Methods of reducing bio-cryptographic algorithms computational complexity

PhD student eng. MARIUS-ALEXANDRU VELCIU 1,3, Prof. dr. VICTOR-VALENIU PATRICIU 2
1,2 Military Technical Academy, Computer Science Department, Bucharest, Romania
3Advanced Technologies Institute, Bucharest, Romania
alexandruvelciu@gmail.com, victorpatriciu@yahoo.com

Abstract – Traditional biometric systems have always been regarded with skepticism, due to the uniqueness and universality features of human biometrics – once stolen or compromised, they cannot be safely replaced. For this reason, common efforts of Cryptography and biometric systems researchers gave birth to Bio-cryptography, a domain which proposes to overcome the main security issues associated with biometrics usage, by generating or securing existing cryptographic keys, using biometric data. The main advantages brought by Bio-cryptography are represented by eliminating the need for storing biometric templates and the flexibility of resulted bio-cryptograms. On the other hand, one major drawback concerning bio-cryptographic algorithms is their high consumption of computational resources, caused by the various mathematical processing and large amounts of data used. This paper proposes two methods of reducing bio-cryptographic algorithms computational complexity: the usage of error-correcting codes in the secret key recovery stage and the shared biocryptograms approach for the entire biometric encryption process.

Keywords – bio-cryptographic algorithms, biometric encryption, Fuzzy Vault scheme, Reed-Solomon error-correcting codes, Lagrange Interpolation, shared biocryptograms.

I. INTRODUCTION

Considering the fact that selling or sharing biometric data has become a large scale business model, practiced by many companies, and even governments, it is quite difficult for biometrics to gain more confidence from modern cryptosystems users. Bio-cryptography was formed as a more secure alternative for the usage of biometric data, either for generating and regenerating sensitive data from it, either by binding it with any kind of private information, with the purpose of securing them both.

As biometric data always have a certain degree of variability, even for biometric samples taken from the same user, in the same conditions, the key generation approach is not well suited for use in large-scale applications, as bit-level key regeneration is mandatory. Therefore, key binding represents the main area of interest within Bio-cryptography, which proposes to combine user biometrics with existing cryptographic keys, in the so-called Biometric Encryption process.

The main advantage brought by Biometric Encryption is represented by the elimination of the need for storing biometric templates, which addresses many of the classical biometric systems vulnerabilities. Moreover, the flexibility of resulted bio-cryptograms, which can be revoked or simply cancelled, without any additional security risks, represents another important benefit of this approach.

On the other side, bio-cryptographic algorithms are high consumers of computational resources, due to the mathematical operations they implement and the large data volumes associated with them. As a direct consequence, the response speed of the input / output operations of a bio-cryptographic system can get affected, depending on its available resources.

For this reason, most of the recent studies concerning Bio-cryptography are focused on developing novel methods of reducing its algorithms computational complexity, especially for the biometric decryption process, the most time and data consuming one.

This paper proposes two such methods, one of them focused on simplifying the biometric decryption process, by using the Reed-Solomon error-correcting codes during the polynomial recovery stage, and the other one aiming to reduce the data overhead of the entire biometric encryption process, by using shared bio-cryptograms for all the users enrolled in the system.

The rest of this paper is organized as follows: Section 2 reviews some of the related work in domain; Section 3 describes the background information of this article, regarding the biometric encryption / decryption processes, the Reed-Solomon error-correcting codes and the Fuzzy Vault scheme, the most well known and widely used bio-cryptographic algorithm; Section 4 details our proposed improvement methods for reducing bio-cryptographic algorithms computational complexity, while Section 5 describes the experimental results obtained. Finally, Section 6 concludes this paper.

II. RELATED WORK

Numerous research works have been written previously, concerning the combination of biometrics and cryptography, as it can be seen in [1],[2],[3] or [4]. Most of them rely of the Fuzzy Vault scheme, as a reliable method of binding biometric traits and credentials.
However, not many of them discussed methods for reducing bio-cryptographic algorithms computational complexity.

In [5], Feng Hao et al. proposed the usage of a combination of Hadamard and Reed-Solomon codes, but for trying to correct the errors encountered during the cryptographic key generation using biometrics.

In [6], Sim Hiew Moi et al. used the Reed-Solomon error-correcting codes for reducing the variability and noise of the iris data, contained in the biometric template used within a classical biometric authentication approach.

Melanie Favre et al. emerged the idea of using Fuzzy Vault for multiple users, in [7], as a suggestion of using shared bio-cryptograms, but they haven’t detailed any ways of implementing it, nor any experimental results of this approach.

