Private key generation from on-line handwritten signatures

Hao Feng
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Chan Choong Wah
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

Keywords
Internet, Computer security, Biometrics

Abstract
n recent years, public key infrastructure (PKI) has emerged as co-existent with the increasing demand for digital security. A digital signature is created using existing public key cryptography technology. This technology will permit commercial transactions to be carried out across insecure networks without fear of tampering or forgery. The relative strength of digital signatures relies on the access control over the individual's private key. The private key storage, which is usually password-protected, has long been a weak link in the security chain. In this paper, we describe a novel and feasible system – BioPKI cryptosystem – that dynamically generates private keys from users' on-line handwritten signatures. The BioPKI cryptosystem eliminates the need of private key storage. The system is secure, reliable, convenient and non-invasive. In addition, it ensures non-repudiation to be addressed on the maker of the transaction instead of the computer where the transaction occurs.

Information Management & Computer Security
10/4 [2002] 159–164

[ISSN 0968-5227]
[DOI 10.1108/09685220210436949]

Introduction

The exchanging of computer-based documents such as electronic mail messages, or documents within e-mail messages, over the Internet is common in commercial transactions. Such documents often contain sensitive information, e.g. legal contracts, financial transactions, shopping records etc. To prevent hackers from intercepting and reading commercial documents traveling through insecure networks, one must encrypt those documents. In addition, the documents need to be signed digitally, which provides assurance of data origin, integrity and non-repudiation.

What is a digital signature?

A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document after being sent is unchanged. Digital signature has been with us since 1976, when Whifield Diffie and Martin Hellman introduced the digital signature as an application of public key cryptography. Only recently have businesses and governments started to use digital signature technology to protect sensitive documents on the Web.

In June 2000, former US President Clinton signed the Electronic Signatures in Global and National Commerce Act (E-Sign) into law. Under the act, digital signatures are placed in the same legal category as pen-on-paper signatures, meaning individuals and businesses can now be legally bound to agreements verified over the Web. This act proves to be a big boost in Web transactions with the industry acceptance. Digital signature is accepted within the USA and also in other parts of the world.

All digital signature technologies employ a public key infrastructure, or PKI. Under public key infrastructure, an individual has a pair of keys: a private key and a public key. A digital signature is obtained as the sender signs a document with his private key. When the recipients receive the signed document, they use the sender's public key to authenticate the document and verify that it has not been tampered with in transit.

How secure is digital signature?

The E-Sign act offers digital signature the same legal status as the traditional written signature. However the act did not technically specify the types of technologies that can be used to create a digital signature. The act left a wide opening in the types of implementation and technologies of digital signature. The most popular accepted technologies by industries for implementing digital signature are RSA and DSA.

Currently no one can make a definitive judgement that the present algorithms are secure or insecure before the algorithms are really broken. It takes a huge cost and tremendous computation time to hack the algorithm. This often thwarts most of the hackers. A study suggests that the vast majority of security failures are due to blunders in implementation and management, and essentially independent of the strength of the underlying cryptographic algorithms (Ross, 1994).

For digital signature, the private key management is often a vulnerable link in the security chain. The relative strength of digital signatures, notwithstanding the strength of the underlying asymmetric cryptography, relies on the access control over the individual's private key (Jeff, 2001). For a public key algorithm, the security of

the private key relies on the difficulty of factoring a large prime number (typically 1,024 bits). But as it comes to the private key storage, the security strength drastically reduces to a six-to-eight-character password. human being cannot endure memorizing many passwords over a long time. He may use the same passwords for his e-mail account, network logon, on-line banking, office access PIN etc. He may write down the passwords on a piece of paper, which could be peeped at. He may choose his alias or date of birth as passwords, which could be guessed by someone close to him. The above highlights that using passwords is an unsure means for authentication. A person authenticated to the access of a private key only means that he has the knowledge of the password but does not necessarily mean that he is the right person. True authentication can only be achieved through biometrics.

What is biometrics?

Biometrics refers to the automated identification of a person based on his/her physiological or behavioral characteristics. This method of identification is preferred over traditional methods involving passwords and PIN numbers. Biometrics describes a person's unique physical or behavioral characteristics. Physical characteristics include fingerprint, palm geometry, retina and iris, etc. Behavioral characteristics include handwritten signature, keystroke pattern, and voice, etc. Owing to these unique-to-person features, biometrics is the only way to identify a person with sufficient legal background.

