Computationally Sound Formalization of Rerandomizable RCCA Secure Encryption

Yusuke Kawamoto[1] Hideki Sakurada[2] Masami Hagiya[1]

[1] Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, JAPAN
{y_kwmt, hagiya} *at* is.s.u-tokyo.ac.jp
[2] NTT Communication Science Laboratories, NTT Corporation
3-1, Morinosato Wakamiya, Atsugi-shi, Kanagawa 243-0198, JAPAN
sakurada *at* theory.brl.ntt.co.jp

Abstract. Rerandomizing ciphertexts plays an important role in protecting privacy in security protocols such as mixnets. We investigate the relationship between formal and computational approaches to the analysis of the security protocols using a rerandomizable encryption scheme. We introduce a new method of dealing with composed randomnesses in an Abadi-Rogaway-style pattern, formalize a rerandomizable RCCA secure encryption scheme, and prove its computational soundness.

1 Introduction

Formal and computational approaches have developed separately in research related to the analysis of security protocols. In the formal approach, a cryptographic message is abstracted into a symbol, called a Dolev-Yao term, and an adversary can only perform several algebraic operations on Dolev-Yao terms [11]. The formal analysis of security protocols is based on the assumption that cryptography is perfectly secure. On the other hand, in the computational approach, a message is a bit string and an adversary is a probabilistic polynomial-time (PPT) algorithm. The computational analysis of security protocols deals with the probability of the adversary performing a successful attack from a complexity-theory perspective.

These two approaches have advantages and disadvantages. Although the formal approach is simpler and amenable to automation, it is based on the unrealistically strong assumption as regards cryptography. While the analysis in the computational approach employs more realistic models, it is very difficult and prone to errors.

In recent years, many researches have related these two approaches [2,1,18]. They define a function, called an encoding, that maps a Dolev-Yao term to a probability distribution over bit strings, and prove the soundness theorem, which claims that the formal equivalence of Dolev-Yao terms implies the computational indistinguishability of the encodings of the terms. This theorem guarantees that the analysis of security protocols in the formal approach is also valid from the viewpoint of the computational approach.

Most of the previous studies related to soundness theorems have dealt with cryptographic primitives with relatively strong computational security, such as IND-CCA2 public key encryption [18], EUF-CMA digital signature [10], and oracle hashing [12]. Although there has been a lot of work on the formal analysis of the security protocols with more complex primitives such as homomorphic encryption [8], their soundness theorems have not been proved.

As a first step to obtaining soundness results for more complex message algebras, we deal with a rerandomizable encryption scheme, which is an encryption scheme with a re-encryption operation that replaces the randomness used in a ciphertext with another without decrypting the ciphertext. Although the randomness used in a probabilistic encryption enables an adversary to observe the occurrences of the same ciphertext, rerandomizing ciphertexts prevents the adversary from tracing them. For this reason, the re-encryption operation plays an important role in protecting privacy in some security protocols such as mixnet [13].

We propose a new formalization of a rerandomizable encryption scheme using Abadi-Rogaway-style formal patterns [2,15], and prove its computational soundness by using the IND-RCCA security [6] of the rerandomizable encryption scheme and the randomness-preserving property of the randomness composition.

In the formalization, we introduce a new method of dealing with composed randomnesses, because the re-encryption operation follows the composition of the randomnesses used in probabilistic encryptions/re-encryptions. Although some studies explicitly represent the randomnesses of probabilistic encryptions in an Abadi-Rogaway-style pattern [12,7,9], they do not deal with the composition of randomnesses. We extend Herzog's formalization [15] to explicitly represent composed randomnesses by a multiset of randomness symbols in an Abadi-Rogaway-style pattern. Due to this, patterns are expressive enough to describe the indistinguishability of a ciphertext from its rerandomization. In addition, we provide a new definition of a renaming of a multiset of randomness symbols that enables us to deal with composed randomnesses. To obtain the soundness result, we deal only with acyclic terms satisfying the freshness assumption, which restricts the usage of honest participants' randomnesses.

In the soundness theorem, we claim that if patterns cannot be distinguished by the Dolev-Yao adversary, then their computational encodings cannot be distinguished by any PPT adversary with access to a decryption oracle. Here, the decryption oracle represents a certain aspect of an active and adaptive adversary. Since the computational indistinguishability introduced in this paper is an extension of Herzog's Abadi-Rogaway public-key indistinguishability [15] to IND-RCCA security, the PPT adversary in our model is not fully active or adaptive. For example, the adversary's nonces and randomnesses are fixed in advance and not adaptive.

The organization of this paper is as follows. Section 2 defines the formal model employed to analyze the security protocols using rerandomizable encryption schemes. Section 3 introduces a computational rerandomizable encryption scheme, and its computational security definitions. Section 4 defines an encoding that maps patterns to distributions over bit strings. Section 5 introduces Abadi-Rogaway RCCA indistinguishability, and proves the soundness theorem. The final section summarizes our work and discusses areas for future research.

2 Formal Model

This section introduces the message algebra used to formalize and analyze the protocols that employs rerandomizable encryption schemes.

2.1 Dolev-Yao Model

Our formal model is an extension of the Dolev-Yao model presented in [15]. A message is abstracted into a term from an appropriate algebra, called a Dolev-Yao term [11], and parties are restricted to performing only pairing, encryption, decryption, and re-encryption operations. There are two kinds of parties: honest participants and an active and adaptive adversary. The honest participants follow a protocol without deviation, and can run multiple sessions of the protocol simultaneously.

The communications between parties are under the control of the adversary. In the same way as in [15], we model the adversary as the communication channel, and assume that the adversary can record, delete, replay, and reorder messages. Each execution of a protocol is defined as an alternating finite sequence of the adversary's messages q_i and honest participants' messages r_i: $r_0, q_1, r_1, q_2, \cdots, r_{n-1}, q_n, r_n$. We assume that the adversary receives the initial knowledge r_0 before executing the protocol, and that each adversary's message q_i must be derivable from $r_0, r_1, \cdots, r_{i-1}$, nonces, and randomnesses. Although the analysis of a security protocol in this model must take account of all non-deterministic choices of the adversary's messages, we do not present an analysis method in the model.

This Dolev-Yao model is explained in detail in [15], and here we concentrate on providing a formalization of a rerandomizable encryption scheme.

2.2 Terms

We define the following sets of atomic symbols, which are mutually exclusive:

- a set *Const* of *constants*, denoting plaintexts of messages for example,
- a set K_{pub} of *public key symbols*,
- a set K_{sec} of *secret key symbols*,
- a set *Nonce* of *nonce symbols*, and
- a set *Rand* of *randomness symbols*, denoting the randomnesses used in encryption.

We denote the secret key corresponding to a public key k_{pub} by $\overline{k_{pub}}$, and the public key corresponding to a secret key k_{sec} by $\overline{k_{sec}}$. Let $K_{adv} \subseteq K_{sec}$ be a finite set of the secret keys of subverted participants.

Let $Nonce_{uni}$ be a set of the nonce symbols of honest participants, $Nonce_{adv}$ be a set of the nonce symbols of the adversary, and $Nonce = Nonce_{uni} \cup Nonce_{adv}$.

Let $Rand_{uni}$ be a set of the randomness symbols denoting uniform randomnesses used only in honest participants' probabilistic encryptions, $Rand_{adv}$ be a set of the randomness symbols used in the adversary's probabilistic encryptions, and $Rand = Rand_{uni} \cup Rand_{adv}$. For a set X, let $FMulti(X)$ be the set of all the non-empty finite multisets of X's elements. Let $X_1 \uplus X_2$ be the disjoint union of two multisets X_1 and X_2.

Using these atomic symbols, a *term* is constructed from a pairing $\langle \_, \_ \rangle$, encryption $\{\!|\, \_ \,|\!\}^{\_}_{\_}$, and re-encryption $(\!|\, \_ \,|\!)^{\_}_{\_}$ operations as follows:

$$\mathit{Term} \ni m ::= c \mid k_{pub} \mid k_{sec} \mid n \mid R \mid \langle m, m \rangle \mid \{\!|\, m \,|\!\}^{R}_{k_{pub}} \mid (\!|\, m \,|\!)^{R}_{k_{pub}},$$

where $c \in \mathit{Const}$, $k_{pub} \in K_{pub}$, $k_{sec} \in K_{sec}$, $n \in \mathit{Nonce}$, and a non-empty finite multiset $R \in \mathit{FMulti}(\mathit{Rand})$. Here the multiset R denotes the randomness composed of all the randomnesses in R. We assume that the value of the composition of the randomnesses in R is uniquely determined.

A term of the form $\langle m_1, m_2 \rangle$ denotes the pair of two messages m_1 and m_2. A term of the form $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ denotes the encryption of a message m by a public key k_{pub} and a composed randomness R. For example, $\{\!|\, m \,|\!\}^{R \uplus R'}_{k_{pub}}$ denotes the encryption of a message m by a public key k_{pub} and the randomness composed of R and R'. A term of the form $(\!|\, m \,|\!)^{R}_{k_{pub}}$ denotes the re-encryption of a ciphertext m by a public key k_{pub} and a composed randomness R. We sometimes abbreviate $\{\!|\, m \,|\!\}^{\{r\}}_{k_{pub}}$ and $(\!|\, m \,|\!)^{\{r\}}_{k_{pub}}$ as $\{\!|\, m \,|\!\}^{r}_{k_{pub}}$ and $(\!|\, m \,|\!)^{r}_{k_{pub}}$, respectively. We can derive a term m from $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ by decrypting $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ using the corresponding secret key $\overline{k_{pub}}$.

2.3 Patterns

This section defines a pattern $\mathit{pattern}(m, T)$ for a term m and a set $T \subseteq K_{sec}$. The intuitive meaning of a pattern $\mathit{pattern}(m, T)$ is the bit string distribution associated with m from the viewpoint of the formal adversary with access to the secret keys T.

