The Fuzzy Vault for Fingerprints is Vulnerable to Brute Force Attack

Preda Mihăilescu, Axel Munk, and Benjamin Tams*
University of Göttingen
preda@uni-math.gwdg.de, munk@math.uni-goettingen.de, btams@math.uni-goettingen.de

Abstract: The fuzzy vault approach is one of the best studied and well accepted ideas for binding cryptographic security into biometric authentication. We present in this paper a brute force attack which improves on the one described by T. Charles Clancy et. al. in 2003 in an implementation of the vault for fingerprints. Based on this attack, we show that three implementations of the fingerprint vault are vulnerable and show that the vulnerability cannot be avoided by mere parameter selection in the actual frame of the protocol. We will report about our experiences with an implementation of such an attack. We also give several suggestions which can improve the fingerprint vault to become a cryptographically secure algorithm. In particular, we introduce the idea of fuzzy vault with quiz which draws upon information resources unused by the current version of the vault. This may bring important security improvements and can be adapted to the other biometric applications of the vault.

1 Introduction

Secure communication relies on trustable authentication. The most widespread authentication methods still use passwords and pass-phrases as a first step towards identity proving. Secure pass-phrases are hard to remember, and the modern user needs a large amount of dynamic passwords for her security. This limitation has been known for a long time and it can in part be compensated by the use of chip cards as universal access tokens.

Biometrical identification, on the other hand, is based on the physical identity of a person, rather then by their control of a token. Reliable biometric authentication would thus put an end to password insecurity, various repudiation disputes and many more shortcomings of phrase or token based identities. Unlike the deterministic keys which are common for cryptography, biometric data are only reproducible within vaguely controlled error bounds, they are prone to various physical distortions and have quite a low entropy.

Overcoming the disadvantages of the two worlds by using their mutual advantages is an important concern. We look back over almost a decade in which the biometrics community developed an increasing concern for the security and privacy of biometrical systems. It is not the purpose of this technical note to clarify the interesting notions and attempts which

*corresponding author, supported by Graduiertenkolleg 1023 Identification in mathematical models of DFG
were developed in this context. For this purpose, we refer to the survey [UMP+04] of Uludag et. al. on biometric cryptosystems.

Researchers from cryptography and coding theory attempted to develop new concepts allowing to model and evaluate, from an information theoretical point of view, algorithms which deal with the specific restraints of biometrics: non-uniformly distributed data with incomplete reproducibility and low, hard to estimate, entropy. Both communities are motivated by the wish to handle biometrics like a classical password, thus protecting it by some variant of one-time functions and performing the verification in the image space. Unlike passwords, the biometrics are not deterministic. This generates substantial challenges for the verification after one-time function transforms. Juels and Wattenberg [JW99] and then Juels and Sudan [JS02] have developed, namely the fuzzy commitments and fuzzy vault, two related approaches with a strong impact on biometric security. The papers of Dodis et. al. [DORS08, BDK+05] can be consulted for further theoretic developments of the concepts of Juels et. al. and their formalisation in an information theoretic framework. It is inherent to the problem that core concepts of the theory, such as the entropy of a biometric template, are hard even to estimate. Thus the security proofs provided by the theory do not translate directly in practical estimates or indications.

In 2003, Clancy, Kiyavash and Lin gave [CKL03] a statistically supported analysis for a realistic implementation of the vault for fingerprints. The authors observe from the start that the possible parameter choices in this context are quite narrow in order to allow sufficient security; they succeed to define a set of parameters which they claim provides the cryptographically acceptable security of $O(2^{69})$ operations for an attack. Our analysis shows that faster attacks are possible in the given frame, thus making brute force possible. The analyses in [CKL03] are outstanding and have been used directly or indirectly in subsequent papers; the good security was obtained at the price of quite a high error probability (20% − 30%). In [UPJ05, Ulu06], Uludag, Pakanti and Jain provided an implementation of the fuzzy vault for fingerprints which uses alignment help-data and was applied to the fingerprints from the FVC2002 database [MMC+02]. This improves the identification rate; however some of the simplifications they make with respect to [CKL03] reduce security quite dramatically, and the ideas of these authors could very well be combined with the more conservative security approach of Clancy et. al.. Yang and Verbauwhede [YV05] describe an implementation of the vault, with no alignment help, which follows closely the concepts of [CKL03] and focuses upon adapting to various template qualities and numbers of minutiae recognised in these templates.

