New One-Time Proxy Signature Scheme based on DLP using the Warrant

Young-Seol Kim†, Jik Hyun Chang†

†Department of Computer Science and Engineering, Sogang University, Seoul, Korea

Summary
The one-time proxy signature scheme is a kind of digital signature scheme. Using the signature scheme, the original signer can delegate his/her signing capability to the proxy signer and the proxy signer can sign the message only once. In this paper, we propose a new one-time proxy signature scheme based on Discrete Logarithm Problem(DLP). The proposed scheme uses Wang's basic proxy signature and the warrant. Because our one-time proxy signature scheme satisfies all security properties, the scheme is secure. Furthermore, we extend our scheme for multiple signing.

Key words:
Digital signature, Proxy signature, One-Time Signature, Public key cryptography

1. Introduction

In 1996, Mambo, Usuda, and Okamoto proposed a new concept, called proxy signature [4], [5]. In a proxy signature scheme, one user Alice, called the original signer, delegates her signing capability to another user Bob, called the proxy signer, and Bob can sign messages in behalf of Alice. A verifier can validate its correctness and can distinguish between a normal signature and a proxy signature. As a result, the verifier can be convinced of the original signer's agreement on the signed message. Proxy signature schemes have been suggested for use in a number of applications, including electronic commerce, mobile agents, and distributed shared object systems, etc. In an example, the president of a company delegates a signing right to his/her secretary before a vacation. The secretary can make a signature in behalf of the president and a verifier is confident that the signature has been made by an authorized secretary, and the verifier can be convinced of the president's agreement on the signed message. Typically, a proxy signature scheme proceeds as follows. The original signer Alice sends the proxy signer Bob a signature that is associated with a specific message. Bob makes a proxy secret key using this information. Bob can then sign on a message with the proxy secret key using a normal signature scheme. After this message and signature is sent to the verifier, he/she recovers a proxy public key using public information and verifies the proxy signature using a normal signature scheme.

One-time signature schemes were proposed by Rabin [12] and Lamport [13]. They are based on the idea of committing public keys to private keys using one-way hash function. A user can sign a message only once by a public, private key pair. If the user wants to sign another message, the user has to have another public, private key pair.

One-time proxy signatures are one-time signatures for which the original signer can delegate his/her signing capability to the proxy signer. In 2003, H. Wang and J. Pieprzyk proposed a one-time proxy signature scheme [14] and another one-time proxy signature scheme was proposed by M. Mehta and L. Harn in 2005 [15]. These two schemes are secure, but they are complicated and not efficient. In this paper, we propose a new one-time proxy signature. It is a simple one-time proxy signature scheme that the proxy signer can sign the message once or for a restrictive number of times. The proposed scheme is very simple and efficient. First of all, we propose a basic scheme for signing once. If the proxy signer signs a multiple messages using the same proxy secret key, the proxy secret is known to everyone. It is illegal. Therefore, the proxy signer has to sign only a message using a proxy secret key. Our scheme uses Wang's basic proxy signature scheme [1] and satisfies all security properties. Therefore, the proposed scheme is secure. Our scheme is useful in several applications such as e-payment and e-voting. Furthermore, because our proxy scheme is a one-time signature, it is more useful.

The rest of this paper is organized as follows. Section 2 introduces the computational assumptions, definitions, and notations for a one-time proxy signature scheme. In Section 3, we briefly recall Wang's basic proxy signature. Section 4 presents the new one-time proxy signature scheme based on DLP and Section 5 discusses its security. Finally, the conclusion is presented in Section 6.

2. Preliminaries

2.1 Assumption

Our scheme uses the following known difficult problem for security.

Manuscript received February 5, 2007
Manuscript revised February 25, 2007

Assumption 1: Discrete Logarithm (DL) assumption. Let $G_q =< g >$ be a cyclic multiplicative group generated by g of order q. Then on inputs (g, g^x) where $x \in_R Z_q$ is a random number, there is no probabilistic polynomial-time algorithm that outputs the value of x with a non-negligible probability.

