A. Brömme, N. Damer, M. Gomez-Barrero, K. Raja, C. Rathgeb,
A. Sequeira, M. Todisco, and A. Uhl (Eds.): BIOSIG 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022

Implementation and performance of a hardened biometric system based on fuzzy vault scheme

Abstract: Biometric systems based on fuzzy vaults schemes ensure users' biometric information using chaff points, unlike other biometric systems. However, these systems are susceptible to different attacks for example brute-force attacks and cross-matching attacks. As consequence, the biometric information or the secret key can be stolen. Some security hardening solutions are the use of two or more biometric features such as fingerprints and iris or on the other side iris and facial recognition. The solution to be implemented and evaluated in this work has biometric-cryptographic characteristics that can secure the user's biometric information. In this sense, we present the implementation of the biometric system based on a fuzzy vault presented in [De22] and we analyze their performance in comparison with a fuzzy vault system. Obtained results show that is possible secure the vault without increasing the processing time and the vault size.

Keywords: Biometric cryptosystem, confidentiality, encrypt, hardening, fuzzy vault, performance.

1 Introduction

Biometric systems based on fuzzy vault 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' [NP08]. 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 [JK15, Ta13].

Since some years ago, the security issues of biometric systems based on fuzzy vault schemes have been addressed by combining cryptography and biometrics. Consequently, hardened biometric systems based on fuzzy vault schemes result in systems with biometric-cryptographic characteristics.

Related work shows the implementation and performance of hardened biometric systems based on fuzzy vault schemes by focusing on different parameters of the biometric side, for example, performance with the percentage of identification instances in which unauthorized persons are incorrectly accepted, the percentage of identification instances in which authorized persons are wrongly rejected, and the percentage of identification instances in which authorized persons are correctly accepted, they all denoted by False Accepted Rate (FAR), False Rejection Rate (FRR), and Genuine Acceptance Rate (GAR), respectively.

Implementation of such hardened solutions has been done considering only the use of passwords leaving aside both the implementation of quantum-safe algorithms from the cryptography side and their performance with biometrics.

The rest of the paper is organized as follows. Section 2 shows the related work that describes the implementation and hardened biometric systems. Section 3 describes how to hardened a biometric system based on fuzzy vault scheme. Section 4 shows the parameters and the software used for the implementation of the biometric system. Section 6 shows the results obtained between a hardened system and one that is not and Section 7 shows the conclusion of this work.

2 Related Work

In [NP08] a fully automatic implementation of the fuzzy vault scheme based on fingerprint minutiae is presented. High curvature points derived from the fingerprint orientation field are extracted and used as helper data to align the template and query minutiae. The helper data itself does not leak any information about the minutiae template, yet it contains sufficient information to accurately align the template and query fingerprints. Further, a minutiae matcher during decoding to account for non-linear distortion is applied and this leads to significant improvement in the actual acceptance rate. The performance of the vault implementation on two different fingerprint databases is demonstrated. Performance improvement can be achieved by using multiple fingerprint impressions during enrollment and verification.

In [Me10] an implementation of a fuzzy vault based on minutiae information in several fingerprints aiming at a security level comparable to current cryptographic applications is presented. They analyze the performances and evaluate the security, efficiency, and robustness of the construction and several optimizations. The results allow an assessment of the capacity of the scheme and an appropriate selection of parameters. Finally, a practical simulation conducted with ten users is reported.

In [Ra21] the application of a well-known biometric cryptosystem like the improved fuzzy vault scheme is investigated. The proposed feature transformation method and template protection scheme are agnostic of the biometric characteristic and, thus, can be applied to virtually any biometric features computed by a deep neural network. An effective face-based fuzzy vault scheme providing privacy protection of facial reference data as well as a digital key derivation from the face is presented in this work. This paper describes the experimental setup for applying the proposed fuzzy vault to deep face representations. Subsequently, the performance of the different quantisation and binarisation methods are evaluated in the different experiments.

In [RB08] a scheme that hardens both fuzzy vault and secret key using a password is proposed because the fuzzy vault is affected by cross-matching. By using passwords an additional layer of security is embedded to achieve high-level security. Within the implementation, there are three stages: transformation, encoding, and decoding. The parameters used in the work are the following. The number of genuine points is 18 to 20 points. The polynomial can be of degrees 6 to 8. The number of existing points inside the vault is between 120 and 220 points. Finally, the number of chaff points is between 100 and 200 points. The performance of the approach is measured in terms of FRR/GAR and FAR with respect to the degree of the polynomial.

