Postprint of a Chinese Remainder Theorem-Based Signature Scheme for Blockchain Voting Scenarios

Authors: Wang Lipeng, Hu Mingsheng, Jia Zhijuan, Publicly available, Zhang Jialei

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

Abstract

Blockchain-based voting systems can be applied to scenarios such as credit evaluation and identity authentication. The underlying cryptographic techniques of corresponding electronic voting protocols are primarily implemented based on blind signatures, ring signatures, and proxy signatures; however, traditional signature algorithms of the aforementioned types may encounter issues such as reliance on central nodes and low efficiency when applied to blockchain. A threshold signature scheme suitable for blockchain voting scenarios is proposed based on the Chinese Remainder Theorem, which generates share signatures and synthesizes the final signature through collaboration among members. The signature method supports node joining and exiting, and the signature process does not require participation of central nodes, thereby enhancing the usability of the scheme; it incorporates verification functionality for communication data while not exposing key information during the communication process, ensuring the security of data transmission over insecure communication channels in blockchain; the algorithm optimizes communication efficiency, saving network bandwidth resources while improving system throughput. Security analysis demonstrates that the difficulty of attack is equivalent to solving the discrete logarithm problem, and it can effectively resist impersonation attacks. Computational complexity analysis shows that the algorithm has low computational overhead and can be effectively adapted to blockchain application scenarios.

Full Text

Preamble

Signature Scheme Applying on Blockchain Voting Scene Based on Chinese Remainder Theorem

Wang Lipeng[1], Hu Mingsheng[1]†, Jia Zhijuan[1], Gong Bei[2], Zhang Jialei[1]
(1. School of Information Science & Technology, Zhengzhou Normal University, Zhengzhou 450044, China;
2. College of Computer Sciences, Beijing University of Technology, Beijing 100124, China)

Abstract: Blockchain-based voting systems are applicable for credit evaluation and identity verification scenarios. The underlying cryptographic schemes of corresponding electronic voting protocols mainly include blind signatures, ring signatures, and proxy signatures. However, traditional algorithms may introduce dependence on central nodes and inefficiency when applied to blockchain. This paper proposes a threshold signature scheme for blockchain voting scenarios based on the Chinese Remainder Theorem. Through cooperation among members, share signatures are generated and synthesized into a final signature. The proposed scheme supports node joining and leaving, eliminates the need for central node participation during signing, and improves availability. The scheme incorporates verification functionality for communication data while not exposing key information during transmission, ensuring data security over blockchain' s insecure communication channels. The algorithm optimizes communication efficiency, saving network bandwidth resources while improving system throughput. Security analysis demonstrates that the attack difficulty is equivalent to solving the discrete logarithm problem, effectively resisting impersonation attacks. Computational complexity analysis shows that the algorithm has low computational overhead and can be effectively adapted to blockchain application scenarios.

Key words: blockchain; confidential computation; threshold signature; Chinese remainder theorem

0 Introduction

Blockchain is a distributed database technology for recording transaction history, characterized by decentralization, anonymity, and trustlessness. It solves data trust issues among different nodes and has rapidly developed in electronic currency, financial investment, Internet of Things, healthcare, energy Internet, and other fields. Blockchain is primarily categorized into three types: public chains, consortium chains, and private chains. Currently, electronic voting systems based on blockchain have emerged on consortium and private chains for credit evaluation and decision-making scenarios.

Electronic voting protocols address security issues in Internet-based voting processes, satisfying requirements for voting legitimacy, anonymity, tally integrity, unforgeability, non-reusability, and immutability. The underlying cryptographic techniques for electronic voting protocols mainly include blind signatures, ring signatures, and proxy signatures, which can be used for identity verification and ensuring vote content trustworthiness [1]. This paper focuses on designing a ring signature scheme for blockchain online voting scenarios

that coordinates all voting participants to ensure fairness and correctness while allowing new members to join and signatures to be revoked. Compared with other voting systems, blockchain-based voting applications feature immutability and non-repudiation, with voting processes executed automatically according to protocols without manual intervention. Their trust mechanisms possess natural neutrality and security, offering significant application prospects.

