International Journal of Information Security and Privacy
Volume 17 • Issue 1

"Every Dog Has His Day": Competitive-Evolving-Committee Proactive Secret Sharing With Capability-Based Encryption

Chuyi Yan, Institute of Information Engineering, Chinese Academy of Sciences, China
Haixia Xu, Institute of Information Engineering, Chinese Academy of Sciences, China*
Peili Li, National Engineering Research Center for Cryptography, China

ABSTRACT

This article proposes a competitive-evolving-committee proactive secret sharing. Every participant in the system has the opportunity to become a member of the holding committee and have sufficient anonymity. During the life cycle of serving as the holding committee members, they only send one message in the protocol without excessive interaction, and achieve receiver strong anonymity with a capability-based encryption scheme different from most public-key encryption schemes, at present named RiddleEncryption, which is also proposed in this paper. In RiddleEncryption the sender does not need to pay attention to the specific identity of the receiver but focuses on what kind of capability the receiver should have. Nobody can determine this kind of capability at the beginning of the system establishment. This article aims at depositing a secret in a distributed manner (e.g., blockchain) without excessive trust and to emphasize more anonymity and capability. The scheme can be used in the dynamic groups, authentication management, rights abuse prevention, and so on.

KEYWORDS

Anonymity, Capability, Communication Protocol, Cryptography, Distributed System, Encryption, Information Theory, Secret Sharing

INTRODUCTION

Distributed systems pursue more rights for each node in the system. The supernodes in the distributed system which appear in some applications, such as the trusted third party, are contrary to the original intention of the distributed system which may cause excessive trust, single point of failure and be tracked.

Considering a scenario that a temporary group is required to do some downstream work depending on the group members capability. How can the dynamic groups be quickly formed? Generally, an authority may point out who the members are or finding some members who you already knew in the real world. But it may cause excessive trust of miss someone who do have such capability. It would

DOI: 10.4018/IJISP.318697

*Corresponding Author

be more secure and ideal if everyone had the opportunity to compete for the group members, which can also mitigate the burden of on single party. This article designed the scheme with the intention of depositing a secret (can be consider as the downstream work requirement) in a distributed manner (e.g., blockchain) without excessive trust and pursuing more anonymity and fairness. Firstly, every node has the opportunity to become the group member, and this group is not permanent, and it will change in the next round. Secondly, it is necessary to consider that people will not expect a single node that handle the downstream work because of the single point failure. So this article considers a group of participants to form groups, which can also be called as holding committee members, and each one holds a part of the secret (can be consider as the symbol of their capability), so put it together and they get the global secret. To resist the collusion attack, the holding committee members should not know who the other holding committee members are during the period of holding the part of the secret. Moreover, they only send one message when something needs to be done in a distributed manner (such as the center generate certificates for users, etc. In this scheme, center members only generate certificates in a distributed manner, and the master private key will not be reconstructed at any time). At the same time, from the attacker's perspective, they do not know who the current holding committee members are, so they cannot launch attacks such as *DDoS (Distributed Denial of Service)*.

To form a dynamic committee, this article use *SS (Secret Sharing)*, and the members of the previous round will send their share to the holding committee members of the next round. However, as long as a message has been sent, the node's identity will be exposed, and there is a risk of being attacked like DDoS. Then the node must complete the secret transmission when sending the message, and the sender needs to know the size of the holding committee in the next round and who they are in advance. Nevertheless, to ensure anonymity, the sender cannot know who they are in advance. So two problems need to consider: 1) How to determine the holding committee members' size to be shared in the next round? 2) How to re-share the secret to the holding committee member in the next round without knowing each other's identity?

In response to the first problem, this article modified the random number generation protocol in Ouroboros (Kiayias et al., 2017). The number of the holding committee members can be determined by all participants in the system together. For the second problem, it means, the sender needs to know the public keys of the holding committee members in the next round, but at this time, these public keys cannot correspond to any receivers like ordinary public-key encryption schemes because this will follow the public key to find the node of the specific receiver and then the adversary can launch a DDoS attack. So this article proposed a capability-based encryption scheme named **RiddleEncryption**. This scheme is similar to the process of guessing a riddle. The public key acts as the $clue$ of the riddle, and the private key acts as the $answer$ to the riddle. All participants can participate in the process of guessing the riddle. If someone guesses the private key correctly, then he will be a holding committee member in the next round. Of course, this will involve difficult problem-solving. The specific parameters set will meet the balance of feasibility and security. In this way, the sender only needs to know what capability the receiver should have without identifying the specific person at all.

Moreover, this article uses the **RiddleEncryption** to construct the **Competitive-Evolving-Committee Proactive Secret Sharing**, and the "authority" in our scheme is obtained by oneself (by solving difficult problems in a limited environment), rather than being granted by a higher-level person in advance. Therefore everyone in the system has the opportunity to be the holding committee member by their capability, which means "Every dog has his day."

Background and Motivation

This subsection compares the strengths and limitations of various popular SS schemes, illustrates the problems of existing schemes with applications that can benefit from SS, and the motivation of using RiddleEncryption to construct our secret sharing scheme.

SS schemes can be broadly classified into three categories: basic SS, roles rights limitation SS, and techniques-based SS. Among them, basic SS can be further subdivided according to the features of shares. Roles rights limitation SS continues to be subdivided according to the roles' abilities, and techniques-based SS, which the authors have listed the studies with some similarity to this study for comparison. The specific type, rationale overview, strengths, limitations, and references of the involved methods are shown in Table 1.

Basic SS has the superiority of simple operation and can meet the long-term preservation of secret information in a relatively secure environment. For example, in preserving sensitive bank information, the authority will segregate the intranet from the extranet in multiple layers, and the number of users is huge, when the bank is suitable to use basic SS to meet the balance of confidentiality and efficiency. But if it is a sensitive business that requires multi-agency networking, basic SS is somewhat lacking. Examples include trusted voting systems and medical services. In the voting system, there are multiple attacks by adversaries, and it is necessary to achieve relative fairness and security in a specific scenario, so the ability of roles is limited, and roles rights limitation SS can be used as needed. Some difficult conditions require multiple specialist consultations, but patient information is very sensitive in this scenario, so the need for SS technology is more urgent. Various technique-based SS are designed according to different scenarios, mainly focusing on the recovery of target key (holding target key can recover secret information).

In summary, most of the existing SS schemes require a trusted third party and the participants are already designated in advance by some trusted channels. However, in areas such as medical information, federal learning, and probate security, the presence of a third party may lead to the disclosure of sensitive information, and the advanced designation of participants may lead to attacks such as DDoS, especially in probate security. While existing schemes that do not require trusted third parties have achieved some success, there are still some problems, such as trusted centers may be involved in the reconstruction or validation phase leading to information leakage (Pedersen, 1991), participants may be maliciously spoofed or attacked (W. & L., 2003) or trusted centers have security risks (Kaya & Selçuk, 2008). Therefore, approaches that do not require trusted third parties and do not rely on trusted channels need further research. Based on the assumption that no trusted third party is required, the assumption of trusted channel is wanted to be further removed, so the idea of RiddleEncryption is introduced in this paper to construct the SS scheme.

Related Work

Secret sharing was proposed by Shamir (Shamir, 1979) and Blakley (Blakley, 1979) in 1979, respectively, and aims to split confidential information and keep it by different participants, which can reconstruct the secret when a threshold of participants is exceeded. The authors can broadly classify SS schemes into three categories, as shown in Table 1.

Early SS schemes were mainly basic SS, classified according to share features: static (Asmuth C, 1983; Shamir, 1979), weighted (Tassa, 2007), and changeable shares (Blundo et al., 1996; A. Herzberg, 1997). This classification is mainly based on the idea that different roles of participants have different statuses, and thus the amount of confidential information in their hands is different. Due to the expansion of scenarios, such as the need for missiles, satellites, etc., the number of confidential information that needs to be shared becomes larger, thus according to the number of information to be shared can be divided into single (Asmuth C, 1983; Shamir, 1979) and multiple secrets (He & Dawson, 1994; Lin & Harn, 2012). As the time of SS use goes on, the secrets are prone to leakage for a long time, so the members' shares need to be updated periodically to fight against the mobile adversary, which can be divided into proactive (A. Herzberg et al., 1995; Ostrovsky & Yung, 1991) and dynamic SS (Blundo et al., 1996; Yuan & Li, 2019) according to the way the shares updated. However, it brings the problem of participant access difficult.

As SS empowers more businesses, limits are placed on the capabilities of participants in each granular domain to balance availability and security within the domain. Roles rights limitation SS can be further divided into computation power limitation (Sohrabi et al., 2020) and honesty limitation

Table 1.
Secret Sharing Methods Comparison

Type	Specific Type		Process Overview	Strengths	Limitations
Basic SS	Static shares	Single secret(Asmuth C, 1983; Shamir, 1979)	Only one secret shared.	Simple and secure.	Low efficiency when more confidential information.
		Multiple secrets (He & Dawson, 1994; Lin & Harn, 2012)	More than or equal to one secret shared.	Multiple confidential information shared in efficiency.	Hard balance between security and computation.
	Weighted shares(Tassa, 2007)		The amount of information carried by each share varies by roles.	Distinguishability of role responsibilities.	Abuse of authority.
	Changeable shares	Proactive SS (A. Herzberg et al., 1995; Ostrovsky & Yung, 1991)	Shares will be updated over time.	Prevents leakage due to long time storage.	Difficult access for participants and no protection against group cheating.
		Dynamic SS (Blundo et al., 1996; Yuan & Li, 2019)	Sharing strategy can be changed.	High flexibility.	Low verifiability.
Roles rights limitation SS	Computation power limitation (Krawczyk, 1994; Sohrabi et al., 2020)		Each role's computation power is limited.	High improvement in efficiency and security level in small scenarios.	The scenario is too limited.
	Honesty limitation(Ogata et al., 2006)		Honest participants can recover secret, dishonest participants can not.	High verifiability and security.	Sacrifice efficiency.
Techniques-based SS	Polynomial-based (Asmuth C, 1983; Shamir, 1979)		Use interpolation techniques such as Lagrange interpolation.	Simple and high fault tolerance.	Slow calculation speed when more participant.
	Counting-based (A. Gutub et al., 2019a; Adnan Gutub & Taghreed AlKhodaidi, 2020)		Count the number of one in shares to recover the secret.	Simple and practical.	Shares limitation when need high tolerance.
	Capability-based(Ours)		Select participants based on their capability to recover secret with no dealer.	High anonymity and prevents leakage due to long time storage.	Protocol is complex with limitation of associated time.

