Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A New Efficient Strong Proxy Signcryption Scheme Based on a Combination of Hard Problems

Hassan Elkamchouchi
Faculty of Engineering
Alexandria University
Egypt
helkamchouchi@yahoo.com

Mohammed Nasr, Roayat Ismail
Faculty of Engineering
Tanta University
Egypt
mnasr@yahoo.com, roaayat@yahoo.com

***Abstract*—Proxy signcryption scheme is the combination of proxy signature and encryption. In this paper we propose a new proxy signcryption scheme which has the following advantages as compared to the previous schemes: (1) It is based on a combination of hard problems: Integer Factorization Problem (IFP), Discrete Logarithm Problem (DLP), Diffie-Hellman Problem (DHP), and irreversibility of a One-Way Hash Function (OWHF). This combination achieves a strong security level. (2) It provides non-repudiation and public verifiability for both the original signcrypter and the proxy signcrypter. (3) It provides forward secrecy. (4) It provides proxy-protection against the original signcrypter. (5) It doesn't require a secure channel between the original signcrypter and the proxy signcrypter. (6) With all these advantages the proposed scheme is efficient.**

***Keywords*—proxy signcryption, forward secrecy, public verifiability, integer factorization, discrete logarithm.**

I. INTRODUCTION

Message security and sender's authentication for communication in the open channel is a basic and important technology of internet. For keeping message confidential and unforged, the sender uses a digital signature algorithm with his private key to sign the message, and encrypts the message and digital signature using a symmetric encryption algorithm . We call this two-step approach "signature-then-encryption". With the current standard signature then encryption approach, the cost for delivering a message in a secure and authenticated way is essentially the sum of the cost for digital signature and that for encryption. Signcryption is a cryptographic primitive first proposed by Zheng [1] to combine the functionality of a digital signature scheme with that of an encryption scheme. A digital signature on an electronic document represents a handwritten signature on a paper document. A problem arises when an original signer is absent. This person may wish to delegate the power of signing to a designated person called proxy signer. The proxy signature is an important technique of modern cryptography. The concept of the proxy signature was first introduced by Mambo et al. in 1996 [2]. They classified proxy signatures based on delegation type as full delegation, partial delegation, and delegation by warrant. Partial delegation is further classified as proxy-unprotected and proxy-protected according to protection of proxy signer. Kim et. al. [3] gave a new type of delegation called partial delegation with warrant. In 1999, Gamage et al. [4] extended the proxy signature and introduced a proxy signcryption scheme by combining proxy signature and encryption technology. It allows an entity to delegate his authority of signcryption to a trusted agent. Gamage's scheme is based on Discrete Logarithm Problem (DLP). But it is desirable to design a proxy signcryption scheme based on other problem, such as Integer Factorization Problem (IFP).

In this paper we propose a new proxy signcryption scheme which has a strong security level as it is based on the intractability of a combination of hard problems: IFP, DLP, DHP, and irreversibility of a OWHF. The scheme is publicly verifiable and forward secure. It provides non-repudiation for both the original signcrypter and the proxy signcrypter and achieves proxy signcrypter protection against original signcrypter impersonation. Our scheme achieves all these advantages with keeping a better efficiency than the previous schemes as Zhang and Dong's proxy signcryption scheme [5] which achieved forward secrecy and public verifiability but at the cost of efficiency. The paper is organized as follows: section II introduces the security properties of the proposed scheme; section III introduces some preliminaries to the proposed scheme; section IV discusses the proposed scheme; Finally, comparison, security analysis are given in section V.

II. SECURITY PROPERTIES

Our scheme is a cryptographic primitive involving four entities: a system authority (SA), an original signcrypter (O), a proxy signcrypter (P), and a receiver (R). The scheme achieves all the following strong proxy signcryption scheme properties:

1. *Correctness*: A properly formed signcrypted ciphertext by the signcryption algorithm must be accepted by the unsigncryption algorithm .
2. *Confidentiality*: Without the knowledge of the sender or the designated receiver's private key, it should be infeasible for an adaptive attacker to gain any partial information on the contents of the signcrypted ciphertext.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

