ISSN 2007-9737

A New Fuzzy Vault based Biometric System Robust to Brute-Force Attack

Alfonso Francisco De Abiega-L'Eglisse[1], Mario Rosas Otero[2], Vladimir Azpeitia Hernández[3], Gina Gallegos-Garcia[4], Mariko Nakano-Miyatake[1]

[1] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Culhuacán, Mexico

[2] Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán, Mexico

[3] Instituto Politécnico Nacional, Escuela Superior de Cómputo, Mexico

[4] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico

{ggallegosg, mnakano}@ipn.mx, adeabiegal1900@alumno.ipn.mx, malttz15@comunidad.unam.mx, vazpeitiah@gmail.com

Abstract. Fuzzy vault based biometric systems use fuzzy vaults during the coding that occurs within the enrollment stage inside a biometric system. In that stage the biometric system creates a vault from the biometric data, user's key and chaff points. In the verification stage, a new biometric data is introduced and user's key can be recovered through the use of the Lagrange polynomial interpolation method. As a consequence, in this kind of systems brute force attack would be successful because fuzzy vaults are finite. Most of the related works focus on designing secure fuzzy vault systems through the use of a password or using hybrid systems. This leads to security falling on the same user or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples. This paper proposes a new system that considers cryptography to achieve a fuzzy vault biometric system robust against brute-force attacks. It is important to say that our system does not need a higher number of chaff points or even a higher polynomial degree. Obtained results show that this new fuzzy vault biometric system not only would be secure for current times but also would be for the future time.

Keywords. Authentication, fuzzy vault, biometric system, confidentiality, cryptography, encryption.

1 Introduction

Fuzzy vault based biometric systems use an algorithm for hiding a secret string S in such way that a user who has the biometric template T can easily recover S. The biometric template T can be fuzzy in the sense that the secret S is locked by some related, but not identical data T' [12]. However, this kind of systems are susceptible to various attacks such as attack against the vault with minutiae descriptors, false-accept attack, intermediate discussion, cross-matching or brute-force attack [6, 20]. Roughly speaking, a brute-force attack consists of an attacker submit

many passwords or passphrases with the hope of eventually guessing the one. The attacker systematically checks all possible passwords and passphrases until the correct one is found.

Related work shows that brute-force attack has attempted to be mitigated through the use of symmetric-key cryptographic schemes that base their security on the difficulty of solving mathematical problems with higher complexity such as the integer factorization problem, the discrete logarithm problem, or the elliptic-curve discrete logarithm problem [8, 17, 20].

Considering the aforementioned, this paper proposes a new fuzzy vault based biometric system that considers three cryptographic primitives in order to mitigate the brute-force in the following manner. The first primitive is a hash function. It is used to obtain a hash value from the original vault. The second primitive is a key encapsulation mechanism. This is used to agree a cryptographic secret key.

Then, in the enrollment stage, the binding data composed by the secret key and user's biometric data are encrypted with a symmetric key encryption. This last one as the third primitive we consider. Subsequently, in the verification stage the biometric template is compared with the new user sample, all this without the need to decrypt them.

Finally, to recover the cryptographic key, Lagrange polynomial interpolation method is executed. As a consequence, our biometric system is able to keep the minutiae values in a confidential way, even thought an attacker steals the templates values, without needing a higher number of chaff points or even a higher polynomial degree. The rest of the paper is organized as follows.

Section 2 shows the related work that describes the way to harden the fuzzy vault biometric system. Section 3 describes the brute-force attack, the fuzzy vault biometric system, and what is a brute-force attack over a fuzzy vault biometric system. Section 4 explicates the cryptographic considerations and describes the work inside this paper as well as the notation and the algorithm.

Section 5 reports all the data needed to made the experiment and shows all the steps in the experiment of brute-force attack over a fuzzy vault biometric system. Section 6 shows the results of the experiment of brute-force attack between no encrypted vault and encrypted vault and Section 7 shows the conclusion of this work.

2 Related Work

This section analyzes the papers related to biometric systems and their improvements in terms of security. These improvements within biometric systems are to prevent attacks such as brute force attack specifically in biometric systems based on fuzzy vaults.

In [10], the idea of the fuzzy vault for the retinal biometric template is presented through a multi-modal biometric fuzzy vault. It includes points from the retina and fingerprint in order to obtain a combined vault, which is hardened with user password for achieving high-level security. The security of the combined vault is measured using min-entropy.

The proposed password hardened multi biometric fuzzy vault is robust towards stored biometric template attacks. In [11], a brute force attack which improves upon the one described in [2] in an implementation of the vault for fingerprints is presented. On base of this attack, they show that the implementations of the fingerprint vault are vulnerable and cannot be avoided by mere parameter selection in the actual frame of the procedure.

They introduce the idea of the fuzzy vault based on information resources not used by the current version of the vault. In [13], a scheme for hardening a fingerprint minutiae-based fuzzy vault using password is proposed. Benefits of the proposed password-based hardening technique include template revocability, enhanced vault security and a reduction in the False Accept Rate of the system without significantly affecting the False Reject Rate.

Since the hardening scheme utilizes password only as an additional authentication factor (independent of the key used in the vault), the security provided by the fuzzy vault framework is not affected even when the password is compromised.

