Fuzzy vault and template-level fusion applied to a binary fingerprint representation

Julien Bringer, Mélanie Favre, Chloé Pelle*, Hugues de Saxcé*

Morpho
firstname.lastname@morpho.com

Abstract: The fuzzy vault is an error-tolerant authentication scheme that stores data in a privacy-preserving way. It allows to hide and retrieve a secret by presenting a set of elements that can slightly differ. This paper aims at applying the scheme to a transformation-invariant binary fingerprint representation. We also show how to perform multi-finger fusion with no extra space requirement and perform fuzzy vault in this case. Our results are quite encouraging with respect to the inherent constraint of decreasing False Accept attacks on template protection schemes in order to ensure higher privacy.

1 Introduction

User privacy is a major concern in our digital world. Unlike cryptography where one single bit error leads to a rejection, biometrics has to deal with noisy data in order to authenticate people. Privacy-preserving approaches to protect biometric data have been proposed in the literature. Among them, biometric template protection is a set of techniques (cf. [RU11] for a survey) that can offer a first layer of security. The underlying idea is to derive a protected version of the enrolled template and to store it instead of the first one. The transformation mechanism is public and it must be hard to reconstruct the template from the protected one without the genuine user's biometric data. Among the well known biometric template protection schemes there are the Fuzzy Commitment [JW99] and the Fuzzy Vault [JS02, JS06] schemes. Both approaches commit to a secret value under a noise-tolerant key. Fuzzy vault encrypts and decrypts secret information using a fuzzy unordered set. Briefly illustrated, Alice locks a secret κ using a set A of elements from a public universe. Bob must present a set B that substantially overlaps A to unlock the vault and retrieve κ.

Applied to biometrics, the fuzzy vault scheme binds a biometric template with a cryptographic key to build a secure sketch. It is computationally hard to retrieve either the template or the key without any knowledge of the biometric data. In [UPJ05] fuzzy vault is applied to fingerprints where elements of the sets A and B are concatenation of quantized x and y minutiae coordinates. A drawback of this is the need of an *alignment* step. Most of the time, alignment is achieved using an auxiliary data, see for instance [UJ06]. To avoid

*Work done during internships at Morpho

this step that leaks some information, an automatic method for absolute pre-alignment has been proposed in [Tam13a]. Specific rotation and translation invariant fingerprint representations have also been studied in [JA06] in order to be suitable for fuzzy vault.

Another binary representation of fingerprints invariant to translation and rotation has been proposed in [BD10]. The idea is to look at the neighborhood-level of minutiae and use a dictionary approach of presence or absence of given patterns in each fingerprint. We propose to apply fuzzy vault on this representation. Compared to minutiae-based fuzzy vault, we don't need a pre-alignment step or the storage of auxiliary data and we don't need to be noise-tolerant for minutiae. In many minutiae-based fuzzy vault implementations there must be space between genuine and chaff points to tolerate small minutiae shifts. The space is then not totally filled with chaff and this can improve cross-matching attacks (see for instance [SB07] or [Tam13b]). We also perform multi-finger fusion by doing a feature-level fusion on the templates to improve biometric results. A similar approach has been studied in [NJ08], the aim being to get a unique template representing multiple biometric traits. Besides the accuracy improvement, the advantage of this kind of fusion is double: first there is a space improvement compared to the storing of multiple templates. Then, an adversary can't unlock a part of the fuzzy vault by providing a single template, thus strengthening the privacy property of the scheme as FA attacks become more costly.

The rest of the paper is structured as follows. Section 2 recalls the fuzzy vault scheme. Section 3 details the fingerprint representation we use and explains how fusion is performed on it. In Section 4 we detail the experiments we carried out and conclude in Section 5.

2 Fuzzy Vault

Let $\mathcal{F}$ be a finite field of cardinality q. We have a secret value $\kappa \in \mathcal{F}^k$ and a secret set $A = \{a_i \in \mathcal{F}, i = 1, \ldots, t\}$. Let r be an integer greater than t. The Fuzzy Vault $V_A \in \mathcal{F}^{2r}$, as described in [JS06], takes as input A and κ. V_A is evaluated with the LOCK algorithm that transforms κ into a polynomial p and evaluates each value in A by p to form a pair. At the end, some chaff points, *i.e.* points taken at random, are added to the vault, so that an attacker can't distinguish a genuine point from a fake one.