First, we introduce the *type trees* of terms [17,15]. We abuse the notation and use a type symbol as an atomic symbol of the same type. The type tree $\mathit{type}(m)$ of a term m is defined as follows: $\mathit{type}(c) = \mathit{Const}$ if $c \in \mathit{Const}$, $\mathit{type}(k_{pub}) = K_{pub}$ if $k_{pub} \in K_{pub}$, $\mathit{type}(k_{sec}) = K_{sec}$ if $k_{sec} \in K_{sec}$, $\mathit{type}(n) = \mathit{Nonce}$ if $n \in \mathit{Nonce}$, $\mathit{type}(R) = \mathit{FMulti}(\mathit{Rand})$ if $R \in \mathit{FMulti}(\mathit{Rand})$, $\mathit{type}(\langle m_1, m_2 \rangle) = \langle \mathit{type}(m_1), \mathit{type}(m_2) \rangle$, $\mathit{type}(\{\!|\, m \,|\!\}^{R}_{k_{pub}}) = \{\!|\, \mathit{type}(m) \,|\!\}^{\mathit{FMulti}(\mathit{Rand})}_{K_{pub}}$, and $\mathit{type}((\!|\, m \,|\!)^{R}_{k_{pub}}) = (\!|\, \mathit{type}(m) \,|\!)^{\mathit{FMulti}(\mathit{Rand})}_{K_{pub}}$. For example, $\mathit{type}(\langle c, \{\!|\, n \,|\!\}^{R}_{k_{pub}} \rangle) = \langle \mathit{Const}, \{\!|\, \mathit{Nonce} \,|\!\}^{\mathit{FMulti}(\mathit{Rand})}_{K_{pub}} \rangle$ holds.

Next, we define the *undecryptable ciphertext symbol* $\Box^{\{\!|\, \mathit{type}(m) \,|\!\}^{R}_{k_{pub}}}$ for each ciphertext $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ to introduce the pattern representing the distribution of the ciphertext that the adversary cannot decrypt. Intuitively, $\Box^{\{\!|\, \mathit{type}(m) \,|\!\}^{R}_{k_{pub}}}$ denotes a random message from which the adversary cannot distinguish the ciphertext $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ when $R \cap \mathit{Rand}_{uni} \neq \emptyset$ holds. The notation $\Box^{\{\!|\, \mathit{type}(m) \,|\!\}^{R}_{k_{pub}}}$ implies that the encryption $\{\!|\, m \,|\!\}^{R}_{k_{pub}}$ reveals the public key k_{pub} and the length of the plaintext m. The set R of randomness symbols in $\Box^{\{\!|\, \mathit{type}(m) \,|\!\}^{R}_{k_{pub}}}$ is used to analyze the relations between probability distributions.

Finally, we define the pattern associated with a term.

Definition 1. A set *Pattern* of patterns is defined by:

$$\mathit{Pattern} \ni m ::= c \mid k_{pub} \mid k_{sec} \mid n \mid R \mid \langle m, m \rangle \mid \{\!|\, m \,|\!\}^{R}_{k_{pub}} \mid (\!|\, m \,|\!)^{R}_{k_{pub}} \mid \Box^{\{\!|\, \mathit{type}(m) \,|\!\}^{R}_{k_{pub}}},$$

where $c \in \mathit{Const}$, $k_{pub} \in K_{pub}$, $k_{sec} \in K_{sec}$, $n \in \mathit{Nonce}$, and a non-empty finite multiset $R \in \mathit{FMulti}(\mathit{Rand})$.

Definition 2. For $k_{pub} \in K_{pub}$, let $Reenc_{k_{pub}}$ be the minimum set of terms recursively defined by:

- $\{\!|\, m \,|\!\}^{R}_{k_{pub}} \in Reenc_{k_{pub}}$ holds for any $m \in Term$ and any $R \in FMulti(Rand)$.
- If $m_{re} \in Reenc_{k_{pub}}$ holds, then $(\!|\, m_{re} \,|\!)^{R}_{k_{pub}} \in Reenc_{k_{pub}}$ holds for any $R \in FMulti(Rand)$.

Note that each $m \in Reenc_{k_{pub}}$ is generated by repeated encryption/re-encryption operations using the same public key k_{pub}.

Definition 3. For a set $T \subseteq K_{sec}$, let $\overline{T}$ be the set $\{\, \overline{k_{sec}} \mid k_{sec} \in T \,\}$. For $m \in Term$ and $T \subseteq K_{sec}$, we define the sets $F(m, T)$ and $G_i(m, T)$ of all the secret keys that the formal adversary can learn from m using the secret keys in T and $G_{i-1}(m, T)$, respectively.

- $F(m, T) = T \quad (\text{if } m \in Const \cup K_{pub} \cup Nonce \cup FMulti(Rand))$
- $F(k_{sec}, T) = \{\, k_{sec} \,\} \cup T$
- $F(\langle m_1, m_2 \rangle, T) = F(m_1, T) \cup F(m_2, T)$
- $F(\{\!|\, m \,|\!\}^{R}_{k_{pub}}, T) = \begin{cases} F(m, T) & (\text{if } k_{pub} \in \overline{T} \text{ or } R \in FMulti(Rand_{adv})) \\ T & (\text{otherwise}) \end{cases}$
- $F((\!|\, m \,|\!)^{R}_{k_{pub}}, T) = \begin{cases} F(\{\!|\, m' \,|\!\}^{R \uplus R'}_{k_{pub}}, T) & (\text{if } R \in FMulti(Rand) \setminus FMulti(Rand_{adv}), \\ & \ \text{and } m = \{\!|\, m' \,|\!\}^{R'}_{k_{pub}} \text{ holds for } m' \in Term \\ & \ \text{and } R' \in FMulti(Rand)) \\ F((\!|\, m' \,|\!)^{R \uplus R'}_{k_{pub}}, T) & (\text{if } R \in FMulti(Rand) \setminus FMulti(Rand_{adv}), \\ & \ \text{and } m = (\!|\, m' \,|\!)^{R'}_{k_{pub}} \text{ holds for } m' \in Reenc_{k_{pub}} \\ & \ \text{and } R' \in FMulti(Rand)) \\ F(m, T) & (\text{otherwise}) \end{cases}$
- $G_0(m, T) = T$
- $G_i(m, T) = F(m, G_{i-1}(m, T))$

We define the function $recoverable\colon Term \times \mathcal{P}(K_{sec}) \to \mathcal{P}(K_{sec})$ that maps a term m and a set $T \subseteq K_{sec}$ to the set of all the secret key symbols recoverable from m by using T.

- $recoverable(m, T) = G_{|m|}(m, T)$

We define the function $pat\colon Term \times \mathcal{P}(K_{sec}) \to Pattern$ that maps a term t and a set $T \subseteq K_{sec}$ to t's pattern with respect to T.

- $pat(m, T) = m \quad (\text{if } m \in Const \cup K_{pub} \cup K_{sec} \cup Nonce \cup FMulti(Rand))$
- $pat(\langle m_1, m_2 \rangle, T) = \langle pat(m_1, T), pat(m_2, T) \rangle$
- $pat(\{\!|\, m \,|\!\}^{R}_{k_{pub}}, T) = \begin{cases} \{\!|\, pat(m, T) \,|\!\}^{R}_{k_{pub}} & (\text{if } k_{pub} \in \overline{T} \text{ or } R \in FMulti(Rand_{adv})) \\ \Box^{\{\!|\, type(m) \,|\!\}^{R}_{k_{pub}}} & (\text{otherwise}) \end{cases}$
- $pat((\!|\, m \,|\!)^{R}_{k_{pub}}, T) = \begin{cases} pat(\{\!|\, m' \,|\!\}^{R \uplus R'}_{k_{pub}}, T) & (\text{if } R \in FMulti(Rand) \setminus FMulti(Rand_{adv}), \\ & \ \text{and } m = \{\!|\, m' \,|\!\}^{R'}_{k_{pub}} \text{ holds for } m' \in Term \\ & \ \text{and } R' \in FMulti(Rand)) \\ pat((\!|\, m' \,|\!)^{R \uplus R'}_{k_{pub}}, T) & (\text{if } R \in FMulti(Rand) \setminus FMulti(Rand_{adv}), \\ & \ \text{and } m = (\!|\, m' \,|\!)^{R'}_{k_{pub}} \text{ holds for } m' \in Reenc_{k_{pub}} \\ & \ \text{and } R' \in FMulti(Rand)) \\ (\!|\, pat(m, T) \,|\!)^{R}_{k_{pub}} & (\text{otherwise}) \end{cases}$

Let $pattern\colon Term \times \mathcal{P}(K_{sec}) \to Pattern$ be the function defined by:

$$pattern(m, T) = pat(m, recoverable(m, T)).$$

In the above definition, $pattern(m, T)$ represents the information that the formal adversary can obtain from the message m using the decryption keys in T. We assume that the formal adversary can see any messages encrypted using a non-uniform randomness. We also assume that he can see the message c in a re-encryption $(\!|\, c \,|\!)^{R}_{k_{pub}}$ if c is not a valid encryption using k_{pub}. In addition, the above definition reflects that the re-encryption of a ciphertext $\{\!|\, m \,|\!\}^{R'}_{k_{pub}}$ by the same public key k_{pub} and a randomness R produces the ciphertext $\{\!|\, m \,|\!\}^{R \uplus R'}_{k_{pub}}$ using the randomness composed of R and R'.

2.4 Acyclicity and Freshness Assumption

First, we introduce a subterm relation. Given a term m, the set $\mathsf{SubTerm}$ of all the subterms of m is recursively defined as follows: $\mathsf{SubTerm}(m) = \{m\}$ if $m \in Const \cup K_{pub} \cup K_{sec} \cup Nonce \cup Rand$, $\mathsf{SubTerm}(m) = \{m\} \cup \mathsf{SubTerm}(m_1) \cup \mathsf{SubTerm}(m_2)$ if $m = \langle m_1, m_2 \rangle$, and $\mathsf{SubTerm}(m) = \{m\} \cup \mathsf{SubTerm}(m')$ if $m = \{\!|\, m' \,|\!\}^{R}_{k_{pub}}$ or $m = (\!|\, m' \,|\!)^{R}_{k_{pub}}$. For two term m and m', we write $m' \sqsubseteq m$ if $m' \in \mathsf{SubTerm}(m)$.

Next, we define the acyclicity of terms in a similar way to that in [2,12].

Definition 4. A secret key symbol k *encrypts* a secret key symbol k' *in* a term m if $\{\!|\, m' \,|\!\}^{R}_{\overline{k}} \sqsubseteq m$ and $k' \sqsubseteq m'$ hold for some $R \in FMulti(Rand)$. A term is *acyclic* if there is no sequence $k_1, k_2, \cdots, k_n, k_{n+1} = k_1$ of secret key symbols such that k_i encrypts k_{i+1} in m for each $1 \leq i \leq n$.

The acyclicity of terms is necessary for us to obtain the soundness theorem.

Then, we define the independence of a uniform randomness symbol.

