Signature Based Fuzzy Vaults With Boosted Feature Selection

George S. Eskander, Robert Sabourin and Eric Granger
Laboratoire d´imagerie, de vision et d´intelligence artificielle
École de technologie superiéure, Université de Quebec
1100, rue Notre-Dame ouest, Montreal, Canada, H3C 1K3
geskander@livia.etsmtl.ca, Robert.Sabourin@etsmtl.ca, Eric.Granger@etsmtl.ca

***Abstract*—Handwritten signatures are commonly employed in many financial and forensic processes, and secure offline signature verification systems (SV) are required to automate such processes. In this context, bio-cryptography systems based on the handwritten signatures may be considered for enhance security. This paper presents a bio-cryptography system that constructs Fuzzy Vaults (FVs) based on the offline signature images. Boosting Feature Selection is employed to select features while training weak classifiers of offline SV systems. The indexes of selected features correspond to the most stable and discriminant features from a user's signature images, and are used to encode user-specific FVs. A password is employed as a second authentication measure, to further enhance system security. During authentication, a user provides both the signature and the password to decode the FV and decouple his private key. If the FV is correctly decoded, the user is authenticated by the verification system. The proposed FV implementation alleviates the security vulnerabilities of the classical SV systems like template security, repudiation, irrevocability, and bypassing the classification decision. Moreover, simulations performed on a real-world signature verification database (with random, simple, and skilled forgeries) indicate security guarantees against stolen authentication measures. While compromised signatures or passwords lead to complete fail ($FAR = 100\%$) of the classical SV or password protected cryptography systems respectively, compromised signatures lead to FAR of 0.1%, and compromised passwords leads to FAR of 15% with the proposed system.**

I. Introduction

Although biometric authentication systems allow for enhanced security, biometric systems are vulnerable to a wide range of attacks [1]. Among different biometric authentication systems, handwritten signature verification systems (SV) are required to automate the authenticity of individuals in many financial and forensic processes such as cashing checks, credit card transactions, document authentication, etc [2]. In literature, most research focused on enhancing the accuracy of the signature verification systems without focusing on the security issues.

Bio-cryptography has been introduced mainly to alleviate the key management problem of cryptographic methods by using biometrics to secure the cryptography private keys, instead of the ordinary passwords [3]. However, bio-cryptography may also be considered as a counter measure, to the security vulnerabilities of the ordinary biometric authentication systems. In this paper bio-cryptography is applied to address both the security of cryptography keys by means of handwritten signature images, and the security vulnerabilities of the classical offline signature verification systems.

Key-binding is generally the most reliable bio-cryptography scheme. The cryptography keys and the biometric keys are coupled in a way that they can not be decoupled unless applying a genuine sample of the biometric trait. A challenging problem for these schemes is managing the fuzziness of the biometric signals that results from the intrapersonal variability and the interpersonal similarity. The Fuzzy Vault scheme (FV) has successfully been applied to design reliable key-binding bio-cryptography systems that absorbs the biometric fuzziness to a great extend [4].

In literature, all reliable FV implementations are mostly based on physiological biometrics, namely, fingerprint [5], face [6], 3D face [7], iris and retina [8], and palmprint [9]. FV implementations based on the handwritten signatures (behavioural traits) are presented in [10],[11]. Although the FV based on the online signatures have shown acceptable performance, it is shown that the features extracted from the offline signature images are not suitable for the fuzzy vault construction.

This paper addresses the design of a FV system based on the offline handwritten signature images. The main contribution is a scheme based on machine learning to extract stable and discriminant representation of the offline signature images. Once this reliable representation is generated, it is used to encode the fuzzy vault in a straightforward way. To this end, multi-scale Extended Shadow Codes features are extracted from the enrolled user signature images. Dissimilarity Representation (DR) [12] approach is employed to overcome the limited training signature samples, and the Boosting Feature Selection (BFS) [13] approach is employed to automate the process of selecting few good features from the huge amount of extracted features. BFS selects features while training weak classifiers of a classical SV system, and consider the indexes of the selected features as a writer-dependent signature representation model. This model is used to extract the most stable and discriminant features from the user signature images. Both the user model and the extracted features are concatenated to produce a set of stable and discriminant points which are used to encode a user-specific FV.

Some authors proposed the usage of a password as a second authentication factor to harden the FV [14]. In these systems,

the password is used to encrypt user features and fuzzy vault. In the proposed system, the second factor is considered in a different way: the password encrypts the user-based signature representation model. User's FV and encrypted representation model are stored in a database as the user template. During authentication, the user applies both the correct password and a genuine query signature image. The password is used to retrieve the user representation model by which the good features are extracted from the query signature image. Finally, decoding points are produced by concatenating both user model and features, and then used to decode the user FV in a straightforward way.

For proof-of-concept simulations, a Brazilian database [15] contains 7,920 signatures images of 168 writers is used to design FV and classical SV systems. Recognition accuracy of the designed systems proved the efficiency of the proposed Boosting Feature Selection method. The proposed FV system security is studied and proved to outperform both of the classical SV systems and the password protected cryptography systems.

