# Improved Fuzzy Vault Scheme for Alignment-Free Fingerprint Features

Benjamin Tams[1], Johannes Merkle[2], Christian Rathgeb[3], Johannes Wagner[3], Ulrike Korte[4], and Christoph Busch[3]

[1]Institute for Mathematical Stochastics, University of Göttingen, Germany, btams@math.uni-goettingen.de
[2]secunet Security Networks AG, Essen, Germany, johannes.merkle@secunet.com
[3]da/sec - Biometrics and Internet Security Research Group, Hochschule Darmstadt, Germany, {christian.rathgeb,johannes.wagner,christoph.busch}@cased.de
[4]Federal Office for Information Security, Bonn, Germany, ulrike.korte@bsi.bund.de

**Abstract:** The *fuzzy vault scheme* is one of the most prominent tools for protecting fingerprint templates, typically being minutiae-based. However, there exist two major problems. Firstly, the fuzzy vault scheme is vulnerable to attacks correlating different templates of the same user. Secondly, auxiliary alignment data may leak information about the protected fingerprints which negatively affects security and privacy. In this paper, we tackle both problems. Our implementation uses alignment-free fingerprint features and fusions thereof, thereby removing the need to store alignment parameters. Furthermore, the features are passed through a quantization scheme and then dispersed in a maximal number of chaff, thereby thwarting correlation attacks.

## 1 Introduction

The *fuzzy vault scheme* [JS06] is a biometric cryptosystem considered eligible for protecting fingerprint features, where the features, typically based on *minutiae*, are hidden within a large number of randomly generated *chaff minutiae*. The eligibility of minutiae templates for being protected with the fuzzy vault scheme has been analyzed by Clancy *et al.* [CKL03], and further explored in a series of minutiae-based fuzzy vault implementations [NJP07, NNJ10, MIK+11].

There exist, however, two major problems with previous implementations. Firstly, the fuzzy vault is generally vulnerable to correlation attacks, which exploits that in two matching vault records genuine minutiae correlate well as opposed to chaff minutiae [SB07]. This property very clearly violates the *unlinkability requirement* [ISO11]. Even worse, the correlation attack allows to efficiently recover the protected feature data. This vulnerability can be avoided by rounding the minutia data to a rigid grid and use these quantizations to encode genuine vault features; chaff minutiae are encoded by all remaining unoccupied grid points. In this way, the templates contain the same set of points, precisely, the all grid points, which makes correlation attacks impossible.

Secondly, in order to successfully verify a query fingerprint, its minutiae are required to be sufficiently close to the genuine minutiae in the fuzzy vault. This may require a preliminary *alignment step* in which an accurate spatial translation and rotation of the query minutiae are achieved. Many implementations outsource the alignment problem by adjusting the query to auxiliary alignment data published along with the vaults [NJP07, NNJ10]. However, such auxiliary alignment data leak information about the protected fingerprints which conflicts with the irreversibility requirement of effective biometric information protection [ISO11, BBGK08] and may even facilitate correlation attacks.[1]

In this paper, we solve both problems. First, we use alignment-free features, which eliminates the need to store auxiliary alignment data. Secondly, we prevent correlation attacks by applying a quantization scheme to the fingerprint features and filling up the whole feature space with chaff points. Another advantage of using quantized features is the possibility of using the *improved fuzzy vault scheme* by Dodis *et al.* [DRS04] which generates significantly smaller records and can also be secured against correlation attacks [MT13].

As alignment-free features, we use *absolutely pre-aligned minutiae*, *i.e.*, minutiae represented w.r.t. a coordinate system that can be robustly estimated from the fingerprint, as well as three different local minutiae descriptors: *minutia orientation descriptors* [TK03], the *minutia frequency descriptors* [Fen08], and *local minutia structures* [JY00]. All of these minutiae descriptors have already been deployed in fuzzy vault schemes [LYC+10, NNJ10]. However, while these schemes solve the alignment problem, they do not use all unoccupied feature points as chaff and are, hence, inherently vulnerable to correlation attacks. A fuzzy vault implementation using another type of alignment-free minutiae descriptors that is also immune against correlation attacks has been presented in [BFPdS14].

The paper is outlined as follows. In Sect. 2 the alignment-free feature types tested in this paper are described. In Sect. 3 we describe the framework of our fuzzy vault. In Sect. 4 the experimental setup is described and results are reported. Finally, in Sect. 5 final discussion are given, conclusions are drawn, and outlook for future research is motivated.

## 2 Analyzed Feature Types

In this section, we outline the local minutiae descriptors deployed in our implementation, *minutia orientation descriptors*, *minutia frequency descriptors*, and *local minutia structures*, as well as appropriate distance and averaging functions used for the quantization based on a $K$-mean clustering algorithm [For65]. We also outline the *absolutely pre-aligned minutiae* used as fourth feature type, which is quantized component-wise.