This computational assumption is widely believed to be true for many cyclic groups, such as the multiplicative subgroup $G_q =< g >$ of the finite field Z_p, where p is a large prime and q is a prime factor of $p-1$. In practice, $|p|=1024$ and $|q|=160$ are considered to be suitable for most current security applications.

2.2 Definitions

Definition 1. A **one-time proxy signature scheme** is usually comprised of the following phases:

-Setup: The original signer Alice and the proxy signer Bob generate their private, public key pairs, respectively. These key pairs are used in a normal signature scheme.
-Proxy delegation: The original signer Alice and the proxy signer Bob perform an interactive protocol to generate a proxy private, public key pair (x_p, y_p). x_p is known only to the proxy signer Bob and y_p is public or revocable publicly.
-Proxy signature generation: The proxy signer Bob signs on a message using proxy private key x_p and sends the signature to Cindy.
-Verification: Cindy verifies m and its signature using verification equation.

The security requirements for proxy signature are specified in K. Zhang [10], [11].

Definition 2. A **secure** one-time proxy signature scheme should satisfy the following requirements:

-Unforgeability: Only the designated proxy signer can create the one-time proxy signature(even the original signer cannot do it).
-Undeniability: Neither the original signer nor the proxy signer can sign the message instead of the other party. Both the original signer and the proxy signer cannot deny their signatures against anyone.
-Identifiability: An original signer can determine the proxy signer's identity from a proxy signature.
-Distinguishability: The one-time proxy signature must be distinguishable from the normal signature.
-Verifiability: The one-time proxy signature can be verified by everyone.

2.3 Notations

For the convenience of describing our work, we define the parameters as follows:

- p, q : two large prime numbers, $q \mid p-1$.
- g : is an element of Z_p^*, its order is q.
- x_U, y_U : a participant U's private key and public key, respectively, $y_U = g^{x_U} \bmod p$.
- $H()$: a public cryptographically strong hash function
- $||$: denotes the concatenation of strings
- m_w : the warrant which specifies the delegation period for the kind of message m is delegated, the identities of the signer, etc.
- $Sig_U[m]$: a participant U 's signing behavior on the message m.

3. Related Work

In 2005, Wang proposed a basic proxy signature scheme [1] based on a provably secure two-party Schnorr signature proposed in [7]. In this section, we briefly review Wang's basic proxy signature scheme.

3.1 Wang's Basic Proxy Signature Scheme

It is assumed that the original signer Alice and the proxy signer Bob have agreed on a warrant m_w before generating a proxy key pair for Bob. Wang's proxy signature scheme [1] is as follows:

Proxy Key Generation. To generate a proxy key pair (x_P, y_P) for the proxy signer Bob, Alice and Bob execute the following interactive protocol jointly.

(1) Alice picks a random number $k_A \in_R Z_q^*$, computes $r_A = g^{k_A} \bmod p$ and $c = H(r_A)$, and then sends c to Bob.
(2) Similarly, Bob chooses a random number $k_B \in_R Z_q^*$, computes $r_B = g^{k_B} \bmod p$, and replies to Alice with (c, r_B).
(3) When (c, r_B) is received, Alice checks whether $r_B{}^q = 1 \bmod p$. If the validation holds, she computes

$r_P = r_A r_B \bmod p$, $s_A = k_A + x_A H(m_w, r_P) \bmod q$, and sends the pair (r_A, s_A) to Bob.

