Liu *et al. Cybersecurity* (2019) 2:22
https://doi.org/10.1186/s42400-019-0039-6

Cybersecurity

RESEARCH Open Access

Predicate encryption against master-key tampering attacks

Yuejun Liu[1,2], Rui Zhang[1,2] and Yongbin Zhou[1,2*]

Abstract

Many real world attacks often target the implementation of a cryptographic scheme, rather than the algorithm itself, and a system designer has to consider new models that can capture these attacks. For example, if the key can be tampered by physical attacks on the device, the security of the scheme becomes totally unclear. In this work, we investigate predicate encryption (PE), a powerful encryption primitive, in the setting of tampering attacks. First, we show that many existing frameworks to construct PE are vulnerable to tampering attacks. Then we present a new security notion to capture such attacks. Finally, we take Attrapadung's framework in Eurocrypt'14 as an example to show how to "compile" these frameworks to tampering resilient ones. Moreover, our method is compatible with the original pair encoding schemes without introducing any redundancy.

Keywords: Tampering resilience, Predicate encryption, Pair encoding, Dual system encryption

Introduction

Predicate Encryption (PE) (Lewko et al. 2010; Okamoto and Takashima 2009, 2010, 2012; Lewko and Waters 2010; Katz et al. 2008) is a new paradigm of public-key encryption that supports fine-grained access control policy. In PE, secret keys are associated with parameters X, ciphertexts are associated with parameters Y and a secret key can decrypt the ciphertext if and only if $R(X, Y) = 1$, where R is a predicate for X and Y. Identity-based encryption (IBE) is the simplest kind of PE where R is a equality predicate. PE is powerful and broadly applicable, however, constructing PE schemes and proving their security are complex. Especially, constructing fully secure PE schemes for complex predicates such as for regular languages is a big challenge.

Wee (2014) and Attrapadung (2014) proposed generic frameworks to construct PE schemes with new primitives called predicate encodings or pair encodings respectively (Here we focus on pair encodings, which are more general) and proved the full security utilizing the powerful tool— dual system encryption introduced by Waters Waters (2009) in composite-order groups. Limited by the inefficiency, Attrapadung (2016) and Agrawal and Chase (2016) presented a similar generic framework in prime-order groups. By far, constructing pair encoding schemes instead of PE schemes for new predicates has significantly simplified the process of designing and analyzing fully secure PE schemes.

The traditional security requirement for PE — full security (or called IND-CPA) assumes that the adversary only has black-box access to the system and thus the master key and secret keys are completely hidden from the adversary. However, such assumption may not always hold in practice. Many real world attacks often target the implementation of a cryptographic scheme, rather than the algorithm itself, and a system designer has to consider new models that can capture these attacks. For example, if the key can be leaked via leakage attacks (Kocher 1996; Kocher et al. 1999) or modified via tampering attacks (Biham and Shamir 1997; Boneh et al. 1997; Agrawal et al. 2003) on the device, the security of the scheme becomes totally unclear. In this work, we only focus on the latter. The key could be a signing key of a signature scheme, a decryption key of an encryption scheme, or the master key of a PE scheme. In PE, the privacy of the master key is the cornerstone of the PE security and it is fatal when the master key is tampered.

*Correspondence: zhouyongbin@iie.ac.cn
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Tampering resilience (i.e., tampering attack security)[1] was initiated by Bellare and Kohno (2003), whose targets were pseudorandom permutation (PRP) and pseudorandom function (PRF). Later, Bellare, Cash, and Miller (2011) generalized the concept to other cryptographic primitives, including PKE, signature and IBE. Roughly speaking, the tampering attacks allow the adversary to modify the key of a cryptographic scheme and observe the output under the falsified key. However, except IBE (PE with a simple equality predicate), tampering attacks were not considered for the complicated primitive, PE. In this work, we try to investigate the problem, namely, PE against tampering attacks.

Overall, from the practical perspective, our work can be seen a stepping stone towards the practicality of PE, in which tampering attacks are unavoidable. From the theoretical perspective, we show how to achieve tampering resilience for more complicated primitives than previously known.

Our Contributions. In this work, we focus on the security of PE when the master key is tampered and construct a generic fully secure PE framework against tampering attacks. Our contributions are three-fold.

First, we find that the existing generic PE frameworks (Attrapadung 2014; 2016; Wee 2014; Agrawal and Chase 2016; Chen et al. 2015) are vulnerable to tampering attacks. Note that tampering attacks are out of the scope of these work, so we did not disprove any of the previous results. However, we'd like to show that tampering attacks are powerful and one should consider countermeasures.

Recall that the crux of constructing PE schemes in these frameworks is designing appropriate encoding schemes. One property of the encoding scheme is linearity, which is important to the security proof but also can be a useful tool to threaten the security of PE in the tampering attack. Concretely, with the help of linearity, the adversary is able to recover the secret key by choosing appropriate tampering functions. Since these frameworks cover most of the existing PE constructions, these PE schemes are vulnerable to tampering attacks.

Second, we extend the full security notion for PE to the setting of tampering attacks. We model tampering attacks by providing the adversary with access a tampering oracle: the adversary is allowed to submit a tampering function[2] ϕ and receives secret keys generated under the falsified master key $\phi(msk)$. The adversary can query the tampering oracle adaptively and repeatedly. If all tampering functions appeared in the tampering oracle are the identity function, our definition is exactly the standard model of full security.

Note that restrictions on tampering functions are necessary, otherwise there are trivial tampering attacks. For instance, if we allow the adversary to make any polynomial of arbitrary tampering queries, the master key can be recovered bit-by-bit, as shown by Gennaro et al. (2004). Therefore, we need to either limit the type of tampering functions or limit the number of tampering queries to bypass the impossibility of unrestricted tampering. And since our attacks on existing PE frameworks are given with linear functions, constant functions and affine functions, here we confine tampering functions are algebraic but allow any polynomial of tampering queries.

Finally, we present a generic, fully secure and tampering resilient PE framework and prove its security within the methodology of dual system encryption. Our main tool is a new primitive called Tampering Resilient Function (TRF). Intuitively, a function H is a tampering resilient function if $\mathrm{H}(x)$ is random even given the output of H applied to related inputs $\phi(x)$ where $\phi(x) \neq x$. We take Attrapadung's framework (Attrapadung 2014) as an example to explain how to construct secure PE schemes against tampering attacks. And, significantly, our costs are comparable to those of the original framework.

We also give concrete constructions of TRF. Obviously, the random oracle is a simple TRF. Whereas these frameworks (Attrapadung 2014; 2016; Wee 2014; Agrawal and Chase 2016; Chen et al. 2015) are proposed to construct fully secure PE schemes in the standard model, we expect to instantiate TRF without the random oracle. With the help of other primitives such as the Non-Malleable Key Derivation Function (NM-KDF), the continuous Non-Malleable Key Derivation Function (cNM-KDF), the Tampering Resilient Pseudorandom Function (TR-PRF)[3] and the Correlated Input Secure Hash (CISH) function, we obtain instantiations of TRF in the standard model.

Our Technique. The reason why the existing PE frameworks are insecure against tampering attacks is the key malleability caused by the linearity of the underling pair encoding schemes and it is nature to construct tampering resilient PE schemes by breaking the property. More specifically, in the model of full security, secret keys SK_X for X that can decrypt the challenge ciphertext associated Y^* are forbidden to query. However, the adversary in our new security model can obtain secret keys SK'_X for such X under the falsified master key except SK_X generated under the original master key. Using key malleability, a valid SK_X can be generated from SK'_X. Hence, we expect that SK'_X are independent from SK_X so that they are useless for the adversary to break PE schemes.

[1] Tampering resilience is equivalent to related-key attacks (RKA) security defined in earlier work (Bellare and Kohno 2003).

[2] A tampering function is also called a related-key derivation (RKD) function (Bellare and Kohno 2003).

[3] The TR-PRF is exactly the RKA-PRF (Bellare and Kohno 2003).

Due to the above observation, we define a new primitive TRF and utilize it mapping the master key before generating secret keys to achieve tampering resilience. The property of TRF that the output $\mathrm{H}(x)$ is random even given $\mathrm{H}(\phi(x))$ as long as $\phi(x) \neq x$ ensures that the adversary cannot utilize SK'_X to obtain the additional advantage. In other words, the adversary cannot generate a properly distributed SK_X correctly without the original master key. Meanwhile, without the key malleability, we cannot reduce the security of the modified PE schemes to the original ones and need to prove using the dual system encryption techniques.

Related Work. Since the seminal work of Bellare and Kohno (2003), a lot of tampering resilient symmetric and asymmetric cryptographic primitives were proposed (Bellare and Cash 2010; Bellare et al. 2011, 2012, 2014; Kalai et al. 2011; Liu and Lysyanskaya 2012; Wee 2012; Damgård et al. 2013, 2015; Fujisaki and Xagawa 2016; Faonio and Venturi 2016; Qin et al. 2015). Gennaro et al. (2004) proved that it is impossible to construct secure cryptographic primitives when the tampering functions are arbitrary and the number of tampering queries is unbounded. Hence, the key to construct secure schemes is how to circumvent these restrictions.

One way is considering restricted tampering. By specifying the type of tampering functions, there existed secure PRF, PRP, PKE, symmetric encryption (SE) and signature resilient to linear functions (Bellare and Cash 2010; Bellare et al. 2011; Wee 2012), IBE resilient to affine and polynomial functions (Bellare et al. 2012), IBE resilient to invertible functions (Fujisaki and Xagawa 2015). By restricting the number of tampering queries the adversary is allowed to make, Damgård et al. (2013) proposed secure PKE schemes and signatures resilient to arbitrary tampering functions. In addition, with the bounded tampering resilient model, Faonio and Venturi (2016) gave the first signature construction in the standard model and the first CCA-secure PKE without NIZK.

Another way to bypass the impossibility is taking advantage of extra mechanisms, such as key-updating mechanisms and self-destruct mechanisms (Kalai et al. 2011; Gennaro et al. 2004; Fujisaki and Xagawa 2016).

Most of above work focused on SE and PKE except the one by Bellare et al. (2012). They proposed similar tampering attacks on Boneh-Franklin IBE scheme (Boneh and Franklin 2001) and Waters IBE scheme (Waters 2005) but repaired IBE schemes in a different way from ours. To an extent, the reason for such tampering attacks on PE or IBE is due to the key malleability property of the original schemes. Bellare et al. (2012) employed the property together with a component called collision resistant identity renaming to reduce tampering resilience of the new schemes to the base ones in the black-box way. However, the way no longer works in the standard model. Our strategy is to destroy the key malleability property directly to prove tampering resilience of new schemes. Moreover, the master public key of modified PE schemes (Bellare et al. 2012) depends on the tampering function form. If tampering functions are complex, such as high-degree polynomials, the master public key is huge. However in our work, we only add a function to map the master key without bringing any redundant elements to the original schemes. Finally, our framework can apply to PE schemes more than IBE schemes.