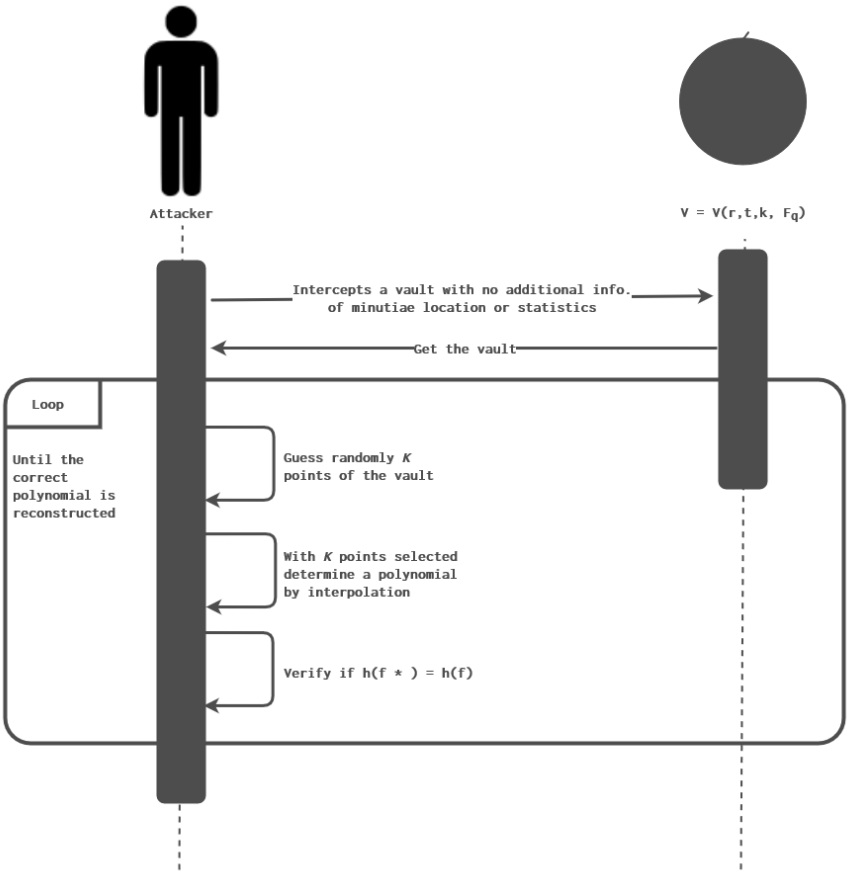

Fig. 1. Brute force attack process

In [16], a review of techniques like hybrid model and multimodal biometrics for security are proposed. They show how such techniques can be effective in enhancing the security of the system. In [18], some of the other known attacks against biometric fuzzy vault and biometric encryption techniques is reviewed.

They introduce three disturbing classes of attacks against Privacy Enhanced Technologies (PET) techniques including attack via record multiplicity, surreptitious key-inversion attack, and novel blended substitution attacks. In [22], a minutiae-based fuzzy vault implementation preventing an adversary from running attacks via record multiplicity is redesigned. Furthermore, they propose a mechanism for robust absolute fingerprint prealignment. Together, they obtain a fingerprint-based fuzzy vault that resists known record multiplicity attacks and that does not leak information about the protected fingerprints from auxiliary alignment data.

In [5], the vulnerabilities of the scheme in [13] is analyzed. After studying various schemes using special data like password a new scheme which is secure against various attacks to fuzzy vaults is proposed to enforce the security.

In [9], a new attack based on the alteration of original user data on fuzzy vault biometric cryptosystem is investigated. They assume that the attacker uses a modified version of the real user image to gain unauthorized access to the system (mobile phone).

ISSN 2007-9737

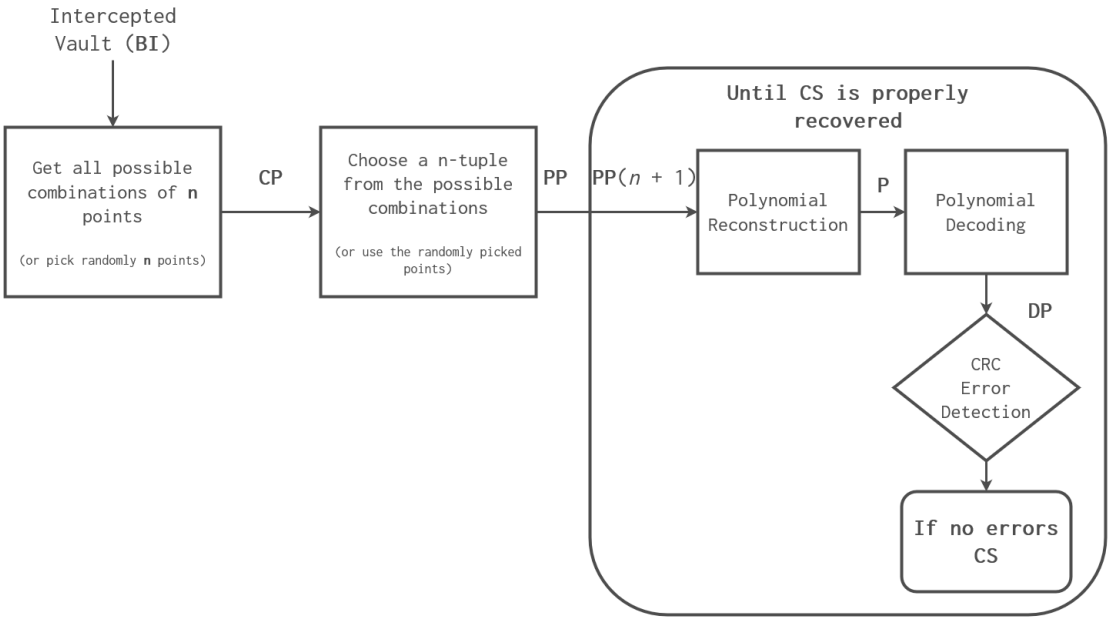

Fig. 2. Modular diagram of the brute force attack process

Experimental results carried out using fingerprint and face modalities show that this assumption has serious impact on the security of this type of biometric cryptosystem.

As we can see most of the related work focuses on designing secure fuzzy vault systems through the use of a password, using hybrid systems, or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples.

In this work, we propose a new system that considers cryptography to achieve a fuzzy vault biometric system robust against brute-force attacks without needing a higher number of chaff points or even a higher polynomial degree.

3 Brute-Force Attack on Fuzzy Vault based Biometric Systems

The attack that we address in this work is brute-force attack. In this sense, on the one hand, we give the definition of the attack. On the other hand, we describe how a fuzzy vault biometric system works.

Finally, we describe the attack over such kind of system. The brute force attack is shown in the literature as viable to violate the fuzzy vault scheme based on fingerprints. Having the advantage that explicit knowledge of the operation of the scheme or of the implementation in the system to be attacked is not necessarily required.

A disadvantage compared to other attacks is that it has a high computational cost. One of the terms that must be taken into account is that to facilitate the verification of success at the time of implementation, it is assumed that the result of the coefficients of the polynomial or the secret is known. Fig. 1 shows the high level brute force attack process diagram, establishing the lines of each entities involved in the attack and their participation in its development.

Fig. 2 shows the modular diagram of the brute force attack where it identifies the transformation of the data and presents the comparison of the data identifying whether it is correct or not. A fuzzy vault based biometric system works over a field $\mathbb{F}$ of cardinality q and a universe $\mathbb{U}$. It assumes in the exposition that $\mathbb{U} = \mathbb{F}$.

