Repeated Games for Generating Randomness in Encryption

Kenji Yasunaga* Kosuke Yuzawa†

March 2, 2017

Abstract

In encryption schemes, the sender may not generate randomness properly if generating randomness is costly, and the sender is not concerned about the security of a message. The problem was studied by the first author (2016), and was formalized in a game-theoretic framework. In this work, we construct an encryption scheme with an optimal round complexity on the basis of the mechanism of repeated games.

1 Introduction

Randomness is essential for many cryptographic primitives. In practice, generating randomness is a complex and difficult task. There are many cryptographic failures [20, 9, 4, 10, 5, 22, 18, 16].

Even though users can access to a good randomness source, they may not use it if generating randomness itself is a costly task. Such a situation arises naturally for energy-saving devices. In encryption schemes, the sender may not generate randomness properly if she is not concerned about the security of a message to be encrypted. Namely, the sender may *rationally* decide not to generate costly randomness.

The problem of such rationality was studied in [21] for public-key encryption schemes. In [21], the author considers a setting in which rational senders and receivers can choose either good randomness and bad randomness. Good one is a truly random string, but is costly. Bad one is a fixed string, e.g., the all-zero string, and can be generated without cost. The author provided constructions of secure encryption schemes depending on the given information to the sender and the receiver. For the most basic situation, in which the receiver does not know whether a message to be sent is valuable to him or not, a two-round scheme is constructed based on any secure public-key encryption scheme. In a more difficult situation, in which the receiver may know the value of a message to him, the two-round scheme is not secure. Then, the author presented a three-round scheme with a peculiar final step, where the receiver encrypts a recovered message with the sender's public key and makes it public.

1.1 This Work

We study the problem of rational behavior for generating randomness in encryption by treating the security game as a *repeated* game. In public-key encryption schemes, after generating a pair of public and secret keys, messages are assumed to be encrypted repeatedly. Thus, it is natural to formalize the security game of encryption schemes as a repeated game. We present a round-efficient scheme based on a mechanism of

*Institute of Science and Engineering, Kanazawa University. Kakuma-machi, Kanazawa, 920–1192, Japan. yasunaga@se.kanazawa-u.ac.jp

†Graduate School of Natural Science and Technology, Kanazawa University. Kakuma-machi, Kanazawa, 920-1192, Japan. makku107@stu.kanazawa-u.ac.jp

repeated games. Specifically, we construct a secure two-round scheme in the setting for which a three-round scheme was presented in [21]. The scheme is the first two-round scheme in the setting where the receiver may know the value of a message to be sent. Since non-interactive schemes cannot be secure [21], the scheme achieves the optimal round complexity.

1.2 Our Model

Our security model is based on the study of [21]. We define a variant of chosen plaintext attack (CPA) game of encryption schemes. The game consists of the key generation phase, which is conducted only once, and the encryption phase, which is played repeatedly. In the encryption phase, an adversary, on input public keys, chooses two challenge messages, and, given a ciphertext, tries to guess which of the two messages was encrypted. The sender and the receiver are rational players, and have their own utility functions. The values of utilities are determined by the outcome of the game. Each rational player needs to choose either good or bad randomness before performing probabilistic algorithms. Roughly speaking, an encryption scheme is said to be secure if the prescribed strategy for rational players is a Nash equilibrium, and a message is securely encrypted in every encryption phase when rational players follow the prescribed strategy.

In this work, we model the above repeated game as an *infinitely* repeated game. More concretely, we consider an infinite sequence of adversaries $A_1, A_2, \ldots$ such that A_i plays only at the i-th encryption phase, called a *stage game*. The i-th stage game is conducted between the sender, the receiver, and A_i. Since A_i cannot communicate with other adversaries, we can avoid the problem of using computationally-secure primitives an exponential number of times. Although the message security is considered for every stage game, rational players are assumed to calculate their utilities as a total of infinitely-many stage games. Thus, we can utilize a mechanism of infinitely repeated games in the framework of CPA games of encryption schemes.

1.3 Related Work

There are many studies using game-theoretic analysis for cryptographic primitives, including secret sharing [14, 1, 6, 3], two-party protocols [2, 8, 17], public-key encryption [21], and protocol design [7]. Among them, repeated games have been introduced only in rational secret sharing [19]. This work shows that the mechanism of repeated games is effective for reducing the round complexity of encryption schemes.

Halpern and Pass [11] introduced the framework of machine games for incorporating the cost of computation, including the cost of randomization, in utility functions. They showed that if randomization is free, there always exists a Nash equilibrium in machine games. In this work, we employ a simpler framework specific to encryption schemes, and show that a Nash equilibrium strategy satisfies CPA security.