Preliminary

In this section, we present some basic notations and definitions that are used in our construction.

Notations. If S is a set, let $|S|$ denote the number of its elements, and $x \xleftarrow{\$} S$ denotes that x is uniformly sampled from S. Let U_n denote the uniform distribution over $\{0,1\}^n$. Denote $[n]$ the set $\{1,2,\ldots,n\}$. A bold face letter represents a vector (e.g. $\mathbf{a}$), and an uppercase letter represents a matrix (e.g. A). For vectors $\mathbf{a} = (a_1,\ldots,a_n)$ and $\mathbf{b} = (b_1,\ldots,b_n)$, we denote dot product as $\mathbf{a}\cdot\mathbf{b} = (a_1b_1,\ldots,a_nb_n)$. Let $g^{\mathbf{a}}$ be the vector $\left(g^{a_1},\ldots,g^{a_n}\right)$. We denote $\mathbf{negl}(\lambda)$ a negligible function of λ.

Tampering resilient function

In this section, we introduce a new primitive called Tampering Resilient Function (TRF), which will be used as a main tool of our tampering resilient PE schemes. Intuitively, a function H is a tampering resilient function if the output $\mathrm{H}(x)$ is still random even if the adversary obtains outputs of $\mathrm{H}(\phi(x))$ with some $\phi(x) \neq x$.

Definition 1 (Tampering Resilient Function) *Let* $\mathcal{F} = \{\phi | \phi : \{0,1\}^n \to \{0,1\}^n\}$ *be any family of functions. Say* $\mathcal{H} = \left\{\mathrm{H} | \mathrm{H} : \{0,1\}^n \to \{0,1\}^k\right\}$ *is a family of tampering resilient functions if for* $\forall x \xleftarrow{\$} U_n, \phi \xleftarrow{\$} \mathcal{F}, \mathrm{H} \xleftarrow{\$} \mathcal{H}, \phi(x) \neq x$, *the following two distributions are indistinguishable:*

$\{\mathrm{H}(x),\ \mathrm{H}(\phi(x))\}$ *and* $\{y,\ \mathrm{H}(\phi(x))\}$

where $y \xleftarrow{\$} U_k$.

Composite order bilinear groups

Let $(\mathbb{G}, \mathbb{G}_T)$ be cyclic groups of composite order $N = p_1p_2p_3$, where p_1, p_2, p_3 are distinct primes. A bilinear group generator $\mathcal{G}$ takes as input a security parameter λ and outputs a description $(\mathbb{G}, \mathbb{G}_T, e, N)$ where e is an efficiently computable bilinear map. Let $\mathbb{G}_{p_i}$ denote the subgroup of $\mathbb{G}$ of order p_i and g_i denote a random generator of $\mathbb{G}_{p_i}$. Each element $h \in \mathbb{G}$ can be written as $h = g_1^a g_2^b g_3^c$. The bilinear map e satisfies the following properties:

- Non-degenerate: For all generators g of $\mathbb{G}$, $e(g,g) \neq 1$.

- Bilinear: For all $a, b \in \mathbb{Z}_N$, $e\left(g^a, g^b\right) = e(g,g)^{ab}$.
- Orthogonality: For $g \in \mathcal{G}_{p_i}, h \in \mathcal{G}_{p_j}$ where $i \neq j$, $e(g,h) = 1 \in \mathbb{G}_T$.

We will take advantage of the following three Subgroup Decision (**SD**) Assumptions (Waters 2009; Lewko and Waters 2010) to prove the security of our construction.

Definition 2 (SD1) *The SD1 problem is to guess* $\beta \in \{0,1\}$, *given* $\left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, T_\beta\right)$, *where*

$$\begin{aligned}\mathcal{G}_\beta^{\mathbf{SD1}}(\lambda) : \mathbf{param}_{\mathbb{G}} = (\mathbb{G}, \mathbb{G}_T, N, e) &\xleftarrow{\$} \mathcal{G}(\lambda),\\ u, v \xleftarrow{\$} \mathbb{Z}_N, g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 &\xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3},\\ T_0 = g_1^u, T_1 &= g_1^u g_2^v,\\ \text{return } D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, T_\beta\right)&.\end{aligned}$$

Definition 3 (SD2) *The SD2 problem is to guess* $\beta \in \{0,1\}$, *given* $\left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^u g_2^z, g_2^v g_3^\rho, T_\beta\right)$, *where*

$$\begin{aligned}\mathcal{G}_\beta^{\mathbf{SD2}}(\lambda) : \mathbf{param}_{\mathbb{G}} = (\mathbb{G}, \mathbb{G}_T, N, e) &\xleftarrow{\$} \mathcal{G}(\lambda),\\ u, v, z, \rho, w, \kappa, \delta \xleftarrow{\$} \mathbb{Z}_N, g_1 &\xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3},\\ T_0 = g_1^w g_3^\delta, T_1 &= g_1^w g_2^\kappa g_3^\delta,\\ \text{return } D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^u g_2^z, g_2^v g_3^\rho, T_\beta\right)&.\end{aligned}$$

Definition 4 (SD3) *The SD3 problem is to guess* $\beta \in \{0,1\}$, *given* $\left(\mathbf{param}_{\mathbb{G}}, g_1, g_2, g_3, g_1^\alpha g_2^u, g_1^s g_2^v, T_\beta\right)$, *where*

$$\begin{aligned}\mathcal{G}_\beta^{\mathbf{SD3}}(\lambda) : \mathbf{param}_{\mathbb{G}} = (\mathbb{G}, \mathbb{G}_T, N, e) &\xleftarrow{\$} \mathcal{G}(\lambda),\\ u, v, s, \alpha \xleftarrow{\$} \mathbb{Z}_N, g_1 &\xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3},\\ T_0 = e(g_1, g_1)^{\alpha s}, T_1 &\xleftarrow{\$} \mathbb{G}_T,\\ \text{return } D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_2, g_3, g_1^\alpha g_2^u, g_1^s g_2^v, T_\beta\right)&.\end{aligned}$$

To construct the tampering resilient PE framework, we additional define a variant of **SD3**, which is called **TRF-SD3**, to handle tampering queries in the security proof. Let $\mathrm{H} : \{0,1\}^n \to \mathbb{Z}_N$ be a tampering resilient function and $\phi_i : \mathbb{Z}_N \to \mathbb{Z}_N$ be an algebraic tampering function.

Definition 5 (TRF-SD3) *The TRF-SD3 problem is to guess* $\beta \in \{0,1\}$, *given* $\left(\mathbf{param}_{\mathbb{G}}, g_1, g_2, g_3, g_1^{\mathrm{H}(\phi_i(\alpha))} g_2^u, g_1^{\mathrm{H}(\alpha)} g_2^u, g_1^s g_2^v, T_\beta\right)$, *where*

$$\begin{aligned}\mathcal{G}_\beta^{\mathbf{SD3}}(\lambda) : \mathbf{param}_{\mathbb{G}} = (\mathbb{G}, \mathbb{G}_T, N, e) &\xleftarrow{\$} \mathcal{G}(\lambda),\\ u, v, s, \alpha \xleftarrow{\$} \mathbb{Z}_N, g_1 &\xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3},\\ T_0 = e(g_1, g_1)^{\mathrm{H}(\alpha)s}, T_1 &\xleftarrow{\$} \mathbb{G}_T,\\ \text{return } D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_2, g_3, g_1^{\mathrm{H}(\phi_i(\alpha))} g_2^u, g_1^{\mathrm{H}(\alpha)} g_2^u, g_1^s g_2^v, T_\beta\right)&.\end{aligned}$$

Due to the property of TRF, $\mathrm{H}(\phi_i(\alpha))$ is independent from $\mathrm{H}(\alpha)$. Thus, if **SD3** holds in $\mathcal{G}$, so does **TRF-SD3**.

Dual system predicate encryption

We consider the predicate family $R = \{R_\kappa\}_{\kappa \in \mathbb{N}^c}$ for some constant c, where $R_\kappa : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is a predicate mapping a key parameter $X \in \mathcal{X}$ and a ciphertext parameter $Y \in \mathcal{Y}$ to $\{0,1\}$. The family index $\kappa = (n_1, n_2, ...)$ specifies the description of a predicate of $R_\kappa \in R$ and the first entry n_1 in κ specifies the domain. In this work, we omit κ and write R_N for simplicity when its domain is Z_N. We say that R is domain-transferable[4] if for p that divides N, then there exist two maps $f_1 : \mathcal{X}_N \to \mathcal{X}_p, f_2 : \mathcal{Y}_N \to \mathcal{Y}_p$ such that for all $X \in \mathcal{X}_N, Y \in \mathcal{Y}_N$:

- **Completeness.** If $R_N(X,Y) = 1$ then $R_p(f_1(X), f_2(Y)) = 1$.
- **Soundness.** If $R_N(X,Y) = 0$, then $R_p(f_1(X), f_2(Y)) = 0$. Otherwise there exists an algorithm which can output a non-trivial factor F such that $p|F, F|N$.

A predicate encryption (PE) for a predicate R consists of four algorithms: **Setup**, **KeyGen**, **Encrypt**, **Decrypt**, while a dual system PE scheme additionally have three algorithms, **SetupSF**, **KeyGenSF** and **EncryptSF**. Note that the last three algorithms are not parts of the PE scheme, they are only needed for security purposes.

Setup$(\lambda) \to (\mathrm{MSK}, \mathrm{PP})$. The setup algorithm takes in a security parameter λ and outputs a public parameter PP and a master key MSK.
KeyGen$(\mathrm{MSK}, X) \to \mathrm{SK}_X$. The key generation algorithm takes in a master key MSK and a parameter X, then it outputs a normal secret key SK_X.
Encrypt$(\mathrm{PP}, Y, M) \to \mathrm{CT}_Y$. The encryption algorithm takes in a public parameter PP, a message M and a parameter Y, and outputs a normal ciphertext CT_Y.
Decrypt$(\mathrm{CT}_Y, \mathrm{SK}_X) \to M$. The decrypt algorithm takes in a ciphertext CT_Y and a secret key SK_X, and outputs a message M if $R(X,Y) = 1$ for X and Y.
SetupSF$(\lambda) \to (\mathrm{MSK}, \mathrm{PP})$. The semi-functional setup algorithm takes in a security parameter λ and outputs a public parameter PP, a master key MSK and some parameters used to generate semi-functional keys and ciphertexts.
KeyGenSF$(\mathrm{MSK}, X) \to \mathrm{SK}_X$. The semi-functional key generation algorithm takes in a master key MSK and a parameter X, then it outputs a semi-functional secret key SK_X.

