Security Considerations in Minutiae-based Fuzzy Vaults

Benjamin Tams, Preda Mihăilescu and Axel Munk

***Abstract*—The *fuzzy vault scheme* is a cryptographic primitive that can be used to protect human fingerprint templates where stored. Analyses for most implementations account for brute-force security only. There are, however, other risks that have to be taken into account such as false-accept attacks, record multiplicity attacks, and information leakage from auxiliary data, such as alignment parameters. In fact, existing work lacks analyses of these weaknesses and are even susceptible to a variety of them. In view of these vulnerabilities, we redesign a minutiae-based fuzzy vault implementation preventing an adversary from running attacks via record multiplicity. Furthermore, we propose a mechanism for robust absolute fingerprint pre-alignment. In combination, we obtain a fingerprint-based fuzzy vault that resists known record multiplicity attacks and that does not leak information about the protected fingerprints from auxiliary alignment data. By experiments, we evaluate the performance of our security-improved implementation which, even though it has slight usability merits as compared to other minutiae-based implementations, provides improved security. However, despite heavy efforts spent in improving security, our implementation is, like all other implementations based on a single finger, subjected to a fundamental security limitation related to the false acceptance rate, *i.e.*, *false-accept attack*. Consequently, this paper supports the notion that a single finger is not sufficient to provide acceptable security. Instead, implementations for multiple finger or even multiple modalities should be deployed the security of which may be improved by the technical contributions of this paper.**

***Index Terms*—fingerprint, minutiae, fuzzy vault scheme, implementation, security, cryptanalyses**

I. Introduction

We start with a preview of the main purposes of the paper. This paper is concerned with two important aspects of the security of biometry. We use the long time investigated example of fingerprint recognition for presenting the details about this security issues. The first aspect, although apparently simple and qualitatively known, has not been taken into account with sufficient consistency so far. It concerns the limitations of fingerprint security. This is due to two major factors:

A. Unlike passwords, biometry is irreplaceable. If it has been once cracked on any system, it is insecure for any further applications. The biometric community faces comparably more difficult problems than the cryptographic one, which has developed during a period of more than one to two decades of intensive academic work a set of reliable and well defined attack scenarios. These are a common base both for cryptologic research and for security assessments and standardisation work. In cryptography, the identification process is based on deterministic primitive, which consistently yield the same output when presented the same input. In biometry however, identification is based on images and visual data which are prone to slight fluctuations, and thus statistical in nature. Therefore, presenting the *same input*, meaning the biometry of one and the same person, will not necessarily result with the extraction of the same identification data: this data is influenced by many, mostly external, physical factors. As a consequence, except for few marginal scenarios, one of which we mention below, we have hardly any investigated attack definitions for biometric security. In particular, the **consequences of the compromise of an individual trait for the security of the respective individual** is very poorly treated. Since the probability for such a compromise is quite high, it is a challenge, that structured attack scenarios should be defined and investigated in the near future in this community.

B. Primitives like *fuzzy vaults* or *fuzzy sketches* rely on an abstract notion of entropy, which was expected by the computer scientists who initially designed this general purpose primitives, to be *high, or at least sufficient*. However, in the context of fingerprints, the statistical character of matching implies that entropy cannot be considered to be more than a *metaphor* that describes intuitively the amount of specific information which can actually be used in repeated matching attempts for the purpose of authentication or identification. This amount is usually quite low for a fingerprint. We argue that the security — and thus the "de facto entropy" — is strictly correlated to the false accept error probability FAR. It is in fact essentially equal to its inverse $S = 1/\mathrm{FAR}$. Using an open database we also provide empirical evidence for the feasibility of false-accept attacks that require the expected amount of S attempts. In particular, this realistic measure is severely lower than figures one encounters in the literature. These are mostly derived by theoretical estimates drawn from models about the security of fuzzy vaults based on fingerprints and scenarios that overlook the possibility of direct false-accept attacks. We therefore urgently recommend that secure applications of fingerprint recognition should migrate to (at least) five finger recognition. This is not a technological challenge, since scanners for simultaneous scanning of five fingers are already on the market — but the academic community is called to insist on their relevance for security. The first

B. Tams and A. Munk are with the Institute for Mathematical Stochastics, University of Goettingen, Goldschmidtstr. 7, 37077, Goettingen, Germany. Emails: { btams , munk }@math.uni-goettingen.de. B. Tams and A. Munk gratefully acknowledge support of the DFG Graduiertenkolleg 1023, the Felix Bernstein Institute for Mathematical Statistics in the Biosciences and the Volkswagen Foundation.

P. Mihăilescu is with the Mathematical Institute, University of Goettingen, Bunsenstr. 3-5, 37073, Goettingen, Germany. Email: preda@uni-math.gwdg.de.

important attack that multi-modal biometry has to be resistant against is the *uncoupling attack*. Suppose that recognition is based on the biometries A and B, where A and B can for instance stand for two different fingers, or for iris and face, etc. It should be infeasible to break a fuzzy vault based on information from A and B into two separate vaults, related separately to A, respectively B. If the separation is achieved, we say that the biometries are uncoupled, and the complexity of an attack for breaking the vault becomes now the *sum* of the complexities for breaking the vaults A and B individually. This is no increase in security, and one expects that, by resisting the uncoupling attack, the single complexities are multiplied rather than added: in other words, if the security of A is $e(A)$ bits and the one of B is of $e(B)$, bits, by avoiding uncoupling, one expects a security of $e(A)+e(B)$, whereas in case the separation is successful, one does not have more than $\max(e(A), e(B))+1$ bits. Therefore, additional care needs to be invested in the algorithms, in order to avoid the possibility of uncoupling attacks.

The urgency of addressing the above limitations is impressively confirmed by the recent breaking of Apple's iPhone fingerprint protection.[1] The attack was made public only few weeks after the announcement of a prize for the breaking of this protection — an event without precedent in the cryptographical branch of security. The attack occurred after the conception of this paper and strongly supports some of our central claims.

Awareness of the above limitations has resulted in the use of the vague term *privacy enhancement* in connection with biometric methods — a term intended to suggest a situation in which there is no guarantee of security, but which is better than no security at all. At a time when biometric security applications are spreading, and numerous countries have started to store fingerprints on reduced function devices (RFDs) embedded in passports and consular personnel are trained to suggest to the citizen that this data uniquely identifies a person, it should be a duty of the academic community to provide serious facts, measures and caveats about the consequences of the known but insufficiently investigated limitations.

While the first part of our contribution calls for a long term migration to system with sufficient security, the second part is concerned with short and middle term applications. We give here a novel algorithm for protecting fingerprint minutiae that is resistant to one of the best discussed scenarios in this domain, linkability attacks in general and correlations attacks in particular. The algorithm is proved to avoid correlation attacks, while providing verification performance which is well-comparable to state-of-the-art implementations of fuzzy vault for fingerprints, which are found in the recent literature, and which are, however, prone to correlation attacks and other vulnerabilities.

A. Overview

The protection of strong user-specific *passwords/keys* on a *server/token* (*e.g.*, via cryptographic hash functions) is well understood and can provide nearly cryptographic security in widely accepted models [1]. On the other hand, secure passwords are hard to remember and may result in a typical user choosing weak passwords or writing them down which reduces the system security. A possible solution to this well-known vulnerability is to replace passwords by measurements of biometrics, such as *fingerprints* or *irises*, which can improve on effective protection achievable with weak passwords: Biometric measurements contain a certain amount of information that do not depend on the owner's ability to memorize passwords. *Authentication* based on biometric measurements requires them to be stored on a server or a token (*e.g.*, a *smart card*). Since biometric measurements may contain sensitive information of which knowledge can threaten the system users' privacy, the biometric measurements must be stored protected on the system's database. Requirements on these so-called *renewable biometric references* (RBR) are motivated in [2]. The most important properties that a valid implementation of an RBR has to provide are *usability* and *security*. The former accounts for the user's convenience including the rate at which an authorized user is accepted on *verification*. The latter is broken down into further requirements as *irreversibility* (*i.e.*, the biometric measurement should not be derivable from an RBR) and *privacy* (most notably the *unlinkability requirement* to prevent the identification of related users across different application's databases, *i.e.*, *cross-matching*).

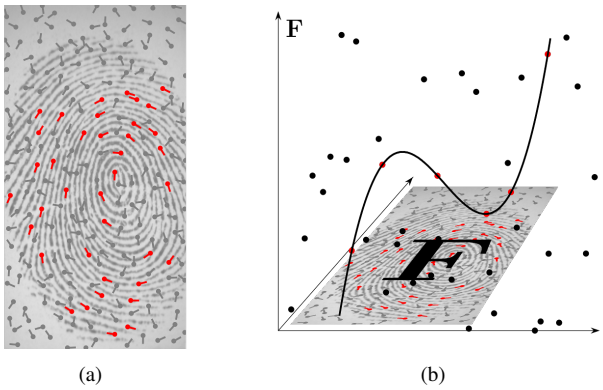
Ideally, the difficulty of defeating irreversibility and privacy of an RBR should be as hard as breaking traditional cryptographic systems (*e.g.*, inverting cryptographic hash values of strong passwords). It is, however, a commonly accepted notion that RBRs *"cannot have the same level of security as cryptographic algorithms"*; a principal motivation for RBRs is to *"replace the vulnerable password-based schemes with more secure and more convenient biometrically managed keys"* (see Section 26.3.3 in [3]). On the other hand, passwords can be replaced nearly arbitrarily many times as compared to biometric measurements and, thus, once an RBR is compromised, its corresponding biometric trait cannot be safely reused. In view of the aforementioned constraints and facts, it is vital to have best practices in implementing and analyzing RBRs to ensure that they are only deployed if they provide *sufficient security*, a bound that has not yet been specified by the community.

In this paper, we focus on RBRs for *fingerprints*, one of the most dominant biometric modalities. We next review approaches that can help in implementing them.

