PLOS ONE

RESEARCH ARTICLE

Novel Threshold Changeable Secret Sharing Schemes Based on Polynomial Interpolation

Lifeng Yuan[1,2], Mingchu Li[1,2], Cheng Guo[1,2]*, Kim-Kwang Raymond Choo[4,5], Yizhi Ren[3]*

1 School of Software Technology, Dalian University of Technology, Dalian, 116620, China, **2** Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, 116620, China, **3** School of Cyberspace, Hangzhou Dianzi University, Hangzhou, 310018, China, **4** Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, United States of America, **5** School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, 5095, Australia

* guocheng@dlut.edu.cn (CG); renyz@hdu.edu.cn (YZR)

Citation: Yuan L, Li M, Guo C, Choo K-KR, Ren Y (2016) Novel Threshold Changeable Secret Sharing Schemes Based on Polynomial Interpolation. PLoS ONE 11(10): e0165512. doi:10.1371/journal.pone.0165512

Editor: Houbing Song, West Virginia University, UNITED STATES

Received: June 1, 2016

Accepted: October 13, 2016

Published: October 28, 2016

Data Availability Statement: All relevant data are within the paper.

Funding: The author Cheng Guo received the National Science Foundation of China under grant No. 61501080 (website: http://www.nsfc.gov.cn/); the author Cheng Guo received the general program of Liaoning Provincial Department of Education Science Research under grants L2014017. The author Cheng Guo received the Fundamental Research Funds for the Central Universities under grant No. DUT16QY09. The author Mingchu Li received the National Science Foundation of China under grant No. 61572095

Abstract

After any distribution of secret sharing shadows in a threshold changeable secret sharing scheme, the threshold may need to be adjusted to deal with changes in the security policy and adversary structure. For example, when employees leave the organization, it is not realistic to expect departing employees to ensure the security of their secret shadows. Therefore, in 2012, Zhang et al. proposed ($t \rightarrow t'$, n) and ($\{t_1, t_2, \cdots, t_N\}$, n) threshold changeable secret sharing schemes. However, their schemes suffer from a number of limitations such as strict limit on the threshold values, large storage space requirement for secret shadows, and significant computation for constructing and recovering polynomials. To address these limitations, we propose two improved dealer-free threshold changeable secret sharing schemes. In our schemes, we construct polynomials to update secret shadows, and use two-variable one-way function to resist collusion attacks and secure the information stored by the combiner. We then demonstrate our schemes can adjust the threshold safely.

Introduction

Rapid advances in Internet technologies have resulted in significant changes in our society (e.g. digitalization of our society), but there are also associated security and privacy risks. In an open communication network, for example, data can be easily intercepted, modified, and even deleted by one or more attackers. It is, therefore, of little surprise that cyber security is a topic of current interest in different disciplines [1, 2]. For example, Javanmardi et al. [3] proposed a fuzzy reputation-based model for trust management in semantic P2P grids, and Li et al. [4] proposed a trust management scheme designed to resist malicious attacks and evaluate the trustworthiness of both data and mobile nodes in securing vehicular ad hoc networks. Butun et al. [5] proposed a cloud-centric, multi-level authentication as a service approach to address both scalability and time constraints for secure public safety device networks. Other research efforts include those reported in [6–9].

(website: http://www.nsfc.gov.cn/); these foundation supported the preparation of this manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Cryptography is an important tool used to ensure data security (e.g. confidentiality). However, the security level is generally determined by the security level of the stored secret key. In 1979, Shamir [10] and Blakley [11] independently proposed (t, n) threshold secret sharing (TSS) scheme designed to protect the secret by distributing a secret among a group of n participants. Only t or more participants in this group can cooperate to recover the secret. The (t, n) threshold secret sharing scheme has been used in various applications, such as in banks to protect the master key, and in certification authorities to protect the private root certificate keys. We refer interested reader to [12–14] for surveys of (t, n) threshold secret sharing schemes.

In practice, the threshold may have to be adjusted if there are changes in the security policies and adversary structures prior to recovering the secret. Examples of changes that require threshold to be adjusted include: (1) an increase or decrease in the importance level of the secret; (2) a change in participants number (i.e., one or more participants joining or leaving the group); (3) a change in the level of mutual trust between participants; and (4) the leakage of some participants' secret shadows. In 1989, Laih et al. [15] proposed the first threshold changeable secret sharing (TCSS) scheme to solve this problem. Since then, several other TCSS schemes based on different methods, such as polynomial interpolation [16–18], lattice basis reduction [19–21], and random noise [22], have been proposed in the literature.

In a naive implementation of the TCSS scheme, a dealer constructs new secret shadows for participants in the new access structure once the threshold is changed. Thus, the dealer needs to hold the secret online and the attacker only needs to defeat the dealer to obtain the secret. To avoid such an attack, Desmedt and Jajodia [23] used the secret shadow redistributing technique in their proposed TCSS scheme, which does not require the dealer's participation after the initialization phase. Similar schemes have also been presented in [24, 25]. In these schemes, each original secret shadow needs to be split into smaller shadows, which are redistributed to all participants in the new access structure. Each participant P_i combines all received smaller secret shadows into one new secret shadow s_i' using a suitable linear combination; thus, each participant only needs to store s_i'. Note that all participants are required to simultaneously maintain mutual secure communication channels. However, this may be impractical when the threshold changes, especially when the change is sudden.

To avoid the requirement of maintaining mutual secure communication channels, several TCSS schemes [16, 17, 26–28] based on broadcasting were proposed. In the schemes described in [17, 26], the dealer validates the new threshold by broadcasting a suitable number of her/his own redundant secret shadows. For example, in the scheme of [26], the dealer constructs a $(n + 1, 2n)$ threshold scheme with n redundant secret shadows, and then, sends n normal secret shadows to n participants. If the threshold needs to be changed to t', the dealer broadcasts $n - t' + 1$ redundant secret shadows. Then, t' or more participants can reconstruct the secret by providing their own secret shadows. In other schemes, in order to validate the new threshold, the dealer broadcasts special information, such as a mask code for the secret [27] and a key for encrypting/decrypting secret shadows [16]. In these schemes discussed, the dealer prepares all secret shadows (also known as advance secret shadows) for potential changeable thresholds during the initialization phase.

Other efforts have also been made on the security and application of TCSS techniques. In 2013, for example, Rao et al. [29] proposed a dynamic threshold multi-secret sharing scheme using Pell's Equation with Jacobi symbol. In their scheme, participants can verify their secret shadows, which avoid the situation of participants receiving the nugatory information given by the dealer. More recently, in 2015, Wang et al. [30] proposed a dynamic threshold changeable multi-policy secret sharing scheme, based on RSA cryptography and discrete logarithm technique. Their scheme reduces the communication costs and can resist multiform cheating. In the same year, Harn and Hsu [31] proposed a threshold changeable secret sharing scheme

based on bivariate polynomial, designed to protect the reconstructed secret from illegal participants.

Zhang et al. [16] proposed $(t \rightarrow t', n)$ and $(\{t_1, t_2, \cdots, t_N\}, n)$ TCSS schemes (hereafter referred to as TCSS-A and TCSS-B schemes). The TCSS-B scheme was the first scheme that could resist collusion attacks launched by participants who have historical secret shadows. However, their schemes suffer from a number of limitations, namely: strict limit on threshold values, large storage space requirement for secret shadows, and significant computation requirement for constructing and recovering polynomials. Thus, in this paper, we propose two improved dealer-free threshold changeable secret sharing schemes, DTCSS-A and DTCSS-B schemes. In our schemes, we construct polynomials to update secret shadows, and use two-variable one-way function to resist collusion attacks and protect the information stored by the combiner. Compared with Zhang et al.'s schemes, our schemes have following advantages:

1. No limitation on threshold values. New threshold t' must be greater than initial threshold t in the TCSS-A scheme, and N potential thresholds $t_1, t_2, \cdots, t_N$ must satisfy $0 < t_{i+1} - t_i < t_1$ $(i = 1, 2, \cdots, N-1)$ in the TCSS-B scheme. However, such limitations are avoided in our schemes.
2. Only one shadow storage requirement. Each participant needs to store $t' - t + 1$ secret shadows in the TCSS-A scheme and N secret shadows in the TCSS-B scheme. In our schemes, only one secret shadow needs to be stored.
3. Less computation. A total of $t' - t + 1$ polynomials need to be constructed and recovered in the TCSS-A scheme, and N polynomials need to be constructed and recovered in the TCSS-B scheme. In our schemes, only one polynomial needs to be constructed and recovered. Thus, our schemes require significantly less computational effort.
4. Dealer-free. Zhang et al.'s schemes require the dealer's assistance in the running phase, unlike our schemes. Thus, our schemes can reduce the single point-of-attack risk (i.e., attackers only need to target the dealer in the attempt to obtain the secret).
5. Secret shadow reusability. In our scheme, the secret shadow can be reused in new secret reconstruction; thus, increasing the efficiency.

The rest of this paper is organized as follows. Section 2 introduces related concepts, two-variable one-way function and the obligations of participants. The proposed threshold changeable schemes are presented in Section 3. In Section 4, we demonstrate the security of our schemes, and evaluate the performance of our schemes with those of Zhang et al.'s. We also discuss how our schemes can deal with the situation where the threshold needs to be adjusted. Section 5 concludes the paper.

