International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(B)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 895–910

THE STUDY ON GENERAL SECURE MULTI-PARTY COMPUTATION

Yu-Fang Chung[1], Tzer-Long Chen[2], Chih-Sheng Chen[3] and Tzer-Shyong Chen[4]

[1]Department of Electrical Engineering
[3]Department of Statistics
[4]Department of Information Management
Tunghai University
No. 181, Sec. 3, Taichung-kang Road, Taichung 40704, Taiwan
{ yfchung; sschen; arden }@thu.edu.tw

[2]Department of Information Management
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan
d97725005@ntu.edu.tw

Received October 2010; revised February 2011

Abstract. *This paper, pertaining to the design of a new security model with multiparty computation for security, aims to study the rational model and UC (universal composability) model as well as design a security protocol for the requirements of the models. The structures of secret sharing agreement, fair computation agreement, bit analysis agreement and the applications of these agreements on security multiparty computation are investigated in the study. Moreover, the study also explores network application technology, such as electronic auction, electronic voting, encrypted data computation and threshold cryptology. This paper further analyzes the combination of rational model and security multiparty computation and proposes a new rational secret sharing method with two rational participants to improve previous methods. Based on $(m+n, t+1)$ threshold, a new $(m+n, t_1+t_2)$ threshold, utilizing the definition of specific permission secret sharing, is proposed and a respondent rational secret sharing protocol is constructed. This paper further studies the theories of UC model and proposes a UC security high-performance voting agreement, which is based on bilinear pairing and secret sharing, by replacing zero-knowledge proof with the new encoding method. Furthermore, this paper studies other applications of security multiparty protocol, such as threshold cryptology and electronic auctions, and constructs more secure agreements with higher performance.*

Keywords: Secure multi-party computation, Secret sharing, Rational model, Threshold cryptography, Electronic auction

1. **Research Background.** In this paper, we conduct a study on secure multi-party computation, including security models, basic agreement and application agreements.

(1). Security Model: rational model and the UC model, where ideas from game theory are used in the rational model.

(2). Basic Agreement: secret sharing, fair computation agreement and bit analysis agreement.

(3). Application Agreement: agreements on applications like threshold cryptography, electronic auction and electronic voting, etc.

The design of game theory model and cryptographic protocol is a study on the interaction between mutually distrustful parties. The designs of these two agreements belong to different fields of study, even if their interests differ, resulting in the development of their own independent fields of study. However, in order to build a more realistic interaction model, more and more researches are working on the integration of these two independent disciplines. In password models, participants can only be either honest or malicious. With the integration of game theory, participants can be both rational and selfish, which is closer to real-world situations. Thus, designing agreements, based on such premise, including researching the degree of influence on original agreements when participants change their strategy and the analysis of the resulting new models, have all become new directions of research.

In multi-party computation, the past methods did not consider the selection strategies of users and the actual requirements on applications. Besides, previous protocol development was not as mature as it is now. The selection and the practicability were not so favorable that it could not achieve the expected effect as well as presented worse integration. However, protocols have been developed that are mature, acceptable and well understood. The new combination of the proposed technique applies game theory as the basis, with user behaviors to choose the optimum dimensions for the users, analyze favorable strategies and establish new applications. Furthermore, with the development of network, it has become more difficult and complex to integrate the new applications and the development. Besides, in consideration of limited account, the past methods are no longer applicable. The proposed technique will benefit the selection and the integration application, enhance the overall performance of the new module and integrate the adaptive capacity.

The proposed technique has the following advantages.

(1). The increase of application range and the convenience. Past techniques used to be applied to single application in which the protocol and the technology were loaded individually. This proposed technique contains diverse protocols and based on the requirement and the actual environment, the communication and the settings could choose various protocols for applications. It will no longer need to load unnecessary protocols and can develop the application of self-protocol to increase the convenience of applications.

(2). High adaptability. Previous protocols were not so mature that the proposed module was limited in development, the applications were restricted and the adaptation problem existed. With mature protocols, the proposed technique could choose the developed protocol as the internal basis and apply the most suitable combination for the environment application, such as electronic auction and electronic voting.

(3). High security. Both cryptographic techniques and protocol development have presented breakthrough and modified several security loopholes in recent years. Both the protocol and the encryption in the proposed technique are well-developed. Having been scrutinized by the academia and the industry, the possible differences are analyzed.

(4). Consideration of users' behaviors. The proposed technique is based on game theory and the optimum method for users is considered as the selection strategy. Previous research used to consider cases that did not conform to practical strategies. Nevertheless, the proposed technique with multi-dimensional thinking presents rational judgment and situational considerations to conform to the actual requirements.

(5). Integration and compatibility. Previous modules did not follow the standard such that the integration and the compatibility were difficult. Since modules contain several protocols, the compatibility and the integration are considered out of question on applications. It is because that new modules integrate the applications of practical selections and diverse protocols, such as verification of identity integrated with electronic auction and voting or combined with other applications.

Halpern and Teague [1] studied secret sharing agreements under rational models. In that study, they defined the behavior of the rational participants. If the participants in a round of calculation can increase the participants' message, the participants will participate in that round of calculation, or else they will not. When there are three or more rational participants, they will give a random agreement during their participation of a secret sharing allowing the restoration of secret key under an expected number of computations, proving the agreement is Nash equilibrium of repeated elimination of comparatively poorer strategies. On this basis, Abraham, Dolev and Gonen [2] took into account conspiring participants and proposed a model which allowed agreement between k conspiring participants. Gordon and Katz [3] proposed a secret sharing agreement when there are only two rational participants and the need to reconfigure allocation after each round is eliminated. In the approach adopted by Lysyanskaya and Triandopoulos [4], it took into account the co-existence of rational and malicious participants and presented an agreement known as Mixed MPC Model. In Kol and Naor [5], the case of non synchronization was considered. In previous studies, rational secret sharing was not possible when there were only two participants, but no convincing proof had ever been given. Therefore, the construction of a rational bi-party secret sharing model [17,19] is still an open question. In addition, research on rational secret sharing with the special characteristics of rational secret sharing based on issuance of special permission is still very rare and is worth further study.

Given the diversity of network environments, a single agreement can no longer meet all needs; mixed uses of multiple agreements are often required. Therefore, it is important to ensure that the combination of independent safety protocols does not cause any safety consideration. In 2001, R. Canetti proposed the concept of UC security [6]. This method utilizes the advantage of modular design concept where the protocols can be designed independently. When a protocol meets UC safety standards, it is secure to operate with other protocols. When UC security was proposed, it immediately received a lot of attention to cryptography academics. In the design of a common model of security protocols, the UC model was used in the calculation of multi-party security [7,8]. Because security requirement was over idealized in the UC model, many agreements under such model could not be implemented. Therefore, additional conditions were considered and mixed protocols that met safety standards were constructed. For example, the researches in KR model and CRS model [9-12] have received a lot of attention. UC security protocol design has now almost become the highest standard for protocol security designs. However, the design of high-security protocols often leads to lower efficiency. Therefore, designing an efficient protocol with high security under the UC framework is an important issue.

