Security analysis for fingerprint fuzzy vaults

Jesse Hartloff, Maxwell Bileschi, Sergey Tulyakov, Jimmy Dobler, Atri Rudra, Venu Govindaraju

Dept. of Computer Sc. and Eng., University at Buffalo, Buffalo, NY, USA

ABSTRACT

In this work we place some of the traditional biometrics work on fingerprint verification via the fuzzy vault scheme within a cryptographic framework. We show that the breaking of a fuzzy vault leads to decoding of Reed-Solomon codes from *random* errors, which has been proposed as a hard problem in the cryptography community. We provide a security parameter for the fuzzy vault in terms of the decoding problem, which gives context for the breaking of the fuzzy vault, whereas most of the existing literature measures the strength of the fuzzy vault in terms of its resistance to pre-defined attacks or by the entropy of the vault. We keep track of our security parameter, and provide it alongside ROC statistics. We also aim to be more aware of the nature of the fingerprints when placing them in the fuzzy vault, noting that the distribution of minutiae is far from uniformly random. The results we show provide additional support that the fuzzy vault can be a viable scheme for secure fingerprint verification.

Keywords: fuzzy vault, security, fingerprints, biometrics, cryptography, Reed-Solomon codes

1. INTRODUCTION

Fingerprints are a popular form of biometrics; however, they are currently mostly collected by governmental agencies and do not have a large-scale commercial deployment. The primary reason for this is the lack of good hashes for fingerprints which both obfuscate the fingerprints and still allow for fast and accurate matching. Most existing work on fingerprints either focuses exclusively on the speed and accuracy of matching or designing hashes for secure storage of fingerprints. In particular, the proposed secure hashes for fingerprints have not been thoroughly compared with existing matching algorithms with respect to their accuracy. Further, the analysis of secure hashes are theoretical and assume that the fingerprints have sufficient entropy, i.e. the fingerprints are distributed somewhat uniformly at random. This paper aims to bridge these two shortcomings.

In this work, we focus on a hash function called *fuzzy vault*, which was first proposed by Juels and Sudan.[1] The fuzzy vault has been studied quite a bit by the biometrics community in recent years[2–6] along with fuzzy commitment[7] which is similar. One advantage of the fuzzy vault is that there is very little information leakage if the vault is not successfully unlocked. Though, unlike traditional work in biometrics which reports full ROC curves, existing results on fuzzy vault only consider a few cases where the false accept rate is very small or zero. We believe that experiments should be performed for the entire range of false accept and false reject rates and report the ROC curve to give a fair comparison between a secure fuzzy vault and traditional matching algorithms. *Our first contribution is a study on the accuracy of the fuzzy vault matching along the lines of traditional fingerprint matching.*

The main reason for studying hash functions for fingerprints is for security. Indeed, it seems very unlikely that hash functions that obscure fingerprints will have the same accuracy as traditional fingerprint matching. Some methods actually improve the matching accuracy while achieving security, though these schemes make use of an encryption key that is assumed to be secret.[8] In this approach, called biometric hardening, at the time of enrollment each enrollee presents a key known only to himself. This way, an impostor fingerprint will use a key different from the enrolled fingerprint, leading to increased security. We do not take this approach in our work.

Our implementation uses only data from the fingerprint itself which makes it more difficult to achieve both goals. The hope, however, is this loss in accuracy will be made up for by increased security against attackers. Some measures of security have been previously studied in the literature. Theoretical work on secure hash

Send correspondence via email to hartloff@buffalo.edu

function have presented formulas for the amount of security (the *security parameter**).[1,9] However, all of these security measures are based on worst-case entropy arguments. For the range of parameters that are practical for fingerprints, we end up with a very small security parameter.† Presumably, it is due to this reason that existing work on fuzzy vault in the biometrics community use the following somewhat ad-hoc notion of security: the time taken by a natural brute force algorithm to break the hash function, or other specified attacks.[4,5,10] Others have used average entropy as a security measure.[11] *Our second main contribution is to point out that breaking the fuzzy vault leads to decoding of Reed-Solomon codes from* random *errors, which has been proposed as a hard problem in the cryptography community.*[12] In other words, we measure the security of the fuzzy vault in terms of an established cryptographically secure primitive to obtain a non-trivial security parameter. Furthermore, unlike in previous works, in our experiments we also keep track of this security parameter.

However, we note that the fuzzy vault is still vulnerable if an attacker has two distinct vaults that were locked with the same fingerprint.[3] The attacker can compare the vaults and observe which points are consistent between them. These points are likely to be the genuine points and the rest are the randomly generated chaff points. Our measures of security are for a single vault; more work must be done to eliminate this type of attack.

There is another issue with the previously observed theoretical security analysis of the fuzzy vault. Due to the way the hash function is defined, the minutia points in the fingerprints have to be uniformly distributed in order to obtain the claimed security parameters. This is because the hash function tries to hide the minutia points (without modification) inside uniformly random "chaff" points. More precisely, the hash function first maps the minutia points to some discrete set of points (in a finite field) and then hides it between uniformly random points from the field. However, the minutia points are not distributed uniformly.[13] To the best of our knowledge, existing work while quantizing the minutia points uniformly quantized the entire possible ranges of the minutia points. For example, if we wanted to use 16 bits to represent the x, y and θ coordinates of a particular minutia, we would simply choose 3 numbers adding to 16 (one for each coordinate), say 4, 4, 8, respectively. And let's say x, y and θ are given to us in values of 8, 8, and 10 bits. We would then simply drop the 4 least significant bits for x and y, and the 2 least significant bits for θ.[2] It has been mentioned that uniform quantization does not account for nonlinear distortion, though we are able to correct for this in the unlocking step.[2]

Therefore, it is important to make the minutiae uniform in the field; otherwise, the scheme is much more prone to probabilistic attacks. *The third main contribution of the paper is to present a more "fingerprint aware" quantization function that results in a (more) uniform distribution in the finite field space.* In other words, instead of binning all the possible minutia points into bins of equal size, we break the range up into bins of unequal size such that in expectation each bin receives (roughly) the same number of minutia points.

