A Novel Approach for Generating Digital Signature using Fingerprint, Password and Smart Card

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

Abstract

Despite the fact that public key algorithms are very strong, their security lies in the safe custody of the private key. In case of private key compromise, extensive damage may be caused to the user as the key can be used to sign forged documents. The usual way to store a private key is through password-based encryption. However, user-chosen passwords have very low entropy, which may be exploited by an attacker to launch password-guessing attacks. In this paper, a new method is proposed that employ fingerprint, password and smart card to dynamically generate a private key for digital signatures. The scheme is sufficiently robust to generate a constant key and is tolerant to errors generated in the fingerprint pattern at the time of key generation. Our proposed algorithm is capable of generating key lengths that can meet the current security requirements of public key algorithms used for digital signatures and is more secure than traditional password-based method of protecting a private key.

1. Introduction

For the past few years, there has been a tremendous advancement in the field of electronic commerce, covering a large number of different applications. The main driving thrust behind this is the Internet that offers low cost means of communication. The Internet is however very insecure due to its inherent topology spreading across a number of different servers, routers at different geographic locations across the globe. Interestingly, public key cryptography whose roots are now spread over three decades forms the fundamental backbone in providing electronic security services to information that is either stored somewhere or passes through any network like the Internet. These services include:

- Confidentiality: The process of keeping information secret between a sender and receiver.
- Integrity: To ensure that information while traveling from sender to the receiver is unchanged.
- Authentication: To confirm data and identities.
- Non-repudiation: To ensure that a sender of a message should not be able to falsely deny later that he didn't sent the message.

In public key cryptography, two keys are used, i.e. a private key and a public key. Usually the public key is used for encrypting messages and is placed in a public register or a file while the private key is used for decrypting messages and is kept secret. If Alice wants to send a message to Bob, she can use Bob's public key to encrypt the message so that it is only possible for Bob to decrypt it using his private key. This provides confidentiality. In addition, digital signatures [1] address the issue of data integrity, authentication and non-repudiation. Using a digital signature, Alice can sign a message she intends sending Bob using her private key. After receiving the message, Bob can verify the signature using Alice public key. If the signature is correctly verified, it is proved that:

- The signature is authentic; when Bob verifies the message with Alice public key, he knows that she can only sign it.
- The signature cannot be forged; only Alice knows her private key.
- The signature is not reusable; the signature is a function of the document and cannot be transferred to any other document.

- The signed document is unalterable; if there is any alteration to the document, the signature can no longer be verified with Alice public key.
- The signature cannot be repudiated.

In a public key cryptosystem, the security of a private key is therefore very important. If the private key is ever compromised, extensive damage may be caused to the user. The key can be used by an attacker to sign forge documents or to decrypt secret messages. Because of the large size of a cryptographically-strong key, it is not feasible for a user to remember the private key and enter each time it is required. Instead, the private key is usually encrypted with a user chosen password. To retrieve the key, the user will have to enter the same password in order to perform successful decryption of the key. A common problem with password-based method is the low entropy, which may be exploited by an attacker to launch password guessing attacks [2]. The space of passwords is very limited. For an eight-character password, there are approximately 2 x 10^{14} combinations of English alphabets, both upper and lower case and digits. This is very small as compared to the size of a 1024-bit RSA key. Moreover, using a low-entropy password as a key to a strong cryptographic algorithm can transform it into a weak one. Therefore efforts are required to devise new methods that can increase the level of security the current password-based techniques offers in the safe custody of the private key.

