generierten Binärvektors $b^{(u)}$ größer als die Länge des Schlüssels sein muss.

Unter dieser Bedingung verbirgt sich die Eigenschaft, dass bei geringerer Entropie des Binärvektors $b^{(u)}$ die Einzigartigkeit desselbigen sinkt. Hohe Entropie bedeutet, dass das Auftreten des Vektors selten zu beobachten ist, während bei geringer Entropie ein Binärvektor sehr oft zu beobachten ist. Tritt ein Binärvektor oft auf, so ist die Diskriminanz des Binärvektors nicht hoch, denn verschiedene Sprecher generieren womöglich denselben Binärvektor. Dies führt dazu, dass der Schlüssel r im System durch einen Binärvektor geschützt werden würde, der mit größerer Wahrscheinlichkeit von verschiedenen Sprechern generiert wird und somit einen schwachen Schutz des Schlüssels bringt.

Im nächsten Schritt wird mit der Encoder-Funktion eines fehlerkorrigierenden Codes (ECC) der Schlüssel codiert. Der eingesetzte fehlerkorrigierende Code ist der Hadamard Code [Moo06, Lecture 8], [Wei], [Dam10, S.25]. Der Hadamard Code ist ein $[n, k, d]$-Code mit der Eigenschaft $[2^n, n+1, 2^{n-1}]$ wobei $t = 2^{n-2} - 1$ Fehler korrigiert werden können:

$$encode(r, k, n) = c \tag{4.10}$$

Das Ergebnis ist das Codewort c der Länge n.

[18] Weiterführende Literatur [Eck09, S.295].

Das Codewort wird nun mit dem Binärvektor bitweise XOR verknüpft und die AD, der Differenzvektor δ gebildet:

$$\delta^{(u)} = b^{(u)} \oplus c^{(u)} \tag{4.11}$$

Aus dem Schlüssel r wird nun mittels einer starken Hashfunktion ein Hashwert berechnet:

$$H(r) = h \tag{4.12}$$

Der Zufallswert wird verworfen und der Differenzvektor δ sowie der Hash h werden abgespeichert. Die Kombination aus der AD und dem PI nennt sich Fuzzy Commitment[19] und kann als 2-Tupel definiert werden:

$$FC^{(u)} = (\delta^{(u)}, h)$$

Der Hash bildet hierbei den PI und der Differenzvektor δ die AD. Weiterhin bildet bei Verwendung des Vektors $RP^{(u)}$ der Vektor eine weitere AD und wird gespeichert, da diese notwendig ist, um den gesuchten Binärvektor zu generieren. Das biometrische Template T entspricht dadurch einem Tripel:

$$T^{(u)} = \left\{h, RP^{(u)}, \delta^{(u)}\right\}$$

Abbildung 4.4 zeigt das soeben beschriebene Schema des Enrollments.

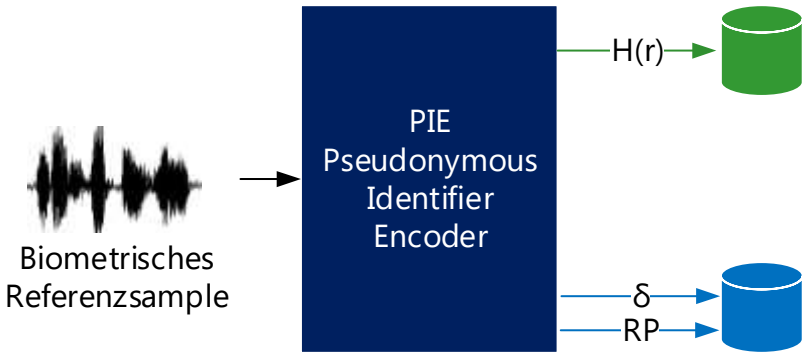

Abbildung 4.4: Schema des Enrollmentprozesses mit konkreten Werten für PI und AD

[19] Vgl. Fuzzy Commitment Scheme von Juels et. al [JW99].

4.3 Verifikation

Um sich bei einem biometrischen System zu verifizieren, muss ein Sprecher eine biometrische Probe abgeben. Gleichzeitig muss eine Angabe gemacht werden, gegen welchen Benutzer er sich verizifieren möchte. Diese Angabe nennt man claimed ID. Den sich verifizierenden Sprecher bezeichnen wir dabei als $\tilde{u}$ und die claimed ID als u. Das aufgezeichnete Sprachsample wird vom PIR Modul wie folgt verarbeitet, um den Vergleich des PI vorzubereiten.

4.3.1 Generierung der Vergleichsreferenz

Aus dem aufgezeichneten Sprachsample werden die Feature-Vektoren x_T extrahiert (vgl. Kapitel 2.3.3). Diese werden nun herangezogen, um den Supervekor des GMM $\lambda^{(\tilde{u})}$ aus dem UBM Λ zu adaptieren. Hierbei werden die gleichen Schritte wie beim Enrollment (Berechnung von Gleichung 4.2 bis 4.6) durchgeführt, wobei $E = 1$ ist. Der adaptierte Supervektor wird anschließend binarisiert (Gleichung 4.8) und alle Indizes, die nicht in $RP^{(u)}$ enthalten sind, werden eliminiert. Mit dem Verifikationsbinärvektor $b^{(\tilde{u})*}$ wird nun das Codewort aus dem abgespeicherten δ extrahiert:

$$c^{(\tilde{u})*} = b^{(\tilde{u})*} \oplus \delta^{(u)} \tag{4.13}$$

Durch die Varianzen in Sprachäußerungen sind die generierten Binärvektoren zu hoher Wahrscheinlichkeit nicht 100% identisch mit dem beim Enrollment generierten Binärvektor. Diese Eigenschaft trifft auch für Sprachsamples desselben Sprechers zu (vgl. Kapitel 2.1). Aus diesem Grund ist das durch die XOR-Operation extrahierte Codewort nicht mit dem beim Enrollment generierten Binärvektor identisch. Es unterscheidet sich an d Stellen, wobei d durch die Hamming-Distanz zwischen $b^{(\tilde{u})*}$ und $b^{(u)}$ bestimmt werden kann.[20]

Die hierdurch im Codewort $c^{(\tilde{u})*}$ entstehenden Fehler müssen korrigiert werden.

4.3.2 Fehlerkorrektur

Für die Fehlerkorrektur wird das extrahierte Codewort durch die Matrix Γ des fehlerkorrigierenden Codes zu einem existierenden Codewort des Codes decodiert.

[20] Die Unterschiedlichkeit trifft gleichzeitig auch eine Aussage über die Trennungsmöglichkeiten von Genuine und Impostor. Eine Evaluation findet sich unter Kapitel 5.3.

Der Hadamard Code hat eine Fehlerkorrekturrate[21] von 50%. Das aus der XOR-Operation hervorgegangene Codewort wird auf das Codewort mit minimalster Distanz bezogen auf die Hamming-Distanz zwischen $c^{(\tilde{u})*}$ und $c \in \Gamma$ gemappt. Hierfür wird mittels Maximum-Likelihood-Decodierung ([BFK+98, S.7]) die Hamming-Distanz zu allen Codewörtern der Matrix Γ bestimmt und $c^{(\tilde{u})*}$ mit dem Codewort $c \in H$ ersetzt, welches die minimalste Distanz besitzt. Für den Fall, dass mehrere Codewörter dieselbe minimalste Distanz aufweisen, wird das aus diesem Prozess zuletzt betrachtete Codewort mit minimalster Distanz verwendet.

Aus dem Codewort wird im letzten Schritt des Decodierungsprozesses die Nachricht $r^{(\tilde{u})*}$ extrahiert:

$$decode(c^{(\tilde{u})*}) = r^* \tag{4.14}$$

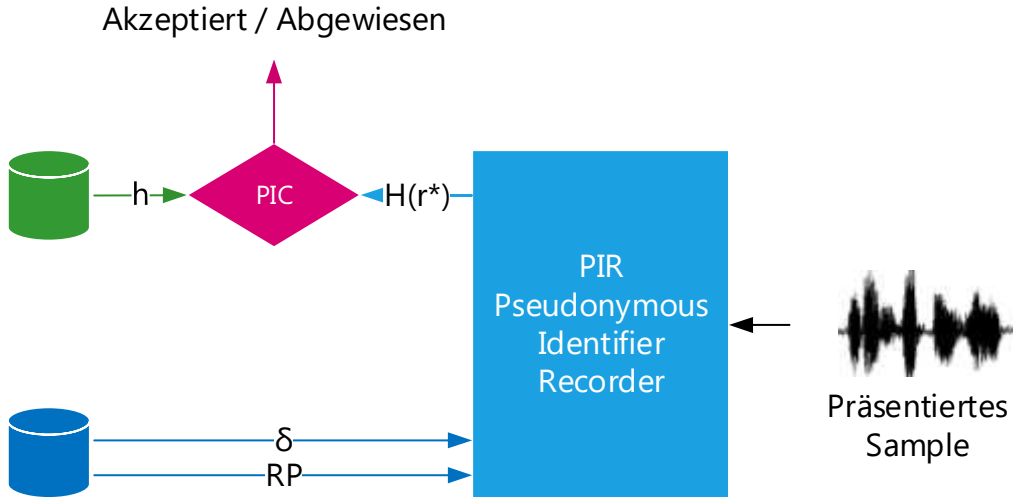

Abbildung 4.5: Schema des Verifikationsprozesses mit konkreten Werten für PI und AD

[21] Gibt den Anteil der in jeder codierten Nachricht korrigierbaren Übertragungsfehler an [BFK+98, S.3].

4.3.3 Verifikationsentscheidung

Der zur Verifikation aus r^* generierte Hash $H(r^*) = h^*$ entspricht auch hier einem PI, den wir nachfolgend als PI^* bezeichnen. Im PIC Modul werden nun der gespeicherte PI und der generierte PI^* verglichen. Die beiden Hashwerte müssen identisch sein, um den Sprecher $\tilde{u}$ als Sprecher u zu verifizieren.

Die Verifikationsentscheidung entspricht somit der Entscheidung:

$$\textit{Verifikationsentscheidung} = \begin{cases} \textit{abgewiesen}, & \text{wenn } H(r) \neq H(r^*) \\ \textit{akzeptiert}, & \text{wenn } H(r) = H(r^*) \end{cases} \tag{4.15}$$

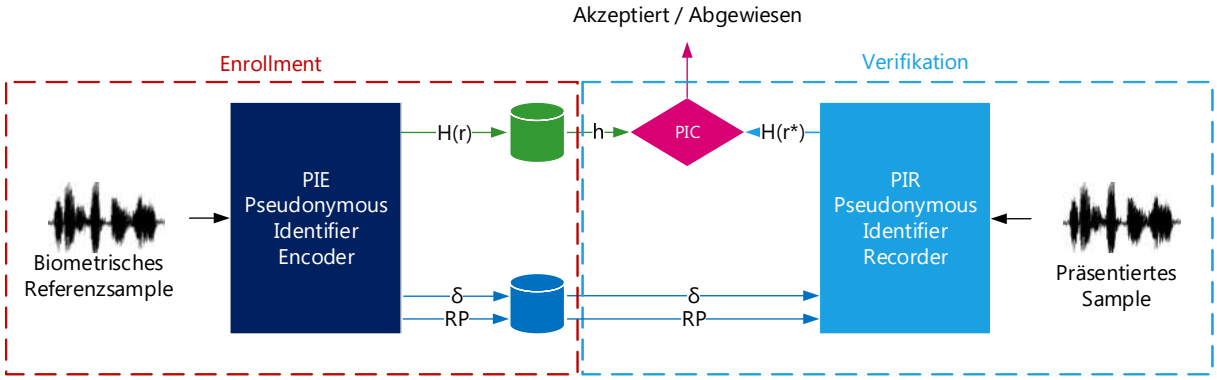

Abbildung 4.6: Architektur des Template Protection Systems mit konkreten Werten für PI und AD

Abbildung 4.6 zeigt die vollständige Architektur sowie das Zusammenspiel aller Komponenten mit konkreten Werten für die Prozesse Enrollment und Verifikation.