This paper is not to be understood as a proof of weakness of the fuzzy vault scheme, in its abstract setting. At the contrary, one needs not to change biometrics or go to more costly multibiometrics to make the current version of fuzzy vault secure. We argue that multi-finger biometrics are more than sufficient in this respect.

In this paper, which extends an earlier unpublished one [Mih07], we describe the original fuzzy vault in Section 2 and argue that the security proofs and remarks given in [JS02] are not useful for fingerprint applications. In Section 3, we discuss the various implementations mentioned above and show that essentially brute force attacks can be performed in feasible time in all instances. In Section 4, we will report about our experiences with an implementation of an attack based on these considerations. Afterwards, in Section 5 we
discuss possible variants and alternatives and suggest some additional sources of information which might render the fuzzy vault secure in connection with fingerprint application. The ideas of fuzzy vaults and commitments are of great importance in biometric security. While their information theoretic foundation [JW99, JS02, DORS08, BDK+05] is well set and understood, the core problem of estimating entropies brings their biometric application at the borderline between skills and science. A minima moralia in this case requires a realistic evaluation of simple attacks, like a well conceived brute force attack. The estimate of a brute force attack to a system is an irrefutable upper bound for the security of that system; if that bound shows to be too low, concerns and improvements are called for. Moreover, using only some statistics of minutiae locations in various images of the same fingerprint, it proves that pushing parameters to the extremes cannot suffice for gaining secure fuzzy vaults for single fingerprints, without bringing some new ideas in play. In this respect, we do not claim novelty to this result. Therefore, we present also a new idea, which we call fuzzy vault with quiz, that is likely to highly increase security, even in the case of one finger identification. The idea is very simple and will be presented in the specific context of fingerprints; conceivably, it may also be regarded as a generalization of the general concepts in fuzzy schemes and commitments. In this paper, we simply give an example of quiz and support its validity by an implementation.

This paper concentrates on fingerprints, for two reasons: first, there is a considerable amount of research concerning the application of fuzzy vaults or fuzzy-vault-inspired hash variants to fingerprint security. Second, the patterns of vulnerability and possible improvements can be discussed more accurately on a single biometrics. The interested reader may find observations which can be applied to other biometrics and also to multibiometrics.

2 The Fuzzy Vault

The fuzzy vault is an algorithm for hiding a secret string \( S \) in such that a user who is in possession of some additional information \( T \) can easily recover \( S \), but an intruder should face computationally infeasible problems in order to achieve this goal. The information \( T \) can be fuzzy, in the sense that the secret \( S \) is locked by some related, but not identical data \( T' \). Juels and Sudan define the vault in quite general terms and allow multiple applications. Biometry is one of them and we shall restrict our description directly to the setting of fingerprints. Generalizations can be found in [JS02, DORS08, BDK+05].