Preliminaries

In this section, we explain the relevant concepts, two-variable one-way function and the obligations of participants.

Conceptions

We introduce the related conceptions as follows:

(1) Communication modes

Two communication modes are used in TCSS schemes (i.e. secure communication channels and broadcasting). It may be impractical to maintain mutual secure communication when the threshold changes, especially when the change is sudden. In our schemes, important

information such as real secret shadows are sent using RSA-based technique, and we validate the new threshold using broadcasting. We refer interested readers to [32–34] for an overview of secure communication techniques, as this is beyond the scope of this paper.

(2) Dealer-free

TSS technology is generally used to protect the secret key. For example, even if $n - t$ participants lose their secret shadows, the remaining t participants are still able to recover the secret. Deploying TSS scheme can also improve the security of the system, as an attacker requires no less than t secret shadows to recover the secret. In traditional TSS schemes, after generating and distributing secret shadows, the dealer destroys the secret and exits. While in some TCSS schemes, the dealer is online until the secret is recovered by participants. For example, in Zhang et al.'s schemes, since the dealer needs to adjust the threshold and deal with the enrollment and disenrollment of the participant, he/she holds the secret and all secret shadows in the running phase until the secret is recovered. Thus, attackers only need to target the dealer in the attempt to obtain the secret. This results in single point of attack.

However, in our schemes, we use the combiner to take the dealer's obligations in the running phase, and use two-variable one-way function to protect the information stored by the combiner. Thus, our schemes can update / revise the threshold in the running phase without the dealer's involvement, which means our schemes is dealer-free. Meanwhile, our schemes protect the secret from being recovered when attackers have access to the information stored by the combiner. Hence, our schemes are more secure.

(3) Collusion attack

The (t, n) threshold secret sharing scheme can resist up to $t - 1$ collusion participants who have secret shadows. However, in the TCSS schemes of [21, 24, 28] based on the advance secret shadow technique, participants have both historical and current secret shadows after changing the threshold. Therefore, such schemes cannot resist attacks carried out by $t - 1$ colluding participants. Many schemes [21, 24, 28] require that all participants destroy the historical shadows if the threshold has been changed, but this may be unrealistic in practice (i.e. we are trusting the bad guys to do the right thing). Thus, Zhang et al. [16] proposed the first scheme (i.e. TCSS-B) designed to resist such collusion attack, by encrypting secret shadows and validating the new threshold with the corresponding key. In our schemes, two-variable one-way function is used to protect secret shadows from collusion attacks.

Two-variable One-way Function

In this section, we introduce two-variable one-way function used in our schemes. Function $f(r, s)$ is a two-variable one-way function, which maps variables r and s into a value with a fixed length. The features of $f(r, s)$ are as follows [35]:

1. Given r and s, it is easy to compute $f(r, s)$.
2. Given s and $f(r, s)$, it is not feasible to compute r.
3. It is not feasible to compute $f(r, s)$ for any r without s.
4. Given s, it is not feasible for r_i and r_j to satisfy $f(r_i, s) = f(r_j, s)$, when $r_i \neq r_j$.
5. Given any pairs of $(r_i, f(r_i, s))$, it is not feasible to compute s.
6. Given any pairs of $(r_i, f(r_i, s))$, it is not feasible to compute $f(r_j, s)$, when $r_i \neq r_j$.

Assume that $|f(r, s)| \leq q$, so $f(r, s) \in GF(q)$. He and Dawson [36] proved the existence of two-variable one-way function, and also brought up the methods to construct it. For example, let S be a secure signature scheme. For a message m, the signature with secure key k is denoted by $S(k, m)$.

Let H be a universal one-way hash function whose existence is based on any one-to-one, one way function [37]. Two-variable one-way function $f(x, y)$ can be constructed as $f(x, y) = H(S(x, y))$.

In our schemes, each participant P_i ($1 \leq i \leq n$) selects his/her own variable s_i (also referred to as real secret shadow), and the combiner has all variables $r_1, r_2, \cdots, r_k$ ($k = t_{\max} - t_{\min} + 1$ in DTCSS-A scheme and $k = N$ in DTCSS-B scheme). By using two-variable one-way function, our schemes have the following advantages:

1. Collusion attack resistance: In our schemes, if and only if no less than current threshold (i.e., t_j) participants wish to recover the secret, the combiner broadcasts the corresponding variant r_j to validate participants' fake secret shadows. Colluding participants cannot obtain the historical shadows to recover the secret. Thus, our schemes can resist attacks carried out by $t_j - 1$ colluding participants who have both current and historical shadows.
2. Single point attack resistance: In our schemes, even if attackers obtain the information r_j stored in the combiner, they cannot compute the $f(r_j, s_i)$ without s_i. Thus, our schemes can avoid the limitation that attackers only need to target a single point in the attempt to obtain the secret.
3. Real secret shadow reusability: In our scheme, the real secret shadow can be reused in new secret reconstruction, thus, increasing the efficiency.

Participants

There are $n + 2$ ($n > 2$) members in our schemes, including n participants, the dealer and the combiner. The obligations of these participants are as follows:

Participants: There are n participants who hold secret shadows. In the running phase, only equal to or greater than threshold value participants can cooperate to recover the secret.

The dealer: In the initialization phase, the dealer generates each participant's advance secret shadows, and prepares for possible threshold change.

The combiner: In the running phase, the combiner adjusts the threshold value according to changes in the security policies and adversary structures prior to recovering the secret. Only if equal to or greater than threshold participants wish to recover the secret, then the combiner broadcasts the corresponding key to validate these participants' current secret shadows. Once successfully validated, these participants can recover the secret.

In generally, there are only participants and dealers in (t, n) threshold secret sharing scheme. However, to avoid the dealer single point attack, we introduce a combiner. The combiner can be used to adjust threshold and validate participants' corresponding secret shadows. We assume both dealer and combiner are trusted.

Proposed Schemes

In this section, we introduce our schemes (i.e., DTCSS-A and DTCSS-B schemes). The notions and parameters used in our schemes are outlined in Table 1.

In our two DTCSS schemes, there are $n + 2$ ($n > 2$) members (i.e., n participants, the dealer and the combiner), and their message flows are shown in Fig 1. Specifically, real secret shadows (sent to the dealer by participants) and shadow activation information (sent to the combiner by the dealer) are sent using RSA-based technique, and other information is sent via broadcasting.

$(t \rightarrow t', n)$ Threshold Changeable Scheme

In this section, we present the dealer-free threshold changeable secret sharing scheme based on broadcasting (DTCSS-A), where the polynomial is used as the secret shadow updating

Table 1. Summary of Notations.

Notation	Meaning
n	Number of participants
t	Threshold value
P_i	Participant i
P	Participant set, $P = \{P_1, P_2, \cdots, P_n\}$
q	A big prime number randomly chosen by the dealer, $q > n$
S	Domain of the secret, $S = GF(q)$
s	Secret, $s \in S$
S_i	Domain of participant P_i's secret shadow, $S_i = GF(q)$
s_i	Participant P_i's secret shadow, $s_i \in S_i$
T	Domain of potential threshold
t'	New threshold in DTCSS-A scheme
N	Number of potential thresholds in DTCSS-B scheme
$h(x)$	A polynomial
$h(x_i)$	Value of polynomial $h(x)$ in a given x_i
y_i^j	Participant P_i's j^{th} advance secret shadow
ψ_i	Participant P_i's secret shadow updating function
$f(r, s)$	A two-variable one-way function
deg(·)	Operator is used for computing the degree of the polynomial
$[x^k]$	Coefficient operator. If $h(x) = \sum_{i\geq 0} a_i x^i$, then $[x^k]\, h(x) = a_k$.
$[\cdot]_k$	Polynomial operator. If $h(x) = \sum_{i\geq 0} a_i x^i$, $[h(x)]_k = \sum_{i=0}^{k-1} a_i x^i$.

doi:10.1371/journal.pone.0165512.t001

function. This scheme is designed to convert a (t, n) scheme into a (t', n) scheme, where $t_{\min} \leq t' \leq t_{\max}$.