B. Fingerprint Protection Schemes

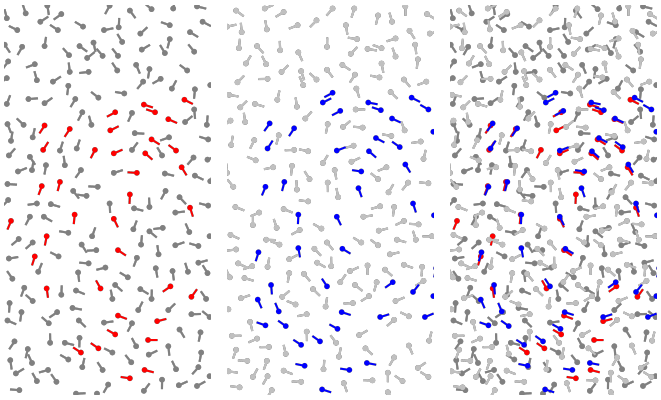
Pioneering work in fingerprint biometrics was done by Tomko *et al.* in 1994 [4], however, usability and security issues were quickly discovered. In 2003, Clancy *et al.* [5] proposed to use the *fuzzy vault scheme* by Juels and Sudan [6], [7] to protect the positions of fingerprint minutiae. Its functioning can be outlined as follows. On *enrollment*, given t minutiae, their positions are encoded as elements x in a fixed finite field $\mathbf{F}$. There is a one-to-one correspondence between minutiae and finite field elements encoding them. A secret polynomial $f \in \mathbf{F}[X]$ in the indeterminate X of degree smaller than k is

[1] *e.g.*, see http://www.telegraph.co.uk/technology/apple/iphone/10327635/iPhone-5s-fingerprint-sensor-hacked-within-days-of-launch.html



(a) (b)

Fig. 1. (a) Genuine (red) and chaff minutiae (gray); (b) each minutia is encoded on a vault point's abscissa where its ordinate binds the minutia to the secret polynomial



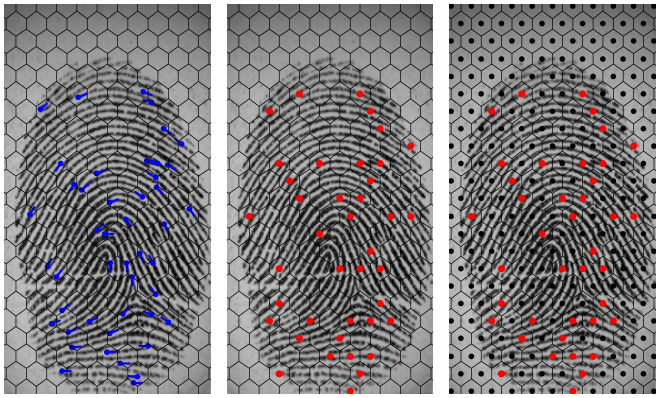
(a) (b) (c)

Fig. 2. Visualization of the correlation attack: Two vaults (a), (b) with chaff minutiae (gray and light-gray) and genuine minutiae (red and blue). (c) The genuine minutiae have a bias to be in agreement.

generated at random and the evaluations $f(x)$ at the minutiae encoded as $x \in \mathbf{F}$ are computed. The *genuine pairs* $(x, f(x))$ are dispersed among a large set of *chaff pairs* not lying on the graph of f, so that a vault of size n is built. On *verification*, using a second (allegedly genuine) query minutiae template, one aims at distinguishing the genuine pairs from the chaff pairs by extracting those vault pairs (called *unlocking pairs*) that correspond to vault minutiae being well approximated by a query minutia. If the unlocking pairs are mostly genuine, one can tolerate errors within certain limits determined by *Reed-Solomon codes* [8]. The fuzzy fingerprint vault draws its security from the difficulty of the problem of distinguishing genuine from chaff (without the help of a second genuine template). This problem can be reduced by the *polynomial reconstruction problem* which is believed to be hard in general if $t \ll \sqrt{(k-1) \cdot n}$ [9]–[13].

Several variants of a minutiae-based fuzzy vault have been implemented [14]–[19] most of which lack resistance to brute-force attack [20], thus conflicting with the irreversibility requirement. As a countermeasure, Nagar *et al.* [18] proposed to incorporate *minutiae descriptors* to protect the vault pair's ordinate values via the *fuzzy commitment scheme* [21]. If carefully implemented, the hybrid implementation can effectively provide improved security against brute-force attacks. However, the practicability of a *dictionary attack* with fingerprints as keys, *i.e.*, *false-accept attack*, remains a serious problem (*e.g.*, see [22]–[25]). Implementations for multiple fingers (or even multiple biometric modalities) may be a valid solution to effectively improve resistance against false-accept attacks at a useful genuine acceptance rate (*e.g.*, see [26]).

Even if *false-accept security* of a fuzzy vault to multiple fingerprints is within acceptable limits, there remains the possibility for an adversary to run *attacks via record multiplicity* [27] to find related record correspondences across different applications' databases, *i.e.*, *cross-matching*, conflicting with the unlinkability requirement, or, even worse, to break multiple related records conflicting with the irreversibility requirement. In 2008, Kholmatov and Yanikoglu [28] demonstrated the practicability of the *correlation attack* in which an attacker aims to differentiate genuine pairs from chaff pairs by correlating the (genuine and chaff) features/minutiae of two (or more) related vaults (see Figure 2 for a visualization). In particular, the authors showed that it is possible to break two matching vault correspondences with a reasonably high probability. Furthermore, some minutiae-based fuzzy vaults are attached with public auxiliary data to support aligning query minutiae templates on verification [15], [16], [29]; this extra information may even ease cross-matching.



(a) (b) (c)

Fig. 3. Visualization of an approach to prevent the correlation attack: The minutiae (blue) are rounded to hexagonal grid points (red) encoding genuine vault pairs (a), (b); each unoccupied grid point (black) is used to encode a chaff pair (c).

To avoid the use of auxiliary alignment data, Li *et al.* (2010) [19] extracted features invariant to the fingerprint's translation and rotation. While the verification performances the authors report look promising and the error-prone step of aligning the query fingerprints to the vault is circumvented, the authors do not provide an analysis against correlation attacks to fulfill the unlinkability requirement. One may argue that a user password can be used to mitigate remaining vulnerabilities including the risk of correlation attack-based cross-matching (*e.g.*, see [17]). On the other hand, the use of passwords requires the users to remember them and may result in usability problems similar to those of mere password-based verification that were meant

to be resolved with biometry.

There are other *biometric template protection schemes* than fuzzy vault against which correlation attacks cannot be applied. These are, for example, the *fuzzy commitment scheme* [21] or the more general *fuzzy sketch* [30], [31]. The fuzzy commitment scheme (or the related *pin sketch* [30], [31]) requires that the fingerprint template can be represented as a (binary) feature vector of (pre-defined) fixed length (*e.g.*, see [32]).

Probably, the easiest way to quantize a fingerprint minutiae template as a binary fixed-length feature vector, is to lay a grid on the fingerprint image in which each grid point encodes a position in the feature vector; if a minutia position can be rounded to a grid point, the entry at its corresponding position of the feature vector is set to 1; for each unoccupied grid point its corresponding entry is set to 0. Another approach to encode a minutiae template as a binary fixed-length feature vector is to quantize the minutiae with the help of their *spectral representation* through Fourier transforms which can even circumvent problems related with aligning query fingerprints in the spatial domain [33], [34]. Also, we could consider the use of *distributed source coding* techniques [35] or *spin glass* constructions [36] in combination with binary feature vectors.

C. Motivation

If a quantization scheme is applied to, say, fingerprint minutiae, *i.e.*, if the minutiae can be robustly represented, the fuzzy vault scheme can be implemented such that it becomes resistant against correlation attacks, too. For example, points of a rigid grid to which a minutia is rounded can be used to encode a genuine vault pair while all other grid points are used to encode a chaff pair. In such a way any correlation between genuine features is avoided since two (or more) vault features (*i.e.*, the union of genuine and chaff features) are equal. Furthermore, the improved fuzzy vault scheme by Dodis *et al.* [30], [31] can be used to significantly reduce the data size consumed by the vault records. On the other hand, there exists an efficient attack against multiple records of the improved fuzzy vault scheme using the *extended Euclidean algorithm* [37] conflicting with both the unlinkability and irreversibility requirement. Fortunately, re-ordering the minutiae' encodings for each record, similar to a *password salt*, may effectively prevent the record multiplicity attack [37]. If, in addition, fingerprints can be robustly pre-aligned w.r.t. an *intrinsic coordinate system*, the basis for an unlinkable minutiae-based fuzzy vault has been laid. Despite its conceptual simplicity, such an approach has not yet been well investigated.

One may argue that if a quantization scheme is applied to the minutiae of absolutely pre-aligned fingerprints, the fuzzy commitment scheme (or *pin sketch* [30], [31]) can be used for template protection instead of the fuzzy vault scheme: The elements of the universe of feature elements are successively labeled with an index, and if a minutiae quantizes as a feature element, its position in a fixed-length feature vector is set to 1 and, otherwise, left unchanged as 0; this relation has been already pointed out in [30], [31]. However, it is important to note that in a binary fuzzy commitment scheme the problem of cross-matching cannot be avoided when based on a linear code [38], [39] unless the parameters are chosen very carefully—even though there has been an attempt to prevent fuzzy commitment schemes from being vulnerable to cross-matching [40]. This already is a substantiated reason for preferring the improved fuzzy vault scheme; for more details, we refer to [39]. Furthermore, in a fuzzy commitment scheme it is necessary for the underlying error-correcting code to match the length of the feature vectors. However, usable error-correcting codes typically require the feature vectors to be of a length of a special form, *e.g.*, $2^m - 1$ for non-trivial *BCH codes* [41]. This is typically solved by dividing the feature vectors into chunks matching the length of usable error correcting codes, and to implement a multi-layer error correction technique [34], [42]. Furthermore, to allow toleration of missing areas of the query template, the decoder should be able to effectively tolerate erasures which can be achieved with Reed-Solomon codes [42]. On the other hand, the use of chunks enables an attacker to run score-based attacks or attacks using the error correcting code's outputs statistics [22], which are vulnerabilities that do not apply to an implementation that merely utilizes a fuzzy vault scheme with a maximal number of chaff pairs. In the authors' view, the use of a fuzzy vault scheme is conceptually simpler as compared to taking measures to make the fuzzy commitment scheme practical—especially in view of the fact that this causes additional security issues. In this paper we focus on the possibility in implementing minutiae-based template protection via fuzzy vault.

D. Contribution and Outline of the Paper

In Section II, we describe an implementation for generating protected fingerprint templates via fuzzy vault, *i.e.*, *vault records*, from absolutely pre-aligned minutiae templates as candidates for an RBR. To achieve resistance against the correlation attack (in order to match the unlinkability requirement), we apply a quantization process to each minutia by rounding them to a hexagonal grid as, for example, discussed in [20]; the minutiae angles are quantized as well. This enables the use of the improved fuzzy vault scheme which has the positive effect that the size of the vault records reduces drastically. In order to make verification feasible, we propose a randomized decoding procedure.

In Section III, to avoid the use of auxiliary alignment data, we present a method for absolute fingerprint pre-alignment, a problem for which no definite solution has been found before [43]. In Section IV, we experimentally demonstrate the practicability of our minutiae-based fuzzy vault in combination with our method for absolute fingerprint pre-alignment.

