A preliminary version of this paper appears in the proceedings of the *21st International Conference on Cryptology in India (INDOCRYPT 2020)*. This is the full version.

Security of Public Key Encryption against Resetting Attacks

Juliane Krämer and Patrick Struck

Technische Universität Darmstadt, Germany
{juliane,patrick}@qpc.tu-darmstadt.de

Abstract. Ciphertext indistinguishability under chosen plaintext attacks is a standard security notion for public key encryption. It crucially relies on the usage of good randomness and is trivially unachievable if the randomness is known by the adversary. Yilek (CT-RSA'10) defined security against resetting attacks, where randomness might be reused but remains unknown to the adversary. Furthermore, Yilek claimed that security against adversaries making a single query to the challenge oracle implies security against adversaries making multiple queries to the challenge oracle. This is a typical simplification for indistinguishability security notions proven via a standard hybrid argument. The given proof, however, was pointed out to be flawed by Paterson, Schuldt, and Sibborn (PKC'14). Prior to this work, it has been unclear whether this simplification of the security notion also holds in case of resetting attacks.
We remedy this state of affairs as follows. First, we show the strength of resetting attacks by showing that many public key encryption schemes are susceptible to these attacks. As our main contribution, we show that the simplification to adversaries making only one query to the challenge oracle also holds in the light of resetting attacks. More precisely, we show that the existing proof can not be fixed and give a different proof for the claim. Finally, we define real-or-random security against resetting attacks and prove it equivalent to the notion by Yilek which is of the form left-or-right.

Keywords: Public Key Encryption · Resetting Attacks · Provable Security

1 Introduction

Encryption is a fundamental cryptographic primitive to achieve confidentiality between communicating parties. The main distinction is between symmetric key encryption and public key encryption. The former requires the communicating parties to exchange a symmetric key a priori, which the latter does not. An encryption scheme is deemed secure if a ciphertext does not leak information about the underlying plaintext. This is typically modelled as an indistinguishability game in which the adversary has to distinguish between the encryption of two messages of its choice, so-called left-or-right security.

For public key encryption schemes this mandates the usage of probabilistic algorithms for encryption to achieve security. This is in contrast to symmetric key encryption schemes which can use a nonce that has to be unique for each encryption but does not have to be chosen at random [24]. The classical security notion for public key encryption (IND-CPA) implicitly assumes that fresh randomness can be used for every encryption. This is modelled by letting the challenger encrypt using fresh random coins for every query.

The natural question that arises is what happens when this assumption is not true. Clearly, security is elusive if the randomness is known by the adversary which can simply re-encrypt its challenge messages using this randomness. Another scenario, and simultaneously the focus of this paper, is one in which the randomness is unknown to the adversary but might be reused across several encryptions. The practical relevance of this scenario has been shown by Ristenpart and Yilek [23] who demonstrated how TLS can be attacked with reused randomness due to virtual machine snapshots. Based on this, Yilek [28] introduced security against *resetting attacks*. In this setting, the adversary can force the challenger to reuse a randomness across several encryptions. This setting has later been generalised to security against *related randomness attacks* by Paterson et al. [20] and Matsuda and Schuldt [18]. For related randomness attacks, the adversary can specify a function which is applied to the (reused) randomness generated by the challenger and the outcome of the function is the randomness that is used to actually encrypt. All notions introduced in [18, 20, 28] are in the left-or-right style. The adversary queries two messages to its challenge oracle, the oracle encrypts either the left or the right message, and the adversary tries to distinguish these cases.

Security should always hold with respect to adversaries making multiple queries to the challenge oracle. For IND-CPA-like security notions, schemes are often proven secure against adversaries making only one query to the challenge oracle [3, 8, 9, 11, 15]. It is folklore that this implies security in the desired case of multiple queries via a standard hybrid argument. Yilek [28] argues that the same holds also for resetting attacks and provides a proof sketch for the claim. Later, however, Paterson et al. [20] pointed out that the proof is flawed as it results in a prohibited query by the reduction. They further argued that they could neither prove Yilek's claim nor give a separation to disprove it. Thus, prior to this work, it has been unclear whether security against a single challenge query implies security against multiple challenge queries in the light of resetting attacks.

1.1 Our Contribution

In this work we revisit the resetting attack model proposed by Yilek [28]. First, we define a class of public key encryption schemes which we show to be insecure in this model. We then prove several schemes insecure by showing that they lie in the defined class. As our main contribution, we close the aforementioned gap by showing that security against a single query to the challenge oracle indeed implies security against multiple queries to the challenge oracle even against resetting attacks, hereby confirming the claim made in [28]. More precisely, we first investigate the flawed proof in [28] and give an adversary that distinguishes the different hybrid games in the proof almost perfectly. We then nevertheless prove the claim by giving a different proof approach, which only yields an additional factor of 2 compared to the claimed bound in [28]. Finally, we define real-or-random security against resetting attacks and prove the equivalence between the existing left-or-right and our new real-or-random security notion.

1.2 Related Work

Garfinkel and Rosenblum [13] pointed out a theoretical threat to the security of a virtual machine, due to the possibility of snapshots. The practical relevance of this threat has later been shown by Ristenpart and Yilek [23], who demonstrated attacks on TLS. Based on these, Yilek [28] defined security against resetting attack, which models the threat pointed out in [13]. Later, Paterson et al. [20] and Matsuda and Schuldt [18] generalised this to security against related randomness attack.

Bellare et al. [5] gave a public key encryption scheme which still achieves a meaningful security notion, yet qualitatively worse than IND-CPA, even if the randomness is bad. The same setting is also considered by Bellare and Hoang [6]. Huang et al. [14] study nonce-based public key encryption schemes in order to avoid the issue of bad randomness. Closer to our setting is the work by Wang et al. [27], which studies both resetting attacks and bad randomness, yet for authenticated key exchange.

1.3 Organization of the Paper

Section 2 covers the necessary background for this work. In Section 3 we show that several public key encryption schemes are susceptible to resetting attacks. Our main contribution is given in Section 4, where we restore the claim made in [28] by giving a different proof. The definition of real-or-random security against resetting attacks and the proof of its equivalence with the left-or-right security is given in Section 5.

2 Preliminaries

2.1 Notation

For an integer x, the set $\{1, \ldots, x\}$ is denoted by $[x]$. We use game-based proofs [7, 26]. For a game G and an adversary $\mathcal{A}$, we write $\mathsf{G}^{\mathcal{A}} \Rightarrow y$ if the game outputs y when played by $\mathcal{A}$. In our case, the game output will either be true or false, indicating whether the adversary has won the game. Analogously, we write $\mathcal{A}^{\mathsf{G}} \Rightarrow y$ to denote that $\mathcal{A}$ outputs y when playing game G. We only use distinguishing games in which the adversary has to guess a randomly chosen bit b. To scale the advantage of an adversary $\mathcal{A}$ to the interval from 0 to 1, its advantage in a distinguishing game G is defined as

$$\mathbf{Adv}^{\mathsf{G}}(\mathcal{A}) = 2 \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1 \,.$$

Reformulation to adversarial advantage yields

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{G}}(\mathcal{A}) &= 2\ \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1 \\
&= 2\left(\Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \wedge b = 0] + \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \wedge b = 1]\right) - 1 \\
&= 2\left(\Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \,|\, b = 0]\Pr[b = 0]\right. \\
&\qquad \left. + \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \,|\, b = 1]\Pr[b = 1]\right) - 1 \\
&= \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \,|\, b = 0] + \Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow \mathsf{true} \,|\, b = 1] - 1 \\
&= \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 0 \,|\, b = 0] + \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 1 \,|\, b = 1] - 1 \\
&= \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 0 \,|\, b = 0] - \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 0 \,|\, b = 1]\,.
\end{aligned}$$

Analogously, we get

$$\mathbf{Adv}^{\mathsf{G}}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 1 \,|\, b = 1] - \Pr[\mathcal{A}^{\mathsf{G}} \Rightarrow 1 \,|\, b = 0]\,.$$

A public key encryption Σ is a triple of three algorithms `KGen`, `Enc`, and `Dec`, where

- `KGen`: $\mathbb{N} \to \mathcal{SK} \times \mathcal{PK}$ is the key generation algorithm which takes a security parameter[1] as input and outputs a secret key and a public key.
- `Enc`: $\mathcal{PK} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ is the encryption algorithm which maps a public key, a message, and a randomness to a ciphertext.
- `Dec`: $\mathcal{SK} \times \mathcal{C} \to \mathcal{M}$ is the decryption algorithm which outputs a message on input a secret key and a ciphertext.

By $\mathcal{SK}$, $\mathcal{PK}$, $\mathcal{M}$, $\mathcal{C}$, and $\mathcal{R}$ we denote the secret key space, public key space, message space, ciphertext space, and randomness space, respectively.

2.2 Security Against Resetting Attacks

Yilek [28] defines security against resetting attacks, which extends the standard notion of IND-CPA.[2] This models a scenario in which the randomness used to encrypt might be reused, for instance, when performed on a virtual machine using snapshots. The security game is displayed in Fig. 1. The adversary gets access to a challenge left-or-right oracle $\mathsf{LR\text{-}Enc}$ and aims to distinguish which of its messages it encrypts. In addition, the adversary gets an encryption oracle Enc which allows to encrypt using arbitrary, adversarial chosen public keys. The crucial part is that the adversary specifies an index for both oracles to determine which randomness is used to encrypt, i.e., repeating an index results in a repeated randomness. This is also the reason for the additional encryption oracle. In the classical IND-CPA setting, there is no need for this oracle since the adversary can encrypt locally.