[4] The property is required for predicates whose domains are composite-order, such as the predicates in Attrapadung's framework (Attrapadung 2014).

EncryptSF(PP, Y, M) $\rightarrow$ CT_Y. The semi-functional encryption algorithm takes in a public parameter PP, a message M and a parameter Y, and outputs a semi-functional ciphertext CT_Y.

Correctness. We require the correctness condition holds that for all $X \in \mathcal{X}, Y \in \mathcal{Y}$, if $R(X, Y) = 1$, we have **Decrypt**(**Encrypt**(PP, Y, M), **KeyGen**(MSK, X)) $= M$.

Security model

In this work, we consider the situation where the adversary has unexpected power to tamper with the master key and it is not captured by the standard full security notion for PE. In this section, we extend the full security notion to the setting of tampering attacks by providing the adversary an additional tampering oracle. Informally, in our new model, besides secret keys generated under the original master key, the adversary is also allowed to obtain secret keys generated under the falsified master key, which is derived from the original master key with the tampering function chosen by the adversary adaptively.

Note that restrictions on tampering functions are necessary, otherwise there are trivial tampering attacks. For instance, as shown by Gennaro et al. (2004), if the adversary is allowed to make any polynomial of arbitrary tampering queries, the master key may be recovered bit-by-bit. Therefore, we need to either limit the type of tampering functions or limit the number of tampering queries to bypass the impossibility of unrestricted tampering. And since our attacks on existing PE frameworks are given with linear functions, constant functions and affine functions, here we confine tampering functions are algebraic but allow any polynomial of tampering queries.

Denote the tampering functions by $\phi : \mathcal{MSK} \rightarrow \mathcal{MSK}$. An PE scheme Π is secure against such tampering functions if for all PPT adversaries $\mathcal{A}$ it holds that

$$Adv^{trpe}_{\mathcal{A},\Pi}(\lambda) = |\mathbf{Pr}\left[\mathbf{Exp}^{trpe}_{\mathcal{A},\Pi}(\lambda) = 1\right] - \frac{1}{2}| \leq \mathbf{negl}(\lambda).$$

where the game $\mathbf{Exp}^{trpe}_{\mathcal{A},\Pi}(\lambda)$ is defined blow. The adversary $\mathcal{A}$ may adaptively make (unbounded) polynomially many key generation queries to the tampering oracle and receive secret keys generated under the falsified master key ϕ(MSK) both in ***Phase 1*** and ***Phase 2***. Key generation queries for X where $R(X, Y^*) = 1$ (Y^* is the target parameter) under the original master key are forbidden. The restriction is natural, otherwise the adversary can get secret keys that can decrypt the challenge ciphertext to win the security game directly. We use a table $\mathcal{L}$ to record X in key generation queries under the original master key. Notice that if all the tampering functions appeared in the key generation queries are the identity functions, our definition is exactly the model of full security (i.e. IND-CPA security).

$\mathbf{Exp}^{trpe}_{\mathcal{A},\Pi}(\lambda)$:

Setup. In this phase, the challenger runs (MSK, PP) $\leftarrow$ **Setup**(λ) and gives PP to the adversary $\mathcal{A}$. Besides, the challenger should initialize the list $\mathcal{L}$.

Phase 1. In this phase, $\mathcal{A}$ is allowed to make key generation queries under the falsified master key with algebraic tampering functions. Upon receiving a parameter X and a tampering function ϕ_i, the challenger runs $\mathrm{SK}_X \leftarrow$ **KeyGen**(MSK$'$, X) in which MSK$' = \phi_i$(MSK) and returns SK_X to $\mathcal{A}$. If ϕ_i is the identity function, the challenger adds X to $\mathcal{L}$.

Challenge. The adversary submits two messages M_0, M_1 and a challenge parameter Y^* with the restriction $R(X, Y^*) = 0$ for $\forall X \in \mathcal{L}$. The challenger flips an unbiased coin $b \xleftarrow{\$} \{0, 1\}$ and runs $\mathrm{CT}^*_{Y^*} \leftarrow$ **Encrypt**(PP, Y^*, M_b). Then it returns $\mathrm{CT}^*_{Y^*}$.

Phase 2. In this phase, the challenger answers key generation queries in the same way as in ***Phase 1***.

Guess. $\mathcal{A}$ outputs $b' \in \{0, 1\}$. If $b' = b$, the output of the game is 1.

A weaker security definition is non-adaptively (or selectively) tampering resilient in which the tampering functions are fixed in advance. That is, the adversary must submit a set of tampering functions before seeing the public parameters. Since the Assumption **TRF-SD3** depends on tampering functions, we are only able to construct PE schemes against non-adaptive tampering attacks.

Tampering attacks on existing PE frameworks

Wee (2014) and Attrapadung (2014) proposed generic frameworks for fully secure PE constructions via notions called predicate encodings or pair encodings respectively in composite-order bilinear groups. Later, similar frameworks in prime-order groups were also proposed (Chen et al. 2015; Attrapadung 2016; Agrawal and Chase 2016). These frameworks are generic in the sense that they can be applied to almost arbitrary predicate R.

In this section, we show these frameworks are not secure against master-key tampering attacks. Note that tampering attacks are beyond the scope of the security model in these work, so we did not disprove any of the previous results. For simplicity, we take Attrapadung's framework (Attrapadung 2014) as an example to illustrate our attacks in the subsequent parts, but these attacks are also applicable for other frameworks (Wee 2014; Attrapadung 2016; Agrawal and Chase 2016; Chen et al. 2015).

Pair encoding scheme: syntax

Recall that a pair encoding scheme (Attrapadung 2014) for predicate family R consists of four deterministic algorithms as follows:

- **Param**$(\kappa) \to n$. The algorithm takes in κ and outputs an integer n, which specifies the dimension of the common variable $\mathbf{h} = (h_1, \ldots, h_n)$.
- **Enc1**$(X) \to (\mathbf{k} = (k_1, \ldots, k_{m_1}); m_2)$. The algorithm takes in a key parameter X and outputs a sequence of polynomials $\mathbf{k}$ and the dimension of the variable $\mathbf{r} = (r_1, \ldots, r_{m_2})$. $\{k_i\}_{i\in[m_1]}$ is a linear combination of the master key α and variables $\mathbf{r}, \mathbf{h}$.

$$k_i = a_i\alpha + \sum_{j\in[m_2]} a_{i,j}r_j + \sum_{j\in[m_2],l\in[n]} a_{i,l,j}h_l r_j$$

- **Enc2**$(Y) \to (\mathbf{c} = (c_1, \ldots, c_{w_1}); w_2)$. The algorithm takes in a ciphertext parameter Y and outputs a sequence of polynomials $\mathbf{c}$ and the dimension of the variable $\mathbf{s} = (s, s_1, \ldots, s_{w_2})$. $\{c_i\}_{i\in[w_1]}$ is a linear combination of variables $\mathbf{s}, \mathbf{h}$.

$$c_i = b_i s + \sum_{j\in[w_2]} b_{i,j}s_j + \sum_{l\in[n]} b'_{i,l}h_l s + \sum_{j\in[w_2],l\in[n]} b_{i,l,j}h_l s_j$$

- **Pair**$(X, Y) \to \mathbf{E}$. The algorithm takes in X, Y and outputs $\mathbf{E} \in \mathbb{Z}_N^{m_1\times w_1}$.

Correctness. For any $X \in \mathcal{X}, Y \in \mathcal{Y}$, let $(\mathbf{k}; m_2) \leftarrow$ **Enc1**(X), $(\mathbf{c}; w_2) \leftarrow$ **Enc2**(Y), $\mathbf{E} \leftarrow$ **Pair**(X, Y), if $R(X, Y) = 1$, the correctness of the pair encoding scheme is required to satisfy the following equation:

$$\mathbf{k}\mathbf{E}\mathbf{c}^T = \alpha s.$$

It is obvious that the syntax of pair encoding implies the following two properties: parameter-vanishing and linearity.

$$\textbf{(Parameter-Vanishing)} \quad \mathbf{k}(\alpha, \mathbf{0}, \mathbf{h}) = \mathbf{k}(\alpha, \mathbf{0}, \mathbf{0})$$

$$\textbf{(Linearity)} \quad \mathbf{k}(\alpha_1, \mathbf{r}_1, \mathbf{h}) + \mathbf{k}(\alpha_2, \mathbf{r}_2, \mathbf{h}) = \mathbf{k}(\alpha_1 + \alpha_2, \mathbf{r}_1 + \mathbf{r}_2, \mathbf{h})$$
$$\mathbf{c}(\mathbf{s}_1, \mathbf{h}) + \mathbf{c}(\mathbf{s}_2, \mathbf{h}) = \mathbf{c}(\mathbf{s}_1 + \mathbf{s}_2, \mathbf{h})$$

When proving the security of PE, parameter-vanishing and linearity make it possible to switch normal ciphertexts and keys to semi-functional ones indistinguishably. However, linearity also makes PE vulnerable to tampering attacks. The detailed attacks is provided in the subsequent sections.

Pair encoding scheme: security definition

Our construction is compatible with previous pair encoding schemes (Attrapadung 2014; Wee 2014; Attrapadung 2016; Agrawal and Chase 2016; Chen et al. 2015). There are two security notions in the pair encoding scheme (Attrapadung 2014)—perfectly master-key hiding security and computationally master-key hiding security. Here we recall the security definitions as follows.

Perfectly Master-key Hiding Security. The pair encoding scheme P is perfectly master-key hiding (PMH) if the following two distributions are identical. If $R(X, Y) = 0$, let $n \leftarrow$ **Param**(κ), $(\mathbf{k}; m_2) \leftarrow$ **Enc1**(X), $(\mathbf{c}; w_2) \leftarrow$ **Enc2**(Y):

$\mathbf{c}(\mathbf{s}, \mathbf{h}), \mathbf{k}(0, \mathbf{r}, \mathbf{h})$ and $\mathbf{c}(\mathbf{s}, \mathbf{h}), \mathbf{k}(\alpha, \mathbf{r}, \mathbf{h})$ where the probability is taken over $\alpha \xleftarrow{\$} \mathbb{Z}_N, \mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$.

