Related Randomness Attacks for Public Key Cryptosystems

Tsz Hon Yuen
Huawei, Singapore
Yuen.Tsz.Hon@huawei.com

Cong Zhang
Rutgers University, NJ, USA
cz200@cs.rutgers.edu

Sherman S. M. Chow
The Chinese University
of Hong Kong, Hong Kong
sherman@ie.cuhk.edu.hk

Siu Ming Yiu
The University of Hong Kong
Hong Kong
smyiu@cs.hku.hk

ABSTRACT

We initiate the study of *related randomness attack* in the face of a number of practical attacks in public key cryptography, ranges from active attacks like fault-injection, to passive attacks like software (mis)implementation on choosing random numbers. Our new definitions cover the well-known related-key attacks (RKA) where secret keys are related, and a number of new attacks, namely, related encryption randomness attacks, related signing randomness attacks, and related public key attacks. We provide generic constructions for security against these attacks, which are efficiently built upon normal encryption and signature schemes, leveraging RKA-secure pseudorandom function and generator.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems

Keywords

related-key attack; related-randomness attack; public key encryption; identity-based encryption; signatures

1. INTRODUCTION

Generating randomness is crucial to the security of modern cryptosystems, from generating password salts and producing nonce for authentication, to picking randomness for public key cryptosystems, including generations of public key pairs, encryptions, and signatures.

There exist cryptographically secure pseudorandom number generators (PRNG), but researches show that many programmers fail to understand its importance. Recently, Lenstra *et al.* [16] tested the validity of the assumption that different random coins are used to generate each key. They collected a large number of openly accessible public keys and discovered common primes in 0.2% of RSA public-keys. Heninger *et al.* [13] performed a large survey and found that 0.75% of TLS certificates share keys due to insufficient entropy during key generation. They obtained RSA private keys for 0.50% of TLS hosts and 0.03% of SSH hosts, because their public keys shared non-trivial common factors, and DSA private keys for 1.03% of SSH hosts, due to insufficient signature randomness. They also discovered boot-time entropy hole in the Linux PRNG.

For PRNGs themselves, Argyros and Kiayias [1] exploits randomness vulnerabilities in PHP applications, which lacks a built-in cryptographically secure PRNG. Michaelis *et al.* [18] uncovered significant weaknesses of PRNG SecureRandom of several Java runtime libraries including Apache Harmony used in Android, GNU classpath, and Bouncy Castle.

One may argue that security is guaranteed if cryptographically secure PRNG is correctly applied and implemented. However, fault injection [11, 8] can induce modifications in a hardware-stored key, including the key of a pseudorandom function (PRF). The key of the PRF itself may be vulnerable to related-key attack [5]. It is important to consider the security of cryptosystems under such weak randomness.

In this paper, we propose the formal model of *related randomness attack*. Firstly, we consider the randomness used to generate secret keys. This type of attacks is termed as the related-key attack (RKA), and it is well-known for blockciphers, PRF [5], and public key cryptosystems [4]. Secondly, we consider the randomness used to encrypt or sign a message. To the best of the authors' knowledge, related randomness attacks in these two contexts are not previously formalized (except a recent and independent work on public-key encryption [19]). Finally, we consider the related randomness in public key. They appear to be the same as the RKA since the public key is usually a deterministic function of the secret key. However, we show examples that related public keys may induce security problems, without accessing the (related) secret keys. This paper proposes new models to capture and constructions for security against these attacks.

1.1 Related-Key Attacks

Related-key attacks (RKA) are practical against real blockciphers such as AES [9]. Under RKA, an adversary can obtain input-output examples of the blockcipher, not only under the target key K, but also under some related keys. Security against RKA is a popular blockcipher design goal. Bellare *et al.* [4] leverage the RKA security for a suite of high-level primitives in public key cryptosystems. They

ASIA CCS'15, April 14–17, 2015, Singapore, Singapore.
Copyright © 2015 ACM 978-1-4503-3245-3/15/04 ...$15.00.
http://dx.doi.org/10.1145/2714576.2714622.

showed how to construct Φ-RKA secure signatures and Φ-RKA secure public key encryption (PKE) against chosen ciphertext attack from Φ-RKA secure identity-based encryption (IBE).

Recently, Bellare *et al.* [6] proposed a framework to convert normal IBE schemes into Φ-RKA secure IBE schemes, which covers schemes with security in the random oracle model and in the standard model. We make a technical observation that Φ-RKA secure IBE is easy to construct in the random oracle model, where $\Phi = \{\phi^*_\Delta : \Delta \in \mathcal{K}\}$ where $(\mathcal{K}, *)$ is a group under the operation $*$, and $\phi^*_\Delta(K) = K * \Delta$ for all $K \in \mathcal{K}$. This allows a larger class of RKA when compared with affine space, the class of RKA supported by the existing RKA-secure IBE instantiated from Boneh-Franklin IBE [6].

1.2 Related Encryption Randomness Attack

Related key attack is widely studied for different primitives. Also, the problem of fresh randomness is considered in leakage-resilient cryptography. However, there is little concern for the use of related randomness in encryption and signatures. To be more explicit, consider the following example in C++. In most programming textbook, it is suggested to generate a random number by the function *rand()*. The seed of the random number is usually set by the function *srand(time(null))*. So, the seed of two different function calls to *srand* is actually related to the time difference of the function calls. If such randomness is used by the encryptor, then security problems may arise. Even the more complicated seeding algorithm of a number of Java runtime libraries are shown to be error prone [18].

Model. In order to model the related encryption randomness attack on the challenge ciphertext, we have to provide an extra *encryption oracle* which takes a message M, an identity ID and a function ϕ as inputs and outputs $\mathtt{Enc}(\mathsf{mpk}, \mathsf{ID}, M; \phi(R^*))$, where R^* is the randomness used in the challenge ciphertext. Obviously, to rule out simple attack, we do not allow M to be one of the challenge message when ID is the challenge identity and ϕ is the identity function.

We can show that all concrete constructions of RKA-secure IBE proposed by Bellare *et al.* [6] are not secure when this new attack is considered. Take the example of the Boneh-Franklin IBE (with g^α as the master public key), the encryption process using the randomness r is as follows:

$$C_0^* = M^* \cdot \hat{e}(g^\alpha, H(\mathsf{ID}^*))^r, \quad C_1^* = g^r.$$

If the attacker can ask for the encryption of arbitrary message M' for the same ID^* with ϕ being an identity function, he can obtain the same $\hat{e}(g^\alpha, H(\mathsf{ID}^*))^r$ easily and learn M^* encrypted by the challenge ciphertext C^*.

Construction. A simple idea is to use the correlated-input secure hash function (CI-hash) [12] H to generate the randomness used in the original encryption $\mathtt{Enc}$. For example, a new encryption algorithm $\bar{\mathtt{Enc}}$ can be:

$$\bar{\mathtt{Enc}}(\mathsf{mpk}, \mathsf{ID}, M; r) = \mathtt{Enc}(\mathsf{mpk}, \mathsf{ID}, M; H(r||\mathsf{ID}||M)).$$

CI-hash [12] provides *one-wayness*, *unpredictability*, and *pseudorandomness* even if related randomness are used as the hash inputs. Goyal *et al.* showed that fully secure CI-hash can be constructed from RKA-secure PRF [3]. However, the CI-hash is a keyed hash function. The key itself may still suffer from "related public key" attack, since the hash key used in different cryptosystems may be related. CI-hash only considers security of correlated inputs, but not related hash keys. We may need to treat the hash key as some common reference strings and assume that it is honestly generated.