In the field of cryptography, applications of secure multiparty computational protocols used to lag behind practical protocols, such as encryption and signature. With the rapid development of network computing environment and the applications of various distributed computing, secure multiparty computational protocols have caught up rapidly in many fields and applications. Among them, including the threshold password, electronic auction and the electronic voting are discussed in this research.

2. Research on Rational Model and Agreements Based on Them.

2.1. **Rational model research.** Game theory model can be divided into cooperative game and non-cooperative game. The distinction depends on whether or not the participants can reach a binding agreement. If a binding agreement can be reached it is called a cooperative game; otherwise, it is called a non-cooperative game. For example, if two oligopolistic firms reach an agreement to maximize monopoly profits and organize production in accordance to the agreement, they are participating in a cooperative game. The problem they mutually face is the sharing of mutual benefits as a result of the cooperation. On the contrary, if the agreement reached is non-binding, such that it is not mandatory for either party to abide by the agreement, where each party can choose their own optimal production or pricing individually, then they are participating in a non-cooperative game. Cooperative game theory values group rationality, that all is just, fair and efficient among all parties. Non-cooperative game theory functions on individual rationality, where each individual can make the best decision; output may be efficient, but it may also be inefficient. Non-cooperative game can be approached and divided into the following two perspectives:

(1). Division of time: The order in which the participants will take action is taken into consideration. The static game and the dynamic game are often considered. In the static game, participants can choose to take action at the same time or at different times. In the dynamic game, participants have in mind the order of action, where later participants can observe the actions taken by the previous participants, and accordingly, determine their own strategies.

(2). Division of information: This is further divided into two cases depending on whether the participants have or do not have complete knowledge of other participants' future strategy and payoff function. A complete-information game is the one where the participant has accurate information on other participants' feature strategy and payment function, while an incomplete-information game is the one where each participant has no accurate information on other participants.

Considering both the division of time and division of information, four different types of games can be obtained: complete-information static game, complete information dynamic game, incomplete-information static game and incomplete-information dynamic game. These four types of games correspond to the four equilibrium concepts, namely, Nash Equilibrium (Nash, 1950-51), Sub-game Perfect Nash Equilibrium (Selten, 1965) and Bayesian-Nash Equilibrium (Harsanyi, 1967-68), the Perfect Bayesian-Nash Equilibrium (Selten, 1975; Kreps and Wilson, 1982; Fudenberg and Tirole, 1991), as shown in Table 1.

Table 1. Types of game theory

Information time	Complete Information	Incomplete Information
Static	Complete-Information Static Game (Nash Equilibrium)	Incomplete-Information Static Game (Bayesian-Nash Equilibrium)
Dynamic	Complete-Information Dynamic Game (Sub-game Perfect Nash Equilibrium)	Incomplete-Information Dynamic Game (Perfect Bayesian-Nash Equilibrium)

There are other classification methods. According to the number of participants, games are divided into one-player game, two-player game, and multi-player game. According to the benefit of the game, games can be divided into zero-sum game, constant-sum game and non-zero-sum game. In a zero-sum game, the total benefit to all players will always amount to zero. In a constant-sum game, the total benefit to all players is a constant. The players' benefit is indeterminable in a non-zero-sum game.

2.2. **New rational secret-sharing agreement.** Halpern and Teague's model considered only two rational players and therefore could not prove secret-sharing. The secret-issuer can only deliver the share correctly and efficiently to the players, resulting in neither wanting to give his share to the other, since by sharing secret with the other player, the other party can obtain his private key, making it impossible to realize secret-sharing between two rational players. In the case of two rational players, we can realize secret-sharing by exploiting the uncertainty of a player in sharing a secret with the other player. A new secret-sharing scheme is proposed in the following section.

2.3. **Parameter setting.** Let p be a large prime number and let $K \neq 0$ be the private key to be shared by two players. The share issuer chooses a positive integer t and randomly selects $s_0, s_1, \ldots, s_t$ from Z_p which satisfy $K = s_0 + s_1 + \ldots + s_{t-1} + s_t \bmod p$. That is, the secret key K is hidden in the $t+1$ components $s_0, s_1, \ldots, s_t$. The secret issuer then constructs $t+1$ linear polynomials.

$$\begin{aligned}
&f_0(x) = s_0 + s_0' \bmod p,\ s_0' \in Z_p \\
&f_1(x) = s_1 + s_1' \bmod p,\ s_1' \in Z_p \\
&\ldots \\
&f_{t-1}(x) = s_{t-1} + s_{t-1}' \bmod p,\ s_{t-1}' \in Z_p \\
&f_t(x) = s_t + s_t' \bmod p,\ s_t' \in Z_p
\end{aligned}$$

The coefficients $s_0', s_1', \ldots, s_t'$ are randomly chosen. We notice that the kth Component s_k of K is the constant term of $f_k(x)$.

2.4. **Multi-round implementation of agreement.** The rules of implementation of the rounds are as follows:

At round k $(0 \leq k \leq t)$ the share issuer gives each player a share in the form of a point on the polynomial $f_k(x)$ in accordance with Shamir's secret sharing method. Upon receiving a share from the secret issuer; each player adopts the strategy which is to send his share to the other player. This signifies a successful execution of a round and both players have enough information to construct the polynomial $f_k(x)$ associated with that round and obtains the component s_k. If one of the players at any round refuses to give his share to the other player, then the secret sharing process terminates. After $t+1$ successful rounds, both players know the values of $s_0, s_1, \ldots, s_t$ and recover the secret key K by $K \equiv s_0 + s_1 + \ldots + s_t \bmod p$.

When the above said method is extended to n rational participants, a status bit can be set for each participant. If a participant does not receive the share from other participants, he is permanently withdrawn and implementation of the agreement terminated. If the share from other participants is received, the agreement will continue to run. Likewise, we can set t to prevent participants from deviating.

2.5. **New rational secret-sharing agreements based on issuance of permission.**

2.5.1. *Definition of secret-sharing agreements based on issuance of permission.* Let A and B be two sets of players with $|A| = m$, $|B| = n$ and $A \cap B = \Phi$. The secret issuer gives the ith player in set A a share k_i $(1 \leq i \leq m)$ and the jth player in set B a share $\overline{k_j}$ $(1 \leq j \leq n)$. Let $t \leq m$ be a positive number and let K be the private key to be shared by the players in sets A and B. The private key K can be recovered if t or more players in set A cooperate with a player in set B by broadcasting their shares. This secret sharing protocol is called the $(m+n, t+1)$ threshold method.

