Game-Theoretic Security for Bit Commitment*

Haruna Higo[1], Keisuke Tanaka[1], and Kenji Yasunaga[2]

[1] Tokyo Institute of Technology, Japan
{higo9, keisuke}@is.titech.ac.jp
[2] Kanazawa University, Japan
yasunaga@se.kanazawa-u.ac.jp

Abstract. Higo, Tanaka, Yamada, and Yasunaga (ACISP 2012) studied oblivious transfer (OT) from a game-theoretic viewpoint in the malicious model. Their work can be considered as an extension of the study on two-party computation in the fail-stop model by Asharov, Canetti, and Hazay (EUROCRYPT 2011).
This paper focuses on bit commitment, and continues to study it from a perspective of game theory. In a similar manner to the work on OT, we consider bit commitment in the malicious model. In order to naturally capture the security properties of bit commitment, we characterize them with a single game where both parties are rational. In particular, we define a security notion from a game theoretic viewpoint, and prove the equivalence between it and the standard security notion.

Keywords: Cryptography, game theory, bit commitment.

1 Introduction

1.1 Motivations

Cryptographic protocols are designed for some parties to accomplish some purposes. When defining their security, we consider situations among honest parties and adversaries. Honest parties always follow the protocol description, while adversaries may deviate from it to attack others, e.g., dig out secrets of others. We usually say a protocol is secure if no adversary can damage the honest parties. The adversaries are assumed to be interested in attacking, however, not interested in protecting their own secret. Also, we assume there is at least one honest party. That is, we do not consider situations where all parties conduct some sort of attack.

Game theory mathematically analyzes decision making of multiple parties. In particular, non-cooperative game theory deals with the situations where the parties act independently. The parties are called rational, since they only care about their own preferences and act to achieve their best satisfactions. If a party

* This research was supported in part by a grant of I-System Co. Ltd., and JSPS Grant-in-Aid for Scientific Research Numbers 23500010, 24240001, 25106509, and 23700010.

has two or more preferences, he considers the trade-offs among them and aims to obtain the most reasonable result.

As described, both non-cooperative game theory and cryptography study the situations where parties act. However, they capture situations from different perspectives. In reality, even adversaries may be reluctant to reveal their secrets. Also, for example, if a party is sure that there is no danger, he may try to obtain more information than expected. That is, all parties may not be completely honest. In a game-theoretic framework, we can formalize such realistic perspectives.

There is a line of work using game-theoretic concepts to study cryptographic protocols. For a survey on the joint work of cryptography and game theory, we refer to [15, 13]. Halpern and Teague [9] introduced such approach of study on secret sharing. Their work has been followed in many subsequent work called rational secret sharing (see [3] and the references therein for the subsequent work). They study it in the presence of rational parties, seeking for secure protocols in a game-theoretic framework. Besides secret sharing, there are several studies using game-theoretic frameworks for cryptographic protocols, e.g., two-party computation [1, 7], leader election [6], byzantine agreement [8], oblivious transfer (OT) [11], and public-key encryption [17]. As an extension of the work by Asharov, Canetti, and Hazay [1] and Higo, Tanaka, Yamada, and Yasunaga [11], we are interested in whether the standard security notions of cryptographic protocols are reasonable in such a realistic model. In order to investigate it, we employ a game-theoretic framework.

In this work, we focus on bit commitment. Two parties, called the *sender* and the *receiver*, interact to implement it. They conduct two phases in series. In the first phase, called the *commit phase*, the sender who has a bit b interacts with the receiver. After that, the receiver obtains a commitment string c, and the sender obtains c and a decommitment string d. In the latter phase, called the *open phase*, the sender persuade the receiver that the committed bit is b through an interaction using c and d . Finally, the receiver outputs a bit representing whether she accepts that b is the committed bit.

In cryptography, we usually require three properties, *hiding property*, *binding property*, and *correctness*, as the security properties for bit commitment. Hiding property guarantees that no receiver can learn the committed bit before starting the open phase. Binding property guarantees that no sender can generate a pair of decommitment strings to open the commitment to both 0 and 1. These two properties are required to protect the sender and the receiver respectively. Correctness guarantees that if two parties honestly follow the protocol description, they can open the bit that was committed in the commit phase. Note that, in cryptography, each of the three properties is defined individually. Thus, for example, we do not consider parties who want to break hiding property and to protect binding property at the same time.

1.2 Previous Studies on Game-Theoretic Security

Asharov et al. [1] studied two-party protocols in the fail-stop model from a game-theoretic viewpoint. Fail-stop adversaries are allowed to abort the protocol rather than continuing at each round, but they cannot conduct other deviation, such as sending illegal messages to the others. They focus on the properties of privacy, correctness, and fairness. They characterized them individually in a game-theoretic manner using a concept called computational Nash equilibrium. For privacy and correctness, they showed the equivalence between the corresponding cryptographic and the game-theoretic notions. For fairness, they showed that their game-theoretic notion is strictly weaker than existing cryptographic ones, and proposed a new cryptographic notion that is equivalent to the game-theoretic one. Groce and Katz [7] continued their consideration on fairness, and showed a way to circumvent impossibility results in cryptography in a game-theoretic framework.