In order to make the issue of key storage more secure, researchers are now looking into the use of biometrics. A biometric is a person's unique physical or behavioural characteristics that can be used to identify an individual. Physical characteristics includes fingerprints, hand geometry, retina, iris and facial characteristics, etc. Behavioural characteristics includes signature, voice, keystroke patterns and gait, etc. [3]. The basic aim of using biometrics is to devise a mechanism that is more secure in protecting the cryptographic key of a user as compared to conventional method of password-based encryption. One of the primary limitations with biometrics is that they can never provide absolutely certain measurements because personal features have a natural range of variation. This makes the task of linking biometrics with cryptographic algorithms quite challenging. Although password-based systems are weak from security perspective, they are 100% reliable. The presentation of a correct (incorrect) password will always correctly result in acceptance (denial) of a service. In this paper, we propose a novel method that combines password with fingerprint and information stored in a smart card to dynamically generate a private key that can be used for generating digital signatures or for decrypting messages. We will show that our proposed scheme is much more secure then conventional methods of storing a key through password-based encryption. In Section 2 we will survey the existing work done using biometrics to make the key storage mechanism more strong. This will be followed by a detailed description of our proposed algorithm in Section 3. In Section 4 we will present the security analysis of our proposed scheme against on-line and off-line attacks. In Section 5 we will show how keys of different lengths can be linked with our proposed algorithm. This will be followed by empirical results, conclusion and future work in Section 6 and 7 respectively.

2. Prior Work

There are two basic methods that can use biometric information for increasing the security of a cryptographic key. The first method involves template matching and key retrieval. In this method, the biometric data is captured and matched with a stored template. If the user is verified, the key is released. One such method is proposed in [4] that uses a combination of public-key encryption and fingerprints for personal authentication. In this scheme, the private key is stored in a smart card and is unlocked only if the fingerprint matching is successful. This provides a safe mechanism for private key storage as the smart card is kept with the legitimate user. However, in case if the card is stolen or lost, there is a chance of the private key being compromised, as smart cards offer some level of tamper-resistance but are not tamper-proof [5]. One of the most powerful attack against smart cards is the Differential Power Analysis (DPA) [6]. DPA relies on statistical inferences drawn on power consumption data measured during smart card computation. Particular computation creates particular patterns of spikes in power consumption. Careful analysis of the peaks in a power consumption pattern can lead to the discovery of information about secret keys used during cryptographic computations. DPA has been successfully used on Triple DES and 2048-bit RSA encryption [7]. Therefore storing a cryptographic key on a smart card with biometric-based access is secure only if the card is safely kept with the user. In case if the card is stolen or lost, it is possible to retrieve the key using DPA attack. In the second type of method that uses biometrics for secure key storage, the key is dynamically generated at run time instead of being stored in some medium. Bodo first proposed such a method in a German patent [8]. In this method, the data derived from the biometrics is used directly as a cryptographic key. This method however 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 the use of that biometric is irrevocably lost since the key is directly derived from the biometric. The compromised key can therefore be used to extract the biometric information. Another innovative idea of generating a cryptographic key from voice characteristics of a user is proposed by Monrose et al. [9]. Using this technique, a 46-bit key can be generated from a roughly two second spoken password. The idea of using online handwritten signatures for private key generation has recently been proposed [10]. The scheme can generate a key of 160-bits. In another method that uses fingerprints [11, 12], the key is linked with the biometric at the time of enrolment to form a data that gives no information of the key or the biometric. In the verification stage, the key is then reconstructed using the stored data and the biometric. The paper illustrates an example of generating a 128-bit key that can be used for cryptographic applications.

3. Our Method

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 et al. [10]. However in public key algorithms like the RSA, key lengths of 1024-bit or higher are required [13], keeping in view the current computational power. In order to solve this problem, we propose a novel technique in this paper that besides generating keys of higher lengths also improves the level of security the current password-based scheme offers for protection of a private key. Rather than storing a private key by password encryption, our method combines password, fingerprint features and data stored in a smart card along with some login information to dynamically generate a private key on demand. Once the key is generated it can be used with the specific public key algorithm to generate digital signatures. As compared to the method proposed in [4], our method uses smart card only for storing part of the enrolment information used for generating the private key i.e. the random seed values. Even if the information stored in the smart card is stolen by an attacker, the private key will never be compromised as the key is dynamically generated using the legitimate user fingerprint sample, password and valid registry files in addition to the information stored in the smart card. The proposed key generation algorithm uses minutia points for image alignment [14] and the orientation field of a ridge map of a fingerprint as a feature for key generation. This feature is designated as the *Ridge angle vector*, $\boldsymbol{R_a}$ and is calculated using the algorithm proposed in [15]. The minutia points are extracted using the algorithm proposed in [14]. We are restricted to use the minutia points as features for key generation because of three main reasons. Since image alignment requires storage of template minutia, this may enable an attacker to generate the key if the template is ever compromised. Secondly, the number of minutia points are not same among different people. In addition, there are a number of spurious minutia detected, which makes the key generation task extremely unreliable. Therefore we only use minutia points for image alignment as the alignment algorithm proposed in [14] requires only a subset of the minutia points between the input and the template to calculate the rotation and translation parameters. The proposed key generation algorithm consists of an *Enrolment Phase* and *Signature Generation Phase*. We denote a legal user by U_a and 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 private key by Kp_a. We assume that the user already has his public/private key pair, as the basic purpose of our algorithm is to link it with the existing private key of a user. In case if a new key pair is required, it can be generated by any public key algorithm like the RSA, etc.