Considering the related work is essential to mention that our work focuses on the implementation and performance of a hardened fuzzy vault scheme. The sizes of hardened vaults and unhardened vaults are compared, measuring the difference in bytes. The processing times of the hardened fuzzy vault scheme and the fuzzy vault scheme are compared, where the difference in clock cycles is measured. This article also focuses on the implementation and performance of the quantum-safe cryptographic algorithms rather than improving biometric sampling or enforcing the scheme using passwords as security.

3 Biometric system based fuzzy vault scheme

3.1 Fuzzy vault scheme

In [Ju06], fuzzy vault scheme works over a field $\mathscr{F}$ of cardinality q and a universe $\mathscr{U}$. It assumes in the exposition that $\mathscr{U} = \mathscr{F}$. The aim of this is to lock a secret value $K \in \mathscr{F}^k$ under a secret set $A \in \mathscr{U}^t = \mathscr{F}^t$ for protocol parameters k and t.

The fuzzy vault scheme is integrated by two algorithms that are the Lock algorithm and the Unlock algorithm. The basic idea of the Lock algorithm is to create a code word representing a secret K as a polynomial p. To protect the code word, chaff points are added as noise in the form of random pairs (x_i, y_i). Since the secret K is assumed to be a polynomial p, it is simply written as $p \leftarrow K$ to represent the conversion.

The basic idea of the Unlock algorithm is to unlock a V vault, in order to determine the code word that encodes the secret K. The set A specifies the x-coordinates of correct points, which lie on the polynomial p. It is written $K' \leftarrow p$ to denote the conversion of a polynomial degree at most k to a secret in $\mathscr{F}^k$. It is denoted as the reverse of the procedure employed in the Lock algorithm.

3.2 Hardened fuzzy vault scheme

In [De22] the hardened fuzzy vault scheme is based on three cryptographic primitives which are a hash function, a key encapsulation mechanism, and a symmetric key algorithm. Tab 1 explains in a broad way how the hardened of the fuzzy vault scheme is assumed out. Notation is maintained as in [De22].

Roughly speaking, a hash function H maps bit-strings of arbitrary finite length to strings of fixed length. Hash functions are one-way function, in other words, they are practically infeasible to invert [FI15]. Key Encapsulation Mechanism or KEM are a class of encryption method designed to protect symmetric cryptographic key material from transmission using a public key scheme [NI09]. Symmetric key algorithm are the most modern block ciphers. They frequently incorporate a sequence of permutation and substitution operations. A commonly used design is an iterated cipher. It requires the specification of a round function, a key schedule and the encryption of a plain text will proceed through Nr similar rounds [FI01].

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: $K' \leftarrow CodingCRC(K)$
$5 : V' \leftarrow Polyproyec(X, Y, P)$
$6 : V \leftarrow L.S(V')$
$7 : h \leftarrow H(V)$
$8 : Sk \leftarrow KEM(h)$
$9 : C^V \leftarrow Enc(Sk, V)$

Tab. 1: Cryptographic protocol definition

4 Implementation in biometric system based fuzzy vault scheme

To implement the hardened fuzzy vault scheme, it was necessary to use several tools. We create the code in a hybrid way. For the part of the biometric system, Matlab was used since this software has libraries capable of analyzing and processing images for the extraction of the minutiae, while the part of the security of the vault was necessary to use C since it is efficient to run the algorithms of the hash function, the key encapsulation mechanism, and the symmetric key encryption. We use a computer with an Intel Core i7-9750H CPU@2.6GHz $\times$ 12 processors, with 16GiB RAM. The operating system used to compile and run the algorithms is Ubuntu 20.04.4 LTS with 64-bit architecture and the GCC version 9.4.0 compiler with the OpenSSL 1.1.1f library.

The modules made in Matlab that belong to the fuzzy vault scheme were optimized with the least number of instructions. Once the algorithms were optimized, some modifications were made for the use and integration with the modules made in C language. To calculate the clock cycles, a C module is created that allows counting the time from the first module of Matlab until the last instruction of the security algorithms, resulting in the total time of the process. The vault that results as an output of the Matlab algorithms is used as input data for the cryptographic primitives.

For cryptographic primitives, the hash function is a Shake-128 function as the input parameter to the KEM. The KEM algorithm used for the symmetric key calculation is Kyber1024, and the symmetric key algorithm used to encrypt the vault is AES-256-CBC. Another parameter taken into account for measuring the performance of the scheme is the size in bytes of the vault, both the vault that is not encrypted and the encrypted vault, and in the end, when retrieving the information after the decryption process, the size of the information obtained from the decrypted vault against the size of the original vault.

