2016

Relations between robustness and RKA security under public-key encryption

Hui Cui
University of Wollongong, hc892@uowmail.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Man Ho Au
University of Wollongong, aau@uow.edu.au

Publication Details

Cui, H., Mu, Y. & Au, M. Ho. (2016). Relations between robustness and RKA security under public-key encryption. Theoretical Computer Science, 628 78-91.

Relations between robustness and RKA security under public-key encryption

Abstract

We revisit the notions of robustness introduced by Abdalla, Bellare and Neven (TCC 2010), and related-key attack (RKA) security raised by Bellare, Cash and Miller (ASIACRYPT 2011). In the setting of public-key encryption (PKE), robustness means that it is hard to produce a ciphertext that is valid for two different users, while RKA security means that a PKE scheme is still secure even when an attacker can induce modifications in a decryption key, and subsequently observe the outcome of this PKE scheme under this modified key. In this paper, we explore the relationship between RKA security and various notions of robustness (weak, strong, complete, and so so). We show, there is no implication between weak (strong) robustness and RKA security while complete robustness implies RKA security but is not implied by RKA security; besides complete robustness, there exist other ROB definitions that can imply RKA security if they meet some security requirements. This result provides a different framework enabling the construction of PKE schemes that are secure under the restricted related key attacks. Also, we instantiate how a robust PKE scheme achieves RKA security, and compare it with other existing ways of achieving RKA security in public-key setting.

Keywords

between, robustness, relations, rka, encryption, security, under, public, key

Disciplines

Engineering | Science and Technology Studies

Publication Details

Cui, H., Mu, Y. & Au, M. Ho. (2016). Relations between robustness and RKA security under public-key encryption. Theoretical Computer Science, 628 78-91.

Relations between Robustness and RKA Security under Public-Key Encryption

Hui Cui*, Yi Mu, Man Ho Au

School of Computer Science and Software Engineering, University of Wollongong, Wollongong NSW2522, Australia

Abstract

We revisit the notions of robustness introduced by Abdalla, Bellare, and Neven (TCC 2010), and related-key attack (RKA) security raised by Bellare, Cash, and Miller (ASIACRYPT 2011). In the setting of public-key encryption (PKE), robustness means that it is hard to produce a ciphertext that is valid for two different users, while RKA security means that a PKE scheme is still secure even when an attacker can induce modifications in a decryption key, and subsequently observe the outcome of this PKE scheme under this modified key. In this paper, we firstly explore the relationship between RKA security and various notions of robustness (weak, strong, complete, and so so), We show, there is no implication between weak (strong) robustness and RKA security while complete robustness implies some kind of RKA security but is not implied by RKA security; besides complete robustness, there exist other ROB definitions that can imply RKA security if they meet some security requirements. This result provides a different framework enabling the construction of PKE schemes that are secure under the restricted related key attacks. Lastly, we instantiate how a robust PKE scheme achieves RKA security, and compare it with other existing ways of achieving RKA security in public-key setting.

Keywords: Robustness, Related-key attack, Public-key encryption.

1. Introduction

The crucial security requirement of encryption is providing privacy of the encrypted data, i.e. data privacy. To capture various requirements of data privacy, formalizations like indistinguishability or non-malleability [17] under either chosen plaintext attacks (CPA) [19] or chosen ciphertext attacks (CCA) are proposed. In recent years, user privacy has become an equally relevant concern, which leads to anonymity, to be another pursued goal in encryption

*Corresponding author

Email addresses: hc892@uowmail.edu.au (Hui Cui), ymu,aau@uow.edu.au (Yi Mu, Man Ho Au) schemes. Anonymity, also known as key privacy, was introduced in [4], meaning that a ciphertext does not leak any information about the public key under which it was created, thereby making the communication anonymous. Under this scenario, a fundamental question was raised in [1]: how does a legal user know whether or not an anonymous ciphertext is intended for him or her? Furthermore, what will happen if a decryption key on a ciphertext was not created for it? To address this issue, robustness was put forward in [1], which guarantees that decryption fails with high possibility if the "wrong" decryption key is used.

1.1. Robust Encryption

Robustness (ROB) ensures a property that a ciphertext cannot be correctly decrypted under two different decryption keys, which has been implicitly used in applications such as bid privacy [25], consistency in searchable encryption [12], anonymous broadcast encryption [3, 21], and anonymous hybrid encryption [23], but the formal definitions was detailed in [1] which introduced two kinds of robustness in encryption: weak robustness (WROB) and strong robustness (SROB). Later, stronger notions called unrestricted strong robustness (USROB), full robustness (FROB), key-less robustness (KROB), mixed robustness (XROB), and complete robustness (CROB) [18] were put forward to provide robustness guarantees in more challenging settings such as the encryption of key-dependent messages [10] or messages encrypted under related keys [6].

WROB can precisely address the issue of using the wrong key that arises in the anonymity contexts such as anonymous broadcast encryption [3, 21], but SROB trumps WROB for applications where ciphertexts can be viciously chosen and goes farther towards making encryption mis-use resistant. CROB is obtained by progressively removing various restrictions on adversarial capabilities in the strong robustness security model. First, the adversary is given access to honestly generated secret keys and arrive at an intermediate notion named unrestricted (strong) robustness (USROB). Next, the honest key-generation requirement is removed to get to the notion of full robustness (FROB). Since full robustness can be viewed as the "decryption-only part" of CROB, another natural notion of robustness called key-less robustness (KROB) arises as the dual notion corresponding to the "encryption-only part" of CROB. Finally, XROB is a "mixed" notion derived from FROB and KROB which is regarded as a useful tool in establishing results about these notions. CROB is the strongest notion among all these notions, different from WROB, SROB and USROB, in CROB the adversary can maliciously generate ciphertexts and the public keys; different from FROB, KROB, and XROB, CROB requires the behavior of the encryption and decryption routines of a scheme with respect to each other.

1.2. RKA Security

In practice, an attacker might induce modifications in a hardware-stored key by fault injection [9] or other means. When the attacker can subsequently observe the outcome of the cryptographic primitive under this modified key, we have a related-key attack (RKA). RKA was first conceived as tools for the cryptanalysis of blockciphers [20, 8], but the ability of attackers to modify keys stored in memory via tampering [13, 9] raises concerns that RKA can actually be mounted to a master key of identity-based encryption (IBE), a signing key of a certificate authority, or a decryption key, making RKA security important for a wide variety of primitives. In this paper, the primitive we target is public-key encryption (PKE), of which the RKA security under chosen ciphertext attacks (CC-RKA security) was defined by Bellare, Cash, and Miller [5].

Efforts to achieve RKA security have been made on a variety of cryptographic primitives [22, 2, 5, 26, 7] such as identity-based encryption, public-key encryption, symmetric encryption, signature. Lucks [22] presented some constructions for block ciphers and pseudorandom function generators. Applebaum, Harnik and Ishai [2] put forward symmetric encryption schemes secure against linear related-key attacks. Bellare, Cash and Miller [5] proposed the approaches to build high-level primitives secure against related-key attacks. Wee [26] designed the CCA secure public-key encryption schemes which are resilient against linear related-key attacks under standard assumptions in the standard model. Bellare, Paterson and Thomson [7] provided a generic framework enabling the RKA-secure identity-based encryption schemes for sets of related-key derivation functions beyond the linear barrier.

1.3. Our Contributions

In this paper, firstly, we study how these notions of robustness as WROB, SROB, CROB relate to RKA security in the setting of public-key encryption. Regarding that the related-key attack is on the decryption key, we are considering their security models under chosen ciphertext attacks. In other words, we talk about the relations between WROB-CCA, SROB-CCA, CROB and CC-RKA under public-key encryption. Figure 1 summarizes the main relations between them. In this figure, the relations among the notions of robustness have been proved in [18], so our emphasis is to explore the relations between them and CC-RKA security. We conclude that neither WROB-CCA nor SROB-CCA has relations with CC-RKA, and CROB implies but is not implied by CC-RKA regarding to *the restricted related-key attacks*, where for a related decryption key, there always exists a corresponding related encryption key, of which the definition we will explain later in Section 3.1.

Next, we investigate under *the restricted related-key attacks*, whether there are other ROB notions that are weaker than CROB but strong enough imply CC-RKA security under the public-key setting. We start from analyzing the relations between other existing ROB notions defined in [18]: unrestricted strong robustness (USROB), full robustness (FROB), key-less robustness (KROB), and mixed robustness (XROB) and CC-RKA security, and sketch our results in Figure 1. During this exploration, we find some interesting results: there do exist other ROB notions that can achieve CC-RKA security, but they need to satisfy two security properties, of which one is the well known CCA security and the other one is termed by us as malicious key attack (MKA) security. We summarize our results in Table 1, and details the analysis in Section 4. The

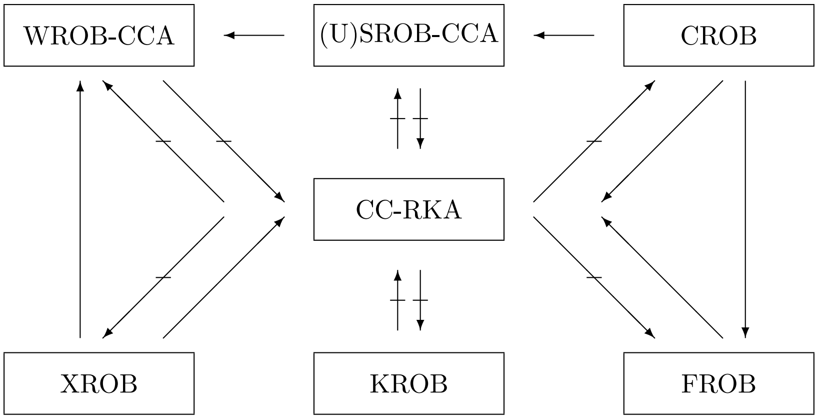

Figure 1: Relations between ROB notions and CC-RKA security.

necessity of CCA security is obvious as RKA security is introduced to resist the attacks on the decryption key. To prove the importance of MKA security in making a robust public-key encryption scheme RKA secure, we show that both WROB and SROB can achieve CC-RKA security if they are MKA secure; besides, the existing ROB notions that can imply RKA security have MKA security in common.