3.1 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 $\boldsymbol{R_a} = \{\theta_m : 1 \leq \theta_m \leq 180$ degrees and $1 \leq m \leq 180\}$. Store $\boldsymbol{M_a}$ in $Fmin_a$.

2. For each $\theta_m \in \boldsymbol{R_a}$, define θ_{Sk} given by Eqn.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 generation.

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

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

4. Each consecutive $P_i \in P$ is treated as a secret and an $(m\text{-}n)$ threshold scheme [16] 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 Eqns. 2 and 3. The function H_{SHA} in Eqn. 2 uses SHA-1 algorithm [1] to give a 160-bit random and unique hashes, ζ is a public value selected arbitrary for every S. Eqn. 3 maps the 160-bit hashes generated by Eqn. 2 into a small hash values 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}.

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

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

6. For each segment of Kp_a, store the secret shares generated in Step 4 in $RandomSecret_a$ by 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 hard disk, etc. We call $EFmin_a$ and $ERandomSecret_a$ as the registry files for user U_a.

3.2. Signature Generation Phase

In this phase, the system will generate 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, $Scard_a$ to the system. The system reads the corresponding registry files $EFmin_a$, $ERandomSecret_a$ and does the following:

1. Decrypts $EFmin_a$ using pwd_a to get $\boldsymbol{M_a}$ and use a minutia alignment algorithm [14] 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 Eqn. 4, where $V = m = 180$ and form Groups G_i ($1 \le i \le 36$) comprising of five consecutive θ_{Vk}.

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

3. Read the encrypted δ values from $Scard_a$, decrypts them with pwd_a, and use Eqns. 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 [17]. 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 is 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 for U_a. Once the private key is generated, it can be used with the corresponding public key algorithm for generating digital signatures.

4. 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 generate the private key for digital signature. This is an obvious improvement in comparison to traditional method of key storage in which only a password is required to decrypt the private key for digital signature. In an off-line attack, we assume the security of the smart card and analyze our scheme against an attack on the registry files *EFmin* and *ERandomSecret.* We assume that the attacker has managed to capture these files and in addition is also successful in guessing the correct password to get the decrypted version *Fmin* and *RandomSecret. Fmin* only contains user minutia points and will not reveal any information regarding *Kp*. Using *RandomSecret* even if the attacker is able to construct all secrets through the *(m-n)* threshold scheme [16], he will still not be able to get the exact sequence that makes up the private key. For this, he will have to perform at most *36!* permutations, i.e. approximately 3.7×10^{41} number of iterations and in each iteration check for key validity. This is a considerable high overhead as compared to an eight-character password space, which has roughly 2×10^{14} different combinations of English alphabets both upper and lower case and digits. Even if a sixteen-character password is chosen, the total possibilities are roughly 5×10^{28} which is still low as compared to 3.7×10^{41}. In addition, since the file *RandomSecret* contains secret shares, it will not reveal any information of the ridge angle features i.e. the biometric used for the key generation process.

5. Key Length

Using our algorithm, keys of any length can be generated easily through Step 3 and 4 in the enrolment phase. The basic idea is to divide the key into 36 segments and treat each segment as a secret. 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 generation 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

6. Genuine Acceptance and Imposter Rejection

The novelty in our design is that it keeps the genuine acceptance high while rejecting an imposter. The genuine acceptance can be increased by increasing the shadow angles in the enrolment phase, this will enable a constant value of the key to be generated for a legitimate user and will be tolerant to errors in the fingerprint pattern. Fig. 1 shows one of our implementation in which 270 genuine fingerprint comparisons were made for different shadow angles. We were able to get a 100% genuine acceptance at *r=10* for all the fingerprint samples. However, increasing the shadow angles will not increase the false acceptance because the hashes produced by Eqns. 2 and 3 not only depends on the fingerprint data 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 random seed values.

