

Security Issues of Biometric Encryption

A. Stoianov

Office of Information and Privacy Commissioner of
Ontario, Toronto, Canada
alex.stoianov@ipc.on.ca

T. Kevenaar, M. van der Veen
priv-ID B.V.

Eindhoven, The Netherlands
{tom.kevenaar, michiel.van.der.veen}@priv-id.com

Abstract — Security issues of Biometric Encryption are discussed in terms of resilience to the attacks. A number of previously unpublished attacks are considered: Nearest Impostors; using Error Correcting Code statistics; and Non-randomness attack.

Keywords — Fuzzy Extractor; Hill Climbing; fingerprints; iris; soft decoding; erasures

I. INTRODUCTION

A new area of research called Biometric Encryption (BE) has been developed since mid-90's [1, 2, 3] (other terms for BE are Biometric Cryptosystem, Fuzzy Extractor, Secure Sketch, Helper Data System, Biometric Locking, Biometric Key Generation, etc.). BE can be defined as a group of technologies that securely bind a digital key to a biometric or extract a digital key from the biometric. Instead of a biometric image or template, so-called 'helper data' is stored and possibly a hashed version of the key. It is computationally difficult to retrieve either the key or the biometric from the stored helper data; the key is recreated only if the correct biometrics is presented on verification. Thus, the output of BE verification is either a key (correct or incorrect) or a failure message. This "encryption/decryption" process is fuzzy because of the natural variability of biometric samples.

BE systems, that bind the key with the biometric on a fundamental level, can enhance both privacy and security of a biometric system. In general, BE is less susceptible to high level security attacks on a biometric system [4], such as substitution attacks, tampering, Trojan horse attacks, overriding Yes/No response, masquerade attack, etc. BE can work in non-trusted or, at least, in less trusted environment. The random keys are usually longer than conventional passwords and do not require user memorization. Unlike conventional biometric templates, the BE helper data are revocable.

Some low level attacks against BE systems are still possible, though. In the low level attacks an attacker is supposed to be familiar with the algorithm and to be able to access the stored helper data. The attacker can also collect or generate a biometric database to use it offline. It is assumed that the attacker does not possess a genuine biometric sample. The attacker's objectives are to obtain the key or to reduce the search space, and/or to obtain a biometric or a masquerade version of it.

With a few exceptions [5 - 8], the security aspects of BE have been largely overlooked. In many cases the theorems that deliver a formal proof of security for a BE scheme are based on unrealistic assumptions about the biometric samples (e.g., the feature sets are considered random and statistically independent), or impose arbitrary constraints on the approaches that could be employed by an attacker. Very few publications deal with practical aspects of the attacks. In our view, designing attacks against a specific BE system should be an integral part of any BE algorithm development, preferably, at early stage. If vulnerabilities are found, the remedies may be put in place.

In this paper, we provide descriptions in practical terms for some previously unpublished attacks on BE systems. As a model for illustration purposes, we chose the first practical BE algorithm (called Mytec2) [9] and discuss the applicability of the attacks to Fuzzy Commitment and some other BE schemes.

II. ATTACKS ON BE SYSTEMS

A general overview of attacks on BE is given in [3]. The most important attacks include:

Inverting the hash; False acceptance (FAR) attack; Hill Climbing attack [5]; Nearest Impostors attack; Running Error Correcting Code (ECC) in a soft decoding and/or erasure mode; ECC Histogram attack; Non-randomness attack against Fuzzy Vault [7]; Non-randomness attack against Mytec2 and Fuzzy Commitment; Re-usability attack [6, 8]; and Blended Substitution attack [8].

Besides obvious attacks (inverting the hash and obtaining a false acceptance), the only published practical attacks are Hill Climbing [5] against Mytec2 and Re-usability against Fuzzy Vault [10]. Other attacks that have been considered include Non-randomness attack against Fuzzy Vault [7] and quantization schemes [11], and Blended Substitution [8].

FAR attack is conceptually the simplest. The attacker needs to collect or generate a biometric database of a sufficient size. The attack can be mitigated by using a secret transform [1], preferably controlled by a user's password; by using slowdown functions; and by using BE within a framework of a homomorphic cryptosystem [12]. Note that the latter would provide an ultimate solution to most BE security problems.

