Efficient Database Techniques for Identification with Fuzzy Vault Templates

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, {$faerber|fries|seidl$}@cs.rwth-aachen.de
[c] BSI Bonn, $ulrike.korte$@bsi.bund.de
[d] secunet Essen, $johannes.merkle$@secunet.com

Abstract: The authentication based on biometric information has several advantages compared to solely password-based systems, which has led to a growing interest of industry and of public authorities in biometric-based systems. To meet the high security standards concerning biometric data, template protection systems such as the fuzzy vault are indispensable to maintain the secrecy of the critical information. Several publications have discussed the application of fuzzy vault to fingerprint authentication systems. However, for identification purposes in large databases the fuzzy vault protection of the biometric reference data poses severe efficiency challenges.

In this work, we examine and compare the performance of three different approaches to enable the identification based on protected fingerprint minutiae templates also for large databases. All three approaches calculate a prioritization of the database entries exploiting filtering techniques and indexing structures. Based on this prioritization a search performing the complex exact comparison of database and query template is steered and thus will faster find a match.

1 Introduction

The need for reliable information security systems is present in all levels of society to prevent unauthorized system access or information theft. Cryptographic techniques are widely implemented and current algorithms (e.g. AES and RSA) excel in a very high proven security [Sta02]. Cryptosystems use one or more encryption keys to transform the critical information (plain text) into cipher text. Without the correct decrypting keys it is infeasible to reverse the encryption and convert the cipher text into the original information. The weakness of these systems is the underlying assumption that the cryptographic keys belong exclusively to authorized users. Consequently, the protection of the keys' secrecy has a very high priority for all cryptosystems. As these keys usually are very complex to be able to guarantee a certain security level, they cannot easily be memorized by users but are securely stored, i.e., the access is protected by an alternative security system. Overall, the security of the whole system is only as high as the security guaranteed by the weakest link, which, usually, is this second, alternative security system. Most popular are password-based systems. Passwords, however, can be easily lost, forgotten, stolen, or attacked, which motivates the application of biometrics for authentication purposes.

In biometric systems the identity verification is based on a person's anatomical and behavioral traits. An advantage over password-based authentication is that biometric features are persistent, personal, and cannot easily be forged. However, these systems require the storage of biometric reference data, which poses high privacy risks itself. Beside being personal data, biometric features can potentially reveal additional information, like health or ethnic aspects of a person [DBHO10, Mor07, Sei06], which are unnecessary for identity verification. The security of biometric templates is a fortiori crucial since once being corrupted a biometric trait cannot be revoked or reissued. Biometric cryptosystems [JNN08, BBGK08] can considerably mitigate these risks by using reference data that do not reveal significant information on the biometric trait. In case of successful authentication a key is extracted from or released by the biometric data presented. Hence, no key needs to be explicitly stored or presented for authentication. Out of the various biometric template protection systems, the fuzzy vault scheme is a very prominent one [JS02]. What qualifies the fuzzy vault scheme for the construction of biometric cryptosystems, is the ability to deal with unordered sets, which are commonly encountered in biometrics, and its error-tolerance, which is necessary because even two samples of the same biometric trait can differ substantially. Basically, the transformation of a biometric template is realized through a generation of random artificial attributes, which are interspersed to obfuscate the actual template. Only this transformed template will be stored and, therefore, a remaining challenge is the matching procedure to identify two corresponding templates.

Several approaches already proposed the application of the fuzzy vault scheme to fingerprint minutiae [UPJ05, NJP07, MNS+10]. These approaches, however, describe the application for authentication processes, where a claim of the subject's identity is made à priori. A naive transfer of these techniques to identification systems will not work efficiently for large databases, because in addition to an efficient verification process the number of potentially matching templates needs to be narrowed down significantly. In [BFF+11] we presented two first approaches for supporting the identification process for fuzzy vault transformed minutiae sets by efficient database techniques. In the present paper, these approaches are further evaluated along with a third approach. While there are proposals for biometric identification with encrypted templates (e.g. [BCK09]), to the best of our knowledge, besides our approaches no efficient algorithms for biometric identification with biometric cryptosystems do exist so far. The problem of identification systems in this setting is the impossibility to rule out any database entry with certainty due to the fuzzy vault transformation and the intraclass variations of biometric templates. Hence, our goal is a good prioritization of the present candidate set by utilizing efficient filtering techniques and indexing structures. The higher the approximated similarity of a protected template compared to a query is, the higher is the priority assigned to it by these techniques. In this work we discuss three approaches for efficient prioritization of the protected templates and evaluate them on the publicly available datasets FVC [MMC+02] and MCYT [OGFAS+].

