

Research in Information Security

Dr. Ashok Kumar Das

IEEE Senior Member

Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: ashok.das@iiit.ac.in

URL: <http://www.iiit.ac.in/people/faculty/ashokkdas>

Personal Home Page: <http://sites.google.com/site/iitkgpakdas/>

Algorithms and Approaches of Proxy Signatures

Digital Signature

- Digital signature is a cryptographic means through which the authenticity, data integrity and signer's non-repudiation can be verified.
- Typically, digital signature of a document is a piece of information encrypted by the signer's private key.
- Numerous researches have shown significant contributions to this field using various cryptographic primitives.
- Nevertheless, there are many practical environments where digital signatures do not possess specific requirements, and thereby digital signatures appear in several other forms, namely proxy signatures, multi signatures, blind signatures, ring signatures etc.

Various signatures

- **Multi-signature:** A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document, and has many potential uses.

Various signatures

- **Multi-signature:** A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document, and has many potential uses.
- **Blind signature:** In many applications involving anonymity, it is desirable to allow a participant to sign a message without knowing what the message is. This is called a blind signature.

Various signatures

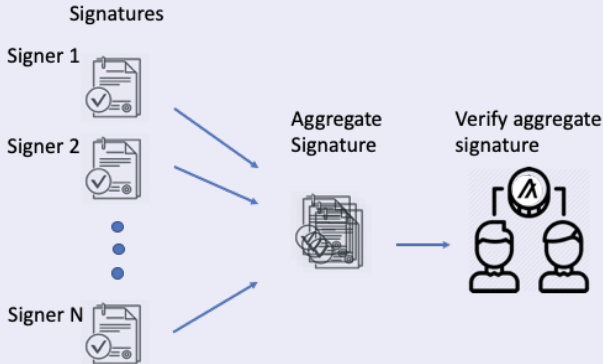
- **Multi-signature:** A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document, and has many potential uses.
- **Blind signature:** In many applications involving anonymity, it is desirable to allow a participant to sign a message without knowing what the message is. This is called a blind signature.
- **Ring signature:** Ring signatures, first introduced by Rivest, Shamir, and Tauman, enable a user to sign a message so that a ring of possible signers (of which the user is a member) is identified, without revealing exactly which member of that ring actually generated the signature.

Various signatures

- **Multi-signature:** A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document, and has many potential uses.
- **Blind signature:** In many applications involving anonymity, it is desirable to allow a participant to sign a message without knowing what the message is. This is called a blind signature.
- **Ring signature:** Ring signatures, first introduced by Rivest, Shamir, and Tauman, enable a user to sign a message so that a ring of possible signers (of which the user is a member) is identified, without revealing exactly which member of that ring actually generated the signature.

Aggregate Signature

- In a general aggregate signature scheme, signatures are generated by individual users. They can then be combined into an aggregate signature by some aggregating party.
- An aggregate signature is the same length as an ordinary signature in the underlying scheme.



Proxy signatures

- Proxy signature is a digital signature where an original signer delegates her signing capability to a proxy signer, and then the proxy signer performs message signing on behalf of the original signer.
- **Example:** A manager of a company wants to go for a long trip. She would need a proxy agent, to whom she would delegate her signing capability, and thereafter the proxy agent would sign the documents on behalf of the manager.

Proxy signatures

- The notion of proxy signature has been evolved over a long time (over 25 years now).
- However, the cryptographic treatment on proxy signature was introduced by Mambo *et al.* in 1996.

M. Mambo, K. Usuda, and E. Okamoto, “Proxy Signatures: Delegation of the Power to Sign Messages,” IEICE Transactions Fundamentals, vol. E79-A, no. 9, pp. 1338-1353, 1996.

- Mambo *et al.* classified the proxy signature on the basis of delegation, namely ***full delegation***, ***partial delegation*** and ***delegation by warrant***.

Full delegation

- In full delegation, an original signer gives her private key to a proxy signer and the proxy signer signs document using original signer's private key.
- The drawback of proxy signature with full delegation is that the absence of a distinguishability between original signer and proxy signer.

Partial delegation

- In partial delegation, the original signer derives a proxy key from her private key and hands it over to the proxy signer as a delegation capability.
- In this case, the proxy signer can misuse the delegation capability, because partial delegation cannot restrict the proxy signer's signing capability.

Delegation by warrant

- The weaknesses of full delegation and partial delegation are eliminated by partial delegation with warrant.
- A warrant explicitly states the signers' identity, delegation period and the qualification of messages on which the proxy signer can sign, etc.

Discrete logarithm problem and its applications

- The discrete logarithm is the inverse of discrete exponentiation in a finite cyclic group.
- **Instance:** A multiplicative group (G, \cdot) , an element $g \in G$ having order n and $y = g^x \bmod n$.

Question: Find x .

This problem is computationally infeasible when n is large.

Formal definition of discrete logarithm problem

Let G be a cyclic group of order n , g a generator of G , and A an algorithm that returns an integer in Z_n , where $Z_n = \{0, 1, \dots, n-1\}$. Let $a \in_R S$ denote that a is chosen randomly from the set S . Consider the following experiment, $EXP_{G,g}^{DLP}(A)$ in Algorithm 1.