(4) Upon receiving (r_A, s_A), Bob first computes $r_P = r_A r_B \bmod p$, and then checks whether $r_A{}^q = 1 \bmod p$, $c = H(r_A)$, and $g^{s_A} = y_A{}^{H(m_w, r_P)} r_A \bmod p$. If all validations pass, he calculates $s_B = k_B + x_B H(m_w, r_P) \bmod q$, and finally sets his proxy key pair (x_P, y_P) by

$$x_P = s_A + s_B \bmod q$$
$$y_P = g^{x_P} \bmod p$$

Proxy Signature Generation. To generate a proxy signature on a message m that conforms to the warrant m_w, the proxy signer Bob performs the same operations as in the standard Schnorr signature scheme [2]. That is, he first selects a random number $k \in_R Z_q^*$, and computes $r = g^k \bmod p$, $s = k + x_P H(m, m_w, r) \bmod q$. The resulting proxy signature on message m is $\sigma = (m_w, r_P, r, s)$.

Proxy Signature Verification. To verify the validity of σ, the verifier Cindy operates as follows:

(1) Check whether the message m conforms to the warrant m_w. If not, stop. Otherwise, continue.

(2) Check whether Alice and Bob are specified as the original signer and the proxy signer in the warrant m_w, respectively.

(3) Recover the proxy public key y_P by computing:

$$y_P = (y_A y_B)^{H(m_w, r_P)} r_P \bmod p$$

(4) Accept the proxy signature $\sigma = (m_w, r_P, r, s)$ if and only if the following equality holds:

$$g^s = y_P{}^{H(m, m_w, r)} r \bmod p$$

4. Description of the One-Time Proxy Signature Scheme based on DLP using the Warrant

We now present new one-time proxy signature scheme based on DLP. Our new scheme is constructed based on Wang's basic proxy signature scheme [1] and uses the warrant.

The original signer Alice and the proxy signer Bob jointly generate a proxy key pair (x_P, y_P) for Bob, and the verifier Cindy can recover the proxy public key y_P for verification. We skip the setup phase.

4.1 Proxy Delegation Phase

The original signer Alice and the proxy signer Bob execute the following interactive protocol jointly.

(1) Alice picks random numbers $k_A, k_1 \in_R Z_q^*$, computes $r_A = g^{k_A} \bmod p$, $r_1 = g^{k_1} \bmod p$ and $c = H(r_A)$, and then sends c to Bob.

(2) Similarly, Bob chooses a random number $k_B \in_R Z_q^*$, computes $r_B = g^{k_B} \bmod p$, and replies to Alice with (c, r_B).

(3) When (c, r_B) is received, Alice checks whether $r_B{}^q = 1 \bmod p$, If the validation holds, she computes $r_P = r_A r_B \bmod p$, $s_A = k_A + x_A H(m_w, r_P) \bmod q$. The warrant m_w is generated by

$$m_o = [\textit{delegation lifetime},\ \textit{Alice's and Bob's identities},\ \textit{the description of message}, r_1],$$
$$m_w = (m_o, Sig_{Alice}[m_o]).$$

And then, she sends the pair (r_A, s_A, k_1, m_w) to Bob by secure manner.

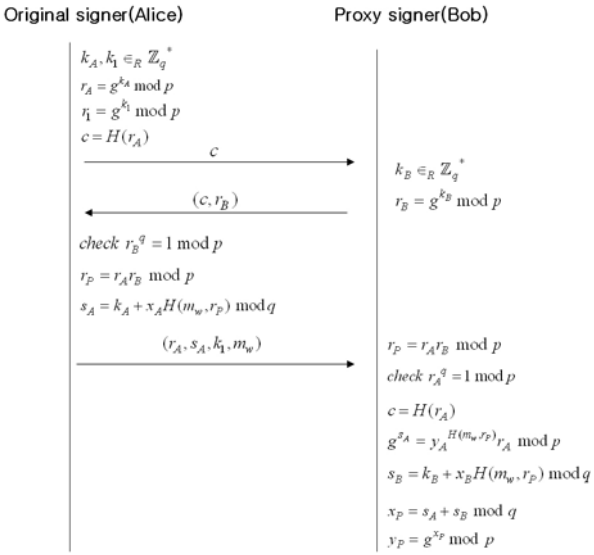

Fig. 1 Proxy delegation phase.