1.1 Overview of our Results

In Section 2, we recall the formal definition of the fuzzy vault, which is based on the well-known Reed-Solomon codes. We also recall that, as proposed by Kiayias and Yung,[12] the cryptographically hard problem based on hardness of decoding Reed-Solomon codes for random errors is beyond the limits of currently known efficient decoding algorithms. We then show that "breaking" the fuzzy vault is the same as breaking this hard cryptographic problem. In particular, we formally define these two hard problems and prove their equivalence.

In Section 3, we present the details of our implementation. Our main contribution here is a more fingerprint-aware quantization of minutia points: the idea is to perform non-uniform "binning" of the minutia points in their corresponding ranges so that the quantized values are uniformly distributed. In particular, we present simulations on the FVC2002 fingerprint database 1 and show that the empirical distribution of our quantization is better than those in existing work as shown in figure 1.

In Section 4, we present the results of our experiments on the FVC2002 fingerprint database 1. Here we would like to highlight two differences from previous related work. First, we present full ROC curves as is the norm of traditional fingerprint matching results. Second, we also plot our security parameter with the FRR and FAR curves.

---

*At a high level, a security parameter of λ means that a randomized polynomial time adversary only has roughly $2^{-\lambda}$ probability advantage over the trivial behavior of outputting a random answer.

†In our experimental settings, the measure of entropy is essentially 0.

1.2 Related Work

We begin with an overview of prior work on fingerprint fuzzy vault in the biometrics community. As has been mentioned earlier, there has been a fair bit of work so here we will concentrate on the work that is most relevant to ours.

Nandakumar, Jain and Pankanti[2] present a fuzzy vault implementation with probably the best known accuracy (our accuracy numbers are worse). To help with alignment, the authors store a "hint" in a high-curvature datum in the locked vault to assist with alignment. However, this work does not present full ROC curves and refers to Chang, Shen and Teo[14] for security. Chang et al. again use pre-defined attacks and entropy measures to describe the security of the vault. By contrast our work with the original fuzzy vault and presents full ROC curves along with a thorough security analysis.

Nagar, Nandakumar and Jain[11] showed that matching accuracy can be improved further by utilizing minutia descriptors. These descriptors capture data from the area surrounding each minutia point. This data is not stored in the vault, yet it used for locking and unlocking the vault. Since it is not stored, an attacker presumably must guess all the bits of this data so the entropy of the vault will be increased by the number of bits used. They show that the vault will have a reasonable of amount entropy in the average case with their setup.

Boult, Scheirer and Woodworth[8] improve both accuracy and security by using additional encryption with independent keys. The keys assists in matching since an impostor with a different key is unlikely to match even with a very similar fingerprint. Part of the security of this new system also comes from this extra encryption. By contrast, we analyze the security of the existing fuzzy vault.

There has been a fair bit of work on attacks on the fingerprint. Scheirer and Boult[3] present multiple attacks for different proposed secure biometrics methods. A few of these are on fuzzy vault. In particular, the authors observe that if the same fingerprint is used to lock different secrets, then this information can be used by an attacker to break the vaults. There have also been work on other specific attacks.[4,5,10,14] By contrast, our work shows that the fuzzy vault can be secure against more general attacks assuming that the adversary has access to only one vault for a given fingerprint.

There has also been work on using "secure sketches" that use binary BCH codes instead of Reed-Solomon codes (which are used for fuzzy vault).[15,16] Further, these works use multiple readings of the same fingerprint to improve the accuracy of the schemes.

Finally, we survey the theoretical results. The fuzzy vault was proposed by Juels and Sudan.[1] Designing secure sketches using BCH codes was proposed by Dodis et al.[9] Both of these works present entropy based security results. By contrast, Kiayias and Yung[12] present a computationally hard problem based on decoding Reed-Solomon codes from random errors. This latter work forms the basis of our security results.

2. FUZZY VAULTS AND REED-SOLOMON CODES

In this section, we recall the definition of the fuzzy vault of Juels and Sudan[1] and Reed-Solomon codes (in particular, the related cryptographic hardness assumption in Kiayias and Yung[12]).[‡] We then point out the connection between the hardness of breaking the fuzzy vault and the hardness assumption from Kiayias and Yung[12] . For the rest of the section, we will assume that the minutia points have been quantized such that they can be mapped to the finite field with q elements, which we will denote by $\mathbb{F}_q$. (We will come back to the question of quantization in Section 3.)

We first recall the definition of polynomials. A polynomial of degree $k-1$ over $\mathbb{F}_q$ is defined by k coefficients $p_0, \ldots, p_{k-1} \in \mathbb{F}_q$ and is given by $P(X) = \sum_{i=0}^{k-1} p_i X^i$. It is well-known that given $r \geq k$ distinct points $\alpha_1, \ldots, \alpha_r \in \mathbb{F}_q$, the set of vectors obtained by evaluating all polynomials of degree at most $k-1$ over $\mathbb{F}_q$ over $\alpha_1, \ldots, \alpha_r$ is the well-known Reed-Solomon codes of *dimension* k and block length r where the vector $(p_0, \ldots, p_{k-1})$ that define P is the message to be encoded.

[‡]We would like to emphasize that the connection between fuzzy vault and Reed-Solomon codes is well-known– in fact the paper of Juels and Sudan presented the fuzzy vault in the language of Reed-Solomon codes. The novelty in this paper is the connection to the hardness result from Kiayias and Yung.[12]

We are now ready to define the fuzzy vault. The fuzzy vault is defined by a subset $\mathbf{f} \subseteq \mathbb{F}_q$ (which denotes the set of minutia points in the fingerprint being "locked") and a (random) secret $\mathbf{s} = (s_0, \ldots, s_{k-1}) \in \mathbb{F}_q^k$, which denotes a secret polynomial $P_{\mathbf{s}}(X) \stackrel{\text{def}}{=} \sum_{i=0}^{k-1} s_i \cdot X^i$. The hash of $\mathbf{f}$ with secret $\mathbf{s}$ is the set $h(\mathbf{f}, \mathbf{s}) \stackrel{\text{def}}{=} \{(\alpha, P_{\mathbf{s}}(\alpha)) | \alpha \in \mathbf{f}\} \cup R$, where $R \subset \mathbb{F}_q \times \mathbb{F}_q$ is defined as follows. Each $(\alpha', \beta') \in R$ is chosen so that α is random (and has not been chosen yet) and β' is a uniformly random element in $\mathbb{F}_q \setminus \{P_{\mathbf{s}}(\alpha')\}$. We will use r to denote $|h(\mathbf{f}, \mathbf{s})|$.