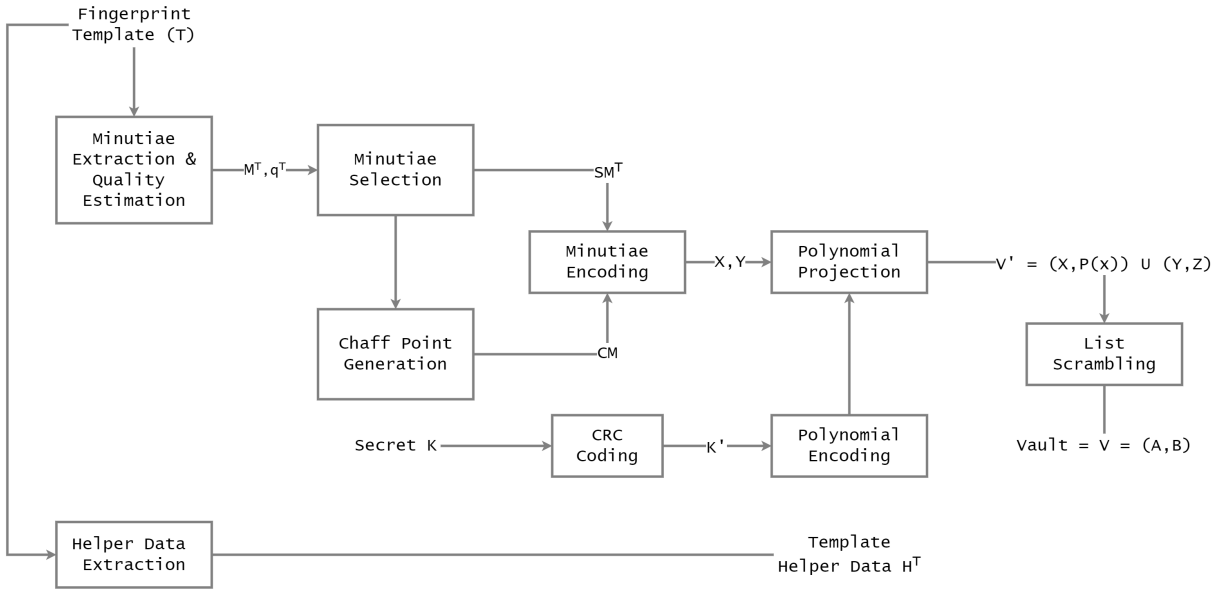

Fig. 3. Fuzzy vault scheme

The target in this kind of systems is to lock a secret value $\mathcal{K} \in \mathbb{F}^k$ under a secret set $\mathbb{A} \in \mathbb{U}^t = \mathbb{F}^t$, for protocol parameter $\mathcal{K}$ and t. Fig 3 shows the original fuzzy vault scheme. It considers a fuzzy vault encryption algorithm that takes, as input, a secret $\mathcal{K}$, a set $\mathcal{A}$ and outputs a vault $\mathcal{V}_\mathcal{A} \in \mathcal{F}^r$ for some security parameter r.

A corresponding decryption algorithm takes as input a vault $\mathcal{V}_\mathcal{A} \in \mathcal{F}^r$ and a decryption set $\mathcal{B} \in \mathcal{U}^t$. The output of the algorithm is a plain text value $\mathcal{K}' \in \mathcal{F}^k$ or null if the algorithm is unable to extract a plain text [12]. The brute-force attack process essentially consists of obtaining a fuzzy vault of size T that is of interest to be breached.

Then all the possible combinations of points existing in the vault are obtained in groups of n elements, where $0 \leq n \leq T$ and T is the total of points that the vault contains. It is also possible to choose n points at random.

If a range of values $[V_1, V_2]$ is known that could be the degree of the polynomial then the combinations are made in groups of $V_1 \leq n \leq V_2$ points of the vault. If the degree V of the polynomial is known then the combinations of points will be made in groups of $n = V$ points.

Subsequently, each of the combinations of points are passed through the Lagrange polynomial interpolation method, the result is obtained and it is verified if the secret obtained is equal to the original secret. If they are the same, the vault has been breached.

4 A New Fuzzy Vault based Biometric System

4.1 Cryptographic Considerations

Cryptographic Hash Functions. take a message as input and produce an output referred to as a hash code, hash-result, hash-value or simply hash. A hash function h maps bit-strings of arbitrary finite length to strings of fixed length, e.g. n bits. For a domain D and range R with $h : D \rightarrow R$ and $\mid D \mid > \mid R \mid$. Hash functions are one-way function, in other words, they are practically infeasible to invert [4].

Key Encapsulation Mechanism. are a class of encryption method designed to protect symmetric cryptographic key material from transmission using a public key scheme. In other words, KEM is a set of functions that can be used to obtain a symmetric encryption key from asymmetric keys.