The string is prepared for the transmission in the vault as follows. Let \( S \in \{0, 1\}^* \) be a secret string of \( l \) bits length. The user (Alice, say) that wishes to be identified by the string \( S \) has her finger scanned and a locking set \( \mathcal{L} \) comprising the cartesian coordinates of \( t \) minutiae in the finger scan is selected from this finger template \( T' \); the couples of coordinates are concatenated to single numbers \( X_i = (x_i || y_i) \in \mathcal{L} \). One selects a finite field \( \mathbb{F}_q \) attached to the vault and lets \( k' + 1 = \lceil \frac{l}{\log_2(q)} \rceil \) be the number of elements in \( \mathbb{F}_q \) necessary to encode \( S \). One assumes that \( 0 < \max_{X \in \mathcal{L}} X < q \) and maps \( X \leftarrow \mathbb{F}_q \) by some convention. Selecting \( f(X) \in \mathbb{F}_q[X] \) to be a polynomial of degree \( k \geq k' \) with coefficients which encode \( S \) in some predetermined way, one builds the genuine set \( \mathcal{G} = \)
\[ \mathcal{G}(\mathbb{F}_q, S, t, k, \mathcal{L}) = \{ (X_i, Y_i) : X_i \in \mathcal{L}; Y_i = f(X_i) \}, \]
which encodes the information on \( S \). The genuine verifier Bob has an original template \( T \) of Alice’s finger and should use this information in order to recover \( f(X) \) and then \( S \). In order to make an intruder’s (Victor’s, say) attempt to recover \( S \) computationally hard, the genuine set is mixed with a large set of chaff points \( \mathcal{C} = \{ (U_j, W_j) : j = 1, 2, \ldots, r - t \} \), with \( U_j \not\in \mathcal{L} \) and \( W_j \neq f(U_j) \); the chaff points should be random uniformly distributed. Chaff points and genuine lists are shuffled to a common vault with parameters \( V = V(k, t, r, \mathbb{F}_q) = \mathcal{G} \cup \mathcal{C} \).

Upon reception, Bob will generate an unlocking set \( \mathcal{U} \) containing those \( X_i \) coordinates of vault points, which well approximate coordinates of minutiae in \( T \). This templates must have negligible nonlinear distortions and be aligned modulo affine transforms. The second condition is addressed in [Ulu06]. The unlocking set may be erorated either by allowing some chaff points which are closer to \( T \) than locking points, or by simple coordinate imprecision. Both problems can be dealt within given limits of error correcting codes. Thus Juels and Sudan suggest using Reed Solomon codes for decoding \( f(X) \).

The security argumentation in [JS02] is based upon the expectation that the chaff points will build an important amount of subsets of \( t \) elements, whose coordinates are interpolated by polynomials of degree \( k \), thus hiding \( f(X) \) from Victor among these random polynomials. The argument is backed up by the following lemma, a proof of which can be found in [JS02, CKL03].

**Lemma 1.** For every \( \mu \), \( 0 < \mu < 1 \) and every vault \( V(k, t, r, \mathbb{F}_q) \), there are at least \( \frac{\mu}{3} q^{k-t} (r/t)^t \) random polynomials \( g \in \mathbb{F}_q[X] \) such that \( V \) contains \( t \) couples \( (U_j, g(U_j)) \).

### 2.1 A brute force attack

If Victor intercepts a vault \( V = V(k, t, r, \mathbb{F}_q) \), but has no additional information about the location of minutiae or some of their statistics, he may still try to recover \( S \) by brute force trials. For this, he needs to find by random trials \( k + 1 \) points in the genuine list \( \mathcal{G} \). The chances that \( k + 1 \) points of the vault are also in the genuine list are:

\[ 1/P = \binom{r}{k+1} \sim (r/t)^{k+1} < 1.1 \cdot (r/t)^{k+1}, \quad \text{for } r > t > 5. \tag{1} \]

This, together with the fact that the odds for a point \((X, Y) \in \mathbb{F}_q^2\) to lay on the graph of a given polynomial \( f \in \mathbb{F}_q[X] \) are equal to the probability \( P[Y = f(X)] = 1/q \) yield the ground for the proof of Lemma 1. Lagrange interpolation of a polynomial of degree \( k \) can be done in \( O(k \log^2(k)) \) operations [Jvz03]; checking whether an additional point \((U, W)\) lays on the graph of \( f(X) \) (so \( W = f(U) \)) requires \( O(k) \) steps, so \( K = O(\log^2(k)) \) such verifications can be done at the cost of one interpolation.

