Forward-Secure Subliminal Channels based on GQ Signature *

Dai-Rui Lin, Chih-I Wang, Chun-I Fan and D-J Guan

Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung 804, Taiwan
{d9134812, d9134810}@student.nsysu.edu.tw

Abstract

A subliminal channel is a communication channel that enables the signer to transmit an extra secret message concealed in a digital signature to an authorized receiver. Several subliminal channels based on various digital signature schemes have been proposed. However, most proposed methods allow the subliminal receivers to obtain more information to forge a valid signature, and if the same of subliminal message are sent twice, then the outsider can distinguish subliminal channels in the signature. Moreover, once the secret key of a subliminal channel is compromised, there have no solution in current research to prevent this kind of situation reword. In this work, the subliminal channel is embedded into key-evolving base GQ signature(kebGQ) to create a subliminal forward-secure channel. The security of the signature scheme can clearly be improved if the subliminal receivers do not share any part of the signer's secret, and the subliminal messages can be hidden in different periods. Outsider cannot identify subliminal channels, even if the message is transmitted in twice.

Keywords: Subliminal channels, Digital signatures, Key-evolving protocol, GQ scheme.

1 Introduction

A digital signature which hides a secret message in it is called a subliminal channel. The verification of the signature by receiver is the same as the original signature scheme, therefore, the public cannot know if there are concealed messages in it. Only the subliminal receiver can retrieve the subliminal message from the digital signature.

*This work was supported in part by TWISC@NCKU under grant NSC 94-3114-P-006-001-Y

1.1 Signature with subliminal channel schemes

In 1983, Simmons first constructed a subliminal channel in a digital signature scheme [14]. Since then, several schemes for the subliminal channel based on various digital signature schemes have been proposed [5, 7, 10, 11, 12, 14, 15]. All of the subliminal channel schemes consists three keys, which are the secrete key, the public key, and the subliminal key. The secret key is used to sign the message, the public key is used to verify the signature, and the subliminal key is used to hide and decode the subliminal message. Subliminal receivers have to share the subliminal key with the signer to protect the subliminal message. In Simmons' scheme, the subliminal key is the same as the secret key for signing the message, for this reason, the subliminal receiver can forge a valid signature easily.

In 1997, Harn and Gong proposed two digital signature schemes with subliminal channels [5]. The main feature of their schemes is that the subliminal receivers do not need to share the whole secret key as the subliminal key. Therefore, no subliminal receivers can forge a valid signature alone. However, each subliminal receiver still share a part of the secret key, both of their schemes will suffer from the conspiracy attack. That is, if there are enough subliminal receivers conspire against the message signer, they can derive the secret key to forge a valid signature or reduce the security of the digital signature.

In 1999, Jan and Tseng proposed two digital signature schemes with subliminal channels based on the discrete logarithm problem [7]. Both schemes are more efficient than the previously proposed scheme [5] and allow multiple subliminal receivers to simultaneously extract different subliminal messages in a signature. Their first scheme has the same conspiracy attack as

in the Harn-Gong's scheme [5]. The security of their second scheme is still not perfect because the subliminal receivers can obtain information of the secret key although he/she does not know the correct secret key. If the length of the secret key is k_1 bits and the length of the subliminal secret key is k_2 bits, the security of the signature is only $k_1 - k_2$ bits for the subliminal receiver, instead of k_1 bits. That is, the subliminal receiver can get more information to forge a signature than other people. Moreover, both Jan-Tseng's two schemes suffer from a new malicious subliminal receiver attack [12]. A malicious subliminal receiver can forge a subliminal message which will be accepted by other subliminal receivers belonging to the same subliminal channel.

1.2 Forward-secure signature schemes

Bellare and Miner initially proposed GQ signatures with forward-security properties [3]. In 2001, Abdalla and Reyzin improved the Bellare-Miner's forward-secure GQ signature schemes with a shorter public key[2]. Many studies about forward-secure related schemes have been presented [13, 6, 9, 16]. The main concept of forward-secure signature scheme is that: the public key is fixed, while the secret signing key is updated at regular intervals. Each secret signing key is adopted to sign messages only during a particular time period. A new secret key is produced, and old one erased, at the end of each time period, which can mitigate the damage caused by key exposure without requiring keys to be distributed.