Halpern et al. [13, 12] used cryptographic primitives for finding equilibria in repeated games played by computationally-bounded players.

Halpin and Naor [15] proposed a method for generating randomness by human game play.

1.4 Notations

A function $\varepsilon(\cdot)$ is called *negligible* if for any constant c, $\varepsilon(\lambda) < 1/\lambda^c$ for every sufficiently large λ. For two families of random variable $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$, we say X and Y are *computationally indistinguishable*, denoted by $X \approx_c Y$, if for every probabilistic polynomial-time distinguisher D, there is a negligible function $\varepsilon(\cdot)$ such that $|\Pr[D(X_n) = 1] - \Pr[D(Y_n) = 1]| \leq \varepsilon(n)$ for every sufficiently large n. For a probabilistic algorithm A, we denote by $A(x; r)$ the output of A running on input x with randomness r.

2 Repeated Games for Public-Key Encryption

We assume that both the sender and the receiver are rational players. Each player has a set of valuable messages, and prefers a message to be sent confidentially if it is valuable to the player. We consider the cost of generating randomness for algorithms. Each player can choose one of the two types of randomness, good randomness and bad randomness. The former is a truly random string but costly to generate. The latter is the all-zero string and can be generated without cost.

We model the interactions between the sender and the receiver as a game. The game is a variant of the chosen plaintext attack (CPA) game of encryption schemes. First, the key generation phase is conducted by the sender and the receiver individually. Then, the guessing game is conducted between the sender, the receiver, and an adversary. The adversary chooses two messages m_0 and m_1. A randomly chosen message is encrypted by the interaction between the sender and the receiver. Given the transcript of the interaction, the adversary tries to guess which message was encrypted. The guessing game is played repeatedly. We assume that the repeated game is *perfect monitoring*, in the sense that the players can observe each other's actions after each guessing game ends. Let $\mathcal{M}_S$ and $\mathcal{M}_R$ be the sets of valuable messages for the sender and the receiver, respectively. In each guessing game, the adversary chooses two messages so that both of them are in either $\mathcal{M}_S \setminus \mathcal{M}_R$, $\mathcal{M}_R \setminus \mathcal{M}_S$, or $\mathcal{M}_S \cap \mathcal{M}_R$. Let p_S, p_R, p_{SR} denote the probabilities that the chosen messages are in $\mathcal{M}_S \setminus \mathcal{M}_R$, $\mathcal{M}_R \setminus \mathcal{M}_S$, and $\mathcal{M}_S \cap \mathcal{M}_R$, respectively. It holds that $p_S, p_R, p_{SR} \geq 0$ and $p_S + p_R + p_{SR} = 1$. We assume that p_S, p_R, p_{SR} are a priori fixed, and the same values are used in each iterated game.

As observed in [21], it is necessary to define a public-key encryption scheme as an *interactive* protocol in which both the sender and the receiver can generate their own public and secret keys.

Definition 1 (Public-key encryption scheme)**.** *An n-round public-key encryption scheme* Π *is the tuple* $(\{\mathsf{Gen}_w\}_{w\in\{S,R\}}, \{\mathsf{Enc}_i\}_{i\in\{1,\dots,n\}}, \mathsf{Dec})$ *such that*

- ***Key generation:*** *For each* $w \in \{S, R\}$*, on input* 1^λ*,* Gen_w *outputs* (pk_w, sk_w)*. Let* $\mathcal{M}$ *denote the message space.*
- ***Encryption:*** *For a message* $m \in \mathcal{M}$*, set* $st_S = (pk_S, pk_R, sk_S, m)$*,* $st_R = (pk_S, pk_R, sk_R)$*, and* $c_0 = \perp$*. Let* $w \in \{S, R\}$ *be the first sender, and* $\bar{w} \in \{S, R\} \setminus \{w\}$ *the second sender. For each round* $i \in \{1, \dots, n\}$*, when i is odd,* $\mathsf{Enc}_i(c_{i-1}, st_w)$ *outputs* (c_i, st'_w)*, and* st_w *is updated to* st'_w*, and when i is even,* $\mathsf{Enc}_i(c_{i-1}, st_{\bar{w}})$ *outputs* $(c_i, st'_{\bar{w}})$*, and* $st_{\bar{w}}$ *is updated to* $st'_{\bar{w}}$*.*
- ***Decryption:*** *After the encryption phase, on input* st_R*,* Dec *outputs* $\hat{m}$*.*
- ***Correctness:*** *For any message* $m \in \mathcal{M}$*, after the encryption phase,* $\mathsf{Dec}(st_R) = m$*.*

We define a formal security game for rational sender and receiver in repeated games. Without loss of generality, we assume that every probabilistic algorithm requires random bits of length equal to the security parameter, and that, in the encryption phase, only the first algorithm for each party is probabilistic.