Extending the ideas of $(m+n, t+1)$ threshold method [18], we consider the $(m+n, t_1+t_2)$ threshold method. Let A and B be two sets of players with $|A| = m$, $|B| = n$ and $A \cap B = \Phi$. As in the $(m+n, t+1)$ threshold method, share k_i $(1 \le i \le m)$ and $\overline{k_j}$ $(1 \le j \le n)$ are given to players in A and B, respectively, by the share issuer. Let K be the private key to be shared by players in A and B. Let t_1 $(\le m)$ and t_2 $(\le n)$ be positive integers. The private key K can be recovered if t_1 or more players in set A cooperate with t_2 or more players in set B by broadcasting their shares. The private key cannot be recovered if either condition is not satisfied. This secret sharing protocol is called the $(m+n, t_1+t_2)$ threshold method.

In the original model, all players are rational and refuse to send their share to other players. The private key K thus cannot be recovered. We resolve this impasse by constructing a new secret sharing method based on the issuance of permissions.

2.5.2. *Secret-sharing based on special permissions.* Let A and B be two sets of players with $|A| = m$, $|B| = n$ and $A \cap B = \Phi$. Let p be a large prime number and let K be the private key to be shared by players in A and B. The share issuer selects two random integers S_A and S_B from Z_p such that $S_A + S_B \equiv K \bmod p$. In other words, the secret key K is broken into two parts S_A and S_B. S_A is to be recovered by players in A and S_B is to be recovered by B. Players in A and B then cooperate to recover the private key K by $S_A + S_B \equiv K \bmod p$.

Let t and t_1 $(\le m)$ be positive integers. The share issuer randomly selects $s_0, s_1, \ldots, s_t \in Z_p$ such that $S_A \equiv s_0 + s_1 + \ldots + s_t \bmod p$. That is, the part S_A is further divided into $t+1$ components.

The share issuer then constructs t random polynomials of degrees $t_1 - 1$.

$$
\begin{aligned}
&f_0(x) = s_0 + s_{0,1}x + \ldots + s_{0,t_1-1}x^{t_1-1} \bmod p,\ s_{0,j} \in Z_p,\ 1 \le j \le t_1 - 1\\
&f_1(x) = s_1 + s_{1,1}x + \ldots + s_{1,t_1-1}x^{t_1-1} \bmod p,\ s_{1,j} \in Z_p,\ 1 \le j \le t_1 - 1\\
&\ldots\\
&f_t(x) = s_t + s_{t,1}x + \ldots + s_{t,t_1-1}x^{t_1-1} \bmod p,\ s_{t,j} \in Z_p,\ 1 \le j \le t_1 - 1
\end{aligned}
$$

At round k $(0 \le k \le t)$, the share issuer selects a subset A_k of A with $|A_k| = t_1$, The share issuer then gives a share, which is a point of $f_k(x)$, to each player in A_k. The players in A_k broadcasting their shares to players in A. This constitutes a successful execution the kth round and every player in A can construct the polynomial $f_k(x)$ and recovers the component S_k.

If at any time any player who received a share form the share issue refuses to broadcast his share, the process terminates. After $t+1$ successful rounds, every player in set A learns $S_A \equiv s_0 + s_1 + \ldots + s_t \bmod p$.

Following the same process with parameters (t, t_2) instead of (t, t_1), the part S_B can be recovered by every player in B after $t+1$ successful rounds.

Players in A and players in B then cooperate to recover the private key K by $S_A + S_B \equiv K \bmod p$. We notice that it takes the cooperation of players in each set to recover a part and it takes the cooperation of player between A and B to recover the private key K.

Through the setting of t, the situation of participants deviating from the agreement can be avoided. Let P_{-i} denote the set of all participants except P_i. Assume that all participants in P_{-i} follow the agreement protocols. When a rational participant P_i deviates, there is $\frac{1}{t}$ probability of obtaining the private key and $\frac{t-1}{t}$ probability of not obtaining the private key. Let U^+, U and U^- be, respectively, the root mean square (RMS) value when P_i obtains the secret key and P_{-i} does not, and all participants obtain the secret key, and no participant obtains the secret key. According to the assumption of rationality, it can be verified that $U^+ > U > U^-$. If P_i follows the agreement, the

RMS value U is obtained. But if he deviates from the agreement, the RMS that can be obtained is $\frac{1}{t} * U^{+} + \frac{t-1}{t} * U^{-}$. That is to say, if the condition $U > \frac{1}{t} * U^{+} + \frac{t-1}{t} * U^{-}$ is satisfied, then participant P_i will not deviate from action. Therefore, by properly setting the value of parameter t, we can effectively prevent participants' deviation. Finally, as all the rational participants will abide by the agreement, the strategy will thus achieve Nash Equilibrium. According to Halpern and Teague [3], Theorem 3.2 can explain the strategy is the Nash Equilibrium retained after repeated elimination of inferior strategies.

3. Research on UC Model and Agreements Based on UC Model.

3.1. **UC model research.** Both the random oracle model and the standard model aim at a single agreement for analyzing and proving its safety and the agreement's implementation. Consideration is given only to security objectives and attacker's ability. In an increasingly complex network environment, the use of a single agreement can no long meet current needs. Thus, researches on safety models aiming at complex agreements, such as the UC models began to develop. The difference between how the random oracle model and standard model solve the problem of isolated security is that the UC model aims at combination of agreements to solve its security issues. Increasing evidence shows that UC safety agreements can be combined with UC framework and ensure overall safety.

UC model is certainly not completely independent from the random oracle model and the standard model. To prove that a protocol can safely realize the desired performance function, we must first define security goal, i.e., the desired ideal model performance function. A security performance function must be able to avoid attacks to prove its security. With regard to the security of a single agreement, random oracle model and standard model are current safe models that are relatively mature and provable. In addition, comparison of these three agreements' basic provable security shows that they are all the same in dealing with secret sharing.

There are two models in the UC framework: the real world model and the ideal process model. Participants implement the agreement protocol and ideal performance function in both models. However, participants in the ideal process model cannot interact directly, but can only interact through ideal performance function. In the implementation of the agreement protocols and the process of ideal performance function, participants are constantly activated and the information output is also constant.

In a real-life environment, participants implementing the agreement protocols often have different weights that need to be re-analyzed. In addition, participants in the UC framework do not consider the extent of future benefits in deciding whether to implement an agreement or not. With regard to these two types of UC models and the participants' implementation of agreement, they are analyzed and studied as follows.