Computationally Master-key Hiding Security. We define two computational security notions: selectively master-key hiding security (SMH) and co-selectively master-key hiding security (CMH) in a bilinear group generator $\mathcal{G}$ through the following game template $\mathbf{Exp}_{\mathcal{G},b,\mathcal{A},G}$ for the flavor $G \in \{\text{SMH}, \text{CMH}\}$. It takes as input the security parameter λ and does the experiment with the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as follows:

$$\begin{aligned}
&\mathbf{Exp}_{\mathcal{G},b,\mathcal{A},G}(\lambda):\\
&(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \leftarrow \mathcal{G}_\lambda,\\
&g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3},\\
&\alpha \xleftarrow{\$} \mathbb{Z}_N, n \leftarrow \mathbf{Param}(\kappa), \mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n,\\
&st \leftarrow \mathcal{A}_1^{\mathcal{O}^1_{G,b,\alpha,\mathbf{h}}(\cdot)}(g_1, g_2, g_3),\\
&b' \leftarrow \mathcal{A}_2^{\mathcal{O}^2_{G,b,\alpha,\mathbf{h}}(\cdot)}(st),
\end{aligned}$$

where st denotes the state information and the oracles $\mathcal{O}^1$ and $\mathcal{O}^2$ are defined below:

- **Selective Security**. $\mathcal{O}^1$ can be queried once while $\mathcal{O}^2$ can be queried many times.
  - $\mathcal{O}^1_{\text{SMH},b,\alpha,\mathbf{h}}(Y^*)$: Run $(\mathbf{c}; w_2) \leftarrow$ **Enc2**(Y^*) and pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Return $\mathbf{C} \leftarrow g_2^{\mathbf{c}(\mathbf{s},\mathbf{h})}$.
  - $\mathcal{O}^2_{\text{SMH},b,\alpha,\mathbf{h}}(X)$: If $R(X, Y^*) = 1$, return $\perp$. Else, run $(\mathbf{k}; m_2) \leftarrow$ **Enc1**(X) and pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}$. Return $\mathbf{K} \leftarrow \begin{cases} g_2^{\mathbf{k}(0,\mathbf{r},\mathbf{h})} & \text{if b} = 0 \\ g_2^{\mathbf{k}(\alpha,\mathbf{r},\mathbf{h})} & \text{if b} = 1 \end{cases}$.
- **Co-selective Security**. Both $\mathcal{O}^1$ and $\mathcal{O}^2$ can be queried only once.
  - $\mathcal{O}^1_{\text{CMH},b,\alpha,\mathbf{h}}(Y^*)$: Run $(\mathbf{k}; m_2) \leftarrow$ **Enc1**(X) and pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}$. Return $\mathbf{K} \leftarrow \begin{cases} g_2^{\mathbf{k}(0,\mathbf{r},\mathbf{h})} & \text{if b} = 0 \\ g_2^{\mathbf{k}(\alpha,\mathbf{r},\mathbf{h})} & \text{if b} = 1 \end{cases}$.
  - $\mathcal{O}^2_{\text{CMH},b,\alpha,\mathbf{h}}(X)$: If $R(X, Y^*) = 1$, return $\perp$.

Else, run $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$ and pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Return $\mathbf{C} \leftarrow g_2^{\mathbf{c}(\mathbf{s},\mathbf{h})}$.

For a PPT adversary $\mathcal{A}$, we define the advantage of the $\mathcal{A}$ in the security game $\mathbf{Exp}_{\mathcal{G},b,\mathcal{A},G}$ for the flavor $G \in \{\mathrm{SMH}, \mathrm{CMH}\}$ as:

$$Adv_{\mathcal{A}}^{G}(\lambda) = \left|\mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{G},0,\mathcal{A},G} = 1\right] - \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{G},1,\mathcal{A},G} = 1\right]\right|$$

A pair encoding scheme P is selectively (resp. co-selectively) master-key hiding in $\mathcal{G}$ if $Adv_{\mathcal{A}}^{SMH}(\lambda)$ (resp. $Adv_{\mathcal{A}}^{CMH}(\lambda)$) is negligible in λ. If both hold, we say P is doubly selectively master-key hiding.

Review Attrapadung's framework (Attrapadung 2014)

With the pair encoding scheme, the PE framework in Attrapadung (2014) is given below:

Setup$(\lambda) \rightarrow (\mathrm{MSK}, \mathrm{PP})$. Run $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Pick $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Run $n \leftarrow \mathbf{Param}(\kappa)$. Choose $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n$ and $\alpha \xleftarrow{\$} \mathbb{Z}_N$. The public key is $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1,g_1)^{\alpha}\right)$ and the master key is $\mathrm{MSK} = \alpha$.

KeyGen$(\mathrm{MSK}, X) \rightarrow \mathrm{SK}_X$. Run $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Output the secret key SK_X as follows:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\alpha,\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Encrypt$(\mathrm{PP}, Y, M) \rightarrow \mathrm{CT}_Y$. Run $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y)$. Pick $\mathbf{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Output the secret key $\mathrm{CT}_Y = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})}, C_1 = e(g_1,g_1)^{\alpha s} M.$$

Decrypt$(\mathrm{CT}_Y, \mathrm{SK}_X) \rightarrow M$. Check whether $R(X, Y) = 1$. If yes, run $\mathbf{E} \leftarrow \mathbf{Pair}(X, Y)$. Compute M as follows:

$$M = \frac{C_1}{e(\mathrm{SK}_X^{\mathbf{E}}, \mathbf{C}_0)}.$$

Tampering attacks on Attrapadung's framework

In the setting of tampering attacks, the adversary is allowed to tamper the master key and gets secret keys generated under the falsified master key. We show that Attrapadung's framework (Attrapadung 2014) is not tampering resilient by specifying particular tampering functions. The master key is an exponent $\alpha \in \mathbb{Z}_N$. Suppose the tampering function is linear, w.l.o.g. $msk' = \phi(\alpha) = t\alpha$. Having access to the key generation oracle under the falsified master key $\phi(\alpha)$, the adversary obtains

$$\mathrm{SK}'_X = g_1^{\mathbf{k}(t\alpha,\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3,$$

where each polynomial k_i of $\mathbf{k}(t\alpha, \mathbf{r}, \mathbf{h})$ is a linear combination of the master key α and variables $\mathbf{r}, \mathbf{h}$. Because of the linearity of α and $\mathbf{r}$ in $\mathbf{k}$, we are able to get a properly distributed secret key $\mathrm{SK}_X = g_1^{\mathbf{k}_X(\alpha,\mathbf{r}',\mathbf{h})}$ under the original master key α with a new randomness $\mathbf{r}' = t^{-1}\mathbf{r}$ by raising SK'_X to t^{-1}. In this way the adversary can win the security game because he can get secret keys for X where $R(X, Y^*) = 1$ and Y^* is used in the challenge ciphertext.

Besides the linear functions, Attrapadung's framework (Attrapadung 2014) is not secure against other algebraic functions. Suppose the tampering function is affine, w.l.o.g. $msk' = \phi(\alpha) = t\alpha + c$. With a call to the key generation oracle for X under the falsified master key $\phi(\alpha)$ where $R(X, Y^*) = 1$, the adversary obtains

$$\mathrm{SK}'_{X,\mathrm{aff}} = g_1^{\mathbf{k}(t\alpha+c,\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Raising $\mathrm{SK}'_{X,\mathrm{aff}}$ to t^{-1} resulting a secret key $\mathrm{SK}_{X,\mathrm{aff}} = g_1^{\mathbf{k}_X(\alpha+t^{-1}c,\mathbf{r}',\mathbf{h})}$ under the original master key α with a new randomness $\mathbf{r}' = t^{-1}\mathbf{r}$. Decrypt the challenge ciphertext using $\mathrm{SK}_{X,\mathrm{aff}}$, the adversary obtains

$$e(\mathrm{SK}_{X,\mathrm{aff}}^{\mathbf{E}}, \mathbf{C}_0) = e(g_1,g_1)^{(\alpha+t^{-1}c)s}.$$

Then the adversary continues to make a key generation call for X with a constant tampering function $\phi(\alpha) = t^{-1}c$ and obtains

$$\mathrm{SK}_{X,\mathrm{con}} = g_1^{\mathbf{k}(t^{-1}c,\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

After decrypting with $\mathrm{SK}_{X,\mathrm{con}}$, the adversary is able to compute $e(g_1,g_1)^{t^{-1}cs}$. Thus, combining an affine function and a constant function, the adversary is able to recover $e(g_1,g_1)^{\alpha s}$ to decrypt the challenge ciphertext.

Remark 1 *Although the master key in other frameworks (Wee 2014; Chen et al. 2015; Attrapadung 2016; Agrawal and Chase 2016) is a group element in the form of g_1^{α} where g_1 is the generator of a prime-order group or a prime-order subgroup of composite-order groups, similar tampering attacks still exist. Suppose the tampering function is a simple polynomial, w.l.o.g. $msk' = \phi(g_1^{\alpha}) = g_1^{t\alpha}$. The remaining parts are the same.*

Note that Bellare et al. (2012) presented similar tampering attacks on Boneh-Franklin IBE scheme (Boneh and Franklin 2001) and Waters IBE scheme(Waters 2005). The cause of such tampering attacks is a paradoxical property called key malleability, which means that there is a simulator can transform a secret key generated by the original master key to one generated by the falsified master key. On the one hand, key malleability allows us to reduce the security of the tampering resilient PE to the security of base one in the black-box way, as Bellare et al. did (Bellare et al. 2012). But on the other hand, we can also derive properly distributed secret keys from those under the falsified master key to break the security of schemes and that is the reason for above tampering attacks. Bellare et al. took advantage of key malleability together with

another component called identity renaming scheme to prove tampering resilience of new IBE schemes. In contrast, we destroy the property to construct secure PE schemes. In the next section, we show how to construct tampering resilient PE schemes.

Our generic tampering resilient PE framework

In this section, we propose a secure and generic PE framework from pair encoding against algebraic tampering functions, which is a slightly modified version of the original one. The only thing we alter is the way the master key used. That is, we keep the master key unchanged, but before using it, we first add a function to map the master key and use the mapped key to generate public parameters and secret keys. Next we will explain how to use the idea to fix Attrapadung's framework (Attrapadung 2014) to achieve tampering resilience.