(Ogata et al., 2006). However, this type of SS scenario is more limited but achieves a high balance between usability and security in small scenarios.

Recently, various technique-based SS schemes have been devised with the further expansion of the scenario. For instance, grounded theory-based approaches such as the Lagrangian difference polynomial principle (Tassa, 2007) and Chinese Remainder Theorem (Asmuth C, 1983) are proposed. In order to improve the practicality of SS (Al-Qurashi & Gutub, 2018), counting-based SS comes into view. This method performs secret recovery by counting the number of ones in shares (Al-Ghamdi et al., 2019; AlKhodaidi & Gutub, 2020; A. Gutub et al., 2019b; A. Gutub & Al-Qurashi, 2020; Adnan Gutub & Taghreed AlKhodaidi, 2020), which avoids complicated calculations and increases practicality, and multimedia information hiding can be performed by counting-based SS (Alaseri & Gutub, 2018; A. Gutub & Al-Ghamdi, 2019; A. Gutub & Al-Ghamdi, 2020; A. Gutub & Alaseri, 2020; A. A.-A. Gutub & Alaseri, 2021).

With the emergence of scenarios such as medical data management, probate management, and federal learning, the demand for SS solutions without trusted third parties and not relying on trusted channels has increased, while relevant SS research is still scarce. Dynamic PSS can be used to alleviate the problem. Calypso (Kokoris-Kogias, 2018) uses threshold encryption to construct DPSS for key management and confidential information storage. Dfinity (Hanke et al., 2018) implement randomness beacon in the dynamic committee, but the global secret needs to be updated in each round. However, in dynamic PSS, the determination of committee members is specified by the external output, not by the protocol itself. Therefore, Benhamouda and Gentry et al. (Benhamouda, 2020, November) proposed an Evolving-Committee PSS (ECPSS) scheme to determine the committee members by the protocol itself. The above studies can alleviate the assumption of trusted third parties or trusted channels, but methods to simultaneously meet the security needs of both still need further research.

Contribution

Firstly, this article proposed **Competitive-*ECPSS*** *(Competitive-Evolving-Committee Proactive Secret Sharing)* and gave an instantiation. Our scheme just uses one type of committee, which is the holding committee. It means that all the participants can compete for the holding committee members by themselves, without passively waiting for the nominator to select them, which can significantly increase the participants' enthusiasm. Moreover, everyone can determine the number of holding committees in the next round and the *capabilities* of the receivers in a distributed manner.

The reason why determine *capability* rather than specific identity is for the anonymity consideration; that is, the adversary cannot infer the specific node from the *capability* in advance to avoid being DDoS. Moreover, before sending a message, the members of the holding committee in the current round do not know each other.

Secondly, for achieving the above goals, this article proposed a capability-based encryption scheme, named **RiddleEncryption** and gave an instantiation that is different from the existing public-key encryption scheme. In the traditional public-key encryption scheme, the corresponding node can be found along with the public key, and then the adversary can launch a DDoS attack.

When one do not pay attention to *who* holds the public key and focus on the *capability* of the receivers, one may think of time-lock puzzles and attribute encryption, but **RiddleEncryption** is still different from them. In the term of time-lock puzzles, it emphasized "fixed period in the future" but this article emphasized "specific capability for the party". In the term of attribute encryption, the public key of it is based on attributes specified in advance. Some authorities have specified the specific attributes of each person at the beginning of the system establishment, and they are not strived based on the capability of the participants. Nevertheless, in **RiddleEncryption**, participants can become receivers as long as they have the capability to solve difficult problems (like the discrete logarithm problem) in specific situations.

Organization

The structure of the rest article is as follows: Section 2 introduces some preliminary information. Section 3 introduces **RiddleEncryption** and its instantiation in detail. Section 4 introduces our **Competitive-ECPSS** and its instantiation in detail. Section 5 analyzes the correctness, complexity, parameters bound, and security of the **Competitive-ECPSS**. Section 6 gives some applications of our scheme. Section 7 concludes the paper.

PRELIMINARIES

In this section, this article will describe the background and definitions that will be used in our scheme.

Broadcast

The only communication method used in our scheme is the broadcast channel (with authentication), which is public, and both honest parties and adversaries participating in the protocol can obtain and send messages from the channel. Assuming that there is no delaying. Besides, this article does not assume the secure channel or the sender-anonymous channel because such assumption makes the realization of the anonymity of transmission simple, but in fact, establishing such a channel is hard. At the same time, in our scheme, the adversary is the mobile adversary, i.e., the adversary can see the messages in the broadcast channel and corrupt any sender, or they can launch a DDoS attack after knowing the identity of a specific node. The assumption used in our scheme is used in many proactive protocols and assuming that all parties can safely erase their own state and secret information.

Secret Sharing

This article uses $t-of-n$ secret sharing, which means sharing a secret σ into n parties, and when t of shares get together can reconstruct the secret. For the instantiation, this article uses the Shamir secret sharing scheme.

Definition 1. (Secret Sharing) On the message space $\boldsymbol{\mathcal{M}}$, the $\boldsymbol{t-of-n}$ secret sharing scheme $\left(\boldsymbol{Sh}, \boldsymbol{Re}\right)$ has the following two properties:

Correctness: Using Sh algorithm to generate correct share and choose any t of them can use Re algorithm to reconstruct m, i.e., $\forall m \in \mathcal{M}$, any participants set $\mathcal{S} = \left\{i_1, i_2, \ldots, i_t\right\} \subseteq \left\{1, 2, .., n\right\}$ of size t, $Pr\left[Re\left(s_{i_1}, s_{i_2}, \ldots, s_{i_t}\right) = m : \left(s_1, s_2, \ldots, s_n\right) \leftarrow Sh\left(m\right)\right] = 1$.

Semantic Secure: If for any two messages of the same length $m, m' \in \mathcal{M}$, any participants set $\mathcal{S} \subseteq \left\{1, 2, \ldots, n\right\}$ and $\left|\mathcal{S}\right| < t$, for any adversary $\mathcal{A}$, $\left|Pr\left[\mathcal{A}\left(s_i : i \in \mathcal{S}\right) = 1 : \left(s_1, \ldots, s_n\right) \leftarrow Sh\left(m\right)\right] - Pr\left[\mathcal{A}\left(s_i^{'} : i \in \mathcal{S}\right) = 1 : \left(s_1^{'}, \ldots, s_n^{'}\right) \leftarrow Sh\left(m'\right)\right]\right| \le negl\left(1^\lambda\right)$ where λ is security parameter, then the secret sharing is semantic secure.

Ouroboros in Our Scheme

Ouroboros (Kiayias et al., 2017) has constructed a safe and efficient way to generate random numbers in a distributed manner in the blockchain environment. This article transforms Ouroboros to applying the scheme in a distributed system rather than a blockchain environment. The following is the protocol after transformation.

```
Ouroboros in our scheme
Participate: All the participants in the system.
Input: Security parameter $\lambda$.
Output: Every participant gets a consistent random number.
Commitment:
1. Each participant randomly chooses a number $u_i, i \in \left[1, n_{total}\right]$, $n_{total}$ is
the number of the participants.
2. Make commitment $c_i \leftarrow Com\left(u_i; r_i\right), r_i$. is the random seed using in
the commitment.
3. Share $u_i$ into $n_{total}$ pieces as $\left(\sigma_1^i, \ldots, \sigma_{n_{total}}^i\right)$ and encrypt the share
with other participants' public keys $\left(ct_1^i, \ldots, ct_{n_{total}}^i\right)$ then broadcast it.
Reveal: All the participants open the commitment and then
broadcast it.
Recovery:
```

```
1. All the participants check whether everyone has opened the
commitment.
2. If someone did not open it, decrypt the corresponding $ct_{id}^{i}$, $id$ is
everyone's identity, and get the $\sigma_{id}^{i}$ to broadcast it so that
everyone can recovery $u_i$.
3. Now, each participant has all the random numbers from others
and then XOR (exclusive OR) them to get the final random number.
```

Remark. This approach does not lead to insecurity because Ouroboros has its own error correction capability and can meet the above expectations. Moreover, the adversary cannot affect the number and parameters of the next round because in Ouroboros, once the adversary does not like the number is chosen and wants to abort, the adversary will not be able to participate in the game without affecting other honest parties to execute the protocol.

ElGamal Encryption

ElGamal encryption was proposed in 1984 by T.ElGamal and this article uses it to instantiate the RiddleEncption and the CECPSS protocol.

Non-Interactive Zero-Knowledge Proofs

This article uses the standard definition of NIZK (Non-Interactive Zero-Knowledge Proofs) (Blum, 1988), using a common reference string.

OUR RIDDLEENCRYPTION SCHEME

In this section, this article will propose an encryption scheme based on the capability of the receiver. The framework of our RiddleEncryption scheme is shown in Figure 1.

