Filtertechniken für geschützte biometrische Datenbanken

Christian Böhm[a], Ines Färber[b], Sergej Fries[b], Ulrike Korte[c], Johannes Merkle[d], Annahita Oswald[a], Thomas Seidl[b], Bianca Wackersreuther[a], Peter Wackersreuther[a]
[a] LMU München, {boehm|oswald|wackersb|wackersr}@dbs.ifi.lmu.de
[b] RWTH Aachen, {faerber|fries|seidl}@informatik.rwth-aachen.de
[c] BSI Bonn, ulrike.korte@bsi.bund.de
[d] secunet Essen, johannes.merkle@secunet.com

Abstract: In immer mehr sicherheitsrelevanten Bereichen werden biometrische Erkennungstechniken für die Zugangskontrolle oder die Identitätsfeststellung einer Person eingesetzt. Da biometrische Merkmale hoch sensibel sind, müssen sie vor unbefugtem Zugriff geschützt werden. Sogenannte Template Protection Verfahren ermöglichen eine biometrische Authentisierung, ohne dass sich diese Merkmale aus den gespeicherten Referenzdaten ermitteln lassen. Allerdings erschweren diese Verfahren die Suche nach passenden Referenzdaten und machen daher die Identifikation innerhalb umfangreicher Datenbestände ineffizient. In diesem Artikel werden erste Ansätze untersucht um auch für große Datenmengen eine Identifikation auf Basis von geschützten Fingerabdrücken durchführen zu können. Die vorgestellten Verfahren erstellen durch Filtertechniken und Indexstrukturen eine geeignete Priorisierung der Datenbankeinträge, sodass der aufwändige exakte Vergleich zwischen Anfrage und den transformierten Einträgen gezielt erfolgen kann.

1 Einleitung

Der Einsatz biometrischer Merkmale in Identifikationssystemen hat in den letzten Jahren stark zugenommen. Biometrische Erkennungsmerkmale sind in der Regel universell, einzigartig, persistent und personengebunden. Die Persistenz der biometrischen Daten bedingt jedoch, dass sie einmal korrumpiert unwiederbringlich als Identifikationsmerkmal für das betreffende Individuum verloren sind. Zudem bergen biometrische Merkmale neben den benötigten Informationen für eine Identifikation auch sehr sensible Informationen, z.B. über die ethnische Zugehörigkeit oder den Gesundheitszustand. Daher ist die Verwendung biometrischer Daten aus Sicht des Datenschutzes nicht unumstritten.

Um die sensiblen Daten sicher zu speichern, haben sich sogenannte *Template Protection Verfahren* etabliert, wobei das Fuzzy Vault-Verfahren [JS06] dabei zu den Bekanntesten zählt. Hier werden die biometrischen Eigenschaften durch künstlich hinzugefügte Merkmale gegen weit verbreitete Angriffstechniken geschützt. Da eine Suche in den transformierten Referenzdaten im Allgemeinen dann aber sehr ineffizient ist, ist der praktische Einsatz für Identifikationszwecke bisher noch ein offenes Problem.

In der Publikation von Korte *et al.* [KMN09] wurde bereits ein Verfahren zum Abgleich eines ungeschützten Fingerabdrucks mit einem geschützten Datenbankeintrag basierend

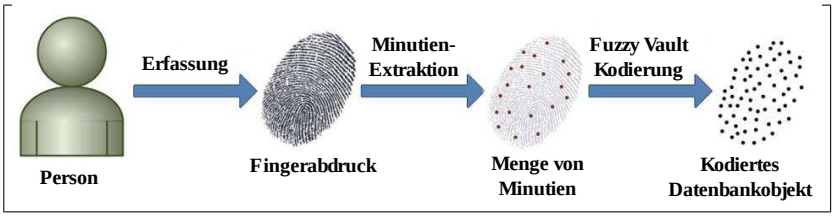


Abbildung 1: Erzeugen eines kodierten Datenbankobjekts für den Identifikationsprozess.

auf Minutien vorgestellt. Dieses System implementiert jedoch nur den Authentifikationsprozess, bei dem die Identität des Nutzers à priori bekannt ist. Um auch die Identifikation in akzeptabler Zeit zu beantworten, ist neben einer effizienten Verifikation auch die Anzahl der in Frage kommenden Datenbankobjekte geeignet einzuschränken. Indexstrukturen sowie Filterarchitekturen ermöglichen eine entsprechende Vorauswahl durch verschiedene Approximationstechniken. Allerdings unterliegen die biometrischen Daten neben der schützenden Transformation meist starkem Rauschen. So können die Finger beim Scannen gedreht oder verschoben aufgelegt werden. Zudem kann der Abdruck durch unterschiedlichen Druck des Fingers auf den Sensor Verzerrungen aufweisen. Zusätzlich können Deletionen oder Insertionen von Merkmalen auftreten. Es muss also davon ausgegangen werden, dass prinzipiell kein Objekt mit Sicherheit für die anschließende Verifikation ausgeschlossen werden kann.