In this paper, instead of using the CI-hash, we use RKA-secure PRF [3] directly. We treat the randomness r as the key of the RKA-secure PRF F and $\mathsf{ID}||M$ as its input[1]. We thus reduce the problem of related encryption randomness ed to the related-key attack of PRF.

1.3 Related Signing Randomness Attack

A signing algorithm can suffer from practical attacks if the signing randomness is chosen from a low entropy source. If a DSA key is used to sign two different messages using the same ephemeral key, then the long-term secret key is immediately computable from the public key and the signatures. Heninger *et al.* [13] recovered DSA private keys for 1.6% of SSH hosts, because of insufficient signature randomness.

Model. The original Φ-RKA security model for signatures allows the adversary to obtain the (related) signature Sign $(\phi(\mathsf{sk}), M)$ for any $\phi \in \Phi$ and any message M, such that M is not equal to the forgery message or ϕ is not the identity function. Our new model allows an extra level of Φ_r-related signing randomness attack, such that the signing oracle outputs $\mathsf{Sign}(\phi(\mathsf{sk}), M; \phi_r(R))$ for any $\phi_r \in \Phi_r$, where R is a fixed string. It allows the attacker to obtain signatures of related signing randomness.

Construction. Our solution is similar to the Φ-RKA-secure signature scheme by Bellare *et al.* [4]. They used Φ-RKA secure pseudorandom number generator (PRG) to generate the randomness of the key generation algorithm, in order to achieve Φ-RKA security for signatures. We also use the same Φ-RKA secure PRG to generate the signing randomness to achieve related signing randomness security at the same time. The advantage of our scheme is that our scheme has minimal modification to the existing signature schemes, and it is easy to update the existing implementation to cope with the related signing randomness attack.

1.4 Related Public Key Attack

One of the motivations of modeling RKA in public key cryptosystems is to protect against fault injection attack. It can be viewed as an *active* attack when the relation injected into the system is chosen by the attacker. On the other hand, it is possible that one may misuse different cryptographic primitives with related secret and public key pairs. For example, Alice uses the key pairs (x, g^x) for her online banking transaction, uses the key pairs $(x+\Delta, g^{x+\Delta})$ for her email communication, uses the key pairs $(x + 2\Delta, g^{x+2\Delta})$, $(x + 3\Delta, g^{x+3\Delta})$ for other applications. Alice may do so voluntarily due to ignorance, or she may not be aware of such relation introduced by the software implementation of choosing key pairs (recall C++ example of the related encryption attack). It can be viewed as a *passive* attack since the attack can be performed by simply observing the relationship between public keys.

Compared to our definition of related public key attack (RPA), Bellare *et al.* [4] also proposed a similar notion of "strong Φ-RKA security", which additionally outputs a related public key during the oracle query of the RKA security

[1] By [12, Theorem 6], it is equivalent to a CI-hash of key $\mathsf{ID}||M$ and input r.

game. However, they claimed that they "*are not aware of this having any application-relevance but wish to highlight it because the constructions possess it*". In this paper, we propose a separation of RPA from RKA by showing a concrete attack on a multiple-recipient encryption scheme [15] with related public keys. This attack does not even require the use of the RKA oracle query.

Model. Modeling the related public key is not trivial. If the relationship between public keys are publicly computable, no oracle is needed and RPA is not properly captured. As a result, the relationship must be computed using the secret key (such that the adversary cannot compute it himself), and yet the computation must be related to the generation of the public key.

We find that the Φ_p-related public key attack model can be applied to *separable* scheme [4], which means that the public key pk is a deterministic function $\mathcal{T}$ of the secret key sk and some public parameters π. In the security model, we additionally allow the adversary to query a key generation oracle with input $\phi_p \in \Phi_p$, to obtain $\mathcal{T}(\pi, \phi_p(\mathsf{sk}))$. This model is applicable to both signatures and encryption schemes.

Construction. Bellare *et al.* proved that a signature scheme can be strengthened to be strong Φ_K-RKA secure by using Φ_K-RKA secure pseudorandom generator (PRG), by assuming either Φ_K is *claw-free*, or the PRG is also secure in the Φ_K*-identity-collision-resistant* model. We want to investigate if one can build a signature scheme from a weaker assumption, if only RPA security is considered. It is useful when the user is confident in defending against active attacks, but still want to avoid the passive RPA attack.

We propose a framework to convert many RPA-insecure signature schemes into one secure in the RPA model. It is simple, efficient, and avoids the claw-free assumption on the leakage function.

Organization. We first review some background knowledge in the next section. Section §3 shows that RKA-secure IBE is easy to construct in the random oracle model. In §4, we define the new security model for IBE for this paper, give a generic construction and its security proof. In §5, we extend the related randomness attack to signatures. In §6, we show the related public key attack and give a solution. Finally, we conclude our paper in §7.

2. BACKGROUND

2.1 Pseudorandom Functions

A pseudorandom function (PRF) family is specified by an efficient probabilistic parameter generation algorithm $\mathsf{Gen}_{\mathsf{prf}}$ which takes as input a security parameter 1^λ and outputs the description of a function π including a description of the function's keyspace $\mathcal{K}$, domain $\mathcal{D}$, and range $\mathcal{R}$, i.e., a PRF function family $F : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ takes a key $k \in \mathcal{K}$ and input $x \in \mathcal{D}$ and returns an output $F(k, x) \in \mathcal{R}$.

The Φ-RKA-security of pseudorandom function [4] is defined by the security game with an adversary $\mathcal{A}$. Firstly, the challenger runs $\pi \leftarrow \mathsf{Gen}_{\mathsf{prf}}$, picks a random key $k \in \mathcal{K}$ and a random bit $b \in \{0, 1\}$. The challenger gives π to the adversary. The adversary $\mathcal{A}$ asks the oracle $RKFn(\phi, x)$ for $\phi \in \Phi$ and $x \in \mathcal{D}$. If $b = 1$, the oracle returns $F(\phi(k), x)$. If $b = 0$, the oracle picks T uniformly at random from $\mathcal{R}$ and returns T. Note that if $(\phi(k), x)$ is the same as a certain oracle query in the past, the oracle always returns the same T. Finally, $\mathcal{A}$ returns a guess bit b' and wins if $b = b'$. The advantage of $\mathcal{A}$ is the probability of $\mathcal{A}$ wins minus $1/2$.

2.2 Pseudorandom Generators

Pseudorandom Generators (PRG) is specified by a parameter generation algorithm $\mathtt{Setup}(1^\lambda)$ which outputs param; a key generation algorithm $\mathcal{K}(\mathsf{param})$ which outputs a key K; an evaluation algorithm $\mathcal{G}(K)$ which outputs T, with $t = |T|$ is publicly known.

