A Multi-Short Signature Scheme for Distributed Approval Workflows Postprint

Authors: Zuo Liming, Chen Lanlan, Zhou Qing

Date: 2018-12-13T00:00:00+00:00

Abstract

Traditional single-path linear workflows struggle to satisfy the requirements of management transactions characterized by high concurrency and stringent timeliness demands, whereas multi-layer mesh distributed architectures can effectively address such issues. To address the data interaction security concerns in distributed approval workflow business systems, we propose a short multi-signature scheme with high security. First, the security of the signature scheme is proven under the random oracle model and the Computational Diffie-Hellman (CDH) hardness assumption; based on this scheme, a distributed approval workflow interaction protocol is designed and subjected to security analysis; the signature scheme is implemented in C language and its efficiency is compared with comparable signature schemes; finally, the advantages of the application system built upon this signature scheme are analyzed. The results demonstrate that the signature scheme exhibits high efficiency and low computational overhead; consequently, the distributed approval workflow based on this scheme is applicable to e-government systems with demanding requirements for high concurrency and timeliness.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
ChinaXiv Partner Journal

A Short Multi-Signature Scheme for Distributed Approval Workflow

Zuo Liming[1],[2], *Chen Lanlan*[1],[2], *Zhou Qing*[1],[2]
([1]School of Science, [2]SEC Institute, East China Jiaotong University, Nanchang 330013, China)

Abstract: Traditional single-route workflows struggle to meet the demands of high-concurrency and time-sensitive management transactions. Multi-layer mesh distributed architectures can effectively address these challenges. To tackle data interaction security issues in distributed approval workflow business systems, this paper proposes a multi-signature scheme with high security and short signature length. First, the security of the signature scheme is proven under the random oracle model and the Computational Diffie-Hellman (CDH) hardness assumption. Based on this scheme, a distributed approval workflow interaction protocol is designed and its security is analyzed. The signature scheme is implemented in C language and compared with similar schemes. Finally, the advantages of the application system based on this signature scheme are analyzed. Results show that the proposed scheme offers high efficiency and low computational overhead, making the distributed approval workflow suitable for e-government systems with high concurrency and time-effectiveness requirements.

Keywords: distributed architecture; workflow; short multi-signature; protocol

0 Introduction

Traditional workflow technologies are designed for centralized applications. However, as modern enterprises continue to expand, information has become distributed, heterogeneous, and loosely coupled, rendering traditional workflow technologies inadequate for managing massive data volumes. This is particularly true for e-government systems aimed at government agencies, where management tasks are cumbersome and require multi-level approval. In high-concurrency, time-sensitive environments, traditional workflows fail to meet these system requirements, whereas multi-layer mesh distributed architectures can effectively solve such problems. In 2018, Pan [1] classified agents in distributed workflow management systems according to their functions, analyzed the execution process of each agent type, and researched and implemented key technologies for process execution and resource management. However, Pan' s research on distributed workflow technology focused primarily on functionality, with little attention to the security protection of approval information. For systems serving government agencies and other governmental institutions, information confidentiality is paramount. Therefore, applying digital signatures to distributed approval workflows and constructing a multi-short signature scheme suitable for such workflows is crucial for ensuring data security in business systems that adopt distributed approval workflows.

The concept of multi-signature (MS) was first proposed by Itakura et al. [2] in 1983. It is a special type of signature where multiple users sign the same message, applicable to scenarios where a document requires signatures from multiple parties, such as multi-level approval documents. This concept has wide applications in e-commerce, e-government, and other fields. In recent years, many

experts and scholars have constructed and improved multi-signature schemes. In 2015, Sahu et al. [3] proposed an identity-based multi-proxy multi-signature scheme using bilinear pairings and proved its existential unforgeability against adaptive chosen-identity attacks under the Computational Diffie-Hellman assumption and random oracle model. In 2016, Du Hongzhen [4] constructed an efficient identity-based sequential multi-signature mechanism using bilinear pairing technology and proved its existential unforgeability against adaptive chosen-message and chosen-identity attacks under the CDH hardness assumption. In 2017, Pankaj et al. [5] proposed an identity-based multi-proxy multi-signature scheme using bilinear pairings and demonstrated its high efficiency and security. Other scholars have subsequently proposed additional multi-signature schemes [6–9]. However, most of these schemes suffer from long signature lengths, resulting in low transmission and storage efficiency in poor network conditions. Therefore, constructing a multi-signature scheme with short signature length and high security is of significant importance.

