University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Anonymous signcryption against linear related-key attacks

Hui Cui

University of Wollongong, hc892@uowmail.edu.au

Yi Mu

University of Wollongong, ymu@uow.edu.au

Man Ho Au

University of Wollongong, aau@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Anonymous signcryption against linear related-key attacks

Abstract

A related-key attack (RKA) occurs 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. In this paper, we concentrate on the security of anonymous signcryption schemes under related-key attacks, in the sense that a signcryption system should contain no information that identifies the sender of the signcryption and the receiver of the message, and yet be decipherable by the targeted receiver. To achieve this, we consider our anonymous signcryption scheme being semantically secure against chosen ciphertext and related-key attacks (CC-RKA), existentially unforgeable against chosen message and related-key attacks (CM-RKA), and anonymous against chosen ciphertext and related-key attacks (ANON-RKA). Specifically, we require that an anonymous signcryption scheme remains secure even when an adversary is allowed to access the signcryption oracle and the designcryption oracle on linear shifts of the private keys of the sender and the receiver, respectively. After reviewing some basic definitions related to our construction, based on the existing work on cryptographic primitives in the setting of related-key attacks, we give a concrete anonymous signcryption scheme from BDH which achieves CC-RKA security, CM-RKA security, ANON-RKA security in the random oracle model.

Keywords

signcryption, against, linear, related, key, anonymous, attacks

Disciplines

Engineering | Science and Technology Studies

Publication Details

Cui, H., Mu, Y. & Au, M. (2013). Anonymous signcryption against linear related-key attacks. *Lecture Notes in Computer Science*, 8209 165-183.

Anonymous Signcryption Against 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. A related-key attack (RKA) occurs 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. In this paper, we concentrate on the security of anonymous signcryption schemes under related-key attacks, in the sense that a signcryption system should contain no information that identifies the sender of the signcryption and the receiver of the message, and yet be decipherable by the targeted receiver. To achieve this, we consider our anonymous signcryption scheme being semantically secure against chosen ciphertext and related-key attacks (CC-RKA), existentially unforgeable against chosen message and related-key attacks (CM-RKA), and anonymous against chosen ciphertext and related-key attacks (ANON-RKA). Specifically, we require that an anonymous signcryption scheme remains secure even when an adversary is allowed to access the signcryption oracle and the designcryption oracle on linear shifts of the private keys of the sender and the receiver, respectively. After reviewing some basic definitions related to our construction, based on the existing work on cryptographic primitives in the setting of related-key attacks, we give a concrete anonymous signcryption scheme from BDH which achieves CC-RKA security, CM-RKA security, ANON-RKA security in the random oracle model.

Keywords: Signcryption, CC-RKA, CM-RKA, Anonymity.

1 Introduction

In recent decades, physical attacks like side channel attacks [29] that exploit information leakage from the implementation of an algorithm are becoming increasingly popular and come in a large variety, 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 some of these situations, the adversary might get some partial information about private key through certain physical methods, which are referred to as key-leakage attacks. However, such attacks are not anticipated by the designer of the system and, correspondingly, not taken into account when arguing its security. Since modern notions of security, such as semantic security [17] and CCA security [30]

in encryption systems, is formulated in a very desired way that the adversary can fully control almost all aspects of the system (that is, the adversary is able to encrypt messages and decrypt ciphertexts at its choice), but have no access to the private keys of the entities in the communication. Unfortunately, this assumption is too ideal to satisfy in the above scenarios.

To achieve such security requirements, it requires to capture security under the context where some information of the private key are leaked to the adversary. In this paper, we consider a special case of such attacks, where 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, called related-key attack (RKA) [16, 8]. 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 a cryptographic system by invoking it with several private keys satisfying some known relations.

Although the RKA security has been achieved in various cryptographic primitives, there are few considering anonymity, which requires that the identities of participants should not be leaked during the communication [1]. With this in mind, in this work, we propose an approach for anonymous signcryption secure against related-key attacks. Suppose the signcryption system is composed of algorithms, public parameters, as well as private and public key pairs of the sender and the receiver respectively, of which the private and public keys are subject to related-key attacks, and the public parameters are system-wide, i.e., they are set beforehand and independent of users. In a protocol run, all these parameters are possible to be tampered when distributed via a public channel.

For an anonymous signcryption system, the designcryption needs the private key of the receiver while the signcryption needs the private key of the sender, hence we consider related-key attacks on private keys of both sides: chosen ciphertext attack security under related-key attack (CC-RKA), chosen message attack security under related-key attack (CM-RKA), as well as anonymity under chosen ciphertext attack and related-key attack (ANON-RKA). The designcryption oracle is forbidden when the signcryption is equal to the challenged signcryption and the derived receiver's private key matches the original one. Also, the signcryption oracle will not be executed if the given plaintext is equal to the challenged plaintext and the derived sender's private key matches the original one. Note that we define our model on the basis of the definitions in [7, 8, 33].

To begin with, we need to solve a problem how to designcryption a signcryption C with the private key $\phi(sk_R)$, where ϕ denotes a linear shift. This can be achieved with key homomorphism [33], which can reduce a signcryption scheme against related-key attacks with chosen ciphertext attack security and anonymity to a general chosen ciphertext attack secure and anonymous signcryption scheme with additional properties that the designcryption of a signcryption C with the private key $\phi(sk_R)$ equals the designcryption of another signcryption C' with the original private key sk_R . To consider the security one step further, key homomorphism fails when the signcryption C' equals the challenge signcryption

in the chosen ciphertext attack security and anonymity games. Anyway, with the adaptive trapdoor relations mentioned in [22, 32, 33], this event will never happen, which can simply formulate that the challenge signcryption is an invalid signcryption for any receiver's private key $sk'_R \neq sk_R$, such that a valid signcryption with the public parameters decides a consistent private key uniquely. Next, we should consider to signcrypt a plaintext m with the private key $\phi(sk_S)$, where ϕ denotes a linear shift, yet the case where the plaintext m in the signcryption C' equals the plaintext in the output signcryption where $\phi(sk_S) \neq sk_S$. We adopt a collision resistant hash function in the signcryption, which disables the adversary to output a valid signcryption for any sender's private key $sk'_S \neq sk_S$. In this way, a valid signcryption with public parameters can only be constructed by a correct private key.

1.1 Related Work

Signcryption, introduced by Zheng [35] in 1997, is a cryptographic primitive "Signcryption" to combine the functions of digital signature and encryption in a single step with a cost lower than that required by signature-then-encryption approach. In 2002, Baek, Steinfeld, and Zheng [5] formalized and defined security notions for signcryption, which are similar to the chosen ciphertext attack security and chosen message attack security. The notion was first defined by Jee Hea An, Yevgeniy Dodis, Tal Rabin [3], where an adversary not only access the public keys of both the sender and the receiver but also know the private key of the sender, which later was extended to the security properties of signcryption [28, 25]. Malone-Lee [28] proposed the first identity-based signcryption scheme, and claimed that their scheme achieves both privacy and unforgeability. Libert and Quisquater [25] pointed out that the scheme in [28] is not semantically secure in privacy as the signature of the message is not hidden in the signcrypted message, and proposed a signcryption scheme with ciphertext anonymity [26] based on gap Diffie-Hellman assumption, but Yang, Wong and Deng [34] found that it is not secure. Chow et al. [15] designed an identity-based signcryption scheme with public verifiability and forward security. Concurrently, Boyen [14] extended the security model in [28] via adding three new security notions: ciphertext unlinkability, ciphertext authentication and ciphertext anonymity. In addition, there are also some works concentrating on efficiency [6, 23, 24]. Barreto et al. [6] constructed an identity-based signcryption scheme which greatly improves the efficiency. Chung et al. [23] described a key privacy preserving signcryption scheme with high efficiency and simple design, and then they extended it to a ring signcryption scheme based on the technique due to Boneh et al. [13].