Definition 2 (Repeated CPA game for rational parties)**.** *Let* $\Pi = (\{\mathsf{Gen}_w\}_{w\in\{S,R\}}, \{\mathsf{Enc}_i\}_{i\in\{1,\dots,n\}}, \mathsf{Dec})$ *be an n-round public-key encryption scheme. For a sequence of adversaries* $A = (A_1, A_2, \dots)$*, the security parameter* λ*, valuable message spaces* $\mathcal{M}_S$ *and* $\mathcal{M}_R$*, and a pair of strategies* (σ_S, σ_R)*, we define the following game* $\mathbf{Game}^{\mathrm{rep}}(\Pi, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S, \sigma_R)$*:*

- ***Key generation phase:***

– ***Choice of randomness:*** *For each* $w \in \{S, R\}$*, compute* $(a_w^{gen}, st_w) \leftarrow \sigma_w(1^\lambda, \mathcal{M}_w)$*, where* $a_w^{gen} \in \{\mathsf{Good}, \mathsf{Bad}\}$ *and* st_w *is the state information of* w*. We assume that* $\mathcal{M}_w$ *has a polynomial-size representation. If* $a_w^{gen} = \mathsf{Good}$*, choose* $r_w^{gen} \in \{0,1\}^\lambda$ *uniformly at random. Otherwise, set* $r_w^{gen} = 0^\lambda$*.*

– ***Key generation:*** *For each* $w \in \{S, R\}$*, generate* $(pk_w, sk_w) \leftarrow \mathsf{Gen}_w(1^\lambda; r_w^{gen})$*, and set the state information for the challenge phase to be* $st_w^1 = (st_w, pk_w, sk_w, pk_{\bar{w}})$*, where* $\bar{w} \in \{S, R\} \setminus \{w\}$*.*

– ***Outcome of the key generation phase:*** *Output* $(\mathsf{Num}_S^{gen}, \mathsf{Num}_R^{gen})$*, where* Num_w^{gen} *takes* 1 *if* $a_w^{gen} = \mathsf{Good}$*, and* 0 *otherwise.*

- ***Challenge phase:*** *For* $i = 1, 2, \ldots,$ *do the following.*

  – ***Challenge generation:*** *Given* (pk_S, pk_R)*,* A_i *outputs* $(m_0, m_1, a_A^S, a_A^R, st_A)$*, where* $m_0, m_1 \in \mathcal{M}_S \cup \mathcal{M}_R$*,* $a_A^S, a_A^R \in \{0,1\}$ *represent the choices of* A_i *for the auxiliary inputs to the sender and the receiver, and* st_A *is the state information of* A_i*. Then,* $b \in \{0,1\}$ *is chosen uniformly at random.*

  – ***Choice of randomness:*** *For each* $w \in \{S, R\}$*, compute* $(a_w^{enc,i}, st_w^{i+1}) \leftarrow \sigma_w(pk_S, pk_R, sk_w, st_w^i, aux_w)$*, where* $a_w^{enc,i} \in \{\mathsf{Good}, \mathsf{Bad}\}$*,* $aux_S = (m_b, v_R^i, a_R^{enc,i-1})$*,* $aux_R = (a_S^{enc,i-1}, v_S^i)$*,* $a_R^{enc,0} = a_S^{enc,0} = \perp$*,* $v_R^i = \mathsf{Val}_R^i$ *if* $a_A^S = 1$*, and* $v_R^i = \perp$ *otherwise,* $v_S^i = \mathsf{Val}_S^i$ *if* $a_A^R = 1$*, and* $v_S^i = \perp$ *otherwise, where* Val_R^i *and* Val_S^i *are defined below. If* $a_w^{enc,i} = \mathsf{Good}$*, then choose* $r_w^{gen} \in \{0,1\}^\lambda$ *uniformly at random. Otherwise, set* $r_w^{gen} = 0^\lambda$*.*

  – ***Guessing the challenge:*** *The challenge message* m_b *is encrypted by the interaction using* $\{\mathsf{Enc}_i\}_{i \in \{1,\ldots,n\}}$*, where* $w \in \{S, R\}$ *uses* r_w^{gen} *as the random bits for the encryption algorithms. Given the transcript of the interaction and* st_A*,* A_i *outputs* $b' \in \{0,1\}$*.*