[1] We will often omit this input.

[2] In [28] the notion is also extended to the IND-CCA case, which is not relevant for this work.

Game $\mathsf{LR\text{-}RA}$	**oracle** $\mathsf{LR\text{-}Enc}(m_0, m_1, i)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m, i)$
$b \leftarrow_{\$} \{0,1\}$	**if** $f[i] = \bot$	**if** $f[i] = \bot$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_{\$} \mathtt{KGen}()$	$\quad f[i] \leftarrow_{\$} \mathcal{R}$	$\quad f[i] \leftarrow_{\$} \mathcal{R}$
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$r^* \leftarrow f[i]$	$r^* \leftarrow f[i]$
return $(b' = b)$	**if** $b = 0$	$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_0; r^*)$	**return** c
	else	
	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$	
	return c	

Fig. 1: Security game to define LR-RA security using different randomnesses.

Yilek gives two lemmas to simplify the notion. One lemma shows that we can restrict the notion to a single randomness which is used for every encryption query by the adversary. The other lemma, identified as flawed in [20], claims that we can restrict the adversary to a single query to its left-or-right oracle $\mathsf{LR\text{-}Enc}$. Below we recall the former.

Lemma 1 ([28, Lemma 1]). *Let Σ be a PKE scheme and the game* $\mathsf{LR\text{-}RA}$ *be defined as in Fig. 1. Then for any adversary $\mathcal{A}$, making q queries to* $\mathsf{LR\text{-}Enc}$ *and querying in total t different indices to* $\mathsf{LR\text{-}Enc}$ *and* Enc*, there exists an adversary $\mathcal{B}$, making q queries to* $\mathsf{LR\text{-}Enc}$ *and querying only a single index to* $\mathsf{LR\text{-}Enc}$ *and* Enc *such that*

$$\mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) \leq t\, \mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{B})\,.$$

Lemma 1 allows to simplify the security game by choosing one randomness r^* at the beginning of the game which is used for every query by the adversary. The simplified security game $\mathsf{LR\text{-}RA}$ is displayed in Fig. 2.

Game $\mathsf{LR\text{-}RA}$	**oracle** $\mathsf{LR\text{-}Enc}(m_0, m_1)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m)$
$b \leftarrow_{\$} \{0,1\}$	**if** $b = 0$	$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_{\$} \mathtt{KGen}()$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_0; r^*)$	**return** c
$r^* \leftarrow_{\$} \mathcal{R}$	**else**	
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$	
return $(b' = b)$	**return** c	

Fig. 2: Security game to define LR-RA security.

To exclude trivial wins, Yilek [28] defines equality-pattern respecting adversaries. Intuitively, these are adversaries which never repeat a message to their encryption oracles and do not make two challenge queries which are equal in the left message but different in the right message, or vice versa. Below we formally define such adversaries.[3] Note that this definition is necessary to achieve a meaningful security notion, but is not an immoderate restriction imposed on the adversary.

Definition 2. *Let $\mathcal{A}$ be an adversary playing* $\mathsf{LR\text{-}RA}$ *which makes q queries to* $\mathsf{LR\text{-}Enc}$*. Let $\mathcal{E}$ be the set of messages m such that $\mathcal{A}$ makes a query* (pk^*, m) *to* Enc*. Let $(m_0^1, m_1^1), \ldots, (m_0^q, m_1^q)$ be the queries to* $\mathsf{LR\text{-}Enc}$*. We say that $\mathcal{A}$ is* equality-pattern respecting *if*

- *for all $b \in \{0,1\}$ and $i \in [q]$, $m_b^i \notin \mathcal{E}$ and*
- *for all $b \in \{0,1\}$ and $i \neq j$, $m_b^i = m_b^j \implies m_{1-b}^i = m_{1-b}^j$.*

The LR-RA advantage of an adversary is defined as follows.

Definition 3. *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a public key encryption scheme and the game* $\mathsf{LR\text{-}RA}$ *be defined as in Fig. 2. For any equality-pattern respecting adversary $\mathcal{A}$, its* LR-RA *advantage is defined as*

$$\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) := 2\Pr[\mathsf{LR\text{-}RA}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1\,.$$

3 LR-RA-Insecure Public Key Encryption Schemes

To show that the security notion LR-RA is strictly stronger than the classical notion of ciphertext indistinguishability (IND-CPA), Yilek gives a separation example, i.e., a PKE scheme that is IND-CPA-secure but LR-RA-insecure. The scheme follows the standard hybrid encryption idea and combines an arbitrary public key encryption scheme with the one-time pad encryption. The concrete scheme[4] is displayed in Fig. 3. The core observation is that in the resetting attack case, the same one-time key will be used for every query as it is derived from the (reused) randomness. By making one query to the oracle Enc, the adversary learns this one-time key which it can then use to decrypt its challenge query.

This clearly shows that the LR-RA security notion is stronger than the classical IND-CPA security notion. However, the attack does not exploit a weakness in the scheme. It essentially bypasses the security by using the one-time pad in an insecure way, namely, using a key twice. We emphasise that this specific attack no longer works if the one-time pad encryption is replaced with a secret key encryption scheme for which using the same secret key does not affect the security. Furthermore, the idea behind the hybrid encryption scheme is to avoid encrypting a large message using a (costly) public key encryption. Since a key for the one-time pad has the same length as the message, instantiating the hybrid encryption scheme with the one-time pad defeats its main advantage. The given

[3] Note that this definition is tailored to the single randomness setting. The equivalent, more complicated definition for multiple randomnesses (see, e.g., [28, Appendix A]) is irrelevant for this work and therefore omitted.

[4] In [28] the schemes also consists of a MAC to achieve CCA security which we omit here for simplicity.

$\mathtt{KGen}(\lambda)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KGen}^P(\lambda)$	$k, r^* \leftarrow r$
return $(\mathsf{sk}, \mathsf{pk})$	$c_1 \leftarrow \mathtt{Enc}^P(\mathsf{pk}, k; r^*)$
	$c_2 \leftarrow k \oplus m$
	return $c \leftarrow (c_1, c_2)$

Fig. 3: Separation example given in [28]. Algorithms $\mathtt{KGen}^P$ and $\mathtt{Enc}^P$ are the key generation and encryption algorithm of the underlying PKE scheme, respectively.

separation is therefore more of theoretical interest and raises the question how critical resetting attacks are in practice.

In this section, we show that resetting attacks are devastating in practice by showing that many PKE schemes are susceptible to these attacks. To this end, in Section 3.1, we define a class of public key encryption schemes that we call *PK-splittable* and show that such schemes are LR-RA-insecure. We then show, in Section 3.2, that every PKE scheme following the LWE-based scheme by Regev [22], several code-based encryption schemes, and *any* instantiation of the hybrid encryption scheme - i.e., not just the one using the one-time pad -, lie in this class of encryption schemes. Hence all these scheme are insecure against resetting attacks.

3.1 A Class of LR-RA-Insecure PKE Schemes

We define the term *PK-splittable* for public key encryption schemes. Intuitively, these are schemes for which the public key and the ciphertext can be divided into two parts such that: 1. each part of the public key affects exactly one part of the ciphertext and 2. only one part of the ciphertext depends on the message that is encrypted. Below we give the formal definition.

Definition 4. *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a public key encryption scheme, where $\mathcal{PK} = \mathcal{PK}_f \times \mathcal{PK}_g$ with $\mathcal{PK}_g \neq \emptyset$ and $\mathcal{C} = \mathcal{X} \times \mathcal{Y}$. If there exist functions $f\colon \mathcal{PK}_f \times \mathcal{R} \times \mathcal{M} \to \mathcal{X}$ and $g\colon \mathcal{PK}_g \times \mathcal{R} \to \mathcal{Y}$ such that for any public key $\mathsf{pk} = (\mathsf{pk}_f, \mathsf{pk}_g)$ it holds that*

$$\mathtt{Enc}(\mathsf{pk}, m; r) = (f(\mathsf{pk}_f, r, m), g(\mathsf{pk}_g, r))\,,$$

then we say that Σ is a PK-splittable *public key encryption scheme with* core encryption function f *and* auxiliary encryption function g.

From Definition 4 it is easy to see that PK-splittable public key encryption schemes are LR-RA insecure. First, the adversary makes a query to the challenge oracle $\mathsf{LR\text{-}Enc}$ on two randomly chosen messages to obtain a challenge ciphertext. Then it queries both messages to the oracle Enc but on a public key which differs from the challenge public key only in the part affecting the auxiliary encryption function g. To determine the secret bit, the adversary simply compares the output of the core encryption function f for its queries. This is formalised in the following theorem.

