Biometric Folder Locking System using Fuzzy Vault for Face

Ketaki .N. Bhoyar
Department of Computer Engineering
Progressive Education Society's Modern College of Engg.,
Shivajinagar, Pune.

ABSTRACT

Locking folders effectively protects data from malicious programs, such as Viruses, Worms, and Trojans. Biometric template protection is a hot topic of research recently. Fuzzy vault is a recently developed construct which binds secret key (Password) and biometric information to provide template protection. In this paper we propose a biometric folder locking system using fuzzy vault for face. At the user's side, biometric template(face) and the password are accepted from the user. At the locking stage, fuzzy vault encoding is implemented by transforming the biometric template and the password. Then, at the unlocking stage, for the biometric template claimed by the user, corresponding fuzzy vault will be checked. If it has been changed then folder unlocking will be failed. Otherwise, by decoding the fuzzy vault the key will be obtained. If the key is same as the password entered by the user, then unlocking the folder will be successful. Otherwise it will be failed. The contribution of this paper include: Instead of using the face image directly for processing, real time video for face is accepted from user and characteristic feature points will be further used for processing. No database is being used for storage of the characteristic feature points.The Galois field algorithm is used for storing the transformed values in the fuzzy vault. This will introduce difficulty in reconstructing the original values from the stored ones. The experimental results shows that the proposed scheme provides promising results.

General Terms

Pattern Recognition, Security, Canny Edge Detection Algorithm, Finite Field Arithmetic Algorithm.

Keywords

Biometrics, Face template, Fuzzy vault, Secret key, Canny Edge.

1. INTRODUCTION

Locking folders is the best way to guarantee that nobody accidentally or intentionally gets access to your financial, health, private, and confidential information. Presently used password based systems have a number of associated inconveniences and problems such as user needs to remember passwords, passwords can be guessed or broken down via brute force and also there is problem of non-repudiation. Biometric recognition offers a reliable and natural solution to the problem in password management systems. Current authentication systems based on physiological and behavioural characteristics of persons (known as biometrics), such as face, inherently provide solutions to many of these problems and may replace the authentication component of the traditional password based systems. But biometric suffers from problem of non-repeatability. Conventional biometric person authentication systems, however, simply store each user's template as-is on the system. If registered templates are not properly protected, the risk arises of template leakage to a third party and impersonation using biometric data restored from a template.

Fuzzy vault is a key binding biometric cryptosystem scheme proposed by Jules and Sudan. It binds a key with the biometric template and obtains the helper data for authentication. The template is hidden in the helper data. At the authentication stage, the query biometric is used for extracting the key. If it is same as the biometric template, the correct key can be recovered and the authentication will be successful. Fuzzy vault doesn't require ordered features; it was implemented for finger print. Recently, several face based fuzzy vault schemes were implemented[1].

In this paper a biometric folder locking system using fuzzy vault for face is proposed. At the user's side, biometric template(face) and the password are accepted from the user. At the locking stage, fuzzy vault encoding is implemented by transforming the biometric template and the password. Then, at the unlocking stage, for the biometric template claimed by the user, corresponding fuzzy vault will be checked. If it has been changed then folder unlocking will be failed. Otherwise, by decoding the fuzzy vault the key will be obtained. If the key is same as the password entered by the user, then unlocking the folder will be successful. Otherwise it will be failed.

2. RELATED WORK

Ferhaoui Chafia, Chitroub Salim, Benhammadi Farid proposed a method based on the storage of the true minutia but under encoded shape, by using hashing function such that SHA1. Giving birth thus to a new approach that uses the advantages of the 'Fuzzy commitment' to fill the weaknesses of 'Fuzzy Vault'[4].

Umut Uludag, Sharath Pankanti, Anil K. Jain presented the implementation of the fuzzy vault, operating on the fingerprint minutiae features. These features are defined as abrupt changes in the regular ridge structure on the fingertip, characterized by either ending or bifurcation of the ridges. Typically, they are represented as (x, y,z) triplets, denoting their row indices (x), column indices (y) and angle of the associated ridge, respectively. These features are used in the proposed fingerprint fuzzy vault system for locking and unlocking the fuzzy vault[3].