(4) Upon receiving (r_A, s_A, k_1, m_w), Bob first computes $r_P = r_A r_B \bmod p$, and then checks whether $r_A{}^q = 1 \bmod p$, $c = H(r_A)$, and $g^{s_A} = y_A{}^{H(m_w, r_P)} r_A \bmod p$. If all validations hold, he calculates $s_B = k_B + x_B H(m_w, r_P) \bmod q$, and finally sets his proxy key pair (x_P, y_P) by

$$x_P = s_A + s_B \bmod q$$
$$y_P = g^{x_P} \bmod p$$

4.2 Signing Phase

To generate a proxy signature for the verifier Cindy, the proxy signer Bob executes the following operations. Because Bob has to use only r_1, he can sign a message only once. The proxy signer Bob computes $r_1 = g^{k_1} \bmod p$, $s = k_1 + x_P H(m, m_w, r_1) \bmod q$ and sends (m, r_P, m_w, r_1, s) to the verifier Cindy.

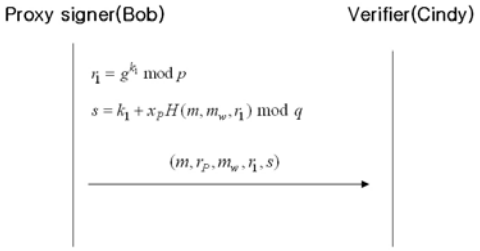

Fig. 2 Signing phase.

4.3 Verification Phase

The verifier Cindy checks Alice's and Bob's identities and delegation lifetime of the warrant m_w. If all validations hold, Cindy follows the next operations. Cindy recovers y_P by $y_P = (y_A y_B)^{H(m_w, r_P)} r_P \bmod p$. Next, Cindy can verify the one-time proxy signature by checking whether

$$g^s = y_P{}^{H(m, m_w, r_1)} r_1 \bmod p \qquad (1)$$

holds. This is because:

$$\begin{aligned} y_P{}^{H(m, m_w, r_1)} r_1 &= g^{x_P H(m, m_w, r_1)} g^{k_1} \bmod p \\ &= g^{x_P H(m, m_w, r_1) + k_1} \bmod p \\ &= g^s \bmod p \end{aligned}$$

Verifier(Cindy)

$y_P = (y_A y_B)^{H(m_w, r_P)} r_P \bmod p$

$check\ g^s = y_P{}^{H(m, m_w, r_1)} r_1 \bmod p$

Fig. 3 Verification phase.

4.4 Extension of the Signing Number of Times

Now, we extend our scheme for multiple signing. If the original signer Alice wants to allow the proxy signer to sign multiple messages, the original signer operates as follows. First of all, the original signer generates the warrant like this in the proxy delegation phase.

$$\begin{aligned} m_o = [&delegation\ lifetime, \\ &Alice's\ and\ Bob's\ identities, \\ &the\ description\ of\ message, \\ &r_1, r_2, r_3, ...], \\ m_w = (m_o, &Sig_{Alice}[m_o]). \end{aligned}$$

In other words, if the original signer wants to allow the proxy signer to sign n messages, the original signer issues $r_1, r_2, r_3, ..., r_n$ to the proxy signer by the warrant and sends $k_1, k_2, k_3, ..., k_n$ to the proxy signer by secure manner. Next, in the signing phase, the proxy signer signs n messages using a pair (k_i, r_i) $i = 1, 2, ..., n$. At this time, the proxy signer has to use a pair (k_i, r_i) $i = 1, 2, ..., n$ for a message. Lastly, in the verification phase, the verifier checks if r_i, $i = 1, 2, ..., n$ is contained in the warrant and verifies the message and its signature

5. Analysis of the Proposed Scheme

Theorem 1. The proposed scheme satisfies the unforgeability property.