Table 1: Achieving the restricted RKA security from ROB notions. Denote ○ by the meaning of "satisfy", × by the meaning of "fails to satisfy", ∧ by the meaning that it can be implied if satisfying both of the other two, and ✓by the meaning of "imply".

	CCA security	MKA security	CC-RKA security
WROB-CCA	○	×	∧
SROB-CCA	○	×	∧
USROB-CCA	○	×	∧
FROB	○	○	✓
KROB	×	○	∧
XROB	○	○	✓
CROB	○	○	✓

Lastly, we take an example to talk about how a robust scheme achieves RKA security under chosen ciphertext attacks, and compare it with other existing methods for the achievement of RKA security.

The remainder of this paper is organized as follows. In Section 2, we outline the concepts and definitions associated to this work. In Section 3, we elaborate the security models of related-key attacks and three kinds of robustness. In Section 4, we summarize the relations between RKA security and various kinds of ROB notions under the setting of CCA security. In Section 5, we use a robust public-key encryption scheme to show how it achieves RKA security. Finally, we conclude this paper in Section 6.

2. Preliminaries

In this section, we recall some basic notions and definitions about public-key encryption schemes and commitment schemes.

2.1. Public-Key Encryption Scheme

A public-key encryption scheme $\mathcal{PKE}$ is composed of the following four algorithms [5]: parameter generation algorithm PG, key generation algorithm KG, encryption algorithm Enc, and decryption algorithm Dec.

- $\mathrm{PG}(1^\lambda) \to pars$: Taking a security parameter λ as input, this algorithm outputs the public parameters $pars$.
- $\mathrm{KG}(pars) \to (ek,\, dk)$: Taking the public parameters $pars$ as input, this algorithm outputs an encryption key ek and a decryption key dk.
- $\mathrm{Enc}(pars,\, ek,\, m) \to C$: Taking the public parameters $pars$, an encryption key ek, and a message m as input, this algorithm outputs a ciphertext C.
- $\mathrm{Dec}(pars,\, ek,\, dk,\, C) \to m/\perp$: Taking the public parameters $pars$, an encryption key ek, a decryption key dk, and a ciphertext C as input, this algorithm outputs m for a valid ciphertext or $\perp$ for an invalid ciphertext.

We require that $\mathcal{PKE}$ is correct if for any $\lambda \in \mathbb{N}$, $pars \leftarrow \mathrm{PG}(1^\lambda)$, $(ek,\, dk) \leftarrow \mathrm{KG}(pars)$, and $C \leftarrow \mathrm{Enc}(pars,\, ek,\, m)$, we have that $\mathrm{Dec}(pars,\, ek,\, dk,\, C) = m$.

AI-CCA Security. Following the definition described in [1], we briefly revisit the game of AI-CCA security, which models the usual indistinguishability and anonymity under chosen ciphertext attacks (IND-CCA and ANON-CCA) of a PKE scheme $\mathcal{PKE}$ in a single game. The advantage of an adversary algorithm $\mathcal{A}$ in the AI-CCA game is defined as

$$\mathbf{Adv}^{\text{AI-CCA}}_{\mathcal{PKE}}(\mathcal{A}) = |2 \cdot \Pr[\text{AI-CCA}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \text{true}] - 1|,$$

where game AI-CCA is shown in Figure 2. Note that to achieve the IND-CCA security game for $\mathcal{PKE}$, we only need to use $\mathrm{LR}(ek^*,\, M_0^*,\, M_1^*)$ to replace $\mathrm{LR}(ek_0^*,\, ek_1^*,\, M_0^*,\, M_1^*)$ in the AI-CCA security game.

2.2. Commitment Scheme

A commitment scheme $\mathcal{CME}$ is composed of the following three algorithms [5]: parameter generation algorithm CPG, committal algorithm Com and deterministic verification algorithm Ver.

- $\mathrm{CPG}(1^\lambda) \to cpars$: Taking a security parameter λ as input, this algorithm outputs the public parameters $cpars$.
- $\mathrm{Com}(cpars,\, x) \to (com,\, dec)$: Taking a the public parameters $caprs$ and data x as input, this algorithm outputs a commitment com to x along with a decommittal key dec.

```
proc Initialize
$pars \leftarrow \mathrm{PG}(1^\lambda)$; $b \leftarrow \{0,1\}$
$S, T, U, V \leftarrow \emptyset$; Return $pars$
proc GetEK($id$)
$(ek,\ dk) \leftarrow \mathrm{KG}(pars)$
$U \leftarrow U \cup \{ek\}$
Return $ek$
proc GetDK($ek$)
If $ek \notin U$ then return $\perp$
If $ek \in S$ then return $\perp$
$V \leftarrow V \cup \{ek\}$; Return $dk$
proc Finalize($b'$)
Return $(b' = b)$

proc Dec($C,\ ek$)
If $ek \notin U$ then return $\perp$
If $(ek,\ C) \in T$ then return $\perp$
$M \leftarrow \mathrm{Dec}(pars,\ ek,\ dk,\ C)$
Return $M$
proc LR($ek_0^*,\ ek_1^*,\ M_0^*,\ M_1^*$)
If $(ek_0^* \notin U) \vee (ek_1^* \notin U)$ then return $\perp$
If $(ek_0^* \in V) \vee (ek_1^* \in V)$ then return $\perp$
If $|M_0^*| \neq |M_1^*|$ then return $\perp$
$C^* \leftarrow \mathrm{Enc}(pars,\ ek_b^*,\ M_b^*)$ then return $\perp$
$S \leftarrow S \cup \{ek_0^*, ek_1^*\}$
$T \leftarrow T \cup \{(ek_0^*, C^*), (ek_1^*, C^*)\}$
Return $C^*$
```

Figure 2: Game defining AI-CCA security for $\mathcal{PKE}$ = (PG, KG, Enc, Dec). Denote S by public keys involved in the challenge phase, T by the pairs of public key and ciphertext in the challenge phase, U by the registered user (public key) set, V by the set of the registered public keys with corresponding private keys. Here GetEK(id) can be removed from this game, and we keep them to make it consistent with the following security games and convenient for the later description.

- Ver($cpars$, x, com, dec) → 1/0: Taking the public parameter $cpars$, data x, a commitment com, and a decommittal key dec as input, this algorithm outputs 1 to indicate that accepts or 0 to indicate that it rejects.

We call that $\mathcal{CME}$ is correct if for any $\lambda \in \mathbb{N}$, $x \in \{0,1\}^*$, $cpars \leftarrow \mathrm{CPG}(1^\lambda)$, $(com,\ dec) \leftarrow \mathrm{Com}(cpars,\ x)$, we have that $\mathrm{Ver}(cpars,\ x,\ com,\ dec) = 1$.

In most schemes the decommittal key is the randomness used by the committal algorithm and verification is by re-applying the committal function, which ensures uniqueness. The advantages

$$\mathbf{Adv}^{\mathrm{Hiding}}_{\mathcal{CMT}}(\mathcal{A}) = 2 \cdot \Pr[\mathrm{Hiding}^{\mathcal{A}}_{\mathcal{CMT}} \Rightarrow \mathrm{true}] - 1, \quad \text{and}$$
$$\mathbf{Adv}^{\mathrm{Binding}}_{\mathcal{CMT}}(\mathcal{A}) = \Pr[\mathrm{Binding}^{\mathcal{A}}_{\mathcal{CMT}} \Rightarrow \mathrm{true}]$$

referring to the games of the standard hiding and binding properties, respectively, are reviewed in Figure 3.

3. Modeling RKA Security and Robustness

In this section, we briefly revisit the security models of related-key attack security and three kinds of robustness security, respectively.

3.1. RKA Security

Related-key deriving functions. Our definition follows the notion of related-key deriving (RKD) functions given in [6]. A class $\Phi = (\mathcal{P},\ \mathcal{Q})$ of related-key deriving functions is a pair of algorithms: parameter generation algorithm $\mathcal{P}$ and evaluation algorithm $\mathcal{Q}$, of which the second is deterministic.

```
proc Initialize
$cpars \leftarrow \mathrm{CPG}(1^\lambda)$; $b \in \{0,1\}$
Return $cpars$
proc LR($x_0$, $x_1$)
$(com, dec) \leftarrow \mathrm{Com}(cpars, x_b)$
Return $com$
proc Finalize($b'$)
Return $(b' = b)$
```

```
proc Initialize
$cpars \leftarrow \mathrm{CPG}(1^\lambda)$
Return $cpars$
proc Finalize($com$, $x_0$, $dec_0$, $x_1$, $dec_1$)
$d_0 \leftarrow \mathrm{Ver}(cpars, x_0, com, dec_0)$
$d_1 \leftarrow \mathrm{Ver}(cpars, x_1, com, dec_1)$
Return $(x_0 \neq x_1 \wedge d_0 = 1 \wedge d_1 = 1)$
```

Figure 3: Game Hiding$_{\mathcal{CMT}}$ (left) achieves the hiding property and Game Binding$_{\mathcal{CMT}}$ (right) achieves the binding property. Note that LR can only be called once.

- $\mathcal{P}(1^\lambda) \rightarrow pars$: On input a security parameter λ, parameter generation algorithm $\mathcal{P}$ produces parameters $pars$.
- $\mathcal{Q}(pars, dk, \phi) \rightarrow dk'/\perp$: On input the parameters $pars$, a key dk and a description ϕ of a related-key deriving function, the evaluation algorithm $\mathcal{Q}$ returns either a modified key dk' or $\perp$.

We require that for all ϕ, $pars$, either $\mathcal{Q}(pars, dk, \phi) = \perp$ for all dk or for no dk. We let $\Phi_{pars,\phi}(\cdot) = \mathcal{Q}(pars, \cdot, \phi)$.

Restricted related-key attacks. To simplify the description in this paper, we give a definition of related-key deriving (RKD) functions for the restricted related-key attacks, called restricted RKD functions. Let $\mathcal{DK}$ be the decryption key space and $dk \in \mathcal{DK}$ be a decryption key. Let $\mathcal{EK}$ be the encryption key space and $ek \in \mathcal{EK}$ be an encryption key. A class Φ of restricted RKD functions is a class of related-key deriving functions that satisfies the following properties.

