An Information-Theoretic Approach for Secure Protocol Composition

Yi-Ting Chiang[1(✉)], Tsan-Sheng Hsu[1], Churn-Jung Liau[1], Yun-Ching Liu[4], Chih-Hao Shen[2], Da-Wei Wang[1], and Justin Zhan[3]

[1] Academia Sinica, Taipei 11529, Taiwan
{ytchiang,tshsu,liaucj,wdw}@iis.sinica.edu.tw
[2] University of Virginia, Charlottesville, VA 22904, USA
shench@gmail.com
[3] North Carolina Agricultural and Technical State University, Greensboro, NC 27411, USA
justinzzhan@gmail.com
[4] University of Tokyo, Tokyo 113-8654, Japan
cipherman@gmail.com

Abstract. Privacy protection has become a crucial issue in the information era. In recent years, many protocols have been developed to accomplish computational tasks collaboratively without revealing the participants' private data. However, developing protocols for each individual application would not be practical. The more natural and efficient approach would be utilizing basic protocols as building blocks for the construction of complex protocol.

In this paper, we proposed the concept of t-certified protocols, which are protocols that are secure when t parties are under the influence of a semi-honest adversary. A composition theorem is given to specify the conditions for secure composition of t-certified protocols, and a framework for constructing complex protocols is developed.

We have adopted an information theoretical approach, and believe that it will be a viable alternative to the classic simulator approach, which is based on the concept of indistinguishability between the ideal model and the real model.

Keywords: Privacy-preserving computation · Secure multiparty computation · Protocol composition

1 Introduction

With the advancements in network and storage technology, massive databases are distributed all over the Internet, and methods for performing collaborative computational tasks between these databases while retaining privacy has gained a great deal of attention in recent years.

The concept of secure two-party computation was proposed by Yao [1] and extended to the multi-party case by Goldreich et al. [2]. It was shown that the

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 405–423, 2015.
DOI: 10.1007/978-3-319-23829-6_28

secure computation of general computable functions is theoretically possible, and protocols for computing fundamental operations has also been proposed, such as Yao's garbled circuit [2]. Currently, the most adopted approach for computing complex functions is by combining several secure protocols together, but the composition of protocols was shown to be not necessarily secure [3]. Methods for secure composition of protocols have been proposed and extensively investigated [4–8]. One example is the Protocol Composition Logic (PCL), which is a logic-based method. The PCL can be applied to prove security properties of network protocols [9], supporting compositional reasoning on both parallel and sequential composition of protocols [10]. Although these methods provides a formal foundation for the security verification of the composition of protocols, the process is rather complex, and hard to apply in practice.

In this paper, we will focus on sequential composition [11], which is the scenario where each new execution begins immediately after the previous one terminates. We proposed the concept of *t-certified protocols*, which are protocols that are information theoretically secure against a semi-honest adversary [12,25] whom controls t parties in an n-party secure computation, where $t < n$. We have identified a set of preconditions and a general method for the secure composition of t-certified protocols. This allows us to develop a framework for constructing secure protocols for computing complex functions by utilizing t-certified protocols as building blocks.

Our framework is under the assumption of sequential composition, and can simplify the complex task of security verification significantly. An information theoretical approach has been adopted in the development of our framework, and we believe it will be a viable alternative to the classic simulator approach, which is based on the concept of indistinguishability between the ideal model and the real model.

In Sect. 2, we will describe the proposed framework for privacy-preserving collaborative computation protocols. We will give a demonstration to our framework in Sect. 3, and some concluding remarks in Sect. 4.

2 A Composition Framework

In this section, we propose a composition framework for secure multi-party computation protocols. First, we consider a set of definitions and basic properties in information theory. Then, we present an information theory paradigm and composition model.

2.1 Definitions

We use the following widely accepted definitions throughout the paper.

Definition 1. *Random variables X, Y, and Z are said to form a* Markov chain, *denoted by $X \rightarrow Y \rightarrow Z$, if the conditional distribution of Z only depends on Y*

and is conditionally independent of X. Specifically, X, Y, and Z form a Markov chain if the joint probability can be written as

$$\Pr(X, Y, Z) = \Pr(X) \Pr(Y|X) \Pr(Z|Y).$$

That is, the random variables X, Y, and Z are said to form a Markov chain if and only if X and Z are conditionally independent given Y.

When protocols are developed, it is inevitable that participants will keep records of historical data that they could use to their advantage. In the simulation paradigm, the historical data is taken into consideration and modelled as auxiliary inputs. Because in Markov chain, given the current state, knowledge of the previous states is irrelevant for predicting the subsequent states. Markov property plays a crucial role in our information-theoretical paradigm preventing history from interfering with current execution after protocol composition.

Definition 2 (Functionality). An n-ary functionality $F(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ is a function that maps n inputs to n outputs stochastically, whereas ordinary functions that map inputs to outputs uniquely are deemed deterministic functionalities.[12]

Functionalities are randomized extensions of ordinary functions. A functionality F may be regarded as a probability distribution over functions such that F equals the function f_i with probability $P(i)$. There are two steps in evaluating $F(x_1, \ldots, x_n)$: tossing coins to decide an index i, and then evaluating the function $f_i(x_1, \ldots, x_n)$.

Definition 3. A protocol Π realizes the n-ary functionality $F(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ if n parties follow the steps in Π such that party i inputs x_i and receives y_i at the end of the execution.

Privacy and correctness are two fundamental requirements in multi-party computation research. The privacy requirement stipulates that only necessary information should be revealed, while the correctness requirement ensures the accuracy of the protocol outputs. In the remainder of this paper, a protocol that realizes a functionality is described as theoretically correct instead of computationally indistinguishable.

Definition 4 (Information-Theoretically Secure Protocol). Let Π be a multi-party protocol, x_i be the private input of party i, and X be the collection of all the private inputs, i.e. $X = (x_1, \ldots, x_n)$. Party i's view during an execution of Π with input X, denoted by $\text{VIEW}_i^{\Pi}(X)$, *is* (x_i, r_i, m_i), where r_i is the internal coin tosses, and m_i represents all the received messages. The protocol is said to be information-theoretically secure if party i does not have more information about X after the execution than before it; that is,

$$I(X; \text{VIEW}_i^{\Pi}(X)) = I(X; x_i), \ i = 1, \ldots, n,$$

where $I(A;B)$ is the mutual information shared by random variables A and B [13].

Therefore, no information about the secret inputs held by the participants are revealed by their local view after executing a function, which is realized by a information-theoretically secure protocol. Hence, we can assure that the privacy of these participants are preserved.

