 CMC, vol.56, no.3, pp.401-414, 2018

Server-Aided Multi-Secret Sharing Scheme for Weak Computational Devices

En Zhang[1, 2], Xintao Duan[1, 2], Siuming Yiu[3], Junbin Fang[4], Zoe L. Jiang[5, *], Tsz HonYuen[6] and Jie Peng[1]

Abstract: In the setting of (t, n) threshold secret sharing, at least t parties can reconstruct the secret, and fewer than t parties learn nothing about the secret. However, to achieve fairness, the existing secret sharing schemes either assume a trusted party exists or require running multi-round, which is not practical in a real application. In addition, the cost of verification grows dramatically with the number of participants and the communication complexity is $O(t)$, if there is not a trusted combiner in the reconstruction phase. In this work, we propose a fair server-aided multi-secret sharing scheme for weak computational devices. The malicious behavior of clients or server providers in the scheme can be verified, and the server provider learns nothing about the secret shadows and the secrets. Unlike other secret sharing schemes, our scheme does not require interaction among users and can work in asynchronous mode, which is suitable for mobile networks or cloud computing environments since weak computational mobile devices are not always online. Moreover, in the scheme, the secret shadow is reusable, and expensive computation such as reconstruction computation and homomorphic verification computation can be outsourced to the server provider, and the users only require a small amount of computation.

Keywords: Secret sharing, server-aided, non-interactive, fairness.

1 Introduction

Secret sharing schemes are important building blocks in modern cryptography and have wide applications in access control, electronic voting, attribute-based encryption, secure multiparty computation, etc. In 1979, Shamir [Shamir (1979)] and Blakely [Blakely (1979)] independently proposed the (t, n) threshold secret sharing scheme. A (t, n) secret sharing scheme involves a dealer and a set of participants. It allows the dealer to

[1] College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China.

[2] Lab of Intelligence Business & Internet of Things of Henan Province, Xinxiang, 453007, China.

[3] Department of Computer Science, University of Hong Kong, Hong Kong, China.

[4] Department of Optoelectronic Engineering, Jinan University, Guangzhou, 510632, China.

[5] School of Computer Science and Technology, Harbin nstitute of Technology Shenzhen Graduate School, Shenzhen, 518055, China.

[6] Huawei, Singapore.

[*] Corresponding Author: Zoe L. Jiang. Email: zoeljiang@hit.edu.cn.

distribute shares of a secret among participants. Any group of *t* or more participants can recover the secret, however, any group of size less than *t* cannot obtain any information about the secret. Later, a series of secret sharing schemes were proposed.

The work of Chor et al. [Chor, Goldwasser, Micali et al. (1985)] first proposed the verifiable secret sharing (VSS) scheme to achieve security against cheaters. The works of [Feldman (1987); Pedersen (1991)] respectively proposed a VSS scheme based on homomorphic commitments. Recently, Cafaro et al. [Cafaro and Pelle (2015)] introduced a new computational problem (i.e. the Exponentiating Polynomial Root Problem) and proposed a space-efficient VSS scheme. Mohamed [Mohamed (2017)] proposed hierarchical threshold secret sharing scheme for color images. Mashhadi [Mashhadi (2017)] proposed a secure publicly verifiable and proactive secret sharing scheme with a general access structure. Miao et al. [Miao, Yan, Wang et al. (2015)] introduced the notion of randomized component (RC) and proposed a (*t*, *m*, *n*)-group oriented secret sharing. In their scheme, once *m* ($m \geq t$) players form a tightly coupled group by generating RCs, the secret can be constructed only if all *m* RCs are correct. Cramer et al. [Cramer, Damgård, Döttling et al. (2015)] presented a novel method for constructing secret sharing schemes from linear error correcting codes and linear universal hash functions. The work of Komargodski et al. [Komargodski and Zhandry (2016)] constructed two very natural extensions of secret sharing. The first one is called distributed secret sharing, and the second one is called functional secret sharing. Liu et al. [Liu, Ma, Wei et al. (2016)] proposed a multi-group dynamic quantum secret sharing scheme.

To share multiple secrets, a series of works [Pang and Wang (2005); Zhang and Cai (2013); Mashhadi and Dehkordi (2015); Shivani (2017); Deng, Wen and Shi (2017); Liu, Zhang and Zhang (2016); Pilaram and Eghlidos (2017)] have explored protocols for a multi-secret sharing scheme. Pang et al. [Pang and Wang (2005)] proposed a new multi-secret sharing scheme based on Shamir's secret sharing. Zhang et al. [Zhang and Cai (2013)] proposed a rational multi-secret sharing scheme in standard point-to-point communication networks. Mashhadi et al. [Mashhadi and Dehkordi (2015)] proposed two verifiable multi-secret schemes based on a linear feedback shift register public key and new nonhomogeneous linear recursions. Recently, Shivani [Shivani (2017)] proposed a multi secret sharing scheme with unexpanded meaningful shares. Deng et al. [Deng, Wen and Shi (2017)] proposed a threshold multi-secret sharing scheme based on phase-shifting interferometry. Liu et al. [Liu, Zhang and Zhang (2016)] analyzed the security of several verifiable multi-secret sharing schemes and showed that these schemes are vulnerable to attack. Next, they presented two improved verifiable multi-secret sharing schemes. Pilaram et al. [Pilaram and Eghlidos (2017)] proposed an efficient lattice based multistage secret sharing scheme which can resist against quantum computers.