5 Evaluation des Template Protection Systems

Neben der gesteigerten Sicherheit der Templates im System spielt vor allem die biometrische Performanz eine wichtige Rolle zur Beurteilung der Leistungsfähigkeit des Systems.

Im folgenden Kapitel wird die Evaluation der biometrischen Performanz des konstruierten Systems durchgeführt. Hierbei wurde das System schrittweise getestet und bewertet. Im ersten Schritt wurde die Trennfähigkeit von Genuines und Impostors anhand von Binärvektoren getestet. Anschließend wurde das System um die Verwendung von PI und AD erweitert. Die hierbei erzielte Verifikationsleistung wird durch Angaben der algorithmischen Fehlerraten FMR und FNMR aufgezeigt (vgl. Kapitel 2.5).

Den Abschluss bildet eine Analyse zur Erfüllung der Anforderungen Widerrufbarkeit, Erneuerbarkeit, Unumkehrbarkeit und Unverkettbarkeit.

5.1 Testanforderungen

Um die algorithmischen Fehlerraten eines biometrischen Systems zu bestimmen, ist ein Datensatz erforderlich, der im Optimalfall die gewünschten realen Einsatzbedingungen abbildet. Für Sprecherverifikationssysteme sind verschiedene Einsatzmöglichkeiten denkbar. Eine Herausforderung stellt hierbei der textunabhängige Einsatz mit kurzen Sprachaufnahmen (wenige Sekunden) dar, da die Datenmenge während der Verifikation sehr gering ist. Für einen realen Einsatz ist sie jedoch besonders interessant. Durch kurze Sprachaufnahmen von wenigen Sekunden ist der Eingabeaufwand des Benutzers nicht groß und der Benutzer wird in der Durchführung seiner eigentlichen Aufgabe nicht behindert. Durch einen Verzicht auf Vorgabe des zu sprechenden Textes könnte die Verifikation sogar nebenläufig[22] durchgeführt werden.

[22]Nebenläufig bedeutet, dass das System z.B. während eines laufenden Telefonats die Verifikation anhand von Mitschnitten durchführt (vgl. [Kun11]).

5.1.1 Sprachkorpus

Für das Testszenario wurde ein bei der atip GmbH vorhandener gemischtgeschlechtlicher Sprachkorpus eingesetzt. Die Sprachaufnahmen bestehen aus drei bis fünf gesprochenen Ziffern und haben eine Dauer von 2 bis 5 Sekunden. Wie in Abbildung 5.1 dargestellt, wurde der Korpus in drei Datensätze aufgeteilt.

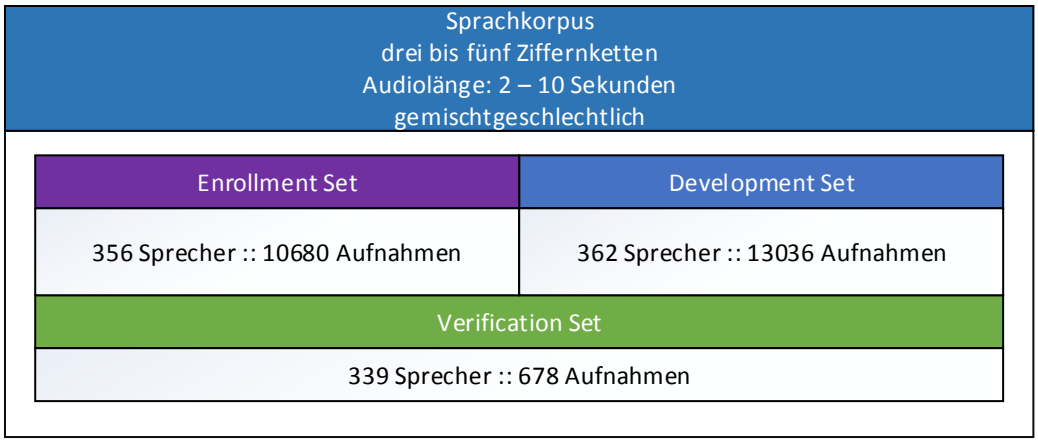

Abbildung 5.1: Übersicht der Aufteilung des verwendeten Sprachkorpus

Das Development Set besteht aus 362 Sprechern. Mit diesem wurde das UBM sowie die Population $\overline{\mu}$ trainiert. Das Enrollment Set wurde verwendet, um Sprecher im System zu registrieren. Für jeden der darin enthaltenen 356 Sprecher sind 30 Aufnahmen vorhanden. Das Verification Set besteht aus 339 Sprechern mit je zwei Aufnahmen. In der Zuordnung Enrollment Set zu Verification Set ergibt sich somit, dass für 17 Sprecher im System das Profil bei Impostor-Verifikationsversuchen verwendet werden konnte, jedoch keine Genuine-Verifikationsversuche durchgeführt wurden.

5.1.2 Testablauf

Die unter Kapitel 2.5 beschrieben Fehlerraten bestimmen sich aus den vorhandenen Daten des Sprachkorpus und einer festgelegten Systemkonfiguration. Hierbei werden die im System vorhandenen Parameter eingestellt und anschließend das Enrollment Set sowie Verification Set mit diesen Konfigurationen vom System verarbeitet. Die im System konfigurierbaren Parameter sind:

- UBM: Das zu verwendende UBM. Hieraus ergibt sich die Anzahl an Komponenten und die maximale Länge des generierten Binärvektors;
- n: Die Länge des generierten Binärvektors. Hieraus resultiert auch die Länge der Codewörter;
- k: Die Länge des Schlüssels – sie ist im Allgemeinen abhängig von n;
- RP: Verwendet den Vektor RP und ermöglicht somit auch Einstellung der Länge n;
- τ: Der Schwellwertparameter zur Verifikationsentscheidung auf Binärvektorebene.

Zu Beginn eines Testdurchlaufs werden die Datensätze aus dem Enrollment Set verwendet, um die Sprecher – wie unter Kapitel 4.2 beschrieben – im System zu registrieren. Anschließend wird das Verification Set verwendet, um die Genuine-Verifikationsversuche durchzuführen. Hierbei wird jeder Sprecher aus dem Verification Set gegen das zugehörige Profil aus dem Enrollment Set verifiziert. Nachdem die Genuine-Verifikationsversuche durchgeführt wurden, werden alle Daten aus dem Verification Set zur Generierung von Impostor-Verifikationsversuchen verwendet. Hierbei werden die Sprecher aus dem Verification Set gegen alle vorhandenen Sprecher des Enrollment Sets, ausschließlich der zugehörigen Profile, getestet. Abbildung 5.2 zeigt beispielhaft den Testablauf. Ein grüner Pfeil beschreibt einen Genuine-Verifikationsversuch und ein roter Pfeil einen Impostor-Verifikationsversuch.

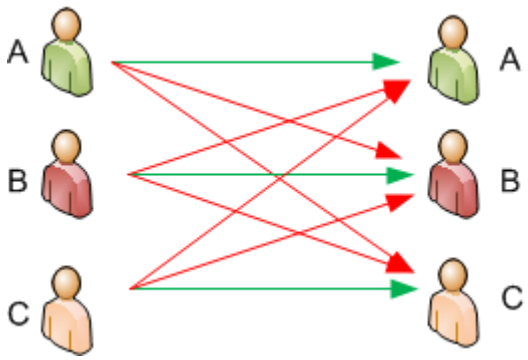

Abbildung 5.2: Beispielhafter Testablauf für Genuine- und Impostor-Verifikationsversuche

Insgesamt ergeben sich durch die Datensätze 678 Genuine-Verifikationsversuche und 240690 Impostor-Verifikationsversuche in einem Testdurchlauf.

5.2 Trennung von Genuine und Impostor anhand von Binärvektoren

Um zu zeigen, dass die generierten Binärvektoren eine Trennung von Genuine und Impostor ermöglichen, muss ein Ähnlichkeitsmaß als Entscheidungskriterium verwendet werden. Unter [Lin98] werden drei Anforderungen über die Ähnlichkeit von zwei Objekten in Bezug auf eine informationstheoretische Betrachtung gemacht:

1. **Gemeinsamkeit**: Je mehr Gemeinsamkeiten zwei Objekte besitzen, desto ähnlicher sind sie.
2. **Unterschiedlichkeit**: Je weniger Unterschiede zwei Objekte besitzen, desto ähnlicher sind sie.
3. **Identität**: Zwei Objekte erreichen die maximale Ähnlichkeit nur dann, wenn sie identisch sind.

Diese drei Anforderungen treffen auch auf die Anforderung an die generierten Binärvektoren zu, um Genuine und Impostor anhand von Binärvektoren zu unterscheiden. Für einen Genuine-Verifikationsversuch ist im Optimalfall der beim Enrollment generierte Binärvektor b mit dem Binärvektor b^*, der bei der Verifikation generiert wird, identisch. Durch die Varianz in Sprachäußerungen ist dies nicht der Fall. Für einen Impostor-Verifikationsversuch muss die Unterschiedlichkeit möglichst groß – im Optimalfall maximal – sein. Hieraus folgt, dass die durch Varianz verursachte Unterschiedlichkeit bei Genuine-Verifikationen möglichst gering – gegenüber Impostor aber möglichst hoch – sein muss.

Eine Beurteilung zur Ähnlichkeit von zwei Binärvektoren b, b^* liefert die Hamming-Distanz. Das Ergebnis der Hamming-Distanz ist die Anzahl an Bitstellen, an denen zwei Binärvektoren unterschiedlich sind. Für die drei genannten Anforderungen kann das Ergebnis der Hamming-Distanz[23] folgende Aussagen treffen:

$$\begin{aligned} \textbf{Identität:}\ & HD(b,b^*) = 0 \ ,\ \textit{für}\ b = b^* \\ \textbf{Unterschiedlichkeit:}\ & HD(b,b^*) = x \ ,\ \textit{für}\ b \neq b^*\ \textit{und}\ 1 \leq x \leq |b| \\ \textbf{Gemeinsamkeit:}\ & HD(b,b^*) = |b| - x \ ,\ \textit{für}\ b \neq b^*\ \textit{und}\ 0 \leq x \leq |b| \end{aligned} \tag{5.1}$$

[23] Eine umfangreichere Definition der Hamming-Metrik findet sich unter [BBF+06, S.13].