Assume that the dealer knows the changeable threshold domain $T = \{t_{\min}, t_{\min} + 1, \cdots, t_{\max}\}$, where $2 \leq t_{\min} \leq t_{\max} \leq n$ and $t_{\max}, t_{\min} \in \mathbb{Z}$. Then, using the RSA-based technique, the

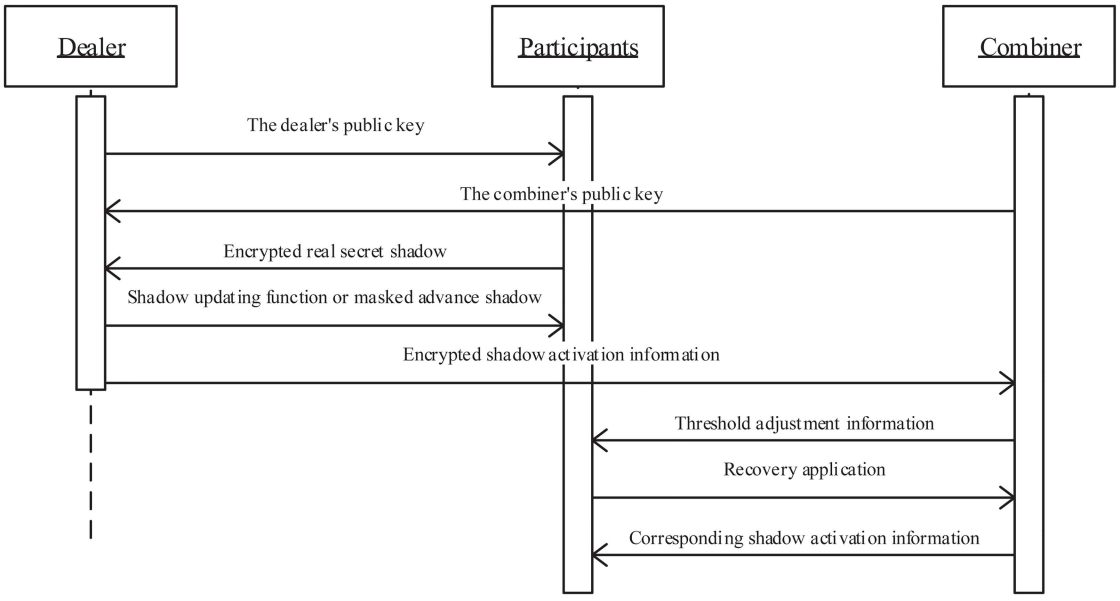

Fig 1. Sequence diagram of our schemes.

doi:10.1371/journal.pone.0165512.g001

dealer negotiates the real shadow with each participant. The dealer generates each participant's advance secret shadows and secret shadow updating function, and publishes these functions. Prior to exiting, the dealer sends the information used to validate the secret shadow to the combiner. Based on any updates to the security policy and adversary structure, the combiner adjusts the threshold to a suitable value t'. If no less than t' participants wish to recover the secret, then the combiner broadcasts $r_{t_{t'-t_{min}+1}}$. Therefore, these participants can recover the secret. DTCSS-A scheme consists of three phases as follows:

1. Secret shadows negotiation phase

In this phase, the dealer creates the notice table. Participants choose their own real secret shadows, and send them to the dealer using the underlying RSA technique. This phase has the following steps:

1. Notice table creation: To broadcast the message, the dealer creates a notice table, which can only be used for broadcasting information by the dealer and the combiner. Participants can obtain the information from the notice table, but they are unable to broadcast or modify the table.

2. Secret shadows negotiation initialization: Let $M_1 = p_1 \times p_2$ and $\varphi(M_1) = (p_1 - 1) \times (p_2 - 1)$, where p_1, p_2 are big prime numbers chosen randomly by the dealer. The dealer chooses an integer $e_1 < \varphi(M_1)$, which is co-prime with $\varphi(M_1)$. Then, the dealer computes the integer d_1, such that $e_1 d_1 \equiv 1(\mathrm{mod}\varphi(M_1))$, and broadcasts $\{e_1, M_1\}$ in the notice table. Similar, the combiner also generates his/her own $\{e_2, d_2, M_2\}$, and broadcasts $\{e_2, M_2\}$ in the notice table.

3. Real secret shadow generation and transfer: Each participant P_i $(1 \leq i \leq n)$ chooses a real secret shadow $s_i \in S_i$ randomly and sends C_i to the dealer, where $C_i = (s_i)^{e_1} \bmod M_1$. After receiving C_i, the dealer recovers the real secret shadow $s_i = (C_i)^{d_1} \bmod M_1$, and then ensures all participants choose distinct secret shadows. If two or more participants choose the same secret shadow, they will be asked to choose their secret shadows again until all secret shadows are distinct.

2. Initialization phase

In this phase, the dealer constructs the polynomial and generates each participant's advance shadows. The work-flow of this phase is shown in Fig 2, and the working steps are as follows:

(1) Polynomial construction: To share a secret $s \in S$, the dealer constructs polynomial $h(x)$ as

$$h(x) = (s + a_1 x^1 + \cdots + a_{t_{\max}-1} x^{t_{\max}-1}) \bmod q, \tag{1}$$

where $a_1, a_2, \cdots, a_{t_{\max}-1} \in GF(q)$ are chosen randomly. Let $h_j(x) = [h(x)]_{t_{\min}+j-1}$ for all $1 \leq j \leq t_{\max} - t_{\min} + 1$. Polynomial $h_j(x)$ can be generated by Algorithm 1.

Algorithm 1: Polynomial generator 1

```
Input: h(x), j, t_min, t_max
Output: h_j(x)
  h_j(x) = h(x);
  d = t_min + j - 1;
  While d ≤ t_max - 1 do
    c = [x^d] h(x);
    h_j(x) = h_j(x) - c x^d;
    d = d + 1;
  end
```

(2) Secret shadow updating functions construction: The dealer selects $t_{\max} - t_{\min} + 1$ distinct and nonzero integers $r_1, r_2, \cdots, r_{t_{\max}-t_{\min}+1} \in GF(q)$ as keys. There is one-to-one correspondence

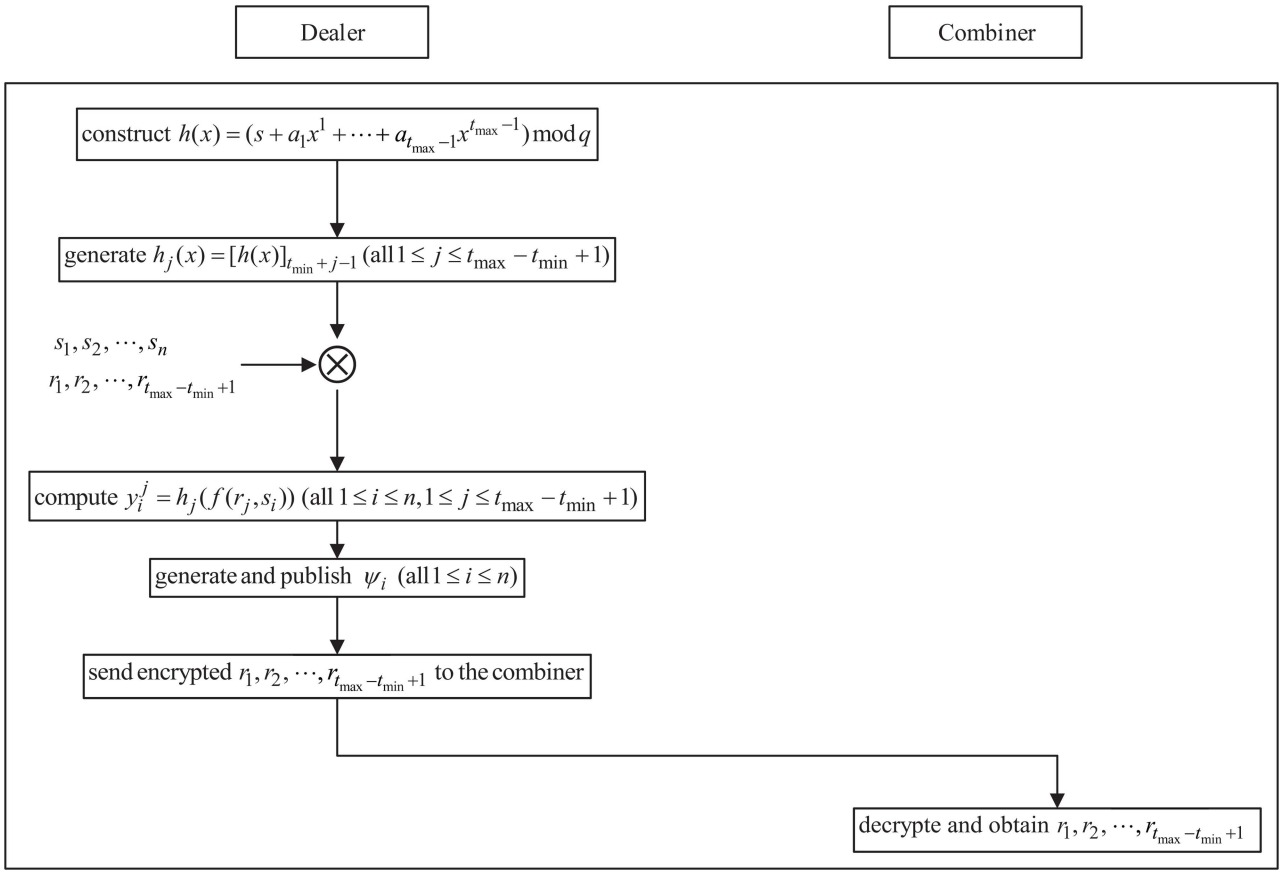

Fig 2. The work-flow of the initialization phase of DTCSS-A scheme.

doi:10.1371/journal.pone.0165512.g002

between keys $r_1, r_2, \cdots, r_{t_{\max}-t_{\min}+1}$ and thresholds $t_{\min}, t_{\min}+1, \cdots, t_{\max}$. Then, each participant's advance secret shadows $y_i^1, y_i^2, \cdots, y_i^{t_{\max}-t_{\min}+1}$ $(1 \le i \le n)$ can be computed as

$$y_i^j = h_j(x_i^j) \ \ (j = 1, 2, \cdots, t_{\max} - t_{\min} + 1), \tag{2}$$

where $x_i^j = f(r_j, s_i)$.