Biometrics complementing PKI

Biometrics, an advanced authentication means, is enjoying a renewed interest in industry security applications. It gained its momentum since the 11 September event, which alerts the importance of authenticating a passenger's true identity before boarding a plane. The two emerging technologies, biometrics and public key infrastructure, can well complement each other in many security applications. Currently researchers are actively looking into ways of combining the two technologies.

Notion of biometric signature

The notion of biometric signature was first seen in Pawan and Siyal's (2001) paper. They define the biometric signature as a process to derive a private key from a biometric sample and use the private key to sign an e-document. The advantages of this approach are obvious. As a unique private key can be dynamically generated from one's biometric sample, no storage of private keys is required. This eliminates the problem of vulnerability of private key storage, which resolves the key management issue. The dynamically generated private key provides great convenience in signing documents as one can sign documents anytime anywhere without having to carry a disk or smart card.

Difficulties in implementation

The implementation of biometric signature in application comprises two parts:

1. highly consistent biometric sample data are obtained; and
2. a private key is derived from the sample data.

The difficulty for the first part is that all the bits in the biometric sample should be "exactly" correct. Pawan and Siyal's paper only addresses the second part. They give a conceptual example of iris biometrics and presume that a 512-byte iris sample has been obtained without a single bit error. Based on the sample, a private key can be derived following some well-established public key algorithms, e.g. RSA or DSA. However, when one's iris image is captured, it is extremely unlikely that every bit in the 512-byte sample is "correct". If it is, then it is most likely an attack.

This paper fills the gap in the first part. We propose a low-cost, reliable and feasible solution based on on-line signatures, a common form of behavioral biometrics.

A feasible implementation

The implementation is based on handwritten signature. The handwritten signature can be on-line or off-line depending on how the signature is obtained. An off-line signature is obtained by scanning a signature on paper and its features are static. This scanned image should ideally be watermarked. In contrast, an on-line signature is obtained by capturing the signing process on a tablet. The dynamic features obtained include speed, pressure, pen angles, etc., which are difficult to forge. In this paper, we only explore the on-line signatures.

What is the BioPKI cryptosystem?

We propose the BioPKI cryptosystem as the solution that demonstrates a novel way to merge the two technologies, biometrics and public key infrastructure (PKI). Figure 1 shows a block diagram of the proposed BioPKI cryptosystem.

The cryptosystem consists of three stages:
1 shape matching;
2 feature coding; and
3 private key generation.

The shape matching stage examines the shape of a test sample and filters out the random and simple forgeries. The feature coding stage finds a feature code for each of the defined features and concatenates each feature code into a code string. Finally, the private key generation stage takes the code string as the input and generates the individual's private key.

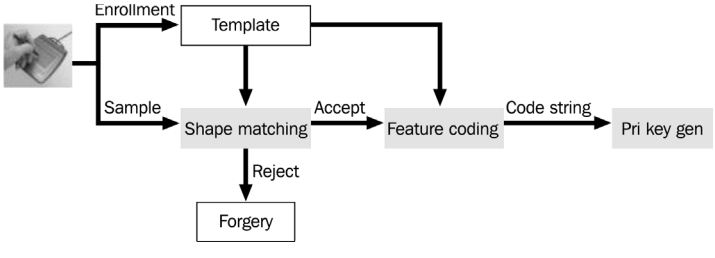
Operation of the system consists of enrollment and verification phases. During the enrollment phase, an individual provides ten sample signatures, from which a template and a pair of keys are derived. The private key is then discarded while the public key is kept. During the verification phase, the person provides a written test sample. After being processed by the three stages, a private key is generated. If the private key matches with the kept public key, the test sample is authentic and the generated private key can be used to digitally sign an e-document.

Stage 1: shape matching

Shape matching consists of the examination of the static features of a test signature with respect to a reference one, i.e. the template. Only signatures with very similar shapes will proceed to the next stage. It is necessary to filter out some random or simple forgeries in the first place. The static features (i.e. the image) of a reference signature do not reflect anything related to the private key generated later. Hence it can be safely written into the template.

In our implementation, we apply dynamic time warping (DTW) (Sankoff and Kruskal, 1983) to shape matching. DTW is to align the shapes of x, y waveforms from a test sample with the reference ones. Figure 2 shows a demonstration of the waveforms before and after DTW.