Aus dieser Interpretation des Ergebnisses der Hamming-Distanz kann nun untersucht werden, ob sich Genuine- und Impostor bei der Verifikation unterscheiden lassen. Hierbei wird die Hamming-Distanz als Score verwendet, der den prozentualen Anteil unterschiedlicher Bitstellen angibt:

$$HD_{score}(b, b^*) = \frac{||b \oplus b^*||}{n} \tag{5.2}$$

Der Score liegt im Wertebereich zwischen 0.0 (Identität) und 1.0 (volle Unterschiedlichkeit) und ermöglicht eine Vergleichbarkeit für verschiedene Systemkonfigurationen.

5.3 Verifikation anhand von Binärvektoren

Für die Ergebnisse aus Abbildung 5.3 wurde das System so konfiguriert, dass keine Relevant Projection $RP^{(u)}$ (vgl. Kapitel 4.2.3) verwendet wird. Hierdurch wird die maximale Anzahl[24] möglicher Bits für die Binärvektoren generiert. Gemessen wurde die Hamming-Distanz für Genuine- und Impostor-Verifikationsversuche zwischen b und b^* und anschließend in einem Histogramm dargestellt.[25]

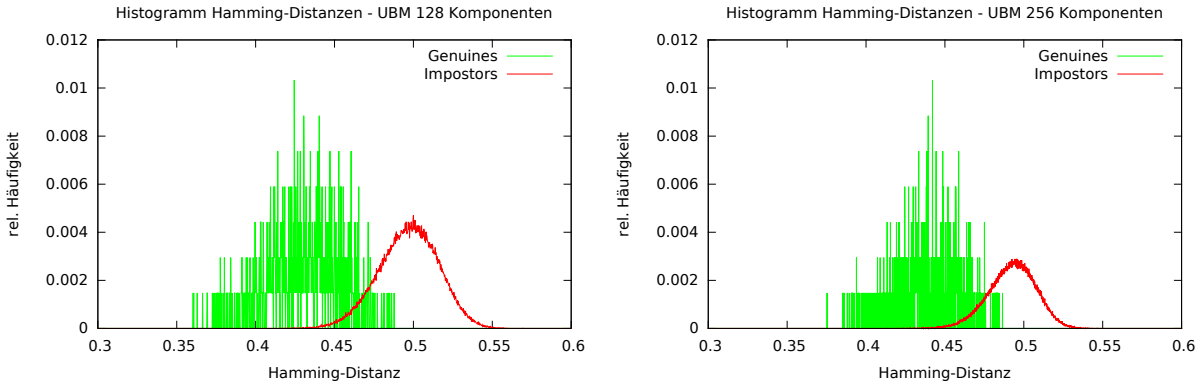

Abbildung 5.3: Histogramme der Hamming-Distanzen zwischen b und b^* ohne Verwendung von $RP^{(u)}$

In den Histogrammen ist zu erkennen, dass auch in diesem System keine perfekte Trennung zwischen Genuine und Impostor erreicht wurde (vgl. Kapitel 2.5). Genuines erreichen im Durchschnitt eine Hamming-Distanz der Binärvektoren b (Enrollmentbinärvektor) und

[24] Dies entspricht 4992 Bit für 128 Komponenten und 9982 Bit für 256 Komponente.
[25] Ein Histogramm ist die graphische Darstellung einer gemessenen Häufigkeitsverteilung.

b^* (Verifikationsbinärvektor) von 43.02% für ein UBM mit 128 Komponenten und 43.94% für ein UBM mit 256 Komponenten. Für Impostor-Verifikationsversuche liegt die durchschnittliche Hamming-Distanz bei 49.67% für 128 Komponenten und 49.24% für 256 Komponenten.

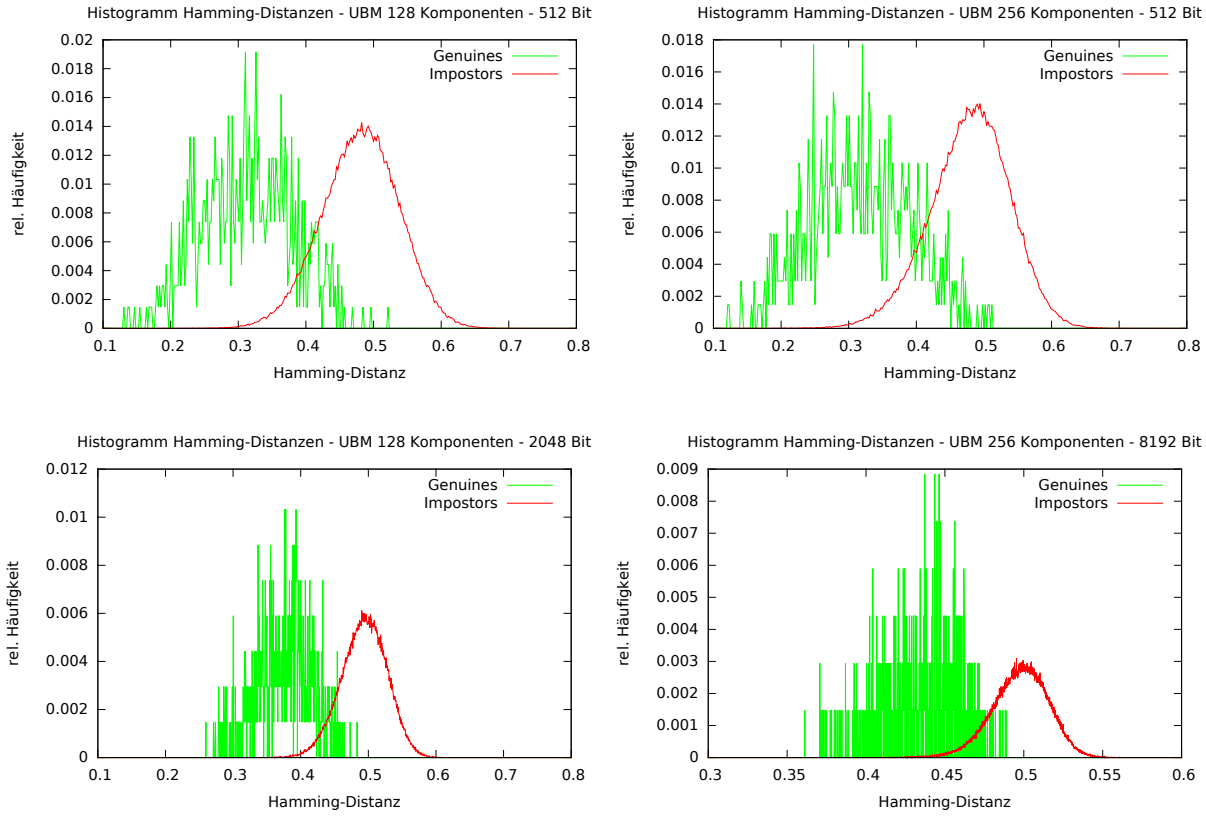

Abbildung 5.4: Histogramme der Hamming-Distanzen zwischen b und b^* unter Verwendung von $RP^{(u)}$

In der Abbildung 5.4 sind Histogramme mit der Verwendung von Relevant Projections $RP^{(u)}$ zu sehen. Die Tabelle 5.1 fasst die durchschnittlichen Hamming-Distanzen einer Systemkonfiguration sowie Fehlerraten zusammen. In Abbildung 5.5 sind die Fehlerraten über die DET-Diagramme für die beiden UBMs zu sehen. Insgesamt zeigt sich, dass durch den Einsatz von $RP^{(u)}$ die durchschnittliche Hamming-Distanz für Genuines stark verringert werden kann, während die Impostor-Verteilung sich nur minimal verändert. Diese Optimierung ist für die spätere Verwendung von PI sowie eines fehlerkorrigierenden Codes sehr förderlich. Bei der Verwendung von 512 Bits verringert sich die durchschnittliche Genuine Hamming-Distanz um 12%-Punkte, während die Impostor-Verteilung sich

UBM	n	∅ HD Genuines	∅ HD Impostors	τ	FNMR	FMR
128	512	0.3131	0.4804	0.408	0.0973	0.1010
128	1024	0.3376	0.4878	0.418	0.0707	0.0698
128	**2048**	**0.3747**	**0.4933**	**0.438**	**0.0516**	**0.0568**
128	4096	0.4197	0.4978	0.462	0.0530	0.0581
128	4992	0.4302	0.4967	0.466	0.0575	0.0578
256	512	0.3167	0.4788	0.41	0.1194	0.1190
256	1024	0.3386	0.4886	0.422	0.0840	0.0828
256	2048	0.3648	0.4931	0.435	0.0663	0.0631
256	4096	0.3966	0.4951	0.451	0.0589	0.0618
256	**8192**	**0.4344**	**0.4976**	**0.469**	**0.0530**	**0.0590**
256	9982	0.4394	0.4924	0.468	0.0678	0.0587

Tabelle 5.1: Auswirkung von $RP^{(u)}$ auf die durchschnittliche Hamming-Distanz zwischen b und b^*, Verifikationsleistung anhand Schwellwert τ

nur um 1.6%-Punkte verschlechtert. Der Überlappungsbereich zwischen der Genuine- und Impostor-Verteilung hat sich gleichzeitig vergrößert, wodurch sich eine Verifikationsleistung bei Wahl eines Schwellwerts $\tau = 0.408$ mit einer ungefähren EER von 10% ergibt. Zu beachten ist, dass eine EER nicht immer durch eine spezifische Konfiguration erreichbar ist. Den größten Einfluss hierauf hat der Schwellwert τ, da dieser die Entscheidung „akzeptiert oder abgewiesen“ beeinflusst und somit direkten Einfluss auf die Fehlerraten hat. Für die Tabelle 5.1 wurden Schwellwerte verwendet, welche annähernd eine EER ergeben.

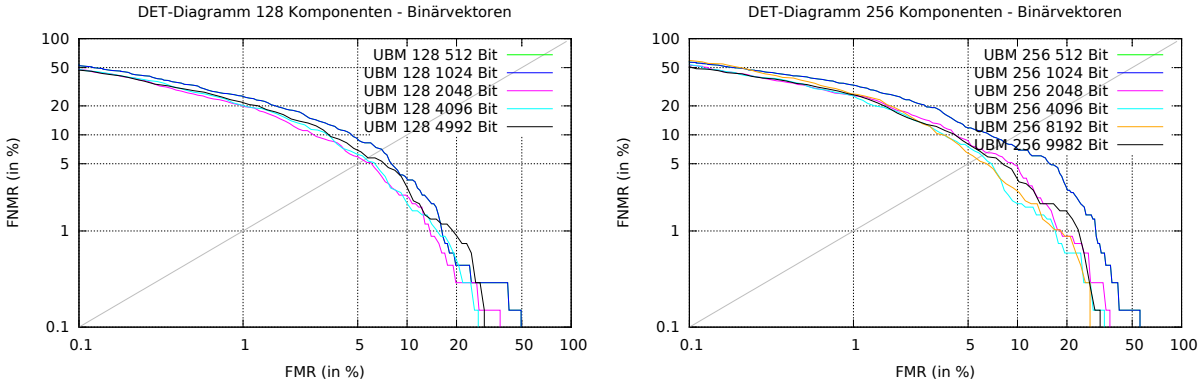

Abbildung 5.5: DET-Diagramme bei Verwendung von Binärvektoren und Hamming-Distanz als Score

Insgesamt konnte für zwei unterschiedliche UBM Größen eine EER von ~5.7% in einem Testdurchlauf ermittelt werden. Das System von Rúa et al. [ARMACC12, Table II] konnte für Online-Signaturen in der unprotected Verifikation nur eine EER von 8.35% erreichen. Hierdurch zeigt sich, dass die Idee, diesen Algorithmus auf Sprachdaten anzuwenden, haltbar ist und es sich dazu eignet in ein Template Protection System für Sprecherverifikation überführt zu werden.