Lu and Shieh proposed the GQ signature scheme with key-evolving protocol in 2004[13], and made the forward-secure GQ signature with low-complexity key-evolving protocol, and more efficient then the previous works. A synchronized key-updating mechanism is utilize to update not only the secret signing key, but also the public key during the same period.

This study develops a subliminal channel scheme derived from Lu and Shieh's scheme [13]. The subliminal key of the proposed approach is independent of signer's secret key. The key evolving protocol is employed to achieve a forward-secure subliminal message, guaranteeing the security of the signature and the subliminal messages in different periods.

1.3 Applications

1.3.1 Application of subliminal channels

The daily military broadcast program is a good application of the subliminal channel. This program broadcast songs, talks or advertisements every day, including digital signatures to prove that the program is official. The commander can send orders secretly hidden in the signatures. The enemy cannot know when the commander transmits the orders, nor what those orders are.

1.3.2 Application of forward-secure subliminal channels

Another application of a subliminal channel is when a credit card provider hide a card holder's credit history and credit limit in a digital signature for an issued credit card. However, each credit card has a valid period time, and has to revoked or destroyed once the valid period time has finished. In this case, we need the concept of forward-subliminal channel.

2 Review of key-evolving protocol for GQ signature scheme

In this section, we briefly review the key-evolving protocol for GQ signature scheme that proposed by Lu and Shieh [13]. The Lu-Shieh's idea is combination of GQ signature and key-evolving protocol [1] to produce the GQ signature over multiple periods with forward-secure, and depicted in Figure 1.

As in the RSA setting, the signer generates two large prime p and $q(N = pq)$. Then, the signer chooses a public exponent prime e and compute the corresponding long term secret key d as $d = e^{-1} mod\ \phi(N)$.

2.1 Key-evolving protocol

The key-evolving protocol can reduces that the number of publishes or redistributions of public keys and secret keys via synchronized key-updating mechanism. The key-evolving protocol between signer and verifier for period i are compute as follows. First, the signer chooses a random number $v_0 \in Z_N^*$ and publishes the v_0 to verifier. In the period i, the signer computes $v_i = h(v_{i-1})$ and then computes $s_i = (1/v_i)^d \pmod N$. At the same period i, the verifier computes $v_i = h(v_{i-1})$ iteratively with v_0. Then, the signer and verifier synchronously update the short

term key s_i and v_i for each period i. Both of s_i and v_i are linked via ${s_i}^e v_i = 1$, where $i = 1, ..., T$, and T to be total number of periods.

2.2 Key-evolving protocol for GQ signature scheme

The scheme is divided into four phases, (1) *key generation*, (2) *signature generation*, (3) *signature verification* and (4) *key updating*.

- *Key generation phase*:
  Without the loss of generality, the key generation is the same as the key-evolving protocol. After the signer chooses a random number $v_0 \in Z_N^*$ as his/her first short term public key, and computes the corresponding first short term secret key $s_0 = (1/v_0)^d$, the signer publishes the verification key set $VK = (N, e, v_0, h, H)$, where h and H are both hash function.

- *Signature generation phase*:
  Then the signer generate the signature for period i by the short term secret key s_i as follows. First, he/she chooses a random number $r \in Z_N^*$ and computes $a = H(r^e||M)$ and $z = r(s_i)^a$, where M is the message to be signed. The signature for period i is (a, z, i) and publish (a, z, M, i) to verifier.

- *Signature verification phase*:
  Upon receiving (a, z, i, M), the verifier updates the verification key to obtain v_i. Then, he/she computes $a' = H(z^e v_i^a||M)$. The signature is valid if $a = a'$.

- *Key updating phase*:
  Via synchronized key-updating mechanism, in each current period i, the signer updating the short term key s_i as $s_i = (1/h(v_{i-1}))^d \pmod N$, and the verifier computes $v_i = h(v_{i-1})$ iteratively with v_0 at the same period i.