Bisher sind keine effizienten Suchverfahren für eine Identifikationslösung mit Template Protection Verfahren bekannt. Diese Arbeit beschreibt daher erste Ansätze, um effiziente Datenbanktechniken in das biometrische Anwendungsgebiet zu integrieren. Durch den Einsatz effizienter Filterarchitekturen bzw. Indexstrukturen ermöglichen wir eine schnelle Identifikation von Personen auf Basis geschützter Fingerabdrucksbilder (vgl. Abbildung 1), wobei die Herausforderung darin liegt, mit einem sehr starken Rauschen innerhalb der Daten, verursacht durch das Fuzzy Vault und den zusätzlichen Rauscheffekten, umzugehen. Zwei Fingerabdrücke werden dabei auf Basis ihrer zuvor extrahierten Minutien (End- und Verzweigungspunkte der Papillarlinien) verglichen. Wir stellen zwei unterschiedliche Ansätze vor, die eine Rangfolge der Datenbankobjekte erstellen, sodass tendenziell ähnlichere Objekte für den anschließenden Verifikationsvorgang priorisiert werden. Unsere Experimente zeigen, dass sich die Zahl der durchzuführenden Verifikationen deutlich reduzieren lässt.

Der Rest dieses Artikels ist wie folgt gegliedert: Abschnitt 2 erläutert den allgemeinen Identifikationsvorgang. Unsere beiden Verfahren GeoMatch und BioSimJoin werden in Abschnitt 3 vorgestellt. Abschnitt 4 beschreibt den Prozess der Erstellung aller verwendeten biometrischen Datenbanken. Eine ausführliche Evaluierung unserer Verfahren folgt in Abschnitt 5. Abschließend fassen wir diesen Artikel in Abschnitt 6 zusammen.

2 Identifikationssysteme für Biometrische Datenbanken

Bei der biometrischen Identifikation steht die Identität des Benutzers nicht à priori fest. Sie entspricht daher einem Scan über die komplette Datenbank, bis eine Übereinstimmung gefunden wird. Als biometrische Merkmale eines Fingerabdrucks verwenden wir die kartesischen Koordinaten seiner Minutien $m = (m_x, m_y)$. Im Folgenden wird ein Finger stets durch die Menge seiner Minutien repräsentiert. Sei die Anfrage repräsentiert durch ein Template Q, das gegen eine Datenbank $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ von Referenztemplates verglichen werden soll. Alle Referenztemplates R_j werden schon beim Enrolment durch das Fuzzy-Vault Verfahren transformiert. Im Folgenden wird zunächst das Enrolment, also das Erstellen von R_j, sowie die anschließende Verifikation von Q, beschrieben.

Enrolment. Das Verfahren Fuzzy-Vault [JS06] zählt zu den bekanntesten und meist akzeptierten Template Protection Verfahren und wurde für ungeordnete biometrische Merkmale unterschiedlicher Länge entwickelt. Aus diesem Grund eignet es sich zum Schutz der Minutien aus Fingerabdrücken. Beim Enrolment wird ein Polynom $p(\alpha)$ mit Grad k bestimmt, dessen Koeffizienten $Z = [a_0, \cdots, a_k] : p(\alpha) = \sum_{i=0}^{k} a_i \cdot \alpha^i$ zufällig gewählt werden. Die Koordinaten (m_x, m_y) jeder Minutie des Fingers werden zusammen als ein Element α des endlichen Körpers $\mathbb{F}_q$ dargestellt, wobei q so groß gewählt werden muss, dass alle Minutienpositionen eindeutig darin abgebildet werden können. Die Minutien-Informationen bilden zusammen mit den jeweiligen Funktionswerten $\beta = p(\alpha)$ die Stützpunkte des Polynoms. Zusätzlich werden sogenannte Chaff-Punkte generiert, die nicht auf p liegen. Diese dienen dazu, die echten Minutien zu verschleiern. Dazu werden für alle Chaff-Punkte zufällige $\beta \neq p(\alpha)$ generiert. Die Stützpunkte von p werden mit den Chaff-Punkten vermischt und als Datenbanktemplate R_j zusammen mit dem Hashwert $h(Z)$ gespeichert.

Verifikation. Bei der Verifikation werden die Minutien $m_Q \in Q$ mit denen des Referenztemplates R_j verglichen. Anhand konkreter globaler Translationen und Rotationen von Q, sowie einer Toleranz bzgl. Positionsabweichungen werden die übereinstimmenden Punkte ermittelt. Diese werden zusammen mit den entsprechenden Werten β_i aus R_j zur Rekonstruktion des Polynoms mit Hilfe des Reed-Solomon-Dekoders [RS60] verwendet, welches anschließend mit dem in der Datenbank hinterlegen Wert $h(Z)$ verglichen wird.