3. *Distinguishability*: The proxy signature must be distinguishable from the normal signature.
4. *Strong unforgeability:* A designated signer, called proxy signer, can create a valid proxy signature for the original signer. But the original signer and third parties who are not designated as a proxy signer cannot create a valid proxy signature
5. *Strong identifiability:* Anyone can determine the identity of the corresponding proxy signer from a proxy signature.
6. *Verifiability:* Validity of a proxy signature as well as the original signer's delegation on signature signing on a particular message can be verified using public parameters.
7. *Non-repudiation of proxy signing:* It is difficult for a proxy signer to falsely deny having signed its proxy signatures.
8. *Non-repudiation of signature delegation:* It is difficult for an original signer to falsely deny that it has delegated the signing power to a proxy signer.
9. *Prevention of proxy key misuse:* It should be confident that proxy key pair should be used only for creating proxy signature, which conforms to delegation information. In case of any misuse of proxy key pair, the responsibility of proxy signer should be determined explicitly.

III. PRELIMINARIES

Our scheme is based on the following hard problems:

Definition 1.Integer Factorization Problem (IFP): given a positive integer n, find its prime factorization; that is, write $n = p_1^{e_1} p_2^{e_2} \ldots . p_k^{e_k}$ where the p_i are pair wise distinct primes and each $e_i \geq 1$ [6].

Definition 2. Discrete Logarithm Problem (DLP): given a prime p, a generator α of Z_p^* and an element $\beta \in Z_p^*$, find the integer x, $0 \leq x \leq p-2$, such that $\alpha^x = \beta \bmod p$ [6]

Definition 3. Diffie-Hellman Problem (DHP): given a prime p, a generator α of Z_p^*, an element $\alpha^a \bmod p$ and $\alpha^b \bmod p$ and, find $\alpha^{ab} \bmod p$ [6].

Definition 4. Irreversibility of a One-Way Hash Function: given n and an input M, computing $H(M) = n$ must be easy and given n, it is hard to compute M such that $H(M) = n$ [6].

In our scheme we will deal with the group of units of Z_n, namely Z_n^*, where $n = p` . q`$ and $p`$ and $q`$ are primes. In other words the DLP and DHP in Z_n^*.

Fact 1: Let n be a composite integer. The DLP in Z_n^* polytime reduces to the combination of the IFP of n and the DLP in Z_p^* for each prime factor of n [6].

This means that finding the integer x, such that $\alpha^x = \beta \bmod n$, where $n = p` . q`$ is equivalent to three hard problems: factorization of n into $p`$ and $q`$, the DLP: $\alpha^x = \beta \bmod p`$ and the DLP: $\alpha^x = \beta \bmod q`$.

Fact 2: The DHP in Z_n^* is at least as difficult as the problem of factoring n [6].

IV. PROPOSED SCHEME

Our scheme consists of seven phases: setup phase, key generation phase, proxy key generation phase, proxy key verification phase, signcryption phase, unsigncryption phase, and public verification phase.

A. Setup Phase

There exists a system authority (SA) whose tasks are to initialize the system and to manage the public directory.SA do the following:

1. Choose a composite modulus $n = p` . q` = (2fp + 1)(2fq + 1)$ where $p`$, $q`$, p, q and f are prime (this form is similar to the form used by Girault in [7].)
2. Keep p and q secret, while publish n and f .
3. Select and publish the following parameters:
   g : denotes an element of Z_n^* of order f.
   $h(.)$: denotes a one-way hash function, e.g. SHA-2 [8].
   m_w : denotes a warrant, which records the identity of the original signcrypter(O), the proxy signcrypter (P), and the valid delegation time, ect.
   $E_k(.)$ and $D_k(.)$: are the symmetric encryption and decryption using a symmetric key k, e.g. AES [9].

B. Key Generation Phase

The keys of each user are as following:

x_O : The original signcrypter's secret key which is a random number in the interval $[1,\ldots,f-1]$.

y_O : The original signcrypter's public key:

$$y_O \equiv g^{x_O} \bmod n. \quad (1)$$

x_P :The proxy signcrypter's secret key which is a random number in the interval $[1,\ldots,f-1]$.

y_P : The proxy signcrypter's public key :

$$y_P \equiv g^{x_P} \bmod n. \quad (2)$$

x_R : The receiver's secret key which is a random number in the interval $[1,\ldots,f-1]$.