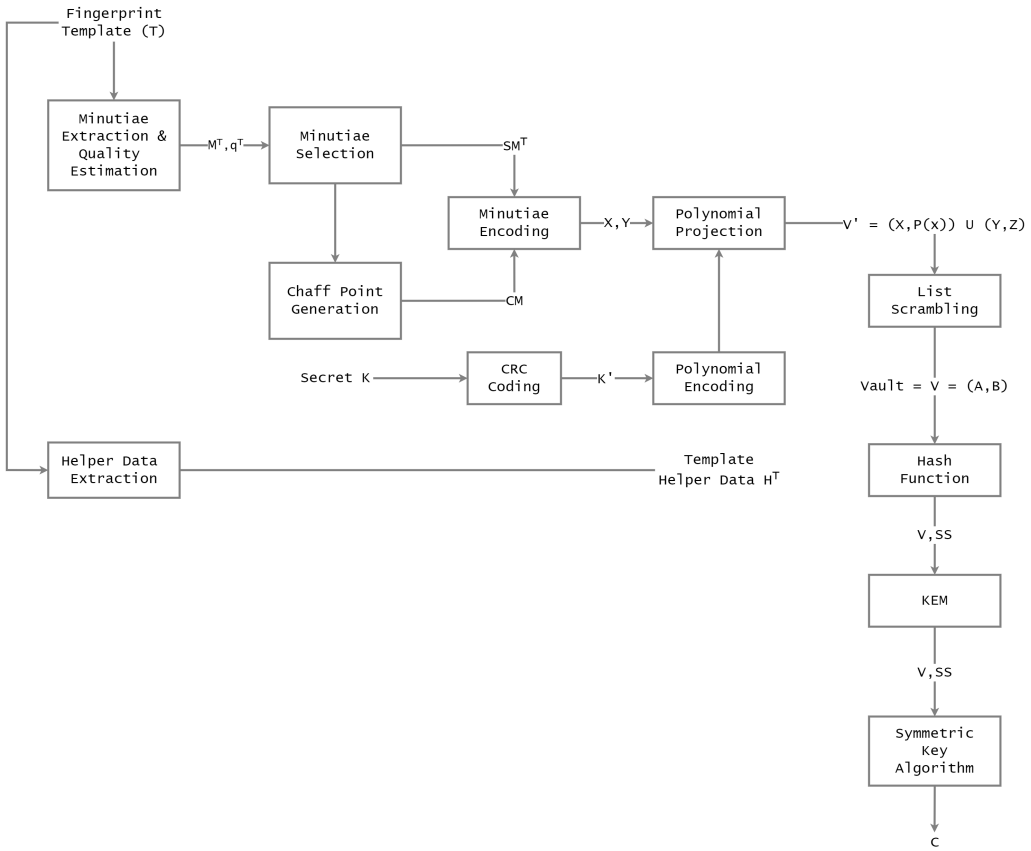

Fig. 4. Cryptographic considerations in our fuzzy vault based biometric system

Then, the symmetric key is used to encrypt the longer message. KEM simplifies the process by generating a random element in the finite group underlying the asymmetric scheme. Then, the symmetric key is derived by hashing it as a consequence the need for padding is eliminated.

KEM is composed of three algorithms named KeyGen, Encaps and Decaps. KeyGen generates a public key pk and a private key sk. Encaps returns a symmetric key K and a ciphertext ct. Decaps returns K [14].

Symmetric Key Algorithm. These are used in the most modern block ciphers. They often incorporate a sequence of permutation and substitution operations.

An iterated cipher is a commonly used design. It requires the specification of a round function, a key schedule and the encryption of a plain text will proceed trough Nr similar rounds [3].

4.2 Description

Our system considers encrypting template values with a symmetric key algorithm. The symmetric key is generated with a hash function of variable length. It is transmitted to the database with a key encapsulation mechanism. The template is protected by encrypting the data, so if it is stolen, a brute force attack can be avoided, since the template is not in plain text.

The entities that interact in our proposed solution during the enrollment are the biometric scanner that takes the fingerprint template, a personal computer used to capture user data and a server that storage all the user data captured. The entities

that interact during the verification are the biometric scanner that takes a new fingerprint and a server to verify the data.

Considering the approach presented in [19] and the entities aforementioned, as we can see in Fig. 4, we take as the input to our first algorithm, a data matrix denoted by $Vault = V = (A, B)$. After that its hash value is obtained with $SS = H(V)$, whose size is the exact length of the input of the key encapsulation mechanism. Once this hash value SS is obtained, it is necessary to transmit it to server. It is made by using the key encapsulation mechanism between the server and the personal computer as follows.

Firstly, the server generates a key pair $\mathsf{KeyGen}(\mathsf{Sk}, \mathsf{Pk})$ to start the encapsulation process. Then, the server sends the public key to the personal computer. Now, it can encapsulate the value SS called secret shared. Secondly, the personal computer encapsulates the secret shared SS by using the public key and sends it to the server. Finally, the server receives the information and recovers the secret shared SS with his private key and the decapsulation algorithm.

When the secret shared SS is obtained, the server stores it for its later usage. After that, the symmetric key algorithm is used with the secret shared, recovered by both entities, to encrypt the Vault. When it is done, the Vault is transmitted, from the personal computer to the server, to be storage in the database.

All of this is depicted in Fig. 5 and can be seen in Table 1. Considering that the decoding stage maintains in the same way with [19], with the difference that the $Vault$ is processed in the filter process.

It must be decrypted with the symmetric key algorithm $V = D_{SS'}(C)$ used in the coding stage and using the secret shared, previously saved, as the key to decrypt such $Vault$. All of this is depicted in Fig. 6 and Fig. 7.

4.3 Cryptographic Protocol in our New Fuzzy Vault Based Biometric System

In this section, we describe the cryptographic protocol defined for the encrypted fuzzy vault biometric system.

The notation of variables used in the protocol is described in Table 2.

Table 1. Cryptographic protocol definition

Cryptographic protocol definition for our new fuzzy vault based biometric system
1 : $M^T, q^T, H^T \leftarrow Ext(T)$
2 : $SM^T \leftarrow (M^T, q^T)$
3 : $CM \leftarrow Ch.P.Gen(u, v, \theta)$
4 : $(X, Y) \leftarrow M.Encod(CM, SM^T)$
4a: $P \leftarrow Polyencod(K')$
4b: $K' \leftarrow CodingCRC(K)$
5 : $V' \leftarrow Polyproyec(X, Y, P)$
6 : $V \leftarrow L.S(V')$
7 : $h \leftarrow H(V)$
8 : $Sk \leftarrow KEM(h)$
$C^V \leftarrow Enc(Sk, V)$

Table 2. Notation

Symbol	Definition
T	Fingerprint template
M^T	Minutiae extracted
q^T	Quality estimation
SM^T	Minutiae selection
CM	Chaff points
(X, Y)	Minutiae encoding
K'	CRC Coding
P	Polynomial encoding
$V'(X, P(x) \cup (Y, Z))$	Polynomial projection
$V = (A, B)$	Vault
E/D	Encryption/Decryption symmetric key algorithm