We assume now with Clancy et. al., that there is a degree \( k < D < t \) which is minimal with the property that among all polynomials \( g \in \mathbb{F}_q[X] \) of degree \( k \) which interpolate vault points, \( f(X) \) is the only one interpolating at least \( D \) points. This yields a criterion for identifying \( f \).
Lemma 2. Let $\mathcal{V} = \mathcal{V}(k, t, r, \mathbb{F}_q)$ be a fuzzy fingerprint vault and $k < D < t$ be chosen as above. Then an intruder having intercepted $\mathcal{V}$ can recover the secret $S$ in $R = C \cdot (r/t)^{k+1}$ operations, where $C < 8rk$.

Proof. We have shown that in less then $< 1.1 \cdot (r/t)^{k+1}$ trials, Victor can find a set of $k+1$ points from the locking set $L$. In order to find such a set and then $S$, for each $(k+1)$-tuple $T = (X_i, Y_i)_{i=0}^k \subset \mathcal{V}$ Victor has to

1. Compute the interpolating polynomial $g_T(X)$. It is proved in [JvzG03] that the implicit constant for Lagrange interpolation is 6.5; let $K = 6.5 \cdot \log^2(k)$. Thus all interpolation polynomials require $< 7.2 \cdot k \cdot \log^2(k) \cdot (r/t)^{k+1}$ operations.

2. Search a point $(U, W) \in \mathcal{V} \setminus T$ such that $g(U) = W$. This requires the equivalent of $r/K$ Lagrange interpolations. If no point is found, then discard $T$.

3. If $T$ was not discarded, search for a further point verifying $g(U) = W$. This step is met with probability $1/q$. If a point is found, add it to $T$; otherwise discard $T$.

4. Proceed until a break condition is encountered (no more points on the graph of $g(X)$) or $D$ points have been found in $T$.

Adding up the numbers of operations required by the steps 1-4., with weights given by the probabilities of occurrence, one finds:

$$R < 7.2 \cdot (r/t)^{k+1} \cdot k \cdot K \cdot \frac{rq}{K(q-1)} < 8.0 \cdot (rk) \cdot (r/t)^{k+1},$$

as claimed. Note in particular that the bound does not depend on $D$, since this value is absorbed in the sum of the series $\sum(1/q)^i$, of probabilities to successfully add $i$ point to $T$.

Here are some remarks on factors that influence the complexity of the brute force attack.

Restricting the region of interest from which Victor chooses points is irrelevant, if minutiae are assumed to be uniformly distributed over the template. In this case, $r$ and $t$ are scaled by the same factor and thus $r/t$ and the complexity of brute force remain unchanged.

The complexity grows with the degree $k$ of the polynomial $f(X)$. However, high degrees $k$ require large unlocking sets, which may not be possible for average quality fingerprints and scanners. Thus one can only augment the degree to an extent to which it is not (too much) increase the work required for unlocking by Bob.

The complexity grows with the number of chaff points. There is a bound to this number, given by the size of the image on the one side and the variance in the minutiae location between various data capturings and extractions [CKL03]. Clancy and his co-authors find empirically the lower bound $d \geq 10$ for the distance between chaff points, and this distance was essentially respected also by the subsequent works.

The complexity grows while reducing $t$. This is however also detrimental for genuine unlocking, since it may reduce the size of the unlocking set below the required minimum.
What can be inferred about the security of fingerprint vaults from the seminal paper [JS02]? First, one observes that Juels and Sudan suggest to use error correcting codes, thus avoiding explicit indications to whether an interpolation polynomial is the correct \( f(X) \). Uludag and Jain suggest on the other hand in [Ulu06] the use of CRC codes: thus \( S \) is padded by a CRC code, adding 1 to the degree \( k' \) needed to encode \( S \). Upon decoding, Bob can check the CRC and ascertain that he found the correct secret. This simplifies the unlocking procedure. Does it bring advantages to Victor? If the degree \( k > k' \) and thus the CRC does not increase additionally the polynomial degree, Victor has a gain of \( O(8 \cdot r / \log^2(k)) \), as follows from Lemma 2. Otherwise, there is no gain.

