ENSURING PRIVACY OF BIOMETRIC FACTORS IN MULTI-FACTOR AUTHENTICATION SYSTEMS

Kikelomo Maria Apampa, Tian Zhang, Gary B. Wills and David Argles
School of Electronics and Computer Science, University of Southampton, U.K.

Keywords: Biometric privacy, Multi-factor authentication, Raw biometric.

Abstract: One of the inherent properties of biometrics is the ability to use unique features for identification and verification of users. The usable biometric features in humans are limited in number and they must be kept secret; if a biometric factor is compromised it presents a challenge that may defy solution. In this paper we present a novel method to preserve privacy of users' biometrics. Using an elastic matching algorithm, we produce a digest that can be substituted for the raw biometric factor. This will ensure that the users' biometric data is never exposed during the authentication phase.

1 INTRODUCTION

Identification and authentication are security requirements that have steadily become more important in private and public sectors. Governments, military, financial communities, medical industries, etc. continually seek effective methods to identify and authenticate their users. Traditionally, username-passwords have been employed in almost all access-control systems; however this method has proved unsatisfactory especially when users insist on very short and easy passwords to memorise (Argles *et al*, 2007). The difficulty of remembering passwords arises from the amount of entropy in the passwords. By allowing passwords with less entropy, we are creating weak passwords that would be easier for the attacker to guess. The migration from single-authentication to dual-factor authentication which provides stronger and effective security schemes is well documented (Jain *et al*, 2000; Bolle *et al*, 2004). Biometric technology is a potential approach to authentication which will create more secure systems since biometric data is unchangeable and not forgettable. A biometric authentication system generally consists of two stages (see figure 1). During the enrolment phase a users' biometric image is acquired, a biometric template is created and the templates are stored in a database or on a portable storage device like a smartcard (Davida *et al*, 1998). During the authentication phase, the user presents a biometric sample which is compared with the stored template. The user is successfully authenticated if there is a near match between the input and the stored template. In this paper biometric raw data refers to the unmodified image of a fingerprint which is extracted and stored on the biometric server. The template data refers to the stored features which were extracted from the fingerprint image; they contain information necessary for comparison. Ironically one of the greatest benefits of biometric factors also poses a challenge i.e., they are unchangeable and easily forgeable. According to Ratha *et al*, (2007) *"if a biometric identifier is compromised it is lost forever and possibly for every application where the biometric is used"*. This is particularly significant as once a biometric factor is exposed it loses its value as a factor and a user may not immediately be aware that their biometric has been compromised. A common concern in biometric security is the privacy issues derived from storage and misuses of the template data (Jain *et al*, 2007).

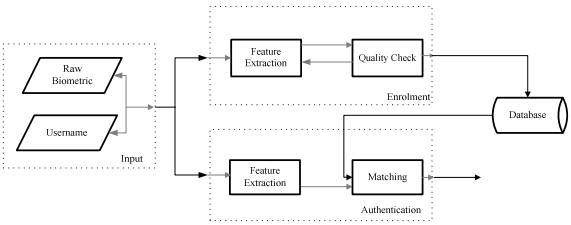

Figure 1: Biometric Architecture (Jain & Panakanti, 2000).

Maria Apampa K., Zhang T., B. Wills G. and Argles D. (2008).
ENSURING PRIVACY OF BIOMETRIC FACTORS IN MULTI-FACTOR AUTHENTICATION SYSTEMS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 44-49
DOI: 10.5220/0001921600440049

