A BIOMETRIC-BASED SCHEME FOR ENHANCING SECURITY OF CRYPTOGRAPHIC KEYS

M.Y. Siyal and Fawad Ahmed
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
Email: {eyakoob, pka534086}@ntu.edu.sg

ABSTRACT

In public key cryptography, the security of private keys is of vital importance. If a private key is ever compromised, it can be used to sign forge documents or to decrypt secret messages. Conventional methods such as password-based encryption that are used for safe custody of private keys do not provide adequate security due to very low entropy in user chosen passwords. In order to enhance the security of private keys, we propose a novel biometric-based method that dynamically regenerates the private key of a user rather than storing it directly in an encrypted form. Our proposed algorithm is capable of regenerating key lengths that can meet the current security requirements of any public key algorithm and is more secure than conventional methods of protecting private keys using password-based encryption.

1. INTRODUCTION

In public key cryptography two keys are used; a private key and a public key. The public key is normally used for encrypting messages and is publicly known, while the private key is used for decrypting messages and is kept secret. However, for generating a digital signature, the private key is used to sign a message that can be verified by the corresponding public key [1]. Therefore, in a public key cryptosystem, security of the private key is very important. If the private key is ever compromised, extensive damage may be caused to the user. The key can be used to sign forge documents or to decrypt secret messages. Because of the large size of a cryptographically strong key, it is not possible for a user to remember the private key and enter each time it is required for a cryptographic application. Instead, the private key is usually stored encrypted with a user chosen password. A common problem with password-based encryption is the low entropy in user chosen passwords that can be exploited to launch password-guessing attacks [2]. Moreover, using a low-entropy password as a key to a strong cryptographic algorithm can transform it into a weak one.

From the past few years, researchers are attempting to use biometrics to enhance the security of cryptographic keys. A biometric is a unique physical or behavioral characteristic of an individual that can be used for identification purpose. Physical characteristics include fingerprints, hand geometry, retina, iris and facial characteristics, etc. Behavioral characteristics include signature, voice, keystroke patterns and gait, etc. [3]. The notion of using biometric template directly as a cryptographic key was first proposed by Bodo [4]. In this method, the data derived from the biometrics is used directly as a cryptographic key. As pointed out in [4], this method suffers from two main limitations. First, since a biometric template is not consistent due to environmental and physiological factors, this may not be able to generate a constant value of the key every time. Secondly, if the key is ever compromised, then that particular biometric may not be used again. Another innovative idea of generating a cryptographic key from voice characteristics of a user is proposed by Monrose et al. [5]. Using this technique, a 46-bit key can be generated from a roughly two second spoken password. The idea of using on-line handwritten signatures for private key generation has recently been proposed in [6], which can be used to generate a 160-bit key. In another method that uses fingerprints [4], the key is linked with the biometric at the time of enrolment to form enrolment data that gives no information of the key or the biometric. In the verification stage, the key is then reconstructed using the enrolment data and the biometric. The authors illustrates an example of generating a 128-bit key that can be used for cryptographic applications.

All the methods discussed above have a limitation of key length. For example, the largest key size that can be produced is 160-bit with the technique presented by Feng and Wah [6]. However, in public key algorithms like the RSA, key lengths of 1024-bit or higher are required [7], keeping in view the current computational power. To address this issue, we propose a novel biometric-based technique in this paper that dynamically regenerates the private key of a user by using his password, fingerprint and information stored in a smart card. One of the primary advantage of our scheme is that it does not require a user to generate any new key pair, as the basic purpose of our scheme is to regenerate the existing private key of a user. In case the user does not possess a public/private key pair, it can be generated by the corresponding public key algorithm

0-7803-8560-8/04/$20.00©2004 IEEE

that is used by the cryptographic application. It is important to note that our method uses smart card for storing only a part of the enrolment information that is required for regenerating the private key and not the private key itself. Storing a private key on a smart card does not provide full proof security as smart cards offer some level of tamper-resistance but are not tamper-proof [8].

In the next section, we present the details of our proposed key regeneration algorithm. In Section 3, we analyze the security of our scheme against on-line and off-line attacks. Section 4 illustrates how keys of different lengths can be regenerated. In Section 5, we present the experimental results on several fingerprint images. This is followed by conclusion in Section 6.