It is made clear in the Chapters 4 and 5 of [JS02] that the amount of chaff points is essential for security. The suggested minimum lays about \( r \sim 10^4 \). For fingerprints, a large amount of chaff points decreases the average distance between these (and also genuine points) in the list. The value \( r = 10^4 \) leads to an average distance of \( 2 - 5 \) pixels between the point coordinates, depending on the resolution of the original image. This is below realistic limits as mentioned above. At this distance, even in presence of a perfect alignment, the genuine verifier Bob should need some additional information (like CRC or other) providing the confirmation of the correct secret. Such a confirmation is contrary to the security lines on which Juels and Sudan make there evaluation. There is an apparent conflict between the general security proofs in [JS02] and realistic applications of the fuzzy vault to fingerprints. In [JS02] the authors explicitly warn that applications involving privacy-protected matching cannot achieve sufficient security. It is not clear, whether fingerprint matching is considered in [JS02] as belonging to this category.

### 3 Implementations Of The Fingerprint Vault

We start with the most in depth analysis of security parameters for the fingerprint vault, which was done by Clancy and co-authors in [CKL03]. The paper focusses on applications to key release on smart cards. They suggest using multiple scans in order to obtain, by correlation, more reliable locking sets. As mentioned above, the variance of minutiae locations which they observed in the process leads to defining a minimum distance between chaff (and genuine) points, which is necessary for correct unlocking. This minimal distance \( d \sim 11 \) implies an upper bound for the size \( r \) of the vault and thus the number of chaff points!

The authors use very interesting arguments on packing densities and argue that in order to preserve the randomness of chaff points, these cannot have maximal packing density. On the other hand, assuming that the intruder has access to a sequence of vaults associated to the same fingerprint and he can align the data of these vaults, then the randomness of chaff points allows a correlation attack for finding the genuine minutiae. This argument suggests rather using perfectly regular high density chaff point packing. These are hexagonal grids with mutual distance \( d \) between the points. The genuine minutiae can be rounded to grid points, and Victor will have no inference point for distinguishing these from the chaff points. We shall comment below on this point.
The implementation documented in this paper suggests the following parameters for optimal security: \( k = 14, D = 17, t = 38, r = 313 \). The brute force attack in Lemma 2 is more efficient than the one of Theorem 1 of [CKL03], on which they base their security estimates. Using the above parameters and Lemma 2, we find an attack complexity of \( \sim 2^{55} \); comparing this to the complexity of genuine unlocking yields a security factor \( F \sim 2^{49} \) which is below cryptographic security, unlike the \( 2^{69} \) deduced in [CKL03]. By the very balanced arguments used in the parameter choice, the security bounds obtained on base of this paper are an indication of the vulnerability of the fingerprint vault in general.

Yang and Verbauwhede describe in [YV05] an implementation of the vault, in which the degree of the polynomials \( f(X) \) varies in dependence of the size \( t \) of the genuine list, which itself depends directly on template quality. From the point of view of security, the paper can be considered as a follow up of [CKL03], which addresses the problem of poor image quality with its consequences for the size of the locking set. Secret sizes and polynomial degrees are adapted to the size of locking sets. The proposal is consistent, its vulnerability to attacks is comparable to [CKL03] in general, and higher, when adapting to poor image quality.

The major contributions of Uludag and Jain in [Ulu06] is to provide a useful set of helper data for easing image alignment. This has an important impact on the identification rate. As mentioned above, they bring the elegant and simple proposal of adding a CRC to the secret, thus easying the unlocking work. We discussed above the issue of the security risk increasement: this is arguably small, below a factor of \( 2^8 \). On the other hand, the degree of \( k = 8 \) for the polynomial \( f(X) \) and vault size \( r = 224, \) whilst \( t = 24, \) makes their system more vulnerable, with an absolute attack complexity of \( \sim 2^{37} \).

