J Ambient Intell Human Comput (2015) 6:807–824
DOI 10.1007/s12652-015-0299-2

ORIGINAL RESEARCH

A brief survey on secure multi-party computing in the presence of rational parties

Yilei Wang[1,2] · Tao Li[1] · Hairong Qin[1] · Jin Li[3] · Wei Gao[4] · Zhe Liu[5] · Qiuliang Xu[6]

Received: 14 May 2015 / Accepted: 11 June 2015 / Published online: 3 July 2015

Abstract Intelligent agents (IA) are autonomous entities which observe through sensors and act upon an environment using actuators to adjust their activities towards achieving certain goals. The architectures of agents have enormous potentials when they are applied into critical systems, where agents choose actions between working with their own and cooperating with others. Rational utility-based agents choose actions to maximize their expected utilities. Rational secure multi-party computing (RSMPC) means secure multi-party computing (SMPC) in the presence of rational utility-based agents. Here, we call them rational parties. In this paper certain goals of rational parties are maximizing their utilities. The introduction of rational parties considers the incentives in executing protocols. The security definitions under rational framework can better demonstrate the executing environment of real protocols. Furthermore, rational two-party computing removes some impossibility in traditional two-party computing, such as fairness. This paper represents the research status of RSMPC and some typical protocols. The advantages and disadvantages of previous rational SMPC protocols are discussed here. As an emerging field, there are still lots of open problems in RSMPC, such as communication channels, utility assumptions and equilibrium notions etc.

Keywords Game Theory · Nash equilibrium · Intelligent agents · Rational secret sharing · Rational secure multi-party computing

1 Introduction

Intelligent computing is an experience-based thoughtful program, which is a branch of artificial intelligent system. Intelligent computing can handle problems in critical systems with independent thinking ability. Intelligent agents are widely used in agent technology, which is a form of artificial intelligence. One basic aim for artificial intelligence is to help social communities like parties in social

✉ Yilei Wang
wang_yilei2000@163.com

Tao Li
litao_888@sina.com

Hairong Qin
qhr6113@163.com

Jin Li
jinli71@gmail.com

Wei Gao
mygaowei@163.com

Zhe Liu
zhe.liu@uni.lu

Qiuliang Xu
xql@sdu.edu.cn

[1] School of Information and Electrical Engineering, Ludong University, Yantai, China

[2] Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, China

[3] School of Computer Science and Educational Software, Guangzhou University, Guangzhou, China

[4] School of Mathematics and Statistics Science, Ludong University, Yantai, China

[5] Laboratory of Algorithmics, Cryptology and Security, University of Luxembourg, Walferdange, Luxembourg

[6] School of Computer Science and Technology, Shandong University, Jinan, China

networks, more adaptive to the changes in their environments. Intelligent agents are often described schematically as a functional system similar to computer programs. Some IA definitions stress on their autonomy and others on goal-directed behaviours (Faiyaz et al. 2014; Amato et al. 2014). Russell et al. (1995) divide agents into five classes according to their degree of perceived intelligence and capability: simple reflex agents, model-based reflex agents, goal-based agents, learning agents (Andreu and Angelov 2013; García et al. 2012) and utility-based agents. Utility-based agents differentiate goal states from non-goal states. Therefore, we should define a measure to show the desirable level of a particular state. This measure can be obtained through a utility function mapping a state to a measure of the utility of the state. Utility-based agents choose their actions to maximize their expected utilities. We borrow the notion of rational parties from economics to present the properties of utility-based agents. In the following of this paper, rational parties can be considered as utility-based agents if there is no special statement.

Multi-agent systems concern the interaction among several utility-based agents (rational parties) including those in distributed problem solving, distributed constraint optimization and multi-agent learning etc. (García et al. 2012). SMPC in the presence of rational parties is such a multi-agent system that can solve security problems in cloud computing, which is based on utility and consumption of computer resources (Chen et al. 2012a; Li et al. 2014; Castiglione et al. 2015; Xu et al. 2014; Ficco et al. 2014). Yao (1982) proposed the problem of millionaires, which introduced the pioneering theory for secure two-party computing (STPC). Goldriech et al. (1987) extended STPC to SMPC. The main task of SMPC is to guarantee security of the computation in the presence of various external attacks. Generally, these attacks to the computation are implemented by a subset of all the parties which are controlled by an external adversary. Parties in the subset are called corrupted parties and others outside the subset honest parties. Once parties are corrupted, they will be controlled by the adversary and execute the programs coming from the adversary. Adversaries are divided into three categories according to their abilities:

1. Semi-honest adversaries: they always abide by the protocol, but try to deduce more information than the protocol allowed about other parties through their intermediate views.
2. Malicious adversaries: they can corrupt other parties and obtain all the information of the corrupted parties, who execute the protocol following the arbitrary programs made by the malicious adversaries, such as premature abortion, false inputs and denial of execution etc.
3. Covert adversary: their abilities are between those of semi-honest and malicious adversaries. They have incentives to cheat honest parties. However, they can only successfully cheat with probability ϵ and fail to cheat with probability $1-\epsilon$, where ϵ is called detect factor (Aumann and Lindell 2007).

The security notions in SMPC consist of the following aspects.

1. Privacy: one party can only obtain his own output and the intermediate values deduced from the views. For example, in the electronic voting (Chen et al. 2011), the only output for parties is who wins in the vote. Beyond that, they can get no more information about other parties' inputs from the vote.
2. Correctness: each party can get correct output. For example, the winners in the vote must be the one who get most votes.
3. Independence of inputs: the corrupted parties must independently choose their inputs such that the inputs are independent of those from honest parties. For example, in electronic tendering system, the bid of each party is secret and it can not depend on other bids. Otherwise, malicious adversary can always win the bidding by adding a small value.
4. Fairness and guarantee of output delivery: adversaries can not get the inputs of honest parties by prematurely abortion. That is, adversaries receive the outputs if and only if honest parties receive the same outputs. For example, in electronic voting and tendering systems, the outputs for adversaries and honest parties are identical (Chen et al. 2012b).

The security definitions are realized by Ideal/Real paradigm and the basic idea is as follows. Considering an ideal world, where existing a trusted third party (TTP). Each party sends his inputs to TTP, who computes certain function using the inputs and then sends back the results to each party. Note that the adversary can arbitrarily choose the inputs for the corrupted parties. In the ideal world, the only information each party received from TTP is his output. Therefore the property of privacy is guaranteed. Furthermore, since TTP is honest and always computes correctly, the property of correctness is guaranteed. In the ideal world the only thing that adversary can do is to replace the inputs for corrupted parties. In the real world, no TTP exists, so the computation must be completed by interactions among parties. If no adversary in the real world have more abilities than the adversaries in the ideal world, the protocol is secure. In other words, security requires that the protocol can resist any attacks in the ideal world. In the real world, the attacks by the adversaries are identical to those in the ideal world. Therefore, if a protocol is secure in the ideal world, it is also secure in the real world.

Although the Ideal/Real paradigm can provide a security definition, it is criticized for being too pessimistic. The ability with which the malicious adversaries are endowed are too strong such that it is hard to find such adversaries in real world. In fact, most adversaries attack the protocol for some incentives instead of simply break the protocol. However, the descriptions of malicious adversaries neglect the incentives. In other words, the existence of malicious adversaries requires the consideration of the worst scenario or the maximized ability for adversaries. Nevertheless, such adversaries may not exist in real world. So the security discussion in the presence of such adversaries seems meaningless. In fact, adversaries participating in the protocol should be assumed for certain incentives instead of for nothing. For example, the incentives for the adversary may be to let someone win the vote in electronic voting instead of to simply prevent the vote. In byzantine protocols, the adversary may hope to reach an agreement of retreat instead of attack. In the execution of protocols, semi-honest or malicious adversaries can not be present with certain incentives. So, defining new types of adversaries according to the incentives has practical significance.

Parties are assumed to participate in protocols with certain incentives or for some profits. For example, someone promises to offer the adversary some favorable policies if he wins the vote. Thus adversary has incentives to make the one win the vote. In Byzantine protocols, the agreement of retreat minimizes the cost for the enemy, so the adversary tries to lead to the retreat agreement. These profits can be described by utility in game theory. The main goal for adversaries is to maximize their utilities. To realize the goal, the adversary may adopt some strategies. Meanwhile, parties who are not controlled by the adversary are no longer assumed to be honest parties. From RMPSC point of view, they also have certain incentives, which is to prevent adversary from attacking the protocol. For example, in electronic voting, parties who are not corrupted hope all parties correctly execute the protocol. They hope that the one who gets the most votes wins. Meanwhile they try to guarantee fairness for voting by preventing some party win the vote by collusion with a subset of all the parties. Therefore, these uncorrupted parties are also motivated by profits. In electronic voting, the profits are fairness such that the one who gets most votes wins. These profits can also be described by utility. Furthermore, they can also adopt some strategies to withstand adversaries. Thus, the protocol designers must consider such incentives when they construct a protocol. The previous notions of honest parties and adversaries are not fit for the scenarios where parties participate in the protocol with certain incentives. Therefore, new models in the presence of these new parties should be proposed. Furthermore, security notions under such scenarios should also be redefined. Since the incentives for parties in cryptography can borrow the corresponding notions in game theory (Esposito et al. 2015), the basic idea of RSMPC is to describe the incentives under the frame of game theory, combine them with utility and discuss the security notions in the presence of such parties. Both game theory and SMPC discuss some distrustful parties hope to complete certain tasks by interacting with each other.

RSMPC which is a combination of game theory and SMPC, solves some problems in SMPC by utilizing some notions in game theory. Figure 1 shows the relationships of RSMPC with game theory and SMPC. In RSMPC, honest parties and corrupted parties are collectively called rational parties, who have respective utilities and strategies. They interact with each other to maximize their utilities. The execution of the protocol can be considered as a process where rational parties take several strategies. Rational parties can choose to abide by the protocol or deviate from the protocol. The choices depend on whether deviation can bring better utilities. The final goal of protocol design is: each party abides by the protocol such that the protocol can be securely executed and all parties achieve optimal utilities. This goal explained in the game theory is: each party abiding by the protocol is Nash equilibrium and no one can get higher utility by deviating from the protocol.

The seminal notion of rational parties is proposed by Halpern and Teague (2004) when they study rational secret sharing (RSS) and multi-party computing. Halpern and Teague mainly discuss the scenario where rational parties are introduced into Shamir secret sharing (Shamir 1979). They think that parties neither honestly abide by the protocol nor arbitrarily destroy the protocol, instead they are driving by utility. If the action of sending shares to others is regarded as cooperating with others and not sending shares is regarded as defecting from others, then RSS is similar to the prisoner's dilemma (PD). Therefore, the utility definitions of rational parties are similar to that of parties in game theory. Consequently, the results are inherited from PD game. That is, rational parties have no incentives sending shares to others just as the results in game theory that parties have no incentives to cooperate with others since cooperating is dominated by defect.

As mentioned above, rational parties have goals in protocols. Towards the view of RSS, the goals of rational parties can be described as follows. We also call the goals as assumptions for utilities.

1. Selfishness: every party hopes to learn the secret;
2. Exclusivity: if he can not learn the secret, he hopes that other parties will not learn the secret either; if he learns the secret, he hopes that less parties will learn the secret.