2 RELATED WORK

One potential means of safe-guarding stored templates is encryption. In a review article, Jain *et al*, (2007) suggest that multiple acquisitions of the same biometric trait will not yield the same feature set and as a result biometric templates cannot be stored in an encrypted form. Furthermore, the biometric templates would need to be decrypted prior to matching; therefore they will be inevitably exposed to potential hacker attacks (Braithwaite *et al,* 2002). Ratha *et al*, (2001) proposed the concept of cancellable transforms to overcome the problems of compromised biometric templates. The technique introduced unique distortions of raw biometric data such that instead of storing the original biometric it is transformed using a one-way function; the transformed biometric and transformation are stored. In their proposal they conclude that transforms are noninvertible therefore it is computationally hard to recover the original biometric identifier from a transformed version thus preserving privacy. Braithwaite *et al*, (2002) argues that it is necessary in some cases to reverse the transformation prior to matching which would expose the raw biometric data and make it susceptible to hacking. To eliminate the need to revert the templates to a non-transformed state during the authentication, Braithwaite *et al*, (2002) propose the use of application-specific biometric templates. In this approach the biometric template assumes a new format that is unique for each application and the transformations are such that the matching can be performed on the transformed templates. Argles *et al*, (2007) consider a similar problem of ensuring privacy of the users' biometric even if the biometric database server is compromised. They suggest a split and merge technique which is a hybrid scheme incorporating an electronic token and biometric verification. In this method the encrypted biometric template and user key is split during storage. One half of the encrypted template is stored on an electronic media and the other is retained inside the secure biometric database. Storing the encrypted data in two separate locations makes it difficult for an intruder to compromise the system. Without the decryption key the attacker will first be required to break the encryption algorithm. Once the key generator is exposed the information leakage becomes problematic, reducing the difficulty of guessing the template by half.

Other approaches which address the issue of ensuring privacy of biometric templates include the use of steganography (Jain & Uludag, 2003) and the secure sketch scheme (Sutcu *et al*, 2007).

3 ANALYSIS OF SPLIT AND MERGE TECHNIQUE

The split and merge technique attempts to ensure privacy of the biometric factor by splitting the factor into multiple components (Argles *et al*, 2007). The system uses a biometric (fingerprint) and physical (USB drive) factor; where the removable storage device is used to secure a user-selectable password (user key). In figure 2 and figure 3 the enrolment and matching processes of the method is shown. To analyse the split and merge system we shall assume that key generation, splitting, merging, encryption and decryption functions have the following properties:

Assumption 1: The key generation function is a good pseudorandom function with a large period - without knowing the seed, we cannot deduce the next outcome of the generator irrespective of how many previous outcomes we have collected

Assumption 2: The splitting function $S : x \mapsto (a \in A, b \in B)$ splits an input x into two components containing equal amounts of information: $|A| = |B| \Leftrightarrow i(a) = i(b)$

Assumption 3: The encryption function is Shannon secure (Shannon, 1951) and leaks no information. For a cryptosystem: $\{E, D, k, m, c\}, H(m) = H(m \mid c)$

These simplifications are made so we can analyse the system independently of any weaknesses that maybe inherited from these functions in implementation.

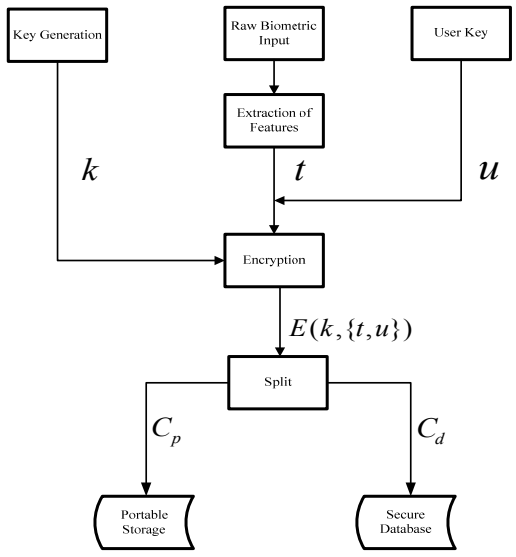

Figure 2: Enrolment using the split and merge method.

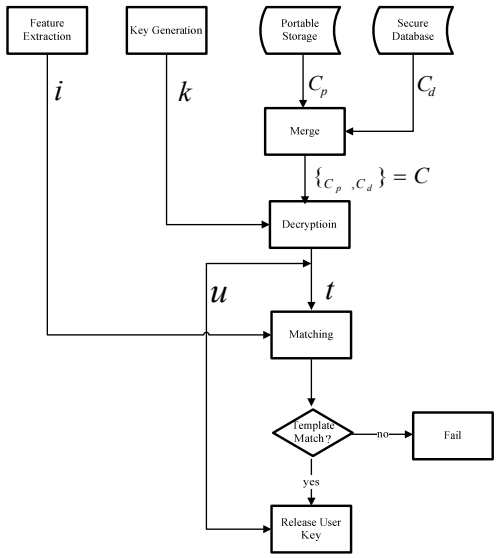

Figure 3: Authentication using the split and merge method.