The concept of short signatures was first introduced by Boneh et al. [10], with signature lengths half that of DSA signatures and higher signing efficiency. This paper incorporates short signatures into multi-signatures to construct a multi-short signature scheme suitable for distributed approval workflows, offering advantages of short signature length and high security. A corresponding interaction protocol is also designed, making it applicable to e-government systems requiring multi-level approval while satisfying the demands of high security, large data volumes, high concurrency, and loose coupling in transaction management.

---

1 Principles of Secure Distributed Approval Workflow

As shown in [Figure 1: see original paper], the secure distributed approval workflow business system discussed in this paper requires a Key Generation Center (KGC) and an approval network. The approval network structure is illustrated in [Figure 2: see original paper], which primarily includes an information submission node, an information receiving node, a rejection node, an approval completion node, an information processing node, and M approval layers. Each approval layer contains several approval nodes. Selecting one approval node from each layer forms an approval chain consisting of the selected M approval nodes along with the start and end nodes. The selection rule for nodes in an approval chain is to choose the approval node with the shortest current queue in each layer, and one approval chain completes a round of workflow approval. Specific parameter descriptions are provided in .

The key components of the approval workflow system are as follows:

a) Key Generation Center: The KGC' s primary function is to generate and distribute key pairs for each approval node in the network. Since user-submitted approval information is private, the KGC must monitor the entire approval process, leading to special system requirements: (1) Only the KGC can verify

signatures, though authorized parties may also verify with KGC authorization; (2) Secure authentication can be performed using the KGC' s master key and the approval node' s index.

b) Approval Network: As shown in [Figure 2: see original paper], the approval network comprises multiple approval chains. For an approval chain $L = \{U_{\text{Start}}, U_1, U_2, \ldots, U_M, U_{\text{End}}\}$, the information submission user submits message m to the information receiving node U_{Start}. Upon receiving m, U_{Start} timestamps it with T to record when the message first enters the approval queue, then encapsulates the information and sends it to the first approval node U_1 in chain L. Node U_1 first reviews the information. If it fails to meet approval requirements, U_1 sends the rejection result and information to the rejection node U_{Fail}; otherwise, U_1 signs the information and sends it to the next approval node U_2. Node U_2 first verifies the signature' s validity. If invalid, U_2 directly returns the information to U_{Fail}; if valid, U_2 reviews the information. If requirements are not met, U_2 sends the rejection result to U_{Fail}; otherwise, U_2 signs the information and forwards it to the next node. This process continues until the final approval node U_M receives the information, verifies the signature, and either rejects or signs the approval result, sending the signature to the completion node U_{End}. Nodes U_{Fail} and U_{End} ultimately send the approval results to the processing node U_{Info}, which returns the final result to the information submission user.

Based on these system principles and requirements, this paper designs a multi-short signature scheme for distributed approval workflows to enhance system security. Developing a secure and efficient multi-short signature scheme is the key challenge.

2.1 Mathematical Foundations

Definition 1 (Computational Diffie-Hellman Problem, CDH). Given $(P, aP, bP) \in G$ where $a, b \in_R \mathbb{Z}_q^*$ are unknown random numbers, computing abP is computationally hard.