The Re-usability attack was shown [10] to be successful against Fuzzy Vault. On the other hand, Boyen proved [6] that the Fuzzy Commitment scheme is secure if the ECC is linear (and, we can add, if there is no component selection or

masking). Mytec2 and other schemes that use the application-dependent “salting” are inherently immune.

The blended substitution attack, which is rather of secondary nature, may work for Fuzzy Vault. Both Fuzzy Commitment and Mytec2 schemes are much more resilient to the attack. The details for the Re-usability and the blended substitution attacks will be published elsewhere.

In the following sections, we present the previously unpublished results for the Nearest Impostor attack, the attacks using ECC statistics, and the Non-randomness attack.

III. MYTEC2 AND FUZZY COMMITMENT BE SCHEMES

A. Mytec2 [9]

Mytec2, which was proposed in 1997, is a predecessor of many other BE schemes, such as Fuzzy Commitment and “BioHashing”. We will give more detailed description of a modified Mytec2 scheme because it will be used as a model for several attacks described in this paper. Mytec2 shares many common vulnerabilities with other BE schemes.

The scheme, which is applicable to fingerprints, can be designed in the following steps:

- Several images of the same finger are captured, and a composite image, $f(x)$, is created (see details in Ref. [9]);
- The Fourier transform, $F(u) = |F(u)| \cdot \exp(i\phi(u))$, of the image is obtained;
- A random phase, $\phi_{\text{rand}}(u)$, is generated, and a phase-only filter, $H(u) = \exp(i\phi_{\text{rand}}(u) - i\phi(u))$, is obtained;
- $\exp(i\phi_{\text{rand}}(u))$ is multiplied with a magnitude part of the “fingerprint composite filter” (which is transitory, i.e. not stored) and the inverse Fourier transform is taken to obtain an output pattern, $c_0(x)$. It is also supposed to be random;
- A 131-bit key is randomly generated and is mapped to the ECC codeword. The ECC consists of two consecutive ECCs: a (23, 1) repetition code (which is run in 255 chunks) followed by the (255, 131) BCH code;
- The ECC codeword is linked to the “the most reliable bits” of $c_0(x)$ via a lookup table;
- The stored helper data include the phase-only filter, $H(u)$, the lookup table, and the hash of the key.

On verification, the Fourier transform of a new fingerprint sample, $F_I(u)$, is multiplied with the stored filter $H(u)$ and the new transitory magnitude part to obtain an output pattern, $c_I(x)$. Then the bits are extracted at the locations stored in the lookup table, and the ECC decoder obtains an output key. The process is repeated until a correct hash is found or the search is exhausted after checking about 1000 shifts of the image (this is done to compensate for image misalignment).

The Mytec2 scheme is quite similar to the Fuzzy Commitment scheme [13] that appeared later, including selection of the most reliable components, or to more general Fuzzy Extractors [14]. However, in Mytec2, the obfuscation part (i.e. creating a filter $H(u)$) and the coding part are separate (there is an inverse Fourier transform in the middle). Note that the product of two phase-only functions is an analog equivalent of XOR operation in the Fuzzy

Commitment scheme. The linkage of the ECC codeword through a lookup table is similar to what was called a “Permutation-based Fuzzy Extractor” [14]. There is also a similarity to BioHashing or “Biometric salting” [1]. However, in Mytec2 the random phase is not kept secret or even stored anywhere.

B. Fuzzy Commitment

This scheme, which was proposed by Juels and Wattenberg [13] in 1999, still remains one of the most suitable for biometrics that have a template in the form of an ordered string. A k -bit key is mapped to a n -bit codeword of an (n, k, d) ECC. The binary biometric template and the codeword are XOR-ed, thus obfuscating each other. The resulting n -bit string is stored into helper data along with the hashed value of the key. On verification, a new biometric n -bit template is XOR-ed with the stored string. The result is decoded by the ECC and the hashed value is checked.

The most notable applications of the Fuzzy Commitment scheme are Hao et al [15] and Bringer et al [16] works on iris, and priv-ID system for face [17] and fingerprints [18].

IV. SCORE-BASED ATTACKS

One of the strengths of BE is that it outputs either a correct key or a failure message. Unlike conventional biometrics or even Cancelable Biometrics [4], it does not have a verification score by design. This makes BE inherently protected from a high-level Trojan horse attacks, when a malicious program always outputs a high score.

