International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:6, 2016

Implementing Fault Tolerance with Proxy Signature on the Improvement of RSA System

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

***Abstract*—**Fault tolerance and data security are two important issues in modern communication systems. 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 proxy signature scheme with fault tolerance over an efficient and secure authenticated key agreement protocol based on the improved 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, improved RSA, key agreement.

I. Introduction

PROXY signature scheme was first introduced by Mambo et al. in 1996 [1]; it is an important inquiry in the field of a digital signature. This scheme allows the original signer to delegate his signing capability to the proxy signer. Based on the delegation type, proxy signatures are divided into three types, namely full delegation, partial delegation, and delegation by warrant. In full delegation, the proxy signer is given the same secret key that the original signer has, so that he can create the same signature as original signer. When the proxy signer signs a document unfavorable for the original signer, his mischievous action is not detected because the signature created by the proxy signer is indistinguishable from the signatures created by the original signer. In partial delegation, the proxy signer has a proxy signature key by combining his private key with a delegation key from the original signer. The proxy signature is different from both the original's standard signature and the proxy's standard signature.

Partial delegation with warrant is implemented by using a warrant. The warrant contains important information such as the validity period of the delegation, the identities of the proxy signer, original signer, and the generated proxy verification key. According to the warrant, any verifier can verify whether or not the signature has expired or is signed by a legal delegate.

Digital signature schemes with fault tolerance make it possible for error detections and corrections during the processes of data computations and transmissions. Zhang, in 1999 [2] Lee and Tsai, in 2003 [3] 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.

The vulnerability of Zhang's scheme is pointed out by [4], i.e. a pernicious client could produce an alternate message with the same signature by permuting the rows or columns in the original message matrix *X*. They suggested a new method; this is certainly improved Zhang's scheme in which the original message matrix is multiplied with two prime matrices with the same length of the original message. They showed that a pernicious client cannot forge a valid message with the same signature by permuting the rows and columns in the matrix. In 2013, Shreenath et al. [5] improved the mechanism of Lin et al. [4] with providing extra security by making use of transpose matrix based on the RSA. If a malicious looks into the message he will find it difficult to understand or calculate checksum/ hash value, thus it will confuse the malicious.

To keep the confidentiality of the data that transfers over a public network, Rivest et al. [6] proposed RSA technique as a public key cryptosystem. According to the proposed scheme, the sender can use the receiver's public key to encrypt a message and the receiver can use his secret key to decrypt the encrypted message. In 2014, Somani et al. [7] proposed a new security scheme for the RSA cryptosystem contains three prime numbers and overcome several attacks possible on RSA. The new scheme has speed improvement on RSA decryption side by using the Chinese Reminder Theorem (CRT).

In order for any two parities to 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.

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

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).

scheme in Section V. Finally, in Section VI, we give our conclusion.

II. Security Requirements of Proxy Signature

The security requirements for any proxy signature are first studied in [1] and later were improved in [8], [9]. According to them, a secure proxy signature scheme is expected to satisfy the following five requirements [15]:

(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 proxy signature.

III. The New Secure Key Agreement Protocol

The used protocol for authenticated key agreement [16] 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
$s_1 \equiv (y_B)^{-r_A} \cdot (t_B)^{x_A}$	$\xrightarrow{s_1}$ $\xleftarrow{s_2}$	$s_2 \equiv (y_A)^{-r_B} \cdot (t_A)^{x_B}$
$v_2 = \left((t_B)^{x_A} \cdot s_2\right)^{-\bar{r}_A}$		$v_1 = \left((t_A)^{x_B} \cdot s_1\right)^{-\bar{r}_B}$
$\equiv y_B$		$\equiv y_A$
$K_{AB} \equiv v_2' t_B{}^{x_A} t_B{}^{r_A}$		$K_{AB} \equiv v_1' t_A{}^{x_B} t_A{}^{r_B}$
$\equiv y_B{}^{r_A} t_B{}^{x_A} t_B{}^{r_A}$		$\equiv y_A{}^{r_B} t_A{}^{x_B} t_A{}^{r_B}$
$K_{AB} \equiv g^{x_B r_A + x_A r_B + r_A r_B}$		$K_{AB} \equiv g^{x_A r_B + x_B r_A + r_B r_A}$

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 $GCD(x, n1) = 1$ and 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 the improved RSA system. The improved RSA scheme provides an enhancement of [10], each user provides a public key (*e*, *N)* and a secret key *d,* where *N* is the product of three large prime numbers p, q *and* s such that $N = p \times q \times s$, and the public key *e* and secret key *d* must satisfy the equation $d = e^{-1}(p-1)(q-1)(s-1)$.

