Published in IET Information Security
Received on 10th June 2009
Revised on 27th January 2010
doi: 10.1049/iet-ifs.2009.0098

Special Issue on Multi-Agent & Distributed Information Security

ISSN 1751-8709

Unconditionally secure social secret sharing scheme

M. Nojoumian D.R. Stinson M. Grainger

David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
E-mail: mnojoumi@cs.uwaterloo.ca

Abstract: The authors introduce the notion of a 'social secret sharing scheme', in which shares are allocated based on a player's reputation and the way he/she interacts with other participants. During the social tuning phase, weights of players are adjusted such that participants who cooperate will end up with more shares than those who defect. Alternatively, newcomers are able to be enrolled in the scheme while corrupted players are disenrolled immediately. In other words, this scheme proactively renews shares at each cycle without changing the secret, and allows trusted participants to gain more authority. The motivation is that, in real-world applications, components of a secure scheme may have different levels of importance (i.e. the number of shares a player has) as well as reputation (i.e. cooperation with other players for the share renewal or secret recovery). Therefore a good construction should balance these two factors, respectively. In the proposed schemes, both the passive and active mobile adversaries are considered in an unconditionally secure setting.

1 Introduction

The growth of Internet has created amazing opportunities for 'secure multiparty computations', where various users, intelligent agents or computer servers cooperate in order to conduct computation tasks based on the private data they each provide [1]. Since these computations could be among untrusted participants or competitors, consequently, the privacy of each participant's input is an important factor.

As stated in the literature, a fundamental method used in secure multi-party computations is the 'secret sharing scheme' [2, 3], where a secret is divided into different shares for distribution among participants (private data), and a subset of participants then cooperate in order to reveal the secret (computation result). In particular, Shamir [2] proposed the '(t, n)-threshold secret sharing scheme', in which the secret is divided into n shares for distribution among players. The shares are constructed such that any t participants can combine their shares to reveal the secret, but any set of $t-1$ participants cannot learn anything about the secret.

Sample applications of such schemes are: 'joint signature or decryption', where a group of players sign documents or decrypt messages with the intention that only if all of them or a subset of participants cooperate, then a signature or a message can be generated [4], 'shared RSA keys', in which a number of players collaborate to jointly construct an RSA key [5], 'electronic auctions with private bids', where a group of agents perform sealed-bid electronic auctions while preserving the privacy of the submitted bids [6].

To construct a secure scheme, first the security model needs to be defined. We consider various types of adversaries. In the 'passive adversary' model, participants follow protocols correctly but are curious to learn the secret information. On the other hand, in the 'active adversary' model, players may deviate from protocols while trying to learn the secret data.

In addition, the passive or active adversary might be classified in a 'static' or 'mobile' setting. The former refers to the adversary who corrupts players ahead of time, whereas in the latter case, the adversary may corrupt different players at different stages of the protocols' executions. Finally, the entire security model might be 'computational', meaning that, the security of the protocols relies on the computational assumptions such as the

hardness of factoring, or 'unconditional', meaning that the adversary has unlimited computation power.

1.1 Motivation

Our motivation is that, in real-world applications, components of a secure system may have different 'levels of importance', that is, the number of shares a player has, as well as 'reputation', that is, cooperation with other players for the share renewal or secret recovery. Therefore a good construction should balance these two factors respectively, that is, adjusting the responsibility based on reliability. Assume a major shareholder has been attacked. If the scheme is not re-arranged the security cost would be severe. On the other hand, if a player with a small number of shares is working reliably for some period of time, it might be rational to assign him more shares.

Although our goal is to focus on the theoretical aspects of such a construction, we motivate the proposed scheme by the following scenario. Suppose shares of a secret have been distributed among various players based on their weights in a secure system, consequently, revealing the secret will trigger an action. The aim is to monitor participants' behaviours over time to regulate players' responsibility. As an example of this scenario, we can refer to real-time systems that are subject to operational deadlines. If a server provides his share in a single time slot when it is needed, he is classified as a cooperative player in that time period. Otherwise, he is not a reliable server.

1.2 Contribution

Our major contribution is the idea of a 'social secret sharing scheme', where shares are allocated based on each player's reputation and behaviours. In fact, shares are proactively renewed at each time period without changing the secret, while allowing the cooperative players to gain more authority.

Our scheme is called a 'social secret sharing scheme' since it can be visualised in terms of players collaborating to recover the secret in a social network, based on their reputations. This is similar to human social life where people share more secrets with whom they really trust and vice versa. In the literature, there exist dynamic schemes with different properties than our constructions, such as schemes in which one can activate various access structures [7], enroll or disenroll participants [8] or change the threshold [9].

As the main contributions, first, the formal definition and necessary conditions of a social secret sharing scheme is provided. Second, a scheme under the passive mobile adversary model is constructed, and the required techniques for the weight escalation/reduction based on the existing trust computation model [10] is proposed. In addition, a new tool, called the 'enrollment protocol', is developed for this primary construction. Third, the unconditionally secure proactive and verifiable secret sharing scheme in [11, 12] is generalised, in order to extend our approach to an active mobile adversary model, where each player has at most $m \geq 1$ shares rather than a single share.

Our constructions are dealer-free, unconditional and working under the passive/active mobile adversary models. In fact, it is quite challenging to design protocols in this setting. In other words, if one relaxes any of these assumptions, then he can decrease the computation and communication complexities. For instance, by using a trusted authority, or constructing the proposed scheme by relying on computational assumptions, or considering the simple passive adversary model without mobility.

1.3 Organisation

This paper is organised as follows. Section 2 provides some preliminaries. Section 3 creates a general picture of our social secret sharing scheme. Section 4 demonstrates the first construction under the passive mobile adversary model. Section 5 extends the first scheme to the active mobile adversary model. Finally, Section 6 contains concluding remarks.

2 Preliminaries

In the following discussions, secret sharing schemes and trust management are quickly reviewed in order to create the required foundations for our proposed social secret sharing scheme.