In 2004, Micali and Reyzin [29] put forward a comprehensive framework for modeling security against side-channel attacks, which relies on the assumption that there is no leakage of information in absence of computation. Halderman et al. [19] in 2008 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 [2]. 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 [11]. Bellare and Kohno [9] 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 [9], Lucks [27] 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, in 2010, Bellare and Cash [7] provided the first constructions to create related-secret pseudorandom bits. On the basis of the work in [7], Applebaum, Harnik, and Ishai [4] put forward some RKA secure symmetric encryption schemes, which can be used in garbled circuits in secure computation. In [8], Bellare, Cash and Miller found the 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. So far, efforts have been made to achieve RKA security about cryptographic systems such as signatures [18, 10], CCA secure public-key encryption [33, 10], identity-based encryption [10], in the setting of related-key deriving function being a class of constant functions, linear functions, affine functions, and polynomial functions.

The remainder of this paper is organized as follows. In Section 2, we briefly present the concepts associated to this work and our defined security model of RKA secure signcryption. In Section 3, we review the bilinear pairs and the complexity assumptions. In Section 4, we propose a specific construction of RKA secure signcryption from BDH, and prove its security in the random oracle model. Finally, we conclude this paper in Section 5.

2 Preliminaries

Firstly, we briefly describe the framework of signcryption, and some concepts related to RKA security. Then we details the security definitions of signcryption schemes with anonymity in the setting of related-key attacks.

2.1 Signcryption

Let $\mathcal{M}$ be the message space. An signcryption scheme is composed of the following four algorithms [24]: Setup, Keygen, Signcrypt, Designcrypt.

- Setup(1^λ) $\rightarrow$ $params$: Taking a security parameter λ as input, this algorithm outputs the public parameters $params$.
- Keygen($1^\lambda, params$) $\rightarrow (sk_R, pk_R), (sk_S, pk_S)$: Taking a security parameter λ and the public parameters $params$ as input, this algorithm outputs two private and public key pairs $(sk_R, pk_R), (sk_S, pk_S)$.
- Signcrypt($1^\lambda, params, m, sk_S, pk_R$) $\rightarrow C$: Taking a security parameter λ , the public parameters $params$, a plaintext $m \in \mathcal{M}$, the private key sk_S and the public key pk_R as input, this algorithm outputs a signcryption C .

- $\text{Designcrypt}(1^\lambda, \text{params}, C, sk_R, pk_S) \rightarrow m/\perp$: Taking a security parameter λ , the public parameters params , a signcryption C , the private key sk_R , and the public key pk_S as input, this algorithm first computes a message and signature pair (m, σ) with sk_R , and checks its validity with pk_S . Then it outputs either $m \in \mathcal{M}$ for a valid signcryption, or $\perp$ in case of a invalid signcryption.

We require that a signcryption system is correct, meaning that if $\text{params} \leftarrow \text{Setup}(1^\lambda)$, $(sk_R, pk_R), (sk_S, pk_S) \leftarrow \text{Keygen}(1^\lambda, \text{params})$ and $C \leftarrow \text{Signcrypt}(1^\lambda, \text{params}, m, sk_S, pk_R)$, then $m \leftarrow \text{Designcrypt}(1^\lambda, \text{params}, C, sk_R, pk_S)$.

2.2 RKA Security

Related-key deriving functions. Our definition follows the notion of related-key deriving functions given in [9]. Briefly speaking, a class Φ of related-key deriving functions $\phi: sk_u \rightarrow sk_u$ 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 this paper, we only consider the class Φ^+ as linear shifts.

The family Φ^+ . Any function $\phi: Z_q^* \rightarrow Z_q^*$ in this class is indexed by $\Delta \in Z_q^*$, where $\phi_\Delta(sk_u) := sk_u + \Delta$.

Informally, we consider a secure anonymous signcryption scheme against related-key attacks to be semantically secure against chosen ciphertext and related-key attacks (CC-RKA), existentially unforgeable against chosen message and related-key attacks (CM-RKA), and anonymous against related-key attacks in the sense that a signcryption should contain no information that identifies the sender of the signcryption and the receiver of the message (ANON-RKA), and yet be decipherable by the targeted receiver.

CC-RKA Security. A signcryption scheme is semantically secure against chosen ciphertext and related-key attacks (CC-RKA security) if no probabilistic polynomial-time adversary has a non-negligible advantages in the following game.

- Initialization. The challenger algorithm $\mathcal{B}$ runs $\text{params} \leftarrow \text{Setup}(1^k)$, and $(sk_R, pk_R), (sk_S, pk_S) \leftarrow \text{Keygen}(1^\lambda, \text{params})$. Algorithm $\mathcal{B}$ gives the public parameters params , the private and public key pair (sk_S, pk_S) , and the public key pk_R to the adversary algorithm $\mathcal{A}$.
- Phase 1. Algorithm $\mathcal{A}$ issues a series of queries to RKA.Designcrypt oracle. On input a signcryption C , and a related-key deriving function $\phi \in \Phi$, algorithm $\mathcal{B}$ runs $(m, \sigma) \leftarrow \text{Designcrypt}(1^\lambda, \text{params}, C, \phi(sk_R))$, and sends (m, σ) to algorithm $\mathcal{A}$. Note that as sk_S is given to algorithm $\mathcal{A}$, we remove the queries to RKA.Signcrypt oracle.
- Challenge. Algorithm $\mathcal{A}$ outputs two messages $M_0^*, M_1^* \in \mathcal{M}$, $|M_0^*| = |M_1^*|$, on which it wishes to be challenged. Algorithm $\mathcal{B}$ chooses a random $d \in \{0, 1\}$, and runs $C^* \leftarrow \text{Signcrypt}(1^\lambda, \text{params}, m_d, sk_S, pk_R)$. Algorithm $\mathcal{B}$ sends C^* as the designcryption to algorithm $\mathcal{A}$.

- Phase 2. Algorithm $\mathcal{A}$ continues to adaptively issue queries to RKA.Designcrypt oracle. On input a signcryption C , and a related-key deriving function $\phi \in \Phi$, with the constraint $(\phi(sk_R), C) \neq (sk_R, C^*)$, algorithm $\mathcal{B}$ responds as in Phase 1.
- Output. Algorithm $\mathcal{A}$ outputs its guess $d' \in \{0, 1\}$ for d and wins the game if $d' = d$.

We define algorithm $\mathcal{A}$'s advantage in this game to be

$$\text{Adv}_{\mathcal{A}}^{\text{CC-RKA}}(\lambda) \stackrel{\text{def}}{=} |\Pr[d = d'] - 1/2|.$$

CM-RKA Security. A signcryption scheme is existentially unforgeable against chosen message and related-key attacks (CM-RKA security) if no probabilistic polynomial-time adversary has a non-negligible advantages in the following game.