Figure 1
A block diagram of the BioPKI cryptosystem



In Figure 2, the top two graphs (a, b) are drawn from the reference data. The middle two graphs (c, d) are from the sample data, while the bottom two graphs (e, f) are from the position-warped sample data. Both x and y are independently warped through DTW. The graphs on the left panel (a, c, e) show signatures in x-y coordinates while the graphs on the right panel (b, d, f) show x, y data along the point serial number. From graphs (b), (d) and (f), one may notice that peaks and valleys of the sample waveforms are shifted to align with those of the reference ones. Some shifts in waveforms have been highlighted in graphs (b), (d) and (f) of Figure 2. Correlation coefficients can be obtained between position-warped x, y data and the reference ones. Low correlation coefficients will result in rejection of the sample.

Test results from a database comprising 25 users (750 authentic samples and 250 forgeries) show that 47.2 per cent of the forgeries are rejected at this stage while only 3.4 per cent authentic samples are rejected.

Stage 2: feature coding

Only "good-quality" signatures will proceed to the feature coding stage. In this stage, it will extract values of pre-defined features and code the feature values in decimal format. For each feature, the decimal number is the feature code. All the feature codes will be later concatenated together to form a code string.

As one remembers, we need to get a code string with every bit "exactly" correct. In our implementation, we first define a scheme of feature coding to achieve this goal. Here we take one of the features, pen-down time, as an example. Figure 3 demonstrates how this could be done.

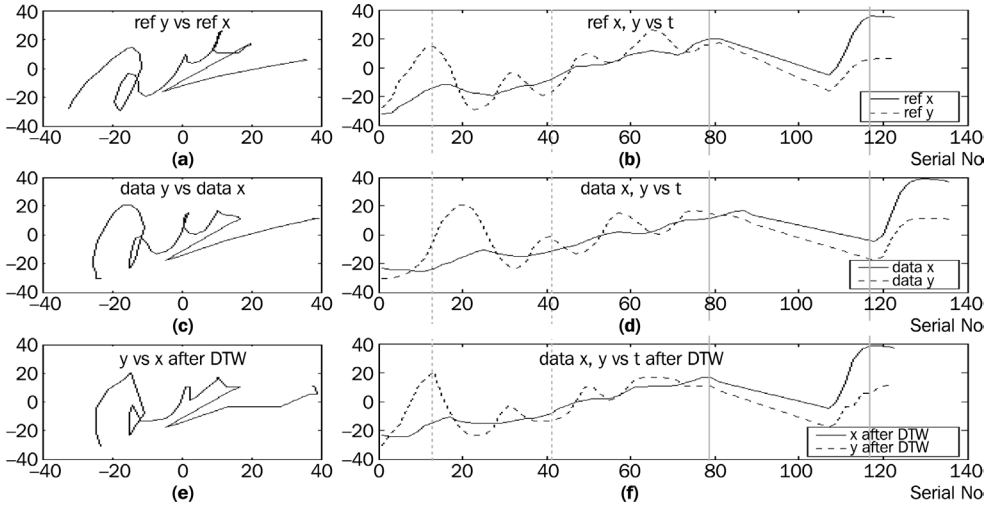
Figure 3(a) shows the histogram for the pen-down time values of 750 authentic samples in our database. Figure 3(b) is a skeleton view of Figure 3(a).

In Figure 3(b), three boundaries are defined. The whole boundary includes all possible values for a feature. For pen-down time, the whole boundary is between 0 and infinite. The database boundary includes values collected from the database. The user boundary includes values for a particular user. The user boundary is defined as:

$$\text{User boundary} = (\overline{T} - b \times \text{std}_T, \overline{T} + b \times \text{std}_T) \quad (1)$$

During enrollment, ten samples will be collected from the user. "$\overline{T}$" is the mean of the ten feature values. "std_T" is the standard deviation of those ten values. The user boundary is flexible and its range is adjusted