In the setting of secret sharing, one desirable property is fairness, which guarantees that if one party learns the secret, the other parties do too. To achieve fairness, Tompa et al. [Tompa and Woll (1989)] proposed a multi-round secret sharing scheme in which the real secret *D* is encoded as a sequence $\{D^1,\ldots,D^k\}$ where $D^i = D$ for i chosen randomly and D^j=*dum* for all $j \neq i$, where dum is a dummy legal value. In the *jth* round for $1 \leq j \leq k$, all active parties pool their shares and reconstruct D^j of the sequence until some $D^j \neq dum$. In their scheme, all the active parties require simultaneously pooling of their

shares, and the cheater has a probability $1/k$ of learning the secret while the others do not. In 2013, Tian et al. [Tian, Ma, Peng et al. (2013)] proposed a fair threshold secret sharing scheme following the approach of Gordon et al. [Gordon, Hazay, Katz et al. (2011)]. In their works, the dealer distributes a list of length v, and the true share is hidden in the list, and each party learns the true share with probability $1/v$ .The reconstruction protocol requires at most v iterations. Recently, the work of Harn et al. [Harn, Lin and Li (2015)] proposed an asynchronous secret reconstruction scheme to protect the secret against both inside and outside adversaries. Another line of works [Halpern and Teague (2004); Liu, Li and Ma (2017); Zhang and Liu (2013); Tian, Peng, Lin et al. (2015); Zhang, Yuan and Du (2015); Jin, Zhou and Ma (2017)] focused on designing rational secret sharing schemes. In this setting, players are rational rather than corrupt or honest. Halpern et al. [Halpern and Teague (2004)] first introduced rational secret sharing and multiparty computation. They pointed out that any method for sharing secret reconstruction with a commonly known upper bound on the running time is unstable, and no parties will send a shadow in the last round since they have no incentive to do so. Following the seminal work of Halpern et al. [Halpern and Teague (2004)], all existing rational secret sharing schemes require running multi rounds.

Unfortunately, all the above schemes are not fully satisfied to guarantee fairness. It is not hard to see that no party has any incentive to broadcast his/her shadow in a traditional single round secret sharing scheme since not sending his/her shadow weakly dominates sending his/her shadow. For example, assume that t active parties cooperate to reconstruct a secret that was shared using a t-out-of-n secret sharing scheme. If each party simultaneously broadcasts his/her share to all others, every party can learn the secret. However, if one party does not broadcast his/her share, it can still reconstruct the secret (because it received the $t-1$ shares of all other parties), but the others cannot (because they only have $t-1$ shares). In contrast, all existing fair secret sharing schemes and rational secret sharing schemes require running multi rounds, which are rather inefficient, in particular in the case of weak computational devices such as smart phones, PDAs or tablets.

1.1 Motivation

With the rapid development of information technologies and smart terminals, cloud computing has become a vital part of daily activity. Using smart phones, users can easily upload or share their personal data to others for diverse purposes or download kinds of applications from the Apple store or Google Play store, such as Google drive, Dropbox, Baidu cloud and Wechat. For instance, patients can upload their own e-health records to the cloud and users can download applications from the Apple store or Google Play store. A server-aided secret sharing model is described in Fig. 1. Although cloud outsourcing computation's benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption.

Recently, several server-aided MPC protocols have been considered to trade off the parties' work at the expense of the servers [Peter, Tews and Katzenbeisser (2013); López-Alt, Tromer and Vaikuntanathan (2012); Zhang, Li, Niu et al. (2017); Gordon, Katz and Liu (2015)]. Unfortunately, all existing server-aided MPC protocols are still far from practical for the secret sharing scheme. Moreover, to achieve fairness, the existing secret

sharing schemes either assume a trusted party exists or require running multi-round. In addition, the cost of verification grows dramatically with the number of participants and the communication complexity is $O(t)$, if there is not a trusted combiner in the reconstruction phase.