II. Challenges to design a FV based on handwritten signatures

The fuzzy vault (FV) is a cryptographic construction that binds a secret message, e.g., cryptographic key, with an unordered and fuzzy locking set. In case of biometric based FV implementations, the FV binds a secret cryptography key with a set of points extracted from the biometric trait. As shown in Figure 1, biometric traits can be characterized by a set of points extracted directly from their images in the spatial space, e.g., minutiae points in fingerprints, cross points in signatures, etc. During authentication, the secret message can be correctly decoupled only if the unlocking points significantly match the locking points. Accordingly, the reliability of a FV implementation depends on the stability of points extracted from the biometric trait. For the physiological biometrics like fingerprints, retina, iris, palmprint, etc, it is easy to locate invariant points in the spatial space. Variability of traits mostly results from the signal acquisition processes, such as unaligned samples, noisy readings, etc, but it is not an intrinsic property of the trait itself. As a result, applying some pre-processing techniques on the trait, e.g. alignment methods, filtering noise through quantization, etc, could absorb most of variation. On the other hand, in case of unstable behavioural biometrics like offline handwritten signatures, variability besides resulting from the trait acquisition processes, it could be an intrinsic property of the trait itself as it results from a changing human behaviour. It is very difficult to locate invariant points from the offline signature image in the spatial space, and to cancel the variability by applying the same techniques applied on some physiological biometrics.

Besides the biometrics variability, the similarities of traits for different persons may result in unreliable FV implementation. For instance, if two fingerprints of different persons matches substantially, their minutiae points will match accordingly and result in false FV decoding. Similarities may occur

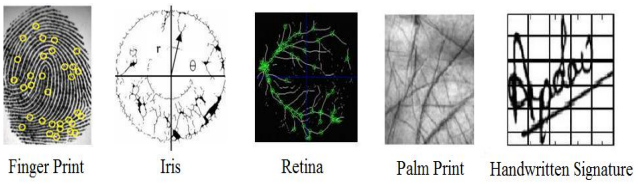


Fig. 1. Extracting Biometric Represenation in the Spatial Space

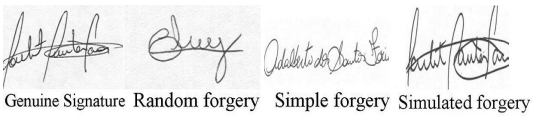


Fig. 2. Examples of the signature images

according to the natural similarities between persons, or it may occur when biometrics traits are imitated by forgeries. With physiological biometrics, the natural interpersonal similarities are rare, and it is difficult to produce forged physiological biometric traits. On the other hand, with offline handwritten signatures, signatures of different persons may have some similarities, and it is easy to imitate signature images. Figure 2 shows a signature image of a genuine user and some forged images. There are three types of signature forgery: random, simple, and simulated. For random forgery, the forger has neither access to the genuine signature nor to the signer's name. So, he produces the signature randomly. For simple forgery, the forger has no access to the genuine signature but he knows the signer's name. He therefore produces a signature based on the name. For the simulated forgery, the forger has access to a sample of the signature. He can therefore simulate the genuine signature. It is clear that extracting representations from signature images in the spatial space may lead to very similar representations of different persons, or forged images and to false matches accordingly.

III. Extraction and Selection of Signature Representation

This scheme applies the Dissimilarity Representation (DR) [12] and the Boosting Feature Selection (BFS) [13] approaches, to select the best features representing the signature images. Multi-scale Extended Shadow Codes feature representation are extracted from signatures images, then it is transformed to the dissimilarity space by applying dichotomy transformation. The resulted dissimilarity representation is then used by a Boosting Feature Selection algorithm to select the best features.

A. Multi-scale Extended Shadow Codes

Extended Shadow Code (ESC) [16] method is considered for multi-scale feature extraction. As illustrated in Figure 3, the ESC consists in the superposition of bar mask array over the image of a handwritten signature. Each bar is assumed to be a light detector related to a spatially constrained area of the 2D signal. A shadow projection is defined as the simultaneous projection of each black pixel into its closest

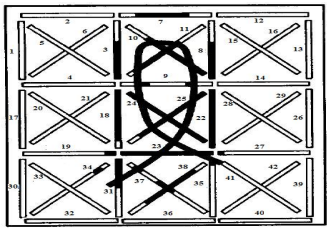
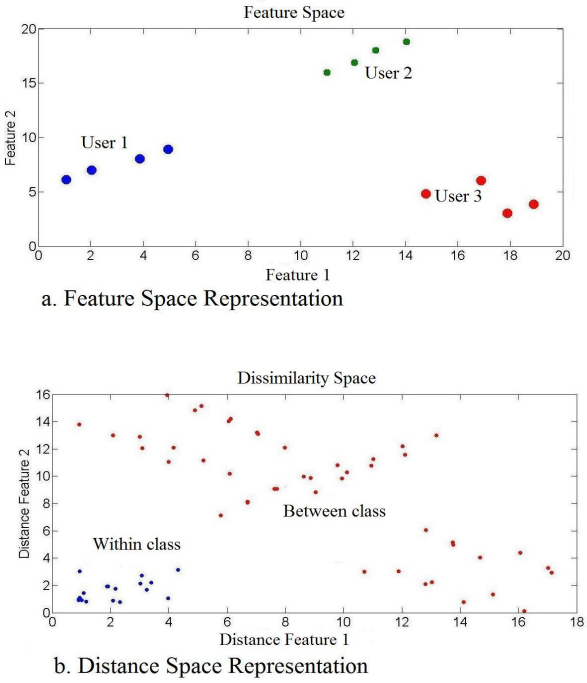