3.1 Security Dependant on Obscurity

In addressing the shortcomings in the design of the split and merge technique, we shall consider a scenario where an attacker has acquired the portable storage device and attempts to recover the user key from the authentication system. (The key generation function must be deterministic; else the system will be unable to recover the key). We shall assume the attacker has access to the key generator and the biometric database. Thus, from figure 3 we make the following observations:

- It is possible to partition the system to only require c_p and c_d to recover the biometrics.
- The attacker can derive the biometrics of the user by acquiring the storage device and having access to the authentication system; it is possible to recover the fingerprint template of the user.

For an authentication session consisting of $\{i, k, c_p, c_d\}$ (see figure 3), we define the following operations:

Merge as M: $(c_p, c_d) \mapsto c$

Decrypt as D: $(k, c) \mapsto (t, u)$

Compares as C: $(i, t) \mapsto \{true, false\}$ (3.1.1)

Then figure 3 is summarised as:

$$(t, u) = D(k, M(c_p, c_d)) \quad (3.1.2)$$

Since the splitting and merging functions must be bijective, they must also be deterministic; extracting t or u from (t,u) should always be possible. Deriving (t,u) in 3.1.2 does not employ the capabilities of the compare function, thus the biometric factor is not used. Exposing the biometric template of the user is a bigger problem than exposing a single biometric input of a single scan. The template is often a better true representation of the biometric feature than an average scan by definition. It is important for the security of the spilt and merge system to keep the key generator private.

3.2 Exposed Biometrics

During authentication the user's biometric data is briefly exposed to allow the matching of the input biometric and the template. The matching phase cannot be performed on the client as it requires both the input biometric and the template, thus it needs to be performed on the server. As an example, a disgruntled employee with access to the server could recover the complete biometric template by compromising the privacy of the matching component. This action defeats the purpose to protect the user biometrics in the event that the server is compromised.

4 PROPERTIES OF BIOMETRICS

Existing fingerprint algorithms do not attempt to directly extract a unique invariant representation of the fingerprints (Jain *et al*, 2000). In practice these would require perfect equipment and conditions; instead they use an approximation of the unique invariant representation (template biometric). The biometric factor is then compared to the template which can be either accepted or rejected depending on the amount of work required to transform one into the other. The transformations may be complex (Ma *et al*, 2004) and the end comparison is the result of a number of probabilistic and heuristic operations. Due to this, the matching will always have a non-zero probability of false acceptances and rejections. Common biometrics such as fingerprints provide a high degree of reliability when identifying a user; however they can also be forged with varying degrees of success. For this reason, fingerprints alone are insufficient for authentication. Biometric factors are more suited as identification factors due to being forgeable and immutable. Thus, the use of biometrics as an authentication factor relies on the

ability to keep the biometrics secret. Exposing the users' biometrics may have severe consequences; a user will have only one set of fingerprints as opposed to having different passwords for access.

We shall define failure of an authentication system by:

- Positive failure: occurs when the system incorrectly supports an identity
- Negative failure: occurs when the system fails to support a correct identity

5 COMPONENTS OF THE NEW SYSTEM

The proposed system aims to achieve strong authentication by the use of a physical and a biometric factor. We assign equal importance to minimising positive failure and keeping biometrics private, since every time the biometrics are exposed, the task of minimising positive failure becomes more difficult. By drawing on the characteristic strengths of the different factor types, we aim to build a system that is less likely to fail positively by means of forgery. The system will still fail negatively should the user forget or lose the physical factor; unfortunately this must be the case as only biometric factors are guaranteed to be always available. The user presents a biometric factor (for identification) and a physical factor (for verification) during the authentication process. The biometric factor consists of two components; i.e., the biometric reader which in implementation could be a standard "off the shelf" biometric device and the transformer could be readily implemented in software on the client. The physical factor also consists of a security token and a small storage device. In implementation the small storage device could be a smartcard or a modified USB storage key.