5.4 Verifikation mit Template Protection

Nachdem gezeigt werden konnte, dass die generierten Binärvektoren eine Trennung von Genuine und Impostor ermöglichen, kann das System um die Verwendung von PI erweitert werden. Hierzu wurde das Enrollment um die Schritte der Generierung und Codierung eines Schlüssels sowie dem Fuzzy Commitment Scheme erweitert. In der Verifikation wird anschließend das Codewort extrahiert und eine Nachricht decodiert. Die Verifikationsentscheidung wird über einen Vergleich des gespeicherten Hashwertes mit dem Hashwert der decodierten Nachricht durchgeführt. Damit die beiden berechneten Hashwerte übereinstimmen, muss der Hashwert des ursprünglichen Schlüssels mit dem Hashwert der decodierten Nachricht identisch sein.

5.4.1 Decodierung des extrahierten Codeworts

Nachdem bei der Verifikation der Binärvektor b^* generiert wurde, wird aus der gespeicherten AD, der Differenzvektor δ, das Codewort extrahiert:

$$\begin{aligned} \textit{Enrollment: } & b \oplus c = \delta \\ \textit{Verifikation: } & b^* \oplus \delta = c^* \\ \textit{zusammengefasst: } & b^* \oplus b \oplus c = c^* \end{aligned} \tag{5.3}$$

Durch die Unterschiedlichkeit von b und b^* unterscheidet sich auch das extrahierte Codewort c^* vom ursprünglichen Codewort c. Der Unterschied zwischen c und c^* entspricht dabei der Hamming-Distanz der Binärvektoren. Es gilt:

$$\begin{aligned} (b^* \oplus b) \oplus c &= c \ , \ \textit{für } b \oplus b^* = 0 \\ (b^* \oplus b) \oplus c &\neq c^* \ , \ \textit{für } b \oplus b^* \neq 0 \end{aligned} \tag{5.4}$$

Da, wie in der bisherigen Auswertung zu erkennen, $b \oplus b^* \neq 0$, müssen diese Fehler korrigiert werden. Hierzu wird c^* vom fehlerkorrigierenden Code im Decodierungsprozess in zwei Phasen verarbeitet. Zuerst wird durch Maximum-Likelihood-Decodierung c^* einem existierenden Codewort zugeordnet. Hierzu berechnet die Maximum-Likelihood-Decodierung die Hamming-Distanz zwischen c^* und jedem Codewort der Hadamard-Matrix Γ. Anschließend wird das Codewort mit der geringsten Hamming-Distanz zu c^* ausgewählt. Aus dem so ausgewählten Codewort wird anschließend die darin enthaltene Nachricht r^* decodiert:

$$\begin{aligned} ML(c^*, \Gamma) &= \textit{arg min } HD(c^*, c \in \Gamma) \\ \textit{decode}(c^*) &= r^* \end{aligned} \tag{5.5}$$

Der eingesetzte Hadamard Code kann bis zu $t = 2^{n-2} - 1$ Fehler korrigieren. Treten mehr als t Fehler auf, so wählt die Maximum-Likelihood-Decodierung auch das Codewort mit geringster Hamming-Distanz aus. Hierbei kann jedoch keine garantiert korrekte Decodierung durchgeführt werden. Es ist daher zu diesem Zeitpunkt nicht garantiert, dass das ausgewählte Codewort und die daraus decodierte Nachricht r^* die beim Enrollment generierte Nachricht r ist.

In einem häufigen Anwendungsbereich fehlerkorrigierender Codes ist die Unterscheidung zwischen korrigierbaren und erkennbaren Fehlern wichtig. Als Beispiel sei der Empfang von Daten über einen Funkkanal genannt. Das entfernte Empfangsgerät sei darauf angewiesen, dass nur die korrekten Nachrichten im System zu einer korrekten Funktionsweise führen. Hierfür muss das Empfangsgerät daher sicherstellen, dass es bei Empfang eines Codeworts die Fehler korrigieren und eine korrekte Nachricht decodieren kann. Ansonsten muss es erkennen, dass mehr Fehler aufgetreten sind als korrigierbar und die empfangenen Daten nicht verwendet werden dürfen, da das Gerät nicht sicherstellen kann, ob es die korrekte Nachricht decodiert hat.

Weil beim Enrollment aus dem Schlüssel r der Hash h berechnet und als PI gespeichert wurde, weiß dieses System jedoch welche Nachricht decodiert werden muss. Durch den Vergleich des gespeicherten Hash $h = H(r^*)$ kann das System grundsätzlich ein Codewort auswählen und die darin enthaltene Nachricht decodieren. Die Verifikationsentscheidung wird nachgelagert im PIC-Modul durch Vergleich der beiden Hashwerte getroffen.

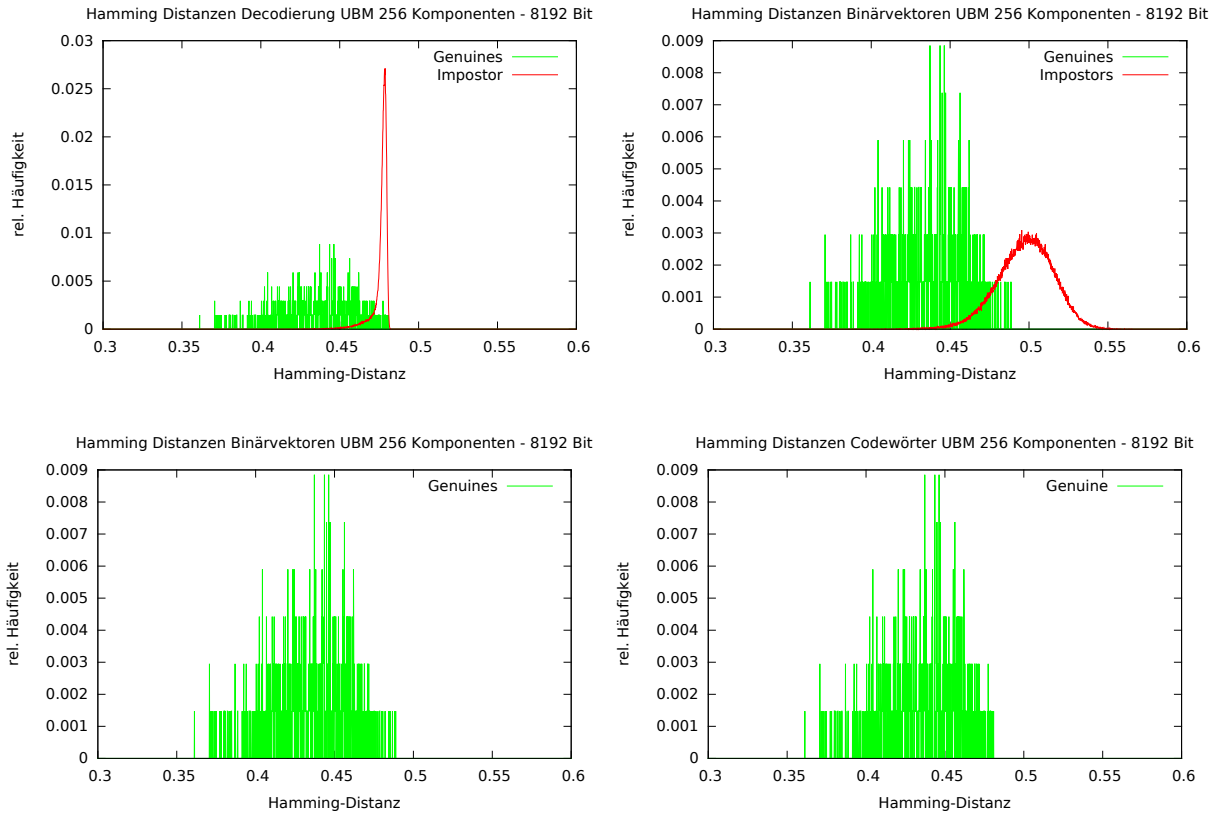

Abbildung 5.6: Histogramme der Hamming-Distanzen zwischen dem extrahierten und dem zurückgegeben Codewort der Maximum-Likelihood-Decodierung für Genuines und Impostors (oben), nur Genuines (unten)

In Abbildung 5.6 sind die Histogramme der Hamming-Distanzen zwischen dem extrahierten Codewort c^* und dem zurückgegebenen Codewort $c \in \Gamma$ mit minimaler Distanz, unter Verwendung des UBM mit 256 Komponenten und Binärvektoren der Länge 8192 Bit, angegeben. In den unteren beiden Diagrammen wurden ausschließlich die Genuines betrachtet. Das linke Diagramm zeigt die Verteilung der Hamming-Distanzen der Binärvektoren und das rechte Diagramm die Verteilung der Hamming-Distanzen zurückgegebener Codewör-

ter. In den oberen beiden Diagrammen wurden für dieselbe Konfiguration Genuine und Impostor eingetragen.

Es ist zu beobachten, dass die Verteilung der Hamming-Distanzen der Codewörter bei Genuines nahezu identisch mit der Verteilung von Hamming-Distanzen der generierten Binärvektoren ist. Auffällig ist, dass sich der Unterschied nur im Bereich höherer Hamming-Distanzen befindet. Codewörter, die durch die Unterschiedlichkeit von b und b^* eine hohe Hamming-Distanz aufweisen, können durch die Minimaldistanz des Hadamard Codes eine maximale Hamming-Distanz von 50% zu einem anderen Codewort aufweisen. Für diesen Fall würde es zwei Codewörter geben, zu denen das extrahierte Codewort die minimalste Distanz aufweist. Dies wurde in der Implementierung aufgelöst, indem das Codewort, welches zuletzt im Ablauf geprüft und dieselbe Distanz aufweist, als das zurückzugebende ausgewählt wurde. Motiviert ist diese Entscheidung durch das seltene Auftreten solcher Hamming-Distanzen für Genuines. Für 50% + 1 Stelle Unterschied existiert durch die Minimaldistanz wieder ein Codewort, welches eindeutig eine minimalste Distanz aufweißt. Somit findet sich für die Impostor Verteilung, durch die hohe Anzahl an Distanzen die zwischen den Binärvektoren mehr als 50% betragen, eine Häufung im Bereich 46% bis 49%. Für einen kleinen Teil an Genuine-Versuchen ist auch bei Distanzen nahe an 50% zu beobachten, dass diese bei der Maximum-Likelihood-Decodierung für ein Codewort eine geringere Distanz als die ursprüngliche Hamming-Distanz zwischen den Binärvektoren aufweisen. In beiden Fällen, für Genuine als auch Impostor, ist jedoch durch diese Zuweisung noch nicht sichergestellt, dass für Genuines das gesuchte Codewort und für Impostor ein anderes Codewort ausgewählt wird.