2.1 Secret sharing

As mentioned earlier, in a (t, n)-threshold secret sharing scheme, the secret is divided into n shares to be distributed among players. Consequently, the secret is reconstructed if at least t players cooperate with each other. On the other hand, any subsets of $t-1$ players cannot learn anything about the secret.

In a 'verifiable secret sharing scheme' [13], participants can verify that their shares are consistent with those of other participants. The authors in [14] present an unconditionally secure VSS when $t \leq n/3$. They only assume the existence of secure private channels between each pair of players. The proposed scheme in [15] uses the same communication model along with a broadcast channel to construct a new VSS when $t \leq n/2$. The former construction has a zero probability of error whereas the latter one has a negligible probability of error.

The authors in [16] illustrate the notion of the 'proactive secret sharing scheme', where the shares of players are updated without changing the secret. This solution is proposed for the mobile adversary model [17], where the adversary can infiltrate and gather the shares of an increasing number of participants over time in order to finally recover the secret.

To assign multiple shares rather than a single share to some players, the 'weighted secret sharing scheme' is introduced [18]. For instance, consider the scenario in which the president and chief executive of a company have the collective authority to open the safe deposit box of the company, but that any two vice-presidents can substitute for a missing party in their absence. In this scenario, the weighted scheme is used in order to prioritise different players.

2.2 Trust management

In the context of the social networks, 'trust' is the expectation that a player has about the future behaviour of another player based on the history of their interactions. On the other hand, 'reputation' is the perception that a player creates by past behaviours about his intentions. The former is a personal quantity while the latter is a social quantity [19]. A comprehensive survey of the existing trust and reputation systems are presented in [20].

Definition 1: We define $\mathcal{T}_i^j(p)$ as the trust value assigned by player P_j to P_i during period p, and $\mathcal{T}_i(p)$: $\mathbb{N} \rightarrow \mathbb{R}$ as the trust function, which represents the reputation of P_i

$$\mathcal{T}_i(p) = \frac{1}{n}\sum_{j=1}^{n} \mathcal{T}_i^j(p)$$

$$\text{where } -1 \leq \mathcal{T}_i(p) \leq +1 \quad \text{and} \quad \mathcal{T}_i(0) = 0$$

It is clear that if all players P_j have the same view (equal trust values for P_i), then $\mathcal{T}_i(p) = \mathcal{T}_i^j(p)$ for $1 \leq j \leq n$, which means, trust values are equal to the reputation value. Our social secret sharing scheme also requires a trust function with the same view for all players in order to distribute shares among participants. The construction of this function is out of the scope of this article, and it is independent of the proposed secret sharing scheme, meaning that, one can apply an arbitrary trust function. Therefore we use the trust management approach in [10], which illustrates and resolves the problem of the well-known solution in [21]. We quickly review this technique in order to give a flavour of the trust calculation in social networks.

The general idea in [10] is to support good players, discredit bad ones and create opportunities for newcomers whom we do not know much about their behaviours. As shown in Table 1, six possible actions and three sets $\mathcal{B}$, $\mathcal{N}$ and $\mathcal{G}$ are defined for bad, new and good players, respectively, where α, and β define boundaries on trust values for different sets of players.

This construction applies monotonically increasing and decreasing functions $\mu(x)$ and $\mu'(x)$ in the case of cooperation and defection to compute the trust function recursively, that is, computing $\mathcal{T}_i(p)$ by using $\mathcal{T}_i(p-1)$. This property leads to a fair trust computation

Table 1 Six possible actions for the trust management

Trust value	Cooperation: $P_i(\mathcal{C})$	Defection: $P_i(\mathcal{D})$
$P_i \in \mathcal{B} \Rightarrow \mathcal{T}_i(p) \in [-1, \beta)$	encourage	penalise
$P_i \in \mathcal{N} \Rightarrow \mathcal{T}_i(p) \in [\beta, \alpha]$	give a chance	take a chance
$P_i \in \mathcal{G} \Rightarrow \mathcal{T}_i(p) \in (\alpha, +1]$	reward	discourage

compared to [21].

$$P_i(\mathcal{C}) \Rightarrow \mathcal{T}_i(p) = \mathcal{T}_i(p-1) + \mu(x)$$

$$\mu(x) \in \begin{cases} [\eta, \theta), & P_i \in \mathcal{B} \\ \theta, & P_i \in \mathcal{N} \\ (\theta, \kappa], & P_i \in \mathcal{G} \end{cases}$$

$$P_i(\mathcal{D}) \Rightarrow \mathcal{T}_i(p) = \mathcal{T}_i(p-1) - \mu'(x)$$

$$\mu'(x) \in \begin{cases} (\theta, \kappa], & P_i \in \mathcal{B} \\ \theta, & P_i \in \mathcal{N} \\ [\eta, \theta), & P_i \in \mathcal{G} \end{cases}$$

For instance, by assigning $\eta = 0.01 < \theta = 0.05 < \kappa = 0.09$, we can simply define various points and construct an appropriate trust function via regression.

3 Social secret sharing scheme

The proposed model consists of n participants, $P_1, P_2, \ldots, P_n$, and a dealer who is available only during the initialisation phase. We assume the existence of private channels between each pair of participants (to be used during the share renewal step), and that the dealer can communicate privately with participants in the dealing stage. We also assume the existence of a synchronised broadcast channel, on which information is transmitted instantly and accurately to all participants. Let $\mathbb{Z}_q$ be a finite field and let ω be a primitive element in this field; all computations are performed in the field $\mathbb{Z}_q$.

Our intention is to construct unconditionally secure schemes, that is, schemes that do not rely on computational assumptions. We consider both the passive and active adversaries with mobility, that is, who are able to change the set of corrupted players from time to time during the execution of protocols. In the first construction, players correctly follow all protocols but are curious to learn the secret, whereas in the second one, players may deviate from the protocols.