Model and Definitions

Definition 3. A RiddleEncryption $RE = \left(Setup, Enc, GuessingRiddle, Dec\right)$ is a quaternion of algorithms as follows, assuming that the system is built under some difficult assumption τ (like discrete logarithm problem, CDH and etc.):

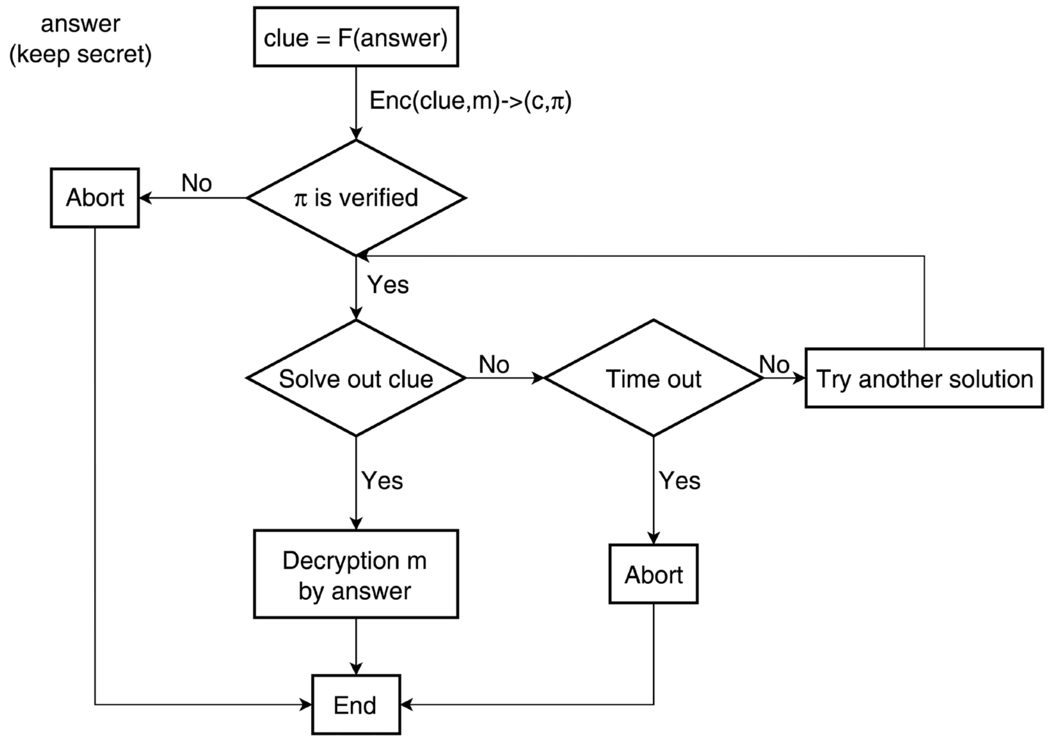
Setup$\left(\lambda, t, \omega\right) \rightarrow \left(answer, pp\right)$, where $pp = clue$ and $answer$ keeps secret. This is a randomized algorithm that takes a security parameter λ, the desired puzzle difficulty ω and the given time t to solve the puzzle which makes it possible to solve τ under this difficulty (It is not always possible to solve the answer to a difficult problem within t, but there is a certain probability that the answer can be calculated). Produce public parameters ***pp*** that consists of an encryption key $clue$ and a decryption key $answer, clue = OWF.Eval\left(answer\right)$. OWF (One way function) in Setup is a one-way function that satisfied τ and ω. This article requires Setup to be polynomial-time in λ. By convention, the public parameters specify a plaintext space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$. This article assumes that $\mathcal{M}$ is efficiently sampleable.

Enc$\left(clue, m\right) \rightarrow \left(c, \pi\right)$ takes a message $m \in \mathcal{M}$ and produces a ciphertext $c \in \mathcal{C}$ and a proof π for the statement that c is m encrypted by $clue$ and $clue$ is consistent with $answer$ (Only as a check during transmission).

GuessingRiddle $\left(clue, t, \omega, c, \pi\right) \rightarrow \left\{answer, Fail\right\}$

1. Many participants try to compete for the receiver position. If π is verified, it means that the sent ciphertext is meaningful and has not been tampered with. At this time, proceed to the next step. Otherwise, abort.

Figure 1.
The framework of our RiddleEncryption scheme



2. Under the limited time t and the desired puzzle difficulty ω, by comparing with $clue$, some participants can get the answer within the time t and proceed to the decryption phase. Participants who did not get an answer will get $Fail$ and abort.

***Dec**$\left(\boldsymbol{answer}, \boldsymbol{c}\right) \rightarrow \boldsymbol{m}$ uses the answer guessing into the **GuessingRiddle** to decrypt c to get the plaintext m.*

Remark. In **GuessingRiddle**, all participants can compete for receivers. At this time, participants can verify whether they have obtained the correct $answer$ (by comparison with $clue$) without interacting with the sender or a trusted third party. If the receiver needs to manipulate the plaintext further (for example, use the plaintext to issue a certificate to other users), it needs to generate a proof of knowledge of the $answer$ to prove its own capability. If the participant only uses the decryption to obtain the plaintext as the ultimate goal, then he does not need to prove its capability to anyone, but it can also privately verify whether it has the capability specified by the sender (use $answer$ comparing with $clue$).

In a RE, the adversary has no advantage to access the decryption oracle. Obviously, it satisfies the IND-CPA (Indistinguishability under chosen-plaintext attack) security within the difficulty of ω and the time t.

A RE has correctness, soundness, and receiver strong anonymity.

Correctness and Soundness. For correctness, it is necessary to ensure that everyone who guesses the $answer$ in **GuessingRiddle** can correctly decrypt the ciphertext in the **Dec** stage. For soundness, it is necessary to ensure 1) If it is not the ciphertext obtained by $clue$ in **Enc**, even if the $answer$

satisfies the $clue$ the receiver cannot decrypt correctly; 2) Participants who have not guessed the correct $answer$ cannot decrypt correctly. More formally,

Definition4.(Correctness)AREiscorrectifforall λ, t, ω *,parameters* $\left(answer, clue\right) \leftarrow Setup\left(\lambda, t, \omega\right)$, *and all* $\boldsymbol{m} \in \boldsymbol{\mathcal{M}}$, *if* $\left(c, \pi\right) \leftarrow Enc\left(clue, m\right)$ *and* $answer \leftarrow Gues\sin gRiddle\left(clue, t, \omega, c, \pi\right)$ *then* $\boldsymbol{Dec}\left(\boldsymbol{answer}, \boldsymbol{c}\right) = \boldsymbol{m'}$ *and* $\boldsymbol{m} = \boldsymbol{m'}$.

Definition 5. (Soundness) A RE is sound if for all algorithms $\boldsymbol{\mathcal{A}}$ that run in time $\boldsymbol{\mathcal{O}}\left(poly\left(t, \lambda, \omega\right)\right)$

$$Pr[c \neq \boldsymbol{Enc}\left(\boldsymbol{clue}, \boldsymbol{m}\right), m = \boldsymbol{Dec}\left(\boldsymbol{answer}, \boldsymbol{c}\right) | \left(answer, pp\right) \leftarrow \boldsymbol{Setup}\left(\lambda, t, \omega\right), answer \leftarrow \mathcal{A}\left(clue, t, \omega, c, \pi\right)] \leq negl\left(\lambda\right) \cdot$$

and

$$Pr[clue \neq OWF.Eval\left(answer'\right), m = \boldsymbol{Dec}\left(answer', c\right) | pp = \left(answer, pp\right) \leftarrow \boldsymbol{Setup}\left(\lambda, t, \omega\right), c \leftarrow \boldsymbol{Enc}\left(\boldsymbol{clue}, \boldsymbol{m}\right), answer' \leftarrow \mathcal{A}\left(clue, t, \omega, c, \pi\right)] \leq negl\left(\lambda\right)$$

Receiver Strong Anonymity. This article calls the security property needed for a RE scheme receiver strong anonymity. The sender does not encrypt for a specific person when encrypting, but for people who have the capability to guess the riddle through $clue$ within a specified time, so the sender will not know who the receiver is. However, the receiver can determine whether to fd the $answer$ to the riddle through $clue$, without interacting with the sender to reveal his identity. That is to say, anyone who has the capability to guess the $answer$ can be a receiver, and it is a negligible probability to determine the identity of the receiver through $clue$ in advance to launch a DDoS attack.

Assume that $\mathcal{A}$ controls n_{Adv} parties, and there are n_{total} parties in the system who can guess the $answer$ correctly in the same probability. This article defines the following game applied to an adversary $\mathcal{A} : \left(\mathcal{A}_0, \mathcal{A}_1\right)$:

$pp \leftarrow Setup\left(\lambda, t, \omega\right)$ // choose a random pp

$m \leftarrow \mathcal{M}$ // choose a random message m

$\left(c, \pi\right) \leftarrow \boldsymbol{Enc}\left(\boldsymbol{clue}, \boldsymbol{m}\right)$ // encrypt m by $clue$

$\mathcal{L} \leftarrow \mathcal{A}_0\left(clue, c, \pi, t, \omega\right)$ // adversary preprocesses $clue, c, \pi$

$\left\{\mathrm{ID}\right\}'_{\mathrm{receivers}} \leftarrow \mathcal{A}_1\left(\mathcal{L}, \mathrm{clue}\right)$ // adversary computes the set of receivers

It says that $\left(\mathcal{A}_0, \mathcal{A}_1\right)$ wins the game if $\left\{\mathrm{ID}\right\}_{\mathrm{receivers}} = \left\{\mathrm{ID}\right\}'_{\mathrm{receivers}}$, $\left\{\mathrm{ID}\right\}_{\mathrm{receivers}}$ is the identity of the participants who found the *answer* within the specified time t. (The participants don't know each other who found the *answer*, and the adversary can only know whether the participants under her control got the *answer*.), i.e., $\mathcal{A}$ can guess correctly more than $\frac{\mathrm{n}_{\mathrm{Adv}}}{\mathrm{n}_{\mathrm{total}}} + \epsilon$ which means $\mathcal{A}$ need to guess all the receivers' identities correctly.

Definition6.(ReceiverStrongAnonymity) ARE scheme $\boldsymbol{RE} = \left(\boldsymbol{Setup}, \boldsymbol{Enc}, \boldsymbol{GuessingRiddle}, \boldsymbol{Dec}\right)$ is receiver strong anonymity if, for every constant $\epsilon > 0$, no feasible adversary can win the above game with non-negligible probability in λ.

