A New Identity-based Proxy Signature Scheme from Bilinear Pairings

Hamid Mala
Department of Electrical & Computer Engineering, Isfahan University of Technology (IUT), Isfahan, Iran
mala@ec.iut.ac.ir

Mohammad Dakhil-alian
Department of Electrical & Computer Engineering, Isfahan University of Technology (IUT), Isfahan, Iran
mdalian@cc.iut.ac.ir

Mehdi Brenjkoub
Department of Electrical & Computer Engineering, Isfahan University of Technology (IUT), Isfahan, Iran
brnjkb@cc.iut.ac.ir

Abstract

Proxy signature schemes allow a proxy signer to generate a proxy signature on behalf of an original signer. In this paper we propose an Identity-based proxy signature scheme from bilinear pairings. In comparison with the Xu et al's scheme, our scheme is more efficient in computation and requires fewer pairing operations especially in verification phase.

1. Introduction

In a certificate-based public key system, before using the public key of a user, the participants must verify the certificate of the user at first. As a consequence, this system requires a large storage and computing time to store and verify each user's public key and the corresponding certificate. In 1984, Shamir introduced the idea of identity(ID)-based public key cryptosystem [1], which enables any pair of users to communicate securely without exchanging public key certificates, without keeping a public key directory, and without using online service of a third party, as long as a trusted key generation center issues a private key to each user when he first joins the network. An identity-based scheme resembles an ideal mail system. If you know somebody's name and address, you can send him a message that only he can read, and you can verify the signatures that only he could have produced. Shamir proposed an identity-based signature scheme in 1984 but invention of an Identity-based encryption scheme last till 2001 which Boneh and Franklin proposed an "Identity-Based Encryption from the Weil Pairing" [2]. Since then, many ID-based crypto primitives have been proposed using bilinear pairings. One of them is proxy signature.

In 1996, Mambo, Usuda, and Okamoto introduced the concept of "proxy signature" [3]. In such a scheme an original signer delegates his signing authority to proxy signer in such a way that the proxy signer can sign any messages on behalf of the original signer. There are three types of delegation: full delegation; partial delegation and delegation by warrant. In the full delegation, the original signer just gives his signing (private) key to the proxy signer as the proxy signing key. Therefore, the signature generated between the original signer and the proxy signer is indistinguishable. In the case of partial delegation, the proxy singing key is derived from the original signer's private key by the original signer. On the other side, it is computational hard for the proxy signer to derive the private key of the original signer. However, the original signer can still forge a proxy signature of the proxy signer. In the delegation by warrant, the original signer signs a warrant that certifies the legitimacy of the proxy signer. Proxy signatures have found numerous practical applications, particularly in distributed computing where delegation of rights is quite common.

Bilinear pairings have attractive properties; consist of "bilinearity", "Non-degeneracy" and "Computability" that have made them suitable for cryptographic applications. The Weil and Tate pairings have recently been used to construct cryptosystems, such as signature schemes of Sakai, Ohgishi and Kasahara [4], the tripartite Diffie-Hellman protocol of Joux [5], the identity-based encryption scheme of Boneh and Franklin [2], the short signature scheme of Boneh et al [6], the ID-based key exchange system of Smart [7] and the ID-based signature scheme of Hess [8].

In this paper we propose an Identity-based proxy signature scheme from bilinear pairings. In our scheme delegation is done by warrant. As compared with the Xu-Zhang-Feng scheme [9], our scheme is more efficient in computation and requires fewer pairing operations especially in verification phase.

From security aspects our scheme provides properties that a strong proxy signature scheme should have, defined by Lee *et al:* [10].

1. Distinguishability: Proxy signatures are distinguishable from normal signatures by everyone.
2. Verifiability: From the proxy signature, the verifier can be convinced of the original signer's agreement on the signed message.
3. Strong non-forgeability: A designated proxy signer can create a valid proxy signature for the original signer. But the original signer and other third parties who are not designated as a proxy signer cannot create a valid proxy signature.
4. Strong identifiability: Anyone can determine the identity of the corresponding proxy signer from the proxy signature.
5. Strong non-deniability: Once a proxy signer creates a valid proxy signature of an original signer, he/she cannot repudiate the signature creation.

0-7803-9521-2/06/$20.00 ©2006 IEEE.

6. Prevention of misuse: The proxy signer cannot use the proxy key for other purposes than generating a valid proxy signature. That is, he/she cannot sign messages that have not been authorized by the original signer.