Fig. 3. ESC features [16].



a. Feature Space Representation

b. Distance Space Representation

Fig. 4. Feature space and dissimilarity space signature representations

horizontal, vertical and diagonal bars. A projected shadow turns on a set of bits distributed uniformly along the bars. After all the pixels of a signature are projected, the number of on bits in each bar is counted and normalized to the range of [0,1] to constitute the ESC feature value. ESC extraction can be done based on different scales. Generally, the dimension of the single-scale ESC feature vector (DSF) is given by: $DSF = 4hv+h+v$, where h, v are the numbers of the single-scale horizontal and vertical bars respectively. For instance, in Figure 3 there are three horizontal and three vertical sections result in a representation scale of 3×3, with 42 ESC features.

Although ESC features selected based on a single extraction scale can represent a user signature image uniquely, it is known that different signatures can be represented by a specific extraction scale and there is no single scale suitable for all signatures. Accordingly, we extract ESC on multiple scales and concatenate the extracted feature vectors to constitute a high-dimensional feature vector. Features that are most suitable for a specific user are included in this high-dimensional feature vector and the learning algorithm selects a concise, stable and discriminant feature vector. These user-based features are used to encode the user FV.

B. Similarity learning

Dissimilarity representation approach is applied in the field of offline handwritten signature verification system (SV) [17]. Instead to design a multi-class classifier or one-class classifier for each writer, producing a writer-dependent signature verification system (WD-SV), a single two-class classifier is designed for the whole population, producing a writer-independent signature verification system (WI-SV).

Figure 4.a illustrates the feature space representation of signature images. The representation of the three users result in three different clusters each has four samples. For simplicity, this Figure plots only two features, while in the real case the feature vector is a multi-dimensional vector that may contain thousands of features. In this example, the two selected features result in non-overlapped and compact clusters. As described in the next subsection, a boosting method is employed for feature selection and training of a classifier. The performance of the classifier declines significantly when the number of users is large and when the number of samples per user is small.

To alleviate this problem, the signature representation is transformed from its original feature space to the dissimilarity space by applying the dichotomy transformation. This transformation can be achieved by computing the distance vector between each pair of signatures. The distance vector is simply computed by subtracting the values of each two coordinate features in the two signature representations, and then takes its absolute value as the new feature value. It is important to note that the resulting distance vectors belong to two different classes. The within-class case includes all distance vectors that result from comparing two signatures belong to the same user. The between-class includes all distance vectors that result from comparing two signatures belong two different users.

Figure 4.b illustrates the dissimilarity representation of signatures from Figure 4.a after applying the dichotomy transformation. It can be noticed that, while the original feature representation results in three classes with four samples each, the dissimilarity representation results in two classes with eighteen samples for the within-class, and forty-eight samples for the between-class. Hence, the twelve samples in the feature space results in sixty-six samples in the dissimilarity space. Accordingly, training of the m-class classifier with few samples per class in the feature space is replaced with the more tractable training of a two-class classifier with several samples per class.

Although the similarity representation results in a more tractable optimization problem, there are some practical limitations. The between-class cluster consists of distances between a specific user's signatures and other systems users' signatures. When starting the system, there is no user enrolled in the system so we can not generate the between-class cluster. To overcome this problem, a dataset that contains simulated signatures, that have the same nature of the expected real

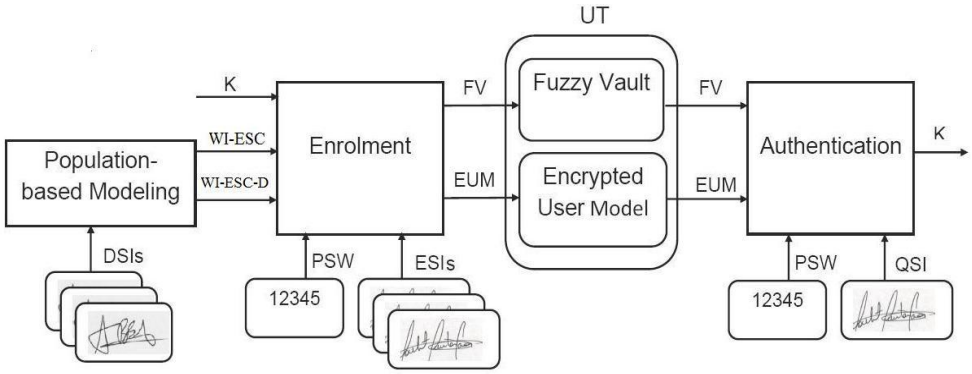


Fig. 5. Block Diagram of the FV System

system users signatures, is used. From these signatures the two classes are generated. Generally, if there are R different simulated users, each have S signatures, the number of the generated distance samples is $\binom{RS}{2}$ different distance vectors, of these $R\binom{S}{2}$ are within-class distance vectors, and $S^2\binom{R}{2}$ are between-class distance vectors. These simulated clusters allow to gain knowledge about the best features for the whole population. The training process employs these clusters and results in a smaller subset of features that suit the whole population. We call this process a population-based feature selection.