2.2 Preliminary Theory

Because of the finite nature of real-world applications, numbers are often considered in finite fields, denoted by $GF(p)$, where p is a large enough prime. When designing a secure protocol, it is intuitive to add a random number in order to hide secrets. The following lemmas demonstrate that, in a finite field, the addition of random numbers is intuitively appealing and also protects private data completely from the perspective of information theory. This masking property is very helpful in protocol design and security analysis.

Lemma 1. *Let X and R be random variables defined on $GF(p)$. If R is uniformly distributed and independent of X, then $(X+R)$ follows a uniform distribution over $GF(p)$.*

Proof. In a finite field, both negation and the addition of a constant are bijective operations. Specifically, the sequence $(i_0, i_1, \ldots, i_{(p1)})$, for all $i \in GF(p)$, is a permutation of $(0, 1, \ldots, p-1)$. As a result, we have

$$\begin{aligned}
&Pr(X+R=i)\\
=&\sum\nolimits_k Pr(X=k, R=i-k)\\
=&\sum\nolimits_k Pr(X=k)\cdot Pr(R=i-k|X=k)\\
=&\sum\nolimits_k Pr(X=k)\cdot Pr(R=i-k)\\
=&\sum\nolimits_k Pr(X=k)\cdot \frac{1}{p} = \frac{1}{p},
\end{aligned}$$

which proves the lemma.

Moreover, in a finite field, an independent uniform random variable R is so powerful that, no matter how the random variable X is distributed, $(X+R)$ will follow a uniform distribution whose entropy (uncertainty) is maximal.

Lemma 2. *Let X and R be random variables defined on $GF(p)$, and let Y be a random variable defined on another field. If R is uniformly distributed and independent of the joint distribution (X,Y), $(X+R)$ is independent of Y.*

Proof. From the proof of Lemma 1, we know that, for $i \in GF(p)$, the conditional probability is

$$\begin{aligned}
&\Pr(X+R=i|Y)\\
=&\sum\nolimits_k \Pr(X=k, R=i-k|Y)\\
=&\sum\nolimits_k \Pr(X=k|Y)\cdot\Pr(R=i-k|X=k,Y)\\
=&\sum\nolimits_k \Pr(X=k|Y)\cdot\Pr(R=i-k)\\
=&\sum\nolimits_k \Pr(X=k|Y)\cdot\frac{1}{p}=\frac{1}{p}=\Pr(X+R=i),
\end{aligned}$$

which proves the independence of $(X+R)$ and Y.

Lemmas 1 and 2 state that the masked variable $(X+R)$ is maximally uncertain and disconnected from Y. From the perspective of protocol design, adding independent random numbers to outgoing messages guarantees the security of the message, and ensures that the messages do not reveal other private information.

Lemma 3. *Let $X_1,\ldots,X_n$, and R be random variables defined on $GF(p)$. If R follows a uniform distribution and is independent of $(X_1,\ldots,X_n)$, then we have* $\Pr(X_1|X_2,\ldots,X_{n-1},X_n+R)=\Pr(X_1|X_2,\ldots,X_{n-1})$.

Proof. Based on the assumption that R is independent of $(X_1,\ldots,X_n)$ and Lemma 2, we know that X_n+R is independent of $(X_1,\ldots,X_{n-1})$.

Finally, we generalize the idea of masked variables and present one of the most useful results in the following theorem:

Theorem 1. *Let $X_1,\ldots,X_n$, and R be random variables defined on $GF(p)$, and let $Y_1,\ldots,Y_m$ are arbitrary functions of $X_1,\ldots,X_n$. If R is uniformly distributed and independent of $(X_1,\ldots,X_n)$, we have*[1]

$$I(Y_1;Y_2,\ldots,Y_{m-1},Y_m+R)=I(Y_1;Y_2,\ldots,Y_{m-1})$$

In addition, $\Pr(Y_1|Y_2,\ldots,Y_{m-1},Y_m+R)=\Pr(Y_1|Y_2,\ldots,Y_{m-1})$*; i.e.,* $H(Y_1|Y_2,\ldots,Y_{m-1},Y_m+R)=H(Y_1|Y_2,\ldots,Y_{m-1})$.

Proof. Since R is uniformly distributed and independent of $(X_1,\ldots,X_n)$, by definition and the above lemmas, we have

$$\begin{aligned}
&I(X_1,\ldots,X_n;R)=0\\
\Rightarrow&I(Y_1,\ldots,Y_{m-1},Y_m;R)=0\\
\Rightarrow&I(Y_1,\ldots,Y_{m-1};Y_m+R)=0 &\text{(Lemma 2)}\\
\Rightarrow&I(Y_2,\ldots,Y_{m-1};Y_m+R)=0.
\end{aligned}$$

[1] Occasionally, Y_m could be an empty function so that the following equation also holds:

$$I(Y_1;Y_2,\ldots,Y_{m-1},R)=I(Y_1;Y_2,\ldots,Y_{m-1}).$$

Moreover, the above results show that

$$\begin{aligned} &I(Y_1; Y_m + R | Y_2, \ldots, Y_{m-1}) \\ =&I(Y_1, \ldots, Y_{m-1}; Y_m + R) - I(Y_2, \ldots, Y_{m-1}; Y_m + R) = 0. \end{aligned}$$

Therefore, we conclude that

$$\begin{aligned} &I(Y_1; Y_2, \ldots Y_{m-1}, Y_m + R) \\ =&I(Y_1; Y_2, \ldots, Y_{m-1}) + I(Y_1; Y_m + R | Y_2, \ldots, Y_{m-1}) \\ =&I(Y_1; Y_2, \ldots, Y_{m-1}) \end{aligned}$$

Finally, it is known that if R is independent of $(X_1, \ldots, X_n)$, then it is also independent of $(Y_1, \ldots, Y_m)$. By combining this result with Lemma 3, we have $Pr(Y_1 | Y_2, \ldots, Y_{m-1}, Y_m + R) = Pr(Y_1 | Y_2, \ldots, Y_{m-1})$, which completes the proof.

Eliminating redundancy helps us analyse the security of multi-party protocols, especially when there is a great deal of unnecessary information, and the redundancy includes the outputs of a function under the presence of the inputs. Furthermore, Theorem 1 states that information masked by independent, uniform random variables is also redundant and can be removed.