3 Effiziente Filter-Techniken

Für die Identifikation einer Person könnten naiv Authentisierungssysteme, wie beispielsweise das in [MIK+10] publizierte Verfahren, derart eingesetzt werden, dass sequentiell die gesamte Datenbank durchsucht wird bis ein Treffer erfolgt. Da diese Verfahren durch das explizite Austesten aller natürlichen Transformationen (Rotation, Translation) des Anfragefingers sehr ineffizient sind, ist dies jedoch für große Datenbanken nicht praktikabel. Stattdessen muss die Menge der zu überprüfenden Referenztemplates zuvor geeignet gefiltert werden, damit die exakte Verifikation nur auf einer reduzierten Menge von Personen durchgeführt wird. Die von uns vorgeschlagenen Verfahren dienen beide dazu, ein Ranking der Personen aufzustellen, absteigend sortiert nach Ähnlichkeit bezüglich der angefragten Person P_Q, wodurch die Zeit für den gesamten Identifikationsprozess verringert wird.

Um die Sicherheit vor Brute-Force Angriffen zu erhöhen, bietet es sich an, eine Person über mehrere Finger zu identifizieren. Eine Anfrage P_Q und das entsprechende Referenzobjekt einer Datenbank $P_j \in DB$ bestehen somit allgemein aus $\theta \in \{1, \cdots, 10\}$ Templates: $P_Q = \{Q_1, \cdots, Q_\theta\}$ und $P_j = \{R_{j,1}, \cdots, R_{j,\theta}\}$. Ein Template der Anfrage $Q_f \in P_Q$ ist eine Menge von Minutienkoordinaten $m_Q = (m_x, m_y)$. Alle Referenztemplates $R_{j,f}$ einer Person P_j werden zusätzlich durch zufällig eingestreute Chaff-Punkte verschleiert. Da für jedes Template $R_{j,f}$ der zugehörige Fingertyp $f \leq \theta$ bekannt ist, kann angenommen werden, dass jeder Fingertyp in einem separaten Datenraum DB_f verwaltet wird.

3.1 GeoMatch

Der Vergleich zweier Templates Q_f und $R_{j,f}$ des entsprechenden Fingertyps f stellt durch das typischerweise vorliegende Rauschen und vor allem durch die Verschleierung eine Herausforderung dar. Ein einfacher Abgleich der Koordinaten beider Punktmengen ist hier nicht zielführend, stattdessen muss eine sehr große Menge von Transformationen in Betracht gezogen werden. Für diesen ersten Ansatz GeoMatch bedienen wir uns einiger Prinzipien aus dem verwandten Dockingproblem für Proteine. Ein typischer Ansatz hier ist es, das Problem in kleinere Elemente zu zerlegen, auf deren Basis individuelle Matchings durchgeführt werden. Anschließend werden diese lokalen Lösungen auf globale Konsistenz hin überprüft. Um einen Großteil der fraglichen Transformationen ausschließen zu können, wird beispielsweise in [Len95] ein Vergleich von Dreiecken, gebildet durch Zentren relevanter Moleküle, zu Grunde gelegt.

Für das vorliegende Problem des Vergleichs zweier Punktmengen Q_f und $R_{j,f}$ werden für beide Mengen Tripel berechnet, deren paarweise euklidischen Distanzen einen Schwellwert l_u übersteigen, sowie einen Schwellwert l_o unterschreiten. Die so gebildeten Dreiecke D_{Q_f} des Anfragetemplates werden mit den Dreiecken $D_{R_{j,f}}$ des Datenbanktemplates anhand der Seitenlängen (euklidiche Distanz zweier Minutien) auf Ähnlichkeit getestet. Der Vergleich der lokalen Strukturen erfolgt somit unabhängig von Koordinatenwerten und vernachlässigt sowohl ihre globale Ausrichtung als auch ihre globale Positionierung. Um den Einfluss lokaler Positionsfehler der Minutien, begründet durch Ungenauigkeiten beim Enrolment oder bei der Minutienextraktion, zu schwächen, wird bei dem Vergleich der Seitenlängen eine Fehlertoleranz δ berücksichtigt. Sind zwei Dreiecke $d_a \in D_{Q_f}$ und $d_b \in D_{R_{j,f}}$ ähnlich, so wird ihre relative Rotation γ zueinander ermittelt. Für eine globale Prüfung auf Konsistenz wird im Anschluss überprüft, für wie viele Dreiecke in D_{Q_f} ein Dreieck in $D_{R_{j,f}}$ mit gleicher relativer Rotation vorliegt. Eine hohe Anzahl solcher Dreiecke lässt neben der vielen lokalen Matches auch Rückschlüsse auf eine globale Drehung ähnlicher Gesamtstrukturen zu. Je höher also die Anzahl ähnlicher lokaler Rotationen, desto wahrscheinlicher die Ähnlichkeit beider Templates. Die Ähnlichkeit zweier Templates Q_f und $R_{j,f}$ bestimmt sich somit wie folgt, wobei $\mathcal{A} = \{0°, \cdots, 360°\}$ die Menge aller Winkel in einer zu wählenden Diskretisierung (z.B. 2°-Schritte) ist:

$$sim(D_{Q_f}, D_{R_{j,f}}) = \max_{\gamma \in \mathcal{A}} \left\{ \left| \left\{ d_a \in D_{Q_f} \mid \exists d_b \in D_{R_{j,f}}.\ |d_a - d_b| \leq \delta \wedge \angle(d_a, d_b) \approx \gamma \right\} \right| \right\}$$