Remark. Assuming that $\mathcal{A}$ can calculate the $answer$ corresponding to the $clue$ in the $\mathcal{A}_0$ stage. At this time, $\mathcal{A}$ can use this $answer$ to compare with the $\mathrm{n}_{\mathrm{Adv}}$ parties controlled by $\mathcal{A}$. If they have this $answer$, they are the receivers. However, apart from blindly guessing the identity of the wild parties (do not consider social engineering and other factors here), $\mathcal{A}$ has no other advantage to guess their identity. The bound of participants will be specifically defined in the analysis part.

At the same time, there is also a problem that after time t, some participants do not stop solving difficult problems. Although there is no way to enforce that participants stop trying to solve for m, the sender can update the plaintext sent after time t so that the original plaintext is no longer valuable, i.e., even after time t, the participant found the $answer$ and decrypted the plaintext, the plaintext is meaningless. The original secret can be made invalid by key management technologies such as secret revocation. This update mechanism using in our **Competitive-Evolving-Committee PSS** and the analysis of the update time $\mathrm{t}_{\mathrm{update}}$ also described in the analysis part.

Construction by ElGamal Encryption

This section the authors use the ElGamal encryption scheme and NIZK to instantiate the RiddleEncryption to become $RE_{ElGamal}$.

```
$\boldsymbol{RE}_{ElGamal}$ **Setup**
**Input:** $\lambda, t, \omega$.
**Output:** Keep $clue$ public and $answer$ secret.
**Setup**$(\lambda, t, \omega)$:
1. Compute $pp_{NIZK} \to NIZK.Setup(\lambda, t)$.
2. Let the secret key of ElGamal is $x \in \mathcal{Z}_p^*$, $p$ is a prime number in the ElGamal system, and the public key is $y = g^x \bmod p$, $g$ is the random number less than $p$.
3. Let $answer := x$ and $clue := (y, pp_{NIZK})$.
4. Output $clue$ and keep $answer$ secret.
$\boldsymbol{RE}_{ElGamal}$ **Encryption**
**Input:** $clue$ and message $m$.
**Output:** Ciphertext $c$ and proof $\pi$.
**Enc**$(\boldsymbol{clue}, \boldsymbol{m})$:
1. $m \in \mathcal{M}$ is the message to encrypt and choose a random $r \leftarrow Z_p^*$.
2. Compute $c := (C_1, C_2) = (g^r \bmod p, y^r m \bmod p)$.
3. $NIZK.Prove(pp_{NIZK}, c, r, answer) \to \pi$, $\pi$ is proven the statement that $c$ is $m$ encrypted by $clue$ and $clue$ is consistent with $answer$ (Only as a check during transmission).
4. Output $(c, \pi)$.
$\boldsymbol{RE}_{ElGamal}$ **GuessingRiddle**
**Input:** $clue, t, \omega, c, \pi$.
**Output:** Fail or keep the $answer$ secret.
**GuessingRiddle** $(clue, t, \omega, c, \pi)$:
1. All the participants call $NIZK.Verify(pp_{NIZK}, c, \pi)$, if output $No$ then abort else go to next step.
2. All the participants use discrete logarithm problem (DLP)
```

solving algorithms in different environments in the limited time t and the specific puzzle difficulty ω.
3. After time t, some participants get the $answer$ (can check with $clue$ by themselves) and others get $Fail$. The one who gets $Fail$ aborts, and the one who gets the $answer$ goes to the **Dec** phase.

$\boldsymbol{RE}_{ElGamal}$ **Decryption**
Input: $answer, c$.
Output: Message m.
Dec $\left(\boldsymbol{answer}, \boldsymbol{c}\right)$: $m = C_2 C_1^{-answer} \bmod p$

Theorem 1. $\boldsymbol{RE}_{\boldsymbol{ElGamal}}$ is a $\boldsymbol{RE}$ scheme with correctness, soundness, and receiver strong anonymity.

Proof. Proof of Theorem 1

Correctness: When the participants get the $clue\left(c, \pi\right)$ and the parameters, after passing the NIZK proof verification and time $\boldsymbol{t}$, some of them work out the **answer** and then $\mathbf{C_2 C_1^{-answer} \bmod p = \left(y^r m\right)\left(g^r\right)^{-answer} \bmod p = \left(g^{answer \cdot r} M\right)\left(g^r\right)^{-answer} \bmod p = m \bmod p}$

Soundness: Assuming there exists an adversary $\boldsymbol{\mathcal{A}}$ with

$$\boldsymbol{Pr}[\boldsymbol{c} \neq \boldsymbol{Enc}\left(\boldsymbol{clue}, \boldsymbol{m}\right), \boldsymbol{m} = \boldsymbol{Dec}\left(\boldsymbol{answer}, \boldsymbol{c}\right) \mid \boldsymbol{pp} = \left(\boldsymbol{answer}, \boldsymbol{clue}\right) \leftarrow Setup\left(\lambda, t, \omega\right), answer \leftarrow \boldsymbol{\mathcal{A}}\left(clue, t, \omega, c, \pi\right)] \geq poly\left(\lambda\right)$$

It is in contrast with the correctness of ElGamal. Or with

$$\boldsymbol{Pr}[\boldsymbol{clue} \neq \boldsymbol{OWF.Eval}\left(\boldsymbol{answer'}\right),\ \boldsymbol{m} = \boldsymbol{Dec}\left(\boldsymbol{answer'}, \boldsymbol{c}\right) \mid \boldsymbol{pp} = \left(\boldsymbol{answer}, \boldsymbol{clue}\right) \leftarrow Setup\left(\lambda, t, \omega\right), c \leftarrow Enc\left(clue, m\right),\ answer' \leftarrow \boldsymbol{\mathcal{A}}\left(clue, t, \omega, c, \pi\right)] \geq poly\left(\lambda\right)$$

this means

$$\boldsymbol{Pr}[\boldsymbol{clue} \neq \boldsymbol{OWF.Eval}\left(\boldsymbol{answer'}\right) \mid \boldsymbol{pp} = \left(answer, clue\right) \leftarrow Setup\left(\lambda, t, \omega\right), c \leftarrow Enc\left(clue, m\right), answer' \leftarrow \boldsymbol{\mathcal{A}}\left(clue, t, \omega, c, \pi\right)] \geq poly\left(\lambda\right)$$

and

2) $$\boldsymbol{Pr}[\boldsymbol{m} = \boldsymbol{Dec}\left(\boldsymbol{answer'}, \boldsymbol{c}\right) \mid \boldsymbol{pp} = \left(answer, clue\right) \leftarrow Setup\left(\lambda, t, \omega\right), c \leftarrow Enc\left(clue, m\right), answer' \leftarrow \boldsymbol{\mathcal{A}}\left(clue, t, \omega, c, \pi\right)] \geq poly\left(\lambda\right)$$

For 1): Assuming that $\boldsymbol{\mathcal{A}}$ can find the **answer** (using kangaroo algorithm or something else), then 1) means that $\boldsymbol{\mathcal{A}}$ has already known $\mathbf{clue = g^x}$ but got the inverse of $\mathbf{g^{x'}}$ which implies the inverse algorithm (like kangaroo algorithm) fail to find the inverse in a high probability. It contrasts with the setting of the parameters (Details in Analysis).

For 2): It contrasts with the correctness of ElGamal.

Receiver Strong Anonymity: It is assumed that $\mathcal{A}$ can calculate the ***answer*** corresponding to the ***clue*** in the $\mathcal{A}_0$ stage. At this time, $\mathcal{A}$ can use this **answer** to compare with the $\mathbf{n}_{\mathrm{Adv}}$ parties controlled by A. If they have this **answer**, they are the receivers. However, if $\mathcal{A}$ has a non-negligible probability of guessing the receivers' IDs more than $\frac{\mathbf{n}_{\mathrm{adv}}}{\mathbf{n}_{\mathrm{total}}} + \epsilon$ with difficulty ω and time $\boldsymbol{t}$, then it indicates that $\mathcal{A}$ has other methods besides blind guessing to find the information of receivers who have never sent relevant messages. Unless the receivers have sent relevant messages, $\mathcal{A}$ cannot obtain more relevant information, so this contradicts with the scheme which defined receiver only sends one message once before receiving the secret but after initialization. The bound of participants will be specifically defined in Analysis.

OUR COMPETITIVE-ECPSS SCHEME

This section will use the **RiddleEncryption** instantiated by the ElGamal encryption scheme, secret sharing, NIZK, and Ouroboros in our scheme to construct a Competitive-Evolving-Committee PSS scheme. The framework of the solution shows in Figure 2.

Model and Definitions

This section proposed a secret sharing scheme that holding members who can be generated independently without any other parties to nominate and also does not require centralized infrastructure such as PKI. The name of it is Competitive-ECPSS (CECPSS).

Definition 7. A competitive-evolving-committee proactive secret sharing scheme consists of the following procedures:

Trusted Setup(optional). *Provide initial state for a universe of k_0 parties;*

Sharing. *Among an initial holding committee of size k_0, each of them choose a secret share $\sigma_{i_0}, i_0 \in [1, k_0]$. These shares can be interpolated in a degree-($th - 1$) polynomial F_0 with $F_0(0) = \sigma$, where th is the threshold and σ is the global secret.*

Committee-Competitive-Selection-Part1.(CCS-Part1) Determine system parameters which consist next round's holding committee members' public key but cannot map to the specific node and the size of the next round's holding committee in a distributed way. This protocol runs among all the participants in the system, whether they are holding committee members or not.

Handover-Part1.(H-Part1) Previous round holding committee members re-share their share, encrypt the share and broadcast on the channel.

Committee-Competitive-Selection-Part2.(CCS-Part2) Decide who is the holding committee member of the next round based on the capability of the participants in the next round using **RiddleEncryption**. The committee-competitive-selection process diagram shown in Figure 3.

Handover-Part2.(H-Part2)} Next-round holding committee members decrypt the ciphertexts for them and get the share of them.