2. OUR METHOD

The proposed key regeneration algorithm uses the orientation field of a ridge map of a fingerprint as a biometric feature for key regeneration. This feature is designated as the *ridge angle vector*, $\boldsymbol{R_a}$ and is calculated using the algorithm proposed in [9]. The orientation field of an original fingerprint image does not give a very accurate measurement of the ridge flow because of the presence of noise and broken ridge lines, as shown in Fig.1 and 2. In order to get a reasonably accurate and consistent estimation of the ridge flow pattern, we propose the use of the orientation field of the ridge map of a fingerprint image. The results are shown in Fig.3 and 4. As compared to Fig. 2, the orientation pattern in Fig. 4 follows the ridges very smoothly. This makes the use of the $\boldsymbol{R_a}$ feature highly desirable for the purpose of key regeneration. The registration of fingerprint images is carried out using the minutia points obtained from the fingerprint image [10]. The proposed key regeneration algorithm consists of an *Enrolment Phase* and *Key Regeneration Phase*. We denote a legal user by U_a, his fingerprint by f_a, the minutia points extracted by $\boldsymbol{M_a}$, password by pwd_a, the smart card by $Scard_a$ and the existing private key by Kp_a.

2.1. Enrolment phase

In this phase, U_a submits f_a, pwd_a, $Scard_a$ and Kp_a to the system. The system then does the following:

1. Compute $\boldsymbol{M_a}$ and $\boldsymbol{R_a}$ from f_a, where $\boldsymbol{R_a} = \{\theta_m : 1 \le \theta_m \le 180$ degrees and $1 \le m \le 180\}$. Store $\boldsymbol{M_a}$ in $Fmin_a$.

2. For each $\theta_m \in \boldsymbol{R_a}$, define θ_{Sk} given by Eq.1, where $S = m = 180$ and form groups $G_i (1 \le i \le 36)$ comprising of five consecutive θ_{Sk}. The variable r is the system parameter defining the number of shadow angles. The purpose of shadow angles is to make the system tolerant to errors in the fingerprint pattern at the time of key regeneration.

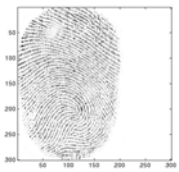

Fig.1 Original image

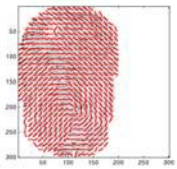

Fig.2 Orientation field of the original image

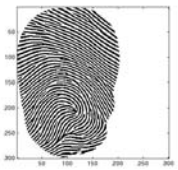

Fig. 3 Ridge map of the original image

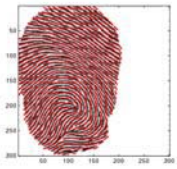

Fig.4 Orientation field of the ridge map

$$\theta_{Sk} = \{\theta_m, \theta_m \pm t\}_{1 \le t \le r} \ , 1 \le k \le (2r+1) \qquad (1)$$

3. Define a set P containing the P_i segments of Kp_a i.e. $P = \{P_1, P_2, \ldots\ldots\ldots, P_{36}\}$.

4. Each consecutive $P_i \in P$ is treated as a secret and an (m-n) secret sharing scheme [11] is used to generate the secret shares, where m is the minimum number of shares required to retrieve the secret and n is the total number of shares. For our experiment, m=2 and n=5 x (2r+1).

5. Compute random, collision free hashes H^*_{Sk} using Eq.2 and 3. The function H_{SHA} in Eq.2 uses SHA-1 algorithm [1] to give 160-bit random and unique hashes, ζ is a public value selected arbitrary for every S. Eq.3 maps the 160-bit hashes generated by Eq.2 into smaller collision free hashes, which are used to index the secret shares. The parameter σ defines the maximum range of H^*_{Sk} and δ is a random seed value chosen for every S in order to avoid any collision in the values of H^*_{Sk}.

$$H_{Sk} = H_{SHA}\left(\theta_{Sk} + H_{SHA}\left(pwd_a + \zeta\right)\right) \qquad (2)$$

$$H^*_{Sk} = f_0(H_{Sk}, \delta, \sigma) \qquad (3)$$

6. For each segment of Kp_a, store the secret shares generated in Step 4 in $RandomSecret_a$ by using the corresponding index pointed by H^*_{Sk}. For example for the n shares of P_1, the first 5 x (2r+1) H^*_{Sk} values will be used where, $1 \le S \le 5$ and $1 \le k \le (2r+1)$.

7. Encrypt δ values with pwd_a and store them in $Scard_a$. Encrypt $Fmin_a$ and $RandomSecret_a$ with pwd_a to get $EFmin_a$, $ERandomSecret_a$ respectively and store them in any medium like the hard disk, etc. We call $EFmin_a$ and $ERandomSecret_a$ as the registry files for user U_a.