Higo et al. [11] studied two-message oblivious transfer (OT) from a game-theoretic viewpoint, characterizing its security using computational Nash equilibrium. They restrict the target protocol from general two-party computation to OT. However, the characterization of Higo et al. [11] has several advantages. First, they investigated the security in the malicious model, where the adversaries can arbitrarily deviate from the protocol description. Second, both parties are rational in their game while a game defined in [1] is essentially played between a rational party and an honest party. Finally, they characterized all security properties by a single game, whereas each security property is defined in an individual game in [1]. Specifically, Higo et al. [11] listed three preferences for each party. Since parties may have different strength of preferences, they formalize them as a weighted sum of the probabilities where each preference is satisfied. This way of formalization was introduced in order to make the model closer to the reality. With this model, they showed the equivalence between their game-theoretic security and the standard cryptographic security.

1.3 This Work

In this paper, we study bit commitment in a game-theoretic framework. In particular, we define a security notion from a game theoretic viewpoint, and examine the relation between it and the standard security notion. As summarized in Table 1., our work has various advantages compared to the previous studies.

We consider bit commitment in the malicious model. In order to naturally capture its security properties, we define a single game where both parties are rational. In other words, we take over the advantages of [11] over [1].

Since both bit commitment and OT are types of two-party computation, one might think that we can simply extend the result of [11] to the case of bit commitment. However, this is not the case. Bit commitment and OT have an essential difference in the functions they compute. The function of OT is a single function, that is, it has a single pair of inputs and a single pair of outputs. On the other hands, what bit commitment computes is a type of reactive

Table 1. Results of [1], [11], and this work.

	Asharov et al. [1]	Higo et al. [11]	This work
Target protocol	Two-party computation	Two-message OT	Bit commitment
# of phases	1	1	2
# of messages	Not restricted	2	Not restricted
Adversary model	Fail-stop model	Malicious model	Malicious model
# of rational parties	1 out of 2 parties	Both of 2 parties	Both of 2 parties
Properties	1	3	3
Utility functions	Fixed	Weighted	General

functionalities [10, 12], which have multiple phases in their computations. Bit commitment has two phases, with a pair of inputs and outputs for each phase, where the second input may depend on the result of the first phase. Moreover, Higo et al. [11] focused on two-message OT, whose interaction has only one round. For bit commitment, we do not only consider multiple phases, but also get rid of the limitation on the number of rounds. When we consider from a game-theoretic perspective, this difference makes the characterization and the analysis more complicated than those in the case of OT.

Generalized utility functions and a simpler solution concept. In the field of game theory, *utility function* mathematically represents the preferences of each party. We formalize the preferences of each party in bit commitment into a form of utility function.

We do not employ a fixed form such as a fixed value in [1] or a weighted sum in [11]. Our utility functions are said to be more general than the ones in the previous work.

Moreover, we reform the way of perceiving the preferences. Since protocols may be used repeatedly, the users are not just interested in a good outcome of a game but prefer to use a good protocol. We characterize the preferences of the parties not over the outcomes of single executions of a protocol, but over the algorithms used by the parties. Although it is not an essential difference, it contributes to employ Nash equilibrium rather than computational Nash equilibrium. As a result, we obtain a simple description of the theorem and its proof.

Non-triviality of our theorem. We prove that our security is equivalent to the standard cryptographic one. The implications between the two security notions are not trivial. Actually, they are, in general, not comparable. In the cryptographic security, we define the three properties individually, whereas rational parties pay attention to the trade-offs among them. That is, if there is a way of attacking some property of a protocol, it is not secure in cryptography. However since rational parties may not perform the attack to the protocol in case this attack together derives a negative result, it may satisfy the game-theoretic security. In this sense, the cryptographic security seems stronger. However, when

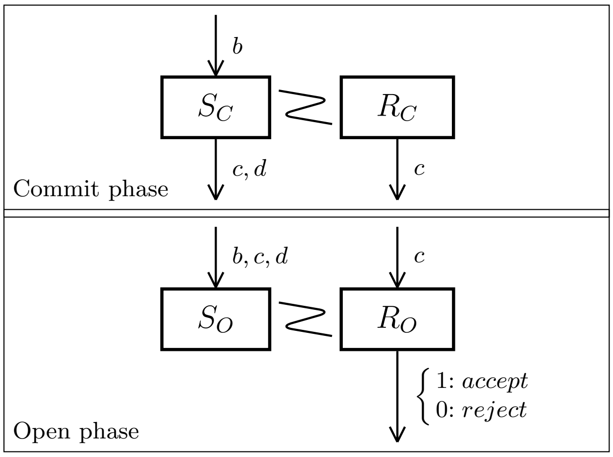

Fig. 1. Bit commitment protocol.

we focus on the number of non-honest parties, the other seems stronger. Considering security in cryptography, we generally assume that there is at least one honest party, but all parties are rational in game theory. That is, everyone is allowed to take arbitrary action.

2 Preliminaries

In this section, we review some cryptographic definitions and game-theoretic concepts.

First, we review some basic definitions. We say a function $\mu : \mathbb{N} \to \mathbb{R}$ is *negligible* if for any polynomial p, there exists $N \in \mathbb{N}$ such that for any $n > N$ it holds that $\mu(n) < 1/p(n)$. We describe a negligible function as $\mathrm{negl}(\cdot)$. An algorithm is *PPT* if it runs in probabilistic polynomial time. In this paper, all the parties are assumed to use PPT algorithms in the security parameter n. Formally, each party has an input 1^n, but we omit this part. For two algorithms A and B, denote the view of A during the interaction with B by $\mathsf{view}_A(B)$, and the output of A after the interaction with B by $\mathsf{out}_A(B)$.

