World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

A Secure Proxy Signature Scheme with Fault Tolerance Based on RSA System

H. El-Kamchouchi, Heba Gaber, Fatma Ahmed, Dalia H. El-Kamchouchi

Open Science Index, Computer and Information Engineering Vol:10, No:6, 2016 publications.waset.org/10004632/pdf

***Abstract*—**Due to the rapid growth in modern communication systems, fault tolerance and data security are two important issues in a secure transaction. During the transmission of data between the sender and receiver, errors may occur frequently. Therefore, the sender must re-transmit the data to the receiver in order to correct these errors, which makes the system very feeble. To improve the scalability of the scheme, we present a secure proxy signature scheme with fault tolerance over an efficient and secure authenticated key agreement protocol based on RSA system. Authenticated key agreement protocols have an important role in building a secure communications network between the two parties.

***Keywords*—**Proxy signature, fault tolerance, RSA, key agreement protocol.

I. Introduction

DIGITAL signature schemes with fault tolerance make it possible for error detections and corrections during the processes of data computations and transmissions. Previously, Zhang [1] and Lee and Tsai [2] have respectively proposed two efficient fault-tolerant schemes based on the RSA cryptosystem. Both of them can efficiently check the sender's identity and keep the confidentiality of the transmitted document. Furthermore, they can detect the errors and correct them. However, these schemes have a common weakness in security.

Huifang Xue [3] has improved the mechanism of Lee and Tsai by providing extra security against Chosen Ciphertext Attacks (CCA) using a permutation matrix. If a malicious looks into the message he will find it difficult to understand or calculate checksum/ hash value due to the randomization of permutation matrix.

Proxy signature scheme is an important inquiry in the field of a digital signature. It was first introduced by Mambo et al. [4]. Proxy signature permits an original signer to delegate his signing rights to a proxy signer, and then the proxy signer performs message signing on behalf of the original signer. For example, a director of a company wants to survive for a long trip. He would require a proxy agent, to whom he would delegate his signing capability, and thereafter the proxy agent would sign the documents on behalf of the director. The classification of the proxy signature is dependent on the basis of delegation, namely full delegation, partial delegation and delegation by warrant, and presents a well-organized strategy.

In full delegation, the proxy signer signs document using the same secret key of the original signer given by the original signer. In partial delegation, the proxy key is derived from the secret key of the original signer and hands it over to the proxy signer as a delegation capability. The weaknesses of full delegation and partial delegation are eliminated by partial delegation with warrant. A warrant, explicitly states the signer's identity, delegation period and the qualification of messages on which the proxy signer can sign.

In order to the two parties communicate securely together over an unreliable public network they must authenticate one another and agree on a secret encryption key. To achieve this, key establishment protocols are applied at the beginning of a communication session in order to verify the parties' identities and build a common session key. Authenticated key agreement protocols have an important role in establishing secure communications between the two parties over the open network [5].

This paper addresses a secure and efficient proxy signature scheme with fault tolerance based on the RSA system. The remaining parts of this paper are organized as follows: In Section II, we elaborate security requirements of proxy signature. Next, we discuss the new secure key agreement protocol in Section III. In Section IV, we elaborate the improved of Xue's scheme. In Section V, we proposed our scheme. We analyze the security properties and common attacks of our proposed scheme in Section VI. Finally, in Section VII, we give our conclusion.

II. Security Requirements of Proxy Signature

The security requirements for any proxy signature are first studied in [4] and later were improved in [7], [8]. According to them, a secure proxy signature scheme is expected to satisfy the following five requirements [6]:

1) Verifiability: A verifier can be confident of the original signer's agreement on the signed message from a proxy signature
2) Strong unforgeability: Only the designated proxy signer can generate a valid proxy signature.
3) Strong identifiability: The identity of the proxy signer can be determined by any verifier from a proxy signature.
4) Strong undeniability: The proxy signer cannot repudiate the signature creation against anyone else, once he creates a valid proxy signature on behalf of an original signer.
5) Prevention of misuse: The responsibility of the proxy signer should be determined explicitly if he misuses the proxy key for the purposes other than generating a valid

Prof H. El-Kamchouchi, Dr Fatma Ahmed, and Dr Dalia H. El-Kamchouchi are with the Electrical Engineering Department, University of Alexandria, Egypt (e-mail: helkamchouchi@ieee.org, moonyally@yahoo.com, Daliakamsh@yahoo.com).