We begin with the following simple observation:

PROPOSITION 1. *Given $h(\mathbf{f}, \mathbf{s})$ and one of $\mathbf{f}$ or $\mathbf{s}$, one can in polynomial time compute the other.*

Proof. Assume we know $\mathbf{f}$.[§] Then consider the pairs (α, β) for $\alpha \in \mathbf{f}$. Then perform polynomial interpolation[¶] to obtain $\mathbf{s}$.

For the other direction, assume that we know $\mathbf{s}$. In that case we know that for any $(\alpha, \beta) \in h(\mathbf{f}, \mathbf{s})$, we know that $\alpha \in \mathbf{f}$ if and only if $\beta = P_{\mathbf{s}}(\alpha)$. □

We now formally define the problem of breaking the fuzzy vault.

DEFINITION 1. *Let r, t, k be parameters then $\mathcal{V}_{r,t,k}$ is defined as the following computational problem. Let $T \subseteq \mathbb{F}_q \times \mathbb{F}_q$ be generated as follows. Pick $\mathbf{f} \subseteq \mathbb{F}_q$ as a random subset of size t. Let $\mathbf{s}$ be picked uniformly at random from $\mathbb{F}_q^k$. Then $T \stackrel{\text{def}}{=} h(\mathbf{f}, \mathbf{s})$ is generated as above. The goal is given $\mathbf{y}$, efficiently compute $\mathbf{s}$.*

Note that by Proposition 1, the goal of computing $\mathbf{s}$ is the same as computing (an approximation) of $\mathbf{f}$, which will constitute breaking the hash function for our application. Further, we note that we can assume (by Section 3) that the fingerprints when quantized to a subset in $\mathbb{F}_q$ given rise to a random subset of $\mathbb{F}_q$ (and hence in the problem above, $\mathbf{f}$ is picked uniformly at random).

The proof of security in Juels and Sudan (at a high level) follows by showing that given $h(\mathbf{f}, \mathbf{s})$ for $|\mathbf{f}| = t$, there are roughly $q^{k-t}\binom{r}{t}$ distinct pairs $(\mathbf{f}', \mathbf{s}') \neq (\mathbf{f}, \mathbf{s})$ that, when using the locking procedure of Juels and Sudan, could have resulted in the vault $h(\mathbf{f}, \mathbf{s})$. For an appropriate choice of the parameters (namely k close to t *and* r close to q), this leads to exponentially many possibilities, which in turn implies a strong information-theoretic/entropy based security. Unfortunately, for our more practical choice of parameters, this number actually turns out to be essentially 0. In other words, for our range of parameters, the pair $(\mathbf{f}, \mathbf{s})$ is the *only* pair that could have led to $h(\mathbf{f}, \mathbf{s})$. Thus, we do not get any information theoretic security through the fuzzy vault. However, this does not necessarily imply that the vault does not have any security. In fact, we argue next that even though $(\mathbf{f}, \mathbf{s})$ might be the only pair that leads to $h(\mathbf{f}, \mathbf{s})$, *computing* the pair (or by Proposition 1 either one) is still hard (subject to a known cryptographic hardness assumption).

Next, we begin with a quick recap of the hard problem from Kiayias and Yung[12] . Consider the following decoding problem

DEFINITION 2. *Given r, t, k and random distinct values $\alpha_1, \ldots, \alpha_r \in \mathbb{F}_q$ (for large enough q), consider the problem $\mathcal{RS}_{r,t,k}$, which is defined as follows. Let $\mathbf{y} \in \mathbb{F}_q^r$ be defined as follows. Pick a set $S \subseteq [r]$ uniformly at random from all sets of size t. For a random $P(X)$ of degree $k-1$ over $\mathbb{F}_q$, define $y_i = P(\alpha_i)$ if $i \in S$ and y_i is picked uniformly at random from $\mathbb{F}_q \setminus \{P(\alpha_i)\}$ if $i \notin S$. The goal is to compute $P(X)$ from $\mathbf{y}$.*

Kiayias and Yung conjecture the above problem to be hard.[12] To be precise, the hardness of the problem in Kiayias and Yung[12] holds for *every* set of distinct $\alpha_1, \ldots, \alpha_r$ (with the rest of the objects picked randomly as above). Thus, the above version of the problem where $\alpha_1, \ldots, \alpha_r$ are picked at random is also hard.[‖] The best known algorithm to solve the above problem is when $t > \sqrt{rk}$ due a *list decoding* algorithm due to Guruswami and Sudan[18]– in fact the latter algorithm works even if the "errors" are also worst-case. To be more precise, Kiayias and Yung[12] conjecture that the above problem of decoding Reed-Solomon codes from random errors is hard when $t < \sqrt{rk}$. In particular, the security parameter is defined to be

[§]Actually one only needs to know another set $\mathbf{f}' \subset \mathbb{F}_q$ such that $\mathbf{f} \cap \mathbf{f}' \geq \frac{k+|\mathbf{f}'|}{2}$.

[¶]In case we know only a noisy version $\mathbf{f}'$ of $\mathbf{f}$, we preform unique decoding of Reed-Solomon code by say the Welch-Berlekamp algorithm.[17]

[‖]Technically the hard problem in Kiayias and Yung[12] is a boolean version of the problem above but if we can solve the version above then we can also solve the boolean version.

$$\lambda \stackrel{\text{def}}{=} \sqrt{rk} - t, ^{**} \tag{1}$$

which we use as our measure of security for fuzzy vaults.