However, in case of BE the attacker may try to derive an *intermediate* score based on the knowledge of the algorithm. The intermediate score may be global or partial, i.e. derived from some parts of the stored information (e.g., if the helper data are structured as a set of smaller chunks).

The published version of Mytec2 uses a repetition $(m, 1)$ code in k chunks. The partial score can be defined as the total number of ones or zeros, whichever is greater, among m bits. A global score is just a sum of k partial scores.

Another example is Hao et al scheme for iris [15] with (64, 7) Reed-Muller (RM) ECC in 32 chunks followed by (32B, 20B) Reed-Solomon (RS) ECC to output a 140 bit key. To derive a partial score in this case, the RM ECC must be run in a soft decoding mode, i.e. when the ECC always outputs the nearest codeword. The partial score is a length of the chunk minus a distance to the nearest codeword, the global score being a sum of partial scores.

The schemes with a correction vector [11] allow deriving an intermediate score based on the distance to the nearest integer. Whether this score is discriminative enough should be a matter of future research.

In the Hill Climbing attack, the attacker makes small changes in the input impostor’s image and watches how the score changes. If it increases, the change is retained; if not, the attacker tries a different change. After a number of iterations, the attacker may be able to retrieve a key. The details of the attack, that uses “quantized hill climber”, are thoroughly described in [5].

A. Nearest Impostors attack

There is an easier way of using the partial score than the Hill Climbing attack. Instead of making changes in the same image, the attacker may run a relatively small, such as a few hundred, number of samples against the helper data. The attacker selects several "nearest impostors", i.e. the attempts with the highest global score. If, for any nearest impostor's attempt, the partial score is large enough for a particular chunk, the attacker may assume that the correct codeword was decoded for this chunk. Other nearest impostors' attempts may decode other chunks. Then the attacker applies a voting technique to the nearest impostors and repeats the process for all the chunks. The attack is considered meaningful if the size of the required attack database is significantly smaller than FAR^{-1} .

The published version of Mytec2 scheme shows a good accuracy for a fingerprint database of 138 unique fingers, 12 - 60 attempts per finger: $FRR \sim 3\% - 10\%$ at $FAR < 10^{-4}$. However, the scheme can be completely cracked by the Nearest Impostors attack. The attacker runs a small database of less than 400 images against each template, and obtains a partial score for each bit out of 255 and the global score.

As shown in Fig. 1, the attacker trains the system by plotting the number of bit errors vs. the global score. There are about 380 impostor attempts; none of them gets even close to the BCH ECC threshold at 18 bits (horizontal line).

However, it can be seen that there is a tendency of obtaining a lower error for a higher score. Based on these training data, the attacker sets a cutoff score at 1.9 (vertical dashed line), which will be used for the attack. There are 5 nearest impostors' samples that passed the cutoff. Then the attacker lists the scores for each bit (out of 255) obtained from these 5 samples and compares them with each other. The decision can be made by a weighed voting technique.

In a typical case, 241 bits out of 255 are found correctly. The following (255, 131) BCH ECC can correct 18 random errors, which means that this score attack succeeds with only 380 impostors' prints. Note that the brute force FAR attack would require about 10000 attempts or more. The Nearest Impostors attack also seems to be more efficient than the Hill Climbing attack which needs about 300 - 400 steps for a weaker face biometrics and a short 20-bit key [5].

Next, we tested two consecutive ECCs of the same style as in [15] but for Mytec2 BE with fingerprints: a (32,6) RM soft decoder followed by (54B, 22B) RS which works with 6-bit bytes output by the RM ECC. The key size is 132 bits.

The attack becomes less effective: the dependence of the number of byte errors vs. the global score tends to be more flat, so that no good nearest impostors' samples are obtained. It is possible to retrieve only 5 out of 54 correct bytes, which is far below the RS threshold of 16 byte errors or 32 erasures. This can be explained by the following property of the (m, l) ECC: when m, l increase, almost all errors for the decoder's failures are concentrated at about $m/2$. The partial score becomes less meaningful for bigger m, l .