Heba Gaber is with the Electrical Engineering Department, Arab Academy for Science and Technology, Egypt, (e-mail: heba.g.mohamed@gmail.com).

Open Science Index, Computer and Information Engineering Vol:10, No:6, 2016 publications.waset.org/10004632/pdf

proxy signature.

III. The New Secure Key Agreement Protocol

The used protocol for authenticated key agreement [5] provides authentication between the two parties A and B before exchanging the session keys. The protocol consists of three phases; The Registration Phase, The Transfer and Substantiation Phase, and The Key Generation Phase. Fig. 1 shows the overall operation of the new protocol.

A		B
$s_1 \equiv (y_B)^{-r_A} \cdot (t_B)^{x_A}$	$\xrightarrow{s_1}$ $\xleftarrow{s_2}$	$s_2 \equiv (y_A)^{-r_B} \cdot (t_A)^{x_B}$
$v_2 = \left((t_B)^{x_A} \cdot s_2\right)^{-\bar{r}_A}$		$v_1 = \left((t_A)^{x_B} \cdot s_1\right)^{-\bar{r}_B}$
$\equiv y_B$		$\equiv y_A$
$K_{AB} \equiv v_2' \cdot {t_B}^{x_A} \cdot {t_B}^{r_A}$		$K_{AB} \equiv v_1' \cdot {t_A}^{x_B} \cdot {t_A}^{r_B}$
$\equiv {y_B}^{r_A} \cdot {t_B}^{x_A} \cdot {t_B}^{r_A}$		$\equiv {y_A}^{r_B} \cdot {t_A}^{x_B} \cdot {t_A}^{r_B}$
$K_{AB} \equiv g^{x_B r_A + x_A r_B + r_A r_B}$		$K_{AB} \equiv g^{x_A r_B + x_B r_A + r_B r_A}$

Fig. 1 Overall operation of the proposed protocol

The system picks short-term private key r_A, r_B, they are random integers $2 \le r_A, r_B < n1$ and $GCD(r, n1) = 1$ $n1 = (p-1)(q-1)$ where p, q are large safe prime numbers normally at least 512 bits. t_A, t_B are short-term public keys where $t_A = g^{r_A} \bmod n$ and $t_B = g^{r_B} \bmod n$, g is a generator of Z_p^* and $n = pq$ long term public key at least 1024 bits. Then the system picks long-term private keys x_A, x_B they are random integer where $2 \le x_A, x_B < n1$ and $GCD(x, n1) = 1$ then, compute long-term public key y_A, y_B where $y_A = g^{x_B} \bmod n$ and $y_B = g^{x_A} \bmod n$. K_{AB} is the shared secret key calculated by the new secure protocol between the two parties A and B.

IV. Proposed Scheme

We propose a secure and efficient proxy digital signature scheme with fault tolerance based on new key agreement protocol RSA system. The proposed scheme is based on Xue's scheme [3]. In the RSA cryptography, each user provides a public key (*e*, *N)* and a secret key *d*, where *N* is the product of two large prime numbers p, q *and* s such that $N = p \times q$, and the public key *e* and secret key *d* must satisfy the equation $d = e^{-1}(p-1)(q-1)$.

A. Initialization

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

- A: the original signer
- P: the proxy signer
- B: the receiver
- p, q : two large prime number
- $(e_A ; d_A)$: secret key of original signer
- $(e_A ; n_A)$: public key of original signer
- $(e_P ; d_P)$: secret key of proxy signer
- $(e_P ; n_P)$: public key of proxy signer
- $(e_B ; d_B)$: secret key of receiver
- $(e_B ; n_B)$: public key of receiver
- n_A, n_P and n_B : is the product of two large safe primes
- $H(\,)$: a secure one-way hash function.
- K_{PB} : shared secret key between P and B
- m_w : a warrant.