Kiayias and Yung also show the implications of the hardness assumption above. In particular, if the problem above is hard for probabilistic polynomial-time algorithm, then the vector $\mathbf{y}$ appears to be a random vector in $\mathbb{F}_q^n$ to such algorithms.

We now quickly argue why breaking the fuzzy vault would imply solving the hard problem from above.

PROPOSITION 2. *If there is a (probabilistic) polynomial-time algorithm to solve $\mathcal{V}_{r,t,k}$, then there is a (probabilistic) polynomial-time algorithm to solve $\mathcal{RS}_{r,t,k}$.*

Proof. We will prove a one to one correspondence between the different objects in $\mathcal{V}_{r,t,k}$ and $\mathcal{RS}_{r,t,k}$, which will immediately prove the claim.

The random secret $\mathbf{s}$ (or more precisely the corresponding polynomial $P_{\mathbf{s}}(X)$, which is a uniformly random polynomial of degree at most $k-1$ over $\mathbb{F}_q$) is in one to one correspondence with the random polynomial $P(X)$ in $\mathcal{RS}_{r,t,k}$. Consider only the "α" part of $h(\mathbf{f}, \mathbf{s})$ from $\mathcal{V}_{r,t,k}$, which corresponds to the set $\{\alpha_1, \ldots, \alpha_r\}$ in $\mathcal{RS}_{r,t,k}$. Further, the set S and $\mathbf{f}$ are in one-to-one correspondence as follows: $i \in S$ if and only if $\alpha_i \in \mathbf{f}$.

Finally, we consider the distributions in both settings. Note that the distribution on the α's is the same in both cases. In particular, picking $\alpha_1, \ldots, \alpha_r$ distinct random elements from $\mathbb{F}_q$ and then picking a random subset S of size t is the same as picking $\mathbf{f} \subset \mathbb{F}_q$ randomly and then picking the rest of $r - t$ α's as distinct random elements in the definition of $h(\mathbf{f}, \mathbf{s})$. Finally, note that for $i \notin S$ ($\alpha \notin \mathbf{f}$ resp.) the distribution on y_i is the same as the distribution on β, where $(\alpha, \beta) \in h(\mathbf{f}, \mathbf{s})$. In other words, the distribution on the "errors" is the same in both cases.

Thus, we have shown that all the objects in $\mathcal{V}_{r,t,k}$ are in one to one correspondence with the objects in $\mathcal{RS}_{r,t,k}$, as desired. □

3. IMPLEMENTATION

In this section we present the details of our implementation of the fuzzy vault scheme. We intend to be thorough in this discussion (while still being brief) to assist further research on this topic.

Center: The algorithm first reads all the fingerprints from the database and stores them in a data structure. As the fingerprints are read, they are centered about the common point of $(250, 250)$. By centering the points far from the origin, rotations and translations unlikely to produce negative values that can cause problems later in the algorithm. Centering each fingerprint also allows us to get close to the proper alignment without much cost.

Quantize: Each minutia point is then converted to a value in the field $\mathbb{F}_{2^{16}}$. This is done by converting the x, y, and θ values into binary strings of length 6, 6, and 4 respectively. We note that if we simply quantize the values by taking the most significant bits, the values will not be uniform in the field, as is required for the security proofs, since minutia points themselves are not uniformly distributed. In our analysis of fingerprint data, we observed that the x, y, and θ values closely follow a normal distribution. By assuming a normal distribution, we convert a value into a b bit binary string by dividing the normal curve into 2^b bins with equal probability and assigning each bin a unique binary value of length b. The x, y, and θ values are binned based on where they lie in the approximate distribution and are then concatenated as $\alpha = x \circ y \circ \theta$. We present some experimental justification in Section 4.

**We also need to pick the parameters such that the brute force algorithm is also negligible in λ. This is true for our settings.

3.1 Locking the Vault

To lock the vault with a fingerprint reading $\mathbf{f}$, the minutia points are first converted into the field as shown above to form the set $\{\alpha_i\}$. Then a secret polynomial, $P_{\mathbf{s}}(X)$, is generated by choosing k elements of $\mathbb{F}_{2^{16}}$ at random. The number of terms in this polynomial is our value k and can be varied between runs. In practice, this secret can be any data that needs to kept secure, such as an encryption key, but for testing purposes random data will suffice. The polynomial is evaluated at each of the genuine values and the points $(\alpha_i, P_{\mathbf{s}}(\alpha_i))$ are stored in the vault. Since the α_i are uniform in the field, the chaff points are created by choosing α' and β' uniformly at random in $\mathbb{F}_{2^{16}}$ and adding (α', β') to the vault with the condition that $\beta' \neq P_{\mathbf{s}}(\alpha')$. We also make sure that the α' are different from the α_i in $\mathbf{f}$. After the specified number of chaff points are added, which can vary from run to run, the vault is sorted by α to mix the chaff with the genuine points. The output of the locking function is the vault V which is a set containing the genuine and chaff points.

For evaluation purposes, the chaff points are marked and the secret is stored with the vault so everything can be checked after an attempted unlock. This will not be done in practice (and doesn't need to be), but it allows us to generate ROC curves that show how the vault will perform when deployed.

3.2 Unlocking the Vault

The unlocking algorithm is given a fingerprint $\mathbf{f}'$ and a vault V and attempts to recover the secret polynomial $P_{\mathbf{s}}$.

Alignment: When a fingerprint is used to attempt to unlock a vault, the alignment parameters are first computed from a baseline algorithm. The baseline matching results and the reference alignment information were obtained using a minutia based fingerprint matching algorithm utilizing secondary features from the triplets of neighboring minutia.[19] This is a simulation that can be replaced by techniques outlined in section 3.3. However, as we are primarily concerned with the security of the vault within a reasonable matching scheme, this simulation fits our needs. The genuine and impostor matchings both use this alignment data so as to ensure fairness. Other fuzzy vault implementations that use pre-aligned data typically don't give the impostors any help, which can artificially improve results.