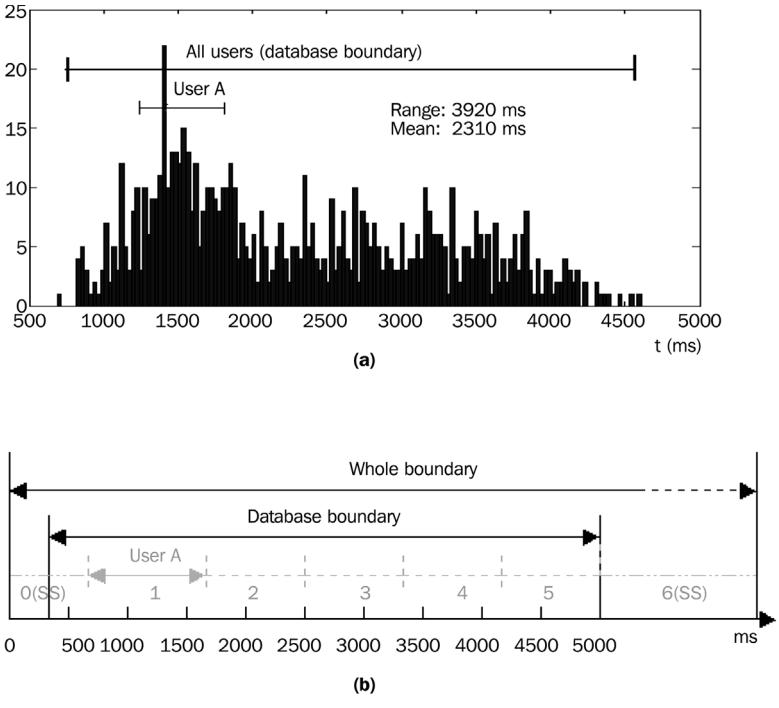
Figure 2
Waveforms before and after DTW



by the variable "*b*". The "*b*" is a system parameter to be adjusted. It can be an arbitrary decimal number. A bigger "*b*" value corresponds to more error tolerance, but, however, easier barrier for forgeries.

In our scheme, the whole boundary will be divided into several segments. Each segment will be assigned a decimal number starting from 0. The segmentation takes place in the following order. First a user boundary is defined with a chosen value "*b*". Then the same boundary is unfolded to both ends before exceeding the database boundary. Finally the superfluous portion at either end would be extended into the whole boundary and becomes one segment. The two superfluous segments in Figure 3(b) are numbers 0 and 6. The boundaries for all the segments will be defined in a template. The system would first extract a particular feature value, fit it into a segment and obtain the feature code, i.e. the sequence number. After processing all the features, the feature codes are concatenated to output a code string.

Figure 3
A demonstration of feature coding for pen-down time



The template includes the boundary definitions without any hint on a particular segment. It does not release any information about the feature code nor arise any privacy concern as usually people have for biometric storage.

For feature selection, a relatively flat histogram of the feature is preferred, which is important as it avoids data humping (much higher probability) in certain regions. In addition, only dynamic features are included for feature coding. The rationale for this is that dynamic features, unlike static features, are transparent to users. Visually it does not give any hint about the feature codes, even with the knowledge of all boundary definitions.

We have defined 43 features. It is beyond the paper's length to list all of them. But to name a few, they are: pen-down time, RMS of *Vx*, RMS of *Vy*, max forward *Vx*, max backward *Vy*, time when the last peak of *Vx* or *Vy* occurs, etc.

We use μ to denote the number of segments defined over the database boundary for each feature. For pen-down time $\mu = 5$ (see

Figure 3), which excludes the 0th and sixth segments as they are superfluous. Hence the bit-information of this feature can be obtained as $\lambda = \log_2 \mu$. The total bit-information for one person's signature is the summation of λ for each of the 43 features. The total bit-information varies from person to person. In general, the more consistent the signature is, the more bit-information it would convey. The average bit-information for 25 users in the database is of 40 bits, which will be introduced later.

Stage 3: private key generation

For private key generation from a pre-obtained code string, digital signature algorithm, or DSA (NIST, 1992) is preferred over RSA (Pawan, 2001). With DSA, the private key and the public key can be computed in the steps below:

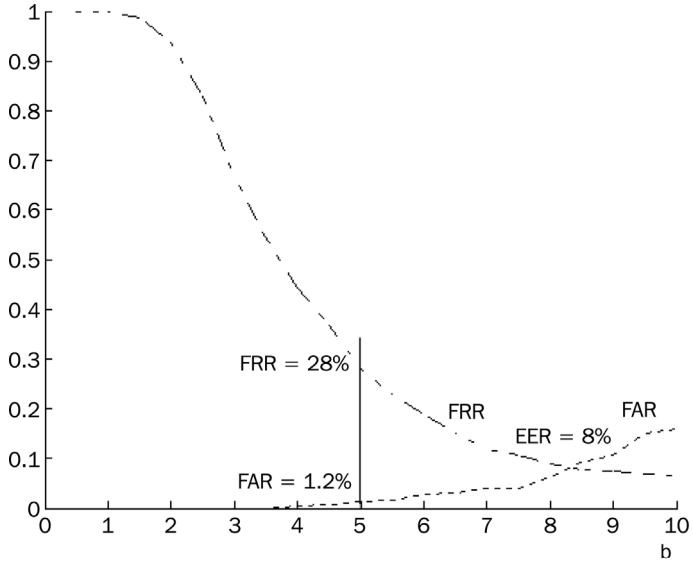
1 *Computation of p, q and g*:
   - $p = 512 \sim 1{,}024$ bit prime number;
   - $q = 160$ bit prime factor of $p-1$;
   - $g = h^{(p-1)/q} \bmod p$, where $h < (p-1)$ and $h^{(p-1)/q} \bmod p > 1$.