Haiping Lu, Karl Martin, Francis Bui, K. N. Plataniotis, Dimitris Hatzinakos developed a system that has a goal to reject, during the verification process, an unauthorized subject who does not possess the original face features used during enrollment. In contrast, a genuine subject with the correct face features will be accepted. More importantly, the verification process needs to be based solely on the helper data, without requiring direct access to the original face features[6].

Thomas Franssen & Xuebing Zhou described the implementation of the Fuzzy Vault based on the description of

A fuzzy vault scheme proposed by A. Juels and M.Sudan[11].Ae-Young Kim, Sang-Ho Lee proposed a fuzzy vault based on the eigenfaces. For this scheme, they use a feature vector, which is called an eigenface, from a face image. The eigenface is calculated by the principle component analysis method and is consist of many components[9].

3. THE PROPOSED FRAMEWORK

In this section, implementation for the folder locking system using fuzzy vault for face is presented. At the first real time video of the user's face is captured. Edge segmentation of that face is done using Canny Edge Detection Algorithm. A set of face features are extracted from the retrieved image. The extracted features are then quantized and mapped to binary representation for feature points matching. The produced binary features and the key entered by the user are bound using the fuzzy vault. The key will be correctly retrieved if the presented face features have substantial overlap with the enrolled ones. The details of the proposed methods are presented in this section.

Overview of the folder locking system as shown in figure 1 consists of two stages. The one is the folder locking stage and the other is the folder unlocking stage. The input for locking stage is the folder to be locked and the folder is initially secured with the help of windows security. The password entered by the user and the face features extracted from the real time video of the users face are encoded. The dummy points are added to the encoded data. These dummy points are nothing but the points except the main face features(viz two centre pupil points, nose tip point, center lip point etc) we have considered while locking. These dummy points are introduced just to confuse the hackers. Thus the combination of encoded data and dummy points are scrambled and stored in fuzzy vault, hence the folder is locked.

At the unlocking stage, only face is provided as the input. The face features are extracted and matched with the data stored in the fuzzy vault. If the points are matched then the user is valid and the password is retrieved and the folder gets unlocked. If the points are not matched the user becomes invalid and the folder remains locked. The detailed encoding and decoding of fuzzy vault is explained below.

