Biometric Cryptosystem Using Online Signatures

Alisher Kholmatov and Berrin Yanikoglu

Sabanci University
Faculty of Engineering and Natural Sciences
Istanbul 34956, Turkey
alisher@su.sabanciuniv.edu, berrin@sabanciuniv.edu

Abstract. Biometric cryptosystems combine cryptography and biometrics to benefit from the strengths of both fields. In such systems, while cryptography provides high and adjustable security levels, biometrics brings in non-repudiation and eliminates the need to remember passwords or to carry tokens etc. In this work we present a biometric cryptosystems which uses online signatures, based on the Fuzzy Vault scheme of Jules et al. The Fuzzy Vault scheme releases a previously stored key when the biometric data presented for verification matches the previously stored template hidden in a vault. The online signature of a person is a behavioral biometric which is widely accepted as the formal way of approving documents, bank transactions, etc. As such, biometric-based key release using online signatures may have many application areas.
We extract minutiae points (trajectory crossings, endings and points of high curvature) from online signatures and use those during the locking & unlocking phases of the vault. We present our preliminary results and demonstrate that high security level (128 bit encryption key length) can be achieved using online signatures.

1 Introduction

Biometric authentication is the task of verifying the identity of individuals based on their physiological (e.g. fingerprint, face) or behavioral traits (e.g. signature). During authentication, biometric traits of a person are matched against the stored biometric profile of the claimed identity and access is granted if there is sufficient match. Biometric systems are gaining popularity as more trustable alternatives to password-based security systems, since there are no passwords to remember and biometrics cannot be stolen and are difficult to copy. Biometrics also provide non-repudiation (an authenticated user cannot deny having done so) to some degree because of the difficulty in copying or stealing someone's biometrics.

On the other hand, biometric traits are known to be variable and noisy. The same biometric may change between consecutive acquisitions (due to injury, ageing, even mood etc.) and noise can be introduced to a biometric signal by an acquisition device or the environment. While it would be very convenient to use biometric traits for encryption, for instance someone using his fingerprint or handwritten signature to encrypt a document and securely send it over public network, this seems very difficult due to the aforementioned variability of the biometric signals and the fact that encryption and decryption operations cannot tolerate the perturbation of even a single bit. In its most basic sense, generating a cryptographic key from a biometric trait, say fingerprints, has not been very successful, as it involves obtaining an *exact* key from a highly variable data. For instance Feng and Wah has only been able to generate a 40-bit private key from online signatures with an 8% equal error rate (error obtained either because the private key generated from a genuine signature does not match the public key of the person, or the private key generated from a forgery matches the genuine public key of a template) [7].

Uludag et al. makes the distinction between two general approaches within what they call *crypto-biometric systems*), according to the coupling level of cryptography and biometrics [6]: *Biometrics-based key release* refers to the use of biometric authentication to release a previously stored cryptographic key. Biometric authentication is used as a wrapper, adding convenience to traditional cryptography where the user would have been in charge of remembering his/her key; however the two techniques are only loosely coupled. *Biometrics-based key generation* refers to extracting/generating a cryptographic key from a biometric template or construct. In this case, biometrics and cryptography are tightly coupled: the secret key is bound to the biometric information and the biometric template is not stored in plain form.

Recent work of Juels et al. [1] and Tuyls et al. [2] are classified as biometrics-based key generation, as they require a tight coupling of cryptography and biometrics. In particular, the work of Juels et al. forms the basis of this paper [1]. The fuzzy vault construct is an example of recent research which focus on combining cryptography and biometrics to take advantage of the benefits of both fields [1][2][3][4][5]: while biometrics provide non-repudiation and convenience, traditional cryptography provides adjustable levels of security and can be used not just for authentication, but also for encryption. Jules and Wattenberg proposed the *fuzzy commitment* scheme [8]; later Juels and Sudan extended it to the *fuzzy vault* scheme [1] and described how it can be used to construct/release an encryption key using one's biometrics: a secret (cryptographic key) is "locked" using a biometric data of a person, such that someone who possesses a substantial amount of the locking elements (e.g. another reading of the same biometric) would be able to decrypt the secret. In summary, using the fuzzy vault scheme, one can store a cryptographic key in what's called a vault and later extract/release it by presenting his/her biometrics.