To confirm that the system becomes more resilient to the Nearest Impostors attack for larger m, l , yet another ECC

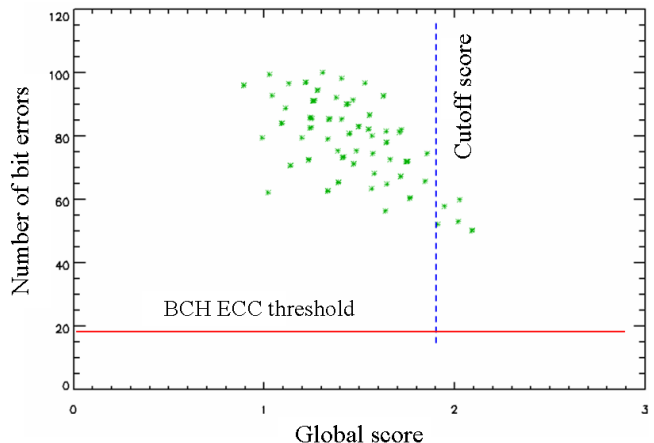


Figure 1. Nearest Impostors attack against Mytec2 BE. Repetition (23, 1) code followed by BCH (255, 131).

combination was tested: (56, 10) ECC (a punctured BCH code) followed by two interleaving (25B, 13B) RS ECCs operating with 5-bit bytes. The first ECC has about the same bit rate, $56/10 = 5.6$, as the (32, 6) RM ECC, $32/6 = 5.33$. The dependence shown in Fig. 1 becomes almost completely flat. The attack is fully unsuccessful, i.e. none of the 25 10-bit chunks has been correctly decoded.

Since the Hill Climbing attack uses the same intermediate score but is even less efficient, it can be expected that any (m, l) single-block ECC with $m \gg 56, l \gg 10$ should be resilient to both Hill Climbing and Nearest Impostors attacks.

Hao et al scheme for iris should be more resilient to the score-based attacks than the schemes that use a simple repetition code. However, based on our results, we think that this scheme is not completely immune, so that some percentage of bytes that are output by RM ECC could be recovered. The same can probably be said about Bringer et al [16] scheme that uses a product of two RM ECCs.

For the parameters of the BCH ECCs used in priv-ID [18], such as $n = 511, k = 76$, most wrong output codewords will have a distance close to $n/2 \sim 255$. The BCH ECC can correct up to 85 random bit errors or 170 erasures. Both numbers are far below $n/2$. In practical terms, this means that the attacker would have to deal with a huge quantization step of about $(255 - 85) = 170$ bits. It is very unlikely that the "quantized hill climber" would be able to handle this.

The scheme using LDPC codes [19], where $n \gg 1800, k \gg 100$, also seems to be very secure in this regard.

V. ATTACKS USING ECC OUTPUT STATISTICS

All practical ECCs were designed for communication and data storage purposes but not for BE. The existing ECCs are not necessarily optimal for BE in terms of bit rates, error correction capabilities, speed, and security. The latter component is usually not taken into consideration in most existing ECCs. Designing an optimal ECC for BE applications is beyond the scope of this paper. This should be a fruitful area of future research, such as, for example, [19].

ECCs consisting of a series of small chunks, such as in [15], seem to be the most vulnerable to the attacks exploiting the output statistics of the ECC.

A. Running ECC in a soft decoding or erasure mode

Many BE schemes use hard decoding algorithm for the ECC, which can correct up to $(d - 1)/2$ random errors, where d is the minimum ECC distance (the code bound). The advantages of hard decoding are high speed, availability, and easiness of implementation.

However, some ECCs can be run in a soft or list decoding mode: the decoder always returns the nearest codeword or a list of the nearest codewords. Soft decoders are usually much slower. If a soft decoder is run for BE, it can correct on average more errors, which results in lower FRR but higher FAR. Therefore, even if the BE algorithm is configured for hard decoding, the attacker can run a soft decoder instead to have a better chance of obtaining a false acceptance (the attacker does not have to follow the official version). Besides, it can help to facilitate the other attacks.

Another option for the attacker is to run the ECC decoder in the erasure mode: if the locations of errors are known (or assumed), the decoder can deal with up to $(d - 1)$ erasures, i.e. twice the number of randomly distributed errors. If the attacker can obtain some bits, e.g., by the Nearest Impostors attack, he will consider the rest as erasures. Even if the number of erasures is more than $(d - 1)$, the attacker can significantly narrow the search space. In general, if the attacker was able to obtain $(n - d + 1)$ bits out of n (or even much less in many cases), the security of the system would diminish from k bits to zero.