Definition 2 (Bilinear Pairings). Given a security parameter k, let G_1 be a cyclic additive group of order q, and G_2 be a cyclic multiplicative group of the same order, where P is a generator of G_1. A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$ that satisfies the following three properties:

a) **Bilinearity:** For all $a, b \in \mathbb{Z}_q^*$, $e(aP, bP) = e(P, P)^{ab}$.

b) **Non-degeneracy:** $e(P, P) \neq 1$.

c) **Computability:** For all $Q, R \in G_1$, there exists an efficient algorithm to compute $e(Q, R)$.

If these properties hold, the map e is called a bilinear pairing.

2.2 Multi-Short Signature Scheme

The general definition of multi-short signature schemes follows that in [4]. Based on the security requirements analysis for approval workflows, this paper constructs a multi-short signature scheme suitable for distributed approval workflows. The scheme consists of five efficient algorithms: system initialization, approval node key generation and injection, approval chain and multi-signature generation, multi-signature verification, and authorized signature verification.

a) System Initialization. The KGC takes a security parameter k as input, selects a large prime q, and defines G_1 and G_2 as order-q cyclic groups with P as a generator of G_1. It selects a secure bilinear map $e : G_1 \times G_1 \to G_2$ and a random $s \in \mathbb{Z}_q^*$ as the master secret key, computing $P_{\text{pub}} = sP$ as the master public key. It also selects secure hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \to G_1$. The KGC publishes system parameters $\text{params} = \{q, G_1, G_2, P, e, P_{\text{pub}}, H_1, H_2\}$ while keeping the master key s secret.

b) Approval Node Key Generation and Injection. Before connecting to the approval network, each node in every approval layer must register with the KGC. For an approval node with identity ID, the KGC randomly selects a key generation token $K_{ID} \in \mathbb{Z}_q^*$, generates the private key $S_{ID} = (s + H_1(ID, K_{ID}))^{-1} P$, and secretly stores K_{ID}. It computes the public key $y_{ID} = H_1(ID, K_{ID}) P_{\text{pub}}$. The KGC manages approval node information by establishing an index through identity ID, using different key generation tokens for different nodes, and injects keys securely into the node with identity ID. This key generation approach differs from existing digital signature schemes to satisfy the special requirements (1) and (2) mentioned earlier.

c) Approval Chain and Multi-Signature. Let $L = \{U_{\text{Start}}, U_1, U_2, \ldots, U_M, U_{\text{End}}\}$ be an approval chain. The user submits approval information $m \in \{0,1\}^*$ to the information receiving node U_{Start}. If the information meets all approval conditions, it proceeds to the approval completion node U_{End}; otherwise, it returns to the rejection node U_{Fail}. The approval chain adds a timestamp T to the submitted information m when U_{Start} receives it, recording when the message first enters the approval queue. The message m and timestamp T are sent to the first approval node U_1.

The detailed process is as follows:

(b) Operations at Node U_1: Upon receiving the information, U_1 computes $h = H_2(m, T)$ and the partial signature $S_1 = H_1(ID_1, K_{ID_1})^{-1} \cdot h$, then sends (S_1, ID_1, m, T) to the next node U_2.

(c) Operations at Node U_i ($i = 2, 3, \ldots, M$): Node U_i first verifies the validity of the received signature before adding its own. The process is: 1. Compute $h = H_2(m, T)$. 2. Verify if $e(S_{i-1}, P) = e(h, \sum_{j=1}^{i-1} H_1(ID_j, K_{ID_j}) P_{\text{pub}})$ holds. If not,

send information m to U_{Fail} and terminate; otherwise, continue. 3. Compute $S_i = S_{i-1} + H_1(ID_i, K_{ID_i})^{-1} \cdot h$, then send $(S_i, ID_1, ID_2, \ldots, ID_i, m, T)$ to the next node U_{i+1}.