- Malleability. Given an RKD function $\phi \in \Phi$, and a decryption key $dk \in \mathcal{DK}$, there should exist an algorithm which outputs a decryption key dk' under ϕ that is distributed identically to the output of $\mathcal{Q}(pars, dk, \phi)$.
- Compatibility. Given an RKD function $\phi \in \Phi$, an encryption key $ek \in \mathcal{EK}$, and a decryption key $dk \in \mathcal{DK}$, there should exist a key generation algorithm KG($pars$) outputting a key pair $(dk', ek') \neq (dk, ek) \in (\mathcal{EK}, \mathcal{DK})$ that equals $dk' \leftarrow \mathcal{Q}(pars, dk, \phi)$ and its corresponding ek' generated via running some RKD function on ek.
- Collision resistance. Given RKD functions $\phi_1, \phi_2 \in \Phi$, and a decryption key $dk \in \mathcal{DK}$, dk_1 generated under $\mathcal{Q}(pars, dk, \phi_1)$ should not be equal to dk_2 generated under $\mathcal{Q}(pars, dk, \phi_1)$ if $\phi_1 \neq \phi_2$.

Note that a lot of key generation algorithms can meet this kind of requirements, but there do exist some algorithms that cannot achieve these conditions. For instance, in RSA encryption algorithm [24], if the decryption key is modified, it is possible that no corresponding encryption key can be found.

CC-RKA Security. On the basis of the games describing RKA security under the chosen ciphertext attacks in [5], we define the games of CC-RKA security for a public-key encryption scheme $\mathcal{PKE}$ = (PG, KG, Enc, Dec) in Figure 4.

```
proc Initialize
$b \leftarrow \{0,1\}$; $C^* \leftarrow \perp$; $pars \leftarrow \mathrm{PG}(1^\lambda)$
Return $pars$
proc GetEK($id$)
$(ek, dk) \leftarrow \mathrm{KG}(pars)$; Return $ek$
proc LR($m_0$, $m_1$)
If $(|m_0| \neq |m_1|)$ then return $\perp$
Return $C^* \leftarrow \mathrm{Enc}(pars, ek, m_b)$

proc Dec($\phi$, $C$)
$dk' \leftarrow \Phi_{pars,\phi}(dk)$
If $dk' = \perp$ then return $\perp$
If $((dk' = dk) \wedge (C = C^*))$
  then return $\perp$
Return $M \leftarrow \mathrm{Dec}(pars, dk', C)$
proc Finalize($b'$)
Return $(b = b')$
```

Figure 4: Game defining CC-RKA security for $\mathcal{PKE}$ = (PG, KG, Enc, Dec). Note that LR can only be called once.

We say that $\mathcal{PKE}$ is CC-RKA secure if the advantage

$$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE},\Phi}(\mathcal{A}) = 2 \cdot \Pr[\text{CC-RKA}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \text{true}] - 1$$

is negligible for any adversary algorithm $\mathcal{A}$.

3.2. Robustness

```
proc Initialize
$U, V \leftarrow \emptyset$; $pars \leftarrow \mathrm{PG}(1^\lambda)$
Return $pars$
proc GetEK($id$)//W, (U)SROB
$(ek, dk) \leftarrow \mathrm{KG}(pars)$
$U \leftarrow U \cup \{ek\}$
Return $ek$
proc GetDK($ek$)//W, (U)SROB
If $ek \notin U$ then return $\perp$
$V \leftarrow V \cup \{ek\}$
Return $dk$
proc Dec($C$, $ek$)//W, (U)SROB
If $ek \notin U$ then return $\perp$
$M \leftarrow \mathrm{Dec}(pars, ek, dk, C)$
Return $M$

proc Finalize($\mathcal{O}$, $ek_0$, $ek_1$, $\mathcal{K}$)
If $(ek_0 \notin U) \vee ((ek_1 \notin U))$
  then return false
If $(ek_1 \in V) \vee ((ek_1 \in V))$//W, SROB
  then return false
If $(ek_0 = ek_1)$ then return false
For $\mathcal{O} = M$, $\mathcal{K} = \perp$//WROB
  $M_0 \leftarrow M$; $C \leftarrow \mathrm{Enc}(pars, ek_0, M_0)$
  $M_1 \leftarrow \mathrm{Dec}(pars, ek_1, dk_1, C)$
For $\mathcal{O} = C$
  For $\mathcal{K} = \perp$//(U)SROB
  For $\mathcal{K} = \{dk_0, dk_1\}$//FROB
    $M_0 \leftarrow \mathrm{Dec}(pars, ek_0, dk_0, C)$
    $M_1 \leftarrow \mathrm{Dec}(pars, ek_1, dk_1, C)$
Return $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$
```

Figure 5: Game defining WROB, (U)SROB and FROB for $\mathcal{PKE}$ = (PG, KG, Enc, Dec).

Robustness is put forth to ensure that it is hard to produce a ciphertext that is valid for two different public keys, which helps to make anonymous encryption more misuse resistant. Following the definitions of SROB and WROB given in [1], we describe them under the CCA security model in the setting of a public-key encryption scheme $\mathcal{PKE}$ in Figure 5. WROB in the setting of CCA security (WROB-CCA) follows the SROB-CCA game except that in the Finalize phase, the adversary in WROB-CCA produces a message M, and C is its encryption under the encryption key of one of the given identities, while in SROB-CCA it produces C directly, and may not obtain it as an honest encryption.

Next, we review the security model of CROB (CROB) for a public-key encryption scheme $\mathcal{PKE}$ in Figure 6. CROB is obtained by progressively removing various restrictions on adversarial capabilities in the strong robustness security model. First, the adversary is given access to honestly generated secret keys, which forms unrestricted (strong) robustness (USROB). Next, the honest key-generation requirement is removed to get to the notion of full robustness (FROB). Since full robustness can be viewed as the “decryption-only part” of CROB, another natural notion of robustness called key-less robustness (KROB) arises as the dual notion corresponding to the “encryption-only part” of CROB. Finally, XROB is a “mixed” notion derived from FROB and KROB which is regarded as a useful tool in establishing results about these notions.

```
proc Initialize
List ← $\emptyset$
$pars \leftarrow \mathrm{PG}(1^\lambda)$
Return $pars$
proc Enc($ek$, $M$, $r$)//CROB
$C \leftarrow \mathrm{Enc}(pars, ek, M; r)$
List ← $(ek, M, C) \cup$ List
proc Dec($ek$, $dk$, $C$)//CROB
$M \leftarrow \mathrm{Dec}(pars, ek, dk, C)$
List ← $(ek, M, C) \cup$ List
```

```
proc Finalize()//CROB
For $(ek_0, M_0, C_0)$, $(ek_1, M_1, C_1)$ ∈ List
  If $(C_0 = C_1 \neq \perp) \wedge (ek_0 \neq ek_1) \wedge$
      $(M_0 \neq \perp \wedge M_1 \neq \perp)$    return true
Return false
proc Finalize($M_0$, $M_1$, $ek_0$, $ek_1$, $r_0$, $r_1$)//KROB
            (($M_0$, $C_1$, $ek_0$, $ek_1$, $r_0$, $dk_1$))//XROB
If $(ek_0 = ek_1)$ then return false
$C_0 \leftarrow \mathrm{Enc}(pars, ek_0, M_0; r_0)$
$C_1 \leftarrow \mathrm{Enc}(pars, ek_1, M_1; r_1)$//KROB
$(M_1 \leftarrow \mathrm{Dec}(pars, ek_1, dk_1, C_1))$//XROB
Return $(C_0 = C_1 \neq \perp)$//KROB
            $(\wedge(M_0 \neq \perp) \wedge (M_1 \neq \perp))$//XROB
```

Figure 6: Game defining CROB, FROB, KROB, and XROB for $\mathcal{PKE}$ = (PG, KG, Enc, Dec).

The WROB, SROB, USROB, FROB, KROB, XROB and CROB advantages of an adversary, for x ∈ {W, S, US, F, K, X, C}, are

$$\mathbf{Adv}^{\mathrm{xROB}}_{\mathcal{PKE}} = \Pr[\mathrm{xROB}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \mathrm{true}].$$

4. Relations Between CC-RKA Security and Notions of ROB-CCA

In this section, we study how the RKA security relates to various notions of robustness under chosen ciphertext attacks.

4.1. From CC-RKA Security to Robustness

Theorem 1. *A CC-RKA secure public-key encryption scheme does not imply WROB-CCA security, SROB-CCA security, and CROB security.*

Proof. It is indicated in [1] that WROB-CCA security is the weakest one among the three robustness definitions, so if a CC-RKA secure public-key encryption scheme is not WROB-CCA secure, then it is definitely neither SROB-CCA secure nor CROB secure. To prove this, we can show that there exists a CC-RKA secure PKE scheme which is insecure in the WROB-CCA security model. In order to cover comprehensively, we use two different cases of RKA secure PKE schemes to discuss their robustness.

- For a query $(\phi,\ C)$ to the decryption oracle, if $\phi(dk) \neq dk$, some message rather than $\perp$ will be returned. In this case, it is straightforward that an adversary can simply win the robustness game, thus this kind of RKA secure schemes are not robust.

- For a query $(\phi,\ C)$ to the decryption oracle, if $\phi(dk) \neq dk$, $\perp$ will be returned. In this case, we assume that there is a PKE scheme which is resilient to related-key attacks, but it fails to achieve even weak robustness. Let $\mathcal{PKE} = (\mathrm{PG},\ \mathrm{KG},\ \mathrm{Enc},\ \mathrm{Dec})$ be a CC-RKA secure pubic-key encryption scheme (for instance, the CCA secure public-key encryption scheme in [26]). Given $\mathcal{PKE}$ and a redundancy code $\mathcal{RED} = (\mathrm{RKG},\ \mathrm{RC},\ \mathrm{RV})$ [1], the encryption with redundancy transform associates to them another encryption scheme $\overline{\mathcal{PKE}} = (\overline{\mathrm{PG}},\ \overline{\mathrm{KG}},\ \overline{\mathrm{Enc}},\ \overline{\mathrm{Dec}})$. Let K be a key generated by the redundancy key generation algorithm RKG, and add K to the public parameters of the above scheme to generate the public parameters for $\overline{\mathrm{PG}}$. $\overline{\mathrm{KG}}$ runs the same as KG. Let $\bar{r}$ be the redundancy generated by the redundancy computation algorithm RC as $\bar{r} \leftarrow \mathrm{RC}(K,\ ek||M)$, and replace m by $m||\bar{r}$ in Enc to generate $\overline{\mathrm{Enc}}$. Now to decrypt a ciphertext C, $\overline{\mathrm{Dec}}$ runs as follows: Run $\overline{\mathrm{Dec}}((pars,\ K),\ ek,\ dk,\ C)$ to obtain $M||\bar{r}$. If $\mathrm{RV}(K,\ ek||M,\ \bar{r}) = 1$, outputs M. Otherwise, it outputs $\perp$.