The fuzzy vault scheme is classified as a key-generation scheme in Uludag et al., because of its tight coupling of cryptography and biometrics [6]. However, in the sense that the biometric data releases a previously stored key, it can also be seen as a releasing mechanism. Yang and Verbauwhede and Uludag et al. [9, 10] implemented the fuzzy vault using fingerprints, making simplifying assumptions about the biometric data.

In this paper we present an implementation of the fuzzy vault construct using online signatures [1].

2 Previous Work

2.1 Fuzzy Vault

Jules and Sudan proposed a scheme called *fuzzy vault*, which they call an error tolerant encryption operation [1]. Fuzzy vault scheme provides a framework to encrypt (”lock”) some secret value (eg. cryptographic key) using an unordered set of locking elements as a key, such that someone who possesses a substantial amount of the locking elements will be able to decrypt the secret. It is based on the difficulty of the polynomial reconstruction problem. The encoding and decoding are done as follows:

Assume that Alice wants to secure her cryptographic key S (a random bit stream) using an arbitrary set of elements A. She selects a polynomial $P(x)$ of degree D and encodes S into the polynomial's coefficients. Encoding can be achieved by slicing S into non-overlapping bit chunks and then mapping these onto the coefficients. The mapping must be invertible meaning that the coefficients can be unambiguously mapped back to the corresponding bit chunks, which when concatenated will reconstruct the S. Then, Alice evaluates the polynomial at each element of her set A and stores these evaluation pairs into the set G, where $G = \{(a_1, P(a_1)), (a_2, P(a_2)), ..., (a_N, P(a_N))\}$, $a_i \in A$ and $|A| = N$. Finally, she generates a random set R of pairs such that none of the pairs in that set lie on the polynomial; and she merges the sets G and R into a final set, to obtain the vault, which she then makes public. Note that within the vault, it is not known which points belong to G and which ones belong to R. All the steps required to lock a secret in the Fuzzy Vault are graphically represented in Figure 1.

Now suppose that Bob has his own set of elements B and he wants to find out (”unlock”) Alice's secret locked in the vault. He will be able to do so only if his set B largely overlaps with Alice's A, so as to identify a substantial number of the pairs that lie on the polynomial, from the vault. Given at least $D+1$ pairs that lie on the polynomial, he applies one of the known polynomial reconstruction techniques (eg. Lagrange interpolating polynomial) to reconstruct the polynomial and thus extracts her secret S. Notice that if Bob does not know which of the points of the vault lie on the polynomial, it should be computationally infeasible for him to unlock the vault.

Whereas perturbation of a single bit in a key of a classical cryptosystem (eg. AES, RSA) hinders decryption completely, the fuzzy vault allows for some minor differences between the encryption & decryption keys; here the unordered sets used to lock & unlock the vault. This fuzziness is necessary for use with biometrics since different measurements of the same biometric often result in (slightly) different signals. Furthermore, for most of the known biometric signals, it is hard to establish a consistent ordering within the measured features. For