1) Davida et al scheme using check bits [20]

The scheme appends $(n - k)$ check bits (stored into the helper data) of (n, k, d) ECC to a binary biometric template. The latter itself, or a value derived from it, serves as a key.

It can be shown that for most practical scenarios the check bits reveal information about the biometric template to the extent that the template can be fully reconstructed.

As an example, consider a $(1023, 193)$ BCH code. It can correct up to $(d - 1)/2 = 118$ random errors, which is only 11.5% of all bits. Most biometrics usually have a much higher error rate. There will be $(1023 - 193) = 830$ check bits stored into the helper data.

To crack the system, the attacker simply prepends arbitrary 193 bits (e.g., all zeros) to the check bits and treats those 193 bits as erasures. The BCH decoder can correct up to $(d - 1) = 236$ erasures, which is more than enough to fully reconstruct the codeword from the helper data. In other words, the scheme would not have any security.

To have at least some security, the parameters of the ECC must satisfy the requirement $n < 2k$. However, such ECC would not be powerful enough for all known biometrics: even for iris, which was targeted by Davida et al, the bit error rate is 25% or higher. In this case, $n \gg k$ is required. Moreover, even if the condition $n < 2k$ is satisfied, the attacker still can significantly reduce the search space by running the ECC list decoder in the erasure mode.

2) Kanade et al scheme with zero insertion [21]

Kanade et al [21] modified the Hao et al scheme by inserting two zeros after every three bits in the iris code. There are 61 blocks of 32 bits in each having from 12 to 14 zeros in known locations. This method boosts the performance of the RM $(32,6)$ ECC from 7 random bit errors in a hard decoding mode, i.e. less than 25%, to $7/20 = 35\%$ of errors (if there are 12 inserted zeros).

Despite good accuracy numbers, the attacker can easily crack the scheme without any additional information [22]: by knowing the locations of only 7 zeros for each 32-bit block, it is possible to reconstruct the entire 198-bit key. For that, the attacker finds the nearest codewords that have the same bits in the known locations (i.e. where the zeros are inserted).

B. ECC Histogram attack

If the ECC consists of a series of chunks, such as Reed-Muller chunks [15], the attacker can run the following attack:

A relatively small database of images is run against the helper data with soft decoding. For each chunk, a number of appearances of each possible output codeword is counted, i.e. a histogram of an output for all impostor attempts is generated. The bin corresponding to the histogram maximum is declared a winner, thus yielding a likely codeword for this chunk. The same is done for all other chunks. The second ECC (Reed-Solomon) wipes out remaining byte errors.

The attack works because the error probability for an impostor distribution is usually below 0.5 (e.g., 0.46 in [15]). The correct codeword has a slightly higher probability of appearance, so that the attacker just needs enough statistics.

The example of the ECC Histogram attack against Mytec2 BE with fingerprints is shown in Fig. 2. There are two consecutive ECCs of the same type as in [15]: the $(64, 7)$ RM soft decoder followed by $(35B, 19B)$ RS working with 7-bit bytes. It can correct up to 8 byte errors or 16 erasures. The key size is 133 bits.

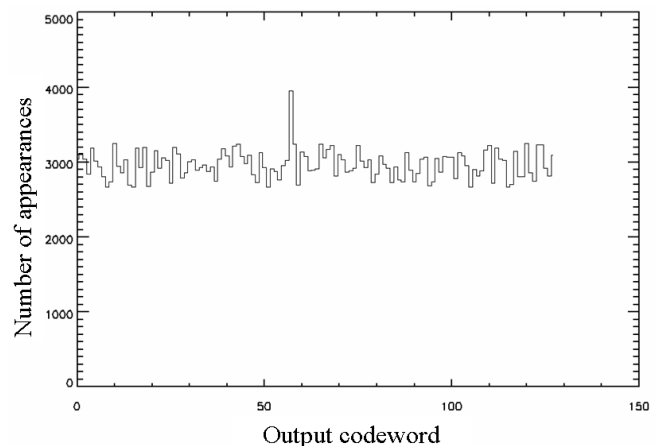


Figure 2. ECC Histogram attack on Mytec2 BE. Reed-Muller $(64, 7)$ ECC followed by Reed-Solomon $(35B, 19B)$ ECC. The winning bin # 57 (the correct codeword) has the highest number of appearances.