III. BACKGROUND INFORMATION

A. Biometric encryption

Bio-cryptography proposes, as an alternative for the traditional password-based security mechanisms, to directly correlate the sensitive data of the users (passwords, PINs, cryptographic keys) with their biometric data. There are three main operating modes introduced by Bio-cryptography, which can achieve this [8]:

- **Key generation**: user secret values are generated/regenerated based exclusively on their biometric data;
- **Key recovery**: the secret value is recovered only after the successfully completion of a biometric authentication process;
- **Key binding**: the private key and the digitized biometric sample of the user (the so-called biometric template) are combined in a bio-cryptographic construction, such as neither of them can be illegitimate recovered from the derived bio-cryptogram. Also known as the Biometric Encryption process.

Although the key generation approach confers the highest level of security, because of the biometric data entropy, it is hardly feasible to regenerate the same bi-cryptographic key, from different biometric samples, even when they are taken from the same person, in almost exactly the same conditions. Analogously, the second operation mode, the key recovery one, is essentially a traditional biometric system authentication mechanism, used to secure and recover an existing biometric key. Thereby, it doesn’t confer any additional advantages.

For these reasons, the key binding approach is the most representative operation mode of the Bio-cryptographic systems, as Biometric Encryption is the most feasible and reliable mechanism from those described above.

As described in Fig. 1, in essence, the secret key is encrypted using the biometric [8]. The resulted bio-cryptogram confers a high level of security, as both the key and the biometric query used to secure it are discarded at the end of the process.

![Figure 1: Biometric encryption / decryption processes](image)

The decryption algorithm either recovers the initial secret key, if the fresh biometric sample is similar enough to the one used in the enrollment phase, either fails and rejects the recovery attempt. What customizes the Biometric Encryption process is the fact that its algorithms are designed to tolerate a certain variance of the biometric samples, variance that can be controlled using a certain threshold [9]. This way, the impediment encountered in the key generation approach can be overcome, as biometric encryption process can work with samples which are not identical.

B. Reed-Solomon error-correcting codes

Reed-Solomon error-correcting codes represent a category of cyclic error-correcting codes, built over finite fields. Such a code is specified by RS(n, k) with s bits symbols, meaning that its encoder input can consist of k symbols of s bits each, which get extended to n-length codewords, by adding parity bits. Hence, a Reed-Solomon decoder can correct up to (n − k) / 2 symbols that contain errors [10].

There are several encoding procedures for this kind of codes, resulting in different ways of defining the codewords. One that stirs interest for the Biometric Encryption process is the one defined in the original approach of Reed and Solomon, back in 1960, in which each codeword represents a sequence of values that reside on the polynomial curve of a low-degree polynomial. This way, to derive a Reed-Solomon codeword, we need to evaluate a polynomial P in n distinct points of the finite field F, the resulting sequence of values representing the codeword itself:

\[ C = \{ p(a_1), p(a_2), \ldots, p(a_n) \} \]

Since any two polynomials of degree k have at most k - l common points, it means that any two Reed-Solomon codewords differ in at least \( n - (k - 1) = n - k + 1 \) positions. Moreover, since there are maximum two polynomials of degree k, that have exactly k - l common points, the Hamming distance for Reed-Solomon error-correcting codes is exactly \( d = n - k + 1 \).

For the decoding stage of the Reed-Solomon error-correcting codes, one of the most suitable methods is represented by the Berlekamp-Massey algorithm, which
represents a reliable mechanism of determining the minimum degree polynomial of a linear recurrent sequence, defined over an arbitrary finite field \[10\].

C. The Fuzzy Vault scheme

The Fuzzy Vault scheme represents a bio-cryptographic algorithm that proposes to secure a secret value by encoding it under a polynomial construction, followed by its binding with the biometric data of the legitimate owner of it. The encryption algorithm used within this scheme is specially designed to work with unordered data sets \[2\]. Its security resides on the infeasibility of the polynomial reconstruction by using brute-force or statistical attacks, when dealing with a large order mixture of both points lying on the polynomial, and points that don’t.

During the first stage of the biometric encryption algorithm, the secret value is used to derive the polynomial construction \(P\). Thus, for a finite field \(F\), such as the Galois field \(GF(2^q)\), a secret value \(k \in F\) is encoded using a subset:

\[ A = \{a_i\}_{i=1}^{q} \text{ where } a_i \in F \]

Basically, each group of \(q\) bits, that the initial secret value is split into, is mapped to a coefficient of the encoding polynomial \(P\), of degree \(k – 1\). Once the polynomial is derived, the next step involves the evaluation of the values (points) contained within the biometric template of the legitimate user (e.g.: can be fingerprint minutiae points value, voice cepstral coefficients etc.). At this stage, we obtain the genuine points set, which, subsequently, needs to be secured within the Fuzzy Vault construction:

\[ G = \{(x_i, P(x_i))\}_{i=1}^{n} \]

where \(x_i\) represents each value from the biometric template, and \(n\) represents their order of magnitude.