2.2. Key regeneration phase

In this phase, the system will regenerate the private key of a legal user if and only if the fingerprint, password and smart card are same as in the enrolment phase. The working of this phase is as follows: U_a submits f_a, pwd_a and $Scard_a$ to the system. The system reads the corresponding registry files $EFmin_a$, $ERandomSecret_a$ and does the following:

1. Decrypt $EFmin_a$ using pwd_a to get $\boldsymbol{M_a}$ and use a minutia alignment algorithm [10] to align f_a at a position presented in the enrolment phase.

2. Compute $\boldsymbol{R_a}$ from f_a, where $\boldsymbol{R_a} = \{\theta_m : 1 \le \theta_m \le 180$ degrees and $1 \le m \le 180\}$. For each $\theta_m \in \boldsymbol{R_a}$, define θ_{Vk} given by Eq.4, where $V = m = 180$ and form groups G_i $(1 \le i \le 36)$ comprising of five consecutive θ_{Vk}.

$$\theta_{Vk} = \{\theta_m - 1, \theta_m, \theta_m + 1\}, \quad 1 \le k \le 3 \tag{4}$$

3. Read the encrypted δ values from $Scard_a$, decrypt them with pwd_a, and use Eq.5 and 6 to calculate H^*_{Vk}

$$H_{Vk} = H_{SHA}\left(\theta_{Vk} + H_{SHA}\left(pwd_a + \zeta\right)\right) \tag{5}$$

$$H^*_{Vk} = f_0(H_{Vk}, \delta, \sigma) \tag{6}$$

4. Decrypt $ERandomSecret_a$ with pwd_a and read the secret shares based on the index pointed by H^*_{Vk}. Compute the secrets i.e. the segments of Kp_a using polynomial interpolation. Each θ_{Vk} will enable three secrets to be calculated. If any two secrets calculated within a θ_{Vk} matches and if within a group any two such θ_{Vk} are found giving the same value of the secret; the secret i.e. the segment of the private key will be considered as valid. Each segment will only be valid if $H_{Vk} \in H_{Sk}$ (*for* V=S) and $H^*_{Vk} \in H^*_{Sk}$ (*for* V=S). The concatenation of the segments P_1 to P_{36} will form the private key of U_a. Once the private key is regenerated, it can be used with the corresponding public key algorithm for cryptographic applications.

3. SECURITY ANALYSIS

We describe two types of attack against our scheme; *on-line attack* and *off-line attack.* In an on-line attack, the attacker has to produce the legitimate user password, fingerprint and smart card to dynamically regenerate the private key. As compared to conventional password-based schemes, our proposed scheme offers more security in an on-line attack since in addition to the knowledge of password; it also requires a valid fingerprint and smart card for successful regeneration of the private key. In an off-line attack, we assume that the attacker captures the login registry files and then use them in an off-line effort to regenerate the private key. We first analyze the security of our scheme against an off-line attack on the files $EFmin_a$ and $ERandomSecret_a$ that stores the minutia points and the secret shares of a legitimate user U_a. To make things more favorable for the attacker, we further assume that the attacker has some how been able to break the password and is able to get the decrypted files $Fmin_a$ and $RandomSecret_a$. The file $Fmin_a$ only contains the minutia points of U_a, therefore, it will not reveal any information regarding Kp_a. The next attack that can be launched is to use $RandomSecret_a$ in order to construct the private key. Since in our implementation of the (m-n) threshold scheme, m=2, therefore, the attacker has to pick two valid shares from the total shares stored in $RandomSecret_a$ to construct a single segment of the private key. However, in order to check the validity of the secret, the attacker will require at least three valid shares. If each pair of share gives the same secret, it will be confirmed that the secret is valid. In this way, it is possible for the attacker to construct all the 36 secrets. However, in order to produce the correct key, the attacker has to concatenate the secrets in the correct order. For this, the attacker will have to perform at most 36! permutations, i.e. 3.7×10^{41} number of iterations and in each iteration check for the key validity. This is a considerable high overhead as compared to the case where a private key is simply stored by password-based encryption. A brute force attack on a simple password based encryption scheme will require the attacker to try approximately 2×10^{14} different combinations of English characters and digits for an eight character password. Even if a sixteen-character password is chosen, the total number of possibilities will roughly be 5×10^{28} that is still very low as compared to 3.7×10^{41}. In addition, since the file $RandomSecret_a$ contains secret shares, it will not reveal any information of the ridge angle features i.e. the biometric used for the key regeneration process. Similarly, an attack on the user's smart card will not reveal any direct information about the key or the biometric as the smart card only contains random seed values that are used at run time along with fingerprint, password and registry files for key regeneration.

4. KEY LENGTH

Using our algorithm, keys of any length can be regenerated through Step 3 and 4 in the enrolment phase. No matter how long is the key size, the basic idea is to divide the key into 36 segments and treat each segment as a secret. Secret shares are then created for each secret using the (m-n) threshold scheme [11]. Table 1 shows the size of the secret for different key lengths obtained by dividing the total key length by 36 and rounding-off to the nearest integer. The extra bits left due to rounding-off are padded with random data bits that are removed in the key regeneration phase.