3.1 Architectural Framework

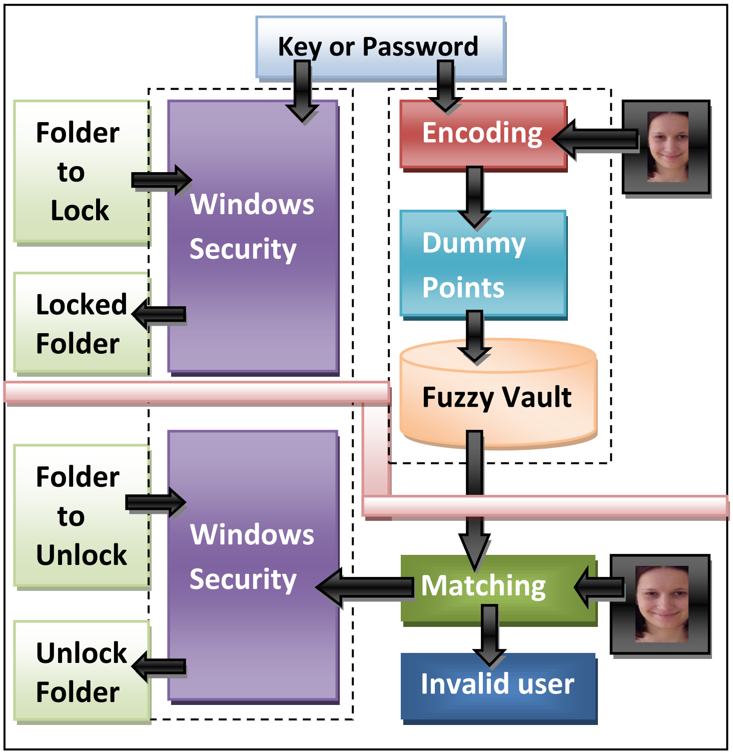

Figure 1. Overview of Folder Locking System

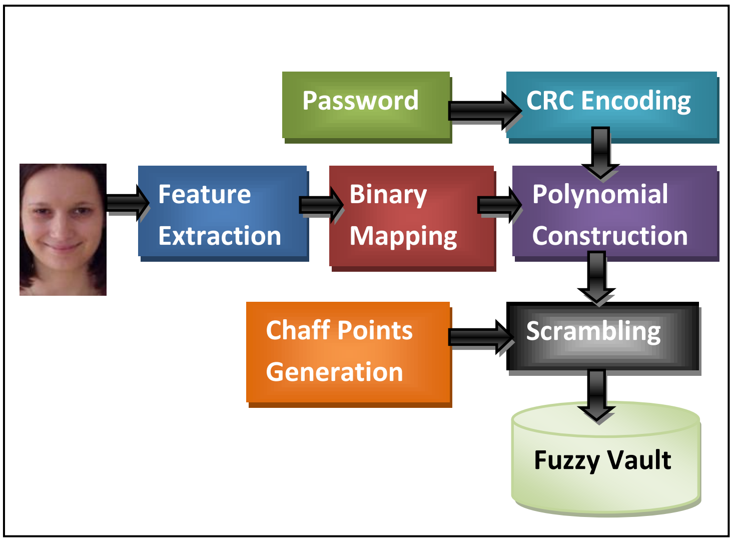

Figure 2. Fuzzy vault encoding

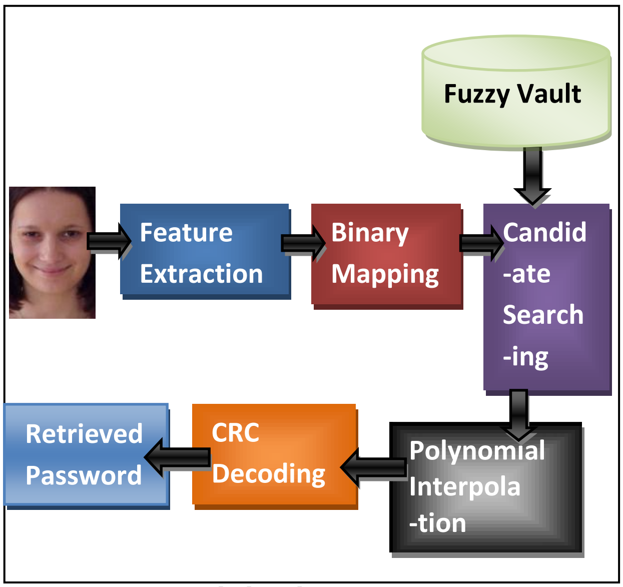

Figure 3. Fuzzy vault decoding

3.2 Encoding

Secret data needs to be protected. Suppose the secret data is S. But the size of S that can be protected feasibly is limited by the capacity of the entity used for locking and unlocking. Currently, *x* and *y* coordinates of face feature points are used for locking/unlocking the vault. Encoding operation is used to secure S in which face features and the textual password are mixed and stired in the fuzzy vault. While decoding the face features claimed by the user is matched with the values stored in the vault. If matched then S is retrieved. Hence the folder gets unlocked. Note that the vault is only responsible for securing S with face feature data and this is not the key in traditional password management system, rather, it has the role of a key for a new cryptographic construct, namely the fuzzy vault. In the current implementation, S is generated as a 128-bit random bit stream.

This current decoding implementation does not include any error-correction scheme, as proposed by Juels and Sudan's [4], since there are serious difficulties to achieve error-correction with biometric data. Developing the necessary polynomial reconstruction via error-correction has not been demonstrated in the literature. Instead, the algorithm proposed decodes many candidate secrets. To identify which one of these candidates is the actual secret, there is a need to put some structure into the secret S. By checking the validity of this structure during decoding, the algorithm can identify whether a given candidate secret is correct or not. Cyclic Redundancy Check (CRC) is a generalization of the simple parity bit checking. It is commonly used in communication channel applications for error detection where the errors are introduced due to channel noise. In this case using incorrect face feature points during decoding will cause an incorrect polynomial reconstruction, resulting in errors.

In the current implementation, a 16-bit CRC data is generated from the secret S. Hence, the chance of a random error being undetected (i.e., failing to identify an incorrect decoding) is 2^{-16} .The 16-bit primitive polynomial, $g_{CRC}\ (a) = a^{16} + a^{15} + a^2 + 1$, which is used for CRC generation is called "CRC-16" and is used for EBCDIC messages by IBM [12]. Appending the CRC bits to the original secret S (128-bits), 144-bit data SC is constructed. From this point on, all operations take place in Galois fields with cardinality 65536, namely GF (2^{16}) in which *x* and *y* coordinates of a feature (8-bits each) are concatenated as [*x* | *y*] to arrive at the 16-bit locking/unlocking data unit *u* . Note that to account for slight variations in feature data (due to nonlinear distortion), raw feature data are first quantized. Namely, each feature is translated to lie in a square tessellation of the 2D image plane.