Definition 5. A uniform randomness symbol $r \in Rand_{uni}$ is *independent* in a set S of terms if there exist a unique multiset $R \in FMulti(Rand)$ such that

– $r \in R$ holds,
– R occurs in some $m \in S$, and
– $r \notin R'$ holds for every $R' \in FMulti(Rand)$ occurring in some $m' \in S$ with $R' \neq R$.

$r \in Rand_{uni}$ is *independent* in a term m if it is independent in $\{ m \}$.

Intuitively, if an independent randomness $r \in Rand_{uni}$ is used in an honest participant's probabilistic encryption/re-encryption, then it is not used in another encryption/re-encryption. For example, let S be the set $\{ \{\!|\, c_1 \,|\!\}^{r_1}_{k}, \{\!|\, c_1 \,|\!\}^{\{r_1, r_2\}}_{k}, \{\!|\, c_2 \,|\!\}^{r_3}_{k}, (\!|\, \{\!|\, c_2 \,|\!\}^{r_3}_{k} \,|\!)^{r_4}_{k} \}$ for $r_1, r_2, r_3, r_4 \in Rand_{uni}$. While r_2, r_3, and r_4 are independent in S, r_1 is not independent.

Finally, we introduce the following freshness assumption.

Definition 6. A multiset $R \in FMulti(Rand)$ *encrypts* a term m' in a term m if $\{\!|\, m' \,|\!\}^{R}_{k_{pub}} \sqsubseteq m$ holds for some public key symbol k_{pub}. A multiset $R \in FMulti(Rand)$ *re-encrypts* a term m' in a term m if $(\!|\, m' \,|\!)^{R}_{k_{pub}} \sqsubseteq m$ holds for some public key symbol k_{pub}. A term m *satisfies the freshness assumption* if it holds that for each $R \in FMulti(Rand) \setminus FMulti(Rand_{adv})$ occurring in m, there exist

– a unique term m' such that every occurrence of R encrypts/re-encrypts m' in m, and
– a uniform randomness symbol $r \in R \cap Rand_{uni}$ independent in m.

Intuitively, the former condition represents the fact that

– no honest participant uses the same composed randomness R to encrypt/re-encrypt another message, and that
– no honest participant uses the randomnesses in $Rand_{uni}$ except when employing them as the randomnesses in probabilistic encryptions/re-encryptions, that is, no uniform randomness symbols in $Rand_{uni}$ are used as plaintexts or keys in m.

The latter condition represents the fact that

– every randomness R used in an honest participant's encryption/re-encryption is composed of at least one independent and uniform randomness r which he never uses in another encryption/re-encryption.

For example, for $c_1, c_2 \in Const$, $k \in K_{pub}$, $r_1, r_2 \in Rand_{uni}$ and $r_{adv} \in Rand_{adv}$, the following four terms do not satisfy the freshness assumption: $\langle \{\!|\, c_1 \,|\!\}_k^{\{r_1\}}, \{\!|\, c_2 \,|\!\}_k^{\{r_1\}} \rangle$, $\{\!|\, r_1 \,|\!\}_k^{\{r_2\}}$, $\langle \{\!|\, c_1 \,|\!\}_k^{\{r_1\}}, \{\!|\, c_1 \,|\!\}_k^{\{r_1, r_2\}} \rangle$, and $\langle \{\!|\, c_1 \,|\!\}_k^{\{r_1\}}, \{\!|\, c_1 \,|\!\}_k^{\{r_1, r_{adv}\}} \rangle$. If two multisets $R, R' \in FMulti(Rand)$ with $R \subsetneq R'$ occur in a term m, then m does not satisfy the freshness assumption. Note that the freshness assumption allows honest participants to copy any ciphertexts.

Hereafter, we deal only with acyclic terms that satisfy the freshness assumption.

2.5 Observational Equivalence

This section defines the renaming of patterns and the observational equivalence of terms.

First, we introduce several notations and the renaming for atomic symbols.

Definition 7. Given $T \subseteq K_{sec}$, let $Atom_T = (K_{pub} \setminus \overline{T}) \cup (K_{sec} \setminus T) \cup Nonce_{uni} \cup (FMulti(Rand) \setminus FMulti(Rand_{adv}))$. Given $P \in Pattern$, let $Atom_T(P)$ be the following set: $\{ P' \in Atom_T \mid P' \text{ occurs in } P \}$.

Definition 8. Given $P \in Pattern$ and $T \subseteq K_{sec}$, a function σ is a *renaming for the atomic symbols in P except for T* if it is a type-preserving injection from $Atom_T(P)$ to $Atom_T$ such that $\sigma(k) = k'$ if and only if $\sigma(\overline{k}) = \overline{k'}$ for any $k, k' \in K_{pub} \setminus \overline{T}$.

Next, we define the renaming of a pattern.

Definition 9. Given a pattern $P \in Pattern$ and a renaming σ for the atomic symbols in P except for $T \subseteq K_{sec}$, we write $\tilde{\sigma} P$ to represent the pattern obtained by replacing each occurrence of $Q \in Atom_T(P)$ in P with $\sigma(Q)$.

Finally, we define the observational equivalence of terms.

Definition 10. Two terms m and m' are *observationally equivalent*, written as $m \cong m'$, if there exists a renaming σ for the atomic symbols in $pattern(m', K_{adv})$ except for K_{adv} such that $pattern(m, K_{adv}) = \tilde{\sigma}\, pattern(m', K_{adv})$.

Example 1. Let $k, k_1, k_2 \in K_{pub} \setminus \overline{K_{adv}}$ and $r_1, r_2 \in Rand_{uni}$.

– $\{\!|\, m \,|\!\}_k^{\{r_1\}} \cong \{\!|\, m \,|\!\}_k^{\{r_1, r_2\}} \cong (\!|\, \{\!|\, m \,|\!\}_k^{r_1} \,|\!)_k^{r_2}$
This represents the fact that the re-encryption operation using the same public key k and a uniform randomness r_2 does not change the probability distribution. Note that we can prove this by employing a renaming σ satisfying $\sigma(\{ r_1, r_2 \}) = \{ r_1 \}$.

- $\{\!|\, m \,|\!\}_k^{r_1} \cong \{\!|\, m \,|\!\}_k^{r_2}$ but $\langle \{\!|\, m \,|\!\}_k^{r_1}, \{\!|\, m \,|\!\}_k^{r_1} \rangle \not\cong \langle \{\!|\, m \,|\!\}_k^{r_1}, \{\!|\, m \,|\!\}_k^{r_2} \rangle$
  This represents the fact that the formal adversary can recognize the repetition of the same ciphertext bit strings. Note that no renaming σ satisfies both $\sigma(\{ r_1 \}) = \{ r_1 \}$ and $\sigma(\{ r_2 \}) = \{ r_1 \}$. In general, our observational equivalence of patterns can deal with the relations of probability distributions unlike [2], because of our definition of the renaming.
- $\{\!|\, m \,|\!\}_k^{r_1} \cong \{\!|\, m \,|\!\}_k^{\{ r_{adv}, r_1 \}} \cong (\!|\, \{\!|\, m \,|\!\}_k^{r_{adv}} \,|\!)_k^{r_1} \quad (r_{adv} \in Rand_{adv})$
  This represents the fact that the re-encryption of the adversary's ciphertext $\{\!|\, m \,|\!\}_k^{r_{adv}}$ using a uniform randomness r_1 produces a uniformly random ciphertext. Note that there exists a renaming σ satisfying $\sigma(\{ r_{adv}, r_1 \}) = \{ r_1 \}$.
- $\langle \{\!|\, m \,|\!\}_k^{r_1}, \{\!|\, m \,|\!\}_k^{r_2} \rangle \not\cong \langle \{\!|\, m \,|\!\}_k^{r_1}, (\!|\, \{\!|\, m \,|\!\}_k^{r_1} \,|\!)_k^{r_{adv}} \rangle \quad (r_{adv} \in Rand_{adv})$
  This represents the fact that the adversary can recognize the re-encryption using the adversary's randomness r_{adv} because he has performed the re-encryption. Note that we have $pattern(\langle \{\!|\, m \,|\!\}_k^{r_1}, \{\!|\, m \,|\!\}_k^{r_2} \rangle, \emptyset) = \langle \Box^{\{\!|\, type(m) \,|\!\}_k^{r_1}}, \Box^{\{\!|\, type(m) \,|\!\}_k^{r_2}} \rangle$ but $pattern(\langle \{\!|\, m \,|\!\}_k^{r_1}, (\!|\, \{\!|\, m \,|\!\}_k^{r_1} \,|\!)_k^{r_{adv}} \rangle, \emptyset) = \langle \Box^{\{\!|\, type(m) \,|\!\}_k^{r_1}}, (\!|\, \Box^{\{\!|\, type(m) \,|\!\}_k^{r_1}} \,|\!)_k^{r_{adv}} \rangle$.
- $\langle \{\!|\, m \,|\!\}_{k_1}^{r_1}, \{\!|\, m \,|\!\}_{k_1}^{r_2} \rangle \not\cong \langle \{\!|\, m \,|\!\}_{k_1}^{r_1}, \{\!|\, m \,|\!\}_{k_2}^{r_2} \rangle$
  This represents the fact that the formal rerandomizable encryption schemes in this paper do not satisfy receiver anonymity [19], i.e., the key-privacy [4] or which-key concealing [2] of rerandomizable encryption schemes.

3 Computational Model

This section introduces the notion of computational indistinguishability, a computational rerandomizable encryption scheme, and its security definitions.

3.1 Preliminaries

In a computational setting, messages are bit strings and adversaries are probabilistic polynomial-time (PPT) algorithms that input and output bit strings. We denote the set of all bit strings by *String*, and the length of a bit string x by $|x|$. The computational security of cryptographic schemes is defined in terms of the notion of a *probability ensemble* over bit strings, which is a sequence $\{D_\eta\}_\eta$ of probability distributions D_η over bit strings indexed by a security parameter η.

We use the following indistinguishability of probability ensembles as a security definition in the computational setting. We write $d \leftarrow D_\eta$ to indicate that d is sampled from a probability distribution D_η, and write $\Pr[d \leftarrow D_\eta : E]$ for the probability of an event E when d is sampled from D_η. We abuse the notation and write $d \leftarrow X$ to indicate that d is sampled from the uniform distribution on a set X. A function f from integers to real numbers is *negligible* in a security parameter η if for every $c > 0$ there exists an integer η_c such that $f(\eta) \leq \eta^{-c}$ holds for any $\eta \geq \eta_c$.