In social secret sharing, each participant initially receives a constant number of shares. As time passes, players are assigned weights based on their behaviours in the scheme. Consequently, each participant receives a number of shares corresponding to his trust value, which is the representation of a player's reputation over time. In fact, weights of

IET Inf. Secur., 2010, Vol. 4, Iss. 4, pp. 202–211
doi: 10.1049/iet-ifs.2009.0098

participants are adjusted such that cooperative players receive more shares compared to non-cooperative ones. Alternatively, newcomers can join the scheme while corrupted players are disenrolled immediately. The reason for a corruption might be an active attack or a computational failure. Therefore the corrupted server is able to re-enroll in the scheme only after being fixed, and in that case, he is treated as a newcomer.

Example 1: We consider a matrix $\mathcal{M}_{n\times m}$ for the participants' identifiers, where n is the maximum number of participants and m is the maximum weight of any participant. As an example shown in Fig. 1, assume we have four participants with different weights. After some period of time, suppose we observe defection (e.g. not being available to send S_4) from the first participant and cooperation from the fourth player. In that case, the scheme decreases w_1 to 3 and increases w_4 to 2. That is, disenrollment of $i = 4$ and enrollment of $i = 14$ take place.

To further illustrate the proposed scheme, different possible behaviours are defined. After that, the required conditions are illustrated in order to ensure the scheme is working correctly. Finally, the formal definition of a social secret sharing scheme is presented.

Definition 2: Cooperation $P_i(\mathcal{C})$: P_i is available at the time of share renewal or secret recovery and sends correct information. Defection $P_i(\mathcal{D})$: P_i is not available at the required time or probably responds with delay. Corruption $P_i(\mathcal{X})$: P_i has been compromised by a passive or active adversary and may send incorrect information.

Definition 3: To recover the secret, the total weight of authorised players $\in \Delta$ (uncorrupted) must be equal or greater than the threshold, that is, $\sum_{P_i\in\Delta} w_i \geq t$. On the other hand, the total weight of colluders $\in \nabla$ (corrupted) must be less than the threshold, that is, $\sum_{P_i\in\nabla} w_i < t$. Finally, the weight of each player is bounded to a parameter much less than t, that is, $w_i \leq m \ll t$ for $1 \leq i \leq n$.

Definition 4: The social secret sharing scheme $\mathcal{S}^4$ is a three-tuple denoted as $\mathcal{S}^4(\mathcal{Sha}, \mathcal{Tun}, \mathcal{Rec})$ consisting of secret sharing, social tuning and secret recovery respectively. The only difference compared to the threshold scheme is the second stage, in which the weight of each player P_i is

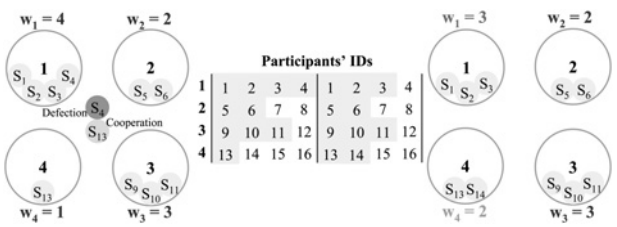

Figure 1 *Social secret sharing scheme*

adjusted based on the player's reputation $\mathcal{T}_i(p)$, after that, shares are updated accordingly.

4 Passive adversary model construction

In this construction, we consider the 'passive adversary model', where players follow all protocols but an unauthorised subset of them may collude to gather information and attempt to reconstruct the secret.

4.1 Secret sharing (Sha)

Suppose, the dealer initiates a secret-sharing scheme by generating a polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree $t-1$ in which its constant term is the secret $f(0) = \zeta$, that is, Shamir's scheme [2]. He sends shares of player P_i for $1 \leq i \leq n$ according to his weight w_i, and then he leaves the scheme

$$\varphi_{ij} = f(\vartheta_{ij}) \quad \text{for } 1 \leq j \leq w_i$$

where $\vartheta_{ij} = im - m + j$ and m is the maximum weight of any participant. The initial trust value is zero for all players. This trust value and, consequently, weights of participants are updated at each cycle during the share renewal stage based on players' behaviours.

4.2 Social tuning (Tun)

Our scheme provides a mechanism for assigning new weights to players based on their behaviours at the end of each time period, where by 'behaviour', we refer to a participant's reputation. We intend to apply a weight adjustment technique that supports reliable participants because of their repeated cooperation, reduces the influence of unreliable players because of their past defection, and protects the scheme from colluders.

In fact, the trust function illustrates how reputable or trustworthy each participant is. One simple solution is to assign an initial trust value to newcomers, increase this value by a constant factor if the participant is cooperating, and decrease it otherwise. However, this naive method does not consider various scenarios when making the adjustment. For that reason, we apply the proposed trust function in [10], as illustrated in Section 2.2.

Although we limited the weight of each player by m, this trust function $\mathcal{T}_i(p)$ also bounds the trust value, both above and below, so that a participant cannot continually build up the reputation in order to be the main shareholder and form a monopoly. In other words, it protects the scheme in a scenario where a malicious player cooperates for a while in order to gather most of the shares for a severe damage.

Furthermore, consider the scenario in which a player cooperates in the share renewal stage for several times

(cheap cooperations) until reaching a high trust value, at which point he may defect the secret recovery stage (an expensive defection) without significant effect on his reputation value. The authors in [10] define the parameter λ as the 'transaction cost'. In that case, the scheme would be able to fairly deal with the players' cooperation and defection.

Finally, since players' weights and consequently trust values are public information, therefore the trust computation and weight adjustment can be done by any authority or a committee of players on a public board. In the next, we illustrate how to increase and/or decrease the weight of different players consistently.

4.2.1 Inactivating non-cooperative players' shares: Now that we have a trust value for each participant, we turn to the task of using that value to adjust the scheme. Clearly, identifier j for $1 \leq j \leq m$ should be inactivated for each player whose trust value has been decreased and activated for players whose trust values are risen or for newcomers. The task is to determine how many id's should be inactivated for non-cooperative participants, and how other id's should be activated for cooperative players and new participants.