- Initialization. The challenger algorithm $\mathcal{B}$ runs $params \leftarrow \text{Setup}(1^k)$, and $(sk_R, pk_R), (sk_S, pk_S) \leftarrow \text{Keygen}(1^\lambda, params)$. Algorithm $\mathcal{B}$ gives the public parameters $params$, the private and public key pair (sk_R, pk_R) , and the public key pk_S to the adversary algorithm $\mathcal{A}$.
- Phase 1. Algorithm $\mathcal{A}$ issues a series of queries to RKA.Signcrypt oracle. On input a message $m \in \mathcal{M}$, and a related-key deriving function $\phi \in \Phi$, algorithm $\mathcal{B}$ runs $C \leftarrow \text{Signcrypt}(1^\lambda, params, m, \phi(sk_S), pk_R)$, and sends C to algorithm $\mathcal{A}$. Note that as sk_R is given to algorithm $\mathcal{A}$, we remove the queries to RKA.Designcrypt oracle.
- Output. Algorithm $\mathcal{A}$ outputs a signcryption C^* , and wins the game if $(m^*, \sigma^*) \leftarrow \text{Designcrypt}(1^\lambda, params, C^*, sk_R)$, and $true \leftarrow \text{Verify}(1^\lambda, params, m^*, \sigma^*)$.

ANON-RKA Security. A signcryption scheme is anonymous against chosen ciphertext and related-key attacks (ANON-RKA security) if no probabilistic polynomial-time adversary has a non-negligible advantages in the following game.

- Initialization. The challenger algorithm $\mathcal{B}$ runs $params \leftarrow \text{Setup}(1^k)$, and $(sk_{R,0}, pk_{R,0}), (sk_{S,0}, pk_{S,0}) \leftarrow \text{Keygen}(1^\lambda, params)$, $(sk_{R,1}, pk_{R,1}), (sk_{S,1}, pk_{S,1}) \leftarrow \text{Keygen}(1^\lambda, params)$, respectively. Algorithm $\mathcal{B}$ gives the public parameters $params$, the private and public key pairs $(sk_{S,0}, pk_{S,0}), (sk_{S,1}, pk_{S,1})$, and the public keys $pk_{R,0}, pk_{R,1}$ to the adversary algorithm $\mathcal{A}$.
- Phase 1. Algorithm $\mathcal{A}$ issues a series of queries to RKA.Designcrypt oracle. On input $sk_S \in \{sk_{S,0}, sk_{S,1}\}$, $pk_R \in \{pk_{R,0}, pk_{R,1}\}$, a signcryption C , and a related-key deriving function $\phi \in \Phi$, algorithm $\mathcal{B}$ runs $(m, \sigma) \leftarrow \text{Designcrypt}(1^\lambda, params, C, \phi(sk_R))$, and sends (m, σ) to algorithm $\mathcal{A}$. Note that as $sk_{S,0}, sk_{S,1}$ are given to algorithm $\mathcal{A}$, we remove the queries to RKA.Signcrypt oracle.
- Challenge. Algorithm $\mathcal{A}$ outputs a message $M^* \in \mathcal{M}$ on which it wishes to be challenged. Algorithm $\mathcal{B}$ chooses random $d, e \in \{0, 1\}$, and runs $C^* \leftarrow \text{Signcrypt}(1^\lambda, params, m, sk_{S,d}, pk_{R,e})$. Algorithm $\mathcal{B}$ sends C^* as the designcryption to algorithm $\mathcal{A}$.

- Phase 2. Algorithm $\mathcal{A}$ continues to adaptively issue queries to RKA.Designcrypt oracle. On input $sk_S \in \{sk_{S,0}, sk_{S,1}\}$, $pk_R \in \{pk_{R,0}, pk_{R,1}\}$, a signcryption C , and a related-key deriving function $\phi \in \Phi$, with the constraint $(\phi(sk_{R,d}), C) \neq (sk_{R,d}, C^*)$, algorithm $\mathcal{B}$ responds as in Phase 1.
- Output. Algorithm $\mathcal{A}$ outputs its guess $d', e' \in \{0, 1\}$ for d, e , and wins the game if $d' = d$ and $e' = e$.

We define algorithm $\mathcal{A}$'s advantage in this game to be

$$\text{Adv}_{\mathcal{A}}^{\text{ANON-RKA}}(\lambda) \stackrel{\text{def}}{=} |\Pr[d = d' \wedge e = e'] - 1/4|.$$

3 Bilinear Maps and Complexity Assumptions

In this section, we review a few facts related to groups with efficiently computable bilinear maps, and the security assumptions that our new schemes based on.

3.1 Bilinear Maps

Let G and G_T be two multiplicative cyclic groups of prime order q . Let g be a generator of G , and $\hat{e} : G \times G \rightarrow G_T$ be a bilinear map with the following properties [12, 20, 21]: (1) Bilinear: for all $g \in G$ and $a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$; (2) Non-degenerate: $\hat{e}(g, g) \neq 1$.

We say that G is a bilinear group if the group action in G can be computed efficiently and there exists a group G_T and an efficiently computable bilinear map $\hat{e} : G \times G \rightarrow G_T$ as above.

3.2 Complexity Assumptions

Computational DL. The computational Discrete Logarithm (DL) problem is that for any probabilistic polynomial-time algorithm, it is difficult to compute b given (g, g^b) , where $g \in G$, $b \in \mathbb{Z}_q^*$ are chosen independently and uniformly at random.

Computational BDH. The computational bilinear Diffie-Hellman (BDH) problem is that for any probabilistic polynomial-time algorithm, it is difficult to compute $\hat{e}(g, g)^{abc}$ given (g, g^a, g^b, g^c) , where $g \in G$, $a, b, c \in \mathbb{Z}_q^*$ are chosen independently and uniformly at random.

Decisional BDH. The decisional bilinear Diffie-Hellman (BDH) problem is that for any probabilistic polynomial-time algorithm, it is difficult to distinguish $(g, g^a, g^b, g^c, \hat{e}(g, g)^{abc})$ from (g, g^a, g^b, g^c, Z) , where $g \in G$, $Z \in G_T$, $a, b, c \in \mathbb{Z}_q^*$ are chosen independently and uniformly at random.

4 Anonymous Signcryption from RKA Security

In this section, we propose a specific anonymous signcryption scheme in the setting of related-key attacks, and analyze its CC-RKA, CM-RKA and ANON-RKA security.

4.1 Techniques in Our Solution

To achieve key homomorphism [33], we make use of a class of functions with an additional input (namely tag), called adaptive trapdoor relations [22, 32], which is easy to compute and invert with tag , but hard to invert without tag .

More specifically, our adaptive trapdoor relation F_{pk_u} satisfies the following features.

- Generation. This is a randomized algorithm G that outputs a pair (pk_u, sk_u) on input a security parameter λ .
- Sampling. On input pk_u and tag , this randomized algorithm F outputs $(\theta, F_{pk_u}(tag, \theta))$ for a random θ .
- Inversion. For all tag, y and (pk_u, sk_u) , this efficient algorithm F' computes $F'(sk_u, tag, y) = F_{pk_u}^{-1}(tag, y)$.
- One-wayness. For a stateful adversary $\mathcal{A}$, it holds that

$$\Pr \left[\theta = \theta' \mid \begin{array}{l} tag^* \leftarrow \mathcal{A}(1^\lambda). \\ (pk_u, sk_u) \leftarrow G(1^\lambda). \\ (\theta, y) \leftarrow F(pk_u, tag^*). \\ \theta' \leftarrow \mathcal{A}^{F_{pk_u}^{-1}(\cdot, \cdot)}(pk_u, y). \end{array} \right]$$

is a negligible function in λ , where adversary $\mathcal{A}$ is allowed to query $F_{pk_u}^{-1}(\cdot, \cdot)$ on any tag different from tag^* .