4 Attacking Fuzzy Fingerprint Vaults

In this section, we will report about our experiences with attacks on fingerprint vaults. Before coming to data we describe how we proceeded in implementing our attack.

4.1 Implementation of our attack

Before we implemented the attack a working en- and decryption should have been available for us. Its implementations essentially requires operations in a finite field. Therefore we worked with Victor Shoups Number Theory Library (NTL) [Sho09]. Furthermore, we strictly used finite fields of characteristic 2 for this gives canonical conventions for identifying finite field elements with positive integers (bitwise).

Our encryption implementation requires the degree \( k \) of the polynomial \( f \), the number \( t \) of points to be extracted from a fingerprint template, the fingerprint template, and the size \( r \) the vault will have. In this first version, the template simply is taken from a list of minutiae locations. For a minutia location \((x, y)\) the concatenation \((x||y) = x + 2^{16} \cdot y\) is computed where we implicitly assume that \( x \) and \( y \) fit into 2 byte length integers. As
already mentioned in section 2 the size $q = 2^m$ is chosen so that for all minutia locations of the template $(x, y)$ the inequality $(x||y) < q$ holds. Similar, given the secret $S \in \{0, 1\}^*$ of $l$ bits length the size $q$ fulfils $k > \left\lceil \frac{t}{\log_2(q)} \right\rceil$. Using NTL’s functionalities a defining polynomial $P(X) \in \mathbb{F}_2[X]$ of a finite field $\mathbb{F}_q$, $q = 2^m$, is built and then the secret $S$ is identified with its corresponding polynomial $f \in \mathbb{F}_q[X]$ of degree $k$. If $\alpha$ is a root of $P$ and $(x||y) = \sum_{i=0}^{m-1} x_i 2^i$ is given in its binary representation the vault point $(X, Y)$ is obtained from this by setting $X = \sum_{i=0}^{m-1} x_i \alpha^i$ and $Y = f(X)$. In this way the locking set is constructed. The chaff points are generated by generating locations $(x, y)$ having a reasonable distances to genuine minutiae locations and then, as above, $X$ is computed but $Y \in \mathbb{F}_q$ is generated at random such that $(X, Y)$ does not lie on the polynomial graph. In such a way the vault $V$ is obtained. If $(X, Y) \in V$ is a point of the vault then, as above, there are unique integers $x, y$ corresponding to $X$ and $Y$, respectively. By this, a partial order is given on the vault points. Thus, in its representation the array of vault points is sorted w.r.t. this order such that no one is able to distinguish genuine points from chaff points just using knowledge about how genuine points are dispersed in the vault (e.g. appended or pushed in front).

Next, the implementation of the decryption was done ignoring the alignment modulo affine transform for this does not affect the attack. Given a list of minutiae locations our implementation simply extracts those vault points (after deconcatiations of their $X$-coordinate into $(x, y)$) which well approximate template locations. Using the Peterson-Berlekamp-Massey algorithm as suggested in [JS02] one succeeds in recovering the polynomial $f \in \mathbb{F}_q[X]$ if at least $k + t^2$ of extracted vault points are also genuine points.

In such a way we implemented a working protocol of fuzzy vault for fingerprints.

Thereafter, the implementation of the attack as in subsection 2.1 was done analogue. The attack requires the vault data and a process id number giving a seed such that parallel running processes will interpolate different polynomials from a randomly chosen sequence of $(k + 1)$-tuple of vault points.

### 4.2 Running brute force attacks

Each attack against our fingerprint vaults were done on a 4 multiprocessor Quad-Core AMD Opteron(tm) Processor 8347 HE with 1.9 GHz and 32GB RAM using 8 processes in parallel.