Algorithm 1: $EXP_{G,g}^{DLP}(A)$

```
1:  $x \in_R Z_n$ 
2:  $X \leftarrow g^x \bmod n$ 
3:  $x' \leftarrow A(X)$ 
4: if  $g^{x'} = X \bmod n$  then
5:   return 1 (Success)
6: else
7:   return 0 (Failure)
8: end if
```

Formal definition of discrete logarithm problem

- The DLP-advantage of A is defined by $Adv_{G,g}^{DLP}(A) = Pr[Exp_{G,g}^{DLP}(A) = 1]$, where $Pr[E]$ denotes the probability of an event E .
- The discrete logarithm problem (DLP) is said to be a hard problem in G if the DLP-advantage of any adversary of reasonable resources is small, where resources are measured in terms of the time complexity of the adversary including its code size as usual.
- In other words, DLP is called a hard problem, if $Adv_{G,g}^{DLP}(A) \leq \epsilon$, for any sufficiently small $\epsilon > 0$.
- **Reference: Ashok Kumar Das. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. Networking Science (Springer), 2(1-2):12-27, 2013.**

Proxy Signatures

The Schnorr signature scheme

The scheme is based on hardness of solving DLP. It consists of the following phases.

- Setup** (\mathcal{SP}_{dlp}): Takes input 1^k and outputs **params-dlp**.
 The **params-dlp** consists of primes q and l such that $2^{k-1} \leq q < 2^k$, an element $g \in Z_q^*$ of order l that divides $q - 1$, and a hash function $h : \{0, 1\}^* \rightarrow Z_l$.
- KeyGen** (\mathcal{KG}_{dlp}): The users agree on a group G (multiplicative group of integers modulo q) for some prime q with generator g of prime order l in which the DLP is hard problem.
 The user chooses a private key $x \in Z_l$ and then computes the public key as $y = g^x \pmod{q}$.
 In other words,
 user public key $\leftarrow \mathcal{KG}_{dlp}(\text{params-dlp}, \text{user private key})$, that is,
 $y \leftarrow \mathcal{KG}_{dlp}(\text{params-dlp}, x)$.

The Schnorr signature scheme (Continued...)

- **Sign** (\mathcal{S}_{dlp}): To sign a message, say m , the signer has to choose a random number $t \in Z_l$ and calculate $r = g^t \pmod{q}$. Then the signer computes $c = h(m||r)$ and $\sigma = (t - xc) \pmod{l}$. The signature on message m is then (σ, c) . In other words, $\sigma \leftarrow \mathcal{S}_{dlp}(\mathbf{params-dlp}, (t, r), x, m)$.
- **Verify** (\mathcal{V}_{dlp}): The verifier calculates $r' = g^\sigma y^c \pmod{q}$ and $c' = h(m||r')$. If the condition $c' = c$ is satisfied, the signature is treated as valid; otherwise, the signature is invalid. In other words, **result** $\leftarrow \mathcal{V}_{dlp}(\mathbf{params-dlp}, y, \sigma, m)$, where **result** $\in \{valid, invalid\}$.

Remark: The Schnorr signature scheme is proven to be secure under the assumption that the DLP is intractable (NP-hard).

Proxy Signatures

Security properties of proxy signature

- **Strong unforgeability:** A designated proxy signer can create a valid proxy signature on behalf of the original signer. But the original signer and other third parties cannot create a valid proxy signature.
- **Strong identifiability:** Anyone can determine the identity of corresponding proxy signer from the proxy signature.
- **Strong undeniability:** Once a proxy signer creates a valid proxy signature on behalf of the original signer, he cannot deny the signature creation.
- **Verifiability:** The verifier can be convinced of the signers' agreement from the proxy signature.
- **Distinguishability:** Proxy signatures are distinguishable from the normal signatures by everyone.
- **Secrecy:** The original signer's private key cannot be derived from any information, such as the shares of the proxy key, proxy signatures, etc.

Generalized DLP-based Proxy Signature Model

Participants

- An **original signer**, who delegates her signing capability to a **proxy signer**.
- A proxy signer, who signs the message (document) on behalf of the original signer.
- A **verifier**, who verifies the proxy signature and decides to accept or reject the signature.
- A **trusted authority**, who certifies the public key.

Mechanism

- An original signer selects a private key x_o and computes her public key y_o as

$$y_o \leftarrow \mathcal{KG}_{dlp}(\mathbf{params-dlp}, x_o)$$

- A proxy signer selects a private key x_p and computes her public key y_p as

$$y_p \leftarrow \mathcal{KG}_{dlp}(\mathbf{params-dlp}, x_p)$$

- **Delegation capability generation:**

It takes **params-dlp**, original signer's chosen parameters (k_o, r_o) , original signer's private key x_o , a warrant w as input; and outputs signature σ_o on w as

$$\sigma_o \leftarrow \mathcal{S}_{dlp}(\mathbf{params-dlp}, (k_o, r_o), x_o, w)$$

Mechanism (Continued...)