Reconstruction. *Take more than th shares from the current holding committee and reconstruct the global secret σ if necessary, depending on the application scenario of the protocol.*

Construction

Now the authors put everything described above together to form the **Competitive-Evolving-Committee PSS Protocol (CECPSS)** below.

Figure 2.
The framework of our Competitive-ECPSS

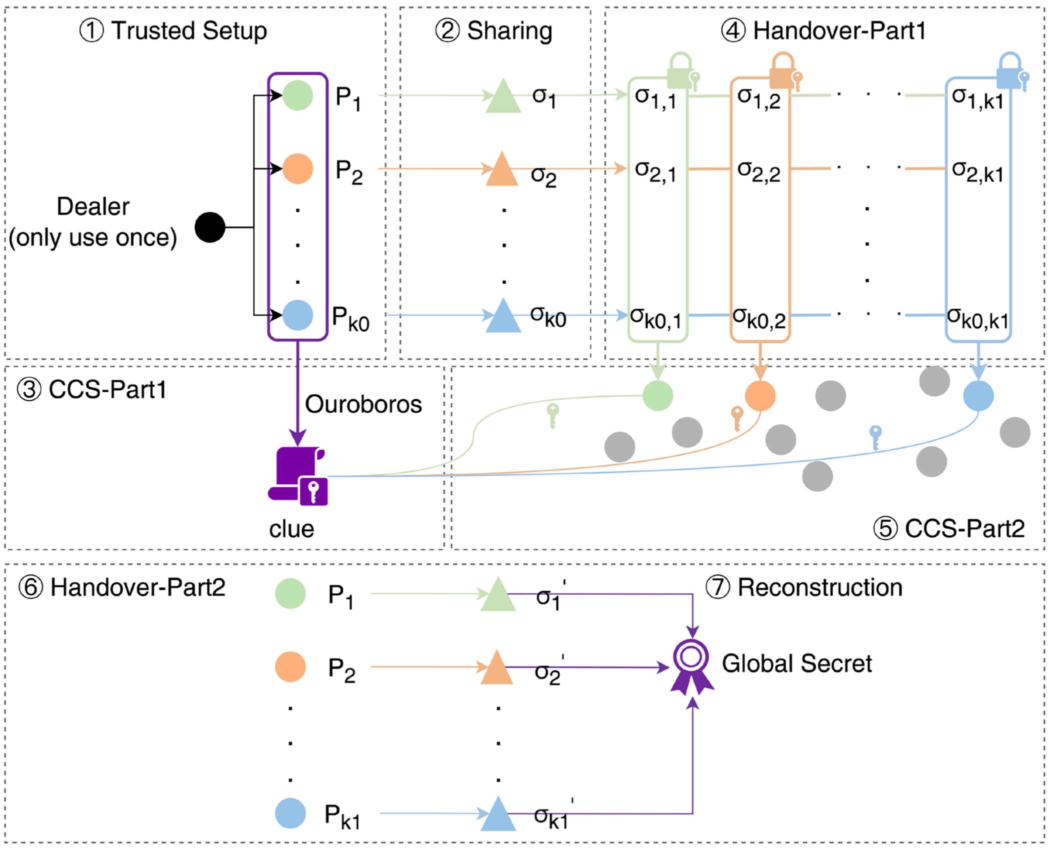
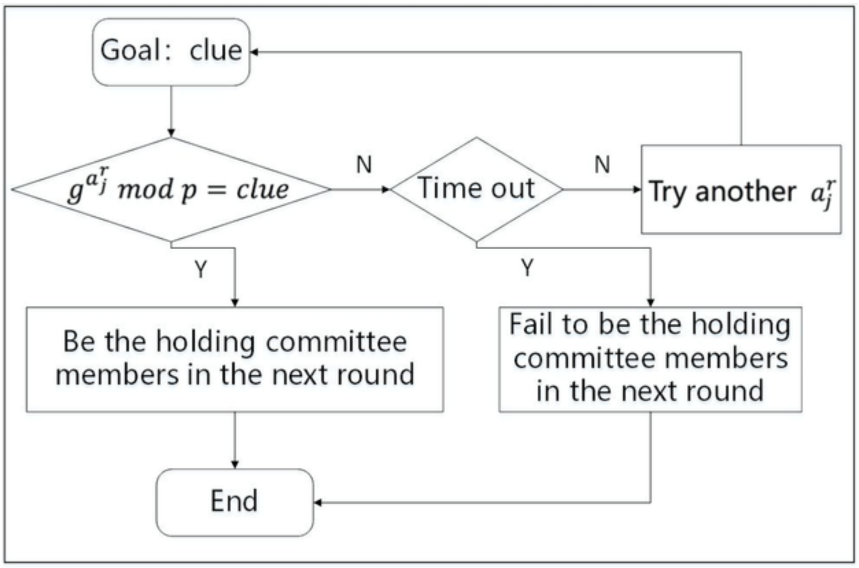


Figure 3.
Committee competitive selection process


Goal: clue
$g^{a_j^r} \, mod \, p = clue$
N
Time out
N
Try another a_j^r
Y
Y
Be the holding committee members in the next round
Fail to be the holding committee members in the next round
End

Algorithm 1.
Competitive-Evolving-Committee PSS Protocol

Input:	t : time to find the answer; E : the desired puzzle difficulty; $\mathrm{n}_{\mathrm{total}}$: the bound of the number of participants; $\mathrm{t}_{\mathrm{update}}$: time to update the global secret; distributed nodes who want to participate in;
Output:	global secret reconstruction for specific scenarios;
	# Dealer do
	# Trusted Setup
1:	round = 0;
2:	$P_1^{round}, P_2^{round}, \dots, P_{k_{round}}^{round} \leftarrow$ DealerRandomChooseHolders(k_{round}, r_{dealer});
3:	$b_1, b_2, \dots, b_{k_{round}} \leftarrow$ DealerGiveHoldersToken($Z_p^*, P_1^{round}, P_2^{round}, \dots, P_{k_{round}}^{round}$);
	# only distribute the token at the Setup Stage;
	# $Holders_{round}$ do (each holder do this loop at the same time)
4:	**for** $i = 1; i \le k_{round}; i++$ **do**
	#Sharing
5:	$\sigma_i^{round} \leftarrow$ ShareGeneration($Z_p^*; r_{P_i^{round}}$);
	# CCS-Part1 (Using Ouroboros to get pp and $clue$ respectively)
6:	$k_{round+1} \leftarrow$ Ouroboros($\lambda_{round+1}$);
7:	**for** $j = 1; j \le k_{round+1}; j++$ **do**
8:	$pp_j^{round} \leftarrow Ouroboros\left(\lambda_{RE_j}\right)$; $clue_j^{round} \leftarrow Ouroboros\left(\lambda_{RE_j}\right)$;
9:	**end for**
	#H-Part1
10:	$\sigma_{i,1}^{round}, \sigma_{i,2}^{round}, \dots, \sigma_{i,k_1}^{round} \leftarrow$ SecretSharing(σ_i^{round});
11:	**for** $j = 1; j \le k_{round+1}; j++$ **do**
12:	$ct_{i,j}^{round} \leftarrow$ RE.Enc($pp_j^{round}, clue_j^{round}, \sigma_{i,j}^{round}$);
13:	$\pi_i^{round} \leftarrow$ NIZK.Prove($ct_{i,j}^{round}, pp_j^{round}, clue_j^{round}, \sigma_{i,j}^{round}$);
14:	**end for**
15:	$\pi_i^{token} \leftarrow$ NIZK.Prove(b_i);

continued on following page

Algorithm 1.
Continued

16:	SecurityErase(σ_i^{round});
17:	Broadcast($ct_{i,1}^{round},\ldots,ct_{i,k_1}^{round},\pi_i^{round},\pi_i^{token}$);
18:	**end for**
	# All the participants do (each participant do at the same time)
	# CCS-Part2
19:	**for** $i=1;i\le k_{round};i++$ **do**
20:	**if** NIZK.Verify(π_i^{round})==FALSE OR NIZK.Verify(π_i^{token})==FALSE
21:	Discard($ct_{i,1}^{round},\ldots,ct_{i,k_{round+1}}^{round}$);
22:	**end if**
23:	**end for**
24:	**for** $n=1;n\le n_{total};n++$ **do**
25:	$s_n \leftarrow Z_j^*$; # Choose one of the RE systems parameters trying to guess the riddle
26:	result $\leftarrow$ RE_{s_n} .GuessingRiddle($clue_{s_n}^{round},t,\omega$);
27:	**if** result==Fail **then**
28:	Abort;
	# H-Part2
29:	**else**
	# Number n participant win the place of holder of next round
30:	**for** $i=1;i\le k_{round};i++$ **do**
31:	$\sigma_{i,s_n}^{round} \leftarrow RE_{s_n}.Dec\left(result,ct_{i,s_n}^{round}\right)$;
32:	**end for**
33:	$\sigma_{s_n}^{round+1} \leftarrow$ ShareReconstruction($\sigma_{1,s_n}^{round},\sigma_{2,s_n}^{round},\ldots,\sigma_{k_0,s_n}^{round}$);
34:	**end if**
35:	**end for**
	# $\mathrm{Holders}_{\mathrm{round+1}}$ do
	# Reconstruction

continued on following page

Algorithm 1.
Continued

36:	$\sigma_{global} \leftarrow$ GlobalShareReconstruction($\sigma_1^{round+1}, \sigma_2^{round+1}, \ldots, \sigma_{k_1}^{round+1}$); (Or doing the downstream work.)
37:	**if** t_{update} is reached **then**
38:	change the next round σ_{global} and secure erase the current round σ_{global} ;
39:	return to Trusted Setup;
40:	**else**
41:	round++;
42:	return CCS-Part1 for other rounds;
43:	**end if**

ANALYSIS FOR OUR COMPETITIVE-ECPSS SCHEME

In this section, the authors give the correctness of the secret-sharing in **Handover** phase and the complexity and security proof of the **Competitive-Evolving-Committee PSS**.