Signer		Verifier
$v_i = h(v_{i-1})$		$a' = H(z^e v_i^a \|\| M)$
$s_i = (\frac{1}{v_i})^d (\mod N)$		$= H(r^e (s_i)^{ae} v_i^a \|\| M)$
$r \in Z_N^*$	$\xrightarrow{a, z, M, i}$	$= H((r^e (\frac{1}{v_i})^{dea} v_i^a \|\| M)$
$a = H(r^e \|\| M)$		$= H(r^e \|\| M) \stackrel{?}{=} a$
$z = r(s_i)^a$		

Fig. 1. The Lu-Shieh's kebGQ signature.

3 Forward-secure subliminal channel based on Key-evolving base GQ signature

In this section, we consider the forward-secure subliminal channel based on kebGQ signature scheme. We implement a subliminal channel on the condition that the verification formula of the underlying signature scheme is kept unchanged.

3.1 The random number r

The subliminal message can embed into any signature schemes where the signature scheme construct with the random number. There are many papers implied that the random number can embed the subliminal message [14, 5, 7, 12, 15, 8]. In order to embed the subliminal message into the GQ signature, we can define the random number r via some processes as follows.

Definition 1 *Given a subliminal message c and subliminal key k, the random number r can be expressed as $r = c^d k$.*

Let the k to be a subliminal key that pre-share in advance between the signer and subliminal receiver. Whenever the signer want to embed the subliminal message c into GQ signature [4], the random number r can define as $r = c^d k$. We can use a trivial scheme to illustrate the basic idea, that is, we can replay the random number $r \in_R Z_N^*$ as $r = c^d k$ for GQ signature.

Definition 2 *Given a set of subliminal message $C = \{c_0, ..., c_n\}$ and a set of subliminal key $K = \{h^0(k_0), h^1(k_0), h^2(k_0), ..., h^n(k_0)\}$, where $h(\cdot)$ to be a one-way hash function, and $k_0 = h^0(k_0), k_1 = h^1(k_0) = h(k_0), k_2 = h^2(k_0) = h(h(k_0)) = h(k_1), k_n = h^n(k_0) = h(k_{n-1})$. We can get a set of random number $R = \{r_0, ..., r_n\}$, where $r_i = c_i^d k_i$, for $i = 0, 1, ..., n$.*

In order to achieve the subliminal channel with specific of forward-secure, during the period i, the random number r_i can define as $r_i = c_i^d k_i$, where the c_i is the ith subliminal message, and the k_i is the ith subliminal key. By employee the hash function H, the ith subliminal key k_i can updating by the i-1th key k_{i-1} as $k_i = H(k_{i-1})$.

3.2 Forward-secure subliminal channel based on GQ signature scheme

Our construction of the proposed scheme depicted in Figure 2. The proposed scheme consist three parties, (1) *The signer*, (2) *The verifier* and (3) *The subliminal receiver*, and divided into five phases, (1) *key generation*, (2) *signature generation*, (3) *signature verification*, (4) *key updating* and (5) *subliminal message recover*. We describe the details of the five phases as follows.

- *Key generation phase*:
  The signer generates two large prime p and $q(N = pq)$. Then, the signer chooses a public exponent prime e and compute the corresponding long term secret key d as $d = e^{-1} mod\ \phi(N)$. Next, the signer chooses a random number $v_0 \in Z_N^*$ as a first short term public key, and computes the corresponding first short term secret key $s_0 = (1/v_0)^d$. The signer publishes the verification key set $VK = (N, e, v_0, h, H)$, where h and H are both hash function. During time period i, the signer computes $v_i = h(v_{i-1})$ and then computes the periodic signing key $s_i = (1/v_i)^d \pmod N$.

- *Signature generation phase*:
  Then the signer generate the signature for period i by the short term secret key s_i as follows. First, he compute r_i as $r_i = c_i^d k_i$ and computes $a_i = H(r_i^e||M_i)$ and $z_i = r_i(s_i)^{a_i}$, where M_i is the ith message to be signed. The signature for period i is (a_i, z_i, i) and then publish (a_i, z_i, M_i, i) to verifier.

- *Signature verification phase*:
  Upon receiving (a_i, z_i, M_i, i), the verifier updates the verification key to obtain v_i. Then, he computes $a_i' = H(z_i^e v_i^{a_i}||M_i)$. The signature is valid if $a_i = a_i'$.

