Secure Proxy Signature Based on Factoring and Discrete Logarithm

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

Abstract—A digital signature is an electronic signature form used by an original signer to sign a specific document. When the original signer is not in his office or when he/she travels outside, he/she delegates his signing capability to a proxy signer and then the proxy signer generates a signing message on behalf of the original signer. The two parties must be able to authenticate one another and agree on a secret encryption key, in order to communicate securely over an unreliable public network. Authenticated key agreement protocols have an important role in building a secure communications network between the two parties. In this paper, we present a secure proxy signature scheme over an efficient and secure authenticated key agreement protocol based on factoring and discrete logarithm problem.

Keywords—Discrete logarithm, factoring, proxy signature, key agreement.

I. INTRODUCTION

The cryptographic treatment of proxy signature scheme was first introduced by Mambo et al. in 1996 [1]. Proxy signature is an important inquiry in the field of a digital signature. It permits an original signer to delegate his signing rights to a proxy signer, and then the proxy signer performs the 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. The drawback of proxy signature with full delegation is the difficulty to distinguish between original signer and proxy 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. Due to partial delegation cannot restrict the proxy signer’s signing capability, he/she can misuse the 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 1997, Kim et al. [2] proposed a scheme using the concept of partial delegation with a warrant to restrict proxy signer signing capability. In 1999, Okamoto et al. [3], for the first time, proposed proposed proxy unprotected signature scheme based on RSA scheme. A proxy-protected signature scheme based on the RSA assumption was proposed by Lee et al. in 2001 [4], [5]. In 2002, Shum and Wei [6] proposed another proxy protected signature scheme. Shao proposed the first proxy signature scheme based on the factoring integer problem in 2003 [7]. In 2005, Zhou et al. [8] proposed two efficient proxy-protected signature schemes. Their first system is based on RSA assumption and the second strategy was based on the integer factorization problem. Also, in 2005 Han et al. [9] introduced a relatively new proxy signature scheme which is as secure as ElGamal signature [10]. Next, a signature based on two hard problems: factoring and discrete logarithms was introduced by Harn [11] and Li et al. [12]. For more security, in 2013, Mat-Isa and Ismail introduced a new proxy signature with the revocation based on factoring and discrete logarithm problems [13].

The two parties must authenticate mutually and agree on a secret encryption key to communicate together securely. 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. The most famous protocol for key agreement was proposed by Diffie and Hellman, which is based on the concept of public-key cryptography (DL) [14]. There are two types of the Diffie-Hellman protocol namely static and ephemeral. In the first one, the parties exchange static public keys, and in the second, they exchange ephemeral public keys [15]. The important feature of the designed protocol is the established session key is formed as a combination of static and ephemeral private keys of two parties.

In this paper, we present a secure proxy signature scheme over an efficient and secure authenticated key agreement protocol based on two hard problems: factoring and discrete logarithm problems. The designed protocol for authenticated key agreement is secure as well as efficient and provides authentication between two entities before exchanging the session keys. The remaining parts of this paper are organized as follows: In Section II, we elaborate security properties of the proxy signature scheme. Next, we discuss the designed
protocol in Section III. In Section IV, we proposed our proxy signature scheme. We analyze the security properties and common attacks of our proposed 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 [4], [5]. According to them, a secure proxy signature scheme is expected to satisfy the following five requirements [16]:

- Verifiability: A verifier can be confident of the original signer’s agreement on the signed message from a proxy signature.
- Strong unforgeability: Only the designated proxy signer can generate a valid proxy signature.
- Strong identifiability: The identity of the proxy signer can be determined by any verifier from a proxy signature.
- 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.
- 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. NEW KEY AGREEMENT PROTOCOL

The used protocol for the authenticated key agreement [17] 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.

\[ A \rightarrow B: \text{warrant} \]

\[ B \rightarrow A: \text{warrant} \]

The system picks short-term private key \( r_A, r_B \), they are random integers \( 2 \leq r_A, r_B < n_1 \), and \( GCD (r_A, n_1) = 1 \). The protocol has low communication overhead where, the total number of transmitted bits is \(|n_1|\). The protocol has low complexity (complexity is 4) since the protocol needs only four exponential operations. So, it provides desirable performance attributes.

IV. THE PROPOSED PROXY SIGNATURE SCHEME

The proposed proxy scheme is based on the new authenticated key agreement protocol with two hard problems factoring and discrete logarithm problems. The system is divided into four phases: System setup, Proxy key generation, Proxy key verification, Proxy signature generation and Proxy signature verification.