To select the features that suit a specific real user, the reference signature images collected during enrolment are used to generate signature representations based on the whole-population feature vector. These vectors are used to generate a within-class cluster by applying the dichotomy transformation on them, and a between-class cluster by finding the distance between them and the whole-population signatures. Finally, the classifier is re-trained using these two clusters to find a smallest subset of features that suits the specific user. We call this process a user-based feature selection.

C. Boosting Feature Selection

Boosting feature selection (BFS) methodology is applied to implement both population-based and user-based feature selection processes while training the classifier. Boosting mechanisms, generally, can be operated as a control layer over any weak learner (e.g. decision stump, MLP, etc) to guide it in the iterative learning of the underlying function behind the training data, and produce a strong classifier. In our scheme, the AdaBoost algorithm is applied for BFS [18]. It is an "adaptive" version of boosting that reacts "adaptively" to the performance of the already trained weak learner. During each iteration, a single weak learner algorithm learns a finite set of training samples, and performance is assessed according to the weights associated with each sample. The performance of the current weak learner affects both the distribution over samples for the next iteration, and the share of that learner in the final strong hypothesis. In this paper, decision stump is used. It is a threshold classifier that provides the classification decision based on the value of a single feature. Accordingly, a single feature is selected in each boosting iteration. At the end, the classifier corresponds to a committee of decision stumps each utilizes a single ESC feature. The indexes of selected features are stored as a generic vector of the best writer-independent ESC feature (WI-ESC).

To increase the discrimination of this population-based representation, user-based feature selection is applied. When a user is enrolled to the system, a dissimilarity representation of his enrolled signatures is used to build a writer-dependent signature verification system (WD-SV) by applying BFS in the same manner. Finally, indexes of the features embedded in this classifier are used as a writer-depended feature model (WD-ESC) model.

IV. A Fuzzy vault system based on the offline signature images

In the proposed FV implementation, user signature images and his password are used together to secure the cryptography private keys. To treat FV as a traditional classifier, its outputs can be transformed to class labels. If the key is decoupled successfully, it produces positive class label, otherwise it produces a negative class labels.

Figure 5 shows the general framework of the proposed system. It consists of three main sub-systems. First, a population-based modelling subsystem is used to model the feature representation of the signature images of the entire population. It uses the Development Signature Images (DSIs) to generate a population-based (Writer-independent) ESC representation model (WI-ESC). This model is used to extract a writer-independent ESC representation of the development images (WI-ESC-D).

The enrolment module is used to enrol new users to the system. It uses the Enrolling Signature Images (ESIs) of enrolled user, his password (PSW), his private key (K), WI-ESC, and WI-ESC-D. It results in an Encrypted version of a User-based signature representation Model (EUM). This User Model (UM) is a vector of the best user-specific feature indexes (WD-ESC). Moreover, it results in a Fuzzy Vault (FV) encodes the user private key (K) and the representation of his Enrolled Signature Images (ESIs). Both the EUM and FV are stored together as the user template (UT).

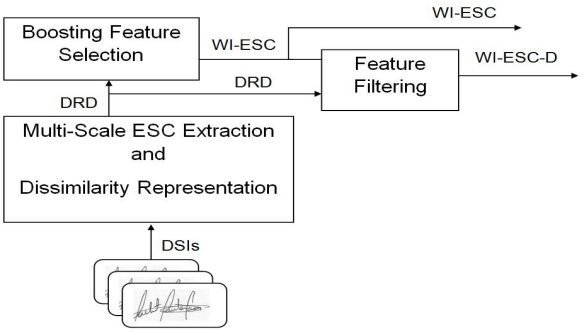


Fig. 6. Block Diagram of the Population-Based Modeling subsystem

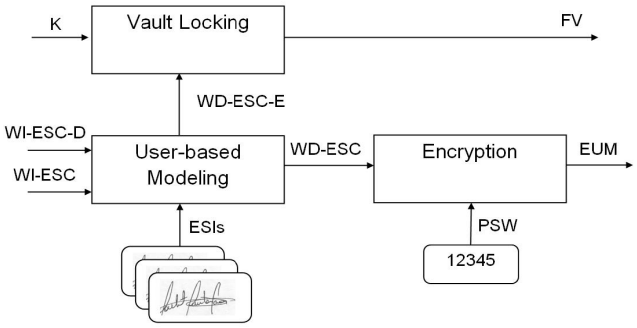


Fig. 7. Block Diagram of the Enrolement subsystem

The authentication sub-system is used to authenticate a user who posses both a genuine signature image and a correct password. The password (PSW) is used to decrypt the EUM. Then, features are extracted from the user Query Signature Image (QSI) based on WD-ESC. Finally, once the right feature vector is extracted, it is used to unlock the FV and release K.

A. Population-Based modeling

As shown in Figure 6, this subsystem consists of three modules: multi-scale ESC feature extraction and dissimilarity representation, Boosting Feature Selection (BFS), and feature filtering. The multi-scale representation of the development signatures is transformed to the dissimilarity space by applying the dichotomy transformation. The resulting dissimilarity representation of the development signature images (DRD) is then sent to the BFS module, to develop a writer-independent signature verification system (WI-SV). The distinct feature indexes embedded in this system are considered as the write-independent ESC signature representation model (WI-ESC). Finally, The representation of the development signature images is filtered by WI-ESC to produce a writer-independent ESC representation of the development dataset (WI-ESC-D).