ISSN 2007-9737

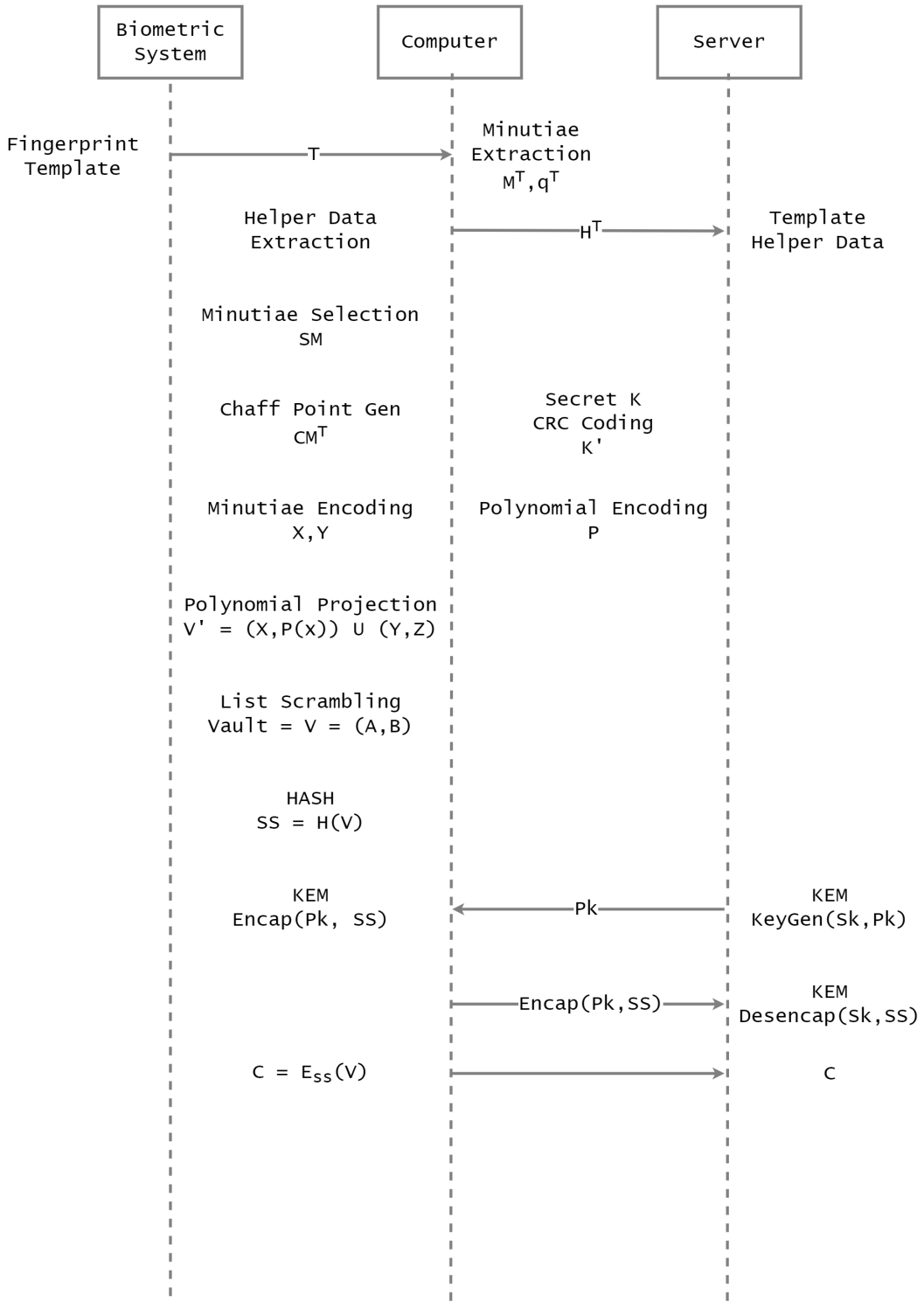

Fig. 5. Interaction between the three entities of our solution proposed

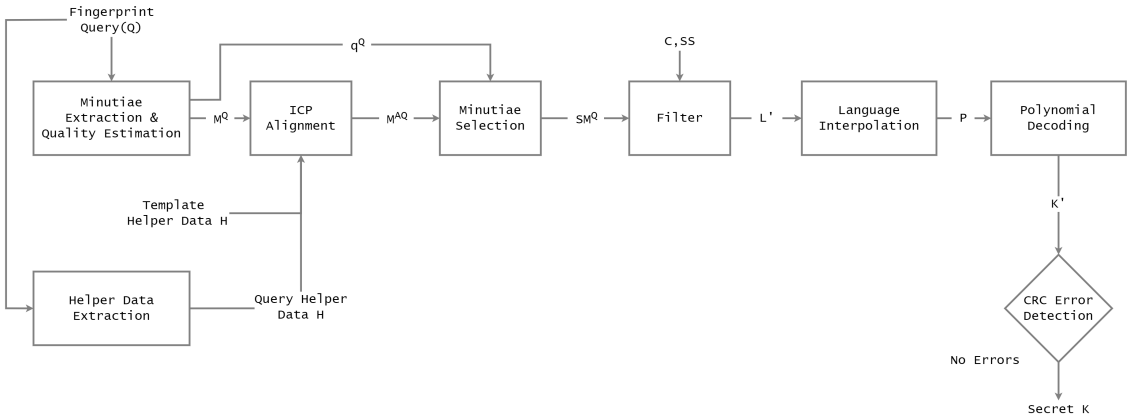

Fig. 6. Verification of fuzzy vault based biometric system considers outputs obtained from cryptographic considerations

5 Experimentation

5.1 Fingerprint Database Overview

For the experimentation, a database of 80 images of fingerprints in total was used. All the images contain different fingerprint impressions, such impressions belong to 10 different users. That is, there are eight different impressions for each user.

With each of the impressions, a different fuzzy vault is obtained. Three types of vaults were created for each fingerprint impression resulting in a total of three sets of 80 fuzzy vaults. Each of the vault types created are described below.

Free Size Vaults. These vaults are based on the quality of the minutiae, a quality filter is used at the time of extraction to select the best samples, resulting in the total set of minutiae being those with sufficient quality for use at the time of an authentication request later. This in turn results in each template having a fuzzy vault with a number of different genuine points and the overall size of the vaults would be randomly sized differently. All these vaults possess 50 chaff points and S genuine points, with S being the amount of high-quality minutiae found.

Standard Size Vaults. These vaults are created by sacrificing a little the quality of the minutiae in order to obtain a certain number of R in all the impressions thus achieving that all the vaults produced have the same size regardless of whether they come from different fingerprint impressions. All these vaults contain 50 chaff points and 23 genuine points.

Encrypted Vault. These vaults are created from standard sized vaults that have the same R elements. All These vaults contain 50 chaff points and 23 genuine points.

5.2 Brute-Force Attack Experimentation

Having the fuzzy vaults created from the available database it is possible to test them for vulnerability relatively easily. For the implementation of the attack, a uniform random distribution was used to apply the Lagrange polynomial interpolation method. It is assumed that the degree of the hidden polynomial is previously known in the experimentation.

The results provided in the Table 3 and Table 4 show the performance against iterations and the time required to successfully break the vaults. The criterion considered that a vault cannot be violated is when the correct polynomial is not found after a million iterations of polynomial reconstruction.

The vaults are named in the following way. Firstly the name of the vault, then the user number and finally the sample number of the user. As in example Vault101_1, Vault102_1,$\cdots$, Vault110_8.