In Section V, we give a detailed analysis for our implementation's security including a serious treatment of the false-accept attack which is, to the best of the authors' knowledge, missing for other existing implementation of the fuzzy fingerprint vault. We prove that our implementation is secure against the specific correlation attack [28] and, furthermore, show that our implementation can effectively be secured against other known record multiplicity attacks to the improved fuzzy vault scheme [37], [44] by incorporating a public random permutation process, thereby adopting the idea from [40].

In Section VI, we compare the verification performance and security with other implementations from the literature and give a conclusion as well as an outlook for future research. Even though there is a degradation in verification performance, our implementation resists a significantly larger variety of attacks and consumes significantly less memory as compared to other implementations of the fuzzy fingerprint vault.

Finally, it seems worth noting that our implementations can be downloaded in form of an open-source C++ library.[2] The library includes an implementation of our method for absolute fingerprint pre-alignment and of our minutiae-based fuzzy vault coming with several optional security enhancing features including an interface for user password combination; details can be found on-line in the documentation and/or source code.

II. Minutiae-Based Fuzzy Vault System

In this section, we describe the functioning of our minutiae-based fuzzy vault implementation. Given an (absolutely pre-aligned) minutiae template, each of its minutia is passed through a quantization process first.

A. Minutia Quantization

Let $\mathfrak{m} = (a, b, \theta)$ be a minutia at (a, b) and $\theta \in [0, 2\pi)$ its angle, and let $\Lambda_0, \ldots, \Lambda_{r-1}$ be a fixed system of r (hexagonal) grid points covering the region in which an absolutely pre-aligned minutia can occur. Now, let Λ_j be a point that best approximates (a, b); furthermore, let $j' = \lfloor \theta/(2\pi) \cdot s \rfloor$ where s is a parameter controlling the number of quanta used to quantize minutiae angles. A quantized minutia is encoded by the integer $j + r \cdot j'$ which in turn encodes (by some fixed convention) a finite field element $x_{j,j'} \in \mathbf{F}$. We use the field element $x_{j,j'}$ to represent the quantization of the minutia $\mathfrak{m}$.

Note that the feature universe in which a minutia's quantization can occur equals $\mathbf{E} = \{\ x_{j,j'} \mid j = 0, \ldots, r-1,\ j' = 0, \ldots, s-1\ \}$. On constructing the vault, the chaff is not generated randomly but each element from $\mathbf{E}$ not encoding a quantized minutia is used to encode a chaff. In such a way, the union of genuine and chaff features between multiple vault records are equal thus preventing an attacker from gaining advantage from any correlation between two vault's features.

B. Enrollment

On enrollment, the user provides a template containing the minutiae $\mathfrak{m}_1, \mathfrak{m}_2, \ldots$ for which we assume that they are sorted decreasingly w.r.t. their quality and that they are already absolutely pre-aligned. Let $t_{\max}$ be an upper bound on the number of feature elements used to encode genuine vault features, and let $\mathbf{A}$ be the maximal subset of $\mathbf{F}$ of the first minutiae quantization such that $|\mathbf{A}| \leq t_{\max}$. We refer to $\mathbf{A}$ as the *feature set*.

The next step is to bind the feature set $\mathbf{A}$ to a secret polynomial $f \in \mathbf{F}[X]$ of degree less than k. This can be done as usual by letting the genuine set $\mathbf{G} = \{\ (x, f(x)) \mid x \in \mathbf{A}\ \}$, generating the chaff set $\mathbf{C} = \{\ (x, y) \mid x \in \mathbf{E} \setminus \mathbf{A},\ y \neq f(x)\ \}$ (with y random), and publishing $\mathbf{V} = \mathbf{G} \cup \mathbf{C}$. This, however, may result in quite a large amount of data consumed by the vault records. Alternatively, we may use the improved fuzzy vault scheme by Dodis *et al.* [30] in which both chaff and genuine pairs are encoded by a monic polynomial of degree $t = |\mathbf{A}|$. An instance of the improved fuzzy vault scheme given the feature set $\mathbf{A}$ and a secret polynomial f is easily generated as $V(X) = f(X) + \prod_{x \in \mathbf{A}} (X - x)$.

If $x \in \mathbf{A}$, then $V(x) = f(x)$ and, thus, $(x, V(x))$ is a genuine pair; otherwise, if $x \notin \mathbf{A}$, then $V(x) \neq f(x)$ and, hence, $(x, V(x))$ is a chaff pair. The monic polynomial V, which is of degree t, encodes the vault instance $\mathbf{V} = \mathbf{G} \cup \mathbf{C}$ hiding the genuine pairs $\mathbf{G} = \{\ (x, V(x)) \mid x \in \mathbf{A}\ \}$ within the chaff $\mathbf{C} = \{\ (x, V(x)) \mid x \notin \mathbf{A}\ \}$ but requiring significantly less memory; for further details we refer to [30].

In addition to the polynomial $V(X)$, we assume that a cryptographic hash value of f is published as part of the vault record. Thus, the candidate for an RBR generated on enrollment is the pair $(V(X), \mathrm{SHA}(f))$ where $\mathrm{SHA}(f)$ can be considered as the RBR's *pseudonymous identifier* and $V(X)$ as the RBR's *auxiliary data* (not to be confused with *auxiliary alignment data*) [2].

C. Verification

On verification, we assume that the vault record $(V(X), \mathrm{SHA}(f))$ protecting the feature set $\mathbf{A}$ is given and that a *query feature set* $\mathbf{B} \subset \mathbf{F}$ has been generated from the minutiae template of a claimant in the same way as the protected feature set $\mathbf{A}$ has been generated from the (alleged same) enrolled user. The verifier computes the *unlocking pairs* as $\mathbf{U} = \{\ (x, V(x)) \mid x \in \mathbf{B}\ \}$. $\mathbf{U}$ consists of exactly $\omega = |\mathbf{A} \cap \mathbf{B}|$ genuine pairs, *i.e.*, pairs that lie on the graph of the secret polynomial f, and, if ω is reasonably large, the polynomial f can be recovered using an algorithm for decoding Reed-Solomon codes which is considered as an *accept* decision, and a *reject* decision otherwise.

D. Parameter Configuration

The verification performance and security of our minutiae-based fuzzy vault can be controlled by the following parameters:

- the equidistant spacing between the coordinates of the hexagonal grid ℓ, which controls the number of grid pairs r within the region in which absolutely pre-aligned minutiae can occur;
- the number of values s into which the minutiae angles are quantized;
- the bound $t_{\max}$ on the number of genuine vault pairs;
- and the length k of the secret polynomial.

We performed a systematic test to determine a good configuration for the above parameters. For each configuration, we estimated the GAR and the FAR using minutiae templates estimated from the FVC 2002 DB2-B (which is intended for training purposes) following an adoption of the FVC protocol [45], thus, yielding 280 genuine verification attempts and 45 impostor verification attempts. On genuine verification, to simulate that the query features are well aligned to the vault features we decoupled the alignment problem from the vault.

[2] http://www.stochastik.math.uni-goettingen.de/biometrics/thimble

The configuration $\ell = 29,\ s = 6,$ and $t_{\max} = 44$ was found to yield 100% genuine feature set correspondences sharing at least 7 common elements while all non-matching feature set correspondence agreed in less than 7 elements, thereby suggesting to choose $k = 7$ as a minimal size of the length of the secret polynomials. Since the number of hexagonal grid pairs fitting in the fingerprint image's dimension 296×560 is $r' = 242$ and since the minutiae angles can fall into $s = 6$ possible quanta, the vault is of size $n' = 1452$. Thus, as *brute-force security* of a fuzzy fingerprint vault is commonly defined as

$$\mathbf{bf}(n', t_{\max}, k) = \binom{n'}{k} \cdot \binom{t_{\max}}{k}^{-1} \tag{1}$$

(*e.g.*, see [20]), for $k = 7$ we obtain a brute-force security of $\mathbf{bf}(1452, 44, 7) \approx 2^{36}$. If a higher security is sought, we may choose a higher k.

At this point, we stress that the decoupling of the alignment from the vault has been performed only for training purposes in order to select a reasonable parameter configuration. In Section IV-B we describe the evaluation of our implementation for absolute pre-aligned minutiae. Furthermore, the number of hexagonal grid points equidistantly arranged with distance $\ell = 29$ pixels fitting in the region in which absolutely pre-aligned minutiae can occur is $r = 1733$ thus yielding a vault of size $n = 10398$; but, in order to estimate brute-force securities, it is safer to assume that minutiae occur in a region of dimension 296×560 and therefore to assume that the vault is of size $n' = 1452$.

E. Vault Record Size

On enrollment, a monic polynomial $V(X)$ of degree at most $t_{\max} = 44$ with coefficients in a finite field $\mathbf{F}$ is generated. Therefore, at most $t_{\max} \cdot \log_2(\mathbf{F})$ bits are required to store $V(X)$. Since $|\mathbf{F}| \geq n = 10398$ must be fulfilled to uniquely encode each minutia quantization by a finite field element, we can choose $\mathbf{F} = \mathbb{F}_{2^{16}}$ as the underlying finite field which results in $44 \cdot 16$ bits required to encode $V(X)$. The hash value of the secret polynomial may consume additional 160 bits (*e.g.*, via the *secure hash algorithm* [46]). Consequently, a vault record generated by our implementation may require $44 \cdot 16 + 160 = 864$ bits or, equivalently, 108 bytes for storage.

In Section V-E we discuss that a public random permutation process for re-ordering the feature elements should be associated with each record to prevent successful application of record multiplicity attacks against the improved fuzzy vault scheme [37], [40], [44]. The (pseudo-)random permutation process can be encoded by a random public seed which can (say) be derived from the hash value $\mathrm{SHA}(f)$.

F. Randomized Decoder

On verification, an unlocking set $\mathbf{U}$ is computed containing $t \leq t_{\max} = 44$ unlocking pairs. If $\mathbf{U}$ contains at least $(t + k)/2$ genuine unlocking pairs, the secret polynomial f can be recovered using a Reed-Solomon decoder [41]. This, however, requires the unlocking sets to contain more than 50% genuine pairs, and it has been pointed out in [14] that this does not appear to be realistic for fingerprint minutiae. Therefore, in [14] an approach has been proposed in which the unlocking set is decoded by iterating through all polynomials interpolating k unlocking pairs. If $\mathbf{U}$ contains k genuine pairs, the correct polynomial f can be recovered whose correctness can, for example, be verified with the help of a cryptographic hash value of the correct polynomial (or via a CRC added to f). This *systematic decoder* has been adopted for the vast majority of fuzzy fingerprint vault implementations [15]–[19]. For our implementation, however, the use of a systematic decoder is problematic because the unlocking sets can be quite large. For example, if a brute-force security at least 2^{40} is sought, we may choose $k = 8$. Then, as an unlocking set can be of size up to $t_{\max} = 44$, up to $\binom{44}{8} \approx 2^{27}$ polynomial iterations have to be performed before an authenticating user is accepted or rejected. This is too expensive for a usable system.