This paper is organized as follows : the bilinear pairing is introduced in section 2, Xu-Zhang-Feng's identity-based proxy signature scheme is reviewed in section 3, our new scheme is proposed in section 4 and the efficiency and security analysis is given in section 5 ;finally we draw the conclusion.

2. Preliminaries

In this section, we briefly review some preliminaries that will be used throughout this paper.

2.1. Bilinear Pairings

Let G_1 be an additive cyclic group of prime order q, generated by P ; and G_2 be a multiplicative cyclic group of the same order. As mentioned in [2] G_1 can be considered as a subgroup of points on an elliptic curve or hyper elliptic curve over a finite field. A bilinear pairing is a map $e:G_1 \times G_1 \to G_2$ with the following properties:

1) Bilinear: for all $P,Q \in G_1$ and $a,b \in Z_q^*$, $e(aP,bQ) = e(bP,aQ) = e(P,Q)^{ab}$.

2) Non-degenerate: there exist a $Q \in G_1$ such that $e(Q,Q) \neq 1$.

3) Computable : given $P,Q \in G_1$ there is an efficient algorithm to compute $e(P,Q)$ in polynomial time.

Such bilinear pairing has been successfully realized on certain elliptic curves, such as the modified Weil pairing and Tate pairing [2].

2.2. Complexity assumptions

Let G_1 be an additive cyclic group generated by P with the prime order q . Assume that the inversion and multiplication in G_1 can be computed efficiently. Following problems are introduced in G_1 . We mean $a \in_R G$, to choose an element in group G at random.

1) Discrete Logarithm Problem (DLP): given two elements $P,Q \in G_1$, find an integer $n \in Z_q^*$ such that $Q = nP$ whenever such an integer exists.

2) Computational Diffie-Hellman Problem (CDHP): given P, aP, bP for some $a,b \in_R Z_q^*$, compute abP .

3) Decisional Diffie-Hellman Problem (DDHP): given P, aP, bP, cP for $a,b,c \in_R Z_q^*$, decide whether $c = ab \bmod q$.

4) Bilinear Pairing Inversion Problem (BPIP) : given $P \in G_1$ and $e(P,Q) \in G_2$, find $Q \in G_1$.

As specified in [2], the decisional diffie-hellman problem in G_1 should be easy. The DDHP in G_2 , the CDHP and DLP in both G_1 and G_2 should be hard. Also the BPIP should be hard. The group G_1 with these conditions is called a Gap Diffie-Hellman (GDH) group.

3. Review of Xu-Zhang-Feng's proxy signature scheme

The scheme uses SOK-IBS[1] as its identity-based plain signature [4]. The scheme consists of following algorithms:

Setup: Assume k is the security parameter of the system. Let G_1 be a GDH group of prime order $q > 2^k$ generated by P , G_2 be a multiplicative cyclic group of the same order, and $e:G_1 \times G_1 \to G_2$ be a bilinear map. Private Key Generator (PKG) picks a random master key $s \in_R Z_q^*$ and sets $P_{pub} = sP$. Then he chooses hash functions $H_1, H_2, H_3 :\{0,1\}^* \to G_1$, and hash function $H_4 :\{0,1\}^* \to Z_q^*$. Then he publishes parameters of the system: $params = \{q, G_1, G_2, e, P, H_1, H_2, H_3, H_4, P_{pub}\}$

Key Extract : given a user's identity ID , PKG computes $Q_{ID} = H_1(ID) \in G_1$ and the associated private key $S_{ID} = sQ_{ID} \in G_1$.

Sign : given the private key S_d of original designator (signer) ID_d , in order to sign the message m_w :

1. Randomly pick $r_d \in_R Z_q^*$ and compute $U_d = r_d P \in G_1$ and then put $H_d = H_2(ID_d, m_w, U_d) \in G_1$.

2. Compute $V_d = S_{ID_d} + r_d H_d \in G_1$.

The signature on m_w is the warrant $w = < U_d, V_d >$.

Verify : to verify a signature $w = < U_d, V_d >$ on a message m_w for the identity ID_d , the verifier computes $Q_{ID_d} = H_1(ID_d)$ and $H_d = H_2(ID_d, m_w, U_d)$ firstly. He then accepts the signature if and only if $e(P, V_d) = e(P_{pub}, Q_{ID_d}) e(U_d, H_d)$.