### 2.1 Minutia Orientation Descriptors

*Minutia orientation descriptors* have been proposed in [TK03] and consists of local estimations of the fingerprint's orientation field sampled from locations around a reference

[1] Auxiliary alignment data could be avoided by basing the implementation on a comparison-fit approach [MIK+11]; however, this requires an existing correlation between genuine vault minutiae — exactly the property assuming vulnerability against correlation attacks.

minutia; the orientation estimations can be represented w.r.t. the orientation at the reference minutia; in this way, the descriptor is independent of the finger's rotation and translation. More specifically, according to [TK03, NNJ10, LYC+10], a minutia orientation descriptor's sample coordinates are arranged on 10, 16, 22, and 28 equidistant points lying on four concentric circles of radius 27, 46, 63, and 81 around a local coordinate system defined by the reference minutia, *i.e.*, its position defines the coordinate system's origin and its angle the direction of the coordinate system's abscissa. Consequently, an orientation descriptor is a 76-length vector with real entries each encoding an orientation measurements relative to the orientation of the reference minutia (see Fig. 1(a) for a visualization). A method for estimating an image pixel's local orientation can be found in [KW87].

**Dissimilarity Computation** We compute the difference between two orientation angles $\phi, \varphi \in [0, \pi)$ as $0.5 \cdot \mathrm{diff}(2\phi, 2\varphi)$ component-wisely where $\mathrm{diff} : [0, 2\pi) \times [0, 2\pi) \to [0, \pi)$ denotes the distance of two angles along the unit circle. In summary, given two orientation descriptors $\boldsymbol{\omega} = (\omega_1, ..., \omega_{76})$ and $\boldsymbol{\omega'} = (\omega'_1, ..., \omega'_{76})$ we may compute their distance as $\mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{\omega'}) = 1/76 \cdot \sum_{i=1}^{76} 0.5 \cdot \mathrm{diff}(2\omega_i, 2\omega'_i)$ where the normalization factor $1/76$ guarantees that the distance between two orientation descriptors lies in the interval $[0, \pi/2)$.

**Averaging** Given a set of orientation descriptors we may determine its arithmetic mean component-wise, where the average should be computed along the unit circle accounting for the fact that orientations are undirected [KW87].

### 2.2 Minutia Frequency Descriptors

*Minutia Frequency Descriptors* have been proposed in [Fen08] and represent the local inter-ridge distances at coordinates placed around the reference minutia. Precisely, a minutia's frequency descriptor is thus a 76-length vector with real positive entries (see Fig. 1(b) for a visualization). A method for estimating a fingerprint pixel's local ridge frequency estimation can be found in [Got12].

**Dissimilarity Computation** We compute the distance between two frequency descriptors, of which components consist of the inverse of local inter-ridge distance measurements, as the normalized Euclidean distance of 76-length vectors. Specifically, given two frequency descriptors $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{76})$, $\boldsymbol{\lambda'} = (\lambda'_1, ..., \lambda'_{76}) \in (0, 1]^{76}$, the distance can be computed as $\mathrm{dist}(\boldsymbol{\lambda}, \boldsymbol{\lambda'}) = 1/76 \cdot \sum_{i=1}^{76} |\lambda_i - \lambda'_i|$.

**Averaging** The mean of a set of frequency descriptors is computed by applying the harmonic mean component-wise. This corresponds to averaging the components' inter-ridge distances first and then re-obtaining the inter-ridge frequencies by inverting the results.

### 2.3 Local Minutia Structures

*Local Minutia Structures* have been proposed in [JY00] and consist of a six-length vector $(d_1, d_2, \theta_1, \theta_2, \phi_1, \phi_2)$ derived from a reference minutia and its two spatially nearest minutiae. Here $(d_1, \theta_1)$ and $(d_2, \theta_2)$ are the polar coordinates of the closest and second

(a) (b) (c)

Figure 1: *Minutia orientation descriptor* (a) and *minutia frequency descriptor* (b) consist of local orientation and ridge frequency estimations, respectively, sampled on 76 coordinates equidistantly spaced around the reference minutia. A *local minutia local structure* (c) is a six-length vector $(d_1, d_2, \theta_1, \theta_2, \phi_1, \phi_2)$ encoding the constellation of a the reference minutia and its two spatially closest neighboring minutiae.

closest minutia relative to the reference minutia; $\phi_1$ and $\phi_2$ denote the angle of the reference minutia formed with the angle of the closest and second closest minutia, respectively. For a visualization we refer to Fig. 1(c).