Generic construction

Let $\mathcal{H} = \{\{0,1\}^n \to \mathbb{Z}_N\}$ be a family of tampering resilient functions. Denote by $P = (\mathbf{Param}, \mathbf{Enc1}, \mathbf{Enc2}, \mathbf{Pair})$ a pair encoding scheme for predicate family R. The PE framework is given as follows:

Setup$(\lambda) \to (\mathrm{MSK}, \mathrm{PP})$. Run $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Pick $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Run $n \leftarrow \mathbf{Param}(\kappa)$. Choose $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n$ and $\alpha \xleftarrow{\$} \mathbb{Z}_N$. Choose $\mathrm{H} \xleftarrow{\$} \mathcal{H}$. The public key is $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1,g_1)^{\mathrm{H}(\alpha)}, \mathrm{H}\right)$ and the master key is $\mathrm{MSK} = \alpha$.

KeyGen$(\mathrm{MSK}, X) \to \mathrm{SK}_X$. Run $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Output the secret key SK_X as follows:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3. \tag{1}$$

Encrypt$(\mathrm{PP}, Y, M) \to \mathrm{CT}_Y$. Run $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y)$. Pick $\mathbf{s} = \left(s, s_1, \ldots, s_{w_2}\right) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Output the ciphertext $\mathrm{CT}_Y = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})}, C_1 = e(g_1,g_1)^{\mathrm{H}(\alpha)s} M. \tag{2}$$

Decrypt$(\mathrm{CT}_Y, \mathrm{SK}_X) \to M$. Check whether $R(X,Y) = 1$. If yes, run $\mathbf{E} \leftarrow \mathbf{Pair}(X,Y)$. Compute M as follows:

$$M = \frac{C_1}{e(\mathrm{SK}_X^{\mathbf{E}}, \mathbf{C}_0)}.$$

Correctness. Due to the correctness of the pair encoding scheme, for $R(X,Y) = 1$, we have

$$\begin{aligned} e\left(\mathrm{SK}_X^{\mathbf{E}}, \mathbf{C}_0\right) &= e\left(\left(g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3\right)^{\mathbf{E}}, g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})}\right) \\ &= e(g_1,g_1)^{\mathbf{k}\mathbf{E}\mathbf{c}^T} \\ &= e(g_1,g_1)^{\mathrm{H}(\alpha)s} \end{aligned}$$

Semi-functional Algorithms. We additionally define semi-functional algorithms used in the security proof.

SetupSF$(\lambda) \to (\mathrm{MSK}, \mathrm{PP}, g_2, \hat{\mathbf{h}})$. The algorithm is the same as **Setup** except it additionally outputs a generator $g_2 \xleftarrow{\$} \mathbb{G}_{p_2}$ and a semi-functional parameter $\hat{\mathbf{h}} \xleftarrow{\$} \mathbb{Z}_N^n$.

KeyGenSF$(\mathrm{MSK}, X, g_2, \hat{\mathbf{h}}, type, \hat{\alpha}) \to \mathrm{SK}_X$. Run $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$. Pick $\mathbf{r}, \hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Output the secret key SK_X depending on the input type $t \in \{1,2,3\}$ as follows:

$$\mathrm{SK}_X = \begin{cases} \left(g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3\right) & \text{if } t = 1 \quad (3) \\ \left(g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3\right) & \text{if } t = 2 \quad (4) \\ \left(g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3\right) & \text{if } t = 3 \quad (5) \end{cases}$$

EncryptSF$(\mathrm{PP}, Y, M, g_2, \hat{\mathbf{h}}) \to \mathrm{CT}_Y$. Run $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y)$. Pick $\mathbf{s} = (s, s_1, \ldots, s_{w_2}), \hat{\mathbf{s}} = (\hat{s}, \hat{s}_1, \ldots, \hat{s}_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Output the ciphertext $\mathrm{CT}_Y = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{h}})}, C_1 = e(g_1,g_1)^{\mathrm{H}(\alpha)s} M. \tag{6}$$

Security proof

If the pair encoding scheme P is doubly selectively master-key hiding and the family $\mathcal{H}$ is tampering resilient, the above framework is fully secure and tampering resilient against algebraic tampering functions. More precisely, we present the following theorem.

Theorem 1 *Suppose that a pair encoding scheme P for predicate R is selectively and co-selectively master-key hiding, the Subgroup Decision Assumptions are intractable*

Table 1 Games in the security proof

$\mathbf{G}_{res}$:	Replace the restriction $R_N(X,Y^*) = 0$ with $R_{p_2}(X,Y^*) = 0$.
$\mathbf{G}_0$:	$(\mathrm{MSK}, \mathrm{PP}, g_2, \hat{\mathbf{h}}) \leftarrow \mathbf{SetupSF}(\lambda)$ $\mathrm{CT}_Y^* \leftarrow \mathbf{EncryptSF}\left(\mathrm{PP}, Y^*, M_b, g_2, \hat{\mathbf{h}}\right)$
$\mathbf{G}_{k,1}$:	$\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N, \mathrm{SK}_j \leftarrow$ { $\mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \mathbf{0}, 3, \hat{\alpha}_j\right)$ if $j < k$; $\mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \hat{\mathbf{h}}, 1, 0\right)$ if $j = k$; $\mathbf{KeyGen}\left(\phi_i(\alpha), X_j\right)$ if $j > k$ }
$\mathbf{G}_{k,2}$:	$\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N, \mathrm{SK}_j \leftarrow$ { $\mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \mathbf{0}, 3, \hat{\alpha}_j\right)$ if $j < k$; $\mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \hat{\mathbf{h}}, 2, \hat{\alpha}_j\right)$ if $j = k$; $\mathbf{KeyGen}\left(\phi_i(\alpha), X_j\right)$ if $j > k$ }
$\mathbf{G}_{k,3}$:	$\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N, \mathrm{SK}_j \leftarrow$ { $\mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \mathbf{0}, 3, \hat{\alpha}_j\right)$ if $j \le k$; $\mathbf{KeyGen}\left(\phi_i(\alpha), X_j\right)$ if $j > k$ }
$\mathbf{G}_{q_1+1}$:	$\mathrm{SK}_j \leftarrow \mathbf{KeyGenSF}(\phi_i(\alpha), X_j, g_2, \hat{\mathbf{h}}, 1, 0)$
$\mathbf{G}_{q_1+2}$:	$\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$ $\mathrm{SK}_j \leftarrow \mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \hat{\mathbf{h}}, 2, \hat{\alpha}\right)$
$\mathbf{G}_{q_1+3}$:	$\mathrm{SK}_j \leftarrow \mathbf{KeyGenSF}\left(\phi_i(\alpha), X_j, g_2, \mathbf{0}, 3, \hat{\alpha}\right)$
$\mathbf{G}_{final}$:	$M \xleftarrow{\$} \mathcal{M}, \mathrm{CT}_Y^* \leftarrow \mathbf{EncryptSF}\left(\mathrm{PP}, Y^*, M, g_2, \hat{\mathbf{h}}\right)$

Fig. 1 Proof profile

in $\mathcal{G}$ and $\mathcal{H}$ is a family of tampering resilient functions. Also suppose that P is domain-transferable. The framework above is fully secure and non-adaptively tampering resilient with algebraic tampering functions. For any PPT adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1, \ldots, \mathcal{B}_6$ with almost the same running time as $\mathcal{A}$ such that for any security parameter λ:

$$\begin{aligned} Adv^{trpe}_{\mathcal{A}}(\lambda) \leq\ & 2Adv^{\mathbf{SD1}}_{\mathcal{B}_1}(\lambda) + (2q_1+3)Adv^{\mathbf{SD2}}_{\mathcal{B}_2}(\lambda) \\ & + Adv^{\mathbf{TRF-SD3}}_{\mathcal{B}_3}(\lambda) + q_1 Adv^{CMH}_{\mathcal{B}_4}(\lambda) \\ & + Adv^{SMH}_{\mathcal{B}_5}(\lambda) + q_{all} Adv^{TRF}_{\mathcal{B}_6}(\lambda) \end{aligned}$$

*where q_1, q_2 is the number of key generation queries in **Phase 1** and **Phase 2** respectively and $q_{all} = q_1 + q_2$.*

The proof is similar to that of Attrapadung's framework (Attrapadung 2014), except the adversary in our security definition is allowed to receive secret keys generated under the falsified master key even for parameters X satisfy $R(X, Y^*) = 1$ where Y^* is used in the challenge ciphertext. We define a sequence of games and complete the proof of Theorem 1 by proving the indistinguishability between adjacent games. Each game is defined as follows and the difference compared with the previous one is given in Table 1.

$\mathbf{G}_{real}$: The original game.

$\mathbf{G}_{res}$: Replace the restriction $R_N(X, Y^*) = 0$ with $R_{p_2}(X, Y^*) = 0$.

$\mathbf{G}_0$: The normal challenge ciphertext is modified to be semi-functional type.

$\mathbf{G}_{k,t}$: The k-th queried secret is modified to be semi-functional of type-t where $k \in [1, q_1], t \in \{1, 2, 3\}$.

$\mathbf{G}_{q_1+t}$: All secret keys queried in **Phase 2** are modified to be semi-functional of type-t where $t \in \{1, 2, 3\}$.

$\mathbf{G}_{final}$: The semi-functional challenge ciphertext is modified to the ciphertext of a random message.

We provide a proof profile in Fig. 1, including all the assumptions and security definitions used in the proof (To clarity, TRF is used throughout the proof and only two places are marked in the figure). The key is to prove that additional secret keys under the falsified master key won't help the adversary break the security of PE. We sketch the proof idea here. Divide these secret keys into two cases. For X that $R(X, Y^*) = 0$, tampering resilience follows from the master-key hiding of the underlying pair encoding scheme P directly. For X that $R(X, Y^*) = 1$, the adversary is able to compute $e(g_1, g_1)^{\mathrm{H}(\phi(\alpha))s}$ with the help of the correctness of P. However, the tampering function ϕ cannot be the identity function in this case so that the adversary still cannot recover the masking $e(g_1, g_1)^{\mathrm{H}(\alpha)s}$ used in the ciphertext from $e(g_1, g_1)^{\mathrm{H}(\phi(\alpha))s}$ due to the property of the tampering resilient function H.

Let $Adv^i_{\mathcal{A}}$ denote the advantage of $\mathcal{A}$ in $\mathbf{G}_i$. Notice that the advantage of $\mathcal{A}$ in $\mathbf{G}_{final}$ $Adv^{final}_{\mathcal{A}}$ is 0 since the challenge ciphertext in the final game is independent of M_b. The complete proof of Theorem 1 is given in Appendix.

In addition, if the underlying pair encoding scheme P is perfectly master-key hiding, we get the following theorem.