2.1 Bit Commitment in Cryptography

In this section, we review security of bit commitment in a cryptographic framework as defined in [5, 2]. Bit commitment (Fig. 1.) has two phases, the *commit phase* and the *open phase*, which are executed in series. Note that this definition allows interactions in both phases.

Definition 1 (Bit commitment protocol) *A bit commitment protocol* Com *is a tuple of four PPT interactive algorithms, denoted by* $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$.

- *The commit phase is an interaction between S_C and R_C, where S_C receives a bit $b \in \{0,1\}$ as an input. The output of the commit phase consists of the* commitment *string c and a private output d for the sender, called the* decommitment *string. Without loss of generality, let c be the transcript of the interaction between $S_C(b)$ and R_C, and d the view of S_C, including the private random coin of S_C.*
- *The open phase is an interaction between S_O and R_O, where S_O receives (b, c, d), and R_O receives c as inputs. We assume that the first message by the sender explicitly contains a bit b, which indicates that the sender is to persuade the receiver that the committed bit is b. After the interaction, R_O outputs 1 if the receiver accepts, and 0 otherwise.*

Next, we review a security notion of commitment in the malicious model. In this model, adversaries are allowed to act arbitrarily. That is, they may follow the description of the protocol, stop the protocol execution, or deviate from it. A protocol is called secure if it satisfies three properties, hiding property, binding property, and correctness. Since we derive a new security notion in terms of game theory in the next section, this one is called the *cryptographic security.*

Definition 2 (Cryptographic security) *Let $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$ be a bit commitment protocol. We say Com is* cryptographically secure *if it satisfies the following three properties.*

Hiding property: *For any $b \in \{0,1\}$, PPT cheating receiver R_C^*, and PPT distinguisher D, it holds that*

$$\Pr[D(\mathsf{view}_{R_C^*}(S_C(b))) = b] \leq 1/2 + \mathrm{negl}(n).$$

Binding property: *For any $b \in \{0,1\}$, PPT cheating sender (S_C^*, S_O^*), and PPT decommitment finder F, it holds that*

$$\Pr[\mathsf{out}_{R_O(c^*)}(S_O^*(0, c^*, d_0)) = \mathsf{out}_{R_O(c^*)}(S_O^*(1, c^*, d_1)) = 1] \leq \mathrm{negl}(n),$$

where c^* is the transcript between $S_C^*(b)$ and R_C, (d_0, d_1) is the output of $F(\mathsf{view}_{S_C^*(b)}(R_C))$.

Correctness: *For any $b \in \{0,1\}$, it holds that*

$$\Pr[\mathsf{out}_{R_O(c)}(S_O(b, c, d)) = 1] \geq 1 - \mathrm{negl}(n),$$

where c is the transcript between $S_C(b)$ and R_C, and $d = \mathsf{view}_{S_C(b)}(R_C)$.

2.2 Game Theory

Game theory [4, 16] studies actions of some parties aiming at their own goals. We characterize the situations as a *game* in terms of game theory. The parties of the game have their own preferences. In games, parties choose the best actions from their alternatives to obtain the most preferable outcome. The series of actions of each party is collectively called *strategies.* When we analyze cryptographic protocols from a game-theoretic viewpoint, the tuple of algorithms of each party accounts for his strategy.

Utility functions stands for the preferences of the parties. A utility function maps from a tuple of strategies of parties to a real number. When all parties choose their strategies, the outcome of the game is (probabilistically) determined. The values of utility functions usually represent the degree of its preference over the outcome. Higher rate represents stronger preference. Each party guesses the actions of the others, and estimate his own utility to choose his best strategy. Every party chooses the algorithm that delivers him the highest utility.

We are interested in how the parties act in the game. Solution concepts characterize which tuples of strategies are likely to be chosen by the parties. While there are many solution concepts introduced, we employ *Nash equilibrium*, which is one of the most commonly used. When all parties choose the Nash equilibrium strategies, no party can gain his utility by changing his strategy unilaterally. Namely, if parties are assumed to choose the Nash equilibrium strategies, no party have any motivation to change his strategy.

3 Bit Commitment in Game Theory

In this section, we introduce game-theoretic definitions with respect to bit commitment. First, we define a game to execute a protocol. Then, we consider the natural preferences of the sender and the receiver. The solution concept we employ is Nash equilibrium [15, 13]. Finally, we characterize the required properties for bit commitment using these notions in the game-theoretic framework.

Game. Given a bit commitment protocol $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$, we define a game between a sender and a receiver. A sender has three PPT algorithms (S'_C, S'_O, F), and a receiver has two PPT algorithms (R'_C, D) in our game. Here is an informal description of the game. (See also Fig. 2.)

First, the sender and the receiver execute a commit phase by using S'_C and R'_C together with a random bit b as the input for the sender. Then, a distinguisher D of the receiver tries to guess the committed bit b using her view in the commit phase. After that, a decommitment finder F of the sender tries to generate two decommitment strings d_0 and d_1, where d_b is used for opening b as the committed bit. Using S'_O and R_O, two open phases are executed, whether d_0 and d_1 are correctly used to open the commitment generated in the commit phase. Note that the receiver has to use R_O as the open phase algorithm. Since otherwise, the receiver can even accept/reject all the commitment, and such strategies should be excluded from her choice.