The Φ-RKA security of PRG [4] is defined by the following security game with an adversary $\mathcal{A}$. Firstly, the challenger runs $\mathsf{param} \leftarrow \mathtt{Setup}(1^\lambda)$, picks a random key $K \in \mathcal{K}(\mathsf{param})$ and a random bit $b \in \{0, 1\}$. The challenger gives param to the adversary. The adversary $\mathcal{A}$ ask the oracle $RKGen(\phi)$ where $\phi \in \Phi$. If $b = 1$, the oracle returns $\mathcal{G}(\phi(K))$. If $b = 0$, the oracle picks T uniformly random from $\{0, 1\}^t$ and returns T. Note that if the value of $\phi(K)$ is the same as some past oracle queries, the oracle always return the same T. Finally, $\mathcal{A}$ returns a guess bit b' and wins if $b = b'$. The advantage of $\mathcal{A}$ is the probability of $\mathcal{A}$ wins minus $1/2$.

The Φ-identity-collision-resistance (ICR) security of PRG defined by the following security game with an adversary $\mathcal{A}$. Firstly, the challenger runs $\mathsf{param} \leftarrow \mathtt{Setup}(1^\lambda)$, picks a random key $K \in \mathcal{K}(\mathsf{param})$ and calculates $T_0 = \mathcal{G}(K)$. The challenger gives param to the adversary. The adversary $\mathcal{A}$ ask the oracle $RKGen(\phi)$ where $\phi \in \Phi$. The oracle returns $S = \mathcal{G}(\phi(K))$. Finally, $\mathcal{A}$ wins if there exists an oracle query with input ϕ such that $K' = \phi(K) \neq K$ and $S = T_0$. The advantage of $\mathcal{A}$ is the probability of $\mathcal{A}$ wins.

3. RKA-SECURE IBE IN THE ROM

We make the technical observation that Φ-RKA secure identity-based encryption (IBE) is easy to construct in the random oracle model. We observe that Lucks [17] showed that it is easy to strengthen a given PRF to be a Φ-RKA one by hashing the key with a random oracle before use, where $\Phi = \{\phi^*_\Delta : \Delta \in \mathcal{K}\}$, $(\mathcal{K}, *)$ is a group under the operation $*$, and $\phi^*_\Delta(K) = K * \Delta$ for all $K \in \mathcal{K}$. Since Bellare *et al.* [4] showed Φ-RKA secure IBE, CCA-secure PKE and signatures can be constructed from a Φ-RKA secure PRF and an instance of IBE with traditional security, CCA-secure PKE, and signatures respectively. Therefore, we observe that Φ-RKA secure IBE can also be constructed by hashing the master secret key with a random oracle before use.

There are two advantages of the our approach. Firstly, applying the existing transformation [6] on Boneh-Franklin IBE is slightly less efficient since the hash function takes a longer input (hashing the identity and the master public key). Secondly, our method allows Φ to be a large group-induced class when compared with linear relation [6].

4. RELATED ENCRYPTION RANDOMNESS

We showed a related encryption randomness attack on RKA-secure IBE [6] in Section 1 for the same identity. Even if the related randomness is used to encrypt other messages to another identity, it may still help the attacker. Here we show a stronger attack against the Boneh-Boyen IBE [10], with (α, β) as the master secret key. The encryption of a challenge message M^* to the challenge identity ID^*

$$C_0^* = M^* \cdot \hat{e}(g, g)^r, \quad C_1^* = g^{r(\alpha + \mathsf{ID}^*)}, \quad C_2^* = g^{\beta r}.$$

If the attacker can ask for the encryption of arbitrary message M' and identity $\mathsf{ID} \neq \mathsf{ID}^*$, with $\phi(r) = Ar+B$ (an affine space related randomness), he can obtain $\hat{e}(g,g)^r$ easily and win the security game.

4.1 Related Key & Randomness Attack for IBE

An IBE scheme consists of four probabilistic polynomial-time (PPT) algorithms:

1. $\texttt{Setup}$: On input a security parameter 1^λ, it generates a master public key mpk and a master secret key msk.
2. $\texttt{Ext}$: On input msk and an identity ID from an identity space $\mathcal{I}$, it outputs an identity-based secret key $\mathsf{sk_{ID}}$.
3. $\texttt{Enc}$: On input $\mathsf{mpk}, \mathsf{ID}$ and a message M from a message space $\mathcal{M}$, it outputs a ciphertext $\mathfrak{C}$.
4. $\texttt{Dec}$: On input $\mathsf{sk_{ID}}$, and $\mathfrak{C}$, it outputs a message M or $\perp$ symbolizing the failure of decryption.

Correctness. $\forall M \in \mathcal{M}$ and $\forall \mathsf{ID} \in \mathcal{I}$, $M \leftarrow \texttt{Dec}(\mathsf{sk_{ID}}, \texttt{Enc}(\mathsf{mpk}, \mathsf{ID}, M))$, where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \texttt{Setup}(1^\lambda)$, $\mathsf{sk_{ID}} \leftarrow \texttt{Ext}(\mathsf{msk}, \mathsf{ID})$.

We consider the following indistinguishability game against adaptive chosen identity and chosen plaintext attacks (IND-ID-CPA) for semantic security with related key and randomness attack. The game (Φ_k, Φ_r)-$\mathsf{RKRA\text{-}IBE}$ is defined as follows.

1. Setup. The challenger runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \texttt{Setup}(1^\lambda)$ and gives mpk to the adversary $\mathcal{A}$.
2. $\mathsf{Query\ 1}$. The following oracles can be queried by $\mathcal{A}$:
   - Extraction Oracle $\mathcal{KEO}(\mathsf{ID}, \phi)$: On input $\mathsf{ID} \in \mathcal{I}, \phi \in \Phi_k$, it outputs $\texttt{Ext}(\phi(\mathsf{msk}), \mathsf{ID})$.
3. $\mathsf{Challenge}$. $\mathcal{A}$ sends two messages $M_0, M_1 \in \mathcal{M}$ and an identity $\mathsf{ID}^* \in \mathcal{I}$ to the challenger. The challenger picks a random bit b' and computes $\mathfrak{C}^* \leftarrow \texttt{Enc}(\mathsf{mpk}, \mathsf{ID}^*, M_{b'}; R^*)$ using a randomness R^*. The challenger sends $\mathfrak{C}^*$ to $\mathcal{A}$.
4. $\mathsf{Query\ 2}$. The following oracles can be queried by $\mathcal{A}$:
   - Extraction Oracle $\mathcal{KEO}(\mathsf{ID}, \phi)$: On input $\mathsf{ID} \in \mathcal{I}, \phi \in \Phi_k$, it outputs $\texttt{Ext}(\phi(\mathsf{msk}), \mathsf{ID})$.
   - Encryption Oracle $\mathcal{EO}(M, \phi, \mathsf{ID})$: On input a message $M, \phi \in \Phi_r$ and an identity ID, it returns $\texttt{Enc}(\mathsf{mpk}, \mathsf{ID}, M; \phi(R^*))$.
5. Output. $\mathcal{A}$ returns a guess b^* of b'.

$\mathcal{A}$ wins the game if $b' = b^*$, there was no query to $\mathcal{KEO}$ with input $\mathsf{ID} = \mathsf{ID}^*$ and ϕ is an identity map, and no query to $\mathcal{EO}$ with input $M = M_0^*$ or M_1^*, $\mathsf{ID} = \mathsf{ID}^*$ and ϕ is an identity map. The advantage of $\mathcal{A}$ is $\left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right|$.