**Distance Computation** We adopt the similarity measure used in [LYC+10] to derive a reasonable distance function for local minutia structures. That is, given two structures $\boldsymbol{s} = (d_1, d_2, \theta_1, \theta_2, \phi_1, \phi_2)$ and $\boldsymbol{s}' = (d_1', d_2', \theta_1', \theta_2', \phi_1', \phi_2')$, we set

$$\begin{aligned} \mathrm{dist}(\boldsymbol{s}, \boldsymbol{s}') =& |d_1 - d_1'| + |d_2 - d_2'| + \\ & \frac{0.3 \cdot 180}{\pi} \cdot \left( \mathrm{diff}(\theta_1, \theta_1') + \mathrm{diff}(\theta_2, \theta_2') + \mathrm{diff}(\phi_1, \phi_1') + \mathrm{diff}(\phi_2, \phi_2') \right). \end{aligned} \tag{1}$$

**Averaging** For a set of $m$ local minutia structures $\boldsymbol{s}^{(j)} = (d_1^{(j)}, d_2^{(j)}, \theta_1^{(j)}, \theta_2^{(j)}, \phi_1^{(j)}, \phi_2^{(j)})$ with $j = 1, ..., m$, the average is computed component-wise, where the average of angles is computed along the unit circle. More specifically, we define $\mathrm{mean}(\{\boldsymbol{s}^{(j)}\}) = (\overline{d_1}, \overline{d_2}, \overline{\theta_1}, \overline{\theta_2}, \overline{\phi_1}, \overline{\phi_2})$ where, for $i = 1, 2$

$$\begin{aligned} \overline{d_i} = 1/m \cdot \sum_j d_i^{(j)}, \quad \overline{\theta_i} = \arg\left( \sum_j \left( \cos(\theta_i^{(j)}) + \sqrt{-1} \cdot \sin(\theta_i^{(j)}) \right) \right), \\ \text{and} \quad \overline{\phi_i} = \arg\left( \sum_j \left( \cos(\phi_i^{(j)}) + \sqrt{-1} \cdot \sin(\phi_i^{(j)}) \right) \right). \end{aligned} \tag{2}$$

### 2.4 Absolutely Pre-aligned Minutiae

As additional alignment-free feature we use its minutiae represented w.r.t. an *intrinsic coordinate system*. This coordinate system is derived from a robust *directed reference*

*point* estimation, *i.e.*, a position and orientation of a reference point: It's position can be used to define a coordinate system's origin while the direction defines its orientation. Further, the minutia's angle is measured relatively to the orientation of the reference point. The directed reference point estimation is taken from [TMM15]

# 3 Proposed Construction of the Improved Fuzzy Vault Scheme

## 3.1 Quantization

Let $U$ be the universe of all features of the same type (*e.g.*, minutia local structures) and let $\mathrm{dist} : U \times U \to \mathbb{R}_{\geq 0}$ be a distance function that measures the similarity between two features of $U$. Assume that we are given a system $\{u_1, ..., u_K\} \subset U$ to which we refer as the *quantization system*. Now, we may determine the quantization of an $x \in U$ by computing the index of its closest element of $\{u_1, ..., u_K\} \subset U$; this essentially corresponds to a rounding procedure. More specifically, we use the integer

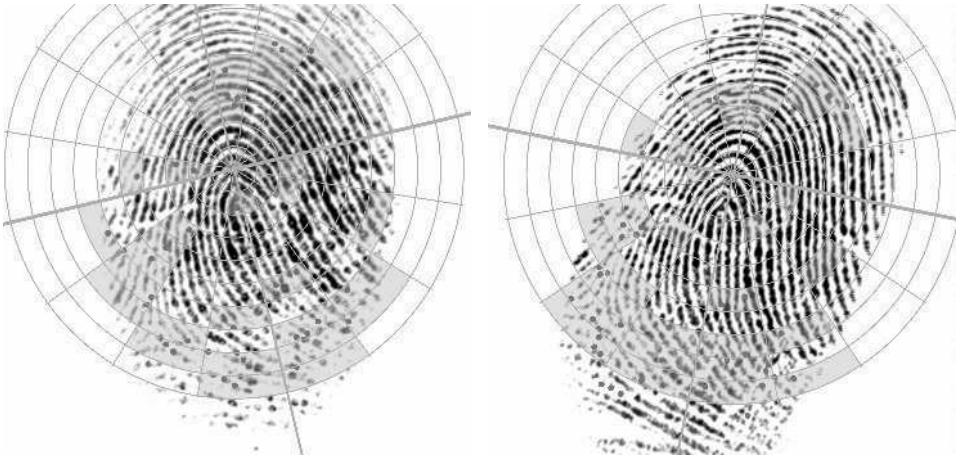
$$\mathrm{quant}(x) = \underset{i=1,...,K}{\arg\min}\, \mathrm{dist}(u_i, x) \tag{3}$$

as the quantization of $x$. Hence, we can easily compute a quantization of a feature $x \in U$ assuming that we are given a reasonable quantization system and a reasonable distance function. To establish the quantization system for a general type of features, we may perform a cluster analysis. We employ the well-known $K$-mean clustering algorithm [For65] (where $K$ is considered as a parameter) and use the final quantization system. Therefore, it is required to utilize a reasonable distance and averaging function; these have been specified for the individual feature types outlined in section 2.1, 2.2, and 2.3.