Matching: We first convert the minutia points in $\mathbf{f}'$ to values in $\mathbb{F}_{2^{16}}$ in the same way we did for the locking set. We then perform a simple matching algorithm between the set of α_i and V. For each α_i, we find the closest value in V by the L_2 distance of the vectors $(x, y, \theta) \in \mathbb{R}^3$ as they are stored in the field. If the closest vault point is within a predetermined threshold, that point is added to a set S which will be the input for a Welch-Berlekamp decoder.[17] We use a threshold of an L_2-distance less than or equal to 2. This value has shown the best results among the threshold values tested. We also experimented with the idea of having a probability distribution such that the smaller the L_2-distance, the higher the probability the point would be added to S. We ran a genetic algorithm to find the best distribution with the condition that the probability distribution was monotone non-increasing in L_2 distance. We saw no remarkable change in the matching accuracy using these distributions and the genetic algorithms seemed to converge to the heuristically (via a sweep of threshold values) best threshold. [††]

Score: To compute the matching score of a given $\mathbf{f}$ and $\mathbf{f}'$, we use the number of genuine points minus the number of chaff points in the set of points S that results from attempting to unlock a vault with $\mathbf{f}'$ that has been locked by $\mathbf{f}$. This score accurately captures the ability of $\mathbf{f}'$ to unlock the vault, since the Welch-Berlekamp decoder will return the secret polynomial exactly when

$$t \geq \frac{n+k}{2} \tag{2}$$

where t is the number of genuine minutia in S, k is the number of terms in the polynomial, and $n = |S|$. We define c to be the number of chaff points in S, then we can rewrite the equation as

$$t - c \geq k \tag{3}$$

[††]In our evaluation of the thresholds, we used EER for FVC2002/DB1

where $t - c$ is our score function. Therefore, if the score is greater than or equal to the number of terms in the polynomial, the vault will unlock and vice-versa. Using this score, we generate FAR, FRR, and ROC curves where the threshold is the number of terms k in the polynomial. This allows us to see how a particular set of parameters for a vault will perform with respect to the size of the polynomial.

3.3 Improvements

Though this paper is mostly concerned about the security of the fuzzy vault scheme, it is worth noting that there are various techniques that can be used to improve the error rates of fuzzy vault and traditional matching schemes. Whether the schemes terribly decrease security is a point of interest, and is a topic for further research. We will list some of possibilities here.

Alignment Our implementation uses alignment data from a separate matching algorithm as described in section 3.2. This is sufficient for testing purposes; however, this is not possible in practice. To overcome this, several approaches have been successfully explored. Geometric hashes, hash functions that are rotation and translation invariant, have been used to construct vaults.[4,5] Using these hash functions eliminates any need to find the alignment data. Without using invariant hashes, the alignment data can be found by storing high-curvature data along with the locked vault as done by Nandakumar et al.[2] Another benefit of this approach is that it effectively adds another iteration to the matching algorithm which will reduce the FAR since most imposers will not match the high curvature stored and will be rejected without attempting to unlock the vault itself. The primary security concern with this technique is that additional information about each fingerprint is being stored in the vault. Since this data is not encrypted or obfuscated in any way, it can be freely used by an attacker as side information to crack the vault.

Descriptors Nagar et al.[11] showed that matching accuracy can be improved further by utilizing minutia descriptors. These descriptors capture data from the area surrounding each minutia point. This data is not stored in the vault, yet it used for locking and unlocking the vault. Since it is not stored, an attacker presumably must guess all the bits of this data so the entropy of the vault will be increased by the number of bits used. They show that the vault will have a reasonable of amount entropy in the average case with their setup. This method avoids many security issues by not storing this information in the vault. The primary security concern would be to ensure the data is close to uniform.

Cyclic Redundant Check(CRC) We mark the genuine and chaff points to obtain a matching score that relates to fingerprint's ability to unlock the vault. However, in practice, any matching set that is larger than the degree of the polynomial will have to be inputted into the Welch-Berlekamp decoder. If the decoder returns a polynomial, there will be no consistent way to tell if it is the secret polynomial. To overcome this, a CRC term can be added to the secret and used to check if the returned polynomial is in fact the secret desired.[2] For a CRC of length k, there is only a 2^{-k} probability that a random polynomial will satisfy the check. With high probability, a returned polynomial that satisfies the CRC is the secret polynomial. Adding this functionality will not affect the security of the vault, and so was not implemented in our vault, but it could easily be added if desired.

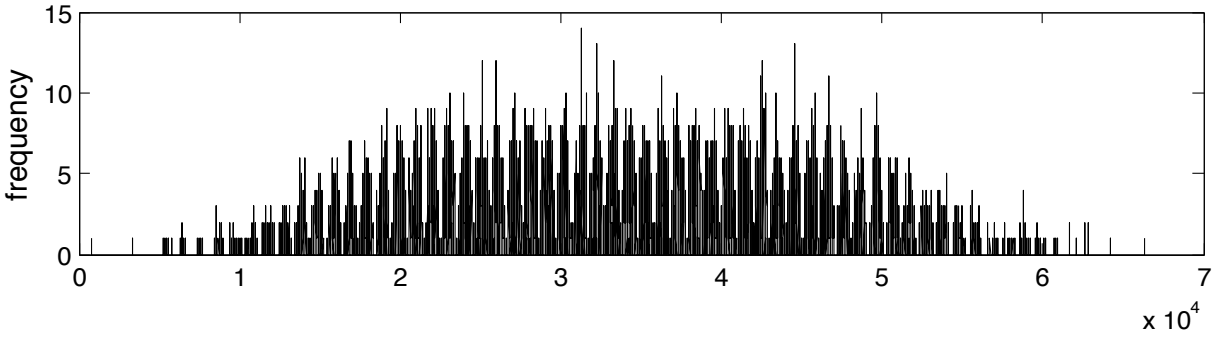
4. EXPERIMENTAL RESULTS AND DISCUSSION

We begin with a description of our experimental results on quantizing the minutiae into $\alpha \in \mathbb{F}_{2^{16}}$. The set of α values are then used to lock and unlock the vault. The resulting distribution of all the minutia points used in our experiments is shown in figure 4. For comparison, a min-max quantization of the same data is shown in figure 4. This data was computed by normalizing the data over the range from the min value to the max value, then dropping the least significant bits to arrive at the desired quantization. For both distributions 6, 6, and 4 bits were used for x, y, and theta respectively. We note that this algorithm can give slightly more uniform data than the method discussed in the introduction which quantizes over the entire scanner range since it eliminates any area around the edge of the scanner that contains no minutia points. This can only help the alternate distribution so we used it here to achieve a more fair comparison.