LOCK$(A, \kappa) = V_A = \{(x_i, y_i) \in \mathcal{F} \times \mathcal{F}, i = 1, \ldots, r\}$
$p \leftarrow \kappa$
for $i = 1$ **to** t **do**
  $x_i = a_i$
  $y_i = p(x_i)$
end
for $i = t + 1$ **to** r **do**
  $x_i \in \mathcal{F} \setminus A$ at random
  $y_i \in \mathcal{F} \setminus \{p(x_i)\}$ at random
end

UNLOCK$(V_A, B) = \kappa$ or $null$
$S = \emptyset$
for $i = 1$ **to** t **do**
  $(x_i, y_i) \overset{b_i}{\leftarrow} V_A$
  $S = S \cup (x_i, y_i)$
end
$\kappa =$RSDecode(k, S)

Given V_A and a set B with a sufficient overlap with A, one should be able to recover κ

(note that B doesn't need to have the same size than A). In [JS06], this is achieved with algorithm UNLOCK that relies on the error correction capacity of Reed-Solomon codes.

Many security analysis of the fuzzy vault scheme have been carried out. In [JS06] the number of polynomials of degree less than k passing through t points of the vault is considered. In fact, for any $\mu > 0$ there exists at least $\frac{\mu}{3}q^{k-t}(r/t)^t$ such polynomials, with probability at least $1 - \mu$. However, this is true only if the distribution of A is uniform. Another approach is to see the fuzzy vault in the secure sketch model. In [DRS04], authors estimate the entropy loss of the scheme to be at most $(t-k)\log(q) + \log\binom{q}{r} - \log\binom{q-t}{r-t} + 2$. The scheme is also vulnerable against cross-matching of templates from different enrollments if the whole space filled by genuine and chaff points is not stable, cf. [SB07]. Finally, a general brute force attack against fuzzy fingerprint vaults is given in [MMT09]. The idea is to pick up any k points in the vault, determine the corresponding interpolation polynomial and try to find at least t points in the vault which lay on the graph of that polynomial. The maximal number of operations necessary to decrypt the vault is given by $8r(k-1)\left(\frac{r}{t}\right)^k$.

3 Fingerprint binary representation

3.1 Original setting

In this work we choose to use the fingerprint binary representation introduced in [BD10]. Authors use local neighborhood of minutiae, called vicinities, to deal with local skin distortions. They have a fixed-size dictionary of possible vicinities, called representatives, which they compare fingerprint vicinities to. This gives birth to a fixed-size binary vector with the information of presence or absence of a close dictionary vicinity in the fingerprint.

In more details, on one side the N representatives rp_i come along with N thresholds th_i indicating the probability of existence of the considered minutiae neighborhoods. Along this, a fingerprint is encoded by considering for each minutia its surrounding neighborhood. Each of these vicinities v_i is then compared to each representative giving a comparison score, which is then compared to the current representative's threshold. If the score is bigger than the threshold, than a one is put in the vector, else a zero (cf. Figure 1).

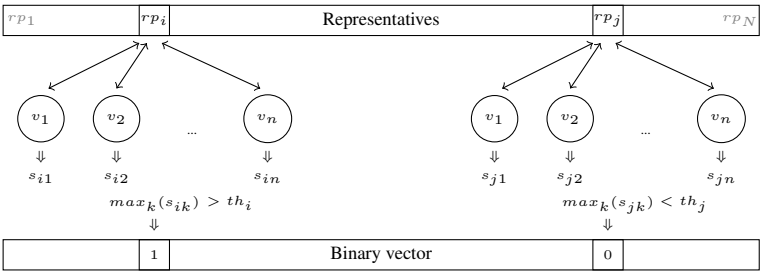

Figure 1: Binary vector construction

In practice, we modify a little the previously described encoding algorithm by constructing binary vectors with a fixed number t of bits set to one, and this independently of the amount of minutiae in the fingerprint. This is simply done by setting a 1 in the vector for the t closest representatives. t is chosen rather small and hence the vectors are finally very sparse. Comparison between two such vectors is performed using a bit-wise AND: the score corresponds to the amount of common bits set to one. See [BD10] for more details.