y_R : The receiver's public key :

$$y_R \equiv g^{x_R} \bmod n. \quad (3)$$

K_{SH} : A shared-secret key between the proxy signcrypter P and the receiver R computed using Diffie-Hellman key exchange method [6] as following:
by the proxy signcrypter, P, as:

$$K_{SH} \equiv y_R^{x_P} \bmod n \equiv (g^{x_R})^{x_P} \bmod n \equiv g^{x_R x_P} \bmod n. \quad (4)$$

and by the receiver, R , as:

$$K_{SH} \equiv y_P^{x_R} \bmod n \equiv (g^{x_P})^{x_R} \bmod n \equiv g^{x_R x_P} \bmod n. \quad (5)$$

C. Proxy Key Generation Phase

The original signcrypter O performs the following steps to delegate the signcrypting capability to the proxy signcrypter P:

1. Choose a random integer k in the interval [1,……, f–1] and compute:
$$r_o \equiv g^k \bmod n. \quad (6)$$
2. Compute the proxy signature key as:
$$S_o \equiv (k + x_O\, h(m_w \| r_o)) \bmod f. \quad (7)$$
where || refers to the concatenation.
3. Send (S_o , r_o) to the proxy signcrypter P via public channel.

D. Proxy Key Verification Phase

The proxy signcrypter P do the following:

1. Accept S_o only if:
$$g^{S_o} \equiv (y_o^{h(m_W \| r_O)} \cdot r_o) \bmod n. \quad (8)$$
2. Compute:
$$x_{Pr} \equiv (S_o + x_P \cdot h(m_w \| r_o) \bmod f. \quad (9)$$
as his proxy signature secret key. Then he computes and publishes the corresponding proxy public key :

$$y_{Pr} \equiv (r_o \cdot (y_P\ y_o)^{h(m_W \| r_O)}) \bmod n$$

$$\equiv (g^k \cdot (g^{x_P} \cdot g^{x_O})^{h(m_W \| r_O)}) \bmod n$$

(substitute for r_o , y_P and y_o from (6), (2) and (1) respectively)

$$\equiv (g^{(k + x_O h(m_W \| r_O)) + x_P \cdot h(m_W \| r_O)}) \bmod n$$

$$\equiv (g^{(S_o + x_P h(m_w \| r_o)}) \bmod n \qquad \text{(from (7))}$$

$$\equiv g^{x_{Pr}} \bmod n. \qquad \text{(from (9))} \quad (11)$$

E. Signcryption Phase

The proxy signcrypter do the following procedures to signcrypt a message m to the receiver R:

1. Choose a random integer x in the interval [1,…, f– 1].
2. Compute:
$$K_1 = h(g^x \bmod n). \quad (12)$$
3. Compute:
$$K_2 = h((g^x \cdot K_{SH}) \bmod n. \quad (13)$$
4. Compute:
$$C = E_{K_2}(m). \quad (14)$$
5. Compute:
$$r = h(m \| K_1). \quad (15)$$
6. Compute:
$$S = (x - x_{Pr} \cdot r) \bmod f. \quad (16)$$
7. Send the signcrypted ciphertext consists of (C, S, r) to the receiver R.

F. Unsigncryption Phase

The receiver R do the following procedures to unsigncrypt the signcrypted ciphertext:

1. Compute:
$$K_1 = h((g^S \cdot y_{Pr}{}^r) \bmod n. \quad (17)$$
2. Compute:
$$K_2 = h((g^S \cdot y_{Pr}{}^r \cdot K_{SH}) \bmod n). \quad (18)$$
3. Compute:
$$m = D_{K_2}(C). \quad (19)$$
4. Compute:
$$r` = h(m \| K_1). \quad (20)$$

The receiver R accepts m if and only if: $r = r`$.

G. Public Verification Phase

If the proxy signcrypter P denies the signature of the message m, the receiver R, after decrypting and verifying the signcrypted ciphertext can prove the dishonesty of the signer by passing (m, r, s) to a third party who can be convinced that it came originally from P by verifying:

$$r = h(m \| h((g^S \cdot y_{Pr}{}^r) \bmod n)). \quad (21)$$

V. Analysis of the Proposed Scheme

A. Security

We will show that our scheme satisfies all the required properties of a strong signcryption scheme mentioned in section II.