In principle, it is also possible to quantize absolutely pre-aligned minutiae with the help of a quantization system. Yet, a more direct way is to quantize minutia representations component-wisely: Given a minutia coordinate and its angle, the coordinate could be rounded to a rigid grid (*e.g.*, rectangular or hexagonal) while its angle can be quantized into a few number of partitions; such an approach has, for example, been used in [TMM15]. In this paper we consider a variation by replacing the quantization of absolutely pre-aligned minutia coordinates by the quantization of their coordinates in polar representation. More specifically, by $(\alpha, \beta)$ we denote an absolutely pre-aligned minutia's coordinate represented w.r.t. a directed reference point; this coordinate can be transformed in polar coordinate representation $(\delta, \Phi)$ where $\delta = \sqrt{\alpha^2 + \beta^2}$ and $\Phi = \arctan_2(\beta, \alpha)$. In this paper, we divide $\delta$ through a parameter $\mathsf{distQuanta} > 0$ and use its nearest integer to encode the quantization; a partition of $\mathsf{phaseQuanta}$ is used to encode the quantization of $\Phi$ (see Fig. 2 for a visualization); further, $\mathsf{angleQuanta}$ is used to quantize an absolutely pre-aligned minutia's angle.

## 3.2 Fusion

Let $U_1, ..., U_N$ be universes of different feature types. Furthermore, assume that each type is *minutia-related*, *i.e.*, its features relate to a single reference minutia. By $q_1, ..., q_N$



(a) (b)

Figure 2: Visualization of how minutiae can be quantized such that their representation is alignment-free. First, minutiae are represented with respect to a Cartesian coordinate system; then the polar coordinate representation of the minutiae's position can be quantized component-wisely; in our implementation, we also account for the minutiae' angles quantizations.

denote respectively the quantizations of $x_1, ..., x_N$ each encoded by an integer in the range $[0, K_1), ..., [0, K_N)$ (Sect. 3.1). A feature-level fusion of $q_1, ..., q_N$ can be encoded by the integer $q_1 + q_2 \cdot K_1 + ... + q_N \cdot K_1 \cdots K_{N-1}$ of the interval $[0, n)$ where $n = K_1 \cdots K_N$.

### 3.3 Fuzzy Vault System

Given a fingerprint, a set of integers $\mathbf{A} \subset \{0, ..., n-1\}$ can be extracted containing fusions of alignment-free feature quantizations. We call $\mathbf{A}$ a *feature set*. Let $\mathbf{F}$ be a finite field of size at least $n$, *i.e.*, $|\mathbf{F}| \geq n$. Then each element of $\mathbf{A}$ can be used to encode an element of $\mathbf{F}$; in this paper we do not necessarily distinguish between feature quantizations from $\mathbf{A}$ and the finite field elements encoding them.

**Enrolment** The fuzzy vault scheme can be used to store $\mathbf{A}$ in a protected way by hiding the set unless a sufficiently similar *query set* $\mathbf{B}$ is presented, *i.e.*, a set $\mathbf{B} \subset \{0, ..., n-1\}$ with $|\mathbf{A} \cap \mathbf{B}|$ being sufficiently large.

On enrolment, a cryptographic key encoded by a secret polynomial $f \in \mathbf{F}[X]$ of degree smaller than $k$ is chosen uniformly at random. In the original fuzzy vault construction [JS06], we would generate a set of genuine pairs from $\mathbf{A}$ lying on the graph of $f$ and hide them among a randomly generated set of chaff pairs containing pairs not lying on the graph of $f$. However, in order to prevent correlation attacks, we would need to use all remaining elements of the feature set $\mathbf{A}$ as chaff points, which would render the vault record very large. Therefore, we apply the improved fuzzy vault scheme by Dodis *et al.* [DRS04] where genuine and chaff points are encoded by a second polynomial. In order to thwart correlation attacks against the improved fuzzy vault [BA13], we apply a record-specific

permutation to the finite field (see [MT13] for details).

More specifically, we first compute a cryptographic hash $\mathrm{SHA}(f)$ and use it as a seed to generate a pseudo-random permutation $\sigma : \mathbf{F} \to \mathbf{F}$. Then, the polynomial $V(X) = f(X) + \prod_{a \in \mathbf{A}}(X - \sigma(a))$ is computed and the pair $(V(X), \mathrm{SHA}(f))$ is published as the vault record.