The genuine points are secured using another set of points, called chaff points, randomly generated, which mustn’t reside on the encoding polynomial curve:

\[ C = \{(c_i, d_i)\}_{i=1}^{m} \text{ with } d_i \neq P(c_i) \]

where \(m\) represents the number of chaff points used. Their role is to diffuse the genuine points within the final derived bio-cryptogram, as the Fuzzy Vault construction is obtained by the scrambled reunion of the two above point sets. The Fuzzy Vault encryption process is summarized in Fig. 2:

In order to recover the initial secret value, from within the Fuzzy Vault construction, one needs to input a fresh biometric sample, or to use the biometric template that was used during the enrollment phase (if it was stored by him on a mobile device or a smart card). If the new biometric sample has a certain degree of similarity, over a certain preset threshold, then the derived biometric template will recover a number of points from the bio-cryptogram, by comparison with the abscissa points value, obtaining the so-called candidate points set:

\[ Q = \{(b_i, x_i)\}_{i=1}^{q} \]

If the order of the candidate points set is higher than \(k + 1\), where \(k\) represents the encoding polynomial degree, then we can recover at least one candidate polynomial, by using Lagrange Interpolation (in mathematics, any polynomial of degree \(k\) can be recovered by knowing minimum \(k + 1\) points that reside on its polynomial curve).

IV. OUR PROPOSED IMPROVEMENT METHODS

A. Reed-Solomon error-correcting codes approach

As stated in the final section of the previous chapter, the polynomial reconstruction stage of the biometric decryption process can produce multiple candidates, by identifying more than \(k+1\) points within the bio-cryptogram, where \(k\) represents the degree of the encoding polynomial, which needs to be recovered, in order to decrypt the initial secret value.

In this case, the genuine polynomial identification sums up to an exhaustive search between the \(C(n, k+1)\) candidate polynomials, where \(n\) represents the number of points “spotted” from the Fuzzy Vault bio-cryptogram. This process needs to be used in conjunction with a checksum mechanism, which adds an appendix during the enrollment phase, in order to be checked against each
candidate polynomial in the decryption stage. As Lagrange interpolation needs to be applied to each of the candidate points subset, in order to recover the current polynomial, this stage is one of the most resource and time consuming in the entire Fuzzy Vault scheme, especially when the candidate points identification stage has revealed a large number of such points.

Error-correcting codes have the potential to simplify the polynomial reconstruction stage, as they can identify the genuine polynomial without conducting an exhaustive search. Adapted to this case, the error-correction process maps to a polynomial-correction one, as the single-run of the decoding algorithm of the code will output the genuine polynomial, of course, if some conditions are met.

The main idea that allows the integration and usage of error-correcting codes within Bio-cryptography is to consider the genuine points set as a codeword, of length \( t \), which encodes a secret message of length \( k + 1 \), where \( k \) has the same meaning as above, the degree of the encoding polynomial.

By extrapolating this principle to the Reed-Solomon error-correction codes, used in conjunction with the Berlekamp-Massey algorithm, during the decoding stage, a code with \( n \) elements, having \( n - k - 1 \) redundancy elements, can correct up to \( (n - k - 1) / 2 \) errors. Therefore, in order to be able to recover the genuine encoding polynomial, during the polynomial reconstruction stage, by using this approach, it is sufficient to identify at least \( (n + k + 1) / 2 \) genuine points from within the vault [12], where \( n \) represents the number of candidate points, and \( k \) is the degree of the polynomial.

\[
\begin{array}{|c|c|}
\hline
\text{Exhaustive polynomial reconstruction} & \text{Error-correcting codes approach} \\
\hline
\text{Biocryptogram} & \text{Biocryptogram} \\
\hline
\text{Candidate points identification} & \text{Candidate points identification} \\
\hline
\text{Candidate points combinations} & \text{Error correction (Berlekamp-Massey decoding)} \\
\hline
\text{Lagrange Interpolation} & \text{Lagrange Interpolation} \\
\hline
\text{Chebyshev verification (CRC OK?)} & \text{Recovered secret} \\
\hline
\text{YES} & \text{YES} \\
\hline
\text{Recovered secret} & \text{Recovered secret} \\
\hline
\end{array}
\]

**Figure 4: Using error-correcting codes for the polynomial reconstruction stage**

In conclusion, the benefit brought by the Reed-Solomon error-correcting codes can be a significant one, especially if we’re dealing with many candidate points, among which there are predominant chaff points. Of course, it is essential for the condition regarding the implementation of the Berlekamp-Massey algorithm to be satisfied, that amongst the candidate points there are at least \( (n + k + 1) / 2 \) genuine ones, otherwise the error-correcting code usage is infeasible.

