Public-Key Encryption Resilient to Linear Related-Key Attacks

Hui Cui, Yi Mu, and Man Ho Au

School of Computer Science and Software Engineering,
University of Wollongong, Wollongong, NSW 2522, Australia
hc892@uowmail.edu.au, {ymu,aau}@uow.edu.au

Abstract. In this paper, we consider the security of public-key encryption schemes under linear related-key attacks, where an adversary is allowed to tamper the private key stored in a hardware device, and subsequently observe the outcome of a public-key encryption system under this modified private key. Following the existing work done in recent years, we define the security model for related-key attack (RKA) secure public-key encryption schemes as chosen-ciphertext and related-key attack (CC-RKA) security, in which we allow an adversary to issue queries to the decryption oracle on the linear shifts of the private keys. On the basis of the adaptive trapdoor relations via the one-time signature schemes, Wee (PKC'12) proposed a generic construction of public-key encryption schemes in the setting of related-key attacks, and some instantiations from Factoring, BDDH with CC-RKA security, and DDH but with a weaker CC-RKA security. These schemes are efficient, but one-time signatures still have their price such that in some cases they are not very efficient compared to those without one-time signatures. Bellare, Paterson and Thomson (ASIACRYPT'12) put forward a generic method to build RKA secure public-key encryption schemes, which is transformed from the identity-based encryption schemes. However, so far, the efficient identity-based encryption schemes are generally based on parings. To generate a specific construction of public-key encryption schemes against related-key attacks without pairings, after analyzing the related-key attack on the Cramer-Shoup basic public-key encryption scheme, we present an efficient public-key encryption scheme resilient against related-key attacks without using one-time signature schemes from DDH. Finally, we prove the CC-RKA security of our scheme without random oracles.

Keywords: Public-key encryption, Related-key attack, CC-RKA security.

1 Introduction

In the traditional security model, it is assumed that the adversary is isolated from the internal states of the honest communication parties. However, with the development of information technologies, the security of cryptographic algorithms in modern cryptography is analyzed in the black-box model, where an adversary may view the algorithm's inputs and outputs, but the private key as

T. Zia et al. (Eds.): SecureComm 2013, LNICST 127, pp. 182–196, 2013.

well as all the internal computation remains perfectly hidden. Unfortunately, this idealized assumption is often hard to satisfy in real systems. In many situations, the adversary might get some partial information about private keys through methods which were not anticipated by the designer of the system and, correspondingly, not taken into account when arguing its security. Such attacks, referred to as key-leakage attacks, come in a large variety. An important example is side-channel [18] attacks that exploit information leakage from the implementation of an algorithm, where an adversary observes some "physical output" of a computation (such as radiation, power, temperature, running time), in addition to the "logical output" of the computation.

In recent two decades, this requirement has been relaxed to capture security under the scenarios where some information of the keys is leaked to the adversary. When an adversary tampers the private key stored in a cryptographic hardware device, and observes the result of the cryptographic primitive under this modified private key, there is a related-key attack (RKA) [4,11]. The key here could be a signing key of a certificate authority or a decryption key of an encryption scheme. In related-key attacks, the adversary attempts to break an encryption scheme by invoking it with several private keys satisfying some known relations.

Wee [20] proposed a generic construction of public-key encryption schemes in the setting of linear related-key attacks. In [20], the constructions exploit certain existing public-key encryption schemes that are susceptible to linear related-key attacks, to obtain public-key encryption schemes that are secure against linear related-key attacks from adaptive trapdoor relations via strong one-time signatures, which generates a tag in the ciphertext of the concrete scheme. The security of this realization is analogous to those for obtaining chosen-ciphertext attack (CCA) security from extractable hash proofs [19], and trapdoor functions [15], which implies a trick that the RKA decryption oracle will return $\perp$ for $tag = tag^*$ generated from an one-time signature scheme, whenever the ciphertext with tag given by the adversary matches the challenge ciphertext with tag^* or not. Briefly, RKA.Decrypt oracle outputs $\perp$ when given a ciphertext with $tag = tag^*$ even $\phi(sk) \neq sk$, where ϕ denotes a linear shift. That is to say, the RKA decryption query will not help the adversary to obtain more information if $tag = tag^*$. Besides, Wee [20] designed some efficient strong one-time signatures to reduce the total overhead of the specific schemes. However, though one-time signatures are easy to construct in theory, and are more efficient than full-fledged signatures, they still have their price. Particularly,

- Known one-time signature schemes based on general one-way functions [10] allow very efficient signing, key generation and signature verification, but they require the expensive valuations of the one-way function. More problematic, such schemes usually have long public keys and signatures, resulting in long ciphertexts.
- Although one-time signature schemes constructed based on number-theoretic assumptions by adapting full-fledged signature schemes have the advantage of shorter public keys and signatures, but this yields schemes of which computational cost for key generation, signing, and verifying is more expensive.