**Verification of Positive Biometric Claim** Given a vault record $(V(X), \mathrm{SHA}(f))$ and a query set $\mathbf{B}$, the verifier first reconstructs the field permutation $\sigma : \mathbf{F} \to \mathbf{F}$ from $\mathrm{SHA}(f)$ and then builds the set of unlocking pairs $\mathbf{U} = \{\ (\sigma(b), V(\sigma(b))) \mid b \in \mathbf{B}\ \}$. It is important to note that if $b \in \mathbf{A}$, then $V(\sigma(b)) = f(\sigma(b))$ and thus $(\sigma(b), V(\sigma(b)))$ is a genuine pair; otherwise, if $b \notin \mathbf{A}$, then $V(\sigma(b)) \neq f(\sigma(b))$ and thus $(\sigma(b), V(\sigma(b)))$ is a chaff pair. Consequently, $\mathbf{U}$ contains exactly $|\mathbf{A} \cap \mathbf{B}|$ genuine pairs lying on the graph of the secret polynomial $f$ which can be recovered if $|\mathbf{A} \cap \mathbf{B}|$ is sufficiently large. The correctness of $f$ can be verified using $\mathrm{SHA}(f)$.

### 3.4 Decoder

In the original version of the fuzzy vault scheme, the use of a Reed-Solomon decoder has been proposed to recover the polynomial on verification; however, extensive experimental investigations suggest that the error-correction capability of a classical Reed-Solomon decoder seems not to be able to result in acceptable verification performances for single-finger systems (*e.g.*, see [NJP07,LYC+10,TMM15]). Instead, most implementations work by iterating through all $k$-sized subsets of $\mathbf{U}$ and for each subset compute its interpolation polynomial $f^*$; if $\mathrm{SHA}(f^*) = \mathrm{SHA}(f)$, then, with very high reliability, $f^* = f$ and recovery of $f$ is considered as successful resulting in an accept decision; otherwise, if for all $\binom{|\mathbf{U}|}{k}$ iterations $\mathrm{SHA}(f^*) \neq \mathrm{SHA}(f)$, then $f$ could not be discovered resulting in a reject decision.

However, for large $|\mathbf{U}|$ the above systematic decoding approach easily becomes infeasible; for example, in [NJP07] the unlocking sets can be of size up to $|\mathbf{U}| = 24$ where $k = 9$ resulting in a worst-case running time of $\binom{24}{9} \approx 2^{20}$ polynomial interpolations. Consequently, in [TMM15] a randomized decoding approach has been proposed in which not all but at most numDecIts polynomial iterations with randomly selected $k$-sized subsets of $\mathbf{U}$ are performed. In our experiments, we strictly utilized this randomized decoding strategy, though it may be easily modified by selecting larger subsets allowing for some errors which can be corrected with a Reed-Solomon decoder; this could result in a more efficient decoder. As a reasonable choice for numDecIts we selected $2^{16}$.

## 4 Experiments

### 4.1 Quantization Systems

Using the FVC 2002 DB1 [MMC+02] we established quantization systems for the three alignment-free feature types minutia orientation descriptors OD, minutia frequency descriptors FD, and local minutia structures LMS as described in Sect. 2. For each feature type we used the first (among eight) impressions of the first 55 fingers to build a "cloud"

Table 1: Parameters selected on base of a training for different feature-level fusion strategies.

| | OD+FD+LMS | APM | APM+OD+FD+LMS |
|---|---|---|---|
| odQuanta | 31 | 1 | 5 |
| fdQuanta | 26 | 1 | 1 |
| lmsQuanta | 31 | 1 | 1 |
| distQuanta | – | 19 | 20 |
| phaseQuanta | – | 11 | 9 |
| angleQuanta | – | 10 | 8 |
| maxFeatures | 34 | | |
| numDecIts | $2^{16}$ | | |

of feature elements; then, for each $K = 1, ..., 32$ the feature cloud has been input to our $K$-mean clustering implementation which resulted in a candidate for the final $K$-sized quantization system. To assess the quality of the quantization system, the *reproducibility rate* has been determined with the help of ground-truth minutia correspondences manually marked between the first and second impressions of the remaining $55$ fingers of the FVC 2002 DB1. We repeated the $K$-mean clustering procedure $1000$ times and selected the system that resulted in the highest reproducibility rate.

### 4.2 Parameters

We performed experiments with our fuzzy vault implementation. Among the feature types OD, FD, LMS, and absolutely pre-aligned minutiae APM, we tested the following three feature-level fusions: 1. OD+FD+LMS; 2. APM only; and 3. APM+ OD+ FD+LMS. For the respective fusions, on base of a previous training, we selected the following parameters which are also listed in Tab. 1:

- odQuanta, fdQuanta, and lmsQuanta denoting the number of clusters/quantization system size for minutia orientation descriptors, minutia frequency descriptors, and local minutia structures, respectively.
- distQuanta, phaseQuanta denoting the quantization parameters for an absolutely pre-aligned minutia's coordinate in polar representation where the first and the second correspond to the distance and angular coordinate, respectively; these parameters are only relevant if the fusion contains the feature type absolutely pre-aligned minutiae (Sect. 2.4).
- angleQuanta denoting the quantization parameter for minutia angles; this parameter is only relevant if the fusion contains the feature type absolutely pre-aligned minutiae APM (Sect. 2.4).
- maxFeatures denoting the maximal number of quantized features protected by a vault; if from a fingerprint more than maxFeatures features can be extracted, those relating to the highest minutia quality are selected.
- numDecIts denoting the number of decoding iterations on verification (Sect. 3.4).