B. Proxy Key Generation

The original signer A does the following:

1. Computes $S_A = H(m_w \| e_P)^{d_A} \bmod n_A$.
2. Sends (S_A, m_w) to the proxy signer over a public channel.

C. Proxy Key Verification

The proxy signer P checks whether $H(m_w \| e_P) = {S_A}^{e_A} \bmod n_A$. If it holds, the proxy signer accepts it as a valid proxy key; otherwise, rejects it.

D. Proxy Signature Generation

To sign message X on behalf of the original signer A, the proxy signer P does the following:

Step1. User A sends an n×n message matrix X to user P:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix} \quad (1)$$

Step2. For the message matrix X , the proxy P now constructs an (n+1)×(n+1) matrix X_h as:

$$X_h = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} & X_1 \\ x_{21} & x_{12} & \cdots & x_{2n} & X_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} & X_n \\ X^1 & X^1 & \cdots & X^1 & S_P \end{pmatrix} \quad (2)$$

where, $X_i = \prod_{j=1}^{n} x_{ij} \bmod n_p$, $X^j = \prod_{i=1}^{n} x_{ij} \bmod n_p$, $for\ 1 \le i, j \le n$

$S_P = \left(S_A \oplus H(X_h \| m_w \| e_P) \oplus Z\right)^{d_P} \bmod n_P$; $\oplus$ is an exclusive OR operation and $Z = H(X_1 \ldots X_n, X^1 \ldots X^n, Y_1 \ldots Y_n, Y^1 \ldots Y^n)$

$Y_i = \prod_{j=1}^{n} x_{ij} \bmod n_p$, $Y^j = \prod_{i=1}^{n} x_{ij} \bmod n_p$, $for\ 1 \le i, j \le n$. The

Open Science Index, Computer and Information Engineering Vol:10, No:6, 2016 publications.waset.org/10004632/pdf

proxy signature of message X_h is (Z, m_w, S_P, e_A, e_P).

Step3. P Compute the following ciphertext matrix:

$$C_h = \begin{pmatrix} x_{11} \oplus H(S_P,1,1,X_1,X^1) & \cdots & x_{1n} \oplus H(S_P,1,n,X_1,X^n) & C_1 \\ x_{21} \oplus H(S_P,2,1,X_2,X^1) & \cdots & x_{2n} \oplus H(S_P,2,n,X_2,X^n) & C_2 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} \oplus H(S_P,n,1,X_n,X^1) & \vdots & x_{nn} \oplus H(S_P,n,n,X_n,X^n) & C_n \\ C^1 & \cdots & C^n & S_P^* \end{pmatrix}$$

$$C_h = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} & C_1 \\ c_{21} & c_{12} & \cdots & c_{1n} & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} & C_m \\ C_1 & C_2 & \cdots & C_n & S_P^* \end{pmatrix} \quad (3)$$

where $C_i = X_i^{e_B} \bmod n_B$, $C^j = X^{je_B} \bmod n_B$, $S_P^* = S_p^{e_B} \bmod n_B$, *for all* $1 \le i, j \le n$. Now, P and B decide a temporary key K_{PB} as discussed in Section II.

Step4. Proxy P generates permutation matrix P with the temporary key K_{PB} and transmit $C' = C_h \times$ P to receiver B.

Step5. B obtains the permuted ciphertext matrix C', permutation matrix P can be produced with the shared temporary key. Then compute $P^{-1} = C' \times K_{PB}$.