2.3 An Information-Theoretical Paradigm

Most studies use the simulator paradigm is to prove the security of protocols. Specifically, a simulator generates an adversary's view in the ideal model that is indistinguishable from the adversary's view in the real model [12]. Canetti proposed a widely accepted design methodology for secure protocols [11]. The steps are as follows:

1. Design a "high-level" protocol for the given functionality under the assumption that some primitive functionalities can be computed securely.
2. Design secure primitive protocols to realize the primitive functionalities.
3. Construct a composite protocol that realizes the given functionality by incorporating the primitive protocols as subroutines into the "high-level" protocol.

The composite protocol is only provably secure when the high-level protocol and the primitive protocols are provably secure in the hybrid model and the real model respectively. The methodology is elegant and allows us to design a large-scale protocol in a recursive manner as follows. When a primitive protocol is proved to be secure as the boundary condition in the recursion, each proof of the security of a high-level protocol that results in a secure composite protocol can be used as another secure primitive to construct a "higher-level" protocol.

To measure security, our approach uses information theory instead of indistinguishability. Next, we define an adversary's ability and propose a definition of protocol security.

Definition 5. *An adversary is t-limited if it can select up to t parties to control. In addition, as an adversary starts to control a party, it can learn the view of this party has until now.*

Definition 6. *Let Π be an n-party protocol that realizes an n-ary functionality $f(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ and let X be the distribution of all parties' private inputs, i.e., $X = (x_1, \ldots, x_n)$. The view of party i during an execution of Π with input X, denoted by $\text{VIEW}_i^{\Pi}(X)$, is (x_i, r_i, m_i, y_i), where r_i is the internal coin tosses, and m_i is the received messages. The protocol is said to be* t-certificated *if it is secure against a t-limited semi-honest adversary. Specifically, the protocol must satisfy the following criteria.*

C1. The internal coin tosses r_i are generated independently.
C2. The protocol operations depend solely on the inputs and internal coin tosses; that is, $(m_1, \ldots, m_n, y_1, \ldots, y_n)$ is a function of $(X, r_1, \ldots, r_n)$.
C3. The adversary does not gain information about X with every possible collusion; that is, for all $I \subset \{1, \ldots, n\}$, and $|I| \leq t$,

$$I(X; \text{VIEW}_I^{\Pi}(X)) = I(X; X_I),$$

where X_I and $\text{VIEW}_I^{\Pi}(X)$ denote the joint inputs and views of collusive parties.

C3 describes protocol security in terms of information theory. C1 and C2 may appear to be unnecessary as they are implied when designing protocols in the stand alone model. However, they are crucial because they ensure the security of the designed protocols. Note that Definition 6 is feasible for Canetti's method, but with a slight modification. Specifically, if there is no communication between the participants in a high-level protocol, our main theorem (Theorem 2) claims that the certification against a t-limited adversary is closed under composition; that is, a protocol composed of t-certificated primitive protocols remains t-certificated. The closure property reduces the effort required to design a large-scale system. Once the primitive protocols are proved to be certificated, the resulting composite protocol is provably certificated without extra burdens. This allows protocol designers to focus on developing more efficient high-level protocols.

2.4 Composition Model

Before presenting our main theorem, we formally define the composition model. Recall that the model is actually a composite protocol constructed by Canetti's methodology with the condition that no communication is allowed in the high-level protocol.

Let Π be an m-round n-party protocol constructed by the modified methodology, and let $X = (x_1, \ldots, x_n)$. Then, protocol Π can be modelled as follows.

1. Party i starts with private input x_i and sets $z_i^0 \leftarrow x_i$.
2. Party i sets $\text{VIEW}_i^{\Pi,0}(X) \leftarrow (x_i, z_i^0)$.
3. Initialize the round number: $l \leftarrow 1$.
4. Repeat while $l \leq m$:
   (a) Party i sets $x_i^l \leftarrow z_i^{l-1}$.

(b) A subset of the parties, k^l parties, collaboratively execute a certificated protocol ρ^l so that
  - Party i, who participates in ρ^l, receives random coin tosses r_i^l, communicated messages m_i^l, and the protocol output y_i^l.
  - Party j, who does not participate in ρ^l, sets $r_j^l \leftarrow m_j^l \leftarrow y_j^l \leftarrow x_j^l$.

(c) Party i locally produces independent coin tosses s_i^l, and sets z_i^l to be a function of own knowledge, i.e. $z_i^l \leftarrow f_i^l(\text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, y_i^l, s_i^l)$.

(d) Party i sets

$$\text{VIEW}_i^{\Pi,l}(X) = (\text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, y_i^l, s_i^l, z_i^l).$$

(e) $l \leftarrow l + 1$.

5. Party i sets $y_i \leftarrow z_i^m$ as the output and halts.

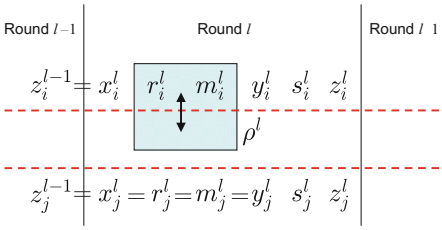

Fig. 1. A summary of the l-th round of protocol Π

Figure 1 summarizes round l. The x-axis represents the time line from left (round $l-1$) to right (round $l+1$). Party i participates in the certificated protocol ρ^l, but party j does not. In the execution of ρ^l, party i has random coin tosses r_i^l, received message m_i^l, and the output y_i^l. In addition, s_i and s_j are locally generated coin tosses; while z_i and z_j are, respectively, functions of party i's knowledge and party j's knowledge up to round l. Recall that communication is only allowed in the execution of ρ^l.

It makes sense to model party j, who does not participate in ρ^l, by assigning r_j^l, m_j^l, and y_j^l to x_j^l. Party j is not allowed to communicate with other parties in round l; thus, his actions can be modelled by local random coin tosses, s_j^l, and private computation, z_j^l.

Theorem 2 (Sequential Composition Theorem). *Given secure channels, if the primitive protocols ρ^l are t-certificated, then the composite protocol Π is t-certificated.*

Proof. First, we outline the proof and show that the theorem is sound when $t = 1$. The proof is divided into three steps.

1. Address a crucial Markov property introduced by the composition model and the proposed security definition.

2. Normalize the k^l-party 1-certificated protocol, ρ^l, to derive another n-party 1-certificated protocol, ϕ^l.
3. Prove by mathematical induction that n-party 1-certification is closed under the proposed composition model.

In our information-theoretical paradigm, the Markovity discussed in Lemma 4 is crucial because it acts as a bridge between primitive protocols. Next, by regarding an adversary who colludes with t parties, we can generalize the closure property from 1-certification to t-certification.

