Jurnal Teknologi, 47(C) Dis. 2007: 37–44

A DIRECTED SIGNATURE SCHEME BASED ON DISCRETE LOGARITHM PROBLEM

EDDIE SHAHRIL ISMAIL[1] & YAHYA ABU HASAN[2]

Abstract. This paper presents a new directed signature scheme based on discrete logarithm problem, a scheme that permits only an intended verifier to verify a resulting signature. In addition, the scheme is able to prove to any third party the validity of the signature. Essentially, we illustrate that the scheme achieves the same level of security but is very efficient compared to Lim-Lee's scheme. Two immediate applications are also presented: flexible shared verification and signing contracts procedure extended from the new directed signature scheme.

Keywords: Cryptography; digital signature; directed signature; discrete logarithm problem; zero-knowledge technique

Abstrak. Kertas ini membangunkan skema baru tandatangan *directed* yang berdasarkan masalah logaritma diskret, suatu skema yang membenarkan hanya pengesah yang layak untuk mengesah tandatangan. Skema ini juga berupaya membuktikan kepada pihak ketiga keaslian tandatangan. Skema baru ini juga mencapai tahap keselamatan yang setara tetapi lebih efisien berbanding dengan skema Lim-Lee. Dua aplikasi langsung dibincangkan: pengesahan perkongsian fleksibel dan prosedur tandatangan kontrak cetusan daripada skema tandatangan *directed* yang baru.

Kata kunci: Kriptografi; tandatangan digital; tandatangan *directed*; masalah logaritma diskret; teknik *zero-knowledge*

1.0 INTRODUCTION

The idea on how one can sign a message electronically (digital signatures) presented by Diffie and Hellman [3] has solved many problems of online communication especially in message authentication. Many such protocols have been invented and explored, for example, blind signatures [2], fail-stop signatures [8], proxy signatures [5] and designated confirmer signatures [7]. Every protocol has its own characteristic and was devised to overcome their unique applicable problem. Consider this specific situation: Alice communicates with King Hospital digitally in order to obtain her digital health certificate with a valid signature on it from an officer, for example, a doctor. This certificate is personally sensitive to Alice and logically only she is able to verify the authenticity and integrity of the resultant signature plus only she could

[1] Fakulti Sains dan Teknologi, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor Darul Ehsan

[2] Universiti Sains Malaysia, 11800 USM Penang, Malaysia

prove the validity of the signature to any requested third party. This is important to ensure that the proof of validity of signature is publicly acceptable. To solve this kind of problem a scheme called directed signature is proposed [6].

This paper describes the construction of a directed signature which is based on discrete logarithm problem (DLP) [4] with two results: first, we managed to reduce the number of secret keys compared to the Lim-Lee's scheme and second we successfully extended the new scheme into a flexible share verification scheme, in which only the intended verifiers (formed into a group) have an ability to verify a resultant signature. The scheme is flexible since the cardinality of the group contains no limited number of verifiers.

2.0 THE DIRECTED SIGNATURE SCHEME

The definition of a digital signature scheme is given below. Refer to Schneier [9] for a complete discussion on signature systems.

Definition 2.1. A digital signature scheme is a scheme containing the following three procedures:

(1) **Setup:** An algorithm for generating all public and private keys used in the scheme.

(2) **Sign:** An algorithm that takes a private key and a message m as inputs and outputs a signature on a message m.

(3) **Verify:** An algorithm for establishing the validity of the received signature, given a public key and a signed message.

It is clear that a directed signature scheme is a basic digital signature scheme plus a fourth procedure stated as follows:

(4) **Proof:** An algorithm that uses a Zero-Knowledge technique for convincing a third party that the signature is valid without revealing the actual signature.

2.1 Setup of Parameters for the Scheme

(a) A trusted generator (TGR) selects two large primes (1024-bits) and q (size 512-bits) such that q divides p – 1.

(b) TGR next finds a generator $g \neq 1$ of order q in $Z_p^* = \{1, 2, \ldots, p-1\}$ satisfying $g^q \equiv 1 \pmod{p}$.

The ordered triple (p, q, g) represents the common public parameters of the scheme for all users. These parameters are then used by the signer, S and the intended verifier, V, to produce their own pair of public and secret keys.

(c) S and V select at random their respective secret keys x_S and x_V in $Z_q^* = \{1,2,\ldots,q-1\}$ and compute their corresponding public keys given by $y_S \equiv g^{x_S}$ (mod p) and $y_V \equiv g^{x_V}$ (mod p).

Now the two parties are ready to communicate via the signing and verifying processes. We show how S can sign a message that is personally sensitive to the verifier V.