Table 2: Verification performance achievable with our fuzzy vault implementation for different feature-level fusion strategies.

| | OD+FD+LMS | | APM | | APM+OD+FD+LMS | |
|---|---|---|---|---|---|---|
| $k$ | GAR (FAR) | GDT (IDT) | GAR (FAR) | GDT (IDT) | GAR (FAR) | GDT (IDT) |
| 1 | 97% (34%) | $3ms$ ($60ms$) | 99% (92%) | $1ms$ ($7ms$) | 98% (81%) | $2ms$ ($18ms$) |
| 2 | 89% (9%) | $13ms$ ($106ms$) | 98% (78%) | $2ms$ ($25ms$) | 98% (58%) | $2ms$ ($50ms$) |
| 3 | 79% (2%) | $33ms$ ($149ms$) | 98% (60%) | $3ms$ ($60ms$) | 98% (38%) | $4ms$ ($95ms$) |
| 4 | 65% (0.3%) | $82ms$ ($202ms$) | 98% (38%) | $4ms$ ($121ms$) | 97% (21%) | $5ms$ ($163ms$) |
| 5 | 44% (0%) | $162ms$ ($257ms$) | 97% (17%) | $7ms$ ($192ms$) | 97% (7%) | $9ms$ ($231ms$) |
| 6 | 25% (0%) | $268ms$ ($331ms$) | 96% (7%) | $11ms$ ($260ms$) | 96% (2%) | $15ms$ ($303ms$) |
| 7 | 12% (0%) | $378ms$ ($416ms$) | 95% (2%) | $19ms$ ($322ms$) | 94% (0.46%) | $29ms$ ($375ms$) |
| 8 | 5% (0%) | $486ms$ ($505ms$) | 94% (0.4%) | $30ms$ ($389ms$) | 91% (0.06%) | $51ms$ ($454ms$) |
| 9 | 2% (0%) | $599ms$ ($607ms$) | 92% (0.12%) | $49ms$ ($458ms$) | 88% (0.02%) | $86ms$ ($540ms$) |
| 10 | 1% (0%) | $714ms$ ($719ms$) | 89% (0.08%) | $77ms$ ($533ms$) | 82% (0%) | $142ms$ ($633ms$) |
| 11 | 0.4% (0%) | $834ms$ ($837ms$) | 85% (0.02%) | $118ms$ ($614ms$) | 76% (0%) | $220ms$ ($733ms$) |
| 12 | 0.2% (0%) | $978ms$ ($980ms$) | 80% (0%) | $179ms$ ($704ms$) | 69% (0%) | $325ms$ ($855ms$) |

### 4.3 Evaluation

For each of the three tested fusions and the parameters determined during training, we evaluated the verification performance of our fuzzy vault implementation with the help of the optical scans of right index fingers contained in the MCYT-100 database [OGFAS03]. To measure the genuine acceptance rate GAR, we used each of the $100$ individual's $j$th scans ($j = 1, ..., 11$) to generate a fuzzy vault-protected record. The remaining scans ($j' = j + 1, ..., 12$) were used to perform a total of $11 \cdot 12/2 = 66$ genuine verification attempts per person. Consequently, we performed up to $6,600$ genuine verification attempts. To measure the false acceptance rate, for each person (labeled $i = 1, ..., 100$) we generated a fuzzy vault record using his first scan. The remaining persons' ($i' = i + 1, ..., 100$) first scans were used to perform impostor verification attempts. In such a way, we ran a total of $4,950$ impostor verification attempts. Furthermore, we kept track of the average verification times on genuine and impostor verification that we denote by GDT and IDT, respectively. The result of our evaluation, conducted on a single core of a 1.9 *GHz* server, can be found in Tab. 2.

As can be seen from Tab. 2, with a fusion of minutiae orientation descriptors, minutia frequency descriptors, and local minutia structures we reached a GAR of $44\%$ at the zero FAR for $k = 5$. In comparison, merely using absolutely pre-aligned minutiae has the capability of providing the significantly better GAR of $80\%$ at a similar FAR for $k = 12$ which can even be slightly improved when combined with minutia orientation descriptors resulting in a GAR of $82\%$ for $k = 10$. Furthermore, we found that the average decoding times can be performed within an amount of time significantly smaller than a second which makes our implementation feasible to be run in verification mode.