For simplicity, let X_ρ^l, R_ρ^l, M_ρ^l, and Y_ρ^l denote the joint distribution of all parties' inputs, coin tosses, received messages, and outputs in the execution of ρ^l. Similarly, let $\text{VIEW}^{\Pi,l}(X)$ be the joint distribution of the views of all parties in round l of protocol Π.

Lemma 4. *The joint historical information and outputs of the current round are conditionally independent given the input of the current round. That is,* $\text{VIEW}^{\Pi,l-1}(X)$, x_i^l, *and* (r_i^l, m_i^l, y_i^l) *form a Markov chain, for* $l = 1, \ldots, m$.

Proof. For party i, who participates in ρ^l, we know that

$$\begin{aligned} & I(\text{VIEW}^{\Pi,l-1}(X); R_\rho^l, M_\rho^l, Y_\rho^l | X_\rho^l) \\ =& I(\text{VIEW}^{\Pi,l-1}(X); R_\rho^l | X_\rho^l) + I(\text{VIEW}^{\Pi,l-1}(X); M_\rho^l, Y_\rho^l | X_\rho^l, R_\rho^l) \qquad \text{(C2)} \\ =& I(\text{VIEW}^{\Pi,l-1}(X); R_\rho^l | X_\rho^l) = 0. \end{aligned}$$

Note that because R_ρ^l is generated independently in ρ^l after $\text{VIEW}^{\Pi,l-1}(X)$ and X_ρ^l have been computed, it must be independent of $(\text{VIEW}^{\Pi,l-1}(X), X_\rho^l)$. From the above result, we know that $I(\text{VIEW}^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | X_\rho^l) = 0$. In addition,

$$\begin{aligned} & I(\text{VIEW}^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | X_\rho^l) \\ =& I(\text{VIEW}^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | x_i^l, X_\rho^l) && (x_i^l \text{ is part of } X_\rho^l) \\ =& I(\text{VIEW}^{\Pi,l-1}(X), X_\rho^l; r_i^l, m_i^l, y_i^l | x_i^l) - I(X_\rho^l; r_i^l, m_i^l, y_i^l | x_i^l) \\ =& I(\text{VIEW}^{\Pi,l-1}(X), X_\rho^l; r_i^l, m_i^l, y_i^l | x_i^l) && \text{(C3)} \\ \Rightarrow& I(\text{VIEW}^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | x_i^l) = 0. \end{aligned}$$

For party j, who does not participate in ρ^l, we prove the Markov property as follows:

$$\begin{aligned} & I(\text{VIEW}^{\Pi,l-1}(X); r_j^l, m_j^l, y_j^l | x_j^l) \\ =& I(\text{VIEW}^{\Pi,l-1}(X); x_j^l | x_j^l) = 0. && (x_j^l = r_j^l = m_j^l = y_j^l) \end{aligned}$$

Lemma 5. *Step (4b) in the composition model normalizes the* k^l*-party 1-certificated protocol* ρ^l *into an* n*-party 1-certificated protocol* ϕ^l.

Proof. Let ϕ^l be an extension of ρ^l that is executed collaboratively by all parties instead of the original k^l parties. For simplicity, let X_ϕ^l, R_ϕ^l, M_ϕ^l, and Y_ϕ^l denote,

respectively, the inputs, coin tosses, messages, and outputs of all participants in the execution of ϕ^l. We have to prove that ϕ^l satisfies the following conditions of Definition 6:

[C1] By the assumption that ρ^l is a certificated protocol, we know that R_ρ^l is generated independently. In addition, $R_\phi^l = R_\rho^l$, ϕ^l satisfies this condition.

[C2] As ρ^l is assumed to be 1-certificated, there exists a function f_ρ such that $f_\rho(X_\rho^l, R_\rho^l) = (M_\rho^l, Y_\rho^l)$. It is trivial to construct a function f_ϕ that exploits f_ρ as a subroutine and outputs $y_j^l = x_j^l$ for party j, who does not participate in ρ^l.

[C3] Given $I(X_\rho^l; \text{VIEW}_i^{\rho^l}(X_\rho^l)) = I(X_\rho^l; x_i)$, we need to prove that $I(X_\phi^l; \text{VIEW}_i^{\phi^l}(X_\phi^l)) = I(X_\phi^l; x_i)$, for $i = 1, \ldots, n$. For party i, who participates in ρ^l, it holds that

$$\begin{aligned} I(X_\phi^l; \text{VIEW}_i^{\phi^l}(X_\phi^l)) =& I(X_\phi^l; x_i^l, r_i^l, m_i^l, y_i^l) \\ =& I(X_\phi^l; x_i^l) + I(X_\phi^l; r_i^l, m_i^l, y_i^l | x_i^l) \\ =& I(X_\phi^l; x_i^l). \qquad \text{(Lemma 4)} \end{aligned}$$

Because $x_i^l = z_i^{l-1} \subset \text{VIEW}^{\Pi, l-1}(X)$, we know that $X_\phi^l = (x_1^l, \ldots, x_n^l)$ must be a subset of $\text{VIEW}^{\Pi, l-1}(X)$; we can apply Lemma 4 to this proof.
For party j, who does not participate in ρ^l, we have

$$\begin{aligned} I(X_\phi^l; \text{VIEW}_j^{\phi^l}(X_\phi^l)) =& I(X_\phi^l; x_j^l, r_j^l, m_j^l, y_j^l) \\ =& I(X_\phi^l; x_j^l). \qquad (x_j^l = r_j^l = m_j^l = y_j^l) \end{aligned}$$

Lemma 6. *The n-party protocol Π comprised of n-party 1-certificated protocols $\phi^1, \ldots, \phi^m$ is also 1-certificated.*

Proof. (C1) and (C2) will be proved under the assumption of semi-honest adversaries, and (C3) will be proved by mathematical induction. Initially,

$$I(X; \text{VIEW}_i^{\Pi,0}(X)) = I(X; x_i, z_i^0) = I(X; x_i). \qquad (x_i = z_i^0)$$

Next, we consider round l,

$$\begin{aligned} & I(X; \text{VIEW}_i^{\Pi,l}(X)) \\ =& I(X; \text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, y_i^l, s_i^l, z_i^l) \\ =& I(X; \text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, y_i^l, s_i^l) \\ =& I(X; \text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, y_i^l) \qquad \text{(Theorem 1)} \\ =& I(X; \text{VIEW}_i^{\Pi,l-1}(X)) + I(X; x_i^l, r_i^l, m_i^l, y_i^l | \text{VIEW}_i^{\Pi,l-1}(X)). \end{aligned}$$