3.2 Feature-level fusion

The use of more than one fingerprint during authentication of a person has a major benefit as it leads to less false accepts (FA). There are several ways to achieve this, we focus on feature-level fusion. The approach described in [BD10] has the advantage of being well suited for it. We follow authors' idea of performing fusion on multi-acquisitions of the same finger by extending this principle to multi-finger fusion. Indeed, in the feature vectors a position indicates if a vicinity is present or absent inside the considered fingerprint. When several fingerprints are considered, one needs to put to one positions in the feature vector related to the vicinities of all fingerprints. This corresponds to a bit-wise OR between all binary feature vectors that have to be merged. The corresponding merged vector has the same size as the underlying ones and it can be compared by a bit-wise AND just as before.

Number of templates	Average weight	Minimum weight	Maximum weight	Absolute standard deviation	Standard deviation (%)
2	1192	1175	1200	3,79	0,32
4	2357	2329	2378	9,15	0,39
8	4603	4537	4646	19,69	0,43

Table 1: Hamming weight of the fused vectors for 600 bits to one on FVC2000 DB2

The last assertion has been validated by the study of the number of bits set to one when a bit-wise OR is performed on several binary vectors. Table 1 sums up our observations for templates containing 600 bits set to one on FVC2000 DB2 [FVCa]. In fact, the average Hamming weight stays high: different fingers do not share many common patterns.

3.3 Applying fuzzy vault on the feature vectors

Traditionally, fingerprint fuzzy vault takes sets of minutiae - often pre-aligned - as inputs and evaluates a thresholded set intersection on it. Here we have to deal with binary vectors, so we simply see positions set to 1 as elements of the set to input in the vault. Regarding notations of Section 2, we have t that equals the amount of bits set to 1 in the binary vector and $r = N$. We have to choose an underlying finite field $GF(q) = \mathcal{F}$ with $q > N$. Figure 2 illustrates the way we construct the fuzzy vault with the binary vectors, with the subset of the N first elements of $\mathcal{F}$, fully filled of t genuine points and $N - t$ chaff points.

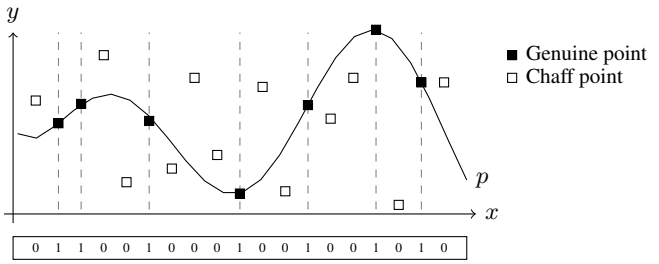

Figure 2: Fuzzy vault construction for binary vectors

4 Experiments

We performed our tests on two public databases: FVC2000 DB2 [FVCa], a low-cost capacitive sensor database with 800 images (8 acquisitions times 100 subjects), and FVC2002 DB2 [FVCb], an optical sensor database with 880 images (8 acquisitions times 110 subjects). None of them has acquisitions of different fingers of a same person, so we created random fused templates coming from 2, 4 or 8 different people for a total of 40 000 comparisons with a ratio of 1 genuine matching test over 40 tests. For reference templates we merged random templates coming from the four first acquisitions while verification templates are fusions of templates of the four last acquisitions.

4.1 Fusion on binary templates

We worked on binary vectors of size $N = 56786$ where there are $t = 600$ bits set to one. Table 2 shows the biometric performances of our feature-level fusion. As we can expect, the more templates are merged, the better are the authentication results.

Number of	FVC2000 DB2			FVC2002 DB2		
templates	Threshold	FAR (%)	FRR (%)	Threshold	FAR (%)	FRR (%)
1	36	0.0094	32.87	40	0.0104	39.03
2	77	0.0076	17.30	82	0.0117	20.91
4	181	0.0051	2.10	177	0.0093	2.54
8	549	0.0102	0.30	525	0.0093	0.27

Table 2: Biometric performances of feature-level fusion

By observing in table 2 the threshold w for FAR around 0.01%, we note that w does not grow linearly with respect to the amount of fused templates: the more fingers we use, the faster w increases. For eight finger-fusion a lot of matching positions do not come from the same underlying sub-templates. Nevertheless, performances are still better this way.