Correctness

The global secret can be recover correctly by using Lagrangian interpolation.

Let $\mathbf{F}_\mathbf{r}$ is the polynomial which shares the global secret σ and λ_i is the Lagrange coefficients for $\mathbf{F}_\mathbf{r}$, where $\mathbf{i} \in \left[1, \mathbf{K}_\mathbf{r}\right]$, $\mathbf{K}_\mathbf{r}$ is the size of the holding committee in the $\mathbf{r}$ -th round.

Let $\mathbf{G}_\mathbf{i}$ is the polynomial which is the $\mathbf{r}$ -th round holding committee members choose for the $\mathbf{r}+1$ -th round committee members and $\mu_{i,j}$ is the Lagrange coefficients for $\mathbf{G}_\mathbf{i}$, where $\mathbf{j} \in \left[1, \mathbf{K}_{\mathbf{r}+1}\right]$, $\mathbf{K}_{\mathbf{r}+1}$ is the size of the holding committee in the $\mathbf{r}+1$ -th round.

Then have the global secret:

where $\mu_{i,j} \cdot \lambda_i$ is the Lagrange coefficients for the polynomial $\mathbf{F}_{\mathbf{r}+1}$, the correctness has been proved.

Complexity

Through the Construction part, it is easy to see that the communication complexity of the whole protocols is fixed polynomial with the security parameter, regardless of the number of rounds or the number of parties $\boldsymbol{n}_{total}$. According to the bound of the parties in the committee, there are only $\mathcal{O}\left(\lambda\right)$ parties and each of them only send a single message for sending the puzzle. The computation complexity performed by each party depends on the methods they solving the difficult problems.

Bound of Time and the Number of Participants

To meet the feasibility and stability of the system, it is necessary to set the range of the parameter.

Theorem 2. (The bound of time $\boldsymbol{t}$) For the system where the underlying difficult problem is the discrete logarithm problem, the time it can be used to solve the discrete logarithm problem should be limited to $\left(1.63 + \boldsymbol{o}\left(1\right)\right)\sqrt{\boldsymbol{N}} < \boldsymbol{t} < \left(2 + \boldsymbol{o}\left(1\right)\right)\sqrt{\boldsymbol{N}}$, where $\boldsymbol{N}$ is the order of the group in the discrete logarithm problem.(B. Qi et al., 2020) The size of the interval in which the exponent is ω and $\sqrt{\omega}$ is smaller than the largest computing power scale so far but $\omega > poly\left(t\right)$.(Fowler & Galbraith, 2015; S. D. Galbraith et al., 2012; B. Qi et al., 2020)

Remark. Restricting t can make sure that at least one participant tries to guess on riddle within the time limit and in the typical scenarios not all the participants can guess the riddle.

Proof. Proof of Theorem 2

In a given time t, some participants are required to solve the discrete logarithm problem (DLP), and some participants cannot. In an extreme state, a single processor solves the DLP serially, which is a state that any participant can reach (as long as they have a PC). Therefore, to prevent everyone from solving the DLP, it is necessary to set the time shorter than the time complexity of the algorithm used by everyone. The more efficient algorithm for solving DLP serially is the Kangaroo algorithm, and its time complexity is $\left(2+\mathbf{o}\left(1\right)\right)\sqrt{\mathbf{N}}$ (Fowler & Galbraith, 2015; S. Galbraith et al., 2013).

At the same time, t cannot tend to 0. If so, no one can solve it. To ensure the system's feasibility, i.e., someone needs to be a member of the holding committee in the next round, so the time needs to be set longer than the time complexity of the fastest algorithm-solving DLP so far. The current efficient algorithm is an improvement of the Kangaroo algorithm, and its time complexity is $\left(1.633+\mathbf{o}\left(1\right)\right)\sqrt{\mathbf{N}}$ (B. Qi et al., 2020; Bin Qi et al., 2020).

For the interval size of ω, considering the system's feasibility, i.e., it is possible to solve the exponent within time t. According to the conclusion given by the birthday paradox, $\sqrt{\omega}$ must be the magnitude that is easy to handle, so $\sqrt{\omega}$ is smaller than the most extensive scale that current computers can handle. At the same time, to prevent brute force cracking by a single processor, $\omega > poly\left(t\right)$. Because there is no single processor that can process a scale larger than $\mathbf{poly}\left(\mathbf{t}\right)$ in a sufficient time, this would lead to a single processor that can solve some difficult problems violently, which contradicts the current difficult assumptions.

Theorem 3. (The bound of the number of participants $\boldsymbol{n}_{total}$) Let $\boldsymbol{n}_{Adv}$ is the number of malicious parties, $\boldsymbol{n}_{Honest}$ is the number of honest parties, $\boldsymbol{k}$ is the number of the holding committee members and $\boldsymbol{th}$ is the threshold to reconstruct the secret. There is $\boldsymbol{n}_{Adv} < \boldsymbol{th}$,

$$\boldsymbol{n}_{Honest} + \boldsymbol{n}_{Adv} > 2\cdot \boldsymbol{th} + 1, \boldsymbol{n}_{Honest} \geq 1 + \boldsymbol{th}\cdot\left[1 + \boldsymbol{ln}\left(\boldsymbol{th} - 1\right)\right]$$

so

$$\boldsymbol{n}_{Honest} + \boldsymbol{n}_{Adv} = \boldsymbol{n}_{total} \geq \boldsymbol{MAX}\left\{2\cdot \boldsymbol{th} + 1, 1 + \boldsymbol{th}\cdot\left[1 + \boldsymbol{ln}\left(\boldsymbol{th} - 1\right)\right]\right\}.$$

Remark. Although the article say "$\mathbf{n}_{\mathrm{Adv}} < \mathbf{th}$", the adversary model is not weaker than the model of ECPSS. As in ECPSS, $\mathbf{n}_{\mathrm{Adv}}$ should be less than th; otherwise it is meaningless.

Proof. Proof of Theorem 3

The secret sharing used in the system is proactive secret sharing. According to its fault tolerance requirements, there are $\mathbf{n}_{\mathrm{Adv}} < \mathbf{th}$ and $\mathbf{n}_{\mathrm{total}} > 2\cdot\mathbf{th} + 1$.

The honest parties in the system succeed in solving the difficult problem with a probability of $1/\mathbf{k}$, and the system can only continue when different honest parties find $\mathbf{th}$ answers, i.e., it is feasible. Therefore, there should not be too few honest parties, and the number of the honest parties required for $\mathbf{th}$ solutions to be found is

$$n_{Honest} = 1 + \frac{th}{th-1} + \frac{th}{th-2} + \frac{th}{th-3} + \ldots + \frac{th}{2} + \frac{th}{1}$$
$$= 1 + th \cdot \left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{th-2} + \frac{1}{th-1}\right)$$

where $1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{\mathrm{th}-2} + \frac{1}{\mathrm{th}-1}$ is the divergence series, therefore,

$$\int_1^{th} \frac{1}{x} dx < 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{th-2} + \frac{1}{th-1} < 1 + \int_1^{th-1} \frac{1}{x} dx$$
$$ln(th) < 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{th-2} + \frac{1}{th-1} < 1 + ln(th-1)$$
$$\therefore 1 + th \cdot ln(th) < n_{Honest} < 1 + th \cdot \left[1 + ln(th-1)\right]$$
$$\therefore n_{Honest} \geq 1 + th \cdot \left[1 + ln(th-1)\right] \text{ is feasiable.}$$

In summary $\mathbf{n}_{\text{total}} \geq \mathbf{MAX}\left\{2 \cdot \mathbf{th} + 1, 1 + \mathbf{th} \cdot \left[1 + \mathbf{ln}(\mathbf{th} - 1)\right]\right\}$.

Theorem 4. (The bound of time t_{update}) Re-enter the Trusted Setup phase after time $\left(1.633 + o(1)\right)\sqrt{N} \cdot k_0$ to update the global secret within a single malicious party, where N is the order of the group in the discrete logarithm problem and k_0 is the number of holding committee members in the Trusted Setup phase(Fowler & Galbraith, 2015; B. Qi et al., 2020).

Remark. If there are $n_{Adv} > 1$ malicious parties can solve the puzzle at the same time, then the update time will be $t = Max\left\{t_1, t_2, \ldots, t_{n_{Adv}}\right\}$, where $t_i, i \in \left[1, n_{Adv}\right]$ is the $i-$th malicious party's time of solving the puzzle. This time may be short, so our scheme has limitation in the case of more parallel malicious parties. This will involve related technologies such as key destruction in key management, and this article does not discuss them in detail.

Proof. Proof of Theorem 4

Suppose that the adversary does not participate in the subsequent competition, and certification work after the previous round holding committee members announce $\mathbf{ct}_{\mathbf{i,j}}^{\mathbf{round}}$ for the first time but continues to violently solve different groups of DLPs just to crack the global secret, so as to obtain all the shares and then recover the global secret. Even if the adversary uses the fastest algorithm to crack, its solution time is also $\left(1.633 + \mathbf{o}(1)\right)\sqrt{\mathbf{N}} \cdot \mathbf{k}_0$. So before this time, return to the Start phase to update the global secret can avoid this attack.

Security

Combining the above theorems, the authors get theorem 5.

Theorem 5. Consider the parameters $\left(1.633 + o(1)\right)\sqrt{N} > t > \left(2 + o(1)\right)\sqrt{N}$, $\sqrt{\omega}$ is smaller than the largest computing power scale so far but

$$\omega > poly(t), n_{total} \geq MAX\left\{2 \cdot th + 1, 1 + th \cdot \left[1 + \ln(th-1)\right]\right\},\ t_{update} < \left(1.633 + o(1)\right)\sqrt{N} \cdot k_0.$$

Let Ouroboros in our scheme is secure as (Kiayias et al., 2017) and RiddleEncryption satisfies correctness, soundness, and receiver strong anonymity. Also, let $\boldsymbol{A}$ be a NIZK argument system.