The attacker runs a small database of ~ 380 images against the helper data. For each RM chunk, the number of

appearances for each of $2^7=128$ codewords is counted. Since the algorithm checks ~ 1000 shifts, the result for each shift is counted as an independent attempt to increase the size of the statistical sample. A histogram of the output is shown in Fig. 2 for one of the chunks (# 11 in this particular case). Note that there is a peak on the histogram for one of the output codewords, # 57. The attacker assigns the chunk # 11 to the codeword # 57, or 0111001.

Overall, the attacker can correctly obtain 24 out of 35 chunks, which is still below the system threshold of 27 chunks set by the RS code. However, the attacker knows the locations of many of those 24 chunks because they have a higher peak in Fig. 2, so that the RS code can be run with erasures. The attack becomes 100% successful.

The experiments show that the ECC Histogram attack works best for powerful (m, l) ECCs with $m \sim 100, l < 12$.

The attacker needs sufficient statistical samples to be able to populate all 2^l bins of the histogram (that is why only a relatively small l could be handled). However, the size of the impostor database does not have to be very large. The attack works even if the same impostor's image or template is re-used many times by applying various rotations and distortions, both global and local. As for Mytec2 scheme, even the image shift can be counted as a separate entry. For the iris schemes [15, 16], the attacker would likely need a relatively small database (~ 1000) of iris images. The algorithm checks globally 7 rotation angles. The attacker can generate additional local distortions by applying 3 rotations and 3 dilations to each of 8 sectors of the iris image, which yields 504 samples with 7 global rotations. Having 1000 images, the attacker can generate 504,000 samples to populate all 128 histogram bins. Since the system FAR in [15] is set to less than 1 in 240,000, it is very unlikely that such database of ~ 1000 images, even with the distortions, would produce a false acceptance. The ECC Histogram attack is more likely to succeed. This assumption, of course, needs to be tested.

The remedy against the ECC Histogram attack is to increase the l size of the (m, l) ECC, such that 2^l becomes so large that the bin content for all output codewords would be either 0 (in most cases) or 1. At some l , the ECC Histogram attack becomes less feasible than the FAR attack.

BE schemes that use a single block ECC, such as [17-19], are secure against this attack by definition.

VI. NON-RANDOMNESS ATTACK

This group of attacks exploits possible non-randomness of the biometric information and/or the helper data. It may not even need a biometric database.

Many BE schemes, such as Fuzzy Commitment and Fuzzy Vault, require biometric input to be random. Under these conditions, it is possible to deliver a formal proof of the system security. Unfortunately, it is difficult to satisfy this condition in real life, because most biometric traits are inherently non-random. The non-randomness often provides redundancy that reduces the false rejection.

There is a view that even if the biometric is not random but, nevertheless, has sufficient entropy to support a long key, then the system remains secure. As will be shown, this is not necessarily true. In some cases the attacker can completely crack the system. Besides non-randomness of the biometric, the structure of the helper data is also important. One of the known examples is the attack against Fuzzy Vault based on the method for generating chaff points [7].

A. Mytec2

The published version of Mytec2 does not have any notion of the randomness control of the output pattern. The stored filter, $H(u)$, is random. However, the output pattern $c_{\theta}(x)$ is obtained by applying a transitory (i.e. not stored) Fourier magnitude factor to $H(u)$ and performing the inverse Fourier transform. If the structure and the parameters of the magnitude factor are not properly controlled, the output pattern may become non-random. After the most reliable components are selected as top positive and top negative, they can be mapped on a 2D array. The result will consist of clusters, that are, in general, well separated (Fig. 3).

The attacker does not know which cluster is white or black, i.e. corresponds to ones or zeros. However, he may try to figure out the parity of each cluster in the following way:

The algorithm uses a $(m, 1)$ repetition code, i.e. every key bit is encoded by m bits of the same parity. As shown in Fig. 3, the attacker can choose the clusters that are well separated from the others. The attacker maps all the bits, a, from the first codeword. All other bits within those clusters, for example, b, c, d will also have the same parity. The attacker interconnects the white clusters containing the bits a, b, c, d. The other group of clusters (black) is interconnected by the bits e, f, g, h. It takes very little effort to sort all the clusters