Key Homomorphism. Let Φ be a set of related-key deriving functions. We say that F_{pk_u} is Φ -key homomorphic if there is a probabilistic polynomial-time algorithm T such that $F'(\phi(sk_u), tag, y) = F'(sk_u, tag, T(\phi, tag, y))$ holds with overwhelming probability for all $\phi \in \Phi$, sk_u , tag and y .

4.2 Construction

Let $\hat{e} : G \times G \rightarrow G_T$ be a bilinear map over a bilinear group G of prime order q with a generator $g \in G$. The scheme is described as follows.

- Setup. To generate the system public parameters, this algorithm works as follows.
 1. Chooses random $\beta, \gamma \in Z_q^*$, and computes $g_1 = g^\beta, g_2 = g^\gamma$.
 2. Chooses collision resistant hash functions $H_0 : G^2 \rightarrow G, H : G^2 \rightarrow Z_q^*, H' : G^5 \times G_T^2 \rightarrow Z_q^*$.
 3. Outputs the public parameters $(g, g_1, g_2, H_0, H, H')$.

- Keygen. To generate two private and public key pairs for receiver R and sender S respectively, the system chooses random $x_R, x_S \in Z_q^*$ as the private keys, and computes $Y_R = g^{x_R}, Y_S = g^{x_S}$ as the public keys.
- Signcrypt. To signcrypt a message $m \in G_T$ for receiver R , sender S runs as follows.
 1. Chooses a random $r \in Z_q^*$, and computes $\mu = g^r, \theta = g_1^r$.
 2. Chooses a random $e \in Z_q^*$, and computes $tag = g^e$.
 3. Computes $\psi = \hat{e}(\theta, g_2) \cdot m$, and

$$\begin{aligned}\tau &= (Y_R \cdot g_1^{H(\mu, tag)})^r \cdot H_0(\mu, Y_R^r), \\ \sigma &= e - x_S \cdot H'(\mu, \tau, \psi, Y_R^r, Y_R^{x_S}, tag, m),\end{aligned}$$

4. Outputs the signcryption $C = (\mu, \tau, \psi, tag, \sigma)$.
- Designcrypt. To designcrypt a signcrypton C from sender S , receiver R runs as follows.
 1. Computes θ as $\theta = (\frac{\tau}{H_0(\mu, \mu^{x_R})} \cdot \mu^{-x_R})^{\frac{1}{H(\mu, tag)}}$. If $\hat{e}(\theta, g) = \hat{e}(\mu, g_1)$, computes $m = \psi / \hat{e}(\theta, g_2)$, and outputs $(\mu, \tau, \psi, m, tag, \sigma)$. Otherwise, it outputs $\perp$.
 2. Check the validity of σ via $tag = g^\sigma \cdot Y_S^{H'(\mu, \tau, \psi, \mu^{x_R}, Y_S^{x_S}, tag, m)}$. If the equation holds, it outputs m . Otherwise, it outputs $\perp$.

4.3 Proof of Security

We analyze the security of our proposed signcryption scheme against related-key attacks by reducing its CC-RKA security, CM-RKA security, and ANON-RKA security under the security games defined in Section 2.

Theorem 1. *Assume that the decisional BDH assumption holds in G, G_T , the computational BDH problem holds in G, G_T , then our signcryption scheme is CC-RKA secure regarding linear related-key deriving functions ϕ^+ in the random oracle model.*

Let $(\mu^*, \tau^*, \psi^*, tag^*, \sigma^*)$ be the challenge signcryption of the message M_d given to algorithm $\mathcal{A}$ by algorithm $\mathcal{B}$. Denote Failure by the event that algorithm $\mathcal{A}$ issues $(\mu^*, \tau^*, \psi^*, Y_1, Y_2, M_0)$ or $(\mu^*, \tau^*, \psi^*, Y_1, Y_2, M_1)$ to random oracle H' , and (μ, Y_1) to random oracle H_0 , where $\hat{e}(Y_1, g) = \hat{e}(Y_R, \mu^*)$.

In what follows we prove that if the event Failure does not occur, then our signcryption scheme is CC-RKA secure. We conclude this proof by showing that the event Failure has a negligible probability to occur.

Lemma 1. *If the decisional BDH assumption holds in G, G_T , and the event Failure does not happen, then our signcryption scheme is CC-RKA secure.*

Proof. Suppose that algorithm $\mathcal{A}$ is an adversary algorithm against the CC-RKA security of our signcrypton scheme, then we can construct a challenger algorithm $\mathcal{B}$ that solves the decisional BDH problem, which is given input a BDH instance

(g, g^a, g^b, g^c, Z) and outputs 1 (Z is $\hat{e}(g, g)^{abc}$) or 0 (Z is a random element in G_T).

Initialization. To simulate the system parameters, algorithm $\mathcal{B}$ runs as follows.

1. Chooses a collision resistant hash function $H : G^2 \rightarrow Z_q^*$.
2. Chooses a random $e^* \in Z_q^*$, computes $\text{tag}^* = g^{e^*}$.
3. Chooses a random $x_S \in Z_q^*$, computes $Y_S = g^{x_S}$.
4. Chooses a random $x_r \in Z_q^*$, computes $Y_R = (g^b)^{-H(g^c, \text{tag}^*)} g^{x_r}$. Note that $x_R = \log_g Y_R = -b \cdot H(g^c, \text{tag}^*) + x_r$ is unknown to algorithm $\mathcal{B}$.
5. Sends the public parameters $(g, g_1, g_2, H_0, H, H')$ of which $g_1 = g^b, g_2 = g^a, H_0, H'$ are the random oracles controlled by algorithm $\mathcal{B}$, receiver R 's public key Y_R , and sender S 's public and private key pair (x_S, Y_S) to algorithm $\mathcal{A}$.

H_0 -query. At any time algorithm $\mathcal{A}$ can query the random oracle on (μ, Y_1) . Algorithm $\mathcal{B}$ maintains a list L_{H_0} of tuples $((\mu, Y_1), H_0(\mu, Y_1))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on Y , algorithm $\mathcal{B}$ responds as follows.

- If (μ, Y_1) already appears in list L_{H_0} , algorithm $\mathcal{B}$ responds with $H_0(\mu, Y_1)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $t_i \in Z_q^*$, sets $H_0(\mu, Y_1) = t_i$, and adds $((\mu, Y_1), t_i)$ to list L_{H_0} .

H' -query. At any time algorithm $\mathcal{A}$ can query the random oracle on $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$. Algorithm $\mathcal{B}$ maintains a list $L_{H'}$ of tuples $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$, algorithm $\mathcal{B}$ responds as follows.

- If $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$ already appears in list $L_{H'}$, algorithm $\mathcal{B}$ responds with $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $s_i \in Z_q^*$, sets $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m) = s_i$, sends s_i to algorithm $\mathcal{A}$, and adds $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), s_i)$ to list $L_{H'}$.

Phase 1. Algorithm $\mathcal{A}$ adaptively issues the RKA designcrypt queries to algorithm $\mathcal{B}$. For a query (C, ϕ) to RKA.Designcrypt oracle where $C = (\mu, \tau, \psi, \text{tag}, \sigma)$, algorithm $\mathcal{B}$ responds as follows.