3.2. **Research on participants' weight function.** In the UC framework, the weights for the participants are the same. However, in a real-life environment, not all participants have the same weight. For example: in a company's decision making process, manger and assistant manager have different voting right. Therefore, for agreements that involve multiple participants, it is necessary and meaningful to consider the role of each participant in a successful implementation of an agreement.

The models in the UC framework guarantee the realization of UC security by the agreement only when honest participants form the majority. Thus, when dishonest participants form the majority, the agreement does not have UC security. This implies that in the process of implementing the agreement, all participants have the same weights. However, in a real-life environment, consideration must be given to how different participants' implementation will have different effects. Therefore, we can assign a weight w_i to the participant P_i in the UC framework, where $0 < w_i < 1$ and $\sum_{i=0}^{n} w_i = 1$. When the sum of the weights of the dishonest participants is greater than $1/2$, the agreement does not meet UC security.

3.3. **Participant type research.** In the basic UC framework, all participants will implement agreements honestly. However, when the participants are invaded by attackers, the attackers can control the participants in order to implement the agreement maliciously. Thus, participants in the implementation of agreements can basically be divided into two categories: honest and dishonest. In game theory model, all participants are rational and selfish and want to realize maximum benefit. In order to develop an interaction model closest to reality and expand applications in cryptography, both disciplines have to be integrated by studying the behavior of rational participants through game theory. During the implementation of an agreement, participants will take into consideration their own benefits. If implementing the agreements can yield greater benefits, they will not honestly implement the agreement; otherwise, they will carry them out honestly.

Halpern and Teague's study, for the first time, introduced game theory into secret-sharing agreement and defined the behavior of rational participants. However, the problem of combining game theory and UC framework models has not yet been thoroughly researched.

3.4. **UC-secure electronic voting agreement of two candidates.** Assume that each voter can only vote for only one candidate in a one-out-of-two electronic voting system. Under the UC framework, a new type of coding method is employed that does not require the use of zero-knowledge to prove the legitimacy of the vote. By using bilinear pairing and identity-based (n, t) threshold method we introduce a method to realize a multi-candidate voting system with multiple counting centers for vote counting.

(1). Construct an ideal performance function for secret-sharing as shown in Figure 1.

Secret-Sharing F_{VSS}

There are l participants $P_1, \ldots, P_l$, an attacker S and a key distributor D. m is the number of participants who request for the private key to be revealed. The execution process of F_{VSS} is as follows:

1. When a participant receives the input value $(input, id, v)$ from D and activates the agreement, $(share, id)$ is sent to S. Upon receiving, the message $(input, id, \cdots)$ is ignored thereafter.
2. After participants accept the input value $(open, id)$ and activate the agreement, if $(input, id, v)$ is received from D, $(open, id, v)$ is then sent to all participants and attackers.

Figure 1. Secret-sharing F_{VSS}

(2). Construct an ideal electronic voting functionality (F_{Voting}) as shown in Figure 2. F_{Voting} accepts the votes from all voters, and calculates the final voting count.

When designing a more complex agreement, a more powerful tool is needed to realize the modular design. We first design UC security sub-agreements from the lowest level. We then utilize UC security to design each level. We then verify the UC security for the top-most level of the agreement, thereby guaranteeing the security of the whole system. In other words, designing complex agreements requires construction of simple

Ideal electronic voting functionality F_{Voting}

$V_1, V_2, \ldots, V_n$ are the n voters. C_1, C_2 are the candidates and HV is the verifier. S is the attacker and $Tallier$ is the counting center who also announces the voting results. F_{Voting} is executed as follows:

1. When a vote $(V_j, Sign(T_{V_j}), Sid, T_{V_j})$ is received from V_j, this vote is saved then $(V_j, Sign(T_{V_j}), Sid)$ is then sent to S and verifier HV. Any information sent later by V_j is ignored. (Sid is the only marker for the gap between the current connection and the next.)
2. When (V_j, Sid) is received from S, check if $(V_j, Sign(T_{V_j}))$ has been saved and decide if T_{V_j} can be counted as a valid vote. Ignore all messages sent later by S.
3. After $(result, Sid)$ is received from S, calculate the voting result C_{win}. Then, send $(result, Sid, C_{win})$ to and counting center $Tallier$.

Figure 2. Ideal electronic voting functionality F_{Voting}

agreement. Similarly, the ideal performance function of complex agreements also requires the construction of idea performance functions of complex agreements.

(3). Combining the two ideal performance functions mentioned above, we can obtain a common formal description of the electronic voting agreement, as shown in Figure 3.

Electronic Voting Agreement $\pi_{VOTING}^{F_{VSS}}$

There are n voters $V_1, V_2, \ldots, V_n$ and two candidates C_1 and C_2. HV is the verifier and S is the attacker. $T_1, T_2, \ldots, T_l$ are the vote counter and $Tallier$ is the counting center which also announces the final voting result. F_{VSS} is the ideal performance function of secret-sharing.

1. Activate and distribute the encrypted votes public key pk to all voters $V_1, V_2, \ldots, V_n$. Then, distribute the decrypted votes private key share sk_i to all vote counters $T_1, T_2, \ldots, T_l$.
2. Voter V_i uses the public key to encrypt his ballot v_i to obtain $C_i \leftarrow E_{pk}(Encode(v_i))$. Voter V_j the computes the signature σ_i of the encrypted ballet C_i and send (C_i, σ_i) to the verifier HV.
3. Upon receiving the pairs (C_j, σ_j), the verifier authenticates the legitimacy of all (C_j, σ_j). Then, all received pairs (C_j, σ_j) are mixed and sent to the electronic announcement board.
4. Upon receiving the pairs (C_j, σ_j) from the electronic announcement center, vote counter Ti decrypts the pairs (C_j, σ_j) using the private key ski and then send the decrypted result to the electronic announcement board.
5. $Tallier$ tallies the votes and announces the final results. (The final voting results can be tallied and verified by anyone.)

Figure 3. Electronic voting based on similar threshold $\pi_{VOTING}^{F_{VSS}}$

3.5. **Electronic voting method UC-EV.** Assume that there are n voters $V_1, V_2, \dots, V_n$, two candidates C_1 and C_2, a ballot verifier HV, l vote counters $T_1, T_2, \dots, T_l$ and an agreement attacker A.

(1). System preparation phase

An identity-based threshold encryption method for short ciphertext designed by Vanesa Daza is employed. The preparation phase is divided into the following steps:

Step 1: A trusted center chooses a security parameter k and generates a large prime number q. Let G_1 be a cyclic group generated by P and let G_2 be a group g order q. Let $e : G_1 \times G_1 \to G_2$ be a bilinear pairing.