2.2 Signature Generation

(a) The signer S picks at random an integer $k \in_R Zq$.

(b) Next he computes $r = H\left(y_V^k \bmod p\right)$ using one-way hash function H as of [9].

(c) He then calculates the number $e = r \oplus \mathrm{H(m)}$ and $u \equiv k\left(x_S - e\right)^{-1}$ (mod q).

He produces a valid signature on message m, given by the pair (r, u).

2.3 Verification Procedure

The verifier V is able to validate the signature to test whether it is genuine or not.

(a) He computes $\lambda \equiv \left(y_S^s g^{-es}\right)^{x_V}$ (mod p).

(b) He accepts the signature as valid if and only if $r = H(\lambda)$.

We prove the correctness of the above signature scheme as follows:

Theorem 2.1: *Assume that OSS is an ordinary signature scheme described as above. If the signer successful ly signs a given message m in Signature Generation, then the validation of signature in Verification Procedure of OSS is correct.*

Proof: Say that (r, s) is the signature of m, where $r = H\left(y_V^k \bmod p\right)$ and $s \equiv k\left(x_S - e\right)^{-1}$ (mod q). It is easy to show that,

$$H(\lambda) = H\left(\left(y_S^s g^{-es}\right)^{x_V}\right) = H\left(\left(g^{sx_S - es}\right)^{x_V}\right) = H\left(g^{kx_V}\right) = H\left(y_V^k\right) = r$$

In some cases, there is a need for the verifier to convince someone else that the signature is indeed valid. For this purpose, he can use the zero-knowledge proof technique [2] such that the third party T can gain nothing except for the validity of the signature.

2.4 Prove to Any Third Party

(a) The verifier V sends $\lambda \equiv \left(y_S^s g^{-es}\right)^{x_V}$ (mod p) and the resultant signature (r, s) to T.

(b) The third party T then checks that $r = H(\lambda)$ and computes $\gamma \equiv y_S^s g^{-es}$ (mod p).

(c) Using the zero-knowledge technique, V proves to T that $\log_\gamma \lambda = \log_g y_V$.

3.0 RESULTS AND DISCUSSIONS

We discuss the scheme's security and efficiency performances. As long as discrete logarithm problem is hard to solve, our scheme is hard to break and therefore is secure to use. This means that, no signer can sign on behalf of other signer and no enemy can claim that this signature is produced or signed by the signer unless he or she has really signed it. In addition, an unintended verifier will not be able to verify the signature and will have no ability to prove its validity to other people.

For efficiency consideration, the following results are tabulated. Each table compares our scheme and Lim-Lee's scheme in terms of the number of keys, computational complexity and communication cost. The following notations are used in the analysis: SK is the number of secret keys; PK is the number of public keys; T(exp) is the time for modular exponentiation; T(mul) is the time for modular multiplication; T(inv) is the time for a modular inverse computation; T(h) is the time for performing a one-way hash function $H(.)$; $|x|$ denotes the bit length of x. The time for performing modular addition/subtraction computation is negligible.

The comparison of the number of keys, computational complexity and communication cost between our scheme and Lim-Lee's scheme are illustrated in Tables 1-3. From the tables, we conclude that, our scheme is better than Lim-Lee's scheme.

Below is the small example of the above scheme.

Table 1 Comparison of the two schemes for signing process

		Lim and Lee's Scheme	Our Scheme
1.	Number of keys used	SK=3, PK=2	SK=2, PK=1
2.	Computational complexity	2T(exp)+T(mul)+T(h)	T(exp)+T(mul)+T(inv)+T(h)
3.	Communication cost	3|p|+|q|	|p|+|q|
4.	The signature's size	4-tuples	2-tuples

Table 2 Comparison of the two schemes for verifying process

		Lim and Lee's Scheme	Our Scheme
1.	Number of keys used	SK=1, PK=2	SK=1, PK=2
2.	Computational complexity	3T(exp)+2(Tmul)+T(h)	3T(exp)+T(mul)+T(h)

Table 3 Comparison of the two schemes for validity process

		Lim and Lee's Scheme	Our Scheme
1.	Number of keys used	SK=0, PK=4	SK=0, PK=4
2.	Computational complexity	5T(exp)+4T(mul)+T(h)	5T(exp)+2T(mul)+T(h)
3.	Communication cost	4|p|+|q|	2|p|+|q|