Bellare, Paterson and Thomson [6] provided a framework to enable the construction of identity-based encryption schemes that are secure under related-key attacks. In [6], a very particular type of framework, which allows to reduce RKA security of a modified identity-based encryption scheme directly to the normal identity-based encryption security of a base identity-based encryption scheme, is used. Because of this framework, exploiting known results on identity-based encryption in a black-box way is allowed and re-entering the often complex security proofs of the base identity-based encryption schemes is avoided. Based on this, they constructed the RKA secure schemes for public-key encryption. Their schemes are achieved in the standard model and hold under reasonable hardness assumptions in the standard model, but they are transformed from the identity-based encryption schemes. Anyway, most of the current identity-based encryption schemes based on bilinear pairings, which are not very efficient.

Our Contributions. Inspired by the above, in this paper, we attempt to bridge this gap in Wee's public-key encryption schemes resilient against related-key attacks from DDH [20] without using any one-time signature schemes. First of all, we review the definition of linear related-key attacks introduced by Wee [20] in the setting of public-key encryption, and describe how to attack the public-key encryption system of Cramer and Shoup [9] in our RKA security model, which is a bit different from that described in [20]. In Wee's attack [20], a related-key deriving function only changes one part of the secret keys, while our attack changes all parts of the secret keys with the same linear shift function ϕ. In the second place, on the practical side, with some trivial modifications to the basic cryptosystem of Cramer and Shoup [9], we obtain an efficient scheme that is RKA secure based on the decisional Diffie-Hellman assumption. Our technique is to hide the functions related to the randomness that appear in the Cramer-Shoup scheme, such that even given the private keys used in the encryption function of the message, the adversary still has no idea to output the message under the modified secret keys without the hiding information. Our scheme is very efficient, as we do not need any pairing computation to implement encryption and decryption. Finally, we prove the CC-RKA security of our scheme under the DDH assumption. Interestingly, regarding the CC-RKA security proof, [20] simulates the RKA decryption queries via key homomorphism and make the adversary fail through key fingerprint, and [6] uses key malleability to simulate the RKA decryption queries and collision-resistant identity renaming to make the proof goes; however, in our specific construction, we avoid to make use of such techniques to claim the security.

To begin with, we briefly describe the framework introduced in [4]. Informally, a public-key encryption scheme is resilient to related-key attacks, then it is chosen-ciphertext attack secure even when the adversary obtains partial information of the message in the scheme under the modified private keys of the adversary. This is modeled by providing the adversary with access to a related-key attack decryption oracle: the adversary can query the decryption oracle with any function (ϕ, C), and then receive $(\phi(sk), C)$, where sk is the secret

key (we note that the related-key deriving functions can be chosen depending on the public key, which is known to the adversary). The adversary can query the related-key attack decryption oracle adaptively, with only one restriction that the decryption of a ciphertext C with the private key $\phi(sk)$ cannot equal the decryption of the challenge ciphertext C^* with the original private key sk.

1.1 Related Works

Micali and Reyzin [17] put forward a comprehensive framework for modeling security against side-channel attacks in 2004, which relies on the assumption that there is no leakage of information in the absence of computation. Later in 2008, Halderman et al. [14] described a set of attacks violating the assumption of the framework of Micali and Reyzin. Specially speaking, their "cold boot" attacks showed that a significant fraction of the bits of a cryptographic key can be recovered if the key is ever stored in memory, of which the framework was modeled by Akavia, Goldwasser and Vaikuntanathan [1]. Similarly, fault injection techniques can be used to falsify, inducing the internal state of the devices being modified, if given physical access to the hardware devices [7].

Bellare and Kohno [5] investigated related-key attacks from a theoretical point of view and presented an approach to formally handle the notion of related-key attacks. Followed the approach in [5], Lucks [16] presented some constructions for block ciphers and pseudorandom function generators. To solve the open problem in related-secret security whether or not related-key secure blockciphers exist, Bellare and Cash [3] provided the first constructions to create related-secret pseudorandom bits. Based on the work in [3], Applebaum, Harnik, and Ishai [2] gave RKA secure symmetric encryption schemes, which can be used in garbled circuits in secure computation. Later, Bellare, Cash and Miller [4] proposed approaches to build high-level primitives secure against related-key attacks like signatures, CCA secure public-key encryption, identity-based encryption, based on RKA secure pseudorandom functions. Also, there are a lot of other works about cryptographic systems with RKA security such as signatures [6,13], CCA secure public-key encryption [6,20], identity-based encryption [6].

The remainder of this paper is organized as follows. In Section 2, we briefly present the basic definitions, and the security assumptions that are used in our construction. In Section 3, we review the concepts associated to this work and the security model of RKA secure public-key key encryption systems. In Section 4, we propose an efficient public-key encryption scheme resilient against related-key attacks, after the analysis of a linear attack on the Cramer-Shoup cryptosystem [9], and prove its security under the hardness of the DDH problem. Finally, we conclude this paper in Section 5.

2 Preliminaries

In this section, we look back some basic notions, definitions, and tools that are used in our construction. We formally state the decisional Diffie-Hellman

assumptions, and present the technical definitions that will be used repeatedly in our analysis.

2.1 Complexity Assumptions

Suppose that Groupgen is a probabilistic polynomial-time algorithm that inputs a security parameter 1^λ, and outputs a triplet (G, p, g) where G is a group of order p that is generated from g, and p is a prime number.