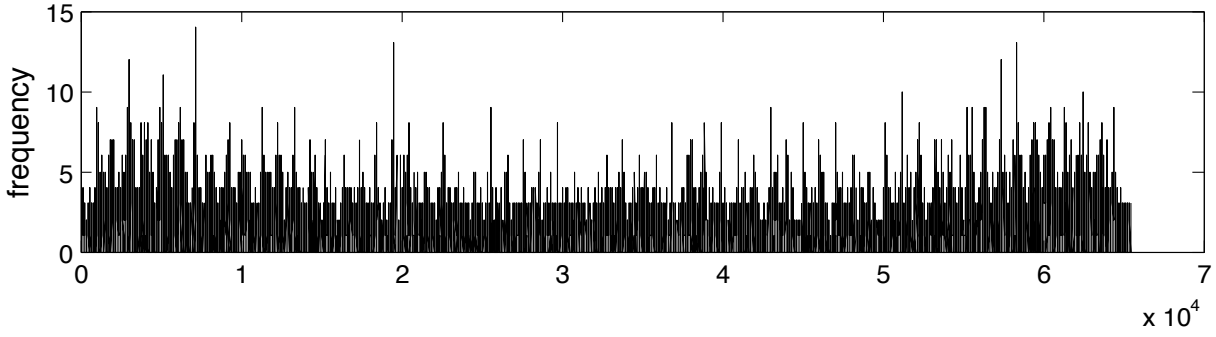
We conducted the experiments on the first fingerprint database from the Second International Fingerprint Verification Competition (FVC2002/DB1) (http://bias.csr.unibo.it/fvc2002/). The minutia positions were obtained by finding large curvature points on the contours extracted from the binarized fingerprint images.[20] Additionally, to improve the performance of minutia extraction, the fingerprints were enhanced by the short time Fourier transform algorithm.[21] A standard testing protocol for calculating all possible 2800 genuine (100 persons with $\frac{8\cdot 7}{2}$ matches) and 4950 (1 print of each person matched against 1 print of another, or $\frac{100\cdot 99}{2}$) impostor matches has been used. To ensure that there is no symmetry error in our results, each pair $(\mathbf{f}, \mathbf{f}')$ is tested with a vault locked by $\mathbf{f}$ and unlocked by $\mathbf{f}'$ and vice-versa.

The results for a run with 200 and 300 chaff points are shown in Figures 2 and 3, respectively. To compute the λ curves, we used the average number of minutia of all the readings in FVC2002/DB1, which is 45. We can see from Figures 4 and 4 that the fuzzy vault has a negative lambda if there are fewer than 9 terms in the secret polynomial with 200 chaff points, or fewer than 6 terms with 300 chaff points. A negative λ indicates that the entire vault can be used as input for the Guruswami-Sudan decoding algorithm[18] and it will successfully list decode to the correct polynomial. As λ increases above zero, the security of the vault increases exponentially.

For reference, the ROC for the baseline algorithm[19] is given in Figures 2(a) and 3(a). With this implementation, matching accuracy decreased in exchange for security. We notice that the fuzzy vault implementation outperforms the baseline algorithm at small values of FAR. This is partly due to the baseline algorithm assigning scores of 1 or 0 (the scores range from 0 to 1) under certain, somewhat common, conditions. This results in many-way ties at these two values and the fairly odd shape of the ROC curve. The performance of fuzzy vault

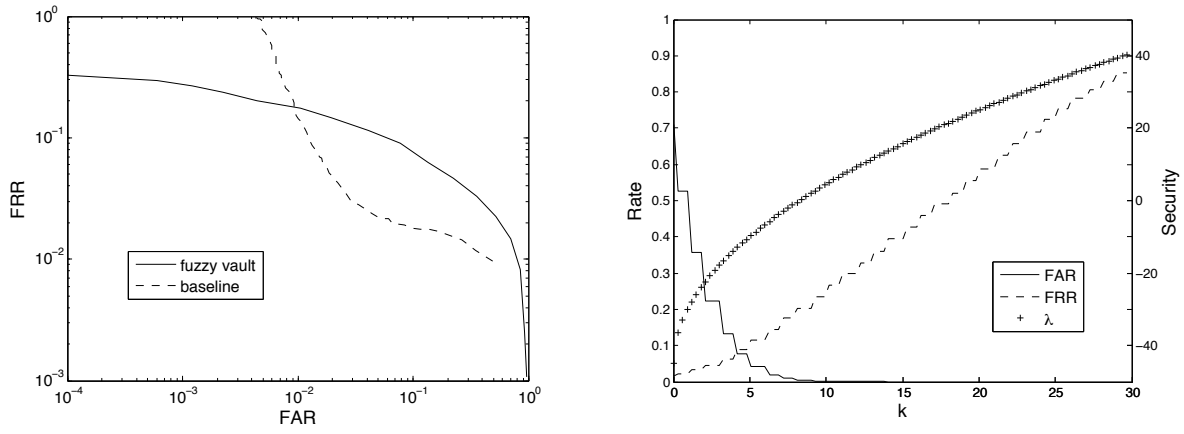


(a) Distribution using a simple min-max quantization.



(b) Distribution using our method of quantizing the minutia points.