B. Enrolment

This subsystem is used to enrol new users to the system. As shown in Figure 7, it consists of three modules: 1) user-based modeling, 2) model encryption, and 3) vault locking.

The user-based modeling module uses the enrolment signature images (ESIs) to develop a write-dependent ESC signature representation model (WD-ESC). As with the population-based modeling module, the signature images are used to extract multi-scale ESC feature vector. However, with this module, the multi-scale ESC feature representation of the enrolled signatures is filtered before being sent to the BFS module. The feature filter uses the population-based model (WI-ESC) to filter the multi-scale ESC feature representation of the enrolled signatures and generate a writer-independent ESC representation of the enrolling dataset (WI-ESC-E). Then, the BFS module uses both WI-ESC-D, and WI-ESC-E to produce the writer-dependent signature verification system (WD-SV). Once the WD-SV is designed, the indexes of the best **r** of its embedded features are considered as a writer-dependent feature representation model (WD-ESC).

The encryption module uses the user password (PSW) to generate an encryption key (EK) by which the WD-ESC is encrypted and stored as encrypted version of user model (EUM). Finally, all of the user template signature, his WD-ESC feature vector, and his WD-SV, are deleted from memory for security.

The vault locking module locks the private key K with features extracted from the enrolled signature images based on WD-ESC (WD-ESC-E). First, features are adapted for the FV construction by concatenating the feature indexes (WD-ESC) and the feature values (WD-ESC-E) to constitute the FV locking points. This step preserves the features location even if being reordered (scrambled), and also it increases the discrimination of the FV locking points and increases the system accuracy accordingly. To this end, the ESC locking index **lind** is represented in a 8-bit locking feature index **lfind**, and the ESC locking value **lval** is quantized based on the quantization size **q**. If $q < 8$ bits, padding bits are added to left of quantized value to constitute a 8-bits locking feature value **lfval**. Both words are then concatenated to constitute a 16-bit locking feature **lf**. All the **r** locking features constitute a **r**-dimensional locking feature vector LF. The user secret key **K** (of any length **k**) is used to build a single dimensional polynomial **P** (of degree **n**) by splitting the key into 16-bits units and assign the i^{th} unit as $c_{i^{th}}$ polynomial coefficient. A CRC-16 field is computed based on the coefficients and added as a coefficient c_n, in order to detect decoding errors at the authentication time. The locking feature vector LF is projected into the constructed polynomial results in a projected locking vector PLF. The projection of each feature is calculated as $P(lf_i) = c_0\, lf_i^n + c_1\, lf_i^{n-1} + ... + c_n$. To hide the **r** genuine points (LF, PLF), **s** chaff points (CF, PCF), where $s >> r$, are synthetically generated and added to the set to constitute an ordered list $V' = \{X', Y'\}$, where $X' = (LF, CF)$, and $Y' = (PLF, PCF)$. Finally, V' is scrambled to constitute the $FV = \{X, Y\}$.

C. Authentication

The authentication sub-system authenticates a user who posses both a valid password (PSW) and a genuine query

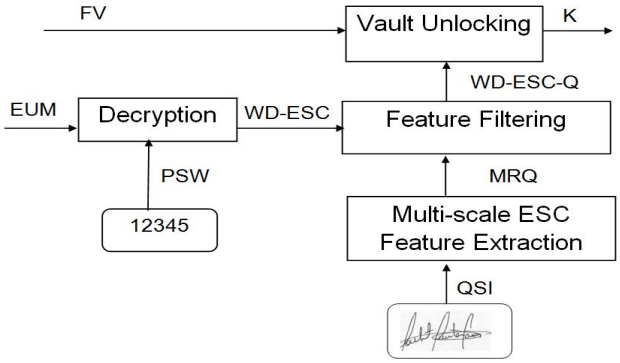


Fig. 8. Block Diagram of the Authentication subsystem

signature image (QSI). As illustrated in Figure 8, the PSW regenerates the user encryption key (EK) by which the EUM is decrypted and generates WD-ESC. The multi-scale ESC feature extraction module generates a multi-Scale representation of the query signature (MRQ). Then, WD-ESC is used to filter MRQ and generates a writer-dependent representation of the query signature (WD-ESC-Q).

This feature vector is then adapted for the FV construction and constitutes the unlocking feature vector UF. This vector is used to filter the matching points in the FV, by comparing its elements with X. The matching points constitute a matching list ML of length r', where $n \leq r' \leq r$. All **nc** combinations of $n+1$ points of the matching list constitute an unlocking list UL. Each vector of UL is used to recover the encoded polynomial P by applying the Lagrange interpolation method. The resulting polynomial coefficients are combined to constitute the secret key K^* and then the CRC is computed to check the correctness of the recovered P. If any of the recovered polynomials is correct, the corresponding K^* is released to the user as his secret key **K**, else the user is not authenticated.

V. Results and discussion

A. Experimental Methodology

1) Signature Database: A Brazilian database [15] is used for proof-of-concept simulations. It contains 7,920 samples of signatures that were digitized as 8-bit grayscale images over 400×1000 pixels, at resolution of 300 dpi. The signatures were provided by 168 writers and are organized as follows: the first 60 writers have 40 genuine signatures, 10 simple forgeries, and 10 simulated forgeries per writer, and the other 108 have just 40 genuine signatures per writer. Figure 2 shows sample signatures of a specific user.