The rest of the paper is organized as follows. In Section 2 we introduce the underlying fuzzy vault construction and formalize the problem setting. The description of the different filtering techniques follows in section 3. In section 4 we describe the databases and their preprocessing on which the experimental evaluation, presented in section 5, is based on. Section 6 summarizes our work and gives an outlook for possible future work.

2 Background

In this work we examine three approaches to support the efficient identification in large fingerprint minutiae databases protected through the fuzzy vault system. In the following we shortly introduce the basic notions for our approaches.

2.1 Fingerprint Feature

The biometric template for our study is a fingerprint's minutiae set. Minutiae of a fingerprint denote local ridge characteristics, of which the most prominent ones are ridge endings and ridge bifurcations. Typically minutiae are represented as triplets (x, y, ϕ), denoting their x- and y-position in the fingerprint image together with the angle ϕ of the associated ridges. In the present paper however, we follow the majority of constructions for the fuzzy fingerprints vault and use only the positions (x, y) of the minutiae. In order to determine the agreement between two fingerprints, the minutiae-based fingerprint matching basically calculates an alignment between the minutiae sets of two fingers that results in the maximum number of minutiae pairings. The difficulty here is that even two samples of the same fingerprint can vary to a very high extent. For example, translation, rotation, nonlinear distortion of the minutiae, spurious and missing minutiae can lead to high intraclass variations, which can lead to high false-positive and false-negative rates in authentication systems.

2.2 Fuzzy Vault

The fuzzy vault is a cryptographic scheme that is perfectly qualified to work with biometric features as it allows to deal with unordered sets of different cardinalities and moreover it is error tolerant - a necessary criteria for the mentioned intraclass variances.

Enrollment. To hide a secret K, we use a biometric template X, which is an unordered set of t (pairwise distinct) attributes (minutiae) $m_1, \cdots, m_t$. As secret key a random polynomial p over the finite field F_q with degree k, where $k \leq t \leq q$, is chosen. All attributes $m_i \in X$ have to be mapped to an element $x_i = f(m_i)$ of the field F_q, where f is an arbitrary injective function from the attribute space to the finite field. The elements x_i represent the supporting points for the chosen polynomial p, which later on will serve for reconstructing the secret polynomial. Instead of the polynomial, we, therefore, only store a list of (pairwise distinct) pairs $(x_i, p(x_i)) \in F_q^2$. To conceal the secret-revealing support points, we intersperse a large number (c) of chaff points $(x_j, y_j) \in F_q^2$, where $t < j \leq (c+t), \forall 1 \leq i, j \leq (c+t) : x_j \neq x_i \Leftrightarrow i \neq j$, and $\forall t < j \leq (c+t) : y_j \neq p(x_j)$ (the chaff points do not lay on the polynomial p). The randomly ordered list of all tuples $(x_1, y_1), \cdots, (x_{c+t}, y_{c+t})$ represents the vault, and is added as reference template R to the database.

Authentication. In order to recover the secret polynomial and, thereby, to verify its identity, a user has to present another set Q of attributes (query), which then will be compared

with the stored reference template R. The set of all pairs in R having a matching attribute in the query set ($\{(x_i, y_i) \in R | \exists q_j \in Q.x_i = q_j\}$) is used for the reconstruction of the polynomial using the Reed-Solomon decoding. Only if the template Q shares a sufficient amount of attributes with the genuine attributes of R, the polynomial reconstruction can be successful and, thus, the secret K be released.

2.3 Biometric Identification

In the identification process the user does not explicitly claim an identity and the system recognizes an individual by searching the entire template database for a match to the query template. Since an application of the authentication process to all entities of a database will be computationally expensive for large databases, we aim at a pre-selection such that only the most promising database candidates have to be matched against the query. Literature does, so far, not discuss solutions for template protected databases, which pose new challenges. One approach is, e.g., to classify a query into one of the pre-specified types (arch, tented arch, left loop, right loop, whorl) (natural proportion of fingerprints in these classes (3.7%,2.9%,33.8%,31.7%,29.7%) [WCW94]) and only compare it to the subset of the database corresponding to this type. Assigning each database template its according class, reveals this important and useful information to any attacker, which cannot be recommended.