Step6. The receiver B uses his/her secret key d_B to decrypt C_h and obtains decrypted message as:

$$\bar{X}_h = \begin{pmatrix} c_{11} \oplus H(S_P,1,1,X_1,X^1) & \cdots & c_{1n} \oplus H(S_P,1,n,X_1,X^n) & \bar{X}_1 \\ c_{21} \oplus H(S_P,2,1,X_2,X^1) & \cdots & c_{2n} \oplus H(S_P,2,n,X_2,X^n) & \bar{X}_2 \\ \vdots & \ddots & \vdots & \vdots \\ c_{n1} \oplus H(S_P,n,1,X_n,X^1) & \vdots & c_{nn} \oplus H(S_P,n,n,X_n,X^n) & \bar{X}_n \\ \bar{X}^1 & \cdots & \bar{X}^n & \bar{S}_p \end{pmatrix}$$

$$\bar{X}_h = \begin{pmatrix} \bar{x}_{11} & \bar{x}_{12} & \cdots & \bar{x}_{1n} & \bar{X}_1 \\ \bar{x}_{21} & \bar{x}_{12} & \cdots & \bar{x}_{2n} & \bar{X}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{x}_{n1} & \bar{x}_{n2} & \cdots & \bar{x}_{nn} & \bar{X}_n \\ \bar{X}^1 & \bar{X}^2 & \cdots & \bar{X}^n & \bar{S}_p \end{pmatrix} \quad (4)$$

E. Proxy Signature Verification

Step 7: Now the verifier or receiver B verify the following:

$$\bar{X}_i = \prod_{j=1}^{n} \bar{x}_{ij} \bmod n_P \;, \bar{X}^j = \prod_{i=1}^{n} \bar{x}_{ij} \bmod n_P \;, \text{for } 1 \le i, j \le n$$

$$\bar{Y}_i = \prod_{j=1}^{n} \bar{x}_{ij} \bmod n_P \;, \bar{Y}^i = \prod_{i=1}^{n} \bar{x}_{ij} \bmod n_P \;, \text{for } 1 \le i, j \le n$$

$$Z' = H\left(\bar{X}_1 \ldots\ldots \bar{X}_n, \bar{X}^1 \ldots\ldots \bar{X}^n, \bar{Y}_1 \ldots\ldots \bar{Y}_n, \bar{Y}^1 \ldots\ldots \bar{Y}^n\right)$$

$$H(m_w \| e_P) = \left(\bar{S}_P^{e_P} \bmod n_P \oplus H(\bar{X}_h \| m_w \| e_P) \oplus Z'\right)^{e_A} \bmod n_A \quad (5)$$

If it holds, it is accepted as a valid proxy signature; otherwise, rejected.

V. Security and Efficiency Analysis

In the following, we show that the proposed schemes satisfy the security features, namely, verifiability, strong unforgeability, strong, undeniability, strong identifiability and prevention of misuse.

A. Verifiability

The verifier of proxy signature, can check whether verification equation

$$H(m_w \| e_P) = \left(\bar{S}_P^{e_P} \bmod n_P \oplus H(\bar{X}_h \| m_w \| e_P) \oplus Z'\right)^{e_A} \bmod n_A$$

holds or not. We prove this as follows

$$\begin{aligned} &\left(\bar{S}_P^{e_P} \bmod n_P \oplus H(\bar{X}_h \| m_w \| e_P) \oplus Z'\right)^{e_A} \bmod n_A \\ &= \{(S_A \oplus H(\bar{X}_h \| m_w \| e_P) \oplus Z' \bmod n_P \oplus \\ &\qquad H(\bar{X}_h \| m_w \| e_P) \oplus Z'\}^{e_A} \bmod n_A \\ &= \{(H(m_w \| e_P)^{d_A} \bmod n_A \bmod n_P \oplus H(\bar{X}_h \| m_w \| e_P) \\ &\qquad \oplus Z' \oplus H(\bar{X}_h \| m_w \| e_P) \oplus Z'\}^{e_A} \bmod n_A \\ &= H(m_w \| e_P) \oplus (H(\bar{X}_h \| m_w \| e_P) \oplus Z')^{e_A} \bmod n_P \\ &\qquad \oplus \left(H(\bar{X}_h \| m_w \| e_P) \oplus Z'\right)^{e_A} \bmod n_P \\ &\qquad = H(m_w \| e_P)) \end{aligned}$$