Definition 11. Two probability ensembles $\{D_\eta\}_\eta$ and $\{D'_\eta\}_\eta$ are *computationally indistinguishable* with respect to an oracle O, written $D_\eta \approx_O D'_\eta$ if for every PPT adversary A,

$$\Pr[d \leftarrow D_\eta : A^{O(\cdot)}(d, \eta) = 1] - \Pr[d' \leftarrow D'_\eta : A^{O(\cdot)}(d', \eta) = 1]$$

is negligible in η.

In the above definition, we assume that the PPT adversary A can send a polynomial number of queries to the oracle O.

3.2 Rerandomizable Encryption Scheme

We consider a rerandomizable encryption scheme where everyone can re-encrypt a ciphertext using a public key and a randomness. It is left to our future work to deal with more complex rerandomizable encryption schemes using re-encryption keys generated from secret keys, such as the proxy re-encryption scheme proposed in [5].

Let *Param* be a set of security parameters, *PubKey* be a set of computational public keys, *SecKey* be a set of computational secret keys, *Plaintext* be a set of computational plaintexts, and *Random* be a set of random bit strings used in encryptions and re-encryptions. Let $\perp$ be the special bit string representing the failures of encryptions, decryptions, and re-encryptions. We denote the secret key corresponding to a public key pk by $\overline{pk}$, and the public key corresponding to a secret key sk by $\overline{sk}$.

Definition 12. A computational *rerandomizable encryption scheme* is a quintuple ($\mathcal{G}$, $\mathcal{E}$, $\mathcal{D}$, $\mathcal{R}$, $\mathcal{CMP}$) consisting of the following five algorithms:

- a key generation algorithm $\mathcal{G}\colon \mathit{Param} \times \mathit{Random} \to \mathit{PubKey} \times \mathit{SecKey}$ that outputs, given a security parameter η and a randomness r, a public key and secret key pair $(pk,\ sk)$.
- an encryption algorithm $\mathcal{E}\colon \mathit{PubKey}\times\mathit{String}\times\mathit{Random} \to \mathit{Cipher}\cup\{\perp\}$ that outputs, given a public key pk, a bit string x, and a randomness r, the encryption of x using pk and r, or the failure message $\perp$.
- a decryption algorithm $\mathcal{D}\colon \mathit{SecKey} \times \mathit{String} \to \mathit{Plaintext} \cup \{\perp\}$ that outputs, given a secret key sk and a bit string x, the decryption of x using sk, or the failure message $\perp$.
- a re-encryption algorithm $\mathcal{R}\colon \mathit{PubKey} \times \mathit{String} \times \mathit{Random} \to \mathit{Cipher} \cup \{\perp\}$ that outputs, given a public key pk, a bit string x, and a randomness r, the re-encryption of x using r, or the failure message $\perp$.
- a randomness-composition algorithm $\mathcal{CMP}\colon \mathit{FMulti}(\mathit{Random}) \to \mathit{Random}$ that outputs the composition of a given finite multiset of randomnesses. We assume that the bit string representing the composition of a multiset of randomnesses is uniquely determined if the multiset is fixed.

We assume that the lengths of the outputs from these algorithms depend only on those of the inputs. These algorithms satisfy the following properties for any $pk \in \mathit{PubKey}$, $sk = \overline{pk}$, any $r,\ r' \in \mathit{Random}$, any $R_1,\ R_2,\ R_3 \in \mathit{FMulti}(\mathit{Random})$, and any $x \in \mathit{String}$.

- $\mathcal{D}(sk,\ \mathcal{E}(pk,\ x,\ r)) = \begin{cases} x & (\text{if } x \in \mathit{Plaintext}) \\ \perp & (\text{otherwise}) \end{cases}$
- $\mathcal{R}(pk,\ \mathcal{E}(pk,\ x,\ r),\ r') = \mathcal{E}(pk,\ x,\ \mathcal{CMP}(\{\,r,\ r'\,\}))$
- $\mathcal{CMP}(\mathcal{CMP}(R_1 \uplus R_2) \uplus R_3) = \mathcal{CMP}(R_1 \uplus \mathcal{CMP}(R_2 \uplus R_3))$

To obtain soundness results for the schemes such that the composition of a multiset of randomnesses is not uniquely determined, it is sufficient to use sequences of randomness symbols instead of multisets of randomness symbols.

3.3 Security Definitions of Rerandomizable Encryption Schemes

We define the IND-RCCA security of the rerandomizable encryption scheme.

Definition 13. Let η be a security parameter and $\mathcal{RE} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{CMP})$ be a rerandomizable encryption scheme. For a PPT adversary A, we define the advantage $Adv^{\mathrm{RCCA}}_{\mathcal{RE},A}$ as follows:

$$
\begin{aligned}
Adv^{\mathrm{RCCA}}_{\mathcal{RE},A}(\eta) = \Pr[\ & (pk, sk) \leftarrow \mathcal{G}(\eta); \\
& (m_0, m_1) \leftarrow A^{D_1(\cdot)}(pk); \\
& \quad (m_0 \neq m_1 \text{ and } |m_0| = |m_1|) \\
& r \leftarrow Random; \\
& b \leftarrow \{0, 1\}; \\
& c := \mathcal{E}(pk, m_b, r); \\
& b' \leftarrow A^{D_2(\cdot)}(c) : \\
& b' = b \qquad\qquad\qquad] - \tfrac{1}{2},
\end{aligned}
$$

where

$$
D_1(x) = \mathcal{D}(sk, x) \ \text{ and } \ D_2(x) = \begin{cases} \mathcal{D}(sk, x) & (\mathcal{D}(sk, x) \neq m_0, m_1) \\ test & (\text{otherwise}) \end{cases}
$$

A rerandomizable encryption scheme $\mathcal{RE}$ is *IND-RCCA secure* if the advantage $Adv^{\mathrm{RCCA}}_{\mathcal{RE},A}$ is negligible in η for every PPT adversary A.

The notion "RCCA", or Replayable CCA, was proposed by Canetti et al. [6] as a relaxation of CCA2 security. Although this security is strictly weaker than CCA2, it is believed to be a necessary and sufficient formalization of "secure encryption" from the applicational point of view [3]. Groth [14] first proposed a rerandomizable encryption scheme satisfying a weaker form of RCCA security, and another scheme satisfying RCCA security in the generic groups model. Prabhakaran and Rosulek [19] improved this rerandomizable scheme to achieve RCCA security in a standard model, and Xue and Feng [21] proposed a more efficient scheme that also achieves receiver anonymity. There are notions similar to IND-RCCA: "benign malleability" [20], "loose ciphertext-unforgeability" [16], and "generalized CCA security" [3].

Finally, we define the notion of *randomness-preserving* composition, because IND-RCCA security cannot describe the security property whereby the re-encryption algorithm $\mathcal{R}$ fully rerandomizes input ciphertexts.

Definition 14. Let η be a security parameter and $\mathcal{RE} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{CMP})$ be a rerandomizable encryption scheme. The randomness composition algorithm $\mathcal{CMP}$ is *randomness-preserving* if it holds for every $r, r_0, r_1 \in Random$ that

1. $\Pr[x_0 \leftarrow Random : \mathcal{CMP}(\{x_0, r\}) = r_1] = \Pr[x_0 \leftarrow Random : x_0 = r_1]$
2. $\Pr[x_0 \leftarrow Random : x_0 = r_0 \wedge \mathcal{CMP}(\{x_0, r\}) = r_1]$
   $= \Pr[x_0 \leftarrow Random : x_0 = r_0] \cdot \Pr[x_0 \leftarrow Random : \mathcal{CMP}(\{x_0, r\}) = r_1]$.

By Lemma 1 of [21], if $\mathcal{CMP}$ is randomness-preserving, then $\mathcal{RE}$ is perfectly rerandomizable [19], which is a security notion of the re-encryption operation $\mathcal{R}$.

4 Encoding

This section introduces an encoding that maps patterns to distributions over bit strings. The definition of the encoding is standard [2,12], but we take the composed randomness into account.

First, we define the set of the symbols that should be encoded using random bit strings.

Definition 15. For a term/pattern m, let $RS(m)$ be the set of atomic symbols:

$$RS(m) = \{\, m' \in K_{pub} \cup Nonce \mid m' \text{ occurs in } m \,\} \cup \{\, \overline{k_{sec}} \mid k_{sec} \in K_{sec}, k_{sec} \text{ occurs in } m \,\}$$
$$\cup \{\, r \in R \mid R \in FMulti(Rand),\ R \text{ occurs in } m \,\}.$$

For a set S of terms/patterns, let $RS(S)$ be the set $\bigcup_{m \in S} RS(m)$.

Next, we define the set $Coins_\ell$ of functions each of which encodes the randomness used to encode key/nonce/random symbols.

Definition 16. For a set X of atomic symbols, let $Coins_\ell(X)$ be the set: $\{\, t\colon X \to \{0,1\}^\ell \,\}$.

Each function $t \in Coins_\ell(RS(m))$ maps each key/nonce/randomness symbol x in m to a random bit string used to encode x. For example, for a public key symbol k_{pub} occurring in m, $t(k_{pub})$ is the random bit string that is used to generate the public key bit string denoted by k_{pub}. Hereafter we sometimes omit the length ℓ from the notation when ℓ is a polynomial in the security parameter η such that $t \in Coins_\ell(X)$ is sufficient to encode all the key/nonce/randomness symbols in X.

Then, we define the algorithms used in the encoding of terms/patterns. Let $\mathcal{RE} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{CMP})$ be a rerandomizable encryption scheme. We use $\mathcal{G}$ to encode public and secret key symbols, $\mathcal{E}$ to encode encryptions, $\mathcal{R}$ to encode re-encryptions, and $\mathcal{CMP}$ to encode a set of randomness symbols. We also use the following algorithms.

Definition 17.
- A *constant encoder* $\mathcal{C}$ is a deterministic algorithm that outputs a fixed bit string corresponding to a given constant c in *Const*.
- A *nonce encoder* $\mathcal{N}$ is an algorithm that outputs, given a randomness $t(n)$ for some $n \in Nonce$, a bit string uniformly and randomly selected from $\{0,1\}^{poly(\eta)}$, where $poly(\eta)$ is a fixed polynomial in η.
- A *type encoder* $\mathcal{T}$ is an algorithm that outputs a fixed bit string of the same length as the encoding of the term m for an input $type(m)$, such as an all-zero string of the same length.
- A *nonce distribution* D_{nonce} is an algorithm that outputs, given a random bit string, a bit string used as the adversary's nonce.
- A *randomness distribution* D_{rand} is an algorithm that outputs, given a random bit string, a bit string used as the adversary's randomness for probabilistic encryptions and re-encryptions.