Current mainstream threshold signature schemes are divided into dealer-based and dealerless schemes based on key distribution methods. Dealer-based schemes have management nodes that handle most trust authentication tasks, but these nodes become performance bottlenecks. In dealerless schemes, each node is highly autonomous at the cost of increased overall network computation. When implementing voting protocols using signature algorithms in blockchain, trusted-center-based group signature schemes face issues of trusted center selection and data leakage. As a decentralized network structure, blockchain requires signature algorithms to be designed as decentralized structures. Additionally, when nodes become unavailable, the signature algorithm must support signature revocation. Designing a secure, decentralized, revocable threshold signature scheme is the key research problem addressed in this paper.

Due to blockchain network heterogeneity, threshold signature schemes for blockchain require low computational resource consumption while providing high-security services for complex scenarios. When initiating voting, communication rounds and bandwidth requirements should be minimized. When voting nodes fail or new members join, related operations should be completed efficiently with minimal computation. Designing a signature scheme with low computational and communication resource requirements is an important prerequisite for adaptation to blockchain application scenarios.

In 2004, Tzer-Shyong et al. integrated threshold methods requiring shorter elliptic curve encryption into a new signature scheme, but did not provide identity tracking and revocation operations [2]. Literature [3] improved the key generation method of the above scheme, making the difficulty of collusion attacks equivalent to the elliptic curve discrete logarithm problem. Literature [4] proposed a threshold signature scheme that effectively resists attacks where members collude to forge signatures. These methods are threshold signature schemes based on Shamir' s secret sharing, and other secret sharing techniques have since emerged.

Literature [5] proposed an identity-based threshold signature scheme using bilinear mapping and secret sharing, employing an identity-based t-out-of-n secret sharing algorithm to improve execution efficiency. Literature [6] proposed a threshold signature scheme based on discrete logarithm difficulty that effectively resists attacks targeting secret sharing techniques. Literature [7] presented a threshold group signature scheme with short key length, low computational load, and bandwidth requirements. Literature [8, 9] proposed ECDSA-based threshold signature systems where s participants reconstruct keys but require signers. Goldfeder et al. [10] proposed using threshold signature technology to

implement multi-party control of Bitcoin keys, utilizing threshold cryptography for trusted key management. Literature [11] proposed a secret sharing scheme based on visual cryptography that effectively resists brute-force attacks on secrets.

In recent years, secret sharing schemes based on the Chinese Remainder Theorem have emerged [12-14]. Among them, the Asmuth-Bloom threshold secret sharing scheme [15] requires less computation compared to Shamir' s secret sharing but cannot guarantee data security when transmitting over insecure communication channels. Literature [16] proposed a scheme combining the ElGamal mechanism with Asmuth-Bloom threshold secret sharing to prevent secret shares from being tampered with during propagation. Literature [17]' s scheme effectively controls data length during computation with good anonymity and anti-forgery properties but must rely on a trusted center for key distribution.

Addressing existing research problems, this paper proposes a blockchain threshold signature scheme based on the Chinese Remainder Theorem with attack difficulty equivalent to solving the discrete logarithm problem. To better adapt to blockchain networks and meet their decentralized and heterogeneous communication channel characteristics, the proposed signature method supports node joining and leaving without requiring central node participation. Additionally, the scheme incorporates communication data verification functionality while not exposing key information during transmission, effectively resisting impersonation attacks. For blockchain application scenarios, the signature algorithm optimizes communication efficiency, saving network bandwidth resources while improving system throughput. Compared with existing threshold signature algorithms, this scheme has lower computational complexity in both signature generation and verification.