  – ***Outcome of the stage game:*** *Output* $(\mathsf{Win}^i, \mathsf{Val}_S^i, \mathsf{Val}_R^i, \mathsf{Num}_S^i, \mathsf{Num}_R^i)$*, where* Win^i *takes* 1 *if* $b' = b$*, and* 0 *otherwise,* Val_w^i *takes* 1 *if* $m_b \in \mathcal{M}_w$*, and* 0 *otherwise, and* Num_w^i *takes* 1 *if* $a_w^{enc,i} = \mathsf{Good}$*, and* 0 *otherwise.*

- *The outcome of the game is* $(\mathsf{Num}_S^{gen}, \mathsf{Num}_R^{gen}, \{\mathsf{Adv}^i, \mathsf{Val}_S^i, \mathsf{Val}_R^i, \mathsf{Num}_S^i, \mathsf{Num}_R^i\}_{i=1,\ldots})$*, where* $\mathsf{Adv}^i = 2|\mathbb{E}[\mathsf{Win}_i] - 1/2|$*.*

In the above game, the adversary can choose whether the sender (and the receiver) can know the value of a message for the other player, namely Val_R^i (and Val_S^i), before interacting with the other player. This setting is challenging as observed in [21].

Note that the adversary A_i plays the CPA game only at the i-th stage game, and does not communicate with other adversaries. Thus, it is possible to achieve a negligible advantage for every state game although we define the whole CPA game as an infinitely-repeated game.

Next, we define the utility functions in the repeated CPA game. In repeated games, the *discount factor* $\delta > 0$ is employed so that the utility of the i-th stage game is discounted by the factor δ^i. We assume that rational players calculate their utilities as if the stage games will be played infinitely.

Definition 3. *For a discount factor* $\delta \in (0,1)$*, the utility of player* $w \in \{S, R\}$ *when the players follow a pair of strategies* (σ_S, σ_R) *is defined by*

$$U_w(\sigma_S, \sigma_R) = -c_w^{rand} \cdot \mathsf{Num}_w^{gen} + \sum_{i=0}^{\infty} \delta^i u_w[i],$$

where $(\mathsf{Num}_S^{gen}, \mathsf{Num}_R^{gen}, \{\mathsf{Adv}^i, \mathsf{Val}_S^i, \mathsf{Val}_R^i, \mathsf{Num}_S^i, \mathsf{Num}_R^i\}_{i=1,\dots})$ *is the outcome of the game* $\mathbf{Game}^{\mathrm{rep}}(\Pi, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S, \sigma_R)$*,* $u_w[i]$ *is the utility of player* w *in the* i*-th stage game, defined by*

$$u_w[i] = u_w^{sec} \cdot (-\widetilde{\mathsf{Adv}}^i) \cdot \mathsf{Val}_w^i - c_w^{rand} \cdot \mathsf{Num}_w^i,$$

and $c_w^{rand}, u_w^{sec} \in \mathbb{R}$ *represent the cost of generating randomness and the utility when the message is sent securely, respectively, and* $\widetilde{\mathsf{Adv}}^i$ *is equal to* Adv^i *except that* $\widetilde{\mathsf{Adv}}^i = 0$ *if* Adv^i *is a negligible function in* λ*.*

We assume that $u_w^{sec} > c_w^{rand} > 0$*, which implies that the security is worth paying the cost of generating randomness.*

A round version $\widetilde{\mathsf{Adv}}^i$ of Adv^i is introduced for the simplicity of arguments. In addition, we assume that rational players are not concerned about a negligible advantage of their utility.

Definition 4 (Nash equilibrium)**.** *A pair of strategies* (σ_S, σ_R) *is called a* Nash equilibrium *if for every* $w \in \{S, R\}$ *and strategy* σ'_w*, it holds that* $U_w(\sigma_S^*, \sigma_R^*) \leq U_w(\sigma_S, \sigma_R)$*, where* $(\sigma_S^*, \sigma_R^*) = (\sigma'_S, \sigma_R)$ *if* $w = S$*, and* $(\sigma_S^*, \sigma_R^*) = (\sigma_S, \sigma'_R)$ *otherwise.*

We define the security of encryption schemes for rational players. For an encryption scheme Π and a pair of strategies (σ_S, σ_R), we require that (1) when the players follow (σ_S, σ_R), Π is secure in every stage game, and (2) the strategy of following (σ_S, σ_R) is a Nash equilibrium.