- *Key updating phase*:
  The signer and verifier synchronously update the short term key s_i and v_i for each period i as: the signer computes $v_i = h(v_{i-1})$ and then computes $s_i = (1/v_i)^d \pmod N$. At the same period i, the verifier computes $v_i = h(v_{i-1})$ iteratively with v_0.

- *Subliminal message recover phase*:
  If the subliminal receiver who receiving the signature (a_i, z_i, M_i, i), after verify the signature in regular process, he can use the subliminal key k_i to extract the subliminal message c_i from the r_i $(r_i^e = (c_i^d k_i)^e = (c_i^{de} k_i^e) = (c_i k_i^e))$ as follows. First, the subliminal receiver updating the subliminal key k_i as $k_i = h(k_{i-1})$, then the subliminal message c_i can recover by computing $c_i = r_i^e / k_i^e$.

4 Security Analysis

It is straightforward that the security of the proposed scheme as equal to the kebGQ signature scheme as describe in [13]. In this section, we will show that our forward-secure subliminal channel scheme is robust.

Signer		Verifier
$v_i = h(v_{i-1})$		$a_i' = H(z_i^e v_i^{a_i}\|\|M_i)$
$s_i = (\frac{1}{v_i})^d (\mod N)$		$= H(r_i^e (s_i)^{a_i e} v_i^{a_i}\|\|M_i)$
$c_i : sub_{msg_i}$		$= H((r_i^e (\frac{1}{v_i})^{dea_i} v_i^{a_i}\|\|M_i)$
$k_i = h(k_{i-1})$	$\xrightarrow{a_i, z_i, M_i, i}$	$= H(r_i^e\|\|M_i) \stackrel{?}{=} a_i$
$r_i = c_i^d k_i$		Subliminal receiver
$a_i = H(r_i^e\|\|M_i)$		$r_i^e / k_i^e = (c_i^d k_i)^e / k_i^e$
$z_i = r_i (s_i)^{a_i}$		$= c_i^{de} k_i^e / k_i^e = c_i k_i^e / k_i^e = c_i$

Fig. 2. Forward secure subliminal channel based on kebGQ signature.

1) *The signature cannot be forged.*
In our scheme, the subliminal key is independent of the secret key which is used to sign the message. It means that the subliminal receiver cannot get any advantages in forging a valid signature. Therefore, it will not compromise the security of the signature.

Moreover, the attacker or the malicious subliminal receiver cannot modify the subliminal message while keeping the signature valid. Because he/she has to change the value of signature S, this is equivalent to factor N.

2) *No body can retrieve c_i, unless the subliminal receiver.*
In our schemes, the attacker cannot determine whether there are hidden subliminal messages or not, because the r_i behaves like a random value. In other words, the r_i is satisfy the specific of indistinguishable [8]. In addition, the verification steps are the same as GQ signature scheme, even if the attacker assumes that there are hidden subliminal messages, it is still impossible to find the subliminal messages, because k_i is unknown. Therefore, no information is disclosed.

3) *Indistinguishable.*
In the proposed schemes, once the same message send is sent twice, any verifier cannot identify potential message hidden in the signature, since the r_i in each period is different. That is, assuming that the subliminal message C send in twice or even multiple times, then $(r_i = C^d k_i) \neq (r_{i+1} = C^d k_{i+1}) \neq ...(r_{i+l} = C^d k_{i+l})$. The result that $(r_i = C^d k_i) = (r_{i+1} = C^d k_{i+1})$ is impossible. Restated, the r_i still acts as a random value.

4) *Forward-secure subliminal channel.*
The proposed schemes enable the subliminal message to be kept secret even if the secret key of the subliminal channel is exposed. If the ith subliminal key k_i was exposed, then k_{i-1} cannot be calculated from k_i unless the one-way hash function is insecure.

5 Discussions

In those applications, the independence between the keys is very important. If the subliminal receivers share a part of the secret key, they can conspire to forge a signature on a fake program. Others may regard the fake program as an official program and this is not allowed.

The malicious subliminal receiver attack [15] is also a problem. It means that a malicious subliminal receiver can forge a fake order while keeping the signature valid. Other soldiers will receive two or more orders with the valid signatures, therefore, they cannot know which one is correct.