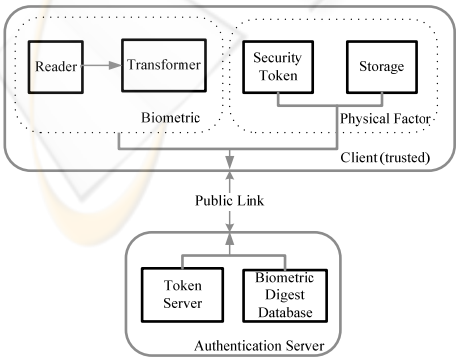

Figure 4: Components of the new system.

6 OBFUSCATION OF BIOMETRICS

During authentication the server must perform an operation of the following form (verification of an identity):

$$C(i,t) \rightarrow \{true, false\} \tag{6.1}$$

In current biometric systems the two inputs (input and template) are both elements from the same space and the function is essentially a piece wise function based on the matching distance function (m)

$$C_b(i,t) = \begin{cases} true\, m(i,t) \le k \\ false\, m(i,t) > k \end{cases} \tag{6.2}$$

In ArgleCs *et al*, (2007) the biometric system was obfuscated by splitting, however the authentication function C remains the same as (6.2). The disadvantage of requiring i and t to be elements from the space is that security of the system depends on the ability to perform C in private. If the privacy of C is not guaranteed the attacker could cause positive failure by acquiring either i or t.

7 HASH FUNCTIONS

The solution for the username/password systems is to use a hash function (or one-way function) resulting in the following authentication function:

$$C_h(i,t) = \begin{cases} true\, h(i) = t \\ false\, h(i) \neq t \end{cases} \tag{7.1}$$

The properties of (7.1) would be ideal in protection of user biometrics as the security of the system will not depend on the privacy of the biometric database. Therefore a biometric data can be hashed and stored on the server. Hashing may seem an appropriate solution for biometrics; however the problem arises when matching an incoming biometric against the stored hashed template. A biometric data will produce a close match and not an exact match. The inability to match input template with the stored template will lead to unacceptably high false rejection rates.

8 SOLUTIONS USING HEURISTICS

An approach for producing a biometric digest is using elastic matching algorithms in place of hash functions. The matching algorithms are ideal candidates since they are already of the form $m:(i,j)\rightarrow D\subset \mathbb{Z}$. We can adapt the matching algorithm to fit the form required by generating an arbitrary biometric input (K) and then comparing the input with the generated input: $d(j)=m(j,K)$ In effect we require an algorithm that can identify close matches i.e. elastic matching algorithm. We define the authentication function as follows:

$$C_i(d(i),d(t))=\begin{cases} true \mid d(i)-d(t)\le f \\ false \mid d(i)-d(t)>f \end{cases} \quad (8.1)$$

Note that the authentication function's range does not expose the input biometric or the template it is not possible to obtain i from $d(i)$.

8.1 Enrolment

Figure 6 depicts the sequence diagram for the enrolment process.

1. The reader acquires the user's raw biometrics (B_u)
2. The authentication server sends a server ID that is unique to the system
3. The transformer (T) generates an arbitrary template (S) from the server ID
4. The transformer then produces a representation of the raw biometric with respect to the arbitrary template. I.e. T (S, B_u) = O. where, O is the origin and T has the properties of an elastic matching function.
5. The new representation of the raw biometric is sent to the server for storage
6. The origin (generated template) is then stored on the physical storage device.
7. User password is acquired and encrypted. The password is stored on the token and sent to the server for storage.

8.2 Authentication

Figure 7 shows the authentication process as a sequence diagram. The Diffe-Hellman (DifHel76) exponent is used to establish an encrypted conversation between the client and server.

1. The client produces the origin (generated template) stored on the token
2. The client gets raw biometrics (B_u^*) from reader and produces a digest. i.e. T (O, B_u^*) = S^*
3. The client requests a one-time password from the token. The client encapsulates the digest (S^*) and one-time passwords and sends the package to the server
4. The server decrypts the package and extracts the digest (S^*) and one-time password.
5. The server queries the digest database for the most likely match
6. The servers checks the current password against the stored password for the most likely match
7. If the passwords match the use is successfully authenticated, else authentication fails.