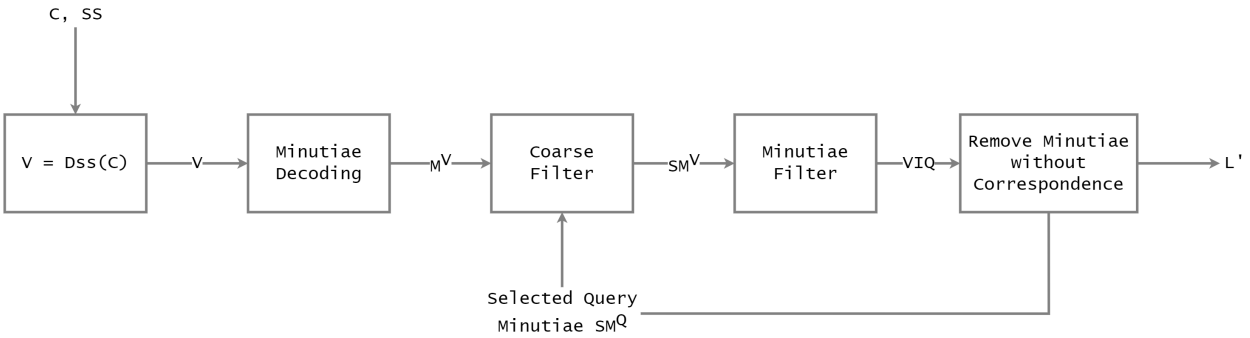

Fig. 7. The process of filtering during the verification stage should obtain the original vault

5.3 Cryptographic Schemes Used in the Experimentation

The cryptographic hash function used into the experimentation is named SHAKE-128 with an output size of 32 bits. SHAKE or SHA-3 extendable-output function (XOFs) is defined from the KECCAK[c] function by appending a four-bit suffix to M, for any output length d : SHAKE$128(M, d)$ = KECCAK$[256](M\|1111, d)$.

KECCAK is the family of sponge functions with the KECCAK $-$ p$[b, 12 + 2l]$ permutation as the underlying function and with $pad10^*1$ as the padding rule. The family is parameterized by any choices of the rate r and the capacity c such that $r + c$ is in $25, 50, 100, 200, 400, 800, 1600$ [4].

The key encapsulation mechanism used into the experiment to calculate the symmetric key is Kyber-1024. CRYSTALS - Kyber is an IND CCA2 secure key encapsulation mechanism (KEM), whose security is based on the hardness of solving the learning-with-errors (LWE) problem over module lattices. Kyber is one of the finalists in the NIST post-quantum cryptography project.

The submission lists three different parameter sets aiming at different security levels. Specifically, Kyber-512 aims at security roughly equivalent to AES-128, Kyber-768 aims at security roughly equivalent to AES-192, and Kyber-1024 aims at security roughly equivalent to AES-256 [1].

The symmetric key algorithm used into the experiments to encrypt the vault is AES-256-CBC. The Advanced Encryption Standard commonly called AES has a specification for the encryption of digital data established by the United States National Institute of Standards and Technology (NIST) [15]. It has a fixed block size of 128 bits and a key length of 128, 192 or 256 bits.

The relation between the number of rounds and the key length is as follows. There are 10 rounds for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys.

According to the specification, AES works on a 4x4 array of bytes, named the state and most of the operations are done in the finite field [3, 7]:

$$\mathbb{F}_{2^8} = \mathbb{Z}_2[x]/(x^8 + x^4 + x^3 + x + 1). \qquad (1)$$

6 Results

As we mentioned before, our new system considers Cryptographic Hash Functions, a Key Encapsulation Mechanism, and Symmetric Key Algorithm in order to be robust against brute-force attacks. Before including cryptographic schemes in our simulation, the attacker executes a brute-force attack on the vault in order to find the polynomial coefficients that contain the secret that was defined during the enrollment process of a user.

The probability of finding the correct combination that forms this polynomial is $1/n$ with n equal to the number of combinations. By encrypting the vault with the symmetric encryption scheme resistant to attacks from computers, robustness is added to the biometric system, since the security of this type of scheme lies on two parameters. The first one is the length of the symmetric key used to transform or encrypt the vaults of the biometric system.

Table 3. Iterations and time results per user to breach the free-size vaults

User	# Vaults	Min. Iterations	Max. Iterations	Min. Time	Max. Time
1	3	7,613	637,964	7.8s	573s
2	3	54,486	462,515	65.4s	545.4s
3	4	517	655,593	0.6s	718.8s
4	7	1,120	79,873	1.2s	85.2s
5	5	15,759	881,182	16.2s	909.6s
6	8	1,309	214,215	1.2s	238.2s
7	6	17,671	979,271	19.8s	1555.8s
8	8	901	134,253	5.4s	149.4s
9	3	89,734	427,978	100.2s	463.2s
10	5	4,029	484,881	7.2s	635.4s

The second one is the ciphertext obtained from the vaults. In other words, the patterns that the attacker can use to guess the polynomial coefficients are absent, as stated in [12]. The existence of symmetric encryption leads to the generation and establishment of symmetric key between two entities, the computer that captures the user's data and the server that stores it.

During this interaction there exist a possibility that the attacker wants to intercept the symmetric key. Our proposed solution incorporates a key encapsulation mechanism (KEM), in charge of agreeing the symmetric key between the computer and server.

KEM scheme uses a hash function, which takes as input parameters one of the vault coefficients of the biometric system, selected randomly. The function output feeds the KEM scheme to agree the symmetric key they use to encrypt the vault coefficients.

6.1 Results in Proposed Solution Against Brute-Force Attack with no Encrypted Vaults

6.1.1 Free Size Vault Experimentation

Table 3 shows the results of the number of iterations that were necessary to violate the fuzzy vaults. In the experimentation, not all the vaults were violated and the table shows the number that could have been violated. For all users, there are cases in which the number of iterations to violate the vaults was low, however, there are also cases in which the number of iterations was very high.

In the end, there were only two users to whom all the vaults were successfully breached. The same Table 3 shows the results measured in seconds needed to break the vaults are shown. The results of the vaults that were successfully breached averaged at least 25 seconds and averaged a maximum of 734 seconds.

In some vaults the polynomials could not be calculated by means of Lagrange polynomial interpolation method because they did not contain enough genuine points to calculate the polynomial reconstruction.

The reason is because the minutiae quality filter did not allow for enough genuine points. Vaults with fewer genuine points than necessary were Vault101_4, Vault101_5, Vault102_5, Vault102_8, Vault107_8, Vault109_6.