3 Efficient Filtering Techniques

In this section, we discuss three different approaches, which reduce the database search space $DB = \{P_1, \cdots, P_{|DB|}\}$ for an individual P_Q to be identified in the database DB. For this purpose, all approaches determine a prioritization of the database, where the entries are ordered in descending order based on their similarity with the querying individual. The most promising entries in the database are thus favored for the exact verification, which accelerates the identification process. Based on the work of [MNS+10], we assume that an individual is identified through multiple fingers. A query P_Q and a database entry $P_j \in DB$, thus, generally consist of $|\theta|$ many templates, where $\theta \subseteq \{1, \cdots, 10\}$: $P_Q = \{Q_f | f \in \theta\}$ and $P_j = \{R_{j,f} | f \in \theta\}$. A template is represented by its set of coordinates $m_i = (x_i, y_i)$. While these coordinates for the query templates $Q_f \in P_Q$ describe only real minutiae, the coordinates for the database templates $R_{j,f} \in P_j$ consist of genuine minutiae as well as of chaff points. As for each template $R_{j,f}$ the according finger type f is known, we, for simplicity, assume that each finger type is treated separately.

3.1 GeoMatch

The comparison of the two templates Q_f and $R_{j,f}$ of a given finger type f is a challenging task due to the intraclass variances and the interspersed noise in the form of chaff points. Usually, for the alignment of the coordinate sets, a large number of transformations has to

be considered, which is too complex for an efficient filter technique. For the *GeoMatch* approach, we avail ourself of the principles of the related docking problem of proteins in biology. There, a common approach is to decompose the problem into small units, on which individual matchings are performed. Afterwards, these local solutions are checked for global consistency to build a solution for the original problem [Len95].

For the comparison of two coordinate sets Q_f and $R_{j,f}$, GeoMatch calculates a set of triangles for both sets, which are defined by those coordinate triplets, whose pairwise Euclidean distances exceed a threshold b_l and, as well, fall below a threshold b_u. In the following a triangle t is represented as a triplet of its edge lengths $t = (e_1, e_2, e_3)$, where, beginning with the shortest side, all edges are enumerated clockwise. Note, that the set of triangles $T_{R_{j,f}}$ for the database templates has to be computed only once and is stored for following identifications. The triangles of the query template T_{Q_f} are compared with the ones $T_{R_{j,f}}$ of the database template based on their edge lengths. To compensate the influence of local misplacements of minutiae caused by inaccuracies during the enrollment, the similarity check of two edges considers a tolerance of δ. The comparison of these local patterns is independent of the global positioning or rotation and therefore translation invariant, an important criteria to deal with intraclass variances. If two triangles $t_a \in T_{Q_f}$ and $t_b \in T_{R_{j,f}}$ are similar to each other, GeoMatch determines their relative rotation γ to each other. After all triangles have been compared, GeoMatch checks for global consistency by determining the maximal number of triangles in T_{Q_f} for which matching triangles with same relative orientation γ have been found in $T_{R_{j,f}}$. The larger this number of similar rotated matchings, the higher is the consensus that $R_{j,f}$ is a rotated version of Q_f and, thus, the higher is the probability for the similarity of both templates. The similarity of two templates Q_f and $R_{j,f}$ is defined as:

$$sim(T_{Q_f}, T_{R_{j,f}}) = \max_{\gamma \in \mathcal{A}} \left\{ \left| \left\{ t_a \in T_{Q_f} \mid \exists t_b \in T_{R_{j,f}}.\ ||t_a - t_b||_\infty \leq \delta \wedge \angle(t_a, t_b) \approx \gamma \right\} \right| \right\}$$

where $\mathcal{A} = \{0°, \cdots, 360°\}$ is the set of all angles for a given discretization (e.g. $2°$-steps). The similarity of $sim(T_{Q_f}, T_{R_{j,f}})$ is calculated for all reference templates j and for all finger types $f \in \theta$. Overall, the similarity between two individuals is determined through $|\theta|$ many of their fingers: $sim(P_Q, P_j) = \sum_{f \in \theta} sim(T_{Q_f}, T_{R_{j,f}})$. Subsequently the database entries P_j are sorted in descending order based on their similarity score and, thus, are prioritized for the verification process.