Durch den Vergleich des gespeicherten Hashwertes als PI und dem bei der Decodierung generierten PI^* wird der Benutzer anschließend akzeptiert oder abgewiesen. Diese Entscheidung führt zu einer messbaren algorithmischen Fehlerleistung. In Tabelle 5.2 sind für die Konfigurationen, wie in Tabelle 5.1, die erreichten Fehlerraten FNMR und FMR eingetragen. Das System wurde dabei so konfiguriert, dass ein Binärvektor mit genau einem Codewort verarbeitet wird. Die daraus resultierende Länge k des Schlüssels (vgl. Kapitel 4.2.4) ist ebenso in der Tabelle in Bits angegeben.

Die Fehlerraten für das Template Protection System aus Tabelle 5.2 zeigen, dass das konstruierte System für die Sprecherverifikation funktioniert. Das durch die Maximum-Likelihood-Decodierung ausgewählte Codewort entspricht bei Genuines zu einem großen Anteil dem durch das Enrollment vorgegebene und gesuchte Codewort. Beispielsweise kann für das UBM mit 256 Komponenten mit einer Binärvektorlänge von 8192 Bit eine FNMR

UBM	n	k	FNMR	FMR
128	512	10	0.0252	0.3032
128	1024	11	0.0073	0.2515
128	2048	12	0.0044	0.2039
128	**4096**	**13**	**0.0162**	**0.1319**
256	512	10	0.0766	0.1767
256	1024	11	0.0398	0.1730
256	2048	12	0.0103	0.1775
256	**4096**	**13**	**0.0103**	**0.1769**
256	8192	14	0.0191	0.1323

Tabelle 5.2: Verifikationsleistung des Template Protection Systems

von 1.9% bei einer FMR von 13.2% erzielt werden. Bei dem UBM mit 128 Komponenten und 4096 Bit konnte eine FNMR von 1.6% bei einer FMR von 13.1% erreicht werden. Diese beiden Konfigurationen sind sich von den Fehlerraten sehr ähnlich. Die erzielte Verifikationsleistung der FNMR ist hierbei bereits als besonders positiv hervorzuheben.

5.4.2 Optimierungsansatz im Decodierungsprozess

Die Histogramme in Abb. 5.6 zeigen eine Überlagerung von Genuine und Impostor. Der Überlappungsbereich ist ähnlich zu denen aus den Verteilungen in Abbildung 5.4. Es konnte bereits gezeigt werden, dass die zurückgegebenen Codewörter der Maximum-Likelihood-Decodierung die gesuchten Codewörter sind. Anhand dieser Erkenntnis wird folgende Idee zur Optimierung der Fehlerraten nachfolgend untersucht:

Analog zur Verwendung eines Schwellwertes τ bei der unprotected Verifikation durch Hamming-Distanzen der Binärvektoren wird in den Decodierungsprozess der Maximum-Likelihood-Decodierung eingegriffen. Ist die Hamming-Distanz des durch die Maximum-Likelihood-Decodierung ausgewählten Codeworts größer als der konfigurierte Schwellwert τ, wird über einen Zufallsprozess ein anderes Codewort der Hadamard Matrix Γ zurückgegeben, jedoch nicht das Codewort, welches vorab ausgewählt wurde.

Diese Optimierungsidee führt zu einer Verbesserung der biometrischen Performanz. Es lässt sich durch diesen Eingriff direkt Einfluss auf die Fehlerraten nehmen, wodurch das System an spezifische Anforderungen eines Kunden angepasst werden kann, während gleichzeitig die Template Protection gewährleistet ist. In Tabelle 5.3 werden analog zur Tabelle 5.1 Fehlerraten nahe an einer EER angegeben.

UBM	n	k	τ	**FNMR**	**FMR**
128	512	10	0.38	0.1091	0.1052
128	1024	11	0.406	0.0722	0.0700
128	**2048**	**12**	**0.432**	**0.0560**	**0.0582**
128	4096	13	0.458	0.0634	0.0596
256	512	10	0.402	0.1327	0.1430
256	1024	11	0.419	0.0929	0.9401
256	2048	12	0.433	0.0707	0.0724
256	**4096**	**13**	**0.449**	**0.0678**	**0.0683**
256	8192	14	0.467	0.0663	0.0632

Tabelle 5.3: Verifikationsleistung des Template Protection Systems bei Eingriff in den Decodierungsprozess mit einem Schwellwert

Für das UBM mit 128 Komponenten konnte eine EER von ~5.7% und für das UBM mit 256 Komponenten eine EER von ~6.8% erreicht werden. Dies entspricht einer minimalen Verschlechterung von 0.2%-Punkte (UBM 128) bzw. 1.3%-Punkte (UBM 256) gegenüber der unprotected Verifikation. Im Vergleich zur Template Protection ohne Optimierung verbessert sich beim UBM 128 mit 2048 Bit die FMR um 14.5%-Punkte und beim UBM 256 wird eine Verbesserung der FMR von 10.9%-Punkten erreicht.

Einen Verlauf der Fehlerraten für das optimierte Template Protection System ist in den DET-Diagrammen unter Abbildung 5.7 zu sehen.

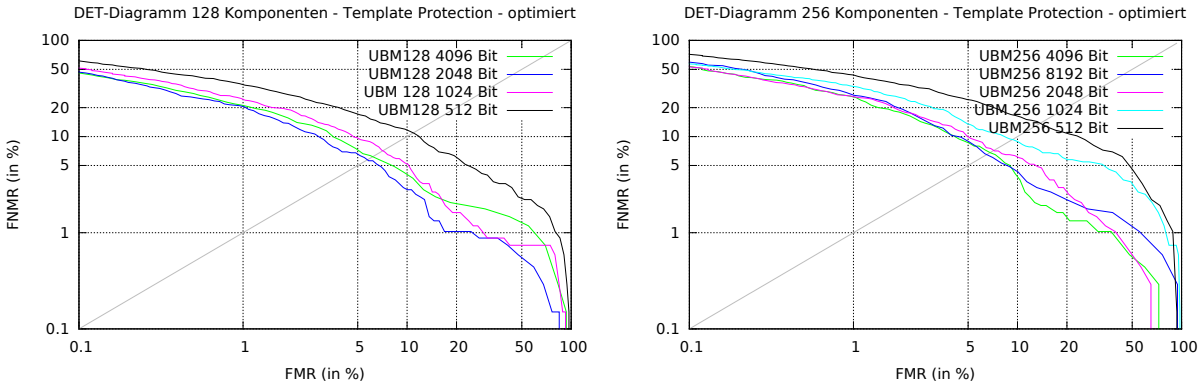

Abbildung 5.7: DET-Diagramme für das optimierte Template Protection System

5.5 Anforderungen an das biometrische Template

Nachdem gezeigt wurde, dass das konstruierte System eine gute biometrische Performanz erzielt, werden im Folgenden die Anforderungen des ISO-Standard ISO 24745 (vgl. Kapitel3, [ISO11]) an das biometrische Template beurteilt. Ziele, die nicht durch Vorschläge des Standards abgedeckt werden sind die Unumkehrbarkeit, die Erneuerbarkeit und Widerrufbarkeit, sowie die Unverkettbarkeit.

5.5.1 Unverkettbarkeit

Die Unverkettbarkeit des Templates über verschiedene Systeme hinweg kann nicht ausschließlich durch den Algorithmus erreicht werden. Insbesondere der Systembetreiber muss sicherstellen, dass jede Applikation einen eigenen Enrollmentprozess besitzt und gleichzeitig nicht auf Templates einer anderen Applikation zugegriffen wird. Wird beispielsweise ein Enrollment für mehrere Applikationen mit denselben Trainingsdaten durchgeführt, so sind der PI und das dazugehörige δ durch den Zufallswert bereits für denselben Benutzer unterschiedlich. Die RP wäre hierbei jedoch für beide Systeme identisch.

Um die Unverkettbarkeit von RP zu erreichen, kann für jede Applikation eine zufällige Permutation eingeführt werden. Nachdem der Binärvektor generiert wird, werden die Positionen permutiert und erst anschließend die RP berechnet. Somit unterscheiden sich RP bei verschiedenen Applikationen unter Verwendung derselben Enrollmentdaten, sodass keine Einzelteile des Template verkettbar sind.

5.5.2 Unumkehrbarkeit

In der Unumkehrbarkeit gibt es zwei Betrachtungsweisen. Auf der einen Seite steht der Schutz des Templates im System auf der anderen Seite gilt es, den Schutz der biometrischen Daten sicherzustellen. Die Eigenschaft zur Unumkehrbarkeit des Templates in biometrische Daten – hier MFCC – ist wichtig, da bereits Ansätze zur Umwandlung von MFCC in Sprachdaten existieren ([Ell05]). Zum jetzigen Zeitpunkt gibt es für keines der Template-Teile eine Umkehrfunktion, um aus diesen biometrische Daten zu errechnen.

Der PI besteht aus einem Hashwert für einen Zufallswert, welcher unabhängig von den biometrischen Daten ist. Da keinerlei Abhängigkeiten von den biometrischen Daten zum Zufallswert existieren, kann aus diesem keine Information zu den MFCC gewonnen werden. Bei Einsatz kryptographisch starker Hashfunktionen kann das Template auch nicht zur ursprünglichen Nachricht zurückgerechnet werden.

Die RP entsprechen einer Maske zur Auswahl relevanter Bitstellen aus dem generierten Binärvektor. Durch sie lässt sich lediglich interpretieren, dass das beim Enrollment berechnete Feature an dieser Position sich von der Population $\overline{\mu}$ unterscheidet. Darüber hinaus können vor allem durch Einsatz von Permutationen keine direkten Zuordnungen zu einem spezifischen MFCC-Feature gemacht werden.

Das δ entspricht einer XOR-Verknüpfung des Binärvektors mit dem Codewort des fehlerkorrigierenden Codes. Generell lässt sich aus einem Binärvektor keine Umkehrung zu einem konkreten Feature-Vektor machen, da die Binarisierung einen großen Informationsverlust darstellt. Diese Verknüpfung dient vor allem dem Schutz des Templates, da sich aus dem Codewort der Schlüssel extrahieren lässt. Die XOR-Operation ist jedoch nicht invertierbar. Die Sicherheit des Differenzvektors δ hängt dabei von der Entropie des Binärvektors ab.[26] Sie muss groß genug sein, sodass ein Angreifer den Binärvektor nicht erraten kann.

Darüber hinaus führen unterschiedliche Schlüssel bei gleichen Enrollmentdaten zu unterschiedlichen Hashwerten und unterschiedlichem δ, da hierbei lediglich der Binärvektor identisch bleiben würde. Für verschiedene Benutzer mit demselben Schlüssel ändert sich der Differenzvektor δ, da sich die Binärvektoren unterscheiden. Der Hashwert wäre für alle Benutzer jedoch identisch. Dies lässt sich durch das Einbringen eines Salt in die Hashfunktion lösen, sodass auch hier unterschiedliche Hashwerte für die Benutzer existieren. Dies soll jedoch nur die Unumkehrbarkeit der einzelnen Bestandteile verdeutlichen, da das Verwenden desselben Schlüssels offensichtlich nicht sinnvoll ist.