A first attack we ran was against a vault consisting of $r = 224$ points hiding a polynomial of degree $k = 8$ interpolating $t = 24$ vault points. Thus, for the probability $P$ for a single trial to lead to the desired polynomial fulfilled $[1/P] = 2.542 \times 10^{11}$. These are the same security parameters as in [Ulu06]. Due to our implementation of the protocol the size of the finite field in which our operations took place was $2^{25}$ contrary to $2^{16}$ in [Ulu06]. We started 8 processes in parallel. All processes together interpolated and tested 11347 polynomials per second of CPU time whether they interpolate $t$ vault points. Hence, we expected the whole attack to succeed in discovering the polynomial after 1 day 7 hours 7
minutes and 33 seconds of CPU time. Lucky as we were it in fact succeeded after 1 hour and 58 seconds of CPU time.

Another vault with same security parameters as before but a finite field of size $2^{108}$ this time (due to a larger bit length of the encrypted secret) was attacked. This time the attack interpolated and tested 17123 polynomials a second of CPU time. This let us expect to succeed in discovering the polynomial after 20 hours 37 minutes and 34 seconds of CPU time. The attack was successful after 10 hours 55 minutes and 8 seconds of CPU time. One may wonder why in this larger field relatively more operations can be performed. This may be due to tuning details of NTL.

We also started a brute force attack against a vault having optimal security parameter as suggested by this paper. Thus, $r = 313$, $t = 38$ and $k = 14$. In fact we did not succeed in breaking such a vault but do a few calculations out of our experiences. The probability $P$ that a randomly selected $(k + 1)$-tuple leads to the hidden polynomial fulfills $\frac{1}{P} = 953\,116\,315\,773\,448$. We interpolated and tested 8124 polynomials a second of CPU time. Thus, we expect to succeed in discovering the polynomial after more than 1860 years. Modern supercomputers thus are able to break such a vault within a feasible amount of time.

5 Security Discussion

We discuss in this section several variants for improving security of the fingerprint vault.

5.1 Using more fingers

We have shown that the parameters $r, t, k$, allowing to control the security factor, are naturally bounded by image size, variance of minutiae location and average number of reliable minutiae. They cannot thus be modified beyond certain bounds and it is likely that this bounds have been very well derived in [CKL03]. It lays thus at hand to propose using for instance the imprints of two fingers rather then only one, for creating the vault. This way the parameters can be virtually doubled, yielding to a literal squaring of the security factor.

5.2 Non-random chaff points

As mentioned above, it is argued in [CKL03] that chaff points should have random distribution; this leads to halving the packing density. However, one can embrace the opposite attitude, consisting in laying a hexagonal grid of size $d = 11$, proposed by the authors. Each grid point will be attached to some vault point - chaff or genuine. Thus Victor will have no means for distinguishing between chaff points and genuine ones, despite of the regularity of the grid. Thanks to the error correcting codes, the genuine points can always
be displaced by a distance at most $d/2$ to a grid point. It is the packing density which, according to the results in [CKL03] doubles, thus doubling the vault size $r$. It is thus conceivable that this strategy may also improve the security of the vault. Nevertheless, the consequences need still be analysed.

5.3 Quizes using additional minutiae information

There is more information in a minutia than its mere coordinates. Such are for instance the orientation, lengths and curvatures of incoming lines, neighbouring data, etc. We propose to attach to each minutia a quiz which can be solved in robust manner by Bob, but which introduces for Victor several (say $b$) bits of uncertainty per minutia. Thus for polynomial degree $k$, the security may be increased by a factor of $2^{kb}$.

We give in the case of the orientation a simple example of how a quiz functions. This, in fact, was added by us to the implementation we reported in Section 4. Let $X$ be the concatenated coordinates of a fixed minutia and let $\alpha$ be its orientation, in a granularity of $\pi/n$, for some small integer $n$. Then, along with $(X, f(X))$, the vault will also contain a random value $\beta$ instead of $\alpha$: thus the minutia is represented by $(X, Y, \beta)$.

Upon reception, Bob computes the integer $0 \leq j < n$ such that $j \pi n = \alpha - \beta \mod \pi$. The value of $j$ will then encode a certain transformation $Y' = T(Y)$ of the received value $Y$ and in fact the interpolating value will be set to be $Y' = f(X)$. Note that the vault creator has control on the generation of $\beta$ and it may be chosen such that the value of $j$ can be safely recovered (thus $\alpha - \beta$ is bounded away from a multiple of $\pi/n$). For chaff points, $\beta$ is random.