To make the verification process practical, we may consider the possibility in randomizing the systematic decoding approach. Given a set of t unlocking pairs $\mathbf{U}$, we iterate through a certain number $\mathcal{D}$ of candidate polynomials each interpolating k different unlocking pairs being selected randomly in each iteration. Consequently, if $\omega = |\mathbf{A} \cap \mathbf{B}| \geq k$, our *randomized decoder* will successfully recover the correct polynomial f with probability $1 - (1 - \mathbf{bf}(t, \omega, k)^{-1})^{\mathcal{D}}$ which approaches 100% as $\mathcal{D} \to \infty$; otherwise, if $|\mathbf{A} \cap \mathbf{B}| < k$, our randomized decoder will, as the systematic decoder, fail in recovering f. Later in Section IV-B, we demonstrate the practicability of the randomized decoder for $\mathcal{D} = 2^{16}$ experimentally.

III. Absolute Fingerprint Pre-Alignment

For our minutiae-based fuzzy vault implementation, we assume that the minutiae templates can be absolutely pre-aligned. While there are proposals for coarse absolute pre-alignment [26], the quality of the pre-alignment should, however, be of reasonable robustness in order to work well with our implementation. Given a reasonably robust method for estimating a fingerprint's *intrinsic coordinate system*, we may represent a template's minutiae w.r.t. the system solving the absolute pre-alignment problem. There are approaches for estimating a fingerprint's intrinsic coordinate system, but their respective applicability is based on assumptions for which no satisfactory solution yet exists (*e.g.*, partitioning a fingerprint into *regular regions* [47]) or that cannot be guaranteed in practice (*e.g.*, require the estimation of each fingerprint's *core* and *delta* [48]).

There is an evident one-to-one correspondence between a Cartesian coordinate system and a reference point being constituted with a direction: The reference point is used as the coordinate system's origin and the direction defines the ordinate axis (or the abscissa axis depending on the chosen convention). If a method is available that robustly estimates a *directed reference point* from a fingerprint, we can use it to represent the minutiae w.r.t. the resulting Cartesian coordinate system using well-known techniques from linear algebra. While there are methods that can extract robust fingerprint reference points (*e.g.*, the *core* [49] or *focal point* [50]), they are, however, not constituted with a direction,

but a robust method for estimating a direction is required as well. In this section, we present an approach to estimate a fingerprint's directed reference point; the method has been introduced recently in [51]. Its practicability, in combination with our implementation of a minutiae-based fuzzy vault, is demonstrated in Section IV-B.

Throughout this section, we represent two-dimensional coordinates (a, b) as complex numbers $a + i \cdot b$.

A. Preliminaries and Outline

Our method for estimating a fingerprint's directed reference point aims at estimating a coordinate on the fingerprint in a region where the orientation field locally looks like the orientation field near the core of a tented arch. To model the orientation field of a tented arch, we use the *quadratic differential model* presented in [52]. We aim at finding the rotation and translation of a *tented arch model* such that it well approximates an orientation field estimated from the fingerprint. Since the tented arch model features a longitudinal axis and a core, the core of a fitted tented arch model can be used as the reference point's coordinate and the direction of the, say, longitudinal axis serves as the reference point's direction.

The Tented Arch Model: The orientation field of a tented arch essentially is the orientation field of an arch whose flow is influenced by a core and a delta placed on the model's longitudinal axis. The orientation field of an arch can be modeled as the complex function $\psi(z) = \lambda^2 \cdot (z^2 - R^2)^2$ where $\mathbf{Im}(z) > 0$ and λ, R are real parameters. The undirected orientation $\varphi \in [0, \pi)$ at (a, b) where $b > 0$ fulfills $\varphi = 0.5 \cdot \mathbf{Arg}(\psi(a + i \cdot b))$. Given the distance of a core d_{core} and the one of a delta d_{delta}, where $0 \leq d_{\mathrm{delta}} \leq d_{\mathrm{core}}$, we model the orientation field of a tented arch as the complex function $\tau(z) = \psi(z) \cdot \frac{z^2 + {d_{\mathrm{core}}}^2}{z^2 + {d_{\mathrm{delta}}}^2}$. For more details on the quadratic differential model we refer to [52].

As our method works by finding a spatial movement of a tented arch model $\tau(z)$ to well approximate a fingerprint's orientation field estimation, we need to model $\tau(z)$ being moved by an isometry $\alpha \cdot z + \beta$ with complex α, β where $|\alpha| = 1$. We obtain such a model $\tau_{\alpha,\beta}(z)$ by plugging $\alpha \cdot z + \beta$ and correct for the rotation by multiplication with α^{-2}, *i.e.*,

$$\tau_{\alpha,\beta}(z) = \alpha^{-2} \cdot \tau(\alpha \cdot z + \beta). \tag{2}$$

By $\gamma_{\alpha,\beta}$ we denote the complex position of the core of $\tau_{\alpha,\beta}(z)$ which is given by

$$\gamma_{\alpha,\beta} = \alpha^{-1} \cdot (i \cdot d_{\mathrm{core}} - \beta). \tag{3}$$

Furthermore, the direction of the longitudinal axis of $\tau_{\alpha,\beta}(z)$ equals $\theta \in [0, 2\pi)$ where $\exp(i \cdot \theta) = i \cdot \alpha^{-1}$.

B. Evaluation of a Fitted Tented Arch

Given a fit of a tented arch $\tau_{\alpha,\beta}(z)$ we may want to valuate how well it agrees with an orientation field estimation of a fingerprint. Therefore, we assume that an orientation field estimation is given by a set $\{(z_j, v_j)\}$ where v_j encodes the (undirected) orientation at the complex z_j, *i.e.*, if $\varphi_j \in [0, \pi)$ is the orientation at z_j, then $v_j = \cos(2\varphi_j) + i \cdot \sin(2\varphi_j)$.

We evaluate the quality of a fit $\tau_{\alpha,\beta}(z)$ using the function

$$\kappa(\alpha, \beta) = \sum_j \exp\left(\frac{(|z - \gamma_{\alpha,\beta}| - \rho)^2}{2 \cdot \sigma^2}\right) \cdot \left| \frac{\tau_{\alpha,\beta}(z_j)}{|\tau_{\alpha,\beta}(z_j)|} - v_j \right|^2 \tag{4}$$

which we want to minimize over α and β. Here, σ denotes the involved Gaussian's standard deviation and $\rho \geq 0$ controls the distance from the core at which orientation measurements are taken into account with the highest weight.

C. Minimization

Essentially, our estimation of a directed reference point aims at minimizing $\kappa(\alpha, \beta)$ for which there exist multiple approaches. In this paper, we consider the following.

1) *Initial model*: For $\alpha = 1$, perform a global search for an initial β with $\kappa(1, \beta)$ being small.
2) *Update rotation*: Rotate the model $\tau_{\alpha,\beta}(z)$ around the core $\gamma_{\alpha,\beta}$ to decrease the cost $\kappa(\cdot, \cdot)$. This may require to update both α and β.
3) *Update translation*: Update β to decrease $\kappa(\alpha, \cdot)$.
4) *Loop*: Repeat steps 2 and 3 until α and β converge.
5) *Output*: Compute the reference point $\gamma_{\alpha,\beta}$ using (3) and its direction θ as the phase of $i \cdot \alpha^{-1}$ and return $(\gamma_{\alpha,\beta}, \theta)$.

It is possible that α and β do not converge or that $\gamma_{\alpha,\beta}$ is not placed on the fingerprint foreground. In these cases, we may repeat the procedure using another translation part β with $\kappa(1, \beta)$ being small. Yet, it is possible that no initial β yields a valid estimation for a directed reference point. Therefore, we should try only a few (around 20) initial $\tau_{1,\beta}(z)$ and report a failure message if none yielded a valid estimation.

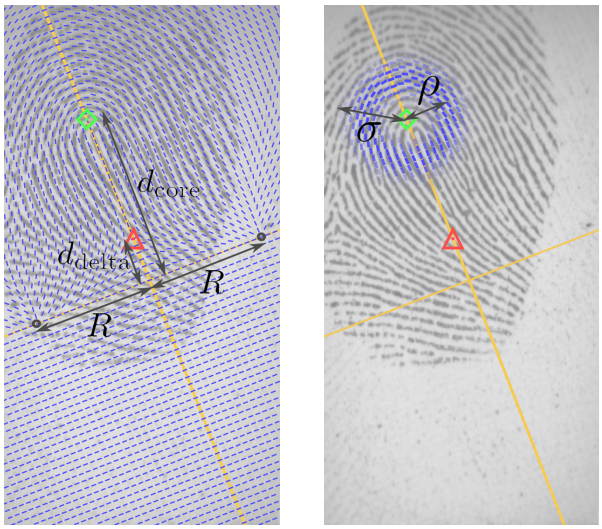
Step 1 can be implemented by iterating $\gamma_{1,\beta}$ over a rectangular grid within the fingerprint's foreground. Therein, we must ensure that the core $\gamma_{1,\beta}$ and the delta $\delta_{1,\beta}$ of the tented arch $\tau_{1,\beta}(z)$ are different from the z_j in the orientation field $\{(z_j, v_j)\}$; otherwise, Formula (4) cannot be evaluated. This can, for example, be ensured by letting the z_j form a rectangular grid and searching β such that the $\gamma_{1,\beta}$ are iterated in between of the z_j; thereby the search of the initial model can be realized by an iteration of the $\gamma_{1,\beta}$ over a grid.

Step 2 and Step 3 can both be implemented using a *steepest descent* method for finding local minimums. Details on steepest descent methods can be found in nearly all textbooks on numerical optimization (*e.g.*, see [53]).

For the experiments in this paper, the orientation fields $\{(z_j, v_j)\}$ have been estimated using the well-known gradient method following the description in [43] in which the z_j form a rectangular grid of which coordinates are systematically arranged with a difference of 7 pixels. The fingerprint foregrounds have been estimated by selecting the largest connected region after *Otsu thresholding* and then choosing the area surrounded by the region's convex hull as the estimation of the foreground.