One option for the share removal is simply to disenroll a single id of a player each time his trust value decreases. However, such an approach does not take into account the total number of shares in the scheme, nor does it consider the number of shares each participant has. For instance, if the player has a large number of shares, inactivation of a single id has a negligible effect. On the other hand, a participant with only one share remaining is totally removed from the scheme. One particular approach is to inactivate a number of id's for each player P_i proportional to the amount that the player's reputation $\mathcal{T}_i(p)$ is decreased

$$P_i(\mathcal{D})\text{: defection} \Rightarrow w_i(p) = \left\lfloor w_i(p-1) \cdot \left(1 - \frac{\tau}{2}\right) \right\rfloor$$

where $\tau = \mathcal{T}_i(p-1) - \mathcal{T}_i(p) \geq 0$ is the coefficient of the weight reduction for the non-cooperative players. If $w_i(p)$ becomes zero, P_i is removed from the scheme, that is, the release of a row in $\mathcal{M}_{n\times m}$. Consequently, the total number of id's to be activated in the entire scheme is given as follows

$$\delta(p) = \sum_{i:P_i(\mathcal{D})} (w_i(p-1) - w_i(p))$$

Example 2: Suppose that trust values of a non-cooperative player P_i and a cooperative player P_j have been decreased from $\mathcal{T}_i(p-1) = -0.2$ to $\mathcal{T}_i(p) = -0.8$ and $\mathcal{T}_j(p-1) = 0.8$ to $\mathcal{T}_j(p) = 0.2$ accordingly. In that case, the weight reduction coefficient $\tau = 0.6$ would be the same because of the symmetric range of the trust function. In this example, P_j has done an expensive defection compared to the P_i so that they obtained the same reduction rate. Since $\mathcal{T}_i(p) \in [-1, +1]$ for every player, we divide τ by 2 in order to compute the rate of the weight reduction in the [0, 1] interval.

4.2.2 Activating cooperative players' shares: Given the number of id's to be activated, we now define which players should receive extra shares and how many newcomers can enter into the scheme. For each participant P_i, consider the ratio of a player's trust value $\mathcal{T}_i(p)$ to the number of shares he/she is holding $w_i(p)$. This ratio $\rho = \mathcal{T}_i(p)/w_i(p)$ increases with the participant's trust value enhancement, and decreases as the participant gains more shares.

As a result, it is reasonable to activate id's in participants for whom this ratio is highest, but this is not enough since we also need to consider newcomers whose trust values are zero. Therefore, to have a fair policy, we give the first priority to cooperative players for whom this ratio is both highest and positive, the second priority to newcomers and the third priority to other cooperative players with negative trust values. However, the conditions of Definition 3 must be satisfied in Fig. 2.

We assume that there are enough cooperative and/or new players P_i with $w_i(p) < t - 1$ to activate their id's, that is, $\delta(p) \leq |\mathcal{A}|$. The scenario in which there are no cooperative participants or newcomers to receive shares seems unlikely. However, our algorithm can easily be modified to handle this situation by assigning the remaining shares to non-cooperative participants who still have relatively high trust values; doing so maintains a constant number of shares in the scheme. By sorting the array $\mathcal{A}$ and assuming $|\mathcal{A}| \simeq n$, the complexity of the algorithm is $O(n + n \log n)$.

To add participants to the scheme, we have two options for assigning id's to new players in $\mathcal{M}_{n\times m}$. The first solution is to add a row for each new player in the matrix. As time passes, this approach leads to a big matrix with empty rows and consequently increases the size of identifiers. The second alternative is to use released rows of the disenrolled players. Since we first remove players from the scheme and then update shares of remaining participants, we can reuse the released id's and assign them to newcomers without leaking any information about the secret. In fact, new players receive updated shares corresponding to those recycled id's.

4.2.3 Share renewal: This stage consists of two phases. First, initial shares for newcomers or newly activated id's of existing players are generated. After that, players proactively update their shares, whereas disenrolled id's do not receive any more shares. As a result, old shares corresponding to those inactivated id's would be useless.

Phase-(I)

To update shares in a proactive scheme, a participant must have his previous shares. Suppose we intend to activate a new

doi: 10.1049/iet-ifs.2009.0098

Algorithm 1

collect cooperative & new players in an array $\mathcal{A}$

compute the trust-to-share ratio ρ for $P_i \in \mathcal{A}$

sort the array $\mathcal{A}$ based on the computed ratio ρ

$k := 0$ \\ assume $\delta(p) \leq |\mathcal{A}|$

for $j := 1$ to $|\mathcal{A}|$ **do**

  select player P_i from $\mathcal{A}[j]$

  if $0 < w_i(p) < t - 1$ **then**

    \\ known player

    activate a new *id* for P_i

    $w_i(p) := w_i(p-1) + 1$

    $k := k + 1$

  else if $w_i(p) := 0$ **then**

    \\ new player

    assign a row in $\mathcal{M}_{n\times m}$ to P_i

    activate a new *id* for P_i

    $w_i(p) := 1$

    $k := k + 1$

  end if

  if $k := \delta(p)$ **then**

    break the loop

  end if

end for

Figure 2 *Activation of players' id's*

id in period p while we do not have its corresponding share in period $p-1$. For the sake of simplicity, assume each participant has one identifier, in that case, this problem can be resolved only if t participants cooperate together in order to generate the old share for the newcomer, where t is the threshold.

The initial solution to this problem, named 'share recovery', was proposed in [16]. That solution is not efficient because of its random shuffling procedure. Saxena *et al.* [22] propose a non-interactive solution by using bivariate polynomials, named 'bivariate admission control', but this protocol is secure only under the discrete logarithm assumption. Our solution, called 'enrollment protocol', is an efficient new construction with unconditional security under the passive adversary model. We assume that this protocol is executed in a single time slot in our social secret sharing scheme.