The following result,

$$\begin{aligned}
&I(X; x_i^l, r_i^l, m_i^l, y_i^l | \text{VIEW}_i^{\Pi,l-1}(X)) \\
=&I(X; r_i^l, m_i^l, y_i^l | \text{VIEW}_i^{\Pi,l-1}(X), x_i^l) && (x_i^l \in \text{VIEW}_i^{\Pi,l-1}(X)) \\
=&I(X, \text{VIEW}_i^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | x_i^l) - I(\text{VIEW}_i^{\Pi,l-1}(X); r_i^l, m_i^l, y_i^l | x_i^l) \\
=&0, && (X, \text{VIEW}_i^{\Pi,l-1}(X) \subset \text{VIEW}^{\Pi,l-1}(X), \text{Lemma 4})
\end{aligned}$$

shows that

$$I(X; \text{VIEW}_i^{\Pi,l}(X)) = I(X; \text{VIEW}_i^{\Pi,l-1}(X)).$$

In other words, given that Π is 1-certificated in round $l-1$, we know that it is 1-certificated in round l. The mathematical induction completes the proof.

Before presenting the last lemma, we have to construct a new protocol ω^l. Recall that we convert the k^l-party protocol ρ^l into an n-party protocol ϕ^l in Lemma 5 by assuming that parties that do not participate in ρ^l take part in ϕ^l and only output their inputs. Here, we construct the protocol ω^l from protocol ϕ^l and a collusion set C whose size is at most t. All collusive parties C in protocol ϕ^l are regarded as a single adversary A in protocol ω^l; that is, ϕ^l is an n-party protocol, whereas ω^l is an $(n - |C| + 1)$-party protocol. If protocol ϕ^l can be certificated against every set C, protocol ω^l can be certificated against the corresponding party A; thus, the protocol comprised of ω^l is 1-certificated because of Lemmas 5 and 6. As a result, the protocol Π comprised of ϕ^l is certificated against C, and is therefore t-certificated.

Lemma 7. *If the protocol ρ^l in the composition model is t-certificated, the protocol ϕ^l is also t-certificated.*

Proof. Recall that ϕ^l is an extension of ρ^l derived by increasing the number of participants from k^l to n. For semi-honest adversaries, conditions (C1) and (C2) are trivial, so we focus on (C3). From the assumption that ρ^l is t-certificated, we know that

$$I(X_\rho^l; \text{VIEW}_S^{\rho^l}) = I(X_\rho^l; X_S), \forall S \in \{1, \ldots, n\}, |S| \leq t.$$

Next, for every subset $C \in \{1, \ldots, n\}$, $|C| \leq t$, there are three possible relations between the collusive parties and the participants in ρ^l, denoted by P. Specifically, the collusive parties C may be part of, disjoint from, or overlap with P. We consider each scenario below.

1. $(C \cap P = \varnothing)$ In this case, the parties in C do not participate in ρ^l and only output their input during the execution of protocol ϕ^l. Trivially, condition (C3) holds that

$$I(X_\phi^l; \text{VIEW}_C^{\phi^l}) = I(X_\phi^l; X_C^l).$$

2. $(C \subset P)$ Because every collusive party participates in ρ^l, condition (C3) in this case is guaranteed by the t-certification of protocol ρ^l.

3. ($C \cap P \neq \varnothing$) In this case, some of the collusive parties participate in ρ^l. Again, the Markov property described in Lemma 4 is demonstrated here:

$$\begin{aligned}
& I(X_\phi^l; \text{VIEW}_C^{\phi^l}) \\
=& I(X_\phi^l; X_{C-P}^l, \text{VIEW}_{C\cap P}^{\phi^l}) \\
=& I(X_\phi^l; X_{C-P}^l, X_{C\cap P}^l, R_{C\cap P}^l, M_{C\cap P}^l, Y_{C\cap P}^l) \\
=& I(X_\phi^l; X_C^l, R_{C\cap P}^l) + I(X_\phi^l; M_{C\cap P}^l, Y_{C\cap P}^l | X_C^l, R_{C\cap P}^l) \\
=& I(X_\phi^l; X_C^l, R_{C\cap P}^l) \qquad \text{(C2)} \\
=& I(X_\phi^l; X_C^l). \qquad \text{(C1)}
\end{aligned}$$

In all the above scenarios, protocol ϕ^l is certificated against every collusive set whose size is at most t; thus, it is t-certificated.

Lemmas 5, 6 and 7 complete the proof of Theorem 2.

3 Demonstration

In this section, we give a two-party integer comparison protocol as an example of the application of our framework. The comparison problem, also known as Yaos millionaire problem [14], has been studied in many literatures [15–21]. The primitive building blocks and the comparison protocol are adopted from [22]. We will show that these protocols are 1-certificated. We will first introduce primitive building blocks, and then construct the integer comparison protocol.

All protocols presented here are based on additive secret sharing over Z_N. That is, a secret value x is split into n shares $x_1, x_2, \ldots, x_n \in Z_N$ to n parties, such that $x = \sum_{i=1}^{n} x_i$, and any $n-1$ subset $\{x_{i_1}, \ldots, x_{i_{n-1}}\}$ is uniformly distributed. The original secret can only be recovered, if and only if all the shares are combined together.

3.1 Primitive Building Blocks

The secure protocols presented in this section are based on the secure Scalar-Product protocol, which is defined as

Definition 7 (Scalar Product). *Party 1 and Party 2 want to collaboratively compute the scalar product of their private input vectors* $X = (x_1, \ldots, x_d)$ *and* $Y = (y_1, \ldots, y_d)$. *That is, they want to execute the secure protocol*

$$((x_1, \ldots, x_d), (y_1, \ldots, y_d)) \mapsto (z_1, z_2),$$

such that

$$z_1 + z_2 = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}^T \begin{bmatrix} y_1 \\ \vdots \\ y_d \end{bmatrix} = \sum_{i=1}^{d} x_i \cdot y_i$$

where $x_i, y_i, z_1, z_2 \in \mathbb{Z}_n$. *Additionally,* $+$ *and* $\cdot$ *are the modular addition and the modular multiplication in* $\mathbb{Z}_n$.

The implementation of scalar product protocols can be found in [23,24]. The specific implementation of the scalar product protocol that we have adopted, runs with a commodity party C, which is assumed to be semi-honest. The commodity party C will not collude with the two parties, nor will it participate directly in the computation of the protocol. It essentially acts only as a random variable generator for the two parties.