Theorem 2 *Suppose that a pair encoding scheme P for predicate R is perfectly master-key hiding, the Subgroup Decision Assumptions are intractable in $\mathcal{G}$ and $\mathcal{H}$ is a family of tampering resilient functions. Also suppose that P is domain-transferable. The framework above is fully secure and non-adaptively tampering resilient with*

Table 2 A comparison of TRF based on different primitives

Primitives	TRF description	Reference	Assumption	Tampering times	Tampering function
(c)NM-KDF	$\mathrm{H}(x) = f(x)$	(Faust et al. 2014)	-	One-Time	Bounded-degree polynomial
$f : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$					
		(Qin et al. 2015)	DDH	Continual	Bounded-degree polynomial
TR-PRF	$\mathrm{H}(x) = f_x(c)$	(Bellare and Cash 2010)	DDH	Continual	Group-induced function
$f : \boldsymbol{\mathcal{K}} \times \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$	where $c \leftarrow \boldsymbol{\mathcal{X}}$ is public				
			DLIN	Continual	Group-induced function
CISH	$\mathrm{H}(x) = f_k(x)$	(Goyal et al. 2011)	q-DHI	Continual	Bounded-degree polynomial
$f : \boldsymbol{\mathcal{K}} \times \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$	where $k \leftarrow \boldsymbol{\mathcal{K}}$ is public				

Table 3 Theoretical analysis of TRF

Reference	Description	Assumption	Size of PP	Cost of evaluation
(Faust et al. 2014)**	-	$f : \mathbb{Z}_p^t \times \mathbb{Z}_p^t \to \mathbb{Z}_N$	$t\|\mathbb{Z}_p\|$	-
(Qin et al. 2015)***	DDH	$f : \mathbb{H}^{n\times n} \times \mathbb{Z}_p^n \to \mathbb{H}^n$	$2\|\mathbb{Z}_p\| + (n^2+1)\|\mathbb{H}\|$	$2n^2$Exp
(Bellare and Cash 2010)	DDH	$f : \mathbb{Z}_p^{n+1} \times \{0,1\}^n \to \mathbb{H}$	$(n+1)\|\mathbb{Z}_p\| + \|\mathbb{H}\|$	1Exp
	DLIN	$f : (\mathbb{Z}_p^{2\times 2})^{n+1} \times \{0,1\}^n \to \mathbb{H}$	$4(n+1)\|\mathbb{Z}_p\| + \|\mathbb{H}\|$	1Exp
(Goyal et al. 2011)	q-DHI	$f : \mathbb{Z}_p \times \mathbb{Z}_p \to \mathbb{H}$	$\|\mathbb{Z}_p\| + \|\mathbb{H}\|$	1Exp

*$\mathbb{H}$ is a group of prime order p over $\mathbb{Z}_N$. $|\mathbb{Z}_p|$ and $|\mathbb{H}|$ denote the size of an element in $\mathbb{Z}_p$ and $\mathbb{H}$, respectively. Exp denotes a modular exponentiation in $\mathbb{H}$

**Here we consider the simplest t-wise independent hash function $f = \sum_{i=0}^{t-1} a_i x_i \bmod p \bmod N$ where $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p$

***Only partial PP of OT-LF is considered here

algebraic tampering functions. For any PPT adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1, \ldots, \mathcal{B}_4$ with almost the same running time as $\mathcal{A}$ such that for any security parameter λ:

$$Adv_{\mathcal{A}}^{trpe}(\lambda) \le 2Adv_{\mathcal{B}_1}^{\mathbf{SD1}}(\lambda) + (2q_{all}+1)Adv_{\mathcal{B}_2}^{\mathbf{SD2}}(\lambda) + Adv_{\mathcal{B}_3}^{\mathbf{TRF-SD3}}(\lambda) + q_{all}Adv_{\mathcal{B}_4}^{TRF}(\lambda)$$

Instantiations of tampering resilient functions

Tampering Resilient Function (TRF) is an important building block for our tampering resilient PE framework. In this section, we present several instantiations of TRF. A function H is a tapering resilient function if the output is random even when the adversary sees outputs of related inputs. A simple construction of TRF is the random oracle. But the existing PE frameworks based on the dual system encryption technique are designed to construct fully secure PE schemes without the random oracle, we'd better instantiate TRF in the standard model. A comparison of TRF based on different cryptographic primitives is given in Table 2.

TRF based on (c)NM-KDF. Faust et al. (2014) introduced a notion called Non-Malleable Key Derivation Function (NM-KDF) for tampering function family $\mathcal{F}$, which uses randomness x to derive $y = f(x)$ in such a way that, even x is tampered to a different value $x' = \phi(x)$, $y' = f(x')$ does not reveal any information about y. The definition is almost the same as ours of TRF. Since we want to give our tampering resilient PE framework in a unified way, we define a new primitive TRF to cover current similar notions. Faust et al. (2014) proposed a simple construction of NM-KDF based on a t-wise independent hash function but the construction is secure against one-time tampering attacks. Soon after, Qin et al. (2015) extended the notion to continuous NM-KDF (cNM-KDF) in which unbounded polynomially tampering queries are allowed. They also present constructions of cNM-KDF for any polynomials of bounded degree under the standard assumptions, e.g., DDH and DCR. Applying these constructions to our PE framework, we obtain tampering resilient PE framework for the same class of tampering functions.

TRF based on TR-PRF. Tampering Resilient Pseudorandom Function (TR-PRF) was first formalized by Bellare and Kohno (2003), in which the adversary can observe the input-output of the PRF not just under the target key k, but under others keys $\phi_1(k), \ldots, \phi_q(k)$ derived from k in adversary-specified ways. Later, Bellare and Cash (2010) gave the first construction of TR-PRF (based on the Naor-Reingold PRF) whose security is proven under the DDH assumption for group-induced functions[5]. They also provided a construction (based on Lewko-Waters PRF) under the DLIN assumption for similar tampering functions. Combining TR-PRF with our PE framework, we obtain secure PE schemes against tampering attacks, in which the key of TR-PRF is the master key and the input is a public randomness. Because of the restricted range of TR-PRF, we can only obtain tampering resilient PE schemes for linear or group-induced functions[6].

TRF based on CISH Functions. Another way of constructing TRF is based on Correlated Input Secure Hash (CISH) functions introduced by Goyal et al. (2011), which can be interpreted as a dual version of TR-PRF. Specifically, a CISH function f with the key k ensures that the output $f_k(x)$ is random even given the hash values of multiple correlated inputs $(f_k(\phi_1(x)), \ldots, f_k(\phi_q(x)))$, where functions ϕ_i are chosen by the adversary. Compare with TRF, the CISH function has an extra key k. In our PE framework with TRF based on the CISH function, the key k is part of the public parameters and the input is the master key. Goyal et al. (2011) give a concrete construction of the CISH function for a large class of polynomial functions of bounded degree under the variant of q-Diffie Hellman Inversion (q-DHI) assumption. However, the construction is only non-adaptively secure, that is, the adversary must submit functions ϕ_i at the begin of the game. This makes our PE framework non-adaptively tampering resilient as well.

[5] A function $\phi = \mathbf{a} * \mathbf{d}$ is a group-induced function if the operation $*$ corresponds to the component-wise multiplication on $\mathbb{Z}_p^*$.

[6] Recent work (Matsuda and Schuldt 2018) provided a TR-PRF construction for arbitrary tampering functions satisfying collision-resistance and output-unpredictability for a bounded number of falsified keys. Since the construction is incompatible with our security model, we omit it here.

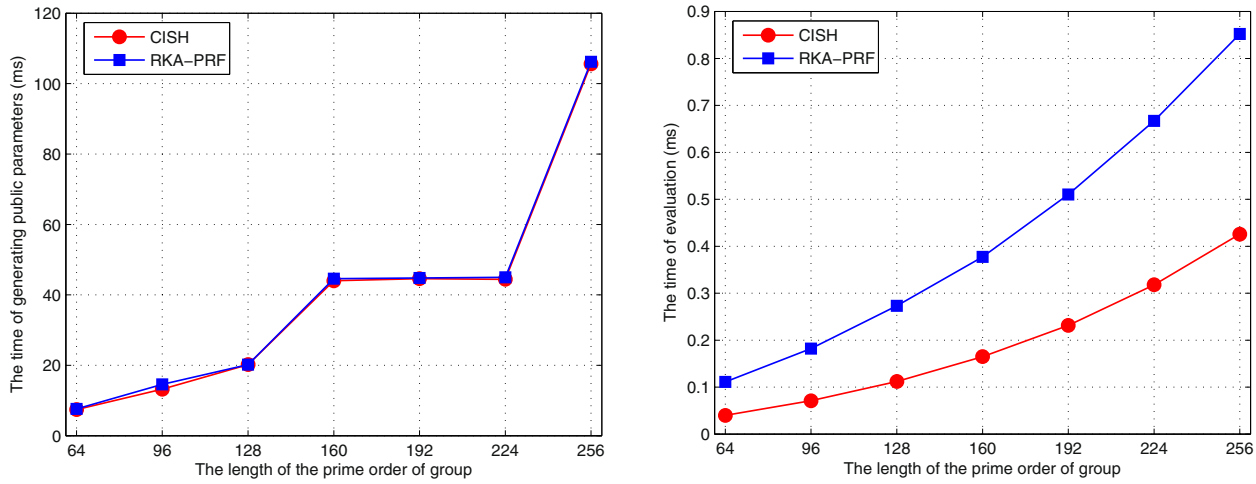

Fig. 2 $\mathbb{Z}_N$ where N is prime. **a** The time of generating PP **b** The time of evaluation

Performance Evaluation. We give a concrete analysis of the performance of TRF, which is the only additional primitive in our tampering resilient PE compared with the original PE schemes. Table 3 presents the public parameters (PP) size and evaluation efficiency of TRF theoretically. Since module exponentiation offers the main cost, we ignore other computations, such as addition and multiplication. In addition, it should be pointed out the construction of cNM-KDF employs three primitives: one-time lossy filter (OT-LF), pairwise independent hash function (i.e. t = 2) and one-time signature (OT-Sig) and here we only present costs of OT-LF based on DDH and pairwise independent hash function, but even so, the performance of TRF based on cNM-KDF is the worst. As shown in Table 3, almost all of TRF need additional public parameters except TRF based on NM-KDF in Faust et al. (2014), which only allows one-time tampering query, however. In general, storage overhead and computation overhead introduced by TRF are negligible compared to PE schemes. Hence, the efficiency of our TR-PE schemes is comparable to that of original PE schemes.