Intuitively no parties have incentives sending shares to others such that they may learn the secret while he has a risk of not learning the secret, e.g. others do not send shares to

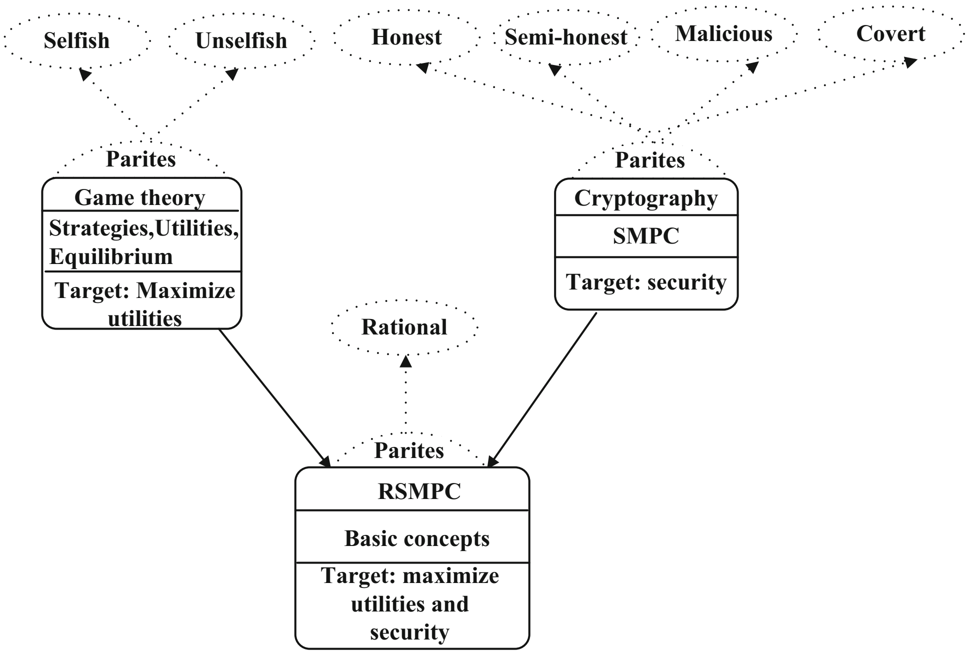

Fig. 1 The relationship of RSMPC between game theory and SMPC

him. Rational parties are assumed to decide whether sending shares simultaneously and the scenario where only one party deviates from the protocol while others all abide by the protocol is considered. More specifically, considering an m-out-of-n Shamir secret sharing scheme in the presence of rational parties, where m denotes the threshold of the scheme and n is the number of parties. Suppose that one party deviates from the scheme (say P_1), there are two cases.

1. If P_1 does not send shares to other $m - 1$ parties who send their shares to P_1, then P_1 learns the secret while others can not. In this case, since P_1 can learn the secret, he has no incentives sending his shares to others according to the selfishness assumption.
2. If P_1 sends shares to other $m - 1$ parties who do not send their shares to P_1, then P_1 does not learn the secret. In this case, P_1 can not learn the secret but others can learn it. If P_1 continue sending shares to others, he will violate selfishness and exclusivity assumptions. Therefore P_1 has no incentives sending shares to other.

In conclusion, P_1 has no incentives sending shares according to selfishness and exclusivity assumptions. For the same reasons, other rational parties also have no incentives sending shares in rational Shamir secret sharing scheme. Finally, no parties will receive shares and no one will learn the secret. In order to endow parties with incentives to send shares, Halpern and Teague proposed a random 3-out-of-3 Shamir secret sharing scheme. They also construct a random rational SMPC on the basis of the RSS such that the computation of function f can be completed in a constant expected time. Halpern and Teague present some open problems about RSMPC, the successive works discuss these problems.

Besides some results in RSS, there are still some results in RSMPC, which consist of two aspects: security under UC model and fairness in rational two-party computing. The former discusses how to realize security in concurrency and the latter discusses how to realize fairness in rational STPC. In traditional STPC, fairness is often neglected for the lack of honest majority. Cleve indicated that fairness can be achieved when most parties are honest (Cleve 1986). In STPC, it can not guarantee the condition that most parties are honest when one party is not honest. Therefore, fairness can not be achieved. This is why fairness is often neglected in STPC. Fortunately, the introduction of rational parties can solve this problem. Rational notions describe the incentive that parties participate in the protocol. So parties can be considered as rational when parties have incentives. In fact, rational notions can also be applied into other fields such as Byzantine protocol besides secret sharing scheme (Ogiela and Ogiela 2012, 2010) and SMPC.

This paper analyzes the research actuality, describes the application fields about RSMPC. Furthermore, some positive and negative results are presented. Finally we also give some open problems in this field.

2 Preliminaries

2.1 Utility functions

The incentives of rational parties can be described by utility, which is important for them since the strategies they choose depend on utilities. The strategies help them to maximize their utilities. Here we inherit the notions in game theory to define actions and utilities for parties who participate in the protocol. Let $\Gamma = (\{P_i\}_{i=1}^n, \{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$ denote a protocol with n parties, where $\{P_i\}_{i=1}^n$ denotes a set with n parities and P_i denotes one party. Let $\{A_i\}_{i=1}^n$ denote an action set, where the possible actions for P_i is A_i. Let $\{u_i\}_{i=1}^n$ denote utility function, which suffices $u_i : A_1 \times \cdots \times A_n \mapsto R$. Let $A \overset{def}{=} A_1 \times \cdots \times A_n$ such that $\mathbf{a} = (a_1, \ldots, a_n) \in A$ is an outcome of a protocol. The utility function u_i denotes the preference of P_i for a certain outcome. For example, if P_i prefer $\mathbf{a}$ with respect to $\mathbf{a}'$), then it can be denoted as $u_i(\mathbf{a}) > u_i(\mathbf{a}')$. If it suffices that $u_i(\mathbf{a}) \geq u_i(\mathbf{a}')$, we call P_i weakly prefers $\mathbf{a}$. Note that in SMPC protocol, action sets and utility functions are common knowledge (Osborne and Rubinstein 2004). However it's a strong assumption and we try to avoid this in the future works.

Many RSMPC protocols are based on RSS schemes. So we first give basic notions such as utility function, Nash equilibrium toward the view of RSS. In m-out-of-n Shamir secret sharing scheme, each party has one share and decides whether to share his share with others. When one party has m shares (m is the threshold of Shamir secret sharing scheme), he can obtain the secret using m shares. If all parties are willing to share their shares with others, they all obtain the secret. However, if one party shares his share while other parties do not send their shares to him, then he will not obtain the secret. Therefore, parties are caught in a dilemma, where it is difficult to decide whether to send shares to others. On one hand, if all parties do not share, no one will get the secret, which violates the selfish assumption. On the other hand, if they send shares to others while not receiving enough shares, then others will get the secret but he can not, which violates the exclusivity assumption. Therefore the best result is: all parties send shares to others and they all get the secret. Unfortunately, parties can not confirm that others will definitely send shares to them. Therefore parties will choose not sending shares to others for insurance purpose. Although all parties hope to get the secret, eventually neither can get. This is the dilemma in RSS scheme just like that in PD game.

Before give the definition of utility function and Nash equilibrium, we first present some basic symbols. Let an action tuple be $\mathbf{a} = (a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n)$, where a_i denotes the action of party P_i. Let $\mathbf{a_{-i}} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ denote the action tuple except P_i and $\mathbf{a}' = (a_i', \mathbf{a_{-i}}) = (a_1, \ldots, a_{i-1}, a_i', a_{i+1}, \ldots, a_n)$ denote the case where P_i choose the action a_i' while other parties choose the actions coming from the action tuple $\mathbf{a_{-i}}$. Let $u_i(\mathbf{a})$ denote the utility of P_i when the action tuple is $\mathbf{a}$. Let $\delta_i(\mathbf{a}) = 1$ denote the event where P_i learns the secret when P_i adopts a_i while others adopt actions in $\mathbf{a_{-i}}$. Let $\delta_i(\mathbf{a}) = 0$ where P_i does not learn the secret when P_i adopts a_i while others adopt actions in $\mathbf{a_{-i}}$. Let $num(\mathbf{a}) = \sum_{i=1}^{n} \delta_i(\mathbf{a})$ denote the number of parties who learn the secret. The utility definition is as follows according to the selfishness and exclusivity assumptions.

1. If $\delta_i(\mathbf{a}) > \delta_i(\mathbf{a}')$, then it suffices $u_i(\mathbf{a}) > u_i(\mathbf{a}')$;
2. If $\delta_i(\mathbf{a}) = \delta_i(\mathbf{a}')$ and $num(\mathbf{a}) < num(\mathbf{a}')$, then it suffices $u_i(\mathbf{a}) > u_i(\mathbf{a}')$.

We will analyze the results about the two outcomes $\mathbf{a}$ and $\mathbf{a}'$.

1. If $\delta_i(\mathbf{a}) = 1$ and $\delta_i(\mathbf{a}') = 0$ suffice such that $\delta_i(\mathbf{a}) > \delta_i(\mathbf{a}')$, it shows that the outcome $\mathbf{a}$ can lead P_i to learn the secret, while $\mathbf{a}'$ can not. P_i prefers the outcome $\mathbf{a}$ according to selfishness assumption. That is, the utility induced by the outcome $\mathbf{a}$ is better than that of $\mathbf{a}'$, therefore we have $u_i(\mathbf{a}) > u_i(\mathbf{a}')$.
2. If $\delta_i(\mathbf{a}) = 1$ and $\delta_i(\mathbf{a}') = 1$ or $\delta_i(\mathbf{a}) = 0$ and $\delta_i(\mathbf{a}') = 0$ such that $\delta_i(\mathbf{a}) > \delta_i(\mathbf{a}')$, it shows that either the outcomes $\mathbf{a}$, $\mathbf{a}'$ both lead P_i to learn the secret or they do not. In both case, P_i hope that the less number parties learn the secret, the better. On the other hand, the condition $num(\mathbf{a}) < num(\mathbf{a}')$ denotes that parties learning the secret when the outcome is $\mathbf{a}$ are less than parties when the outcome is $\mathbf{a}'$. P_i prefers the outcome $\mathbf{a}$ according to the exclusivity assumption. That is, the utility induced by the outcome $\mathbf{a}$ is better than that of $\mathbf{a}'$, therefore we have $u_i(\mathbf{a}) > u_i(\mathbf{a}')$.

For simplicity, we give the basic definitions in protocol $\Gamma = (P_1, P_2, A_1, A_2, u_1, u_2)$ with only two parties. P_1 and P_2 denote two rational parties. The threshold of RSS is 2. That is, one party must receive the other party's share to retrieve the secret. Let $A_1 = A_2 = \{Send, Not\,Send\}$, where $Send$ denotes the action that party sends share to the other and $Not\,Send$ denotes the action that party does not send share to the other. Let u_1, u_2 denote the utility of P_1 and P_2, respectively. Here we only give the definition of u_1. The definition of u_2 is similar to u_1. The utility definition is related with the outcomes. There are four outcomes in protocol Γ according to the actions.

1. $\mathbf{a}^1 = (Not\,Send, Send)$. P_1 does not send share to P_2 while P_2 sends share to P_1. P_1 has two shares and can learn the secret while P_2 can not learn the secret for lack of the other share. So it suffices that $\delta_1(\mathbf{a}^1) = 1$, $\delta_2(\mathbf{a}^1) = 0$ and $num(\mathbf{a}^1) = 1$ according to the utility definition.
2. $\mathbf{a}^2 = (Send, Send)$. Both parties send shares to the other and all of them learn the secret. So it suffices that $\delta_1(\mathbf{a}^2) = 1$, $\delta_2(\mathbf{a}^2) = 1$ and $num(\mathbf{a}^2) = 2$ according to the utility definition.