An IBE scheme is (t,ϵ)-(Φ_k, Φ_r)-RKRA IND-ID-CPA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the (Φ_k, Φ_r)-$\mathsf{RKRA\text{-}IBE}$ game above.

If there is no related randomness attack, then the attacker cannot query the encryption oracle. In this case, we say that an IBE scheme is (t,ϵ)-Φ_k-RKA (related key attack) IND-ID-CPA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the $(\Phi_k, \emptyset)$-$\mathsf{RKRA\text{-}IBE}$ game above.

If there is no related-key attack, then the attacker cannot query the extraction oracle. In this case, we say that an IBE scheme is (t,ϵ)-Φ_r-RRA (related encryption randomness attack) IND-ID-CPA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the $(\emptyset, \Phi_r)$-$\mathsf{RKRA\text{-}IBE}$ game above.

A similar definition for public key encryption (PKE) is given in Appendix A for completeness.

4.2 Generic Construction

We propose the use RKA-secure pseudorandom function (PRF) [3] to handle the encryption randomness. We treat the randomness r as the key of the RKA-secure PRF F and $\mathsf{ID}||M$ as its input. The identity and the message are treated as the input of the PRF, to prevent the attack we have shown against Boneh-Boyen IBE. The problem of related encryption randomness is reduced to the related-key attack of PRF.

Suppose $\Pi = (\texttt{Setup}, \texttt{Ext}, \texttt{Enc}, \texttt{Dec})$ is a IND-ID-CPA secure IBE scheme with security parameter 1^λ, the identity space $\mathcal{I}$, the message space $\mathcal{M}$ and the randomness space used in encryption $\mathcal{R}$. Suppose $F : \mathcal{K} \times (\mathcal{I}||\mathcal{M}) \rightarrow \mathcal{R}$ is a Φ-RKA secure PRF with generator $\mathsf{Gen_{prf}}$. Define

- $\widehat{\texttt{Setup}}(1^\lambda)$: It runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \texttt{Setup}(1^\lambda)$ and runs $\mathsf{Gen_{prf}}(1^\lambda)$ to get $F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$, where $\mathcal{D}$ is the identity space $\mathcal{I}$ concatenates with the message space $\mathcal{M}$. It outputs $\widehat{\mathsf{mpk}} = (\mathsf{mpk}, F)$ and msk.
- $\widehat{\texttt{Enc}}(\mathsf{mpk}, \mathsf{ID}, M)$: It picks a random key $r \in \mathcal{K}$ and returns $\texttt{Enc}(\mathsf{mpk}, \mathsf{ID}, M;\ F(r, \mathsf{ID}||M))$.

Then we obtain an RRA-secure IBE scheme $\hat{\Pi} = (\widehat{\texttt{Setup}}, \texttt{Ext}, \widehat{\texttt{Enc}}, \texttt{Dec})$.

Theorem 1. *The IBE scheme $\hat{\Pi}$ is Φ-RRA IND-ID-CPA secure if Π is IND-ID-CPA secure and $\mathcal{F}$ is a Φ-RKA secure PRF.*

Proof. We can prove the theorem by hybrid security games. The main part of the proof is to show statistical indistinguishability between these games. The output of each games consists of the view of the adversary $\mathcal{A}$ as well as a bit b chosen by the challenger $\mathcal{C}$ representing its choice of which message M_0^*, M_1^* to encrypt. Denote $\mathsf{Game\ 0}$ to be the original Φ-RRA security game. Denote $\mathsf{Game\ 1}$ to be the same as the $\mathsf{Game\ 0}$, except that for the encryption oracle query and the challenge phase, it outputs $\widehat{\texttt{Enc}}(\mathsf{mpk}, \mathsf{ID}, M; r')$ where r' is randomly chosen from $\mathcal{R}$.

Lemma 1. *No PPT adversary can distinguish* Game *1 and* Game *0 if $\mathcal{F}$ is a Φ-RKA secure PRF.*

Proof. Suppose the simulator $\mathcal{B}$ is given a Φ-RKA PRF family F from its challenger $\mathcal{C}$. For the encryption oracle query with input (ID, M, ϕ):

- It asks its challenger $\mathcal{C}$ for the oracle $RKFn(\phi, \mathsf{ID}||M)$ and obtains $\bar{r}$. It outputs $\widehat{\texttt{Enc}}(\mathsf{mpk}, \mathsf{ID}, M; \bar{r})$.

For the challenge phase, $\mathcal{B}$ picks a random bit b, sets ϕ as an identity map and asks its challenger $\mathcal{C}$ for the oracle $RKFn(\phi, \mathsf{ID}^*||M_b^*)$. $\mathcal{B}$ obtains r^* and outputs $C^* \leftarrow \widehat{\texttt{Enc}}(\mathsf{mpk}, \mathsf{ID}^*, M_b^*;\ r^*)$.

It is easy to see that if the challenger $\mathcal{C}$'s choice of $b=1$, then $\mathcal{B}$ simulates Game 0; if the challenger $\mathcal{C}$'s choice of $b=0$, then $\mathcal{B}$ simulates Game 1. Therefore Game 1 is indistinguishable from Game 0 if F is a Φ-RKA secure PRF. □

Lemma 2. *The advantage of $\mathcal{A}$ in* $\mathsf{Game\ 1}$ *is negligible if* Π *is IND-ID-CPA secure.*

Proof. Observe that in $\mathsf{Game\ 1}$, all oracle outputs are not related to r^*. Therefore, the encryption oracle does not help the attacker to win the game. The rest of the simulation is the same as the underlying IBE scheme. Thus, the advantage of $\mathcal{A}$ in $\mathsf{Game\ 1}$ is negligible if Π is IND-ID-CPA secure. □

Combining the above lemmas, the IBE scheme $\hat{\Pi}$ is Φ-RRA IND-ID-CPA secure. □

Extension. Our result can be applied to different areas. Similarly, we can prove the following lemma for RKRA-secure IBE.

Lemma 3. *The IBE scheme $\hat{\Pi}$ is (Φ_k, Φ_r)-RKRA IND-ID-CPA secure if Π is Φ_k-RKA IND-ID-CPA secure and $\mathcal{F}$ is a Φ_r-RKA secure PRF.*

It can also been easily shown that the theorem still holds for CCA security.

5. RELATED SIGNING RANDOMNESS

For the case of signatures, both the secret key and the singing randomness are chosen by the signer. The adversary only needs to obtain leakage from a single party. For some signature schemes, anyone can re-randomize a signature σ for a message M into another signature σ' for M, and hence it is legitimate to generate a signature with related randomness for the same message. On the other hand, it is not possible for many signature schemes, such as signatures from the Fiat-Shamir transformation.

For the related signing randomness attack, we should avoid the attacker to generate a new signature for *different* messages, even if we know certain ways to obtain related randomness. We first consider the existential unforgeability against related key and related (signing) randomness attack, such that the attacker cannot re-randomize a signature for a different message.

5.1 Related Randomness Attack for Signatures

A signature scheme consists of four PPT algorithms:

1. $\texttt{Setup}$: On input a security parameter 1^λ, it generates a system parameter param.
2. $\texttt{Gen}$: On input param, it generates a public key pk and a secret key sk.
3. $\texttt{Sign}$: On input $\mathsf{param}, \mathsf{sk}$ and a message M from a message space $\mathcal{M}$, it outputs a signature σ.
4. $\texttt{Verify}$: On input $\mathsf{param}, \mathsf{pk}$, M and σ, it outputs 1 for valid signature or 0 otherwise.