Definition 5. *Let* $\Pi = (\{\mathsf{Gen}_w\}_{w\in\{S,R\}}, \{\mathsf{Enc}_i\}_{i\in\{1,\dots,n\}}, \mathsf{Dec})$ *be a public-key encryption scheme, and* (σ_S, σ_R) *a pair of strategies of the game* $\mathbf{Game}^{\mathrm{rep}}$*. We say* $(\Pi, \sigma_S, \sigma_R)$ *is* CPA secure with a Nash equilibrium *if for any sequence of probabilistic polynomial-time adversaries* $A = (A_1, A_2, \dots)$*, and any sets of message spaces* $\mathcal{M}_S$ *and* $\mathcal{M}_R$*,*

1. *there is a negligible function* $\varepsilon(\cdot)$ *such that* $\mathsf{Adv}_i \leq \varepsilon(\lambda)$ *for every* i *in* $\mathbf{Game}^{\mathrm{rep}}(\Pi, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S, \sigma_R)$*; and*
2. (σ_S, σ_R) *is a Nash equilibrium in* $\mathbf{Game}^{\mathrm{rep}}(\Pi, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S, \sigma_R)$*.*

3 Two-Round Scheme

We propose a two-round scheme that achieves a CPA security with a Nash equilibrium. The scheme can be based on any usual CPA-secure encryption scheme. In the key generation phase, both the sender and the receiver generate their own public key and secret key. In the encryption phase, a key agreement protocol is conducted to share a key. The shared key has the property such that it is a uniformly random string if and only if both players use good randomness in the encryption phase.

The above two-round scheme is not secure in a *one-shot* game, since if a message to be sent is valuable only to the receiver, the sender never uses good randomness. We overcome the insecurity in a one-shot game by a mechanism of infinitely-repeated games. In repeated games, each player can choose an action depending on the actions in the previous stage games. We employ a *grim trigger* strategy as a punishment strategy in repeated games. Initially, players choose good randomness in the encryption phase regardless of the value of a message to be sent. In any stage game, if some player chooses bad randomness, then bad randomness will be chosen in every subsequent game. This strategy is effective when valuable messages to each player will be chosen with at least a certain probability. The mechanism is similar to the repeated prisoners' dilemma.

We present a formal description of our two-round scheme $\Pi_{\mathsf{two}} = (\{\mathsf{Gen}_w\}_{w\in\{S,R\}}, \{\mathsf{Enc}_1, \mathsf{Enc}_2\}, \mathsf{Dec})$ and the grim trigger strategy for the repeated CPA game. The scheme is based on a public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- $\mathsf{Gen}_w(1^\lambda)$: Generate $(pk_w, sk_w) \leftarrow \mathsf{Gen}(1^\lambda)$, and output (pk_w, sk_w). The state information is set to be $st_w^1 = (pk_w, pk_{\bar{w}}, sk_w)$, where $\bar{w} \in \{S, R\} \setminus \{w\}$.
- $\mathsf{Enc}_1(st_R^1)$: Sample $r_1 \in \{0,1\}^\lambda$ uniformly at random, compute $c_1 \leftarrow \mathsf{Enc}(pk_S, r_1)$, and output (c_1, st_R^2), where $st_R^2 = (st_R^1, r_1)$.

  $\mathsf{Enc}_2(m, c_1, st_S^1)$: Sample $r_2 \in \{0,1\}^\lambda$ uniformly at random and compute $c_2 \leftarrow \mathsf{Enc}(pk_R, r_2)$ and $\hat{r}_1 \leftarrow \mathsf{Dec}(sk_S, c_1)$. Then, set $r = \hat{r}_1 \circ r_2$, compute $c_3 \leftarrow m \oplus r$, and output (c_2, c_3), where $\circ$ denote the concatenation operation.
- $\mathsf{Dec}(c_2, c_3, st_R^2)$: Compute $\hat{r}_2 \leftarrow \mathsf{Dec}(sk_R, c_2)$ and $\hat{r} = r_1 \oplus \hat{r}_2$. Then output $\hat{m} = c_3 \oplus \hat{r}$.

The above scheme Π_{two} is similar to the three-round scheme presented in [21]. In the scheme, a shared key in the encryption phase has a property such that it is a uniformly-random string if one of the sender and the receiver uses good randomness. This scheme is not secure without the final step. This is because the sender does not have any incentive to use good randomness in the key generation phase. The sender can achieve their own security only by using good randomness in the encryption phase. To prevent such laziness of the sender, we need the final step in which the receiver encrypts a recovered message using the sender's public key and makes it public. In repeated CPA games, the final step is not needed since a punishment can be imposed in subsequent stage games.