We assume that each of these algorithms outputs bit strings of the same length for inputs of the same length. Let $\mathcal{I} = \langle \mathcal{RE}, \mathcal{C}, \mathcal{N}, \mathcal{T}, D_{nonce}, D_{rand} \rangle$.

Finally, we define the encoding of terms/patterns. We abuse the notations and use $\langle \cdot, \cdot \rangle$ to represent the concatenation of bit strings. Let *fst* and *snd* be the two algorithms that map a concatenation of bit strings to the first and second component, respectively.

Definition 18. Let e be a function from some set $dom(e)$ of terms/patterns to bit strings, η be a security parameter, and $t \in Coins(RS(m) \setminus dom(e))$. The *encoding* $\llbracket m \rrbracket^{e,t}_{\eta,\mathcal{I}}$ of a term/pattern m is recursively defined as follows:

$$
\begin{aligned}
&\text{if } m \in Dom(e),\\
&\text{then} \quad \llbracket m \rrbracket^{e,t}_{\eta,\mathcal{I}} = e(m)\\
&\text{else} \quad \llbracket c \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle C(c), \text{“Const”}\rangle\\
&\llbracket k_{pub} \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle fst(\mathcal{G}(\eta,\ t(k_{pub}))), \text{“PubKey”}\rangle\\
&\llbracket k_{sec} \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle snd(\mathcal{G}(\eta,\ t(\overline{k_{sec}}))), \text{“SecKey”}\rangle\\
&\llbracket n \rrbracket^{e,t}_{\eta,\mathcal{I}} = \begin{cases} \langle D_{nonce}(\mathcal{N}(\eta,\ t(n))), \text{“Nonce”}\rangle & (\text{if } n \in Nonce_{adv})\\ \langle \mathcal{N}(\eta,\ t(n)), \text{“Nonce”}\rangle & (\text{otherwise}) \end{cases}\\
&\llbracket \{ r \} \rrbracket^{e,t}_{\eta,\mathcal{I}} = \begin{cases} \langle D_{rand}(t(r)), \text{“Rand”}\rangle & (\text{if } r \in Rand_{adv})\\ \langle t(r), \text{“Rand”}\rangle & (\text{otherwise}) \end{cases}\\
&\llbracket R \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle \mathcal{CMP}(\{ fst(\llbracket \{ r \} \rrbracket^{e,t}_{\eta,\mathcal{I}}) \mid r \in R \}), \text{“Rand”}\rangle\\
&\llbracket \langle m_1,\ m_2\rangle \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle\langle \llbracket m_1 \rrbracket^{e,t}_{\eta,\mathcal{I}},\ \llbracket m_2 \rrbracket^{e,t}_{\eta,\mathcal{I}}\rangle, \text{“pair”}\rangle\\
&\llbracket \{\!| m |\!\}^R_k \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle \mathcal{E}(fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}),\ \llbracket m \rrbracket^{e,t}_{\eta,\mathcal{I}},\ fst(\llbracket R \rrbracket^{e,t}_{\eta,\mathcal{I}})),\ fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}), \text{“enc”}\rangle\\
&\llbracket (\!| m |\!)^R_k \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle \mathcal{R}(fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}),\ fst(\llbracket m \rrbracket^{e,t}_{\eta,\mathcal{I}}),\ fst(\llbracket R \rrbracket^{e,t}_{\eta,\mathcal{I}})),\ fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}), \text{“enc”}\rangle\\
&\llbracket \Box^{\{\!| type(m) |\!\}^R_k} \rrbracket^{e,t}_{\eta,\mathcal{I}} = \langle \mathcal{E}(fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}),\ \mathcal{T}(type(m)),\ fst(\llbracket R \rrbracket^{e,t}_{\eta,\mathcal{I}})),\ fst(\llbracket k \rrbracket^{e,t}_{\eta,\mathcal{I}}), \text{“enc”}\rangle
\end{aligned}
$$

where $c \in Const$, $k_{pub} \in K_{pub}$, $k_{sec} \in K_{sec}$, $n \in Nonce$, and $r \in Rand$, $R \in FMulti(Rand)$. For any pattern m and any security parameter η, the encoding $\llbracket m \rrbracket^{e}_{\eta,\mathcal{I}}$ is the probability distribution $\{ t \leftarrow Coins(RS(m) \setminus dom(e)) \colon \llbracket m \rrbracket^{e,t}_{\eta,\mathcal{I}} \}$. We omit e when $dom(e) = \emptyset$. When $Dom(e) = \{ x_1, x_2, \cdots, x_n \}$ and $y_i = e(x_i)$ for each $1 \le i \le n$, we sometimes write $[x_1 \mapsto y_1, x_2 \mapsto y_2, \cdots, x_n \mapsto y_n]$ instead of e. Hereafter we omit $\mathcal{I}$ from the notations, and abbreviate $\llbracket \{ r \} \rrbracket^{e,t}_{\eta,\mathcal{I}}$ as $\llbracket r \rrbracket^{e,t}_{\eta}$.

In the above definition, each encoding is followed by a type tag representing one of the bit string types “Const”, “PubKey”, “SecKey”, “Nonce”, or “Random” and the bit string operation types “pair” and “enc”. The algorithm *fst* is used to remove type tags, and we omit *fst* for readability hereafter. A ciphertext bit string contains the public key used to generate the ciphertext. We introduce the algorithm $\mathcal{PK}$ that outputs the public key pk from a given encryption using pk. $\mathcal{PK}$ satisfies the equation: $\mathcal{PK}(\langle \mathcal{E}(\llbracket k \rrbracket^{e,t}_{\eta}, \llbracket m \rrbracket^{e,t}_{\eta}, \llbracket R \rrbracket^{e,t}_{\eta}), \llbracket k \rrbracket^{e,t}_{\eta}, \text{“enc”}\rangle) = \llbracket k \rrbracket^{e,t}_{\eta}$.

Note that $\llbracket m \rrbracket^{e,t}_{\eta}$ is a unique bit string, because $t \in Coins(RS(m) \setminus dom(e))$ determines all the randomnesses in m.

5 Soundness

5.1 Abadi-Rogaway Indistinguishability

First, we define a function $undec_{\tau}$ that maps a bit string to a set of undecryptable bit strings. Intuitively, given an encoding μ of a term M and a set τ of encodings of a set

$T \subseteq K_{sec}$, $x \in undec_\tau(\mu)$ is an encoding of an undecryptable message in $pattern(M, T)$.

Definition 19. Let μ be a bit string, and τ be a set of computational secret keys. Let $undec_\tau$ be the algorithm defined in Fig. 1.

```
algorithm $undec_\tau(\mu)$
  Set $B, B' := \{\mu\}$;
  do
    $B := B'$;
    $B' := \emptyset$;
    for each $b \in B$
      if $b = \langle b_1, b_2, \text{“pair”}\rangle$
        then $B' := B' \cup \{b_1, b_2\}$;
      if $b = \langle c, \mathcal{PK}(c), \text{“enc”}\rangle$ and $\langle \mathcal{PK}(c), \text{“PubKey”}\rangle \in \overline{\tau}$
        then $B' := B' \cup \{\mathcal{D}(\overline{\mathcal{PK}(c)}, c)\}$;
      if $b = \langle c, \mathcal{PK}(c), \text{“enc”}\rangle$ and $\langle \overline{\mathcal{PK}(c)}, \text{“SecKey”}\rangle \in \tau$
        then $B' := B' \cup \{\mathcal{D}(\overline{\mathcal{PK}(c)}, c)\}$;
      otherwise
        $B' := B' \cup \{b\}$;
  while $B' \neq B$;
  return $B'$;
```

Fig. 1. Algorithm $undec_\tau$.

Roughly speaking, $undec_\tau(\mu)$ is the set of all the challenge ciphertexts, and is used to specify the ciphertexts that cannot be decrypted by the decryption oracle in Definition 21.

Next, we define the set $forbid_{\eta,t}(M, T)$ of bit strings that is used in the oracle of Definition 21.

Definition 20. Let $M \in Term$ and $T \subseteq K_{sec}$. Let $forbid_{\eta,t}(M, T)$ be the set:

$$\left\{ \langle pk, \mathcal{D}(\overline{pk}, y)\rangle, \langle pk, Type(\mathcal{D}(\overline{pk}, y))\rangle \;\middle|\; \begin{array}{l} y \in undec_{[\![T]\!]^t_\eta}([\![M]\!]^t_\eta), \\ pk = \mathcal{PK}(y) \end{array} \right\},$$

where $Type$ is the algorithm defined by $Type([\![m]\!]^t_\eta) = \mathcal{T}(type(m))$ for every $m \in Term$.

Finally, we define a computational indistinguishability between the two probability distributions each encoding a term. This indistinguishability is defined in the presence of an active and adaptive PPT adversary A, and is almost the same as that in [15] except for the definition of the oracle $O^{M,M',T}_{\eta,t}$.

Definition 21. Let η be any security parameter, T be any finite set of secret key symbols, and M and M' be any two acyclic terms satisfying the freshness assumption and $M \cong M'$. A rerandomizable encryption scheme $\mathcal{RE}$ provides *Abadi-Rogaway RCCA indistinguishability* if for every PPT adversary A, it holds that

$$[\![M]\!]_\eta \approx_{O^{M,M',T}_{\eta,t}} [\![M']\!]_\eta,$$

that is, the advantage $Adv^{\text{AR-RCCA}}_{\mathcal{RE},A}$ defined below is negligible in η.

$$\begin{aligned} Adv^{\text{AR-RCCA}}_{\mathcal{RE},A}(\eta) = \Pr[t \leftarrow Coins(M),\ d \leftarrow [\![M]\!]^t_\eta :\ & A^{O^{M,M',T}_{\eta,t}(\cdot,\cdot)}(d,\eta) = 1] \\ - \Pr[t \leftarrow Coins(M'),\ d \leftarrow [\![M']\!]^t_\eta :\ & A^{O^{M,M',T}_{\eta,t}(\cdot,\cdot)}(d,\eta) = 1] \end{aligned}$$

$$O^{M,M',T}_{\eta,t}(pk,x) = \begin{cases} \mathcal{D}(\overline{pk},x) & \text{(if either} \\ & \quad \text{(i) } pk \in [\![K]\!]^t_\eta \text{ for some } K \in \overline{T}, \text{ or} \\ & \quad \text{(ii) (a) } pk \in [\![K]\!]^t_\eta \text{ for some } K \in K_{pub} \setminus \overline{T}, \\ & \quad \quad \text{(b) } \langle pk, \mathcal{D}(\overline{pk},x)\rangle \notin forbid_{\eta,t}(M,T), \\ & \quad \quad \text{and} \\ & \quad \quad \text{(c) } \langle pk, \mathcal{D}(\overline{pk},x)\rangle \notin forbid_{\eta,t}(M',T)) \\ \perp & \text{(if } pk \notin [\![K]\!]^t_\eta \text{ for any } K \in K_{pub}) \\ test & \text{(otherwise)} \end{cases}$$

In this definition, the adversary A can learn some relations between plaintexts and their encryptions by having access to the oracle $O^{M,M',T}_{\eta,t}$. As opposed to the access to D_1 and D_2 in Definition 13, the adversary A needs to send a public key pk to the oracle $O^{M,M',T}_{\eta,t}$ to specify the corresponding secret key $\overline{pk}$ used for the decryption, because two messages M, M', and their patterns can be thought of as many possible different challenge ciphertexts under many possible different keys.