1.1 Digital Signatures

Digital signatures are cryptographic protection techniques using cryptography to verify data origin or integrity, primarily used in asymmetric key encryption and digital digest scenarios. A typical digital signature process involves signing by the owner, verification by others, and validity only for the current verification entity, with steps as follows [18]: , where is the private key and is the public key. , where is the plaintext message and is the generated signature. , verifying data integrity based on public key, plaintext, and signature information.

Currently commonly used signature algorithms mainly include the Elliptic Curve Digital Signature Algorithm and partial blind signature algorithm. The Elliptic Curve Digital Signature Algorithm is designed based on the elliptic curve discrete logarithm problem, with security depending on the difficulty of solving elliptic curve problems. The partial blind signature algorithm was proposed by Abe et al. [19] in 1996. Its main idea is that except for consensus messages pre-negotiated with the signer, the signer cannot obtain the content of the signed message, thereby protecting the signer' s privacy.

1.2 Secret Sharing Protocol

The concept of secret sharing was first proposed by Shamir [20] and Blakley [21]. The idea is to split a secret into N shares using appropriate methods and distribute each share to different participants. During secret recovery, the number of participants must reach a certain threshold to reconstruct the message content. Classic secret sharing algorithms include Shamir' s algorithm and the Asmuth-Bloom algorithm based on the Chinese Remainder Theorem.

1.2.1 Shamir Algorithm

Shamir's secret sharing algorithm splits secret into n sub-secrets, where any sub-secrets can recover while any sub-secrets cannot. The process consists of three steps: a) Initialization. Assuming n participants, a prime number, the trusted center' s encoding range is the finite field , the threshold value is , and each participant's ID is , where . b) Share generation. Construct a degree polynomial . Substitute the above equation to obtain and send these information pairs to each participant. c) Decryption. participants arbitrarily select message pairs and reconstruct the polynomial using Lagrange interpolation.

1.2.2 Asmuth-Bloom Algorithm

The Asmuth-Bloom algorithm operates as follows: Let be the threshold value and be the secret. Select a large prime and n integers satisfying the following conditions: strictly monotonically increasing; ; . b) Share generation. Let be greater than the product of any of the . Random select an integer where and compute . It follows that . Split the secret as where . c) Secret recovery. Any members can exchange their secret shares to recover secret . Assuming participants submit secret shares , construct the congruence system: . According to the Chinese Remainder Theorem, this system has a solution in and the solution is , where satisfies: . From this, the secret can be obtained.

2.1 Blockchain Threshold Signature System Architecture

The blockchain threshold signature system architecture further ensures data security.

2.2 Blockchain Threshold Signature System Detailed Design

The detailed process of blockchain threshold signature is described below. For convenience, symbols are defined in Table 1. Table 1 Symbols of the proposed scheme.

This paper proposes a decentralized threshold signature algorithm on blockchain based on the Chinese Remainder Theorem. Participants mainly include three roles: blockchain nodes, signature verifiers, and signature combiners. The secret

share shadow generated by node is . The partial signature satisfies the Asmuth-Bloom scheme' s large prime . The large prime for generating group public key is . The message to be signed is .

1) Initialization: Let the node set in the blockchain be with members, where the threshold value is . Select two large primes and , a positive integer sequence , and a generator on finite field , where and satisfy Asmuth-Bloom scheme requirements. Note that is public information known to all nodes. Node randomly generates node private key for key sharing member key and corresponding , letting satisfy: . Node computes and obtains node public key , broadcasting it to other nodes. After receiving messages from other nodes, each node computes the group public key: .