Correctness. $\forall M \in \mathcal{M}$, $1 \leftarrow \texttt{Verify}(\mathsf{param}, \mathsf{pk}, M, \texttt{Sign}(\mathsf{param}, \mathsf{sk}, M))$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param})$, $\mathsf{param} \leftarrow \texttt{Setup}(1^\lambda)$.

We consider the following unforgeability game against chosen message attacks (EUF-CMA) for security with related-key, related signing randomness attack. The game (Φ_k, Φ_r)-$\mathsf{RKRA\text{-}Sig}$ is defined as follows.

1. Setup. The challenger runs $\mathsf{param} \leftarrow \texttt{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param})$ and gives $\mathsf{param}, \mathsf{pk}$ to the adversary $\mathcal{A}$. The challenger samples the signing randomness r.
2. $\mathsf{Query\ 1}$. The following oracles can be queried by $\mathcal{A}$:
   - Signing Oracle $\mathcal{SO}(M, \phi_k, \phi_r)$: On input a message M, a key leakage $\phi_k \in \Phi_k$ and a randomness leakage $\phi_r \in \Phi_r$, it returns a signature $\sigma \leftarrow \texttt{Sign}(\mathsf{param}, \phi_k(\mathsf{sk}), M; \phi_r(r))$.
3. Output. $\mathcal{A}$ returns a signature σ^* on a message M^*.

$\mathcal{A}$ wins the game if $1 \leftarrow \texttt{Verify}(\mathsf{param}, \mathsf{pk}, M^*, \sigma^*)$, there was no query to $\mathcal{SO}$ with input message M^* and $\phi_k(\mathsf{sk}) = \mathsf{sk}$. The advantage of $\mathcal{A}$ is $\Pr[\mathcal{A}$ wins$]$.

A signature scheme is (t, ϵ)-(Φ_k, Φ_r)-RKRA EUF-CMA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the (Φ_k, Φ_r)-$\mathsf{RKRA\text{-}Sig}$ game above.

5.2 Generic Construction

We give a generic construction of RKRA EUF-CMA secure signature scheme from RKA-secure pseudorandom generator (PRG). The construction is similar to the RKA EUF-CMA secure signature scheme by Bellare *et al.* [4]. Suppose $(\texttt{Setup}, \texttt{Gen}, \texttt{Sign}, \texttt{Verify})$ be a EUF-CMA secure signature scheme, and $(\texttt{Setup}', \mathcal{K}, \mathcal{G})$ be an RKA-secure PRG. We give our new construction as follows.

- $\hat{\texttt{Setup}}(1^\lambda)$: It runs $\mathsf{param} \leftarrow \texttt{Setup}(1^\lambda)$ and $\mathsf{param}' \leftarrow \texttt{Setup}'(1^\lambda)$, and outputs $\hat{\mathsf{param}} = \mathsf{param}||\mathsf{param}'$.
- $\hat{\texttt{Gen}}(\hat{\mathsf{param}})$: It samples a random $K \leftarrow \mathcal{K}(\mathsf{param}')$ and runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param}; \mathcal{G}(K))$. It outputs $\hat{\mathsf{pk}} = \mathsf{pk}$ and $\hat{\mathsf{sk}} = K$.
- $\hat{\texttt{Sign}}(\hat{\mathsf{param}}, K, M)$: It runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param}; \mathcal{G}(K))$ and samples a random $R \leftarrow \mathcal{K}(\mathsf{param}')$. It outputs $\sigma = \hat{\texttt{Sign}}(\mathsf{param}, \mathsf{sk}, M; \mathcal{G}(R))$.
- $\hat{\texttt{Verify}}(\hat{\mathsf{param}}, \mathsf{pk}, M, \sigma)$: It outputs the result of $\texttt{Verify}$ $(\mathsf{param}, \mathsf{pk}, M, \sigma)$.

Theorem 2. *The above signature scheme is $(\Phi_k \cup \Phi_c, \Phi_r)$-RKRA EUF-CMA secure, if the underlying signature is EUF-CMA secure, the underlying PRG is $(\Phi_r \cup \Phi_k)$-RKA secure and Φ_k-ICR secure; and Φ_c is the class of constant related-key deriving functions associated to Φ_k.*

Proof. We prove the security of our scheme by a sequence of security games:

- Game_0 is the original $(\Phi_k \cup \Phi_c, \Phi_r)$-RKRA EUF-CMA security game.
- Game_1 is the same as Game_0, except that for each signing oracle query, the simulator runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param}; \mathcal{G}(K))$ and picks R' uniformly random from the range of $\mathcal{G}$. It returns $\sigma = \hat{\texttt{Sign}}(\mathsf{param}, \mathsf{sk}, M; R')$.

We have to show that the advantage of $\mathcal{A}$ in Game_1 and Game_0 differs by ϵ, where ϵ is the probability of breaking the Φ_r-RKA security of the underlying PRG. Suppose the simulator $\mathcal{B}$ is given param' from the challenger of the Φ_r-RKA security game. It runs $\mathsf{param} \leftarrow \texttt{Setup}(1^\lambda)$, samples a random $K \leftarrow \mathcal{K}(\mathsf{param}')$ and runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{Gen}(\mathsf{param}; \mathcal{G}(K))$. It sets $\hat{\mathsf{sk}} = K$, and outputs ($\hat{\mathsf{param}} = \mathsf{param}||\mathsf{param}'$, $\hat{\mathsf{pk}} = \mathsf{pk}$) to the adversary $\mathcal{A}$.

For the signing oracle queries with input (M, ϕ_k, ϕ_r), $\mathcal{B}$ runs $(\mathsf{pk}', \mathsf{sk}') \leftarrow \texttt{Gen}(\mathsf{param}; \mathcal{G}(\phi_k(K)))$ and queries the $\mathsf{GenOracle}(\phi_r)$ to its challenger. The challenger returns a value T which is either equal to $\mathcal{G}(\phi_r(R))$ or a uniformly random number from the range of $\mathcal{G}$. Then $\mathcal{B}$ outputs $\sigma = \hat{\texttt{Sign}}(\mathsf{param}, \mathsf{sk}', M; T)$ and sends to $\mathcal{A}$.

Therefore if the adversary can distinguish between Game_0 and Game_1, then $\mathcal{B}$ can answer $T = \phi_r(R)$ or T is a random number, which breaks the Φ_r-RKA security of the underlying PRG.

Finally, observe that Game_1 is the same as the $(\Phi_k \cup \Phi_c)$-RKA EUF-CMA security game. The adversary cannot win with a non-negligible probability if the underlying signature is EUF-CMA secure, the underlying PRG is Φ_k-RKA secure and Φ_k-ICR secure, where Φ_c is the class of constant related-key deriving functions associated to Φ_k. □

6. RELATED PUBLIC KEY ATTACK

The related public key attack (RPA) seems to be a natural extension of RKA. In this section, we will propose a security model to capture the RPA. Bellare *et al.* [4] considered a similar model with combined RPA with RKA, but they did not show any relevant attack. We will demonstrate two concrete attacks on encryption and signature schemes with related public keys. In fact, our attacks do not require the oracles provided by the RKA, and hence it shows the separation between the RKA and the RPA models.