Für alle Referenztemplates $R_{j,f} \in DB_f$ wird die maximale Anzahl der Anfragedreiecke bestimmt, für die eine gleiche relative Rotation bzgl. Q_f ermittelt wurde. Dieser Prozess wird parallel für alle θ Finger der Anfrage Q durchgeführt. Für alle Personen $P_j \in DB$ werden anschließend die θ Ähnlichkeiten aller Finger $R_{j,i} \in P_j$ aufaddiert:

$$sim(P_Q, P_j) = \sum_{f \in \theta} sim(D_{Q_f}, D_{R_{j,f}})$$

Im Anschluss kann die Datenbank anhand dieser Anzahl absteigend sortiert und so für den Verifikationsprozess priorisiert werden.

Das Verfahren GeoMatch zeichnet sich durch seine Robustheit gegenüber globaler Verschiebung, sowie globaler Rotation der Templates aus, da lediglich Distanzen zwischen Minutien, nicht aber ihre Koordinaten verglichen werden.

3.2 BioSimJoin

Der Nachteil von GeoMatch ist die hohe Laufzeit, die der Abgleich mehrerer Dreiecksstrukturen zwischen Q und einem Referenztemplate R_j mit sich bringt. Aus diesem Grund werden bei BioSimJoin nicht die geometrischen Beziehungen zwischen Minutien und Chaff-Punkten betrachtet, sondern vielmehr Vergleiche der Punktmengen mittels Bereichsanfragen unterstützt durch eine Indexstruktur durchgeführt.

BioSimJoin speichert die Minutien bzw. Chaff-Punkte aller Personen $m_{R_{j,f}} \in R_{j,f}$ in einem Datenraum des entsprechenden Fingertyps f. Dabei werden die x- und y-Koordinaten aller Minutien bzw. Chaff-Punkte in einer Indexstruktur aus der Familie der R-Bäume [Gut84] organisiert. Diese hierarchischen Indexstrukturen, die ursprünglich zur Speicherung von hochdimensionalen Daten entwickelt wurden, eignen sich daher für die Verwaltung von biometrischen Punktdaten. Sie ermöglichen eine effiziente Beantwortung von Bereichsanfragen (d. h. Rechtecks- bzw. Intervall-Anfragen), und sind zudem dynamisch, d.h. durch effiziente Einfüge- bzw. Lösch-Operationen kann die Struktur bei Veränderung des Datenbestandes effizient aktualisiert werden.

Aufgrund der vorliegenden Rauscheffekte wie Rotation bzw. Translation zwischen der Anfrage Q_f und den Datenbanktemplates $R_{j,f}$ kann kein direkter Punktvergleich durchgeführt werden. Daher wird bei der Anfrage für jede Minutie $m_{Q_f} \in Q_f$ eine Bereichsanfrage mit Radius r durchgeführt. Für jede dieser Minutien m_{Q_f} werden im Datenraum des entsprechenden Fingertyps f diejenigen Punkte (Minutien oder Chaff-Punkte) bestimmt, die sich innerhalb des Bereiches mit Radius r um m_{Q_f} befinden, d.h. deren euklidischer Abstand r nicht überschreitet. Die Information, ob es sich bei den Punkten um Minutien oder Chaff-Punkte handelt ist dabei nicht bekannt, lediglich welcher Person P_j sie zugeordnet sind. Falls eine Minutie $m_{R_{j,f}}$ einer Person P_j in den Radius der Anfrageminutie m_{Q_f} fällt, wird die Anzahl der Treffer für P_j um 1 erhöht. Die resultierende Kandidatenliste entspricht einer Liste an Personen P_j die absteigend nach Anzahl der Treffern für diese Person sortiert ist. Der algorithmische Ablauf von BioSimJoin ist in Algorithmus 1 zusammengefasst. Wie auch bei GeoMatch kann die Berechnung für verschiedene Finger-

typen parallelisiert erfolgen. Schließlich werden die Treffer aller Finger für jede Person aufsummiert.