Proxy designation : in order to designate user ID_p as a proxy signer, the original signer sends user ID_p a message m_w and corresponding warrant w . The user ID_p verifies this signature w and if it is valid he computes a proxy signing key as : $skp = H_4(ID_d, ID_p, m_w, U_w) S_{ID_p} + V_d$

[1] Sakai-Ogishi-Kasahara Identity Based Signature

Proxy signing: given proxy signing key skp, proxy signer signs a message m on behalf of user ID_d as follows:

1. Randomly picks $r_p \in Z_q^*$ and computes $U_p = r_p P \in G_1$ and then puts $H_p = H_2(ID_p, m, U_p)$.
2. Computes $V_p = skp + r_p H_p \in G_1$.

The proxy signature on message m on behalf of user ID_d produced by user ID_p is $psig = (m_w, ID_p, U_d, U_p, V_p)$.

Proxy verification : The verifier first takes $Q_{ID_d} = H_1(ID_d) \in G_1$, $Q_{ID_p} = H_1(ID_p) \in G_1$, $H_d = H_2(ID_d, m_w, U_d)$ and $H_p = H_2(ID_p, m, U_p)$. He then accepts the signature if

$$e(P, V_p) = e(P_{pub}, Q_{ID_d})^{H_4(ID_d, ID_p, m_w, U_d)} \times e(P_{pub}, Q_{ID_p}) \times e(U_p, H_p) \times e(U_d, H_d)$$

4. Review of Hess's signature scheme

This scheme consists of four algorithms, setup, key extract, sign and verify. Setup and key extract algorithms are the same as Xu *et al*'s except that instead of hash functions H_2, H_3 and H_4 we define $H : \{0,1\}^* \to Z_q^*$. The two latter algorithms are defined as follows.

Sign: a user with identity ID and public/private key pair Q_{ID} / S_{ID} signs a message m in the following steps:

1. Randomly pick $k \in_R Z_q^*$ and compute $r = e(P, P)^k$, $c = H(m, r)$.
2. Computes $U = cS_{ID} + kP$.

The signature on m will be $< c, U >$.

Verify: to verify a signature $< c, U >$ on a message m for the identity ID, the verifier

1. Firstly computes $Q_{ID} = H_1(ID)$ and $r' = e(U, P)e(Q_{ID}, P_{pub})$ which if the signature is valid, should be equal to r.
2. He then accepts the signature if and only if $c = H(m, r')$.

The signature and verification algorithms are consistent, because from bilinearity of the pairing map e we have:

$$\begin{aligned} r' &= e(U, P)e(Q_{ID}, P_{pub})^{-c} \\ &= e(cS_{ID} + kP, P)e(-csQ_{ID}, P) \\ &= e(cS_{ID} + kP - cS_{ID}, P) \\ &= e(kP, P) = e(P, P)^k = r \end{aligned}$$

5. Our proposed scheme

Although Xu *et al*'s identity-based proxy signature provides all the security requirements defined in section 1, but from efficiency viewpoint it doesn't have any basic difference with the most natural proxy signature scheme which follows: "the designator arranges a warrant consist of proxy's name and conditions of the proxy and signs this warrant. Whenever the proxy signer wants to sign a message on behalf of the designator, attach the signed warrant to his signed message and sends them to the verifier. The verifier first verifies the signature of the designator on the warrant and then if it is valid, verifies the signature on message m with the proxy's public key whose identity is mentioned in the warrant." In this scenario and using SOK-IBS whose verification needs one hash evaluation and three pairing evaluation, we will need 2 hash and 6 pairing evaluation. While verification in Xu *et al*'s proxy signature scheme preserves only one pairing evaluation and still needs 2 hash and 5 pairing evaluations. The only difference between this scenario and Xu *et al*'s proxy signature is that in the latter, proxy signing key is different from proxy's private key. We propose a more efficient proxy signature scheme based on Hess's signature scheme.

In Our ID-based proxy signature original signer uses Hess's signature scheme to sign the warrant for the proxy. Having verified the signed warrant, proxy signer uses one of its part and his private key to form the proxy key. Then he uses proxy key in a mathematically attractive way to sign a message on behalf of the original signer. Verifier can verify the signature just by two pairing evaluation, two elliptic curve point multiplication, one hash computing and two point addition. The complete description of the scheme is given as a set of sequential algorithms:

Setup: Let G_1 be a GDH group of prime order q generated by P, G_2 be a multiplicative cyclic group of the same order, and $e : G_1 \times G_1 \to G_2$ be a bilinear map. PKG picks a random master key $s \in_R Z_q^*$ and sets $P_{pub} = sP$. Then he chooses hash functions $H_1 : \{0,1\}^* \to G_1$, and hash function $H : \{0,1\}^* \to Z_q^*$. Then he publishes parameters of the system: $params = \{q, G_1, G_2, e, P, H_1, H_2, H_3, H_4, P_{pub}\}$

Key Extract: given a user's identity ID, PKG computes $Q_{ID} = H_1(ID) \in G_1$ and the associated private key $S_{ID} = sQ_{ID} \in G_1$.