The experimental database is split into two sets, the development dataset D composed of the last 108 writers with only 30 signatures out of the 40 genuine signatures, and the exploitation dataset E composed of the first 60 writers. The exploitation dataset E is used to generate two subsets: the reference set R composed of 30 randomly selected genuine signatures per writer in E and the questioned set Q composed of the 10 remaining genuine signatures, 10 simple, and 10 simulated forgeries from each writer, plus 10 random forgeries selected from the genuine signatures of 10 different writers.

To perform unbiased evaluation of the different designed systems, only one signature from **R** is used at a time to test the verification performance of SV systems, and only one reference is used to encode each FV. The questioned set **Q** is used, so the total number of testing samples is: 60 user $\times30$ genuine reference signatures $\times40$ query signatures $= 72000$ samples. Of these, 18000 are genuine, 18000 are random forgery, 18000 are simple forgery, and 18000 are simulated forgery samples.

2) Feature Extraction: The features are extracted based on different scales depends on the size of the extraction grid. Let $H = \{2, 5, 10\}$ be a set of 3 horizontal scales defined by their number of grid rows and $V = \{3, 6, 12\}$ be a set of 3 vertical scales defined by their number of grid columns. The Cartesian product $H \times V$ results in the 9 single scales (2×3, 2×6, 2×12, 5×3, 5×6, 5×12, 10×3, 10×6, 10×12). Concatenating these single scale features results in a multi-scale feature vector of 1575 dimensions.

3) Feature Selection:

a) Population-Based Feature Selection: The development dataset **D**, is used to design 10 WI-SV for the 9 single scales ESC feature vectors and a multi-scale vector (consists of the 9 single scale vectors concatenated in one vector). For each vector, the learning set **L** is generated from **D** by applying dichotomy transformation on all genuine signatures from every writer to constitute a within-class set, and on genuine signatures of every writer against random forgeries selected from other writers' signatures to constitute a between-class set with equivalent number of counterexamples. AdaBoost has been run on **L** for 200 boosting iterations results in 10 different WI-SV each consists of 200 weak classifiers. Figure 9 shows the DET (Detection Error Tradeoff) curve, that illustrate the trade-off between the FAR (False Accept Rate) and the FRR (False Reject Rate), of the designed SV and FV systems. It is clear that the multi-scale based SV system outperforms the single-scale based SV systems. Also, the best single scale is 10×6 and the worst single-scale is 1×1.

b) User-Based Feature Selection: As the multi-scale ESC based SV system outperforms the other single-scale based SV systems, only the multi-scale WI-ESC feature vector is utilized to design a WD-SV system. Each reference signature in **R** is used to extract multi-scale WI-ESC feature vector for the specific user. Then, dichotomy transformation is applied on all genuine signatures vectors of the user to constitute a within-class set. The whole-population between-class set and the user within-class set are used to train the AdaBoost for a WD-SV system. The designed WD-SV is used to find the indexes of the best 20 WD-ESC features. This feature vector is then encrypted by the user password and stored as the user model (EUM).

4) Fuzzy Vault Encoding/Decoding: The FV design parameters are as follows: $k = 128$ bits, $n = 8$, $r = 20$, and $s = 200$. For each user in **E**, two types of FVs are encoded: WI-FV encoded using the best single-scale WI-ESC feature vector that

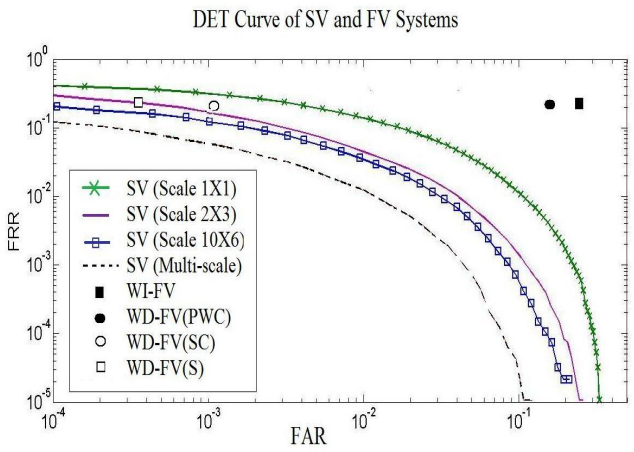


Fig. 9. DET curves for SV and FV systems

is scale: 10×6 (using the best 20 out of the 256 features in this scale representation), and WD-FV using the multi-scale WD-ESC feature vector. All feature vectors are represented using 5 different quantization sizes q: $\{3,4,5,6,7bits/ESCvalue\}$. So for each reference signature in **R**, there are 5 WI-FVs, and 5 WD-FVs each encodes a single reference signature. The 40 signatures per user of **Q** are verified using all the WI-FV and WD-FV versions for each user.