3. $\mathbf{a}^3 = (Not\,Send, Not\,Send)$. No party sends shares to the other and no one will learn the secret. So it suffices that $\delta_1(\mathbf{a}^3) = 0$, $\delta_2(\mathbf{a}^3) = 0$, $num(\mathbf{a}^3) = 0$ according to the utility definition.
4. $\mathbf{a^4} = (Send, Not\,Send)$. P_1 sends shares to P_2 while P_2 does not send shares to P_1. P_2 have two shares and can learn the secret while P_1 can not learn the secret for lack of the other share. So it suffices that $\delta_1(\mathbf{a^4}) = 0$, $\delta_2(\mathbf{a^4}) = 1$ and $num(\mathbf{a^4}) = 1$ according to the utility definition.

In the following, we give the relationships among the above utility definitions.

1. For $\delta_1(\mathbf{a}^1) = \delta_1(\mathbf{a}^2) = 1$ and $num(\mathbf{a}^1) = 1 < num(\mathbf{a}^2) = 2$, it suffices that $u_1(\mathbf{a}^1) > u_1(\mathbf{a}^2)$ according to the utility definition.
2. For $\delta_1(\mathbf{a}^2) = 1 > \delta_1(\mathbf{a}^3) = 0$, it suffices that $u_1(\mathbf{a}^2) > u_1(\mathbf{a}^3)$ according to the utility definition.
3. For $\delta_1(\mathbf{a}^3) = \delta_1(\mathbf{a^4}) = 0$ and $num(\mathbf{a}^3) = 0 < num(\mathbf{a^4}) = 1$, it suffices that $u_1(\mathbf{a}^3) > u_1(\mathbf{a^4})$ according to the utility definition.

In conclusion, it suffices that $u_1(\mathbf{a}^1) > u_1(\mathbf{a}^2) > u_1(\mathbf{a}^3) > u_1(\mathbf{a}^4)$. To better illustrate the relationships among the utilities, let $u_1(\mathbf{a}^1) = U^+$, $u_1(\mathbf{a}^2) = U$, $u_1(\mathbf{a}^3) = U^-$, $u_1(\mathbf{a^4}) = U^{--}$ such that $U^+ > U > U^- > U^{--}$. We follow the way in game theory to present utility in a matrix. Table 1 gives the utility matrix of rational parties.

Currently, most utilities are defined according to the selfishness and exclusivity assumptions. Recently, reputation is considered as one part in utility (Milgrom and Roberts 1986). Reputation is a positive evaluation to certain social group for their special ability (Achim et al. 2011; Serbanescu et al. 2012; Visan et al. 2011). Reputation exists in real world especially in economics and politics. For example, one company may cooperate with other companies who have good reputation. So if a company wants to have more cooperative chances, he should consider to build good reputation when they interact with others. In RSMPC, if sending shares to others may lead to good reputation, then rational parties are willing to earn good reputation by sending shares to others. Although sending shares may not directly lead to a higher utility in the current round, it may attract others to send shares to him in the following rounds. On the other hand, parties who do not send shares in the current round earn bad reputation and others may not send shares to them in the following rounds. Therefore, parties have incentives to earn good reputation for the sake of more shares. Furthermore, punishment is given to parties who do not send share at the same time when parties gain good reputation when they send shares to others. If parties send shares, then they will earn a positive reputation increment, otherwise a negative one. The higher the reputation one party has, the bigger the possibility of receiving shares it has. Considering the effect of reputation on utility, another assumption about the utility is needed. The new assumption is reputation, where every rational party hopes to earn good reputation. For simplicity, the increment is 1 when one party sends shares to others and -1 when he does not send. There are three assumptions about utility except for reputation: selfishness, exclusivity and reputation. Three factors ρ^1, ρ^2 and ρ^3 are assigned to the above three assumptions, respectively in order to denote the impacts on utility. Let $\rho^1 > \rho^2 > \rho^3$, this setting denotes that parties firstly hope earn good reputation, then hope to learn the secret and lastly hopes less parties learn the secret. The utility definition is given below when considering the impact of reputation.

Table 1 The utility matrix of rational parties

P_i P_j	Send	Not send
Send	(U, U)	(U^{--}, U^+)
Not send	(U^+, U^{--})	(U^-, U^-)

$$u_i(\mathbf{a}) = \rho_1 \cdot \tau_i(\mathbf{a}) + \rho_2 \cdot \delta_i(\mathbf{a}) + \rho_3 \cdot \frac{1}{num(\mathbf{a+1})} \qquad (1)$$

In Eq. (1), $\tau_i(\mathbf{a})$ denotes the impact of reputation on utility, where $\tau_i(\mathbf{a}) = 1$ when party sends share, otherwise $\tau_i(\mathbf{a}) = -1$. Let $\delta_i(\mathbf{a})$ denote the impact of selfishness on utility, where $\delta_i(\mathbf{a}) = 1$ when party learns the secret, otherwise $\delta_i(\mathbf{a}) = 0$. Let $\frac{1}{num(\mathbf{a+1})}$ denote the impact of exclusivity on utility, which means that this part is small when a lot of parties learn the secret and is big when few parties learn the secret. The denominator is set to be $num(\mathbf{a+1})$ rather than $num(\mathbf{a})$ in order to avoid the case denominator is zero when no one learns the secret. This part is ρ_3 when no one learns the secret then, $\frac{1}{2}\rho_3$ when only one party learns the secret and $\frac{1}{3}\rho_3$ when both parties learn the secret.

To describe the utility definitions considering reputation assumption, we still use protocol $\Gamma = (P_1, P_2, A_1, A_2, u_1, u_2)$ with two rational parties. The utility definition is related with the outcomes as mentioned above. There are still four outcomes in protocol Γ according to the actions.

1. $\mathbf{a}^1 = (Not\ Send, Send)$. P_1 does not send share to P_2 while P_2 sends share to P_1. P_1 has two shares and can learn the secret while P_2 can not learn the secret for lack of the other share. So it suffices that $\tau_1(\mathbf{a}^1) = -1$, $\tau_2(\mathbf{a}^1) = 1$, $\delta_1(\mathbf{a}^1) = 1$, $\delta_2(\mathbf{a}^1) = 0$ and $num(\mathbf{a}^1) = 1$

according to the utility definition. The utility function of P_1 and P_2 are $u_1(\mathbf{a}^1) = -\rho_1 + \rho_2 + \frac{\rho_3}{2}$ and $u_2(\mathbf{a}^1) = \rho_1 + \frac{\rho_3}{2}$, respectively.

2. $\mathbf{a}^2 = (Send, Send)$. Both parties send shares to the other and all of them learn the secret. So it suffices that $\tau_1(\mathbf{a}^2) = 1,\ \tau_2(\mathbf{a}^2) = 1,\ \delta_1(\mathbf{a}^2) = 1,\ \delta_2(\mathbf{a}^2) = 1$ and $num(\mathbf{a}^2) = 2$ according to the utility definition. The utility function of P_1 and P_2 are $u_1(\mathbf{a}^2) = \rho_1 + \rho_2 + \frac{\rho_3}{3}$ and $u_2(\mathbf{a}^2) = \rho_1 + \rho_2 + \frac{\rho_3}{3}$, respectively.
3. $\mathbf{a}^3 = (Not\ Send, Not\ Send)$. No party sends shares to the other and no one will learn the secret. So it suffices that $\tau_1(\mathbf{a}^3) = -1,\ \tau_2(\mathbf{a}^3) = -1,\ \delta_1(\mathbf{a}^3) = 0,\ \delta_2(\mathbf{a}^3) = 0$ and $num(\mathbf{a}^3) = 0$ according to the utility definition. The utility function of P_1 and P_2 are $u_1(\mathbf{a}^3) = -\rho_1 + \rho_3$ and $u_2(\mathbf{a}^3) = -\rho_1 + \rho_3$, respectively.
4. $\mathbf{a}^4 = (Send, Not\ Send)$. P_1 sends shares to P_2 while P_2 does not send shares to P_1. P_2 have two shares and can learn the secret while P_1 can not learn the secret for lack of the other share. So it suffices that $\tau_1(\mathbf{a}^4) = 1$, $\tau_2(\mathbf{a}^4) = -1$, $\delta_1(\mathbf{a}^4) = 0$, $\delta_2(\mathbf{a}^4) = 1$ and $num(\mathbf{a}^4) = 1$ according to the utility definition. The utility function of P_1 and P_2 are $u_1(\mathbf{a}^4) = \rho_1 + \frac{\rho_3}{2}$ and $u_2(\mathbf{a}^4) = -\rho_1 + \rho_2 + \frac{\rho_3}{2}$, respectively.

Since the utilities of P_1 and P_2 are symmetric, here we only analyze the relationship of P_1. It suffices that $u_1(\mathbf{a}^2) > u_1(\mathbf{a}^4) > u_1(\mathbf{a}^1) > u_1(\mathbf{a}^3)$ because of $\rho_1 > \rho_2 > \rho_3$. To better illustrate the relationships of the utilities, let $u_1(\mathbf{a}^1) = RU^-$, $u_1(\mathbf{a}^2) = RU^+$, $u_1(\mathbf{a}^3) = RU^{--}$, $u_1(\mathbf{a}^4) = RU$. So we have $RU^+ > RU > RU^- > RU^{--}$. Table 2 is the utility matrix of rational parties when considering reputation.

2.2 Notions of equilibrium

2.2.1 *Nash equilibrium*

Suppose utility function is common knowledge, if P_1 knows that other parties choose $a_2, \ldots, a_n$ then P_1 is sure to adopt $a_1 \in A_1$ to maximize his utility. a_1 is called best response of P_1 with respect to $a_2, \ldots, a_n$. Given a_1,P_2 will choose $a_2' \in A_2$ and so on such that each party will choose his best response. The tuple $\mathbf{a}$ is called self-enforcing if a_i and only if is a best response with respect to P_i. If one action tuple is a self-enforcing, it is called Nash equilibrium. Definition 1 gives the formal definition of Nash equilibrium.

Table 2 The utility matrix of rational parties (considering reputation)

P_iP_j	Send	Not send
Send	(RU^+, RU^+)	(RU, RU^-)
Not send	(RU^-, RU)	(RU^{--}, RU^{--})

Definition 1 (*Nash equilibrium of pure strategy*) Let $\Gamma = (\{P_i\}_{i=1}^n, \{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$ denote a normal game with n parties. An action tuple $\mathbf{a}$ is Nash equilibrium of pure strategy, if for every i and $a_i' \in A_i$, we have

$u_i(a_i', \mathbf{a}_{-\mathbf{i}})$.

In Table 2 $(Not\ Send, Not\ Send)$ is Nash equilibrium of pure strategy. That is, if P_1 chooses *Not Send*, then the best response of P_2 is *Not Send*. The utilities for both parties are all U^-. Otherwise, if P_1 chooses *Not Send* while P_2 deviates from Nash equilibrium to choose $a_i' = Send$, then the outcome is $(Not\ Send, Send)$. The utility of P_2 is U^{--} which is smaller than U^-. Therefore, P_2 has no incentive to deviate from Nash equilibrium since deviation lead to an inferior utility. P_1 will not deviate from Nash equilibrium for the same reason. The final result is that both P_1 and P_2 following the action tuple in Nash equilibrium.