Sign : given the private key S_d of original signer ID_d, in order to sign the message m_w He uses Hess's signature:

1. Randomly picks $k_d \in_R Z_q^*$ and computes $r_d = e(P, P)^{k_d}$ and $c_d = H(m_w, r_d)$.
2. Computes $U_d = c_d S_d + k_d P \in G_1$.

The signature on m_w is the warrant $w =< c_d, U_d >$.

Verify: to verify a signature $< c_d, U_d >$ on a message m_w for the identity ID_d, the verifier

1. Firstly computes $Q_{IDd} = H_1(ID_d)$ and $r' = e(U,P)e(Q_{ID}, P_{pub})$.

2. He then accepts the signature if and only if $c_d = H(m_w, r')$.

Proxy designation : in order to designate user ID_p as a proxy signer, the original signer sends user ID_p a message m_w and corresponding warrant w. The user ID_p verifies this signature w and if it is valid he computes a proxy signing key using his private key S_p and the first element of the warrant:

$$skp = c_d S_p \qquad (1)$$

Proxy signing: given proxy signing key skp, proxy signer signs a message m on behalf of user ID_d as follows:

1. Randomly picks $k_p \in Z_q^*$ and computes $r_p = e(P,P)^{k_p}$ and then puts $c_p = H(m, r_p r_d)$.

2. Computes $U_p = c_p.skp + k_p P$.

The proxy signature on message m on behalf of user ID_d produced by user ID_p is announced as:

$$psig = (m_w, ID_p, ID_d, U_d, U_p, c_p, c_d) \qquad (2)$$

Proxy verification: The verifier first takes $Q_{ID_d} = H_1(ID_d) \in G_1$, $Q_{ID_p} = H_1(ID_p) \in G_1$, and then by calculating two pairing operation can obtain :

$$r' = e(Up + U_d, P)e(Q_d + c_p Q_p, P_{pub})^{-c_d} \qquad (3)$$

He then accepts the signature as a valid proxy signature from user ID_p on behalf of user ID_d if and only if equation (4) is hold.

$$c_p = H(m, r') \qquad (4)$$

5.1. Correctness

The signature and verification algorithms are consistent, because from bilinearity of the pairing map e we have:

$$\begin{aligned} r' &= e(U_p + U_d, P)e(Q_d + c_p Q_p, P_{pub})^{-c_d} \\ &= e(c_p.skp + k_p P + c_d S_d + k_d P, P)e(S_d + c_p d_p, P)^{-c_d} \\ &= e(c_p.c_d S_p + k_p P + c_d S_d + kP, P)e(-c_d S_d - c_d c_p S_p, P) \\ &= e(k_p P + k_d P, P)e(0,P) \\ &= e(P,P)^{k_d + k_p}.1 \\ &= r_d.r_p \end{aligned} \qquad (5)$$

6. Security and Efficiency analysis

The identity based proxy signature we proposed is more efficient than Xu *et al*'s scheme, especially in proxy verification p phase. We can divide a proxy signature into four phases: "phase (1), signing the proxy and issuing the warrant", "phase (2) warrant verification and proxy signing key generation", "phase (3) proxy signature generation" and "phase (4) final verification". Table (1) gives a complete comparison between our proxy signature scheme and Xu *et al*'s one in their four phases.

Table (1): Efficiency comparison

scheme	Xu *et al*'s	Proposed
phase (1)	$2M_{G_1} + H + A_{G_1}$	$3M_{G_1} + H$ $+ A_{G_1} + e$
phase (2)	$3P + M_{G_1}$ $+ 2H + A_{G_1}$	$2P + 2M_{G_1}$ $+ H + e$
phase (3)	$2M_{G_1} + H + A_{G_1}$	$3M_{G_1} + H$ $+ A_{G_1} + e$
phase (4)	$5P + M_{G_1}$ $+ 2H + e$	$2P + 2M_{G_1} + H$ $+ 2A_{G_1} + e$

In this table M_{G_1} and A_{G_1} mean scalar multiplication and Addition in group G_1 respectively , H is a hash function evaluation whose output is an elliptic curve point, P is a pairing operation which is the most time-consuming operation and e is exponentiation in Z_q^*. Other computation costs are negligible. Notice that it is unnecessary to do a pairing operation to compute r_p or r_d each time we generate a signature, because $e(P,P)$ can be recomputed and then with an exponentiation in G_2, r_p or r_d is computed.