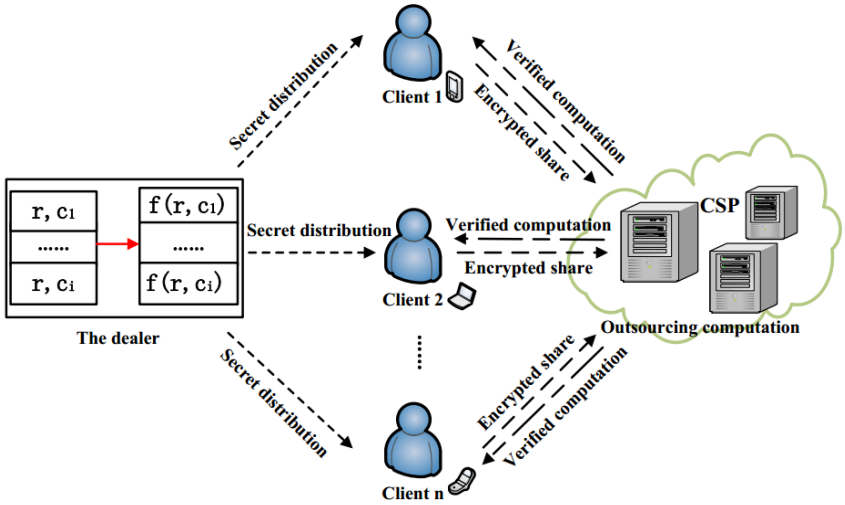


Figure 1: Outsourcing secret sharing system model

1.2 Contributions

In this work, we propose an outsourcing multi-secret sharing scheme, as shown in Fig. 1. Given the recent emergence of public clouds, such as Amazon EC2 and Microsoft Azure, clients are able to delegate expensive computation to a powerful cloud server, and in the scheme, it is fair for every client to obtain the multi secrets with limited computation cost. Our approach is particularly suited to applications where clients are very resource-constrained and not always online. To the best of our knowledge, this is the first server-aided secret sharing scheme. Compared with previous secret sharing schemes, the proposed scheme has a number of advantages as follows.

(a) The scheme is suitable for reusable multi-stage multi-secret sharing and expensive computation can be outsourced to an untrusted cloud server.

With the amount of data generated, collected, and analyzed by computing systems growing at an amazing rate, reusable multi-stage multi-secret sharing schemes have attracted much attention from researchers. In our scheme, expensive computation such as secret reconstruction and homomorphic encryption can be performed by the cloud service provider. The m secrets that have been reconstructed cannot reveal any information about the next m secrets that have not been reconstructed. Moreover, the cloud service provider (CSP) learns nothing about the secret shadow and the secret.

(b) It is fair for every active client to obtain the multi secrets.

To achieve fairness, all existing fair or rational cryptographic protocols require running many rounds, which is not efficient in a real application. In this work, we propose a server-aided secret sharing scheme. Following Gordon et al. [Gordon, Katz and Liu

(2015)], we assume that the CSP cannot collude with the parties, this is due to a connection we establish between our scheme and virtual black-box (VBB) obfuscation, which is already known to be impossible [Barak, Goldreich, Impagliazzo et al. (2001)]. In addition, there are many settings where collusion does not occur; for example, given the consequences of legal action and bad publicity, it is reasonable to assume that the large CSP (e.g. Google, Amazon or Microsoft) will not collude with the parties. Finally, it is fair for any t active clients to obtain multiple secrets, which means if one client obtains the secrets, all the other $t-1$ clients do too.

(c) The scheme does not require user interaction or always being online.

There are many settings in practice where users usually access their profiles via resource constrained mobile devices that are not always online, and a completely non-interactive solution is crucial for such systems to work in practice. Our scheme can reconstruct the secret without any interaction among the clients. Once online, clients can retrieve the secret, while the server learns nothing at all.

(d) The scheme is verifiable, and the verifiable stage can be done by the weak computational devices in the idle time.

Generally, the cloud service provider may have a strong financial incentive to try to cheat and return a wrong result to the client. General non-interactive verifiable computation can be used to address this problem. Unfortunately, existing solutions are either inefficient or rely heavily on user interaction. In this work, we use a one-way hash function to add verification capabilities. Given the secret s, the dealer computes $k = h(s)$ and makes it public, and after reconstructing a secret s', the client verifies whether $h(s') = k$. If $h(s') \neq k$, the CSP is cheating. In addition, the verifiable stage can be performed by the weak computational devices during idle time.

1.3 Overview

The remainder of this paper is organized as follows. In Section 2, the preliminary of cryptography and secret sharing are introduced. Section 3 introduces the server-aided secret scheme. In Section 4, we analyze the new scheme, and we compare our solution with the existing secret sharing schemes in Section 5. Finally, we present our conclusions in Section 6.

2 Preliminaries

2.1 Two-variable one-way function

Definition 2.1 Let k be a security parameter, and let $f[x]$ denote the set of values $f(x, y)$ for all possible y. A function $f(x, y)$ is called a two-variable one-way function if it is easily computable, but for any $y_i \in \{0,1\}^k$ where $1 \leq i \leq l$, for any polynomial time algorithm A, for all polynomials p and all sufficiently large k, the probability that A on inputs $f(x, y_i)$ and outputs a y such that $y \neq y_i$ is held that

$$Pr[f(x, A(f(x, y_1), f(x, y_2), \ldots f(x, y_l))) \in f[x] \| \ x \in_R \{0,1\}^k, y \neq y_i, 1 \leq i \leq l] < 1/\ p(k) \quad (1)$$

Remark: The two-variable one-way function has several properties as follows.