D. Parameter Configuration

In Section II-D, we described the determination of the parameters for our minutiae-based protection system. In the



(a) (b)

Fig. 4. Visualization of the parameters for the tented arch model (a) and the parameters for the cost function κ controlling with which weight orientation measurements around the core are taken into account (b); note the parameter λ controls how "tented" the tented arch model is and is not visualized.

following we outline how we used the result to resume the parameter training for our automatic method for estimating a fingerprint's directed reference point. The parameters that control the estimation of our method are given by $(d_{\text{core}}, d_{\text{delta}}, \rho, \sigma, \lambda, R)$ and are visualized in Figure 4.

We aimed at maximizing the number of genuine feature set correspondences (generated from absolutely pre-aligned minutiae templates) sharing at least $k = 7$ common elements. The maximum has been taken over the tested parameters $(d_{\text{core}}, d_{\text{delta}}, \rho, \sigma, \lambda, R)$ yielding directed reference points w.r.t. which the minutiae templates could be shifted, *i.e.*, pre-aligned. The minutiae templates were the same as those used for training the vault parameters (*i.e.*, estimated from the FVC 2002 DB2-B training set; see Section II-D). If for a fingerprint no valid reference point was found, the corresponding minutiae template has been temporarily excluded for testing the current configuration. As absolutely pre-aligned minutiae can have negative coordinates, the hexagonal grid used to generate the feature sets was centered in $[-634, 634] \times [-634, 634]$ which covers the region of absolutely pre-aligned minutiae. The tuple $(d_{\text{core}}, d_{\text{delta}}, \rho, \sigma, \lambda, R) = (160, 22, 45, 12, 1.81, 175)$ was found to yield 263 (among 280) genuine feature set correspondences sharing at least $k = 7$ elements.

IV. Experiments

A. Directed Reference Point Estimation

For each fingerprint in the FVC 2002 DB2-A [45] we estimated its directed reference point using the method and parameters as in Section III. An excerpt of the result is visualized in Figure 5. For 18 of the 800 fingerprints, no valid reference point has been output, yielding a *failure to align rate* of 2.3%. We furthermore observed that the average time in estimating a fingerprint's directed reference was $\approx 3.3s$ on a single core of a $3.2GHz$ desktop computer.

B. Evaluation

TABLE I
Performance and securities of our minutiae-based fuzzy vault with absolute fingerprint pre-alignment

k	GAR (FAR)	GDT (IDT)	brute-force security	false-accept security
7	91% (0.87%)	$0.08s$ ($0.27s$)	2^{36}	2^{22}
8	88% (0.12%)	$0.14s$ ($0.35s$)	2^{41}	2^{25}
9	87% (0.04%)	$0.20s$ ($0.41s$)	2^{47}	2^{28}
10	79% (0%)	$0.29s$ ($0.51s$)	2^{52}	2^{31}
11	73% (0%)	$0.36s$ ($0.57s$)	2^{57}	2^{34}
12	72% (0%)	$0.50s$ ($0.69s$)	2^{63}	2^{37}

To demonstrate the practicability of both our minutiae-based protection scheme and our automatic method for absolute fingerprint pre-alignment, we evaluated the verification performance on the FVC 2002 DB2-A [45] for the fixed finite field $\mathbf{F} = \mathbb{F}_{2^{16}}$ and varying $k = 7, \ldots, 12$. The minutiae templates have been estimated using a commercial extractor (Neurotechnology Ltd. Verifinger SDK 5.0). We evaluated the genuine acceptance rates as for the vast majority of minutiae-based fuzzy vault implementations [14]–[19]. Specifically, for each of the 100 fingers in the database, the first impression was used for enrollment and the second was used as the query. For each of the first two impressions, no failure message on directed reference point estimation has been reported, thus, yielding a total of 100 genuine verification attempts. To estimate the false acceptance rates, in [14]–[19], the FVC protocol has been modified to increase the number of observed impostor verification attempts. However, the increase is problematic from a statistical point of view, *i.e.*, the impostor verification attempts are not at all statistically independent (also see [54]). Therefore, we strictly followed the FVC protocol [45] yielding a total of 4950 impostor verification attempts.

In each verification attempt (genuine and impostor) that we simulated, we created a reference vault record $(V(X), \text{SHA}(f))$ as described in Section II-B from an absolutely pre-aligned minutiae template (the minutiae template was pre-aligned w.r.t. the coordinate system given by the directed reference point estimation method described in Section III). Therein, a hexagonal grid of distance $\ell = 29$ centered in $[-634, 634] \times [-634, 634]$ has been used which covers the region of potential absolutely pre-aligned minutiae. The query minutiae template (also absolutely pre-aligned) has been used to build the unlocking pairs as described in Section II-C. Using the randomized decoder (Section II-F) with $\mathcal{D} = 2^{16}$, we aimed at finding an $f^* \in \mathbf{F}[X]$ of degree smaller than k with $\text{SHA}(f^*) = \text{SHA}(f)$. Whenever successful, the verification attempt was considered as an accept; otherwise, as a reject. We also kept track of the average times GDT and IDT needed by the randomized decoder on genuine and on impostor verification, respectively, both determined on a single core of a $3.2\ GHz$ desktop computer. The result of our evaluation can be found in Table I.

V. Security

In addition to usability, of which GAR is an important value to assess it, security is clearly an important property

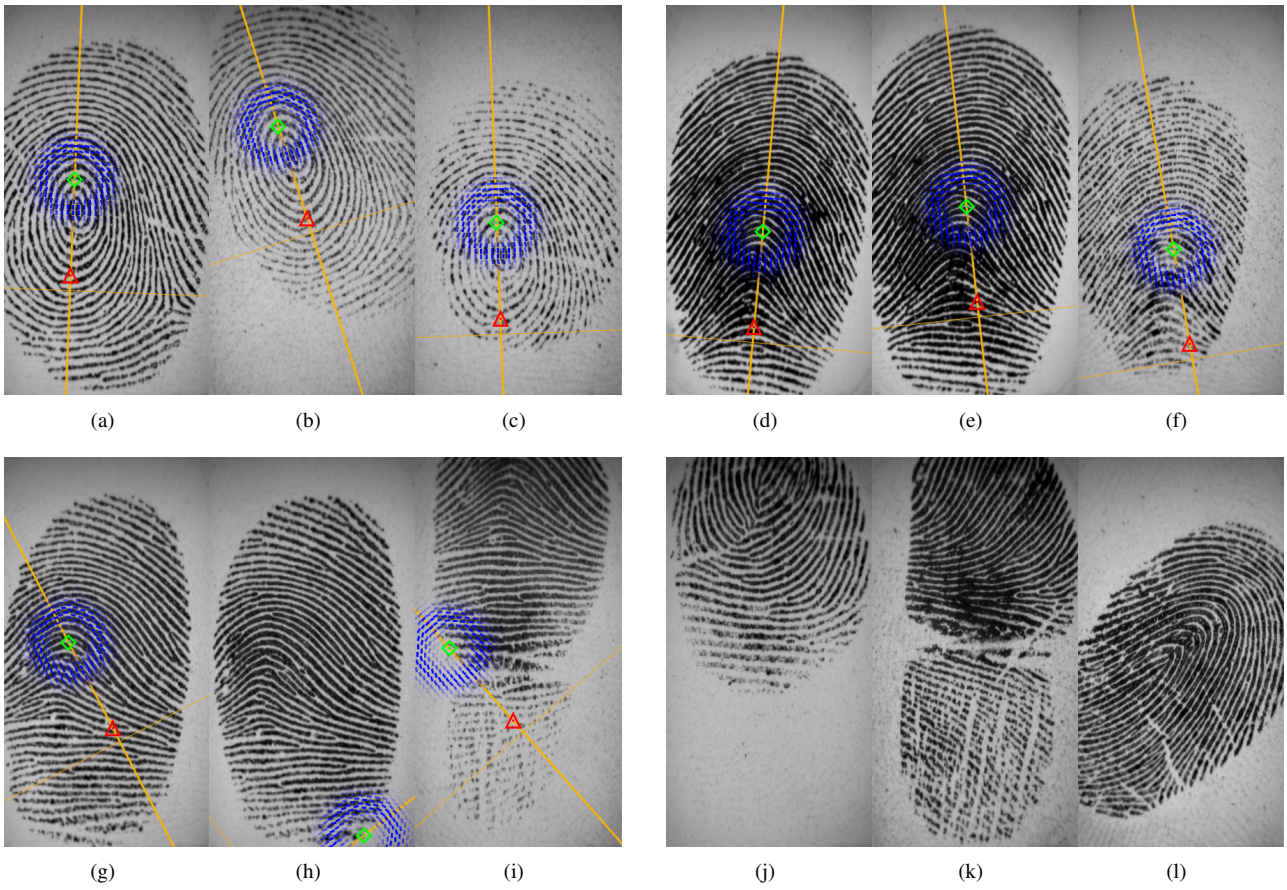


Fig. 5. Excerpt from the directed reference point estimation on the FVC 2002 DB2-A — The core (green diamond) and the direction of the longitudinal axis (bold yellow line) give the coordinate and the direction, respectively, of the estimated directed reference point. The blue lines correspond to the orientation of the quadratic tented arch model. Their transparencies indicate the weight of which the orientations around the core were taken into account. Note that the red triangle indicates the position of the delta of the tented arch model and not a delta of the fingerprint. The estimated directed reference points are quite robust (a)–(c) and our method has the potential to work even for arches (d)–(f). However, there are case in which our method had problems in robustly estimating the directed reference points (g)–(i) and cases in which our method failed in estimating a valid directed reference point (j)–(l).

of a template protection scheme which should not be at the cost of greed for verification performance. In this section, we discuss the security of our implementation against a variety of attacks. Before we describe analyses against the serious attack scenarios of false-accept and record multiplicity attacks, we point out our implementation's security against brute-force and mere key inversion attacks.

A. Key Inversion Attack

On enrollment, a secret key of bit length $16 \cdot k$ is generated at random and used to hide a quantized minutiae template by binding the one to the other. An attacker having intercepted a vault record has therefore the possibility to guess a candidate for the secret key repeatedly and, as soon as successful, he can recover the quantized minutiae template; note that the attacker can check whether the candidate key is correct since he knows a cryptographic hash value of the correct key. Thus, the effective security level of the vault records is bounded by the difficulty in guessing the correct key which varies between 2^{112} and 2^{192} as k varies between 7 and 12. Even though the security provided by the difficulty in guessing the correct key appears to be sufficient for most applications, there are, however, attacks that can recover the minutiae template from our vault records much more efficiently than possible by attempting to guess the correct key.

