IMPROVED FUZZY VAULT SCHEME FOR FINGERPRINT VERIFICATION

C. Örencik, T. B. Pedersen, E. Savaş and M. Keskinoz
Faculty of Engineering & Natural Sciences, Sabanci University, Istanbul, 34956, Turkey

Keywords: Fuzzy Vault, Template Protection, Biometrics, Fingerprint, Privacy.

Abstract: Fuzzy vault is a well-known technique to address the privacy concerns in biometric identification applications. We revisit the fuzzy vault scheme to address implementation, efficiency, and security issues encountered in its realization. We use the fingerprint data as a case study. We compare the performances of two different methods used in the implementation of fuzzy vault, namely brute force and Reed Solomon decoding. We show that the locations of fake (chaff) points in the vault leak information on the genuine points and propose a new chaff point placement technique that makes distinguishing genuine points impossible. We also propose a novel method for creation of chaff points that decreases the success rate of the brute force attack from 100% to less than 3.5%. While this paper lays out a complete guideline as to how the fuzzy vault is implemented in an efficient and secure way, it also points out that more research is needed to thwart the proposed attacks by presenting ideas for future research.

1 INTRODUCTION

Identification for access control and other purposes can be achieved by utilizing three factors: 1) what you know (e.g. passwords), 2) what you have (e.g. smartcards), and 3) what you are (biometric data identifying a person). Either these factors can be used alone or any combination of the three can be used together to increase security and compensate the weaknesses of one factor. While passwords can be forgotten and smartcards can be stolen, a biometric is inseparable from an individual and always accessible providing comparably high level of security. In addition, it can easily be combined with other factors to increase security further. Biometric identification, on the other hand, also suffers from two major drawbacks: 1) the noisy nature of biometrics measurement process and 2) privacy issues due to the fact that biometric data reveals private information about the individuals which is not intended to be revealed otherwise.

(Juels and Sudan, 2002) proposed the so-called *fuzzy vault* scheme to overcome these two drawbacks associated with biometrics usage in identification. The main idea is to exploit the relationship between error correction and secret sharing — the biometric data together with a *secret* defines a codeword from an appropriate error correction code. Given the fingerprint, the codeword can be corrected, and the secret is extracted. However, the secret does not reveal anything about the biometric data. If the secret is compromised, one can always choose another secret to combine with the same biometric.

A successful application of fuzzy vault to fingerprint biometrics is due to (Uludag et al., 2005) that basically uses the brute force approach. Different from Clancy's work they used alignment *helper data* which decreases the error rates, and also a Cyclic Redundancy Check (CRC) embedded in a coefficient of the secret polynomial is used to guarantee that the correct polynomial is found.

In this paper, we focus on several issues involving fuzzy vault implementation for biometrics usage. The first issue is to compare the computational efficiencies of the aforementioned two methods, namely brute-force and RS decoding methods. Another issue we deal with is to analyze some security drawbacks of the fuzzy vault scheme and propose solutions to those weaknesses as outlined below:

- The locations of the points in the vault may reveal some information as to which points are genuine depending on the chaff point generation. We propose a method in Section 4.1 that makes distinguishing genuine points impossible.
- Mihailescu (Mihailescu, 2007) pointed out that the fuzzy vault scheme is vulnerable to brute force attack. We propose a new method in Section 4.2 to

Örencik C., B. Pedersen T., SavaÈŹ E. and Keskinoz M. (2008).
IMPROVED FUZZY VAULT SCHEME FOR FINGERPRINT VERIFICATION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 37-43
DOI: 10.5220/0001917600370043

decrease the success rate of this attack from 100% to less than 3.5%. This countermeasure proves to be useful in certain settings.

- We study the effects of distances between chaff points and between a chaff and a genuine point on the security of the fuzzy vault.
- We also study limitations on the vault size and its effects on the security and performance of the fuzzy vault.