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, s : three 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 three large safe primes
- $h(\,)$: a secure one-way hash function.
- K_{AP} : shared secret key between A and P
- m_w : a warrant.

B. Proxy Key Generation

The original signer A does the following:

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

C. Proxy Key Verification

The proxy signer P checks whether $h(m_w \,||\, e_P \,||\, K_{AP}) = S_A{}^{e_A} \bmod n_A$. 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×m message matrix X to user P:

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

where $x_{ij}, 1 \le i \le n, 1 \le j \le m$, is a message block which has the same length as N_A and N_B

Step2. Now, P takes the transpose of the original matrix:

$$T=\begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix}=\begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1m} & x_{2m} & \cdots & x_{nm} \end{pmatrix} \quad (2)$$

Step3. User P then creates two prime number matrix P and Q as:

$$P=\begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ p_1 & p_2 & \cdots & p_n \\ \vdots & \vdots & \ddots & \vdots \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}, \; Q=\begin{pmatrix} q_1 & q_1 & \cdots & q_1 \\ q_2 & q_2 & \cdots & q_2 \\ \vdots & \vdots & \ddots & \vdots \\ q_m & q_m & \cdots & q_m \end{pmatrix} \quad (3)$$

where matrix P and Q both have the same dimensions with the message matrix T, which is a (m × n) matrix.

Step4. The sender *P* computes a new message matrix $\bar{T}$ which is the entry-wise product of the matrix T, *P* and *Q:*

$$\bar{T}=\begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix}\begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ p_1 & p_2 & \cdots & p_n \\ \vdots & \vdots & \ddots & \vdots \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}\begin{pmatrix} q_1 & q_1 & \cdots & q_1 \\ q_2 & q_2 & \cdots & q_2 \\ \vdots & \vdots & \ddots & \vdots \\ q_m & q_m & \cdots & q_m \end{pmatrix}$$

$$=\begin{pmatrix} t_{11}\times p_1\times q_1 & t_{12}\times p_2\times q_1 & \cdots & t_{1n}\times p_n\times q_m \\ t_{21}\times p_1\times q_2 & t_{22}\times p_2\times q_2 & \cdots & t_{2n}\times p_n\times q_m \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1}\times p_1\times q_m & t_{m2}\times p_2\times q_m & \cdots & t_{mn}\times p_n\times q_m \end{pmatrix}$$

$$=\begin{pmatrix} \bar{t}_{11} & \bar{t}_{12} & \cdots & \bar{t}_{1n} \\ \bar{t}_{21} & \bar{t}_{22} & \cdots & \bar{t}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{t}_{m1} & \bar{t}_{m2} & \cdots & \bar{t}_{mn} \end{pmatrix} \quad (4)$$

Step5. For the message matrix $\bar{T}$, the proxy P now constructs an (n+1)×(m+1) matrix T_h as:

$$T_h=\begin{pmatrix} \bar{t}_{11} & \bar{t}_{12} & \cdots & \bar{t}_{1n} & T_1 \\ \bar{t}_{21} & \bar{t}_{12} & \cdots & \bar{t}_{1n} & T_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{t}_{m1} & \bar{t}_{m2} & \cdots & \bar{t}_{mn} & T_m \\ T_1 & T_2 & \cdots & T_n & S_P \end{pmatrix} \quad (5)$$

where,

$$T_i=\prod_{j=1}^{n} t_{ij} * p_j \; mod N_P \;, for \; 1 \le i \le m$$

$$T_j=\prod_{i=1}^{m} t_{ij} * q_i \; mod N_P \;, for \; 1 \le j \le n \;,$$

$$h=\prod_{j=1}^{n}\left(\prod_{i=1}^{m} t_{ij} \; mod N_P\right) mod N_P$$

$$S_P=\left(S_A \oplus h\left(T_h \| m_w \| e_P\right) \oplus h\right)^{d_P} \; mod \; n_P \,,$$