The independence of the keys can avoid the above problems. In our forward-secure subliminal channel scheme, the subliminal key and the signer's secret key are independent, and the subliminal key will update in different periods for the multiple subliminal messages. Moreover, the malicious subliminal receiver attack and the conspiracy attack [5, 7] will not work in our scheme, because the subliminal receiver do not share any part of signer's secret key. The comparisons between our scheme and previous schemes [15, 5, 7, 10, 11] are shown in Table 1.

6 Conclusions

This investigation presents a forward-secure subliminal channel. The proposed schemes can generate subliminal channels without sharing the signer's secret key. The properties of the proposed schemes are summarized the below.

- The proposed schemes achieve the subliminal channels with specific of forward-secure.
- The subliminal receivers do not share the signer's secret key, so conspiracy is not a problem.
- The subliminal messages can be updated and concealed in different periods.

Table 1. The comparisons between [15, 5, 7, 10, 11] and our scheme.

	[15]	[5]	[7]	[10]	[11]	Our scheme
Attacking method	Forgery attack	Conspiracy attack	Conspiracy attack	No	No	No
Share signer's secret key	Yes	Part	Part	No	No	No
Forward secure	No	No	No	No	No	Yes
Potential # of subliminal messages	1	2	2	1	1	n

- The subliminal channel is indistinguishable even if the message is sent twice.
- The subliminal message has the non-repudiation property.

The subliminal channels created by the proposed schemes are more useful then those in previous works, since they permit a different subliminal message in each period i.

References

[1] M. Abdalla and M. Bellare. Increasing the lifetime of a key: a comparative analysis of the security of re-keying techniques. In *Proc. ASIACRYPT '00, LNCS 1976*, pages 431–448, 1999.

[2] M. Abdalla and L. Reyzin. A new forward-secure digital signature scheme. In *Proc. ASIACRYPT '00, LNCS 2139*, pages 116–129. Springer-Verlag, 2001.

[3] M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In *Proc. CRYPTO '99, LNCS 1666*, pages 431–448. Springer-Verlag, 1999.

[4] L. C. Guillou and J. J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both tranmission and memory. In *Proc. EUROCYPT '88, LNCS 330*, pages 123–128, 1988.

[5] L. Harn and G. Gong. Digital signature with a subliminal channel. In *IEE Proc. Comput. Digit. Tech*, volume 144, pages 387–389, 1997.

[6] G. Itkis and L. Reyzin. Forward-secure signatures with optimal signing and verifying. In *Proc. CRYPTO '01, LNCS 2139*, pages 332–354. Springer-Verlag, 2001.

[7] J. K. Jan and Y. M. Tseng. New digital signature with subliminal channels based on the discrete logarithm problem. In *Proceedings of the 1999 International Workshops on Parallel Processing*, pages 198–203, 1999.

[8] K. Kobara and H. Imai. Self-synchronized message randomization methods for subliminal channels. In *Proc. ICICS '97, LNCS 1334*, pages 325–334. Springer-Verlag, 1997.

[9] H. Krawczyk. Simple forward-secure signatures from any signature scheme. In *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pages 108–115, 2000.

[10] H. Kuwakado and H. Tanaka. New subliminal channel embedded in the esign. *IEICE Trans. Fundamentals*, E82-A(10):2167–2171, 1999.

[11] N. Y. Lee and P. S. Ho. Digital signature with threshold subliminal channel. *IEEE Transactions on Consumer Electronics*, 49(4):1240–1242, 2003.

[12] N. Y. Lee and D. R. Lin. Robust digital signature scheme with subliminal channels. *IEICE Trans. Fundamentals*, E86-A(1):187–188, 2003.

[13] C. F. Lu and S. Shieh. Efficient key-evolving protocol for the gq signature. *Journal of information science and engineering*, 20:763–769, 2004.

[14] G. J. Simmons. The prisoner's problem and the subliminal channel. In *Proceedings IEEE Workshop Communications Security CRYPTO'83*, pages 51–67, 1983.

[15] G. J. Simmons. Subliminal communication is easy using the dsa. In *Proc. EUROCYPT '93, LNCS 765*, pages 218–232. Springer-Verlag, 1993.

[16] W. G Tzeng and Z. J Tzeng. Robust forward-secure signature schemes with proactive security. In *Proc. PKC '00, LNCS 1992*, pages 264–276, 2001.