The rest of the paper is organized as follows. Section 2 gives a review of the fuzzy vault scheme and explains the most important details of the brute force and Reed Solomon decoding algorithms (Roth, 2006) used for reconstructing the secret polynomial hidden in the fuzzy vault. Section 3 provides comparative analysis for the performance of two techniques used in polynomial reconstruction. Section 4 outlines two proposed modifications to the enrollment stage to increase the security of the fuzzy vault and summarizes the security analysis of the scheme against brute force attack. Section 5 explores the effects of the vault and threshold sizes on the performance and security of the fuzzy vault using experimental data. It also provides a timing comparison between brute force and RS decoding methods. Section 6 is devoted to the summary of the paper and proposes a new idea for future work.

2 REVIEW OF FUZZY VAULT

In this section we give a brief outline for the techniques used in the application of fuzzy vault scheme to fingerprint biometrics. The identification process using the fuzzy vault consists of two major stages: the enrollment and verification. In the enrollment stage, the fuzzy vault is created by embedding a secret polynomial after the fingerprint of the user is obtained. The fuzzy vault hides the fingerprint and the secret polynomial, which can be revealed if the same finger is used in verification. The verification stage contains two phases: 1) the alignment of the measured fingerprint to the points in the fuzzy vault, and 2) the reconstruction of the secret polynomial. The enrollment and alignment stages are the same for both brute force and RS decoding methods that differ in the polynomial reconstruction phase. The details of enrollment and allignment stages are provided in (Juels and Sudan, 2002), (Clancy et al., 2003), therefore only the polynomial reconstruction stage is described briefly in the following two sections.

2.1 Brute Force Approach

To reconstruct the secret polynomial using brute force approach requires trying all the combinations of size k given m matching minutiae points. Note that some of the m matching minutiae points are the ones that actually match random chaff points in the fuzzy vault. When k minutiae points that match the real minutiae points are found during the exhaustive search, the scheme is said to be successful.

In brute force approach, first k pairs of (x_i, y_i) are chosen randomly from the verification list and the polynomial on which the selected k pairs lie is calculated using Lagrange interpolation method. Then whether μ of the remaining vault points satisfies $y_i = p(x_i)$ is tested. If more points that lie on the same polynomial are found, the fingerprint is verified; otherwise rejected. If insufficient number of pairs satisfy $y_i = p(x_i)$ condition, another random k pairs are taken as input and the process is repeated. The maximum number of trials is set to a high value, after which the program rejects the fingerprint if no polynomial satisfying the condition is found. The drawback of the brute force approach is high computation complexity when the tested fingerprint is too noisy.

2.2 Reed Solomon Decoding Approach

When we construct the fuzzy vault we evaluate the secret polynomial for all n minutiae points, i.e. $y_i = p(x_i)$ for $i = 1, \ldots n$. This can be put into a matrix-vector formulation as follows:

$$\left[\begin{array}{cccc} y_1 & y_2 & \ldots & y_n \end{array}\right] = \left[\begin{array}{cccc} p_0 & p_1 & \ldots & p_{k-1} \end{array}\right] \mathbf{G}$$

where the matrix **G** is the generator matrix (Roth, 2006).

It is crucial to notice that the generator matrix changes for each user, which differ from the conventional application of RS encoding method. Since the enrollment stage essentially utilizes the RS encoding, the reconstruction of the secret polynomial in the verification stage can be achieved by employing an *RS-Decoder*.

Utilizing error correcting codes for the implementation of fuzzy vault is first proposed by (Juels and Sudan, 2002). The authors state that after matching minutiae points are obtained, use of Reed-Solomon (RS) decoder is a more efficient approach than the brute force. The Reed-Solomon decoders have an error correction capability of $\tau = \frac{m-k}{2}$ errors. The best choice to implement RS decoder is to use the Berlekamp-Massey (BM) algorithm as explained also in (Clancy et al., 2003) since it is fast, easy to implement and widely studied.