Proof. Firstly, we show that the original signer cannot forge the proxy signature without the proxy signer's help. Assume that the original signer Alice can forge proxy signature (m, r_P, m_w, r_1, s). Then, we set c by:

$$c = H(m, m_w, r_1) \bmod q.$$

But, we can get c', another result, using the same input (m, m_w, r_1) and the other hash function H'. In other words,

$$c' = H'(m, m_w, r_1) \bmod q.$$

Therefore, the original signer Alice can forge another signature (m, r_P, m_w, r_1, s') for message m. And, because $g^s = {y_P}^{H(m,m_w,r_1)} r_1 \bmod p$, then

$$g^s {y_P}^{-H(m,m_w,r_1)} = r_1 = g^{s'} {y_P}^{-H'(m,m_w,r_1)} \bmod p.$$

And,

$$g^s {y_P}^{-c} = r_1 = g^{s'} {y_P}^{-c'} \bmod p.$$

Therefore,

$$g^{(s-s')} = {y_P}^{(c-c')} \bmod p$$
$$= g^{x_P(c-c')} \bmod p$$

Consequently,

$$g^{x_P} = g^{\frac{(s-s')}{(c-c')}} \bmod p,$$
$$x_P = \frac{(s-s')}{(c-c')} \bmod q.$$

That is to say, the original signer Alice can compute x_P, the discrete log of y_P. However, we already know that this is infeasible as previously stated in Assumption 1. Therefore, the original signer Alice cannot forge the proxy signature without the proxy signer's help. Also, in case Alice tries to generate $(m, r_P', m_w', r_1', s')$ by replacing proxy private key, x_P', this is also infeasible. When the proxy signature is verified, not only the proxy signer's but also the original signer's public keys are used in the verification equation. Thus the forged proxy signature that is generated without valid proxy private key x_P cannot pass the verification. Also, in this case, the original signer Alice cannot forge the proxy signature.

Secondly, the third party cannot forge the valid proxy signatures as the third party knows less than the original signer Alice. Since the original signer Alice is unable to generate valid proxy signatures, the third party cannot generate valid proxy signatures either. Therefore, the proposed scheme satisfies the unforgeability property.

Theorem 2. The proposed scheme satisfies the undeniability property.

Proof. As to the property of undeniability, it implies that the original and the proxy signers cannot deny having signed the message on behalf of the original signer to any person. In the proposed scheme, when the one-time proxy signature (m, r_P, m_w, r_1, s) is verified, the warrant m_w is checked and the public keys y_A and y_B of the original signer and the proxy signer are used in the verification equation. So the original signer and proxy signer cannot deny having signed the message m on behalf of the original signer to any person. Therefore, the proposed scheme satisfies the undeniability property

Theorem 3. The proposed scheme satisfies the identifiability property.

Proof. If the original signer wants to know the proxy signer's identity, the original signer can check the warrant m_w. It is easy to determine the identity. Therefore, the proposed scheme satisfies the identifiability property.

Theorem 4. The proposed scheme satisfies the distinguishability property.

Proof. In the proposed scheme, when the one-time proxy signature (m, r_P, m_w, r_1, s) is verified, not only the proxy signer's but also the original signer's public keys and identities are used in the verification equation, so we can consider it as a proxy signature and not a normal signature. Thus, anyone can distinguish the one-time proxy signature from normal signatures and the proposed scheme satisfies the distinguishability property.

Theorem 5. The proposed scheme satisfies verifiability property.

Proof. The security property implies that from the proxy signature, a verifier can be convinced of the original signer's agreement to the signed message. In the proposed scheme, on the one hand, from the warrant m_w, the verifier can know who the original signer and the proxy signer are. On the other hand, when the proxy signature is verified, the original signer's and the proxy signer's public keys and identities are used in the verification equation. Thus the original signer cannot deny having delegated his signing capability to the designated proxy signer. That is to say, any verifier can be convinced of the original signer's agreement to the signed message and verify the message and its signature. Therefore, the proposed scheme satisfies the verifiability property.