1. Algorithm $\mathcal{B}$ computes θ' with $\phi(x_R)$. To see how algorithm $\mathcal{B}$ obtains θ without x_R , we rewrite τ such that

$$\begin{aligned} \frac{\tau}{t_i} &= (Y_R \cdot g_1^{H(\mu, \text{tag})})^r = \mu^{-b \cdot H(g^c, \text{tag}^*) + x_r + b \cdot H(\mu, \text{tag})} \\ &= (\mu^b)^{H(\mu, \text{tag}) - H(g^c, \text{tag}^*)} \cdot \mu^{x_r} = \theta^{H(\mu, \text{tag}) - H(g^c, \text{tag}^*)} \cdot \mu^{x_r} \\ \Rightarrow \theta &= \left(\frac{\tau}{t_i \cdot \mu^{x_r}} \right)^{\frac{1}{H(\mu, \text{tag}) - H(g^c, \text{tag}^*)}}. \end{aligned}$$

On the other hand,

$$\begin{aligned}\theta' &= \left(\frac{\tau}{t_i} \cdot \mu^{-(x_R + \Delta)}\right)^{\frac{1}{H(\mu, tag)}} = \left(\left(\frac{\tau}{t_i} \cdot \mu^{-\Delta}\right) \cdot \mu^{-x_R}\right)^{\frac{1}{H(\mu, tag)}} \\ &= \theta \cdot (\mu^{-\Delta})^{\frac{1}{H(\mu, tag)}}.\end{aligned}$$

Note that this reflects how key homomorphism works in the adaptive trapdoor relation [33].

2. If $\hat{e}(\theta', g) = \hat{e}(\mu, g_1)$, algorithm $\mathcal{B}$ outputs $m = \psi / \hat{e}(\theta', g_2)$. Otherwise, it outputs $\perp$.

Challenge. Algorithm $\mathcal{A}$ outputs two messages $M_0, M_1 \in G_T$ on which it wishes to be challenged. Algorithm $\mathcal{B}$ chooses random $s^*, t^* \in Z_q^*$, a random $d \in \{0, 1\}$, sets $\mu^* = g^c$, and computes

$$\tau^* = (g^c)^{x_r} \cdot t^*, \quad \psi^* = Z \cdot M_d, \quad \sigma^* = e^* - x_s \cdot s^*.$$

Algorithm $\mathcal{B}$ outputs the signcryption $C^* = (\mu^*, \tau^*, \psi^*, tag^*, \sigma^*)$, and adds $((\mu^*, \tau^*, \psi^*, Y_R^c, Y_R^{x_s}, M_d), s^*)$ to list $L_{H'}$, $((\mu, Y_R^c), t^*)$ to list L_{H_0} .

Phase 2. Algorithm $\mathcal{A}$ adaptively issues the RKA designcryption queries to algorithm $\mathcal{B}$. For a query (C, ϕ) to RKA.Designcrypt oracle where $C = (\mu, \tau, \psi, tag, \sigma)$, algorithm $\mathcal{B}$ responds as follows.

- $H(\mu, tag) \neq H(g^c, tag^*)$. Algorithm $\mathcal{B}$ responds as in Phase 1.
- $H(\mu, tag) = H(g^c, tag^*)$, and $(\mu, \tau, \psi, \sigma) \neq (\mu^*, \tau^*, \psi^*, \sigma^*)$. If algorithm $\mathcal{B}$ accepts this signcryption, it means algorithm $\mathcal{A}$ breaks the security of the CM-RKA security of our scheme, which we will analyze later. Therefore, algorithm $\mathcal{B}$ outputs $\perp$ except with negligible probability.
- $H(\mu, tag) = H(g^c, tag^*)$, and $(\mu, \tau, \psi, \sigma) = (\mu^*, \tau^*, \psi^*, \sigma^*)$ and $\phi(x_R) \neq x_R$. If algorithm $\mathcal{B}$ accepts this signcryption, it means algorithm $\mathcal{A}$ can output $\phi \in \Phi$ such that $(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-\phi(x_R)})^{\frac{1}{H(g^c, tag^*)}} \neq \perp$. That is, $\hat{e}(\theta', g) = \hat{e}(\mu, g_1)$, to guarantee this,

$$\left(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-x_R}\right)^{\frac{1}{H(g^c, tag^*)}} = \left(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-\phi(x_R)}\right)^{\frac{1}{H(g^c, tag^*)}} \Rightarrow x_R = \phi(x_R)$$

should hold. Therefore, algorithm $\mathcal{B}$ outputs $\perp$ except with negligible probability.

In fact this is the one-wayness property of the adaptive trapdoor relation, which on the other hand reflects how the key fingerprint property, which is indispensable according to the definitions given in [7, 4, 33], works in our construction.

Note that (C, ϕ) satisfying $H(\mu, tag) = H(g^c, tag^*)$, $(\mu, \tau, \psi, \sigma) = (\mu^*, \tau^*, \psi^*, \sigma^*)$ and $\phi(x_R) = x_R$, is not allowed by the definition of the CC-RKA security game.

Output. Algorithm $\mathcal{A}$ outputs a guess $d' \in \{0, 1\}$. If $d = d'$, algorithm $\mathcal{A}$ wins the game, and algorithm $\mathcal{B}$ outputs 1 indicating $Z = \hat{e}(g, g)^{abc}$. Otherwise, algorithm $\mathcal{B}$ outputs 0 indicating Z is random in G_T .

Let ϵ be the advantage that algorithm $\mathcal{A}$ breaks the CC-RKA security of the above game. We can see that if algorithm $\mathcal{B}$'s input tuple is (g, g^a, g^b, g^c, Z) where $Z = \hat{e}(g, g)^{abc}$, then algorithm $\mathcal{A}$'s view of this simulation is identical to the real attack, thus algorithm $\mathcal{A}$'s probability in outputting $d' = d$ must satisfy $\Pr[d = d'] = 1/2 + \epsilon$. On the other hand, if algorithm $\mathcal{B}$'s input tuple (g, g^a, g^b, g^c, Z) where $Z \in G_T$, then algorithm $\mathcal{A}$'s advantage is nil and thus $\Pr[d' = d] = 1/2$. To sum up, algorithm $\mathcal{B}$'s probability in solving the decisional BDH problem is

$$\Pr[\mathcal{B}(g, g^a, g^b, g^c, Z)] = 1/2 \cdot (1/2 + \epsilon) + 1/2 \cdot 1/2 = 1/2 + \epsilon/2.$$

In the following, we prove that the event Failure has a negligible probability to occur due to the security of the computational DH problem hiding in hash function H' .

Lemma 2. *If the computational BDH problem holds in G, G_T , then the event Failure happens with a negligible probability.*

Proof. Given algorithm $\mathcal{A}$ for which the event Failure happens with a noticeable probability, we construct an algorithm $\mathcal{B}'$ that solves the computational BDH problem. Specifically, we consider the following game where algorithm $\mathcal{B}'$ solves the computational BDH problem. Suppose that algorithm $\mathcal{B}'$ is given a random tuple (g, g^a, g^b, g^c) as input and outputs $\hat{e}(g, g)^{abc}$.

Initialization. The same as in Lemma 1.

H_0 -query. At any time algorithm $\mathcal{A}$ can query the random oracle on (μ, Y_1) . Algorithm $\mathcal{B}$ maintains a list L_{H_0} of tuples $((\mu, Y_1), H_0(\mu, Y_1))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on (μ, Y_1) , algorithm $\mathcal{B}$ responds as follows.