With $t_{\max} - t_{\min} + 1$ points $(x_i^1, y_i^1), (x_i^2, y_i^2), \cdots, (x_i^{t_{\max}-t_{\min}+1}, y_i^{t_{\max}-t_{\min}+1})$, each participant's secret shadow updating function ψ_i $(1 \le i \le n)$ can be constructed as follows:

$$\psi_i(x) = \sum_{j=1}^{t_{\max}-t_{\min}+1} y_i^j \prod_{k=1,k\neq j}^{t_{\max}-t_{\min}+1} \frac{x - x_i^k}{x_i^j - x_i^k} \ \bmod \ q. \tag{3}$$

Then, these functions are placed into notice table.

(3) Data transfer: To validate the new threshold in next phase, the dealer sends keys $r_1, r_2, \cdots, r_{t_{\max}-t_{\min}+1}$ to the combiner before exiting. Note that these keys need to be encrypted by the combiner's public key. Upon receiving this encrypted information, the combiner decrypts it.

3. Running phase

In this phase, the security policy and adversary structure may change, which necessitates a threshold change prior to recovering the secret. Once the threshold has been revised, the secret can then be recovered with the most recently broadcasted threshold. The work-flow of the running phase is shown in Fig 3, and the working steps are as follows:

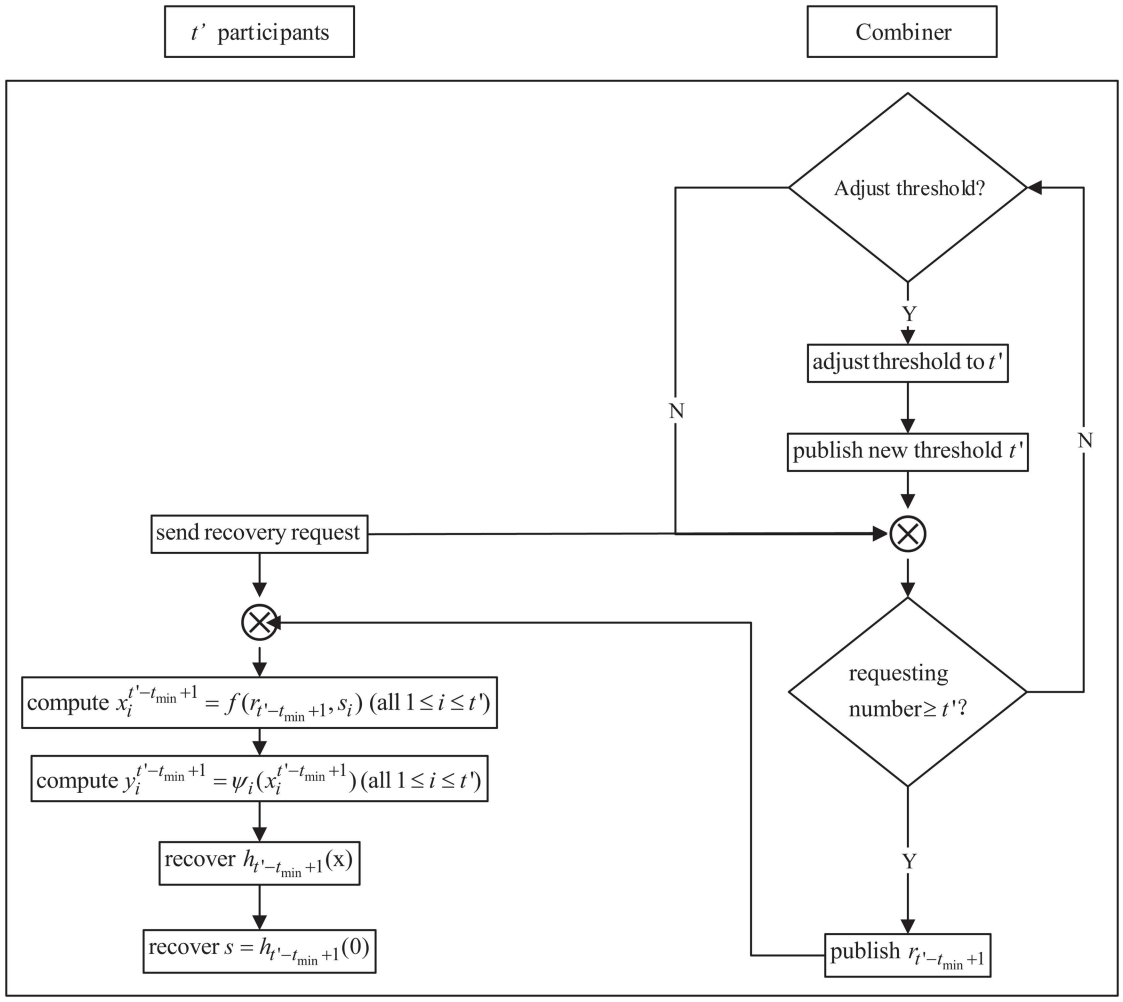

Fig 3. The work-flow of the running phase of DTCSS-A scheme.

doi:10.1371/journal.pone.0165512.g003

(1) Threshold adjustment: Based on the changes in the security policy and adversary structure, the combiner selects a suitable threshold t' ($t_{\min} \le t' \le t_{\max}$), and inserts it into the notice table. The threshold can be adjusted many times before the secret is recovered.

(2) Shadow activation: If participants wish to recover the secret, they can send the recovery requests to the combiner. When t' or more participants wish to recover secret s, the combiner broadcasts corresponding key $r_{t_{t'-t_{min}+1}}$. Then, each participant P_i ($1 \le i \le n$) can obtain her/his current secret shadow $y_i^{t'-t_{\min}+1} = \psi_i(x_i^{t'-t_{\min}+1})$, where $x_i^{t'-t_{\min}+1} = f(r_{t'-t_{\min}+1}, s_i)$.

(3) Secret recovery: Without loss of generality, we assume that t' participants $P_1, P_2, \cdots, P_{t'}$ wish to recover secret s. With t' points $(x_1^{t'-t_{\min}+1}, y_1^{t'-t_{\min}+1}), (x_2^{t'-t_{\min}+1}, y_2^{t'-t_{\min}+1}), \cdots, (x_{t'}^{t'-t_{\min}+1}, y_{t'}^{t'-t_{\min}+1})$, polynomial $h_{t'-t_{\min}+1}(x)$ can be recovered as

$$h_{t'-t_{\min}+1}(x) = \sum_{i=1}^{t'} y_i^{t'-t_{\min}+1} \prod_{j=1, j\neq i}^{t'} \frac{x - x_j^{t'-t_{\min}+1}}{x_i^{t'-t_{\min}+1} - x_j^{t'-t_{\min}+1}} \mod q, \quad (4)$$

where $x_i^{t'-t_{\min}+1} = f(r_{t'-t_{\min}+1}, s_i)$ $(1 \le i \le t')$.

Then, we can recover secret s by computing $s = h_{t'-t_{\min}+1}(0)$.

$(\{t_1, t_2, \cdots, t_N\}, n)$ Threshold Changeable Scheme

In this section, we present our $(\{t_1, t_2, \cdots, t_N\}, n)$ TCSS scheme (i.e., DTCSS-B scheme). Usually, N is a small integer. For example, when $N = 3$, $\{t_1, t_2, t_3\}$ correspond to the "low, middle, high" level of security in computers. Meanwhile, even if t_N is small, we always have $t_j \geq j$ for all $1 \leq j \leq N$.

Let t_k $(1 \leq k \leq N)$ be the value of initial threshold, and t_j $(1 \leq j \leq N)$ be the value of the new threshold. Assume that the dealer knows N potential thresholds $t_1, t_2, \cdots, t_N$, where the threshold may be changed in the future and $t_1 < t_2 < \cdots < t_N$. Similar to the DTCSS-A scheme, the dealer negotiates the real shadow. The dealer generates each participant's advance secret shadow. Prior to exiting, the dealer sends the information used to update and validates the secret shadow to the combiner. If the security policy or adversary structure changes, then the combiner adjusts the threshold to a suitable value t_j, and broadcasts corresponding masked advance secret shadows. If no less than t_j participants wish to recover the secret, then the combiner broadcasts r_j. Thus, these participants can recover the secret. The DTCSS-B scheme consists of three phases, namely: secret shadows negotiation, initialization and running.

1. Secret shadows negotiation phase

Similar to the DTCSS-A scheme, the dealer creates the notice table, and participants choose their own real secret shadows and send them to the dealer based on the underlying RSA technique.

2. Initialization phase

In this phase, the dealer constructs the polynomial to protect the secret and generates the advance secret shadows for all participants. The work-flow of the running phase is shown in Fig 4, and the working steps are as follows:

(1) Polynomial construction: To share a secret s, polynomial $h(x)$ is constructed as

$$h(x) = (s + a_1 x^1 + \cdots + a_{t_N-1} x^{t_N-1}) \mod q, \tag{5}$$

where $a_1, a_2, \cdots, a_{t_N-1} \in GF(q)$ are chosen randomly. For all $1 \leq j \leq N$, let polynomial $h_j(x) = [h(x)]_{t_j}$. Polynomial $h_j(x)$ can be generated by Algorithm 2.