In our implementation, the transformation given by $j$ was chosen as a kind of shift of $Y$. If $\theta$ is a root of a defining polynomial of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$ then any $Y \in \mathbb{F}_{2^m}$ can be written as $Y = \sum_{i=0}^{m-1} y_i \theta^i$. The value $j$ then defines $T(Y) = \sum_{i=0}^{m-1} y_r(i, j) \theta^i$, where $r(i, j) = i + j \mod n$. Its inverse is then given by $T^{-1}(Y) = \sum_{i=0}^{m-1} y_r(i, -j) \theta^i$.

Several robust additional informations may as well increase the security of the fingerprint vault to a cryptographically acceptable level.

5.4 The alternative of cryptographic security

These observations lead to the question: is the use of one-way functions and template hiding an intrinsic security constraint, or just one in many conceivable approaches to securing biometric authentication? The second is the case, and it is perfectly feasible to construct a secure biometric authentication system based on the mechanisms used by state of the art certification authorities. Basically, the scanners of the biometric system need to:

1. Have enclosed, temper proof, cryptographic units.
2. Encrypt templates immediately after the image generation.
3. Build up secure channels to the matching servers, using challenge response mecha-
nisms.

4. Create distinguished templates e.g. by endowing them with time stamps, scanner credentials and signature.

On the server side, template databases should be encrypted and the matching be performed in secure, temper proof environments. These requirements are quite general and must be fulfilled in cryptographically secure environments, so adding them to a biometric system is possible. Note that the template is transmitted in encrypted form and is event-bound. Only upon verification of signature, credentials and time stamps will the verification proceed with the template matching. If the cryptographic verification fails, no subsequent action is taken: in particular, a compromised template is not sufficient to break the system. At the contrary, in order to use a fake template, one needs to gain control upon the scanner and force its credentials and signatures upon a stolen template: this is assumed to be hard. This eliminates the stringent and possibly unachievable condition to protect the templates as if their revelation would compromise their usage in any system at any ulterior time.

6 Conclusions

It has been attempted to achieve security in biometric application either by using one-way functions adapted to the specifics of biometric data, or by direct application of strong cryptographic techniques. We showed that one of the leading methods of the first category, the fuzzy vault, allows a simple attack to its instantiation for fingerprint data [CKL03, UPJ05, Ulu06, YV05].

The attack described and implemented is a brute force attack, the worst case for the attacker, and thus the best case for the genuine user. It is an indication of the security limitation of the current applications of the fuzzy vault to fingerprints. The attack can definitely be improved. E.g., by using some meet in the middle strategy in the combinatorial search. More important, the upper bound in Lemma 2 was estimated on the assumption that the attacker does not distinguish the chaff points in his search. This is not realistic, since an intruder should use some statistics on the minutiae locations in a fingerprint to derive probabilities for points to be genuine ones. This would lead to a useful order of priorities in the brute force search described above; the result would be conceivably comparable to a reduction of the number of chaff points to less than a half! Note that the helper data proposed in [Ulu06] are in this case also a major help for the attacker.

It would be interesting to conduct such attacks in the future. However, considering that the upper bound found in this paper, together with these natural improvement strategies clearly show that security is insufficient, one may argue that the development and investigation of more secure alternatives to the present fuzzy vault implementation should have higher research priority. We have brought some suggestions which may help raising the security level of the fingerprint vault to cryptographically acceptable values.

One may argue that similar attacks could be possible to other related methods and thus cryptographic security is preferable, whenever it can be achieved or afforded. Subsequent work should consider variants of the one-way function ideas which could have higher secu-
rity, even if they do not meet the standards of cryptographic security. Also, cryptographic security can be achieved by in a wide scale of variants; analysing pros and cons of such variants is an open topic.

References