B. Verification performance

The performance measures used are: 1) GAR is the percentage of all genuine signatures that are correctly authenticated, 2) FAR-random, 3) FAR-simple, 4) FAR-simulated are the percentages of all random, simple, and simulated forgeries respectively that are false authenticated, and 5) Average FAR is the percentage of all forged signatures that are false authenticated. As different quantization sizes are used, we report here only the results of the best quantization size, i.e. $q=3$. For FV based on WI features (WI-FV), only signature images are used for verification. For FV based on WD features, both signatures and passwords are used as users shall know there passwords in order to retrieve the indexes of their features.

To evaluate the impact of each of the identification elements (signature and password), three scenarios are implemented to decode the WD-FV; i) WD-FV(PWC): (password compromised access) in which forgeries used their forged signatures but they have the password compromised, ii) WD-FV(SC) (signature compromised access) in which forgeries compromised the genuine signatures and used it but they entered random passwords, iii) WD-FV(S) (secure access) in which forgeries used their forged signatures and entered random passwords. Table I shows the performance of the FV systems.

With WD-FV(PWC), we simulated that forgeries had access to the password, i.e. the feature indexes. So, we cancelled the impact of the using the password, and hence any performance improvement of WD-FV than WI-FV is due to the quality of selected features. It is clear that the personalization of features has much improved FAR for all types of forgeries. Also, the

TABLE I
COMPARISON OF WI-FV AND WD-FV SYSTEMS

Performance%	WI-FV	WD-FV		
		PWC	SC	S
GAR	74	75	75	75
FAR-random	9	3	—	0
FAR-simple	18	7	—	$3.33E^{-}4$
FAR-simulated	47	36	—	$8.33E^{-}4$
Average FAR	24.66	15.33	$1.1E^{-}3$	$4E^{-}4$

benefit of using a signature to secure the private cryptography keys, as another authentication measure besides the ordinary password, is clear. Even if the password is compromised, the possibility of compromising the cryptography key is about $\%15$. With WD-FV(SC), the benefit of using user password is clear. Even if forgeries could capture genuine signature from any paper or document, and use it to access a system, the possibility for their success is about 0.1%. If both the identification elements were secure as in case of WD-FV(S), the possibility of forgeries success is 0.04%.

From Figure 9, it is clear that the ideal performance of WD-FV, i.e. WD-FV(S), is close to the performance of SV of scale 2×3 at $FAR\simeq0.04\%$. Performance of WD-FV in case the signatures are compromised, i.e. WD-FV(SC), is close to the performance of SV of scale 2×3 at $FAR\simeq0.1\%$. However, if the password compromised, the performance of WD-FV is worse than all the SV systems, but better than WI-FV.

C. Security Analysis

Although the proposed system shows enhanced security, there are security vulnerabilities related the FV construction. Assume the attacker could access the user FV template; in this case he can attack the system by brute-force search. If the attacker can locate $n+1$ genuine points out of the total $r+s$ vault point, he will be able to reconstruct the correct polynomial and decode the private key accordingly. The upper-limit of the brute-force attack search time for a FV of r encoding points, s chaff points, and polynomial degree of n is: $\binom{r+s}{n+1}$. So, for $r=20, s=200, n=8$, the upper-limit of search space is: $\binom{220}{9}\simeq2.8\times10^{15}$. Among these combinations $\binom{20}{9}\simeq1.6\times10^{5}$ will decode the FV. So, the expected number of polynomial evaluation to decode the FV $\simeq 2.8\times10^{15}\div1.6\times10^{5}\simeq1.7\times10^{10}\simeq2^{34}$. So for such system, the actual cryptography key of 128 bits is secured by a FV of security level equivalent to a 34 bits key.

The worst brute-force attack occurs if the attacker could compromise the password; in this case he can filter the genuine points using the compromised feature indexes, and shrink the search space. To alleviate this problem, we generated half of the chaff features (CF) with the same values of the genuine feature indexes. For instance, when generating 200 chaff features, 100 of them have their x-axis as the same as of the genuine features, while the other 100 locates on different positions. In this case, if the password is compromised half of the chaff points will be filtered and excluded accordingly from the brute-force search space. For instance, the new system

parameters will be, $r = 20, s = 100, n = 8$. Accordingly, the new of search space is: $\binom{120}{9} \simeq 1.04 \times 10^{13}$. Among these combinations $\binom{20}{9} \simeq 1.6 \times 10^5$ will decode the FV. So, the expected number of polynomial evaluation to decode the FV $\simeq 1.04 \times 10^{13} \div 1.6 \times 10^5 \simeq 6.5 \times 10^7 \simeq 2^{26}$. For an 8-charachter password, the guessing entropy is between 18-30 bits [19]. So, when attackers try to guess the password, and then apply it to reduce the search space by filtering the chaff points, the total guessing entropy of this trial is between 44-56 bits which is higher than the 34-bits original entropy of the system. So, using a password to secure a part of the FV locking information, does not affect the FV security.

VI. Conclusions and recommendations

In this paper, a Fuzzy Vault (FV) system is designed based on the handwritten signature images and ESC features. A machine learning scheme is proposed to select good features from the signature images by employing the Dissimilarity Representation and Boosting Feature Selection approaches. The selected features are used to design classical signature verification systems and FV systems. The proof-of-concept computer simulations have shown reliability of the feature selection scheme, as the designed SV systems had good performance even for a small number of boosting iterations. For instance, using a multi-scale feature extraction of only 9 scales that results in a total of 1575 features, and applying only 200 boosting iterations, results in SV system with GAR $\simeq 88\%$ at FAR $\simeq 0.01\%$. Using only 20 features to encode WI-FVs results in GAR of 74% at FAR for random forgeries of 9%. Repeating the BFS process for the specific users to design WD-FVs, enhanced the quality of feature representation; For instance, the FAR of random forgeries reduced from 9% to 3%. Increasing number of feature extraction scales and boosting iterations, improves considerably the quality of selected features. Also, increasing number of FVs encoding features increases the matching points and the GAR accordingly.