In Table 2, (*Send*, *Send*) is Nash equilibrium of pure strategy. That is, both parties have incentives to send shares considering the impact of reputation. Finally, both will learn the secret. This result is different with that of Table 1, where no parties have incentives to send shares. P_1 either adopts Send or Not Send in Nash equilibrium of pure strategy. The strategies in this case are pure strategies. However, Nash equilibrium of pure strategy may not exists in games. To solve this problem, rational parties are allowed to adopt mixed strategies, which mean that parties may choose actions with certain probabilities. For example, P_1 adopts *Send* with probability of $\frac{1}{3}$, *Not Send* with probability of $\frac{2}{3}$. Letσ_i denote a probability distribution on A_i or σ_i can be redeemed as a strategy of some party, where P_i samples fromA_i such that $\sum_{a_i' \in A_i} pr(a_i) = 1$. Given a strategy vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{i-1},\sigma_i, \sigma_{i+1}, \ldots, \sigma_n)$, where σ_i denotes mixed strategy of P_i. If parties play the game according to the strategies in $\boldsymbol{\sigma}$, then we use $u_i(\sigma)$ to denote the expected utility. Other notions are similar to those of pure strategies. Let $\boldsymbol{\sigma}_{-i} = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$ denote the mixed vector except P_i and $\sigma' = (\sigma_i', \sigma_{-i}) = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \ldots,\sigma_n)$ denote that P_i adopts mixed strategies in σ_i' while others adopt strategies in σ_{-i}. If σ_i can maximize $u_i(\sigma_i, \sigma_{-i})$, then it is a best response with respect to σ_{-i}. The definition of mixed strategy Nash equilibrium is given in Definition 2.

Definition 2 (*Mixed strategy Nash equilibrium*) Let $\Gamma = (\{P_i\}_{i=1}^n, \{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$ denote a normal game with n parties. Mixed strategy σ_i denotes a probability distribution on A_i, where $\sigma_i \in \Delta(A_i)$. $\sigma = (\sigma_1, \ldots, \sigma_n)$ is mixed strategy Nash equilibrium, if for every i and $\sigma_i' \in \Delta(A_i)$, we have

$$u_i(\sigma_i', \sigma_{-i}) \leq u_i(\sigma).$$

In SMPC, the computation ability is bounded, so computational Nash equilibrium is often considered in RSMPC.

Definition 3 (*Computational Nash equilibrium*) Let $\Gamma = (\{P_i\}_{i=1}^n, \{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$ denote a normal game with n parties. An action tuple $\mathbf{a}$ is computational Nash equilibrium, if for every i and $a_i' \in A_i$, there exists a negligible function ε such that

$$u_i(a_i', \mathbf{a_{-i}}) \leq u_i(\mathbf{a}) + \varepsilon.$$

2.2.2 *Other equilibrium notions*

Nash equilibrium is basic equilibrium notions in game theory. Besides, there also some other refined equilibrium notions. If there exists a random strategy $\sigma_i \in \Delta(A_i)$ and for each $\mathbf{a_{-i}} \in \mathbf{A_{-i}}(\mathbf{A_{-i}} \overset{def}{=} \times_{j \neq i} A_j)$ such that $u_i(\sigma_i, \mathbf{a_{-i}}) > u_i(a_i, \mathbf{a_{-i}})$, then $a_i \in \Delta(A_i)$ is a strictly dominated strategy with respect to A_I. If there exists a random strategy $\sigma_i \in \Delta(A_i)$ such that:

1. For every $\mathbf{a_{-i}} \in \mathbf{A_{-i}}$ such that $u_i(\sigma_i, \mathbf{a_{-i}}) \geq u_i(a_i, \mathbf{a_{-i}})$;
2. There exists $\mathbf{a_{-i}} \in \mathbf{A_{-i}}$ such that $u_i(\sigma_i, \mathbf{a_{-i}}) \geq u_i(a_i, \mathbf{a_{-i}})$.

Then $a_i \in A_i$ is a weakly dominated strategy with respect to A_{-i}.

Definition 4 Iterated deletion of weakly dominated strategy, IDWDS. Let $DOM_i(\hat{A})$ denote a set of weakly dominated strategies in $\hat{A}_i$ with respect to $\hat{A}_{-i}$. For $k \geq 1$, let $A_i^{k\,def} {=} A_i^{k-1} \backslash DOM(A^{k-1})$ and $A_i^{\infty} \overset{def}{=} \cap_k A_i^k$. If for every i, it suffices that $\sigma_i \in \Delta(A_i^{\infty})$, then Nash equilibrium σ is IDWDS.

Another refinement of Nash equilibrium is called trembling hand perfect equilibrium. Parties may deviate from the protocol with a small probability, which is called "trembling hand". Trembling hand perfect equilibrium can not only guarantee the strategies are optimal when there are no trembling hands but also optimal when there are trembling hands.

The above equilibrium notions discuss the scenarios where only one party deviates from the protocol. However, some parties may collude to deviate from the protocol for optimal utilities. In SMPC, some parties are corrupted by an external adversary, which can be considered as collusion among the parties. Let $\mathbf{C} = \{\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_t}\} \subset \mathbf{P}$ be a set of collusion parties, where P denotes the set of all parties. $\mathbf{A_C} \overset{\mathbf{def}}{=} \times_{\mathbf{i \in c}} \mathbf{A_i}$ denotes the action set of collusion parties, $\sigma_c \overset{def}{=} (\sigma_{p_1}, \sigma_{p_2}, \ldots, \sigma_{p_t})$ denotes the strategy set of collusion parties, and $\sigma_{-c} \overset{def}{=} \sigma_{\mathbf{p|c}}$ denotes the strategy set of parties other than collusion parties.

Definition 5 t-Resilient equilibrium. If for every $\mathbf{C} = \{\mathbf{P_1}, P_2, \ldots, P_t\} \subset P(|C| \leq t)$, i and $\sigma_c' \in (A_c)$, it suffices that $u_i(\sigma_c', \sigma_{-c}) \leq u_i(\sigma)$, then $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is t-resilient equilibrium.

Correlated equilibrium describes such scenario, where exists a trusted mediator, who will "recommend" some strategies to parties such that they will obtain optimal utilities if they play the game according to the strategies. The efficiency of computational correlated equilibrium is better than Nash equilibrium, where the former can be achieved in polynomial time (Gilboa and Zemel 1989) while the latter is an NP hard problem. Urbano and Vila (2004) in correlated equilibrium in computational environment. Dodis et al. (2000), Gradwohl et al. (2013), Lepinski et al. (2004) also discussed the notions of correlated equilibrium. Dodis et al. (2000) proved that correlated equilibrium can be reached between two parties if they are computationally bounded and they can communicate before the game. Gradwohl et al. continued the work of Dodis et al. (2000) and considered how to implement a mediator to reach correlated equilibrium. They proposed a sequential rational protocol for non-trivial correlated equilibrium and prove that mediator can be replaced by a relative stable protocol (Gradwohl et al. 2013). Lepinski et al. (2004) proposed a fully fair secure function evaluation, which is secure in the presence of many malicious adversaries. However, the protocol needs physical communication channels, which are hard to realize in reality. In addition, other equilibrium notions include Bayesian Nash equilibrium, which considers the situation where parties have types under incomplete information. Therefore, optimal strategies are related to not only strategies but also types of other parties.

2.3 Iterated games and backward induction

In most cases, one game may include several rounds. That is, parties repeatedly play one game Γ and it is denoted as $(\Gamma_1, \Gamma_2, \ldots, \Gamma_T)$, where T denotes the number of rounds. History H records actions which are adopted by parties in each round. Let $a^k = (a_1^k, a_2^k, \ldots, a_n^k)$ denote the action adopted in the kth round. For simplicity, the initial round is set to be 0 and denoted as $H = \Phi$. Parties choose their actions according to history $h^k = (a^0, a^1, \ldots, a^k)$. In each round, P_i obtain utility $u_i (i \in N)$. Maleka and Shareef (denoted as MS protocol) first propose RSS using iterated games, which introduce punishment into the protocol. However, their protocol can only construct RSS within unlimited rounds and invalid for limited rounds (Maleka et al. 2008b). In complete information games with T rounds, parties do not worry to be punished in the last round since there is no chance for others to punish them. So parties have no incentives to send their shares to others in the last round. For the same reason, parties have no incentives to send their shares in the penultimate round. In a similar fashion, parties will not share their share in $T-2, T-3, \ldots 1$ round. The above process is backward induction. To avoid this in complete information where parties know exactly when the protocol ends, Abraham et al. (2006), Kol and Naor (2008a, b), Katz (2008) propose various methods.

2.4 Extensive games

The games where parties alternatively adopt actions should be described by extended games.

Definition 6 An extensive game with imperfect information $<\mathcal{P}, A, H, P, f_c, (\mathcal{I}_{P_i})_{i\in\{1,2\}}, \succsim_i>$ has the following components.

1. The set of two parties $\mathcal{P} = \{P_1, P_2\}$.
2. The action set $A = A_1 \times A_2$. P_i $(i \in \{1,2\})$ chooses his action from the set of A_i. The action profile $a = (a_1, a_2) \in A$.
3. A set H of sequences (finite or infinite) that satisfies the following three properties.
   (a) The empty sequence $\varnothing$ is a member of H.
   (b) If $(a^k)_{k=1,2,\ldots,K} \in H$ (where K may be infinite) and $L < K$ then $(a^k)_{k=1,2,\ldots,L} \in H$.
   (c) An infinite sequence $((a^k))_{k=1}^{\infty} \in H$.
   Each member in the set of H is a history which is a sequence of actions profiles.
   A history $(a^k)_{k=1,2,\ldots,K} \in H$ is terminal if it is infinite or if there is no a^{k+1} such that $(a^k)_{k=1,2,\ldots,K+1} \in H$. The set of actions available after the non-terminal histories is denoted Z.
4. A function P that assigns to each non-terminal history a member coming from $\{P_1, P_2\} \sqcup \mathfrak{c}$. P is the *party function* and $P(h)$ $(h \in H)$ assigns the party who should take actions after the history h. $P(h) = \mathfrak{c}$ means that an exterior *chance* will determine the actions after the history h. Note that $P(h) = \mathfrak{c}$ occurs only at the initial round of the protocol.
5. A function f_c where $P(h) = \mathfrak{c}$ assigns an independent probability measure $fc(\cdot|h)$ on $A(h)$. Note that $fc(a|h)$ is the probability that $a \in A$ occurs after history h.
6. For each party, a partition is denoted by $\mathcal{I}_{P_i}$ $(i \in \{1,2\})$ such that $h \in H : P(h) = P_i$. The property of the partition is that $A(h) = A(h')$ whenever h and h' are the same member of partition. For each $I_i \in \mathcal{I}_i$, denote $A(I_i)$ as the set $A(h)$ and $P(I_i)$ as the party $P(h)$ for any $h \in I_i$. Note that, $\mathcal{I}_i$ is the information partition of P_i $(i \in \{1,2\})$, while $I_i \in \mathcal{I}_i$ is an information set of party P_i.
7. For each party, a preference relation $\succsim_i$ on lotteries over Z (the *preference relation* of party P_i) that can be described as the expected value of a payoff function defined on Z.

The following games after the first round are sub-games of the initial game (Myerson 2013). Sub-games are similar to the initial games, which include initial information set and all information needed in the following games. In extensive games, parties who choose actions lately (denoted as PCAL) may choose beneficial actions with respect to parties who choose actions firstly (denoted as PCAF). This is equal to some kind of "commitment". On the other hand, in the following games, PCAL may also adopt disadvantage actions with respect to PCAF. This is equal to some kind of "threat". In some games, threat is incredible, also called empty threat. The reason for empty threat is: if PCAL acts according to the threat strategies he claimed, he may obtain an inferior utility than not according to the threat strategies. That is, he only uses the threat strategies to frighten others and may not really adopt the threat strategies he claimed. Therefore, other parties do not believe the threat. Empty commits are defined similarly. To exclude the empty threat or commit, a refinement of Nash equilibrium is sub-game perfect Nash equilibrium.