Some vaults that needed more than a million iterations in the polynomial reconstruction to be able to find the coefficients of the correct polynomial are Vault101_2, Vault101_3, Vault101_8, Vault102_3, Vault102_4, Vault102_7, Vault103_2, Vault103_3, Vault103_4, Vault103_8, Vault104_5, Vault105_4, Vault105_6, Vault105_8, Vault107_3, Vault109_1, Vault109_3, Vault109_4, Vault109_8, Vault110_1, Vault110_2, Vault110_4.

Table 4. Iterations and time results per user to breach the standard-size vaults

User	# Vaults	Min. Iterations	Max. Iterations	Min. Time	Max. Time
1	8	488	208,412	1.2s	346.2s
2	8	8,128	83,129	16.8s	159s
3	8	9,403	617,940	9s	885.6s
4	8	8,123	211,829	7.8s	288s
5	8	80,589	218,181	123s	294.6s
6	8	710	60,441	1.2s	71.4s
7	8	26,040	482,439	30s	596.4s
8	8	25,476	253,338	24.6s	277.8s
9	8	214	406,164	0.18s	632.7s
10	8	1,558	242,562	1.8s	211.8s

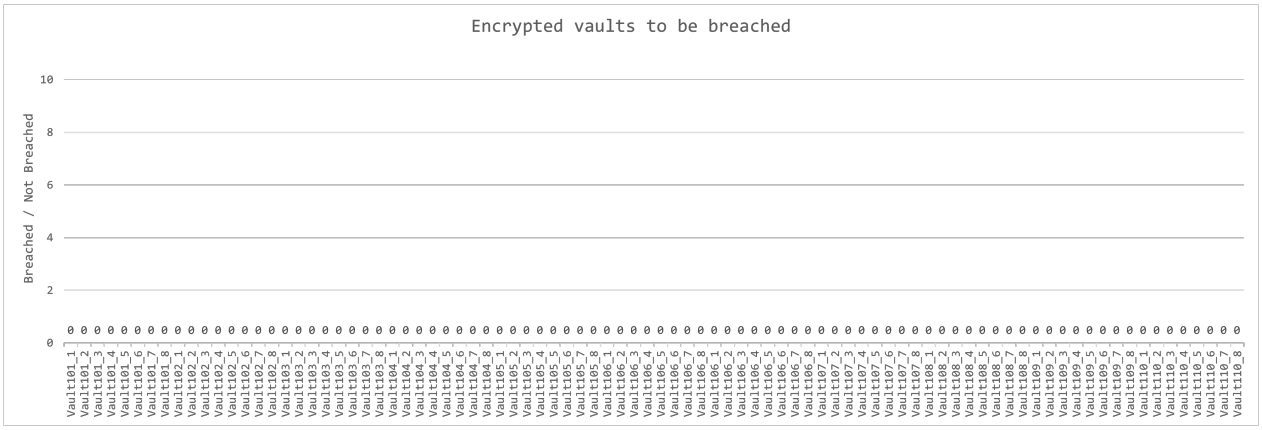

Fig. 8. Encrypted vaults breached

6.1.2 Standard Size Experimentation

Table 4 shows the results of the number of iterations that were necessary to breach the vaults. During the development of the experiment, 100% of the vaults used could be successfully breached and 65% of the vaults could be breached with less than 100,000 iterations.

The same Table 4 shows the results measured in seconds of the time required to breach the vaults.

During the experiment, most of the vaults were breached in less than 300 seconds, that is, the attack shows low times to be a brute force attack considering that all the vaults were breached.

6.2 Results in Proposed Solution Against Brute-Force Attack with Encrypted Vaults

Since the viability and performance of the brute force attack on the fuzzy vault scheme has been verified in this work, a test was carried out with the same vaults but that were passed through an encryption function, specifically using the symmetric key algorithm.

Encrypted vaults, instead of being a set of points, are a string of coded characters. Because of that, it is not feasible to apply a polynomial reconstruction directly on that character string. The alternative to show a comparison is to transform the encoded character string to a set of points that are contained in the finite field bounded by the vault values.

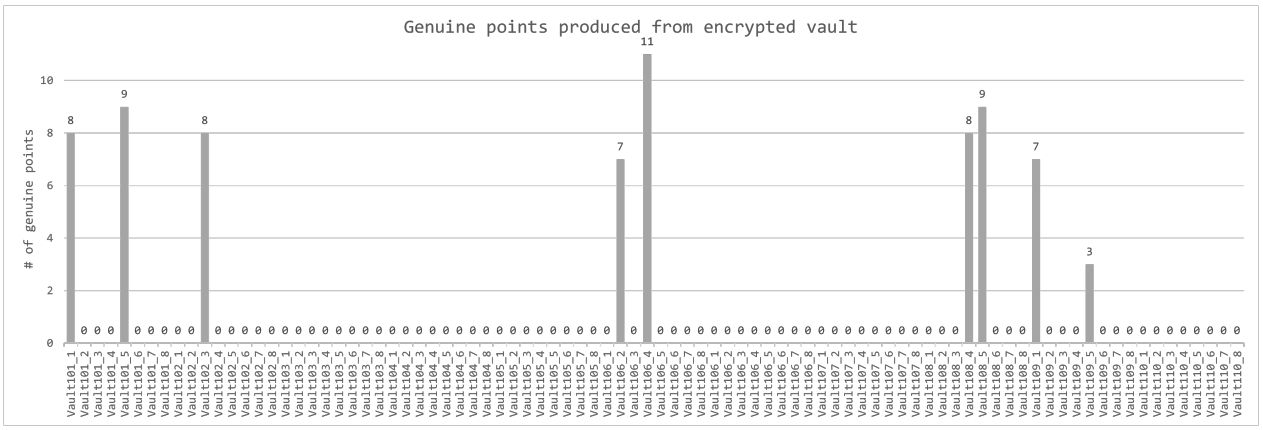

Fig. 9. Genuine point results recovered on all vaults from their encrypted versions

To achieve this, the following process was done. The encrypted vault is read as a binary file. The values obtained are transformed to their equivalent in the decimal system at values between 0 and 255. Later, these values are normalized and re-scaled to values between 0 and 2^{16}. Finally, it is verified that with the transformation made to the set of points, sufficient points have been obtained, this is done by comparing the points obtained in the encrypted vaults with the genuine points of the unencrypted vaults.

The number of points that must be obtained are at least $(k + 1) \times 2$, where k is the degree of the hidden polynomial inside the vault, since both the dependent and independent values that intervene in the polynomial are needed to be able to rebuild it.

In case of finding the number of points necessary to carry out the polynomial reconstruction, the correct order of the data pairs must still be known to be able to interpolate successfully, which increases complexity and computational cost.