6.1 Examples of Related Public Key Attack

We demonstrate two practical related public key attacks, one for encryption and one for signatures.

Multi-Recipient Encryption. Bellare *et al.* [2] showed that randomness re-use is secure for *multi-recipient encryption* schemes. It can also be viewed as a special case of *related encryption randomness* security, where the relation is an identity map. Their security model rules out the *rouge-key* attack, where an adversary registers public keys created as a function of other public keys. They assume that the adversary cannot register a public key without knowing the corresponding secret key. However, if the public keys of two or more recipients are related (e.g. Bob encrypts an email to multiple email accounts of Alice using different public keys, but the public keys are actually related), the rouge-key attack also succeeds. It is because such relation is generated by an legitimate user knowing the secret key. One example of rouge-key attack on ElGamal-based scheme [15] was presented [2, Section 4]. We demonstrate it in the context of related public key attack as follows.

Scheme. Suppose a sender wants to send message M_i to receiver i encrypted under the latter's ElGamal public key g^{x_i}, for $i \in [1, n]$. Kurosawa [15] considered picking just one r at random, setting $C_i = (g^r, g^{x_i r} \cdot M_i)$ and send C_i to receiver i for all $i \in [1, n]$. This scheme is known as the ElGamal-based multi-recipient encryption.

Attack. Now suppose the public keys of some receivers are related, e.g. g^x, g^{2x}, g^{3x}. Thus the adversary sees the three corresponding ciphertexts $(g^r, g^{xr} \cdot M_1), (g^r, g^{2xr} \cdot M_2), (g^r, g^{3xr} \cdot M_3)$. From them it can compute $[g^{rx} \cdot M_1] \cdot [g^{2rx} \cdot M_2] \cdot [g^{3rx} \cdot M_3]^{-1} = M_1 M_2 / M_3$. If two plaintexts are known, the third one can be calculated easily without using any related secret key during the calculation. This scheme is not secure in this sense if there is related public key.

Ring Signatures. Related public key attack also applies to the *ring signatures*. We can show that if the attacker obtains a ring signature with respect to $(pk_1, pk_2, \ldots, pk_n)$, the attacker can convert it into a signature with respect to $(\phi(pk_1), pk_2, \ldots, pk_n)$, where $\phi(pk_1)$ is a related public key.

Scheme. Consider the ring signature scheme [14] using the classic Schnorr signatures. For simplicity, we consider the two users case. Denote g to be a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q and $H : \{0,1\}^* \to \mathbb{Z}_q$ is a collision resistant hash function. Suppose Alice and Bob have key pairs $(x_1, y_1 = g^{x_1})$ and $(x_2, y_2 = g^{x_2})$ respectively. To sign a message M, Alice randomly picks $r, r_1 \in \mathbb{Z}_q^*$ and computes

$$R_2 = g^{r_2}, \quad R_1 = g^r y_2^{-H(M,R_2)}, \quad \sigma = r + r_2 + x_1 H(M, R_1).$$

The ring signature is $(R_1, R_2, h_1, h_2, \sigma)$, where $h_i = H(M, R_i)$ for $i = 1, 2$. The verification algorithm is to check if h_1, h_2 are correct hash outputs, and if

$$R_1 R_2 y_1^{h_1} y_2^{h_2} = g^{\sigma}.$$

Attack. Suppose the attacker issues a typical query for the signing oracle of x_1 to sign M with public keys (y_1, y_2) and receives $(R_1, R_2, h_1, h_2, \sigma)$. The attacker then picks Δ and asks the $\mathsf{KeyGen\ Oracle}$ to obtain y_3, which is a public key related to y_2 via the relation $y_3 = g^{x_2 + \Delta}$. Then the attacker can calculate

$$\sigma' = \sigma - \Delta h_2.$$

$(R_1, R_2, h_1, h_2, \sigma')$ is a valid signature for the message M and public keys (y_1, y_3) since

$$R_1 R_2 y_1^{h_1} y_3^{h_2} = R_1 R_2 y_1^{h_1} (y_2 g^{\Delta})^{h_2} = g^{\sigma + h_2 \Delta} = g^{\sigma'}.$$

A similar attack has been shown to illustrate the difference between fixed-ring attack and chosen-subring attack in ring signatures [7]. Our related public key is a special kind of *adversarially-chosen key* in the chosen-subring attack, yet the adversary does not know the corresponding secret key in our model.

6.2 Modeling Related Public Key Attack

We propose the security model of related public key attack (RPA). In the usual syntax of public key cryptosystems, the public and secret keys are generated together by the algorithm $\mathsf{KeyGen}(1^\lambda)$. In most practical schemes, the secret key is usually chosen first and the public key is a deterministic function of the secret key and some public parameters. A scheme is called *separable* [4] if there exists algorithms:

- $\mathsf{Setup}(1^\lambda)$: It is a probabilistic algorithm which outputs the public parameters π.
- $\mathsf{KeyGen}(\pi)$: It is a probabilistic algorithm which outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, where $\mathsf{pk} = \mathcal{T}(\pi, \mathsf{sk})$ and $\mathcal{T}$ is a deterministic algorithm.

We only consider separable schemes in the rest of this section.

In the security game with Φ_P-RPA security for separable schemes, we additionally allow the adversary to query a $\mathsf{KeyGen\ Oracle}$:

- $\mathsf{KeyGen\ Oracle}(\phi)$: On input a function $\phi \in \Phi_P$, it returns $\hat{\mathsf{pk}} = \mathcal{T}(\pi, \phi(\mathsf{sk}))$.

The adversary can win the game not only by attacking on the (challenge) public key pk, but also on any $\hat{\mathsf{pk}}$ returned by the $\mathsf{KeyGen\ Oracle}$.

Comparison between RPA and Strong Φ_K-RKA. Bellare *et al.* [4] proposed a similar notion of strong Φ_K-RKA security for signatures and PKE. In their security model, there is only a signing oracle with input a message M and $\phi \in \Phi_K$. The oracle outputs $\sigma \leftarrow \mathsf{Sign}(\phi(\mathsf{sk}), M)$ and $\hat{\mathsf{pk}} = \mathcal{T}(\pi, \phi(\mathsf{sk}))$. The adversary wins the game if the forgery signature is valid for any $(M, \hat{\mathsf{pk}})$ used by the oracle. Conceptually, they embed our $\mathsf{KeyGen\ Oracle}$ into the signing oracle for RKA security.

Bellare *et al.* [4] also proposed analogous security definitions for CCA-secure PKE. The decryption oracle outputs a related public key $\hat{\mathsf{pk}}$ and the decryption under the related secret key $\phi(\mathsf{sk})$. In the challenge phase, the adversary can ask for the challenge ciphertext encrypted for some $\hat{\mathsf{pk}}$.

We note that our model is more general than the strong Φ_K-RKA security [4]. It is because the attacker may obtain related public keys (which is a passive attack by just observing available public keys) but cannot launch related (secret) key attack (which is an active attack, such as fault injection). It is not necessary to consider both attacks together. Therefore, our model provides a separation for different attacks and provides better classification.