The Decisional Diffie-Hellman Assumption. The decisional Diffie-Hellman (DH) assumption is that the ensembles $\{G, g, f, g^r, f^r\}$ and $\{G, g, f, g^{r_1}, f^{r_2}\}$ are computationally indistinguishable, where $(G,\ p,\ g) \leftarrow \text{Groupgen}(1^\lambda)$, and the elements $g,\ f \in G,\ r,\ r_1,\ r_2 \in Z_p$ are chosen independently and uniformly at random.

A Basic Scheme Based on DDH. Since the introduction of DDH assumption [8], it has already found several interesting applications. Note that the DDH assumption readily gives a chosen-plaintext attack (CPA) secure public-key encryption scheme. Let the public key consist of random elements $g,\ f,\ g^{x_1},\ f^{x_2} \in G$, and the secret key consist of random element $x_1,\ x_2 \in Z_p$. The encryption of a message $M \in G$ is given by $(C_1,\ C_2,\ C_3) = (g^r,\ f^r,\ (g^{x_1} f^{x_2})^r \cdot M)$, where $r \in Z_p$ is a random element. The message M can be recovered with the secret key $x_1,\ x_2$ by computing $M = C_3 \cdot (C_1)^{-x_1} \cdot (C_2)^{-x_2}$.

2.2 Public-Key Encryption

A public-key encryption scheme is composed of the following four randomized algorithms [12]: Keygen, Encrypt, and Decrypt.

- $\text{Keygen}(1^\lambda) \rightarrow (sk,\ pk)$: Taking a security parameter λ as input, this algorithm outputs a private key and a public key pair $(sk,\ pk)$.
- $\text{Encrypt}_{pk}(m) \rightarrow C$: Taking a plaintext m (in some implicit message space), and a public key pk as input, this algorithm outputs a ciphertext C.
- $\text{Decrypt}_{sk}(C) \rightarrow m$: Taking a plaintext m, a ciphertext C, and a private key sk as input, this algorithm outputs m for a valid ciphertext or $\perp$ for an invalid ciphertext.

We require that a public-key encryption system is correct, meaning that if $(sk,\ pk) \leftarrow \text{Keygen}(1^\lambda)$, and $C \leftarrow \text{Encrypt}_{pk}(m)$, then $\text{Decrypt}_{sk}(C) \rightarrow m$.

3 Modeling Related-Key Attacks

In this section, we define the notion of a chosen-ciphertext attack; in addition, we present a natural extension of this notion to the setting of related-key attacks, as introduced by Bellare, Cash and Miller [4]. Also, we introduce some notions about related-key attacks, as proposed in [2].

3.1 Chosen-Ciphertext Attacks

A public-key encryption scheme (Keygen, Encrypt, Decrypt) is secure against chosen-ciphertext attacks (CCA security) if for a stateful adversary algorithm $\mathcal{A}$, the advantage in the following game is negligible in the security parameter λ.

1. $(sk, pk) \leftarrow \text{Keygen}(1^\lambda)$.
2. $(m_0, m_1) \leftarrow \mathcal{A}^{\text{Decrypt}_{sk}(\cdot)}(pk)$ such that $|m_0| = |m_1|$.
3. $C^* \leftarrow \text{Encrypt}_{pk}(m_d)$ where $d \in \{0, 1\}$.
4. $d' \leftarrow \mathcal{A}^{\text{Decrypt}_{sk}(\cdot)}(C^*)$.
5. Output d'.

Here $\text{Decrypt}_{sk}(\cdot)$ is an oracle that on an input C, it returns $\text{Decrypt}_{sk}(C)$.

The weaker security notion of CPA security (i.e. secure against CPAs) is obtained in the above security game when depriving adversary $\mathcal{A}$ of the the access to the decryption oracle.

3.2 RKA Security

Related-Key Deriving Functions. Our definition follows the notion of related-key deriving functions given in [5]. Briefly speaking, a class Φ of related-key deriving functions ϕ: $sk \rightarrow sk$ is a finite set of functions with the same domain and range, which map a key to a related key. Additionally, Φ should allow an efficient membership test, and ϕ should be efficiently computable. Note that in our concrete constructions, we only consider the class Φ^+ as linear shifts.

The family Φ^+. Any function $\phi : Z_p \rightarrow Z_p$ in this class is indexed by $\triangle \in Z_p$, where $\phi_\triangle(sk) : = sk + \triangle$.

We constraint that if sk is composed of several elements as $(sk_1, \ldots, sk_n)$ with $n \in Z^+$, for any sk_i where $i \in \{1, \ldots, n\}$, $\phi_\triangle(sk_i) : = sk_i + \triangle$ with $\triangle \in Z_p^n$.

CC-RKA Security. A public-key encryption scheme (Keygen, Encrypt, Decrypt) is Φ-CC-RKA secure if for a stateful adversary algorithm $\mathcal{A}$, the advantage in the following game is negligible in the security parameter λ.

1. $(sk, pk) \leftarrow \text{Keygen}(1^\lambda)$.
2. $(m_0, m_1) \leftarrow \mathcal{A}^{\text{RKA.Decrypt}_{sk}(\cdot,\cdot)}(pk)$ such that $|m_0| = |m_1|$.
3. $C^* \leftarrow \text{Encrypt}_{pk}(m_d)$ where $d \in \{0, 1\}$.
4. $d' \leftarrow \mathcal{A}^{\text{RKA.Decrypt}_{sk}(\cdot,\cdot)}(C^*)$.
5. Output d'.