Now we formally define a bit commitment game.

Definition 3 (Game) *For a bit commitment protocol* $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$*, and PPT algorithms* S'_C*,* S'_O*,* F*,* R'_C*, and* D*, the game* $\Gamma^{\mathsf{Com}}((S'_C, S'_O, F), (R'_C, D))$ *is executed as follows.*

1. *Choose a bit* b *uniformly at random and set* $\mathsf{guess} = \mathsf{amb} = \mathsf{suc} = \mathsf{abort} = 0$.

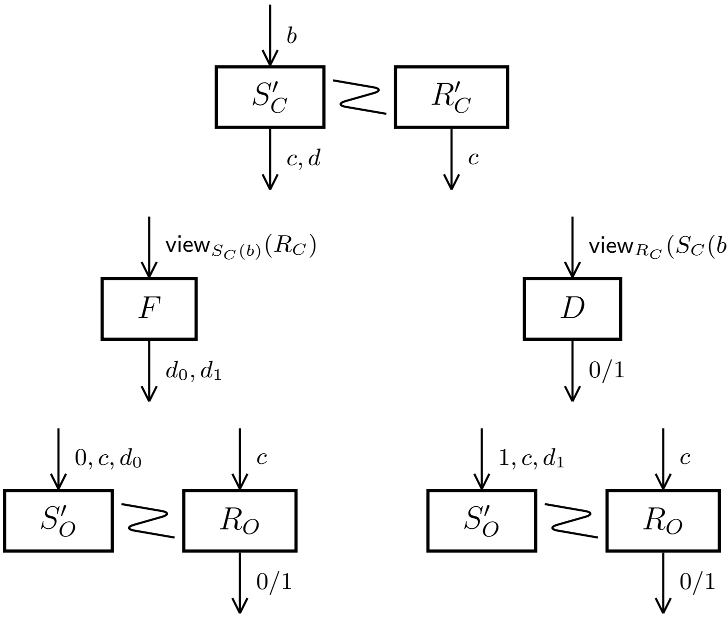

Fig. 2. Bit commitment game.

2. *Observe an interaction between* $S'_C(b)$ *and* R'_C*, and* c *denotes the transcript during the interaction. Set* $\mathsf{abort} = 1$ *if some party aborts the protocol.*
3. *Set* $\mathsf{guess} = 1$ *if* $b = D(\mathsf{view}_{R'_C}(S'_C(b)))$*.*
4. *Run* $F(\mathsf{view}_{S'_C(b)}(R'_C))$ *and get* (d_0, d_1) *as output.*
5. *Observe an interaction between* $S'_O(0, c, d_0)$ *and* $R_O(c)$*, and between* $S'_O(1, c, d_1)$ *and* $R_O(c)$*. Set* $\mathsf{abort} = 1$ *if some party aborts.*
6. *Set* $\mathsf{amb} = 1$ *if* $\mathsf{out}_{R_O(c)}(S'_O(0, c, d_0)) = \mathsf{out}_{R_O(c)}(S'_O(1, c, d_1)) = 1$*, and* $\mathsf{suc} = 1$ *if either* $\mathsf{out}_{R_O(c)}(S'_O(b, c, d_b)) = 1$ *or* $\mathsf{abort} = 1$*.*

The tuple $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ is the outcome of this game, and is explained as follows.

After the commit phase, the receiver tries to learn the committed bit b beforehand. If she succeeded in guessing, then $\mathsf{guess} = 1$. Otherwise, $\mathsf{guess} = 0$. The sender tries to find two decommitment strings d_0 and d_1 in order that $d_{b'}$ can be opened to b'. Acceptance of both bits implies that he can choose the bit to be opened. If the sender succeed in finding such values, then $\mathsf{amb} = 1$. Otherwise, $\mathsf{amb} = 0$. If the receiver can accept the commitment for the committed bit b, or one of the parties aborts the protocol, then $\mathsf{suc} = 1$. Otherwise, $\mathsf{suc} = 0$.

Utility functions. We consider that each party of bit commitment has multiple goals listed as the following preferences.

We consider that the sender has the following two preferences:

– He does not prefer the receiver to know the committed bit b before executing the open phase.
– On executing the open phase, he prefers to be able to choose a bit to be opened.

Next, the receiver is considered to have the following three preferences:

– She prefers to learn the committed bit b before executing the open phase.
– She does not prefer the sender to change the bit to be opened in the open phase.
– She prefers to open the committed bit b in the open phase unless the protocol was aborted.

We formalize these preferences as utility functions. Similar to the work of Higo et al. [11], each party has a single utility function that represents all the preferences in a lump. However, our utility functions are not in a fixed form such as weighted sum used in [11]. Moreover, to describe the preferences over the algorithms used in the game, the arguments of utility functions are the algorithms. They are evaluated using the prescribed three random variables guess, amb and suc that represents the outcome of the game.

For simplicity, we use the following notations. We denote by $a \prec b$ or $b \succ a$ for $a(n), b(n) \in \mathbb{R}$, if it holds that $a(n) < b(n) - \epsilon(n)$ for some non-negligible function ϵ. Also, $a \approx b$ denotes that $|a(n) - b(n)| \leq \mathrm{negl}(n)$.