One may argue that, for example in [NJP07, NNJ10], a better genuine acceptance has been reached. We stress, however, that these implementations are vulnerable to cross-matching and information leakage from auxiliary alignment data. When compared with another existing implementation avoiding these problems [TMM15] in which a GAR of $79\%$ at the zero FAR has been reached, we may conclude that the performance of our

implementation is slightly better.

### 4.4 Security

A very important aspect of a fuzzy vault implementation is its resistance to recovery attacks, *i.e.*, the effort for an attacker given a vault record to recover the original feature sets or, equivalently, the secret polynomial. Generally, the fuzzy vault can be attacked by a *brute-force attack*, where the attacker repeatedly samples $k$ points from the vault and tries to interpolate the secret polynomial from these. The expected number of attempts of this attack can be estimated by combinatorial means [MMT09].

In contrast, the *false-accept attack* exploits the specific distribution of the biometric features, by repeatedly simulating (impostor) verifications using the features of randomly chosen (real) fingerprints, e.g., chosen from a biometric database [TMM15]. The success probability of the false-accept attack is equal to the FAR for the parameters used. In general, the attacker can deviate in her simulation from the parameters used in actual operation to optimize her success rate; however, in our fuzzy vault implementation, the number of decoding iterations numDecIts is the only parameter that is not already fixed in the enrolment. It has been proven in [TMM15] the expected number of decoding attempts of the false-accept attack is minimized for $\mathsf{numDecIts} = 1$. Hence, we estimate the security against false-accept attacks using this optimal strategy.

Estimating very high security levels assumes sharp estimations of FARs when they are close to zero. In biometric systems with deterministic verification algorithm, the FAR can only be estimated down to the magnitude of $1/N$, where $N$ is the number of impostor verifications performed in the evaluation. However, the verification of our implementation is probabilistic as soon as the unlocking set contains more than $k$ points. This property allows us to give heuristic estimates of FARs that are much smaller than $1/N$: For each single impostor verification, we compute the success probability based on the size of the unlocking set and the number of correct points contained, and, finally, we estimate the FAR as the mean over all verifications. For details, we refer to [TMM15].

It turns out that for the parameters chosen, the false-accept attack is much more efficient than the brute-force attack and, hence, we estimate the security against recovery attacks by the expected number of attempts required for a false-accept attack, i.e., by the reciprocal of the FAR achieved with $\mathsf{numDecIts} = 1$.[2] Fig. 3 shows a plots of the genuine acceptance rate versus the security level (depending on $k$), for different combinations of features.

Another very important security aspect concerns the risk of correlation attacks on two or more vault records of the same user. Since we use the improved fuzzy vault scheme, which effectively uses all finite field elements as vault points [DRS04], the correlation attack from [SB07] cannot be applied. On the other hand, there are specific correlation attacks against the improved fuzzy vault scheme based on solving systems of polynomial equations [BA13] or deploying the extended Euclidean algorithm [MT13]. However, these attacks only work, if in both vault records the features are represented by the same finite field elements, and, hence, are prevented by our use of a record-specific permutation $\sigma$ of

[2] This estimate is conservative insofar as we neglect the attacker's computational effort for verification.

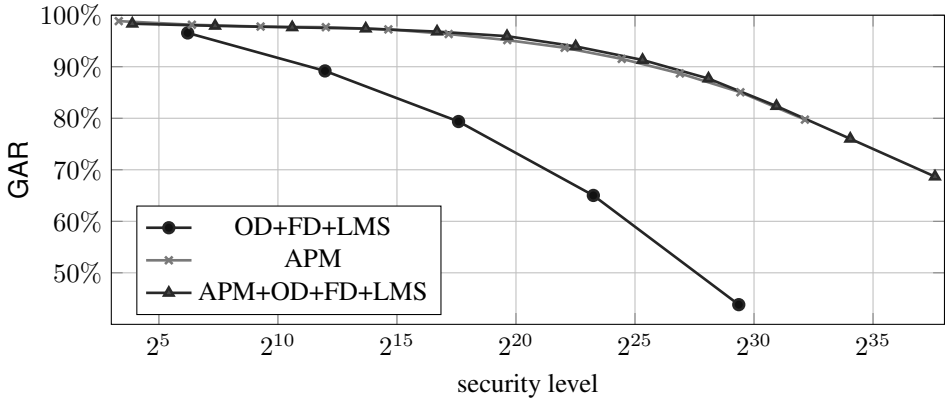



Figure 3: Genuine acceptance rate plotted versus false-accept security.

the field elements (see Sect. 3.3); for further details we refer to [TMM15].

## 5 Discussion

In this work, we designed an implementation of the improved fuzzy vault scheme for three fusions of alignment-free fingerprint feature types. We considered four different feature types one of which is given by absolutely pre-aligned minutiae. The choice of the other three types has been motivated by the work of Li *et al.* [LYC+10]; for these feature types, a generic quantization scheme based on the $K$-mean clustering algorithm is proposed in the present paper which can be padded with a maximal number of chaff in a fuzzy vault to achieve resistance against linkage attacks.