Algorithm 1 $filter_{BioSimJoin}(Q_f, r)$

```
candidates = [(P_1,0), (P_2,0), ..., (P_n,0)]
DB_f = [(x_{1,f},y_{1,f},R_{1,f}),..., (x_{n,f},y_{n,f},R_{n,f})]
for all minutia m_{Q_f} in Q_f do
  for all minutia m_{R_{j,f}} in DB_f do
    if dist(m_{Q_f}, m_{R_{j,f}}) ≤ r then
      candidates[j].increment()
    end if
  end for
end for
return Kandidatenliste sortiert nach Anzahl der Treffer
```

4 Biometrische Datenbanken

Für die Evaluierung beider Verfahren verwenden wir die zwei Datenbanken für Fingerabdrucksbilder NIST SD14 [WGT+07] und FVC-2002 DB1 [MMJP09]. Während die Bilder der wesentlich größeren NIST-Datenbank durch Scans von Tintenabdrücken entstanden und somit sehr starkes Rauschen enthalten, wurden die Aufnahmen der für heutige Erkennungssysteme wesentlich repräsentativeren FVC-Datenbank direkt digital erfasst. Um ein möglichst praxisnahes Enrolment zu simulieren, haben wir uns an die in [MIK+10] beschriebenen Anforderungen an eine biometrische Datenbank gehalten. Gleichzeitig gewährleisten wir einen hohen Schutz der Daten vor bekannten Angriffen, wie in [MMT09] formuliert. Die Minutien wurden mittels des NIST Algorithmus mindtct [WGT+07] extrahiert, nach Qualität gefiltert und anschließend mit Hilfe des Fuzzy Vault [JS06] geschützt.

Multi-Finger. Die Verwendung mehrerer Finger pro Person für das Enrolment steigert exponentiell die Sicherheit vor Brute-Force Attacken. Aus diesem Grund identifizieren wir eine Person anhand von drei Fingern, wie in [MIK+10] empfohlen.

Feature-Selektion des Referenztemplates $\mathbf{R_j}$. Um eine möglichst hohe Qualität der Referenztemplates zu garantieren und Ungenauigkeiten aus Scanvorgang sowie Minutienextraktion zu verringern, werden lediglich zuverlässige Minutien verwendet. Dazu werden, wie in [MIK+10] beschrieben, die Minutien ermittelt, die aus mehreren Aufnahmen eines Fingers extrahiert wurden. Diese werden zudem anhand ihres von mindtct ausgegebenen Qualitätswerts rel derart gefiltert, dass über alle drei Finger einer Person hinweg die besten 90 Minutien gewählt werden. Diese 90 Minutien werden anschließend durch insgesamt 112 zufällig eingestreute Chaff-Punkte verschleiert.

Diese Parameter gewährleisten ein Sicherheitslevel von 2^{70} gegen Angriffe, die versuchen, „echte“ von „unechten“ Minutien, also Chaff-Punkten, unterscheiden zu können [MIK+10].

Feature-Selektion des Anfragetemplates Q. Da für Q in der Regel nur eine Aufnahme vorliegt, entfällt die Feature-Extraktion hinsichtlich korrespondierender Minutien meh-

rerer Aufnahmen. Stattdessen erfolgt die Filterung ausschließlich mittels des Qualitätskriteriums rel, wobei dieser jedoch einen Mindestwert von 0.25 übersteigen muss.

Die original Datenbank FVC-2002 DB1 enthielt ursprünglich jeweils acht Aufnahmen für 110 Finger. Nach allen Vorverarbeitungsschritten resultiert eine Datenbank, in der jeweils drei Finger zu insgesamt 27 Personen zusammengefasst wurden. Für diese Personen sind insgesamt 2.430 Minutien und 3.024 Chaff-Punkte gespeichert.

Die original NIST SD14 enthält zu 2.700 Personen jeweils zwei Aufnahmen für alle zehn Finger. Da die Qualität dieser Datenbank nicht an den heutigen Standard heranreicht, und einige Aufnahmen beispielsweise durch handschriftliche Bemerkungen stark verunreinigt sind, wurden nur die Bildpaare verwendet, die durch den BOZORTH Matchingalgorithmus [WGT$^+$07] als Abdrücke des gleichen Fingers erkannt werden. Für die verbleibenden 2.365 Personen werden die drei Anfragefinger gemäß folgender Priorisierung gewählt: linker/rechter Zeigefinger, linker/rechter Mittelfinger, linker/rechter Ringfinger, linker/rechter Daumen und linker/rechter kleiner Finger. Das entspricht einer Reihenfolge von $(7, 2, 8, 3, 9, 4, 6, 1, 0, 5)$ gemäß der Codierung des Fingertyps nach [WGT$^+$07]. Insgesamt enthält der erstellte Datenbestand 212.850 Minutien und 264.880 Chaff-Punkte.

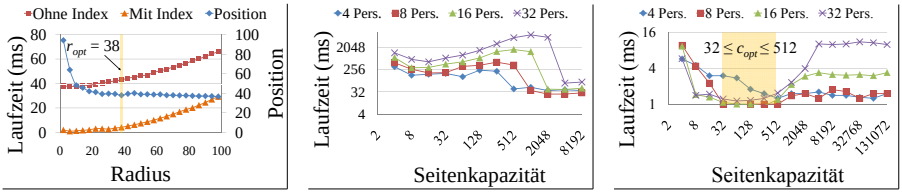
5 Experimente

Zunächst untersuchen wir die Parametereinstellung beider Verfahren sowohl anhand der Datenbank FVC-2002 DB1 als auch der Datenbank NIST SD14. Die so ermittelten optimalen Parameter werden hinterher zur Untersuchung der Effektivität, sowie der Effizienz der Verfahren auf beiden Datenbanken eingesetzt. Anschließend evaluieren wir die Robustheit beider Verfahren gegenüber gedrehten oder verschobenen Daten und untersuchen, inwieweit Insertionen bzw. Deletionen von Minutien die Ergebnisse beeinflussen. Alle Ergebnisse sind stets über alle Personen der entsprechenden Datenmenge gemittelt. Das bedeutet, dass in allen Experimenten jede Person ein Mal als Anfrage verwendet wird. Die Resultate entsprechen somit jeweils repräsentativen Durchschnittswerten.