3.2 DistMatch

The approach *DistMatch* is based on [JY00] and [CCG06], which are techniques for local structure matching. Here local structures, consisting of one central minutia m and k additional minutiae $\{m_1, m_2, \cdots, m_k\}$ of its neighborhood, are used to compare two minutiae sets. For our problem setting, where the reference templates are protected, we have to expect a large number of chaff points in the neighborhood of a genuine minutia. Therefore, these structures, where the neighborhood is restricted to a size k or to a certain distance range [RPBV00], is not expedient. Instead, DistMatch determines for each minutia m_i

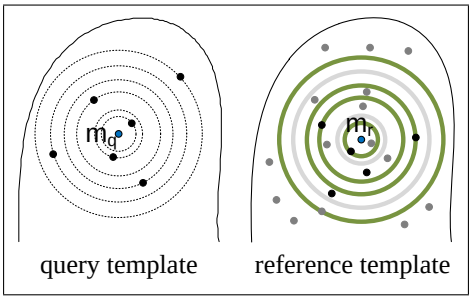


Figure 1: Comparing minutiae for DistMatch.

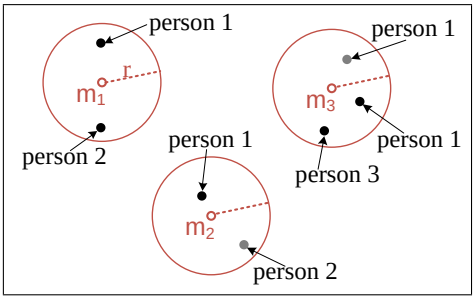


Figure 2: Identification for BioSimJoin.

of a template T a sorted list $l_i = (d_1, \cdots, d_{|T|-1})$ of the Euclidean distances of m_i to all remaining $|T| - 1$ minutiae. For each template T we yield a set L^T of $|T|$ such lists of neighborhood distances l_i, one for each minutia in the template. The similarity of two minutiae m_q and m_r is determined based on the agreement between their neighborhood distances, which is illustrated in Figure 1. All distances (represented as circles) of the query m_q, which find matches for m_r in the reference template, are colored green in the reference template. Here, not only distances to genuine minutiae (black) but also distances to the large amount of chaff points (grey) will match the query distances, which leads to high agreements between the minutiae. Analogously to GeoMatch, also DistMatch applies an error tolerance δ for considering two distances as equal. The similarity of a template $R_{j,f}$ given the template Q_f, is defined as the degree to which minutiae of L^{Q_f} are recovered in $L^{R_{j,f}}$:

$$sim(Q_f, R_{j,f}) = sim(L^{Q_f}, L^{R_{j,f}}) = \sum_{l_q \in L^{Q_f}} \max_{l_i \in L^{R_{j,f}}} |\{d_v \in l_q | \exists d_w \in l_i . |d_v - d_w| \leq \delta\}|$$

The similarity $sim(L^{Q_f}, L^{R_{j,f}})$ is determined for all reference templates j and for all finger types $f \in \theta$. The similarity of two individuals P_Q and P_j is determined based on all fingers $f \in \theta$: $sim(P_Q, P_j) = \sum_{f \in \theta} sim(L^{Q_f}, L^{R_{j,f}})$. Based on this similarity score, all entries P_j in the database are prioritized for the verification process. Similarly to GeoMatch, also DistMatch is robust against linear transformations of two templates Q_f and $R_{j,f}$ as the comparison is only based on distances. The complexity of $sim(Q_f, R_{j,f})$ is in $O(|Q_f| \cdot \max\{|Q_f|, |R_{j,f}|\} \cdot |R_{j,f}|)$, where $O(\max\{|Q_f|, |R_{j,f}|\})$ is the complexity for comparing two sorted lists $l_q \in L^{Q_f}$ and $l_i \in L^{R_{j,f}}$.

3.3 BioSimJoin

GeoMatch and DistMatch suffer from the drawback, that the comparison of the query $Q_f \in P_Q$ to a reference template $R_{j,f} \in P_j$ is very time-consuming. Therefore, in the approach *BioSimJoin*, the coded fingerprint information, including minutiae and chaff points are stored in an index structure, i.e. the R-tree [Gut84]. However, the minutiae of different fingers, e.g. thumb and index, are stored in different data spaces.