(a) Given x and y, it is easy to evaluate $f(x, y)$; (b) Without the knowledge y, it is hard to calculate $f(x, y)$ given any x; (c) Given x and $f(x, y)$, it is hard to compute y; (d) Given y and $f(x, y)$, it is hard to evaluate x; (e) Given y, it is hard to find different values x and x' such that $f(x, y) = f(x', y)$; (f) When pairs of x and $f(x, y)$ are given, it is hard to compute $f(x', y)$ where $x \neq x'$.

2.2 Secret sharing scheme

Definition 2.2 Let F_q be a finite field. A t-out-of-n secret sharing scheme is a pair of algorithms $Share(\cdot)$ and $Rec(\cdot)$. For every secret $s \in F_q$, $Share(\cdot)$ outputs a vector $c_1, \ldots, c_n$. For any t-subset $[i_1, \ldots, i_t] \subseteq [n]$, it holds that

$$Rec((i_1, c_{i_1}), \ldots, (i_t, c_{i_t})) = s \quad (2)$$

Definition 2.3 Let $P = \{P_1, \ldots, P_n\}$ be a set of participants. A collection $\mathcal{A} \subseteq 2^P$ is monotone if $B \in \mathcal{A}$ and $B \subseteq B' \subseteq P$ imply that $B' \in \mathcal{A}$. Sets in $\mathcal{A}$ are called qualified subsets, and sets in $2^P \setminus \mathcal{A}$ are called unqualified subsets.

Definition 2.4 Let $P = \{P_1, \ldots, P_n\}$ be a set of participants, S_i be the space from which the i^{th} secret s_i can be selected, $\{S_A\}_{A \subseteq P}$ be the family of A-secrets-sets, and $H(X)$ be the entropy of X. A multi-secret sharing scheme for $\{S_A\}_{A \subseteq P}$ is a sharing of the secrets in S among participants in P that satisfies the following requirements:

(a) For all subsets $A \subseteq P$, it holds $H(S_A | A) = 0$.

(b) For all subsets $A \subseteq P$ and $T \subseteq \{S_1, \ldots, S_m\} \setminus S_A$, it holds $H(T | A) = H(T)$.

Definition 2.5 A verification secret sharing scheme satisfies the following requirements:

(a) If the dealer and participant P_i both follow the scheme, then P_i accepts the share with probability 1.

(b) For all subsets T_1 and T_2 of $\{1, \ldots, n\}$ of size t, if all participants $(P_i)_{i \in T_1}$ and $(P_i)_{i \in T_2}$ have accepted their shares, then the following holds except with negligible probability: If s_i is the secret constructed by the parties in T_i (for i = 1,2), then $s_1 = s_2$.

3 The server-aided multi-secret sharing scheme (SMSSS)

In this section, with some modifications, Pang's MSS approach can be used in our server-aided multi-secret sharing scheme.

3.1 System parameters

Let $P = P_1,\ldots,P_n$ be n participants and let $f(r,c)$ be a two-variable one-way function. Suppose that h is a collision resistant hash function and that p is a safe prime, such as $q|(p-1)$, where q is a big prime. We use α to denote the generator of order q over Z_p^* and use $s_1,\ldots,s_m$ to denote m secrets shared among n participants. The dealer selects ρ_i from $[m,q-1]$ as P_i's public identity information for $1 \le i \le n$ and creates a public notice board that can be accessed by the participants and the cloud service provider. However, the public information on the board can only be updated by the dealer.

3.2 Protocol for sharing phase

The dealer executes the following steps:

Step 1: The dealer randomly chooses n distinct integers c_i and an integer ξ and then sends it to P_i by a secret channel.

Step 2: Randomly choose an integer r and compute $f(r,c_i)$ for $1 \le i \le n$.

Step 3: With the knowledge of $(n+m)$ pairs of $(0,\xi \oplus s_1),(1,\xi \oplus s_2),\ldots,(m-1,\xi \oplus s_m)$ and $(\rho_i, f(r,c_i))$ for $1 \le i \le n$, the dealer constructs a $(n+m-1)$th degree polynomial as follows.

$$W(x) = a_0 + a_1 x + \ldots + a_{n+m-1}x^{n+m-1} \bmod q \tag{3}$$

Step 4: The dealer generates the public verification information: $\alpha_k = \alpha^{a_k} mod\ p$ for $0 \le k \le n+m-1$, chooses the $(n+m-t)$ minimum integers $\sigma_1,\sigma_2,\ldots,\sigma_{n+m-t}$ from the set $\{[m,q-1]-\rho_j\}$ for $1 \le j \le n$, and computes $W(\sigma_i)$ for $1 \le i \le n+m-t$.

Step 5: The dealer publishes $(r,W(\sigma_i),\alpha_k,h(s_j))$ on the notice board, where $1 \le i \le n+m-t$, $0 \le k \le n+m-1$ and $1 \le j \le m$.