A. System Setup

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

- \( A \): Original signer
- \( B \): Proxy signer
- \( p \cdot q \): Two large prime number
- \( (e_A, d_A) \): Secret key of original signer, \( d_A = e_A^{-1} \mod n_1 \)
- \( (e_A, n_A) \): Public key of original signer
- \( n_A, n_B \): The product of two large safe primes
- \( h() \): A secure one-way hash function.
- \( K_{AB} \): Shared secret key between A and B
- \( m_w \): A warrant
- \( ID_A, ID_B \): Identity of A and B
- \( G \): Subgroup of \( Z_p^* \) of order \( p \cdot q \).
- \( g \): Generator of \( G \).
- \( x_A, x_B \): Long-term private keys of A and B.
- \( y_A, y_B \): Long-term public keys:
  \[ y_A = g^{x_A} \mod p \]
  \[ y_B = g^{x_B} \mod p \]

B. Proxy Key Generation

1) The original signer entity A should do the following:
- Select an arbitrary integer value \( k_A \in Z_{p-1} \).
- Find \( r_A = g^{k_A} \mod p \).
- Calculate warrant \( m_w \), where, \( m_w \) must be created from \( ID_A, ID_B \) and other data on the delegation.
- Compute \( h(m_w \parallel r_A \parallel K_{AB}) \).
- Find \( \sigma_A = k_A + x_A \cdot h(m_w \parallel r_A \parallel K_{AB}) \mod p - 1 \).
- Compute \( u_A = \sigma_A A \mod n_A \).
• Send \((m, r, K_{ab}, \sigma, u)\) to the proxy signer in the secure channel.

2) The proxy signer does the following:
• Shares a key \(d\) with original signer
• Checks the validity of \((m, r, K_{ab}, \sigma, u)\) by verifying whether or not the following equation holds

\[
g^{u_d} \equiv r^h(m || r || K_{ab}) \quad (1)
\]

If the verification is successful, the proxy signer then
• Computes an alternative proxy private/public key pair \(\sigma_p\) and \(y_p\) respectively, such that

\[
\sigma_p = \sigma + x_g \cdot h(m || r || K_{ab}) \mod p - 1
\]

\[
y_p = g^{\sigma_p} \mod p
\]

C. Signature Generation

Now, the proxy signer B will sign a message \(m\) on behalf of the original signer, he uses \(\sigma_p\) to perform an ordinary signing operation. The proxy signature on the message \(m\) is then

\[(m, m', r, \text{Sign}_{\sigma_p}(m), K_{ab}, \sigma_p)\]

D. Signature Verification

Any verifier first uses the same verification procedures of the original signature scheme to check \(\text{Sign}_{\sigma}(m)\). Furthermore, the verifier has to check whether or not the following equations hold

\[
y_p^e = r^h_y(m || r || K_{ab}) \mod p \quad (3)
\]

V. SECURITY 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:

\[
y_p = g^{\sigma_p} \mod p
\]

\[
= g^{\sigma + x_g \cdot h(m || r || K_{ab}) \mod p}
\]

\[
= g^{\sigma} \cdot g^{x_g \cdot h(m || r || K_{ab}) \mod p}
\]

\[
= g^{k_A \cdot x_g \cdot h(m || r || K_{ab}) \mod p}
\]

\[
= g^{k_A \cdot g^{x_g \cdot h(m || r || K_{ab}) \mod p}}
\]

\[
= r^h_A \cdot y_A \cdot y_B \mod p
\]

B. Strong Unforgeability

In this scheme, from (2), the proxy signature is created with the proxy signer's secret key \(x_g\) and delegated proxy key \(\sigma_p\). The proxy key is bound with the original signer's secret key \(x_A\) and the session key \(K_{ab}\). No one (including the original signer) can construct the proxy signature. If the original signer tries to construct the proxy private key from a proxy public key, he/she will need to solve the discrete logarithm problem. However, the discrete logarithm problem is difficult. Moreover, from (1) the verification of \(h(m || r || K_{ab})\) with the signed message prevents the dishonest party from the creation of forged proxy signature. 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

Any verifier can determine the identity of the proxy signer from the proxy signatures created by the proxy signer. Therefore, in the proposed scheme, any verifier can identify the identity of the proxy signer from the proxy signature generated by him \((m, m', r, \text{Sign}_{\sigma_p}(m), K_{ab}, \sigma_p)\) on the message \(m\).