- If $\hat{e}(Y_1, g) = \hat{e}(Y_R, g^c)$, algorithm $\mathcal{B}$ solves the computational BDH problem immediately. To see this, we have

$$\begin{aligned} Y_1 &= Y_R^c = (g^{bc})^{-H(g^c, tag^*)} g^{c \cdot x_r} \\ \Rightarrow g^{bc} &= \left(\frac{Y_1}{g^{c \cdot x_r}} \right)^{-\frac{1}{H(g^c, tag^*)}} \Rightarrow \hat{e}(g, g)^{abc} = \hat{e}(g^a, g^{bc}). \end{aligned}$$

- If (μ, Y_1) already appears in list L_{H_0} , algorithm $\mathcal{B}$ responds with $H_0(\mu, Y_R^r)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $t_i \in Z_q^*$, sets $H_0(\mu, Y_1) = t_i$, and adds $((\mu, Y_1), t_i)$ to list L_{H_0} .

H' -query. At any time algorithm $\mathcal{A}$ can query the random oracle on $(\mu, \tau, \psi,$

$Y_1, Y_2, \text{tag}, m)$, Algorithm $\mathcal{B}$ maintains a list $L_{H'}$ of tuples $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$, algorithm $\mathcal{B}$ responds as follows.

- If $\hat{e}(Y_1, g) = \hat{e}(Y_R, g^c)$, the same as that in H_0 query.
- If $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$ already appears in list $L_{H'}$, algorithm $\mathcal{B}$ responds with $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $s_i \in Z_q^*$, sets $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m) = s_i$, sends s_i to algorithm $\mathcal{A}$, and adds $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), s_i)$ to list $L_{H'}$.

Phase 1. The same as in Lemma 1.

Challenge. Algorithm $\mathcal{A}$ outputs two messages $M_0, M_1 \in G_T$ on which it wishes to be challenged. Algorithm $\mathcal{B}'$ chooses random $r^*, s^*, t^* \in Z_q^*$, and computes

$$\begin{aligned}\mu^* &= g^{r^*}, \quad \tau^* = (Y_R \cdot g_1^{H(\mu^*, \text{tag}^*)})^{r^*} \cdot t^*, \\ \psi^* &= \hat{e}(g_1, g_2)^{r^*} \cdot M_d, \quad \sigma^* = e^* - x_S \cdot s^*,\end{aligned}$$

Algorithm $\mathcal{B}$ outputs the signcryption $C^* = (\mu^*, \tau^*, \psi^*, \text{tag}^*, \sigma^*)$, and adds $((\mu^*, \tau^*, \psi^*, Y_R^{r^*}, Y_R^{x_S}, \text{tag}^*, M_d), s^*)$ to list $L_{H'}$, $(Y_R^{r^*}, t^*)$ to list L_{H_0} .

Phase 2. The same as in Lemma 1.

Lemma 1 makes sure that as long as the event Failure does not happen, then our signcryption scheme preserves CC-RKA security. Lemma 2 guarantees that as long as the event Failure does not happen, algorithm $\mathcal{B}'$ is the same as algorithm $\mathcal{B}$ such that algorithm $\mathcal{A}$ cannot differentiate between algorithm $\mathcal{B}$ and algorithm $\mathcal{B}'$.

This completes the proof of CC-RKA security of our signcryption scheme.

Theorem 2. *Assume that the computational DL problem holds in G , then our signcryption scheme is CM-RKA secure regarding linear related-key deriving functions ϕ^+ in the random oracle.*

Proof. Suppose that algorithm $\mathcal{A}$ is an adversary breaks the CM-RKA security of our signcryption scheme, we construct algorithm $\mathcal{B}$ that solves the computational DL problem which is given input a random tuple (g, g^b) and outputs b .

Initialization. To simulate the system parameters, algorithm $\mathcal{B}$ runs as follows.

1. Chooses collision resistant hash functions $H_0 : G^2 \rightarrow G$, $H : G^2 \rightarrow Z_q^*$,
2. Chooses a random $a \in Z_q^*$, computes $g_2 = g^a$, and then chooses a random $x_R \in Z_q^*$, computes $Y_R = g^{x_R}$, and a random $x_S \in Z_q^*$, computes $Y_S = (g^b)^{x_S}$. Note that $x_S = \log_g Y_S = b \cdot x_S$, which is unknown to algorithm $\mathcal{B}$.
3. Sends the public parameters $(g, g_1, g_2, H_0, H, H')$ of which $g_1 = g^b$, where H' is a random oracle controlled by algorithm $\mathcal{B}$, receiver R 's public and private key pair (x_R, Y_R) , and sender S 's public key Y_S to algorithm $\mathcal{A}$.

H' -query. At any time algorithm $\mathcal{A}$ can query the random oracle on $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$. Algorithm $\mathcal{B}$ maintains a list $L_{H'}$ of tuples $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$, algorithm $\mathcal{B}$ responds as follows.

- If $(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$ already appears in list $L_{H'}$, algorithm $\mathcal{B}$ responds with $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $s_i \in Z_q^*$, sets $H'(\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m) = s_i$, sends s_i to algorithm $\mathcal{A}$, and adds $((\mu, \tau, \psi, Y_1, Y_2, \text{tag}, m), s_i)$ to list $L_{H'}$.

Phase 1. Algorithm $\mathcal{A}$ adaptively issues the RKA signcryption queries to algorithm $\mathcal{B}$. Once algorithm $\mathcal{A}$ queries (m, ϕ) to RKA.Signcrypt oracle, algorithm $\mathcal{B}$ responds as follows.

1. Chooses a random $r \in Z_q^*$, and computes $\mu = g^r$.
2. Chooses random $\sigma, s_i \in Z_q^*$, and computes $\text{tag} = g^\sigma \cdot Y_S^{s_i}$, $\tau = (Y_R \cdot g_1^{H(\mu, \text{tag})})^r \cdot H_0(\mu, Y_R^r)$, and $\psi = \hat{e}(g_1^r, g_2) \cdot m$.
3. Outputs the signcryption $C = (\mu, \tau, \psi, \text{tag}, \sigma)$, and adds $((\mu, \tau, \psi, Y_R^r, Y_R^{x_S + \Delta}, \text{tag}, m), s_i)$ to list $L_{H'}$.

Output. Algorithm $\mathcal{A}$ outputs a signcryption $C^* = (\mu^*, \tau^*, \psi^*, \text{tag}^*, \sigma^*)$, and algorithm $\mathcal{B}$ designcrypts it following the designcryption algorithm. If this is a valid signcryption, from the Forking Lemma in [31], after a polynomial replay attack of algorithm $\mathcal{A}$, we obtain two valid signcryption $(\mu^*, \tau^*, \psi^*, \text{tag}^*, \sigma^*)$ and $(\mu^*, \tau^*, \psi^*, \text{tag}^*, \sigma)$ with $s_i \neq s^*$, from which we have

$$\text{tag}^* = g^{\sigma^*} \cdot Y_S^{s^*} = g^\sigma \cdot Y_S^{s_i} \Rightarrow Y_S = g^{\frac{\sigma - \sigma^*}{s^* - s_i}} \Rightarrow b = \frac{\sigma - \sigma^*}{x_s \cdot (s^* - s_i)}.$$

That is, algorithm $\mathcal{B}$ solves the computational DL problem.

This completes the proof of CM-RKA security of our signcryption scheme.