Step 2: Choose a hash function $h : \{0,1\}^* \to Z_q$ and a random number $\gamma \in Z_q^*$. Calculate $P_1 = \gamma P$ and choose a random number $\rho \in Z_q^*$. The elements (q, G_1, G_2, P, e, h, P_1, Q, ρ) are included in the export of the agreement.

Step 3: Let ID_i be the identity of vote counter T_i. A hash function $g : \{0,1\}^* \to Z_q$ is used to created a hash value $a_i = g(ID_i)$ of ID_i. The trusted center calculates (PK_i, SK_i), where

$$PK_i = g(ID_i)$$
$$SK_i = \gamma PK_i$$

Step 4: Let PK_i be the public key of vote counter T_i $1 \leq i \leq l$. Generate an element P_2 by

$$P_2 = \sum_{V_i \in V} \lambda_{i0}^{T} PK_i$$

Step 5: Let R be the set of all participants and let R_i be the ith participant.

(2). Voting phase

Assume that each voter can vote for one and only candidate. A vote $M_{V_i} = \{b_1^i, b_2^i\}$ is generated with $b_j^i = 1$ if vote i votes for candidate j. Hence, (b_1, b_2) if a valid vote is and only if one component is 1 and the other component is 0. Associated with each valid vote, a vector $\{\rho^{b_1}, \rho^{b_2}\}$ is constructed, where the public parameters is generate at Step 3 of the system preparation phase.

Voter V_i generates his vote $M_{V_i} = \{b_1^i, b_2^i\}$ and completes the following steps:

Step 1: Generate a two-dimensional vector $v_i = (\rho^{b_1^i}, \rho^{b_2^i})$.

Step 2: Randomly choose two random numbers $s_1^i, s_2^i \in Z_q^*$ that satisfy the following condition: when $b_j^i = 0$ $(j = 1, 2)$, $\gcd(\rho^2, e(P_1, P_2)^{s_j^i}) = \rho$ and when $b_j^i = 1$, $\gcd(\rho, e(P_1, P_2)^{s_j^i}) = 1$.

Step 3: Calculate $S^i = \sum_{j=1}^{2} s_j^i$, $S^{i'} = \sum_{j=1}^{2} (s_j^i)^2$, $C_{1,j}^i = s_j^i P$ and $C_{2,j}^i = \rho^{b_j^i} e(P_1, P_2)^{s_j^i}$. Next, sign $C_{1,j}^i$ with $Sign_{SK_i}(C_{1,j}^i, C_{2,j}^i) \to \sigma_j^i$. Finally, voter V_i sends his vote $T_{V_i} = \{(C_{1,1}^i, C_{2,1}^i, \sigma_1^i), (C_{1,2}^i, C_{2,2}^i, \sigma_2^i), S^i\}$ to HV.

(3). Verification and ballot-mixing phase conducted by HV

The verification process is divided into two steps:

Step 1: Verify the legitimacy of the identity of voter V_i by verifying the signature $\sigma^i = (\sigma_1^i, \sigma_2^i)$.

Step 2: Verify the legitimacy of the vote by checking if $\gcd(\rho^2,\ C_{2.j}^i) = \rho$. If it is not satisfied, then the vote is not valid because the voting system permits each voter to vote for only one candidate.

The Ballot-Mixing process:

When HV receives the vote $T_{V_i} = \{(C^i_{1,1}, C^i_{2,1}, \sigma^i_1), (C^i_{1,2}, C^i_{2,2}, \sigma^i_2), S^i\}$, two random numbers $s'^i_1, s'^i_2 \in Z^*_q$ are generated. HV then calculates $C'^i_{1,j} = C^i_{1,j} + s'^i_j P$ and $C'^i_{2,j} = C^i_{2,j} \times e(P_1, P_2)^{s'^i_j}$ and sends the result $\{(C'^i_{1,1}, C'^i_{2,1}), (C'^i_{1,2}, C'^i_{2,2})\}$ to the electronic announcement board.

(4). Tallying phase

After vote counter T_i receives the votes from the electronic announcement board, the following vote matrix is generated:

$$\begin{pmatrix} (s'^1_1 P, \rho^{-b^1_1} e(P_1, P_2)^{s'^1_1}) & (s'^1_2 P, \rho^{-b^1_2} e(P_1, P_2)^{s'^1_2}) \\ (s'^2_1 P, \rho^{-b^2_1} e(P_1, P_2)^{s'^2_1}) & (s'^2_2 P, \rho^{-b^2_2} e(P_1, P_2)^{s'^2_2}) \\ \vdots & \vdots \\ (s'^n_1 P, \rho^{-b^n_1} e(P_1, P_2)^{s'^n_1}) & (s'^n_2 P, \rho^{-b^n_2} e(P_1, P_2)^{s'^n_2}) \end{pmatrix}$$

Step 1: Vote counter T_i computes $(k = 1, 2)$

$$C_{1,k} = \sum_{i=1}^{n} s'^i_k P, \text{ and}$$

$$C_{2,k} = \prod_{i=1}^{n} \rho^{b^i_k} e(P_1, P_2)^{s'^i_k} = \rho^{\sum\limits_{i=1}^{n} b^i_k} e(P_1, P_2)^{\sum\limits_{i=1}^{n} s'^i_k}$$

$$\text{Let } S_k = \sum_{i=1}^{n} s'^i_k. \text{ Then, } C_{1,k} = S_k P \text{ and } C_{2,k} = \rho^{\sum\limits_{i=1}^{n} b^i_k} e(P_1, P_2)^{S_k}.$$

Step 2: Vote counter T_i calculates $d^j_k = \frac{1}{e(C_{1,k}, SK_j)}$, and sends a m dimensional vector $v_j' = (d^j_1, d^j_2, \ldots, d^j_m)$ to the announcement board.

Tallying Results:

Vote counter T_j receives n vectors v_i' $(i = 1, 2, \ldots, n)$ and calculates D_k (the number of votes candidate k received). The tallying process can be carried out and verified by anyone.

4. **Threshold Cryptography Research.**

We now consider the problem of secure multiparty computation agreement of threshold cryptographic applications, and propose a new undeniable identity-based threshold proxy signature scheme and an identity-based threshold ring signature scheme.