The oracle $O^{M,M',T}_{\eta,t}$ is similar to that in [15] except that the two sets $forbid_{\eta,t}(M,T)$ and $forbid_{\eta,t}(M',T)$ are used to determine whether or not $O^{M,M',T}_{\eta,t}$ returns the decryption of x to the adversary A. The challenge ciphertexts that the oracle $O^{M,M',T}_{\eta,t}$ should not decrypt are those encryptions that the decryption oracle D_2 is not allowed to decrypt in the IND-RCCA game. They are either undecryptable ciphertexts $\mathcal{E}(pk,m,r)$ derivable from $[\![M]\!]^t_\eta$ or $[\![M']\!]^t_\eta$, or the corresponding encryptions $\mathcal{E}(pk, Type(m), r)$. Therefore, $forbid_{\eta,t}(M,T) \cup forbid_{\eta,t}(M',T)$ specifies the set of all the challenge ciphertexts that $O^{M,M',T}_{\eta,t}$ should not decrypt.

5.2 Soundness of Formal Rerandomizable Encryption

We obtain the following soundness theorem.

Theorem 1. Let $\mathcal{RE}$ be an IND-RCCA secure rerandomizable encryption scheme with a randomness-preserving composition $\mathcal{CMP}$. For any two acyclic terms M and M' satisfying the freshness assumption, $M \cong M'$ implies $[\![M]\!]_\eta \approx_{O^{M,M',K_{adv}}_{\eta,t}} [\![M']\!]_\eta$.

Proof. By Lemmas 1 and 2 presented below, we have the following equation for some renaming σ for the atomic symbols in $pattern(M', K_{adv})$ except for K_{adv}.

$$\begin{aligned} [\![M]\!]_\eta \approx_{O^{M,M',K_{adv}}_{\eta,t}} [\![pattern(M,K_{adv})]\!]_\eta &= [\![\tilde{\sigma}\, pattern(M',K_{adv})]\!]_\eta \\ &= [\![pattern(M',K_{adv})]\!]_\eta \approx_{O^{M,M',K_{adv}}_{\eta,t}} [\![M']\!]_\eta . \end{aligned}$$

□

Lemma 1. Let M and M' be any two acyclic terms satisfying the freshness assumption, and T be any finite set of secret key symbols. Let $\mathcal{RE} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{CMP})$ be an IND-RCCA secure rerandomizable encryption scheme where $\mathcal{CMP}$ is randomness-preserving. Then we have $[\![M]\!]_\eta \approx_{O^{M,M',T}_{\eta,t}} [\![pattern(M,T)]\!]_\eta$.

Proof. Suppose that there exists a PPT adversary A with access to $O_{\eta,t}^{M,M',T}$ who can distinguish between samples from $\llbracket M \rrbracket_\eta$ and $\llbracket \mathit{pattern}(M, T) \rrbracket_\eta$. Then we derive a contradiction by using a hybrid argument similar to [2,15]. Between the two rows M and $\mathit{pattern}(M, T)$, we create a new row for each encryption/re-encryption, so that two consecutive rows differ only in one of the following cases:

(1) a single re-encryption $(\!|\, (\!|\, P \,|\!)_K^{R'} \,|\!)_K^{R}$ being replaced with $(\!|\, P \,|\!)_K^{R \uplus R'}$ for $P \in \mathit{Reenc}_K$, $R \in \mathit{FMulti}(\mathit{Rand}) \setminus \mathit{FMulti}(\mathit{Rand}_{adv})$, and $R' \in \mathit{FMulti}(\mathit{Rand})$,

(2) a single re-encryption $(\!|\, \{\!|\, P \,|\!\}_K^{R'} \,|\!)_K^{R}$ being replaced with $\{\!|\, P \,|\!\}_K^{R \uplus R'}$ for $R \in \mathit{FMulti}(\mathit{Rand}) \setminus \mathit{FMulti}(\mathit{Rand}_{adv})$ and $R' \in \mathit{FMulti}(\mathit{Rand})$,

(3) a single encryption $\{\!|\, P \,|\!\}_K^{R}$ being replaced with $\square^{\{\!|\, \mathit{type}(P) \,|\!\}_K^R}$ for $R \in \mathit{FMulti}(\mathit{Rand}) \setminus \mathit{FMulti}(\mathit{Rand}_{adv})$ and $K \in K_{pub} \setminus \overline{T}$.

Because of the definition of patterns, we obtain a sequence of rows: $M = M_0, M_1, \cdots, M_i, M_{i+1}, \cdots, M_n = \mathit{pattern}(M, T)$ where for each $0 \le i < n$, M_i and M_{i+1} are identical except for one of the above cases. Unlike [2,15], it is necessary to consider cases **(1)** and **(2)** that deal with re-encryption patterns. Furthermore, in case **(3)** we take account of the condition with the randomnesses of probabilistic encryptions.

Example 2. For example, let M be the following sequence of terms, and T be the following set for $c \in \mathit{Const}$, $k_1, k_2, k_3, k_4 \in K_{pub}$, and $R_1, R_2, R_3, R_4, R_5, R_6 \in \mathit{Rand}_{uni}$.

$$M = \{\!| c |\!\}_{k_2}^{R_2},\ \{\!| \{\!| c |\!\}_{k_2}^{R_2},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ (\!| (\!| \{\!| c |\!\}_{k_4}^{R_5} |\!)_{k_4}^{R_4} |\!)_{k_4}^{R_3} \qquad T = \{ \overline{k_1} \}$$

Here we have omitted parentheses for readability. We obtain the secret key symbols:

$$\mathit{recoverable}(M, T) = \{ \overline{k_1}, \overline{k_3} \}.$$

We obtain the sequence of rows $M = M_0, M_1, M_2, M_3, M_4, M_5 = \mathit{pattern}(M, T)$.

$$\begin{array}{ll}
M_0 = \{\!| c |\!\}_{k_2}^{R_2},\ \{\!| \{\!| c |\!\}_{k_2}^{R_2},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ (\!| (\!| \{\!| c |\!\}_{k_4}^{R_5} |\!)_{k_4}^{R_4} |\!)_{k_4}^{R_3} & \\
 & (3)\ k_2 \\
M_1 = \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| \{\!| c |\!\}_{k_2}^{R_2},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ (\!| (\!| \{\!| c |\!\}_{k_4}^{R_5} |\!)_{k_4}^{R_4} |\!)_{k_4}^{R_3} & \\
 & (3)\ k_2 \\
M_2 = \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ (\!| (\!| \{\!| c |\!\}_{k_4}^{R_5} |\!)_{k_4}^{R_4} |\!)_{k_4}^{R_3} & \\
 & (1)\ k_4 \\
M_3 = \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ (\!| \{\!| c |\!\}_{k_4}^{R_5} |\!)_{k_4}^{R_3 \uplus R_4} & \\
 & (2)\ k_4 \\
M_4 = \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ \{\!| c |\!\}_{k_4}^{R_3 \uplus R_4 \uplus R_5} & \\
 & (3)\ k_4 \\
M_5 = \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| \square^{\{\!| \mathit{Const} |\!\}_{k_2}^{R_2}},\ \{\!| c |\!\}_{k_3}^{R_6},\ \overline{k_3} |\!\}_{k_1}^{R_1},\ \square^{\{\!| \mathit{Const} |\!\}_{k_4}^{R_3 \uplus R_4 \uplus R_5}} &
\end{array}$$

Since A can distinguish between $\llbracket M_0 \rrbracket_\eta$ and $\llbracket M_n \rrbracket_\eta$, there exist two consecutive rows M_i and M_{i+1} such that A can distinguish between $\llbracket M_i \rrbracket_\eta$ and $\llbracket M_{i+1} \rrbracket_\eta$. Fix M_i and M_{i+1}. Then the two rows M_i and M_{i+1} are the same except for one of the above three cases **(1)** - **(3)**. In each case, we derive a contradiction.

(1) Consider the first case: M_i and M_{i+1} are the same except that a re-encryption $\llparenthesis \llparenthesis P \rrparenthesis_K^{R'} \rrparenthesis_K^{R}$ in M_i is replaced with $\llparenthesis P \rrparenthesis_K^{R \uplus R'}$ in M_{i+1} for $P \in Reenc_K$. Since $P \in Reenc_K$ holds, we obtain the following equation for every $t \in Coins(RS(M_i))$:

$$\mathcal{R}(\llbracket K \rrbracket_\eta^t, \mathcal{R}(\llbracket K \rrbracket_\eta^t, \llbracket P \rrbracket_\eta^t, \llbracket R' \rrbracket_\eta^t), \llbracket R \rrbracket_\eta^t) = \mathcal{R}(\llbracket K \rrbracket_\eta^t, \llbracket P \rrbracket_\eta^t, \mathcal{CMP}(\llbracket R \rrbracket_\eta^t \uplus \llbracket R' \rrbracket_\eta^t)$$

Therefore, we have $\llbracket M_i \rrbracket_\eta = \llbracket M_{i+1} \rrbracket_\eta$, which contradicts the assumption that A can distinguish $\llbracket M_i \rrbracket_\eta$ and $\llbracket M_{i+1} \rrbracket_\eta$.