Table(1)
Size of secret for different key length

Key Length (bit)	Size of Secret
512	15-bit
1024	29-bit
2048	57-bit
4096	114-bit

5. EXPERIMENTAL RESULTS

The novelty in our design is that it keeps the genuine acceptance rate high while rejecting an imposter. The genuine acceptance rate can be increased by increasing the shadow angles in the enrolment phase. This will enable a constant value of the key to be regenerated for a legitimate user and will be tolerant to errors in the fingerprint pattern. Fig.5 shows the results obtained using 270 genuine fingerprint comparisons for different shadow angles. We were able to get a 100% genuine acceptance rate at r=10 for all the fingerprint samples in our database. However, increasing the shadow angles will not increase the false acceptance rate because the hashes produced by Eqs.2 and 3 not only depends on the fingerprint, but also on the password and random seed values. Therefore, even if two different individuals have fingerprints that are quite similar, our algorithm can discriminate them very sharply since the indexing of the secret shares that forms the private key is a function of password, fingerprint and the random seed values.

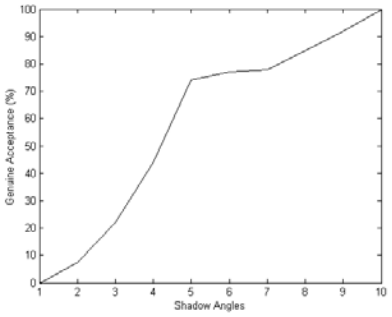

Fig 5. Genuine Acceptance Rate

6. CONCLUSION

The proposed scheme is aimed to enhance the security of cryptographic keys. Instead of storing a key with password-based encryption, our method enables the key to be regenerated dynamically when required for a cryptographic application. The results shown in Section 5 illustrate the robustness of our algorithm against variations in the fingerprint pattern at the time of key regeneration. By increasing the shadow angles, the genuine acceptance rate can be brought to 100% thus ensuring the correct regeneration of the cryptographic key. In addition, we have also carried out the security analysis of our proposed algorithm against on-line and off-line attacks. Our scheme in fact offers more security against such attacks as compared to password-based schemes since in addition to the knowledge of the password, our scheme also requires correct fingerprint and a valid smart card to regenerate the private key. Currently, we are using minutia points for the purpose of image alignment. In future, we will investigate the use of Trace Transform [12], which is a rotation, translation and scale invariant transform to estimate the rotation and translation parameters for image alignment. By using the trace transform, minutia points will not be required for image alignment and can therefore be used along with the ridge angle features for key regeneration. This will make the role of biometric stronger and shall improve the security of the overall scheme.

7. REFERENCES

[1] Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., USA, 1996.

[2] Shai Halevi and Hugo Krawczyk, "Public-Key Cryptography and Password Protocols", *ACM Transactions on Information and System Security*, 1999, Vol. 2, No. 3, pp. 230-268.

[3] A.K. Jain, R. Bolle, and S. Pankanti, editors, *Biometrics: Personal Identification in a Networked Society*, Kluwer Academic Publishers, 1999.

[4] Nichols, P.K, *ICSA Guide to Cryptography,* McGraw-Hill, 1999.

[5] Fabian Monrose, Michael K. Reiter, Qi Li, Susanne Wetzel, "Cryptographic Key Generation from Voice", *Proc. IEEE Symposium on Security and Privacy,* May 2001.

[6] Hao Feng, Chan Choong Wah, "Private Key Generation from On-Line Handwritten Signatures", *Information Management & Computer Security*, Vol. 10, No. 4, pp. 159 – 164, 2002.

[7] RSA Laboratories Frequently Asked Questions About Today's Cryptography. http://www.rsasecurity.com/rsalabs/faq/3-1-5.html

[8] Ville Taponen, "Tamper-resistant Smart Cards - Too Much To Ask For?", *Helsinki University of Technology*, 2000.

[9] A. Ravishankar Rao, *A Taxonomy of Texture Description and Identification*, Springer-Verlag, New York, 1990.

[10] A.K. Jain, L. Hong, S. Pankanti and R. Bolle, "An Identity Authentication System Using Fingerprints", *Proc. IEEE*, Vol. 85, No. 9, pp. 1365-1388, 1997.

[11] Adi Shamir, "How to Share a Secret", *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November, 1979.

[12] Alexander Kadyrov and Maria Petrou, "The Trace Transform and Its Applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 8, August 2001, pp. 811-828.