Here $\text{RKA.Decrypt}_{sk}(\cdot, \cdot)$ is an oracle that on an input (ϕ, C), it returns $\text{Decrypt}_{\phi(sk)}(C)$. We constraint that algorithm $\mathcal{A}$ can only make queries (ϕ, C) such that $\phi \in \Phi$ and $(\phi(sk), C) \neq (sk, C^*)$.

We say that algorithm $\mathcal{A}$ succeeds if $d' = d$, and algorithm $\mathcal{A}$'s advantage can be defined as

$$\text{Adv}^{\text{CCRKA}}_{\Phi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} |\text{Pr}^{\text{CCRKA}}_{\Phi,\mathcal{A}}[\text{Succ}] - 1/2|,$$

where $\Pr^{\mathrm{CCRKA}}_{\Phi,\mathcal{A}}[\mathrm{Succ}]$ denotes the event that algorithm $\mathcal{A}$ outputs the bit $d' = d$.

Briefly speaking, key fingerprint means that any attempt to forge sk induces a random output of $\mathrm{Decrypt}_{sk}(c')$.

4 An Efficient Construction without Pairings

In this section, we put forward our construction based on the Cramer-Shoup cryptosystem [9], and present its security proof under the DDH assumption. To begin with, we describe a simple linear related-key attack on the Cramer-Shoup public-key encryption scheme, which to some extent illustrate some technical obstacles in achieving RKA security.

4.1 Related-Key Attacks on Cramer-Shoup Cryptosystem

We point out a linear related-key attack on the CCA secure encryption scheme based on the DDH assumption proposed by Cramer and Shoup [9]. The details of the Cramer-Shoup public-key encryption scheme is given as follows.

- Key generation. Choose random g, $f \in G$, x, y, a, b, α, $\beta \in Z_p$, a collision resistant hash function $H\colon G^3 \to Z_p$, and sets $u_1 = g^x f^y$, $u_2 = g^a f^b$, $u_3 = g^\alpha f^\beta$.
  The public key is $PK = (g, f, u_1, u_2, u_3, H)$, and the secret key is $SK = (x, y, a, b, \alpha, \beta)$.
- Encryption. To encrypt message $M \in G$,
  1. choose random $r \in Z_p$, and set $C_1 = g^r$, $C_2 = f^r$, $C_3 = {u_1}^r \cdot M$.
  2. compute $t = H(C_1, C_2, C_3)$, $C_4 = (u_2 {u_3}^t)^r$.
  3. output ciphertext $C = (C_1, C_2, C_3, C_4)$.
- Decryption. To decrypt ciphertext $C = (C_1, C_2, C_3, C_4)$,
  1. compute $t = H(C_1, C_2, C_3)$, and output $\perp$ if $C_4 \neq {C_1}^{a+t\alpha} {C_2}^{b+t\beta}$.
  2. otherwise, output $M = C_3 \cdot {C_1}^{-x} \cdot {C_2}^{-y}$.

Suppose we are given a valid ciphertext (C_1, C_2, C_3, C_4) of some message M. We can recover M by making decryption queries to RKA.Decrypt oracle on related secret keys via the following attack. For any $\triangle \in Z_p$, we change the secret key $(x, y, a, b, \alpha, \beta)$ to $(x+\triangle, y+\triangle, a+\triangle, b+\triangle, \alpha+\triangle, \beta+\triangle)$, then $(C_1, C_2, C_3, C_4 \cdot (C_1 \cdot C_2)^{\triangle + t\cdot\triangle})$ can be decrypted to $M \cdot (C_1 \cdot C_2)^{-\triangle}$ under the modified secret keys. As C_1, C_2 and $\triangle$ are known to us, we can obtain M easily by computing $M \cdot (C_1 \cdot C_2)^{-\triangle} \cdot (C_1 \cdot C_2)^{\triangle}$.

Obviously in the above cases, message M can be easily recovered given the output of the decryption algorithm on the modified secret keys.

4.2 Our Construction

Let G be a group of prime order p. We present a public-key encryption scheme which is CCA secure under the linear related-key attacks as follows.

- Key generation. Choose random elements g, f, $h \in G$, x, y, a, b, α, β, $\gamma \in Z_p$, a collision resistant hash function H: $G^4 \to Z_p$, and sets $u_1 = g^x f^y$, $u_2 = g^a f^b$, $u_3 = g^\alpha f^\beta$, $v = h^\gamma$.
  The public key is $PK = (g, h, f, u_1, u_2, u_3, v)$, and the secret key is $SK = (x, y, a, b, \alpha, \beta, \gamma)$.
- Encryption. To encrypt message $M \in G$,
  1. choose random elements $r, r' \in Z_p$, and set
  $$C_1 = g^r v^{r'}, \quad C_2 = f^r v^{r'}, \quad C_3 = h^{r'}, \quad C_4 = {u_1}^r \cdot M.$$
  2. compute $t = H(C_1, C_2, C_3, C_4)$, $C_5 = (u_2 {u_3}^t)^r$.
  3. output ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$.
- Decryption. To decrypt ciphertext $C = (C_1, C_2, C_3, C_4, C_5)$,
  1. compute $t = H(C_1, C_2, C_3, C_4)$, and output $\perp$ if the following equation holds.
  $$C_5 \neq (C_1 \cdot {C_3}^{-\gamma})^{a+t\alpha} (C_2 \cdot {C_3}^{-\gamma})^{b+t\beta}.$$
  2. otherwise, output M as $M = C_4 \cdot (C_1 \cdot {C_3}^{-\gamma})^{-x} \cdot (C_2 \cdot {C_3}^{-\gamma})^{-y}$.