Consider the attack on multi-recipient encryption schemes in §6.1. The successful attacker only requires the formation of related public keys and the knowledge of some plaintexts. The former is captured by our $\mathsf{KeyGen\ Oracle}$, while the latter is the challenge message chosen by the attacker. Therefore the attack in §6.1 is captured in our model, but not by the strong Φ_K-RKA security [4].

6.3 Generic Construction

Let $\Pi' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}')$ be a signature scheme which is not RPA secure in the sense that, given a signature σ on a message m for parameter π and public key pk, and a group induced operation $\phi \in \Phi$, there exists a polynomial time algorithm to find $\bar{\sigma}$ such that

$$\mathsf{Verify}'(\pi, \mathsf{pk}, \sigma, M) = \mathsf{Verify}'(\pi, \mathcal{T}(\pi, \phi(\mathsf{sk})), \bar{\sigma}, M).$$

We call such scheme Φ-*RPA fix-message insecure.*

We find that we can achieve RPA security for Φ-RPA fix-message insecure signatures if the underlying scheme has a property called Φ-*public key malleability.* It means that for a separable scheme with parameter π, public key pk, secret key sk, and $\phi \in \Phi$, there exists a polynomial time algorithm to find $\bar{\mathsf{pk}}$ such that

$$\bar{\mathsf{pk}} = \mathcal{T}(\pi, \phi(\mathsf{sk})).$$

Many existing key systems have the public key malleability property. For example, consider the discrete logarithm based system with $\mathsf{sk} = x$ and $\mathsf{pk} = g^x$, public key malleability holds for affine space transformation for Φ. Let $\phi(x) = Ax + B$ for some $A, B \in \mathbb{Z}$. Then it is easy to compute g^{Ax+B}.

We start with the Φ-RPA fix-message insecure secure signature scheme $\Pi' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}')$ with Φ-public key malleability. We now build another signature scheme Π as follows:

- Setup: On input the security parameter 1^λ, it outputs $\pi \leftarrow \mathsf{Setup}'(1^\lambda)$.
- KeyGen: On input param, it outputs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}'(\pi)$.
- Sign: On input a message M, sk and π, returns $\sigma \leftarrow \mathsf{Sign}'(\pi, \mathsf{sk}, M||\mathsf{pk})$.
- Verify: On input a signature σ, a message M, π and pk, it outputs $\mathsf{Verify}(\pi, \mathsf{pk}, \sigma, M||\mathsf{pk})$.

THEOREM 3. *If Π' is a EUF-CMA secure separable signature scheme with Φ_A-public key malleability but Φ_B-RPA fix-message insecure, then Π is a Φ_P-RPA, EUF-CMA secure signature scheme, where $\Phi_P = \Phi_A \cap \Phi_B$ is some class of group induced operations.*

PROOF. Suppose the simulator $\mathcal{B}$ is given π, pk from the challenger $\mathcal{C}$ of Π'. $\mathcal{B}$ forwards π, pk to the adversary $\mathcal{A}$.

For the $\mathsf{Signing\ Oracle}$ query with input M, $\mathcal{B}$ asks $\mathcal{C}$ to answer the signature on the message $M||\mathsf{pk}$. For the $\mathsf{KeyGen\ Oracle}$ query with input $\phi \in \Phi_P$, $\mathcal{B}$ answers $\bar{\mathsf{pk}} = \mathcal{T}(\pi, \phi(\mathsf{sk}))$ by the Φ_A-public key malleability property.

Finally, $\mathcal{A}$ returns a forgery σ^* on a message M^* with respect to the public key $\bar{\mathsf{pk}}$ such that $\mathsf{Verify}(\pi, \bar{\mathsf{pk}}, \sigma^*, M^*||\bar{\mathsf{pk}}) = 1$. If $\bar{\mathsf{pk}} = \mathsf{pk}$, then M^* must not be asked in the signing oracle. Then $\mathcal{B}$ returns σ^* to $\mathcal{C}$ as the signature to the message $M^*||\mathsf{pk}$. Else, $\bar{\mathsf{pk}}$ is the output of the $\mathsf{KeyGen\ Oracle}$ for some input ϕ. Then $\mathcal{B}$ transforms σ^* to $\hat{\sigma}$ such that $\mathsf{Verify}(\pi, \mathsf{pk}, \hat{\sigma}, M^*||\bar{\mathsf{pk}})$ by the Φ_B-RPA fix-message insecurity. This transformation is possible since ϕ^{-1} exists for group induced operations in Φ_P. $\mathcal{B}$ returns $\hat{\sigma}$ to $\mathcal{C}$ as the signature to the message $M^*||\bar{\mathsf{pk}}$, since such message was never asked to the signing oracle of $\mathcal{C}$. If $\mathcal{A}$ wins, then $\mathcal{B}$ breaks the EUF-CMA security of Π'. □

We can obtain a similar conversion for strongly unforgeable (SUF-CMA) signatures. Suppose the signature scheme Π' is not RPA secure in a way that given a signature σ on a message m for a public key pk, and some $\phi \in \Phi$, there exists a polynomial time algorithm to find $(\bar{\sigma}, \bar{M}) \neq (\sigma, M)$ such that

$$\mathsf{Verify}'(\pi, \mathsf{pk}, \sigma, M) = \mathsf{Verify}'(\pi, \mathcal{T}(\pi, \phi(\mathsf{sk})), \bar{\sigma}, \bar{M}).$$

We call such scheme Φ-*RPA insecure.* Using such signature scheme Π', we can build another signature scheme Π same as above.

THEOREM 4. *If Π' is a SUF-CMA secure separable signature scheme with Φ_A-public key malleability but Φ_B-RPA insecure, then Π is a Φ_P-RPA, SUF-CMA secure signature scheme, where $\Phi_P = \Phi_A \cap \Phi_B$ is some class of group induced operations.*

PROOF. Suppose the simulator $\mathcal{B}$ is given π, pk from the challenger $\mathcal{C}$ of Π'. $\mathcal{B}$ forwards π, pk to the adversary $\mathcal{A}$.

For the $\mathsf{Signing\ Oracle}$ query with input M, $\mathcal{B}$ asks $\mathcal{C}$ to answer the signature on the message $M||\mathsf{pk}$. For the $\mathsf{KeyGen\ Oracle}$ query with input ϕ, $\mathcal{B}$ answers $\bar{\mathsf{pk}} = \mathcal{T}(\pi, \phi(\mathsf{sk}))$ by the Φ_A-public key malleability property.