D. Strong Undeniability:

In the proposed scheme, from (2) the involvements of both original signer and proxy signer are determined by the secret keys \(x_g\) and \(d\) from the proxy signature. 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

In the proposed scheme, the proxy signer cannot forge the delegated rights. The responsibility of the proxy signer is determined from the warrant \(m\) in the case of the proxy signer’s misuse. Therefore, the original signer's misuse is also prevented because he/she cannot compute a valid proxy signature against the proxy signer.

Next, we show that our scheme is heuristically secured by considering the following most common attacks.

1) Known-Key Security (K-KS): In the proposed scheme, if an established session key between original signer and proxy signer is disclosed, the adversary is unable to learn other established session keys. In each run of the proposed scheme, between the two parties, a unique session key which depends on \(r\) and \(r\) should be produced. Therefore, the adversary cannot compute \(K_{ab}\) and cannot calculate \(\sigma_p = x_g \cdot h(m || r || K_{ab}) \mod p - 1\).

2) (Perfect) Forward Secrecy: If both secret keys of two parties are compromised, the adversary is unable to derive old session keys, established by two parties. The protocol also possesses forward secrecy. Suppose that adversary compromises the private keys \(x, x\), he/she cannot...
calculate \( \sigma_x = k_x + x_A \cdot h(m_u \parallel r_x \parallel K_{AB}) \mod p - 1 \). Moreover, the secrecy of previous session keys established by honest parties is not affected, because an adversary who captured the private key \( x_A \) should extract the ephemeral keys \( r_x \) or \( r_B \) from the exchanged values to know the previous or next session keys between them. However, this is DL (Discrete Logarithm Problem). On the other hand, assume adversary is able to solve FAC problem that means he/she knows the prime factorization of \( n \) and can compute \( d_A \); however, he/she cannot compute \( \sigma_x = k_x + x_A \cdot h(m_u \parallel r_x \parallel K_{AB}) \mod p - 1 \) since no information is available for \( x_A \). Thus, he/she still fails to produce \( \sigma_x \) send to proxy signer.

(3) Key-Compromise Impersonation (K-CI): When the private key of original signer is compromised, it may be desirable that this event does not enable an adversary to impersonate the other entities to \( A \). Suppose that \( A \)'s long-term private key \( x_A \) is disclosed. Now, an opponent who knows this value can clearly impersonate \( A \). In the proposed scheme, the opponent cannot impersonate to \( B \) and compute \( \sigma_x = \sigma_x + x_A \cdot h(m_u \parallel r_x \parallel K_{AB}) \mod p - 1 \) without knowing the \( B \)'s long-term private key \( x_B \). From the success of the impersonation, the opponent must know \( A \)'s ephemeral key \( r_x \). So, in this case, the opponent should extract the value \( r_x \) from \( t_x \equiv g^{r_x} \mod n \); however, he/she cannot calculate the sharing key, and this is DL. Furthermore, he cannot compute \( u_A = \sigma_A \cdot r_x \mod n \), which is the RSA factorization problem.

(4) Unknown Key-Share (UK-S): The original signer cannot be coerced into sharing a key with the proxy signer without the knowledge of original signer; i.e., one \( A \) believes the key is shared with some entity \( C \neq B \), and \( B \) believes the key is shared with \( A \). The used protocol prevents unknown key-share. Corresponding to the proxy signer’s public static and ephemeral keys \( y_B, t_B \) an adversary cannot register the proxy signer’s public keys \( y_B, t_B \) as its own, and according to the assumption of this protocol, \( s_x \) has verified that \( B \) possesses the private static and ephemeral keys \( x_B, r_B \) respectively. So an adversary cannot deceive the original signer into believing that \( \sigma_x = \sigma_x + x_A \cdot h(m_u \parallel r_x \parallel K_{AB}) \mod p - 1 \) was originated from him/her. Therefore, the original signer cannot be coerced into sharing \( K_{AB} \) with the proxy signer without his knowledge.

VI. CONCLUSION

In this paper, we proposed a new secure proxy signature with the new key agreement protocol based on factoring and discrete logarithms. Our scheme does not consider the proxy revocation mechanism. The scheme which provides a higher level of security than a single hard problem is based on two hard problems. The proposed system meets the security attributes and strong against most of potential attacks.

REFERENCES