Die Zeitmessungen wurden für jeden Finger parallelisiert auf folgenden Rechnern durchgeführt: Intel Dual Core Xeon 7120 M CPUs bzw. Intel XEON E5345 CPUs mit je 2.33 bis 3.0 GHz und 16 GB RAM. Alle Verfahren wurden mittels Java JDK 6.0 implementiert.

5.1 Parameterevaluierung

Die Seitenbeschränkungen für das Verfahren GeoMatch l_u und l_o wurden so gewählt, dass für nahezu alle Minutien aller Referenztemplates ein Dreieck ohne Chaff-Punkte konstruiert werden kann und gleichzeitig die Gesamtzahl aller Dreiecke möglichst gering ist. Diese Bedingungen erfüllt beispielsweise die gewählte Beschränkung der Seitenlänge der Dreiecke auf $14 - 80$ Pixel. Für die Fehlertoleranz zeigte ein Wert von $\delta = 1$ Pixel die besten Ergebnisse.



(a) Laufzeit und Effektivität in Abhängigkeit von r.

(b) Laufzeit für den Indexaufbau in Abhängigkeit von c.

(c) Laufzeit für die Suche in Abhängigkeit von c.

Abbildung 2: Bestimmung der optimalen Parameter von BioSimJoin.

Bei BioSimJoin müssen der Radius der Bereichsanfrage r und die maximale Kapazität einer Indexseite c geeignet gewählt werden. Da c lediglich die Effizienz des Verfahrens beeinflusst, kann zunächst eine Optimierung von r alleine durchgeführt werden. Abbildung 2(a) zeigt die entsprechenden Ergebnisse auf einem 10%igem Sample der Datenbank SD 14. Die durch Dreiecke und Quadrate markierten Kurven illustrieren die durchschnittlich benötigte Laufzeit, um die Kandidatenliste mit und ohne Indexunterstützung zu ermitteln. Ein höherer Radius impliziert die Überprüfung einer größeren Zahl von Datenbankelementen und daher eine erhöhte Laufzeit. Die mit Rauten markierte Kurve stellt die durchschnittliche Position der angefragten Person P_Q innerhalb der Kandidatenliste dar. Wenn r zu klein gewählt wird, kann P_Q erst relativ spät verifiziert werden. Den besten Kompromiss erzielt ein Radius $r_{opt} = 38$. BioSimJoin liefert nach durchschnittlich 6.23 ms bzw. 45.97 ms (ohne Indexunterstützung) eine Kandidatenliste, innerhalb derer sich P_Q durchschnittlich an Position 37.94 befindet. Unsere Experimente auf anderen biometrischen Datenbanken mit gleichem Setting ergaben, dass dieser Radius-Wert allgemein gute Ergebnisse verspricht.

Abbildung 2(b) zeigt den Zeitaufwand für die Indexierung, Abbildung 2(c) für die anschließende Suche, jeweils in Abhängigkeit von c. Diese Experimente wurden für unterschiedliche Datenbankgrößen durchgeführt. Die initialen Schwankungen sind durch Implementierungs-Overhead zu erklären. Für steigende Werte von c kann eine Zunahme der Laufzeit für den Indexaufbau beobachtet werden, da beim Splitten einer Seite sequenziell auf deren Elemente zugegriffen wird. Sobald c groß genug ist um alle Einträge in einer einzigen Seiten zu speichern, nimmt die Laufzeit der Indexierung rapide ab, wodurch allerdings auch keinerlei Indexunterstützung mehr für die anschließende Ähnlichkeitssuche gegeben ist. Diese Experimente lassen vermuten, dass eine optimale Kapazität $32 \leq c_{opt} < 512$, unabhängig von der Größe der angegebenen Datenbanken, sowohl für die Indexierung als auch die darauf aufbauende Suche gewählt werden sollte. Mittels eines sechsstufig gewichteten Mittelwerts der Laufzeiten für den relevanten Wertebereich $32 \leq c_{opt} < 512$ ergab sich ein globales Minimum bei $c_{opt} = 46$.

Diese Parametrisierungen beider Verfahren zeigten auf beiden Datenbanken gute Ergebnisse, sodass diese für alle folgenden Experimente übernommen wurden.