1. Correctness

a) The proxy key verification equation (8) can be proved as following:

The right hand side of (8) $= (y_o^{h(m_W \| r_O)} \cdot r_o) \bmod n$

$= (g^{x_O \cdot h(m_W \| r_O)} \cdot g^k) \bmod n$ (from (1) and (6))

$= g^{k + x_O \cdot h(m_W \| r_O)} \bmod n = g^{S_o} \bmod n$ (from (7))

= the left hand side of (8).

b) The message m can be recovered successfully by the receiver R if the proxy signcrypter P produced the signcrypted ciphertext honestly, since :

the right-hand side of (17) is:

$h((g^S \cdot y_{Pr}{}^r) \bmod n$

$= h((g^{x - x_{Pr} \cdot r} \cdot y_{Pr}{}^r) \bmod n)$ (substitute for S from (16))

$= h((g^{x - x_{Pr} \cdot r} \cdot g^{x_{Pr} \cdot r}) \bmod n)$ (substitute for y_{Pr} from (11))

$= h(g^x \bmod n) = K_1$ (from (12))

= the left-hand side of (17)

then using K_{SH} to compute K_2 as in (18) and using K_2 to recover m as in (19).

2. Confidentiality

The only way to decrypt C and obtain the message m is to have the shared-secret key K_{SH}. But it is difficult to obtain this key from the public keys y_P and y_R due to the intractability of DHP mod n. For a passive adversary, the information available is only (C, r, S). From this data he can only obtain $K_1 = h((g^S . y_{Pr}{}^r) \bmod n) = h(g^x \bmod n)$ but he cannot guess the corresponding m. Also it is difficult to obtain the secret value x from K_1 due to the intractability of both reversing OWHF and DLP modulo n which is equivalent to: IFP of n into $p`$ and $q`$, DLP mod $p`$ and DLP mod $q`$. If an intruder intends to reveal the secret parameters x_{Pr} and x from (16). This will be difficult because there are two unknown variables (x_{Pr} , x) in one equation.

Then even if x_{Pr} is revealed, a person cannot compute K_{SH} as we mentioned previously and so cannot decrypt C. Therefore, our scheme provides forward secrecy property with respect to the proxy signcrypter.

3. Distinguishability

This is obvious, because there is a warrant m_w in a valid proxy signature, at the same the proxy signature public key $y_{Pr} = (r_o . (y_P \ y_o)^{h(m_w||r_O)}) \bmod n$ includes original signcrypter's public key y_o and proxy signcrypter's public key y_P. So the proxy signature is easy to be distinguishable from the normal signature.

4. Strong unforgeability

It is easy to see that, the signature used by both the original signcrypter O and the proxy signcrypter P is essentially the same as the Schnorr signature [10] with only some modification. This signature is proven to be secure against extential forgery on adaptive chosen-message attack under the random oracle model assumption.

For a third adversary who wants to forge the proxy signature of the message m for the proxy signcrypter , he must have the original signcrypter's private key x_o and the secret value k which are protected due to the intractability of: IFP of n into $p`$ and $q`$, DLP modulo $p`$ and DLP modulo $q`$. If he tries to obtain either x_o or k from the equation of S_o, this will be difficult because he have only one equation in two unknowns as we mentioned previously.

For the original signcrypter, he cannot create a valid proxy signature as according to (9), the proxy key includes the private key x_P of the proxy signcrypter which is protected due to the intractability of: IFP of n into $p`$ and $q`$, DLP modulo $p`$ and DLP modulo $q`$. This provides proxy signcrypter protection against the impersonation of the original signcrypter.

From this discussion we find that anyone except the proxy signcrypter, including the original signcrypter, cannot create a valid proxy signature. So our scheme provides strong unforgeability.

5. Strong identifiability

In our scheme the identity information of the proxy signcrypter is included explicity in a valid proxy signature equation ((21)) as a form of the proxy signature public key y_{Pr}. So anyone can determine the identity of the corresponding proxy signcrypter from a proxy signature and therefore our scheme provides strong identifiability.

6. Verifiability

The valid proxy signcryption for message m will be the tuple (C, r, S), and from the construction of r and S and the public verification phase, the verifier can be convinced that the proxy signcrypter has the original signcrypter's signature on the warrant m_w.

In general the warrant contains the identity information and the limit of the delegated signing capacity and so satisfies the verifiability.

7. Non-repudiation of proxy signing

As we indicated previously the proxy signcrypter is the only one who can compute the proxy key pairs.

So once he creates a valid proxy signature, he cannot repudiate it and hence our scheme provides non-repudiation of proxy signing.

8. Non-repudiation of signature delegation

Since the original signcrypter is the only one who can compute S_o as his private key x_o is included in it, he cannot repudiate the signing capability delegation to the proxy signcrypter and this can be verified publicly as we indicated previously. So our scheme provides non-repudiation of signature delegation.