We decided to use these algorithms because they are algorithms that are resistant not only to current attacks but also to attacks using quantum computers. Another characteristic

taken into account is that these algorithms have been made efficient to be executed on ARM devices such as fingerprint scanners. Finally, the Kyber1024 algorithm was selected for being one of the best algorithms in security and execution time shown in [NI17] by the National Institute of Standards and Technology (NIST).

To evaluate the performance of the biometric system, the work focuses on the number of clock cycles necessary to carry out all the instructions that make up the enrollment phase and the verification phase. It is taken into account that in both phases the codes used are hybrid and the numbers that will be recorded will be the sum of the clock cycles of all the modules that make up the two phases.

5 Test

To begin with, a database containing fingerprints was used in the implementation. This database is made up of 80 samples that are divided into eight different fingerprints from ten different users. This database is publicly available on the internet. The use of a public database guarantees that the results of the experiment are not biased.

	Values
Genuine points	20
Polynomial degree	8
Chaff points	200
Existing points inside the vault	220

Tab. 2: Parameters used for the test of this work.

Tab 2 shows the input parameters used for the fuzzy vault scheme. These parameters were used in the tests made to measure performance. These values are taken from the numbers used in [NP08] evaluated in FVC2002-DB2.

To do the tests, the main function is created for the integration of each of the modules and the results will be printed in a CSV file. First, the variables necessary for the measurement of times are created. Variables to be used throughout the program are then created, with sizes set in the Kyber1024 header files. The program is executed so that later the size of the vault is obtained and then the results are printed in a CSV file. Having the vault and the file, we proceed to calculate the hash function. Subsequently, the key pair for the KEM is created, these keys will serve for encapsulation, in addition to the result of the previously calculated hash function being used as input. Finally, the symmetrical key obtained in the KEM is used to encrypt the vault. The encrypted vault is measured, the time count is finished and then the results are printed in the previously created CSV file.

For decryption it is similar, the encrypted vault is used as input and the output will be the original vault. The times generated by the decryption process are also saved in the CSV file.

6 Performance results

The result of Tab 3 mentions the size of the vaults in their normal form as well as in their hardened form. The hardened vaults are 0.3% larger than the original vaults but that is not a problem for today's event processing equipment. These are processes that do not need to worry as the size is not enough for processing-level concern. The hardened size provides much greater security to the vaults because they are secure with cryptography primitives that are resistant to attack today.

User	Normal Size (byte)	Hardened Size (byte)	Difference	Percentage
User 1	2,380.125	2,388	7.875	0.33%
User 2	2,409	2,416	7	0.29%
User 3	2,392.5	2,400	7.5	0.31%
User 4	2,404.875	2,412	7.125	0.30%
User 5	2,409	2,416	7	0.29%
User 6	2,404.875	2,412	7.125	0.30%
User 7	2,409	2,416	7	0.29%
User 8	2,404.875	2,412	7.125	0.30%
User 9	2,409	2,416	7	0.29%
User 10	2,396.625	2,404	7.375	0.31%

Tab. 3: Average size in bytes of the vault in the implementation.

Tab 4 indicates the processing times that were considered for this system and indicates that the system is 0.1% slower than the original system, this can predict that in reality, the hardened system is almost as efficient as the original system given times similar to the original times. The new times can show that even if the vault is hardened the processing is almost as efficient as the original processing resulting in a hardened system just as efficient as the original.

User	Normal Time (clock cycles)	Hardened Time (clock cycles)	Difference	Percentage
User 1	1,695,093.891	1,695,599.891	506	0.03%
User 2	626,671.648	627,168.898	497.25	0.08%
User 3	610,929.258	611,603.383	674.125	0.11%
User 4	648,333.508	648,923.883	590.375	0.09%
User 5	598,974.492	599,620.117	645.625	0.11%
User 6	586,409.547	587,289.172	879.625	0.15%
User 7	595,327.934	596,158.059	830.125	0.14%
User 8	616,305.367	616,879.742	574.375	0.09%
User 9	519,167.168	519,736.793	569.625	0.11%
User 10	595,323.309	595,931.184	607.875	0.10%

Tab. 4: Average time in clock cycles of the creation of the vault in the implementation.

Finally, Tab 5 shows the vaults that could be obtained from the previously described database. This database can provide enough information to be able to think that the system is efficient enough to protect the information of the users and that in the future it could be

the best option to consider for teams that have limited resources, in a few words this the system could be used in equipment with constrained resources like.