3.3 Protocol for outsourcing phase

Let P_u be the set of the t active participants for $u = 1',2',\ldots,t'$.

Step 1: P_u encrypts his/her pseudo shadow $f(r,c_u)$ using the cloud service provider (denoted by CSP)'s public key PK and sends the ciphertext $E_{PK}(f(r,c_u))$ to CSP.

Step 2: CSP decrypts the ciphertext and checks the following equation to verify whether P_u's secret shadow is valid:

$$\alpha^{f(r,c_u)} = \prod_{j=1}^{n+m-1} \alpha_j^{(\rho_u)^{n+m-1}} mod\ p. \tag{4}$$

If the verification fails, CSP informs all the t active participants and aborts the protocol, else the protocol continues.

Step 3: With the knowledge of t pairs of $(\rho_u, f(r,c_u))$, $u=1',2',\ldots,t'$ and $n+m-t$ pairs of $(\sigma_v, W(\sigma_v))$, $1\le v\le n+m-t$. The $(n+m-1)^{\text{th}}$ degree polynomial $W(x)$ can be uniquely determined as

$$\begin{aligned} W(x) &= \sum_{u=1'}^{t'} f(r,c_u) \prod_{j=1',j\neq u}^{t'} \frac{x-\rho_j}{\rho_u-\rho_j} \prod_{v=1}^{n+m-t} \frac{x-\sigma_v}{\rho_u-\sigma_v} \\ &+ \sum_{v=1}^{n+m-t} W(\sigma_v) \prod_{k=1,k\neq v}^{n+m-t} \frac{x-\sigma_k}{\sigma_v-\sigma_k} \prod_{u=1'}^{t'} \frac{x-\rho_u}{\sigma_v-\rho_u} \\ &= a_0 + a_1 x + \ldots + a_{n+m-1} x^{n+m-1} mod\ q \end{aligned} \tag{5}$$

3.4 Protocol for reconstruction phase

Step 1: CSP sends $W(\chi)$ for $0\le\chi\le m-1$ to P_u.

Step 2: P_{a_i} verifies whether $h(W(i)\oplus\xi)=h(s_{i+1})$, for $(i=0,\ldots,m-1)$. If the check succeeds, the m secrets can be obtained as $s_1=W(0)\oplus\xi$, $s_2=W(1)\oplus\xi,\ldots,s_m=W(m-1)\oplus\xi$, else the CSP has deviated the protocol and P_{a_i} aborts the protocol.

4 Scheme analysis

In this section, we analyze the security and performance of the scheme.

Theorem 4.1 The SMSSS is secure against malicious adversaries and any *t*-1 or fewer clients' collusion cannot reconstruct the secrets.

Proof. The security of our outsourcing secret sharing scheme can be analyzed as follows.

(a) The malicious behaviors of clients can be verified by checking Eq. (4).

Due to the homomorphic property of exponentiation, a commitment of a share $f(r, c_u)$ can be written as:

$$\begin{aligned} \alpha^{f(r,u)} = \alpha^{W(\rho_u)} &= \alpha^{a_0+a_1\rho_u+\ldots+a_{n+m-1}(\rho_u)^{n+m-1}} \\ &= \alpha^{a_0}\alpha^{a_1\rho_u}\ldots\alpha^{a_{n+m-1}(\rho_u)^{n+m-1}} \\ &= \alpha_0\alpha_1^{\rho_u}\ldots\alpha_{n+m-1}^{(\rho_u)^{n+m-1}} \end{aligned} \tag{6}$$

Therefore, the cheating behaviors of clients can be detected by the CSP.

(b) Any *t*-1 or fewer clients' collusion cannot learn the secrets.

In the scheme, any *t*-1 or fewer clients' cooperation cannot determine the polynomial $W(x)$, so they cannot learn the secrets. However, any *t* or more clients' cooperation can determine the polynomial $W(x)$ and then learn the secrets by using $s_i=W(i)\oplus\xi$ $(i=0,\ldots,m-1)$.

(c) The CSP learns nothing about the secrets.

The scheme provides input and output privacy for the client, and this is a critical feature for many real-life scenarios. Every active client only submits a pseudo shadow $f(r, c_u)$ in

the secret reconstruction stage, and the real shadow is protected by the properties of the two-variable one-way function. Therefore, the CSP does not learn any information about the input and output of the client. In the scheme, following the seminal work of Gordon et al. [Gordon, Katz and Liu (2015)], we assume that the CSP does not collude with the clients, since Gordon et al. [Gordon, Katz and Liu (2015)] pointed out that simulation-secure multi-client verifiable computation is impossible to realize when the server colludes with clients. Actually, in real life, it is reasonable to assume that the large CSP (e.g. Google, Amazon or Microsoft) will not collude with the parties.

(d) The client can verify the output provided by the CSP.