A trusted generator T of the scheme selects $p = 1319$ and $q = 659$. Then a generator g of Z_{1319} such that $g^{659} \equiv 1 \bmod 1319$ and $g \neq 1$ is selected and choose $g = 27$. Finally the trusted generator broadcasts a triple (1319, 659, 27) as common parameters and that will be used by interested signers and verifiers, say Sani and Vella respectively. Sani and Vella select their secret keys as $x_s = 200$ and $x_V = 300$ respectively and compute the corresponding public keys as $y_s \equiv 27^{200} \equiv 332 \bmod 1319$ and $y_V \equiv 27^{300} \equiv 498 \bmod 1319$. Sani is now ready to attach a signature onto the certificate by first selecting a random secret key $k = 400$ in Z_{659} and computing the two integers $r = \mathrm{H}(498^{400} \bmod 1319) = \mathrm{H}(409)$ and assume that $e = \mathrm{H}(409)\oplus\mathrm{H}(357) = 246$ where H(357) is a hash value of certificate. Finally, Vella calculates $\mathrm{s} \equiv 400(200 - 246)^{-1} \equiv 593 \bmod 659$ and produces a valid signature on Sani's health certificate as (H(409),539). Vella accepts the signature if $(332^{593}27^{(-246)(593)})^{300} \equiv 409 \bmod 1319$ holds.

4.0 APPLICATIONS

The directed signature scheme presented has immediate applications; such as in flexible shared verification and in signing contracts. The former permits a group of intended verifiers to jointly verify the arrived signature while the latter allows any two countries/companies to sign a message simultaneously.

4.1 Flexible Shared Verification

Let a group of n verifiers be denoted as $V = \{V_1, V_2, ..., V_n\}$. E can decide on a set Q which consists of only the intended verifiers to verify his signature. For shared verification, it is required that Q should contain two or more verifiers.

All verifiers in V generate their own public and secret keys as shown in the above scheme. Now let the owner chooses $Q = \{V_1, V_3, V_5\}$ and a signer now selects a random integer k and computes $r = H\left(\left(y_{V_1} y_{V_3} y_{V_5}\right)^k \bmod p\right)$ and $s = k(x_s - e)^{-1} \pmod q$ where $e = r \oplus H(m)$. The signer then outputs his valid signature as a pair (r, s). All verifiers in Q now can jointly verify that the signature is indeed valid. They first compute $\lambda_{V_i} \equiv \left(y_S^s g^{-es}\right)^{x_{V_i}} \pmod p$ and finally jointly verify the validity of signature by checking that $H\left(\Pi_{v_i \in Q} \lambda_{v_i} \pmod p\right) = r$ is hold. We state this in the following theorem.

Theorem 4.1. *Let $V = \{V_1, V_2, ..., V_n\}$be a group of n verifiers and $Q \subseteq V$ be the selected intended verifiers. A signer outputs his valid signature as a pair (r, s) given by*

$$r = H\left(\left(\Pi_{V_i \in Q}\, y_{V_i}\right)^k \bmod p\right) \textit{and } s = k\left(x_s - e\right)^{-1} \bmod q$$

where $e = r + H(m)$. The signature can be jointly verified if $H\left(\Pi_{V_i \in Q}\, \lambda_{V_i}\, (\mathrm{mod}\ p)\right) = r$ provided that $\lambda_{V_i} \equiv \left(y_S^s g^{-es}\right)^{x_{V_i}}\, (\mathrm{mod}\ p)$.

Proof: It is easy to verify that

$$H\left(\prod_{\mathrm{V_i} \in Q} \lambda_{V_i} \bmod p\right) = r \Leftrightarrow \left(\prod_{V_i \in Q} y_{V_i}\right)^k = \prod_{V_i \in Q} \lambda_{V_i} \bmod p$$

$$\Leftrightarrow g^{\sum V_i \in Q^x V_i k} = \left(y_S^s g^{-es}\right)^{\sum V_i \in Q^x V_i k} \bmod p$$

$$\Leftrightarrow g^{\sum V_i \in Q^x V_i k} = g^{\left(\sum V_i \in Q^x V_i k\right)(x_S - e)(x_S - e)^{-1}} \bmod p \Leftrightarrow \left(x_S - e\right)\left(x_S - e\right)^{-1} = 1 \bmod q$$

The only drawback of this shared verification is that, each verifier has to store secretly their private keys and broadcasts their respective public keys. The data file containing these keys will start to increase in size as new verifiers registered. However, we believe that there are some techniques that would change this into a more practical scheme.