4.1. **New undeniable identity-based threshold proxy signature scheme.**

We improve upon the method of HLL (Hindustan Lever Limited), but use the same system of parameters. The construction of the new method is as follows:

(1). Generation of the proxy signature secret sharing share

Step 1: Initial parameter generation:

Every proxy signer $P_i \in G$ randomly chooses a secret parameter $k_i \in Z_q$ and calculates $r_i = g^{k_i} \bmod p$. r_i is broadcasted to all other proxies. Therefore, every P_i can compute the value R, where

$$R = \prod_{j=1}^{n} r_j \bmod p.$$

Step 2: Group parameter generation

Every proxy signer $P_i \in G$ randomly chooses a secret polynomial $f_i(x) = x_i h(R) + k_i + a_{i,1} x + \cdots + a_{i,t-1} x^{t-1} \bmod q$, where $a_{i,1}$, $a_{i,2}, \ldots, a_{i,t-1}$ are random numbers in Z_q. signer P_i obtains values $f_j(i)$ from signer P_j, for all $1 \leq j \leq$

$n, j \neq i$. Therefore, signer P_i can obtain $s_i = f(i) = f_1(i) + f_2(i) + \cdots + f_n(i) \equiv h(R)\sum_{j=1}^{n} x_j + \sum_{j=1}^{n} k_j + a_1 i + a_2 i^2 + \cdots + a_{t-1} i^{t-1} \bmod q$, where $a_j = \sum_{i=1}^{n} a_{i,j} \bmod q$. The proxy signer group makes public $y_G \equiv \prod_{i=1}^{n} g^{x_i} \equiv \prod_{i=1}^{n} y_i \bmod p$ and $A_j \equiv g^{a_j} \bmod p$; $j = 1, 2, \ldots, t-1$.

Step 3: Generation of proxy key

The original signer chooses a random value k and calculates parameter $K \equiv g^k \bmod p$. He then calculates a proxy signature key $\sigma = ex_0 + kK \bmod q$, when $e = h(m_w, K)$.

Step 4: Sharing of proxy signature key

The original signer distributes proxy signature key σ using the (n, t) threshold method to share proxy signature σ among proxy signers. He generates a secret polynomial $f'(x) = \sigma + b_1 x + b_2 x^2 + \cdots + b_{t-1} x^{t-1} \bmod q$, where $b_j \in Z_q$. He then sends $\sigma_i = f'(i)$ to proxy signers P_i $(i = 1, 2, \ldots, n)$ through a secure channel and broadcasts $B_j = g^{b_j} \bmod p$ and (m_w, K) to group G.

Step 5: Generation of proxy signature key share

After σ_i is received, proxy signer $P_i \in G$ uses the equation $g^{\sigma_i} \equiv y_0^{h(m_w,K)} K^K \prod_{j=1}^{t-1} B_j^{i^j} \bmod p$, to verify the legitimacy of (σ_i, m_w, K). If the equation holds, P_i calculates ${\sigma_i}' = \sigma_i + s_i \cdot h(m_w, K) \bmod q$, where σ_i' is the proxy signature key share of P_i.

(2). Generation of proxy signature:

As described above, in order to simplify the generation of the symbols during proxy signature generation, the participants $P_1, P_2, \ldots, P_t$ collectively create a proxy signer P_0 to sign a message m. Assume that $D = \{P_1, P_2, \ldots, P_t\}$, where every P_i generates a proxy signature for the message m through the following steps:

Step 1: P_i randomly chooses a secret value $k_i' \in Z_q$ and calculates $r_i' = g^{k_i'} \bmod p$. Next, broadcast r_i' to proxy signer group D. Thus, P_i can obtain the value $R' \equiv \prod_{i=1}^{t} r_i' \bmod p$.

Step 2: This step is same as Step 1 in the HLL method for proxy signature generation. However, we replace the original polynomial with a new polynomials $f_i''(x) \equiv x_i h(R') + k_i' + c_{i,1} x + \cdots + c_{i,t-1} x^{t-1} \bmod q$. After receiving $f_j''(i)$ from all P_j $(j \neq i)$, P_i obtains $s_i' = f''(x) = f_1''(i) + f_2''(i) + \cdots + f_t''(i) \equiv h(R') \sum_{j=1}^{t} x_j + \sum_{j=1}^{t} k_j' + c_1 i + \cdots + c_{t-1} i^{t-1} \bmod q$. Finally, make public $C_j \equiv g^{c_j} \bmod p$ $(j = 1, 2, \ldots, t-1)$.

Step 3: P_i calculates $\gamma_i = s_i' + \sigma_i' h(m) \bmod q$ and sends γ_i to proxy signer P_j, $j = 1, 2, \ldots, t$, $j \neq i$.

Step 4: After receiving γ_j $(j \neq i)$, P_j can verify the legitimacy of γ_j by checking.

$$g^{\gamma_j} \equiv R' \left(\prod_{i=1}^{t-1} C_i^{j^i} \right) \left(\prod_{i=1}^{t-1} y_i \right)^{h(R')} \times \left[\left(y_0^{h(m_w,K)} K^K \prod_{i=1}^{t-1} B_i^{j^i} \right) \left(y_G^{h(R)} R \prod_{i=1}^{t-1} A_i^{j^i} \right)^{h(m_w,K)} \right]^{h(\mathrm{ASID},m)} \bmod p$$

Step 5: P_i uses Lagrange interpolation equation on γ_i to calculate $T \equiv f''(0) + (f(0) h(m_w, K) + f'(0)) h(\mathrm{ASID}, m) \bmod q$, the proxy signature of m is $(m, T, K, R', R, m_w, \mathrm{ASID})$.

(3). Verification of proxy signature

The verifier can verify the legitimacy of proxy signature (m, T, K, R', R, m_w, ASID) using the following steps:

Step 1: With m_w and ASID, the verifier can differentiate the identity of the original signer and proxy signer. The verifier then can obtain the proxy signer's public key from the CA (Certificate Authority) center.

Step 2: The verifier can verify proxy signature using the following equation:

$$g^T \equiv \left[\left[y_0 R\left(\prod_{i=1}^{n} y_i\right)^{h(R)}\right]^{h(m_w,K)} K^K\right]^{h(\mathrm{ASID},m)} \times R'\left(\prod_{P_i\in\mathrm{ASID}}^{y_i}\right)^{h(R')} \bmod p$$

If the equation holds, the proxy signature (m, T, K, R', R, m_w, ASID) of m is legitimate.

4.2. **New identity-based threshold ring signature scheme.** In this method, all users' keys are in a group G, where the order n of the group is a product of two large primes p and q which are kept secret. The method is described as follows:

(1). Construction phase

Construct a group G of order $n = pq$ as described in 4.3 (2) and randomly choose elements $u', u_1, \ldots, u_k \in G$. A trusted entity (TA) randomly chooses $a, b_0 \in Z_n$ and calculates $A = g^a$, $B_0 = g^{b_0}$ and $\hat{A} = h^a$. Finally, choose two hash functions $H : \{0,1\}^* \to G$ and $H_1 : \{0,1\}^* \to \{0,1\}^k$. H maps the user's identity to the group G and H_1 maps the user's message to a bit string of length k.