Definition 4 (Utility functions) *For a bit commitment protocol* Com*, and PPT algorithms* S_C*,* S_O*,* R_C*,* S'_C*,* S'_O*,* R'_C*,* D*, and* F*, let* $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ *and* $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$ *be the random variables representing the outcome of* $\Gamma^{\mathsf{Com}}((S_C, S_O, F), (R_C, D))$ *and* $\Gamma^{\mathsf{Com}}((S'_C, S'_O, F), (R'_C, D))$*, respectively. The utility function* U_S^{Com} *for the sender satisfies* $U_S^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) > U_S^{\mathsf{Com}}((S'_C, S'_O, F), (R'_C, D))$ *if one of the following conditions holds.*

S-1. $|\Pr[\mathsf{guess} = 1] - 1/2| \prec |\Pr[\mathsf{guess}' = 1] - 1/2|$ *and* $\Pr[\mathsf{amb} = 1] \approx \Pr[\mathsf{amb}' = 1]$.
S-2. $\Pr[\mathsf{guess} = 1] \approx \Pr[\mathsf{guess}' = 1]$ *and* $\Pr[\mathsf{amb} = 1] \succ \Pr[\mathsf{amb}' = 1]$.

The utility function U_R^{Com} *for the receiver satisfies* $U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) > U_R^{\mathsf{Com}}((S'_C, S'_O, F), (R'_C, D))$ *if one of the following conditions holds.*

R-1. $|\Pr[\mathsf{guess} = 1] - 1/2| \succ |\Pr[\mathsf{guess}' = 1] - 1/2|$*,* $\Pr[\mathsf{amb} = 1] \approx \Pr[\mathsf{amb}' = 1]$*, and* $\Pr[\mathsf{suc} = 1] \approx \Pr[\mathsf{suc}' = 1]$.
R-2. $\Pr[\mathsf{guess} = 1] \approx \Pr[\mathsf{guess}' = 1]$*,* $\Pr[\mathsf{amb} = 1] \prec \Pr[\mathsf{amb}' = 1]$*, and* $\Pr[\mathsf{suc} = 1] \approx \Pr[\mathsf{suc}' = 1]$.
R-3. $\Pr[\mathsf{guess} = 1] \approx \Pr[\mathsf{guess}' = 1]$*,* $\Pr[\mathsf{amb} = 1] \approx \Pr[\mathsf{amb}' = 1]$*, and* $\Pr[\mathsf{suc} = 1] \succ \Pr[\mathsf{suc}' = 1]$.

Note that we use the value $|\Pr[\mathsf{guess} = 1] - 1/2|$ rather than $\Pr[\mathsf{guess} = 1]$. After a single execution of the game, the sender prefers guess to be 0, and the receiver 1. However, focusing on what the parties hope the algorithm to be, we consider that the sender prefers guess to be close to $1/2$, and the receiver prefers it to be far from $1/2$.

Nash equilibrium. As mentioned in Section 2.2., we use Nash equilibrium as the solution concept in this paper. When a pair of strategies in a Nash equilibrium is chosen by the parties, neither party can gain more no matter how he changes his strategy unilaterally. Although all strategies we consider are polynomially bounded, we do not need to use the extended notion named computational Nash equilibrium as is used in the previous work [1, 11]. This conversion is attributed to the reformation of the utility. Since our utility functions describe the preferences over the strategies not over the outcomes of the games, the discussion of computability is done with evaluating utility functions.

Definition 5 (Nash equilibrium) *Let* Com *be a bit commitment protocol. A tuple of PPT strategies* $((S_C, S_O), R_C)$ *is in a* Nash equilibrium, *if for any PPT algorithms* S_C^*, S_O^*, R_C^*, D, *and* F, *neither of the followings hold.*

- $U_S^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) < U_S^{\mathsf{Com}}((S_C^*, S_O^*, F), (R_C, D))$
- $U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) < U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C^*, D))$

Note that the strategies of the parties are (S_C, S_O) and R_C. D and F are excluded from strategies. That is because, informally, the parties always choose the best D and F to improve their utilities.

Game-theoretic security. We characterize the required properties for bit commitment using the prescribed notions. If a protocol is in a Nash equilibrium, it means that the parties will prefer to take the strategies according to the protocol. In other words, the parties do not have a motivation to deviate from the protocol. We call such protocols *game-theoretically secure.*

Definition 6 (Game-theoretic security) *Let* $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$ *be a bit commitment protocol. We say* Com *is* game-theoretically secure *if the tuple of the strategies* $((S_C, S_O), R_C)$ *is in a Nash equilibrium.*

4 Equivalence between the Two Security Notions

In this section, we prove the equivalence between the cryptographic security (Definition 2) and the game-theoretic security (Definition 6). In other words, we show that a protocol is cryptographically secure if and only if the protocol itself is in a Nash equilibrium.

Theorem 1 *Let* Com *be a bit commitment protocol.* Com *is cryptographically secure if and only if* Com *is game-theoretically secure.*

As mentioned in Section 1.3., this relationship is not trivial. We provide both directions of implication one by one.

First, we show that the cryptographic security implies the game-theoretic security.

Lemma 1 *If a bit commitment protocol* Com *is cryptographically secure, then* Com *is game-theoretically secure.*

We prove the contrapositive of this statement. If a protocol is not game-theoretically secure, that is, it is not in a Nash equilibrium, at least one party can gain with using some alternative strategies rather than the protocol description. From the definitions of the utility functions, it is natural that the alternative strategies break some of the cryptographic property, which implies that the protocol is not cryptographically secure. Actually, the definition of Nash equilibrium makes the proof a little complicated. The formal proof is as follows.