Algorithm 2: Polynomial generator 2

```
Input: h(x), j, t_j, t_N
Output: h_j(x)
  h_j(x) = h(x);
  d = t_N - t_j;
  While d > 0 do
    c = [x^{t_j+d-1}] h(x);
    h_j(x) = h_j(x) - c x^{t_j+d-1};
    d = d - 1;
  end
```

(2) Advance secret shadows generation: The dealer chooses N distinct and nonzero integers $r_1, r_2, \cdots, r_N \in GF(q)$ as keys. There is one-to-one correspondence between keys $r_1, r_2, \cdots, r_N$ and potential thresholds $t_1, t_2, \cdots, t_N$. Each participant's advance secret shadows $y_i^1, y_i^2, \cdots, y_i^N$ $(1 \leq i \leq n)$ are computed as follows:

$$y_i^j = h_j(f(r_j, s_i)) \ \ (j = 1, 2, \cdots, N) \tag{6}$$

Then, the dealer places masked advance secret shadows $y_1^k / f(r_k, s_1), y_2^k / f(r_k, s_2), \cdots, y_n^k / f(r_k, s_n)$ into the notice table.

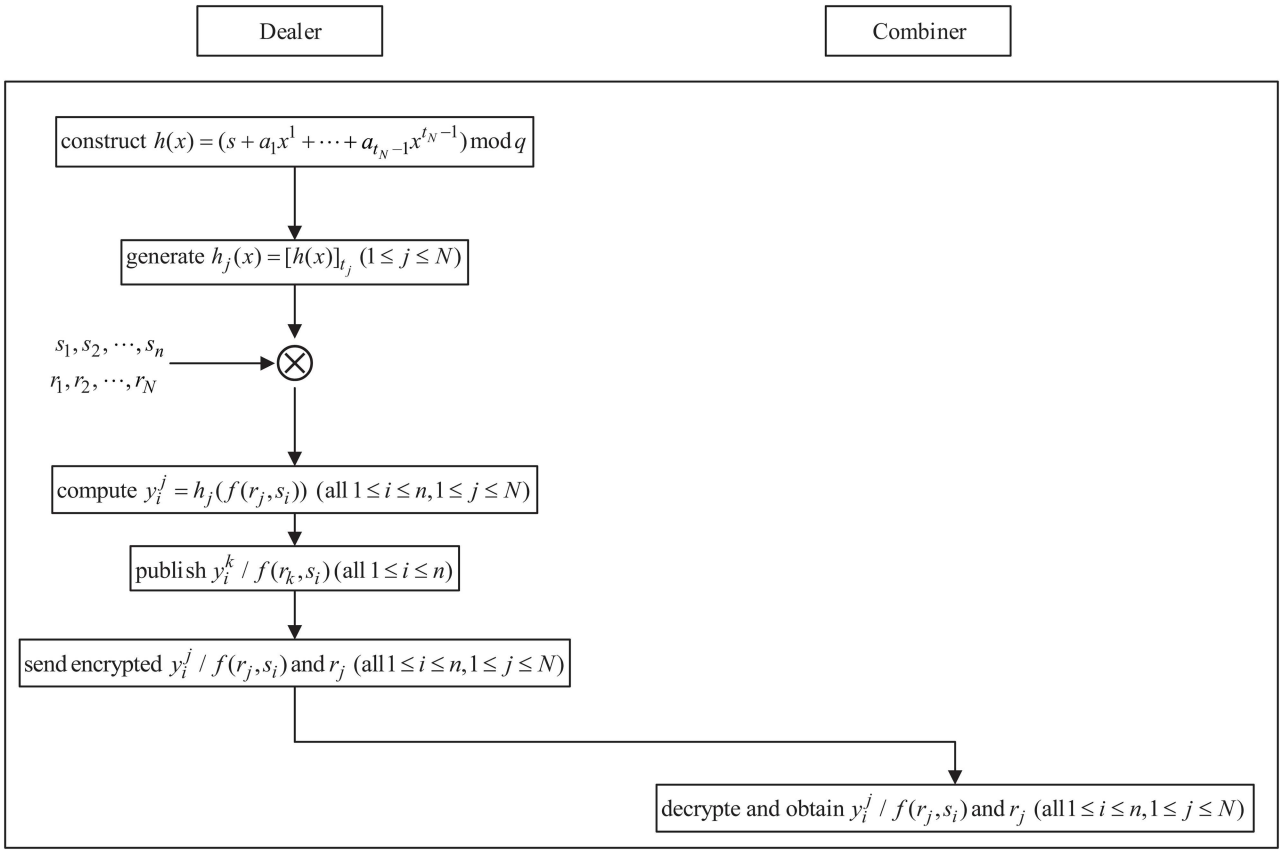

Fig 4. The work-flow of the initialization phase of DTCSS-B scheme.

doi:10.1371/journal.pone.0165512.g004

(3) Data transfer: To validate the new threshold by the combiner in the next phase, the dealer sends $y_1^j/f(r_j,s_1), y_2^j/f(r_j,s_2), \cdots, y_n^j/f(r_j,s_n)$ (j = 1, 2,· · ·, N) and r_1, r_2,· · ·, r_N to the combiner, and then, he/she exits. Note that this information needs to be encrypted by the combiner's the public key. After receiving this encrypted information, the combiner decrypts it.

3. Running phase

In this phase, the security policy and adversary structure may change; thus, updating / revising the threshold is necessary before recovering the secret. The secret can then be recovered with the most recently broadcasted threshold. The work-flow of the running phase is shown in Fig 5, and the working steps are as follows:

(1) Threshold adjustment: Based on the changes in the security policy and adversary structure, the combiner selects a suitable threshold t_j $(1\le j\le N)$. Then, he/she places t_j and $y_1^j/f(r_j,s_1), y_2^j/f(r_j,s_2), \cdots, y_n^j/f(r_j,s_n)$ into the notice table. The threshold can be changed many times.

(2) Shadow activation: If participants wish to recover the secret, they can send the recovery requests to the combiner. When t_j or more participants wish to recover secret s, the combiner broadcasts key r_j to validate participants' secret shadows, and then, each participant P_i $(1\le i\le n)$ can obtain the current secret shadow $(f(r_j,s_i), y_i^j)$.

(3) Secret recovery: Without loss of generality, assume that t_j participants $P_1, P_2, \cdots, P_{t_j}$ provide their current secret shadows $(f(r_j,s_1), y_1^j), (f(r_j,s_2), y_2^j), \cdots, (f(r_j,s_{t_j}), y_{t_j}^j)$. Polynomial