We first show the Lagrange interpolation formula [23], and then present the enrollment protocol. Suppose q is a prime number and $x_1, x_2, \ldots, x_t$ are distinct elements in $\mathbb{Z}_q$. In addition, suppose $f_1, f_2, \ldots, f_t$ are elements in $\mathbb{Z}_q$. Then, there is a unique polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree at most $t-1$ such that $f(x_i) = f_i$ for $1 \leq i \leq t$

$$f(x) = \sum_{i=1}^{t} \left(\prod_{1\leq j\leq t, i\neq j} \frac{x - x_j}{x_i - x_j} \times f_i \right) \tag{1}$$

1. First, each player P_i for $1 \leq i \leq t$ computes his corresponding Lagrange interpolation constant

$$\gamma_i = \prod_{1\leq j\leq t, i\neq j} \frac{k-j}{i-j} \tag{2}$$

where i, j, k represent players' id's

2. After that, each participant P_i multiplies his share φ_i by his Lagrange interpolation constant, and randomly splits the result into t portions, that is, a row in a 'share-exchange matrix'

$$\varphi_i \times \gamma_i = \partial_{1i} + \partial_{2i} + \cdots + \partial_{ti} \quad \text{for } 1 \leq i \leq t \tag{3}$$

3. Players exchange ∂_{ji}'s accordingly through pairwise channels. Therefore each P_j holds t values, that is, a column in the share-exchange matrix. P_j adds them together and sends the result to P_k.

$$\sigma_j = \sum_{i=1}^{t} \partial_{ji} \quad \text{where } \partial_{ji} \text{ is the } j\text{th share-portion of the } i\text{th participant} \tag{4}$$

4. Finally, player P_k adds these values σ_j for $1 \leq j \leq t$ together to compute his share φ_k.

$$\varphi_k = \sum_{j=1}^{t} \sigma_j \tag{5}$$

Example 3: Assume $t = 3$ and the dealer has generated shares of three players P_1, P_2 and P_3 based on $f(x) = 9 + 2x + 5x^2 \in \mathbb{Z}_{13}[x]$, that is, $\varphi_1 = 3$, $\varphi_2 = 7$ and $\varphi_3 = 8$. After some time, players are asked to create a share for a newcomer (for instance P_4) in the absence of the dealer. First each player P_i privately computes $\varphi_i \times \gamma_i$ as follows: $\varphi_1 \times \gamma_1 = 3 \times (4-2)(4-3)/(1-2)(1-3) = 3$, $\varphi_2 \times \gamma_2 = 7 \times (4-1)(4-3)/(2-1)(2-3) = 5$ and $\varphi_3 \times \gamma_3 = 8 \times (4-1)(4-2)/(3-1)(3-2) = 11$. After that, they randomly split the results and exchange them, as shown in the share-exchange matrix $\mathcal{E}_{t\times t}$. Players then

compute and send $\sigma_1 = 7$, $\sigma_2 = 4$ and $\sigma_3 = 8$ to P_4. Finally, he adds up these values to compute his share $\varphi_4 = 6$.

$$\mathcal{E}_{t\times t} = \begin{pmatrix} \partial_{11} = 1 & \partial_{21} = 1 & \partial_{31} = 1 \\ \partial_{12} = 2 & \partial_{22} = 1 & \partial_{32} = 2 \\ \partial_{13} = 4 & \partial_{23} = 2 & \partial_{33} = 5 \end{pmatrix}$$

Theorem 1: The presented enrollment protocol is correct and unconditionally secure under the passive adversary model.

Proof: We first show the protocol is correct and then prove its unconditional security. The following computation illustrates that the new value is in fact the P_k's share on $f(x)$, that is, the correctness.

$$\begin{aligned} \varphi_k &= \sum_{j=1}^{t} \sigma_j \quad \text{by (5)} \\ &= \sum_{j=1}^{t}\sum_{i=1}^{t} \partial_{ji} = \sum_{i=1}^{t}\sum_{j=1}^{t} \partial_{ji} \quad \text{by (4)} \\ &= \sum_{i=1}^{t} (\varphi_i \times \gamma_i) \quad \text{by (3)} \\ &= \sum_{i=1}^{t} \left(\varphi_i \times \prod_{1\le j\le t, i\ne j} \frac{k-j}{i-j} \right) \quad \text{by (2)} \\ &= f(k) \quad \text{by (1)} \end{aligned}$$

As shown in the enrollment protocol, each P_i first multiplies his share φ_i in the corresponding Lagrange interpolation constant γ_i, and then splits the result into t pieces. We defined the share-exchange matrix $\mathcal{E}_{t\times t}$, where each row shows various fractions of a single share and each column represents portions of different shares that each player receives. In other words, all values in ith row, that is, $\partial_{1i}, \partial_{2i}, \ldots, \partial_{ti}$, belongs to a single player P_i and all entries in jth column, that is, $\partial_{j1}, \partial_{j2}, \ldots, \partial_{jt}$, represent values that player P_j receives from other participants

$$\mathcal{E}_{t\times t} = \begin{pmatrix} \partial_{11} & \partial_{21} & \cdots & \partial_{t1} \\ \partial_{12} & \partial_{22} & \cdots & \partial_{t2} \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{1t} & \partial_{2t} & \cdots & \partial_{tt} \end{pmatrix}$$

We consider the following two scenarios to see if a coalition of $t-1$ participants can determine any information regarding the secret.

First, suppose $t-1$ of t cooperating participants collude. In this case, colluders have access to all entries of $t-1$ rows. In addition, they also know $t-1$ entries of the single unknown row because $t-1$ columns belong to them. Therefore just one entry remains unknown, which prevents colluders to find the newcomer's share and consequently the secret (as presented in Example 3, if P_1 and P_2 collude, $\partial_{33} = 5$ in the third row remains unknown).