B. Strong Unforgeability

In this scheme, the proxy signature is created with the proxy signer's secret key d_P and delegated proxy key S_A. The proxy key is binding with the original signer's secret key d_A. No one (including the original signer) can construct the proxy signature without having the knowledge of the secret keys d_P and d_A. Obtaining these secret keys by any other party is as difficult as breaking RSA. Moreover, the verification of $H(m_w \| e_P)$ with the signed message prevents the dishonest party from the creation of forged proxy signatures. Therefore, any party including the original signer cannot forge a valid proxy signature and thus the proposed scheme satisfies the unforgeability property.

C. Strong Identifiability

The verification process of the proposed scheme requires proxy signer's public key e_P and warrant m_w. Any verifier can determine the identity of the proxy signer from the signed message, because the signed message is

Open Science Index, Computer and Information Engineering Vol:10, No:6, 2016 publications.waset.org/10004632/pdf

$S_P = \left(S_A \oplus H\left(X_h \| m_w \| e_B\right) \oplus Z\right)^{d_P} \bmod n_P$, where S_A the signed warrant by the original signer is. Therefore, in the verification process any verifier can determine the identity of the proxy signer from m_w .

D.Strong Undeniability

From a proxy signature of the proposed scheme, the involvements of both original signer and proxy signer are determined by the warrant m_w and the connection of the public keys e_P and e_A in the verification process. Thus the proxy signer and the original signer cannot deny their involvement in a valid proxy signature. So the scheme satisfies the undeniability property.

E. Prevention of Misuse

Both the proxy signer and the original signer's misuse are prevented in our scheme. The proxy signer cannot forge the delegated rights. In case of the proxy signer's misuse, the responsibility of the proxy signer is determined from the warrant m_w . The original signer's misuse is also prevented because he cannot compute a valid proxy signature against the proxy signer, which is the unforgeability property of our scheme.

Next, we show that our scheme is heuristically secured by considering the following attacks [5].

1) *Known-Key Security (K-KS):* The session key is a unique secret key which is produced in each run of a key agreement protocol between *P* and *B.* The protocol provides known-key security, in each run a unique session key should be produced between two parties *B* and *P* which depends on r_B and r_P . Although an opponent has learned some other session keys, he can't compute ephemeral private keys r_B and r_P . Therefore, the protocol still achieves its goal in the face of the opponent.
2) *(Perfect) Forward Secrecy:* The secrecy of previous session keys established by honest entities is not affected if long-term private keys of one or more entities are compromised. The protocol also possesses forward secrecy. Suppose that static private keys x_B and x_P of two parties are compromised. Even so, the secrecy of previous session keys established by honest parties is not affected, because an opponent who captured their private keys x_B or x_P should extract the ephemeral keys r_B or r_P from the exchanged values to know the previous or next session keys between them. However, this is RSA factorization.
3) *Key-Compromise Impersonation (K-CI):* When P's static private key is compromised, it may be desirable that this event does not enable an adversary to impersonate other entities to P. Suppose P's long-term private key x_P , is disclosed. Now an opponent who knows this value can clearly impersonate P. But he can't impersonate B to P without knowing the B's long-term private key x_B . From the success of the impersonation, the opponent must know the P's ephemeral key r_P . So, in this case, the opponent should extract the value r_P from $t_P = g^{r_P} \bmod n$, then compute r_P ' from $r_p' r_P = 1 \bmod n1$ which is the RSA factorization problem.
4) *Unknown Key-Share (UK-S):* Entity B cannot be coerced into sharing a key with entity P without B's knowledge, i.e., when B believes the key is shared with some entity C ≠ P, and P correctly believes the key is shared with B. The designed protocol prevents unknown key-share. Consequent to the assumption of this protocol that s_1 has verified that P possesses the private key x_P corresponding to his static public key y_P , an opponent can't register P's public key y_P as its own and subsequently deceive B into believing that P's messages are originated from the opponent. Therefore, B cannot be coerced into sharing a key with entity P without B's knowledge.
5) *Subgroup Confinement Attack:* Also small subgroup attack, the generator g in is a primitive root of the prime p. If the selected prime p is such that p -1 has several small prime factors, then some values between 1 and p -1 do not generate groups of order p -1, but of subgroups of smaller orders. If the public parameter of either P or B lies within one of these small subgroups, so the shared secret key would be confined to that subgroup. The intruder may launch a brute force attack to determine the exact value of the shared secret key. The Solution to counter this kind of an attack is to choose a Safe Prime and use g that generates a large prime order subgroup or at the very least make sure that composite order subgroup is not vulnerable for instance the order's prime number factorization contains only large primes, which we provided in our protocol, we choose two safe prime numbers and use generator of order p'q'.
6) *Chosen Cipher Text Attack:* According to step 4, if the malicious attempt to capture the transmitted message, five variables will need to be known to decrypt the ciphertext $c_{ij} = x_{ij} \oplus H\left(S_P, 1, 1, X_1, X^1\right)$. The positions of ciphertext, check value and S_P^* are randomized, due to the permutation matrix P. Furthermore, malicious have no means of obtaining the temporary key used to produce P and therefore cannot compute P. In order to P is pseudorandomized, malicious would not know the location of the variables. In attempt to capture the message, S_P^* and 2n check sums will need to be verified to find out the original locations of the data blocks.