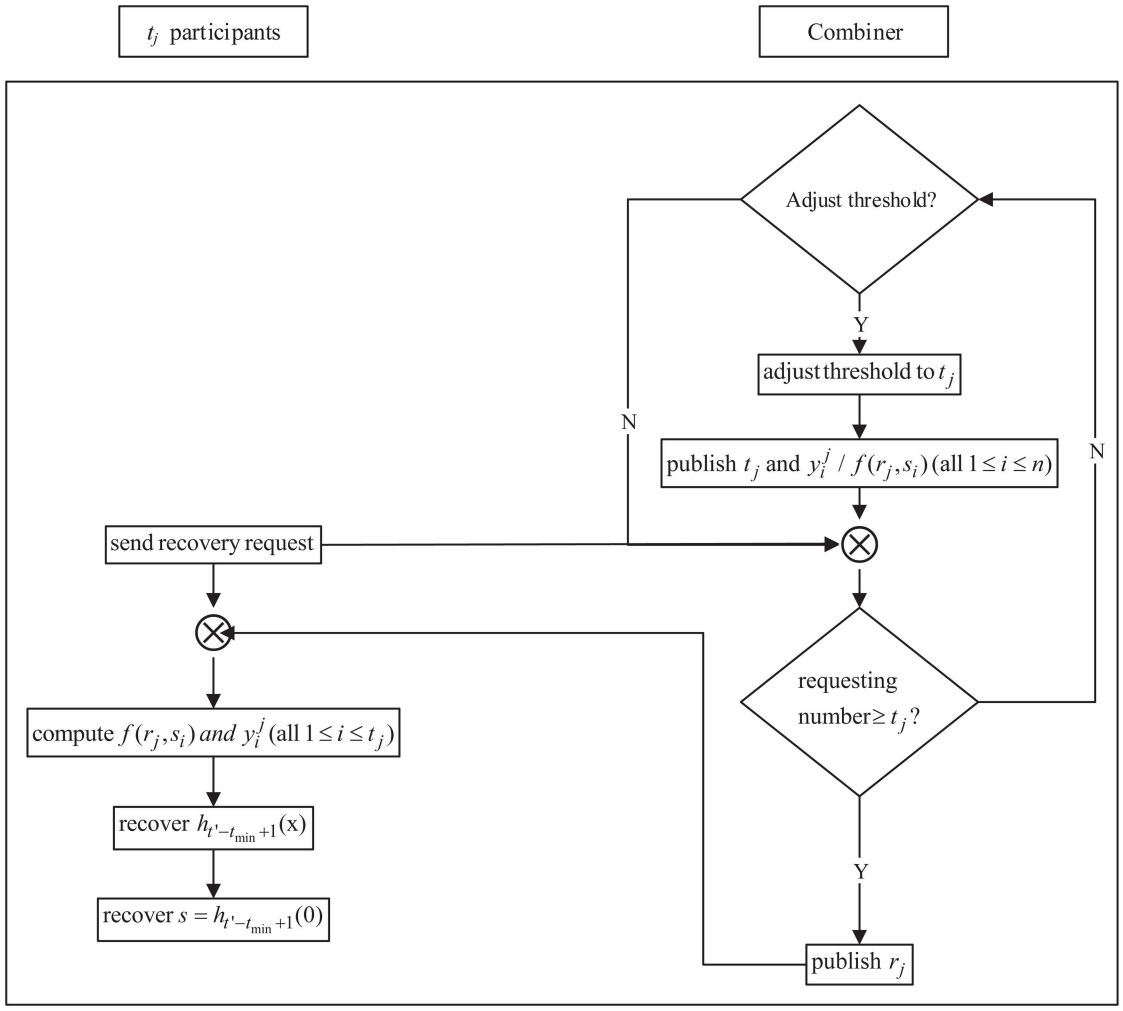

Fig 5. The work-flow of the initialization phase of DTCSS-B scheme.

doi:10.1371/journal.pone.0165512.g005

$h_j(x)$ can be restructured as follows:

$$h_j(x) = \sum_{i=1}^{t_j} y_i^j \prod_{k=1,k\neq i}^{t_j} \frac{x - f(r_j, s_k)}{f(r_j, s_i) - f(r_j, s_k)} \mod q \quad (7)$$

Then, secret s can be recovered as $s = h_j(0)$.

Analysis

Security Analysis

In this section, we discuss and analyze the security of our schemes.

Theorem 1. In our schemes, each participant P_i $(1 \le i \le n)$ is unable to obtain the valid secret shadow before the combiner broadcasts the key which corresponds to the current threshold.

Proof. Let the current threshold be t' in the DTCSS-A scheme, and t_j in the DTCSS-B scheme. In the DTCSS-A scheme, according to the features of two-variable one-way function, each participant P_i $(1 \le i \le n)$ is unable to obtain the valid secret shadow $(x_i^{t'-t_{\min}+1}, y_i^{t'-t_{\min}+1})$

before the combiner broadcasts $r_{t'-t_{\min}+1}$, where $x_i^{t'-t_{\min}+1} = f(r_{t'-t_{\min}+1}, s_i)$ and $y_i^{t'-t_{\min}+1} = \psi_i(x_i^{t'-t_{\min}+1})$.

Similarly, in the DTCSS-B scheme, each participant P_i is unable to obtain the valid secret shadow (x_i^j, y_i^j) without r_j, where $x_i^j = f(r_j, s_i)$ and $y_i^j = (y_i^j / f(r_j, s_i)) \times x_i^j$.

Thus, in our schemes, each participant P_i $(1 \le i \le n)$ is unable to obtain the valid secret shadow prior to the combiner broadcasting the corresponding key.

The combiner broadcasts the key, if and only if the number of participants who wish to recover the secret is equal to or more than the current threshold value. Meanwhile, by Theorem 1, we know that each participant P_i $(1 \le i \le n)$ is unable to compute the valid secret shadow without the key. Thus, participants do not have access to the historical secret shadow, or cannot undermine the security of our schemes using historical secret shadows.

Theorem 2. Less than current threshold value participants are unable to recover the secret.

Proof. Let the current threshold be t' in the DTCSS-A scheme, and t_j in the DTCSS-B scheme. In the DTCSS-A scheme, without loss of generality, assume that $t' - 1$ participants P_1, $P_2, \cdots, P_{t'-1}$ wish to recover the secret after the combiner broadcasts $r_{t'-t_{\min}+1}$. Then, they can obtain $t'-1$ points $(x_1^{t'-t_{\min}+1}, y_1^{t'-t_{\min}+1}), (x_2^{t'-t_{\min}+1}, y_2^{t'-t_{\min}+1}), \cdots, (x_{t'}^{t'-t_{\min}+1}, y_{t'}^{t'-t_{\min}+1})$, where $x_i^{t'-t\min^{+1}} = f(r_{t'-t\min^{+1}}, s_i)$ and $y_i^{t'-t\min^{+1}} = \psi_i(x_i^{t'-t\min^{+1}})$. Utilizing these points for each candidate point (x', y') $(x', y' \in GF(q))$, they can reconstruct one and only one polynomial $h^*_{t'-t\min^{+1}}(x)$ of degree $t'-1$, which satisfy $h^*_{t'-'-t\min^{+1}}(x') = y'$ and $h^*_{t'-t\min^{+1}}(x_i^j) = y_i^j$ $(1 \le i \le t')$. Constructed in the same way, these possible polynomials are equally likely; thus, there is nothing an attacker can deduce about the real polynomial $h_{t'-t_{\min}+1}(x)$. Thus, they cannot recover secret s as in $s = h_{t'-t_{\min}+1}(0)$.

Similarly, in the DTCSS-B scheme, after the combiner has broadcasted r_j $(1 \le j \le N)$, $t_j - 1$ participants are unable to recover the corresponding polynomial $h_j(x)$. Thus, they cannot recover secret s as in $s = h_j(0)$.

Thus, in our schemes, less than the current threshold value participants are unable to recover the secret.

By Theorem 1 and Theorem 2, we know that the secret can be recovered, if and only if equal to or greater than current threshold participants provide their valid secret shadows. Thus, our schemes are secure under changing threshold.

Theorem 3. Attackers are unable to recover secret s using only the information stored by the combiner.

Proof. In the DTCSS-A scheme, according to the features of two-variable one-way function, attackers who obtain keys $r_1, r_2, \cdots, r_{t_{\max}-t_{\min}+1}$ cannot compute any participant's secret shadow (x_i^j, y_i^j) $(1 \le i \le n, 1 \le j \le t_{\max} - t_{\min} + 1)$ without s_i, where $x_i^j = f(r_j, s_i)$ and $y_i^j = \psi_j(x_i^j)$. Since s_i is only known by the dealer and participant P_i, attackers cannot recover secret s.

Similarly, in the DTCSS-B scheme, even if attackers obtain $r_1, r_2, \cdots, r_N$ and $y_1^j / f(r_j, s_1), y_2^j / f(r_j, s_2), \cdots, y_n^j / f(r_j, s_n)$ $(j = 1, 2, \cdots, N)$ stored by the combiner, they cannot obtain any participant's secret shadows (x_i^j, y_i^j) $(1 \le i \le n)$ without s_i, where $x_i^j = f(r_j, s_i)$ and $y_i^j = (y_i^j / f(r_j, s_i)) \times x_i^j$.

Thus, attackers are unable to recover secret s using only the information stored by the combiner.

In no dealer-free schemes, the dealer may be compromised in the running phase, which results in the leakage of secrets and/or secret shadows. By Theorem 3, we know that our schemes can resist such attack. However, if the combiner's information is stolen by attackers, the attackers only need to collude with no less than the minimum threshold participants to

recover the secret. Note that the minimum thresholds are $t_{\min}$ and t_1 in the DTCSS-A scheme and DTCSS-B scheme, respectively.

Theorem 4. In our schemes, attackers are unable to obtain any legitimate participant's real secret shadow.

Proof. In the DTCSS-A scheme, assume that attackers wish to obtain the participant' secret shadow s_i $(1 \leq i \leq n)$, and they can obtain the exchanged information between the combiner and participant P_i. Then, they can obtain ψ_i, $r_{t'-t_{\min}+1}$ and $(x_i^{t'-t\min+1}, y_i^{t'-t\min+1})$, where $x_i^{t'-t\min+1} = f(r_{t'-t\min+1}, s_i)$ and $y_i^{t'-t\min+1} = \psi_i(x_i^{t'-t\min+1})$. According to the features of two-variable one-way function, attackers are unable to compute s_i from $r_{t'-t_{\min}+1}$ and $x_i^{t'-t\min+1}$.

Similarly, in the DTCSS-B scheme, attackers can obtain $r_j, y_i^j/f(r_j, s_i)$ and (x_i^j, y_i^j), where $x_i^j = f(r_j, s_i)$ and $y_i^j = y_i^j/f(r_j, s_i) \times x_i^j$. Thus, they cannot obtain s_i from r_j and x_i^j.

In summary, attackers cannot obtain any legitimate participant's real secret shadow in our schemes.

By Theorem 4, we know that attackers cannot obtain any participant's real secret shadow s_i $(1 \leq i \leq n)$, so s_i can be reused in subsequent scheme.

Comparative Summary

A comparative summary between our schemes and Zhang et al.'s schemes [16] are listed in Table 2.