Because clients lose control on their data and do not have access to the cloud's internal operational details, the CSP may have a strong financial incentive to try to cheat and return a wrong result to the client without detection. Therefore, the client needs some guarantees that the answer returned from the server is correct. General non-interactive verifiable computation can be used to ensure that the CSP performs the computation correctly. Unfortunately, existing solutions are either inefficient or rely heavily on user interaction. In contrast, we use a one-way hash function to add verification capabilities. Given the secrets, the dealer computes $k=h(\mathrm{s})$ and makes it public, and after reconstructing a secret s', the client verifies whether $h(\mathrm{s}')=k$. If $h(s')\neq k$, the CSP is cheating. The proposed verifiable method is more efficient and practical than previous non-interactive verifiable computation.

Theorem 4.2 The SMSSS is a multi-secret sharing scheme, and each client can use his/her secret shadow many times.

Proof. During the protocol for the sharing phase, the dealer uses the pseudo shadow $f(r,c_i)$ instead of the real shadow c_i, and the m secrets $(s_1,\ldots,s_m)$ are hidden by $\xi\oplus s_j$ for $1\le j\le m$. With the knowledge of $(n+m)$ pairs of $(0,\xi\oplus s_1),(1,\xi\oplus s_2),\ldots,(m-1,\xi\oplus s_m)$ and $(\rho_i,f(r,c_i))$ for $1\le i\le n$, the $(n+m-1)^{\text{th}}$ degree polynomial $W(x)$ can be uniquely determined. Each active client sends his/her pseudo shadow to the CSP in the reconstruction phase. Given $f(r,c_u)$, it is hard to compute c_u for $u=1',\ldots,t'$ by the properties of the two-variable one-way function. With the knowledge of t pairs of $(\rho_u,f(r,c_u))$ and $n+m-t$ pairs of $(\sigma_v,W(\sigma_v))$, the CSP can construct the $(n+m-1)^{\text{th}}$ degree polynomial $W(x)$ and learn $W(\chi)$ for $0\le\chi\le m-1$. However, the CSP learns nothing about the m secrets since m secret are hidden by $\xi\oplus s_j$ for $1\le j\le m$. In addition, each client can use his/her shadow many times. For example, to reconstruct the next m secrets, the dealer randomly chooses a new r, which is not equal to the last one and computes $f(r,c_i)$ for $1\le i\le n$. Then, the dealer runs Step 3-Step 5 of the sharing phase without redistributing the secret shadow. In contrast, each active client P_u calculates pseudo shadow $f(r,c_u)$ and sends it to the CSP, and finally, the other m secrets can be constructed.

Theorem 4.3 The SMSSS is complete fairness for every client and can work in an asynchronous network.

Proof. There are two major kinds of fair secret sharing: One is the protocols relying on a Trusted Third Party (TTP) to ensure the fairness of the protocol, and the other is the protocols without TTP. However, in real life, it is very hard to find a TTP in a distributed network. Therefore, we turn to consider the second case: Most previous protocols assumed that the parties had access to a simultaneous channel (meaning that all parties can simultaneously send messages and no party can see what the others broadcast before sending its own), which is problematic to implement in practice.

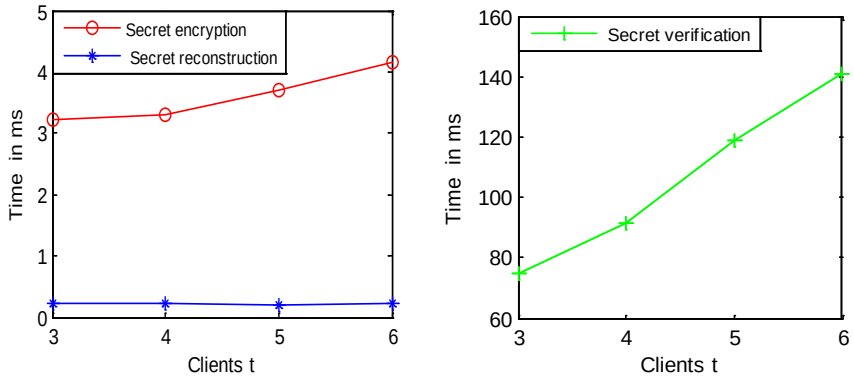
Throughout this paper, we assume that all information is communicated asynchronously. However, in an asynchronous secret reconstruction, when all other parties honestly release their shares, a dishonest party can always exclusively recover the secret by presenting a fake share last. Thus, the other honest parties get nothing but a fake secret. In our scheme, the above problem is addressed by using an untrusted service provider that cannot collude with participants. Given the consequences of legal action and bad publicity, it is reasonable to assume that the large CSP (e.g. Google, Amazon or Microsoft) will not collude with the parties. In our scheme, all the active t clients send pseudo shadow $f(r,c_u)$ asynchronously to the CSP, and no client can see what the others send before sending its own. Once one client sends a fake pseudo shadow, the cheating behavior will be detected by the CSP, and the cheating client learns nothing about the secret. Finally, in the reconstruction phase, every active client can obtain the m secrets.