Theorem 5. *For any PK-splittable public key encryption scheme Σ, there exists an adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) = 1\,.$$

Proof. We construct the following adversary $\mathcal{A}$ playing LR-RA. Upon receiving the target public key $\mathsf{pk}^* = (\mathsf{pk}_f^*, \mathsf{pk}_g^*)$, $\mathcal{A}$ chooses two messages m_0 and m_1 at random and queries (m_0, m_1) to LR-Enc to obtain a challenge ciphertext c. Subsequently, $\mathcal{A}$ runs `KGen` to obtain a public key $\mathsf{pk}' = (\mathsf{pk}'_f, \mathsf{pk}'_g)$, sets $\mathsf{pk} = (\mathsf{pk}_f^*, \mathsf{pk}'_g)$, and queries both (pk, m_0) and (pk, m_1) to Enc to obtain ciphertexts c_0 and c_1. Let c_0^f, c_1^f, and c^f be the core encryption function parts of the ciphertext c_0, c_1, and c, respectively. If $c^f = c_0^f$, $\mathcal{A}$ outputs 0. If $c^f = c_1^f$, it outputs 1.

Since Σ is a PK-splittable scheme, we have $c_0 = (f(\mathsf{pk}_f^*, r^*, m_0), g(\mathsf{pk}'_g, r^*))$ and $c_1 = (f(\mathsf{pk}_f^*, r^*, m_1), g(\mathsf{pk}'_g, r^*))$. The ciphertext c depends on the secret bit b of the game LR-RA. If $b = 0$, c equals $(f(\mathsf{pk}_f^*, r^*, m_0), g(\mathsf{pk}_g^*, r^*))$ and if $b = 1$, it equals $(f(\mathsf{pk}_f^*, r^*, m_1), g(\mathsf{pk}_g^*, r^*))$. Hence, if $b = 0$, the core encryption function part of the ciphertext c is equal to the core encryption function part of c_0. Likewise, if $b = 1$, the core encryption function parts of c and c_1 are equal. This enables $\mathcal{A}$ to perfectly distinguish the cases $b = 0$ and $b = 1$.

It remains to argue that $\mathcal{A}$ is a valid adversary against LR-RA. Since it queries m_0 and m_1 both to LR-Enc and Enc it looks like $\mathcal{A}$ is not equality-pattern respecting. However, the property equality-pattern respecting only prohibits querying a message to LR-Enc which has been queried to Enc together with the target public key. Our adversary never queries Enc on the target public key since it replaces pk_g^*, i.e., the part that affects the auxiliary encryption function, for both queries. Hence the set $\mathcal{E}$ is empty which yields that $\mathcal{A}$ is an equality-pattern respecting adversary. □

3.2 Real-World PKE Schemes that are LR-RA-Insecure

The backbone of many lattice-based encryption schemes [3,10,12,16,17,19,21,25] is the LWE-based public key encryption scheme due to Regev [22], which is displayed in Fig. 4 (for sake of simplicity we give the scheme in a generic form without specifying concrete sets). It is easy to see that from the two ciphertext parts c_1 and c_2, only c_1 depends on the message. Furthermore, each entry of the public key (a and b) affects exactly one ciphertext part. Thus, this scheme is PK-splittable. A similar argument applies to the code-based PKE schemes HQC [1], RQC[5] [2], and ROLLO-II [4], which are also displayed in Fig. 4 (again in a generic form for sake of simplicity). For all schemes, only c_2 is affected by the message and the public key can be split into a core encryption function and auxiliary encryption function related part (for ROLLO-II there is no core encryption function related part of the public key). This is formalised in the lemma below, the proof is given in Appendix A.1.

[5] RQC is very much akin to HQC, hence we provide the description and the formal proofs only for HQC.

$\mathtt{KGen}(\lambda; \overline{r})$	$\mathtt{KGen}(\lambda; \overline{r})$	$\mathtt{KGen}(\lambda; \overline{r})$
$\mathsf{a}, \mathsf{s}, \mathsf{e} \leftarrow \overline{r}$	$\mathbf{h}, \mathbf{x}, \mathbf{y}, \mathbf{G} \leftarrow \overline{r}$	$\mathbf{x}, \mathbf{y} \leftarrow \overline{r}$
$\mathsf{b} \leftarrow \mathsf{as} + \mathsf{e}$	$\mathbf{s} \leftarrow \mathbf{x} + \mathbf{hy}$	$\mathbf{h} \leftarrow \mathbf{x}^{-1}\mathbf{y}$
$\mathsf{pk} \leftarrow (\mathsf{a}, \mathsf{b})$	$\mathsf{pk} \leftarrow (\mathbf{h}, \mathbf{s}, \mathbf{G})$	$\mathsf{pk} \leftarrow \mathbf{h}$
$\mathsf{sk} \leftarrow \mathsf{s}$	$\mathsf{sk} \leftarrow (\mathbf{x}, \mathbf{y})$	$\mathsf{sk} \leftarrow (\mathbf{x}, \mathbf{y})$
return $(\mathsf{sk}, \mathsf{pk})$	**return** $(\mathsf{sk}, \mathsf{pk})$	**return** $(\mathsf{sk}, \mathsf{pk})$
$\mathtt{Enc}(\mathsf{pk}, m; r)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$
parse pk **as** (a, b)	**parse** pk **as** $(\mathbf{h}, \mathbf{s}, \mathbf{G})$	**parse** pk **as** $\mathbf{h}$
$\mathsf{e}_1, \mathsf{e}_2, \mathsf{d} \leftarrow r$	$\mathbf{r}_1, \mathbf{r}_2, \mathbf{e} \leftarrow r$	$\mathbf{e}_1, \mathbf{e}_2 \leftarrow r$
$c_1 \leftarrow \mathsf{bd} + \mathsf{e}_1 + \mathsf{Encode}(m)$	$c_1 \leftarrow \mathbf{r}_1 + \mathbf{hr}_2$	$c_1 \leftarrow \mathbf{e}_1 + \mathbf{e}_2\mathbf{h}$
$c_2 \leftarrow \mathsf{ad} + \mathsf{e}_2$	$c_2 \leftarrow m\mathbf{G} + \mathbf{sr}_2 + \mathbf{e}$	$c_2 \leftarrow m \oplus \mathsf{Hash}(\mathsf{Supp}(\mathbf{e}_1, \mathbf{e}_2))$
return $c \leftarrow (c_1, c_2)$	**return** $c \leftarrow (c_1, c_2)$	**return** $c \leftarrow (c_1, c_2)$

Fig. 4: LWE-based public key encryption schemes (left) and code-based public key encryption schemes HQC (middle) and ROLLO-II (right).

Lemma 6. *The LWE-based public key encryption scheme and the code-based public key encryption schemes HQC and ROLLO-II (cf. Fig. 4) are PK-splittable.*

Corollary 7. *The LWE-based public key encryption schemes and the code-based public key encryption schemes HQC and ROLLO-II are* LR-RA *insecure. For each scheme there exists an adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) = 1\,.$$

Proof. Follows directly from Theorem 5 and Lemma 6. □

Now we turn our attention towards the security of the hybrid encryption scheme against resetting attacks. As discussed above, the attack proposed in [28] exploits the insecurity of the one-time pad when a key is used more than once. The attack no longer works when using an arbitrary symmetric key encryption scheme instead of the one-time pad, as the adversary does not learn the symmetric key from a single query. However, we show that the hybrid encryption scheme (cf. Fig. 5) is PK-splittable, irrespective of the underlying schemes. This shows that any instantiation is susceptible to resetting attacks.

Lemma 8. *Let* $(\mathtt{KGen}^P, \mathtt{Enc}^P, \mathtt{Dec}^P)$ *be a public key encryption scheme and* $(\mathtt{Enc}^S, \mathtt{Dec}^S)$ *be a symmetric key encryption scheme. The resulting hybrid encryption scheme* $(\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$*, see Fig. 5, is a PK-splittable scheme.*

Proof. The scheme written as a PK-splittable scheme is displayed in Fig. 5. □

Corollary 9. *The hybrid encryption scheme is* LR-RA *insecure. There exists an adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) = 1\,.$$

Proof. Follows directly from Theorem 5 and Lemma 8. □

$\mathtt{KGen}(\lambda)$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KGen}^P(\lambda)$
return $(\mathsf{sk}, \mathsf{pk})$

$\mathtt{Enc}(\mathsf{pk}, m; r)$

$k, r^*, r' \leftarrow r$
$c_1 \leftarrow \mathtt{Enc}^P(\mathsf{pk}, k; r^*)$
$c_2 \leftarrow \mathtt{Enc}^S(k, m; r')$
return $c \leftarrow (c_1, c_2)$

$\mathtt{KGen}(\lambda; \overline{r})$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KGen}^P(\lambda; \overline{r})$
$\mathsf{pk}_g \leftarrow \mathsf{pk}$
$\mathsf{pk}_f \leftarrow \emptyset$
return $(\mathsf{sk}, (\mathsf{pk}_f, \mathsf{pk}_g))$

$\mathtt{Enc}(\mathsf{pk}, m; r)$

parse pk **as** $(\mathsf{pk}_f, \mathsf{pk}_g)$
$c_1 \leftarrow g(\mathsf{pk}_g, r)$
$c_2 \leftarrow f(\mathsf{pk}_f, r, m)$
return $c \leftarrow (c_1, c_2)$

$f(\mathsf{pk}_f, r, m)$

parse pk_f **as** $\emptyset$
$k, r^*, r \leftarrow r$
return $\mathtt{Enc}^S(k, m; r')$

$g(\mathsf{pk}_g, r)$

parse pk_g **as** pk
$k, r^*, r \leftarrow r$
return $\mathtt{Enc}^P(\mathsf{pk}, k; r^*)$

Fig. 5: Left: Hybrid encryption scheme combining a public key encryption scheme $(\mathtt{KGen}^P, \mathtt{Enc}^P, \mathtt{Dec}^P)$ and a symmetric key encryption scheme $(\mathtt{Enc}^S, \mathtt{Dec}^S)$. Right: Hybrid encryption scheme written as a PK-splittable scheme.