To evaluate the practical performance of TRF, we implement TRF with DDH-based TR-PRF in Bellare and Cash (2010) and TRF with q-DHI based CISH in Goyal et al. (2011). The output of both TRF is $\mathbb{H}$, a prime-order group over $\mathbb{Z}_N$. Since TRF can apply to PE in both composite-order ($N = p_1p_2p_3$) and prime-order (N is a prime larger than p) groups, we set N composite or prime accordingly and both cases are tested here.

Our experiments are conducted on an Intel Core(TM) i7-4790 CPU @3.6GHz and 12 GB RAM. The prime order

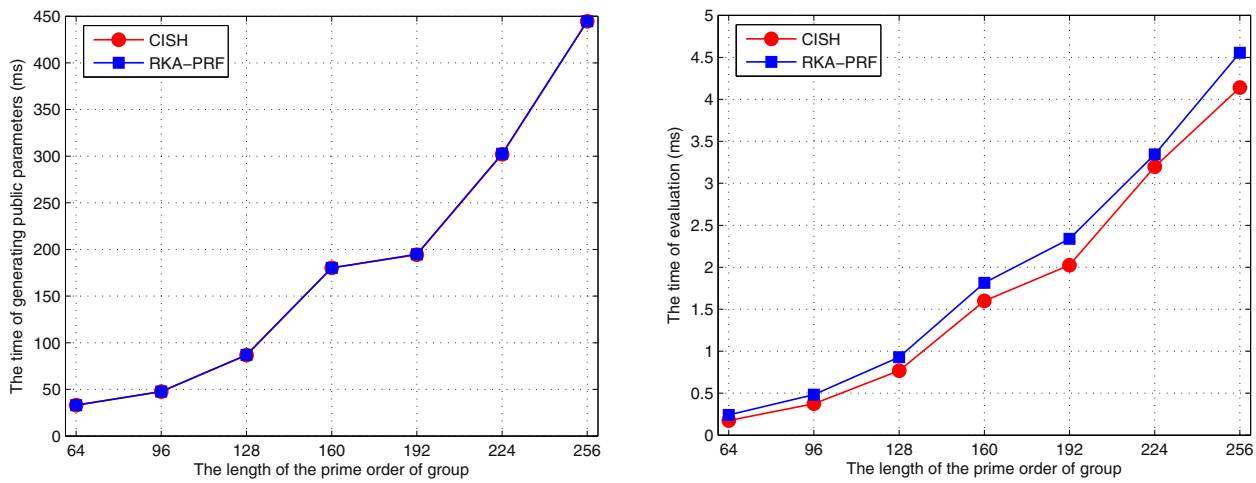

Fig. 3 $\mathbb{Z}_N$ where N is composite. **a** The time of generating PP **b** The time of evaluation

p influences computational cost. We set the length of p increasing from 64 to 256, and repeat each instance 5 times for group generation and 5000 times for function evaluation, then take the average. As depicted in Figs. 2a (Figs. 3a) and 2b (3b), we show the time of generating PP and evaluation when N is prime (N is composite) and the time is given in milliseconds. The most-consuming operation is generating group, specifically generating the prime-order generator, which takes about 100 ms in case of $|p| = 256$ when N is prime, while the evaluation operation only takes less than 1 ms. Even when N is composite, the runtime of TRF is on the order of milliseconds. Overall, TRF is practical that has little impact on the efficiency of PE.

Further directions

In this work we explore the security of PE frameworks against tampering attacks on the master key. In our tampering resilient model, the tampering functions are restricted to algebraic functions and the number of tampering queries is unbounded. In practice, algebraic tampering functions may not describe specific attacks and constructing tampering resilient PE schemes with broader classes of tampering functions is an important direction. Besides, in PE there are two type of keys: the master key and secret keys. Compared to the master key, secret keys are more vulnerable to tampering attacks because we may store the master key in tamper-proof hardware which is expensive for secret keys. How to design secure PE schemes in this case is another direction of our further work.

Appendix

Proof of Theorem 1

Lemma 1 *For any adversary $\mathcal{A}$ can distinguish* $\mathbf{Game}_{real}$ *from* $\mathbf{Game}_{res}$*, there exists an adversary $\mathcal{B}$, such that for any security parameter λ,* $\left|Adv_{\mathcal{A}}^{real} - Adv_{\mathcal{A}}^{res}\right| \leq Adv_{\mathcal{B}}^{\mathbf{SD1}} + Adv_{\mathcal{B}}^{\mathbf{SD2}}$.

Proof The proof is the same as that in Attrapadung (2014) which reduced to the soundness of domain-transferability. If the adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{res}$ from $\mathbf{Game}_0$, then we can find a factor of N to break the assumption **SD1** or **SD2**.

Lemma 2 *For any adversary $\mathcal{A}$ can distinguish* $\mathbf{Game}_{res}$ *from* $\mathbf{Game}_0$*, there exists an adversary $\mathcal{B}$, such that for any security parameter λ,* $|Adv_{\mathcal{A}}^{res} - Adv_{\mathcal{A}}^{0}| \leq Adv_{\mathcal{B}}^{\mathbf{SD1}}$.

Proof We will build a PPT adversary $\mathcal{B}$ against Assumption **SD1** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, T_\beta\right)$ and will simulate either $\mathbf{Game}_{res}$ or $\mathbf{Game}_0$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)}\right)$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$. Let g_2 be an unknown random generator of G_{p_2}. $\mathcal{B}$ implicitly sets $\hat{\mathbf{h}} \mod p_2 = \mathbf{h} \mod p_2$. $\hat{\mathbf{h}}$ is properly distributed and is independent from $\mathbf{h} \mod p_1$ due to the Chinese Remainder Theorem.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ with the master key and sending (tampered) normal secret keys to $\mathcal{A}$.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = \left(s', s_1', \ldots, s_{w_2}'\right) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}_{Y^*}^* = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = T_\beta^{\mathbf{c}(\mathbf{s}', \mathbf{h})}, C_1 = e(T_\beta, g_1)^{\mathrm{H}(\alpha)s'} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}_{Y^*}^*$.

Phase 2. $\mathcal{B}$ answers key generations from $\mathcal{A}$ as same as in ***Phase 1***.

- If $\beta = 0$,

  $$\mathbf{C}_0 = g_1^{u\mathbf{c}(\mathbf{s}', \mathbf{h})} = g_1^{\mathbf{c}(u\mathbf{s}', \mathbf{h})}, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

  $\mathrm{CT}_{Y^*}^*$ is a properly distributed and normal ciphertext where $\mathbf{s} = u\mathbf{s}' \mod p_1$ is uniform. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{res}$.
- If $\beta = 1$,

  $$\mathbf{C}_0 = g_1^{u\mathbf{c}(\mathbf{s}, \hat{\mathbf{h}})} g_2^{v\mathbf{c}(\mathbf{s}, \hat{\mathbf{h}})} = g_1^{\mathbf{c}(u\mathbf{s}', \hat{\mathbf{h}})} g_2^{\mathbf{c}(v\mathbf{s}', \hat{\mathbf{h}})},$$

  $$C_1 = e\left(g_1^u g_2^v, g_1\right)^{\mathrm{H}(\alpha)s'} M_b = e(g_1, g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

  $\mathrm{CT}_{Y^*}^*$ is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{h}} = \mathbf{h} \mod p_2$ and $\hat{\mathbf{s}} = v\mathbf{s}' \mod p_2$ are uniform and independent from $\mathbf{h}, \mathbf{s}$ due to the Chinese Remainder Theorem. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_0$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{res}$ from $\mathbf{Game}_0$ with negligible probability, otherwise $\mathcal{B}$ can break Assumption **SD1**.

Lemma 3 *For any adversary $\mathcal{A}$ can distinguish* $\mathbf{Game}_{k-1,3}$ *from* $\mathbf{Game}_{k,1}$*, there exists an adversary $\mathcal{B}$, such that for any security parameter λ,* $|Adv_{\mathcal{A}}^{k-1,3} - Adv_{\mathcal{A}}^{k,1}| \leq Adv_{\mathcal{B}}^{\mathbf{SD2}}$.

Proof We will build a PPT adversary $\mathcal{B}$ against Assumption **SD2** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^u g_2^z, g_2^v g_3^\rho, T_\beta\right)$ and will simulate either $\mathbf{Game}_{k-1,3}$ or $\mathbf{Game}_{k,1}$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)}\right)$ and

the master key MSK $=\alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$. Let g_2 be an unknown random generator of G_{p_2}. $\mathcal{B}$ implicitly sets $\hat{\mathbf{h}} \mod p_2 = \mathbf{h} \mod p_2$. $\hat{\mathbf{h}}$ is properly distributed and is independent from $\mathbf{h} \mod p_1$ due to the Chinese Remainder Theorem.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ in the following way:

- $j < k$: In this case, $\mathcal{B}$ generates type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \left(g_2^{v} g_3^{\rho}\right)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = v\hat{\alpha}'$ is uniform.
- $j = k$: Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}', \hat{\mathbf{r}}' \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (T_\beta)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3.$$

- $j > k$: In this phase, $\mathcal{B}$ generates normal secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = \left(s', s'_1, \ldots, s'_{w_2}\right) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = (g_1^u g_2^z)^{\mathbf{c}(\mathbf{s}',\mathbf{h})} = g_1^{\mathbf{c}(u\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(z\mathbf{s}',\mathbf{h})},$$

$$C_1 = e((g_1^u g_2^z), g_1)^{\mathrm{H}(\alpha)s'} M_b = e(g_1, g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$, which is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{s}} = z\mathbf{s}' \mod p_2$ is uniform and independent from $\mathbf{s} = u\mathbf{s}' \mod p_1$ due to the Chinese Remainder Theorem.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key and sending normal secret keys to $\mathcal{A}$.

- If $\beta = 0$,

$$\begin{aligned}\mathrm{SK}_X &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot \left(g_1^{w} g_3^{\delta}\right)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3 \\ &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}'_3.\end{aligned}$$

  SK_X is a properly distributed and normal secret key under the falsified master key $\phi_i(\alpha)$ where $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ is uniform. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k-1,3}$.
- If $\beta = 1$,

$$\begin{aligned}\mathrm{SK}_X &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot \left(g_1^{w} g_2^{\kappa} g_3^{\delta}\right)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3 \\ &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\kappa\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}'_3.\end{aligned}$$

  SK_X is a properly distributed and type-1 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\mathbf{r}} = \kappa\hat{\mathbf{r}}' \mod p_2$ is uniform and independent from $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,1}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{k-1,3}$ from $\mathbf{Game}_{k,1}$ with negligible probability, otherwise $\mathcal{B}$ can break Assumption **SD2**.