Figure 1 [Figure 1: see original paper] Architecture of the proposed scheme. As shown in Figure 1, the blockchain threshold signature system includes seven steps: initialization, secret splitting, partial signature generation, signature synthesis, signature verification, member joining, and member revocation. a) Generate public parameters required for the signature algorithm. Each node generates its private key information and public information, broadcasting its public information to other nodes in the network. b) Split node secret information based on the Chinese Remainder Theorem. The split secret shares are broadcast to other nodes for partial signature generation. c) Each node solves for secret information from received secret shares according to the Chinese Remainder Theorem, combines it with its key to generate partial signatures, and broadcasts them to signature combiners. d) The signature combiner synthesizes received partial signatures. Note that in blockchain scenarios, each node can act as both signature combiner and signature verifier. The synthesized signature is calculated as follows: . The signature combiner sends to the signature verifier for verification. e) When new members join, all nodes in the blockchain receive related messages and initiate the member joining process. f) When a node leaves the blockchain network, other nodes receive the exit information and initiate signature revocation. Note that most blockchain applications are built on heterogeneous networks without a trusted center for resource optimization, requiring high robustness and security from signature schemes to meet blockchain' s decentralized and heterogeneous communication channel characteristics. To better adapt to blockchain scenarios, signature algorithms must support node joining and leaving to improve availability. This scheme' s signature algorithm meets these requirements and offers higher computational efficiency compared to Shamir's secret sharing protocol. Since most blockchain applications are built on insecure communication channels vulnerable to man-in-the-middle attacks, this scheme incorporates communication data verification while not exposing key information.

2) Secret Splitting: The secret share that node sends to node is calculated as: . This will be broadcast to other nodes. To prevent malicious tampering during transmission, verification is required. The verification information generated by is and , calculated as: . Node publishes information to other

nodes. Assuming node receives this information, it performs verification to ensure data integrity: . If verification passes, the message was not tampered with in the transmission channel and is trustworthy; otherwise, blockchain node requests node to retransmit the message.

3) Partial Signature Generation: After node successfully verifies the message, it first calculates : . After computing the result, it sends related information to the signature combiner. Each node can compute where is calculated as: . Then for message , compute its corresponding partial signature: . From this, updating the group public key only requires one division operation locally without further interaction with other nodes, saving network bandwidth and improving update efficiency.

Since is public, node deletes and content. When initiating a signature, the process starts from step 3 (partial signature generation).

3.1 Correctness Proof

Theorem 1: After node receives the secret splitting message, the message verification equation holds.

Proof: Since , we have . QED.

Theorem 2: When a node receives shadow shares from other nodes, it can recover the final secret and the value is unique.

Proof: Since are sub-secrets generated by each node, can be viewed as the synthesized secret. Because , the solution to the congruence system is unique. QED.

Theorem 3: During signature synthesis, the signature verification formula holds.

Proof: Since , we have . QED.

3.2.1 Threshold Security Analysis

Implementing a threshold signature scheme on blockchain requires at least nodes to collaborate to generate the final signature in an node network. For a well-designed threshold signature algorithm, if an attacker compromises a certain number of nodes, the final voting result will not be affected as long as the number of legitimate signing nodes is greater than or equal to .

For participants, each node in the blockchain network splits sub-secret during secret splitting, with group public key . Since is public, even if a third party steals this message, solving for the group private key is a discrete logarithm problem, which is computationally difficult. Additionally, since each node stores its own sub-secret information and does not directly transmit sub-secret content during communication, group private key information cannot be obtained unless all members collude.

During partial signature generation, node receives message and verifies it to ensure the message content was not tampered with during transmission. If a third party steals this message content, solving for , , and given , , and is a discrete logarithm problem, which is difficult. Since , the verification message cannot be used to solve for .

After verification passes, at least nodes must send for secret synthesis. If more than messages are received, only groups need to be selected for synthesis. Conversely, if fewer than groups of signatures are received, the congruence system cannot be solved according to the Chinese Remainder Theorem.

Nodes sign messages and generate corresponding partial signature information sent to the signature combiner, where the partial signature generation formula is . Solving for node private key from is a discrete logarithm problem, and cannot be obtained from values.

During partial signature synthesis, the synthesis formula is , yielding: . This is sent to the signature verifier for verification using the formula . Since and are large primes, we can assume . The actual transmission does not contain private key content, so even if a third party steals this content, no meaningful information can be obtained.