B. Brute-Force Attack

To estimate a fuzzy fingerprint vault's brute-force security, one typically assumes that the minutiae of a fingerprint are uniformly and independently distributed over the fingerprints' image region (*e.g.*, see [5], [14]–[20]). Even though there are 1733 points of a hexagonal grid of distance $\ell = 29$ fitting in the region of absolutely pre-aligned minutiae, we cannot assume that absolutely pre-aligned minutiae round to them independently from each other. In fact, the minutiae are placed within a rectangle of dimension 296×560. Hence, we assume that our vaults are generated using a hexagonal grid centered in $[0, 296) \times [0, 560)$ where neighboring coordinates are equidistantly spaced by $\ell = 29$ pixels; as a consequence, the hexagonal grid contains $r' = 242$ coordinates. As the minutiae angles are quantized into s values, we obtain a vault of size $n' = 1452$, and since there are at most $t_{\max}$ genuine vault pairs, the difficulty in successfully guessing k genuine vault pairs is at least $\mathbf{bf}(1452, 44, k)$ (see Equation (1)) to

which we refer as the *brute-force security*. For $k = 7, \ldots, 12$ the brute-force securities of our minutiae-based fuzzy vault implementation are listed in Table I.

C. False-Accept Attack

A brute-force security of a minutiae-based fuzzy vault as a candidate for its overall security, is based on the assumption that minutiae are uniformly and independently distributed. This assumption, however, does not account for the statistics of fingerprint features which an adversary may exploit to derive a biometric measurement from an intercepted vault record. Therefore, if the adversary has collected a large database containing absolutely pre-aligned minutiae templates, he can successively use them to simulate impostor verification attempts. For each simulated verification attempt, he can expect to successfully open the vault with probability equals to the non-zero FAR, thereby performing a *false-accept attack*. At this point we stress that the efficiency of a false-accept attack is a hint for the existence of a similarly efficient statistical attack, for example, attacks that make use of artificial fingerprint generators; therefore, the false-accept security of an implementation must be taken seriously when assessing its overall security, and false-accept security is not just of interest where large databases containing fingerprints are available.

If the FAR is known, the difficulty in running a false-accept attack for an attacker who uses the randomized decoder with $\mathcal{D}$ decoding iterations to simulate impostor verification attempts, equals $\mathrm{FAR}^{-1} \cdot \mathcal{D}$. For $k = 7, 8, 9$, our performance evaluation yielded non-zero point estimations for the FAR, and the false-accept security estimations are 2^{23}, 2^{26}, and 2^{27}, respectively. Since these securities are potentially very weak, we should choose $k \geq 10$ for which no false accepts have been observed. On the other hand, we cannot assume that FAR is zero just from a limited number of observations.

Rule of Three: In order to find an upper bound for the FAR, we use the *rule of three* [54]–[56]: Assume that we observed no false accepts within N independently and uniformly distributed observations for impostor verification attempts; then $\mathrm{FAR} \leq 3/N$ with *confidence level* at least 95%. For our minutiae-based fuzzy vault where $k \geq 10$ we can therefore assume (with a confidence of 95%) that the FAR is not larger than 0.061%, and we may thus assume that the absolute false-accept security is not smaller than $\approx 2^{27}$.

Sharper Analysis: With the rule of three, the FAR can only be bounded by $3/N$ which depends on N. Consequently, the estimation of a very small false-accept security with the rule of three does not appear to be appropriate.

In the following, we give a heuristic on how a sharper estimate of the FARs for our minutiae-based fuzzy vault implementation may be achieved. Therefore assume that on an impostor verification attempt an unlocking set of size t containing ω genuine vault pairs is built by an attacker. Let

$$p(t, \omega, \mathcal{D}) = \begin{cases} 1 - (1 - \mathbf{bf}(t, \omega, k)^{-1})^{\mathcal{D}} & , \text{ if } \omega \geq k \\ 0 & , \text{ otherwise} \end{cases} \quad (5)$$

be the probability that the attacker successfully decodes the unlocking set using the randomized decoder. Consequently, if we observe N impostor verification attempts, in which the jth unlocking set is of size t_j containing ω_j genuine pairs, we may estimate the FAR as $\mathrm{FAR}^* = \frac{1}{N} \sum_j p(t_j, \omega_j, \mathcal{D})$.

So far, we ignored the fact that the attacker is not restricted to a certain verification protocol. In particular, if the attacker has intercepted a vault record that he desires to break via a false-accept attack, he can use a number of decoding iterations $\mathcal{D}$ minimizing the overall effort. There is an optimal $\mathcal{D}$ that the attacker may choose.

Proposition 1. *The effort for a successful false-accept attack using the randomized decoder is minimal for $\mathcal{D} = 1$.*

Proof: Let $\varepsilon(x)$ be the false acceptance rate as a function of the number of decoding iterations x. Then, for any fixed $q \in (0, 1)$, after an effort of $c(x) = \log(q) / \log(1 - \varepsilon(x)) \cdot x$, an attacker will succeed in breaking an intercepted vault record with probability $1 - q$. Note that we can write $1 - \varepsilon(x) = \frac{1}{N} \sum \zeta_j^x$ where $0 \leq \zeta_j \leq 1$. Using *Jensen's inequality* we can bound $1 - \varepsilon(x) \geq \left(\frac{1}{N} \sum \zeta_j\right)^x$. Thus,

$$\begin{aligned} c(x) &= \frac{|\log(q)|}{|\log(1 - \varepsilon(x))|} \cdot x \geq \frac{|\log(q)|}{\left|\log\left(\left(\frac{1}{N} \sum \zeta_j\right)^x\right)\right|} \cdot x \\ &= \frac{|\log(q)|}{\left|\log\left(\frac{1}{N} \sum \zeta_j\right)\right| \cdot x} \cdot x = \frac{|\log(q)|}{\left|\log\left(\frac{1}{N} \sum \zeta_j\right)\right|} = c(1) \end{aligned}$$

which proves the proposition. ∎

The proposition states that, in order to analyze the false-accept security of our implementation, we should prefer to estimate the FAR assuming the adversary does not run more than one random decoding iteration with the randomized decoder for each simulated impostor verification attempt. Hence, we may estimate the false-accept security as

$$\left(\frac{1}{N} \sum_j p(t_j, \omega_j, 1) \right)^{-1} \quad (6)$$

where N is the number of observed impostor verification attempts, t_j denotes the size of the jth unlocking set, ω_j is the number of its genuine pairs, and $p(t_j, \omega_j, 1)$ is computed as in Equation (5).

To estimate the security for $k = 7, \ldots, 12$, we kept track of the $N = 4950$ observed (t_j, ω_j) occurred during evaluation (Section IV-B) which enabled us to estimate the false-accept security via Equation (6). The results are listed in Table I.

D. Correlation Attack

For most implementations of a fingerprint fuzzy vault, no analyses against correlation attack have been given [14]–[16], [18], [19]. In fact, in [28] it has been demonstrated that two genuine vault correspondences can be broken at quite a high rate with the correlation attack. To convince the reader that our attack does resist this specific attack (*i.e.*, that it does not yield more advantage to an attacker than by breaking one of the vault records individually), we give a proof.

Note that for a vault $\mathbf{V} = \{(x, y)\} \subset \mathbf{F} \times \mathbf{F}$ there is a one-to-one correspondence between $\{x\}$ and the *vault feature space* of (possibly quantized) features $\{\mathfrak{m}\}$ (minutia, say) some of which are genuine and the others are chaff. In the scenario of a

correlation attack, we assume that an intruder has intercepted two vaults protecting the features $\{\mathfrak{m}\}$ and $\{\mathfrak{m}'\}$. Now, the intruder aims at finding a rotation and translation of $\{\mathfrak{m}'\}$ such that the transformed features $\{T(\mathfrak{m}')\}$ well correlate with $\{\mathfrak{m}\}$, *i.e.*, such that the number of $d(\mathfrak{m}, T(\mathfrak{m}')) \leq \epsilon$, for some non-negative threshold ϵ, becomes maximal (here $d(\cdot,\cdot)$ denotes a distance function between two features). Let $\mathbf{U}$ consist of those vault pairs that belong to the features $\mathfrak{m}$ with $d(\mathfrak{m}, T(\mathfrak{m}')) \leq \epsilon$. If the expectation of $d(\mathfrak{m}, T(\mathfrak{m}'))$ for matching feature correspondences $(\mathfrak{m}, \mathfrak{m}')$ is smaller than for non-matching correspondences, the intruder can expect that $\mathbf{U}$ consists of a reasonable amount of genuine pairs (see Figure 2 for a visualization) and can hope that an algorithm for decoding Reed-Solomon codes will break the vault.

If the two vault feature sets $\{\mathfrak{m}\}$ and $\{\mathfrak{m}'\}$ are equal, then for each $\mathfrak{m}$ (genuine and chaff) there exists an $\mathfrak{m}'$ with $0 = d(\mathfrak{m}, \mathfrak{m}') \leq \epsilon$ for any threshold $\epsilon \geq 0$. Consequently, the correlation attack does not yield any advantage to the attacker because $\mathbf{U}$ consists of the entire vault pairs and, thus, the difficulty in running the correlation attack reduces to the difficulty in breaking a single vault individually. In fact, for our implementation the feature space for each vault consists of exactly the (coarsely quantized) minutiae that are encoded by the elements in $\mathbf{E}$. This proves that our implementation is resistant against a correlation attack. □

Note that, as a side-effect of resistance against the correlation attack, our implementation also resists attacks exploiting a possible larger *free area* of genuine vault features compared to the free area of chaff features [57]: The free area of a vault feature is defined as the area of the largest "circle" around the vault feature containing no other vault features; in [57] it has been worked out that the free area of genuine feature may have a bias to be larger than the free area of chaff features. In our implementation, the free areas for each vault feature is constant, thus, preventing the attack.

E. Other Record Multiplicity Attacks

The correlation attack is an attack via record multiplicity and our implementation resists this specific attack. One may ask whether general record multiplicity attacks can be applied against multiple vault records generated by our implementation from the same individual to link them across different application's databases, *i.e.*, cross-matching, or even to break them. It has been shown in [44] that the (improved) fuzzy vault scheme (as well as the fuzzy commitment scheme and all other constructions studied in [30], [31]) are in principle vulnerable against record multiplicity attack, and in view of this fact, we should analyze the security of our implementation in the presence of record multiplicity. Therefore, let $(V(X), \mathrm{SHA}(f))$ and $(W(X), \mathrm{SHA}(g))$ be two vault records protecting the feature sets $\mathbf{A}$ and $\mathbf{B}$, respectively. Furthermore, we assume that $\mathbf{A}$ and $\mathbf{B}$ contain t and $u \leq t$ elements, respectively, and that f and g are two polynomials of degree less than k. Furthermore, let $\omega = |\mathbf{A} \cap \mathbf{B}|$.