If quantizations of the feature types from Li *et al.* are fused using the techniques of this paper, we found that the achievable verification performance is clearly inferior as compared to the use of absolutely pre-aligned minutiae (see Fig. 3). Yet, our investigations indicate that, if absolutely pre-aligned minutiae are fused with other alignment-free feature types, verification performance can be slightly improved. From our experiments, we may therefore conclude that absolutely pre-aligned minutiae seem to be an indispensable feature type for the verification performance of a fingerprint-based fuzzy vault. In this view, it seems worthwhile to improve the robustness of existing directed reference point estimation methods during future research. However, this research should be conducted while having in mind that a single finger only seems not be capable of providing sufficient security.

## References

[BA13] M. Blanton and M. Aliasgari. Analysis of Reusability of Secure Sketches and Fuzzy Extractors. *IEEE Trans. Inf. Forensics Security*, 8(9):1433–1445, 2013.

[BBGK08] J. Breebart, C. Busch, J. Grave, and E. Kindt. A Reference Architecture for Biometric

Template Protection based on Pseudo Identities. In *Proc. BIOSIG*, pages 25–37, 2008.

[BFPdS14] J. Bringer, M. Favre, C. Pelle, and H. d. Saxcé. Fuzzy vault and template-level fusion applied to a binary fingerprint representation. In *BIOSIG 2014*, pages 235–242, 2014.

[CKL03] T. Charles Clancy, Negar Kiyavash, and Dennis J. Lin. Secure Smartcard-Based Fingerprint Authentication. In *Proc. ACM SIGMM workshop on Biometrics methods and applications*, pages 45–52, New York, NY, USA, 2003. ACM.

[DRS04] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *EUROCRYPT*, pages 523–540, 2004.

[Fen08] J. Feng. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 41(1):342–352, 2008.

[For65] E. W. Forgy. Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications. *Biometrics*, 21:768–769, 1965.

[Got12] C. Gottschlich. Curved Regions Based Ridge Frequency Estimation and Curved Gabor Filters for Fingerprint Image Enhancement. *IEEE Trans. Image Process.*, 21:2220–2227, 2012.

[ISO11] ISO/IEC JTC1 SC2 Security Techniques. ISO/IEC 24745:2011. Information Technology - Security Techniques - Biometric Information Protection. International Organization for Standardization, 2011.

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

[JY00] X. Jiang and W.-Y. Yau. Fingerprint minutiae matching based on the local and global structures. *Proc. Int. Conf. on Pattern Recognition ICPR*, 2:1038–1041, 2000.

[KW87] M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, 37(3):362–385, 1987.

[LYC+10] P. Li, X. Yang, K. Cao, X. Tao, R. Wang, and J. Tian. An alignment-free fingerprint cryptosystem based on fuzzy vault scheme. *J. Netw. Comput. Appl.*, 33:207–220, 2010.

[MIK+11] J. Merkle, H. Ihmor, U. Korte, M. Niesing, and M. Schwaiger. Performance of the Fuzzy Vault for Multiple Fingerprints. In *Proc. BIOSIG'11*, pages 57–72, 2011.

[MMC+02] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. FVC2002: Second Fingerprint Verification Competition. In *Proc. Int. Conf. on Pattern Recognition*, pages 811–814, 2002.

[MMT09] Preda Mihăilescu, Axel Munk, and Benjamin Tams. The Fuzzy Vault for Fingerprints is Vulnerable to Brute Force Attack. In *Proc. of BIOSIG*, pages 43–54, 2009.

[MT13] J. Merkle and B. Tams. Security of the Improved Fuzzy Vault Scheme in the Presence of Record Multiplicity. *CoRR abs/1312.5225*, 2013. available online: `http://arxiv.org/abs/1312.5225`.

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

[NNJ10] A. Nagar, K. Nandakumar, and A. K. Jain. A hybrid biometric cryptosystem for securing fingerprint minutiae templates. *Pattern Recogn. Lett.*, 31:733–741, June 2010.

[OGFAS03] J. Ortega-Garcia, J. Fierrez-Aguilar, and D. Simon *et al.* MCYT baseline corpus: a bimodal biometric database. *IEE Proc. on Vision, Image and Signal Processing*, 150(6):395–401, 2003.

[SB07] W. J. Scheirer and T. E. Boult. Cracking Fuzzy Vaults and Biometric Encryption. In *Proc. of Biometrics Symp.*, pages 1–6, 2007.

[TK03] M. Tico and P. Kuosmanen. Fingerprint Matching Using an Orientation-Based Minutia Descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):1009–1014, August 2003.

[TMM15] B. Tams, P. Mihăilescu, and A. Munk. Security Considerations in Minutiae-based Fuzzy Vaults. *IEEE Trans. Inf. Forensics Security*, 10(5):985–998, 2015.