Vaults	Normal Size	Hardened Size	Diff Size	Normal Time	Hardened Time	Diff Time
Vault101_1	2,409	2,416	7	8,781,164	8,781,643.0000	479
Vault101_2	2,376	2,384	8	690,436.75	690,945.7500	509
Vault102_1	2,409	2,416	7	602,281.5	602,733.5000	452
Vault102_2	2,409	2,416	7	751,890.3125	752,319.3125	429
Vault103_1	2,409	2,416	7	601,353.25	602,042.2500	689
Vault103_2	2,409	2,416	7	627,147.9375	627,869.9375	722
Vault104_1	2,409	2,416	7	688,307.5625	688,817.5625	510
Vault104_2	2,409	2,416	7	647,736.0625	648,302.0625	566
Vault105_1	2,409	2,416	7	529,984	530,546.0000	562
Vault105_2	2,409	2,416	7	634,058	634,708.0000	650
Vault106_1	2,409	2,416	7	522,838	523,933.0000	1095
Vault106_2	2,409	2,416	7	644,935.3125	645,607.3125	672
Vault107_1	2,409	2,416	7	611,881.9375	613,003.9375	1122
Vault107_2	2,409	2,416	7	610,594.125	611,605.1250	1011
Vault108_1	2,409	2,416	7	668,876.6875	669,738.6875	862
Vault108_2	2,409	2,416	7	615,037.8125	615,784.8125	747
Vault109_1	2,409	2,416	7	545,585.375	546,348.3750	763
Vault109_2	2,409	2,416	7	447,076.5	447,648.5000	572
Vault110_1	2,409	2,416	7	700,383.4375	700,942.4375	559
Vault110_2	2,409	2,416	7	523,721.4063	524,206.4063	485

Tab. 5: Some vaults used for testing in the implementation (sizes are in bytes and times are in clock cycles).

7 Conclusions and Future Work

In the work is previously done, we have determined that it is possible to secure all biometric data with cryptographic primitives, even better it is possible to strengthen the security of these, however it is also possible to reduce the security of these systems so that the protection falls on the part of cryptography. It is possible to decrease the degree of the polynomial or the number of chaff points, however, the security will depend on the strength of the cryptographic primitives proposed for the solution.

It is important to take into account that all biometric data are susceptible to attacks since they are unique data, but protecting them with some cryptographic primitives could make these data much more secure. The execution times observed in the work show us that in reality, it is not necessary to sacrifice more time or space to have systems that are much more secure than those that currently exist. A special feature is that passwords are now becoming obsolete, so biometric information takes on great importance in security.

Finally, it is necessary to verify that these systems are not susceptible to any type of attack within these systems. It is known that symmetric key algorithms are very resistant, but it would be necessary to be able to corroborate this security to know that biometric systems would be secure in the near future.

References

[De22] De Abiega-L'Eglisse, Alfonso Francisco; Rosas Otero, Mario; Azpeitia Hernández, Vladimir; Gallegos-Garcia, Gina; Nakano-Miyatake, Mariko: A New Fuzzy Vault based Biometric System Robust to Brute-Force Attack. Computación y Sistemas, 26(3), 2022. To be published.

[FI01] FIPS-197: , Advanced Encription Standard (AES), 2001. Last updated October 05, 2021.

[FI15] FIPS-202: , SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, 2015. Last updated November 11, 2020.

[JK15] Jain, Rubal; Kant, Chander: Attacks on Biometric Systems: An Overview. International Journal of Advances in Scientific Research, 1:283, 09 2015.

[Ju06] Juels, Ari: A Fuzzy Vault scheme. Designs, Codes and Cryptography, 38:237–257, 02 2006.

[Me10] Merkle, Johannes; Niesing, Matthias; Schwaiger, Michael; Ihmor, Heinrich; Korte, Ulrike: Performance of the Fuzzy Vault for Multiple Fingerprints. pp. 57–72, 01 2010.

[NI09] NIST: , SP 800-56B - Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography, 2009. Last updated March, 2019.

[NI17] NIST: , Post-Quantum Cryptography Standardization, 2017. Last updated December 02, 2021.

[NP08] Nandakumar, Karthik; Pankanti, S.: Fingerprint-Based Fuzzy Vault: Implementation and Performance. Information Forensics and Security, IEEE Transactions on, 2:744 – 757, 01 2008.

[Ra21] Rathgeb, Christian; Merkle, Johannes; Scholz, Johanna; Tams, Benjamin; Nesterowicz, Vanesa: Deep Face Fuzzy Vault: Implementation and Performance. Computers & Security, 113:102539, 11 2021.

[RB08] Reddy, Edara; Babu, I.: Performance of Iris Based Hard Fuzzy Vault. volume 8, pp. 248–253, 08 2008.

[Ta13] Tams, Benjamin: Attacks and Countermeasures in Fingerprint Based Biometric Cryptosystems. arXiv, p. 32, 04 2013.