Proof. To prove this lemma, we assume that $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$ is not game-theoretically secure, and show that Com is not cryptographically secure. Namely, Com does not satisfy at least one of the three properties, hiding property, binding property, and correctness.

Suppose Com is not game-theoretically secure. Then, there exist a tuple $((S_C^*, S_O^*), R_C^*)$ of PPT strategies, a PPT distinguisher D and a PPT decommitment finder F such that at least one of the following two inequalities holds:

$$U_S^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) < U_S^{\mathsf{Com}}((S_C^*, S_O^*, F), (R_C, D)), \tag{1}$$
$$U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C, D)) < U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C^*, D)). \tag{2}$$

First, assume that Equality (1) holds. It implies that the sender can get a higher utility by changing his strategy from (S_C, S_O) to (S_C^*, S_O^*). There are two possibilities for the cause of this increase:

Case S-1: $|\Pr[\mathsf{guess} = 1] - 1/2|$ decreases with the change of the strategy.
Case S-2: $\Pr[\mathsf{amb} = 1]$ increases with the change of the strategy.

Case S-1 implies that $|\Pr[\mathsf{guess} = 1] - 1/2| \succ 0$ holds when both parties choose its honest strategy. This means that Com does not satisfy hiding property for R_C.

Case S-2 implies that $\Pr[\mathsf{amb} = 1] \succ 0$ holds for the strategy tuple $((S_C^*, S_O^*), R_C)$. Hence, Com does not satisfy binding property for (S_C^*, S_O^*).

Next, assume that Equality (2) holds. It implies that the receiver can get a higher utility by changing her strategy from R_C to R_C^*. There are three possibilities for the cause of this increase:

Case R-1: $|\Pr[\mathsf{guess} = 1] - 1/2|$ increases with the change of the strategy.
Case R-2: $\Pr[\mathsf{amb} = 1]$ decreases with the change of the strategy.
Case R-3: $\Pr[\mathsf{suc} = 1]$ increases with the change of the strategy.

Case R-1 implies that $|\Pr[\mathsf{guess} = 1] - 1/2| \succ 0$ holds for the strategy tuple $((S_C, S_O), R_C^*)$. This means that Com does not satisfy hiding property for R_C^*.

Case R-2 implies that $\Pr[\mathsf{amb} = 1] \succ 0$ holds when both parties choose their honest strategies. Hence, Com does not satisfy binding property for (S_C, S_O).

Case R-3 implies that $\Pr[\mathsf{suc} = 1] \prec 1$ holds when both parties choose their honest strategy. This means that Com does not satisfy correctness.

In every case, we have shown that Com is not cryptographically secure. Therefore, the statement follows. □

Next, we show that the game-theoretic security implies the cryptographic security.

Lemma 2 *If a bit commitment protocol* Com *is game-theoretically secure, then* Com *is cryptographically secure.*

The proof of this direction is more technical than that of Lemma 1. We prove it by showing that the contrapositive is true. Assume that a protocol is not cryptographically secure, at least one of the security properties, hiding property, binding property and correctness, does not hold. Provided that an algorithm breaks one of the properties, we cannot simply say that the protocol is not in a Nash equilibrium. That is because, the parties consider the tradeoffs among the preferences. If the algorithms together leads to some negative result, the party cannot gain his utility by using this algorithm. This cannot be the reason of the protocol being not game-theoretically secure. This lemma seems not trivial at this point.

Despite this point, the lemma holds because the definition of Nash equilibrium requires the inequality to hold for any D and F. If an algorithm breaks some property, then some D and F makes a situation where only the probability related to the broken property ($\Pr[\mathsf{guess} = 1]$, $\Pr[\mathsf{amb} = 1]$ or $\Pr[\mathsf{suc} = 1]$) changes by using the algorithm rather than following the protocol. That is, when at least one of the security properties does not hold, some tuple of algorithm makes the protocol not in Nash equilibrium.

Here, we provide a formal proof.

Proof. Suppose that $\mathsf{Com} = ((S_C, S_O), (R_C, R_O))$ is not cryptographically secure. We consider the following five cases, and show that Com is not game-theoretically secure in each case.

Case 1: Com does not satisfy correctness.
Case 2: Com satisfies correctness and does not satisfy binding property for (S_C, S_O).
Case 3: Com satisfies correctness and binding property for (S_C, S_O), and does not satisfy binding property for some $(S_C^*, S_O^*) \neq (S_C, S_O)$.
Case 4: Com satisfies correctness and binding property, and does not satisfy hiding property for R_C.
Case 5: Com satisfies correctness, binding property, and hiding property for R_C, and does not satisfy hiding property for some $R_C^* \neq R_C$.