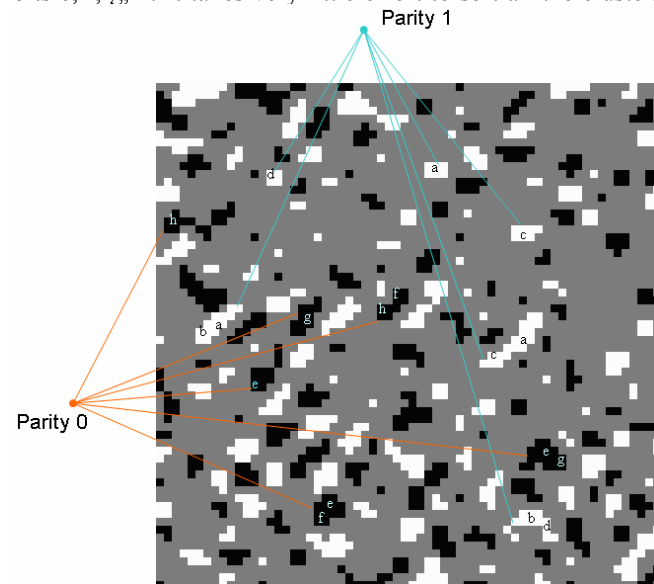


Figure 3. Non-randomness attack against Mytec2 BE with a repetition code and no randomness control.

The white clusters of parity one are interconnected via the bits a, b, c, d. The black clusters of parity zero are interconnected via the bits, e, f, g, h.

into two groups. The attacker can correct possible mistakes by using a voting technique. Overall, the system security is reduced to 1 bit (the attacker still has to distinguish between ones and zeros).

For Mytec2 scheme, it is easy to make the output pattern random by properly choosing the structure and parameters of the transitory filter. For example, if the magnitude part is not used at all, the pure phase-only functions will be completely random. By varying the parameters, it is possible to find a reasonable tradeoff between accuracy and security.

B. Fuzzy Commitment scheme

Unlike Mytec2, the Fuzzy Commitment scheme does not have a randomness control. Therefore, if the biometric template is not random, the problems of the same type as was described for the published version of Mytec2 can occur.

For example, in [15], the standard iris template contains 2048 bits with 249 degrees of freedom, meaning that the redundancy rate can be as high as 8. If the attacker knows (or can guess) how those bits are distributed in the output pattern, he can interconnect the bits in (64, 7) RM ECC. The attack becomes much easier if there is extraction of the most reliable components, as it helps the attacker to identify the clusters. The original scheme [15] uses all 2048 components, which makes the attack more difficult. However, extracting the most reliable components seems to be a natural step in improving the system performance and, therefore, should be considered from a security perspective.

In general, a larger (n, k) ECC block makes the non-randomness attack more difficult. The attacker would have to choose among 2^k codewords rather than between only two codewords (i.e. all zeros and all ones). However, the attack is still possible by using a syndrome decoding which will reduce the problem to solving a set of linear equations.

To make the scheme more secure against the Non-randomness attack, some kind of randomization could be introduced into the system. In Mytec2, this is achieved by multiplying the fingerprint Fourier transform with a random phase-only function. Since a product in the Fourier domain is equivalent to the convolution in the image domain, the continuous input feature vector can be convolved with a randomly generated vector. This random vector should be user- and application-dependent and can be obtained from a seed stored in the system or even from a user's password. The convolved feature vector will be random. After that, the reliable features can be extracted and binarized.

A generalization of this method (called BioHashing) is described in [23]: the feature vector is multiplied with a random matrix, R , of $m \times n$ size, where $m < n$. In other words, the vector is randomized and has the dimension reduced. Again, the matrix R is application-dependent and is generated from a random seed. By choosing $m \sim 0.9 \cdot n$, it is possible to maintain the same level of accuracy [23].

CONCLUSIONS

We presented the results for the Nearest Impostors attack, the attacks exploiting ECC output statistics, and the Non-

randomness attack. As a model, we used modified Mytec2 BE scheme for fingerprints, but most results are also applicable to the Fuzzy Commitment scheme. The attacks can be thwarted or at least mitigated by increasing the length of the ECC block and by applying a randomizing "transform-in-the-middle".