Second, suppose $t-2$ of t cooperating participants plus the newcomer collude. In this case, colluders have access to all entries of $t-2$ rows, in addition, they also know $t-2$ entries of two unknown rows because $t-2$ columns belong to them. Therefore four entries remain unknown. On the other hand, the newcomer also knows the summation of column's entries for all columns, as a consequence, he/she can just construct two equations with four unknowns that does not reveal any information about the secret (as presented in Example 3, if P_1 and the newcomer P_4 collude, $\partial_{22} = 1$ and $\partial_{32} = 2$ in the second row and $\partial_{23} = 2$ and $\partial_{33} = 5$ in the third row remain unknown and P_4 can only construct the following two equations: $1 + \partial_{22} + \partial_{23} = 4$ and $1 + \partial_{32} + \partial_{33} = 8$). □

Phase-(II)

1. To update shares, each player P_u generates a random polynomial $g^u(x) \in \mathbb{Z}_q[x]$ of degree $t-1$ with a zero constant term.

2. Player P_u then sends w_i shares to P_i for $1 \le i \le n$, as shown below, where $\vartheta_{ij} = im - m + j$ and m is the maximum weight of any participant

$$\psi_{ij}^u = g^u(\vartheta_{ij}) \quad \text{for } 1 \le j \le w_i$$

3. Finally, each player P_i updates his share by adding up the auxiliary shares ψ_{ij}^u to his share φ_{ij}

$$\varphi_{ij} = \varphi_{ij} + \sum_{u=1}^{n} \psi_{ij}^u \quad \text{for } 1 \le j \le w_i$$

Since the constant terms of $g^u(x)$-s are zero, the secret ζ remains the same and shares of players are updated in order to overcome the mobile adversary [16]. As we mentioned, inactivated id's do not receive any shares at this stage, that is, they are disenrolled.

4.3 Secret recovery ($\mathcal{R}ec$)

As stated earlier, authorised players are able to recover the secret if their total weight is equal or greater than the threshold, that is, $\sum_{P_i\in\Delta} w_i \ge t$. In this case, players $P_i \in \Delta$ send their shares φ_{ij} for $1 \le j \le w_i$ to a selected participant to reconstruct $f(x)$ by Lagrange interpolation, consequently, the secret $f(0) = \zeta$ is recovered.

Theorem 2: The social secret sharing scheme $\mathcal{S}^4(\mathcal{S}ha, \mathcal{T}un, \mathcal{R}ec)$ presented in Section 4 is unconditionally secure under the passive mobile adversary model.

Proof: The security of $\mathcal{S}ha$ and $\mathcal{R}ec$ is the same as the security of the Shamir's secret sharing scheme [2]. The security of the $\mathcal{T}un$ depends on the share renewal step. The first phase is secure as shown in Theorem 1. The second phase is also proven to be secure as illustrated in [16].

doi: 10.1049/iet-ifs.2009.0098

5 Active adversary model construction

In this section, we consider the active adversary model, where players may deviate from protocols or collude to reconstruct the secret. We review the verifiable and proactive secret sharing scheme, first proposed in [11] (a flaw in the scheme was fixed in [12]). We modify those protocols accordingly to fit them to our social secret sharing scheme.

First of all, the pairwise check is changed since each participant has multiple shares rather than a single share. Second, the 'recovery' protocol is used to generate new shares for newly activated id's. In the original paper, this protocol is used to reboot corrupted servers after running a detection procedure. Third, the accusation process in the second phase of the share renewal stage is changed to make it applicable for a weighted scheme. More precisely, our construction is a generalisation of the proposed scheme in [11, 12], where, in our scheme each player has at most $m \geq 1$ shares while in the initial construction $m = 1$.

We assume that the share renewal step is instantaneous, therefore the adversary cannot corrupt more participants while shares are being updated. A corrupted participant may send incorrect data to other players, both through the public broadcast channel and/or private channels. The modified scheme is secure against $|\nabla|$ colluders who have a total weight of ξ under the following assumption

$$\xi = \sum_{P_i \in \nabla} w_i : \begin{cases} \xi + 1 < t \\ \xi \leq \dfrac{\sum_{i=1}^{n} w_i}{4} - 1 \end{cases}$$

As stated earlier, we consider the matrix $\mathcal{M}_{n\times m}$ for the participants' identifiers, where n is the maximum number of players and m is the maximum weight of any player. We also define the variable $\vartheta_{ij} = im - m + j$ to generate the entries of matrix $\mathcal{M}_{n\times m}$ in the following protocols.

5.1 Secret sharing (Sha)

The following protocol is initiated by a dealer who is not required to participate in the scheme after the share distribution:

1. Dealer chooses a random symmetric polynomial $f(x, y) \in \mathbb{Z}_q[x, y]$ of degree $t - 1$ in order to send shares $\varphi_{ij}(x) = f(x, \omega^{\vartheta_{ij}})$ to P_i for $1 \leq j \leq w_i$ through a private channel, where $a_{00} = \zeta$

$$f(x, y) = \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} a_{rs} x^r y^s \quad \text{and} \quad \forall r, s: a_{rs} = a_{sr}$$

2. To verify distributed shares, players P_i and P_j perform the pairwise checks through secure channels as shown below, where $1 \leq k \leq w_i$ and $1 \leq l \leq w_j$

$$\varphi_{ik}(\omega^{\vartheta_{jl}}) \stackrel{?}{=} \varphi_{jl}(\omega^{\vartheta_{ik}})$$

3. Consequently, if a player P_i finds that the above equation does not hold while checking it with P_j, he/she then broadcasts (i, j), meaning that, P_i is accusing P_j.

4. Each player P_i computes a subset $\Gamma \subseteq \{1, \ldots, n\}$ such that any ordered pair $(i, j) \in \Gamma \times \Gamma$ is not broadcasted. If $|\Gamma| \geq n - |\nabla|$, then P_i outputs $\text{ver}_i = 1$, otherwise, P_i outputs $\text{ver}_i = 0$.