It is shown in [37] that if $\omega \geq (t+k)/2$, then the differences $\mathbf{A} \setminus \mathbf{B}$ and $\mathbf{B} \setminus \mathbf{A}$ can be recovered efficiently via *partial recovery attacks*; furthermore, if in addition $t-\omega \geq k$, then the two vault records $(V(X), \mathrm{SHA}(f))$ and $(W(X), \mathrm{SHA}(g))$ can be even fully broken efficiently. Otherwise, if $\omega < (t+k)/2$, there is no attack known that performs better than an algorithm that is exponential in the finite field size or by breaking the vault records individually [37].

TABLE II
SUCCESSFUL LINKING RATES FOR RELATED (GLR) AND FOR NON-RELATED (ILR) VAULT RECORDS AS WELL AS SUCCESSFUL RECOVERY RATES FOR RELATED (GRR) AND NON-RELATED (IRR) VAULT RECORDS. THE RATES HAVE BEEN MEASURED WITH AND WITHOUT RE-ORDERING THE FINITE FIELD ENCODING VIA A RECORD-SPECIFIC RANDOM PUBLIC PERMUTATION PROCESS.

	without re-ordering		with re-ordering	
k	GLR (ILR)	GRR (IRR)	GLR (ILR)	GRR (IRR)
7	41% (0%)	40% (0%)	0% (0%)	0% (0%)
8	39% (0%)	35% (0%)	0% (0%)	0% (0%)
9	37% (0%)	33% (0%)	0% (0%)	0% (0%)
10	35% (0%)	30% (0%)	0% (0%)	0% (0%)
11	34% (0%)	27% (0%)	0% (0%)	0% (0%)
12	31% (0%)	20% (0%)	0% (0%)	0% (0%)

In view of the existence of efficient partial recovery attacks, we observed during our experiments (Section IV) that an attacker has in fact quite good chances to distinguish related from non-related vault records via a partial recovery attack (the corresponding *related linking rate* GLR versus *non-related linking rate* ILR are listed in Table II). Even worse, in most of the cases, the recovered differences suffice to fully break the vaults (see the related recovery rates GRR in Table II). It is therefore necessary to implement measures that prevent an intruder from successfully applying partial recovery attacks.

Partial recovery attacks require that the feature sets protected by the vault polynomials $V(X)$ and $W(X)$ share at least $(t+k)/2$ elements, and it seems reasonable that this property can be destroyed via a vault-specific public random permutation process. Let $P : \mathbf{E} \to \mathbf{E}$ be a random public bijection and set $\mathbf{A}' = P(\mathbf{A})$. Instead of publishing $(f(X) + \prod_{x \in \mathbf{A}}(X - x), \mathrm{SHA}(f))$, we publish $(f(X) + \prod_{x' \in \mathbf{A}'}(X - x'), \mathrm{SHA}(f))$ as the vault record, *i.e.*, $V(X) = f(X) + \prod_{x' \in \mathbf{A}'}(X - x')$. Note that, since $P : \mathbf{E} \to \mathbf{E}$ is public, it is easy to adjust the verification process without affecting its performance. Now, assume that $(W(X), \mathrm{SHA}(g))$ is a second vault record with re-ordering, *i.e.*, $W(X) = f + \prod_{x' \in \mathbf{B}'}(X - x')$ where $Q : \mathbf{E} \to \mathbf{E}$ is another random public bijection and $\mathbf{B}' = Q(\mathbf{B})$ is of size $u \leq t$. For efficient known partial recovery attacks to reveal the feature sets' differences explicitly, the fulfillment of the inequality $|\mathbf{A} \cap \mathbf{B}| \geq (t+k)/2$ is neither sufficient nor necessary; the inequality $|\mathbf{A}' \cap \mathbf{B}'| \geq (t+k)/2$ must be fulfilled instead. Since both involved bijections are random, the sets $\mathbf{A}'$ and $\mathbf{B}'$ are random. As the probability for $\mathbf{A}'$ and $\mathbf{B}'$ to share exactly ω' elements follows the *hypergeometric distribution*, *i.e.*, $\mathbb{P}(\ |\mathbf{A}' \cap \mathbf{B}'| = \omega'\ ) = h(\omega'|n;u;t) = \binom{u}{\omega'} \cdot \binom{n-u'}{t-\omega'} \cdot \binom{n}{t}^{-1}$, we can compute the probability that $\mathbf{A}'$ and $\mathbf{B}'$ share at least $\omega' > 0$ elements by $\mathbb{P}(\ |\mathbf{A}' \cap \mathbf{B}'| \geq \omega'\ ) = 1 - \sum_{j=0}^{\omega'-1} h(j|u;t;n)$. This yields a formula for computing the probability that the feature sets differences of two related/non-related vault can be recovered via a partial recovery attack.

The probability that direct application of a partial recovery attacks can link two vault records generated by our

implementation (where $t = u = 44$, $n = 10398$, and $\omega' = \lceil (t + k)/2 - 1 \rceil$) in which the feature sets have been passed through a public random individual permutation process approximately equals 2^{-169} for $k = 7$, and one easily verifies that the probability is not larger than approximately 2^{-71} for any $7 \leq k \leq u \leq t \leq t_{\max} = 44$. During our experiments described in Section IV-B, we in fact have not observed any vault record correspondences (related or non-related) protecting re-ordered features sets of more than 3 common elements which is much too few for partial recovery attacks to yield any advantage for breaking the vaults.

This demonstrates that the incorporation of a public record-specific permutation process may be a promising countermeasure to prevent record multiplicity attacks with records generated by our implementation, thereby preventing these attacks without dealing with a key that needs to be kept secret. Furthermore, note that the permutation process does not need to be stored explicitly along with the vault records which would require $\mathcal{O}(n)$ additional memory for storing them. Instead, a seed for a pseudo-random number generator can be stored. For our experiments briefly described above, we used the 160 bit SHA hash values $\mathrm{SHA}(f)$ as seeds for a pseudo-random number generator to encode the record-specific permutations.

The attacker can compute the inverse of the public permutation processes, which have been chosen independently from the protected feature sets; but, he may not be able to re-order the feature sets while protected by the vault in order to run any known efficient and effective record multiplicity attack. For this, he has first to break the vaults and thus, attacking two related vaults may be essentially as hard as breaking them individually. Yet, it is not known whether an attacker can exploit the knowledge of the permutation processes by another (currently unknown) attack. To date, no such attack has been discovered, and therefore we may conclude that our implementation is resistant against known record multiplicity attacks. It would nonetheless be useful to search for a mathematical proof during future research to strengthen this notion. Alternatively, we could search for a counterexample by finding an efficient and effective attack.

F. Other Attacks

Authentication systems based on biometric measurements are subjected to more risks than those discussed in this article. The *surreptitious key-inversion attack* and *blended substitution attack*, both discussed by Scheirer and Boult in 2007 [27], are examples. These attacks can be classified as *man-in-the-middle attacks* and cryptographic techniques may be used to prevent them, *e.g.*, utilizing *digital signatures*, *tamper-proof devices*, and encrypted communication between the authentication modules.

VI. Discussion

A. Summary

In this paper, we presented an implementation of a minutiae-based fuzzy vault for generating vault records as candidates for renewable biometric references [2]. Our implementation has been designed with the aim to remove one of the most serious vulnerabilities of current fuzzy vault implementations to fingerprint: the correlation attack [28]. Therefore, we designed a minutiae quantization process, thereby also preventing an attacker from exploiting a possibly larger free area of genuine features [57]. Another positive effect of (minutiae) quantization is that it enables an improved fuzzy vault scheme in which vault records consume a significantly smaller amount of memory as compared to the original fuzzy vault scheme. A very delicate problem for a minutiae-based fuzzy vault is the alignment problem which is typically solved by storing auxiliary data publicly along with the vault records. In order to prevent an attacker from exploiting public auxiliary alignment data for attacks, we designed our implementation for absolutely pre-aligned minutiae; thereby, we presented an automatic method for estimating directed reference points from fingerprints. We evaluated the performance of our minutiae-based fuzzy vault on a dataset typically used for evaluating fingerprint-based fuzzy vault implementations. We also provided a comprehensive security analysis against scenarios and attacks that are, to the best of the authors' knowledge, state-of-the-art and representative in the context of fuzzy vault. In particular, we provided a serious treatment of the false-accept attack. Furthermore, we analyzed our implementation accounting for recently discovered record multiplicity attacks [37], [44] and showed that our implementation can effectively be secured against them.

B. Comparison

The GAR that we measured for our security-improved minutiae-based fuzzy vault is 79% for a parameter configuration at which no false accepts have been observed. Furthermore, our implementation provides a brute-force security of 2^{52}, resists known attacks via record multiplicity, and does not leak information about the protected fingerprint templates from auxiliary alignment data.

When compared to a GAR of 86% achievable with an original minutiae-based fuzzy vault implementation [16], our implementation is clearly less user-friendly. On the other hand, our implementation significantly improves on the very weak brute-force security of 2^{39} achieved in [16]. Furthermore, the implementation in [16] may be vulnerable to the correlation attack and does leak information about the protected fingerprints from auxiliary alignment data (*e.g.*, constellations of high ridge curvature coordinates which can be exploited for cross-matching), all vulnerabilities that do not exist for our implementation. One may argue, that the need for auxiliary alignment data can be circumvented if alignment-free features are used [19]: At a brute-force security of 2^{52} a GAR of 92% can be achieved. The scheme proposed in [19] does not have an accompanying analysis that shows it to be resistant to the correlation attack. In summary, it remains unclear whether a GAR significantly larger than 79% and false-accept security at least 2^{31} can be achieved with a password-free fingerprint fuzzy vault implementing measures to prevent correlation attacks.

Another advantage of the vault records generated by our implementation are their compact record sizes. The records

generated by the minutiae-based fuzzy vault implementation in [16] consume 896 bytes (plus the data being required to store auxiliary alignment data in form of a three-dimensional point cloud), and the records generated by the alignment-free fuzzy fingerprint vault in [19] consume 1780 bytes. For our implementation the vault records only require 108 bytes which is a significant improvement.

Our implementation has several security benefits as compared to other implementations of the fuzzy vault scheme to fingerprints, but it provides a less usable GAR. In order to allow fair assessment between different implementations as candidates for generating RBRs, a clear concept of sufficient security and sufficient usability has to be specified by our community; to the best of the authors' knowledge only vague requirements (such as usability and privacy) for biometric authentication systems have been stated [2], but the biometric security community lacks well-defined security and usability notions including commonly accepted attack scenarios and specific state-of-the-art attacks.