The RS decoding with BM algorithm takes two vectors ($\mathbf{v} = [x_1, x_2, \ldots, x_m]$ and $\mathbf{y} = [y_1, y_2, \ldots, y_m]$) as explained previously and returns the error locater polynomial (ELP). Since *ELP* shows the locations of all errors, the rest of the data must be correct.

Finally, the secret polynomial can be reconstructed by using the Lagrange interpolation method with the correct minutia points if the number of errors does not exceed τ. Otherwise, the function returns a wrong polynomial of degree $k-1$. Again we check if more points that lie on the same polynomial exist. Otherwise, the function is called with fewer number of pairs. This process is repeated several times with some of the different random pairs being removed from the list. If the algorithm still returns a wrong polynomial as output after several attempts, then the fingerprint is rejected.

3 COMPUTATION COMPLEXITY OF POLYNOMIAL RECONSTRUCTION

In (Clancy et al., 2003), Clancy et al. argue that using the RS decoder is a better approach than the brute force method if the attacker cannot eliminate some of the chaff points from the verification list. But the authors do not provide a comparison between the two approaches. In this paper we try to clarify as to which method is optimal depending on the parameters of m and k where m is the number of matched points and k is one more than the degree of the secret polynomial.

For comparing the two approaches, we calculate the number of operations in the secret polynomial reconstruction phase for both methods. For sake of simplicity, we ignore addition and assignment operations and only count multiplication and inverse operations in F_q since the latter two operations dominate the computation.

3.1 Complexity of RS Decoder

The Reed Solomon decoder has four steps as explained in (Roth, 2006) and complexity of each step and the total complexity is given in Table 1. We assume that Step 3 always returns an error locater polynomial; i.e. the measured fingerprint always leads to matchings to chaff points.

From the perspective of complexity comparison, the main difference between the brute force and the RS decoding approaches is that, RS decoder can distinguish a genuine fingerprint in only one trial if the number of incorrect matchings is less than the error correcting capability of the RS code τ. On the other hand, the brute force approach may have to perform excessively many trials to complete the verification process.

Table 1: Operational Complexity of RS Decoding Method.

Step	Multiplication	Inv.
1. Constructing H	$3m^2 - 2mk$	m
2. Syndrome Computation	$m(m-k)$	-
3. Finding Error Locations	$m(k^2/3+2)$	-
4. Polynomial Construction	k^2	-
Total	$4m^2 + m(k^2/3)$ $-m(3k+2)+k^2$	m

3.2 Complexity of Brute Force Method

Complexity of the brute force method is given in Table 2. Selecting k random points out of m matched points (i.e. Step 1 in the table) involves a randomized algorithm, whose complexity we estimate as equivalent to k multiplication operations. The variable l in the last row of Table 2 stands for the number of trials needed on average, which naturally increases with the error in the tested fingerprint. Without knowing the number of trials l in the brute force method it is not easy to compare two methods. Comparison is only possible with experiments on real and synthetic data, which we achieve in Section 5.

Table 2: Operational Complexity of Brute Force Method.

Phase	Multiplication	Inv.
1. Choosing k random points	k	-
2. Polynomial Construction	k^2	-
3. Verification of the result	$5m$	-
Total	$l(k^2+5m+k)$	-

4 NEW ENROLLMENT STAGE

In this section, we explain two proposed modifications to the enrollment stage in order to strengthen the fuzzy vault against possible attacks.

4.1 Distribution of Chaff Points

Creation of random chaff points is crucial since they should be uniformly distributed in the minutiae space so that an attacker, having access to the vault, should not be able to distinguish between genuine minutiae points and random chaff points (Kholmatov et al., 2006). If the chaff points were created with the condition that every point in the vault should be at least

t Euclidean distance apart to supply a uniform distribution as proposed by (Juels and Sudan, 2002), the *fuzzy vault* could leak some information about the location of genuine points. We have no control over the locations of genuine points, as some of them might be very close to each other as exemplified in Figure 1 where chaff points are represented as circles and genuine points as circles with crosses. If an attacker intercepts a fuzzy vault, he can locate some of the genuine points correctly by checking the distances between the points; i.e. if the distance between two points is closer than the threshold t, then these points are genuine.