4 Left-or-Right Security against Resetting Attacks

In this section, we show that security against adversaries making a single query to the challenge oracle implies security against adversaries making multiple queries to the challenge oracle. This confirms the claim made in [28] by using a different proof that does not suffer from the issue pointed out in [20]. In Section 4.1 we recall the proof given in [28] and its flaw that has been identified in [20]. We construct an adversary which distinguishes the hybrid games almost perfectly, which entails that the existing proof can not be fixed. We then give a different proof for the claim in Section 4.2.

4.1 Shortcomings of Yilek's Proof

We specify two special queries, which are not forbidden by Definition 2. It turns out that these queries are the ones that invalidate the proof in [28]. First, after making a query (m_0, m_1) to $\mathsf{LR\text{-}Enc}$, the adversary can make the same query to $\mathsf{LR\text{-}Enc}$. We call this a *repeating query*. Second, after making a query (m_0, m_1) to $\mathsf{LR\text{-}Enc}$, the adversary can query (m_1, m_0) to $\mathsf{LR\text{-}Enc}$. We call this a *flipping query*.

The proof in [28] uses a sequence of hybrid games $\mathsf{H}_0, \ldots, \mathsf{H}_q$ (cf. Fig. 6). In H_i, the first i queries are answered by encrypting the right message m_1, while the remaining $q - i$ queries are answered by encrypting the left message m_0. By construction, H_0 and H_q equal game $\mathsf{LR\text{-}RA}$ with secret bit $b = 0$ and $b = 1$, respectively. To bound two consecutive hybrids H_{i-1} and H_i, the following reduction $\mathcal{R}_i$ is constructed. Each query by $\mathcal{A}$ to Enc is forwarded by $\mathcal{R}_i$ to its own oracle Enc. The first $i - 1$ challenge queries are answered by querying the left message to Enc, the last $q - i$ challenge queries by querying the right message to Enc, in both cases together with the target public key pk^*. The i-th challenge query is forwarded by $\mathcal{R}_i$ to its own challenge oracle $\mathsf{LR\text{-}Enc}$.

We now elaborate why the reduction does not work if the adversary makes a repeating query or a flipping query.[6] Let (m_0, m_1) be the i-th challenge query by $\mathcal{A}$. Let $j, k > i$ and, wlog, assume that the j-th and k-th query are (m_0, m_1) and (m_1, m_0), respectively. Thus, the j-th query is a repeating query and the k-th query is a flipping query. For the j-th query, the reduction would query its oracle Enc on m_0 and for the k-th query it would query it on m_1. Neither of these two queries is allowed, as both m_0 and m_1 have been queried to $\mathsf{LR\text{-}Enc}$. Thus, this makes the reduction not equality-pattern respecting.

At the first glance, this looks like an issue in the reduction, but we show that the issue lies in the hybrid games. More precisely, we give an equality-pattern respecting adversary that can distinguish two consecutive hybrid games with probability 1. This adversary rules out any proof using these hybrid games, thereby preventing a simple fix of the proof in [28].

Lemma 10. *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a perfectly correct public key encryption scheme and H_i be the hybrid game displayed in Fig. 6. For any $i \in [q]$, there exists an adversary $\mathcal{A}_i$ such that*

$$\Pr[\mathcal{A}_i^{\mathsf{H}_i} \Rightarrow 0] - \Pr[\mathcal{A}_i^{\mathsf{H}_{i-1}} \Rightarrow 0] = 1\,.$$

Game H_i	**oracle** $\mathsf{LR\text{-}Enc}(m_0, m_1)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m)$
$b \leftarrow_{\$} \{0,1\}$	$ctr \leftarrow ctr + 1$	$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
$ctr \leftarrow 0$	**if** $ctr \leq i$	**return** c
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_{\$} \mathtt{KGen}()$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$	
$r^* \leftarrow_{\$} \mathcal{R}$	**else**	
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_0; r^*)$	
	return c	

Fig. 6: Hybrid games in the proof in [28].

Proof. For $i \in [q-1]$, we construct the following adversary $\mathcal{A}_i$. The first $i-1$ and the last $q-i-1$ queries are randomly chosen messages that have never been queried. For the i-th query, $\mathcal{A}_i$ picks two messages m_0 and m_1 at random and queries $\mathsf{LR\text{-}Enc}$ on (m_0, m_1) to obtain a ciphertext c_i. For the $(i+1)$-th query, $\mathcal{A}_i$ invokes $\mathsf{LR\text{-}Enc}$ on the flipping query (m_1, m_0), resulting in a ciphertext c_{i+1}. If $c_i = c_{i+1}$, $\mathcal{A}_i$ outputs 0. Otherwise, it outputs 1.

In game H_{i-1}, both the i-th and the $(i+1)$-th query are answered by encrypting the left message. Since the i-th and $(i+1)$-th queries by $\mathcal{A}_i$ are (m_0, m_1) and (m_1, m_0), this yields $c_i \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_0; r^*)$ and $c_{i+1} \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$.

[6] The issue described in [20] corresponds to the issue for flipping queries we show here.

Then we have $c_i \neq c_{i+1}$ since the scheme is perfectly correct. In game H_i, the i-th query is answered by encrypting the right message instead. This yields $c_i \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$ and $c_{i+1} \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$. Hence we have $c_i = c_{i+1}$.

The adversary $\mathcal{A}_q$ performs $q-2$ challenge queries on random messages, followed by querying first (m_1, m_0) and then (m_0, m_1). The same argument as above allows $\mathcal{A}$ to distinguish with probability 1. □

Remark 11. When considering public key encryption schemes with negligible probability for decryption failures, the distinguishing advantage decreases negligibly. That is because the ciphertexts c_i and c_{i+1} for the message m_0 and m_1 might be equal in H_{i-1}. Nevertheless, two consecutive hybrids can be distinguished almost perfectly.

4.2 Alternative Proof for Yilek's Claim

Having established that the proof approach in [28] does not work, we now turn our attention towards providing a different proof for the statement. Recall that the flawed proof uses a single hybrid argument over the number of queries by the adversary. To deal with the issue of flipping and repeating queries, we change the overall approach as follows. First, instead of a single hybrid argument where we switch from encryption of the left messages to encryption of the right messages, we use two hybrid arguments: one where we first switch from encrypting the left messages to encrypting random messages and one where we switch from encrypting random messages to encrypting the right messages. Second, instead of doing the hybrid argument over the number of queries, we do the hybrid argument over the number of *distinct queries*, i.e., non-repeating queries, by the adversary. The former change avoids the issue of flipping queries while the latter change circumvents the issue of repeating queries.

Below we state our main result. It shows that, in the case of resetting attacks, security against adversaries making a single query to their challenge oracle implies security against adversaries making multiple queries to their challenge oracle. It confirms the claim in the flawed lemma in [28] at the cost of an additional factor of 2 in the bound.

Theorem 12. *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a public key encryption scheme and the* LR-RA *security game be defined as in Fig. 2. Then for any equality-pattern respecting adversary $\mathcal{A}$ making q distinct queries to* $\mathsf{LR\text{-}Enc}$*, there exists an equality-pattern respecting adversary $\mathcal{R}$ making* 1 *query to* $\mathsf{LR\text{-}Enc}$*, such that*

$$\mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) \leq 2q\, \mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{R})\,.$$

Proof. We prove the theorem using hybrid games $\mathsf{L}_0, \ldots, \mathsf{L}_q, \mathsf{R}_0, \ldots, \mathsf{R}_q$ which are displayed in Fig. 7. In game L_i, the first i distinct challenge queries are answered by encrypting a random message while the remaining $q-i$ distinct challenge queries are answered by encrypting the left message. Game R_i is defined analogously except that the right message, rather than the left message, is encrypted. Note that, in any game, repeating queries are answered by looking

Games L_i, R_i

$b \leftarrow_\$ \{0,1\}$
$ctr \leftarrow 0$
$\mathcal{Q} \leftarrow \emptyset$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_\$ \mathtt{KGen}()$
$r^* \leftarrow_\$ \mathcal{R}$
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$

oracle $\mathsf{Enc}(\mathsf{pk}, m)$

$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
return c

oracle $\mathsf{LR\text{-}Enc}(m_0, m_1)$ in L_i

if $\exists c$ s.t. $(m_0, m_1, c) \in \mathcal{Q}$
  return c
$ctr \leftarrow ctr + 1$
if $ctr \leq i$
  $m_* \leftarrow_\$ \mathcal{M}$
  $c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_*; r^*)$
else
  $c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_0; r^*)$
$\mathcal{Q} \leftarrow_\cup (m_0, m_1, c)$
return c

oracle $\mathsf{LR\text{-}Enc}(m_0, m_1)$ in R_i

if $\exists c$ s.t. $(m_0, m_1, c) \in \mathcal{Q}$
  return c
$ctr \leftarrow ctr + 1$
if $ctr \leq i$
  $m_* \leftarrow_\$ \mathcal{M}$
  $c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_*; r^*)$
else
  $c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_1; r^*)$
$\mathcal{Q} \leftarrow_\cup (m_0, m_1, c)$
return c