Figure 1. A comparison of the distribution of genuine minutia points in $\mathbb{F}_{2^{16}}$. Our method achieves a distribution that is closer to uniform than standard quantization algorithms. The distribution is still noisy due to the nature of fingerprint data and we will explore new ideas to reduce this noise.

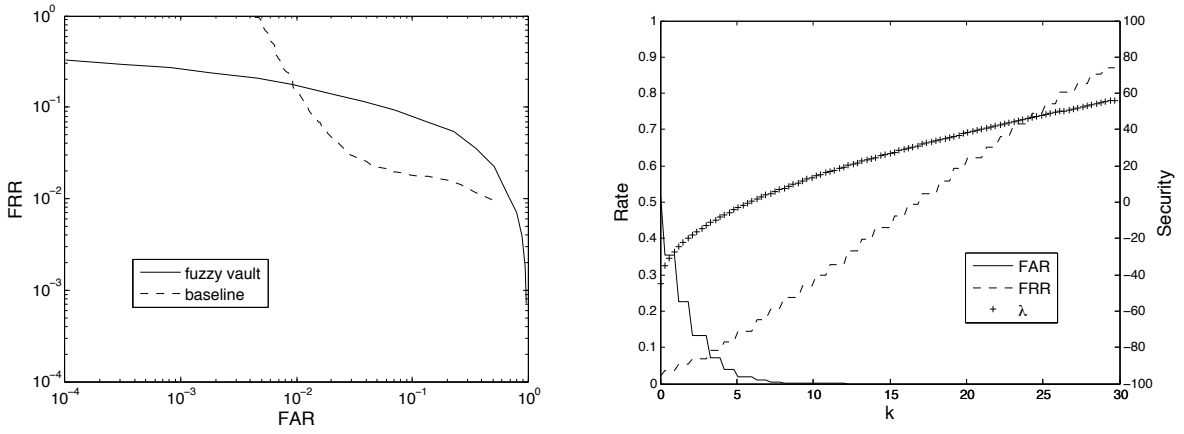


(a) (b)

Figure 2. A summary of the performance and security results for a vault with 200 chaff points. Figure 4 shows a comparison of the ROC curves of the vault and the baseline algorithm, while figure 4 shows a comparison of the matching performance and security of the vault. The vault has a positive λ for $k \geq 9$ indicating that smaller polynomials should not be used due to the lack of security. We computed λ using the average number of genuine minutia which is 45. The EER is 0.0856 and we note that there is no security at the EER.

seems to be comparable to other works[2,4,11] reporting results on different FVC2002 datasets. Nandakumar et al. reported some results of their fuzzy vault system on FVC2002 database 2. Though they do not give ROC curves, they show that they achieve an FAR of .024, .04, and 0 with FRR of .03, .04, and .1 respectively using a mosaiced template and 2 queries. For these results, their failure to capture rate was .01.[2] As mentioned in section 3.3, the addition of side infomation stored in the vault raises questions about the security of this method.

We also list the results for some specific polynomial powers in Table 1. The most accurate practical matching results we present here occurred with 200 chaff points and a polynomial with 9 terms. For these values we



(a) (b)

Figure 3. A summary of the performance and security results for a vault with 300 chaff points. Figure 4 shows a comparison of the ROC curves of the vault and the baseline algorithm, while figure 4 shows a comparison of the matching performance and security of the vault. The vault has a positive λ for $k \geq 6$ indicating that smaller polynomials should not be used due to the lack of security. We computed λ using the average number of genuine minutia which is 45. The EER is 0.0867 and we note that there is no security at the EER.

Table 1. Summary of results and security on FVC2002 DB1 for select choices of parameters. We note that λ is a logarithmic measure of security so the actual security grows exponentially with λ

	200 chaff points			300 chaff points		
# of terms in the polynomial(k)	FRR	FAR	λ	FRR	FAR	λ
9	0.2025	0.0044	1.96	0.2389	0.0018	10.72
11	0.2668	0.0012	6.91	0.2975	0.0003	16.60
13	0.3286	0.0001	11.44	0.3646	0	21.97

observed an FRR of 0.2025, FAR of 0.0044, and λ of 1.96. At this λ there is very little security for the vault. For a vault with 300 chaff points and a polynomial with 13 terms, we observe a λ of 21.97 and zero false accepts. This represents a fairly secure vault. The FRR for this setup was 0.3646. We note that at the EER there is no security.

We note that for our implementation, we gave the impostor readings the same alignment opportunity as the genuine readings. Since the matching algorithm[19] used for alignment favors the larger number of matched minutia pairs, it gives to the fuzzy vault a maximal possible subset of true, or non-chaff, minutia during unlocking. Such idealized alignment procedure could be equivalent to trying to unlock the vault by exploring all possible rotations and translations, and finding the one which delivers the maximum number of matched non-chaff minutia. This sometimes resulted in some rotations which were over 90 degrees and large translations. These transformations may not be realistic and gave the impostors much higher scores than they would without the helper data. Running the same setup without assisting the impostor matches effectively reduced the EER by a factor of two. We note that many implementations that pre-align readings only align the genuines and don't give any help to the impostors. We feel that giving the impostors the same chance as the genuines results in more realistic error rates. Also note that we do not discard the fingerprints with small number of detected minutia, which might have resulted in decreased performance compared to other published results which have non-zero Failure-To-Enroll rates.

5. FUTURE WORK

It is convenient to have λ as a security parameter as it quantizes fundamentally how hard it is to break a vault. However, this parameter is inconsistent between vaults because it depends on the number of genuine minutia points used to lock the vault, which varies between fingerprint readings. This makes it difficult to control the security of a fuzzy vault system in general and prompts the need to control the number of minutia used to lock the vault. By using multiple fingerprints for enrollment as in[7, 15] we can limit the effect that a single poor reading can have on the vault. We can require a user to provide multiple scans and combine these images to obtain an accurate template of the fingerprint. This will effectively increase the number of minutia we can use to lock the vault since more of the minutia points will be captured.