3.2.2 Unforgeability Analysis

Unforgeability means that nodes in the blockchain cannot impersonate other members to generate signature information. Advanced unforgeability also includes signature traceability. Since this scheme eliminates the trusted center and all nodes have equal status, generating final signatures through collaboration avoids the trusted center impersonation problems in traditional schemes. The following analysis assumes any blockchain node can impersonate other nodes to sign sent messages . For convenience, the malicious node is defined as node and the impersonated node as .

If member impersonates member and generates its own key , based on public information, member ' s private key cannot be calculated. When is randomly generated with , since the group private key is and each blockchain node locally stores group private key information, if causes group private key calculation errors, node cannot join the signature generation process. Therefore, member cannot impersonate by generating corresponding key information .

If member impersonates member and generates node private key , to normally generate signatures, must hold. Calculating from is a discrete logarithm problem, which is difficult. Therefore, member cannot impersonate by generating corresponding node private key .

If member impersonates member and generates , since is an integer and is prime, . Member cannot generate to impersonate .

If member impersonates member and generates , since member cannot imper-

sonate to generate corresponding , , and , this implies . At this point, , so member cannot generate to impersonate .

In summary, the proposed blockchain threshold signature scheme ensures that nodes cannot impersonate other members to generate signature information, guaranteeing scheme security.

4.1 Efficiency Analysis

The proposed blockchain threshold signature algorithm has difficulty equivalent to solving the discrete logarithm problem. To compare performance with existing signature algorithms, symbols are defined in Table 2 .

Table 2 Symbols of computational complexity for the proposed scheme.

Since modular addition and subtraction have low computational overhead, they are not considered. Modular exponentiation is essentially modular multiplication and can be simplified through Montgomery exponentiation. Performance evaluation will treat modular exponentiation separately.

This section analyzes efficiency from three aspects: secret splitting, signature generation, and signature verification. Signature generation includes the previously discussed partial signature generation and signature synthesis steps. For the same computational task, only one instance is counted. Computational complexity is shown in Table 3 .

Table 3 Computational complexity of the proposed scheme.

For comparison with existing threshold signature methods, the analysis examines signature generation and verification. Existing methods are complex, mainly including Lagrange interpolation-based and Chinese Remainder Theorem-based approaches. Table 4 compares computational complexity between the proposed blockchain threshold signature algorithm and existing algorithms. Literature [22] and this paper are based on the Chinese Remainder Theorem, while literature [23, 24] use Lagrange interpolation and literature [25] uses zero-knowledge proof.

Table 4 Comparison of computational complexity.

Table 4 shows that the proposed algorithm outperforms literature [25] in both signature generation and verification. The latter includes user identity information in secret shares for authorization management and detecting participant deception, requiring additional hash functions to blind identity information and extra operations for permission management, resulting in higher computational complexity.

Generally, hash function computational complexity exceeds modular operations, making literature [24] less efficient than the proposed algorithm in signature generation. Blockchain, as a heterogeneous network with limited computing

resources, demands high algorithm execution efficiency. Since threshold signature algorithm computation concentrates on signature generation rather than verification, improving signature generation efficiency provides greater value for blockchain execution efficiency. Therefore, although the proposed algorithm is inferior to literature [23] in signature verification, its higher signature generation efficiency results in better system throughput when adapted to blockchain scenarios.

As a decentralized distributed network, blockchain distributes signature algorithm computation tasks evenly across nodes. Since blockchain node computing capabilities vary, simply increasing resources for certain nodes cannot effectively improve signature algorithm execution efficiency. The key factor affecting efficiency is communication resource consumption; reducing communication rounds shortens execution time. Although literature [22] outperforms the proposed algorithm in both signature generation and verification, it requires generating and broadcasting temporary public key information during partial signature generation, with other nodes needing to receive messages from t nodes before synthesizing the final signature. The proposed algorithm eliminates this step, reducing one communication round, saving computational resources, improving synthesis efficiency, and effectively increasing task throughput.