The grim trigger strategy $(\sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$ for repeated CPA games is defined as follows.

- For each $w \in \{S, R\}$, $\sigma_w^{\mathsf{tri}}(1^\lambda, \mathcal{M}_w)$ outputs $(a_w^{gen}, st_w) = (\mathsf{Good}, \mathsf{Good})$.
- For each $w \in \{S, R\}$, $\sigma_w^{\mathsf{tri}}(pk_S, pk_R, sk_w, st_w^i, aux_w)$ outputs $(a_w^{enc,i}, st_w^{i+1})$, where $st_w^{i+1} = (st_w^i, a_w^{enc,i}, a_{\bar{w}}^{enc,i-1})$, $\bar{w} \in \{S, R\} \setminus \{w\}$, and $a_w^{enc,i} = \mathsf{Good}$ if $a_x^{enc,1} = \cdots = a_x^{enc,i-1} = \mathsf{Good}$ for every $x \in \{S, R\}$, and $a_w^{enc,i} = \mathsf{Bad}$ otherwise.

Recall that p_S, p_R, p_{SR} denote the probabilities that chosen messages in repeated CPA games are in $\mathcal{M}_S \setminus \mathcal{M}_R$, $\mathcal{M}_R \setminus \mathcal{M}_S$, and $\mathcal{M}_S \cap \mathcal{M}_R$, respectively. We show that the scheme Π_{two} is secure under the trigger strategy if valuable messages to each player will be chosen in stage games with at least a certain probability.

Theorem 1. *The tuple* $(\Pi_{\mathsf{two}}, \sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$ *is CPA secure with a Nash equilibrium if* $p_w + p_{SR} > \max\{(2-\delta)(c_w^{rand}/u_w^{sec}), c_w^{rand}/(\delta u_w^{sec})\}$ *for each* $w \in \{S, R\}$.

Proof. Let $A = (A_1, A_2, \dots)$ be a sequence of probabilistic polynomial-time adversaries, and $\mathcal{M}_S$ and $\mathcal{M}_R$ sets of message spaces.

First, we show that there is a negligible function $\varepsilon(\cdot)$ such that $\mathsf{Adv}_i \le \varepsilon(\lambda)$ for every i in $\mathbf{Game}^{\mathrm{rep}}(\Pi_{\mathsf{two}}, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$. Since the players follow $(\sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$, they choose good randomness at the key generation phase and every encryption phase. For the i-th stage game, suppose A_i chooses m_0, m_1 as a pair of challenge messages. The view of A_i is

$$
\begin{aligned}
&\{pk_S, pk_R, m_0, m_1, \mathsf{Enc}(pk_S, r_1), \mathsf{Enc}(pk_R, r_2), m_b \oplus (r_1 \circ r_2)\} \\
&\approx_c \{pk_S, pk_R, m_0, m_1, \mathsf{Enc}(pk_S, r_1'), \mathsf{Enc}(pk_R, r_2'), m_b \oplus (r_1 \circ r_2)\} \\
&= \{pk_S, pk_R, m_0, m_1, \mathsf{Enc}(pk_S, r_1'), \mathsf{Enc}(pk_R, r_2'), r\} \\
&= \{pk_S, pk_R, m_0, m_1, \mathsf{Enc}(pk_S, r_1'), \mathsf{Enc}(pk_R, r_2'), m_{1-b} \oplus (r_1 \circ r_2)\} \\
&\approx_c \{pk_S, pk_R, m_0, m_1, \mathsf{Enc}(pk_S, r_1), \mathsf{Enc}(pk_R, r_2), m_{1-b} \oplus (r_1 \circ r_2), \}
\end{aligned}
$$

where $r_1, r_2, r'_1, r'_2 \in \{0,1\}^\lambda$ and $r \in \{0,1\}^{2\lambda}$ are uniformly random strings. We have used the CPA security of the underlying scheme Π for the relation $\approx_c$. The above implies that there is a negligible function $\varepsilon(\cdot)$ such that $\mathsf{Adv}_i \le \varepsilon(\lambda)$ for every i.