SC is used to find the coefficients of the polynomial *p* : 144-bit SC can be represented as a polynomial with 9 (144/16) coefficients in GF(2^{16}), with degree *D*=8 . Hence, $p(u) = c_8u^8 + c_7u^7 + \cdots + c_1u + c_0$. Simply, SC is divided into non-overlapping 16-bit segments, and each segment is declared as a specific coefficient, $c_i, i = 0,1,2,3, \ldots, 8$. Note that this mapping method (from SC to c_i) should be known during decoding, where the inverse operation takes place: decoded coefficients (c_i^*) are mapped back to decoded secret SC^*. Then, two sets composed of point pairs need to be generated. The first one, called genuine set G, is found by evaluating *p*(*u*) on the face features (T). Starting with *N* features (if we have more than *N* features, we choose the first *N* sorted according to ascending *u* values), $u_1, u_2, \ldots, u_N$, we find G=$\{(u_1, p(u_1)), (u_2, p(u_2)), \ldots, (u_N, p(u_N))\}$. Note that the face features are selected to be unique, namely, $u_i \neq u_k$, if i≠k, i=1,2,…,N, k=1,2,…,N . The second set, called the chaff set C, determines the security of the system. Assuming there is a need to add *M* chaff points, first *M* unique random points are generated, $c_1, c_2, \ldots, c_M$ in the field GF(2^{16}), with the constraint that the do not overlap with $u_1, u_2, \ldots \ldots \ldots . u_N$, namely , $c_j \neq u_i, j = 1,2, \ldots \ldots M, i = 1,2, \ldots \ldots .., N$.Then, another set of *M* random points are generated, $d_1, d_2, \ldots ., d_M$, with the constraint that the pairs (c_j, d_j) ,*j=1,2,……,M* do not fall onto the polynomial *p*(*u*) . Chaff set C then becomes as $= \{(c_1, d_1), (c_2, d_2), \ldots \ldots ., (c_M, d_M)\}$, where d should be $d_j \neq p(c_j), j = 1,2, \ldots, M$. Union of these two sets, *G*∪*C*, is finally passed through a list scrambler which randomizes the list, with the aim of removing any stray information that can be used to separate chaff points from genuine points. This results in vault set VS = $\{(v_{1,}w_1), (\mathrm{v}_{2,}w_2), \ldots ., (v_{N+M,}w_{N+M})\}$ Along with *VS*, the polynomial degree *D* forms the final vault, *V*.

3.3 Decoding

Here, a user tries to unlock the vault *V* using the face features. Assuming that there are *N* (note that this number is the same as the number of genuine face features in order to balance the complexity conveyed via the number of required access attempts to reveal the secret) claimed face features (Q), $u_1^*, u_2^*, \ldots, u_N^*$, the points to be used in polynomial reconstruction are found by comparing $u_i^*, i = 1,2, \ldots, N$, with the values of the vault *V* , namely , $v_l, l = 1,2, \ldots, (M + N)$: if any $u_i^*, i = 1,2, \ldots, N$ is equal to v_l, *l*=1, 2,...,(M+N) , the corresponding vault point(v_l, w_l) is added to the list of points to be used. Assume that this list has *K* points, where *K*<=*N* . Now, for decoding a *D* -degree polynomial, (*D*+1) unique projections are necessary. Thus all possible combinations of (*D*+1) points are found out, among the list with size *K*. Hence, with *C*(*K*,*D*+1) combinations end. For each of these combinations, the Lagrange interpolating polynomial is constructed. For a given
specific combination, consider a set for the given a, $L = \{(v_1, w_1), (v_2, w_2), \ldots, (v_{D+1}, w_{D+1})\}$, the corresponding polynomial is

$$p^*(u) = \frac{(u - v_2)(u - v_3) \ldots (u - v_{D+1})}{(v_1 - v_2)(v_1 - v_3) \ldots (v_1 - v_{D+1})} w_1 +$$

$$\frac{(u-v_1)(u-v_3)\ldots(u-v_{D+1})}{(v_2-v_1)(v_2-v_3)\ldots(v_2-v_{D+1})}w_2+\cdots$$
$$+\frac{(u-v_1)(u-v_2)\ldots(u-v_D)}{(v_{D+1}-v_1)(v_{D+1}-v_2)\ldots(v_{D+1}-v_D)}w_{D+1}$$