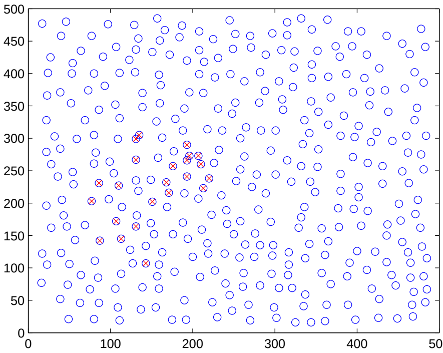


Figure 1: Fuzzy vault as implemented described in (Juels and Sudan, 2002) where genuine points are marked.

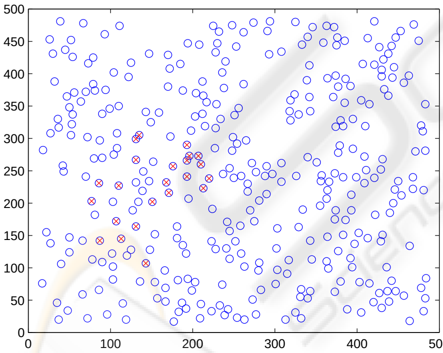


Figure 2: Fuzzy vault with chaff points being placed according to a new scheme.

As a remedy, we generate the chaff points with the condition that every chaff point in the vault should be at least t Euclidean distance apart from a genuine point and should be at least t' Euclidean distance apart from any other chaff point. Note that $t' < t$ since having chaff points far from the genuine points are desirable and have a positive effect on false reject rate (FRR) as demonstrated in Section 5. Smaller threshold t', on the other hand, for inter-chaff point distance is necessary to imitate the distribution of genuine points where close genuine points occasionally occur in the vault. While t value depends on the fingerprint image size and the total number of points in the vault, t' should be chosen depending on the distribution of genuine points. For finger print image of 500 pixels, Figure 2 shows the fuzzy vault constructed with the new chaff point placement strategy, where the threshold values t and t' are chosen as 18 and 8, respectively. As seen in Figure 2, the distribution of chaff points in the fuzzy vault closely resembles the distribution of genuine points, hence an attacker cannot easily distinguish the genuine points from the chaff points.

4.2 A Novel Method for Chaff Point Placement

Note that there are C points in the vault where n of them are genuine and the secret polynomial has degree $k-1$. Mihailescu proved that in less than $8Ck(C/n)^k$ operations[1], the intruder can recover the secret polynomial (Mihailescu, 2007). The idea of the attack relies on the established fact (Juels and Sudan, 2002) that when there are more than D vault points on a polynomial of degree $k-1$ for $D \in (k-1, n)$, this polynomial is the secret polynomial with a very high probability.

Our proposed method to improve the security involves the idea that, by choosing the chaff points at random, but in a more clever way, we can embed some other (randomly chosen) polynomials of degree $k-1$ other than the secret polynomial in the vault. If we guarantee that the number of chaff points that lie on these (fake) polynomials, is around n - the same number of genuine points on the secret polynomial on average - the attacker cannot distinguish the secret polynomial from the fake ones. Otherwise the attacker who succeeds to construct a polynomial can discard it if there are fewer points. One way of choosing non-random points is described below:

1. Place the genuine points in the vault
2. Keep the unassigned chaff points in a pool
3. Keep a list of fake polynomials which is initially empty
4. Repeat until the pool is empty
   (a) Pick a random number r close to n
   (b) For the first fake polynomial, take random $k-1$ points from the vault and take one random point

[1] By operation it is meant that atomic arithmetic operations such as additions and multiplications.

from the pool. For others take random k points from the vault.