In a first step, a range query is performed for each minutia $m_i \in Q_f$ for the query person P_Q with radius r. In this way, we want to answer questions like "Find all minutiae/chaff points in the database that are located in the range r around the query minutia". This procedure is illustrated in Figure 2. The query fingerprint consists of three different minutiae m_1, m_2 and m_3. For each of these minutiae $m_i \in Q_f$, we determine all $m_k \in R_{j,f}$, where $1 \leq j \leq |DB|$ that are located within a range of radius r around m_i. Genuine minutiae are illustrated as black points, whereas grey points indicate chaff points. However, this information is not used during the identification process. Finally, we determine a list of minutiae/chaff points that are located within the specific range of each minutia m_i. For each of these minutia/chaff points the corresponding individual P is known. In our example, for minutia m_3 of the query two genuine minutiae and one chaff point within a range r are identified. The grey chaff point, as well as the minutia on the right belong to the finger of person 1. The minutia on the left refers person 3. Finally, we get the following list of candidates (w.r.t. all three minutiae of the query person): Four hits for person 1, two hits for person 2, and one hit for person 3. This ordering corresponds to the result of the approach BioSimJoin. Supported by an index structure, this can be done very efficiently.

Note that rotations and translations are not handled explicitly by BioSimJoin. However, these effects are offset by an adequate parameterisation for the radius r.

4 Biometric Databases

We used the two publicly available data sets FVC [MMC+02] and MCYT [OGFAS+] to test the effectiveness and efficiency of GeoMatch, DistMatch, and BioSimJoin. In order to guarantee a realistic identification scenario, we followed the suggestions in [MNS+10]. To ensure high protection of the biometric data against brute force attacks, we used the same approach as formulated in [MMT09]. We obtained the minutiae of each fingerprint using the minutiae extraction algorithm MINDTCT [WGT+07]. Each data set contains multiple records of the same fingerprint. Thus, we used BOZORTH [WGT+07] to find the three best matching records according to the BOZORTH matching score. The first record is used as query and the remaining two records as reference. We discarded those minutiae having a quality value below 0.25 and encoded the remaining minutiae using the template protection method Fuzzy Vault [JS02]. For each person we used three fingers for identification as this exponentially increases the safety against brute force attacks [MNS+10].

Feature Selection and Encoding of the Reference Template. Caused by inaccuracies during the scan process, it may occur that some minutiae are missing or are newly compared to the other records of the same finger. Therefore, we applied the comparison algorithm by Merkle et al. [MNS+10] to determine those minutiae that are present in all reference records of the same finger, to get one final reference template. Those reliable minutiae are filtered according to the quality value determined by MINDTCT so that the final reference template for one person contains the best 90 minutiae over three fingers. At last, we obfuscated these 90 minutiae by 120 chaff points. These parameters guarantee a security level of 2^{70} against attacks that try to uncover the genuine minutiae in a reference template [MNS+10].

Feature Selection of the Query Template. As in real application scenarios there is only one record as query available, we used the first record that best matches the remaining records according to the BOZORTH matching score. Those minutiae whose quality value is below a threshold 0.25, were discarded.

The original database MCYT contains 12 optical (dp) and capacitive (pb) fingerprints for each finger of 330 persons. For identification, we used the index, middle and ring finger of the left and right hand separately to get a set of 660 different persons for dp and pb respectively. After preprocessing, the database for the optical fingerprints contains 57.387 minutiae and 73.696 chaff points, and the database for capacitive fingerprints contains 47.058 minutiae and 73.472 chaff points. The original database FVC comprises eight records for 110 fingers. After preprocessing, we result in a database, where three fingers are combined to a total number of 27 persons. More precisely, the database contains 2.430 minutiae and 3.024 chaff points.