Theorem 3. *Assume that the computational BDH assumption holds in G, G_T , then our signcryption scheme is ANON-RKA secure regarding linear related-key deriving functions ϕ^+ in the random oracle model.*

Proof. This part is similar to that of Theorem 1. Denote Failure by the event that algorithm $\mathcal{A}$ issues $(\mu^*, \tau^*, \psi^*, Y_1, Y_2, M^*)$ to random oracle H' , and (μ, Y_1) to random oracle H_0 , where $\hat{e}(Y_1, g) = \hat{e}(Y_R, \mu^*)$. We firstly prove that if the event Failure does not occur, our signcryption scheme is ANON-RKA secure; then conclude it by that the event Failure has a negligible probability to occur.

Suppose there is an adversary algorithm $\mathcal{A}$ against the anonymity of our RKA secure signcryption scheme. We construct a challenge algorithm $\mathcal{B}$ that

solves the computational BDH problem, which is given a random tuple (g, g^a, g^b, g^c) as input and outputs $Z = \hat{e}(g, g)^{abc}$.

Initialization. To simulate the system parameters, algorithm $\mathcal{B}$ runs as follows.

1. Chooses a collision resistant hash function $H : G^2 \rightarrow Z_q^*$.
2. Chooses a random $e^* \in Z_q^*$, computes $tag^* = g^{e^*}$.
3. Chooses random $x_{S,0}, x_{S,1} \in Z_q^*$, computes $Y_{S,0} = g^{x_{S,0}}, Y_{S,1} = g^{x_{S,1}}$.
4. Chooses random $x_{r,0}, x_{r,1} \in Z_q^*$, computes $Y_{R,0} = (g^b)^{-H(g^c, tag^*)} g^{x_{r,0}}, Y_{R,1} = (g^b)^{-H(g^c, tag^*)} g^{x_{r,1}}$.
5. Sends $(g, g_1, g_2, H_0, H, H', (x_{S,0}, Y_{S,0}), (x_{S,0}, Y_{S,1}), Y_{R,0}, Y_{R,1})$ to algorithm $\mathcal{A}$, where $g_1 = g^b, g_2 = g^a, H_0, H'$ are the random oracles controlled by algorithm $\mathcal{B}$.

H_0 -query. At any time algorithm $\mathcal{A}$ can query the random oracle on (μ, Y_1) . Algorithm $\mathcal{B}$ maintains a list L_{H_0} of tuples $((\mu, Y_1), H_0(\mu, Y_1))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on (μ, Y_1) , algorithm $\mathcal{B}$ responds as follows.

- If $\hat{e}(Y_1, g) = \hat{e}(Y_{R,e}, g^c)$ for $e \in \{0, 1\}$, algorithm $\mathcal{B}$ solves the computational BDH problem immediately. To see this, we have

$$\begin{aligned} Y_1 &= Y_{R,e}^c = (g^{bc})^{-H(g^c, tag^*)} g^{c \cdot x_r} \\ &\Rightarrow g^{bc} = \left(\frac{Y_1}{g^{c \cdot x_r}} \right)^{-\frac{1}{H(g^c, tag^*)}} \Rightarrow \hat{e}(g, g)^{abc} = \hat{e}(g^a, g^{bc}). \end{aligned}$$

- If (μ, Y_1) already appears in list L_{H_0} , algorithm $\mathcal{B}$ responds with $H_0(\mu, Y_1)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $t_i \in Z_q^*$, sets $H_0(\mu, Y_1) = t_i$, and adds $((\mu, Y_1), t_i)$ to list L_{H_0} .

H' -query. At any time algorithm $\mathcal{A}$ can query the random oracle on $(\mu, \tau, \psi, Y_1, Y_2, tag, m)$. Algorithm $\mathcal{B}$ maintains a list $L_{H'}$ of tuples $((\mu, \tau, \psi, Y_1, Y_2, tag, m), H'(\mu, \tau, \psi, Y_1, Y_2, tag, m))$ which is initially empty. When algorithm $\mathcal{A}$ issues a hash query on $(\mu, \tau, \psi, Y_1, Y_2, tag, m)$, algorithm $\mathcal{B}$ responds as follows.

- If $\hat{e}(Y_1, g) = \hat{e}(Y_{R,e}, g^c)$ for $e \in \{0, 1\}$, the same as that in H_0 query.
- If $(\mu, \tau, \psi, Y_1, Y_2, tag, m)$ already appears in list $L_{H'}$, algorithm $\mathcal{B}$ responds with $H'(\mu, \tau, \psi, Y_1, Y_2, tag, m)$.
- Otherwise, algorithm $\mathcal{B}$ chooses a random $s_i \in Z_q^*$, sets $H'(\mu, \tau, \psi, Y_1, Y_2, tag, m) = s_i$, sends s_i to algorithm $\mathcal{A}$, and adds $((\mu, \tau, \psi, Y_1, Y_2, tag, m), s_i)$ to list $L_{H'}$.

Phase 1. Algorithm $\mathcal{A}$ chooses (x_S, Y_R) , where $x_S \in \{x_{S,0}, x_{S,1}\}$, $Y_R \in \{Y_{R,0}, Y_{R,1}\}$, and adaptively issues the RKA designcryption queries to algorithm $\mathcal{B}$. For a query (C, ϕ) to RKA.Designcrypt oracle where $C = (\mu, \tau, \psi, tag, \sigma)$, algorithm $\mathcal{B}$ responds as follows.

1. Algorithm $\mathcal{B}$ computes θ' with $\phi(x_R)$. To see how algorithm $\mathcal{B}$ obtains θ without x_R , we rewrite τ such that

$$\begin{aligned}\frac{\tau}{t_i} &= (Y_R \cdot g_1^{H(\mu, tag)})^r = \mu^{-b \cdot H(g^c, tag^*) + x_r + b \cdot H(\mu, tag)} \\ &= (\mu^b)^{H(\mu, tag) - H(g^c, tag^*)} \cdot \mu^{x_r} = \theta^{H(\mu, tag) - H(g^c, tag^*)} \cdot \mu^{x_r} \\ \Rightarrow \theta &= \left(\frac{\tau}{t_i \cdot \mu^{x_r}} \right)^{\frac{1}{H(\mu, tag) - H(g^c, tag^*)}}.\end{aligned}$$

On the other hand,

$$\begin{aligned}\theta' &= \left(\frac{\tau}{t_i} \cdot \mu^{-(x_R + \Delta)} \right)^{\frac{1}{H(\mu, tag)}} = \left(\left(\frac{\tau}{t_i} \cdot \mu^{-\Delta} \right) \cdot \mu^{-x_R} \right)^{\frac{1}{H(\mu, tag)}} \\ &= \theta \cdot (\mu^{-\Delta})^{\frac{1}{H(\mu, tag)}}.\end{aligned}$$

2. If $\hat{e}(\theta', g) = \hat{e}(\mu, g_1)$, algorithm $\mathcal{B}$ outputs $m = \psi / \hat{e}(\theta', g_2)$. Otherwise, it outputs $\perp$.

Challenge. Algorithm $\mathcal{A}$ outputs a message $M^* \in G_T$ on which it wishes to be challenged. Algorithm $\mathcal{B}$ chooses random $s^*, t^* \in Z_q^*$, $d, e \in \{0, 1\}$, $Z \in G_T$, sets $\mu^* = g^c$, and computes

$$\tau^* = (g^c)^{x_{r,e}} \cdot t^*, \quad \psi^* = Z \cdot M^*, \quad \sigma^* = e^* - x_{S,d} \cdot s^*.$$

Algorithm $\mathcal{B}$ outputs the signcryption $C^* = (\mu^*, \tau^*, \psi^*, tag^*, \sigma^*)$, and adds $((\mu^*, \tau^*, \psi^*, (Y_{R,e})^c, Y_{R,e}^{x_{S,d}}, tag^*, M^*), s^*)$ to list $L_{H'}$, $((g^c, (Y_{R,e})^c), t^*)$ to list L_{H_0} .