9 CONCLUSIONS

In this paper we have shown that the resilience of a multifactor authentication system could be improved by combining the factors to preserve the privacy of the user biometric. A novel approach is presented in constructing a digest from the biometric and physical factors. The digest is used in place of the raw biometric in authentication; therefore the raw biometric is never exposed which minimises the risk of exposure. An elastic matching algorithm was used for producing the digest. One of the benefits of using the digest is its ability for trivial sorting and indexing; thus making the system scalable. Further work will be to examine the suitability of different matching algorithms for constructing the digest.

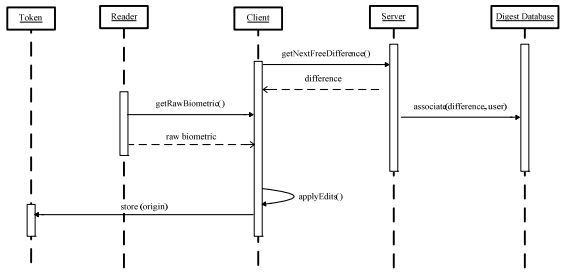

Figure 6: Enrolment in the proposed system.

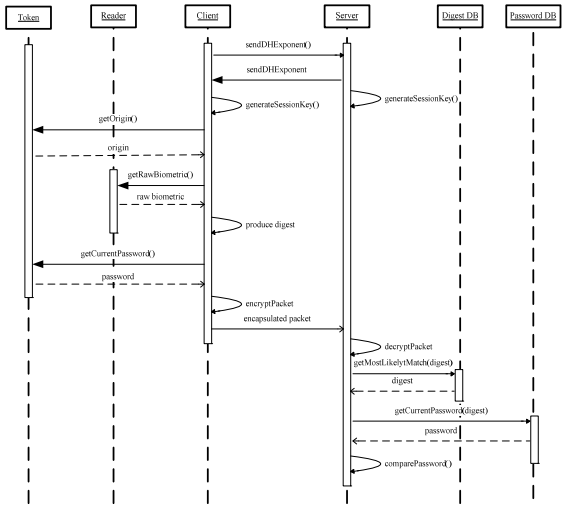

Figure 7: Authentication in the proposed system.

REFERENCES

Argles, D., A. Pease, R. Walters (2007). An Improved Approach to Secure Authentication and Signing. Advanced Information Networking and Applications Workshops 2007. AINAW '07.

Bolle, R., J. Connell, S. Pankanti, N. Ratha, A. Senior (2004). Guide to Biometrics. New York, Springer Professional Computing.

Braithwaite, M., C. vonSeelen, J.Cambier, J. Daugman, R. Glass, R. Moore, I. Scott (2002). Application-specific biometrics templates. Proceedings IEEE Workshop on Automatic Identification Advanced Technologies, Tarrytown NY.

Davida, G. I., Y. Frankel, B. J. Matt (1998). On enabling secure applications through off-line biometric identification. Proceedings IEEE Symposium on Security and Privacy, Oakland California U.S.A, IEEE Computer Society.

Diffie, W. and M. E. Hellman (1976). "New Directions in Cryptography." IEEE Transactions on Information Theory IT-22: 644-654.

Jain, A. and S. Pankanti (2000). Fingerprint Classification and Matching, Academic Press.

Jain, A. K., K. Nandakumar, A. Nagar (2007). Biometric Template Security. EURASIP Journal on Advances in Signal Processing, Hindawi Publishing Corporation.

Jain, A. K. and U. Uludag (2003). "Hiding biometric data." IEEE Transactions on Pattern Aanalysis and Machine Intelligence 25(11): 1493-1498.

Ma, H.-M. and R.-K. Huang (2004). A new fingerprint translation finding algorithm. Proceedings of the 3rd International Conference on Machine Learning and Cybernetics.

Ratha, N. K., J. H. Connell, R. Bolle (2001). "Enhancing security and privacy in biometrics-based authentication systems." IBM systems Journal 40(3): 614-634

Ratha, N. K., S. Chikkerur, J. Connell, R. Bolle (2007). "Generating Cancelable Fingerprint Templates." IEEE Transactions on Pattern Analysis and Machine Intelligence 29(4): 561-572.

Shannon, C. (1951). "Prediction and Entropy of Printed English." The Bell Systems Technical Journal 30: 50-64

Sutcu, Y., Q. Li, N. Memon (2007). How to protect biometric templates. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference on Security, Steganography and Watermarking of Multimedia Contents, San Jose CA