From Table 2, we observe that our schemes have following advantages:

1. No limit on the threshold

The new threshold t' must satisfy $t < t' \leq n$ in the TCSS-A scheme and $t' \in \{t_1, t_2, \cdots, t_N\}$ in the TCSS-B scheme, where $0 < t_{i+1} - t_i < t_1$ $(i = 1, 2, \cdots, N-1)$. In both TCSS-A and TCSS-B schemes, the threshold can be changed only once. In our schemes, however, the new threshold t' can be smaller than the initial threshold t in our DTCSS-A scheme, and N potential thresholds do not need to satisfy $t_{i+1} - t_i < t_1$ $(i = 1, 2, \cdots, N-1)$ in our DTCSS-B scheme. In addition, the threshold of our schemes can be changed more than once.

2. Only one shadow storage requirement

In the TCSS-A scheme, each participant P_i $(1 \leq i \leq n)$ needs to store $t' - t + 1$ secret shadows $s_i^1, s_i^2, \cdots, s_i^{t'-t+1}$ in threshold t and one secret shadow $s_i^{t'-t+1}$ in threshold t'. In the TCSS-B scheme, all participants must store N secret shadows. However, in our schemes, each participant only has to store one secret shadow (i.e., s_i), which results in significant savings for storage.

Table 2. Comparative Summary.

	TCSS-A	DTCSS-A	TCSS-B	DTCSS-B
NTL	$t < t'$	$t_{\min} \leq t' \leq t_{\max}$	$t' \in \{t_1, t_2, \cdots, t_N\}$ $(0 < t_{i+1} - t_i < t_1)$	$t' \in \{t_1, t_2, \cdots, t_N\}$
NCP	$t' - t + 1$	1	N	1
NRP	$\frac{t'-t+2}{2}$	1	$\frac{N+1}{2}$	1
SSS	$(t' - t + 1) \log q$	$\log q$	$N \log q$	$\log q$
BCS	$\log q$	$\log q$	$\frac{N+1}{2} \log q$	$(n+1) \log q$

NTL: New threshold limitation; NCP: Number of constructing polynomials;
NRP: Number of recovering polynomials; SSS: Shadows storage space;
BCS: Broadcasting message space.

doi:10.1371/journal.pone.0165512.t002

3. Less computation

In the TCSS-A scheme, in order to change the threshold, $t' - t + 1$ polynomials $h_1(x)$, $h_2(x)$, $\cdots$, $h_{t'-t+1}(x)$ must be constructed in the initialization phase, and secret s is hidden in $t' - t$ coefficients of polynomial $h_{t'-t+1}(x)$, where $\deg(h_{t'-t+1}(x)) = t' - 1$. If the threshold is changed, t' or more participants can reconstruct polynomial $h_{t'-t+1}(x)$ directly. However, if the threshold is not changed, they have to determine polynomial $h_1(x)$ by polynomial interpolation, and then determine polynomials $h_2(x)$, $h_3(x)$, $\cdots$, $h_{t'-t+1}(x)$ in turn by computing:

$$h_{j+1}(x) = h_j(x) + \frac{y_i^{j+1} - h_j(x_i)}{(x_i)^{t+j-1}} x^{t+j-1} \ (1 \le j \le t' - t), \tag{8}$$

where (x_i, y_i^{j+1}) are provided by participant P_i who wants to recover the secret. Then, the secret can be obtained from the coefficients of polynomial $h_{t'-t+1}(x)$.

In the TCSS-B scheme, N polynomials $h_1(x)$, $h_2(x)$, $\cdots$, $h_N(x)$ must be constructed in the initialization phase. If the threshold is changed to t_j $(1 \le j \le N)$, then they have to determine polynomial $h_j(x)$ using polynomial interpolation, and then, in turn, determine polynomials $h_{j+1}(x)$, $h_{j+2}(x)$, $\cdots$, $h_N(x)$ to recover the secret from polynomial $h_N(x)$.

However, in our schemes, only one polynomial needs to be constructed, and other corresponding polynomials can be obtained using polynomial operator $[\cdot]_k$. In addition, participants only need to recover polynomial $h_{t'-t_{\min}+1}(x)$ in our DTCSS-A scheme and polynomial $h_j(x)$ in our DTCSS-B scheme. Thus, the computational cost in our schemes is significantly lower than those of Zhang et al.'s schemes.

4. Dealer-free

Unlike our proposed schemes, Zhang et al.'s schemes require the dealer's involvement in the running phase. By Theorem 3, we know that attackers are unable to recover the secret using only the information stored by the combiner. Thus, our schemes are more secure.

5. Secret shadow reusability

In Zhang et al.'s schemes, the secret shadow can be used to reconstruct only once, because those secret shadows are known to the participants who participate in recovering secret. However, in our schemes, the real secret shadow will not be leaked in recovering secret, which is demonstrated in Theorem 4. Thus, the real secret shadow can be reused to recover new secret, which results in increased efficiency.

Application

In practice, the threshold may have to be adjusted if there are changes in the security policies and adversary structures prior to recovering the secret. Examples of changes that require threshold adjustment include: (1) an increase or decrease in the importance level of the secret; (2) a change in participant number (i.e., one or more participants joining or leaving the group); (3) a change in the level of mutual trust between participants; and (4) the leakage of some participants' secret shadows. Our schemes can efficiently deal with these situations.

According to whether one or more secret shadows have been leaked, there are two kinds of situation in which the threshold needs to be adjusted:

1. No secret shadow leakage

This type includes the following 2 situation, i.e., an increase (decrease) in the importance level of the secret and a change in the level of mutual trust between participants. In these situations, all participants' secret shadows are secure, so we only need to adjust threshold directly. For example, if the importance level of the secret increases (decreases), we only need to increase (decrease) the threshold.

2. one or more secret shadows leakage

This type includes the following two situations, namely: a change of participant number and the leakage of some participants' secret shadows. In both situations, one or more participants' secret shadows would be leaked (or could be easily stolen by attackers). Therefore, it is not sufficient to only adjust the threshold. Here, we use a change in the participant number as an example to discuss the threshold changeability for enrollment and disenrollment.

1. Enrollment: If some person joins the group in the running phase, then new secret shadows would be distributed to these enrolled participants. Therefore, the threshold does not need to be adjusted. Since our schemes are dealer-free, new secret shadows cannot be generated in the running phase. However, we can prepare redundant secret shadows for enrolled participants in the initialization phase. If any user wishes to be enrolled to the group, he/she will send the request to the combiner. Then, the combiner will distribute the redundant secret shadow to this requesting user. Note that the number of enrolled participants is less than or equal to the number of the redundant secret shadows.
2. Disenrollment: If some participants leave the group, then their secret shadows are useless to them. It is reasonable to assume that we should not rely on these departing participants to secure their secret shadows. Thus, we assume that these shadows could be easily compromised. In Zhang et al.'s scheme, if participants leave the group, then the dealer will broadcast their secret shadows and adjust the threshold. This could result in an increased risk since attackers only have to defeat the dealer to obtain all secret shadows. To avoid such a limitation, we propose two dealer-free schemes. If participants publish their shadows to all other participants (i.e. they leave the group), then the threshold will be adjusted by the combiner, without the involvement of a dealer. Therefore, the security of our schemes can be guaranteed. The detailed solutions dealing with the disenrollment are described below.

Let k be the number of disenrolled participants, t the current threshold value, and $P = \{P_1, P_2, \cdots, P_n\}$ be the set of n participants. In the DTCSS-A scheme, if k participants broadcast their real secret shadows (e.g., when they leave the set P), then we obtain a new set P' that does not include the disenrolled participants. The combiner would then adjust the threshold from t to $t + k$. In other words, the original (t, n) scheme is changed to a $(t + k, n - k)$ scheme. Thus, t participants in set P' can use their own real secret shadows and k published real secret shadows to recover the secret. Note that the actual minimum number of participants (i.e., t) for recovering the secret does not change. In this situation, the actual maximum changeable threshold is changed to $t_{\max} - k$, and the maximum number of disenrolled participants is limited to $t_{\max} - t$. In the DTCSS-B scheme, k is required to satisfy the condition (i.e., $t + k \in \{t_1, t_2, \cdots, t_N\}$), and no more than $t_N - t$ participants can be allowed to leave the group.

We can also use the above described method to deal with secret shadow leakage. For example, if a participant's secret shadow leaks, then this person can leave the group before rejoining. Through these operations, the security of the scheme can be guaranteed.

In summary, our schemes can efficiently deal with the situation in which the threshold needs to be adjusted. Thus, our schemes have broad and promising application potential.

Conclusion

In this paper, we propose two improved dealer-free threshold changeable secret sharing schemes. By using two-variable one-way function, both schemes can resist collusion attacks launched by participants who hold both historical and current secret shadows. We also prove that our schemes can adjust the threshold safely, in the event that the security policy and adversary structure change. A comparative summary demonstrate that our schemes outperform Zhang et al.'s scheme, in terms of security and performance. Lastly, we discuss how our

schemes can deal with situations where the threshold needs to be adjusted; thus, demonstrating the utility of our schemes in real-world deployments.

However, in order to minimize the size of broadcast message, the proposed DTCSS-A scheme requires significant computations to construct the secret shadow updating function. Meanwhile, in order to resist collusion attacks, our schemes can only validate participants' secret shadow once. Thus, future research will include refining the scheme with the aim of improving its efficiency.