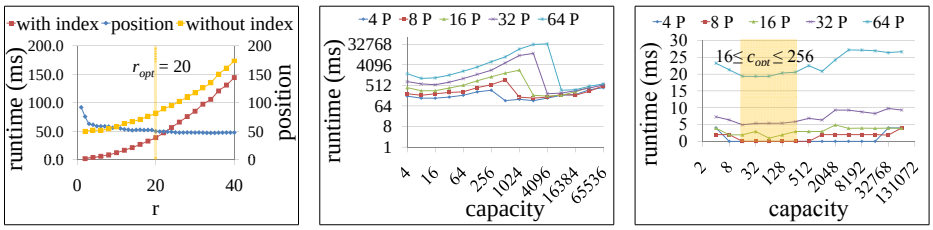
5 Experiments

In a first step, we determined the optimal parameters of all methods on a subset of 100 persons for the databases MCYT-dp and MCYT-pb. This includes the parameters b_l, b_u, and δ of GeoMatch, the parameter δ of DistMatch, and for BioSimJoin, the optimal range r and the capacity c of each index unit. The optimal parameters are then used to evaluate all methods in terms of efficiency and effectiveness. Due to the small size of the data set FVC, we use the parameters determined for MCYT-dp for the evaluation. For each data set, we report the average result over $|DB|$ runs, where we use each person once as query. The runtime was measured parallelized for each finger on following workstations: Intel Dual Core Xeon 7120 M CPUs and Intel XEON E5345 CPUs with each 2.33 to 3.0 GHz and 16 GB RAM. All approaches were implemented in Java 6.0.

5.1 Evaluation of Parameters

For the approach GeoMatch, the edge length restrictions b_l and b_u serve the purpose of considerably reducing the resulting amount of triangles for one template. Therefore, we choose these parameters such that, on the one hand, we do not lose information for any minutia and, on the other hand, get as few triangles as possible. On a sample of 100 individuals, these criteria are fulfilled for $(b_l, b_u) = (55, 123)$ (avg. 1462 triangles per reference template) in MCYT-dp and for $(b_l, b_u) = (28, 120)$ (avg. 2660 triangles per reference template) in MCYT-pb. For the error tolerance parameter a value of $\delta = 1$ showed best quality results for both databases.

For the approach DistMatch, similarly to GeoMatch, low values for the error tolerance parameter δ of $\delta = 2$ for MCYT-dp and $\delta = 1$ for MCYT-pb showed best results regarding the quality of the prioritization.



(a) Runtime and effectivity w.r.t. r. (b) Runtime for index construction w.r.t. c. (c) Runtime for search w.r.t. c.

Figure 3: Optimal Parameters for BioSimJoin.

For BioSimJoin, the radius of the range query r and the maximum capacity c of an index unit have to be optimized. As the parameter settings for MCYT-dp and MCYT-pb are equal, we discuss the results for both databases simultaneously. The curves marked by yellow and red squares in Figure 3(a) illustrate the runtime to generate the candidate list for different radius values in a range of $[0 - 40]$. The runtime is depicted on the left side of the diagram. Higher radius values implicate higher runtimes, as more candidates have to be tested in this case. The curve marked by blue diamonds illustrates the average position of the query in the resulting list of candidates. Extremely small radius values result in candidate lists where the reference of the actual query person is located at relatively backmost position, as small radius values do not tolerate rotated or translated data. With r=20 a trade-off between runtime and effectivity can be achieved.

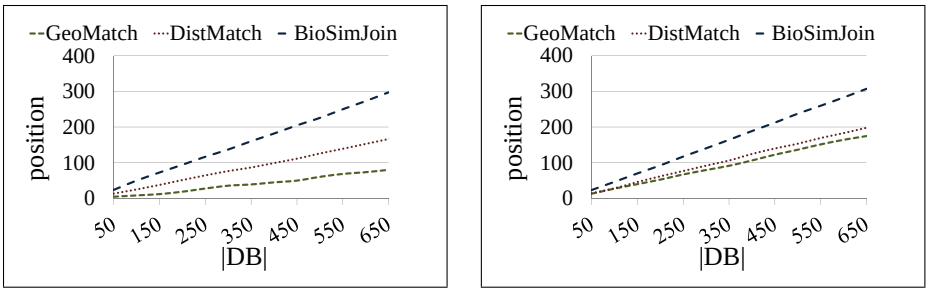
In Figure 3(b), we evaluate the impact of different capacity values on the runtime for the index construction based on different database sizes. The capacity is the amount of data that can be stored in an index unit of the R-tree. We tested the runtime for databases that consist of 4, 8, 16, 32 and 64 persons which refers to an actual database size of 808, 1.613, 3.217, 6.449 and 12.844 database entries for minutiae and chaff points. Higher capacity values implicate a higher effort for the index construction, as the split of an index unit is based on a sequential scan. Extremely high capacities lead to a significant decrease of runtime, as in this case all database entries are stored in only one index unit. However, no index support is given for the subsequent search. Figure 3(c) shows the time for generating the candidates list w.r.t. different capacities. Thus, optimal runtime for index construction and search can be achieved with a capacity in a range of $[16, 256]$ independently of the database size. Using a 6-stage weighted mean value of the runtime for different capacity values in the range of $[16 - 256]$, a significant optimum can be found at c=102. Therefore, we use r=20 and c=102 for all experiments.