Correctness. For any sequence of the key generation and encryption algorithms, it holds that

$$\begin{aligned}(u_2 {u_3}^t)^r &= (C_1 \cdot {C_3}^{-\gamma})^{a+t\alpha} (C_2 \cdot {C_3}^{-\gamma})^{b+t\beta} \\ &= (g^a f^b (g^\alpha f^\beta)^t)^r, \\ M &= C_4 \cdot (C_1 \cdot {C_3}^{-\gamma})^{-x} \cdot (C_2 \cdot {C_3}^{-\gamma})^{-y} \\ &= C_4 \cdot (g^x f^y)^{-r},\end{aligned}$$

and therefore the decryption algorithm is always correct.

Remarks. Note that compared to the scheme proposed in [20], our construction is more efficient. The CCA-RKA secure public-key encryption schemes in [20] are built from adaptive trapdoor relations [15] to generate a *tag* for every ciphertext via a strong one-time signature scheme, which implies a trick in it such that the adversary cannot obtain more information if *tag* of a ciphertext C equals tag^* of the challenge ciphertext C^*, not to mention $C = C^*$; while in our construction, we use the Cramer-Shoup public-key encryption scheme [9] as the basis, and the strong one-time signature schemes are replaced by the ciphertext to generate *tag*, such that RKA.Decrypt oracle will still not facilitate the adversary when a given ciphertext C matches the challenge one C^*, as long as SK does not equal to $\phi(SK)$ for any $\phi \in \Phi$.

4.3 Security

Theorem 1. *Assume the hardness of decisional DH problem, the above public-key encryption scheme is secure in the CC-RKA security game regarding linear related-key deriving function* ϕ^+.

Proof. The proof of security is based on augmenting the proof of Cramer and Shoup with the ideas of generating a generic construction. Specifically, we show that any algorithm $\mathcal{A}$ that breaks the security of the scheme, we can build an algorithm $\mathcal{B}$ that can distinguish between a DH instance and a non-DH instance, which is given a random tuple $(g, f, Z_1 = g^r, Z_2 = f^r) \in G^4$ as input.

Setup. Algorithm $\mathcal{B}$ chooses random elements $h \in G$, $x, y, a, b, \alpha, \beta, \gamma \in Z_p$, and a collision resistant hash function H: $G^4 \to Z_p$, and then sets $u_1 = g^x f^y$, $u_2 = g^a f^b$, $u_3 = g^\alpha f^\beta$, $v = h^\gamma$.

Algorithm $\mathcal{B}$ sends the public key $PK = (g, h, f, u_1, u_2, u_3, v)$ to algorithm $\mathcal{A}$, and keeps the private key $SK = (x, y, a, b, \alpha, \beta, \gamma)$.

Phase 1. Algorithm $\mathcal{A}$ queries (ϕ, C) to RKA.Decrypt oracle. Algorithm $\mathcal{B}$ responds using the private key $\phi(SK)$.

Challenge. Algorithm $\mathcal{A}$ outputs two messages M_0, M_1 on which it wishes to be challenged. Algorithm $\mathcal{B}$ chooses a random bit $d \in \{0, 1\}$, and a random element $r' \in Z_p$, and then responds with the ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$, where

$$C_1^* = Z_1 v^{r'}, \quad C_2^* = Z_2 v^{r'}, \quad C_3^* = h^{r'},$$
$$C_4^* = {Z_1}^x {Z_2}^y \cdot M_d, \quad C_5^* = {Z_1}^{a+\alpha t^*} {Z_2}^{b+\beta t^*}.$$

Here $t^* = H(C_1^*, C_2^*, C_3^*, C_4^*)$.

Phase 2. Algorithm $\mathcal{A}$ continues to adaptively issue queries (ϕ, C) to RKA.Decrypt oracle.

- If $\phi(SK) = SK$ and $C = C^*$, such queries are ruled out by the definition of CC-RKA security game, so algorithm $\mathcal{B}$ responds with $\perp$.
- Otherwise, algorithm $\mathcal{B}$ responds as in Phase 1.

Output. Algorithm $\mathcal{A}$ output a guess $d' \in \{0, 1\}$. If $d' = d$, algorithm $\mathcal{B}$ output 1; otherwise, algorithm $\mathcal{B}$ outputs 0.

Obviously, if (g, f, Z_1, Z_2) is a DH instance, then the simulation will be identical to the actual attack, such that algorithm $\mathcal{A}$ has a non-negligible advantage in outputting the bit $d' = d$.