PROTOCOL Scalar Product

1. C generates two $1 \times n$ random matrix R_a, R_b.
2. Let $r_a + r_b = R_a \cdot R_b^T$. C sends R_a and r_a to Party 1, and R_b and r_b to Party 2.
3. Party 1 compute $X' = X + R_a$, and Party 2 computes $Y' = Y + R_b$.
4. Party 1 sends X' to Party 2, and Party 2 sends Y' to Party 1.
5. Party 2 generates a random value z_2 as its output, and computes $s = X' \cdot X^T + r_b - z_2$.
6. Party 2 sends s to Party 1.
7. Party 1 computes its output $z_1 = s - (R_a \cdot X'^T) + r_a$.

Each party in this scalar product protocol can not get any information about the other parties' private input from the messages that are exchanged between them, and the output he or she produces [24]. Therefore, this protocol is 1-certificated because it satisfies the three conditions we list in Definition 6.

Before presenting the secure comparison protocol, we will first introduce two protocols, $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ and $\mathbb{Z}_2$-to-$\mathbb{Z}_n$, performs conversions between $\mathbb{Z}_n$ sharing and bitwise $\mathbb{Z}_2$ sharing.

Definition 8 ($\mathbb{Z}_n$-to-$\mathbb{Z}_2$). *Party 1 and Party 2 additively share a number in* $\mathbb{Z}_n$, *and they want to securely convert the* $\mathbb{Z}_n$ *sharing into bitwise* $\mathbb{Z}_2$ *sharing. More specifically, Party 1 and Party 2 want to collaboratively execute the secure protocol*

$$(x_1, x_2) \mapsto ((y_1^0, \ldots, y_1^k), (y_2^0, \ldots, y_2^k)),$$

such that

$$(y^k y^{k-1} \cdots y^1 y^0)_2 = x_1 + x_2$$

where $x_1, x_2 \in \mathbb{Z}_n$, $y_1^l, y_2^l \in \mathbb{Z}_2$, *and* $y^l = y_1^l + y_2^l \pmod 2$.

To convert from $\mathbb{Z}_n$ sharing to bitwise $\mathbb{Z}_2$ sharing, we emulate the carry ripple adder with binary Scalar-product protocol, whose $n = 2$. Let $x_1 = (x_1^k \cdots x_1^0)_2$, $x_2 = (x_2^k \cdots x_2^0)_2$, and the adder operates as the following long addition:

$$\begin{array}{rr} & c^{k+1}\ c^k \cdots c^1\ c^0 \\ & x_1^k \cdots x_1^1\ x_1^0 \\ +) & x_2^k \cdots x_2^1\ x_2^0 \\ \hline & y^k \cdots y^1\ y^0 \end{array}$$

where $c^0 = 0$ and $c^{l+1} = c^l x_1^l + c^l x_2^l + x_1^l x_2^l \pmod 2$ are the carry bits; $y^l = c^l + x_1^l + x_2^l \pmod 2$ is the l-th summation bit. Next, we present the $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ protocol as follows:

PROTOCOL $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ $(n = 2^{k+1})$

1. Party i locally sets $c_i^0 = 0$, and $y_i^0 = x_i^0$, $i = 1, 2$.
2. For $l = 0, \ldots, k-1$, repeat Step 2a to Step 2b.[2]
   (a) Party 1 and Party 2 collaboratively execute the binary Scalar-product protocol
   $$((c_1^l, x_1^l, x_1^l), (x_2^l, c_2^l, x_2^l)) \mapsto (z_1^l, z_2^l),$$
   such that
   $$z_1^l + z_2^l \pmod 2 = \begin{bmatrix} c_1^l \\ x_1^l \\ x_1^l \end{bmatrix} \begin{bmatrix} x_2^l \\ c_2^l \\ x_2^l \end{bmatrix}^T \pmod 2$$
   (b) For $j = 1, 2$, Party j computes
   $$\begin{aligned} c_j^{l+1} &= c_j^l x_j^l + z_j^l \pmod 2 \\ y_j^{l+1} &= x_j^{l+1} + c_j^{l+1} \pmod 2 \end{aligned}$$

The $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ protocol is 1-certificated. The parties run step 1 locally without communication. Therefore, the only step we need to examine is step 2. In Step 2, the two parties collaboratively execute the scalar product protocol.

Let $y_i = (y_i^0, y_i^1, \ldots, y_k^i)$ and $c_i = (c_i^0, c_i^1, \ldots, c_{k+1}^i)$, for $i \in \{1, 2\}$. This protocol can be reformulated using the composition model proposed in Sect. 2.4 as follows.

PROTOCOL $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ $(n = 2^{k+1}$, reformulated using composition model)

1. Party i locally sets $c_i^0 = 0$, and $y_i^0 = x_i^0$, for $i = 1, 2$.
2. Party i sets $z_i^0 \leftarrow (x_i, c_i^0, y_i^0)$, for $i = 1, 2$.
3. Party i sets $\text{VIEW}_i^{\Pi,0}(X) \leftarrow (x_i^0, z_i^0)$, for $i = 1, 2$.
4. For $l = 0, \ldots, k-1$, repeat the following steps
   (a) Party 1 and Party 2 collaboratively execute the binary Scalar-product protocol
   $$((c_1^l, x_1^l, x_1^l), (x_2^l, c_2^l, x_2^l)) \mapsto (z_1^l, z_2^l), \text{ such that}$$
   $$z_1^l + z_2^l \pmod 2 = \begin{bmatrix} c_1^l \\ x_1^l \\ x_1^l \end{bmatrix} \begin{bmatrix} x_2^l \\ c_2^l \\ x_2^l \end{bmatrix}^T \pmod 2,$$
   and receives random coin tosses r_i^l, communicated messages m_i^l.

[2] Since $n = 2^{k+1}$, the overflow bit c^{k+1} is discarded.

(b) For $j = 1, 2$, Party j computes the output of the current step $o_i^l = (c_j^{l+1}, y_j^{l+1})$ as:

$$\begin{aligned} c_j^{l+1} &= c_j^l x_j^l + z_j^l \pmod 2 \\ y_j^{l+1} &= x_j^{l+1} + c_j^{l+1} \pmod 2 \end{aligned}$$

(c) Party i locally produces independent coin tosses s_i^l, and sets

$$\begin{aligned} z_i^l &= f_i^l(\text{VIEW}_i^{\Pi,l-1}(X), x_i, r_i^l, m_i^l, o_i^l, s_i^l) \\ &= (x_i, c_i^l, y_i^l), \end{aligned}$$

for $i = 1, 2$.