5.5.3 Erneuerbarkeit und Widerrufbarkeit

Die Erneuerbarkeit wird erreicht, indem der Benutzer eine Schnittstelle zum Widerrufen seines Templates erhält. Beim Widerrufen wird das Template vollständig und unwiderruflich gelöscht.

Auf Wunsch des Benutzers kann anschließend der Prozess des Enrollments erneut durchlaufen werden, um das Template zu erneuern. Durch den Zufallsprozess zur Generierung der PI ändert sich der im System hinterlegte Hashwert. Bei der Codierung des Zufallswertes mit dem fehlerkorrigierenden Code ändert sich nach der XOR-Verknüpfung des Binärvektors mit dem Codewort auch die abzuspeichernde AD.

[26] Eine Beurteilung zu diesem Punkt ist unter Kapitel 5.5.4 zu finden.

5.5.4 Sicherheitsniveau des Pseudonymous Identifier

Neben den Anforderungen des ISO-Standard an das Template ist die erzielte Schlüssellänge[27] des PI wichtig. Aktuell werden Schlüssellängen von 128 Bit und mehr als Mindestanforderung angegeben (vgl. [Bun13], [Eck09, S.290f]). Betrachtet man die Tabellen 5.3 und 5.2, so entspricht die Schlüssellänge für die beste biometrischen Performanz für das UBM 128 12 Bit und beim UBM 256 13 Bit. Unter Verwendung des längsten Binärvektors entspricht die Schlüssellänge 13 Bit beim UBM 128 und 14 Bit beim UBM 256. Dies liegt weit unterhalb der als Mindestanforderung angegebenen 128 Bit. Auch wenn der Schlüssel über eine Hashfunktion in einen wesentlich längeren Bitstring (z.B. generiert SHA-256 eine 256 Bit lange Ausgabe [Eck09, S.364]) überführt werden kann, ist der zu untersuchende Schlüsselraum für einen Angreifer sehr klein. Er muss lediglich 2^{12} bzw. 2^{13} Kombinationen generieren und in die Hashfunktion eingeben, um alle möglichen Kombinationen, die in diesem System vorkommen können, zu konstruieren. Aus diesem Grund ist es nicht möglich, den PI als Eingabeschlüssel für ein Verschlüsselungsverfahren einzusetzen.

Um einen 128 Bit langen Schlüssel mit dem Hadamard-Code zu codieren, müsste der Hadamard-Code die Form $[2^{127}, 128, 2^{126}]$ haben. Es ist offensichtlich, dass der Hadamard-Code somit nicht praktisch einsetzbar ist, um diese Anforderung zu erfüllen.

Darüber hinaus gilt es jedoch auch zu überprüfen, ob das System im Allgemeinen einen hinreichenden Schutz eines 128 Bit langen Schlüssels bietet. Wie bereits als Bedingung erwähnt, muss die Entropie des generierten Binärvektors größer als die Schlüssellänge sein. Eine Schätzung für die durchschnittliche Entropie für die aus biometrischen Feature-Vektoren generierten Binärvektoren liefert die Berechnung der Degrees-of-Freedom ([Dau])

$$dof = \frac{p(1-p)}{\sigma^2} \tag{5.6}$$

wobei p die durchschnittliche Hamming-Distanz zwischen b und b^*, sowie σ^2 die Varianz dieser Hamming-Distanzen bei Impostor-Verifikationsversuchen ist. Das Ergebnis ist die Anzahl unabhängiger Bits im Binärvektor ([RBBB13, S.6]). Eine weitere Interpretation hierfür ist: Ein Binärvektor der Länge 2048 Bit und 219 Degrees-of-Freedom entspricht demselben Ergebnis wie 219 unabhängige Münzwürfe hintereinander.

[27] Die Schlüssellänge beschreibt die Größe des Schlüsselraums.

UBM	n	∅ HD Impostors	DoF
128	512	0.4804	77.73
128	1024	0.4878	122.40
128	2048	0.4933	219.84
128	4096	0.4978	510.71
128	4992	0.4967	695.61
256	512	0.4788	72.62
256	1024	0.4886	115.05
256	2048	0.4931	184.33
256	4096	0.4951	325.84
256	8192	0.4976	787.89
256	9982	0.4924	1102.63

Tabelle 5.4: Degrees-of-Freedom für die generierten Binärvektoren unter Verwendung von *RP*

In Tabelle 5.4 sind für das UBM 128 und das UBM 256 die berechneten Degrees-of-Freedom eingetragen. Hierbei kann man erkennen, dass bei beiden UBM Varianten ab einer Binärvektorlänge von 2048 Bit die Degrees-of-Freedom größer als 128 Bit sind. Somit kann das System mit diesen Konfigurationen grundsätzlich den Schutz für Schlüssellängen von 128 Bit sicherstellen.

5.5.5 Ansätze zur Vergrößerung der Schlüssellänge

Um die Schlüssellänge des Templates zu erweitern, wäre es möglich, nicht nur ein Codewort zu generieren, sondern mehrere Codewörter. Hierfür könnte der Schlüssel auf Bitebene aufgesplittet werden, sodass mehrere Codewörter für kleinere k generiert werden, wodurch aber die Gesamtlänge des Schlüssels größer wird.

Beispiel: Der Binärvektor besitzt die Länge $2^{11} = 2048$ Bit. Durch das Aufteilen in zwei Codewörter kann mit dem Hadamard Code ein Schlüssel der Gesamtlänge $k = 22$ Bit verarbeitet werden, da jedes der zwei Codewörter die Länge $2^n = 2^{10} = 1024$ besitzt und somit eine Nachricht der Länge $k = n + 1 = 11$ Bit codiert. Bei der Verwendung von nur einem Codewort entspricht die Gesamtlänge $k = 12$ Bit, da ein Codewort der Länge $2^{n-1} = 2^{11} = 2048$ Nachrichten der Länge $k = 12$ Bit codiert.

UBM	n	**Codewörter**	k_Σ	**FNMR**	**FMR**	**Anmerkung**
128	2048	2	$2 * 11 = 22$	0.0855	0.0594	
128	**2048**	**2**	$2 * 11 = 22$	**0.0471**	**0.0773**	*permutiert*
128	2048	4	$4 * 10 = 40$	0.3082	0.0072	
128	2048	4	$4 * 10 = 40$	0.2477	0.0103	*permutiert*
128	2048	8	$8 * 9 = 72$	0.7418	0.0002	
128	2048	8	$8 * 9 = 72$	0.6622	0.0004	*permutiert*
256	8192	2	$2 * 13 = 26$	0.1224	0.0416	
256	**8192**	**2**	$\mathbf{2 * 13 = 26}$	**0.0855**	**0.0455**	*permutiert*
256	8192	4	$4 * 12 = 48$	0.3702	0.0006	
256	8192	4	$4 * 12 = 48$	0.3244	0.0081	*permutiert*
256	8192	8	$8 * 11 = 88$	0.7640	0.0003	
256	8192	8	$8 * 11 = 88$	0.7197	0.0004	*permutiert*
256	8192	16	$16 * 10 = 160$	0.9764	0.0000	
256	8192	16	$16 * 10 = 160$	0.9705	0.0000	*permutiert*

Tabelle 5.5: Fehlerraten des Template Protection Systems für 2048 Bit und 8192 Bit und Aufsplitten in mehrere Codewörter

Dies bedeutet aber, dass jedes der Codewörter im Decodierungsprozess erfolgreich decodiert werden muss. Würde eines der Codewörter nicht korrekt decodiert, entspricht der Hashwert des zusammengesetzten Schlüssels r^* nicht dem Hashwert des beim Enrollment generierten Schlüssels. Hierfür müssen die fehlerhaften Bitstellen über die gesamte Länge im Optimalfall gleichverteilt sein. Zumindest dürfen in keinem Anteil, den ein Codewort umfasst, so viele Fehler auftreteten, dass die Decodierung nicht erfolgreich ist. Tabelle 5.5 zeigt die erreichten Fehlerraten für 2048 Bit und 8192 Bit Binärvektoren. Darüber hinaus wurden in einem ersten Ansatz die Bits permutiert. Dabei wurde die zweite Bitstelle mit der letzten Bitstelle, die vierte mit der drittletzten usw. vertauscht. Dies soll zu einer Verteilung möglicher Fehler über den Binärvektor führen und ist in der Tabelle mit dem Zusatz „permutiert“ angegeben.

Die Ergebnisse zeigen, dass das Permutieren der Bitstellen tatsächlich dazu führt, dass auftretende Fehler sich gleichmäßiger auf den Binärvektor verteilen. Mit steigender Anzahl an Codewörtern verschlechtert sich die FNMR sehr schnell.

Grundsätzlich ist hieraus zu erkennen, dass die Fehlerverteilung über den Binärvektor die Möglichkeiten der Schlüssellänge beschränken. Können die auftretenden Fehler reduziert werden, kann das System auch theoretisch mit dem Hadamard-Code Schlüssel von 128 Bit generieren. Eine mögliche Kombination wäre hierbei bereits mit dem UBM 128 bei 2048 Bit und 16 Codewörtern erreicht. Für das UBM 256 kann bei 8192 Bit mit 16 Codewörtern bereits ein Schlüssel von 160 Bit erreicht werden. Praktisch sind diese Schlüssel in der momentanen Konfiguration nicht einsetzbar, da die erreichte biometrische Performanz das System unbrauchbar macht.

6 Zusammenfassung

In dieser Arbeit wurde ein Template Protection System für biometrische Sprecherverifikation konstruiert, welches die Anforderungen des ISO Standards 24745 erfüllt. Es wurden die Anforderungen des ISO Standards bezüglich Sicherheit und Datenschutz vorgestellt und anschließend basierend auf einem Template Protection Algorithmus für Online-Signatur eine Anpassung und Implementierung für die Sprecherverifikation durchgeführt.

Hierzu wurde basierend auf einem UBM und GMM ein erneuerbares biometrisches Template generiert, welches aus einem PI und zwei AD besteht. Im ersten Schritt wird während des Enrollments eine binäre Repräsentation des GMM generiert. Mit einem Zufallszahlengenerator wird anschließend ein Schlüssel generiert, der mit einer Hashfunktion geschützt wird. Der Hashwert bildet die PI des Systems. Der Schlüssel wird von einem fehlerkorrigierenden Code codiert und mit dem Binärvektor über die XOR-Verknüpfung verschleiert. Dies bildet eine AD im System.

Bei der Verifikation wird für eine Sprachaufnahme ein Binärvektor generiert, welcher anschließend über die XOR-Verknüpfung das Codewort entschleiert. Der fehlerkorrigierende Code weist dem hieraus entstandenen, durch Varianzen im Binärvektor veränderten Codewort, ein Codewort des Codes über die Maximum-Likelihood-Decodierung zu. Die aus diesem Codewort extrahierte Nachricht wird über eine Hashfunktion in einen Hashwert überführt. Die Verifikationsentscheidung wird zum Schluss mittels eines Vergleichs des Hashwertes aus der Verifikation mit dem gespeicherten Hashwert des Enrollments durchgeführt. Stimmen die Hashwerte überein, gilt der Benutzer als akzeptiert. Zur Optimierung der Stabilität der generierten Binärvektoren und gleichzeitiger Anpassung an einen fehlerkorrigierenden Code wurde eine weitere AD während des Enrollments berechnet. Diese AD entspricht einer Maske zur Auswahl möglichst stabiler Bitstellen im Binärvektor. Das Template besteht somit aus drei Teilen: dem PI (Hashwert h) und zwei AD (Bitmaske RP und Differenzvektor δ).