6. Conclusion

In this paper, we propose a new one-time proxy signature scheme. Our new scheme is simple. We use Wang's basic proxy signature scheme for constructing the scheme and use the warrant for one-time signature. Our new scheme is secure because the scheme satisfies all security properties: Unforgeability, Undeniability, Identifiability, Distinguishability, and Verifiability.

References

[1] G. Wang. Designated-Verifier Proxy Signature Schemes. In: Security and Privacy in the Age of Ubiquitous Computing (IFIP/ SEC 2005), pp. 409-423. Springer, 2005.
[2] C.P. Schnorr. Security of blind discrete log signatures against interactive attacks. In: Information and Communications Security (ICICS 2001), LNCS 2229, pp. 1-12. Springer-Verlag, 2001.
[3] D. Chaum. Blind signatures for untraceable payments. In Crypto'82, pp. 199-203. New York: Plenum Press, 1983.
[4] M. Mambo, K. Usuda, and E. Okamoto. Proxy signatures: Delegation of the power to sign messages. IEICE Trans. Fundamentals, Sep. 1996, Vol. E79-A, No. 9, pp. 1338-1353.
[5] M. Mambo, K. Usuda, E. Okamoto. Proxy signatures for delegating signing operation. In: 3rd ACM Conference on Computer and Communications Security (CCS'96), pp. 48-57. New York: ACM Press, 1996.
[6] Guilin Wang, Feng Bao, Jianying Zhou, and Robert H. Deng. Security Analysis of Some Proxy Signatures. In: Information Security and Cryptology - ICISC 2003, LNCS 2971, pp. 305-319. Springer-Verlag, 2004.
[7] A. Nicolosi, M. Krohn, Y. Dodis, and D. Mazieres. Proactive two-party signatures for user authentication. In: Proceedings of 10th Annual Network and Distributed System Security Symposium (NDSS'03). The Internet Society, 2003.
[8] J.L. Camenisch, J.-M. Piveteau, and M.A. Stadler. Blind signatures based on the discrete logarithm problem. In: Eurocrypt'94, LNC 950, pp. 428-432. Springer-Verlag, 1994
[9] L.-C. Wu, Y.-S. Yeh, T.-S. Liu, Analysis of Sun et al.'s linkability attack on some proxy blind signature schemes, Journal of Systems and Software, 2006
[10] K. Zhang. Threshold proxy signature schemes. In: Information Security (ISW'97), LNCS 1396, pp. 282-290. Berlin: Springer-Verlag, 1997
[11] K. Zhang. Nonrepudiable proxy signature schemes. Manuscript, 1997. Available at http://citeseer.ist.psu.edu/zhang97nonrepudiable.html
[12] M.O. Rabin. Digitalized signatures. Foundations of Secure Communication, Academic Press, 1978, 155-168.
[13] L. Lamport. Password authentication with insecure communication. Communication of the ACM, 24(11), 1981, 770-772.
[14] Huaxion Wang and Josef Pieprzyk. Efficient One-time proxy signatures. In ASIACRYPT 2003, pp. 507--522, 2004, Springer- Verlag
[15] M. Mehta and L. Harn, IEE Proceedings Communications, Vol. 152, No. 2, 2005, pp. 129-134)

Young-Seol Kim received the B.S. and M.S. degrees in Computer Science and Engineering from Sogang University. He is currently a Ph.D. candidate at Sogang University, Korea. His research interests include cryptography and information security.

Jik Hyun Chang received the B.S. and M.S. degrees in Mathematics from Seoul National University, Korea. He received his Ph.D degree in the department of Computer Science & engineering from University of Minnesota, USA. Since 1986, he serves as a professor at Sogang University, Korea. His research interests include algorithms design and analysis, and cryptographic algorithms.