Phase 2. Algorithm $\mathcal{A}$ chooses (x_S, Y_R) , where $x_S \in \{x_{S,0}, x_{S,1}\}$, $Y_R \in \{Y_{R,0}, Y_{R,1}\}$, and adaptively issues the RKA designcryption queries to algorithm $\mathcal{B}$. For a query (C, ϕ) to RKA.Designcryption oracle where $C = (\mu, \tau, \psi, tag, \sigma)$, algorithm $\mathcal{B}$ responds as follows.

- $H(\mu, tag) \neq H(g^c, tag^*)$. Algorithm $\mathcal{B}$ responds as in Phase 1.
- $H(\mu, tag) = H(g^c, tag^*)$, and $(\mu, \tau, \psi, \sigma) \neq (\mu^*, \tau^*, \psi^*, \sigma^*)$. If algorithm $\mathcal{B}$ accepts this signcryption, it means algorithm $\mathcal{A}$ breaks the security of the CM-RKA security of our scheme. Therefore, algorithm $\mathcal{B}$ outputs $\perp$ except with negligible probability.
- $H(\mu, tag) = H(g^c, tag^*)$, $(\mu, \tau, \psi, \sigma) = (\mu^*, \tau^*, \psi^*, \sigma^*)$ and $\phi(x_R) \neq x_R$. If algorithm $\mathcal{B}$ accepts this signcryption, it means algorithm $\mathcal{A}$ can output $\phi \in \Phi$ such that $(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-\phi(x_R)})^{\frac{1}{H(g^c, tag^*)}} \neq \perp$. That is, $\hat{e}(\theta', g) = \hat{e}(\mu, g_1)$. To guarantee this,

$$\left(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-x_R} \right)^{\frac{1}{H(g^c, tag^*)}} = \left(\frac{\tau^*}{t^*} \cdot (\mu^*)^{-\phi(x_R)} \right)^{\frac{1}{H(g^c, tag^*)}} \Rightarrow x_R = \phi(x_R)$$

should hold. Therefore, algorithm $\mathcal{B}$ outputs $\perp$ except with negligible probability.

Analysis. Algorithm $\mathcal{A}$ has negligible probability to issue (g^c, Y_1) to random oracle H_0 such that $\hat{e}(Y_1, g) = \hat{e}(g^c, Y_{R,e})$ for $e \in \{0, 1\}$. If so, algorithm $\mathcal{B}$ can solve the computational BDH problem immediately. On the other hand, without the value of $H_0(\mu, Y_{R,e}^c)$, algorithm $\mathcal{A}$ has no idea about the identity of receiver R from the challenge signcryption C^* . Likewise, algorithm $\mathcal{A}$ has negligible probability to issue $(\mu^*, \tau^*, \psi^*, Y_1, Y_2, M^*)$ to random oracle H' such that $\hat{e}(Y_1, g) = \hat{e}(g^c, Y_{R,e})$; otherwise, algorithm $\mathcal{B}$ can solve the computational BDH problem immediately. Obviously, without the value of $H'(\mu^*, \tau^*, \psi^*, Y_{R,e}^c, Y_{R,e}^{x_{S,d}}, M^*)$, algorithm $\mathcal{A}$ cannot distinguish the identity of sender S from the challenge signcryption C^* via verification.

This completes the proof of ANON-RKA security of our signcryption scheme.

5 Conclusions

With the development of information technology, there has been a great interest in anonymous systems. On the other hand, traditional security notions cannot meet the requirements in the scenarios where the adversaries might get some partial information about private keys through certain physical methods. Motivated by the above, following the work in [8, 33], in this paper, we focus on the construction of anonymous signcryption schemes secure against related-key attacks. We put forward a specific anonymous signcryption scheme from BDH under the setting of related-key attacks, where an adversary can subsequently observe the outcome of the signcryption and designcryption algorithms under a series of modified private keys of the sender and the receiver (related to the original private keys of the sender and the receiver), respectively. On the basis of the work in [10, 33], we define the security model for anonymous signcryption systems which can resist related-key attacks while maintaining chosen ciphertext attack security (CC-RKA security), chosen message attack security (CM-RKA security) and anonymity, in the sense that a signcryption should contain no information that identifies the sender of the signcryption and the receiver of the message (ANON-RKA), where an adversary is allowed to issue queries to designcryption oracle on linear shifts of the private key of the receiver, and signcryption oracle on linear shifts of the private key of the sender.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer, 2005.
2. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.
3. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.

4. B. Applebaum, D. Harnik, and Y. Ishai. Semantic security under related-key attacks and applications. In *ICS*. Tsinghua University Press, 2011.
5. J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. In *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 80–98. Springer, 2002.
6. P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2005.
7. M. Bellare and D. Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 666–684. Springer, 2010.
8. M. Bellare, D. Cash, and R. Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2011.
9. M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2003.
10. M. Bellare, K. G. Paterson, and S. Thomson. Rka security beyond the linear barrier: Ibe, encryption and signatures. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 331–348. Springer, 2012.
11. E. Biham. New types of cryptoanalytic attacks using related keys (extended abstract). In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 1993.
12. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–219. Springer-Verlag, 2001.
13. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.
14. X. Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer, 2003.
15. S. S. M. Chow, S.-M. Yiu, L. C. K. Hui, and K. P. Chow. Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2003.
16. R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 258–277. Springer, 2004.
17. S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377. ACM, 1982.
18. V. Goyal, A. O’Neill, and V. Rao. Correlated-input secure hash functions. In *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 182–200. Springer, 2011.
19. J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.
20. A. Joux. A one round protocol for tripartite diffie-hellman. In *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer-Verlag, 2000.

21. A. Joux and K. Nguyen. Separating decision diffie-hellman from computational diffie-hellman in cryptographic groups. *J. Cryptology*, 16(4):239–247, 2003.
22. E. Kiltz, P. Mohassel, and A. O’Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 673–692. Springer, 2010.
23. C. K. Li, G. Yang, D. S. Wong, X. Deng, and S. S. M. Chow. An efficient signcryption scheme with key privacy. In *EuroPKI*, volume 4582 of *Lecture Notes in Computer Science*, pages 78–93. Springer, 2007.
24. C. K. Li, G. Yang, D. S. Wong, X. Deng, and S. S. M. Chow. An efficient signcryption scheme with key privacy and its extension to ring signcryption. *Journal of Computer Security*, 18(3):451–473, 2010.
25. B. Libert and J.-J. Quisquater. New identity based signcryption schemes from pairings. *IACR Cryptology ePrint Archive*, 2003:23, 2003.
26. B. Libert and J.-J. Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2004.
27. S. Lucks. Ciphers secure against related-key attacks. In *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 2004.
28. J. Malone-Lee. Identity-based signcryption. *IACR Cryptology ePrint Archive*, 2002:98, 2002.
29. S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.
30. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437. ACM, 1990.
31. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 1996.
32. H. Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 314–332. Springer, 2010.
33. H. Wee. Public key encryption against related key attacks. In *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2012.
34. G. Yang, D. S. Wong, and X. Deng. Analysis and improvement of a signcryption scheme with key privacy. In *ISC*, volume 3650 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2005.
35. Y. Zheng. Digital signcryption or how to achieve $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.