Finally, $\mathcal{A}$ returns a forgery σ^* on a message M^* with respect to the public key $\bar{\mathsf{pk}}$ such that $\mathsf{Verify}(\pi, \bar{\mathsf{pk}}, \sigma^*, M^*||\bar{\mathsf{pk}}) = 1$. If $\bar{\mathsf{pk}} = \mathsf{pk}$, then M^* must not be asked in the signing oracle. Then $\mathcal{B}$ returns σ^* to $\mathcal{C}$ as the signature to the message $M^*||\mathsf{pk}$. Else, $\bar{\mathsf{pk}}$ is the output of the $\mathsf{KeyGen\ Oracle}$ for some input ϕ. Then $\mathcal{B}$ transforms $(\sigma^*, M^*||\bar{\mathsf{pk}})$ to $(\hat{\sigma}, \hat{M})$ such that $\mathsf{Verify}(\pi, \mathsf{pk}, \hat{\sigma}, \hat{M}) = 1$ by the Φ_B-RPA fix-message insecurity. This transformation is possible since ϕ^{-1} exists for group induced operations in Φ_P. $\mathcal{B}$ returns $\hat{\sigma}$ to $\mathcal{C}$ as the signature to the message $\hat{M}$, since such pair of $(\hat{\sigma}, \hat{M})$ was never asked to the signing oracle of $\mathcal{C}$. If $\mathcal{A}$ wins, then $\mathcal{B}$ breaks the SUF-CMA security of Π'. □

7. CONCLUSION

We propose a framework which includes the existing related (secret) key attack, and the new models of related encryption randomness attack, related signing randomness attack, and related public key attack. Corresponding, we propose generic constructions to achieve security under these related randomness attacks. We hope our work to be useful in building cryptosystems in the face of practical attacks resulted from fault injection or poor implementation of pseudorandom number generators.

8. ACKNOWLEDGMENTS

Sherman S.M. Chow would like to acknowledge the support from Early Career Scheme and Early Career Award of the Research Grants Council, Hong Kong (CUHK 439713). S.M. Yiu would like to acknowledge the support from the Seed Funding Programme for Basic Research (201311159040), HKU and NSFC/RGC Joint Research Scheme, Hong Kong (N_HKU 729/13).

9. REFERENCES

[1] G. Argyros and A. Kiayias. I forgot your password: randomness attacks against PHP applications. In *USENIX*, Security'12, page 6. USENIX Association, 2012.

[2] M. Bellare, A. Boldyreva, and J. Staddon. Randomness Re-use in Multi-recipient Encryption Schemes. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 85–99. Springer, 2003.

[3] M. Bellare and D. Cash. Pseudorandom Functions and Permutations Provably Secure against Related-Key Attacks. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 666–684. Springer, 2010.

[4] M. Bellare, D. Cash, and R. Miller. Cryptography Secure against Related-Key Attacks and Tampering. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, 2011.

[5] M. Bellare and T. Kohno. A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, 2003.

[6] M. Bellare, K. G. Paterson, and S. Thomson. RKA Security beyond the Linear Barrier: IBE, Encryption and Signatures. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 331–348. Springer, 2012.

[7] A. Bender, J. Katz, and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3816 of *LNCS*, pages 60–79. Springer, 2006.

[8] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In B. S. K. Jr., editor, *CRYPTO '97*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.

[9] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and Related-Key Attack on the Full AES-256. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 231–249. Springer, 2009.

[10] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[11] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In W. Fumy, editor, *EUROCRYPT '97*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

[12] V. Goyal, A. O'Neill, and V. Rao. Correlated-Input Secure Hash Functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, 2011.

[13] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your Ps and Qs: detection of widespread weak keys in network devices. In *USENIX*, Security'12, page 35. USENIX Association, 2012.

[14] J. Herranz and G. Sáez. Forking Lemmas for Ring Signature Schemes. In T. Johansson and S. Maitra, editors, *INDOCRYPT 2003*, volume 2904 of *LNCS*, pages 266–279. Springer, 2003.

[15] K. Kurosawa. Multi-recipient Public-Key Encryption with Shortened Ciphertext. In D. Naccache and P. Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 48–63. Springer, 2002.

[16] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Public Keys. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 626–642. Springer, 2012.

[17] S. Lucks. Ciphers Secure against Related-Key Attacks. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 359–370. Springer, 2004.

[18] K. Michaelis, C. Meyer, and J. Schwenk. Randomly Failed! The State of Randomness in Current Java Implementations. In E. Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 129–144. Springer, 2013.

[19] K. G. Paterson, J. C. N. Schuldt, and D. L. Sibborn. Related Randomness Attacks for Public Key Encryption. In *PKC 2014*, volume 8383 of *LNCS*, pages 465–482. Springer, 2014.

APPENDIX

A. RELATED KEY AND RANDOMNESS ATTACK MODEL FOR PKE

An encryption scheme consists of three PPT algorithms:

1. `Setup`: On input a security parameter 1^λ, it generates a public key pk and a secret key sk.
2. `Enc`: On input pk and a message M from a message space $\mathcal{M}$, it outputs a ciphertext $\mathfrak{C}$.
3. `Dec`: On input sk, and $\mathfrak{C}$, it outputs a message M or $\perp$ symbolizing the failure of decryption.

Correctness. $\forall M \in \mathcal{M}$, $M \leftarrow \mathtt{Dec}(\mathsf{sk}, \mathtt{Enc}(\mathsf{pk}, M))$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{Setup}(1^\lambda)$.

We consider the following indistinguishability based game against chosen ciphertext attacks (IND-CCA) for semantic security with related-key, related encryption randomness attack. The game (Φ_k, Φ_r)-RKRA-PKE is defined as follows.

1. Setup. The challenger runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{Setup}(1^\lambda)$ and gives pk to the adversary $\mathcal{A}$.
2. Query 1. The following oracles can be queried by $\mathcal{A}$:
   - Decryption Oracle $\mathcal{DO}(C, \phi)$: On input a ciphertext C and a key leakage $\phi \in \Phi_k$, it returns a message $m/\perp \leftarrow \mathtt{Dec}(\phi(\mathsf{sk}), C)$.
3. Challenge. $\mathcal{A}$ sends two messages $M_0, M_1 \in \mathcal{M}$ to the challenger. The challenger picks a random bit b' and computes $\mathfrak{C}^* \leftarrow \mathtt{Enc}(\mathsf{pk}, M_{b'}; R^*)$ using the randomness R^*. The challenger sends $\mathfrak{C}^*$ to $\mathcal{A}$.
4. Query 2. The following oracles can be queried by $\mathcal{A}$:
   - Decryption Oracle $\mathcal{DO}(C, \phi)$: On input a ciphertext C and a key leakage $\phi \in \Phi_k$, it returns a message $m/\perp \leftarrow \mathtt{Dec}(\phi(\mathsf{sk}), C)$.
   - Encryption Oracle $\mathcal{EO}(M, \phi)$: On input a message M and a randomness leakage $\phi \in \Phi_r$, it returns $\mathtt{Enc}(\mathsf{pk}, M; \phi(R^*))$.
5. Output. $\mathcal{A}$ returns a guess b^* of b'.

$\mathcal{A}$ wins the game if $b' = b^*$, there was no query to $\mathcal{DO}$ with input $C = C^*$ and ϕ is an identity map, and no query to $\mathcal{EO}$ with input $M = M_0^*$ or M_1^*, and ϕ is an identity map. The advantage of $\mathcal{A}$ is $\left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right|$.

An encryption scheme is (t, ϵ)-(Φ_k, Φ_r)-RKRA IND-CCA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the (Φ_k, Φ_r)-RKRA-PKE game above.

If there is no related randomness attack, then the attacker cannot query the encryption oracle. In this case, we say that an encryption scheme is (t, ϵ)-Φ_k-RKA IND-CCA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the game above.

If there is no related key attack, then the attacker cannot query the decryption oracle with input $\phi \in \Phi_K$. In this case, we say that an encryption scheme is (t, ϵ)-Φ_r-RRA IND-CCA secure if there is no t-time attacker $\mathcal{A}$ with advantage ϵ in the game above.