5 Performance evaluation

We performed the test at Python 3.5.4rc1, and we ran all clients and the server on the same host, so that network latency was able to be ignored. A full overview of the computation time of each protocol step is given in Tab. 1. The data in Fig. 2(a) (marked by the blue line) shows the runtime of the secret reconstruction. In addition, we computed how much time it takes to encrypt the secret, and the data in Fig. 2(a) (marked by the red line) shows that the runtime of the secret encryption grows linearly with the number of clients. In contrast, it is clear that the verification algorithm takes more time than secret encryption and reconstruction in Fig. 2(b) and the runtime of secret verification is 0.14 seconds when t=6.

Table 1: Computation time (in s) of each protocol step

Clients	t=3	t=4	t=5	t=6
Time of secret Encryption	0.00322	0.00331	0.00369	0.00415
Time of secret Verification	0.07458	0.09133	0.11869	0.14095
Time of secret Reconstruction	0.00023	0.00024	0.00021	0.00022



(a) The runtime of the secret encryption and secret reconstruction

(b) The runtime of the secret verification

Figure 2: The runtime of secret encryption, reconstruction and verification

Table 2: Comparison with the existing secret sharing schemes

Feature	Liu (2016)	Pang (2005)	Pilaram (2017)	Harn (2015)	Liu (2017)	Ours
Fairness	no	no	yes	yes	yes	yes
Interactive	yes	yes	no	yes	yes	no
Trusted party	no	no	yes	no	no	no
Communication Cost expensive	$O(t)$	$O(t)$	$O(1)$	$O(t)$	$O(t)$	$O(1)$
Computation	users	users	users	users	users	CSP

In addition, we compare our solution with the existing secret sharing schemes. Tab. 2 shows the main comparative information of our scheme and five schemes analyzed. Based on Shamir's secret sharing, Pang et al. [Pang and Wang (2005)] proposed a new (t, n) multi-secret sharing scheme. In the scheme, multi-secret can be shared among n participants, and t or more participants can reconstruct these secrets, but t-1 or fewer participants can derive nothing about these secrets. Liu et al. [Liu, Zhang and Zhang (2016)] analyzed the security of several proposed verifiable multi-secret sharing and proposed two improved schemes. These schemes can not only resist cheating by the dealer or participants, but also remove the use of private channels. Unfortunately, the above schemes cannot ensure fairness and the communication complex is $O(t)$. Pilaram et al. [Pilaram and Eghlidos (2017)] proposed an efficient lattice based multistage secret sharing scheme which can resist against quantum computers. However, the scheme requires a trusted combiner to recover the secret. To achieve fairness, the protocols of Harn et al. [Harn, Lin and Li (2015); Liu, Li and Ma (2017)] require running multi-round, which are not suitable for weak computational devices.

In contrast, our scheme only requires running single-round and the communication cost is $O(1)$. In the scheme, expensive computation such as secret reconstruction and homomorphic verification can be performed by the cloud servers, users only require a small amount of computation. Moreover, our scheme does not require interaction among users. Non-interaction is very desirable in the setting of mobile networks or cloud computation since weak computational mobile devices are not always online. The secrets can be reconstructed by the CSP, and once clients are online, they can retrieve the secret by themselves.

6 Conclusion

In this paper, we propose the first server-aided multi-secret sharing scheme. The protocol, which only requires running a single round, offers complete fairness for every active client. Expensive computation such as reconstruction computation and homomorphic verification computation can be outsourced to the server provider. In contrast, every client only requires a small amount of computation. Moreover, our scheme does not require interaction among users; any malicious behavior by clients and the CSP can be verified and can work in an asynchronous mode, which is very suitable for mobile networks or cloud computing environments.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (U1604156, 61602158, 61772176) and Science and Technology Research Project of Henan Province (172102210045).

References

Barak, B.; Goldreich, O.; Impagliazzo, R.; Rudich, S.; Sahai, A. et al. (2001): On the (im) possibility of obfuscating programs. *Lecture Notes in Computer Science*, vol. 2139, no. 2, pp. 1-18.

Blakely, G. (1979): Safeguarding cryptographic keys. *Proceedings of the National Computer Conference*, vol. 48, pp. 313-317.

Cafaro, M.; Pelle, P. (2015): Space-efficient verifiable secret sharing using polynomial interpolation. *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 1-12.

Chor, B.; Goldwasser, S.; Micali, S.; Awerbuch, B. (1985): Verifiable secret sharing and achieving simultaneity in the presence of faults. *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pp. 383-395.

Cramer, R.; Damgård, I. B.; Döttling, N.; Fehr. S.; Spini, G. (2015): Linear secret sharing schemes from error correcting codes and universal hash functions. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 313-336.

Deng, X. P.; Wen, W.; Shi, Z. G. (2017): Threshold multi-secret sharing scheme based on phase-shifting interferometry. *Optics Communications*, vol. 387, pp. 409-414.