  Thus, according to [1], we obtain an anonymous public-key encryption scheme $\overline{\mathcal{PKE}}$. Next, to show a counterexample for WROB, we modify it to an encryption scheme $\mathcal{PKE}^* = (\mathrm{PG}^*,\ \mathrm{KG}^*,\ \mathrm{Enc}^*,\ \mathrm{Dec}^*)$, which differs with $\mathcal{PKE}$ only in the decryption phase. To decrypt a ciphertext C, Dec^* runs as follows: Run $\mathrm{Dec}^*(pars,\ ek,\ dk,\ C)$ to obtain M. If $M = \perp$, outputs $M = M^*(pars)||\mathrm{RC}(\varepsilon,\ ek||M^*(pars);\ 0^l)$. If $\mathrm{RV}(\varepsilon,\ ek||M^*(pars),$ $\mathrm{RC}(\varepsilon,\ ek||M^*(pars);\ 0^l)) = 1$, outputs $M^*(pars)$. Otherwise, outputs $\perp$.

  Here, $\mathcal{RED}$ is actually an unkeyed redundancy code [1], *which can be simply replaced by a collision resistant hash function H.* This, to some degree, reflects that AI-CCA does not imply WROB-CCA [1].

  This resulting scheme $\overline{\mathcal{PKE}}$ is still CC-RKA secure (the examples of the proof about CC-RKA security mostly follows that in [26, 15]), since for all the RKA decryption queries, if the decryption result of m is $\perp$, $m^*(pars)||\mathrm{RC}(\varepsilon, ek||m^*(pars); 0^l)$ will be returned rather than $\perp$, which reveals nothing about m_b in the challenge ciphertext.

  However, $\overline{\mathcal{PKE}}$ is not WROB-CCA following the attack in [1]. For id_0, id_1 of distinct identities of its choice, adversary algorithm $\mathcal{A}$, on input $(pars,$

ε), makes queries $ek_0 \leftarrow$ GetEK(id_0) and $ek_1 \leftarrow$ GetEK(id_1) and returns (id_0, id_1, $m^*(pars)$). Let dk_1 be the decryption key corresponding to ek_1, we can show how the adversary breaks the WROB-CCA security of $\overline{\mathcal{PKE}}$. Let $r_0 \leftarrow$ RC(ε, $ek_0||m^*(pars)$) and $C \leftarrow$ Enc($pars$, ek_0, $m^*(pars)||r_0$). The security of $\overline{\mathcal{PKE}}$ implies that the result of Dec($pars$, ek_1, dk_1, C) is $m^*(pars)||r_1(pars)$ where $r_1(pars)$ = RC(ε, $ek_1||m^*(pars)$; 0^l), and the correctness of $\mathcal{RED}$ implies that RV(ε, $ek_1||m^*(pars)$, $r_1(pars)$) = 1, therefore Dec(($pars$, ε), ek_1, dk_1, C) returns $m^*(pars)$ rather than $\perp$.

Remark 1. CC-RKA security cannot provide any robustness in pubic-key encryption schemes. One reason is that CC-RKA security is not able to guarantee that for any related key dk' of a decryption key dk, the decryption of the ciphertext C generated under dk's corresponding encryption key ek on a message M will failure. For example, the linear related-key attack secure public-key encryption scheme in [16], where the decryption of C under dk' returns some message independent of the original plaintext M. In fact, even this problem is solved, it still cannot ensure ROB security, as a RKA secure public-key encryption scheme is still RKA secure if its decryption algorithm is modified to when the decryption fails, a random message in message space will be returned; whilst in this case, robustness will be totally broken.

4.2. From Robustness to CC-RKA Security

Theorem 2. *A WROB-CCA secure public-key encryption scheme does not imply CC-RKA security.*

Proof. To prove this, we show that there exists a WROB-CCA secure PKE scheme which is insecure in the CC-RKA security model. Take the CCA secure encryption scheme based on the DDH assumption proposed by Cramer and Shoup [14] for an instance. In [1], it was concluded that the Cramer-Shoup encryption scheme achieves WROB-CCA security, but [26] pointed out two simple linear related-key attacks on the Cramer-Shoup encryption scheme.

Also, we can show that a public-key encryption scheme $\mathcal{PKE}'$ = (PG$'$, KG$'$, Enc$'$, Dec$'$) transformed from a CCA secure public-key encryption scheme $\mathcal{PKE}$ = (PG, KG, Enc, Dec) of which the algorithms are the same as those in Section 2.1, and an integer parameter k (representing the length of K) based on the weak robustness transform in [1] is not CC-RKA secure.

- PG$'(1^\lambda)$: Run PG($1^)$ to obtain public parameters $pars$, and choose K from $\{0,1\}^k$. Return ($pars$, K).
- KG$'$($pars$, K): Run KG($pars$) to obtain an encryption and decryption key pair (ek, dk). Return (ek, dk).
- Enc$'$(($pars$, K), ek, M): Run Enc($pars$, ek, $M||K$) to obtain ciphertext C. Return C.

- Dec$'$(($pars$, K), ek, dk, C): Run Dec($pars$, ek, dk, C) to obtain the plaintext M^*. If $M^* = \perp$, then return $\perp$. Otherwise, decompose M^* to $M||K'$. If $K' = K$, then return M; else return $\perp$.

$\mathcal{PKE}'$ may not be CC-RKA secure. Assume that $\mathcal{PKE}'$ is CC-RKA secure, then $\mathcal{PKE}$ should be CC-RKA secure as well. Because decryption algorithm Dec$'$ differs from Dec in the validity of the public parameter K, which is irrelevant with the keys involved in $\mathcal{PKE}$, and will not make $\mathcal{PKE}$ be CC-RKA secure. Anyway, in the above transform, $\mathcal{PKE}$ is a normal public-key encryption scheme, and it need not to be CC-RKA secure. Another reason of this conclusion is that a CC-RKA adversary algorithm $\mathcal{A}$ may issue a related-key decryption query (ϕ, C) such that C decrypted under $\phi(dk)$ returns $M^* = M'||K$. Now M' will be returned to algorithm $\mathcal{A}$, but there is no guarantee that M' leaks nothing about M. It could happen that algorithm $\mathcal{A}$ obtains M from M' according to the known information.

Remark 2. Here we can find that WROB-CCA security fails to achieve CC-RKA security is because weak robustness assumes honestly generated public keys — this makes the encryption under one encryption key decrypted under another decryption key succeed with negligible possibility, whereas the adversary, under relate-key attacks, could maliciously modify the stored keys such that there is no idea for the key forms.

Theorem 3. *A SROB-CCA secure public-key encryption scheme does not imply CC-RKA security.*

Proof. To prove this, we show that there exists a SROB-CCA secure PKE scheme which is insecure in the CC-RKA security model. In [1], it was proved that a modified version of the Cramer-Shoup encryption scheme [14] called $\mathcal{CS}^*$ achieves SROB-CCA security, but it is easy to see that the relate-key attacks on the Cramer-Shoup encryption scheme [26] also works on $\mathcal{CS}^*$. The reason is that $\mathcal{CS}^*$ differs from the original Cramer-Shoup encryption scheme by having Enc choose the randomness u to be non-zero, and then having Dec reject (a_1, a_2, c, d) if $a_1 = \mathbf{1}$, which cannot thwart relate-key attacks.

It may be asked for a more general picture about this proof. To start with, we assume that a SROB-CCA secure public-key encryption scheme implies CC-RKA security. This is to say, for a public-key encryption scheme $\mathcal{PKE}$, if there is an adversary algorithm $\mathcal{A}$ that breaks the CC-RKA security, then there exists an adversary algorithm $\mathcal{B}$ that wins the SROB-CCA game by outputting (C, ek, ek'), where (C, ek') is from a winning RKA decryption query (ϕ, C) with the corresponding encryption key ek' for $dk' = \phi(dk)$. However, there are some problems here: firstly, both algorithm $\mathcal{A}$ and algorithm $\mathcal{B}$ have no idea about dk, thereby it is almost impossible for either of them to obtain dk' from dk, not to mention the corresponding ek'; secondly, even algorithm $\mathcal{B}$ is given the access to dk and can compute dk', the possibility that it makes ek' a valid key is negligible. This shows that the assumption that a SROB-CCA secure public-key encryption scheme implies CC-RKA security is wrong, i.e., a SROB-CCA secure public-key encryption scheme does not imply CC-RKA security.

Remark 3. It is not difficult to see that SROB-CCA security fails to achieve CC-RKA security due to that it requires honest key generation, which excludes, with high possibility, the case of different users with the keys being related to each other (i.e., there exists some relations among the keys). This kind of key leakage information could useful for malicious users to explore the hidden messages.

Theorem 4. *Let $\mathcal{PKE}$ be a public-key encryption scheme which is CROB secure. Then $\mathcal{PKE}$ is also CC-RKA secure with respect to the restricted RKD functions.*

Proof. Let algorithm $\mathcal{A}$ be a CC-RKA adversary. Let C be a ciphertext generated under an encryption key ek and a message m. Let ϕ be the RKD function. To make the proof clear, we divide algorithm $\mathcal{A}$ into two types according to its queries.