(c) Find the $(k-1)^{st}$ degree polynomial that passes through the selected points. Add the polynomial to the list if it is not already in it.

(d) Check the vault if there are any other points that lie on the polynomial. Decrement r by the number of points on this polynomial.

(e) Pick r points from the pool (or the remaining points if their number is less than r) and evaluate these points on the fake polynomial and place the resulting values in the fuzzy vault.

With the proposed chaff placement method, we allow each polynomial intersect with other polynomials in at least k vault points which increases the maximum number of polynomials we can embed into the vault. Note that any two polynomials cannot intersect with each other in more than $k-1$ points. As a result of our experiments in our setting described above, we are able to hide around 30 fake polynomials in the vault. Therefore, this method decreases the probability of finding the secret polynomial using Mihailescu's attack from 100% to approximately 3.3% after the brute force attack is applied. Due to the fact that most of the identification applications allow only limited number of trials, the proposed method enhance the security considerably. Moreover, the method does not affect the false accept or false reject rates since the matching algorithm considers only the x coordinates of the points and this method changes only the y coordinates.

4.3 Security Analysis

The attacks on the fuzzy vault scheme, mostly assume the interception of a vault from a database. The basic attack is the brute force attack over a single vault. The analysis in this work based on the work of Mihailescu (Mihailescu, 2007), shows that this attack is not computationally infeasible, therefore fuzzy vault scheme is insecure without additional security.

If an attacker intercepts a vault, but has no other information about the locations of the genuine points, the best method to recover the secret polynomial is the brute force trials (Clancy et al., 2003)(Mihailescu, 2007). Mihailescu provides a strong brute force attack in (Mihailescu, 2007), which finds the secret polynomial in less than $8(Ck)(C/n)^k$ operations where C is the number of points in the vault, n is the number of genuine points in the vault and the degree of the secret polynomial is $k-1$.

In our tests n parameter is on the average 35 and k is constant 10. For $C = 300$, which gives a better FRR, breaking the system requires $8 \times 300 \times 10 \times (300/35)^{10} \approx 2^{46}$ operations. For $C = 350$, which gives a worse FRR, the system provides a better security; breaking the system requires this time $8 \times 350 \times 10 \times (350/35)^{10} \approx 2^{48}$ operations.

Without the use of the proposed method in section 4.2, the secret polynomial is found with probability 1 after this attack. However, our proposed method decrease the probability to approximately 0.03 since the polynomial found as a result of brute force attack is not guaranteed to be the secret polynomial.

5 TEST RESULTS

We implement polynomial reconstruction phase using two previously discussed approaches: 1) brute force method and 2) RS decoding.

For the implementations, we use a database of 180 people where there are two fingerprint images for each finger, totaling 360 fingerprints. The first 180 fingerprint images are used for enrollment and the second 180 images are used for verification of the corresponding fingerprints. Later, all fingerprints are cross-tested for false accept rates. In the experimental setting bitmap images of 500 × 500 pixels are created for each fingerprint.

All computations and tests are performed on a computer with 1.7 GHz Intel Celeron M processor and 448MB of RAM. The codes are developed in either Matlab or C++ (Microsoft Visual Studio) depending on the nature of the problem.

We basically investigate two issues; firstly, the effects of the vault and threshold sizes on the performance and security of the fuzzy vault, and secondly time efficiencies of two methods used in the polynomial reconstruction phase of the verification stage.

5.1 Effects of Vault and Threshold Sizes

The false reject rates (FRR) and false accept rates (FAR) are calculated in four settings where different values for vault size and minimum distance threshold are used for our database of fingerprints. We use vault sizes of 300 and 350 points and minimum distance thresholds of $(t = 15)$ and $(t = 18)$. The minimum distance between any two chaff points is taken as 8. We calculate the FAR and FRR results for both the brute force and the RS decoding methods.