5.2 Effektivität

Für die FVC Datenbank gibt Tabelle 1 für GeoMatch sowie BioSimJoin jeweils die Position an, die eine angefragte Person P_Q gemittelt über alle 27 Anfragen in der Kandidatenliste einnimmt. Bei BioSimJoin wird die Kandidatenliste im Schnitt in 16.29 ms erzeugt, und P_Q ist auf Position 11.44 zu finden. Das Verfahren GeoMatch erzielt hier eine deutlich bessere Positionierung der korrekten Referenz in der Datenbank, benötigt für die entsprechenden Berechnungen allerdings signifikant mehr Zeit.

Abbildung 3 stellt das Ergebnis von GeoMatch und BioSimJoin anhand der Datenbank SD14 für unterschiedliche Datenbankgrößen dar. Trotz starkem Rauschen bei dieser Datenbank finden beide Verfahren die angefragte Person im Schnitt im vorderen Drittel der Kandidatenliste. Bei einer Datenbank bestehend aus 2.300 Personen kann mit GeoMatch die bei der anschließenden Verifikation zu überprüfende Anzahl an Personen um 69% reduziert werden. BioSimJoin schließt für die Verifikation 66% der Personen aus.

	GeoMatch	BioSimJoin
Position	2.07	11.44
Laufzeit	91.19 ms	16.29 ms

Tabelle 1: Effektivität bzgl. Datenbank FVC.

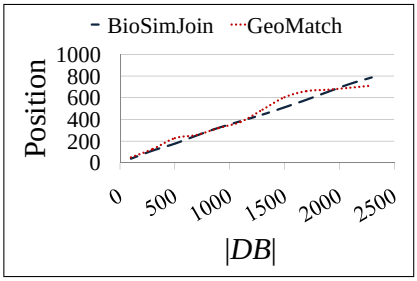


Abbildung 3: Effektivität bzgl. Datenbank SD14.

5.3 Effizienz

Abbildung 4 untersucht die Skalierbarkeit von GeoMatch und BioSimJoin anhand der Datenbank SD14. GeoMatch zeigt hier eine lineare Laufzeit mit zunehmender Anzahl an Datenbankeinträgen (Minutien bzw. Chaff-Punkten). Durch den Einsatz einer Indexstruktur erzielt BioSimJoin eine deutlich geringere Laufzeit. So benötigt BioSimJoin für eine Anfrage auf eine Datenbank bestehend aus knapp 500.000 Einträgen lediglich 2.4 Sekunden, GeoMatch hingegen 2 Minuten und 10 Sekunden.

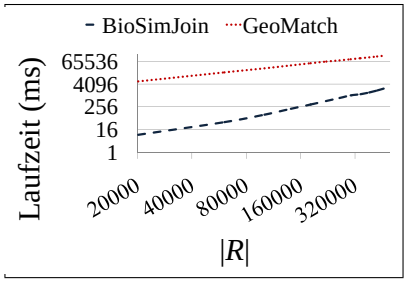
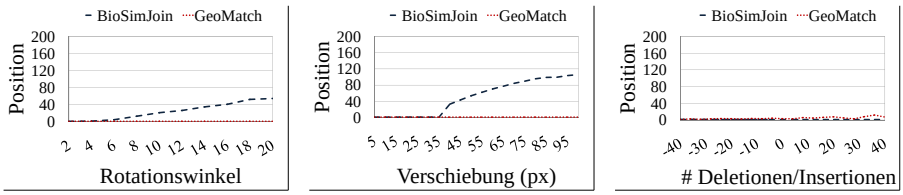


Abbildung 4: Skalierbarkeit der Verfahren.



(a) Rotation. (b) Translation. (c) Deletionen/Insertionen.

Abbildung 5: Robustheit von GeoMatch und BioSimJoin gegenüber unterschiedlichem Rauschen.

5.4 Evaluierung der Robustheit anhand synthetischer Daten

Um die Stabilität von GeoMatch und BioSimJoin gegenüber Rotation, Translation, fehlenden oder zusätzlichen Minutien zu testen, wurden jeweils die Minutien von 200 zufällig ausgewählten Personen der Referenzdatenbank SD14 gezielt manipuliert und als Anfrage verwendet. Für jedes Experiment ist die Position der angefragten Person P_Q innerhalb der Kandidatenliste gemittelt über 200 Anfragen angegeben.

Rotation. Um die Auswirkung eines rotierten Anfragetemplates Q auf die Effektivität der Verfahren zu untersuchen, drehten wir das Koordinatensystem von Q um einen Rotationswinkel ϕ im Intervall $[2°, 4°, 6°, \ldots, 20°]$. In der Regel wird beim Enrolment lediglich eine Rotation um bis zu $20°$ toleriert. Minutien, die durch die Rotation aus dem Bildbereich fallen, wurden verworfen. Abbildung 5(a) zeigt, dass die Effektivität von BioSimJoin mit zunehmender Drehung der Minutien aus Q leicht abnimmt, da die Minutien der Referenz nicht mehr optimal in die entsprechenden Anfrageradien fallen. Allerdings liegt P_Q bei einer starken Drehung von $20°$ im Mittel in der Kandidatenliste immer noch auf einem guten Platz 54 von insgesamt 200, obwohl die Drehung bei BioSimJoin nicht explizit berücksichtigt wird. Das Verfahren GeoMatch ist robust gegen Rotationen.