Xu *et al* propose a security proof for their scheme. Their proof has been done under the random oracle model (The random oracle model means that underlying hash functions used in the scheme are assumed to be ideal random functions [11]) and we now that security in this model can not be a good support for the whole security of the scheme [12]. Security requirements mentioned in section 1, distinguish ability, verifiability, strong identifiably, strong non-deniability and prevention of misuse are achieved in our scheme obviously. We show that our scheme provides "strong non-forgeability" property too.

6.1. Achievement of strong non-forgeability

It is obvious that the original signer has more facilities than the other users to forge a proxy signature from his proxy signer. We shoe that even the original signer can not forge a proxy's proxy signature. Suppose the designator wants to forge a proxy signature on a message m. The only secrets he doesn't know is the private key of the proxy, S_p and proxy signing key skp. He picks a random $k_p \in Z_q^*$ and computes r_p and c_p afterwards. Now he must find a U_p such that $r_p.r_d$

is equal to r' from equation (3). To find such a U_p he should solve the equation

$e(U_p, P) = a$

for U_p, in which

$$a = e(U_d, P)^{-1} e(Q_d + c_p Q_p, P_{pub})^{c_d} .r_p r_d$$

Which is a BPIP. So assuming BPIP is a NP-complete problem, our identity based proxy signature scheme is strongly nonforgabe even for the designator signer.

7. Conclusion

In this paper, we proposed a new identity-based proxy signature scheme that is based on Hess's ID-based signature scheme and has more efficiency than Xu *et al*'s scheme. Our scheme provides all the six security requirements of a proxy signature.

8. References

[1] A. Shamir, Identity-based Cryptosystems and *Signature Schemes*, Proceedings of CRYPTO'84, LNCS 196, pages 47-53, Springer-Verlag, 1984.

[2] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of CRYPTO 2001, LNCS 2139, pages 213{229, Springer-Verlag, 2001.

[3] M. Mambo, K. Usuda and E. Okamoto, "*Proxy signatures for delegating signing operation*," Proc. 3rd ACM Conference on Computer and Communications Security, ACM Press, pp.48-57, 1996.

[4] R. Sakai, K. Ohgishi, M. Kasahara, Cryptosystems based on pairing, In The 2000 Sympoium on Cryptography and Information Security, Okinawa, Japan, January 2000.

[5] A. Joux, One Round Protocol for Tripartite *Diffie-Hellman*, Algorithmic Number Theory Symposium { Proceedings of ANTS 2002, LNCS 1838, pages 385{394, Springer-Verlag, 2000.

[6] D. Boneh, B. Lynn, and H. Shacham, *Short* Signatures from the Weil Pairing, Advances in Cryptology - Proceedings of ASIACRYPT 2001, LNCS 2248, pages 566-582, Springer-Verlag, 2001.

[7]. N.P. Smart. *An Identity based authenticated Key greement protocol based on the Weil Pairing*. Cryptology ePrint rchive, Report 2001/111, 2001. http://eprint.iacr.org/.

[8] F. Hess, Efficient Identity Based Signature Schemes ased on Pairings, Selected Areas in Cryptography { Proceedings of SAC 2002, LNCS 2595, pages 310-324, Springer-Verlag, 2002.

[9] J. Xu, Z. Zhang, D. Feng. ID-Based Proxy Signature Using Bilinear Pairings, available at http://eprint.iacr.org/2004/206/

[10] B. Lee, H. Kim and K. Kim, *Secure mobile agent using strong non-designated proxy signature*, Proc. of ACISP2001, LNCS 2119, pp.474-486, Springer Verlag, 2001.

[11] M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing E±cient Protocols*, Proceedings of the First ACM Conference on Computer and Communications Security 1993, pages 62-73.

[12] R. Canetti, O. Goldreich and S. Halevi, *The Random Oracle Methodology, Revisited* , Proceedings of 30th Annual ACM Symposium on the Theory of Computing, pages 209-218, May 1998, ACM