In Case 1, even if both parties follow the protocol description, the probability that they cannot open the committed bit is non-negligible. That is, for some $b \in \{0, 1\}$, it holds that

$$\Pr[\mathsf{out}_{R_O(c)}(S_O(b, c, d)) = 1] \prec 1,$$

where c is the transcript between $S_C(b)$ and R_C, and $d = \mathsf{view}_{S_C(b)}(R_C)$. Let D^{rand} be an algorithm that outputs 0 or 1 uniformly at random,

F^{honest} an algorithm that outputs (d_0, d_1) where $d_b = \mathsf{out}_{S_C(b)}(R'_C)$ and $d_{1-b} = \perp$, where R'_C is an algorithm of the receiver in the commit phase, and R_C^{abort} a strategy of sending the abort message right after starting the protocol. Note that the three algorithms, D^{rand}, F^{honest}, and R_C^{abort}, are PPT algorithms. We denote the outcome of the games $\Gamma^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D^{\mathsf{rand}}))$ and $\Gamma^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C^{\mathsf{abort}}, D^{\mathsf{rand}}))$ by $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ and $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$, respectively. Now we obtain the following equalities:

- $|\Pr[\mathsf{guess} = 1] - 1/2| \approx |\Pr[\mathsf{guess}' = 1] - 1/2| \approx 0$,
- $\Pr[\mathsf{amb} = 1] = \Pr[\mathsf{amb}' = 1] = 0$,
- $\Pr[\mathsf{suc} = 1] \prec \Pr[\mathsf{suc}' = 1] = 1$.

Hence, it holds that $U_R^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D^{\mathsf{rand}})) < U_R^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C^{\mathsf{abort}}, D^{\mathsf{rand}}))$, which implies that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium.

In Case 2, the sender can break binding property with the honest strategy (S_C, S_O). That is, for some PPT decommitment finder F and $b \in \{0, 1\}$, it holds that

$$\Pr[\mathsf{out}_{R_O(c)}(S_O(0, c, d_0)) = \mathsf{out}_{R_O(c)}(S_O(1, c, d_1)) = 1] \succ 0,$$

where (d_0, d_1) is the output of $F(\mathsf{view}_{S_C(b)}(R_C))$. We denote the outcome of the games $\Gamma^{\mathsf{Com}}((S_C, S_O, F), (R_C, D^{\mathsf{rand}}))$ and $\Gamma^{\mathsf{Com}}((S_C, S_O, F), (R_C^{\mathsf{abort}}, D^{\mathsf{rand}}))$ by $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ and $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$, respectively. Now we obtain the following equalities:

- $|\Pr[\mathsf{guess} = 1] - 1/2| \approx |\Pr[\mathsf{guess}' = 1] - 1/2| \approx 0$,
- $\Pr[\mathsf{amb} = 1] \succ \Pr[\mathsf{amb}' = 1] = 0$,
- $\Pr[\mathsf{suc} = 1] = \Pr[\mathsf{suc}' = 1] = 1$.

Hence, it holds that $U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C, D^{\mathsf{rand}})) < U_R^{\mathsf{Com}}((S_C, S_O, F), (R_C^{\mathsf{abort}}, D^{\mathsf{rand}}))$, which implies that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium.

In Case 3, the sender cannot break binding property with honest strategy (S_C, S_O) but with some strategy $(S_C^*, S_O^*) \neq (S_C, S_O)$. That is, for some PPT decommitment finder F and $b \in \{0, 1\}$, it holds that

$$\Pr[\mathsf{out}_{R_O(c)}(S_O^*(0, c^*, d_0)) = \mathsf{out}_{R_O(c)}(S_O^*(1, c^*, d_1)) = 1] \succ 0,$$

where c^* is the transcript between $S_C^*(b)$ and R_C, and (d_0, d_1) is the output of $F(\mathsf{view}_{S_C^*(b)}(R_C))$. For the same F and b, it holds that

$$\Pr[\mathsf{out}_{R_O(c)}(S_O(0, c, d_0)) = \mathsf{out}_{R_O(c)}(S_O(1, c, d_1)) = 1] \approx 0.$$

We denote the outcome of the games $\Gamma^{\mathsf{Com}}((S_C, S_O, F), (R_C, D^{\mathsf{rand}}))$ and $\Gamma^{\mathsf{Com}}((S_C^*, S_O^*, F), (R_C, D^{\mathsf{rand}}))$ by $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ and $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$, respectively. Now we obtain the following equalities:

- $|\Pr[\mathsf{guess} = 1] - 1/2| \approx |\Pr[\mathsf{guess}' = 1] - 1/2| \approx 0$,

- $0 = \Pr[\mathsf{amb} = 1] \prec \Pr[\mathsf{amb}' = 1]$.

Hence, it holds that $U_S^{\mathsf{Com}}((S_C, S_O, F), (R_C, D^{\mathsf{rand}})) < U_S^{\mathsf{Com}}((S_C^*, S_O^*, F), (R_C, D^{\mathsf{rand}}))$, which implies that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium.

In Case 4, the receiver can break hiding property with the honest strategy R_C. That is, for some PPT distinguisher D, it holds that

$$\Pr[D(\mathsf{view}_{R_C}(S_C(b))) = b] \succ 1/2.$$

Let S_C^{abort} be a strategy of sending the abort message right after starting the protocol. We denote the outcome of the games $\Gamma^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D))$ and $\Gamma^{\mathsf{Com}}((S_C^{\mathsf{abort}}, S_O, F^{\mathsf{honest}}), (R_C, D))$ by $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ and $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$, respectively. Now we obtain the following equalities:

- $|\Pr[\mathsf{guess} = 1] - 1/2| \succ |\Pr[\mathsf{guess}' = 1] - 1/2| \approx 0$,
- $\Pr[\mathsf{amb} = 1] = \Pr[\mathsf{amb}' = 1] = 0$.