Translation. Um abweichende Auflagepositionen des Anfragefingers auf dem Scanner zu simulieren, wurden die Minutien-Koordinaten aus Q einheitlich um jeweils x Pixel verschoben. Auch hier wurden verschobene Minutien außerhalb des Bildbereichs ausgeschlossen. Bei einer Verschiebung um einen Wert kleiner dem Radius r der Bereichsanfrage von BioSimJoin bleiben die gesuchten Personen innerhalb der Kandidatenliste stabil auf dem ersten Platz (vgl. Abbildung 5(b)). Erst bei einer darüber hinausgehenden Verschiebung, fallen einige Minutien der Referenz aus der Bereichsanfrage, sodass die Effektivität leicht abnimmt. GeoMatch ist hier robust und zeigt konstant optimale Ergebnisse.

Insertionen und Deletionen. Bedingt durch Ungenauigkeiten im Scanvorgang oder der Minutienextraktion werden für Q teilweise andere Minutien erkannt als für die Referenz. Für entsprechende Untersuchungen wurden Q im Vergleich zu den Referenzdaten zufällig Minutien hinzugefügt bzw. entfernt. Abbildung 5(c) zeigt, dass für das Verfahren BioSimJoin das Fehlen bzw. Hinzukommen von bis zu 40 Minutien keine Auswirkung auf die optimale Erkennungsleistung haben. Die Ergebnisse von GeoMatch unterliegen hingegen leichten Schwankungen. Besonders bei einer großen Anzahl zusätzlicher Minutien kommt es vor, dass einige der zusätzlichen Minutien im Anfragetemplate auf Chaff-Punkte fremder Referenztemplates matchen, wodurch diese fälschlich im Ranking begünstigt werden.

6 Zusammenfassung

Wir haben erste Ansätze vorgestellt, die eine Personenidentifikation anhand ihres geschützten Fingerabdrucks effizient ermöglichen. Bisherige Verfahren unterstützen lediglich eine Identifikation auf ungeschützten Daten oder eine Authentifikation auf sehr kleinen Datenmengen. Die von uns entwickelten Filtertechniken erzeugen ein priorisiertes Ranking, anhand dessen ein genauer Abgleich von Anfrage und geschütztem Referenzobjekt durchgeführt wird. Experimente auf realen und synthetischen Daten zeigen, dass trotz starkem Rauschen wie Rotation, Translation und Insertionen bzw. Deletionen bei der Anfrage, und der Verschleierung der Referenzdaten, eine effektive und effiziente Identifikation ermöglicht wird. Während sich das Verfahren GeoMatch haupsächlich durch Rotations- und Translationsinvarianz auszeichnet, werden starke Effizienzsteigerungen in erster Linie durch das zweite Verfahren BioSimJoin erzielt. In naher Zukunft werden wir uns damit beschäftigen, beide Kriterien in ein Verfahren zu integrieren.

Danksagung: Diese Arbeit wurde innerhalb des Projekts BioKeyS des Bundesamt für Sicherheit in der Informationstechnik (BSI) durch den Zukunftsfond gefördert. Wir danken allen Partnern insbesondere Sebastian Abt, Christoph Busch, Heinrich Ihmor, Claudia Nickel, Alexander Nouak, Alexander Opel und Xuebing Zhou für die erfolgreichen Diskussionen und zahlreichen Kommentare.

Literatur

[Gut84] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *SIGMOD Conference*, Seiten 47–57, 1984.

[JS06] A. Juels und M. Sudan. A Fuzzy Vault Scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.

[KMN09] U. Korte, J. Merkle und M. Niesing. Datenschutzfreundliche Authentisierung mit Fingerabdrücken. *Datenschutz und Datensicherheit - DuD*, 33(5):289–294, May 2009.

[Len95] H.-P. Lenhof. An Algorithm for the Protein Docking Problem. In *Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism*, Seiten 125–139, 1995.

[MIK+10] J. Merkle, H. Ihmor, U. Korte, M. Niesing und M. Schwaiger. Performance of the Fuzzy Vault for Multiple Fingerprints (Extended Version). *CoRR*, abs/1008.0807, 2010.

[MMJP09] D. Maltoni, D. Maio, A.K. Jain und S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2009.

[MMT09] P. Mihailescu, A. Munk und B. Tams. The Fuzzy Vault for Fingerprints is Vulnerable to Brute Force Attack. In *BIOSIG*, Seiten 43–54, 2009.

[RS60] I. S. Reed und G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[WGT+07] C.I. Watson, M.D. Garris, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet und K. Ko. User's Guide to NIST Biometric Image Software (NBIS), National Institute of Standards and Technology, 2007.