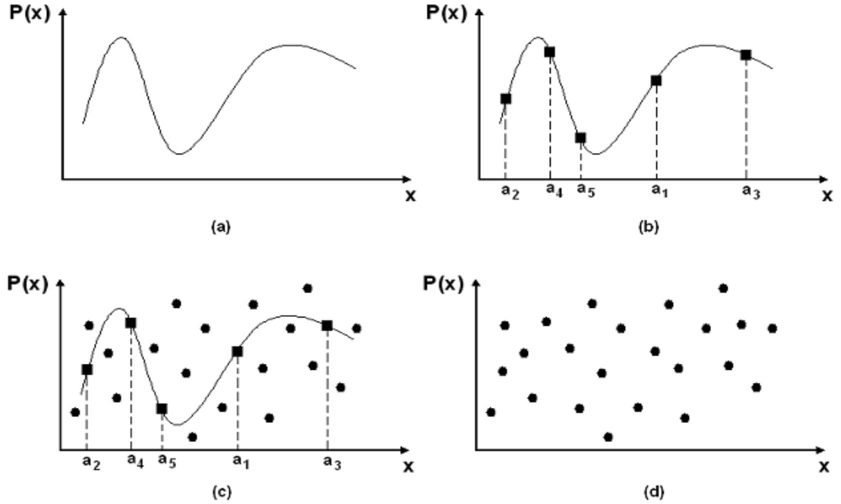

Fig. 1. Vault *Locking* phase: (a) Create a polynomial by encoding the *Secret* as its coefficients. (b) Project genuine features onto the polynomial: a_i represents the subject's i'th feature. (c) Randomly create chaff points (represented by small black circles) and add to the Vault. (d) Final appearance of the Vault, as stored to the system database.

instance two impressions of the same fingerprint can have substantial distortion and the number of features may vary between the two impressions.

Since the fuzzy vault scheme can tolerate some difference between the locking & unlocking sets and does not requiring ordering amongst the set elements, it has a potential to be used in biometric cryptosystems. However it is not straightforward how to implement a fuzzy vault using a particular biometric data, due to the difficulty of matching the template and query biometric signals (i.e. locking and unlocking sets, respectively) within the presence of random data (the chaff points).

2.2 Fuzzy Vault with Fingerprints

Uludag et al. [10] demonstrated a preliminary implementation of the fuzzy vault scheme using fingerprints. Yang and Verbauwhede [9] also implemented the fuzzy vault with fingerprints, but they made the assumption that rotation & translation invariant features can be reliably extracted from minutiae -which is difficult in practice. Furthermore, they store reference minutia point along with the vault, which may also leak some information. We will review the system by Uludag et al. as it relates the most to our proposed scheme.

Minutia points of template & query fingerprints were used as locking & unlocking sets, respectively, to lock a 128-bit long data (S) which forms the cryptographic key. More precisely, the values obtained by concatenation of the corresponding x & y coordinates of minutiae points were used as set elements.

To make sure that the desired S was unlocked from the vault through an error-prone process, cyclic redundancy check bits (16 bits) were concatenated to S. Then, S, together with its check bits, was divided into non-overlapping chunks (16 bits each), giving the coefficients, of an 8th degree polynomial. To lock the secret, template minutiae set was projected onto this polynomial and random chaff points not lying on the polynomial are added, to form the vault. Based on their empirical estimations, they used only 18 minutia points and 200 chaff points.

To unlock the secret, i.e. reconstruct S, they first match the query minutia set with the abscissa part of the vault and identify candidate points lying on the polynomial. Since $D + 1$ points are required to reconstruct a polynomial of degree D, all possible 9 point combinations of the candidate set are tried, to find the one with the correct check bits. S is successfully unlocked when the check bits verify. Authors report a 79% of correct reconstruction rate with 0% false accept rate.

To bypass the problem of matching the minutiae points and finding an upper bound for the performance of the scheme, the authors have used a fingerprint database where minutia points and the correspondence between template & query fingerprints were established by an expert. During their experiments, the minutiae sets of mating fingerprints were pre-aligned (i.e. rotated & translated) according to the established correspondence, and used as such.

3 Proposed Method

In this section we demonstrate an implementation of the Fuzzy Vault scheme using online signatures. Online (dynamic) signature are captured by pressure-sensitive tablets that extract dynamic properties of a signature in addition to its shape, which is the only available information in offline (static) signatures found on bank checks and documents. Dynamic features include the number and order of the strokes, the overall speed of the signature, the pen pressure at each point, etc. and make the signature more unique and more difficult to forge. As a result, online signature verification is more reliable than offline signature verification and much more commonly used as a biometric. Throughout the rest of the paper, we use the term signature to mean an online signature.