$\oplus$ is an exclusive OR operation. The proxy signature of message T_h is $(h, m_w, S_P, e_A, e_P, K_{AP})$

Step6. P Compute the following ciphertext matrix:

a) Select k as a random integer $GCD(k, N_P)=1$ and $1< k < N_P - 1$.

b) Compute C1= $k^{e_B} \; mod \; N_B$.

c) Compute C2= $T_h^{e_B} k \; mod \; N_B$.

$$C2=\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} & C_1 \\ c_{21} & c_{12} & \cdots & c_{1n} & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} & C_m \\ C_1 & C_2 & \cdots & C_n & h_c \end{pmatrix} * k \quad (6)$$

where,

$$c_{ij}=t_{ij}^{e_B} mod \, N_B, C_i=T_i^{e_B} mod N_B, C_j=T_j^{e_B} mod \, N_B,$$

$$h_c=S_P^{d_P} mod \, N_P$$

for all $1 \le i \le n, 1 \le j \le m$

d) Send the cipher text values (C1, C2) to user B

E. Proxy Signature Verification

Step 7: To recover the message X from cipher text C2 user B should do the following:

a) Calculate

$C_p = C1 mod \; p, C_q = C1 mod \; q, C_s = C1 mod \; s$ and then calculate

$k_p = C_p^{d_p} \; mod \; p, \; k_q = C_q^{d_q} \; mod \; q \; and \; k_s = C_s^{d_s} \; mod \; s$

b) By using the formula calculate k

$$k = k_p.(qs)^{(p-1)} \; mod \; N_B + k_p.(ps)^{(q-1)} \; mod \; N_B + k_s.(pq)^{(s-1)} \; mod \; N_B .$$

c) By using the Euclidean algorithm, calculate the value of the unique integer t, $t * k \bmod N_B = 1$ and $1< t < N_B$.

d) Then compute $T_h^{e_B}$, C2*t = $(T_h^{e_B}$ .k)*t = $(T_h^{e_B})$ k*t = $T_h^{e_B} \bmod N_B$.

e) For getting the value of message $\bar{T_h}$ should do the following steps:
First calculate

$$C'_p = T_h^{e_B} \bmod p, C'_q = T_h^{e_B} \bmod q, C'_s = T_h^{e_B} \bmod s$$

then calculate

$$T_p = C'_p \bmod p, T_q = C'_q \bmod q, T_s = C'_s \bmod s.$$

f) Finally, recover the message T_h by using:

$$T_h = T_p.(qs)^{(p-1)} \bmod N_B + T_p.(ps)^{(q-1)} \bmod N_B + T_s.(pq)^{(s-1)} \bmod N_B .$$

$$\bar{T_h} = \begin{pmatrix} \bar{t}_{11} & \bar{t}_{12} & \cdots & \bar{t}_{1n} & \bar{T}_1 \\ \bar{t}_{21} & \bar{t}_{12} & \cdots & \bar{t}_{1n} & \bar{T}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{t}_{m1} & \bar{t}_{m2} & \cdots & \bar{t}_{mn} & \bar{T}_m \\ \bar{T}_1 & \bar{T}_2 & \cdots & \bar{T}_n & \bar{S}_P \end{pmatrix} \quad (7)$$

Step 8: Now the verifier or receiver B verify the checksum to check the following

$$\bar{T_i} = \prod_{j=1}^{n} \bar{t}_{ij} * p_j \bmod N_P \ , for\ 1 \le i \le m \ ,$$

$$\bar{T_j} = \prod_{i=1}^{m} \bar{t}_{ij} * q_i \bmod N_P \ , for\ 1 \le j \le n$$

$$\bar{h} = \prod_{j=1}^{n} \left(\prod_{i=1}^{m} \bar{t}_{ij} \bmod N_P \right) \bmod N_P \quad (8)$$

$$h(m_w \| e_P \| K_{AP}) = \left(s_P^{e_P} \bmod n_P \oplus h(T_h \| m_w \| e_P) \oplus h \right)^{e_A} \bmod n_A$$

If it holds, he accepts it as a valid proxy signature otherwise, rejects it.