The dealer erases all the data on his/her end if at least $n - |\nabla|$ players output $\text{ver}_i = 1$, otherwise, he/she reboots the system for another initialisation. It is worth mentioning that, this scheme can tolerate a dishonest dealer. Moreover, a corrupted player may act honestly during *Sha* because of a future harmful plan, therefore Γ consists of uncorrupted players and possibly malicious players who act honestly during the initialisation.

5.2 Social tuning (Tun)

This section is similar to its counterpart in the passive adversary model construction (Section 4). The only difference is the share renewal stage.

Share renewal. This stage consists of two phases. In the first one, initial shares for newcomers or newly activated id's of existing players are generated, that is, they are enrolled. Then, in the second phase, players proactively update their shares, while disenrolled id's do not receive any updates.

Phase-(I)

1. Each player P_i where $i \in \Gamma$ sends $\varphi_{ik}(\omega^{\vartheta_{jl}})$ for $1 \leq k \leq w_i$ to P_j in order to generate his lth shares, that is, $\varphi_{jl}(x)$.

2. After that, player P_j computes a polynomial $\varphi_{jl}(x)$ such that $\varphi_{jl}(\omega^{\vartheta_{ik}}) = \varphi_{ik}(\omega^{\vartheta_{jl}})$ for at least $n - 2|\nabla|$ values of i.

In fact, share $\varphi_{jl}(x)$ is constructed through the interpolation of pairs $(\omega^{\vartheta_{ik}}, \varphi_{ik}(\omega^{\vartheta_{jl}}))$ in the second step. We explain the main reason behind the condition $n - 2|\nabla|$ in Section 5.3.

Phase-(II)

1. To update shares, each player P_u where $u \in \Gamma$ generates a random symmetric polynomial $g^u(x, y) \in \mathbb{Z}_q[x, y]$ of degree $t - 2$ with a zero constant term, that is, $a_{00} = 0$

$$g^u(x, y) = \sum_{r=0}^{t-2} \sum_{s=0}^{t-2} a_{rs} x^r y^s \quad \text{and} \quad \forall r, s: a_{rs} = a_{sr}$$

2. Each P_u sends $\psi^u_{ik}(x) = g^u(x, \omega^{\vartheta_{ik}})$ by a private channel to P_i for $1 \leq k \leq w_i$. As a result, for every single share $\varphi_{ic}(x)$ of

degree $t-1$, P_i has many auxiliary shares $\psi_{ic}^{u}(x)$ of degree $t-2$.

3. To verify the shares distributed by P_u, each pair of players P_i and P_j perform the pairwise checks through secure channels, where $1 \leq k \leq w_i$, $1 \leq l \leq w_j$ and both $i, j \neq u$

$$\psi_{ik}^{u}(\omega^{\vartheta_{jl}}) \stackrel{?}{=} \psi_{jl}^{u}(\omega^{\vartheta_{ik}})$$

4. If player P_i, for instance, finds that $\exists c \in [1, w_i]$ such that $\psi_{ic}^{u}(\omega^{\vartheta_{jl}}) \neq \psi_{jl}^{u}(\omega^{\vartheta_{ic}})$ for 'more than' $|\nabla|$ values of j, he/she then broadcasts an accusation of P_u.

5. If P_u is accused by 'at most' $|\nabla|$ players, accusations could be from colluders. In this case, player P_u can defend himself by broadcasting the shares that he generated for those accusers

$$\psi_{ic}^{u}(x) = g^{u}(x, \omega^{\vartheta_{ic}}) \quad \text{where } i \in \text{accusers'id's}$$

6. Then, other players P_j, excluding the conflicting parties P_u and P_i, check $\psi_{ic}^{u}(\omega^{\vartheta_{jl}}) \stackrel{?}{=} \psi_{jl}^{u}(\omega^{\vartheta_{ic}})$ and broadcast yes or no. If, for every broadcasted $\psi_{ic}^{u}(x)$, at least $|\Gamma| - |\nabla| - 1$ players broadcast yes, then P_u is not malicious. In this case, if P_i has a share $\psi_{ic}^{u}(x)$ different from the one that P_u has broadcasted, he stores the broadcasted one.

7. Finally, each participant P_i first updates the list Γ of good players who are not found guilty in the previous step, and then updates his/her shares for $1 \leq k \leq w_i$ as follows

$$\varphi_{ik}(x) = \varphi_{ik}(x) + (x + \omega^{\vartheta_{ik}}) \sum_{u \in \Gamma} \psi_{ik}^{u}(x)$$

5.3 Secret recovery ($\mathcal{R}ec$)

Players are able to recover the secret ζ at any time by performing the following recovery protocol:

1. Each player P_i where $i \in \Gamma$ sends $\varphi_{ik}(0)$ for $1 \leq k \leq w_i$ to a selected participant P_j, that is, the constant terms of shares.

2. After that, the selected player P_j computes a polynomial $f'(0, y)$ such that $f'(0, \omega^{\vartheta_{ik}}) = \varphi_{ik}(0)$ for at least $n - 2|\nabla|$ values of i.

3. In fact, $f'(0, y)$ is part of the original symmetric polynomial $f(x, y)$, therefore the selected P_j computes the secret $\zeta = f'(0, 0)$.

As we mentioned, the scheme itself can tolerate $|\nabla|$ dishonest players. In addition, a dishonest dealer may cheat on $|\nabla|$ of honest players during $\mathcal{S}ha$ in order to eliminate them from the scheme. As a result, the set Γ of good players has at least $n - 2|\nabla|$ members. Therefore an error correction technique, such as the one proposed in [24], can be used to find the maximum consistent set of shares for the interpolation of $f'(0, y)$.

Theorem 3: The social secret sharing scheme $\mathcal{S}^4(\mathcal{S}ha, \mathcal{T}un, \mathcal{R}ec)$ presented in Section 5 is unconditionally secure under the active mobile adversary model.

Proof: The security proofs of the modified protocols are the same as the ones presented in [11, 12]. □

6 Conclusion

We introduced the notion of a social secret sharing scheme, in which a player's weight are adjusted based on his reputation and behaviours over time. We demonstrated two constructions based on the passive and active mobile adversary models.