It is very challenging to find a representation of a particular biometric suitable for the fuzzy vault scheme. Similar to the minutiae points defining branch and end points in fingerprints, we extract the event points of a signature (hereafter called minutiae points) and use them as locking or unlocking sets in the fuzzy vault construct. Given a signature's trajectory, we consider crossings, endings and places of high curvature as minutiae points, where each minutia is a two dimensional point (x & y coordinates) defined in the Cartesian space of the pressure sensitive tablet .

Figure 2 demonstrates an example signature with marked minutiae points points. For the time being, minutiae points are marked by experts, i.e. they are not extracted automatically. This is done in order to measure true performance of the vault, i.e. to prevent error which could be introduced by an imperfect minutiae extraction algorithm.

Note that these minutiae points do not capture the timing of a signature, nor the ordering of the strokes; in fact they only use the shape of the signature. Nonetheless, the results are encouraging compared to the results obtained with previous implementations of fuzzy vault.

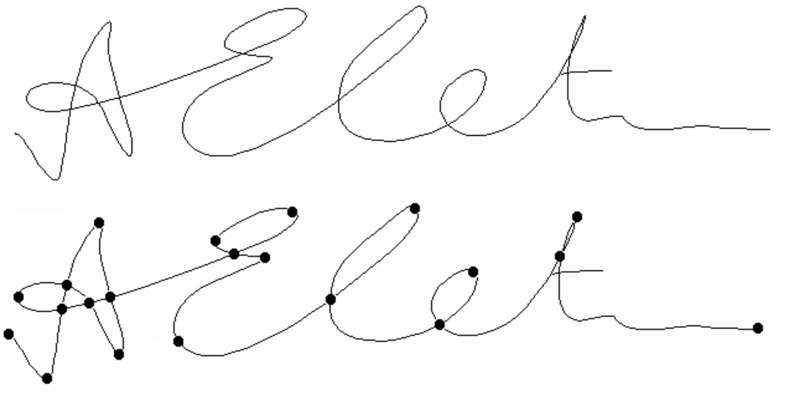

Fig. 2. The Figure demonstrates a genuine signature (on top) and minutiae points marked for that signature (at the bottom).

During the locking phase, we concatenate the x & y coordinates of minutiae points and project these on to the secret polynomial, as described in Section 2.1. Minutiae coordinates and their corresponding projections, along with random chaff points constitute the locking set of the vault. Although chaff points are created randomly, special attention must be paid to the situations where chaff points are generated too close to the genuine points or other chaff points. If placed too close to minutiae points, the unlocking performance will be reduced since chaff point located in close proximity to the genuine points may be mistakenly matched during unlocking phase. Closely generated chaff points may leak information if a malicious attacker knows the closest possible distance between two genuine points. Finally, chaff points must be homogeneously distributed in the vault space; otherwise, they may leak information, enabling an attacker to reduce his search space and decrease the vault's strength. Figure 3 shows a sample vault which is generated within this system.

During the unlocking phase, the correspondence between points of unlocking minutiae set and those of the vault must be determined. Although there are numerous point matching algorithms, we used exhaustive matching to reduce the error which may be introduced by the matching algorithm. Exhaustive