Fig. 7: Hybrid games L_i and R_i used to prove Theorem 12.

up the previous response in the set $\mathcal{Q}$. From this description we can deduce that hybrid games L_0 and R_0 correspond to the game $\mathsf{LR\text{-}RA}$ with secret bit $b = 0$ and $b = 1$, respectively. Furthermore, hybrid games L_q and R_q are identical, as they both answer all q distinct challenge queries by encrypting a random message. Thus we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) &= \Pr[\mathcal{A}^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \mid b = 0] - \Pr[\mathcal{A}^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \mid b = 1] \\
&= \Pr[\mathcal{A}^{\mathsf{L}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{R}_0} \Rightarrow 0] \\
&= \Pr[\mathcal{A}^{\mathsf{L}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{L}_q} \Rightarrow 0] + \Pr[\mathcal{A}^{\mathsf{R}_q} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{R}_0} \Rightarrow 0] \\
&\leq \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{L}_{i-1}} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{L}_i} \Rightarrow 0]\right) \\
&\qquad + \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{R}_i} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{R}_{i-1}} \Rightarrow 0]\right) .
\end{aligned}$$

To bound the consecutive hybrids L_{i-1} and L_i, we construct the following adversary $\mathcal{B}_i$ playing $\mathsf{LR\text{-}RA}$. On input pk^*, $\mathcal{B}_i$ runs $\mathcal{A}$ on input pk^*. When $\mathcal{A}$ makes a query m to Enc, $\mathcal{B}_i$ forwards m to its own oracle Enc and the response back to $\mathcal{A}$. For the first $i-1$ distinct queries $(m_0^1, m_1^1), \ldots, (m_0^{i-1}, m_1^{i-1})$ by $\mathcal{A}$ to $\mathsf{LR\text{-}Enc}$, $\mathcal{B}_i$ responds by querying its oracle Enc on a random message $m_* \leftarrow_\$ \mathcal{M}$ and the target public key pk^* to obtain a ciphertext that it forwards to $\mathcal{A}$. For the i-th distinct query (m_0^i, m_1^i), $\mathcal{B}_i$ chooses $m_* \leftarrow_\$ \mathcal{M}$, queries (m_0^i, m_*) to its oracle $\mathsf{LR\text{-}Enc}$, and sends the response back to $\mathcal{A}$. For the last $q - i$ distinct queries $(m_0^{i+1}, m_1^{i+1}), \ldots, (m_0^q, m_1^q)$ by $\mathcal{A}$ to $\mathsf{LR\text{-}Enc}$, $\mathcal{B}_i$ queries its oracle Enc $q-i$ times on $m_0^{i+1}, \ldots, m_0^q$ together with the target public key pk^* and forwards the response to $\mathcal{A}$. For all these distinct queries, $\mathcal{B}_i$ stores the queried messages along with the returned ciphertext in a set $\mathcal{Q}$. For any repeating query, $\mathcal{B}_i$ returns the same ciphertext which it looks up in the set $\mathcal{Q}$. When $\mathcal{A}$ halts and outputs a bit b', $\mathcal{B}_i$ outputs b'.

Recall that the proof in [28] does not hold since the reduction has to make a forbidden query. Since our reduction makes only one query to $\mathsf{LR\text{-}Enc}$, we have to ensure that it never queries its oracle Enc on one of the messages queried to $\mathsf{LR\text{-}Enc}$. The critical part is the simulation of the oracle $\mathsf{LR\text{-}Enc}$ for $\mathcal{A}$ using the oracle Enc. By construction, $\mathcal{B}_i$ queries $\mathsf{LR\text{-}Enc}$ on (m_0^i, m_*), where m_0^i is from the i-th query (m_0^i, m_1^i) to $\mathsf{LR\text{-}Enc}$ by $\mathcal{A}$ and m_* is a randomly chosen message by $\mathcal{B}_i$. Prior to this query, $\mathcal{B}_i$ invokes Enc only on random messages, hence the probability that one of these is equal to either m_0^i or m_* is negligible. Subsequent to its challenge query, $\mathcal{B}_i$ queries Enc on the left messages that $\mathcal{A}$ queries to its challenge oracle. We know that the only queries of the form $(m_0^i, \cdot)$ are repeating queries, i.e., (m_0^i, m_1^i). Any query (m_0^i, m') where $m' \neq m_1^i$ is excluded as $\mathcal{A}$ is equality-pattern respecting. Since repeating queries are answered using the set $\mathcal{Q}$, they do not involve an oracle query by $\mathcal{B}_i$. Hence $\mathcal{B}_i$ is a valid adversary, i.e., equality-pattern respecting, playing $\mathsf{LR\text{-}RA}$.

By construction we have that $\mathcal{B}_i$ simulates L_{i-1} and L_i if its own challenge bit b is 0 and 1, respectively. This yields

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{L}_{i-1}} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{L}_i} \Rightarrow 0] &\leq \Pr[\mathcal{B}_i^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 0] - \Pr[\mathcal{B}_i^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 1] \\ &\leq \mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{B}_i)\,.\end{aligned}$$

Analogously, we can construct adversaries $\mathcal{C}_i$ to bound consecutive hybrid games R_{i-1} and R_i with the following two differences. First, for the i-th distinct query (m_0^i, m_1^i) by $\mathcal{A}$, $\mathcal{C}_i$ queries its own challenge oracle $\mathsf{LR\text{-}Enc}$ on (m_*, m_1^i), for a randomly chosen message m_*, and sends the result back to $\mathcal{A}$. For the last $q - i$ distinct queries $(m_0^{i+1}, m_1^{i+1}), \ldots, (m_0^q, m_1^q)$ by $\mathcal{A}$, $\mathcal{C}_i$ invokes its oracle Enc on the right messages, i.e., $m_1^{i+1}, \ldots, m_1^q$, and sends the responses back to $\mathcal{A}$. At the end, $\mathcal{C}_i$ outputs b', where b' is the output of $\mathcal{A}$.

Just as above, $\mathcal{C}_i$ simulates R_i if $b = 0$ and R_{i-1} if $b = 1$, which yields

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{R}_i} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{R}_{i-1}} \Rightarrow 0] &\leq \Pr[\mathcal{C}_i^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 0] - \Pr[\mathcal{C}_i^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 1] \\ &\leq \mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{C}_i)\,.\end{aligned}$$

Let $\mathcal{R}$ be the adversary with the highest advantage among $\mathcal{B}_1, \ldots, \mathcal{B}_q, \mathcal{C}_1, \ldots, \mathcal{C}_q$. Then it holds that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) &\leq \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{L}_{i-1}} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{L}_i} \Rightarrow 0]\right) \\ &\qquad + \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{R}_i} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{R}_{i-1}} \Rightarrow 0]\right) \\ &\leq \sum_{i=1}^{q} \Big(\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{B}_i)\Big) + \sum_{i=1}^{q} \Big(\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{C}_i)\Big) \\ &\leq 2q\,\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{R})\,.\end{aligned}$$

This proves the claim. □

We briefly discuss why the issue of the proof in [28] does not occur here. In [28], the reduction did not work as the adversary can query $\mathsf{LR\text{-}Enc}$ on (m_0, m_1) in the i-th query and later make a flipping query (m_1, m_0). Then the reduction would query (m_0, m_1) to its oracle $\mathsf{LR\text{-}Enc}$ and later m_1 to its oracle Enc which makes the reduction not equality-pattern respecting. The adversary can do the same in our case. However, our reduction invokes its oracle $\mathsf{LR\text{-}Enc}$ on (m_0, m_*), rather than (m_0, m_1), which allows it to later invoke its oracle Enc on m_1. The issue of repeating queries does not occur as the reduction never makes an oracle query when the adversary makes a repeating query.

5 An Equivalent Security Notion

In this section we study security against resetting attacks in a real-or-random sense. In Section 5.1, we first define the corresponding security game and then show that security against adversaries making a single query to the challenge oracle implies security against adversaries making multiple queries to the challenge oracle. In classical security notions for public key encryption, it is known that left-or-right security and real-or-random security are equivalent. In Section 5.2, we show that this equivalence also holds for resetting attacks. As a side effect, these results yield a modular proof for our main result Theorem 12, which we give in Section 5.3.

5.1 Real-or-Random Security against Resetting Attacks

In Fig. 8 we give the real-or-random security game $\mathsf{RR\text{-}RA}$ against resetting attacks. It is easy to see that this notion is unachievable, even when imposing the standard equality-pattern restrictions on the adversary (cf. Definition 2). An adversary simply queries the same message twice to its real-or-random oracle $\mathsf{RR\text{-}Enc}$. In case $b = 0$, it obtains the same ciphertext since the same message is encrypted under the same randomness. In case $b = 1$, however, the ciphertexts will be different (even though they used the same randomness) as two different messages will be encrypted. This allows the adversary to distinguish with overwhelming probability.