(2) Consider the second case: M_i and M_{i+1} are the same except that a re-encryption $\llparenthesis \{\!| P |\!\}_K^{R'} \rrparenthesis_K^{R}$ in M_i is replaced with $\{\!| P |\!\}_K^{R \uplus R'}$ in M_{i+1}. We have the following equation for every $t \in Coins(RS(M_i))$:

$$\mathcal{R}(\llbracket K \rrbracket_\eta^t, \mathcal{E}(\llbracket K \rrbracket_\eta^t, \llbracket P \rrbracket_\eta^t, \llbracket R' \rrbracket_\eta^t), \llbracket R \rrbracket_\eta^t) = \mathcal{E}(\llbracket K \rrbracket_\eta^t, \llbracket P \rrbracket_\eta^t, \mathcal{CMP}(\llbracket R \rrbracket_\eta^t \uplus \llbracket R' \rrbracket_\eta^t)$$

Therefore, we obtain $\llbracket M_i \rrbracket_\eta = \llbracket M_{i+1} \rrbracket_\eta$, which contradicts the assumption that A can distinguish $\llbracket M_i \rrbracket_\eta$ and $\llbracket M_{i+1} \rrbracket_\eta$.

(3) Consider the third case: M_i and M_{i+1} are the same except that an encrypted message $\{\!| P |\!\}_K^R$ in M_i is replaced with $\Box^{\{\!| type(P) |\!\}_K^R}$ in M_{i+1} for $R \in FMulti(Rand) \setminus FMulti(Rand_{adv})$ and $K \in K_{pub} \setminus \overline{T}$.

Now we construct an adversary A_0 that breaks the IND-RCCA security of the rerandomizable encryption scheme $\mathcal{RE}$. The definition of A_0 is presented in Figs. 2 and 3.

Let (pk, sk) be a pair consisting of a public key and a secret key generated using the key generation algorithm $\mathcal{G}$. Because of the freshness assumption in Definition 6, we can take a randomness symbol $r_0 \in R \cap Rand_{uni}$ such that $r_0 \notin R'$ holds for every $R' \in FMulti(Rand)$ occurring in M_i with $R' \neq R$. Note that r_0 does not occur in P. Assume that $x_0 \leftarrow Random$. We treat pk and x_0 as the encoding of the public key symbol K and the randomness symbol r_0, respectively.

```
$A_0^{D_1(\cdot)}(pk)$
    $t \leftarrow Coins(RS(M_i) \setminus \{ K, r_0 \})$;
    $m_0 := \llbracket P \rrbracket_\eta^{[K \mapsto \langle pk, \text{“PubKey''} \rangle], t}$;
    $m_1 := \mathcal{T}(type(P))$;
    **return** $(m_0, m_1)$;
```

Fig. 2. The behavior of A_0 on input pk.

```
$A_0^{D_2(\cdot)}(c)$
    $s := \llbracket M_i \rrbracket_\eta^{e,t}$;
    $b' \leftarrow A^{O_{\eta,t}^{\widehat{M_i, M_{i+1}}, T}(pk, \cdot)}(s, \eta)$;
    **return** $b'$;
```

Fig. 3. The behavior of A_0 on input c.

In Fig. 2, A_0 receives the public key pk and generates two bit strings m_0 and m_1 of the same length. D_1 is the decryption oracle defined in Definition 13.

Then assume that $b \leftarrow \{0, 1\}$, $x := \mathcal{CMP}(\{ x_0 \} \uplus \{ \llbracket r' \rrbracket_\eta^t \mid r' \in R \setminus \{ r_0 \} \})$, and $c := \mathcal{E}(pk, m_b, x)$. Since $\mathcal{CMP}$ is randomness-preserving and x_0 is selected independently and uniformly, x is also independent and uniform. Therefore, we can use x as the randomness of the probabilistic encryption generating the challenge ciphertext c in the IND-RCCA game.

In Fig. 3, A_0 receives the ciphertext c and guesses b by invoking the adversary A as a subroutine. Let e be the function $[\{\!| P |\!\}_K^R \mapsto \langle c, pk, \text{“enc”} \rangle, K \mapsto \langle pk, \text{“PubKey”} \rangle]$, and s be the bit string $\llbracket M_i \rrbracket_\eta^{e,t}$. The adversary A receives s from A_0, and answers which of the two distributions $\llbracket M_i \rrbracket_\eta$ and $\llbracket M_{i+1} \rrbracket_\eta$ s is sampled from. Note that we have the equations:

$$\left\{\begin{array}{l} t \leftarrow Coins(RS(M_i) \setminus \{K, r_0\}),\ b := 0, \\ pk \leftarrow fst(\mathcal{G}(\eta)),\ x_0 \leftarrow Random \end{array}\right. : [\![M_i]\!]_\eta^{e,t} \Big\} = [\![M_i]\!]_\eta$$

$$\left\{\begin{array}{l} t \leftarrow Coins(RS(M_i) \setminus \{K, r_0\}),\ b := 1, \\ pk \leftarrow fst(\mathcal{G}(\eta)),\ x_0 \leftarrow Random \end{array}\right. : [\![M_i]\!]_\eta^{e,t} \Big\} = [\![M_{i+1}]\!]_\eta$$

Here, e depends on the bit b, which was used to produce the challenge ciphertext $c := \mathcal{E}(pk, m_b, x)$. Since A can distinguish between $[\![M_i]\!]_\eta$ and $[\![M_{i+1}]\!]_\eta$ with non-negligible probability, A_0 can guess the bit b with non-negligible probability by receiving b' from A. Hence, A_0 breaks the IND-RCCA security. This contradicts the assumption.

There remains a problem with the oracle $O_{\eta,t}^{M,M',T}$. Recall that A uses the oracle $O_{\eta,t}^{M,M',T}$ defined in Definition 21. Since the definition of IND-RCCA security allows A_0 to use only the decryption oracles D_1 and D_2, we consider an algorithm $O_{\eta,t}^{\widehat{M,M'},T}$ that uses only D_2 and simulates the oracle $O_{\eta,t}^{M,M',T}$. We assume that the adversary A uses the algorithm $O_{\eta,t}^{\widehat{M,M'},T}$ presented in Fig. 4, instead of the oracle $O_{\eta,t}^{M,M',T}$. Note that A can efficiently decide $\langle pk, D_2(x)\rangle \in forbid_{\eta,t}(M,T) \cup forbid_{\eta,t}(M',T)$ by computing $forbidden1^{D_2(\cdot)}([\![M]\!]_\eta^t, [\![M']\!]_\eta^t, D_2(x), [\![T]\!]_\eta^t)$ in Fig. 5, and $\langle pk', \mathcal{D}(sk', x)\rangle \in forbid_{\eta,t}(M,T) \cup forbid_{\eta,t}(M',T)$ by computing $forbidden2([\![M]\!]_\eta^t, [\![M']\!]_\eta^t, \mathcal{D}(sk', x), [\![T]\!]_\eta^t, sk')$ in Fig. 6. □

```
algorithm $O_{\eta,t}^{\widehat{M,M'},T^{D_2(\cdot)}}(pk, x)$
    if $pk \neq [\![ k_{pub0} ]\!]_\eta^t$
       for any $k_{pub0} \in K_{pub}$
    then  return $\perp$;
    else if $k_{pub0} = K$
        then if $\langle pk, D_2(x)\rangle \in forbid_{\eta,t}(M, T)$
                or $\langle pk, D_2(x)\rangle \in forbid_{\eta,t}(M', T)$
             then  return $test$;
             else  return $D_2(x)$;
        else $(pk', sk') := \mathcal{G}(\eta, t(k_{pub0}))$;
             if $k_{pub0} \in \overline{T}$
             then  return $\mathcal{D}(sk', x)$;
             else if $\langle pk', \mathcal{D}(sk', x)\rangle \in forbid_{\eta,t}(M, T)$
                     or $\langle pk', \mathcal{D}(sk', x)\rangle \in forbid_{\eta,t}(M', T)$
                 then  return $test$;
                 else  return $\mathcal{D}(sk', x)$;
```

Fig. 4. Algorithm $O_{\eta,t}^{\widehat{M,M'},T^{D_2(\cdot)}}$.

```
algorithm $forbidden1^{D_2(\cdot)}(s_1, s_2, \mu, \tau)$
    Set $F := \emptyset$
    for each $y \in undec_\tau(s_1) \cup undec_\tau(s_2)$
        $F := F \cup \{D_2(y)\}$;
    if $\mu \in F$
    then  return "yes";
    else  return "no";
```

Fig. 5. Algorithm $forbidden1^{D_2(\cdot)}$.

```
algorithm $forbidden2(s_1, s_2, \mu, \tau, sk')$
    Set $F := \emptyset$
    for each $y \in undec_\tau(s_1) \cup undec_\tau(s_2)$
        $F := F \cup \{\mathcal{D}(sk', y)\}$;
    if $\mu \in F$
    then  return "yes";
    else  return "no";
```

Fig. 6. Algorithm $forbidden2$.

Lemma 2. Let $\mathcal{RE} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{CMP})$ be an IND-RCCA secure rerandomizable encryption scheme where $\mathcal{CMP}$ is randomness-preserving. Let M be any acyclic term satisfying the freshness assumption, and T be any set of secret key symbols. Let σ be a renaming for the atomic symbols in $pattern(M,T)$ except for T such that $\tilde{\sigma}\, pattern(M,T)$

$= \mathit{pattern}(M', T)$ for some acyclic term M' satisfying the freshness assumption. Then we have $[\![\, \tilde{\sigma}\, \mathit{pattern}(M, T) \,]\!]_\eta = [\![\, \mathit{pattern}(M, T) \,]\!]_\eta$.

Proof. Given a term/pattern Q, let $X(Q) = \{ R \in \mathit{FMulti}(\mathit{Rand}) \setminus \mathit{FMulti}(\mathit{Rand}_{adv}) \mid R \text{ occurs in } Q \}$. Let $P = \mathit{pattern}(M, T)$. Let $\sigma|_{X(P)}$ be the renaming for the atomic symbols in P such that $\sigma|_{X(P)}(R) = \sigma(R)$ if $R \in X(P)$ and $\sigma|_{X(P)}(Q) = Q$ otherwise. Let $\sigma|_{\mathit{Atom}_T(P)\setminus X(P)}$ be the renaming for the atomic symbols in P such that $\sigma|_{\mathit{Atom}_T(P)\setminus X(P)}(R) = R$ if $R \in X(P)$ and $\sigma|_{\mathit{Atom}_T(P)\setminus X(P)}(Q) = \sigma(Q)$ otherwise. Clearly, we have $[\![\, \tilde{\sigma}\, P \,]\!]_\eta = [\![\, \tilde{\sigma}|_{\mathit{Atom}_T(P)\setminus X(P)}\, \tilde{\sigma}|_{X(P)}\, P \,]\!]_\eta$ and $[\![\, \tilde{\sigma}|_{\mathit{Atom}_T(P)\setminus X(P)}\, \tilde{\sigma}|_{X(P)}\, P \,]\!]_\eta = [\![\, \tilde{\sigma}|_{X(P)}\, P \,]\!]_\eta$. Hence, it is sufficient to prove $[\![\, \tilde{\sigma}|_{X(P)}\, P \,]\!]_\eta = [\![\, P \,]\!]_\eta$.