This calculation is done in GF(2^{16}) and yields $p^*(u) = c_8^*u^8 + c_7^*u^7 + \cdots + c_1^*u + c_0^*$. The coefficients are mapped back to the decoded secret SC^*. For checking whether there are errors in this secret, the polynomial corresponding to SC^* are divided with the CRC primitive polynomial, $g_{CRC}(a) = a^{16} + a^{15} + a^2 + 1$. Due to the definition of CRC, if the remainder is not zero, it is certain that there are errors. If the remainder is zero, with very high probability, there are no errors. For the latter case, SC^* is segmented into 2 parts: the first 128-bits denote S^* while the remaining 16-bits are CRC data. Finally, the system outputs S^*. If the feature list (Q) overlaps with claimed feature list (T) in at least (D+1) points, for some combinations, the correct secret will be decoded, namely, S^* = S will be obtained. This denotes the desired outcome when claimed features and features are from the same face. Note that CRC is an error detection method, and it does not leak information that can be utilized by an imposter attacker (Bob). He cannot learn which one of the polynomial projections is wrong; hence he cannot separate genuine points from chaff points.

4. CONCLUSION

In this paper, a biometric folder locking system using fuzzy vault for face is proposed. In this scheme a new security system that combines the Windows file system security along with face based fuzzy vault. By combing the biometric face features and the password entered by the user at the time of locking is stored in the fuzzy vault. System will take care of password management by accepting only face while unlocking.

Thus, a new security tool will be created that lets you lock/unlock your folders with your personal password and Face using fuzzy vault for Face.

5. ACKNOWLEDGEMENT

A large measure of any credit for the “Biometric Folder Locking System using Fuzzy Vault for Face” must go to Prof. Dr. S. R. Patil, HOD of Computer Department, Bharti Vidyapeeth’s College of Engineering for Women,Pune, who with the author has assisted in the preparation of this paper. We admire his infinite patience and understanding that he guided us in field we had no previous experience.We are grateful to him for having faith in us.

6. REFERENCES

[1] Lifang Wua,b Songlong Yuana, “A face based fuzzy vault scheme for secure online authentication”, Second International Symposium on Data, Privacy, and E-Commerce,2010.

[2] LI Fen, LIU Quan, PANG Liaojun, PEI Qingqi“Identity Authentication Based on Fuzzy Vault and Digital Certificate”, 2010.

[3] Umut Uludag, Sharath Pankanti, Anil K. Jain,“Fuzzy Vault for Fingerprints” ,2010.

[4] Ferhaoui Chafia, Chitroub Salim, Benhammadi Farid , “A Biometric Crypto-system for Authentication”,2010.

[5] Youn Joo Lee, Kang Ryoung Park, Sung Joo Lee, Kwanghyuk Bae, and Jaihie Kim , “A New Method for Generating an Invariant Iris Private Key Based on the Fuzzy Vault System”, 5th oct 2008.

[6] Haiping Lu, Karl Martin, Francis Bui, K. N. Plataniotis, Dimitris Hatzinakos“Face recognition with biometric encryption for privacy-enhancing” , 2008.

[7] Thomas Franssen & Xuebing Zhou,“Fuzzy Vault for 3D Face Recognition Systems”,2008.

[8] Yongjin Wang, KN. Plataniotis, “ Fuzzy Vault for face based cryptographic key generation”, The Edward S. Rogers Sr. Department of Electrical and Computer Engineering,University of Toronto, 10 King's College Road, Toronto, ON, Canada, M5S 3G4,2007

[9] Ae-Young Kim, Sang-Ho Lee“Authentication Protocol using Fuzzy Eigenface Vault”, Feb 12th 2007.

[10] U. Uludag, S. Pankanti, and A. K. Jain, “Fuzzy Vault for Fingerprints”, July 2005.

[11] G. A. Juels and M. Sudan, “A Fuzzy Vault Scheme,” Proceedings of IEEE International Symposium on Information Theory, 2002.