Game $\mathsf{RR\text{-}RA}$	**oracle** $\mathsf{RR\text{-}Enc}(m)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m)$
$b \leftarrow_\$ \{0,1\}$	**if** $b = 0$	$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_\$ \mathtt{KGen}()$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m; r^*)$	**return** c
$r^* \leftarrow_\$ \mathcal{R}$	**else**	
$b' \leftarrow \mathcal{A}^{\mathsf{RR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$\quad m_* \leftarrow_\$ \mathcal{M}$	
return $(b' = b)$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_*; r^*)$	
	return c	

Fig. 8: Security game to define RR-RA security.

To circumvent this trivial win, we can use the security game as displayed in Fig. 9. The game keeps a list of messages queried to the real-or-random oracle and ensures that, in case $b = 1$, the same random message is encrypted when the adversary queries the same challenge message. In the game displayed in Fig. 9 this is done via the table f. This prevents the trivial attack described above. However, it also renders repeating queries obsolete as it does not give the adversary any additional information.

Game $\mathsf{RR\text{-}RA}$	**oracle** $\mathsf{RR\text{-}Enc}(m)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m)$
$b \leftarrow\$ \{0,1\}$	**if** $b = 0$	$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow\$ \mathtt{KGen}()$	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m; r^*)$	**return** c
$r^* \leftarrow\$ \mathcal{R}$	**else**	
$b' \leftarrow \mathcal{A}^{\mathsf{RR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$\quad$ **if** $f[m] = \bot$	
return $(b' = b)$	$\qquad f[m] \leftarrow\$ \mathcal{M}$	
	$\quad m_* \leftarrow f[m]$	
	$\quad c \leftarrow \mathtt{Enc}(\mathsf{pk}^*, m_*; r^*)$	
	return c	

Fig. 9: Real-or-random security dealing with repeating queries.

Due to this, we stick with the security game displayed in Fig. 8 and exclude repeating queries via the definition of equality-pattern respecting. Instead of having two different definitions for equality-pattern respecting, depending on the left-or-right and real-or-random case, we use the unified definition below. It extends Definition 2 to also cover real-or-random adversaries.

Definition 13. *Let $\mathcal{A}_{LR}$ and $\mathcal{A}_{RR}$ be adversaries playing* $\mathsf{LR\text{-}RA}$ *and* $\mathsf{RR\text{-}RA}$*, respectively. Let further $\mathcal{E}_{LR}$ (resp. $\mathcal{E}_{RR}$) be the set of messages m such that $\mathcal{A}_{LR}$ (resp. $\mathcal{A}_{RR}$) makes a query (pk^*, m) to* Enc*. Let $(m_0^1, m_1^1), \ldots, (m_0^q, m_1^q)$ be the queries that $\mathcal{A}_{LR}$ makes to* $\mathsf{LR\text{-}Enc}$ *and $m^1, \ldots, m^q$ be the queries of $\mathcal{A}_{RR}$ to* $\mathsf{RR\text{-}Enc}$*.*
We say that $\mathcal{A}_{LR}$ is equality-pattern respecting *if*

- *for all $b \in \{0,1\}$ and $i \in [q]$, $m_b^i \notin \mathcal{E}_{LR}$ and*
- *for all $b \in \{0,1\}$ and $i \neq j$, $m_b^i = m_b^j \implies m_{1-b}^i = m_{1-b}^j$.*

We say that $\mathcal{A}_{RR}$ is equality-pattern respecting *if*

- *for all $i \in [q]$, $m^i \notin \mathcal{E}_{RR}$ and*
- *for $i \neq j$ it holds that $m^i \neq m^j$.*

Just as for the left-or-right case, the real-or-random (RR-RA) advantage of an adversary is defined as its advantage over random guessing scaled to the interval from 0 to 1.

Definition 14. *Let $\Sigma = (\texttt{KGen}, \texttt{Enc}, \texttt{Dec})$ be a public key encryption scheme and the game* $\mathsf{RR\text{-}RA}$ *be defined as in Fig. 8. For any equality-pattern respecting adversary $\mathcal{A}$, its* RR-RA *advantage is defined as*

$$\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{A}) \coloneqq 2\Pr[\mathsf{RR\text{-}RA}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1\,.$$

We now show that real-or-random security against adversaries making a single query to the challenge oracle $\mathsf{RR\text{-}Enc}$ implies real-or-random security against adversaries making multiple queries to the challenge oracle $\mathsf{RR\text{-}Enc}$. It turns out that the standard hybrid technique, which failed in the left-or-right case, works for this setting. This stems from the fact that the real-or-random adversary submits only one message to its challenge oracle. This makes it impossible to make a flipping query which was the main issue in the left-or-right setting.

Theorem 15. *Let Σ be a public key encryption scheme and the game* $\mathsf{RR\text{-}RA}$ *be defined as in Fig. 8. Then for any equality-pattern respecting (cf. Definition 13) adversary $\mathcal{A}$ making q queries to its challenge oracle* $\mathsf{RR\text{-}Enc}$*, there exists an equality-pattern respecting adversary $\mathcal{B}$ making 1 query to* $\mathsf{RR\text{-}Enc}$ *such that*

$$\mathbf{Adv}^{\mathsf{RR\text{-}RA}}_{\Sigma}(\mathcal{A}) \leq q\, \mathbf{Adv}^{\mathsf{RR\text{-}RA}}_{\Sigma}(\mathcal{B})\,.$$

Game H_i	**oracle** $\mathsf{LR\text{-}Enc}(m)$	**oracle** $\mathsf{Enc}(\mathsf{pk}, m)$
$b \leftarrow_{\$} \{0,1\}$	$c \leftarrow c+1$	$c \leftarrow \texttt{Enc}(\mathsf{pk}, m; r^*)$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_{\$} \texttt{KGen}()$	**if** $c \leq i$	**return** c
$r^* \leftarrow_{\$} \mathcal{R}$	$\quad m_* \leftarrow_{\$} \mathcal{M}$	
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$	$\quad c \leftarrow \texttt{Enc}(\mathsf{pk}^*, m_*; r^*)$	
	else	
	$\quad c \leftarrow \texttt{Enc}(\mathsf{pk}^*, m; r^*)$	
	return c	

Fig. 10: Hybrid games used to prove Theorem 15.

Proof. The theorem can be proven using a standard hybrid argument. We use $q+1$ hybrid games $\mathsf{H}_0, \dots, \mathsf{H}_q$ which are displayed in Fig. 10. In hybrid H_i, the first i queries are answered by encrypting a random message, while the remaining $q-i$ queries are answered by encrypting the message provided by the adversary. It holds that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{A}) &\leq 2\Pr[\mathsf{RR\text{-}RA}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1\\ &\leq \Pr[\mathcal{A}^{\mathsf{RR\text{-}RA}} \Rightarrow 0 \mid b=0] - \Pr[\mathcal{A}^{\mathsf{RR\text{-}RA}} \Rightarrow 0 \mid b=1]\\ &\leq \Pr[\mathcal{A}^{\mathsf{H}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_q} \Rightarrow 0]\\ &\leq \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_i} \Rightarrow 0]\right)\,.\end{aligned}$$

We construct the following adversary $\mathcal{B}_i$ to bound the distinguishing advantage between H_{i-1} and H_i. It runs $\mathcal{A}$ on the same public key pk^* it receives as input and answers any query to Enc by $\mathcal{A}$ using its own oracle Enc. For the first $i-1$ challenge queries $m^1, \ldots, m^{i-1}$ by $\mathcal{A}$, $\mathcal{B}_i$ invokes its oracle Enc on randomly chosen messages. For the i-th query m^i, $\mathcal{B}_i$ invokes its own challenge oracle on m^i and sends the obtained ciphertext back to $\mathcal{A}$. For the last $q-i$ queries $m^{i+1}, \ldots, m^q$ by $\mathcal{A}$, $\mathcal{B}_i$ invokes its oracle Enc on $m^{i+1}, \ldots, m^q$.

The adversary $\mathcal{B}_i$ perfectly simulates H_{i-1} and H_i for $\mathcal{A}$ if its own challenge bit is 0 and 1, respectively. To show that $\mathcal{B}_i$ is equality-pattern respecting, we have to show that it never repeats a query to $\mathsf{RR\text{-}Enc}$ and never queries a message to both $\mathsf{RR\text{-}Enc}$ and Enc. The former is trivial as $\mathcal{B}_i$ makes exactly one query to $\mathsf{RR\text{-}Enc}$. For the latter, recall that only challenge queries by $\mathcal{A}$ that are answered using Enc can be problematic (otherwise, $\mathcal{A}$ would not be equality-pattern respecting). Since $\mathcal{A}$ is equality-pattern respecting, all its queries are on fresh messages, which yields that $\mathcal{B}_i$ never queries its challenge message also to Enc. Hence we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{A}) &\leq \sum_{i=1}^{q} \left(\Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_i} \Rightarrow 0]\right) \\ &\leq \sum_{i=1}^{q} \left(\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_i)\right) .\end{aligned}$$

Let $\mathcal{B}$ be the adversary with the highest advantage among $\mathcal{B}_1, \ldots, \mathcal{B}_q$. Then

$$\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{A}) \leq \sum_{i=1}^{q} \left(\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_i)\right) \leq q\, \mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}) ,$$

which proves the claim. □

5.2 Real-or-Random and Left-or-Right Security are Equivalent

In this section we show that our real-or-random security notion against resetting attacks is equivalent to the left-or-right security notion given in [28]. We show the equivalence by proving two lemmas. The former shows that left-or-right security implies real-or-random security while the latter shows that real-or-random security implies left-or-right security. The proofs appear in Appendix A.2 and Appendix A.3, respectively.