Lemma 1. *If (g, f, Z_1, Z_2) is a DH instance then algorithm $\mathcal{A}$'s view is identical to the actual attack.*

Proof. The actual attack and simulated attack are identical except for the challenge ciphertext. It remains to prove that the challenge ciphertext has the correct distribution when (g, f, Z_1, Z_2) is a DH instance. Actually, in this case, for a random $r \in Z_p$, $Z_1 = g^r$ and $Z_2 = f^r$, the ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$ as it should be. Assume that algorithm $\mathcal{A}$'s advantage in breaking the CC-RKA

security of the above scheme is ϵ, then we can see that algorithm $\mathcal{A}$'s probability in outputting the bit $d = d'$ could be $1/2 + \epsilon$.

Next, we show that if (g, f, Z_1, Z_2) is a non-DH instance, then algorithm $\mathcal{A}$ has a negligible advantage in outputting the bit $d' = d$. We assume that (g, f, Z_1, Z_2) is a non-DH instance, where $\log_g Z_1 = r_1$, $\log_f Z_2 = r_2$, and $r_1 \neq r_2$.

Let $(C_1^*, C_2^*, C_3^*, C_4^*)$ be the challenge ciphertext given to algorithm $\mathcal{A}$ by algorithm $\mathcal{B}$. We use Failure to denote the event where for RKA decryption queries (ϕ, C) it holds that $(C_1, C_2, C_3, C_4) \neq (C_1^*, C_2^*, C_3^*, C_4^*)$, and $H(C_1, C_2, C_3, C_4) = H(C_1^*, C_2^*, C_3^*, C_4^*)$. Note that the event Failure has a negligible probability to occur because hash function H is collision resistant. We say that a ciphertext C is invalid if $\log_g \frac{C_1}{{C_3}^{\gamma+\triangle}} \neq \log_f \frac{C_2}{{C_3}^{\gamma+\triangle}}$ for any $\triangle \in Z_p^n$.

Below we prove that algorithm $\mathcal{A}$ has a negligible advantage in outputting the bit $d' = d$ if the event Failure does not happen. Specifically speaking, we perform it in two cases: (1) if the event Failure does not happen, then the RKA decryption oracle rejects all invalid ciphertexts except with a negligible probability; (2) if the RKA decryption oracle rejects all invalid ciphertexts, then algorithm $\mathcal{A}$ has a negligible advantage in outputting the bit $d' = d$. We conclude by the fact that the event Failure occurs with a negligible probability.

Lemma 2. *If (g, f, Z_1, Z_2) is a non-DH instance and the event Failure does not happen, then the RKA decryption algorithm rejects all invalid ciphertexts except with a negligible probability.*

Proof. The probability of the invalid ciphertexts happening in our security game is analogous to that in the Cramer-Shoup public-key encryption scheme [9] except that for the RKA decryption oracles, some invalid ciphertexts which will be rejected in the security game of the Cramer-Shoup scheme will be accepted in our security game. Suppose that algorithm $\mathcal{A}$ is given the public key $PK = (g, h, f, u_1, u_2, u_3, v)$, and the challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$. We prove this lemma via considering $(a, b, \alpha, \beta) \in Z_p$ from algorithm $\mathcal{A}$'s point of view, such that for $k = \log_g f$, (a, b, α, β) is uniformly random subject to

$$\begin{cases} \log_g u_2 = a + kb \\ \log_g u_3 = \alpha + k\beta \\ \log_g C_5^* = r_1 a + r_2 kb + t^* r_1 \alpha + t^* r_2 k\beta \end{cases}.$$

Note that algorithm $\mathcal{A}$ learns nothing on (a, b, α, β) by querying valid ciphertexts to the decryption oracle. Actually, from submitting a valid ciphertext, algorithm $\mathcal{A}$ only learns a linear combination of the constraint $\log_g u_1 = x + ky$ which is know from the public key.

We denote $(C_1, C_2, C_3, C_4, C_5) \neq (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$ as the first invalid ciphertext queried by algorithm $\mathcal{A}$, where $C_1 = g^{r_1} v^{r'}$, $C_2 = f^{r_2} v^{r'}$, $r_1 \neq r_2$, and $t = H(C_1, C_2, C_3, C_4)$. In this case, there are three cases we need to take into consideration.

- $(C_1, C_2, C_3, C_4) \neq (C_1^*, C_2^*, C_3^*, C_4^*)$ and $t = t^*$. This is impossible since we assume that the event Failure does not happen.

Note that the event Failure will never happen because the hash function H in our construction is collision resistant.

- $(C_1, C_2, C_3, C_4) \neq (C_1^*, C_2^*, C_3^*, C_4^*)$ and $t \neq t^*$. In this case, if the RKA decryption algorithm accepts the invalid ciphertext, we obtain the following equations.

$$\begin{cases} \log_g u_2 = a + kb \\ \log_g u_3 = \alpha + k\beta \\ \log_g C_5^* = r_1 a + r_2 kb + t^* r_1 \alpha + t^* r_2 k\beta \\ \log_g C_5 = r_1'(a + \triangle) + r_2' k(b + \triangle) + t r_1'(\alpha + \triangle) + t r_2' k(\beta + \triangle) \end{cases}.$$

where $w = \log_g h$.