Definition 7 Sub-game perfect Nash equilibrium. In imperfect information extensive games, if strategies from each party form a strategy profile, which is Nash equilibrium in initial extensive games and its sub-games, then this strategy profile is sub-game perfect Nash equilibrium.

3 Classical schemes

3.1 Rational secret sharing schemes

The RSS schemes proposed by Halpern and Teague (denoted as HT protocol), consisting of three parties P_1, P_2 and P_3. For simplicity, we use index $i \in \{1,2,3\}$ to denote each party. Let i^+ denote $i+1$, where 3^+ denotes 1 when $i=3$. Similarly i^- denotes $i-1$ when $i=1$. HT protocol is described as follows.

– Stage 0. The secret dealer assigns each party one share using 3-out-of-3 Shamir secret sharing scheme.
– Stage 1. Each party i chooses on bit c_i such that $c_i = 1$ with probability α and $c_i = 0$ with probability $1-\alpha$. Meanwhile party i chooses another random bit $c_{(i,+)}$

such that $c_{(i,+)} = 1$ and $c_{(i,+)} = 0$ with probability $\frac{1}{2}$. Let $c_{(i,-)} = c_i \oplus c_{(i,+)}$. Party i sends $c_{(i,+)}$ to i^+, $c_{(i,-)}$ to i^-. Note that this means that party i receives $c_{(i^+,-)}$ from i^+ and $c_{(i^-,+)}$ from i^-.

- Stage 2. Each party i sends $c_{(i^+,-)} \oplus c_i$ to i^-. So i should receive $c_{((i^+)^+,-)} \oplus c_{(i)^+} = c_{(i_-,-)} \oplus c_{i^+}$ from i^+.
- Stage 3. Each party i computes $p = c_{i^-,+} \oplus c_{i^-,-} \oplus c_{i^+} \oplus \ c_i = c_{i^-} \oplus c_{i^+} \oplus c_i = c_1 \oplus c_2 \oplus c_3$. If $p = c_i = 1$, then i sends his share to others.
- Stage 4. If $p = 0$ and i does not receive any shares, or $p = 1$ and i only receives one share. Note that this share may be his. That is, i does not receive any share from others. Meanwhile, the secret dealer requires restarting the protocol. Otherwise i aborts the protocol. i aborts the protocol either because he receives all three shares or he detects someone is cheating.

The expected running time of HT secret sharing scheme is $\frac{5}{\alpha^3}$. However it is only fit for three parties. For the case where $m \geq 3$, $n > 3$, the solution is as follows. These parties are divided into three groups, each of which has a leader. Parties in each group send their shares to the leader, then three leaders run HT secret sharing scheme. When leaders receive shares from other leaders, they will send these shares to group members. Halpern and Teague construct RSMPC on the basis of HT secret sharing scheme, denoted as HT protocol, which combines HT secret sharing scheme and SMPC of Goldreich et al. (1987). In HT protocol, parties are allowed to replace their initial inputs and to prematurely abort. Once they confirm their initial inputs, they must abide by what the protocol told them to. The basic idea of HT protocol is as follows. Simulate a circuit to compute function f such that the value of each node in the circuit is considered as a secret and all parties own one share of this secret. HT protocol replaces the last stage with HT secret sharing scheme in the protocol of Goldreich.

HT protocol relies on simultaneous channel, which is hardly to realize in reality. Other schemes relying on simultaneous channel (Maleka et al. 2008b; Abraham et al. 2006; Cai and Peng 2012; Zhang and Cai 2012; Gordon and Katz 2006; Maleka et al. 2008a; Luo et al. 2012; Zhang and Liu 2013; Zhang and Cai 2010; Isshiki et al. 2010). In traditional SMPC, semi-honest behaviors can be converted into malicious behaviors using tools such as zero knowledge and bit commitment. Here we use similar tools in HT protocol. To set the value of α, the utility function should be common knowledge. Furthermore, HT protocol only fits for three parties. For more than three parties, HT protocol can not resist the case where three leaders collude. Halpern and Teague also give some open problems at the end of their paper.

1. Can HT protocol also establish in asynchronous data channels?
2. HT protocol discusses Nash equilibrium and IDWDS. They wonder if there are any other stronger equilibriums satisfying RSMPC.
3. The utility function is assumed to be common knowledge. The following question is what is the result when removing this restriction.

Asharov and Lindell (2011), Fuchsbauer et al. (2010), Micali (2009), Izmalkov et al. (2005), Tian et al. (2011a) solved these open problems through different utility settings and communication assumptions. In secret sharing schemes, it gives solutions for these problems through various assumptions such as utility functions and communication channels etc. The rational secret sharing schemes in Izmalkov et al. (2005) and Tian et al. (2011a) need a trusted party when reconstructing the secret. However, it's hard to find a trusted party. William et al. (2011) proposed rational secret sharing schemes under the asynchronous channels (Moses et al. 2011), but they need honest parties to join in. Gordon and Katz (2006) (denoted as GK protocol) introduced the notion of active parties guaranteeing parties to recover the secret when the secret dealer was offline. There, parties interact with others within constant rounds. Parties can choose to abort the protocol or enter the next round of the protocol at the end of each round. The basic idea of HT protocol as follows:

1. Initialization phase: secret dealer selects a probability β (the selection of β depends on the utility function), then he assigns the correct secret $S \in F$ with probability β, where F is finite field. He also assigns a random generated secret $\hat{S} \in F \backslash S$ with probability $1 - \beta$.
2. Secret assignation phase, the secret dealer distributes the shares of secret to all parties, but neither of them know whether the shares are true.
3. In the execution phase, rational parties set the variable $all\_honest = true$, then they use broadcast channel to share the secret. If they get enough shares to reconstruct the secret $a' \in S$, it means that it's the correct secret. Then the rational parties send a signal to secret dealer, finish the protocol. If the constructed secret $s' \in F \backslash S$, it means that the secret is not true, then the protocol enter the next round. If the shares do not reach the threshold value, then set the variable $all\_honest = false$, and the protocol enter to next round.

Katz (2008) discusses the relation between Game theory and Cryptographic protocol. He points out that both of Game theory and Cryptographic protocol study the interaction problem among distrustful parties. However, the two seemingly unrelated fields can interpenetrate. At the end of this paper, he points out two research directions:

1. Apply Cryptographic protocol into Game theory. Some equilibriums in Game theory can acquire through setting trusted mediator. The major problem of this direction is whether the trusted mediator can be replaced by the parties in distributed Cryptographic protocols.
2. Apply Game theory in Cryptographic protocol, this is the major research direction of rational secure multi-party computing. Traditional Cryptographic models suppose that parties execute protocols honestly or maliciously. However, Game theory models regard parties as self-interest or rational. The major problem of this direction is how to design practical Cryptographic protocol towards the view of Game theory.

Currently, there is lack of formal definitions about rational secure multi-party computing and most utility functions derive from HT protocol. Just as Katz points out, the problem is how to use other utility functions and consider more complicated situations than function evaluation are main topic in the near future. Now the functions in rational secure multi-party computing are limited to non-cooperatively computable (NCC) (Shoham and Tennenholtz 2005). In addition, Katz also points out that parties in rational secure multi-party computing don't live in a vacuum. Instead, they exist in certain legal framework, so we can consider to apply Game theory into covert models. Abraham et al. (2006) (denoted as Abraham protocol) discussed the rational secure multi-party computing in the presence of coalition among parties. Comparing with HT protocol, the advantages of Abraham protocol are as follows:

1. HT protocol discusses the situation which only one party deviate from the protocol. However, Abraham protocol discusses the situation which less than $t-1$ parties collusion and proposes Nash equilibrium resisting $t-1$ collusion.
2. HT protocol is not suitable for rational secure multi-party computing in the presence of two parties, while Abraham protocol is.
3. For the sake of avoiding the negative conclusions of HT protocol, Abraham protocol set a parameter β. The protocol rounds depend on the parameter, which degrades the efficiency of the protocol. Abraham et al. prove that as long as one party tends to get the result of the utility function, then it will reach k-resist Nash equilibrium, where $k < \lceil \frac{n}{3} \rceil$.

The collusion resistance protocol of Kol and Naor's (2008a) can give parties incentives after the execution of the protocol. Therefore, these protocols can be finished within constant rounds and are immune to backward induction. Kol and Naor put forward a new tool of Cryptographic, which is called Meaningful/meaningless encryption. The features of encryption mechanism are as follows: The cipher-text generated by public key cannot be decoded (even parties have infinite computation ability). This kind of secret keys is called meaningless and others are called meaningful. Moreover, the meaningful secret keys provide semantic security. Unless a party has secret key, he cannot distinguish meaningless and meaningful secret keys. In simultaneous broadcast channel, the rational secure multi-party computing proposed by Kol and Naor is called secure-clean-slate protocol by using garbled circuits. Garbled circuits refer to forms of original circuit encryptions. Garbled circuits allow calculating the value of circuit, but they cannot reveal anything except this value. One garbled circuit consists of the following sections: two random strings (they are given to the each one input circuit, the first corresponds to value 0 and the second corresponds to value 1), gates tables and translation tables for outputs. Calculate the value of garbled circuit according to gates tables with respect to corresponding random trains. Then translate the output of Garbled circuit by using translation tables. The basic steps of secure-clean-slate protocol are as follows:

1. Generate garbled circuit. In each round, the protocol constructs a new garbled circuit function f. Make gates tables and translation tables public, commit to random stings of each input, and disrupt the order of commitments. However, parties have no ideas of these random strings. For any x'_i, it's likely to make parties learn $f(x_{-i}, x'_i)$. Divide random strings of each input circuit's into n shares and commit to these shares. If parties hope to reconstruct the random strings, they have to learn all of n shares.
2. Learn the output of garbled circuit. Each party acquires a random string, which is chosen for each input line according to the input of that input line. Party i learns all shares of each random string by executing 1-out-of-2 oblivious transmission protocol with other party j. In 1-out-of-2 oblivious transmission party j is considered as the sender and it's values are random strings chosen for party i. Meanwhile party i is considered as receiver and it's purpose is acquire the value that the matching bit value.
3. Encryption and verification. Parities use β-meaningful/meaningless scheme to encrypt their random strings other than their original inputs. In a meaningful encryption round, the protocol decrypts the ciphers and reconstructs the random strings. Then the protocol certificates each random string can open its corresponding commit. Since each commitment of random strings are published in random order at the beginning of the circuit construction, there is no information about inputs can be revealed to other parities. In the exchange step of the meaningful encryption, random

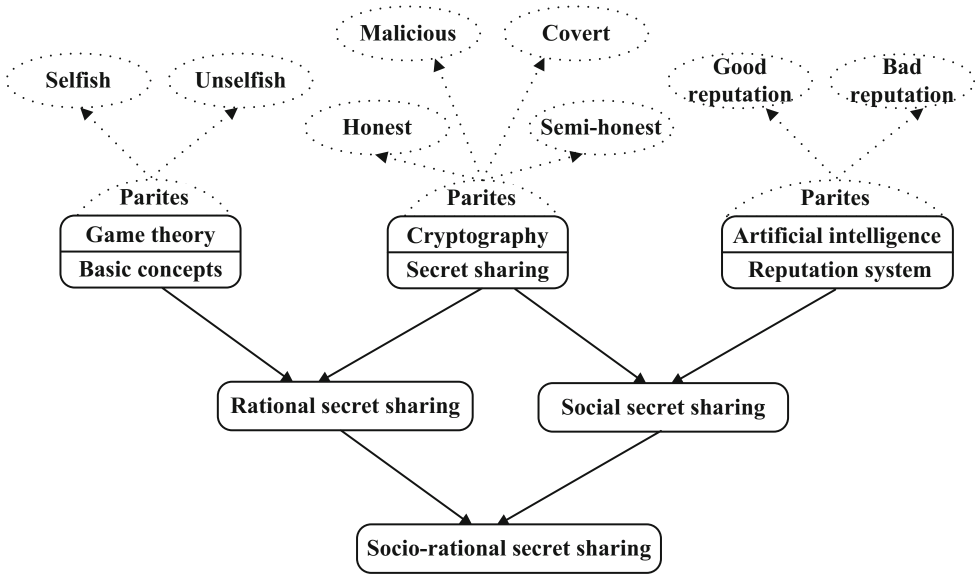

Fig. 2 The relationship of socio-rational secrete sharing scheme

strings can be reconstructed from the encryption information such that parities can acquire the value of the function. However, it protects the original value of parities. In the meaningless encryption round, no information about random strings of the input values is revealed, so no information about the true input value can be revealed.