Lemma 16. *Let Σ be a public key encryption scheme and the games* $\mathsf{RR\text{-}RA}$ *and* $\mathsf{LR\text{-}RA}$ *be defined as in Fig. 8 and Fig. 2, respectively. Then for any equality-pattern respecting (cf. Definition 13) adversary $\mathcal{A}$ making q queries to its challenge oracle* $\mathsf{RR\text{-}Enc}$*, there exists an equality-pattern respecting (cf. Definition 13) adversary $\mathcal{B}$ making q distinct queries to* $\mathsf{LR\text{-}Enc}$ *such that*

$$\mathbf{Adv}_{\Sigma}^{\mathsf{RR\text{-}RA}}(\mathcal{A}) \leq \mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{B}) .$$

Lemma 17. *Let Σ be a public key encryption scheme and the games* RR-RA *and* LR-RA *be defined as in Fig. 8 and Fig. 2, respectively. Then for any equality-pattern respecting (cf. Definition 13) adversary $\mathcal{A}$ making q distinct queries to its challenge oracle* LR-Enc, *there exists an equality-pattern respecting (cf. Definition 13) adversary $\mathcal{B}$ making q queries to* RR-Enc *such that*

$$\mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) \leq 2\,\mathbf{Adv}_{\Sigma}^{\mathsf{RR\text{-}RA}}(\mathcal{B})\,.$$

5.3 A Modular Proof for Yilek's Claim

Having established the equivalence between left-or-right and real-or-random via Lemma 16 and Lemma 17, we can leverage Theorem 15 to prove Theorem 12 more modular.

Proof (of Theorem 12). From Lemma 17, Theorem 15, and Lemma 16 there exist equality-pattern respecting adversaries $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{R}$, respectively, where

- $\mathcal{B}$ makes q real-or-random queries,
- $\mathcal{C}$ makes 1 real-or-random query, and
- $\mathcal{R}$ makes 1 left-or-right query,

such that

$$\mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) \overset{(17)}{\leq} 2\,\mathbf{Adv}_{\Sigma}^{\mathsf{RR\text{-}RA}}(\mathcal{B}) \overset{(15)}{\leq} 2q\,\mathbf{Adv}_{\Sigma}^{\mathsf{RR\text{-}RA}}(\mathcal{C}) \overset{(16)}{\leq} 2q\,\mathbf{Adv}_{\Sigma}^{\mathsf{LR\text{-}RA}}(\mathcal{R})\,.$$

This proves the claim. □

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

References

1. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, and Gilles Zémor. HQC. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`.
2. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillippe Gaborit, Gilles Zémor, Alain Couvreur, and Adrien Hauteville. RQC. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`.
3. Martin R. Albrecht, Emmanuela Orsini, Kenneth G. Paterson, Guy Peer, and Nigel P. Smart. Tightly secure ring-LWE based key encapsulation with short ciphertexts. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *ESORICS 2017, Part I*, volume 10492 of *LNCS*, pages 29–46. Springer, Heidelberg, September 2017.

4. Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, Gilles Zémor, Carlos Aguilar Melchor, Slim Bettaieb, Loic Bidoux, Magali Bardet, and Ayoub Otmani. ROLLO. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
5. Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Heidelberg, December 2009.
6. Mihir Bellare and Viet Tung Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 627–656. Springer, Heidelberg, April 2015.
7. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.
8. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367. IEEE, 2018.
9. Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. On quantum indistinguishability under chosen plaintext attack. *IACR Cryptol. ePrint Arch.*, 2020:596, 2020.
10. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
11. Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. Quantum indistinguishability for public key encryption. *IACR Cryptol. ePrint Arch.*, 2020:266, 2020.
12. Oscar Garcia-Morchon, Zhenfei Zhang, Sauvik Bhattacharya, Ronald Rietman, Ludo Tolhuizen, Jose-Luis Torre-Arce, Hayo Baan, Markku-Juhani O. Saarinen, Scott Fluhrer, Thijs Laarhoven, and Rachel Player. Round5. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
13. Tal Garfinkel and Mendel Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *Proceedings of HotOS'05: 10th Workshop on Hot Topics in Operating Systems, June 12-15, 2005, Santa Fe, New Mexico, USA*. USENIX Association, 2005.
14. Zhengan Huang, Junzuo Lai, Wenbin Chen, Man Ho Au, Zhen Peng, and Jin Li. Hedged nonce-based public-key encryption: Adaptive security under randomness failures. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 253–279. Springer, Heidelberg, March 2018.
15. Juliane Krämer and Patrick Struck. Encryption schemes using random oracles: From classical to post-quantum security. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 539–558. Springer, Heidelberg, 2020.

16. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.
17. Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, Zhenfei Zhang, Zhe Liu, Hao Yang, Bao Li, and Kunpeng Wang. LAC. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
18. Takahiro Matsuda and Jacob C. N. Schuldt. Related randomness security for public key encryption, revisited. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 280–311. Springer, Heidelberg, March 2018.
19. Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
20. Kenneth G. Paterson, Jacob C. N. Schuldt, and Dale L. Sibborn. Related randomness attacks for public key encryption. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 465–482. Springer, Heidelberg, March 2014.
21. Thomas Poppelmann, Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Peter Schwabe, Douglas Stebila, Martin R. Albrecht, Emmanuela Orsini, Valery Osheter, Kenneth G. Paterson, Guy Peer, and Nigel P. Smart. NewHope. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
22. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
23. Thomas Ristenpart and Scott Yilek. When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography. In *NDSS 2010*. The Internet Society, February / March 2010.
24. Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, February 2004.
25. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
26. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. http://eprint.iacr.org/2004/332.
27. Guomin Yang, Shanshan Duan, Duncan S. Wong, Chik How Tan, and Huaxiong Wang. Authenticated key exchange under bad randomness. In George Danezis, editor, *FC 2011*, volume 7035 of *LNCS*, pages 113–126. Springer, Heidelberg, February / March 2012.
28. Scott Yilek. Resettable public-key encryption: How to encrypt on a virtual machine. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 41–56. Springer, Heidelberg, March 2010.

A Appendix

A.1 Proof of Lemma 6

Proof. The LWE-based public key encryption scheme and the code-based public key encryption schemes HQC and ROLLO-II written as PK-splittable schemes are displayed in Fig. 11. $\square$

LWE-based PKE	HQC	ROLLO-II
$\mathtt{KGen}(\lambda; \overline{r})$	$\mathtt{KGen}(\lambda; \overline{r})$	$\mathtt{KGen}(\lambda; \overline{r})$
$\mathsf{a}, \mathsf{s}, \mathsf{e} \leftarrow \overline{r}$	$\mathbf{h}, \mathbf{x}, \mathbf{y}, \mathbf{G} \leftarrow \overline{r}$	$\mathbf{x}, \mathbf{y} \leftarrow \overline{r}$
$\mathsf{b} \leftarrow \mathsf{as} + \mathsf{e}$	$\mathbf{s} \leftarrow \mathbf{x} + \mathbf{hy}$	$\mathbf{h} \leftarrow \mathbf{x}^{-1}\mathbf{y}$
$\mathsf{pk}_f \leftarrow \mathsf{b}$	$\mathsf{pk}_f \leftarrow (\mathbf{s}, \mathbf{G})$	$\mathsf{pk}_g \leftarrow \mathbf{h}$
$\mathsf{pk}_g \leftarrow \mathsf{a}$	$\mathsf{pk}_g \leftarrow \mathbf{h}$	$\mathsf{pk}_f \leftarrow \emptyset$
$\mathsf{sk} \leftarrow \mathsf{s}$	$\mathsf{sk} \leftarrow (\mathbf{x}, \mathbf{y})$	$\mathsf{sk} \leftarrow (\mathbf{x}, \mathbf{y})$
return $(\mathsf{sk}, (\mathsf{pk}_f, \mathsf{pk}_g))$	**return** $(\mathsf{sk}, (\mathsf{pk}_f, \mathsf{pk}_g))$	**return** $(\mathsf{sk}, (\mathsf{pk}_f, \mathsf{pk}_g))$
$\mathtt{Enc}(\mathsf{pk}, m; r)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$
parse pk **as** $(\mathsf{pk}_f, \mathsf{pk}_g)$	**parse** pk **as** $(\mathsf{pk}_f, \mathsf{pk}_g)$	**parse** pk **as** $(\mathsf{pk}_f, \mathsf{pk}_g)$
$c_1 \leftarrow f(\mathsf{pk}_f, r, m)$	$c_1 \leftarrow g(\mathsf{pk}_g, r)$	$c_1 \leftarrow g(\mathsf{pk}_g, r)$
$c_2 \leftarrow g(\mathsf{pk}_g, r)$	$c_2 \leftarrow f(\mathsf{pk}_f, r, m)$	$c_2 \leftarrow f(\mathsf{pk}_f, r, m)$
return $c \leftarrow (c_1, c_2)$	**return** $c \leftarrow (c_1, c_2)$	**return** $c \leftarrow (c_1, c_2)$
$f(\mathsf{pk}_f, r, m)$	$f(\mathsf{pk}_f, r, m)$	$f(\mathsf{pk}_f, r, m)$
parse pk_f **as** b	**parse** pk_f **as** $(\mathbf{s}, \mathbf{G})$	**parse** pk_f **as** $\emptyset$
$\mathsf{e}_1, \mathsf{e}_2, \mathsf{d} \leftarrow r$	$\mathbf{r}_1, \mathbf{r}_2, \mathbf{e} \leftarrow r$	$\mathbf{e}_1, \mathbf{e}_2 \leftarrow r$
$x \leftarrow \mathsf{bd} + \mathsf{e}_1 + \mathsf{Encode}(m)$	**return** $m\mathbf{G} + \mathbf{sr}_2 + \mathbf{e}$	$E \leftarrow \mathsf{Supp}(\mathbf{e}_1, \mathbf{e}_2)$
return x		**return** $m \oplus \mathsf{Hash}(E)$
$g(\mathsf{pk}_g, r)$	$g(\mathsf{pk}_g, r)$	$g(\mathsf{pk}_g, r)$
parse pk_g **as** a	**parse** pk_g **as** $\mathbf{h}$	**parse** pk_g **as** $\mathbf{h}$
$\mathsf{e}_1, \mathsf{e}_2, \mathsf{d} \leftarrow r$	$\mathbf{r}_1, \mathbf{r}_2, \mathbf{e} \leftarrow r$	$\mathbf{e}_1, \mathbf{e}_2 \leftarrow r$
return $\mathsf{ad} + \mathsf{e}_2$	**return** $\mathbf{r}_1 + \mathbf{hr}_2$	**return** $\mathbf{e}_1 + \mathbf{e}_2\mathbf{h}$