(d) Party i sets

$$\text{VIEW}_i^{\Pi,l}(X) = (\text{VIEW}_i^{\Pi,l-1}(X), x_i^l, r_i^l, m_i^l, o_i^l, s_i^l, z_i^l)$$

5. Party i outputs y_i and halts.

Therefore, the protocol $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ is 1-certificated.

Definition 9 ($\mathbb{Z}_2$-to-$\mathbb{Z}_n$). *Party 1 and Party 2 bitwise, additively share a number in $\mathbb{Z}_2$, and they want to securely convert the bitwise $\mathbb{Z}_2$ sharing into the $\mathbb{Z}_n$ sharing. More specifically, Party 1 and Party 2 want to execute the secure protocol $((x_1^0, \ldots, x_1^k), (x_2^0, \ldots, x_2^k)) \mapsto (y_1, y_2)$, such that*

$$y_1 + y_2 = (x^k x^{k-1} \cdots x^1 x^0)_2$$

where $x_1^l, x_2^l \in \mathbb{Z}_2$, $y_1, y_2 \in \mathbb{Z}_n$, and $x^l = x_1^l + x_2^l \pmod 2$.

According to the above requirement, the outputs can be rewritten as the following function:

$$\begin{aligned} y_1 + y_2 &= \sum_{l=0}^{k} x^l \cdot 2^l = \sum_{l=0}^{k} (x_1^l + x_2^l \mod 2) \cdot 2^l \\ &= \sum_{l=0}^{k} (x_1^l + x_2^l - 2x_1^l x_2^l) \cdot 2^l \\ &= \sum_{l=0}^{k} x_1^l \cdot 2^l + \sum_{l=0}^{k} x_2^l \cdot 2^l - \sum_{l=0}^{k} x_1^l x_2^l \cdot 2^{l+1} \end{aligned}$$

In the above function, we divide the computation into two parts. One is locally computable ($\sum x_1^l \cdot 2^l$ and $\sum x_2^l \cdot 2^l$), and the other needs the scalar product protocol ($\sum x_1^l x_2^l \cdot 2^{l+1}$).

PROTOCOL $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ ($n = 2^{k+1}$)

1. Party 1 and Party 2 execute the Scalar-product protocol

$$((x_1^0, \ldots, x_1^k), (2x_2^0, \ldots, 2^{k+1} x_2^k)) \mapsto (t_1, t_2),$$

such that

$$t_1 + t_2 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_1^k \end{bmatrix}^T \begin{bmatrix} 2 \cdot x_2^0 \\ \vdots \\ 2^{k+1} \cdot x_2^k \end{bmatrix}$$

2. Party j computes $y_j = \sum_{l=0}^{k} x_j^l \cdot 2^k - t_j$, for $j = 1, 2$.

Protocol $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ is rather simple than $\mathbb{Z}_n$-to-$\mathbb{Z}_2$. It uses the scalar product protocol for only one time, while $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ calls the scalar product protocol for k times. We can find that the $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ protocol reduces the function $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ to the scalar product that is implemented using the 1-certificated protocol. Therefore, the $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ protocol is 1-certificated.

3.2 The Integer Comparison Protocol

The comparison protocol proposed in [22] compares two values v_1 and v_2 by computing the most significant bit of $(v_1 - v_2)$. According to the binary system on modern computers, if the most significant bit of $(v_1 - v_2)$ is 1, $(v_1 - v_2)$ is a negative number inferring that v_1 is less than v_2. Therefore, the comparison is defined as

Definition 10 (Comparison). *Party 1 and Party 2 additively share a number in $\mathbb{Z}_n$, and they want to know whether the number is positive or negative. As a result, Party 1 and Party 2 want to collaboratively execute the secure protocol $(x_1, x_2) \mapsto (y_1, y_2)$, such that*

$$y_1 + y_2 = \begin{cases} 1 \text{ if } x_1 + x_2 < 0, \\ 0 \text{ otherwise.} \end{cases}$$

That is, one party sets x_1 to v_1 and another party sets x_2 to $-v_2$. Then they can compare v_1 and v_2 according to the above definition. The comparison protocol checks whether the most significant bit of the shared number is 1 as follows.

PROTOCOL Comparison

1. Party 1 and Party 2 collaboratively execute the $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ protocol $(x_1, x_2) \mapsto ((b_1^0, \ldots, b_1^k), (b_2^0, \ldots, b_2^k))$, such that $b^i = b_1^i + b_2^i \pmod 2$, and $(b^k \cdots b^0)_2 = x_1 + x_2$.
2. Party 1 and Party 2 collaboratively execute the $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ protocol $(b_1^k, b_2^k) \mapsto (y_1, y_2)$, such that $y_1 + y_2 = (b^k)_2$ and $b^k = b_1^k + b_2^k \pmod 2$.

We can also formulate this protocol using the composition model given in Sect. 2.4 as follows.

PROTOCOL Comparison (reformulated using composition model)

1. Party i starts with private input x_i and sets $z_i^0 \leftarrow x_i$, for $i = 1, 2$.
2. Party i sets $\text{VIEW}_i^{\Pi,0}(X) \leftarrow (x_i, z_i^0)$, for $i = 1, 2$.
3. Party 1 and Party 2 collaboratively execute the $\mathbb{Z}_n$-to-$\mathbb{Z}_2$ protocol $(x_1, x_2) \mapsto ((b_1^0, \ldots, b_1^k), (b_2^0, \ldots, b_2^k))$, such that $b^i = b_1^i + b_2^i \pmod 2$, and $(b^k \cdots b^0)_2 = x_1 + x_2$. Party i receives random coin tosses r_i^1, communicated messages m_i^l, and the protocol output $o_i^1 = (b_i^0, \ldots, b_i^k)$.

4. Party i locally produces independent coin tosses s_i^1, and sets z_i^1 to be a function of own knowledge, i.e.

$$\begin{aligned} z_i^1 &= f_i^1(\text{VIEW}_i^{\Pi,0}(X), x_i^1, r_i^1, m_i^1, o_i^1, s_i^1) \\ &= b^i, \end{aligned}$$

for $i = 1, 2$.
5. Party i sets

$$\text{VIEW}_i^{\Pi,1}(X) = (\text{VIEW}_i^{\Pi,0}(X), x_i^0, r_i^0, m_i^0, o_i^0, s_i^0, z_i^0),$$

for $i = 1, 2$.
6. Party i sets the new private input as b_i^k, for $i = 1, 2$.
7. Party 1 and Party 2 collaboratively execute the $\mathbb{Z}_2$-to-$\mathbb{Z}_n$ protocol $(b_1^k, b_2^k) \mapsto (y_1, y_2)$, such that $y_1 + y_2 = (b^k)_2$ and $b^k = b_1^k + b_2^k \pmod 2$. Party i receives random coin tosses r_i^2, communicated messages m_i^2, and the protocol output $o_i^2 = y_i$.
8. Party i locally produces independent coin tosses s_i^2, and sets z_i^2 to be a function of own knowledge, i.e.

$$\begin{aligned} z_i^2 &= f_i^2(\text{VIEW}_i^{\Pi,1}(X), x_i^2, r_i^2, m_i^2, o_i^2, s_i^2) \\ &= y_i, \end{aligned}$$

for $i = 1, 2$.
9. Party i sets

$$\text{VIEW}_i^{\Pi,2}(X) = (\text{VIEW}_i^{\Pi,1}(X), x_i^1, r_i^1, m_i^1, o_i^1, s_i^1, z_i^1),$$

for $i = 1, 2$.
10. Party i sets z_i^2 as the output and halts, for $i = 1, 2$.