To prevent parities from aborting the protocol after they received the message, Kol and Naro (2008b) (denoted as KN protocol) set a parameter β to make the parities have no idea about whether it is the last round. As far as the agreement that have n rational parities, they set a short message and $n-1$ long message. KN protocol does not rely on any difficult problems, is immune to the backward induction and is satisfied with strict Nash equilibrium. On the contrary, the secure-clean-slate protocol relies on difficult problems and it uses some cryptographic tools such as, commitment mechanism, traditional multiparty secure computing and vacant transmission etc. Luo et al. (2012) The shortcoming of KN protocol lies in failing to resist collusion of two parities (one short party with one long party) such that other parities can not complete the protocol.

Maleka and Amjedt put forward a rational secret sharing scheme based on repeated game (Maleka et al. 2008a, b). It needs consider the influence of the discount factor on parities' utility. The efficiency of their solution is low and it can not resist collusion among parities. Nojoumian combines rational secret sharing scheme and social secret sharing and puts forward the socio-rational secret sharing scheme (Nojoumian and Stinson 2012). The relationship between socio-rational secret sharing scheme and the other two schemes is shown in Fig. 2. The authors construct a public trusted network among parities, where parities' weights are updated constantly. The updating rule is: the weight of parties who cooperate is bigger than those who defect (Nojoumian et al. 2010; Nojoumian and Stinson 2010). This credible network can be regarded as a reputation system in the field of artificial intelligence. They all consider the effect of reputation on parties.

Domestic scholars also studied socio-rational secret sharing scheme. Yilei Wang et al. constructed a socio-rational secure multi-party computing protocol based on the socio-rational secret sharing scheme (Wang et al. 2014). The basic idea is as follows: rational parities hope to complete the computation of a function through cooperation in a social network. Meanwhile parties consider their reputation in the network for subsequent calculations. Three problems should be taken into consideration while implementing the protocol: (1) the network composed of these parities may not be a complete network; (2) communications in the network may not be safe; (3) parities may compute under incomplete information. In order to solve these problems, they put forward a socio-rational secure multi-party computing, which allows parities to complete computation safely and efficiently in constant rounds. Zhang and Liu (2011) designed an unconditionally secure 2-out-of-2 Social rational secret sharing on standard peer-to-peer channel. The rational secret sharing scheme reach ε Nash equilibrium, where ε is a negligible function introduced when the internal scheme uses MAC. Then they designed a t-out-of-n rational secret sharing scheme based on 2-out-of-2 rational secret sharing scheme and the scheme can resist collusion of t-1 participants. Tian et al. (2011b) proposed the concept of rational secret distributor and constructed a secret reconstruction scheme based on oblivious transfer. This scheme makes the non cooperative participants effectively recover the secret. Yongquan

Zhang proposed a verifiable rational secret sharing scheme based on bilinear pairing, which not only can improve the efficiency of the secret distribution but are also resistant to the collusion of most parties (Zhang and Cai 2012). Cai et al. put forward a kind of rational multi-secrets sharing scheme by using bit commitment based on one-way Hash function. The scheme share multiple secrets among parties through broadcasting channel. However, it can't prevent the collusion of several parties (Luo et al. 2012).

3.2 Rational multiple function calculation

3.2.1 *Rational multiple function calculation under the UC model*

The UC model (Canetti 2000) is proposed by Canetti to define frameworks for security protocol. Furthermore, it defines security of multi-party computing according to the method of indistinguishability between ideal and real protocol. The UC model designs secure protocols by the idea of modularization. The main idea is as follows. Firstly divide the protocol into several parts and select UC model for each part. Secondly, combine these parts into a complex security protocol by using UC theorem. The security level under UC model is higher than those in the general form. Lysyanskaya and Triandopoulos (2006) (denoted as LT protocol) firstly discuss rational multi-party computing in the presence of rational and malicious parties. They propose a protocol allowing rational parties to simulate a trusted third party and compute a function, it suffices the following conditions.

1. Assume that each rational party tends to learn his own results and others can't learn the results.
2. Rational parties can be protected in some ways.

For example, If the adversary corrupt more than $\lceil \frac{n}{2} \rceil - 2$ parties, the adversary either make all rational parties withdraw from the protocol or only get the information, which have the same distribution as their input and output. The basic idea of LT protocol is to first construct ideal-real model and then prove that it is a computational t-security priority function. The followings are a description of LT protocol.

1. Inputting phase of the protocol. Each party P_i receives input x_i and security parameters l^k. If P_i receives any information from other channels and the information comes from the adversary, they will ignore the information. Otherwise, abort the protocol.
2. Preparing phase. All parties reach an agreement on a public string of length $l(k)$ and public key infrastructure PK_i of the participants P_i.
3. Inputting phase of random values. Parse $CRS = CRS_{COM} \circ CRS_{MPC} \circ CRS_{NIZK}$, Let $SECom$ denote simulating and extractable commitment and CRS_{COM} be input. Each party P_i broadcast $z_i = SECom(CRS_{COM}, x_i, r_i^{SEC})$, and r_i^{SEC} is a random value in the commitment. If P_j observed that P_i does not broadcast a valid commitment, then P_j will abort the protocol.
4. Multi-party computing phases. At this phase P_i wants to get the share of $y_i = 0^{p_i}(1 \le i \le n)$, where $P_i = | out_i |$ is known, The protocol hopes that each type of outputs has the same probability. It an be achieved through the following ways: P_i chooses r_i^{MPC} a random value, send $(x_i,\ r_i^{SEC},\ r_i^{MPC})$ to multi-party computing protocol computing $g_{PKI,z}$ though the broadcast channel and used CRS_{MPC} as a public random string.
5. Recovering phase. Each P_i broadcast news $(\{d_{j,i} : 1 \le j \le n\}, \pi)$, where π is a non-interactive simulator of zero-knowledge proof. Let $d_{j,i} = Enc(PK_j, y_{j,i}, r_{j,i})$, where $y_{j,i}$ is a correct decryption of $c_{j,i}$ by using public key PK_i and $r_{j,i}$ is a random value generated in probabilistic encryption algorithm. If P_i receives m valid proofs, he can decrypt all cipher text $\{d_{i,j}\}$, then gain m valid shares of 0 or $1 \circ f_i(\mathbf{x})$. In the first case, if P_i receives less than n valid proof of the message, then P_i abort the protocol; Otherwise, P_i returns to the multi-party computing phase. In the second case, P_i outputs $f_i(\mathbf{x})$.

Recently Garay et al. (2013) considered the incentives of rational parties and discussed how to design rational protocols. They point out that threaten of protocols are modeled into an external individuals–Adversary. The adversary can corrupt limited parties and make them participate in the protocol arbitrarily. In the ideal-real para-digm, it is proved that the existing protocols are still secure in the presence of malicious adversary. Assuming the existence of malicious adversaries, although the protocols can be proved to be secure, it has long been criticized as too pessimistic. The main reason is that the malicious adversary assumption is too strong and it ignores the incentives for parties to deviate from the protocol. Protocols under such assumptions may be designed to resist some meaningless attacks. Since practical adversaries hardly attack the protocol without any incentives, on the contrary, practical adversaries always attack protocols with certain purposes, which are called incentives. The designation of rational protocol is that following the protocol is an equilibrium for rational participants (Halpern and Teague 2004; Abraham et al. 2006; Kol and Naor 2008a; Fuchsbauer et al. 2010; Asharov et al. 2011; Groce and Katz 2012; Halpern and Pass 2008; Pass and Halpern 2010; Gradwohl et al. 2013). The above protocols based on game-theory models are useful for building incentives among distributed and distrustful parties. However, it cannot be directly applied to the following scenario, where some distrustful parties hope to complete the protocol (Huang and Su 2006; Alfredo and Ahmed 2011). What they care about is the strategies of the attackers, who have their own behavioral biases. These biases will affect the strategies of attackers in the protocol.

The basic idea of Garay protocol is to convert the incentives of motivation driving attacks into a two-party game between protocol designer D and protocol attacker *Adv*. The protocol designers D specifies a protocol Π for most honest parties and protocol attacker *Adv* specifies a polynomial attacking strategy for the external adversary *A*, such that he can corrupt other parties trying to undermine the protocol. D and *Adv* 's have unbounded computational ability, so the protocol is similar to a zero-sum extensive game with perfect information and observable actions after conversion. A typical game of this kinds is Stackelberg game (see reference Osborne and Rubinstein 1994, 6.2). Attacker's aim is to choose strategies maximizing his utility. Since it is a zero-sum game, the designer aims to minimize the utility of the attacker. Towards the view of game theory, the equilibrium is an ε-subgame perfect equilibrium, which is a refine of sub-game perfect equilibrium. This equilibrium is difficult to achieve when parties have polynomial computing abilities (Kol and Naor 2008a, b; Gordon and Katz 2006; Fuchsbauer et al. 2010; Gradwohl et al. 2013). The equilibrium in Garay protocol is not affected by this restriction since they discuss unbounded computing power. The main results of Garay protocol are as follows:

1. Assuming that the cost of the adversary when corrupting parties is higher than the utility when he breaks the selfishness of the protocol, then there exists a protocol which can calculate any function. Conversely, if the utility when he breaks selfishness or correct is higher than the cost, it is impossible to construct a valid protocol to compute a function. Assuming the utility when breaking selfishness is higher than the cost and the utility when breaking correctness is lower than the cost, it can also construct a protocol for an arbitrary function.
2. Assuming the cost while breaking selfishness and correctness is higher than the cost, they propose a generic two-party secure function evaluation protocol and prove that their protocol is optimal for some natural functions.
3. For any $\frac{1}{p}$ secure functions *f* (Gordon and Katz 2012; Beimel et al. 2011), they provide an attack-payoff secure protocol, which can be evaluated *f* according to attacker effectiveness.

Garay and Katz constructed a model based on simulation paradigm (Garay et al. 2013) according to UC framework. Their protocol and ideal function are similar to Canetti's synchronization model (Canetti 2000). If specific functions are valid to the protocols in Katz et al. (2013), the security of this model can be reduced to UC model. Pass and Halpern (2010) convert protocol in the presence of a number of parties with limited computing power into a game, which studies the relationship between traditional Cryptography security and the concepts of equilibrium in game theory. Aumann and Lindell (2007) consider the scenario where parties have incentives to cheat but still have small chance to be detected. They propose an efficient protocol. If the adversary get a negative utility when he abort the protocol, then models in Garay et al. (2013) and the frameworks in Pass and Halpern (2010) are the same.