Let $t \in \mathit{Coins}(\mathit{RS}(\{ M \} \cup T))$. Since M satisfies the freshness assumption, for each $\tilde{R} \in X(M)$, there exists a uniform randomness symbol $\tilde{r} \in \tilde{R} \cap \mathit{Rand}_{uni}$ that is independent in M. Therefore, $[\![\, \tilde{r} \,]\!]^t_\eta$ is a random bit string independently and uniformly selected from Random. Since $\mathcal{CMP}$ is randomness-preserving, for each $\tilde{R} \in X(M)$, the randomness $[\![\, \tilde{R} \,]\!]^t_\eta$ composed of $[\![\, \tilde{r} \,]\!]^t_\eta$ is also independent and uniform.

Let $R_1, R_2, \cdots, R_n$ be all the distinct finite multisets of randomness symbols in $X(P)$. It is immediate from Definition 3 that for every $1 \leq i \leq n$, there exist some $\tilde{R}_{i_1}$, $\tilde{R}_{i_2}, \cdots, \tilde{R}_{i_k} \in \mathit{FMulti}(\mathit{Rand})$ occurring in M for $k \geq 1$ such that $R_i = \tilde{R}_{i_1} \uplus \tilde{R}_{i_2} \uplus \cdots \uplus \tilde{R}_{i_k}$, a term of the form $(\!|\, \cdots (\!|\, \{\!|\, m \,|\!\}^{\tilde{R}_{i_1}}_k \,|\!)^{\tilde{R}_{i_2}}_k \cdots \,|\!)^{\tilde{R}_{i_k}}_k$ occurs in M, and $\tilde{R}_{i_k} \in X(P)$. Since $\mathcal{CMP}$ is randomness-preserving and $[\![\, \tilde{R}_{i_k} \,]\!]^t_\eta$ is independent and uniform, $[\![\, \tilde{R}_i \,]\!]^t_\eta = \mathcal{CMP}([\![\, \tilde{R}_{i_1} \uplus \tilde{R}_{i_2} \uplus \cdots \uplus \tilde{R}_{i_{k-1}} \,]\!]^t_\eta, [\![\, \tilde{R}_{i_k} \,]\!]^t_\eta)$ is also independent and uniform.

On the other hand, since $\sigma|_{X(P)}$ is injective, $\sigma|_{X(P)}(R_1), \sigma|_{X(P)}(R_2), \cdots, \sigma|_{X(P)}(R_n)$ are all the distinct multisets of randomness symbols in $X(\tilde{\sigma}|_{X(P)}\, P)$. Then, $[\![\, \sigma|_{X(P)}(R_i) \,]\!]^t_\eta$ is independent and uniform, because $\tilde{\sigma}|_{X(P)}\, P$ is also the pattern of some term satisfying the freshness assumption.

Since both $[\![\, \sigma|_{X(P)}(R_i) \,]\!]^t_\eta$ and $[\![\, R_i \,]\!]^t_\eta$ are independent bit strings uniformly distributed on Random for any $1 \leq i \leq n$, we obtain $[\![\, \tilde{\sigma}|_{X(P)}\, P \,]\!]_\eta = [\![\, P \,]\!]_\eta$. □

5.3 Example: Analysis of Simple Re-encryption Mixnet

We present an example of an analysis of a security protocol in our model.

Example 3. Consider a simple re-encryption mixnet protocol in which there are two honest senders X_1 and X_2, an honest mixnet server Y, and a formal adversary A. We assume that they all have a public key $k_{pub} \in K_{pub}$, and that only Y has the corresponding secret key $\overline{k_{pub}}$.

First, each X_i encrypts a message $c_i \in \mathit{Const}$ using k_{pub} and a uniformly selected randomness $r_i \in \mathit{Rand}_{uni}$. Next, each X_i sends the ciphertext $\{\!|\, c_i \,|\!\}^{r_i}_{k_{pub}}$ to the server Y. Then, Y receives the two ciphertexts and re-encrypts them using the same public key k_{pub} and uniformly selected randomnesses $r'_1, r'_2 \in \mathit{Rand}_{uni}$. Finally, Y outputs $(\!|\, \{\!|\, c_1 \,|\!\}^{r_1}_{k_{pub}} \,|\!)^{r'_1}_{k_{pub}}$ and $(\!|\, \{\!|\, c_2 \,|\!\}^{r_2}_{k_{pub}} \,|\!)^{r'_2}_{k_{pub}}$ in a random order.

The sequence of the honest participants' messages in this protocol is either M or M'.

$$M = \{\!|\, c_1 \,|\!\}^{r_1}_{k_{pub}}, \ \{\!|\, c_2 \,|\!\}^{r_2}_{k_{pub}}, \ (\!|\, \{\!|\, c_i \,|\!\}^{r_i}_{k_{pub}} \,|\!)^{r'_i}_{k_{pub}}, \ (\!|\, \{\!|\, c_{3-i} \,|\!\}^{r_{3-i}}_{k_{pub}} \,|\!)^{r'_{3-i}}_{k_{pub}}$$

$$M' = \{\!|\, c_2 \,|\!\}^{r_2}_{k_{pub}}, \ \{\!|\, c_1 \,|\!\}^{r_1}_{k_{pub}}, \ (\!|\, \{\!|\, c_j \,|\!\}^{r_j}_{k_{pub}} \,|\!)^{r'_j}_{k_{pub}}, \ (\!|\, \{\!|\, c_{3-j} \,|\!\}^{r_{3-j}}_{k_{pub}} \,|\!)^{r'_{3-j}}_{k_{pub}}$$

Note that M and M' are acyclic and satisfy the freshness assumption. For these two sequences of terms M and M', we obtain the following two patterns.

$$pattern(M, \emptyset) = \Box^{\{\!|Const|\!\}_{k_{pub}}^{r_1}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{r_2}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{\{r_i, r'_i\}}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{\{r_{3-i}, r'_{3-i}\}}}$$
$$pattern(M', \emptyset) = \Box^{\{\!|Const|\!\}_{k_{pub}}^{r_2}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{r_1}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{\{r_j, r'_j\}}},\ \Box^{\{\!|Const|\!\}_{k_{pub}}^{\{r_{3-j}, r'_{3-j}\}}}$$

Since the uniform randomness symbols r_1, r_2, r'_1, and r'_2 are independent in M', there exists a renaming σ such that $\sigma(\{r_j, r'_j\}) = \{r_i, r'_i\}$, $\sigma(\{r_{3-j}, r'_{3-j}\}) = \{r_{3-i}, r'_{3-i}\}$, and $\sigma(\{r_i\}) = \{r_{3-i}\}$ for $i,\ j = 1, 2$. Then we obtain $pattern(M, \emptyset) = \tilde{\sigma}\ pattern(M', \emptyset)$, that is, $M \cong M'$.

Assume that the rerandomizable encryption scheme used in this protocol satisfies IND-RCCA security and the randomness-preserving property. Let η be a security parameter, and $[\![\cdot]\!]_\eta$ be the encoding that uses the scheme. Since M and M' are acyclic and satisfy the freshness assumption, it follows from Theorem 1 that we obtain $[\![M]\!]_\eta \approx_{O_{\eta,t}^{M,M',\emptyset}} [\![M']\!]_\eta$. This implies that no active and adaptive PPT adversary can identify the sender of each plaintext c_i. Hence, we obtain sender anonymity with this simple re-encryption mixnet protocol in the computational sense.

6 Conclusion

We proposed a new formalization of a rerandomizable encryption scheme by using Abadi-Rogaway-style formal patterns, and proved its computational soundness by using IND-RCCA security and the randomness-preserving property. In the formalization, we introduced a new method for dealing with composed randomnesses.

Our method of defining patterns using multisets is not limited to the formalization of rerandomizable encryption schemes. We believe it is also useful in order to provide a computationally sound formalization of other cryptographic primitives, such as threshold cryptography and blind signature.

Acknowledgments

We thank Gergei Bana and the members of the Computing Theory Research Group at NTT for their helpful suggestions. We also thank the reviewers for useful comments.

References

1. M. Abadi and J. Jürjens. Formal eavesdropping and its computational interpretation. In *TACS '01: Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software*, pages 82–94, 2001.
2. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103 – 127, 2002.
3. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Theory and Application of Cryptographic Techniques*, pages 83–107, 2002.
4. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 566–582, 2001.

5. R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 185–194. ACM, 2007.
6. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, pages 565–582, 2003.
7. H. Comon-Lundh. Soundness of abstract cryptography lecture notes, 2007. Available at http://www.lsv.ens-cachan.fr/˜comon/Soundness/.
8. V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *JCS*, 14(1):1–43, 2006.
9. V. Cortier, S. Kremer, R. Küsters, and B. Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. In *FSTTCS*, pages 176–187, 2006.
10. V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *ESOP 2005*, pages 157–171, 2005.
11. D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.
12. F. D. Garcia and P. van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In *Advances in Information and Computer Security. International Workshop on Security (IWSEC 2006)*, pages 33–47, 2006.
13. P. Golle, M. Jakobsson, A. Juels, and P. F. Syverson. Universal re-encryption for mixnets. In *CT-RSA*, pages 163–178, 2004.
14. J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *TCC*, pages 152–170, 2004.
15. J. Herzog. A computational interpretation of Dolev-Yao adversaries. *Theoretical Computer Science*, 340(1):57–81, 2005.
16. H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In *CRYPTO*, pages 310–331, 2001.
17. D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *TCC 2005*, pages 169–187, 2005.
18. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *TCC 2004*, pages 133–151, 2004.
19. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In *CRYPTO*, pages 517–534, 2007.
20. V. Shoup. A proposal for an ISO standard for public key encryption. Input for Committee ISO/IEC JTC 1/SC 27, 2001.
21. R. Xue and D. Feng. Toward practical anonymous rerandomizable RCCA secure encryptions. In *ICICS*, pages 239–253, 2007.