B. Using shared bio-cryptograms approach

Another highly consumer of resources process, encountered during the enrollment stage, is represented by the random chaff points generation. These points are essential to any Vault-based bio-cryptographic algorithm, as they are meant to diffuse the genuine information among them, making a brute-force attack infeasible.

Most of the papers in domain, such as in [2] and [13], recommend that the order of magnitude for the diffusion points should be at least ten times higher than the genuine points one. The imposed condition leads inevitably to a data overhead of at least 90%, for each enrolled user bio-cryptogram, thus, for the entire system.

Moreover, besides the high data payload needed to generate and store these points, there are some additional mathematical processing that need to be done, in order to generate the chaff points correctly. Thereby, each pair of such points needs to be checked against the genuine polynomial curve, such as it doesn’t belong to it. If this condition isn’t met, chaff points can get mixed up with genuine ones and might generate false acceptance points, facilitating illegitimate access. Considering their order of magnitude, the fact that the encoding polynomial’s degree value is around 8 or 9, usually, and all the calculus is yield in a finite Galois field, the random chaff points generation process is by far the most time and resource consuming one, during the biometric encryption stage.

Our proposed improvement method, addressed theoretically in our previous work [14], is to use shared bio-cryptograms, in order to eliminate the need for random chaff points. Basically, the genuine points belonging to all the users enrolled in the system get mixed up in a single bio-cryptogram. This way, the entire data overhead of the encoding scheme can get reduced to zero, as the Vault construction will contain only genuine points, as illustrated in Fig. 5, below:

\[
\begin{array}{c}
\text{Figure 5: Shared bio-cryptograms}
\end{array}
\]
To sum up, the benefit brought by using shared bio-cryptograms could be an important one, as the data overhead of the entire biometric encryption scheme can get significantly reduced and the computational speed of the enrollment process can increase.

The following chapter describes the experimental results obtained by implementing such an approach, by comparison with an original Fuzzy Vault scheme, which uses random chaff points.

V. EXPERIMENTAL RESULTS

Further research following our theoretical survey on Shared Vault constructions [14] led us to conducting some experimental results, in order to evaluate the real benefit conferred by this approach.

We've deployed two scenarios, both representing implementations of a voice-based Fuzzy Vault biometric encryption algorithm. The first one consisted of a traditional distributed Vault encoding scheme, while the second one used the shared bio-cryptograms approach. The voice-based algorithm used cepstral analysis as a speech parameterization method, for deriving the digitized biometric template, represented under the form of the calculated cepstral coefficients.

Both approaches performances were analyzed using a database containing 600 fixed phrase audio samples, from 300 different users (two samples for each user). The goal of the biometric encryption scheme was to secure a 128-bits AES key. Each 16-bits group of the key was mapped into a coefficient of the encoding polynomial. For the polynomial reconstruction stage, we used a 16-bits checksum, as in [11]. This way, the final initial secret payload consisted of:

\[ PC = P \mid CRC (128+16 = 144 \text{ bits}) \]

The encoding polynomial coefficients were obtained from the 144-bits payload, its degree being k=8:

\[ P(u) = c_8u^8 + c_7u^7 + \ldots + c_1u + c_0 \]

The evaluation of the cepstral coefficients values, taken as abscissa points values on the polynomial curve, derived the genuine points of the final Vault.

Moreover, during first experiments concerning the shared bio-cryptograms approach, we ran into the necessity of imposing a minimum threshold value for the distance between any two points within the Vault, as we encountered some duplicate genuine points, which could generate false acceptance. This approach had the potential to decrease the False Acceptance Rate (FAR) of the system, but at some major costs concerning the increase of the False Rejection Rate (FRR) of the system.

Various FAR and FRR rates were obtained for different values of this threshold and different speech parameterization, summarized in the table below, together with data usage and computational speed comparison.