3.2.2 The fairness of the rational multi-party computing

Nowadays, most rational secret sharing scheme and rational multi-party computing protocols discuss how to achieve fairness when all parties are rational (Halpern and Teague 2004; Kol and Naor 2008a, b; Gordon and Katz 2006; Gordon et al. 2011). There are also some works (Lepinski et al. 2004; Izmalkov et al. 2005; Lepinksi et al. 2005; Alwen et al. 2008, 2009, 2012; Izmalkov et al. 2008) achieve fairness by using powerful communication tools such as the physical envelopes and the ballot boxes etc.

Ong et al. (2009) (denoted as Ong Protocol) discussed the fairness in the presence of majority of rational parties and a minority of honest parties. They assume there are *k* honest parties, where *k* is much smaller than the secret share threshold *t* and the remaining are rational parties. In this case, Ong et al. put forward a simple protocol under simulcast channel, which achieves fairness with a high probability and satisfies with a stronger equilibrium–trembling hand perfect equilibrium.

Asharov et al. (2011) (denoted as Asharov protocol) discussed some basic requirements for secure multi-party computing: privacy, correctness and fairness. First of all, they give definitions of privacy and correctness in the framework of game theory and prove that privacy and correctness in Nash protocols are equivalent to those in game theory. Then they give the definition of fairness under the framework of game theory and the gradual release property of fairness and prove that the gradual release property are equal to fair computation. Finally they point out that their conclusions are not suitable for protocols, in which correctness is higher than $\frac{1}{2}$. This is the limitation of the Asharov protocol. Groce and Katz (2012) (denoted as Groce Protocol) present a fair protocol for any function in the presence of rational parties. They point out that the reason for Asharov protocol's limitation is due to the unfair definition. In order to avoid this limitation, Groce and Katz propose a new utility function, as shown in Table 3.

For a rational secure two-party computing, the line in Table 3 denote the utility of party P_0 and the column denotes the utility of party P_1. Correct means that parties have a correct output and incorrect means that parties have a wrong output. The first item in the bracket stands for utility of P_0 and the second is for P_1. Groce protocol achieve fairness by letting $b_0 > a_0 > d_0 > c_0$ and $b_1 \geq a_1 \geq d_1 \geq c_1$. According to this definition, the utility matrix of Asharov protocol is shown in Table 4:

As shown in Table 4, two parties have the same utility when both get correct output and false output. Therefore parties have no incentives to get correct output. It is obviously unfair. In other words, rational multi-party computing cannot achieve fairness is not due to itself but due to the inappropriate definitions of utility. Groce and Katz constructed a rational two-party computing protocol according to Table 3. They discussed fairness achievement under ideal and real world model, respectively.

The results of Groce protocol are different from Asharov protocol since they get positive conclusions about fairness. They prove that as long as the function in the ideal world reaches strictly Nash equilibrium, a fair rational protocol can be constructed with a Bayesian strict Nash equilibrium (BNE).

Wallrabenstein and Clifton made a further research about rational computation (denoted as WC protocol) on the base of Asharov protocol (Wallrabenstein and Clifton 2013). They pointed out that although Asharov et al. (2011) and Groce and Katz (2012) protocols define privacy, correctness and fairness toward the view of game theory and the latter protocol gets positive conclusions about fairness, their protocols only consider Bayesian strict Nash equilibrium under perfect information of extensional game. Besides, parties' behavior are limited by the action set of $\{\sigma^{continue}, \sigma^{abort}\}$. From this point, The non-credible threats in fail-stop is useless. Once the parties abort protocol, they will not be punished. In Groce protocol, parties' beliefs about the game state are designed outside. Bayesian strict Nash equilibrium can not dynamically describe the belief. Therefore they constructed a perfect Bayesian equilibrium rational protocol based on Asharov and Groce protocols. They allow parties to arbitrarily deviate from the protocol instead of only aborting the protocol. When parties adopt actions simultaneously, BNE and PBE reach the same equilibrium.

Table 3 Utility definition of the Groce protocol

P_iP_j	Correct	Incorrect
Correct	(a_0, a_1)	(b_0, c_1)
Incorrect	(c_0, b_1)	(d_0, d_1)

Table 4 Utility definition of the Asharov protocol

P_iP_j	Correct	Incorrect
Correct	(0, 0)	$(1, -1)$
Incorrect	$(-1, 1)$	(0, 0)

Domestic researches on fairness in rational computation have just started. Zhang and Cai (2012), Zhang and Liu (2013) analyzed the unfairness and instability in traditional secure two-party computing and build the game model and eventually realize fairness and correctness. Yilei Wang et al. constructed a complex rational two-party computing protocol under the incomplete information (Wang et al. 2015). Their protocol allows parties to have private types such that parties with different types have different utility functions. The utility definition derives from the store-chain game (Osborne and Rubinstein 1994). They proposed a strong equilibrium concept according to the complex protocol, which consists of two parts: computational sequential rationality and consistency. Finally they achieve fairness among two parties.

3.3 Rational Byzantine protocol

The concept of rational parties can not only be applied to rational secret sharing and rational secure multi-party computing, but also can be applied to other cryptographic protocol, such as Byzantine protocol. In Byzantine protocol, the generals must agree to whether to attack in order to resist foreign invasion. When all generals agree to attack, they will win. However, there may be a traitor among them. If the traitor confuses the decision, then no agreement will be reached. The main aim of Byzantine protocol is reach an attacking agreement even they know the existence of a traitor. Byzantine protocol is a model to describe such problem in the real world and Lamport first gives the description of the problem and solutions (Lamport et al. 1982).

Groce and Katz describe the adversary as rational in Byzantine protocol, which hopes the protocol reach a certain kind of preference (Groce et al. 2012). When they execute the protocol, they may deviate from the protocol and try to make the result of the protocol according to their preference. There are two types of parties in the Byzantine protocol of Groce and Katz (denoted as GroceKatz protocol): selfish corrupted parties and honest parties. Malicious adversaries are assumed to arbitrarily break the protocol while selfish corrupted parties break the protocol with certain incentives. In GroceKatz protocol, the utility functions are defined as follows:

$$U[protocol\, on\, 0] := u_0, U[protocol\, on\, 1] := u_1,$$

$$U[disprotocol] := u_2.$$

In the presence of rational parties, Groce and Katz get a conclusion different from previous Byzantine protocol. For example, (1) when $t \geq \frac{n}{2}$, they can not reach an agreement; (2) using the help of statistical security and computational security; (3) reduce the consistency to a broadcast. They also study the probability for rational Byzantine protocol when rational adversary has different preferences sequence. The Grocekatz protocol assumes that parties have two types: honest and rational parties. Similar rational Byzantine protocols include the BAR models in Aiyer et al. (2005), which discusses altruism and rational behaviors of Byzantine protocols in distributed systems (Li et al. 2006). Bei et al. (2012) studied collusion problems when all parties are rational (Clement et al. 2008) and discussed how to realize distributed consensus under crash failure and strategy manipulation in synchronous systems when all parties are assumed to be rational. They borrow the concept of collusion-resistant Nash equilibrium to resist crash failure and strategy manipulation. For a distributed system with n parties, they design a 2-resistance collusion protocol and $n-1$-resistance collusion random protocol. They also point out that if colluders are allowed to communicate though other communication channels, then no protocol can resist the two parties' collusion and one rash failure.

4 Conclusions and future works

Rational secure multi-party computing is a new research direction in the field of SMPC. With the development of intelligent computing and distributed computing, rational secure multi-party computing has a broad application fields. Starting from rational secret sharing scheme, RSMPC discuss how to reach various equilibrium under different communication channels and utility definitions. Recently, game theory is combined with SMPC to study how to realize fairness in multi-party computing. The future works include: asynchronous communication channels, the introduction of new types of parties, UC model of mixed rational multi-party computing protocol.

1. The problem of multi-party computing in asynchronous communication channel is equal to a dynamic game. Therefore we need to consider much complex equilibrium since Nash equilibrium cannot satisfy the new requirements. Further works include how to extend rational multi-party computing protocol to a sub-game perfect equilibrium under complete information, or sequential equilibrium under incomplete information. Meanwhile we should still consider the computational bound in cryptography.
2. The types of rational parties consist of honest parties, semi-honest parties, malicious adversaries, covert adversaries and rational parties. Further works include how to find new types of parties to make them more suitable for practical situations.
3. Previous works, discussing multi-party computing under the framework of game theory, assume that the priori information among parties is symmetric. However, it is not always the case. Similar to game theory, there also exists asymmetry information about parties in RSMPC. For example, rational parties may have a private type, the prior probability of this private type may be different. It is a hot topic in RSMPC to achieve security under incomplete information.
4. Fairness is highlighted in multi-party computing. Future works in this field are to realize complete fairness under mixed model in the presence of different kind of parties; how to realize relaxed fairness under a relaxed condition.
5. Utilize the notion of reputation game in rational multi-party computing, redefine utility for rational parties and reduce round complexity of rational multi-party computing. Furthermore, how to improve the efficiency of rational secure multi-party computing is the key point of future works.

In a word, the study on rational secure multi-party computing is still in its infancy. There are lots of problems to be discussed. Researchers continue to relate new knowledge in game theory to rational secure multi-party computing in order to promote the diversification development of RSMPC.

Acknowledgments This work is partially supported by National Natural Science Foundation of China (No. 61170161, 61202475), Guangzhou Zhujiang Science and Technology Future Fellow Fund (No. 2012J2200094), Outstanding Young Scientists Foundation Grant of Shandong Province (No. BS2014DX016), Nature Science Foundation of Shandong Province (No. ZR2012FQ029), Ph.D. Programs Foundation of Ludong University (No. LY2014033, LY2015033), Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund (Fujian Normal University) (No. 15004).

References

Abraham, Dolev D, Gonen R, Halpern J (2006) Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Proceedings of the 25th annual ACM symposium on principles of distributed computing, pp 53–62. ACM

Achim OM, Oracle TSBU, Bucharest R, Pop F, Cristea V (2011) Reputation based selection for services in cloud environments. In: NBiS 2011, pp 268–273. IEEE

Aiyer AS, Alvisi L, Clement A, Dahlin M, Martin JP, Porth C (2005) Bar fault tolerance for cooperative services. In: ACM SIGOPS operating systems review, vol 39, pp 45–58. ACM

Alfredo V, Ahmed ZF (2011) Cooperative fuzzy controllers for autonomous voltage regulation in smart grids. J Ambient Intell Hum Comput 2(1):1–10

Alwen J, Shelat A, Visconti I (2008) Collusion-free protocols in the mediated model. In: Advances in cryptology-CRYPTO 2008, pp 497–514. Springer

Alwen J, Katz J, Lindell Y, Persiano G, Shelat A, Visconti I (2009) Collusion-free multiparty computation in the mediated model. In: Advances in cryptology-CRYPTO 2009, pp 524–540. Springer

Alwen J, Katz J, Maurer U, Zikas V (2012) Collusion-preserving computation. In: Advances in cryptology-CRYPTO 2012, pp 124–143. Springer

Amato A, Beniamino DM, Venticinque S (2014) Agents based multi-criteria decision-aid. J Ambient Intell Hum Comput 5(5):747–758

Andreu J, Angelov P (2013) An evolving machine learning method for human activity recognition systems. J Ambient Intell Hum Comput 4(2):195–206

Asharov G, Lindell Y (2011) Utility dependence in correct and fair rational secret sharing. J Cryptol 24(1):157–202

Asharov G, Canetti R, Hazay C (2011) Towards a game theoretic view of secure computation. In: Advances in cryptology-EUROCRYPT 2011, pp 426–445. Springer

Aumann Y, Lindell Y (2007) Security against covert adversaries: efficient protocols for realistic adversaries. In: Theory of cryptography, pp 137–156. Springer

Bei X, Chen W, Zhang J (2012) Distributed consensus resilient to both crash failures and strategic manipulations. arXiv preprint arXiv:1203.4324