4.2 Fuzzy vault on feature-level merged vectors

As w is quite small, it implies to correct up to 87.5% of errors in the fuzzy vault setting. We therefore use Guruswami and Sudan's list-decoding algorithm [GS98] to achieve the needed error-correction capacity. Recalling that we have a $k-1$-degree polynomial, the algorithm has a theoretical bound $w > \sqrt{(k-1)t}$. Our experiments were run on a computer with a 2.93GHz Intel Core2 Duo processor and 4GB of RAM. We implemented the fuzzy vault scheme in C using PARI library for some complex operations. We met a computational barrier on a big linear system and needed to review the bound in a harder version $w > \sqrt{1+2t(k-1)}$. This technical condition limits the achieved size of k. Table 3 gives biometric and security (according to [MMT09]) results of our experiments on FVC2000 DB2 (similar results for FVC2002 DB2). Execution times are given for one decoding.

Number of fused templates	FAR (%)	FRR (%)	t	w	k	Computational Security (bits)	Execution time (s)
1	0.0094	32.87	600	36	2	31	1.7
2	0,0076	17,30	1192	77	3	36	14
4	0,0051	2.10	2356	181	7	53	101
8	0.0102	0.30	4599	549	33	143	763

Table 3: Performances of fuzzy vault on $\mathcal{F} = GF(2^{17}-1)$ on FVC2000 DB2

Performing fusion improves biometric efficiency and helps to increase security regarding computational attacks (the most efficient among those mentioned in Section 2 being [MMT09]). As there are more common points between two sets when several templates are fused, the degree of the polynomial increases, and so does security. These good results come with an increase of execution time too due to the higher complexity of the decoding.

4.3 Security against biometric attacks

We succeed to have a FAR lower than 10^{-4} with a very low FRR for 4 or 8 fused templates. This leads to relatively more costly FA attacks against privacy of a template protection scheme. Although 10^{-4} is not negligible, it is lower than one can expect in this setting without fusion. This is an encouraging consequence of our construction.

We also investigate the case where an attacker has access to partial biometric information, suppose for instance he's able to recover 1, 2 or 3 fingerprints out of 8. We evaluate his ability to authenticate with this partial knowledge. To do so we create fused templates with partial matching abilities with other ones, like for instance for eight-finger fusion:

- One false match: $\text{fus}(01\_1, 02\_2, 06\_1, 40\_3, 75\_4, 12\_1, 99\_3, 27\_4)$ vs $\text{fus}(01\_5, 10\_7, 23\_8, 04\_7, 62\_6, 16\_8, 23\_5, 77\_6)$
- Two false matches: $\text{fus}(01\_1, 02\_2, 06\_1, 40\_3, 75\_4, 12\_1, 99\_3, 27\_4)$ vs $\text{fus}(01\_5, 02\_7, 23\_8, 04\_7, 62\_6, 16\_8, 23\_5, 77\_6)$
- Three false matches: $\text{fus}(01\_1, 02\_2, 06\_1, 40\_3, 75\_4, 12\_1, 99\_3, 27\_4)$ vs $\text{fus}(01\_5, 02\_7, 06\_8, 04\_7, 62\_6, 16\_8, 23\_5, 77\_6)$

We compare the corresponding score distributions to standard matching and non-matching score distributions (see Figure 3). Without surprise, the more sub-templates match to the reference sub-templates, the better the scores are. We also evaluate more accurately the amount of partially matching vectors that would be authenticated if the system was tuned for some FAR. Table 4 shows us that with 1 genuine sub-template among 8, an attacker has only 6% of chances to authenticate in a system tuned for a 0.1% FAR. This shows the positive impact of fusion on the security of the system against biometric attacks.