- **Delegation capability verification:** It takes **params-dlp**, y_o , w , σ_o as input; produces the output **Result** where **Result** $\in \{Valid, Invalid\}$, that is,

$$\mathbf{Result} \leftarrow \mathcal{V}_{dlp}(\mathbf{params-dlp}, y_o, \sigma_o, w)$$

- **Proxy key generation ($PKeyGen_{dlp}$):** It takes **params-dlp**, σ_o , x_p and random number as input; and outputs proxy key ρ_p . Typically, the proxy signer applies simple arithmetic operation to form a proxy key as

$$\rho_p \leftarrow y_o \sigma_o + x_p y_p \pmod{q}$$

Procedurally,

$$\rho_p \leftarrow PKeyGen_{dlp}(\mathbf{params-dlp}, \sigma_o, x_p, \text{pub-parameters})$$

Mechanism (Continued...)

- **Proxy signature generation:**

It takes **params-dlp**, proxy key σ_p and message m as input; outputs proxy signature σ_p on m , that is,

$$\sigma_p \leftarrow \mathcal{S}_{dlp}(\mathbf{params-dlp}, \rho_p, m)$$

- **Proxy signature verification:**

It takes **params-dlp**, y_o , y_p , m and σ_p as input; outputs **Result**, that is,

$$\mathbf{Result} \leftarrow \mathcal{V}_{dlp}(\mathbf{params-dlp}, (y_o, y_p), \sigma_p, m)$$

Mambo *et al.*'s scheme: A case study of generalized DLP-based proxy signature model

M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages," IEICE Transactions Fundamentals, vol. E79-A, no. 9, pp. 1338-1353, 1996.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Assumptions

- This scheme is based on the discrete logarithm problem.
- $v \equiv g^s \pmod{p}$, where $s \in_R \mathbb{Z}_{p-1} \setminus \{0\}$ is a secret of the original signer and p is a prime number whose length is taken greater than 512 bits. g is a generator for $\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$.
- To determine the discrete logarithm s .

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Proxy Generation

- An original signer generates a random number $k \in_R \mathbb{Z}_{p-1} \setminus \{0\}$ and computes $K = g^k \pmod{p}$. After that, he calculates $\sigma = s + kK \pmod{p-1}$.

Proxy Delivery

- The original signer gives (σ, K) to a proxy signer in a secure way.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Proxy Verification

- The proxy signer checks a congruence such that

$$g^{\sigma} \equiv vK^K \pmod{p}$$

- If (σ, K) passes this congruence, she accepts it as a valid proxy. Otherwise, she rejects it and requests him a valid one, or she stops this protocol.

Mambo *et al.*'s Proxy Signature Scheme: The Basic Protocol

Signing by the proxy signer

- When the proxy signer signs a document m_p for the sake of the original signer, she uses the σ as an alternative to s , and executes the ordinary signing operation. Then, $(m_p, (\text{signature of the original signer}), K)$ serves as a created proxy signature.

Verification of the proxy signature

- The verification of the proxy signature is carried out by the same checking operation as in the original signature scheme except for the extra computation $vK^K \pmod{p}$. The computed value $v'(\equiv vK^K \pmod{p})$ is dealt with as a new public value.

Mambo *et al.*'s Proxy Signature Scheme: An Examples of Proxy Signature

Assumptions

- A public key v of an original signer is computed by $v = g^s \pmod{p}$. p satisfies $|p| \leq 512$ for preserving scheme's security.
- s_p is a secret of a proxy signer, and v_p is a proxy signer's public key satisfying $v_p \equiv g^{s_p} \pmod{p}$.
- After proxy submission process, (σ_p, K) satisfying the following congruences is obtained.
- $\sigma_p \equiv s + kK \pmod{p-1}$, where $K \equiv g^k \pmod{p}$ and $k \in_R \mathbb{Z}_{p-1} \setminus \{0\}$

Mambo *et al.*'s Proxy Signature Scheme: An Examples of Proxy Signature

Main Phases

- 1 **Signing.** A proxy signer generates a random number $r \in_R \mathbb{Z}_{p-1}^*$, and computes $x = g^r \pmod{p}$. For a message m_p to be signed the proxy signer computes $y = r^{-1}(m_p - x\sigma_p) \pmod{p-1}$. After that, $(m_p, (x, y, K))$ is sent to a verifier.
- 2 **Verification.** A verifier checks $(m_p, (x, y, K))$ by the congruence $g^{m_p} \equiv (vK^K)^x x^y \pmod{p}$.
If the check has succeeded, $(m_p, (x, y, K))$ is properly signed. Otherwise, signature is not valid, or the message might be manipulated.

Lu-Huang's scheme: Another case study of generalized DLP-based proxy signature model

E. J. L. Lu, and C. J. Huang, “A time-stamping proxy signature scheme using time-stamping service,” International Journal of Network Security, vol. 2, no. 1, pp. 43-51, 2006.