VI. Conclusion

In this paper, we have proposed a proxy signature scheme with fault tolerance based on a secure and efficient protocol authenticated key agreement on RSA cryptosystem. The proposed scheme satisfies the necessary security requirements of proxy signature. Furthermore, it permits the receiver to

verify sender's identity and has a more secure encrypting method by using the permutation matrix and the temporary shared key. The temporary shared key is secure and the key itself has never been used in the past. So our system can be used to improve the security in an open Internet network.

REFERENCES

[1] C.N. Zhang, "Integrated Approach for Fault Tolerance and Digital Signature in RSA," IEEE Proceedings-Computers & Digital Techniques, vol. 146, no. 3, pp. 151-159, 1999

[2] N. Lee and W. Tsai, "Efficient Fault-tolerant Scheme basde on the RSA system," IEEE Proceedings – Computer and Digital Techniques, vol. 150, no. 1, pp. 17-20, 2003.

[3] Xue, H. (2010) Improving the Fault-Tolerant Scheme Based on the RSA System. International Symposium on Computational Intelligence and Design, Hangzhou, 29-31 October 2010, 31-33.

[4] M. Mambo, K. Usuda, E. Okamoto, Proxy signature: delegation of the power to sign the message, IEICE Trans. Fundamentals E79-A (9) (1996) PP. 1338 - 1353.

[5] H. Elkamchouchi, M. R. M. Rizk, and Fatma Ahmed," A New Secure Protocol for Authenticated Key Agreement," IACSIT International Journal of Engineering and Technology, Vol. 5, No. 2, April 2013, pp.245-248

[6] Swati Verma and Birendra Kumar Sharma," An Efficient Proxy Signature Scheme Based On RSA Cryptosystem," International Journal of Advanced Science and Technology Vol. 51, February, 2013, pp.121-126

[7] B. Lee, H. Kim and K. Kim, "Secure mobile agent using strong non-designated proxy signature", In: Information security and private (ACISP01), LNCS 2119, Springer-Verlag, (2001), pp. 474-486.

[8] B. Lee, H. Kim and K. Kim, "Strong proxy signature and its applications", In: Proceeding of the 2001 symposium on cryptography and information security (SCIS01), vol. 2, no. 2, (2001), pp. 603-608.

Open Science Index, Computer and Information Engineering Vol:10, No:6, 2016 publications.waset.org/10004632/pdf