Next, we show that the pair of strategies $(\sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$ is a Nash equilibrium in $\mathbf{Game}^{\mathrm{rep}}(\Pi_{\mathsf{two}}, \lambda, A, \mathcal{M}_S, \mathcal{M}_R, \sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}})$. It follows from the above security analysis that

$$\begin{aligned} U_w(\sigma_S^{\mathsf{tri}}, \sigma_R^{\mathsf{tri}}) &\ge -c_w^{rand} + \sum_{i=0}^{\infty} \delta^i(-c_w^{rand}) \\ &\ge -c_w^{rand} - \frac{c_w^{rand}}{1-\delta} = -\frac{(2-\delta)c_w^{rand}}{1-\delta}. \end{aligned} \tag{1}$$

Let consider the case that player $w \in \{S, R\}$ chooses Bad in the key generation phase. Suppose S chose Bad in the key generation phase. (The same argument can be applied to the case for R.) The adversary A_i can obtain r_1 by decrypting $\mathsf{Enc}(pk_S, r_1)$ by using sk_S, which is a part of the output of $\mathsf{Gen}(1^\lambda; 0^\lambda)$. Thus A_i can win every stage game by choosing m_0, m_1 such that the first halves of m_0 and m_1 are different, and, on receiving c_3, outputting $b' = 0$ if the first half of $c_3 \oplus (r_1 \circ 0^\lambda)$ is equal to that of m_0, and $b' = 1$ otherwise. Since the advantage $\widetilde{\mathsf{Adv}}^i$ will be 1 for every i, the utility U_S will be at most

$$\sum_{i=0}^{\infty} \left(\delta^i u_S^{sec}(-1)(p_S + p_{SR})\right) = -\frac{(p_S + p_{SR})u_S^{sec}}{1-\delta},$$

which is less than (1) since $p_S + p_{SR} > (2-\delta)(c_S^{rand}/u_S^{sec})$. Thus, each player cannot increase the utility by choosing Bad in the key generation phase.

In the following, we consider players who choose Good in the key generation phase, but may choose Bad in the encryption phase. In the analysis, we use the following three values for the utility $u_w[i]$ of the i-th stage game:

- $u_1 = 0$, which is the case that $\mathsf{Val}_w^i = 0$ and $\mathsf{Num}_w^i = 0$;
- $u_2 = -c_w^{rand}$, which is the case that $\mathsf{Val}_w^i = 0, \mathsf{Num}_w^i = 1$ or that $\mathsf{Adv}_i = 0, \mathsf{Val}_w = 1, \mathsf{Num}_w^i = 1$;
- $u_3 = -u_w^{sec}$, which is the case that $\widetilde{\mathsf{Adv}}^i = 1$, $\mathsf{Val}_w^i = 1$, and $\mathsf{Num}_w^i = 0$.

Note that $u_1 > u_2 > u_3$.

Suppose that player R follows σ_R^{tri}, and player S chooses Good for stages $i = 1, \ldots, r-1$, but chooses Bad at the r-th stage game. Then, the utility $u_S[i]$ for the i-th stage game is u_2 for $i = 0, \ldots, r-1$ since both players chooses Good, and thus $\widetilde{\mathsf{Adv}}^i = 0$. After the r-th stage, player R uses Bad in every subsequent stage game since R follows σ_R^{tri}. Then, an adversary A_i can win the i-th stage game for every $i > r$. Since R uses Bad, $r_1 = 0^\lambda$. Thus A_i can win the game by choosing m_0, m_1 such that the first halves of m_0 and m_1 are different, and outputting $b' = 0$ if the first half of c_3 is equal to that of m_0, and $b' = 1$ otherwise. For such A_i's, $u_S[i]$ is u_1 with probability p_R, and u_3 with probability $p_S + p_{SR}$. Thus, the utility U_S is at most

$$\begin{aligned} &-c_w^{rand} + u_2 + \delta u_2 + \cdots + \delta^{r-1}u_2 + \delta^r u_1 + \sum_{i=r+1}^{\infty} \delta^i \left(p_R u_1 + (p_S + p_{SR})u_3\right) \\ &= -c_w^{rand} + \frac{1-\delta^r}{1-\delta}u_2 + \delta^r u_1 + \frac{\delta^{r+1}}{1-\delta}\left(p_R u_1 + (p_S + p_{SR})u_3\right) \\ &= -\frac{(2-\delta-\delta^r)c_S^{rand} + \delta^{r+1}(p_S + p_{SR})u_S^{sec}}{1-\delta}, \end{aligned}$$

which is less than (1) since $p_S + p_{SR} > c_S^{rand}/(\delta u_S^{sec})$.