When the final node U_M receives the partial signature S_{M-1} and related information, it verifies validity, computes $S_M = S_{M-1} + H_1(ID_M, K_{ID_M})^{-1} \cdot h$, and sends $(S_M, ID_1, ID_2, \ldots, ID_M, m, T)$ to U_{End}. The signature S_M is the multi-short signature on message m by approval chain L.

d) Multi-Signature Verification. Upon receiving $(S_M, ID_1, \ldots, ID_M, m, T)$, U_{End} verifies $e(S_M, P) = e(h, \sum_{i=1}^{M} H_1(ID_i, K_{ID_i}) P_{\text{pub}})$. If valid, it sends an approval result to U_{Info}; otherwise, it rejects and sends a failure result to U_{Fail}. The correctness verification is as follows:

$$e(S_M, P) = e\left(\sum_{i=1}^{M} H_1(ID_i, K_{ID_i})^{-1} \cdot h, P\right) = e\left(h, \sum_{i=1}^{M} H_1(ID_i, K_{ID_i})^{-1} P\right) = e\left(h, \sum_{i=1}^{M} H_1(ID_i, K_{ID_i}) P_{\text{pub}}\right)$$

e) Authorized Signature Verification. The KGC publicly releases authorization information $Q_{ID} = H_1(ID, K_{ID})$ for each approval node. Anyone can verify signature validity through the node' s identity ID and Q_{ID} by checking if $e(S_M, P) = e(H_2(m, T), \sum_{i=1}^{M} Q_{ID_i} P_{\text{pub}})$ holds. If so, the signature is valid and approval is confirmed; otherwise, the signature is invalid and approval fails.

2.3 Security Model and Security Analysis

Definition 3 (Security Game). The interaction between adversary $\mathcal{A}$ (signature attack algorithm) and challenger $\mathcal{C}$ is shown in [Figure 3: see original paper]. $\mathcal{A}$ wins the game if it forges a signature $\hat{S}_n$ for some message m^* and identity set L^*_{ID} that was not queried for multi-signature, and the forged signature passes verification.

If no probabilistic polynomial-time signature attack algorithm $\mathcal{A}$ can win this game, the scheme is said to be existentially unforgeable against adaptive chosen-message attacks.

Theorem 1. Under the random oracle model and the CDH hardness assumption, the proposed multi-short signature scheme is existentially unforgeable against adaptive chosen-message attacks.

Lemma 1. If there exists an adversary $\mathcal{A}$ that breaks the scheme with non-negligible probability ε in polynomial time t, making at most q_{H_1} hash queries, q_{H_2} hash queries, q_K key generation queries, and q_S multi-signature queries with times t_{H_1}, t_{H_2}, t_K, and t_S respectively, then there exists an algorithm $\mathcal{C}$ that can solve the CDH problem with advantage ε' in time t'.

Proof. Let the CDH challenge instance be (P, aP, bP) where $a, b \in_R \mathbb{Z}_q^*$ are unknown. $\mathcal{C}$ must compute abP using $\mathcal{A}$' s capabilities.

$\mathcal{C}$ runs the system initialization algorithm with security parameter k, setting $P_{\text{pub}} = aP$ (embedding the challenge a) and generating system parameters params $= \{q, G_1, G_2, P, e, P_{\text{pub}}, H_1, H_2\}$, which are sent to $\mathcal{A}$. To facilitate signature forgery, $\mathcal{C}$ maintains a special message set Ω initialized as empty. $\mathcal{C}$ assumes $\mathcal{A}$ has made all necessary H_1, H_2, and key generation queries before signature queries, with all record lists initially empty.

a) H_1 Queries. $\mathcal{C}$ maintains a list L_1 with storage structure (ID_i, K_i, r_i). When $\mathcal{A}$ queries $H_1(ID_i)$, $\mathcal{C}$ checks L_1. If found, it returns the stored value; otherwise, it randomly selects $r_i \in \mathbb{Z}_q^*$ and returns r_i to $\mathcal{A}$, recording (ID_i, K_i, r_i) in L_1.

b) Public Key Queries. $\mathcal{C}$ maintains a list KL with structure (ID_i, K_i, y_i). For a query on ID_i, $\mathcal{C}$ checks KL. If found, it returns y_i; otherwise, it first makes an H_1 query to obtain r_i, computes $y_i = r_{iP_{\text{pub}}}$, returns y_i to $\mathcal{A}$, and adds (ID_i, K_i, y_i) to KL.