The FAR rates turn out to be 0% in all settings after cross testing all fingerprint images with different fingers in four settings.

Table 3 shows the FRRs for different vault sizes and threshold values. The results clearly demonstrate

Table 3: FRR for four different settings.

	Vault Size	
Threshold (t)	300	350
15	6.5%	9%
18	1.5%	4.5%

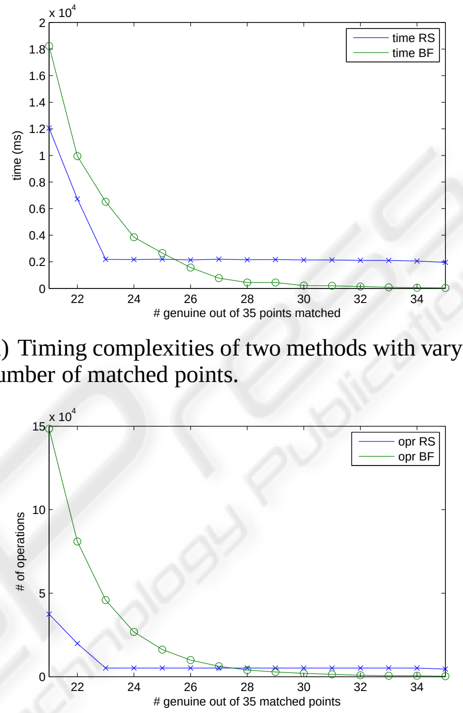
that as the minimum distance between two points increases, the possibility of a genuine point matching to a chaff point decreases resulting in lower FRRs. Although the two values of threshold used in the experiments have the same security level, increasing it further may become impossible after certain point since we cannot place as many points as we desired. Similarly, when more chaff points are added to the vault, the possibility of a genuine point matching to a chaff point increases. Larger vault size results in higher security. The security impact of vault size was analyzed in Section 4.3.

5.2 Timing Results for Polynomial Reconstruction Phase

As explained before, verification phase is the main part where two approaches are compared in terms of timing and success performance. The "Number Theory Library" (NTL) (Shoup, 2008) is used for all the operations in the polynomial reconstruction. For both approaches, the polynomial reconstruction algorithms take two vectors $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ where $y_i = P(x_i)$ for the points matched to a genuine point and $y_i \neq P(x_i)$ for the points matched to a chaff point.

We first compare the timing results of the two approaches using our database of real fingerprints. In case of true fingerprints we find that for 15% of the data, the RS decoding approach is faster. For forgery data, RS decoding method is naturally always faster since concluding that a fingerprint is a forgery takes 1000 trials in our experiments.

Considering that the database used in the experiments may not fully represent all cases, especially the ones with high levels of noise in measurement process, we use synthetic data to control the number matched points in the alignment process. We generate synthetic fingerprints and corresponding fuzzy vaults of 350 total points, where the number of matched points are in the range of $[21, 35]$. Timing results of the experiment for changing number of matched points are shown in Figure 3(a). As clearly observed in the figure, the brute force method takes much longer when the number of matching points are low due to excessive number of trials to find the correct matching points. As the number of matching points increases, the brute force becomes faster. We also provide the same graphic for number of operations in Figure 3(b), that we obtain in our theoretical analysis. The similarity of two figures show that the experimental results are in line with the theoretical analysis



(a) Timing complexities of two methods with varying number of matched points.

(b) Operational complexities of two methods with varying number of matched points.

Figure 3: Comparison of complexities of two methods.