The public system parameters consist of the group G and the subgroup generated by the elements g and h. $\left(A, B_0, \hat{A}\right)$ and $(u', u_1, \ldots, u_k)$ are public parameters. b_o and the factorization of n kept are secret. a is the private key of TA and A is the corresponding public key. Anyone can verify whether $\left(A, \hat{A}\right)$ has been generated correctly.

(2). Key generation

User (signer) sends identity $ID \in \{0,1\}^*$ to trusted entity (TA). TA calculates user's public key pk as $H(ID)$ and the corresponding private key sk as $H(ID)^a$. TA then sends the private key over a secure channel then to user.

(3). Signature algorithm Sig(L, R, M)

Let $ID_1, ID_2, \ldots, ID_n$ be the identities of n users. Assume that the set $\{ID_1, ID_2, \ldots, ID_t\}$ is a ring. Here ID_i is used as the private key of user i. t of the n users cooperate to produce a ring signature for the message M. Without loss of generality, assume that $\{ID_1, ID_2, \ldots, ID_t\}$ is the set of the identities of the signers and $\{ID_{t+1}, \ldots, ID_n\}$ is the set of identities of non-signers.

Step 1: There are t signers and anyone of them can represent the whole group of t signers to sign a message. Using to user identity ID_i of ring R, the corresponding public key $pk_i = H(ID_i)$ of group G can be calculated. Define numbers f_i $(1 \le i \le n)$ by

$$f_i = \begin{cases} 1 & i = 1, \ldots, t \\ 0 & i = t+1, \ldots, n \end{cases}$$

For $i = 1, \ldots, n$, choose an integer $x_i \in Z_n$ randomly and compute $C_i = (pk_i/B_0)^{f_i}\, h^{x_i}$ and $\pi_i = \left((pk_i/B_0)^{2f_i-1}\, h^{x_i}\right)^{x_i}$. If $C = \prod_{i=1}^{n} C_i$, then $B_0^t C = h^x \prod_{i=1}^{t} pk_i$. Thus, C must act as the ciphertext of participating signer's public key.

Step 2: For $i = 1, \ldots, n$, participating signer i first calculates $(m_1, \ldots, m_k) = H_1(M, R)$. Participating signer i then randomly chooses $r_i \in Z_n$, uses private key sk_i to calculate $S_{1i} = sk_i \cdot \left(u' \prod_{j=1}^{k} u_j^{m_j}\right)^{r_i}$ and computes $S_{2i} = g^{r_i}$. Participating signer i then sends (S_{1i}, S_{2i}) to one of the signer in the group of t participating signers.

Step 3: After receiving all the participating signers' (S_{1i}, S_{2i}), the signer sets $x = \sum_{i=1}^{n} x_i$, and calculates $S_1 \leftarrow \hat{A}^x \prod_{i=1}^{t} S_{1i}$ and $S_2 \leftarrow \prod_{i=1}^{t} S_{2i}$. Finally, the encrypted message of M obtained using the (n, t) threshold signature σ of ring R is $((S_1, S_2), \{(C_i\, , \pi_i)\}_{i=1}^{n}) \in G^{2n+2}$.

(4). Verification algorithm

Upon receiving message M and (n, t) threshold signature σ of ring R, the verifier calculates $(m_1, \ldots, m_k) = H_1(M, R)$, and breaks down signature σ into S_1, S_2 and $\{(C_i, \pi_i)\}_{i=1}^{n}$. Next, with the identity ID_i of the ring members, derive the corresponding public key $pk_i = H\left(ID_i\right)$, $i = 1, \ldots, n$.

First check for the legitimacy of π_i: for $i = 1, \ldots, n$, the verifier examines if $e\left(c_i, {}^{c_i}\!/\!\left({}^{pk_i}\!/_{B_0}\right)\right) = e\left(h, \pi_i\right)$. If one of the equations does not hold the verifier rejects the signature. Otherwise, let $C = \prod_{i=1}^{n} C_i$ and check the following equation:

$$e\left(A, B_0^t C\right) = e(S_1, g) \cdot e\left(S_2^{-1}, u' \prod_{j=1}^{k} {u_j}^{m_j}\right)$$

The verifier accents the signature if above equation holds.

4.3. **Secure multiparty agreements in other applications.**

(1). Electronic auction system design based on new group signature

Using identity-based and bilinear mapping group signature methods, we now consider a secure and revocable-registration electronic auction system [20,21].

An auction process involves the bidders, an auction manager (AM) and a verifying center (VC). The bidders place their bids on the item to be auctioned off. The auction manager conducts the auction process, verifies bidders' qualification, issues bidders' identity certificated, accepts bids, and awards the winning bid. The verifying center sends system parameters and assists the auction manager in determining the winning bid.

(2). Initialization of verifying center VC

Suppose that the auction system can support up to 2^k users and that signature message $\in \{0, 1\}^m$, where k, m are polynomial-related parameters. Let $n = p \cdot q$, where p and q are randomly chosen large prime numbers. Let G be a group of order $n = p \cdot q$. Let G_p and G_q be subgroups of G of order p and q, respectively. Let $g \in G$, $h \in G_q$ and $\alpha \in Z_n$ be a random integer.

Choose identity-related generating elements $u', u_1, \ldots, u_k \in G$, $v', v_1, \ldots, v_m \in G$. Make public bilinear group parameter (n, G, G_T, e) and other public parameters $PP = (g, h, u', u_1, \ldots, u_k, v', v_1, \ldots, v_m, A = e(g, g)^{\alpha}) \in G \times G_q \times G^{k+m+2} \times G_T$.

(3). Registering bidder B

After bidder B requests to place his bid to AM, AM verifies the qualification of bidder B. If it passes, bidder B receives a random identity ID, where $0 \le ID < 2^k$. Using the main key MK and ID, bidder B picks a random number $s \in Z_n$ and calculates his own private key $K_{ID} = (K_1, K_2, K_3) = (g^{\alpha} \cdot (u' \prod\limits_{i=1}^{k} u_i^{k_i})^s, g^{-s}, h^s) \in G^3$.

(4). Auction

Let $M = (\mu_1 \cdots \mu_m \in \{0,1\}^m)$ be the bidding information. Bidder B chooses random numbers $t_1, \dots, t_k \in Z_n$ and computes.

$$c_i = u_i^{k_i} \cdot h^{t_i}$$
$$\pi_i = (u_i^{2k_i - 1} \cdot h^{t_i})^{t_i}, \ i = 1, 2, \dots, k.$$

Define $t = \sum\limits_{i=1}^{k} t_i$, $c = u' \prod_{i=1}^{k} c_i = (u' \prod_{i=1}^{k} u_i{}^{k_i}) \cdot h^t$ and let $V = v' \prod_{i=1}^{m} v_i^{\mu_i}$. Choose two random number $\overline{s_1}, s_2 \in Z_n$ and generate the following three parameters:

$$\sigma_1 = K_1 \cdot K_3^t \cdot c^{\overline{s_1}} \cdot V^{s_2}$$
$$\sigma_2 = K_2 \cdot g^{-\overline{s}_1}$$
$$\sigma_3 = g^{-s_2}$$

Let $s_1 = \overline{s_1} + s$, where s is the private key of B.

The signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, c_1, \dots, c_k, \pi_1, \dots, \pi_k) \in G^{2k+3}$ is generated. Bidder B sends V, the signature σ and bidding information M to AM. When AM receives the message, he carries out the following two verification process to verify the identity legitimacy of bidder B:

1). Calculate $c = u' \prod\limits_{i=1}^{k} c_i$, $\forall i = 1, \dots, k$ and verify whether $e(c_i, u_i^{-1} c_i) = e(h, \pi_i)$ holds.

2). Check to see whether $e(\sigma_1, g) \cdot e(\sigma_2, c) \cdot e(\sigma_3, V) = A$ holds.

If both equations hold, then AM accepts the bid as a valid bid.

(5). Ending the auction

AM makes public the signature σ of the bid winner and bidding information M. Legitimacy of the bid winner's identity can be verified by anyone.

Using the signature σ, the verifying center can recover the ID of the bid winner. Let $k_i = 0$, if $(c_i)^q = g^0$ and $k_i = 1$ otherwise. The identity of the bid winner is given by $(k_1, k_2 \dots, k_k) \in \{0,1\}^k$. VC sends the bid winner's identity ID to AM. The bid winner can use ID to carry out the payment process.

5. **Conclusion.** Regarding secure multi-party computation, it contains several types of protocols, such as secret sharing and fair computation. Those protocols used to be proposed for single application without analyses on possible rational behaviors of users to choose the optimum method for practical applications. With game theory, the proposed technique utilizes the most practical behaviors of the user to analyze the optimum choice. Besides, in consideration of the possible events, the optimum protocol is proposed to conform to the actual requirements and applications. The practical and rational methods contain diverse protocols that the new module is further designed, including the improvement of the past HLL problem. The proposed new idea is applied to undeniable proxy signature and the identity authentication in electronic auctions to achieve the requirements of practical application and the security.

The aim of this paper to study the secure multiparty computation agreement, especially the design of security agreements that satisfy the needs of two secure multiparty computation models: rational model and UC model. This paper also addresses secure multiparty computation basic agreement, including sub-agreements on secret sharing, fairness calculation, bit decomposition and their applications in the design of secure multiparty computation agreements. In addition, this paper also addresses various applications on secure multiparty computation agreements, such as electronic auction, electronic voting, encrypted data computation and threshold cryptography. The theoretical model of secure model computation and basic agreements and their related applications will remain a focus of research. The sharing property of secure multiparty computation will be used to design a more general secure multiparty computational agreement.

Acknowledgment. This work was supported partially by National Science Council of Taiwan under Grants NSC 100-2410-H-029-007.

REFERENCES

[1] J. Halpern and V. Teague, Rational secret sharing and multiparty computation: Extended abstract, *Annual ACM Symposium on Theory of Computing*, pp.623-632, 2004.

[2] I. Abraham, D. Dolev, R. Gonen and J. Halpern, Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation, *Proc. of the 25th ACM Symposium on Principles of Distributed Computing*, pp.53-62, 2005.

[3] S. D. Gordon and J. Katz, Rational secret sharing, revisited, *Security and Cryptography for Networks*, pp.229-241, 2006.

[4] A. Lysyanskaya and N. Triandopoulos, Rationality and adversarial behavior in multiparty computation, *CRYPTO, LNCS*, vol.4117, pp.180-197, 2006.

[5] G. Kol and M. Naor, Cryptography and game theory: Designing protocols for exchanging information, *TCC, LNCS*, pp.320-339, 2008.

[6] R. Canetti, Universally composable security: A new paradigm for cryptographic protocols, Extended, *abstract in 42nd FOCS*, 2001. A revised version (2005) is available at IACR Eprint Archive, eprint.iacr.org/2000/067/.

[7] R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai, Universally composable two-party and multiparty secure computation, *Proc. of the 34th Annual ACM symposium on Theory of Computing*, New York, NY, USA, pp.494-503, 2002.

[8] Y. Lindell, General composition and universal composability in secure multi-party computation, *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp.394-403, 2003.

[9] R. Canetti and T. Rabin, Universal composition with joint state, *Crypto, LNCS*, vol.2729, pp.265-281, 2003.

[10] B. Barak, R. Canetti, J. B. Nielsen and R. Pass, Universally composable protocols with relaxed setup assumptions, *Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pp.186-195, 2004.

[11] J. Katz, Universally composable multi-party computation using tamper-proof hardware, *Eurocrypt, LNCS*, pp.115-128, 2007.

[12] R. Canetti, Y. Dodis, R. Pass and S. Walfish, Universally composable security with pre-existing setup, *TCC, LNCS*, 2007.

[13] R. Canetti and M. Fischlin, Universally composable commitments, *Crypto, LNCS*, vol.2139, pp.19-40, 2001.

[14] R. Canetti and H. Krawczyk, Universally composable notions of key exchange and secure channels, *Eurocrypt, LNCS*, vol.2332, pp.337-351, 2002.

[15] D. Hofheinz, J. Muller-Quade and D. Unruh, Universally composable zero-knowledge arguments and commitments from signature cards, *Tatra Mountains Mathematical Publications*, 2005.

[16] K. Gjosteen and L. Krakmo, Universally composable signcryption, *EuroPKI, LNCS*, vol.4582, pp.346-353, 2007.

[17] C.-T. Li and M.-S. Hwang, An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards, *International Journal of Innovative Computing, Information and Control*, vol.6, no.5, pp.2181-2188, 2010.

[18] T.-Y. Lin, T.-C. Wu, C.-I. Lee and T.-S. Wu, A (t, n)-fair dynamic threshold secret sharing scheme, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1395-1406, 2009.

[19] H. Zhao, M. Li, K. Sakurai and Y. Ren, Mechanized analysis of verifiable multi-secret sharing in the applied pi-calculus, *ICIC Express Letters*, vol.4, no.3(B), pp.1053-1058, 2010.

[20] C. Rao, Y. Zhao and Q. Wang, Uniform price auctions of divisible goods with variable supply, *ICIC Express Letters*, vol.4, no.4, pp.1127-1134, 2010.

[21] C.-Y. Lee, H.-D. Tsui and Y.-C. Tai, Job scheduling of retrieving dynamic pages from online auction websites on grid architecture, *ICIC Express Letters*, vol.4, no.5(B), pp.2015-2019, 2010.