Beimel A, Lindell Y, Omri E, Orlov I (2011) 1/p-secure multiparty computation without honest majority and the best of both worlds. In: Advances in cryptology-CRYPTO 2011, pp 277–296. Springer

Cai Y, Peng X (2012) Rational secret sharing protocol with fairness. Chin J Electron 21(1):149–152

Canetti R (2000) Security and composition of multiparty cryptographic protocols. J Cryptol 13(1):143–202

Castiglione A, Pizzolante R, De Santis A, Carpentieri B, Castiglione A, Palmieri F (2015) Cloud-based adaptive compression and secure management services for 3d healthcare data. Future Gener Comput Syst 43:120–134

Chen X, Wu Q, Zhang F, Tian H, Wei B, Lee B, Lee H, Kim K (2011) New receipt-free voting scheme using double-trapdoor commitment. Inform Sci 181(8):1493–1502

Chen X, Li J, Ma J, Tang Q, Lou W (2012a) New algorithms for secure outsourcing of modular exponentiations. In: Computer security-ESORICS 2012, pp 541–556. Springer

Chen X, Li J, Susilo W (2012b) Efficient fair conditional payments for outsourcing computations. IEEE Trans Inform Forensics Secur 7(6):1687–1694

Clement A, Li H, Napper J, Martin JP, Alvisi L, Dahlin M (2008) Bar primer. In: Dependable systems and networks with FTCS and DCC, 2008. DSN 2008. IEEE International Conference on, pp 287–296. IEEE

Cleve R (1986) Limits on the security of coin flips when half the processors are faulty. In: Proceedings of the eighteenth annual ACM symposium on theory of computing, pp 364–369. ACM

Dodis Y, Halevi S, Rabin T (2000) A cryptographic solution to a game theoretic problem. In: Advances in cryptology°*TM* Crypto 2000, pp 112–130. Springer

Esposito C, Ficco M, Palmieri F, Castiglione A (2015) Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory. IEEE Trans Comput 99:1–14

Faiyaz D, Rahat I, Raouf NG (2014) A fuzzy ambient intelligent agents approach for monitoring disease progression of dementia patients. J Ambient Intell Hum Comput 5(1):147–158

Ficco M, Tasquier L, Aversa R (2014) Agent-based intrusion detection for federated clouds. In: INCoS 2014, pp 586–591. IEEE

Fuchsbauer G, Katz J, Naccache D (2010) Efficient rational secret sharing in standard communication networks. In: Theory of cryptography, pp 419–436. Springer

Garay J, Katz J, Maurer U, Tackmann B, Zikas V (2013) Rational protocol design: cryptography against incentive-driven adversaries. In: 2013 IEEE 54th annual symposium on foundations of computer science (FOCS), pp 648–657. IEEE

García Ó, Tapia DI, Alonso RS, Rodríguez S, Corchado JM (2012) Ambient intelligence and collaborative e-learning: a new definition model. J Ambient Intell Hum Comput 3(3):239–247

Gilboa I, Zemel E (1989) Nash and correlated equilibria: some complexity considerations. Games Econ Behav 1(1):80–93

Goldreich O, Micali S, Wigderson A (1987) How to play any mental game-a completeness therem for protocols with honest majority. In Proc. 19th ACM symposium on the theory of computing, pp 218–229

Gordon SD, Katz J (2006) Rational secret sharing, revisited. In: Security and cryptography for networks, pp 229–241. Springer

Gordon SD, Katz J (2012) Partial fairness in secure two-party computation. J cryptol 25(1):14–40

Gordon SD, Hazay C, Katz J, Lindell Y (2011) Complete fairness in secure two-party computation. J ACM (JACM) 58(6):24

Gradwohl R, Livne N, Rosen A (2013) Sequential rationality in cryptographic protocols. ACM Trans Econ Comput 1(1):2

Groce A, Katz J (2012) Fair computation with rational players. In: Advances in cryptology-EUROCRYPT 2012, pp 81–98. Springer

Groce A, Katz J, Thiruvengadam A, Zikas V (2012) Byzantine agreement with a rational adversary. In: Automata, languages, and programming, pp 561–572. Springer

Halpern J, Pass R (2008) Game theory with costly computation. arXiv preprint arXiv:0809.0024

Halpern J, Teague V (2004) Rational secret sharing and multiparty computation: extended abstract. In: STOC 2004: proceedings of the 36th annum ACM symposium on theory of computing, New York, USA: ACM, pp 623–632

Huang L, Su C (2006) Facial expression synthesis using manifold learning and belief propagation. Soft Comput 10(12):1193–1200

Isshiki T, Koichiro W, Tanaka K (2010) A rational secret-sharing scheme based on rsa-oaep. IIEICE Trans Fundam Electron Commun Comput Sci 93(1):42–49

Izmalkov S, Micali S, Lepinski M (2005) Rational secure computation and ideal mechanism design. In: Foundations of computer science, 2005. FOCS 2005. 46th Annual IEEE symposium on, pp 585–594. IEEE

Izmalkov S, Lepinski M, Micali S (2008) Verifiably secure devices. In: Theory of cryptography, pp 273–301. Springer

Katz J (2008) Bridging game theory and cryptography: recent results and future directions. In: Theory of cryptography, pp 251–272. Springer

Katz J, Maurer U, Tackmann B, Zikas V (2013) Universally composable synchronous computation. In: Theory of cryptography, pp 477–498. Springer

Kol G, Naor M (2008a) Cryptography and game theory: designing protocols for exchanging information. In: Theory of cryptography, pp 320–339. Springer

Kol G, Naor M (2008b) Games for exchanging information. In: Proceedings of the fortieth annual ACM symposium on theory of computing, pp 423–432. ACM

Lamport L, Shostak R, Pease M (1982) The Byzantine generals problem. ACM Trans Program Lang Syst (TOPLAS) 4(3):382–401

Lepinski M, Micali S, Peikert C, Shelat A (2004) Completely fair sfe and coalition-safe cheap talk. In: Proceedings of the twenty-third annual ACM symposium on principles of distributed computing, pp 1–10. ACM

Lepinksi M, Micali S, Shelat A (2005) Collusion-free protocols. In: Proceedings of the thirty-seventh annual ACM symposium on theory of computing, pp 543–552. ACM

Li HC, Clement A, Wong EL, Napper J, Roy I, Alvisi L, Dahlin M (2006) Bar gossip. In: Proceedings of the 7th symposium on operating systems design and implementation, pp 191–204. USENIX Association

Li J, Huang X, Li J, Chen X, Xiang Y (2014) Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 25(8):2201–2210

Luo Z, Cai Y, Yang Y (2012) Rational multi-secret sharing scheme based on bit commitment protocol. J Netw 7(4):738–745

Lysyanskaya A, Triandopoulos N (2006) Rationality and adversarial behavior in multi-party computation. In: Advances in cryptology-CRYPTO 2006, pp 180–197. Springer

Maleka S, Shareef A, Rangan CP (2008a) The deterministic protocol for rational secret sharing. In: Parallel and distributed processing, 2008. IPDPS 2008. IEEE international symposium on, pp 1–7. IEEE

Maleka S, Shareef A, Rangan CP (2008b) Rational secret sharing with repeated games. In: Information security practice and experience, pp 334–346. Springer

Micali S (2009) Purely rational secret sharing. In: Theory of cryptography, pp 54–71. Springer

Milgrom P, Roberts J (1986) Relying on the information of interested parties. RAND J Econ 17(1):18–32

Moses JR, William K, Rangan CP (2011) Rational secret sharing with honest players over an asynchronous channel. In: Advances in network security and applications, pp 414–426. Springer

Myerson RB (2013) Game theory. Harvard University Press, Cambridge

Nojoumian M, Stinson DR (2010) Brief announcement: secret sharing based on the social behaviors of players. In: Proceedings of the 29th ACM SIGACT-SIGOPS symposium on principles of distributed computing, pp 239–240. ACM

Nojoumian M, Stinson DR (2012) Socio-rational secret sharing as a new direction in rational cryptography. In: Decision and game theory for security, pp 18–37. Springer

Nojoumian M, Stinson DR, Grainger M (2010) Unconditionally secure social secret sharing scheme. IET Inform Secur 4(4):202–211

Ogiela MR, Ogiela U (2010) The use of mathematical linguistic methods in creating secret sharing threshold algorithms. Comput Math Appl 60(2):267–271

Ogiela MR, Ogiela U (2012) Linguistic protocols for secure information management and sharing. Comput Math Appl 63(2):564–572

Ong SJ, Parkes DC, Rosen A, Vadhan S (2009) Fairness with an honest minority and a rational majority. In: Theory of cryptography, pp 36–53. Springer

Osborne MJ, Rubinstein A (1994) A course in game theory. MIT press, Cambridge

Osborne M, Rubinstein A (2004) A course in game theory. MIT Press, Cambridge

Pass R, Halpern J (2010) Game theory with costly computation: formulation and application to protocol security. In: Proceedings of the behavioral and quantitative game theory: conference on future directions, p 89. ACM

Russell S, Norvig P, A. Intelligence (1995) A modern approach. Artificial Intelligence. Prentice-Hall, Eglnewood Cliffs, p 25

Serbanescu VN, Pop F, Cristea V, Achim OM (2012) Web services allocation guided by reputation in distributed soa-based environments. In: ISPDC 2012, pp 127–134. IEEE

Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

Shoham Y, Tennenholtz M (2005) Non-cooperative computation: Boolean functions with correctness and exclusivity. Theor Comput Sci 343(1):97–113

Tian Y, Ma J, Peng C, Chen X, Ji W (2011a) One-time rational secret sharing scheme based on bayesian game. Wuhan Univ\ J Nat Sci 16(5):430–434

Tian Y, Ma J, Peng C, Ji W (2011b) Game-theoretic analysis for the secret sharing scheme. Dianzi Xuebao (Acta Electronica Sinica) 39(12):2790–2795

Urbano A, Vila JE (2004) Computationally restricted unmediated talk under incomplete information. Econ Theory 23(2):283–320

Visan A, Pop F, Cristea V (2011) Decentralized trust management in peer-to-peer systems. In: ISPDC 2011, pp 232–239. IEEE

Wallrabenstein JR, Clifton C (2013) Equilibrium concepts for rational multiparty computation. In: Decision and game theory for security, pp 226–245. Springer

Wang Y, Liu Z, Wang H, Xu Q (2014) Social rational secure multi-party computation. Concur Comput Pract Exp 26(5):1067–1083

Wang Y, Wong DS, Zhao C, Xu Q (2015) Fair two-party computation with rational parties holding private types. Secur Commun Netw 8(2):284–297

William K, Moses JR, Rangan CP (2011) Rational secret sharing over an asynchronous broadcast channel with information theoretic security. arXiv preprint arXiv:1112.4033

Xu B, Peng Z, Xiao F, Gates AM, Yu JP (2014) Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. Soft Comput 1–9. doi:10.1007/s00500-014-1406-6

Yao AC (1982) Protocols for secure computations. In: 2013 IEEE 54th annual symposium on foundations of computer science, pp 160–164. IEEE

Zhang E, Cai Y (2010) A new rational secret sharing. China Commun 7(4):18–22

Zhang E, Cai Y (2012) A verifiable rational secret sharing scheme based on bilinear pairing [j]. Acta Electronica Sinica 40(5):1050–1054

Zhang Z, Liu M (2011) Unconditionally secure rational secret sharing in standard communication networks. In: Information security and cryptology-ICISC 2010, pp 355–369. Springer

Zhang Z, Liu M (2013) Rational secret sharing as extensive games. Sci China Inform Sci 56(3):1–13