These equations are linearly independent as long as $k^2(r_1 - r_2)(r_1' - r_2')(t - t^*) \neq 0$, so algorithm $\mathcal{A}$ can be used to guess (a, b, α, β). Therefore, the probability that the decryption algorithm accepts the first invalid ciphertexts is at most $1/p$.

- $(C_1, C_2, C_3, C_4) = (C_1^*, C_2^*, C_3^*, C_4^*)$, $t = t^*$ but $C_5 \neq C_5^*$, In this case, if the RKA decryption algorithm accepts the invalid ciphertext, we obtain the following equations.

$$\begin{cases} \log_g u_2 = a + kb \\ \log_g u_3 = \alpha + k\beta \\ \log_g C_5^* = r_1 a + r_2 kb + t^* r_1 \alpha + t^* r_2 k\beta \\ \log_g C_5 = r_1(a + \triangle) + r_2 k(b + \triangle) + t^* r_1(\alpha + \triangle) + t^* r_2 k(\beta + \triangle) \\ \qquad\qquad - r' w \triangle (a + \triangle + t^*(\alpha + \triangle) + b + \triangle + t^*(\beta + \triangle)) \end{cases}.$$

where $w = \log_g h$.

These equations are linearly independent as long as $\triangle \neq 0$, which is ruled out by the definition of CC-RKA security, so algorithm $\mathcal{A}$ can be used to guess (a, b, α, β).

For all the subsequent invalid decryption queries, the above analysis holds except that each time the RKA decryption oracle rejects an invalid ciphertext algorithm $\mathcal{A}$ can rule out one more value of (a, b, α, β).

Lemma 3. *If (g, f, Z_1, Z_2) is a non-DH instance and the RKA decryption algorithm rejects all invalid ciphertexts, then algorithm $\mathcal{A}$ has a negligible advantage in outputting the bit $d' = d$.*

Proof. We prove this lemma by considering the distribution of $(x, y, \gamma) \in Z_p$ from the view of algorithm $\mathcal{A}$. Algorithm $\mathcal{A}$ is given the public key $PK = (g, h, f, u_1, u_2, u_3, v)$, such that algorithm $\mathcal{A}$'s point of view, (x, y, γ) is uniformly random subject to $\log_g u_1 = x + ky$ where $k = \log_g f$ and $\log_g v = k'\gamma$ where $k' = \log_g h$. We suppose that the RKA decryption algorithm rejects all invalid ciphertexts, and note that by querying valid ciphertexts to the RKA decryption oracle, algorithm $\mathcal{A}$ does not learn any more information about (x, y, γ) except the relations of the constraint $\log_g u_1 = x + ky$ and $\log_g v = k'\gamma$.

Hence, algorithm $\mathcal{A}$ cannot learn any information about (x, y, γ) through the RKA decryption queries.

Let $C_1 = Z_1 v^{r'}$, $C_2 = Z_2 v^{r'}$, $C_3 = h^{r'}$. Note that as long as $k'k(r_1 - r_2) \neq 0$,

$$\begin{cases} \log_g u_1 = x + ky \\ \log_g v = k'\gamma \\ \log_g {Z_1}^x {Z_2}^y = r_1 x + k r_2 y \end{cases}$$

are linearly independent. In the following, we consider two cases.

- $\phi(SK) = SK$ and $(C_1, C_2, C_3, C_4, C_5) = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$. In this case, from the definition of the CC-RKA security game, such queries will be ruled out, therefore the RKA decryption algorithm outputs $\perp$ with noticeable probability.
- $\phi(SK) \neq SK$ and $(C_1, C_2, C_3, C_4) = (C_1^*, C_2^*, C_3^*, C_4^*)$. If the verification of C_5 on (C_1, C_2, C_3, C_4) with $\phi(SK)$ fails, the RKA decryption algorithm outputs $\perp$. Otherwise, the RKA decryption algorithm responds as

$$\begin{aligned} M' &= C_4^* \cdot (C_1^* \cdot {C_3^*}^{-\gamma-\triangle})^{-x-\triangle} \cdot (C_2^* \cdot {C_3^*}^{-\gamma-\triangle})^{-y-\triangle} \\ &= M_d \cdot g^{-r\cdot\triangle} \cdot h^{r'\cdot\triangle\cdot(x+\triangle)} \cdot f^{-r\cdot\triangle} \cdot h^{r'\cdot\triangle\cdot(y+\triangle)} \\ &= M_d \cdot g^{-r\cdot\triangle} \cdot f^{-r\cdot\triangle} \cdot h^{r'\cdot\triangle\cdot(x+y+\triangle+\triangle)}. \end{aligned}$$

We can see that even the all the ciphertexts submitted to RKA.Decrypt oracle are exactly the same as the challenge ciphertext, algorithm $\mathcal{A}$ procures nothing about (x, y, γ) from the RKA decryption queries under $(x + \triangle, y + \triangle, \gamma + \triangle)$, as long as $(x + \triangle, y + \triangle, \gamma + \triangle) \neq (x, y, \gamma)$. On the one hand, without (x, y, γ), algorithm $\mathcal{A}$ fails to compute $d' = d$ under the modified secret keys $(x + \triangle, y + \triangle, \gamma + \triangle)$. Therefore algorithm $\mathcal{A}$'s probability in outputting the bit $d' = d$ is $1/2$.