Step 9: The receiver B takes the transpose of the matrix which will result in message as:

$$\bar{X} = \begin{pmatrix} \bar{t}_{11} & \bar{t}_{21} & \cdots & \bar{t}_{m1} \\ \bar{t}_{12} & \bar{t}_{22} & \cdots & \bar{t}_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{t}_{1n} & \bar{t}_{2n} & \cdots & \bar{t}_{mn} \end{pmatrix} = \begin{pmatrix} \bar{x}_{11} & \bar{x}_{12} & \cdots & \bar{x}_{1m} \\ \bar{x}_{21} & \bar{x}_{22} & \cdots & \bar{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_{n1} & \bar{x}_{n2} & \cdots & \bar{x}_{nm} \end{pmatrix} \quad (9)$$

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

$$h(m_w \| e_P \| K_{AP}) = \left(s_P^{e_P} \bmod n_P \oplus h(T_h \| m_w \| e_P) \oplus h \right)^{e_A} \bmod n_A$$

holds or not. We prove this as follows:

$$\begin{aligned} &\left(s_P^{e_P} \bmod n_P \oplus h(T_h \| m_w \| e_P) \oplus h \right)^{e_A} \bmod n_A \\ &= \{ (S_A \oplus h(T_h \| m_w \| e_P) \oplus h \bmod n_P \\ &\qquad \oplus h(T_h \| m_w \| e_P) \oplus h \}^{e_A} \bmod n_A \\ &= \{ (h(m_w \| e_P \| K_{AP})^{d_A} \bmod n_A \bmod n_P \oplus h(T_h \| m_w \| e_P) \\ &\qquad \oplus h \oplus h(T_h \| m_w \| e_P) \oplus h \}^{e_A} \bmod n_A \\ &= h(m_w \| e_P \| K_{AP}) \oplus (h(T_h \| m_w \| e_P) \oplus h)^{e_A} \bmod n_P \\ &\qquad \oplus \left(h(T_h \| m_w \| e_P) \oplus h \right)^{e_A} \bmod n_P \\ &= h(m_w \| e_P \| K_{AP}) \end{aligned}$$

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 and the session key K_{AP}. 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 $h(m_w \| e_P \| K_{AP})$ 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 $S_P = \left(S_A \oplus h(T_h \| m_w \| e_B) \oplus h \right)^{d_P} \bmod n_P$, 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 [11]

(1) Common Modulus Attack: The common modulus attack (CMA) [11] can be occurred by using the same modulus n, when the same message X is encrypted twice and by that attack one can retrieve the message X algorithm. The CMA is applicable in Lin et. al [4] scheme method because it uses the encryption and decryption as same as original RSA. In the proposed scheme using a unique integer k by that there are two cipheretext generated and it appears to be impractical to apply that attack on proposed scheme.

(2) Chosen Cipher Text Attack: Chosen-cipher text attack (CCA) [12] is possible in RSA due to the multiplicative property of the modular arithmetic [13] following by RSA. That means product of the two cipher texts is equal to the encryption of the product of the corresponding plaintexts. The CCA is applicable in both original RSA algorithm, and in the proposed one but by applying CCA on proposed scheme for getting the value of message X, it appears to be complex and more time consuming as compared to the original RSA algorithm.

(3) Timing Attack: An attacker can determine the value of private key by maintaining the track of how much time a computer takes to decrypt the encrypted message this because of Timing attack that occurs at RSA implementation [14]. Timing attack is applicable in majority digital signature fault tolerant schemes based on original RSA algorithm because by measuring the time for encryption and decryption, and time for key generation one can determine the value of the secrete key exponent d, but in proposed scheme by using a random unique integer k in both the encryption and decryption process makes it difficult to distinguish between the time for public key e or private key d and the time for k.