1. Type I: There exists at least one Dec(ϕ, C) query from algorithm $\mathcal{A}$ such that Dec(ϕ, C) $\neq \perp$ and $\phi(dk) \neq dk$. In this case, we can perform the simulation as follows. Let algorithm $\mathcal{S}$ be the challenger of the CROB security game. We construct a CROB adversary algorithm $\mathcal{B}$, which receives the parameters $pars$ from $\mathcal{S}$, and answers algorithm $\mathcal{A}$'s various queries as follows.
   - GetEK(id): algorithm $\mathcal{B}$ generates a key pair (ek, dk) of encryption key and decryption key, respectively, and passes ek to algorithm $\mathcal{A}$.
   - Dec(ϕ, C): algorithm $\mathcal{B}$ runs $\Phi_{pars,\phi}(dk)$ to obtain dk', uses dk' to compute the corresponding ek', forwards (ek', C) to algorithm $\mathcal{S}$ to obtain the result of the decryption, and returns the result to algorithm $\mathcal{A}$. Algorithm $\mathcal{B}$ continues the above process until it obtains an RKA decryption query Dec(ϕ, C) with Dec(ϕ, C) $\neq \perp$ and $\phi(dk)$ $\neq dk$. We can see how algorithm $\mathcal{B}$ wins the CROB security game as follows. For a query Dec(ϕ, C) that Dec(ϕ, C) $\neq \perp$ (let m' denote this output) and $\phi(dk) \neq dk$, it firstly runs $\Phi_{pars,\phi}(dk)$ to get dk' and computes its corresponding encryption key ek', and then it outputs (ek, m, C) and (ek', m', C).

     As a result, we have

     $$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CROB-CCA}}_{\mathcal{PKE}}(\mathcal{B}).$$

     Note that *the restricted related-key attacks* plays an important role here. Obviously, algorithm $\mathcal{B}$ cannot simulate the challenger perfectly to answer the related-key decryption queries from algorithm $\mathcal{A}$ without the restriction of the RKD functions, because it may not have the ability to generate the corresponding the encryption keys of the related decryption keys.
   - LR(M_0, M_1): algorithm $\mathcal{B}$ forwards (ek, M_b) where $b \in \{0, 1\}$ to algorithm $\mathcal{S}$ to get the challenge ciphertext C^*, and returns C^* to algorithm $\mathcal{A}$.

2. Type II: Any adversary that is not type I. In other words, for any Dec(ϕ, C) queries from algorithm $\mathcal{A}$, Dec(ϕ, C) $= \perp$ if $\phi(dk) \neq dk$. In this case, we can see that the related-key decryption queries do not facilitate algorithm $\mathcal{A}$, so it can be reduced to the adversary in the IND-CCA security game in the case of Dec(ϕ, C) $= m$, $C \neq C^*$ and $\phi(dk) = dk$ with C^* being the challenge ciphertext. Let algorithm $\mathcal{S}'$ be the challenger of the IND-CCA security game. We construct an adversary algorithm $\mathcal{B}'$ breaking the IND-CCA security, which receives the parameters $pars$ from $\mathcal{S}'$, and answers algorithm $\mathcal{A}$'s various queries as follows.

   - GetEK(id): algorithm $\mathcal{S}'$ generates a key pair (ek, dk) of encryption key and decryption key, and sends ek to algorithm $\mathcal{B}'$. Algorithm $\mathcal{B}'$ passes ek to algorithm $\mathcal{A}$.
   - Dec(ϕ, C): algorithm $\mathcal{B}'$ decides whether $\phi(dk) = dk$ from the given ϕ. If $\phi(dk) \neq dk$, it returns $\perp$ to algorithm $\mathcal{A}$; otherwise, it forwards (ek, C) to algorithm $\mathcal{S}'$ to obtain the result of the decryption, and returns the result to algorithm $\mathcal{A}$. Note that $C = C^*$ and $\phi(dk) = dk$ will be ruled out by the definition of the CC-RKA security game.
   - LR(M_0, M_1): algorithm $\mathcal{B}'$ forwards M_0, M_1 to algorithm $\mathcal{S}'$, which chooses $b \in \{0, 1\}$, and runs Enc($pars$, ek, M_b) to generate the challenge ciphertext C^*. Algorithm $\mathcal{B}'$ returns C^* to algorithm $\mathcal{A}$.

   Finally, algorithm $\mathcal{A}$ outputs a guessing $b' \in \{0, 1\}$, and sends b' to algorithm $\mathcal{B}'$. Algorithm $\mathcal{B}'$ outputs b' as its guessing of b.
   Here, we have

   $$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{IND-CCA}}(\mathcal{B}').$$

It is easy to see that algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ in an environment identical to CC-RKA, and algorithm $\mathcal{B}'$ runs algorithm $\mathcal{A}$ in an environment identical to IND-CCA. Furthermore, whenever algorithm $\mathcal{A}$ wins this game, algorithm $\mathcal{B}$ breaks the CROB security or algorithm $\mathcal{B}'$ breaks the IND-CCA security.

To conclude the proof, we have

$$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{CROB}}(\mathcal{B}) + \mathbf{Adv}_{\mathcal{PKE}}^{\text{IND-CCA}}(\mathcal{B}').$$

From the above, it is not difficult to conclude that under chosen ciphertext attacks, both WROB and SROB are too weak to imply RKA security, while CROB is strong enough to achieve RKA security. One question might be raised here: whether there exist other intermediate ROB notions that are weaker than CROB but sufficient to imply RKA security as well, i.e., these ROB notions are in between WROB (or SROB) and CROB? Besides CROB, [18] also defined several other ROB notions: unrestricted strong robustness (USROB), full robustness (FROB), key-less robustness (KROB), and mixed robustness (XROB). In the following, we show their relations to RKA security one by one.

- USROB. It still models the scenarios where keys are honestly generated, which could not simulate the environment of RKA security. Definitely, USROB could not achieve CC-RKA security. To prove this, we assume that a USROB-CCA secure public-key encryption scheme implies CC-RKA security. This is to say, for a public-key encryption scheme $\mathcal{PKE}$, if there is an adversary algorithm $\mathcal{A}$ that breaks the CC-RKA security, then there exists an adversary algorithm $\mathcal{B}$ that wins the USROB-CCA game by outputting $(C,\, ek,\, ek')$, where $(C,\, ek')$ is from a winning RKA decryption query $(\phi,\, C)$ with the corresponding encryption key ek' for $dk' = \phi(dk)$. The problem here is that ek' should be a honestly generated encryption key according to the property of USROB, so should dk', which contradicts with the setting of related-key attacks where there is no such restriction on dk'. Therefore, our assumption of a USROB-CCA secure public-key encryption scheme implying CC-RKA security is not right, and a USROB-CCA secure public-key encryption scheme does not CC-RKA security.

- FROB. Let $\mathcal{PKE}$ be a public-key encryption scheme which is FROB. We assume that $\mathcal{PKE}$ is also CC-RKA secure. To prove this assumption, we divide algorithm $\mathcal{A}$ into two types according to its queries. Let algorithm $\mathcal{A}$ be a CC-RKA adversary. Let C be a ciphertext generated under an encryption key ek and a message m. Let ϕ be the RKD function.

  1. Type I: There exists at least one Dec$(\phi,\, C)$ query from algorithm $\mathcal{A}$ such that Dec$(\phi,\, C) \neq \perp$ and $\phi(dk) \neq dk$. In this case, the same as that in Theorem 4, the simulation can be performed as follows.
     - GetEK(id): The same as that in Theorem 4.
     - Dec$(\phi,\, C)$: The same as that in Theorem 4 except in the winning phase. We can see how algorithm $\mathcal{B}$ wins the FROB security game as follows. For a query Dec$(\phi,\, C)$ that Dec$(\phi,\, C) \neq \perp$ (let m' denote this output) and $\phi(dk) \neq dk$, it firstly runs $\Phi_{pars,\phi}(dk)$ to get dk' and computes its corresponding encryption key ek', and then it outputs $(C,\, ek,\, ek',\, dk,\, dk')$.

       As a result, we have
       $$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{FROB}}_{\mathcal{PKE}}(\mathcal{B}).$$
     - LR$(M_0,\, M_1)$: algorithm $\mathcal{B}$ encrypts $(ek,\, M_b)$ where $b \in \{0, 1\}$ by running Enc$(pars,\, ek,\, M_b)$ to get the challenge ciphertext C^*, and returns C^* to algorithm $\mathcal{A}$.
  2. Type II: Any adversary that is not type I. In other words, for any Dec$(\phi,\, C)$ queries from algorithm $\mathcal{A}$, Dec$(\phi,\, C) = \perp$ if $\phi(dk) \neq dk$. The proof of this part is the same as that in Theorem 4, so we omit the details here.

It is easy to see that algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ in an environment identical to CC-RKA, and algorithm $\mathcal{B}'$ runs algorithm $\mathcal{A}$ in an environment identical to IND-CCA. Furthermore, whenever algorithm $\mathcal{A}$ wins this game, algorithm $\mathcal{B}$ breaks the FROB security or algorithm $\mathcal{B}'$ breaks the IND-CCA security.

To conclude the proof, we have

$$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{FROB}}(\mathcal{B}) + \mathbf{Adv}_{\mathcal{PKE}}^{\text{IND-CCA}}(\mathcal{B}').$$

- XROB. Let $\mathcal{PKE}$ be a public-key encryption scheme which is XROB. We assume that $\mathcal{PKE}$ is also CC-RKA secure. To prove this assumption, we divide algorithm $\mathcal{A}$ into two types according to its queries. Let algorithm $\mathcal{A}$ be a CC-RKA adversary. Let C be a ciphertext generated under an encryption key ek and a message m. Let ϕ be the RKD function.

  1. Type I: There exists at least one Dec(ϕ, C) query from algorithm $\mathcal{A}$ such that Dec(ϕ, C) $\neq \perp$ and $\phi(dk) \neq dk$. In this case, the same as that in 4, we can carry out the simulation as follows.
     - GetEK(id): The same as that in Theorem 4.
     - LR(M_0, M_1): algorithm $\mathcal{B}$ encrypts (ek, M_b) where $b \in \{0,1\}$ by running Enc($pars$, ek, M_b; r) to get the challenge ciphertext C^*, and returns C^* to algorithm $\mathcal{A}$.
     - Dec(ϕ, C): The same as that in Theorem 4 except in the winning phase. Algorithm $\mathcal{B}$ continues the decryption query process until it obtains an RKA decryption query Dec(ϕ, C^*) with Dec(ϕ, C^*) $\neq \perp$ and $\phi(dk) \neq dk$. We can see how algorithm $\mathcal{B}$ wins the XROB security game as follows. For a query Dec(ϕ, C^*) that Dec(ϕ, C^*) $\neq \perp$ (let m' denote this output) and $\phi(dk) \neq dk$, it firstly runs $\Phi_{pars,\phi}(dk)$ to get dk' and computes its corresponding encryption key ek', and then it outputs (M_b, ek, r, C^*, ek', dk'). As a result, we have

     $$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{XROB}}(\mathcal{B}).$$

  2. Type II: Any adversary that is not type I. In other words, for any Dec(ϕ, C) queries from algorithm $\mathcal{A}$, Dec(ϕ, C) $= \perp$ if $\phi(dk) \neq dk$. The proof of this part is the same as that in Theorem 4, so we omit the details here.