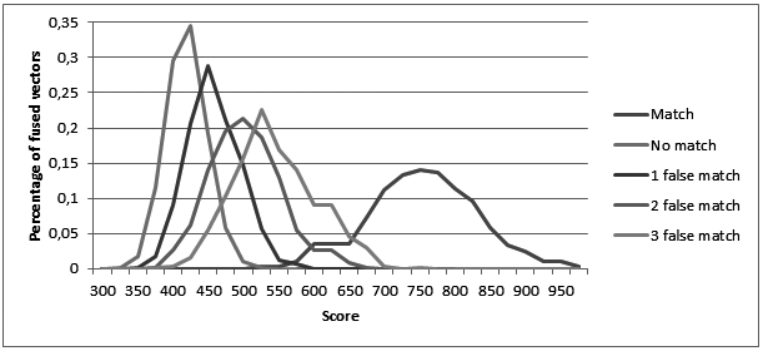

Figure 3: Score distributions for partial matching tests on FVC2000 DB2

FAR	% accepted for 1 FM	% accepted for 2 FM	% accepted for 3 FM
0.1%	6.0%	36.5%	65.5%
0.01%	2.2%	22.1%	47.4%

Table 4: False match vector acceptance rates on FVC2000 DB2

5 Conclusion

In this paper, we apply the fuzzy vault scheme to a binary transformation-invariant fingerprint representation. We show how to increase biometric expectations by performing template-level fusion with no storage overhead and adapt the scheme to this end, leading to high computational security and encouraging properties against biometric attacks. We discuss how this helps to increase the privacy and security of the template protection system. Future works will be to further improve the advantages of such idea.

Acknowledgments

This work has been partially funded by the European FP7 FIDELITY project (SEC-2011-284862).

References

[BD10] Julien Bringer and Vincent Despiegel. Binary feature vector fingerprint representation from minutiae vicinities. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6. IEEE, 2010.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004*, pages 523–540. Springer, 2004.

[FVCa] Fingerprint Verification Competition (2000) . http://bias.csr.unibo.it/fvc2000/.

[FVCb] Fingerprint Verification Competition (2002) . http://bias.csr.unibo.it/fvc2002/.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes. In *FOCS*, pages 28–39, 1998.

[JA06] Jason Jeffers and Arathi Arakala. Minutiae-based structures for a fuzzy vault. In *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pages 1–6. IEEE, 2006.

[JS02] Ari Juels and Madhu Sudan. A Fuzzy Vault Scheme. In *Proceedings of IEEE Internation Symposium on Information Theory*, Lecture Notes in Computer Science, page 408, 2002.

[JS06] Ari Juels and Madhu Sudan. A Fuzzy Vault Scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.

[JW99] Ari Juels and Martin Wattenberg. A Fuzzy Commitment Scheme. In Juzar Motiwalla and Gene Tsudik, editors, *ACM Conference on Computer and Communications Security*, pages 28–36. ACM, 1999.

[MMT09] Preda Mihailescu, Axel Munk, and Benjamin Tams. The Fuzzy Vault for Fingerprints is Vulnerable to Brute Force Attack. In Arslan Brömme, Christoph Busch, and Detlef Hühnlein, editors, *BIOSIG*, volume 155 of *LNI*, pages 43–54. GI, 2009.

[NJ08] Karthik Nandakumar and Anil K Jain. Multibiometric template security using fuzzy vault. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6. IEEE, 2008.

[RU11] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):1–25, 2011.

[SB07] Walter J Scheirer and Terrance E Boult. Cracking fuzzy vaults and biometric encryption. In *Biometrics Symposium, 2007*, pages 1–6. IEEE, 2007.

[Tam13a] Benjamin Tams. Absolute fingerprint pre-alignment in minutiae-based cryptosystems. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, pages 1–12. IEEE, 2013.

[Tam13b] Berend-Benjamin Tams. *Cryptanalysis of the Fuzzy Vault for Fingerprints: Vulnerabilities and Countermeasures*. PhD thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, 2013.

[UJ06] Umut Uludag and Anil Jain. Securing fingerprint template: Fuzzy vault with helper data. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 163–163. IEEE, 2006.

[UPJ05] Umut Uludag, Sharath Pankanti, and Anil K Jain. Fuzzy vault for fingerprints. In *Audio- and Video-Based Biometric Person Authentication*, pages 310–319. Springer, 2005.