The proposed construction has a variety of desirable properties: it is unconditionally secure, meaning that it does not rely on any computational assumptions; proactive, refreshing shares at each cycle without changing the secret; dynamic, allowing changes to the access structure after the initialisation; weighted, allowing the cooperative players to gain more authority in the scheme; and verifiable in the case of the active adversary model.

In addition, the proposed scheme gradually reduces the influence of unreliable participants because of the self-reinforcement property of social interactions among players. In other words, players collaborate with those whom they really trust; conversely, they tend not to cooperate with those whom they do not trust. This issue creates an increasing gap between reliable and unreliable players unless a participant undergoes a sustained change in his behaviour. Applications of such a paradigm are: electronic auctions with private bids running by intelligent agents, joint signature and shared decryption keys.

7 Acknowledgment

We would like to thank the anonymous reviewers for their helpful and constructive comments.

8 References

[1] DU W., ATALLAH M.J.: 'Secure multi-party computation problems and their applications: a review and open problems'. Proc. Workshop on New Security Paradigms, NSPW'01, 2001, pp. 13–22

[2] SHAMIR A.: 'How to share a secret', *Commun. ACM*, 1979, **22**, (11), pp. 612–613

[3] BLAKLEY G.R.: 'Safeguarding cryptographic keys'. National Computer Conference, New York, Montvale, NJ, USA, 1979, vol. 48 of AFIPS Conf. Proc., pp. 313–317

doi: 10.1049/iet-ifs.2009.0098

[4] GOLDWASSER S.: 'Multi party computations: past and present'. Proc. 16th Ann. ACM Symp. on Principles of Distributed Computing, PODC'97, 1997, pp. 1–6

[5] BONEH D., FRANKLIN M.: 'Efficient generation of shared rsa keys', *J. ACM*, 2001, **48**, (4), pp. 702–722

[6] HARKAVY M., TYGAR J.D., KIKUCHI H.: 'Electronic auctions with private bids'. Proc. Third Conf. on USENIX Workshop on Electronic Commerce, WOEC'98, 1998, pp. 61–74

[7] BLUNDO C., CRESTI A., SANTIS A.D., VACCARO U.: 'Fully dynamic secret sharing schemes', *Theoret. Comput. Sci.*, 1996, **165**, (2), pp. 407–440

[8] ZHANG Y., LIU Z.: 'Dynamic and verifiable secret sharing among weighted participants', *J. Syst. Sci. Complexity*, 2007, **20**, (4), pp. 481–485

[9] TARTARY C., WANG H.: 'Dynamic threshold and cheater resistance for shamir secret sharing scheme', in LIPMAA H., YUNG M., LIN D. (EDS.): 'Inscrypt' (Springer, 2006), (*LNCS*, **4318**), pp. 103–117

[10] NOJOUMIAN M., LETHBRIDGE T.: 'A new approach for the trust calculation in social networks'. E-business and Telecommunication Networks: Third Int. Conf. on E-Business, Selected Papers, 2008, vol. 9, pp. 64–77

[11] STINSON D.R., WEI R.: 'Unconditionally secure proactive secret sharing scheme with combinatorial structures', in HEYS H.M., ADAMS C.M. (EDS.): 'Selected areas in cryptography', (Springer, 1999), (*LNCS*, **1758**), pp. 200–214

[12] D'ARCO P., STINSON D.R.: 'On unconditionally secure robust distributed key distribution centers'. Proc. Advances in Cryptology, ASIACRYPT'02, 2002, (*LNCS*), pp. 346–363

[13] CHOR B., GOLDWASSER S., MICALI S., AWERBUCH B.: 'Verifiable secret sharing and achieving simultaneity in the presence of faults'. FOCS, 1985, pp. 383–395

[14] BEN-OR M., GOLDWASSER S., WIGDERSON A.: 'Completeness theorems for non-cryptographic fault-tolerant distributed computation'. STOC, 1988, pp. 1–10

[15] RABIN T., BEN-OR M.: 'Verifiable secret sharing and multiparty protocols with honest majority'. STOC, 1989, pp. 73–85

[16] HERZBERG A., JARECKI S., KRAWCZYK H., YUNG M.: 'Proactive secret sharing or: How to cope with perpetual leakage'. CRYPTO, 1995, (*LNCS*, **963**), pp. 339–352

[17] OSTROVSKY R., YUNG M.: 'How to withstand mobile virus attacks (extended abstract)'. Proc. 10th Ann. ACM Symp. on Principles of Distributed Computing, PODC'91, 1991, pp. 51–59

[18] BENALOH J.C., LEICHTER J.: 'Generalized secret sharing and monotone functions'. CRYPTO, 1988, (*LNCS*, **403**), pp. 27–35

[19] MUI L., MOHTASHEMI M., HALBERSTADT A.: 'A computational model of trust and reputation for e-businesses'. HICSS, 2002, pp. 2431–2439

[20] JØSANG A., ISMAIL R., BOYD C.: 'A survey of trust and reputation systems for online service provision', *Decis. Support Syst.*, 2007, **43**, (2), pp. 618–644

[21] YU B., SINGH M.P.: 'A social mechanism of reputation management in electronic communities', in KLUSCH M., KERSCHBERG L. (EDS.): 'CIA' (Springer, 2000), (*LNCS*, **1860**), pp. 154–165

[22] SAXENA N., TSUDIK G., YI J.H.: 'Efficient node admission for short-lived mobile ad hoc networks'. Proc. 13th IEEE Int. Conf. on Network Protocols, 2005, pp. 269–278

[23] STINSON D.R.: 'Cryptography: theory and practice' (CRC Press, 2005, 3rd edn.)

[24] REES R.S., STINSON D.R., WEI R., VAN REES G.H.J.: 'An application of covering designs: determining the maximum consistent set of shares in a threshold scheme', *Ars. Comb.*, 1999, **53**, pp. 225–237