We will also explore the NIST MINDTCT minutia extraction algorithm[22] as an alternative to locating the points themselves. This algorithm has the advantage of a quality assessment of each minutia based on local image quality. Since we would be able to rank the minutia by quality, we can use the top j points by quality to lock the vault, where j is a chosen value for how many points to use. By combining multiple readings to increase the number of minutia, and quality ranking to limit the number of minutia, the number of genuine points in the vault, and thus λ, can be chosen. Since only the poor quality points are exempt from the vault, we expect the performance will not suffer considerably while the security of the vault will greatly increase.

Another concern that should be more adequately addressed in future work is how to protect against enrollment of a fingerprint in many different fuzzy vaults. As we mentioned earlier, this can lead to a straightforward attack, and more work must be performed to eliminate it.

Our future work will also include further experimentation using the genetic algorithm. This algorithm has been used to confirm that the matching threshold we are using is optimal, and we will apply the algorithm to other parameters as well. By applying this algorithm to the other parameters, we can find the optimal tradeoff between security and matching performance. We will also consider the avenues for future work discussed in Section 3.3.

6. ACKNOWLEDGMENTS

This research is supported by NSF grant TC 1115670.

REFERENCES

[1] Juels, A. and Sudan, M., "A fuzzy vault scheme," *Des. Codes Cryptography* **38**(2), 237–257 (2006).

[2] Nandakumar, K., Jain, A. K., and Pankanti, S., "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Transactions on Information Forensics and Security* **2**(4), 744–757 (2007).

[3] Scheirer, W. J. and Boult, T. E., "Cracking fuzzy vaults and biometric encryption," in [*Proceedings of Biometrics Symposium*], 1–6 (2007).

[4] Guo, X. Q. and Hu, A. Q., "The automatic fuzzy fingerprint vault based on geometric hashing: Vulnerability analysis and security enhancement," *Multimedia Information Networking and Security, International Conference on* **1**, 62–67 (2009).

[5] Lee, S., Moon, D., Jung, S., and Chung, Y., "Protecting secret keys with fuzzy fingerprint vault based on a 3d geometric hash table," in [*Adaptive and Natural Computing Algorithms*], Beliczynski, B., Dzielinski, A., Iwanowski, M., and Ribeiro, B., eds., *Lecture Notes in Computer Science* **4432**, 432–439, Springer Berlin / Heidelberg (2007).

[6] Chung, Y., Moon, D., Lee, S., Jung, S., Kim, T., and Ahn, D., "Automatic alignment of fingerprint features for fuzzy fingerprint vault," in [*Information Security and Cryptology*], Feng, D., Lin, D., and Yung, M., eds., *Lecture Notes in Computer Science* **3822**, 358–369, Springer Berlin/Heidelberg (2005).

[7] Tuyls, P., Akkermans, A. H. M., Kevenaar, T. A. M., Schrijen, G.-J., Bazen, A. M., and Veldhuis, R. N. J., "Practical biometric authentication with template protection," in [*Proceedings of the 5th international conference on Audio- and Video-Based Biometric Person Authentication*], *AVBPA'05*, 436–446, Springer-Verlag, Berlin, Heidelberg (2005).

[8] Boult, T. E., Scheirer, W. J., and Woodworth, R., "Revocable fingerprint biotokens: Accuracy and security analysis," in [*CVPR*], (2007).

[9] Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A., "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.* **38**(1), 97–139 (2008).

[10] Mihailescu, P., Munk, A., and Tams, B., "The fuzzy vault for fingerprints is vulnerable to brute force attack," in [*BIOSIG*], 43–54 (2009).

[11] Nagar, A., Nandakumar, K., and Jain, A. K., "Securing fingerprint template: Fuzzy vault with minutiae descriptors," in [*ICPR*], 1–4 (2008).

[12] Kiayias, A. and Yung, M., "Cryptographic hardness based on the decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory* **54**(6), 2752–2769 (2008).

[13] Hsu, R. and Martin, B., "An analysis of minutiae neighborhood probabilities," in [*Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*], 1 –6 (29 2008-oct. 1 2008).

[14] Chang, E.-C., Shen, R., and Teo, F. W., "Finding the original point set hidden among chaff," in [*Proceedings of the 2006 ACM Symposium on Information, computer and communications security*], *ASIACCS '06*, 182–188, ACM, New York, NY, USA (2006).

[15] Bringer, J., Chabanne, H., and Kindarji, B., "The best of both worlds: Applying secure sketches to cancelable biometrics," *Science of Computer Programming* **74**(12), 43 – 51 (2008). Special Issue on Security and Trust.

[16] Bringer, J., Chabanne, H., Cohen, G. D., Kindarji, B., and Zémor, G., "Theoretical and practical boundaries of binary secure sketches," *IEEE Transactions on Information Forensics and Security* **3**(4), 673–683 (2008).

[17] Guruswami, V., Rudra, A., and Sudan, M., [*Essential Coding Theory*] (2012). Draft available at http://www.cse.buffalo.edu/ atri/courses/coding-theory/book/index.html.

[18] Guruswami, V. and Sudan, M., "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory* **45**, 1757–1767 (1999).

[19] Jea, T.-Y., Chavan, V. S., Govindaraju, V., and Schneider, J. K., "Security and matching of partial fingerprint recognition systems," in [*SPIE Defense and Security Symposium*], (2004).

[20] Govindaraju, V., Shi, Z., and Schneider, J., "Feature extraction using a chaincoded contour representation of fingerprint images," in [*4th international conference on Audio- and video-based biometric person authentication*], **2688**, 268–275, Springer-Verlag, Guildford, UK (2003).

[21] Chikkerur, S., Cartwright, A. N., and Govindaraju, V., "Fingerprint enhancement using stft analysis," *Pattern Recognition* **40**(1), 198–211 (2007). doi: 10.1016/j.patcog.2006.05.036.

[22] Watson, C. I., Garris, M. D., Tabassi, E., Wilson, C. L., Mccabe, R. M., Janet, S., and Ko, K., "User's guide to nist biometric image software (nbis)."