c) H_2 Queries. $\mathcal{C}$ maintains a list L_2 with structure (m_j, T_j, t_j, h_j). When $\mathcal{A}$ queries $H_2(m_j, T_j)$, $\mathcal{C}$ checks L_2. If found, it returns h_j; otherwise: - With probability δ, $\mathcal{C}$ randomly selects $t_j \in \mathbb{Z}_q^*$ and returns $h_j = t_{jP}$, recording (m_j, T_j, t_j, h_j) in L_2 and adding m_j to Ω. - With probability $1-\delta$, $\mathcal{C}$ randomly selects $t_j \in \mathbb{Z}_q^*$ and returns $h_j = t_j(bP)$, recording (m_j, T_j, t_j, h_j) in L_2.

d) Multi-Signature Queries. For a query on message m and ordered identity set $\{ID_1, \ldots, ID_j\}$: - If $m \in \Omega$ (Event E_1), $\mathcal{C}$ aborts and outputs "FAILURE" . - Otherwise, $\mathcal{C}$ retrieves records from L_1 and L_2, computes the partial signature $S_j = \sum_{i=1}^{j} r_i^{-1} \cdot h$ where $h = H_2(m, T)$, and returns S_j to $\mathcal{A}$.

After polynomially many adaptive queries, $\mathcal{A}$ outputs a forged multi-signature S_n^* on message m^* and identity set $L_{ID}^* = \{ID_1^*, \ldots, ID_M^*\}$: - If $m^* \notin \Omega$ (Event E_2), $\mathcal{C}$ aborts. - Otherwise, $\mathcal{C}$ retrieves records from L_1 and L_2. Since $h^* = t^* bP$ and the forged signature verifies, we have:

$$e(S_M^*, P) = e\left(h^*, \sum_{i=1}^{M} r_{iP_{\text{pub}}}\right) = e\left(t^* bP, a\sum_{i=1}^{M} r_{iP}\right) = e\left(P, P\right)^{t^* ab \sum r_i}$$

Thus $\mathcal{C}$ can compute:

$$abP = \left(t^* \sum_{i=1}^{M} r_i\right)^{-1} S_M^*$$

The probability that $\mathcal{C}$ successfully solves the CDH instance satisfies:

$$\varepsilon' \geq \frac{1}{2}\varepsilon(1-\delta-\delta q_S)$$

And the time bound satisfies:

$$t' < t + (2q_{H_1} + 2q_{H_2} + q_K + 2q_S)t_{\exp} + (q_{H_1} + q_{H_2} + q_K + q_S)t_{\text{pair}}$$

Therefore, $\mathcal{C}$ solves a CDH problem instance with non-negligible advantage ε' in polynomial time t', contradicting the CDH hardness assumption. Hence, the scheme is existentially unforgeable, proving Theorem 1.

3.1 Protocol Design

Traditional approval workflows are single-route and linear, making them unsuitable for high-concurrency, time-sensitive scenarios, particularly in e-government systems requiring multi-departmental, multi-level approvals. Multi-layer mesh distributed architectures effectively address these issues, but cross-server approval processes may introduce security vulnerabilities. Using our proposed scheme, we construct a secure distributed approval workflow protocol. The interaction process is as follows:

1. $User \rightarrow Start$: Submit message m to U_{Start}.
2. $Start \rightarrow U_1$: Send (m, T) to the first approval node.
3. $U_{i-1} \rightarrow U_i$: Forward $(S_{i-1}, ID_1, \ldots, ID_{i-1}, m, T)$ for $i = 2, \ldots, M$.
4. $U_M \rightarrow End$: Send final signature $(S_M, ID_1, \ldots, ID_M, m, T)$ to U_{End}.
5. $End \rightarrow Info$: Verify signature $e(S_M, P) = e(H_2(m, T), \sum_{i=1}^{M} Q_{ID_i} P_{\text{pub}})$. If valid, send approval result; otherwise, reject and notify U_{Fail}.
6. $Info \rightarrow User$: Return final result "true" or "false" .