REFERENCES

- [1] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric Template Security". EURASIP J. Adv. Signal Proc., v. 2008, pp. 1-17, 2008.
- [2] P. Tuyls, B. Škorić, and T. Kevenaar, eds., "Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting". Springer-Verlag, London, 2007.
- [3] A. Cavoukian and A. Stoianov, "Biometric Encryption: The New Breed of Untraceable Biometrics." Chapter 26 in Boulgouris, N. V., Plataniotis, K. N., Micheli-Tzanakou, E., eds.: Biometrics: fundamentals, theory, and systems. Wiley, London, 2009, in press.
- [4] N. K. Ratha, J. H. Connell, and R. M. Bolle. "Enhancing security and privacy in biometrics-based authentication systems". IBM Syst. J., v. 40, No. 3, pp. 614-634, 2001.
- [5] A. Adler, "Vulnerabilities in Biometric Encryption Systems". LNCS, Springer, v. 3546, pp. 1100-1109, 2005.
- [6] X. Boyen, "Reusable cryptographic fuzzy extractors." In 11th ACM Conf. CCS 2004, Washington, DC, pp. 82-91, Oct. 2004.
- [7] E.-C. Chang, R. Shen, and F. W. Teo, "Finding the Original Point Set Hidden among Chaff". In Proc. ACM Symp. ASIACCS'06, Taipei, Taiwan. pp. 182-188, 2006.
- [8] W. J. Scheirer and T. E. Boult, "Cracking Fuzzy Vaults And Biometric Encryption". In Biometric Consortium Conference, Baltimore, Sept. 2007.
- [9] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy and B.V.K. Vijaya Kumar, "Biometric Encryption." In ICSA Guide to Cryptography, Ch. 22, McGraw-Hill, 1999.
- [10] A. Kholmatov and B. Yanikoglu, "Realization of correlation attack against fuzzy vault scheme." Proc. of SPIE, Vol. 6819, 2008.
- [11] I.R. Buhan, J.M. Doumen, P.H. Hartel, and R.N.J. Veldhuis, "Fuzzy extractors for continuous distributions". In Proc. 2nd ACM Symp. ASIACCS, Singapore, pp. 353-355, March 2007.
- [12] J. Bringer and H. Chabanne, "An authentication protocol with encrypted biometric data". LNCS, Springer, v. 5023, pp. 109-124, 2008.
- [13] A. Juels and M. Wattenberg, "A fuzzy commitment scheme". In Sixth ACM Conf. Comp. Commun. Secur., pp. 28-36, 1999.
- [14] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data". Proc. Eurocrypt 2004, pp. 523-540, 2004.
- [15] F. Hao, R. Anderson, and J. Daugman, "Combining Crypto with Biometrics Effectively". IEEE Trans. Comp., v. 55, pp. 1081-1088, 2006.
- [16] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji, and G. Zemor, "Optimal iris fuzzy sketches". In IEEE BTAS'07, 2007.
- [17] M. van der Veen, T. Kevenaar, G.-J. Schrijen, T. H. Akkermans, and Fei Zuo, "Face Biometrics with Renewable Templates". Proc. SPIE, v. 6072, 2006.
- [18] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, "Practical Biometric Authentication with Template Protection". LNCS, v. 3546, pp. 436 - 446, Springer, 2005.
- [19] E. Martinian, S. Yekhanin, and J. S. Yedidia, "Secure biometrics via syndromes", in Allerton Conf. Comm., Control and Comp., Monticello, IL, Sep. 2005.
- [20] G.I. Davida, Y. Frankel, and B.J. Matt, "On enabling secure applications through off-line biometric identification." Proc. IEEE 1998 Symp. on Security and Privacy, pp. 148-157, Oakland, Ca., 1998.
- [21] S. Kanade, D. Camara, E. Krichen, D. Petrovska-Delacretaz, and B. Dorizzi, "Three factor scheme for biometric-based cryptographic key regeneration using iris". In BSYM '08, Tampa, FL. Pp. 59-64, 2008.
- [22] A. Stoianov, "Security of Error Correcting Code for Biometric Encryption: a critical note on Kanade et al paper," 2009, unpublished.
- [23] A. B. J. Teoh and C. T. Yuang, "Cancelable Biometrics Realization With Multispace Random Projections". IEEE Trans. Systems, Man, And Cybernetics—Part B: Cybernetics, v. 37, No. 5, pp. 1096 - 1106, 2007.