It is easy to see that algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ in an environment identical to CC-RKA, and algorithm $\mathcal{B}'$ runs algorithm $\mathcal{A}$ in an environment identical to IND-CCA. Furthermore, whenever algorithm $\mathcal{A}$ wins this game, algorithm $\mathcal{B}$ breaks the XROB security or algorithm $\mathcal{B}'$ breaks the IND-CCA security.

To conclude the proof, we have

$$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{XROB}}(\mathcal{B}) + \mathbf{Adv}_{\mathcal{PKE}}^{\text{IND-CCA}}(\mathcal{B}').$$

- KROB. It does not involve decryption oracle, so it is impossible to achieve CC-RKA security, which should be analyzed under chosen ciphertext attacks. Note that if we added IND-CCA security to the security game of KROB, then it is easy to see that KROB becomes similar to CROB, or we can say that it is a loose definition of CROB security game. There is no doubt that it can imply CC-RKA security.

Remark 4. From the above analysis, it is easy to conclude that IND-CCA security plays an important role in achieving CC-RKA security. It is meaningless to talk about RKA security under a robust public-key encryption scheme which is not CCA secure.

4.3. MKA Security

Previously, we did an analysis on the relations between various ROB notions and RKA security. A natural question could be asked: if there is an additional property that could be added to a WROB scheme to achieve CC-RKA security? Indeed, there is a notion which we call malicious key attack (MKA) security can be made use of to obtain CC-RKA security from WROB. We say a robust public-key encryption scheme is MKA secure, means that it is secure under malicious key attacks, where an adversary is allowed to adversarially choose the keys in the system. To prove that an additional property called MKA security is able to achieve CC-RKA security from a CCA secure public-key encryption scheme with weak robustness, we modify the security game of WROB, and describe it Figure 7, which we call stronger weak robustness (SWROB),

```
proc Initialize
$pars \leftarrow \mathrm{PG}(1^\lambda)$
Return $(pars)$
proc Dec($ek$, $dk$, $C$)
$M \leftarrow \mathrm{Dec}(pars, ek, dk, C)$
Return $M$

proc Finalize($M$, $ek_0$, $ek_1$))
If $(ek_0 = ek_1)$ then return false
$M_0 \leftarrow M$; $C \leftarrow \mathrm{Enc}(pars, ek_0, M_0)$
$M_1 \leftarrow \mathrm{Dec}(pars, ek_1, dk_1, C)$
Return $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$
```

Figure 7: Game defining SWROB-CCA for $\mathcal{PKE}$ = (PG, KG, Enc, Dec).

Theorem 5. *Let $\mathcal{PKE}$ be a public-key encryption scheme which is SWROB-CCA secure. Then $\mathcal{PKE}$ is also CC-RKA secure with respect to the restricted RKD functions.*

Proof. Let algorithm $\mathcal{A}$ be a CC-RKA adversary. Let C be a ciphertext generated under an encryption key ek and a message m. Let ϕ be the RKD function. To make the proof clear, we divide algorithm $\mathcal{A}$ into two types according to its queries.

- Type I: There exists at least one Dec(ϕ, C) query from algorithm $\mathcal{A}$ such that Dec(ϕ, C) $\neq \perp$ and $\phi(dk) \neq dk$. In this case, we can do the simulation as follows. Let algorithm $\mathcal{S}$ be the challenger of the SWROB-CCA security game. We construct a SWROB-CCA adversary algorithm $\mathcal{B}$, which receives the parameters $pars$ from $\mathcal{S}$, and answers algorithm $\mathcal{A}$'s various queries as follows.

- GetEK(id): The same as that in Theorem 4.
- Dec(ϕ, C): The same as that in Theorem 4 except in the winning phase. We can see how algorithm $\mathcal{B}$ wins the SWROB-CCA security game as follows. For a query Dec(ϕ, C) that Dec(ϕ, C) $\neq \perp$ (let m' denote this output) and $\phi(dk) \neq dk$, it firstly runs $\Phi_{pars,\phi}(dk)$ to get dk' and computes its corresponding encryption key ek', and then it outputs (m, ek, ek').

  As a result, we have

$$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{SWROB-CCA}}_{\mathcal{PKE}}(\mathcal{B}).$$

- LR(M_0, M_1): algorithm $\mathcal{B}$ encrypts (ek, M_b) where $b \in \{0, 1\}$ by running Enc($pars$, ek, M_b) to get the challenge ciphertext C^*, and returns C^* to algorithm $\mathcal{A}$.

- Type II: Any adversary that is not type I. In other words, for any Dec(ϕ, C) queries from algorithm $\mathcal{A}$, Dec(ϕ, C) $= \perp$ if $\phi(dk) \neq dk$. The proof of this part is the same as that in Theorem 4, so we omit the details here.

It is easy to see that algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ in an environment identical to CC-RKA, and algorithm $\mathcal{B}'$ runs algorithm $\mathcal{A}$ in an environment identical to IND-CCA. Furthermore, whenever algorithm $\mathcal{A}$ wins this game, algorithm $\mathcal{B}$ breaks the CROB security or algorithm $\mathcal{B}'$ breaks the IND-CCA security.

To conclude the proof, we have

$$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{SWROB-CCA}}_{\mathcal{PKE}}(\mathcal{B}) + \mathbf{Adv}^{\text{IND-CCA}}_{\mathcal{PKE}}(\mathcal{B}').$$

In short, SWROB is an improved version of WROB by removing the condition of honest key generation, and differs from CROB in that it allows an adversary to create a ciphertext that decrypts correctly under another decryption key. If an adversary wins WROB game, definitely it wins SWROB game; besides, when an adversary wins SWROB game, definitely it wins CROB game. However, **SWROB does not imply SROB**. That is, for a public-key encryption scheme $\mathcal{PKE}$ which is SWROB, there exists a scheme $\mathcal{PKE}'$ which is SWROB, but fails to be SROB. We define $\mathcal{PKE}'$ to be the same as $\mathcal{PKE}$ (of which the algorithms are the same as those in Section 2.1) except for the encryption and decryption algorithms, which will be modified as follows.

- Enc$'$($pars$, ek, M): Run Enc($pars$, ek, M) to obtain ciphertext C, and output $0||C$.
- Dec$'$($pars$, ek, dk, $c||C$): If $c = 0$, output Dec($pars$, ek, dk, C). If $c = 1$, output a fixed message M^* (in the message space for ek).

$\mathcal{PKE}'$ is a correct public-key encryption scheme, but it is not SROB-CCA secure. Suppose that the adversary algorithm $\mathcal{A}$ queries KG twice to obtain two distinct encryption keys ek_0 and ek_1 with overwhelming probability. Then algorithm $\mathcal{A}$ chooses a random C from the ciphertext space and gives $(1||C$, pk_0, $pk_1)$ as its final output. Obviously, algorithm $\mathcal{A}$ wins the game with the output $(ek_0$, ek_1, $1||C)$ where $ek_0 \neq ek_1$ which always outputs a valid message M^*. Anyway, $\mathcal{PKE}'$ is still SWROB, because the tweaks in the modification do not affect the SWROB game — the new encryption algorithm just adds a zero-bit to all ciphertexts.

In order to convince the essentiality in achieving RKA security, we also simply add MKA security to (U)SROB game to make it become CC-RKA secure, which we call weak complete robustness (WCROB) because it is very close to the security game of CROB, and depict its security game in Figure 8. WCROB is a strengthened definition of SROB by removing the condition of honest key generation. If an adversary wins (U)SROB game, then it wins WCROB game as well. Compared to the security game of CROB, WCROB removes the adversary's interaction with the encryption routine on plaintexts, keys, and random coins of its choice. If an adversary breaks WCROB security, it breaks CROB security as well.

```
proc Initialize
  $pars \leftarrow \mathrm{PG}(1^\lambda)$
  Return $(pars)$
proc Dec($ek$, $dk$, $C$)
  $M \leftarrow \mathrm{Dec}(pars, ek, dk, C)$
  Return $M$

proc Finalize($C$, $ek_0$, $ek_1$))
  If $(ek_0 = ek_1)$ then return false
  $M_0 \leftarrow \mathrm{Dec}(pars, ek_0, dk_0, C)$
  $M_1 \leftarrow \mathrm{Dec}(pars, ek_1, dk_1, C)$
  Return $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$
```

Figure 8: Game defining WCROB-CCA for $\mathcal{PKE}$ = (PG, KG, Enc, Dec).

Theorem 6. *Let $\mathcal{PKE}$ be a public-key encryption scheme which is WCROB-CCA secure. Then $\mathcal{PKE}$ is also CC-RKA secure with respect to the restricted RKD functions.*

Proof. Let algorithm $\mathcal{A}$ be a CC-RKA adversary. Let C be a ciphertext generated under an encryption key ek and a message m. Let ϕ be the RKD function. To make the proof clear, we divide algorithm $\mathcal{A}$ into two types according to its queries.

- Type I: There exists at least one Dec(ϕ, C) query from algorithm $\mathcal{A}$ such that $\mathrm{Dec}(\phi, C) \neq \perp$ and $\phi(dk) \neq dk$. In this case, the same as that in Theorem 5, we can do the simulation as follows.
  - GetEK(id): The same as that in Theorem 5.
  - Dec(ϕ, C): The same as that in Theorem 4 except in the winning phase. We can see how algorithm $\mathcal{B}$ wins the WCROB-CCA security

game as follows. For a query Dec(ϕ, C) that Dec(ϕ, C) $\neq \perp$ (let m' denote this output) and $\phi(dk) \neq dk$, it firstly runs $\Phi_{pars,\phi}(dk)$ to get dk' and computes its corresponding encryption key ek', and then it outputs (C, ek, ek').

As a result, we have

$$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{WCROB-CCA}}(\mathcal{B}).$$

  - LR(M_0, M_1): The same as that in Theorem 5.