(4) 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 *A* and *P*. A protocol should still achieve its goal in the face of an adversary who has learned some other session keys. The protocol provides known-key security, in each run a unique session key should been produced between two parties *A* and *P* which depends on r_A and r_P. Although an opponent has learned some other session keys, he cannot compute ephemeral private keys r_A and r_P .

(5) (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_A 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 keys x_A or should extract the ephemeral keys or from the exchanged values to know the previous or next session keys between them. However, this is RSA factorization.

(6) Key-Compromise Impersonation (K-CI): When A's static private key is compromised, it may be desirable that this event does not enable an adversary to impersonate other entities to A .Suppose A's long-term private key x_A, is disclosed. Now an opponent who knows this value can clearly impersonate A. But he cannot impersonate P to A without knowing the P's long-term private key. For the success of the impersonation, the opponent must know A's ephemeral key r_A. So, also in this case, the opponent should extract the value r_A from $t_A = g^{r_A} \bmod n$, then compute $r_A^{'}$ from $r_A^{'} r_A = 1 \bmod n_1$ which is RSA factorization problem.

(7) Unknown Key-Share (UK-S): Entity P cannot be coerced into sharing a key with entity A without P's knowledge, i.e., when P believes the key is shared with some entity C ≠ A, and A correctly believes the key is shared with P. The designed protocol prevents unknown key-share. Consequent to the assumption of this protocol that s_1 has verified that A possesses the private key x_A corresponding to his static public key y_A , an opponent cannot register A's public key y_A as its own and subsequently deceive B into believing that A's messages are originated from the opponent. Therefore, P cannot be coerced into sharing a key with entity A without P's knowledge.

VI. Conclusion

In this paper, we have presented a proxy signature scheme with fault tolerance based on a secure and efficient protocol authenticated key agreement on improved RSA cryptosystem. The proposed scheme satisfies the necessary security requirements of proxy signature by making use of transpose matrix of the original message and has a secure channel to deliver the proxy key, through the designed protocol that meets the security attributes under the assumption that the RSA factorization problem. Also, it is trying to provide speed improvement on the decryption side of digital signature scheme fault tolerance based on improved RSA algorithm using the concept of Chinese remainder theorem. The algorithm for the proposed scheme can be protected us from

several common attacks.

REFERENCES

[1] 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.
[2] 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
[3] N. Lee and W. Tsai, "Efficient Fault-tolerant Scheme basd on the RSA system," IEEE Proceedings – Computer and Digital Techniques, vol. 150, no. 1, pp. 17-20, 2003.
[4] Iuon-Chang Lin and Hsing-Lei Wang, "An Improved Digital Signature Scheme with Fault Tolerance in RSA", Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE, 2010
[5] Shreenath Acharya, Sunaina Kotekar, Seema S Joshi, Shradda Shetty and Supreetha Lobo," Implementing Digital Signature based Secured Card System for Online Transactions", International Journal of Computer Applications 65(24):27-32, March 2013.
[6] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.
[7] Nikita Somani and Dharmendra Mangal, "An Improved RSA Cryptographic System", International Journal of Computer Applications 105(16):18-22, November 2014.
[8] 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.
[9] 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.
[10] A. H. Al-Hamami and I. A. Aldariseh, "Enhanced Method for RSA Cryptosystem Algorithm," IEEE International Conference on Advanced Computer Science Applications and Technologies, pp. 402-408, 2012.
[11] D. Boneh, "Twenty Years of Attacks on the RSA Cryptosystem," Notices of the AMS, vol. 46, no. 2, pp. 203-213, 1999.
[12] Y. Desmedt and A. M. Odlyzko, "A Chosentext Attack on RSA Cryptosystem and some Discrete Logarithm Schemes," Advances in Cryptology CRYPTO '85, vol. 218, pp. 5116-521, 1986.
[13] R. Kumar, "Security Analysis and Implementation of an Improved Cch2 Proxy Multi-Signature Scheme," International journal of computer network and Information security, vol. 4, pp. 46-54, 2014.
[14] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptology-CRYPTO '96, pp. 104-113, 1996.
[15] 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
[16] 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