The same argument holds when player S follows σ_S^{tri}, but player R tries to choose Bad in the encryption phase. Thus, both players S and R cannot increase their utility by changing their strategies from $(\sigma_S^{tri}, \sigma_R^{tri})$. Hence, $(\sigma_S^{tri}, \sigma_R^{tri})$ is a Nash equilibrium, and thus the statement follows. □

4 Conclusions

In this work, we employed repeated games for reducing the round complexity of encryption schemes performed by rational players. Our two-round scheme achieves a Nash equilibrium in the repeated CPA games. However, a Nash equilibrium may not be a satisfying solution concept in repeated games. The notion of a *subgame-perfect equilibrium* is known as a stronger solution concept in repeated games. Thus, one of the future work is to achieve a subgame-perfect equilibrium in our framework. For the one-shot game, a stronger solution concept, *strict Nash equilibrium*, was achieved in the previous work [21]. Another future work is to explore the possibility of the mechanism of repeated games for other cryptographic primitives. Since a repeated game models a long-term relationship, the mechanism may be applied to cryptographic protocols by considering a long-term relationship.

Acknowledgements

This work was supported in part by JSPS/MEXT Grant-in-Aid for Scientific Research Numbers 24240001, 25106509, 15H00851, and 16H01705.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In E. Ruppert and D. Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006*, pages 53–62. ACM, 2006.

[2] G. Asharov, R. Canetti, and C. Hazay. Toward a game theoretic view of secure computation. *J. Cryptology*, 29(4):879–926, 2016.

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

[4] D. R. L. Brown. A weak-randomizer attack on RSA-OAEP with e = 3. *IACR Cryptology ePrint Archive*, 2005:189, 2005.

[5] L. Dorrendorf, Z. Gutterman, and B. Pinkas. Cryptanalysis of the windows random number generator. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, pages 476–485. ACM, 2007.

[6] G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In D. Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2010.

[7] J. A. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 648–657. IEEE Computer Society, 2013.

[8] A. Groce and J. Katz. Fair computation with rational players. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 7237 of *Lecture Notes in Computer Science*, pages 81–98. Springer, 2012.

[9] Z. Gutterman and D. Malkhi. Hold your sessions: An attack on java session-id generation. In A. Menezes, editor, *Topics in Cryptology - CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2005.

[10] Z. Gutterman, B. Pinkas, and T. Reinman. Analysis of the linux random number generator. In *2006 IEEE Symposium on Security and Privacy (S&P 2006)*, pages 371–385. IEEE Computer Society, 2006.

[11] J. Y. Halpern and R. Pass. Algorithmic rationality: Game theory with costly computation. *J. Economic Theory*, 156:246–268, 2015.

[12] J. Y. Halpern, R. Pass, and L. Seeman. Not just an empty threat: Subgame-perfect equilibrium in repeated games played by computationally bounded players. In T. Liu, Q. Qi, and Y. Ye, editors, *Web and Internet Economics - 10th International Conference, WINE 2014*, volume 8877 of *Lecture Notes in Computer Science*, pages 249–262. Springer, 2014.

[13] J. Y. Halpern, R. Pass, and L. Seeman. The truth behind the myth of the folk theorem. In M. Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14*, pages 543–554. ACM, 2014.

[14] J. Y. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In L. Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 623–632. ACM, 2004.

[15] R. Halprin and M. Naor. Games for extracting randomness. *ACM Crossroads*, 17(2):44–48, 2010.

[16] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In T. Kohno, editor, *Proceedings of the 21th USENIX Security Symposium*, pages 205–220. USENIX Association, 2012.

[17] H. Higo, K. Tanaka, A. Yamada, and K. Yasunaga. Game-theoretic security for two-party protocols. *IACR Cryptology ePrint Archive*, 2016:1072, 2016.

[18] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Public keys. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference*, volume 7417 of *Lecture Notes in Computer Science*, pages 626–642. Springer, 2012.

[19] S. Maleka, A. Shareef, and C. P. Rangan. Rational secret sharing with repeated games. In L. Chen, Y. Mu, and W. Susilo, editors, *Information Security Practice and Experience, 4th International Conference, ISPEC 2008*, volume 4991 of *Lecture Notes in Computer Science*, pages 334–346. Springer, 2008.

[20] P. Q. Nguyen and I. E. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *J. Cryptology*, 15(3):151–176, 2002.

[21] K. Yasunaga. Public-key encryption with lazy parties. *IEICE Transactions*, 99-A(2):590–600, 2016.

[22] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: results from the 2008 debian openssl vulnerability. In A. Feldmann and L. Mathy, editors, *Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference, IMC 2009*, pages 15–27. ACM, 2009.