4.2 Signing Contracts

In this case, two countries A and B can sign a document m simultaneously. The two delegates respectively choose at random secret keys k_A, $k_B \in_R Z_p^* = \{a \mid \gcd(a, p) = 1\}$ and jointly compute $r = H\left(y_B^{k_A} y_A^{k_B}\, (\mathrm{mod}\ p)\right)$ where $y_A \equiv g^{x_A}\, (\mathrm{mod}\ p)$ and $y_B \equiv g^{x_B}$ (mod p) are their public keys associated with the chosen private keys x_A, $x_B \in Z_p^*$. Each of them then calculates $s_A \equiv k_A(x_A - e)^{-1}(\mathrm{mod}\ q)$ and $s_B \equiv k_B(x_B - e)^{-1}(\mathrm{mod}\ q)$ where $e = r \oplus \mathrm{H}(m)$.

The two delegates produce a valid signature on the document m as (r, s_A, s_B). Upon signing, the two countries can validate the signature. Representatives from delegates A and B respectively compute $\lambda_A \equiv \left(y_A^{s_B} g^{-es_B}\right)^{x_B}$ (mod p) and $\lambda_B \equiv \left(y_A^{S_A} g^{-es_A}\right)^{x_B}$ (mod p). They accept the signature as genuine if and only if the following equation holds by co-operatively check that $r = H(\lambda_A \lambda_B(\mathrm{mod}\ p))$.

The size of signature depends on the number of countries involved. If there are t countries involved in the signing ceremony, then the size of signature is $(t + 1)$-tuple. It is obvious to note that the scheme is practical if $t = 2$.

We present a slightly different scenario: Let the keys of the two countries A and B be x_A and x_B respectively. Supposed that A and B have to generate a common published key $\alpha \equiv g^{x_A x_B} \pmod{p}$ and are requested by country C to sign a document m. The two countries produce $r = H\left(y_C^{k_A k_B} \pmod{p}\right)$ where y_C is a public key of country C and compute s_A and s_B as in the above signing contract.

They next produce $(r, s_A s_B)$ as a valid signature on the document m. It can also be proven that delegate C accepts the signature if $r = H\left(\left(\alpha \left(y_A y_B\right)^{-e} g^{e^2}\right)^{s_A s_B x_C} \pmod{p}\right)$. Alternatively, the two countries can produce $r = H\left(g^{k_A} +^{k_B} \pmod{p}\right)$ and $(r, s_A + s_B)$ as their valid signature, whose verification is given by $r = H\left(y_A^{S_A} y_B^{S_B} \left(g^{S_A + S_B}\right)^{-e}\right)$.

5.0 CONCLUSIONS

A directed signature scheme based on the discrete logarithm problem is presented. This type of digital signatures is very attractive when implemented in the scenario, where the content of a message or document is personally sensitive to the intended verifier. No one can verify the signature except the intended verifier. Our scheme is better than the scheme proposed in [6] in terms of the number of keys required publicly and secretly, the computational complexity and communication cost. A flexible group-oriented shared verification scheme is also proposed, where the owner of a message has the ability to decide which verifiers should verify his or her message. Two different kinds of signing contract schemes are also given as the alternatives for the existence schemes.

ACKNOWLEDGEMENTS

We thank the anonymous referees for their suggestions that have led to the improvement in the presentation of this paper.

REFERENCES

[1] Camenisch, J. 1998. Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. PhD Thesis. *ETH Series in Information Security and Cryptography*. 2: 33-37.

[2] Chaum, D. 1983. Blind Signatures for Untraceable Payments. *Advances in Cryptology-Proceeding of CRYPTO'82*. Plenum Press. 199-203.

[3] Diffie, W. and M. Hellman. 1976. New Directions in Cryptography. *IEEE Trans. Info. Theory.* 22: 644-654.

[4] ElGamal, T. 1985. A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithm Problem. *IEEE Trans. Info. Theory.* 469-472.

[5] Kim, S., S. Park and D. Won. 1997. Proxy Signatures, Revisited, *Information and Communications Security, Proceedings.* LNCS 1334. Springer-Verlag. 223-232.

[6] Lim, C. H. and P. J. Lee. 1996. Directed Signatures and Applications to Threshold Cryptosystem. *Workshop on Security Protocol.* Cambridge. 131-138.

[7] Okamoto, T. 1994. Designated Confirmer Signatures and Public-Key Encryption Are Equivalent. *Advances in Cryptology-CRYPTO'94*. Springer-Verlag. LNCS 839. 61-74.

[8] Pfitzmann, B. 1991. Fail-Stop Signatures: *Principles and Applications, Proc. Compsec'91, 8th Word Conference on Computer Science, Audit and Control*. 125-134.

[9] Schneier, B. 1996. *Applied Cryptography*. 2nd Edition. John-Wiley & Sons.