We can say that as the original signcrypter doesn't obtain the proxy signcrypter's private x_P and the proxy signcrypter doesn't obtain the original signcrypter's private key x_o, neither the original signcrypter nor the proxy signcrypter can sign in place of the other party.

9. Prevention of proxy key misuse:

Using the warrant m_w in our scheme had determined the limit of the delegated signing capacity.

So the proxy signcrypter cannot sign some messages that have not been authorized by the original signcrypter and this prevent abuse of the proxy key.

In addition to all these properties, our scheme needs no secure channel for the delivery of the signed warrant.

More precisely, the original signcrypter can send (r_o, S_o) to the proxy signcrypter through a public channel, in other word, any third adversary can get the original signcrypter's signature on the warrant m_w without being able to forge it, because this is equivalent to forge Schnorr signature which is is intractable in our scheme due to the intractability of both IFP and DLP as we discussed previously.

B. Efficiency

Our scheme is more efficient as compared to the previous schemes as Zhang and Dong's proxy signcryption scheme [5] which achieved forward secrecy and public verifiability but at the cost of efficiency.

The detailed costs in each phase are compared in Table I. To simplify the estimation of computational costs, we count only the major operation. For example, the computational cost of modular multiplication, hash function, symmetric key encryption and decryption and hash function is ignored as compared with the expensive costs of modular exponentiation.

TABLE I. NUMBER OF MODULAR EXPONENTIATIONS

Phase	Our scheme	Zhang and Dong's scheme
Proxy key generation phase	1	1
Proxy key verification phase	2	3
Signcryption phase	1	2
Unsigncryption phase	2	3
Public verification phase	2	2
Total	8	11

VI. CONCLUSIONS

In this paper, we have presented a new proxy signcryption scheme which is based on the intractability of a combination of hard problems: IFP, DLP, DHP and reversing a OWHF. This combination of hard problem enhances the security level of the algorithm

Our scheme achieves all the properties of a strong proxy signcryption scheme: correctness, confidentiality with forward secrecy, distinguishability, strong unforgeability which achieves proxy signcrypter protection against the original signcrypter impersonation, strong identifiability, verifiability, non-repudiation of both the original signcrypter and the proxy signcrypter and prevention of proxy key misuse.

Also our scheme requires no secure channel between the original signcrypter and the proxy signcrypter for the delivery of signcryption capability.

With all these advantages our scheme has a better efficiency than the previous schemes which provide forward secrecy and public verifiability as Zhang and Dong's scheme.

REFERENCES

[1] Y. Zheng, "Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost (Signature) + Cost (Encryption)", Advances in Cryptology – CRYPTO' 97, volume 1294 of Lecture Notes in Computer Science, Springer-Verlag, 1997, pages 165-179.

[2] M. Mambo, K. Usuda, E. Okamoto, "Proxy signatures: delegation of the power to sign message", IEICE Transactions on Fundamentals E79-A(9), 1338-1354, 1996.

[3] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited", In Pro. Of ICICS 97, LNCS 1334, Springer-verlag, pages 223-232, 1997.

[4] G. Gamage, J. Leiwo, and Y. Zheng, "An efficient scheme for secure message transmission using proxy signcryption", Technical report 98-01, Monash University, 1998.

[5] Z. Zhang, Q. Dong, and M. Cai, "A New publicly verifiable proxy signcryption csheme", In Progress on Cryptography, 2004.

[6] A. Menezes, P. Oorschot, S. Vanstone, "Handboock of Applied Cryptography", *CRC Press*, 1997. (reference)

[7] M. Girault, "An identity-based identification scheme based on discrete logarithms modulo a composite number", In Eurocrypt' 90, LNCS 473, pages 482-486, Springer-verlag, Berlin, 1991.

[8] National Institute of Science and Technology, "Secure Hash Standard", USA, Federal Information Processing Standard (FIPS) 180-2, Aug. 2002.

[9] J. Daemen and R. Rijmen, "Rijndael: The Advanced Encryption Standard", Dr. Dobb's Journal, pages 137-139, Mar. 2001 (available via http:\\ www.nist.gov/CryptoToolkit).

[10] C. P. Schnorr, "Efficient Signature Generation by Smart Cards", Journal of Cryptology, volume 4, page 161-174, 1991.