Feldman, P. (1987): A practical scheme for non-interactive verifiable secret sharing. *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pp. 427-438.

Gordon, S. D.; Katz, J.; Liu, F. H.; (2015): Multi-client verifiable computation with stronger security guarantees. *Theory of Cryptography*, pp. 144-168.

Gordon, S. D.; Hazay, C.; Katz, J.; Lindell, Y. (2011): Complete fairness in secure two-party computation. *Journal of the ACM*, vol. 58, no. 6, pp. 24.

Halpern, J.; Teague, V. (2004): Rational secret sharing and multiparty computation. *Proceedings of the 36th annual ACM symposium on Theory of computing*, pp. 623-632.

Harn, L.; Lin, C. L.; Li, Y. (2015): Fair secret reconstruction in (*t*, *n*) secret sharing. *Journal of Information Security and Applications*, vol. 23, pp. 1-7.

Jin, J. H.; Zhou, X.; Ma, C. G. (2017): A new socio-rational secret sharing scheme paper withdrawn. *International Journal of Innovative Computing and Applications*, vol. 8, no. 1, pp. 21-30.

Komargodski, I.; Zhandry, M. (2016): Cutting-edge cryptography through the lens of secret sharing. *Theory of Cryptography Conference*, pp. 449-479.

Liu, H.; Li, X. H.; Ma, J. F. (2017): Reconstruction methodology for rational secret sharing based on mechanism design. *Science China Information Sciences*, vol. 60, no. 8.

Liu, H. W.; Ma, H. Q.; Wei, K. J.; Yang, X. Q.; Qu, W. X. et al. (2016): Multi-group dynamic quantum secret sharing with single photons. *Physics Letters A*, vol. 380, no. 31, pp. 2349-2353.

Liu, Y. H.; Zhang, F. T.; Zhang, J. (2016): Attacks to some verifiable multi-secret sharing schemes and two improved schemes. *Information Sciences*, vol. 329, pp. 524-539.

López-Alt, A.; Tromer, E.; Vaikuntanathan, V. (2012): On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pp. 1219-1234.

Mashhadi, S.; Dehkordi, M. H. (2015): Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem. *Information Sciences*, vol. 294, pp. 31-40.

Mashhadi, S. (2017): Secure publicly verifiable and proactive secret sharing schemes with general access structure. *Information Sciences*, vol. 378, pp. 99-108.

Miao, F. Y.; Yan, X.; Wang, X. F.; Badawy, M. (2015): Randomized component and its application to (*t*, *m*, *n*)-group oriented secret sharing. *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 889-899.

Mohamed, P. (2017): Hierarchical threshold secret sharing scheme for color images. *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5489-5503.

Pang, L. J.; Wang, Y. M. (2005): A new (*t*, *n*) multi-secret sharing scheme based on shamir's secret sharing. *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 840-848.

Pedersen, T. P. (1991): Distributed provers with applications to undeniable signatures. *Lecture Notes in Computer Science. Workshop on the Theory and Application of Cryptographic Techniques*, vol. 547, no.1, pp. 221-242.

Peter, A.; Tews, E.; Katzenbeisser, S. (2013): Efficiently outsourcing multiparty computation under multiple keys. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2046-2058.

Pilaram, H.; Eghlidos, T. (2017): An efficient lattice based multi-stage secret sharing scheme. *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 2-8.

Shamir, A. (1979): How to share a secret. *Communications of the ACM*, vol. 22, no. 11, pp. 612-613.

Shivani, S. (2017): Multi secret sharing with unexpanded meaningful shares. *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6287-6310.

Tian, Y. L.; Peng, C.; Lin, D.; Ma, J.; Jiang, Q. et al. (2015): Bayesian mechanism for rational secret sharing scheme. *Science China Information Sciences*, vol. 58, no. 5, pp. 1-13.

Tian, Y. L.; Ma, J. F.; Peng, C.; Jiang, Q. (2013): Fair (*t, n*) threshold secret sharing scheme. *IET Information Security*, vol. 7, no. 7, pp. 106-112.

Tompa, M.; Woll, H. (1989): How to share a secret with cheaters. *Journal of Cryptology*, vol. 1, no. 3, pp. 133-138.

Zhang, E.; Cai, Y. (2013): Rational multi-secret sharing scheme in standard point-to-point communication networks. *International Journal of Foundations of Computer Science*, vol. 24, no. 6, pp. 879-897.

Zhang, E.; Yuan, P. Y.; Du, J. (2015): Verifiable rational secret sharing scheme in mobile networks. *Mobile Information Systems*, vol. 2015, no. 40, pp. 1-7.

Zhang, E.; Li, F. H.; Niu, B.; Wang, Y. C. (2017): Server-aided private set intersection based on reputation. *Information Sciences*, vol. 387, pp. 180-194.

Zhang, Z. F.; Liu, M. L. (2013): Rational secret sharing as extensive games. *Science China Information Sciences*, vol. 56, no. 3, pp. 1-13.