5.2 Effectivity

Table 1 denotes the results of GeoMatch, DistMatch, and BioSimJoin on the database FVC. It specifies the position of a query person in the candidates list and the runtime to construct the candidates list each averaged over 27 queries. GeoMatch is able to achieve a database reduction of 92.03% for the exact verification, wheras DistMatch and BioSimJoin

	GEOMATCH	DISTMATCH	BIOSIMJOIN
Position	2.15	11.67	11.44
Runtime	162.4 ms	295.23ms	16.29 ms

Table 1: Effectivity on database FVC.



(a) Result on MCYT-dp. (b) Result on MCYT-pb.

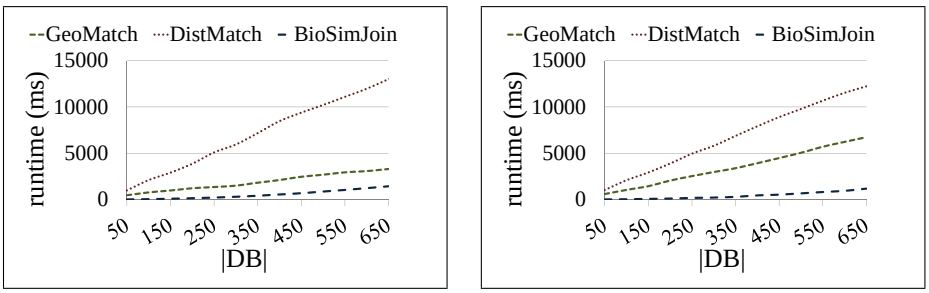
Figure 4: Effectivity on MCYT-dp and MCYT-pb.

only achieve a datareduction of 56.78% and 57.63%, respectively. However, in terms of runtime, the index supported BioSimJoin clearly outperforms GeoMatch and DistMatch.

Figure 4(a) and 4(b) illustrate the results for both MCYT databases. Whereas the results of BioSimJoin and of DistMatch for both databases are comparable, GeoMatch benefits from the better quality of the optical records in MCYT-dp and achieves here better results than on MCYT-pb. GeoMatch averagely ranks the query at 11% of MCYT-dp and at 27% of MCYT-pb, which is significantly better than DistMatch (25% and 30%) and BioSimJoin (45% and 46%). One explanation is that the comparison based on triangles, especially, with the global consistency check considering the relative angle γ, is more selective than the mere distance comparison in DistMatch. Due to the intraclass variances, BioSimJoin is not able to achieve a good prioritization as it considers the actual coordinate values.

5.3 Efficiency

Figure 5 denotes the scalability of GeoMatch, DistMatch and BioSimJoin. As the database size is the same for MCYT-dp and MCYT-pb, all methods, except for GeoMatch, result in similar runtimes for both data sets. Because of the less restrictive parameter setting $(b_l, b_u, \delta) = (28, 120, 1)$ of GeoMatch for MCYT-pb compared to $(b_l, b_u, \delta) = (55, 123, 1)$ for MCYT-dp, a reference template has on average more triangles in MCYT-dp than in MCYT-pb, which results in a higher runtime for MCYT-pb. GeoMatch and DistMatch show linear runtime with increasing database size. Supported by an index structure, BioSimJoin yields significantly lower runtime. Thus, to request a database comprising 650 database objects in MCYT-dp takes only 1429 ms with BioSimJoin, whereas GeoMatch takes 3294 ms and DistMatch takes even 13144 ms. Due to its quadratic complexity for the comparison of two templates, DistMatch has the worst runtime of our approaches.



(a) Result on MCYT-dp.

(b) Result on MCYT-pb.