Fig. 11: The LWE-based PKE scheme (left) and the code-based PKE schemes HQC (middle) and ROLLO-II (right) written as PK-splittable schemes.

A.2 Proof of Lemma 16

Proof. The adversary $\mathcal{B}$ runs $\mathcal{A}$ on the same public key pk^* that it receives. The oracle Enc for $\mathcal{A}$ is simulated by $\mathcal{B}$ using its own oracle Enc. When $\mathcal{A}$ makes a challenge query m_i, $\mathcal{B}$ chooses a random message m_* and invokes its own challenge oracle of (m_i, m_*) and sends the response back to $\mathcal{A}$. When $\mathcal{A}$ outputs its guess b', $\mathcal{B}$ outputs b' as its own guess.

It is easy to see that $\mathcal{B}$ perfectly simulates game RR-RA with secret bit b for $\mathcal{A}$, where b coincides with the secret bit that $\mathcal{B}$ is asked to find in game LR-RA. Likewise, $\mathcal{B}$ is equality-pattern respecting. Every challenge query is on two fresh messages since one is the message by $\mathcal{A}$ which never repeats a challenge message and the other one is always sampled at random, i.e., $\mathcal{B}$ makes only distinct queries. Every query to Enc stems from a query to Enc by $\mathcal{A}$. □

A.3 Proof of Lemma 17

Proof. This proof resembles the proof of Theorem 12. We use three hybrid games H_0, H_1, and H_2 which are displayed in Fig. 12. It holds that H_0 corresponds to game LR-RA with secret bit $b = 0$. The same holds for H_2 except that the secret bit b is 1. This yields

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) &\le 2\Pr[\mathsf{LR\text{-}RA}^{\mathcal{A}} \Rightarrow \mathsf{true}] - 1 \\
&\le \Pr[\mathcal{A}^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 0] - \Pr[\mathcal{A}^{\mathsf{LR\text{-}RA}} \Rightarrow 0 \,|\, b = 1] \\
&\le \Pr[\mathcal{A}^{\mathsf{H}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_2} \Rightarrow 0] \\
&\le \Pr[\mathcal{A}^{\mathsf{H}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] + \Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_2} \Rightarrow 0]\,.
\end{aligned}$$

To bound the first difference, we construct the following adversary $\mathcal{B}_1$. It runs $\mathcal{A}$ on the same public key pk^*. Queries by $\mathcal{A}$ to Enc are forwarded by $\mathcal{B}_1$ to its own oracle Enc as are the responses back to $\mathcal{A}$. For every distinct challenge query (m_0^i, m_1^i) by $\mathcal{A}$, $\mathcal{B}_1$ invokes its own challenge oracle RR-Enc on m_0^i and sends the received ciphertext back to $\mathcal{A}$. Every repeating query is answered with the same ciphertext using a set $\mathcal{Q}$. When $\mathcal{A}$ outputs a bit b', $\mathcal{B}_1$ forwards b' as its own output.

Game H_i

$b \leftarrow_\$ \{0,1\}$
$\mathcal{Q} \leftarrow \emptyset$
$(\mathsf{sk}^*, \mathsf{pk}^*) \leftarrow_\$ \mathtt{KGen}()$
$r^* \leftarrow_\$ \mathcal{R}$
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Enc}}(\mathsf{pk}^*)$

oracle $\mathsf{Enc}(\mathsf{pk}, m)$

$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r^*)$
return c

oracle $\mathsf{LR\text{-}Enc}(m_0, m_1)$ in H_1

if $\exists c$ s.t. $(m_0, m_1, c) \in \mathcal{Q}$
  return c
$m_* \leftarrow_\$ \mathcal{M}$
$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m_*; r^*)$
$\mathcal{Q} \leftarrow_\cup (m_0, m_1, c)$
return c

oracle $\mathsf{LR\text{-}Enc}(m_0, m_1)$ in H_0

if $\exists c$ s.t. $(m_0, m_1, c) \in \mathcal{Q}$
  return c
$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m_0; r^*)$
$\mathcal{Q} \leftarrow_\cup (m_0, m_1, c)$
return c

oracle $\mathsf{LR\text{-}Enc}(m_0, m_1)$ in H_2

if $\exists c$ s.t. $(m_0, m_1, c) \in \mathcal{Q}$
  return c
$c \leftarrow \mathtt{Enc}(\mathsf{pk}, m_1; r^*)$
return c

Fig. 12: Hybrid games used to prove Lemma 17.

If the secret bit b in game RR-RA equals 0, $\mathcal{B}_1$ perfectly simulates H_0 for $\mathcal{A}$ as it receives back the encryption of the left message. On the other hand, if $b = 1$, $\mathcal{B}_1$ receives back the encryption of a random message, hence it perfectly simulates H_1. It remains to argue that $\mathcal{B}_1$ is equality-pattern respecting. It clearly does not query Enc on on any message that it queries to $\mathsf{RR\text{-}Enc}$ as this would entail that $\mathcal{A}$ is not equality-pattern respecting. It also never repeats a query to $\mathsf{RR\text{-}Enc}$ since it queries it exactly on the left messages that $\mathcal{A}$ queries to $\mathsf{LR\text{-}Enc}$. Since repeating queries by $\mathcal{A}$ are answered using set $\mathcal{Q}$, the only possibility would be that $\mathcal{A}$ makes two queries (m_0^i, m_1^i) and (m_0^j, m_1^j) with $m_0^i = m_0^j$ and $m_1^i \neq m_1^j$. As an equality-pattern respecting adversary, $\mathcal{A}$ never makes such queries. Thus we have

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{H}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] &\le \Pr[\mathcal{B}_i^{\mathsf{RR\text{-}RA}} \Rightarrow 0 \,|\, b = 0] - \Pr[\mathcal{B}_i^{\mathsf{RR\text{-}RA}} \Rightarrow 0 \,|\, b = 1] \\ &\le \mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_1)\,.\end{aligned}$$

In the same way, we can construct an adversary $\mathcal{B}_2$ to bound the advantage of $\mathcal{A}$ in distinguishing H_1 and H_2. The difference is that $\mathcal{B}_2$ forwards the right message of $\mathcal{A}$ as its own challenge message to $\mathsf{RR\text{-}Enc}$ and when $\mathcal{A}$ outputs a bit b', $\mathcal{B}_2$ outputs $1 - b'$. It holds that $\mathcal{B}_2$ perfectly simulates H_1 and H_2 if its own challenge bit b equals 1 and 0, respectively. Equality-pattern respecting follows by the same argument as above. This yields

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_2} \Rightarrow 0] &\le \Pr[\mathcal{B}_i^{\mathsf{RR\text{-}RA}} \Rightarrow 1 \,|\, b = 1] - \Pr[\mathcal{B}_i^{\mathsf{RR\text{-}RA}} \Rightarrow 1 \,|\, b = 0] \\ &\le \mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_2)\,.\end{aligned}$$

Let $\mathcal{B}$ be the adversary with higher advantage among $\mathcal{B}_1$ and $\mathcal{B}_2$, then we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{LR\text{-}RA}}(\mathcal{A}) &\le \Pr[\mathcal{A}^{\mathsf{H}_0} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] + \Pr[\mathcal{A}^{\mathsf{H}_1} \Rightarrow 0] - \Pr[\mathcal{A}^{\mathsf{H}_2} \Rightarrow 0] \\ &\le \mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B}_2) \\ &\le 2\,\mathbf{Adv}^{\mathsf{RR\text{-}RA}}(\mathcal{B})\,.\end{aligned}$$

This proves the claim. □

Remark 18. The proof is very much akin the one for Theorem 12, where H_0 and H_2 correspond to L_0 and R_0, respectively, while H_1 equals $\mathsf{L}_q = \mathsf{R}_q$.