C. Conclusion and Outlook

We showed that a security-improved minutiae-based fuzzy vault, in particular removing the issue of record multiplicity attacks, can provide a GAR of 79% at a brute-force security of 2^{52} and false-accept security of 2^{31}. Our analyses clearly confirm that false-accept attacks, which are hints for the existence of similarly efficient statistical attacks, are the weakest link of fuzzy fingerprint vaults. Our work and analyses given in this paper do not prove but indicate low security and usability limitations for the implementation of password-free RBRs [2] based on mere single fingerprints.

For our future research we plan to consider methods for improving the GARs on basis of our minutiae-based fuzzy vault implementation. One approach to obtain this is to improve robustness of our presented approach for absolute fingerprint pre-alignment, for example, by combining it with a method for estimating a fingerprint's focal point [50] and/or its core [49]. In addition, it may be possible to reconsider template protection techniques for quantized minutiae in their polar representation [58] w.r.t. coordinate systems given by more robust estimations of directed reference points. Furthermore, it appears worthwhile to investigate the possibility of removing distortion from a fingerprint [59] before they are passed through a quantization process. Also, even though absolutely pre-aligned minutiae in fact are alignment-free features, it would be interesting to examine whether the use of other alignment-free features can significantly improve on a GAR of 79% while allowing the generation of unlinkable password-free records.

Even if the GAR can be brought to an acceptable level, the rather low security limitations are likely to remain. To some extent the security limitations can be improved by implementations for multiple fingers (or even multiple biometric modalities). On the other hand, the use of multi-biometrics is clearly at the cost of usability and, in view of this fact, it may be important to assess the usability of heavy multi-biometrics versus the use of measurements of a few (still conveniently usable) biometric traits in combination with easy-to-use passwords (*e.g.*, 4-digit PINs) to achieve sufficient security and sufficient usability.

REFERENCES

[1] B. Schneier, *Applied Cryptography (2Nd Ed.): Protocols, Algorithms, and Source Code in C*. New York, NY, USA: John Wiley & Sons, Inc., 1995.

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

[3] A. Cavoukian and A. Stoianov, *Biometrics: theory, methods, and applications*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009, ch. 26 - Biometric Encryption: The New Breed of Untraceable Biometrics.

[4] G. J. Tomko, C. Soutar, and G. J. Schmidt, "Fingerprint controlled public key cryptographic system," US Patent 5,541,994, 1994.

[5] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in *Proc. ACM SIGMM workshop on Biometrics methods and applications*, ser. WBMA '03. New York, NY, USA: ACM, 2003, pp. 45–52.

[6] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. Int. Symp. Inf. Theory*, A. Lapidoth and E. Teletar, Eds., 2002, p. 408.

[7] ——, "A fuzzy vault scheme," *Des. Codes Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[8] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[9] M. Sudan, "Decoding of reed solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, pp. 180–193, 1997.

[10] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," *IEEE Trans. Intell. Transp. Syst.*, vol. 45, pp. 1757–1767, 1998.

[11] D. Bleichenbacher and P. Q. Nguyen, "Noisy polynomial interpolation and noisy chinese remaindering," in *Proc. Int. Conf. on Theory and application of cryptographic techniques*, ser. EUROCRYPT'00, Berlin, Heidelberg, 2000, pp. 53–69.

[12] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of reed-solomon codes is np-hard," in *Proc. of the ACM-SIAM Symp. on Discrete algorithms*, ser. SODA '05. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, pp. 470–478.

[13] A. Kiayias and M. Yung, "Cryptographic hardness based on the decoding of reed-solomon codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2752–2769, Jun. 2008.

[14] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, 2005, pp. 310–319.

[15] U. Uludag and A. K. Jain, "Securing fingerprint template: fuzzy vault with helper data," in *Proc. Workshop on Privacy Research In Vision*, 2006, pp. 163–169.

[16] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 744–757, 2007.

[17] K. Nandakumar, A. Nagar, and A. Jain, "Hardening fingerprint fuzzy vault using password," in *Proc. Int. Conf. on Biometrics*, ser. LNCS 4642, 2007, pp. 927–937.

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

[19] 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.*, vol. 33, pp. 207–220, May 2010.

[20] P. Mihăilescu, A. Munk, and B. Tams, "The fuzzy vault for fingerprints is vulnerable to brute force attack," in *Proc. of BIOSIG*, 2009, pp. 43–54.

[21] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. of ACM Conf. on Computer and Communications Security*, 1999, pp. 28–36.

[22] A. Stoianov, T. Kevenaar, and M. v. d. Veen, "Security issues of biometric encryption," in *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, 2009, pp. 34–39.

[23] T. Kevenaar, U. Korte, J. Merkle, M. Niesing, H. Ihmor, C. Busch, and X. Zhou, "A reference framework for the privacy assessment of keyless biometric template protection systems," in *BIOSIG*, 2010, pp. 45–56.

[24] J. Merkle, M. Niesing, M. Schwaiger, H. Ihmor, and U. Korte, "Security capacity of the fuzzy fingerprint vault," *Int. J. on Advances in Security*, vol. 3, no. 3&4, 2011.

[25] K. Simoens, B. Yang, X. Zhou, F. Beato, C. Busch, E. M. Newton, and B. Preneel, "Criteria towards metrics for benchmarking template protection algorithms." in *ICB*. IEEE, 2012, pp. 498–505.

[26] J. Merkle, H. Ihmor, U. Korte, M. Niesing, and M. Schwaiger, "Performance of the fuzzy vault for multiple fingerprints (extended version)," *CoRR*, vol. abs/1008.0807v5, 2011.

[27] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proc. of Biometrics Symp.*, 2007, pp. 1–6.

[28] A. Kholmatov and B. Yanikoglu, "Realization of correlation attack against the fuzzy vault scheme," in *Proc. SPIE*, vol. 6819, 2008.

[29] J. Li, X. Yang, J. Tian, P. Shi, and P. Li, "Topological structure-based alignment for fingerprint Fuzzy Vault," in *Proc. Int. Conf. on Pattern Recognition*, 2008, pp. 1–4.

[30] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT*, 2004, pp. 523–540.

[31] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[32] A. Arakala, J. Jeffers, and K. Horadam, "Fuzzy extractors for minutiae-based fingerprint authentication," in *Proc. Int. Conf. on Biometrics*, ser. LNCS 4642, 2007, pp. 760–769.

[33] H. Xu and R. N. J. Veldhuis, "Binary representations of fingerprint spectral minutiae features," in *Int. Conf. on Pattern Recognition*, 2010, pp. 1212–1216.

[34] X. Shao, H. Xu, R. N. Veldhuis, and C. H. Slump, "A 3-layer coding scheme for biometry template protection based on spectral minutiae," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Singnal Processing*, 2011, pp. 1948–1951.

[35] S. Draper, A. Khisti, E. Martinian, A. Vetro, and J. Yedidia, "Using distributed source coding to secure fingerprint biometrics," in *Int. Conf. Acoutics Speech Signal Proc*, 2007, pp. 129–132.

[36] C. A. Trugenberger, "The glass maze: Hiding keys in spin glasses," in *Proc. of BIOSIG*, 2011, pp. 89–102.

[37] J. Merkle and B. Tams, "Security of the improved fuzzy vault scheme in the presence of record multiplicity (full version)," *CoRR*, vol. abs/1312.5225, 2013, submitted.

[38] K. Simoens, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches." in *IEEE Symp. on Security and Privacy*. IEEE Computer Society, 2009, pp. 188–203.

[39] B. Tams, "Decodability attack against the fuzzy commitment scheme with public feature transforms," *CoRR*, vol. abs/1406.1154, 2014.

[40] E. J. C. Kelkboom, J. Breebaart, T. A. M. Kevenaar, I. Buhan, and R. N. Veldhuis, "Preventing the decodability attack based cross-matching in a fuzzy commitment scheme," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 107–121, 2011.

[41] E. R. Berlekamp, *Algebraic coding theory*. Laguna Hills, CA, USA: Aegean Park Press, 1984.

[42] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Trans. Comput.*, vol. 55, no. 9, pp. 1081–1088, Sep. 2006.

[43] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2009.

[44] M. Blanton and M. Aliasgari, "Analysis of reusability of secure sketches and fuzzy extractors," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1433–1445, 2013.

[45] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "FVC2002: Second Fingerprint Verification Competition," in *Proc. Int. Conf. on Pattern Recognition*, 2002, pp. 811–814.

[46] National Institute of Standards and Technology, "Announcing the Secure Hash Standard," available online http://csrc.nist.gov/, 1995.

[47] A. M. Bazen and S. H. Gerez, "An intrinsic coordinate system for fingerprint matching," in *Proc. Int. Conf. on Audio- and Video-based Biometric Person Authentication*, 2001, pp. 198–204.

[48] T. Hotz, "Intrinsic coordinates for fingerprints based on their longitudinal axis," in *Proc. Int. Symp. on Image and Signal Processing and Analysis*, 2009, pp. 501–504.

[49] S. O. Novikov and V. S. Kot, "Singular feature detection and classification of fingerprints using hough transform," in *Proc. SPIE*, vol. 3346, 1998, pp. 259–269.

[50] K. Rerkrai and V. Areekul, "A new reference point for fingerprint recognition," in *Proc. Int. Conf. on Image Processing*, 2000, pp. 499–502.

[51] B. Tams, "Absolute fingerprint pre-alignment in minutiae-based cryptosystems," in *Proc. of BIOSIG*, 2013, pp. 75–86.

[52] S. Huckemann, T. Hotz, and A. Munk, "Global models for the orientation field of fingerprints: An approach based on quadratic differentials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1507–1519, 2008.

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[54] A. Mansfield and J. Wayman, "Best practices in testing and reporting performance of biometric devices," available online at http://www.csee.wvu.edu/~natalias/biom426/BestPractice02.pdf, 2002.

[55] J. A. Hanley and A. Lippman-Hand, "If nothing goes wrong, is everything allright? interpreting zero numerators," *Journal of the American Medical Association*, vol. 249, no. 13, pp. 1743–1745, 1983.

[56] B. Jovanovic and P. Levy, "A look at the rule of three," *The American Statistician*, vol. 51, no. 2, pp. 137–139, 1997.

[57] E.-C. Chang, R. Shen, and F. W. Teo, "Finding the original point set hidden among chaff," in *Proc. ACM Symp. on Information, computer and communications security*, ser. ASIACCS '06. New York, NY, USA: ACM, 2006, pp. 182–188.

[58] J. Jeffers and A. Arakala, "Fingerprint Alignment for a Minutiae-Based Fuzzy Vault," in *Proc. Biometrics Symp.*, 2007, pp. 1–6.

[59] A. Senior and R. Bolle, "Improved fingerprint matching by distortion removal," *IEICE Trans. Information and Systems*, vol. E84-D, pp. 825–831, 2001.