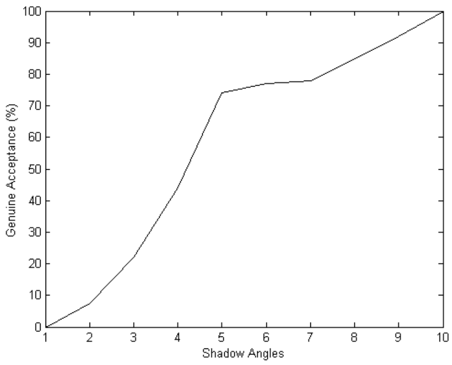

Figure 1. Genuine Acceptance Rate

7. Conclusion and Future Work

Despite the fact that public key algorithms are very strong, their security lies in the fundamental assumption that the private key is kept safe. For an eight-character password, there are approximately 2×10^{14} combinations of English alphabets, both upper and lower case and digits. This is very small as compared to the size of a 1024-bit RSA key. In order to make the key storage mechanism more strong, we have presented a scheme that provides more security to digital signature generation by using a password, fingerprint and smart card. Our method enables the private key to be generated dynamically whenever a message is required to be signed with a digital signature. The algorithm is sufficiently robust to generate a constant private key and is tolerant to errors generated in the fingerprint pattern at the time of signature generation. Our scheme can be used with any public key cryptosystem for digital signature generation as it can produce a key of any desired length. The security analysis shows that our scheme is much more secure as compared to traditional method of protecting a private key through password-based encryption. Our proposed scheme offers a distributed security approach. Instead of encrypting a private key with a password and storing it at one place, we have proposed an idea that requires a valid smart card, correct password and fingerprint to generate the key every time it is required for digital signature or any other cryptographic application. We have shown how to make a biometric enrolment information dynamic and distributed. For example, by simply changing the password, the entire enrolment information will be changed without requiring to revoke the cryptographic key or the biometric. In addition, the complete enrolment information is not stored at one place. If an attacker get access to user smart card he will not be able to deduce any information merely from the random seed values. Similarly the registry files cannot give any information of the biometric or the key. In our current implementation of key generation algorithm, we are using minutia points for the purpose of image alignment. In future we will investigate the use of Trace Transform [18] which is a rotation, translation and scale invariant transform to calculate the rotation and translation parameters for image alignment. By using the trace transform, minutia points will not be required to be stored which will improve the security of the overall system. Furthermore, since the trace transform is a one way transform, therefore no information regarding the fingerprint can be deduced from the trace transform parameters.

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] Yoshiaki Isobe, Yoichi Seto and Masanori Kataoka, “Development of Personal Authentication System using Fingerprint with Digital Signature Technologies”, *34th Annual Hawaii International Conference on System Sciences* (HICSS-34)-Vol. 9 January, 2001.

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

[6] Kocher, P. & Jaffe, J. & Jun, B., “Differential Power Analysis”, *CRYPTO 1999, Proceeding, Springer-Verlag*, pp. 388-397, 1999.

[7] How safe is your smart card?, http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci868301,00.html

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

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

[10] 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.

[11] Colin Soutar, Danny Roberge, Alex Stoianov, Rene Gilroy, and B.V.K. Vijaya Kumar, “Biometric Encryption™ using image processing”, *Proc. SPIE* 3314, 178-188, 1998.

[12] Colin Soutar, Danny Roberge, Alex Stoianov, Rene Gilroy, and B.V.K. Vijaya Kumar, "Biometric Encryption™ - Enrollment and Verification Procedures", *Proc. SPIE* 3386, 24-35, 1998.

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

[14] 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.

[15] A. Ravishankar Rao, *A Taxonomy of Texture Description and Identification*, Springer-Verlag, New York, 1984.

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

[17] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[18] 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.