The computer simulations have shown enhance security of the proposed FV system than the single-factor signature verification systems. As SV systems relies on the signature images only, so when signatures are compromised the FAR will be 100%. On the other hand, in the proposed system, it is shown that FAR is expected to be 0.1% when signatures are compromised.

From the cryptography perspective, the proposed FV system is more secure than the classical cryptography systems. In the classical systems, the password is the weakest link that if compromised the whole system will be compromised as it encrypts the secret key directly. In this case FAR will be 100%. On the other hand, in the proposed system, the password is used to secure the signature representation as if the signature is compromised it will not be much valuable as attackers do not know which features are taken from it. It is shown that the expected FAR is about 15% when the password is compromised.

Although a trade-off between recognition accuracy and system security is shown when comparing performance of the SV and FV systems, the proposed "less-accurate" system is secure against the security vulnerabilities that, if attacked a SV system, its accuracy will be meaningless. Enhancing the FV accuracy have to be more investigated by applying different methodologies for instance: i) enhance the feature level fusion by embedding higher number of ESC or other types of features at different scales, ii) apply decision level fusion by combine the decision of multiple FVs, iii) propose an intelligent quantization method that adapt the quantization size based on the feature variability.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada and BancTec Inc.

References

[1] A. K. Jain et al., Biometrics: A Tool for Information Security. *IEEE Transactions on Information Forensics and Security*, vol.1, no.2. pp.125-143, 2006.

[2] D. Impedovo and G. Pirlo. Automatic Signature Verification: The state of the Art. *IEEE transactions on Systems, Man, and cybernetics, Part C: Applications and Reviews* vol.38, no.5, pp.609- 635, September 2008.

[3] U. Uludag et al., Biometric Cryptosystems: Issues and Challenges. *Proceedings of IEEE*, vol 92, pp. 948–960, 2004.

[4] A. Juels, M. Sudan. A Fuzzy Vault Scheme. *In proc. IEEE int. Symp. Inf. Theory*, Switzerland, pp. 408, 2002.

[5] K. Nandakumar et al., Fingerprint Based Fuzzy Vault: Implementation and Performance. *IEEE transactions on information forensics and security*, vol.2, no.4, pp.744-757, December 2007.

[6] Y. Wang et al. Fuzzy Vault for Face Based Cryptographic Key Generation. *Biometrics Symposium, 2007*, pp.1-6, September 2007.

[7] T. Franssen. et al. Fuzzy Vault for 3D Face Recognition Systems. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008.

[8] V. S. Meenakshi, G. Padmavathi. Retina and Iris Based Multimodal Biometric Fuzzy Vault. *IJCSIS International Journal of CSIS*, vol.7, no.2, February 2010

[9] A. Kumar. et al. Development of a New Cryptographic Construct Using Palmprint-Based Fuzzy Vault. *EURASIP Journal on Advances in Signal Processing*, 2009

[10] Manuel Freire et al., Cryptographic key generation using handwritten signatures *In the proc of SPIE* , vol.6202, pp. 225–231, 2006.

[11] Manuel Freire et al., On the Applicability of Off-line Signatures to the Fuzzy Vault Construction. *ICDAR2007*, Brazil, Sept 2007.

[12] P, Pekalska, E., Duin, R.P.W. Dissimilarity representations allow for building good classiffiers *Pattern Recognition Letters* vol.23, pp.943-956, 2002.

[13] D.B. Redpath and K. Lebart. Boosting Feature Selection. *Pattern Recognition and Data Mining* , Lecture Notes in Computer Science, vol.3686, pp. 305-314, 2005

[14] K. Nandakumar et al. Hardening Fingerprint Fuzzy Vault using password. *Lecture Notes in Computer Science*, vol.4642, pp.927-937, 2007.

[15] C. Freitas, M. Morita, L. Oliveira, E. Justino, A. Yacoubi, E. Lethelier, F. Bortolozzi, and R. Sabourin. Bases de dados de cheques bancarios brasileiros. *XXVI Conferencia Latinoamericana de Informatica*, 2000.

[16] R. Sabourin, G. Genest. An Extended-Shadow-Code based Approach for Off-Line Signature Verification. *12th ICPR*, Jerusalem, pp. 450–453, October 1994.

[17] Bertolini, D. et al., Reducing Forgeries in Writer-Independent Off-Line Signature Verification through Ensemble of Classifiers. *Pattern Recognition*, vol.43, pp. 387-396, 2010.

[18] Schapire et al., The boosting approach to machine learning: An overview. *Workshop on Nonlinear Estimation and Classification*, 2002.

[19] W. E. Burr, D. F. Dodson and W. T. Polk Information Security: Electronic Authentication Guideline. *NIST Special Report*, 800-63, April 2006.