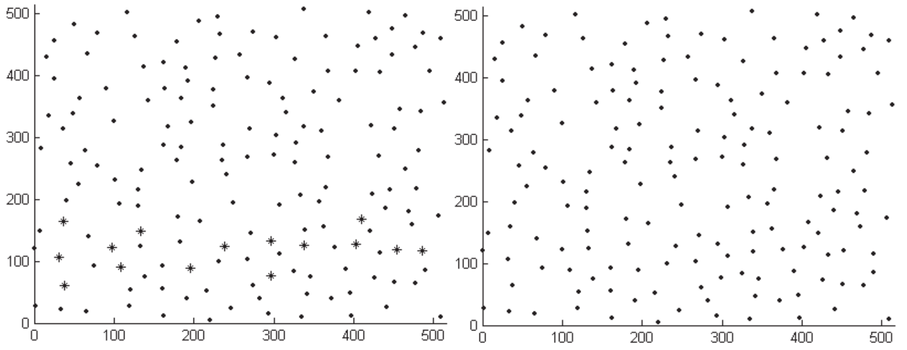

Fig. 3. Fuzzy Vault Locking: genuine points (stars) and chaff (dots) points are represented separately on the left for the sake of clarity. The actual vault shown on the right, only contains the points without any information about their source (genuine or chaff).

matching is performed by applying all possible rotations & translations (in the vault space) to the unlocking set, to find the alignment with the most number of matching points. Figure 4 shows the result of matching genuine (left) and forgery (right) minutiae sets with the vault (matched vault points are circled). As can be seen, while genuine unlocking set substantially overlaps with the vault's genuine points, the forgery set dioes not.

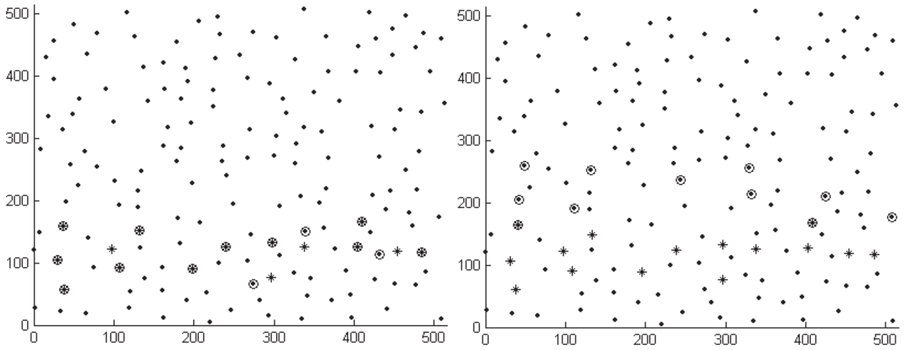

Fig. 4. Fuzzy Vault Matching: matching of genuine (left) and forgery (right) minutiae sets with the vault from fig. 3. matched vault points are shown circled and minutiae (stars) and chaff (dots) points of the vault are represented differently for the sake of clarity.

As a result of matching, we obtain a candidate set of points which are then used for decoding the secret key. During each iteration of the decoding phase, we select $D + 1$ points from candidate set where D is the degree of the secret polynomial and use them to decode the secret as described in Section 2.1. The polynomial degree was fixed at nine and didn't change during testing.

During the locking phase, a cryptographic hash of the secret is also stored in the system database, along with the vault points. During unlocking, we calculate the hash of the decoded secret and compare it with the one stored in the system's database. Decoding phase terminates when both hash values match (secret is decoded) or when maximum number of iterations is performed (secret not revealed). These steps are similar to the fuzzy vault implementation of Uludag et al. [10].

4 Results

The system performance was evaluated using sample signatures supplied by 10 subjects enrolled to our system. All signatures in our dataset were acquired using same tablet and sampled at 100 sample points per second rate. Each subject supplied 4 genuine signatures for a total of 40 signatures. There were no constraints on how to sign, nor was any information given about the working principles of the system, so that the subjects signed in their most natural way.

All possible combinations of 2 signatures out of 4 reference signatures supplied by each subject were used to measure the vault's genuine performance (i.e. correct unlocking rate of the vault). Thus, 6 such signature pairs were obtained for each user and 60 such pairs were tested in total.

To collect skilled forgeries, we added a signing simulation module to our system. Simulation module animates the signing process of a given signature so that the forger could see not only the signature trajectory's points sequence but also the signing dynamics (speed and acceleration). Forgers had a chance of watching the signature's animation several times and practice tracing over the signature image a few times before forging it. We have totally collected 30 skilled forgeries (3 forgeries for each subject), following above mentioned protocol. Each forgery signature was paired with each of the corresponding reference signatures (4 such pairs per forgery signature) and used as such during locking & unlocking phases, where reference signature was used to lock and forgery signature to unlock the vault, respectively. Totally 120 such pairs were obtained for our dataset, which are used to measure weakness of the vault against fraud.