3.2 Security Analysis of Protocol Interaction

1) Resistance to Existential Forgery Attacks
The protocol' s node-to-node transmission is unidirectional with single interaction, and its security is guaranteed by the signature scheme' s security. As analyzed in Section 2.3, the distributed workflow approval protocol based on multi-short signatures is existentially unforgeable against adaptive chosen-message attacks.

2) Resistance to Man-in-the-Middle Attacks
From the multi-short signature scheme construction, a malicious man-in-the-middle attacker must satisfy the verification equation to forge a signature for a target identity. However, as proven in Section 2.3, forging a valid signature

requires solving the CDH problem. Therefore, malicious attackers cannot forge signatures for target identities.

3) Resistance to Message Replay Attacks
The workflow approval process employs a timestamp mechanism. Each user-submitted approval message is timestamped T when first entering the workflow, ensuring message freshness. Any attacker who intercepts approval information cannot pass verification through replay. Each approval node re-verifies the timestamp; if invalid, the message is rejected and returned.

4.1 Implementation

The signature scheme was implemented in C language on Windows 7 64-bit using Microsoft Visual Studio 2012. The implementation results are shown in [Figure 4: see original paper], with core code as follows:

```
// 3. Approval chain and multi-signature
// Calculate h = H2(m, T)
element_{{{from}}_{{hash}}}(h, m, strlen(m));

// User 1 partial signature
element_{mul}(data1, ID1, X1);
element_{mul}(S1, data1, h);

// User 2 verification
element_{pairing}(left1, S1, P);  // Left side of equation
element_{mul}(temp1, ID1, Y1);    // ID1 * Y1
element_{pairing}(right1, h, temp1); // Right side: e(h, ID1*Y1)

// Compare left and right
if (element_{cmp}(right1, left1))
    printf("User 1 signature invalid!\n");
else {
    printf("User 1 signature valid!\n");

    // User 2 partial signature
    element_{mul}(data1, ID2, X2);
    element_{mul}(temp4, data1, h);
    element_{add}(S2, S1, temp4);

    // User 3 verification
    element_{pairing}(left1, S2, P); // e(S2, P)
    element_{mul}(temp2, ID2, Y2);  // ID2 * Y2
    element_{add}(temp3, temp1, temp2); // ID1*Y1 + ID2*Y2
    element_{pairing}(right1, h, temp3);
```

```
    if (element_{cmp}(right1, left1))
        printf("User 2 signature invalid!\n");
    else {
        printf("User 2 signature valid!\n");

        // User 3 signature, S3 is final multi-signature
        element_{mul}(data1, ID3, X3);
        element_{mul}(temp5, data1, h);
        element_{add}(S3, S2, temp5);

        // 4. Multi-signature verification
        element_{pairing}(left1, S3, P); // e(S3, P)
        element_{mul}(temp5, ID3, Y3);  // ID3 * Y3
        element_{add}(temp6, temp3, temp5);
        element_{pairing}(right1, h, temp6);
    }
}
```

4.2 Efficiency Analysis

Research on multi-short signature schemes is limited in existing literature, so we compare our scheme with other multi-signature schemes. compares the computational costs of signature generation and verification, where P denotes bilinear pairing operations, M denotes scalar multiplication, and E denotes exponentiation.

As shown in , our scheme requires 2 scalar multiplications for signing and 1 scalar multiplication plus 2 bilinear pairings for verification. Compared with [12], the computational overhead is similar. Compared with [5, 11], our scheme demonstrates advantages in both computational performance and signature length.