- Type II: Any adversary that is not type I. In other words, for any Dec(ϕ, C) queries from algorithm $\mathcal{A}$, Dec(ϕ, C) $= \perp$ if $\phi(dk) \neq dk$. The proof of this part is the same as that in Theorem 4, so we omit the details here.

It is easy to see that algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ in an environment identical to CC-RKA, and algorithm $\mathcal{B}'$ runs algorithm $\mathcal{A}$ in an environment identical to IND-CCA. Furthermore, whenever algorithm $\mathcal{A}$ wins this game, algorithm $\mathcal{B}$ breaks the CROB security or algorithm $\mathcal{B}'$ breaks the IND-CCA security.

To conclude the proof, we have

$$\mathbf{Adv}_{\mathcal{PKE}}^{\text{CC-RKA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathcal{PKE}}^{\text{WCROB-CCA}}(\mathcal{B}) + \mathbf{Adv}_{\mathcal{PKE}}^{\text{IND-CCA}}(\mathcal{B}').$$

4.4. Summary

We have seen that both WROB and SROB could not achieve CC-RKA secure because they are not MKA secure, and FROB, XROB, KROB are already MKA secure and they imply CC-RKA security; KROB is MKA secure but it fails to achieve CC-RKA security as it is not secure under chosen ciphertext attacks; if KROB satisfies IND-CCA security, it becomes a variant security game of CROB and implies CC-RKA security; when adding MKA security to WROB and SROB, they can imply CC-RKA security as well. Therefore, we can say that a robust public-key encryption scheme that implies CC-RKA security under restricted related-key attacks should be both IND-CCA secure and MKA secure; otherwise, it could not be resilient to restricted related-key attacks.

If looking deeper to the proof part of all the above theorems, we can find that algorithm $\mathcal{B}$ does the least work in the CROB one for the simulation of queries from algorithm $\mathcal{A}$, it prevents other adversarial behaviors that could happen in the ROB security games. This, to some degree, reflects the flexibility of complete robust, and its strongest property among all the ROB notions. Furthermore, it convinces the correctness of an underlying public-key encryption scheme.

Let “Enc” be Encryption, “Dec” be decryption, “KG” be key generation, “Rand” be randomness, “Out ROB” be other attacks outside the ROB model secure (i.e., the adversary may not follow the encryption or decryption routine of the original public-key encryption scheme). We summarize the above discussed ROB notions in Table 2.

Remark 5. As mentioned in [18], CROB is obtained by progressively removing various restrictions on adversarial capabilities in the strong robustness security

Table 2: Summary of all the ROB notions

	Out ROB	Rand output	*dk* output	Enc query	Dec query	Access to honest *dk*	Honest KG
WROB	no	no	no	no	yes	no	yes
SROB	no	no	no	no	yes	no	yes
USROB	no	no	no	no	yes	yes	yes
FROB	no	no	yes	no	yes	yes	no
KROB	no	yes	no	yes	no	yes	no
XROB	no	yes	yes	yes	yes	yes	no
CROB	yes	no	no	yes	yes	yes	no

model, and viewing robustness in terms of the behavior of the encryption and decryption algorithms with respect to each other. On the other side, in the security game of CROB, an adversary is given the shared parameters of the system, which then arbitrarily interacts with the encryption and decryption routines on plaintexts, ciphertexts, keys, and even random coins of its choice, which meets the underlying intuition behind not only the original weak robustness notion, but also the standard correctness criterion for a PKE scheme, and excludes the other attacks falling outside the other ROB security models. Due to all these facts, CROB is a very precise definition, which covers all the other ROB security games, and is flexible to achieve RKA security.

5. Discussion

In the previous section, we summarize the relations between RKA security and notions of robustness under PKE setting. Here, we describe a framework for creating RKA-secure PKE schemes beyond the linear barrier (affine, polynomial, and so on), and compare it with the method proposed in [7].

5.1. Construction

Given a public-key encryption scheme $\mathcal{PKE}$ = (PG, KG, Enc, Dec) and a commitment scheme $\mathcal{CMT}$ = (CPG, Com, Ver), we transform them to a public-key encryption scheme $\overline{\mathcal{PKE}}$ = ($\overline{\text{PG}}$, $\overline{\text{KG}}$, $\overline{\text{Enc}}$, $\overline{\text{Dec}}$) via the strong robustness transform in [1], of which the algorithms are depicted in Figure 9.

Note that we require *ek* in Ver(*cpars*, *ek*, *com*, *dec*) = 1 to be recomputed by running the key generation algorithm on *dk*.

5.2. Security Proof

Theorem 7. *Let* $\mathcal{PKE}$ = (PG, KG, Enc, Dec) *be a secure public-key encryption scheme, and let* $\overline{\mathcal{PKE}}$ = ($\overline{\text{PG}}$, $\overline{\text{KG}}$, $\overline{\text{Enc}}$, $\overline{\text{Dec}}$) *be the public-key encryption scheme resulting from applying the strong robustness transform to* $\mathcal{PKE}$ *and a commitment scheme* $\mathcal{CMT}$ = (CPG, Com, Ver)*. Then*

```
Algorithm $\overline{\mathrm{PG}}(1^\lambda)$
$pars \leftarrow \mathrm{PG}(1^\lambda)$
$cpars \leftarrow \mathrm{CPG}(1^\lambda)$
Return Pars $= (pars,\ cpars)$
Algorithm $\overline{\mathrm{Enc}}((\mathrm{Pars}),\ ek,\ \overline{M})$
$(com,\ dec) \leftarrow \mathrm{Com}(cpars,\ ek)$
$C \leftarrow \mathrm{Enc}(pars,\ ek,\ \overline{M}||dec)$
Return $(C,\ com)$

Algorithm $\overline{\mathrm{KG}}(\mathrm{Pars})$
$(ek,\ dk) \leftarrow \mathrm{KG}(pars)$
Return $(ek,\ dk)$
Algorithm $\overline{\mathrm{Dec}}((\mathrm{Pars}),\ ek,\ dk,\ (C,\ com))$
$M \leftarrow \mathrm{Dec}(pars,\ ek,\ dk,\ C)$
If $M = \perp$ then return $\perp$
$\overline{M}||dec \leftarrow M$
If $(\mathrm{Ver}(cpars,\ ek,\ com,\ dec) = 1)$
   then return $\overline{M}$
Else Return $\perp$
```

Figure 9: A public-key encryption scheme $\overline{\mathcal{PKE}} = (\overline{\mathrm{PG}}, \overline{\mathrm{KG}}, \overline{\mathrm{Enc}}, \overline{\mathrm{Dec}})$ resulting from applying strong robustness transform [1] to a public-key encryption scheme $\mathcal{PKE} = (\mathrm{PG}, \mathrm{KG}, \mathrm{Enc}, \mathrm{Dec})$ and a commitment scheme $\mathcal{CMT} = (\mathrm{CPG}, \mathrm{Com}, \mathrm{Ver})$.

1. *AI-CCA: Let $\mathcal{A}$ be an adversary algorithm against the AI-CCA security of $\overline{\mathcal{PKE}}$. Then there is an adversary algorithm $\mathcal{W}$ against the WROB-CCA security of $\mathcal{PKE}$, an adversary algorithm $\mathcal{H}$ against the the hiding security of $\mathcal{CMT}$, and an adversary algorithm $\mathcal{B}$ against the AI-CCA security of $\mathcal{PKE}$ such that*

$$\begin{aligned}\mathbf{Adv}^{\text{AI-CCA}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) &\leq 2 \cdot \mathbf{Adv}^{\text{WROB-CCA}}_{\mathcal{PKE}}(\mathcal{W})\\ &+ 2 \cdot \mathbf{Adv}^{\text{Hiding}}_{\mathcal{CMT}}(\mathcal{H})\\ &+ 3 \cdot \mathbf{Adv}^{\text{AI-CCA}}_{\mathcal{PKE}}(\mathcal{B}).\end{aligned}$$

2. *SROB-CCA: Let $\mathcal{A}$ be an adversary algorithm against the SROB-CCA security of $\overline{\mathcal{PKE}}$ making q GetEK queries. Then there is an adversary algorithm $\mathcal{B}$ against the the biding security of $\mathcal{CMT}$ such that*

$$\mathbf{Adv}^{\text{SROB-CCA}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{Binding}}_{\mathcal{CMT}}(\mathcal{B}) + \binom{q}{2} \cdot \mathbf{Coll}_{\mathcal{PKE}}.$$

__Coll__ denotes the event that there exist distinct ek_0 and ek_1 queried by algorithm $\mathcal{S}$ to the GetEK oracle such that the encryption keys returned in response are the same.

3. *CC-RKA: Let $\mathcal{A}$ be an adversary algorithm against the CC-RKA security of $\overline{\mathcal{PKE}}$ under the restricted RKD functions. Then there is an adversary algorithm $\mathcal{I}$ against the the AI-CCA security of $\overline{\mathcal{PKE}}$, and an adversary algorithm $\mathcal{B}$ against the the binding security of $\overline{\mathcal{CMT}}$ such that*

$$\mathbf{Adv}^{\text{CC-RKA}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{AI-CCA}}_{\overline{\mathcal{PKE}}}(\mathcal{I}) + \mathbf{Adv}^{\text{Binding}}_{\mathcal{CMT}}(\mathcal{B}).$$

Proof. The first part of the theorem implies that if $\mathcal{PKE}$ is AI-CCA secure and WROB-CCA secure, and $\mathcal{CMT}$ is Hiding secure, then $\overline{\mathcal{PKE}}$ is AI-CCA secure. The second part of the theorem implies that if $\mathcal{CMT}$ is binding secure, and $\mathcal{PKE}$ has low encryption key collision probability then $\overline{\mathcal{PKE}}$ is SROB-CCA secure. The encryption key collision probability $\mathbf{Coll}_{\mathcal{PKE}}$ of $\mathcal{PKE}$ is defined as the maximum probability that $ek_0 = ek_1$ in the following game.

1. $pars \leftarrow \mathrm{PG}(1^\lambda)$;
2. $(ek_0,\, dk_0) \leftarrow \mathrm{KG}(pars,\, ek_0)$;
3. $(ek_1,\, dk_1) \leftarrow \mathrm{KG}(pars,\, ek_0)$.

Here the maximum is over all the distinct ek_0, ek_1. Obviously, $\mathcal{PKE}$ being AI-CCA implies that $\mathbf{Coll}_{\mathcal{PKE}}$ is negligible, so the requirement for low encryption key collision probability is actually not an extra assumption [1]. Proofs of both parts follows that of Theorem 4.2 in [1], and we omit the details here.