Das konstruierte System wurde in der Evaluation stufenweise getestet und die erzielten Fehlerraten wurden diskutiert. Weiterhin wurde der Schutz des biometrischen Templa-

tes diskutiert und gezeigt, dass die Anforderungen des Standards an das Template erfüllt werden. Darüber hinaus wurden Ansätze zur Erweiterung der Schlüssellänge und Optimierungen zur Steigerung der biometrischen Performanz untersucht.

6.1 Fazit

Es konnte gezeigt werden, dass die generierten Binärvektoren eine Trennung von Genuines und Impostor ermöglichen. Über einen gewählten Schwellwert konnte hierbei eine EER von ~5.7% für verschiedene Systemkonfigurationen erzielt werden. Im weiteren Verlauf wurde in der Evaluation gezeigt, dass die Erweiterung zu einem Template Protection System unter Verwendung des Hadamard Codes funktioniert. Durch einen Optimierungsansatz aus den Ergebnissen der Maximum-Likelihood-Decodierung konnte eine EER von ~5.7% erzielt werden. Das System hat somit nach der Erweiterung keinen signifikanten Einbruch in der biometrischen Performanz erlitten.

Das generierte Template erfüllt die Anforderungen zum Schutz der biometrischen Daten und ermöglicht gleichzeitig eine Widerrufbarkeit und Erneuerbarkeit der Referenz im System. Gleichzeitig wurde die Unumkehrbarkeit zu biometrischen Daten sichergestellt und eine Möglichkeit zur Unverkettbarkeit des Templates über Applikationen hinweg vorgeschlagen.

Während die erzielte biometrische Performanz und das Erfüllen der Sicherheitsanforderungen des ISO Standards an das Template insgesamt als positiv zu bewerten sind, muss die bisher erreichte Länge des Schlüssels als nicht ausreichend bewertet werden. Es konnten mit dem Hadamard Code und einer Aufteilung des Schlüssels auf zwei Codewörter im besten Falle ein Schlüssel der Länge 26 Bit generiert werden. Zu beachten ist, dass zwar grundsätzlich größere Schlüssel durch das Aufteilen möglich sind, die biometrische Performanz sich jedoch mit zunehmender Aufteilung rapide verschlechtert. Durch das bereits in der Evaluation eingeführte Permutieren konnten die erzielten Fehlerraten beim Aufteilen leicht gesteigert werden. So konnte beispielsweise beim UBM 128 und zwei Codewörtern durch eine Permutation der Bits die FNMR von 8.5% um 3.8%-Punkte auf 4.7% gesenkt werden. Die FMR verschlechterte sich hierbei von 5.9% um 1.8%-Punkte auf 7.7%. Dies lässt vermuten, dass sich die Fehler nicht gleichmäßig auf den Binärvektor verteilen, was jedoch nicht abschließend in dieser Arbeit bewiesen werden konnte.

6.2 Ausblick

Ein wichtiger Aspekt des Systems ist die stabile Generierung von Binärvektoren. Neben der in dieser Arbeit bereits eingebrachten Bitmaske *RP* könnte auch versucht werden, über die Trainingsdaten beim Binarisierungsprozess stabile Bitstellen zu identifizieren. Hierzu könnten alle Trainingsdaten wie bei der Verifikation binarisiert und anschließend die Stabilität der einzelnen Positionen beurteilt und abgespeichert werden. Eine weitere Überlegung ist, die Verbindung der *RP* und anschließend die soeben beschriebene Messung durchzuführen. Stabile Binärvektoren zu generieren, sodass sich Enrollment und Verifikation für Genuines so gering wie möglich unterscheiden, ist für den Decodierungsprozess des fehlerkorrigierenden Codes eine wichtige Eigenschaft. Gleichzeitig muss aber auch die Trennung von Genuines und Impostor erhöht werden. Würde bereits jetzt die Differenz zwischen Genuine und Impostor Binärvektoren mindestens 51% der Bits betragen, würde die FMR in diesem System unter Beibehaltung der FNMR bei 0% liegen.

Ein weiterer Aspekt ist die Aufteilung des Schlüssels auf mehrere Codewörter. Anhand der Erkenntnisse aus dem Decodierungsprozess sollte die Fehlerverteilung der generierten Binärvektoren genauer untersucht werden. Da die Verifikationsentscheidung nur erfolgreich ist, wenn alle Codewörter korrekt decodiert werden und den richtigen Teilschlüssel generieren, könnten in einem Bereich, der ein Codewort umfasst, mehr Bits fehlerhaft sein als sich korrigieren ließen. Würden diese Fehler gleichmäßig auf den Binärvektor verteilt, könnten alle Codewörter das richtige Ergebnis liefern.

Darüber hinaus sollte auch der Einsatz anderer fehlerkorrigierender Codes in Betracht gezogen werden. Eine Möglichkeit stellt die Familie des Reed-Solomon-Code (RS-Code) ([BFK+98, S.112]) dar. Mit diesen können die Binärvektoren als Codierung von Symbolen betrachtet werden. So könnten beispielsweise 8 Bit zu einem Symbol zusammengefasst werden. Sind in solch einem Symbol mehrere Bitfehler zu finden, so entspricht dies beim RS-Code dennoch nur einem Fehler. Der Hadamard-Code operiert auf der Bitebene und würde jeden dieser Fehler betrachten. Eine weitere Möglichkeit stellt die Verknüpfung von Hadamard- und RS-Code dar. Hao et al. haben unter [HAD05] eine Verknüpfung für Binärvektoren von Iris-Daten bewiesen. Hierbei konnten Schlüssel von 140 Bit generiert werden, was über der empfohlenen Mindestlänge von 128 Bit liegt. Dazu wurden zuerst einzelne Blöcke des Binärvektors mit dem Reed-Solomon-Code und anschließend mit dem Hadamard-Code codiert. Bei der Decodierung werden somit zuerst Fehler auf Bit-Ebene korrigiert und anschließend mit dem RS-Code versucht, die hierbei potentiell falsch decodierten Codewörter zu korrigieren.

Neben der Optimierung des Systems können die Erkenntnisse aus dieser Arbeit auch außerhalb des Kontextes der Template Protection verwendet werden. Die Erkenntnisse der Binarisierung, sowie die Untersuchung der Ähnlichkeit generierter Binärvektoren für Genuines, ermöglichen eine zweistufige Kombination im Einsatz großer Sprecheridentifikationssysteme. Bei diesen muss für ein präsentiertes Sample ein umfangreicher $1 : n$ -Vergleich mit allen Referenzen im System durchgeführt werden. Beim Einsatz von GMM entspricht dieser Aufwand für sehr große Datenbanken einem sehr großen Rechenaufwand. Der Vergleich von zwei Binärvektoren über die Hamming-Distanz ist hingegen hochperformant bei gleichzeitig guter Erkennungsleistung. Die Kombination dieser beiden Verfahren kann zu einem zweistufigen Sprecheridentifikationssystem ausgebaut werden, bei dem zuerst über den Vergleich von Binärvektoren eine Vorauswahl möglicher Kandidaten im System durchgeführt wird. Die z.B. aus zehn Kandidaten entstandene Vorauswahl wird anschließend mit dem aufwändigeren GMM-Verfahren zur Ermittlung des Platz 1–Kandidaten verarbeitet, wodurch sich die Reaktionszeit des Systems massiv verringert. In einer Testreihe auf einem Server mit Intel Xeon CPU (3.7 GHz) und 8 GB RAM konnte der Vergleich der Hamming-Distanz zwischen einer Probe und einer Referenz in durchschnittlich 0.007 ms durchgeführt werden, während das GMM-Verfahren 42.1 ms für einen Vergleich benötigte. Je größer die Datenbank des Identifikationssystems ist, desto größer sind die Auswirkungen auf die Reaktionszeit des Systems, die durch den vorgesetzten Binärvektorenvergleich massiv reduziert werden kann.

Zusammengefasst ist es als sinnvoll zu betrachten, weitere Investitionen in das entwickelte Template Protection System zu tätigen. Die in der Forschungsgemeinschaft entstehenden Ideen und Ansätze werden auch in der Zukunft für das Gebiet der Template Protection neue Impulse bei verschiedenen biometrischen Charakteristiken liefern. Das in dieser Arbeit entwickelte System bietet bereits eine gute Grundlage, die durch weitere Forschung hinsichtlich der biometrischen Performanz und Schlüssellänge ausbaufähig ist.

Abkürzungsverzeichnis

A/D-Wandlung Analog-Digital-Wandlung

AD Auxiliary Data

AES Advanced Encryption Standard

DET Detection Error Tradeoff

DET Detection-Error-Tradeoff

EER Equal Error Rate

FAR False Acceptance Rate

FFT Fast-Fourier-Transformation

FMR False Match Rate

FNMR False Non-Match Rate

FRR False Rejection Rate

GMM Gaussian Mixture Model

HMM Hidden Markov Model

MAC Message Authentication Code

MAP Maximum A Posteriori

MFCC Mel-Frequency Cepstral Coefficients

PCM Pulse-Code-Modulation

PIC Pseudonymous Identifier Comparator

PIE Pseudonymous Identifier Encoder

PII Personally Identifiable Information

PIR Pseudonymous Identifier Recorder

PI Pseudonymous Identifier

RS-Code Reed-Solomon-Code

RSA Rivest, Shamir und Adleman Verfahren

UBM Universal Background Model

UUID Unique Universal Identifier

Literaturverzeichnis

[ARMACC12] Argones-Rùa, Enrique ; Maiorana, Emanuele ; Alba-Castro, Josè L. ; Campisi, Patrizio: Biometric Template Protection Using Universal Background Models: An Application to Online Signature. In: *IEEE Transactions on Information Forensics and Security* (2012), S. 269–282

[BBF+04] Bimbot, Frédéric ; Bonastre, Jean-François ; Fredouille, Corinne ; Gravier, Guillaume ; Magrin-Chagnolleau, Ivan ; Meignier, Sylvain ; Merlin, Teva ; Ortega-García, Javier ; Petrovska-Delacrétaz, Dijana ; Reynolds, Douglas A.: A Tutorial on Text-independent Speaker Verification. In: *EURASIP J. Appl. Signal Process.* 2004 (2004), Januar, 430–451. http://dx.doi.org/10.1155/S1110865704310024. – DOI 10.1155/S1110865704310024. – ISSN 1110–8657

[BBF+06] Betten, Anton ; Braun, Michael ; Fripertinger, Harald ; Kerber, Adalbert ; Kohnert, Axel ; Wassermann, A.: Error-Correcting Linear Codes: Classification by Isometry and Applications. In: *Springer e-books* 18 (2006)

[BBGK08] Breebaart, J. ; Busch, C. ; Grave, J. ; Kindt, E.: A Reference Architecture for Biometric Template Protection based on Pseudo Identities. Special Interest Group on Biometrics and Electronic Signatures, 2008 (2008:25-37). – Forschungsbericht. – BIOSIG 2008