TABLE:2 Efficiency Comparison of Multi-Signature Schemes

Scheme	Signature Generation	Verification
[5]	$4M$	$4P + 2E$
[11]	$2M + E$	$3P + E$
[12]	$3M + E$	$2P + 2E$
Ours	$2M$	$2P + M$

5.1 System-Wide Security Advantages

General distributed workflow systems can meet high-concurrency and time-sensitive transaction management needs but often require cross-server operation, facing security threats such as data forgery, repudiation, impersonation, and tampering during transmission. The distributed workflow system based on our multi-short signature scheme effectively prevents these issues. Multi-short signatures are suitable for scenarios where multiple users sign the same message. With this scheme, each approval node in the workflow' s approval network can verify information validity upon receipt, significantly improving data source reliability and transmission integrity while preventing nodes from denying or forging information.

5.2 Signature Performance Advantages

The proposed multi-short signature scheme reduces signature length to 160 bits —half the length of DSA signatures—with high computational efficiency, making it suitable for poor and unstable network conditions. In mobile network environments, for a 1480-byte message, a short signature requires only 20 bytes compared to 40 bytes for a standard signature, saving 20 bytes per interaction. Since maximum packet sizes are limited, shorter signatures allow more business information per interaction, avoiding efficiency loss from packet fragmentation. This provides clear advantages for transaction-intensive systems with frequent interactions.

---

6 Conclusion

This paper proposes a multi-short signature scheme for distributed approval workflows and designs a corresponding secure interaction protocol. Under the CDH hardness assumption and random oracle model, we prove the scheme' s existential unforgeability against adaptive chosen-message attacks. The scheme was implemented and its efficiency analyzed. The solution is suitable for e-government systems requiring multi-level approval, employing a multi-layer mesh distributed architecture to meet high-concurrency and time-sensitive business demands while maintaining high security for safe and efficient system operation.

References

[1] Pan Tianheng. Research and implementation of key technologies in multi-agent system support distributed workflow [C]//IOP Conference Series: Earth and Environmental Science, 2018.

[2] Itakura K, Nakamura K. A public-key cryptosystem suitable for digital multisignatures [J]. NEC Research & Development, 1983(71): 1-8.

[3] Sahu R A, Padhye S. Identity-based multi-proxy multi-signature scheme provably secure in random oracle model [J]. Transactions on Emerging Telecommunications Technologies, 2015, 26(4): 547-558.

[4] Du Hongzhen. Secure and efficient sequential multisignature scheme for VANET [J]. Application Research of Computers, 2016, 33(10): 3105-3108.

[5] Sarde P, Banerjee A, Dewangan C L. A secure and an efficient ID-based multi-proxy multi-signature scheme from bilinear pairings [J]. International Journal of Computer Applications, 2017, 157(10): 1-6.

[6] Naoto Y, Eikoh C, Masahiro M, et al. A CDH-based ordered multisignature scheme provably secure without random oracles [J]. Journal of Information Processing, 2014, 22(2): 366-375.

[7] Wang F, Chang C C, Lin C, et al. Secure and efficient identity-based proxy multi-signature using cubic residues [J]. International Journal of Network Security, 2016, 18(1): 90-98.

[8] Yang Qing, Xin Xiaolong, Li Xiaoguang. Improved structured blind multisignature schemes based on hyperelliptic curves [J]. Chinese Journal of Engineering Mathematics, 2017, 34(3): 247-261.

[9] Wei L, Zhang L, Huang D, et al. Efficient and provably secure identity-based multi-signature schemes for data aggregation in marine wireless sensor networks [C]//Proc of IEEE International Conference on Networking, Sensing and Control. IEEE, 2017: 593-598.

[10] Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing [C]//Proc of the 7th International Conference on Theory and Application of Cryptology and Information Security. Berlin: Springer, 2001: 514-532.

[11] Qin Yanlin, Wu Xiaoping. Efficient certificateless sequential multi-signature scheme [J]. Journal on Communications, 2013, 34(7): 105-110.

[12] Wei Lifei, Cao Zhenfu, Dong Xiaolei. Secure identity-based multisignature schemes under quadratic residue assumptions [J]. Security and Communication Networks, 2013, 6(6): 689-701.

Note: Figure translations are in progress. See original paper for figures.