In addition, it was necessary to take into account that when transforming the encrypted vault to numerical values, the amount of values obtained depends on the length of the chain that represents the encrypted vault, which for our experiments more than 2000 values are obtained.

This increases the complexity of finding the correct set of coefficients out of such a large number of possible values. So the attacker must try all the combinations of points and prove the polynomial reconstruction since he did not know the genuine points. Fig. 9 shows all the vaults where genuine points could be recovered.

These points were recovered from the encrypted vaults. However, none of them obtained a sufficient number of genuine points for the polynomial reconstruction, since having a polynomial of degree 8 requires at least 18 genuine points recovered to attempt the reconstruction. Given the results shown in the previous graph and due to no vault's enough genuine points were found to even attempt a polynomial reconstruction, that is $(k+1)\times 2$ points, there is no encrypted vault at risk of being compromised as shown in Fig. 8.

Even considering that in a real attack, the attacker would not have a quick way to verify that he is recovering genuine points.

7 Conclusion and Future Work

Fuzzy vault based biometric systems use fuzzy vaults within the enrollment stage inside a biometric system. These systems are susceptible to multiple security attacks. In this paper, we propose a new system that considers cryptography to achieve a fuzzy vault based biometric system robust against brute-force attacks. Our solution was designed based on the effects generated by this kind of attack due to fuzzy vaults are finite.

Most of the related works focus on designing secure fuzzy vault systems through the use of

a password or using hybrid systems. The main difference between our new fuzzy vault biometric system and related work is that we do not need a higher number of chaff points or even a higher polynomial degree. This leads to security falling on the same user or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples.

Obtained results show that an important piece of information is that to recover many more polynomials, it is necessary to have a standard vault size since when there was a free-sized vault, the recovery of the polynomial was lower.

Moreover, we were able to verify that when the vault was encrypted the brute force attack was not successful in recovering the polynomial and therefore the security of this system could not be violated. The test in this paper was made with 256 security bits, as a consequence, this new fuzzy vault biometric system not only would be secure for current times but also would be for the future.

In future work, it is necessary to test other types of attacks such as a correlation attack or an attack through the multiplicity of records to continue testing the security of the proposed solution. In this way, it could be demonstrated that this solution proposal can be effective for protection against multiple attacks.

Acknowledgments

The authors thank the Instituto Politecnico Nacional and the Consejo Nacional de Ciencia y Tecnologia. The research for this paper was financially supported by SIP-IPN 20221427 and CONACYT 321068.

References

1. **Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., Stehlé, D. (2021).** CRYSTALS-Kyber algorithm specifications and supporting documentation. NIST PQCRYPT round 3 submission, pp. 43.
2. **Clancy, T. C., Kiyavash, N., Lin, D. J. (2003).** Secure smartcard-based fingerprint authentication. WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, pp. 42–52.
3. **FIPS-197 (2001).** Advanced encription standard (AES). Last updated October 05, 2021.
4. **FIPS-202 (2015).** SHA-3 standard: Permutation-based hash and extendable-output functions. Last updated November 11, 2020.
5. **Hong, S., Jeon, W., Kim, S., Won, D., Park, C. (2008).** The vulnerabilities analysis of fuzzy vault using password. FGCN '08: Proc of the 2008 Second Int Conf on Future Generation Communication and Networking, pp. 76–83.
6. **Jain, R., Kant, C. (2015).** Attacks on biometric systems: An overview. International Journal of Advances in Scientific Research, Vol. 1, pp. 283.
7. **Katz, J., Lindell, Y. (2020).** Introduction to Modern Cryptography. CRC Press.
8. **Kholmatov, A., Yanikoglu, B. (2008).** Realization of correlation attack against the fuzzy vault scheme. Proceedings of SPIE, Vol. 6819, pp. 7.
9. **Lafkih, M., Lacharme, P., Rosenberger, C., Mikram, M., Ghouzali, S., Haziti, M., Aboutajdine, D. (2015).** Vulnerabilities of fuzzy vault schemes using biometric data with traces. 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 822–827.
10. **Meenakshi, V., Ganapathi, P. (2009).** Security analysis of password hardened multimodal biometric fuzzy vault. World Academy of Science, Engineering and Technology, Vol. 32, pp. 312–320.
11. **Mihailescu, P., Munk, A., Tams, B. (2009).** The fuzzy vault for fingerprints is vulnerable to brute force attack. BIOSIG 2009 - Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, pp. 43–54.
12. **Nandakumar, K., Jain, A., Pankanti, S. (2008).** Fingerprint-based fuzzy vault: Implementation and performance. Information Forensics and Security, IEEE Transactions on, Vol. 2, pp. 744–757.
13. **Nandakumar, K., Nagar, A., Jain, A. (2007).** Hardening fingerprint fuzzy vault using password. ICB 2007: Advances in Biometrics, pp. 927–937.

14. NIST (2009). SP 800-56b - Recommendation for pair-wise key establishment schemes using integer factorization cryptography. Last updated March, 2019.

15. NIST (2017). Post-quantum cryptography standardization. Last updated December 02, 2021.

16. Panwar, A., Singla, P., Kaur, M. (2018). Techniques for enhancing the security of fuzzy vault: A review. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 205–213.

17. Rathgeb, C., Wagner, J., Tams, B., Busch, C. (2015). Preventing the cross-matching attack in bloom filter-based. 3rd International Workshop on Biometrics and Forensics, IWBF 2015.

18. Scheirer, W., Boult, T. (2007). Cracking fuzzy vaults and biometric encryption. Biomet Symp, pp. 1–6.

19. Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. Proceedings of 35th Annual Symposium on Foundations of Computer Science, pp. 124–134.

20. Tams, B. (2013). Attacks and countermeasures in fingerprint based biometric cryptosystems. PhD. Thesis, pp. 32.

21. Tams, B. (2013). Cryptanalysis of the Fuzzy Vault for Fingerprints: Vulnerabilities and Countermeasures. Ph.D. thesis, Georg-August-Universität Göttingen, 37073 Göttingen.

22. Tams, B., Mihailescu, P., Munk, A. (2015). Security considerations in minutiae-based fuzzy vaults. IEEE Transactions on Information Forensics and Security, Vol. 10, No. 5, pp. 985–998.

23. Uludag, U., Pankanti, S., Jain, A. (2005). Fuzzy vault for fingerprints. volume 3546, pp. 310–319.

Article received on 03/02/2022; accepted on 25/05/2022.
Corresponding author is Gina Gallegos-Garcia.