Acknowledgments

The authors are grateful to Jia Liu for the insightful comments and discussions, and the Editor and the anonymous reviewers for their invaluable feedback.

Author Contributions

Conceptualization: LFY MCL CG YZR.

Formal analysis: LFY YZR CG KKRC.

Funding acquisition: MCL CG.

Investigation: LFY CG YZR.

Methodology: YZR LFY KKRC.

Project administration: MCL YZR CG.

Resources: MCL CG.

Supervision: MCL YZR CG.

Visualization: LFY.

Writing – original draft: LFY CG YZR MCL KKRC.

Writing – review & editing: LFY YZR CG KKRC.

References

1. Choo KKR. The cyber threat landscape: Challenges and future research directions. Computers & Security. 2011; 30(8):719–731. doi: 10.1016/j.cose.2011.08.004
2. Choo KKR. In: Kaur H, Tao X, editors. A Conceptual Interdisciplinary Plug-and-Play Cyber Security Framework. ICTs and the Millennium Development Goals: A United Nations Perspective. Boston, MA: Springer US; 2014. p. 81–99. doi: 10.1007/978-1-4899-7439-6_6
3. Javanmardi S, Shojafar M, Shariatmadari S, Ahrabi SS. FRTRUST: a fuzzy reputation based model for trust management in semantic P2P grids. International Journal of Grid and Utility Computing. 2015; 6(1):57–66.
4. Li W, Song H. ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks. IEEE Transactions on Intelligent Transportation Systems. 2016; 17(4):960–969. doi: 10.1109/TITS.2015.2494017
5. Butun I, Erol-Kantarci M, Kantarci B, Song H. Cloud-centric multi-level authentication as a service for secure public safety device networks. IEEE Communications Magazine. 2016; 54(4):47–53. doi: 10.1109/MCOM.2016.7452265
6. Cordeschi N, Shojafar M, Amendola D, Baccarelli E. Energy-efficient adaptive networked datacenters for the QoS support of real-time applications. The Journal of Supercomputing. 2015; 71(2):448–478. doi: 10.1007/s11227-014-1305-8
7. Wang Z, Song H, Watkins DW, Ong KG, Xue P, Yang Q, et al. Cyber-physical systems for water sustainability: challenges and opportunities. IEEE Communications Magazine. 2015; 53(5):216–222. doi: 10.1109/MCOM.2015.7105668

8. Song J, Han C, Wang K, Zhao J, Ranjan R, Wang L. An integrated static detection and analysis framework for android. Pervasive and Mobile Computing. 2016. doi: 10.1016/j.pmcj.2016.03.003
9. Zhao J, Wang L, Tao J, Chen J, Sun W, Ranjan R, et al. A security framework in G-Hadoop for big data computing across distributed Cloud data centres. Journal of Computer and System Sciences. 2014; 80 (5):994–1007. doi: 10.1016/j.jcss.2014.02.006
10. Shamir A. How to share a secret. Communications of the ACM. 1979; 22(11):612–613. doi: 10.1145/359168.359176
11. Blakley GR. Universal one-way hash functions and their cryptographic applications. In: AFIPS 1979 National Computer Conference; 1979. p. 313–317.
12. Stinson DR. An explication of secret sharing schemes. Designs, Codes and Cryptography. 1992; 2 (4):357–390. doi: 10.1007/BF00125203
13. Beimel A. In: Chee Y, Guo Z, Ling S, Shao F, Tang Y, Wang H, et al., editors. Secret-Sharing Schemes: A Survey. Coding and Cryptology: Lecture Notes in Computer Science. Berlin Heidelberg: Springer Berlin Heidelberg; 2011. p. 11–46.
14. Peng K. Critical survey of existing publicly verifiable secret sharing schemes. Iet Information Security. 2012; 6(4):249–257. doi: 10.1049/iet-ifs.2011.0201
15. Laih CS, Harn L, Lee JY, Hwang T. In: Brassard G, editor. Dynamic threshold scheme based on the definition of cross-product in an N-dimensional linear space. Coding and Cryptology: Advances in Cryptology—Crypto 89: Proceedings. Berlin Heidelberg: Springer Berlin Heidelberg; 1990. p. 286–297.
16. Zhang Z, Chee YM, Ling S, Liu M, Wang H. Threshold changeable secret sharing schemes revisited. Theoretical Computer Science. 2012; 418(1):106–115. doi: 10.1016/j.tcs.2011.09.027
17. Martin MK. In: Ganley MJ, editor. Untrustworthy participants in perfect secret sharing schemes. Cryptography and Coding III. Oxford: Oxford University Press; 1993. p. 255–264.
18. Nojoumian M, Stinson DR. On dealer-free dynamic threshold schemes. Adv in Math of Comm. 2013; 7 (1):39–56. doi: 10.3934/amc.2013.7.39
19. Steinfeld R, Pieprzyk J, Wang H. Lattice-based threshold-changeability for standard CRT secret-sharing schemes. Finite Fields and Their Applications. 2006; 12(4):653–680. doi: 10.1016/j.ffa.2005.04.007
20. Khorasgani HA, Asaad S, Eghlidos T, Aref M. A lattice-based threshold secret sharing scheme. In: 2014 11th International ISC Conference on Information Security and Cryptology (ISCISC). IEEE; 2014. p. 173–179.
21. Steinfeld R, Pieprzyk J, Wang H. Lattice-based threshold changeability for standard shamir secret-sharing schemes. IEEE Transactions on Information Theory. 2007; 53(7):2542–2559. doi: 10.1109/TIT.2007.899541
22. Tartary C, Wang H. In: Lipmaa H, Yung M, Lin D, editors. Dynamic Threshold and Cheater Resistance for Shamir Secret Sharing Scheme. Information Security and Cryptology: Lecture Notes in Computer Science. Berlin Heidelberg: Springer Berlin Heidelberg; 2006. p. 103–117.
23. Desmedt Y, Jajodia S. Redistributing secret shares to new access structures and its applications; 1997. ISSE TR-97-01. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.2968&rep=rep1&type=pdf.
24. Chen L, Gollmann D, Mitchell CJ. In: Lomas M, editor. Key escrow in mutually mistrusting domains. Security Protocols: International Workshop Cambridge, United Kingdom, April 10-12, 1996 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 1997. p. 139–153.
25. Martin KM, Safavi-Naini R, Wang H. Bounds and techniques for efficient redistribution of secret shares to new access structures. The Computer Journal. 1999; 42(8):638–649. doi: 10.1093/comjnl/42.8.638
26. Blundo C, Cresti A, De Santis A, Vaccaro U. Fully dynamic secret sharing schemes. Theoretical Computer Science. 1996; 165(2):407–440. doi: 10.1016/0304-3975(96)00003-5
27. Martin KM, Pieprzyk J, Safavi-Naini R, Wang H. Changing thresholds in the absence of secure channels. Australian Computer Journal. 1999; 31(2):34–43.
28. Barwick SG, Jackson WA, Martin KM. Updating the parameters of a threshold scheme by minimal broadcast. IEEE Transactions on Information Theory. 2005; 51(2):620–633. doi: 10.1109/TIT.2004.840857
29. Rao MK, Sarma KVSSRSS, Avadhani PS, Bhaskari DL. In: Latifi S, Arai K, Dehnath N, Dias IAV, Garuba M, Hashemi R, et al., editors. A Model on Dynamic Threshold Multi-Secret Sharing Scheme using Pell's Equation with Jacobi Symbol. Information Technology: New Generations (ITNG), 2013 Tenth International Conference on. New York, NY, USA: IEEE; 2013. p. 773–776.

30. Wang F, Zhou YS, Li DF. Dynamic threshold changeable multi-policy secret sharing scheme. Security and Communication Networks. 2015; 8(18):3653–3658. doi: 10.1002/sec.1288

31. Harn L, Hsu CF. Dynamic threshold secret reconstruction and its application to the threshold cryptography. Information Processing Letters. 2015; 115(11):851–857. doi: 10.1016/j.ipl.2015.06.014

32. Song H, Brandt-Pearce M. A Discrete-Time Polynomial Model of Single Channel Long-Haul Fiber-Optic Communication Systems. In: 2011 IEEE International Conference on Communications (ICC); 2011. p. 1–6.

33. Nnanna E, Houbing S, Wei Y, Chao L, Yan W. In: Al-Sakib KP, editor. Securing Transportation Cyber-Physical Systems. Securing Cyber-Physical Systems. Boca Raton, Florida: CRC Press; 2015. p. 163–196.

34. Xu Q, Ren P, Song H, Du Q. Security Enhancement for IoT Communications Exposed to Eavesdroppers With Uncertain Locations. IEEE Access. 2016; 4:2840–2853. doi: 10.1109/ACCESS.2016.2575863

35. Chien HY, Jinn-Ke J, Tseng YM. cret sharing scheme. Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences. 2000; 83(12):2762–2765.

36. He J, Dawson E. Multisecret-sharing scheme based on one-way function. Electronics Letters. 1995; 31(2):93–95.

37. Naor M, Yung M. Universal one-way hash functions and their cryptographic applications. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing; 1989. p. 33–43.