<table>
<thead>
<tr>
<th>Nr. of genuine points</th>
<th>Nr. of diffusion points</th>
<th>Data overhead</th>
<th>Genuine points threshold</th>
<th>Min. distance between points</th>
<th>Average total run time</th>
<th>FAR</th>
<th>FRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2048 / user</td>
<td>20480 / user</td>
<td>90%</td>
<td>0.005</td>
<td>0</td>
<td>27 s</td>
<td>15.2%</td>
<td>4%</td>
</tr>
<tr>
<td>204800</td>
<td>0</td>
<td>0%</td>
<td>0.005</td>
<td>0</td>
<td>11 s</td>
<td>16.3%</td>
<td>4%</td>
</tr>
<tr>
<td>204800</td>
<td>0</td>
<td>0%</td>
<td>0.005</td>
<td>0</td>
<td>18 s</td>
<td>13.8%</td>
<td>10%</td>
</tr>
<tr>
<td>4096 / user</td>
<td>40960 / user</td>
<td>90%</td>
<td>0.003</td>
<td>0</td>
<td>39 s</td>
<td>10.6%</td>
<td>8%</td>
</tr>
<tr>
<td>409600</td>
<td>0</td>
<td>0%</td>
<td>0.003</td>
<td>0</td>
<td>15 s</td>
<td>11.8%</td>
<td>8%</td>
</tr>
<tr>
<td>409600</td>
<td>0</td>
<td>0%</td>
<td>0.003</td>
<td>0.003</td>
<td>26 s</td>
<td>8.5%</td>
<td>12%</td>
</tr>
<tr>
<td>8192 / user</td>
<td>81920 / user</td>
<td>90%</td>
<td>0.002</td>
<td>0</td>
<td>1m 25s</td>
<td>8.1%</td>
<td>14%</td>
</tr>
<tr>
<td>819200</td>
<td>0</td>
<td>0%</td>
<td>0.002</td>
<td>0.002</td>
<td>35 s</td>
<td>9%</td>
<td>14%</td>
</tr>
<tr>
<td>819200</td>
<td>0</td>
<td>0%</td>
<td>0.002</td>
<td>0.002</td>
<td>56 s</td>
<td>6.5%</td>
<td>25%</td>
</tr>
</tbody>
</table>

The most important aspect pointed out by the experimental results is the fact that eliminating the diffusion points has speed up the algorithm considerably, getting it almost three times faster, in average.

As we can see, the simple reunion of all the genuine points, within a single shared vault, has slightly increased the FAR. On the other hand, forcing all the points within the shared vault to lie at a minimum distance between them (distance set to exactly the maximum threshold value for genuine points) has slightly improved the FAR, but with the cost of increasing the FRR at a higher rate (since many of the genuine points were eliminated from the bio-cryptogram) and loosing almost half of the computational speed gain.

VI. CONCLUSIONS

Biometrics confer a high degree of portability and ease-of-use, but also exhibit a significant level of risk, associated with unauthorized access to users biometric data. As this information is unique per individual, its theft or loss is equivalent with its compromise.

Biometric encryption has an important potential of improving traditional biometric systems security, as its main benefits include eliminating the need for biometric templates storage and the flexibility of generated bio-cryptograms, resulting in better public acceptance.

On the other hand, the biometric encryption gain comes with some drawbacks, regarding its algorithms high computational complexity, due to various mathematical processing and high data overhead. As a direct consequence, the response speed of bio-cryptographic algorithms it’s lower than for traditional biometric systems.

This paper proposes two methods for reducing bio-cryptographic algorithms computational complexity. First of them, addressed, for now, only theoretically, proposes the use of error-correcting codes during the polynomial reconstruction stage of the biometric decryption process. This method has an important potential of simplifying this process, as it can eliminate the exhaustive search of the genuine encoding polynomial.

The second method, addressed both theoretically and experimentally, proposes the elimination of the random
chaff points generation stage, for Fuzzy Vault based bio-cryptographic algorithms, by replacing it with a shared bio-cryptograms approach. This way, genuine points of each user enrolled in the system get diffused among other users genuine points. Experimental results have revealed that this approach has got the biometric encryption algorithm run almost three times faster, in average.

Moreover, imposing a constraint regarding the minimum distance between any two points within the Shared Vault led to a slight improvement of the FAR, but with the cost of increasing the FRR at a much higher rate and loosing almost half of the previous speed gain. Still, for a high security level system, this scenario might be more suitable, despite the raised number of failed legitimate login attempts and the lower system response time.

ACKNOWLEDGMENT

This paper has been financially supported within the project entitled “Horizon 2020 – Doctoral and Postdoctoral Studies: Promoting the National Interest through Excellence, Competitiveness and Responsibility in the Field of Romanian Fundamental and Applied Scientific Research”, contract number POSDRU/159/1.5/S/140106. This project is co-financed by European Social Fund through Sectoral Operational Programme for Human Resources Development 2007-2013. Investing in people!

REFERENCES