Then the construction is a Competitive-Evolving-Committee PSS scheme is secure in model with erasures and the broadcast channel.

Proof. Proof of Theorem 5

Assume the adversary chooses two global secrets σ_0, σ_1 and then interacts with our CECPSS protocol. If the adversary has a negligible advantage in guessing which of σ_0, σ_1 was shared, the protocol is secure. The authors use hybrid techniques to do the proof.

$\boldsymbol{H}_0$ (the real protocol): The challenger controls all the honest parties and holds the global secret and all the shares.

$\boldsymbol{H}_1$ (NIZK soundness): When the challenger receives the proof from the adversary, then aborts. Because the challenger holds the global secret and all the shares, he can detect whether the proof is correct. Assume the adversary successfully cheat the challenger, then there is

$$\Pr\left[\sigma \leftarrow CRS\left(1^{\lambda}\right); \boldsymbol{\mathcal{V}}^{\mathcal{O}}\left(\sigma, f\left(\sigma\right), P^{*\mathcal{O}}\left(\sigma\right)\right) = 1\right] > poly\left(\lambda\right).$$

It is contrast with the soundness of NIZK.

$\boldsymbol{H}_2$ (NIZK zero-knowledge): The challenger using the simulator of the NIZK to generate the proof. When the adversary sees the proof, as the zero-knowledge of NIZK, the adversary has a negligible probability of distinguishing the real proof or the proof generated by the simulator.

$\boldsymbol{H}_3$ (receiver strong anonymity of RE): When the challenger observes the honest parties are fewer than $\boldsymbol{th}$, then aborts. As the setting of the parameters, the malicious parties which the adversary controls should be fewer than $\boldsymbol{th}$. However, when it is more than $\boldsymbol{th}$, it means the adversary gets the identity of the holding committee members from the message, which only be sent once so that the adversary can control the receivers. It is in contrast with the receiver strong anonymity of RE.

$\boldsymbol{H}_4$ (RE secrecy): The challenger uses RE to encrypt σ', which is different from σ_0 and σ_1, and now the adversary has a negligible probability of distinguishing the ciphertext that is encrypted by σ' or by σ_0 or σ_1. This is the basic semantic security of the encryption scheme. (Now, it has already ensured the malicious members are fewer than $\boldsymbol{th}$).

The authors can undo the changes in these hybrids, arriving at a game where the adversary gets σ_{1-b} rather than σ_b.

Limitation

If there are $\mathbf{n}_{\mathbf{Adv}} > 1$ malicious parties can solve the puzzle at the same time, then the update time will be $\mathbf{t} = \mathbf{Max}\left\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{\mathbf{n}_{\mathbf{Adv}}}\right\}$, where $\mathbf{t}_{\mathbf{i}}, \mathbf{i} \in \left[1, \mathbf{n}_{\mathbf{Adv}}\right]$ is the i-th malicious party's time of solving the puzzle. This time may be short, so our scheme has limitations in the case of more parallel malicious parties. This will involve related technologies such as key destruction in key management, and the authors do not discuss them in detail.

Our scheme needs to update the secret periodically, which is the inherent flaw of it. The authors have given a simple solution, but it is not perfect enough, so in the next step, the authors will study more detailed strategies to solve this problem, which may involve secret updating and secret revocation technologies in the key management field.

APPLICATION

Dynamic Group. In some scenarios, specific group tasks need to be performed according to "capability". If only some participants are initially designated as group members, problems such as low work enthusiasm and lack of "capability" in the group may occur. Although there are also some researches on distributed tasks (Kate et al., 2009; Liu et al., 2007; Mwitende et al., 2020), there is no such competition mechanism of our scheme. Using our solution, the authors can make groups of "strong capability" participants within a specified time, so that the downstream work can be executed more efficiently. At the same time, in order to stimulate the continuous strengthening of "capability", dynamic grouping can prevent a certain degree of inaction within the group.

Authentication management. Nowadays, most websites need to register information with the server, but the adversary can know these server nodes somehow (like social engineering). If the adversary launches a DoS attack on the server or performs an off-database attack on its database, a large amount of personal information will leak. The essence of login is that the user needs an authorized certificate to prove that he is authenticated without storing so much sensitive information on a specific server. Then use our solution to allow users to register with a server group "floating" in cyberspace. The members of this group will be updated without storing too much sensitive information on a specific server. Moreover, since the group members are specified through competition, these group members will cherish the hard-won opportunities to maintain relevant information more seriously under the assumption of rational people.

Prevent abuse of rights. A decentralized system can improve the enthusiasm of participants through the protocol proposed in this article, forming a situation where "everyone is the backbone". Encourage the emphasis on "capability" rather than "right" to prevent abuse and arbitrary of rights.

Most places that require a trusted third party can be replaced by our solution so that everyone has the opportunity to become a member of a trusted third party, which providing opportunities for more people and being fairer and in addition, it can avoid the single point of failure and excessive trust.

CONCLUSION AND FUTURE WORK

The authors propose a **Competitive-Evolving-Committee PSS** with a competition mechanism and receiver strong anonymity using our capability-based encryption scheme **RiddleEncryption** and other primitives. Our secret sharing scheme is divided into 7 functions: Trusted Setup, Sharing, CCS-Part1, Handover-Part1, CCS-Part2, Handover-Part2, and Reconstruction. The main innovation is the CCS and Handover part, using **RiddleEncryption**, the core of the difficult problem solving, which competes with each participant's ability. It is a capability-based scheme. Holding committee members only send one message in the life cycle, and they can send it to the corresponding receiver who only knows the receiver's capabilities. Therefore, it will not reveal the identity of the receivers.

In the future, our **RiddleEncryption** scheme can be used not only for the construction of this secret-sharing scheme, but also as a primitive for other schemes that need to guarantee receiver anonymity. **Competitive-Evolving-Committee PSS** can be used as an alternative protocol to the current trusted third party and can also be used as the underlying foundation for other applications of secret sharing technology. Our solution can mitigate excessive trust, provide more anonymity and fairness, and minimize the risk of being attacked like DDoS. Therefore, it has a wide range of application scenarios, such as probate management, medical data sharing, and federal learning, etc.

ACKNOWLEDGMENT

Conflict of Interest

The authors of this publication declare there is no conflict of interest.

Funding Agency

This work is supported by National Key RD Program of China (2017YFB0802500), Guangxi Key Laboratory of Cryptography and Information Security (GCIS201909), Beijing Municipal Science and Technology Project (No. Z191100007119007) and Shandong province major science and technology innovation project (2019JZZY020129).

REFERENCES

Al-Ghamdi, M., Al-Ghamdi, M., & Gutub, A. (2019). Security enhancement of shares generation process for multimedia counting-based secret-sharing technique. *Multimedia Tools and Applications*, *78*(12), 16283–16310. doi:10.1007/s11042-018-6977-2

Al-Qurashi, A., & Gutub, A. (2018). Reliable secret key generation for counting-based secret sharing. *J Comput Sci Comput Math*, *8*(4), 87–101. doi:10.20967/jcscm.2018.04.006

Alaseri, K., & Gutub, A. (2018). Merging secret sharing within Arabic text steganography for practical retrieval. *International Journal of Research Development Organisation -. Journal of Computing Science and Engineering: JCSE*, *4*, 1–17.

AlKhodaidi, T., & Gutub, A. (2020). Trustworthy target key alteration helping counting-based secret sharing applicability. *Arabian Journal for Science and Engineering*, *45*(4), 3403–3423. doi:10.1007/s13369-020-04422-9

Amarasinghe, N., Boyen, X., & Mckague, M. (2021). The cryptographic compexity of anonymous coins A systematic exporation. *Cryptography*, *5*(1), 10. doi:10.3390/cryptography5010010

Asmuth, C. B. J., & Bloom, J. (1983). A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, *29*(2), 208–210. doi:10.1109/TIT.1983.1056651

Baron, J., Defrawy, K. E., Lampkins, J., & Ostrovsky, R. (2015, June). Communication-optimal proactive secret sharing for dynamic groups. In *International Conference on Applied Cryptography and Network Security*, (pp. 23-41). Springer. doi:10.1007/978-3-319-28166-7_2

Benhamouda, F., Gentry, C., Gorbunov, S., Halevi, S., Krawczyk, H., Lin, C., & Reyzin, L. (2020, November). Can a public blockchain keep a secret? Paper presented at the *Theory of Cryptography Conference*. Springer. doi:10.1007/978-3-030-64375-1_10

Blakley, G. R. (1979). Safeguarding cryptographic keys. Paper presented at *the Managing Requirements Knowledge, International Workshop on*. IEEE. doi:10.1109/MARK.1979.8817296

Blum, M. (1988). Non-Interactive Zero-Knowledge and Its Applications. Paper presented at the *Proc. 20th STOC*. IEEE.

Blundo, C., Cresti, A., De Santis, A., & Vaccaro, U. (1996). Fully dynamic secret sharing schemes. *Theoretical Computer Science*, *165*(2), 407–440. doi:10.1016/0304-3975(96)00003-5

Dolev, S., Garay, J., Gilboa, N., & Kolesnikov, V. (2010). Brief Announcement: Swarming Secrets. Paper presented at the *ACM Sigact-sigops Symposium on Principles of Distributed Computing*. ACM. doi:10.1145/1835698.1835750

Fowler, A., & Galbraith, S. (2015). Kangaroo Methods for Solving the Interval Discrete Logarithm Problem. *Computer Science*.

Galbraith, S., Pollard, J., & Ruprai, R. (2013). Computing discrete logarithms in an interval. *Mathematics of Computation*, *82*(282), 1181–1195. doi:10.1090/S0025-5718-2012-02641-X

Galbraith, S. D., Pollard, J. M., & Ruprai, R. S. (2012). Computing discrete logarithms in an interval. *Mathematics of Computation*, *82*(282), 617. doi:10.1090/S0025-5718-2012-02641-X

Gamal, T. E. (1984). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, *31*, 469–472.