[BFK+98] Betten, Anton ; Fripertinger, Harald ; Kerber, Adalbert ; Wassermann, Alfred ; Zimmermann, Karl-Heinz: *Codierungstheorie: Konstruktion und Anwendung linearer Codes*. Berlin and Heidelberg : Springer Berlin Heidelberg, 1998 http://dx.doi.org/10.1007/978-3-642-58973-7. – ISBN 9783540645023

[bio12] *The International library of ethics, law and technology*. Bd. vol. 11: *Second generation biometrics: The ethical, legal and social context*. Dordrecht : Springer, 2012. – ISBN 9400738927

[Bun] Bundesamt für Sicherheit in der Informationstechnik: Einführung in die technischen Grundlagen der biometrischen Authentisierung. `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Biometrie/Technische_Grundlagen_pdf.pdf?__blob=publicationFile`. – Forschungsbericht. – zuletzt aufgerufen am: 11.02.2014

[Bun13] Bundesamt für Sicherheit in der Informationstechnik: BSI - Technische Richtlinie. Version: Januar 2013. `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102_pdf.pdf?__blob=publicationFile`. 2013. – Forschungsbericht. – zuletzt aufgerufen am: 26.01.2014

[Bus11] Busch, Christoph: *Biometrische Systeme*. Skript: `http://christoph-busch.de/teaching-biometrie.html`, 2011. – zuletzt aufgerufen am: 06.02.2014

[Bus12] Busch, Christoph: *Harmonized Biometric Vocabulary (HBV)*. 2012. – `http://www.christoph-busch.de/standards.html`, zuletzt aufgerufen am: 06.02.2014

[BWS06] Beutelspacher, Albrecht ; Wolfenstetter, Klaus-Dieter ; Schwenk, Jörg: *Moderne Verfahren der Kryptographie: Von RSA zu Zero-Knowledge*. 6., verb. Aufl. Wiesbaden : Vieweg, 2006 (Vieweg Mathematik). – ISBN 9783834800831

[CGU] Claremont Graduate University, Web Interface for Statistics Education (.: *Review of Z Score*. `http://wise.cgu.edu/sdtmod/reviewz.asp`. – zuletzt aufgerufen am: 28.12.2013

[Dam10] Damm, Carsten: *Codierungstheorie*. `http://user.informatik.uni-goettingen.de/~damm/CT/Skript/master.pdf`. Version: 2010. – zuletzt aufgerufen am: 14.01.2013

[Dau] Daugman, J.: The importance of being random: statistical principles of iris recognition. , Februar, 279–291. http://www.ingentaconnect.com/content/els/00313203/2003/00000036/00000002/art00030

[Der10] Der Bundesbeauftragte für den Datenschutz und die Informationsfreiheit: *Bundesdatenschutzgesetz*. http://www.bfdi.bund.de/cae/servlet/contentblob/409518/publicationFile/25234/BDSG.pdf. Version: Juli 2010. – Aktualisierte, nicht amtliche Fassung, zuletzt aufgerufen am: 30.01.2014

[DF] Dr. Fornadi, Ferenc: *Sprechstörungen bei der idiopathischen Parkinson-Krankheit*. http://www.parkinson-web.de/content/was_ist_parkinson/symptome/sprechstoerungen/index_ger.html. – zuletzt aufgerufen am: 16.09.2013

[DPK08] Dr. Pfiste, Beat ; Kaufmann, Tobias: *Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Berlin Heidelberg : Springer Verlag, 2008. – ISBN 9783540759102

[DR02] Daemen, Joan ; Rijmen, Vincent: *The Design of Rijndael: AES - The Advanced Encryption Standard*. Berlin and Heidelberg : Springer, 2002 (Information Security and Cryptography, Texts and Monographs). http://dx.doi.org/10.1007/978-3-662-04722-4. – ISBN 9783642076466

[Eck09] Eckert, Claudia: *IT-Sicherheit*. 6., überarb. und erw. Aufl. München [u.a.] : Oldenbourg Verlag München, 2009. – ISBN 978–3–486–58999–3

[Ell05] Ellis, Daniel P. W.: *PLP and RASTA (and MFCC, and inversion) in Matlab*. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/mfccs.html. Version: 2005. – zuletzt aufgerufen am: 27.01.2014

[HAD05] Hao, F. ; Anderson, R. ; Daugman, J.: Combining Cryptography with Biometrics Effectively. / University of Cambridge. Version: 2005. http://www.cl.cam.ac.uk/~jgd1000/biokeycrypto.html. 2005. – Forschungsbericht

[Heg13] Hegenbart, Steve: *Sprecherverifikation - Implementierung eines Verifikationssystems unter Verwendung der Joint Factor Analysis*, Hochschule Mannheim, Institut für Digitale Signalverarbeitung, Masterthesis, 2013

[Inf13] Informationssicherheit, Europäische A.: *Algorithms, Key Sizes and Parameters Report*. https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report/at_download/fullReport. Version: 2013. – zuletzt aufgerufen am: 27.01.2014

[ISO06] ISO: *Information technology: Biometric performance testing and reporting*. Bd. ISO/IEC 19795-1:2006(E). 1st ed. Genève : ISO/IEC, 2006. – Norme internationale

[ISO11] ISO: Information technology - Security techniques - Biometric information protection / International Organization for Standardization. Geneva, Switzerland : ISO/IEC, 2011 (24745:2011). – ISO. – First edition 2011-06-15

[ISO12] ISO: Information technology – Vocabulary – Part 37: Biometrics / International Organization for Standardization. Geneva, Switzerland : ISO/IEC, 2012 (2382-37:2012). – ISO

[JNN08] Jain, Anil K. ; Nandakumar, Karthik ; Nagar, Abhishek: Biometric Template Security. In: *EURASIP JOURNAL ON ADVANCES IN SIGNAL PROCESSING*, 2008

[JW99] Juels, Ari ; Wattenberg, Martin: A fuzzy commitment scheme. In: *Proceedings of the 6th ACM conference on Computer and communications security*. New York, NY, USA : ACM, 1999 (CCS '99). – ISBN 1–58113–148–8, 28–36

[KJW+] Krause, D Prof. Dr. m. ; Jachau, K. ; Wittig, H. ; Muschke, P. ; Szibor, R.: *DNA-Muster und herkömmlicher Erkennungsdienst. Ein Vergleich*. http://www.springerimages.com/Images/MedicineAndPublicHealth/1-10.1007_s00194-005-0338-y-0. – zuletzt aufgerufen am: 16.09.2013

[KL10] Kinnunen, Tomi ; Li, Haizhou: An overview of text-independent speaker recognition: From features to supervectors. In: *Speech Commun.* 52 (2010), Januar, Nr. 1, 12–40. http://dx.doi.org/10.1016/j.specom.2009.08.009. – DOI 10.1016/j.specom.2009.08.009. – ISSN 0167–6393

[Kna09] Knaut, Andrea: *Die Konstruktion menschlicher Identität durch biometrische Erkennungsverfahren bei Personenkontrollen an Nationalstaatengrenzen*, Humbold-Universität zu Berlin, Institit für Informatik, Diplomarbeit, 2009

[KRB13] Krupp, A. ; Rathgeb, C. ; Busch, C.: Social acceptance of biometric technologies in Germany: A survey. In: *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, 2013, S. 1–5

[Kun11] Kunz, Max: *Kontinuierliche Echtzeitverifikation von Sprechern*, Hochschule Darmstadt, Fachbereich Informatik, Bachelorthesis, 2011

[Lin98] Lin, Dekang: An Information-Theoretic Definition of Similarity. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1998 (ICML '98). – ISBN 1–55860–556–8, 296–304

[MDK+97] Martin, A. ; Doddington, G. ; Kamm, T. ; Ordowski, M. ; Przybocki, M.: The DET curve in assessment of detection task performance. In: *EUROSPEECH'97*, 1997

[Moo06] Moon, Professor Todd K.: *Error Control Coding Course*. `http://ocw.usu.edu/Electrical_and_Computer_Engineering/Error_Control_Coding/index.html`. Version: 2006. – zuletzt aufgerufen am: 03.01.2014

[PR13] Pflug, Anika ; Rathgeb, Christian: *Biometrische Systeme*. Skript: `http://www.dasec.h-da.de/teaching/`, 2013. – Webseite zur Vorlesung nicht mehr verfügbar (Stand: 24.01.2014 10:34 Uhr), Foliensatz auf CD abgespeichert

[PS02] Petermann, Thomas ; Sauter, Arnold: *Biometrische Identifikationssysteme - Sachstandsbericht, Arbeitsbericht Nr. 7*. Februar 2002

[Rab89] Rabiner, Lawrence R.: A tutorial on hidden markov models and selected applications in speech recognition. In: *Proceedings of the IEEE*, 1989, S. 257–286

[RBBB13] Rathgeb, Christian ; Breitinger, Frank ; Busch, Christoph ; Baier, Harald: On the Application of Bloom Filters to Iris Biometrics. In: *IET Biometrics* (2013), Dezember. – to appear

[Rey] Reynolds, Douglas A.: *Gaussian Mixture Models*. http://www.ll.mit.edu/mission/communications/ist/publications/0802_Reynolds_Biometrics-GMM.pdf. – zuletzt besucht am: 05.02.2014

[RQD00] Reynolds, Douglas A. ; Quatieri, Thomas F. ; Dunn, Robert B.: Speaker Verification Using Adapted Gaussian Mixture Models. In: *Digital Signal Processing* 10 (2000), Nr. 1-3, 19-41. http://dblp.uni-trier.de/db/journals/dsp/dsp10.html#ReynoldsQD00

[RU11] Rathgeb, Christian ; Uhl, Andreas: A survey on biometric cryptosystems and cancelable biometrics. In: *EURASIP Journal on Information Security* 2011 (2011), Nr. 1, S. 1–25

[Sie09] Siegert, Ingo: *Implementierung einer Sprecherverifikation für ein generisches Telefon-Dialogsystem*, Otto von Guericke Universität Marburg, Diplomarbeit, Juni 2009

[ST95] Schukat-Talamazzini, Ernst G.: *Automatische Spracherkennung*. Vieweg Verlag, 1995

[Wag95] Waggener, Bill: *Pulse code modulation techniques: With applications in communications and data recording*. New York : Van Nostrand Reinhold, 1995. – ISBN 9780442014360

[Wei] Weisstein, Eric W.: *Hadamard Matrix*. http://mathworld.wolfram.com/HadamardMatrix.html. – zuletzt aufgerufen am: 03.01.2014

[Wun01] Wunsch, Holger: *Der Baum-Welch Algorithmus für Hidden Markov Models, ein Spezialfall des EM-Algorithmus*. 2001. – http://www.sfs.uni-tuebingen.de/resources/em.pdf, zuletzt aufgerufen am: 07.02.2014

[YG08] Yampolskiy, Roman V. ; Govindaraju, Venu: Behavioural biometrics a survey and classification. In: *Int. J. Biometrics* 1 (2008), Juni, Nr. 1, 81–113. http://dx.doi.org/10.1504/IJBM.2008.018665. – DOI 10.1504/IJBM.2008.018665. – ISSN 1755–8301