Therefore, by Theorem 2, we know the comparison protocol is 1-certificated.

4 Concluding Remarks

In this paper, we proposed a composition theorem for secure multi-party computation by adopting an information theoretical approach. The theorem can be used to develop a framework for constructing application protocols with primitive building blocks. Any existing secure protocols can serve as building blocks in our framework, as long as they satisfy the necessary conditions.

The security of the derived protocol is guaranteed as long as the preconditions of the composition theorem are satisfied. Our proposed method provides a significant simplification to the process of verifying the security of the derived composite protocols, which may be quite complex if other available verification methods are applied. We have demonstrated the practicality and effectiveness of our framework by applying it to verify the security of an existing protocol.

In real applications, although perfect privacy would be ideal, sometimes "adequate" privacy is acceptable. When secure multi-party computation is utilized in the public sector, privacy must be compromised sometimes in order to accommodate other important social values. To exploit the enormous amounts of now widely available high quality data, a balance must be found between ensuring adequate privacy protection and the efficient execution of computational tasks [26]. Therefore, quantifying the amount of privacy preserved by the protocols is not only essential for exploring the trade-off between privacy and complexity, but also allows practitioners to determine if the achieved privacy level is adequate.

The information theoretical approach is a strong candidate for quantifying the amount of information preserved or revealed by a protocol [24]. Therefore, we expect to extend our framework to accommodate further mechanism for balancing privacy and performance.

References

1. Yao, A.: How to generate and exchange secrets. In: Proceedings of the 27rd Annual IEEE Symposium on Foundations of Computer Science, pp. 162–167, November 1986
2. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game, or: a completeness theorem for protocols with honest majority. In: Proceedings of 19th ACM Symposium on Theory of Computing, pp. 218–229 (1987)
3. Kushilevitz, E., Lindell, Y., Rabin, T.: Information-theoretically secure protocols and security under composition. In: Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing, STOC 2006, pp. 109–118. ACM, New York (2006)
4. Beaver, D.: Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. J. Cryptol. **4**(2), 75–122 (1991)
5. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. J. ACM **51**(6), 851–898 (2004)
6. Lindell, Y.: Composition of Secure Multi-Party Protocols: A Comprehensive Study. LNCS, vol. 2815. Springer, Heidelberg (2003)
7. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, FOCS 2001, p. 136. IEEE Computer Society, Washington, D.C. (2001)
8. Canetti, R.: Security and composition of cryptographic protocols: a tutorial (part i). SIGACT News **37**(3), 67–92 (2006)
9. Durgin, N., Mitchell, J., Pavlovic, D.: A compositional logic for proving security properties of protocols. J. Comput. Secur. **11**(4), 677–721 (2003)
10. Datta, A., Derek, A., Mitchell, J.C., Roy, A.: Protocol composition logic (pcl). Electron. Notes Theoret. Comput. Sci. **172**, 311–358 (2007)
11. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**, 143–202 (2000)
12. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2, 1st edn. Cambridge University Press, Cambridge (2004)

13. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991). Schilling, D.L. (ed.)
14. Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, pp. 160–164, November 1982
15. Kerschbaum, F., Biswas, D., de Hoogh, S.: Performance comparison of secure comparison protocols. In: 20th International Workshop on Database and Expert Systems Application, 2009, DEXA 2009, pp. 133–136 (2009)
16. Damgard, I., Geisler, M., Kroigard, M.: Homomorphic encryption and secure comparison. Int. J. Appl. Cryptogr. **1**(1), 22–31 (2008)
17. Shundong, L., Yiqi, D., Qiyou, Y.: Secure multi-party computation solution to yao's millionaires' problem based on set-inclusion. Prog. Nat. Sci. **15**(9), 851–856 (2005)
18. Garay, J., Schoenmakers, B., Villegas, J.: Practical and secure solutions for integer comparison. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 330–342. Springer, Heidelberg (2007)
19. Zhao, B., Delp, E.J.: Secret sharing in the encrypted domain with secure comparison. In: Global Telecommunications Conference (GLOBECOM 2011), pp. 1–5. IEEE (2011)
20. Kaghazgaran, P., Sadeghyan, B.: Secure two party comparison over encrypted data. In: World Congress on Information and Communication Technologies (WICT 2011), pp. 1123–1126 (2011)
21. Toft, T.: Sub-linear, secure comparison with two non-colluding parties. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 174–191. Springer, Heidelberg (2011)
22. Shen, C.-H., Zhan, J., Hsu, T.-S., Liau, C.-J., Wang, D.-W.: Scalar-product based secure two-party computation. In: IEEE International Conference on Granular Computing, GrC 2008, pp. 556–561 (2008)
23. Du, W., Atallah, M.J.: Privacy-preserving cooperative statistical analysis. In: ACSAC 2001: Proceedings of the 17th Annual Computer Security Applications Conference, pp. 102–110. IEEE Computer Society, Washington, D.C. (2001)
24. Chiang, Y.-T., Wang, D.-W., Liau, C.-J., Hsu, T.: Secrecy of two-party secure computation. In: Jajodia, S., Wijesekera, D. (eds.) Data and Applications Security 2005. LNCS, vol. 3654, pp. 114–123. Springer, Heidelberg (2005)
25. Du, W., Zhan, Z.: Building decision tree classifier on private data. In: Proceedings of the IEEE International Conference on Privacy, Security and Data Mining, CRPIT 2014, pp. 1–8. Australian Computer Society Inc., Darlinghurst (2002)
26. Du, W., Zhan, J.: A practical approach to solve secure multi-party computation problems. In: Proceedings of New Security Paradigms Workshop, Virginia Beach, Virginia, USA, September 2002