Hence, it holds that $U_S^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D)) < U_S^{\mathsf{Com}}((S_C^{\mathsf{abort}}, S_O, F^{\mathsf{honest}}), (R_C, D))$, which implies that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium.

In Case 5, the receiver can not break hiding property with honest strategy R_C but with some strategy $R_C^* \neq R_C$. That is, for some PPT distinguisher D, it holds that

$$\Pr[D(\mathsf{view}_{R_C^*}(S_C(b))) = b] \succ 1/2, \text{ and } \Pr[D(\mathsf{view}_{R_C}(S_C(b))) = b] \approx 1/2.$$

Let $\tilde{R}_C^*$ be a strategy of following R_C^* in the commit phase and not participating in the open phase. Then, it holds that $\Pr[D(\mathsf{view}_{\tilde{R}_C^*}(S_C(b)) = b] \succ 1/2$. We denote the outcome of the games $\Gamma^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D))$ and $\Gamma^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (\tilde{R}_C^*, D))$ by $(\mathsf{guess}, \mathsf{amb}, \mathsf{suc})$ and $(\mathsf{guess}', \mathsf{amb}', \mathsf{suc}')$, respectively. Now we obtain the following equalities:

- $0 \approx |\Pr[\mathsf{guess} = 1] - 1/2| \prec |\Pr[\mathsf{guess}' = 1] - 1/2|$,
- $\Pr[\mathsf{amb} = 1] = \Pr[\mathsf{amb}' = 1] = 0$,
- $\Pr[\mathsf{suc} = 1] = \Pr[\mathsf{suc}' = 1] = 1$.

Hence, it holds that $U_R^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (R_C, D)) < U_R^{\mathsf{Com}}((S_C, S_O, F^{\mathsf{honest}}), (\tilde{R}_C^*, D))$, which implies that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium.

In every case, we show that the tuple $((S_C, S_O), R_C)$ is not in a Nash equilibrium. Therefore, the statement follows. □

5 Concluding Remarks

This paper has focused on bit commitment and characterized its security in a game-theoretic manner. Our work is based on the work of OT by Higo et al. [11].

Since bit commitment and OT computes different numbers of functions in their protocols, the characterization of bit commitment is more complicated. In this paper, we have defined a game in which parties execute a bit commitment protocol, and picked up the natural preferences of the sender and the receiver. Using Nash equilibrium as a solution concept, we have defined the notion of game-theoretic security. We have shown the equivalence between the game-theoretic security and the cryptographic security.

Although we have introduced game-theoretic concepts as a formalization of realistic perspectives, no practical application has been known. Further work is expected in this area to describe some practical implication or limitations.

References

1. Asharov, G., Canetti, R., Hazay, C.: Towards a game theoretic view of secure computation. In: Paterson, K.G. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 6632, pp. 426–445. Springer (2011)
2. Chung, K.M., Liu, F.H., Lu, C.J., Yang, B.Y.: Efficient string-commitment from weak bit-commitment. In: Abe, M. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 6477, pp. 268–282. Springer (2010)
3. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient rational secret sharing in standard communication networks. In: Micciancio [14], pp. 419–436
4. Fudenberg, D., Tirole, J.: Game theory (3. pr.). MIT Press (1991)
5. Goldreich, O.: The Foundations of Cryptography - Volume 2, Basic Applications. Cambridge University Press (2004)
6. Gradwohl, R.: Rationality in the full-information model. In: Micciancio [14], pp. 401–418
7. Groce, A., Katz, J.: Fair computation with rational players. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 7237, pp. 81–98. Springer (2012)
8. Groce, A., Katz, J., Thiruvengadam, A., Zikas, V.: Byzantine agreement with a rational adversary. In: Czumaj, A., Mehlhorn, K., Pitts, A.M., Wattenhofer, R. (eds.) ICALP (2). Lecture Notes in Computer Science, vol. 7392, pp. 561–572. Springer (2012)
9. Halpern, J.Y., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: Babai, L. (ed.) STOC. pp. 623–632. ACM (2004)
10. Hazay, C., Lindell, Y.: Efficient Secure Two-Party Protocols: Techniques and Constructions. Springer-Verlag New York, Inc., New York, NY, USA, 1st edn. (2010)
11. Higo, H., Tanaka, K., Yamada, A., Yasunaga, K.: A game-theoretic perspective on oblivious transfer. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP. Lecture Notes in Computer Science, vol. 7372, pp. 29–42. Springer (2012)
12. Jeffs, R.A., Rosulek, M.: Characterizing the cryptographic properties of reactive 2-party functionalities. In: Sahai, A. (ed.) TCC. Lecture Notes in Computer Science, vol. 7785, pp. 263–280. Springer (2013)
13. Katz, J.: Bridging game theory and cryptography: Recent results and future directions. In: Canetti, R. (ed.) TCC. Lecture Notes in Computer Science, vol. 4948, pp. 251–272. Springer (2008)
14. Micciancio, D. (ed.): Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings, Lecture Notes in Computer Science, vol. 5978. Springer (2010)

15. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York, NY, USA (2007)
16. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT Press (1994)
17. Yasunaga, K.: Public-key encryption with lazy parties. In: Visconti, I., Prisco, R.D. (eds.) SCN. Lecture Notes in Computer Science, vol. 7485, pp. 411–425. Springer (2012)