We have obtained the 8.33% of failure rate to unlock the vault when a genuine signature was presented (i.e. can be considered as false reject rate) and 2.50% of false unlocking rate when the vault was attempted to be opened using forgery signatures. Obtained results are promising. Most of the failures to unlock the vault with genuine signatures are due to the high variability within reference signatures, supplied by a corresponding user. On the other hand, false unlocking rate obtained used forgery signatures is due to the fact that we use only minutiae points of the signature, which don't incorporate dynamic features of corresponding genuine signer thus easier to compromise. The false accept rate can be reduced by increasing the polynomial degree, of course at the expense of slight increase in FRR. Conversely, FRR can be lowered by more efficient chaff point generation, which could assure that number of genuine points matching chaff points doesn't exceed a certain threshold. On average, it took approximately 30 seconds to unlock a vault with it's corresponding genuine signature. Troughout the test a notebook computer with Intel Celeron (M) 1.5GHz and 512 megabyte of RAM hardware configuration was used. All algorithms were implemented using Matlab.

5 Summary and Conclusion

Bio-cryptosystems combine cryptography and biometrics to take advantage of the benefits of both fields: while biometrics provide non-repudiation and convenience, traditional cryptography provides adjustable levels of security and can be used not just for authentication, but also for encryption. Fuzzy Vault scheme [1] is a promising framework for such bio-cryptosystems, as it doesn't require ordered representation of a biometric and it can tolerate variations within biometric up to some extent.

We have demonstrated an implementation of the Fuzzy Vault scheme using online signatures, which runs in real time. Even though our method is a relatively straightforward extension of the fuzzy vault scheme implementation by Uludag et al. [10], the issues encountered in implementing the fuzzy vault with online signatures were non-trivial. Besides, it is the first realisation of the scheme using online signatures which demonstrated promising performance results.

Acknowledgments

We would like to thank TUBİTAK (The Scientific and Technical Research Council of Turkey) for the Ph.D. fellowship support granted to Alisher Kholmatov throughout this research.

References

1. Juels, A., Sudan, M.: A fuzzy vault scheme. IEEE International Symposium on Information Theory (2002) 408
2. Tuyls, P., Verbitskiy, E., Ignatenko, T., Denteneer, D., Akkermans, T.: Privacy protected biometric templates: Acoustic ear identification. Proceedings of SPIE: Biometric Technology for Human Identification **Vol. 5404** (2004) 176–182
3. Davida, G., Frankel, Y., Matt, B.: On enabling secure applications through on-line biometric identification. In IEEE Symposium on Privacy and Security (1998) 408
4. Soutar, C., Roberge, D., Stojanov, S., Gilroy, R., Kumar, B.V.: Biometric encryption using image processing. In Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II **Vol. 3314** (1998) 178–188
5. Linnartz, J., Tuyls, P.: New shielding functions to enhance privacy and prevent misuse of biometric templates. Proceeding of AVBPA (LNCS 2688) (2003) 393–402
6. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.: Biometric cryptosystems: Issues and challenges. In: Proceedings of the IEEE. Volume 92(6). (2004)

7. Feng, H., Wah, C.: Private key generation from on-line handwritten signatures. Information Management & Computer Security, **10/4** (2002) 159–164
8. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. Conference on Computer and Communications Security, ACM Press. (1999) 28–36
9. Yang, S., Verbauwhede, I.: Secure fuzzy vault based fingerprint verification system. Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on (2004) 577–581
10. Uludag, U., Pankanti, S., Jain., A.: Fuzzy vault for fingerprints. Proceeding of International Conference on Audio- and Video-Based Biometric Person Authentication (2005) 310–319