Gutub, A., & Al-Ghamdi, M. (2019). Image based steganography to facilitate improving counting-based secret sharing. *3D Research, 10*, 1-36.

Gutub, A., & Al-Ghamdi, M. (2020). Hiding shares by multimedia image steganography for optimized counting-based secret sharing. *Multimedia Tools and Applications*, *79*(11-12), 7951–7985. doi:10.1007/s11042-019-08427-x

Gutub, A., Al-Juaid, N., & Khan, E. (2019a). Counting-based secret sharing technique for multimedia applications. *Multimedia Tools and Applications*, *78*(5), 5591–5619. doi:10.1007/s11042-017-5293-6

Gutub, A., & Al-Qurashi, A. (2020). Secure shares generation via M-blocks partitioning for counting-based secret sharing. *Journal of Engineering Research*, *8*(3), 91–117. doi:10.36909/jer.v8i3.8079

Gutub, A., & Alaseri, K. J. A. J. S. (2020). Hiding shares of counting-based secret sharing via Arabic text steganography for personal usage. *Arabian Journal for Science and Engineering*, *45*(4), 2433–2458. doi:10.1007/s13369-019-04010-6

Gutub, A., & AlKhodaidi, T. (2020). Smart expansion of target key for more handlers to access multimedia counting-based secret sharing. *Multimedia Tools and Applications*, *79*(25-26), 17373–17401. doi:10.1007/s11042-020-08695-y

Gutub, A. A.-A., & Alaseri, K. A. (2021). Refining Arabic text stego-techniques for shares memorization of counting-based secret sharing. *Journal of King Saud University-Computer and Information Sciences*, *33*(9), 1108–1120. doi:10.1016/j.jksuci.2019.06.014

Hanke, T., Movahedi, M., & Williams, D. (2018). *DFINITY Technology Overview Series*. Consensus System.

He, J., & Dawson, E. (1994). Multistage secret sharing based on one-way function. *Electronics Letters*, *30*(19), 1591–1592. doi:10.1049/el:19941076

Herzberg, A. (1997). Proactive Public Key and Signature Systems. Paper presented at *the Proc ACM Ccs96*. ACM. doi:10.1145/266420.266442

Herzberg, A., Jarecki, S., Krawczyk, H., & Yung, M. (1995). *Proactive Secret Sharing Or: How to Cope With Perpetual Leakage.* Paper presented at the Advances in Cryptology-crypto 95, International Cryptology Conference, Santa Barbara, California, USA. doi:10.1007/3-540-44750-4_27

Kate, A., Goldberg, I., & Cheriton, D. R. (2009). *Asynchronous Distributed Private-Key Generators for Identity-Based Cryptography.* Paper presented at the Security and Cryptography for Networks.

Kaya, K., & Selçuk, A. A. (2008). *A verifiable secret sharing scheme based on the chinese remainder theorem.* Paper presented at the Progress in Cryptology-INDOCRYPT 2008: 9th International Conference on Cryptology in India, Kharagpur, India. doi:10.1007/978-3-540-89754-5_32

Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). *Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol.* Paper presented at the *Annual International Cryptology Conference*. Springer. doi:10.1007/978-3-319-63688-7_12

Kokoris-Kogias, E., Alp, E. C., Siby, S. D., Gailly, N., Gasser, L., Jovanovic, P., & Ford, B. (2018). Verifiable management of private data under byzantine failures. *IACR ePrint, 209:2018.*

Krawczyk, H. (1994). *Secret sharing made short.* Paper presented at the Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA.

Lin, C., & Harn, L. (2012). Unconditionally secure multi-secret sharing scheme. Paper presented at the *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*. IEEE. doi:10.1109/CSAE.2012.6272572

Liu, J., Rong, S., Kou, W., & Wang, X. (2007). Efficient ID-based Signature Without Trusted PKG. *Cryptology ePrint Archive*.

Mwitende, G., Ye, Y., Ali, I., & Li, F. (2020). Certificateless Authenticated Key Agreement for Blockchain-Based WBANs. *Journal of Systems Architecture*, *110*(4), 101777. doi:10.1016/j.sysarc.2020.101777

Ogata, W., Kurosawa, K., & Stinson, D. R. (2006). Optimum secret sharing scheme secure against cheating. *SIAM Journal on Discrete Mathematics*, *20*(1), 79–95. doi:10.1137/S0895480100378689

Ostrovsky, R., & Yung, M. (1991). *How To Withstand Mobile Virus Attacks*. UCLA.

Pedersen, T. P. (1991). *A threshold cryptosystem without a trusted party.* Paper presented at the Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK. doi:10.1007/3-540-46416-6_47

Qi, B., Ma, J., & Lv, K. (2020). Computing Interval Discrete Logarithm Problem with Restricted Jump Method. *Fundamenta Informaticae*, *177*(2), 189–201. doi:10.3233/FI-2020-1986

Qi, B., Ma, J., & Lv, K. (2020). Improved algorithm for solving discrete logarithm problem by expanding factor. *China Communications*, *17*(4), 31–41. doi:10.23919/JCC.2020.04.004

Ran, C., & Herzberg, A. (1994). *Maintaining Security in the Presence of Transient Faults*. Springer-Verlag.

Schultz, D., Liskov, B., & Liskov, M. (2010). MPSS: Mobile Proactive Secret Sharing. [TISSEC]. *ACM Transactions on Information and System Security*, *13*(4), 1–32. doi:10.1145/1880022.1880028

Shamir, A. (1979). *How to share a secret.* Paper presented at the Communications of the ACM. doi:10.1145/359168.359176

Sohrabi, N., Yi, X., Tari, Z., & Khalil, I. (2020). BACC: blockchain-based access control for cloud data. Paper presented at the *Proceedings of the Australasian Computer Science Week Multiconference*. ACM. doi:10.1145/3373017.3373027

Tassa, T. (2007). Hierarchical threshold secret sharing. *Journal of Cryptology*, *20*(2), 237–264. doi:10.1007/s00145-006-0334-8

(1581-1584). W., B., & L., J. (2003). (t,n)-Threshold Signature Scheme without Trusted Third Party. *Chinese Journal of Computers*, *26*(11).

Yuan, J., & Li, L. (2019). A fully dynamic secret sharing scheme. *Information Sciences*, *496*, 42–52. doi:10.1016/j.ins.2019.04.061

APPENDIX

The appendix explains the symbols and terms in the order in which they appear in this paper shown in Table 2.

Table 2.
The description of symbols using in this paper

Section	Symbols	Description
Secret Sharing	$\mathcal{M}$	message space
	Sh	secret sharing algorithm
	Re	reconstruction secret algorithm
	$\mathcal{S}$	participants set
	$\mathcal{A}$	adversary
	$negl$	negligible function
	λ	security parameter
Ouroboros in Our Scheme	u_i	a random number chosen by the i-th participant
	n_{total}	the number of the participants
	c_i	commitment of u_i
	Com	commitment algorithm
	r_i	random seed using in commitment algorithm
	$\left(\sigma_1^i, \ldots, \sigma_{n_{total}}^i\right)$	share pieces of u_i
	$\left(ct_1^i, \ldots, ct_{n_{total}}^i\right)$	share pieces of u_i after encryption
Our RiddleEncryption Scheme	$Setup$	algorithm generates $\left(answer, pp\right)$
	Enc	encryption algorithm for message with $clue$
	$GuessingRiddle$	algorithm to decide the one can decrypt cipher text
	Dec	decryption algorithm for cipher text with $answer$
	ω	desired puzzle difficulty
	t	given time to solve the puzzle
	pp	public parameters consisting $clue$

continued on following page

Table 2.
Continued

Section	Symbols	Description
Our RiddleEncryption Scheme	π	NIZK proof for the cipher text and $clue$
	$\mathcal{A}$	adversary algorithms
	n_{Adv}	the number of adversary
	n_{total}	the number of all participants in the system
	ϵ	constant number
Our Ccompetitive-ECPSS Scheme	k_0	the number of participants in 0-th rounds
	σ_{i_0}	secret share of i_0 -th participant
	th	the threshold of reconstruction
	F_0	polynomial which $F_0\left(0\right)=\sigma$
	σ	global secret
	$\mathrm{t}_{\mathrm{update}}$	time to update the global secret
	k_{round}	the number of participants in the round -th round
	r_{dealer}	random number chosen by dealer
	$P_1^{round}, P_2^{round}, \ldots, P_{k_{round}}^{round}$	participant 1, participant 2,…,participant k_{round}, in the round -th round
	Z_p^*	positive integer group which generator is p
	$b_1, b_2, \ldots, b_{k_{round}}$	the token which can verify the NIZK proofs
	σ_i^{round}	secret share of i-th participant in the round -th round
	$k_{round+1}$	the number of participants in the $\mathrm{round}+1$ -th round
	$»_{\mathrm{round}+1}$	security parameter for Ouroboros protocol in the $\mathrm{round}+1$ -th round
	k_1	the number of participants in 1-th rounds
	$\sigma_{i,1}^{round}$, $\sigma_{i,2}^{round}, \ldots, \sigma_{i,k_1}^{round}$	secret share of $\tilde{\mathrm{A}}_1^{\mathrm{round}}$
	s_n	one of the RE systems parameters

continued on following page

Table 2.
Continued

Section	Symbols	Description
Analysis for Our Ccompetitive-ECPSS Scheme	F_r	the polynomial which shares the global secret A in the $\mathbf{r}$ -th round
	»$_i$	Lagrange coefficients for F_r
	G_i	the polynomial which is the $\mathbf{r}$ -th round holding committee members choose for the $\mathbf{r}+1$ -th round committee members
	¼$_{i,j}$	the Lagrange coefficients for $\mathbf{G}_i$

Chuyi Yan received the bachelor's degree from Beijing Forestry University in 2019. She is currently a Ph.D. student at the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include blockchain security, intrusion detection, cyber situation awareness, etc.

Haixia Xu is a professor in SKLOIS, Institute of Information Engineering, CAS. She writes and presents widely on issues of information security, cryptography, security protocol and blockchain.