Literature [23-25] do not provide member joining and signature revocation functions, while literature [22] lacks signature revocation. Since blockchain is a complex network with random node status changes, power failures, and faults causing node unavailability, signature algorithms must support efficient signature revocation and member joining. The proposed signature algorithm optimizes functionality and performance for blockchain scenarios, adapting more effectively than other algorithms.

4.2 Simulation Experiments

The simulation experiments used Windows 7, Intel CPU i7-6700, Microsoft VC++ 6.0. The proposed scheme was compared with literature [23] in terms of execution efficiency, measuring the total time consumption of signature generation and verification steps in milliseconds. Both and are 150-bit integers. Experiments examine the relationship between time consumption and threshold value and member number respectively.

Experiment 1: Member number , threshold value takes values 10, 15, 20, 25, 30, 35, 40 to examine the relationship between time consumption and threshold value .

Experiment 2: Threshold value , member number takes values 40, 45, 50, 55, 60, 65, 70, 75, 80 to examine the relationship between time consumption and member number .

Simulation results are shown in Figures 2 and 3. Figure 2 [Figure 2: see original paper] shows the relationship of time consuming over the threshold t. Figure 3

[Figure 3: see original paper] shows the relationship of time consuming over the member number n.

Figure 2 shows that as threshold value increases, both schemes'time consumption increases because signature generation computational complexity is positively correlated with threshold value . Literature [23] consumes more time than the proposed scheme, with time consumption increasing faster as threshold value grows. When threshold value is small, both schemes have similar time consumption because their computational complexities are close at small threshold values.

Figure 3 shows that as member number increases, the proposed scheme' s time consumption remains basically stable and is always less than literature [23]. Combined with Figures 2 and 3, the proposed algorithm shows relatively small time consumption fluctuations when threshold value and member number change, maintaining stable performance. For blockchain' s heterogeneous network with frequently changing node numbers, the proposed algorithm' s performance does not fluctuate significantly, demonstrating better robustness and improved adaptation to blockchain voting protocols.

5 Conclusion

When applying ring signature algorithms to blockchain voting, issues of untrusted nodes and low efficiency arise. This paper proposes a blockchain threshold signature scheme based on the Chinese Remainder Theorem with attack difficulty equivalent to solving the discrete logarithm problem. Blockchain features decentralization and heterogeneous communication channels. To adapt to blockchain networks, the proposed signature method supports node joining and leaving without central node participation, improving availability. Since blockchain is built on insecure communication channels vulnerable to man-in-the-middle attacks, the scheme incorporates communication data verification and does not expose key information during transmission, ensuring data security.

Security analysis demonstrates that the proposed threshold signature scheme effectively resists impersonation attacks, overcoming security defects in native blockchain systems. For blockchain application scenarios, the algorithm optimizes communication efficiency, saves computational resources, and improves system throughput. Performance analysis shows that compared with existing threshold signature algorithms, the proposed scheme has lower computational complexity in both signature generation and verification, with better robustness.

References

[1] Dong Youkang, Zhang Dawei, Han Zhen, et al. Board voting system based on the consortium blockchains [J]. Chinese Journal of Network and Information Security, 2017, 3(12): 17-23.

[2] Chen Tzershyong, Hsiao Tsungchih, Chen Tzerlong. An efficient threshold group signature scheme [C]//Proc of IEEE Region 10 Conference Tencon. Piscataway, NJ: IEEE Press, 2004: 13-16.

[3] Peng Ya. Research on threshold digital signature theory and application [D]. Guangzhou: Sun Yat-sen University, 2010.

[4] Xie Dong, Li Jiajia, Shen Zhonghua. A new threshold signature scheme based on elliptic curve crypto system [J]. Journal of Hangzhou Normal University: Nature Science Edition, 2013, 12(1): 57-60.

[5] Liu Hongwei, Xie Weixin, Yu Jianping, et al. Efficiency identity-based threshold group signature scheme [J]. Journal on Communications, 2009, 30(5): 122-127.