Lemma 2 makes sure that as long as the event Failure does not happen, the RKA decryption algorithm rejects all invalid ciphertexts except with a negligible probability. Lemma 3 proves that as long as the RKA decryption algorithm rejects all the invalid ciphertexts, algorithm $\mathcal{A}$ has a negligible advantage in outputting the bit $d' = d$. Therefore, we can say that algorithm $\mathcal{A}$'s probability in outputting the bit $d' = d$ is $1/2$.

To sum up, we can see that if (g, f, Z_1, Z_2) is a DH tuple, algorithm $\mathcal{A}$ wins the CC-RKA game with the probability $1/2 + \epsilon$, such that algorithm $\mathcal{B}$'s probability in solving the decisional DH problem is $1/2 + \epsilon$; if (g, f, Z_1, Z_2) is a non-DH tuple, algorithm $\mathcal{A}$ wins the CC-RKA game with the probability $1/2$, such that algorithm $\mathcal{B}$'s probability in solving the decisional DH problem is $1/2$. Denote by $\mathcal{B}(g, f, Z_1, Z_2) = 1$ the event that algorithm $\mathcal{B}$ solves the decisional DH problem. Hence, algorithm $\mathcal{B}$ has a non-negligible probability

$$\Pr[\mathcal{B}(g, f, Z_1, Z_2) = 1] = 1/2 \cdot (1/2 + \epsilon) + 1/2 \cdot 1/2 = 1/2 + \epsilon/2$$

of solving the decisional DH problem.

This concludes the proof of Theorem 1.

4.4 Efficiency

We compare Wee's CC-RKA secure public-key encryption scheme from factoring, from BDH, from DDH with weaker security and ours from DDH in Table 1.

In this table, "Pairing-E" means the sum of paring computation executed during the encryption phase, and "Pairing-D" means the sum of paring computation executed during the decryption phase. "Ex-E" means the the sum of exponentiation computation executed during the encryption phase, "Ex-D" means the the sum of exponentiation computation executed during the decryption phase.

Table 1. Comparison between public-key encryption schemes with CC-RKA security

Scheme	Ciphertext Size	Pairing-E	Pairing-D	Ex-E	Ex-D
Factoring[20]	6	0	0	9	7
BDH[20]	6	1	3	7	5
DDH[20]	7	0	0	9	9
Ours	5	0	0	7	5

5 Conclusions

Followed the work in [4], Wee [20] proposed the first public-key encryption scheme against related-key attacks via adaptive trapdoor relations [19] while paying a small overhead in efficiency, of which the existing public-key set-ups can be maintained without changing. In the constructions of [20], to make sure the efficiency of the specific constructions, Wee [20] designed some efficient strong one-time signatures in their instantiations. However, though one-time signatures are easy to construct in theory, and are more efficient than full-fledged signatures, (i.e., those which are strongly unforgeable under adaptive chosen-message attack), they still have their price.

Based on a framework to enable the construction of identity-based encryption schemes that are secure under related-key attacks, Bellare, Paterson and Thomson [6] provided a framework to enable the construction of public-key encryption schemes that are secure under related-key attacks. Public-key encryption schemes in [6] are achieved in the standard model and hold CC-RKA under reasonable hardness assumptions in the standard model, but they are transformed from the identity-based encryption schemes such that pairing computation is inevitable in the efficient instantiations.

To construct an efficient public-key encryption scheme under the setting of CC-RKA security without pairings and any one-time signature schemes, in this paper, we focus on the achievement of a full fledged CCA secure public-key encryption scheme in the context of related-key attack security. After a succinct review of the security notions related to public-key encryption schemes with RKA security, we start with pointing out a simple linear related-key attack on the Cramer-Shoup basic CCA secure public-key encryption scheme [9]. Next, we propose an efficient public-key encryption scheme which is resilient against

related-key attacks from DDH, which is in fact a variant of the Cramer-Shoup public-key encryption scheme [9]. Finally, we prove its CC-RKA security under the difficulty of solving the DDH problem.

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
2. Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: ICS. Tsinghua University Press (2011)
3. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)
4. Bellare, M., Cash, D., Miller, R.: Cryptography secure against related-key attacks and tampering. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 486–503. Springer, Heidelberg (2011)
5. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)
6. Bellare, M., Paterson, K.G., Thomson, S.: RKA security beyond the linear barrier: IBE, encryption and signatures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 331–348. Springer, Heidelberg (2012)
7. Biham, E.: New types of cryptanalytic attacks using related keys. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 398–409. Springer, Heidelberg (1994)
8. Boneh, D.: The decision diffie-hellman problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
9. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. IACR Cryptology ePrint Archive, 2001:108 (2001)
10. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptology 9(1), 35–67 (1996)
11. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (2004)
12. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)
13. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
14. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: USENIX Security Symposium, pp. 45–60. USENIX Association (2008)
15. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)

16. Lucks, S.: Ciphers secure against related-key attacks. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 359–370. Springer, Heidelberg (2004)
17. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
18. Quisquater, J.-J., Samyde, D.: ElectroMagnetic analysis (EMA): Measures and counter-measures for smart cards. In: Attali, S., Jensen, T. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001)
19. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)
20. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 262–279. Springer, Heidelberg (2012)