2 *Generation of private key*. Compute SHA1-hash of code string obtained. The hash value is a 160-bit private key, denoted by x.

3 *Generation of public key*. Compute $y = g^x \bmod p$. "y" is a p-bit public key.

From step 2, one may appreciate the importance of all-bits-correctness since a hash function is involved. A single bit difference at the input would result in very big difference at hashed output (Bruce, 1996).

Figure 4
FRR and FAR for overall system



Results

We use false rejection ratio (FRR) and false acceptance ratio (FAR) to evaluate the performance of the overall system. Figure 4 shows FRR and FAR plots versus variable "b".

The results appear encouraging. The equal error rate (EER), where FRR intersects with FAR, is only 8 per cent. FAR indicates the percentage that a legal private key would be generated from a forgery. A low FAR is desired because it would give people high confidence about the legal validity of the private keys generated this way. As an example, we could choose $b = 5$. When b equals 5, the authentic sample's false rejection ratio (FRR) is 28 per cent and the forgeries acceptance ratio (FAR) is only 1.2 per cent.

An interesting note is on 28 per cent FRR, which may alleviate some people's concern on false alarm. From common perception, a string of all-bits-correct biometric data is difficult to obtain and it may be at the expense of high false alarm. However, in our implementation based on on-line signatures, the false alarm is at a reasonable level. Each user may try 1.4 times on average to get the private key from the signature, which is not too annoying to most of the people.

Is the private key unique?

By nature, handwritten signatures may not be as unique as other biometrics, i.e. fingerprint, iris, and retina. When "b" is chosen to be 5, the bit-information for an individual's signature is on average of 40 bits. Ideally if the data distributions (see Figure 3(a)) of all defined features are uniform, the uniqueness of a signature is 1 in 2^{40}. When considering selecting features into the system, we choose those features with flat data distribution in addition to their consistency. However, ideal uniformity is impossible, which makes the actual uniqueness to be much less. Besides uniqueness, 40-bit information may not be strong enough against brute-force attack if someone tries different combinations, i.e. one bit by bit. We are in the process of improving the bit information. Two remedies are suggested below:

1 Include more features into the system. The more features are included, the more bit information will be added up. However, one may expect the false rejection ratio (FRR) to be higher since it would be more stringent to achieve all-bits-correctness.

2 Add padding information to the code string. Before the 40-bit code string is

hashed with SHA-1, it is concatenated with some padding information. The padding information could be either from user's keying in, i.e. user name or pass phase, or from the template. For example, the template may save the timestamp (in milliseconds) of user's first-time registration, which is unique to each user.

In both ways, the uniqueness of the private key can be guaranteed.

Conclusions

A novel system to perform digital signatures using biometrics has been proposed. The system combines the advantages of both PKI (integrity, confidentiality and non-repudiation) and biometrics (addressed-to-the-person authentication). A unique private key is dynamically generated from the user's hand written signature.

A feasible implementation of the BioPKI cryptosystem has been discusses in detail. The system takes on-line signatures as biometric samples. It performs shape matching to rule out poor-quality signatures in the initial verification phase. It then extracts feature codes and concatenates them into an all-bits-correct code string. Finally a private key is derived from that code string. Two remedies are also suggested to improve uniqueness and enhance security strength.

The results are encouraging. With the false acceptance ratio (FAR) as low as 1.2 per cent, the false rejection ratio (FRR) is found to be at a reasonable level, i.e. 28 per cent.

References

NIST (1992), "The digital signature standard proposed by NIST", *Communications of the ACM*, Vol. 35 No. 7, July, pp. 120-6.

Pawan, K.J. and Siyal, M.Y. (2001), "Novel biometric digital signatures for Internet based applications", *Information Management & Computer Security*, Vol. 9 No. 5, pp. 205-12.

Ross, J.A. (1994), "Whither cryptography?", *Information Management & Computer Security*, Vol. 2 No. 5, pp. 13-20.

Sankoff, D. and Kruskal, J.B. (1983), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing, Reading, MA, pp. 125-60.

Scheneier, B. (1996), *Applied Cryptography*, 2nd ed., John Wiley & Sons, New York, NY.

Stapleton, J. (2001), "PKI forum: biometrices", available at: www.pkiforum.org/pdfs/biometricsweb.pdf