Figure 5: Efficiency on MCYT-dp and MCYT-pb.

6 Conclusion

We discussed our first approaches, which realize an efficient identification process based on fuzzy vault protected templates. Existing approaches, so far, only support identification for unprotected templates or realize authentication systems. Our developed filter techniques determine a prioritization for the database entries, based on which a subsequent verification processes the database. Our experiments showed that the approach GeoMatch, which is translation and rotation invariant, achieves high data reduction rates, but solely the index supported approach BioSimJoin is able to perform efficiently on large databases. As future work we plan to combine the two approaches GeoMatch and BioSimJoin, to achieve a high quality and, simultaneously, a low runtime.

Acknowledgment: This work was funded by the Bundesamt für Sicherheit in der Informationstechnik (BSI) within the project BioKeyS-Testing. We want to thank all partners, namely Sebastian Abt, Christoph Busch, Heinrich Imohr, Claudia Nickel, Alexander Nouak, Alexander Opel, and Xuebing Zhou for successful discussions and their support.

References

[BBGK08] J. Breebaart, C. Busch, J. Grave, and E. Kindt. A Reference Architecture for Biometric Template Protection based on Pseudo Identities. In *BIOSIG*, pages 25–38, 2008.

[BCK09] J. Bringer, H. Chabanne, and B. Kindarji. Error-Tolerant Searchable Encryption. In *IEEE Int. Conf. on Communications*, pages 1–6, 2009.

[BFF+11] C. Böhm, I. Färber, S. Fries, U. Korte, J. Merkle, A. Oswald, T. Seidl, B. Wackersreuther, and P. Wackersreuther. Filtertechniken für geschützte biometrische Datenbanken. In *BTW*, pages 379–389, 2011.

[CCG06] S. Chikkerur, A.N. Cartwright, and V. Govindaraju. K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm. In *Proc. Int. Conf. on Biometrics*, volume 3832 of *LNCS*, pages 309–315. 2006.

[DBHO10] M. Drahanský, E. Brezinová, D. Hejtmánková, and F. Orság. Fingerprint Recognition Influenced by Skin Diseases. *Int. Journal of Bio-Science and Bio-Technology*, 3(4):11–22, 2010.

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

[JNN08] A. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Process*, 8(2):113:1–113:17, January 2008.

[JS02] A. Juels and M. Sudan. A Fuzzy Vault Scheme. *Proc. Int. Symp. on Information Theory*, page 408, 2002.

[JY00] X. Jiang and W.Y. Yau. Fingerprint Minutiae Matching Based on the Local and Global Structures. *Proc. Int. Conf. on Pattern Recognition (15th)*, vol. 2:1042–1045, 2000.

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

[MMC+02] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2002: Second Fingerprint Verification Competition. *Proc. Int. Conf. on Pattern Recognition (16th)*, 3:30811, 2002.

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

[MNS+10] J. Merkle, M. Niesing, M. Schwaiger, H. Ihmor, and U. Korte. Performance of the Fuzzy Vault for Multiple Fingerprints. In *BIOSIG*, pages 57–72, 2010.

[Mor07] R. Morelle. 3d face scans spot gene syndromes. *BBC News*, 2007.

[NJP07] K. Nandakumar, A. Jain, and S. Pankanti. Fingerprint-Based Fuzzy Vault: Implementation and Performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[OGFAS+] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro. MCYT baseline corpus: a bimodal biometric database. *IEE Proc. on Vision Image and Signal Processing*.

[RPBV00] N.K. Ratha, V.D. Pandit, R.M. Bolle, and V. Vaish. Robust fingerprint authentication using local structural similarity. In *Proc. Workshop on Applications of Computer Vision*, pages 29 –34, 2000.

[Sei06] J. Seidel. Zusatzinformationen in Fingerbildern. Master's thesis, Hochschule Darmstadt, September 2006.

[Sta02] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Pearson Education, 3rd edition, 2002.

[UPJ05] U. Uludag, S. Pankanti, and A. Jain. Fuzzy Vault for Fingerprints. In *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication (5th)*, pages 310–319. 2005.

[WCW94] C.L. Wilson, G.T. Candela, and C.I. Watson. Neural Network fingerprint classification. In *Journal of Artificial Neural Networks*, volume 1, pages 203–228, 1994.

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