Proof of Part 3 of Theorem 7: In CC-RKA security game (See Figure 10), it is required that all the queries issued by the adversary should be responded with a related decryption key dk' rather than the original one dk.

```
proc Initialize
$pars \leftarrow \mathrm{PG}(1^\lambda)$
$cpars \leftarrow \mathrm{CPG}(1^\lambda)$
$b \leftarrow \{0,1\}$
$S, U, V \leftarrow \emptyset$; $C^* \leftarrow \perp$
$com^* \leftarrow \perp$; $ek_0^* \leftarrow \perp$; $ek_1^* \leftarrow \perp$
Return $(pars,\, cpars)$
proc GetEK($id$)
$(ek,\, dk) \leftarrow \mathrm{KG}(pars)$
$U \leftarrow U \cup \{ek\}$
Return $ek$
proc GetDK($ek$)
If $ek \notin U$ then return $\perp$
$V \leftarrow V \cup \{ek\}$
Return $dk$
proc Finalize($b'$)
Return $(b' = b')$
```

```
proc LR($ek_0^*,\, ek_1^*,\, \overline{M}_0^*,\, \overline{M}_1^*$)
If $(ek_0^* \notin U) \vee (ek_1^* \notin U)$ then return $\perp$
If $(ek_0^* \in V) \vee (ek_1^* \in V)$ then return $\perp$
$(com^*,\, dec^*) \leftarrow \mathrm{Com}(cpars,\, ek_b^*)$
$C^* \leftarrow \mathrm{Enc}(pars,\, ek_b^*,\, \overline{M}_b^* || dec^*)$
Return $(C^*,\, com^*)$
proc Dec($\phi$, $(C,\, com)$, $ek$)
$dk' \leftarrow \Phi_{pars,\phi}(dk)$
If $(dk' = dk_b^*) \wedge (C, com) = (C^*, com^*)$
    then return $\perp$
If $(dk' = dk_{1-b}^* \neq dk_b^*) \wedge$
    $(C,\, com) = (C^*,\, com^*)$ then return $\perp$
$M' \leftarrow \mathrm{Dec}(pars,\, ek,\, dk',\, C)$
If $M' = \perp$ then return $\perp$
$\overline{M'} || dec' \leftarrow M'$
If $\mathrm{Ver}(cpars,\, ek',\, com,\, dec') = 1$
    then return $\overline{M'}$
Else return $\perp$
```

Figure 10: The first game of the proof of Part 3 of Theorem 7. Note that ek' is the corresponding encryption key of dk'.

The trick here is that for any input $(\phi,\, (C,\, com))$ to the decryption oracle, as long as $dk' \neq dk$, a failure symbol $\perp$ will be returned. This is guaranteed by the binding property of the commitment scheme $\mathcal{CMT}$, from which we have

$$\begin{aligned}&\Pr[\mathrm{Ver}(cpars, ek', com, dec') = 1 \wedge \\ &\quad \mathrm{Ver}(cpars, ek, com, dec) = 1] \leq \mathbf{Adv}_{\mathcal{CMT}}^{\mathrm{Binding}}(\mathcal{B}),\end{aligned}$$

where ek' is related encryption key corresponding to dk' (note that this can be achieved if the queries are from the restricted RKD functions). Based on this, we can modify the process of the first game in the proof of Part 1 of Theorem 7 to that of game G_1 (See Figure 11). We can see that the two games becomes equivalent when $dk' = dk$, because for all $dk' \neq dk$ it will return $\perp$. At this level, CC-RKA security becomes consistent to SROB-CCA security. As for other parts of this proof, we can modify game G_1 following the proof of Part 1 of Theorem 7, we omit the details here.

```
proc Dec($(C, com), ek$)
If $ek \notin U$ then return $\perp$
If $(ek = ek^*_b) \wedge (C, com) = (C^*, com^*)$
  then return $\perp$
If $(ek = ek^*_{1-b} \neq ek^*_b) \wedge$
  $(C, com) = (C^*, com^*)$
  then return $\perp$
$M \leftarrow \mathrm{Dec}(pars, ek, dk, C)$
If $M = \perp$ then return $\perp$
$\overline{M}||dec \leftarrow M$
If $\mathrm{Ver}(cpars, ek, com, dec) = 1$
  then return $\overline{M}$
Else return $\perp$
```

```
proc Dec($\phi, (C, com), ek$) // $\mathrm{G}_1$
If $ek \notin U$ then return $\perp$
$dk' \leftarrow \Phi_{pars,\phi}(dk)$
If $(dk' = dk^*_b) \wedge (C, com) = (C^*, com^*)$
  then return $\perp$
If $(dk' = dk^*_{1-b} \neq dk^*_b) \wedge$
  $(C, com) = (C^*, com^*)$ then return $\perp$
If $(dk' = dk)$ then
  $M' \leftarrow \mathrm{Dec}(pars, ek', dk', C)$
  If $M' = \perp$ then return $\perp$
  $\overline{M'}||dec' \leftarrow M'$
  If $\mathrm{Ver}(cpars, ek', com, dec') = 1$
    then return $\overline{M'}$
Else return $\perp$
```

Figure 11: Left: The first game of the proof of Part 1 of Theorem 4.2 in [1]. Right: The second game of the proof of Part 3 of Theorem 7.

To conclude the proof, we have

$$\mathbf{Adv}^{\text{CC-RKA}}_{\mathcal{PKE}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{AI-CCA}}_{\mathcal{PKE}}(\mathcal{I}) + \mathbf{Adv}^{\text{Binding}}_{\mathcal{CMT}}(\mathcal{B}).$$

5.3. Comparison

In the following, we compare our framework of enabling RKA-secure public-key encryption beyond the linear barrier with the one depicted in [7].

- In [7], one general framework was described for creating RKA-secure identity-based encryption schemes via Identity Renaming Transform, which allows to reduce RKA security of a modified identity-based encryption scheme directly to the normal identity-based encryption security of a base identity-based encryption scheme. After that, they obtain the first constructions of RKA-secure schemes for public-key encryption with CCA security derived from identity-based encryption via the CHK transform [11]. Our generic method, from a different point of view, constructs RKA-secure public-key encryption schemes from public-key encryption schemes and commitment schemes, which reduces RKA security of a modified public-key encryption scheme to the normal public-key encryption security of this scheme because of the binding property of the commitment scheme.

- In [7], it is required that the identity-based encryption scheme used to achieve RKA security should be key malleable with a key simulator and the renaming scheme is statistically collision-resistant. Our construction puts some restrictions on the related-key deriving functions, which we name as *the restricted related-key deriving functions*, to perfectly simulate the proof; otherwise, the proof cannot go well.

To sum up, we can see that achieving RKA security in PKE setting by applying a transform to a public-key encryption scheme and a commitment scheme is very convenient yet it can only hold under the restricted RKD functions. However, a lot of key generation algorithms can meet the requirements of RKD functions, so it is still a feasible way to achieve RKA secure PKE schemes in practice.

6. Conclusions

We review the notions of robustness introduced in [1], and related-key attack (RKA) security proposed in [5]. In public-key encryption (PKE) schemes, robustness means that it is hard to produce a ciphertext that is valid for two different users, while RKA security means that a PKE scheme is still secure even when an adversary can modify a stored decryption key and observe outcomes of the public-key encryption scheme under this modified key. Inspired by this, in this paper, we put emphasis on thinking whether there are relations between (weak, strong and complete) robustness and RKA security. After analyzing several robust PKE schemes and PKE schemes secure against related-key attacks, we conclude that there is no implication between weak (strong) robustness and RKA security, and complete robustness implies the restricted RKA security but it is not implied by RKA security. This result shows another framework enabling the construction of PKE schemes that are secure against related-key attacks beyond the linear barrier. Moreover, we make an analysis on what kind of properties a robust public-key encryption scheme should satisfy in order to achieve RKA security. At last, we take an instance of a robust public-key encryption scheme that is secure against related-key attacks, and do a comparison between this method and other known ways for the achievement of RKA security in public-key encryption.

References

[1] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2010.

[2] B. Applebaum, D. Harnik, and Y. Ishai. Semantic security under related-key attacks and applications. In *ICS*, volume 2011 of *Tsinghua University Press*. Tsinghua University Press.

[3] A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer, 2006.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001.

[5] M. Bellare, D. Cash, and R. Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2011.

[6] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2003.

[7] M. Bellare, K. G. Paterson, and S. Thomson. Rka security beyond the linear barrier: Ibe, encryption and signatures. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 331–348. Springer, 2012.

[8] E. Biham. New types of cryptoanalytic attacks using related keys (extended abstract). In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 1993.

[9] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[10] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. *IACR Cryptology ePrint Archive*, 2002:100, 2002.

[11] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[12] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[13] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[14] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *IACR Cryptology ePrint Archive*, 2001:108, 2001.

[15] H. Cui, Y. Mu, and M. H. Au. Anonymous signcryption against linear related-key attacks. In *Provable Security - 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23-25, 2013. Proceedings*, volume 8209, pages 165–183. Springer, 2013.

[16] H. Cui, Y. Mu, and M. H. Au. Public-key encryption resilient to linear related-key attacks. In *Security and Privacy in Communication Networks - 9th International ICST Conference, SecureComm 2013, Sydney, NSW, Australia, September 25-28, 2013, Revised Selected Papers*, volume 127, pages 182–196. Springer, 2013.

[17] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552. ACM, 1991.

[18] P. Farshim, B. Libert, K. G. Paterson, and E. A. Quaglia. Robust encryption, revisited. In *Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 352–368. Springer, 2013.

[19] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[20] L. R. Knudsen. Cryptanalysis of loki91. In *AUSCRYPT*, volume 718 of *Lecture Notes in Computer Science*, pages 196–208. Springer, 1992.

[21] B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 206–224. Springer, 2012.

[22] S. Lucks. Ciphers secure against related-key attacks. In *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 2004.

[23] P. Mohassel. A closer look at anonymity and robustness in encryption schemes. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2010.

[24] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[25] K. Sako. An auction protocol which hides bids of losers. In *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 422–432. Springer, 2000.

[26] H. Wee. Public key encryption against related key attacks. In *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2012.