The experiments demonstrate that the optimal method changes depending on the number of genuine points matched. The brute force approach is faster if the matching is very good (i.e. most of the minutiae points are matched to genuine points) since it will require very few trials to find the polynomial. On the other hand, the RS decoder is very fast to reject a forgery since brute force will require excessive number of trials to decide on a reject. Also for a weak matching of a valid fingerprint, the RS decoder is again faster.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the implementation, security, and performance issues of fuzzy vault for biometric identification. We first provided a guideline that briefly explains the implementation steps of decoding using two methods: brute force and Reed-Solomon (RS). We then compared the efficiencies of the two methods. The results shows that for weak matching, the RS decoding method is more efficient; but for good matching, the brute force method works faster. For the success rate of verification, they both give the same false accept rate (FAR) and false reject rate (FRR).

We proposed a new chaff point placement method to prevent some inferences on the location of genuine points. By adjusting the distance between points (genuine-to-chaff and chaff-to-chaff) we showed that it is possible to increase security and performance of the fuzzy vault implementation.

We explored the effects and limitations of the vault size on the security and performance of the fuzzy vault. The higher number of chaff points in the vault is demonstrated to strengthen the method against the most successful attack. However, placing more chaff points takes more time and becomes impossible after certain number and has an adverse effect on the FRR rate.

We proposed to embed a number of fake polynomials in the fuzzy vault along with the secret polynomial to reduce the success rate of the attacker. We succeeded in placing only limited number of fake polynomials in the vault. We believe that further research will reveal more efficient methods to place higher number of fake polynomials in the vault, that will further increase the security.

Other than the brute force attack, there is another attack that can be applied in the presence of two vaults that belong to the same biometric (Kholmatov and Yanikoglu, 2008). The correlation attack basically utilizes two vaults of the same fingerprint to reveal most of the minutiae points by cross-matching the two vaults. The correlation attack takes advantage of the fact that minutiae point locations are almost the same in two vaults and increasing the vault size has only limited effect on the security. A more effective solution will be keeping a distorted version of the biometric that preserves the invariants of the biometric image. For instance, Sutcu et al. (Sutcu et al., 2005) proposed a method that uses Gaussian function for distortion of the biometric. We plan to use a special hash function that gives similar outputs for inputs that differ by a few bits. By keeping the hash values of the fingerprint minutiae values and performing matching in the hash space, we can avoid the correlation attacks. The effectiveness of this method remains to be analyzed which we plan to pursue as future work.

ACKNOWLEDGEMENTS

We would like to thank Eren Camlikaya for his fingerprint image processing codes. We also would like to thank TUBITAK (The Scientific and Technical Research Council of Turkey) for the M.Sc. fellowship supports granted to Cengiz Orencik.

REFERENCES

Clancy, C., Kiyavash, N., and Lin, D. (2003). Secure smartcard - based fingerprint authentication. In *ACM Workshop on biometric methods and applications, (WBMA)*.

Juels and Sudan, M. (2002). Fuzzy vault scheme. In *IEEE International Symposium on Information Theory*, page 408.

Kholmatov, A. and Yanikoglu, B. (2008). Realization of correlation attack against fuzzy vault. In *Security, Forensics, Steganography and Watermarking of Multimedia Contents X, Electronic Imaging*, San Jose CA, USA.

Kholmatov, A., Yanikoglu, B. A., Savas, E., and Levi, A. (2006). Secret sharing using biometric traits. In *Biometric Technology For Human Identification III*, volume 62022006, Orlando, Florida USA. In Proceedings of SPIE.

Mihailescu, P. (2007). The fuzzy vault for fingerprints is vulnerable to brute force attack. http://arxiv.org/abs/0708.2974v1.

Roth, R. M. (2006). *Introduction to Coding Theory*. Cambridge University Press.

Shoup, V. (2008). Ntl: A library for doing number theory.

Sutcu, Y., Sencar, H. T., and Memon, N. (2005). A secure biometric authentication scheme based on robost hashing. In *Proceedings of the 7th workshop on multimedia and security*, NY, USA.

Uludag, U., Pankanti, S., and Jain, A. (2005). Fuzzy vault for fingerprints. In *Proceeding of International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 310–319.