[6] Yan Jie, Yin Xuri, Zhang Wujun. Research on group signature with threshold value based on elliptic curve [J]. Journal of Southeast University: Nature Science Edition, 2008, 38(1): 43-46.

[7] Chung Yufang, Chen Tzerlong, Chen Tzershyong, et al. A study on efficient group-oriented signature schemes for realistic application environment [J]. International Journal of Innovative Computing Information & Control, 2012, 8(4): 2713-2727.

[8] Gennaro R, Jarecki S, Krawczyk H, et al. Robust threshold DSS signatures [J]. Information and Computation, 2001, 164(1): 354-371.

[9] Gennaro R, Jarecki S, Krawczyk H, et al. Secure distributed key generation for discrete-log based cryptosystems [C]//Proc of International Conference on Theory and Application of Cryptographic Techniques. Berlin: Springer Press, 1999: 295-310.

[10] Goldfeder S, Gennaro R, Kalodner H. Securing Bitcoin wallets via a new DSA/ECDSA threshold signature scheme [EB/OL]. (2015) [2018-07-23]. https://www.cs.princeton.edu/~stevenag/threshold_sigs.

[11] Jia Xingxing, Wang Daoshun, Nie Daxin, et al. Collaborative visual cryptographic schemes [J]. IEEE Trans on Circuits & Systems for Video Technology, 2018, 8(5): 1056-1070.

[12] Hou Zhengfeng, Tan Mengna. A CRT-based (t,n) threshold signature scheme without a dealer [J]. Journal of Computational Information Systems, 2015, 11(3): 975-986.

[13] Shi Nan, Hou Zhengfeng, Tan Mengna, et al. A threshold encryption scheme without a dealer based on Chinese remainder theorem [C]//Proc of IEEE International Conference on Communication Software and Networks. Piscataway, NJ: IEEE Press, 2017: 90-96.

[14] Xu Pu, Ma Jingjin. Improvement of threshold RSA signature scheme based on Chinese remainder theorem [J]. Journal of Electronics & Information Technology, 2015, 37(10): 2495-2500.

[15] Asmuth C, Bloom J. A modular approach to key safeguarding [J]. IEEE Transactions on Information Theory, 1983, 29(2): 208-210.

[16] Cheng Yu, Liu Huanping. The Asmuth-Bloom verifiable threshold sharing scheme [J]. Natural Sciences Journal of Harbin Normal University, 2011, 27(3): 35-38.

[17] Dang Jiali, Yu Huifang. Group signature scheme using Chinese remainder theorem [J]. Computer Engineering, 2015, 41(2): 113-116.

[18] Chen Si. Research on anonymity and key management of Bitcoin [D]. Xian: XiDian University, 2017.

[19] Abe M, Fujisaki E. How to date blind signatures [C]//Advances in Cryptology-ASIACRYPTO. Beijing: Springer Press, 1996: 244-251.

[20] Shamir A. How to share a secret [J]. Communications of the ACM, 1979, 22(11): 612-613.

[21] Blakley G R. Safeguarding cryptographic keys [C]//Proc of International Workshop on Managing Requirements Knowledge. New York: AFIPS Press, 1979: 313-317.

[22] Wang Yan, Hou Zhengfeng, Zhang Xueqi, et al. Dynamic threshold signature scheme based on Chinese remainder theorem [J]. Journal of Computer Applications, 2018, 38(4): 1041-1045.

[23] Xu Fu. Proactive threshold RSA signature scheme based on polynomial secret sharing [J]. Journal of Electronics & Information Technology, 2016, 38(9): 2280-2286.

[24] Shang Guanglong, Zeng Xuesong. Threshold group signature scheme without TA [J]. Journal of Hebei North University: Natural Science Edition, 2017, 33(5): 4-8.

[25] Cao Yang. Digital signature scheme based on secret sharing [J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2015, 27(3): 418-421.

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