Lemma 4 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv^{k,1}_{\mathcal{A}} - Adv^{k,2}_{\mathcal{A}}| \leq Adv^{\mathbf{CMH}}_{\mathcal{B}} + Adv^{\mathbf{TRF}}_{\mathcal{B}}$.*

If the adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$, we can build a PPT adversary $\mathcal{B}$ against the CMH security of the pair encoding scheme P or the TRF family $\mathcal{H}$. Denote $\mathbf{K}^*$ by the challenge secret key. Define F as the event that $\mathbf{K}^*$ is generated under the original master key and $\neg F$ as the event that $\mathbf{K}^*$ is generated under the falsified master key. In order to complete the proof of Lemma 4, it suffices to prove Claims 1 and 2.

Claim 1 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $\left|Adv^{k,1}_{\mathcal{A}} - Adv^{k,2}_{\mathcal{A}}\right| \leq Adv^{\mathbf{CMH}}_{\mathcal{B}}$, conditioned on F occurs.*

Proof We will build a PPT adversary $\mathcal{B}$ against the CMH security of the pair encoding scheme P with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given (g_1, g_2, g_3) and will simulate either $\mathbf{Game}_{k,1}$ or $\mathbf{Game}_{k,2}$.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)}\right)$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ in the following way:

- $j < k$: In this case, $\mathcal{B}$ generates type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$.
- $j = k$: Upon receiving X_k, $\mathcal{B}$ makes a key query to its challenger and receives back $T_\beta = g_2^{\mathbf{k}(\beta,\hat{\mathbf{r}},\hat{\mathbf{h}})}$. Then $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X_k)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. It computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot T_\beta \cdot \mathbf{R}_3.$$

- $j > k$: In this phase, $\mathcal{B}$ generates normal secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^* such that for all $X \in \mathcal{L}, R(X, Y^*) = 0$. This query can be made since $R(X_k, Y^*) = 0$ when the event F occurs. $\mathcal{B}$ makes a ciphertext query to its challenger and receives back $D = g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{h}})}$. Then $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})} \cdot D, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$, which is a properly distributed and semi-functional ciphertext.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key and sending normal secret keys to $\mathcal{A}$.

- If $\beta = 0$,

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-1 semi-functional secret key under the original master key α. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,1}$.
- If $\beta = 1$,

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-2 semi-functional secret key under the original master key α. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,2}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$ with negligible probability, otherwise $\mathcal{B}$ can break the CMH security of P.

Claim 2 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv^{k,1}_{\mathcal{A}} - Adv^{k,2}_{\mathcal{A}}| \leq Adv^{\mathbf{TRF}}_{\mathcal{B}}$, conditioned on $\neg F$ occurs.*

Proof We will build a PPT adversary $\mathcal{B}$ against the family of tampering resilient functions $\mathcal{H}$ with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given (g_1, g_2, g_3) and will simulate either $\mathbf{Game}_{k,1}$ or $\mathbf{Game}_{k,2}$.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)}\right)$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ in the following way:

- $j < k$: In this case, $\mathcal{B}$ generates type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$.
- $j = k$: Upon receiving X_k and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X_k)$ and picks $\mathbf{r}, \hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\mathbf{h}} \xleftarrow{\$} \mathbb{Z}_N^n$. It computes

$$T_\beta \leftarrow \begin{cases} g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} & \text{if } \beta = 0 \\ g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\alpha,\hat{\mathbf{r}},\hat{\mathbf{h}})} & \text{if } \beta = 1 \end{cases}$$

  and the secret key:

$$\mathrm{SK}_X = T_\beta \cdot \mathbf{R}_3.$$

- $j > k$: In this phase, $\mathcal{B}$ generates normal secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s} = (s, s_1, \ldots, s_{w_2}), \hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes $D \leftarrow g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{h}})}$ and the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})} \cdot D, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$, which is a properly distributed and semi-functional ciphertext.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key and sending normal secret keys to $\mathcal{A}$.

- If $\beta = 0$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-1 semi-functional secret key under the falsified master key $\phi_i(\alpha)$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,1}$.
- If $\beta = 1$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-2 semi-functional secret key under the falsified master key $\phi_i(\alpha)$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,2}$.

In this case, $R(X_k, Y^*) = 1$ (Here $\phi_i(\alpha) \neq \alpha$), SK_X can be used to decrypt the challenge ciphertext and the adversary will get $e(g_1, g_1)^{\mathrm{H}(\phi_i(\alpha))s}$ or $e(g_1, g_1)^{\mathrm{H}(\phi_i(\alpha))s} e(g_2, g_2)^{\hat{\alpha}\hat{s}}$. Due to the property of $\mathcal{H}$, $e(g_1, g_1)^{\mathrm{H}(\phi_i(\alpha))s}$ is still a random element in $\mathbb{G}_T$ given $e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b$, meaning that both secret keys cannot decrypt correctly. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,1}$ from $\mathbf{Game}_{k,2}$ with negligible probability, otherwise $\mathcal{B}$ can break the TRF family $\mathcal{H}$.

Lemma 5 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,2}$ from $\mathbf{Game}_{k,3}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv_{\mathcal{A}}^{k,2} - Adv_{\mathcal{A}}^{k,3}| \leq Adv_{\mathcal{B}}^{\mathbf{SD2}}$.*

Proof We will build a PPT adversary $\mathcal{B}$ against Assumption **SD2** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^u g_2^z, g_2^v g_3^\rho, T_\beta\right)$ and will simulate either $\mathbf{Game}_{k,2}$ or $\mathbf{Game}_{k,3}$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = (g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)})$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$. Let g_2 be an unknown random generator of G_{p_2}. $\mathcal{B}$ implicitly sets $\hat{\mathbf{h}} \mod p_2 = \mathbf{h} \mod p_2$. $\hat{\mathbf{h}}$ is properly distributed and is independent from $\mathbf{h} \mod p_1$ due to the Chinese Remainder Theorem.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ in the following way:

- $j < k$: In this case, $\mathcal{B}$ generates type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = v\hat{\alpha}'$ is uniform.
- $j = k$: Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}', \hat{\mathbf{r}}' \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot (T_\beta)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3.$$
- $j > k$: In this phase, $\mathcal{B}$ generates normal secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \mathbf{R}_3.$$

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = (g_1^u g_2^z)^{\mathbf{c}(\mathbf{s}',\mathbf{h})} = g_1^{\mathbf{c}(u\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(z\mathbf{s}',\mathbf{h})},$$

$$C_1 = e((g_1^u g_2^z), g_1)^{\mathrm{H}(\alpha)s'} M_b = e(g_1, g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$, which is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{s}} = z\mathbf{s}' \mod p_2$ is uniform and independent from $\mathbf{s} = u\mathbf{s}' \mod p_1$ due to the Chinese Remainder Theorem.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key and sending normal secret keys to $\mathcal{A}$.

- If $\beta = 0$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot (g_1^w g_3^\delta)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3$$
  $$= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(v\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}'_3.$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ and $\hat{\alpha} = v\hat{\alpha}'$ are uniform. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,3}$.

- If $\beta = 1$,

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot \left(g_2^{v} g_3^{\rho}\right)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \left(g_1^{w} g_2^{\kappa} g_3^{\delta}\right)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3$$
$$= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(v\hat{\alpha}',\kappa\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3'.$$

SK_X is a properly distributed and type-2 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\mathbf{r}} = \kappa\hat{\mathbf{r}}' \mod p_2$ and $\hat{\mathbf{h}} = \mathbf{h} \mod p_2$ are uniform and independent from $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ and $\mathbf{h}$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k,2}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{k,2}$ from $\mathbf{Game}_{k,3}$ with negligible probability, otherwise $\mathcal{B}$ can break Assumption **SD2**.

Lemma 6 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1,3}$ from $\mathbf{Game}_{q_1+1}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv_{\mathcal{A}}^{q_1,3} - Adv_{\mathcal{A}}^{q_1+1}| \leq Adv_{\mathcal{B}}^{\mathbf{SD2}}$.*

Proof We will build a PPT adversary $\mathcal{B}$ against Assumption **SD2** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^{u} g_2^{z}, g_2^{v} g_3^{\rho}, T_\beta\right)$ and will simulate either $\mathbf{Game}_{q_1,3}$ or $\mathbf{Game}_{q_1+1}$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = (g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)})$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$. Let g_2 be an unknown random generator of G_{p_2}. $\mathcal{B}$ implicitly sets $\hat{\mathbf{h}} \mod p_2 = \mathbf{h} \mod p_2$. $\hat{\mathbf{h}}$ is properly distributed and is independent from $\mathbf{h} \mod p_1$ due to the Chinese Remainder Theorem.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ by generating type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot \left(g_2^{v} g_3^{\rho}\right)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = v\hat{\alpha}'$ is uniform.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = (s', s_1', \ldots, s_{w_2}') \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}_{Y^*}^* = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = (g_1^{u} g_2^{z})^{\mathbf{c}(\mathbf{s}',\mathbf{h})} = g_1^{\mathbf{c}(u\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(z\mathbf{s}',\mathbf{h})},$$

$$C_1 = e((g_1^{u} g_2^{z}), g_1)^{\mathrm{H}(\alpha)s'} M_b = e(g_1, g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}_{Y^*}^*$, which is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{s}} = z\mathbf{s}' \mod p_2$ is uniform and independent from $\mathbf{s} = u\mathbf{s}' \mod p_1$ due to the Chinese Remainder Theorem.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}', \hat{\mathbf{r}}' \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (T_\beta)^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3.$$

- If $\beta = 0$,

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_1^{w} g_3^{\delta})^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3$$
$$= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3'.$$

SK_X is a properly distributed and normal secret key under the falsified master key $\phi_i(\alpha)$ where $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ is uniform. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{q_1,3}$.

- If $\beta = 1$,

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_1^{w} g_2^{\kappa} g_3^{\delta})^{\mathbf{k}(0,\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3$$
$$= g_1^{\mathbf{k}(\mathrm{H}\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\kappa\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3'.$$

SK_X is a properly distributed and type-1 semi-functional secret key the falsified master key $\phi_i(\alpha)$ where $\hat{\mathbf{r}} = \kappa\hat{\mathbf{r}}' \mod p_2$ is uniform and independent from $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{q_1+1}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1,3}$ from $\mathbf{Game}_{q_1+1}$ with negligible probability, otherwise $\mathcal{B}$ can break Assumption **SD2**.

Lemma 7 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $\left|Adv_{\mathcal{A}}^{q_1+1} - Adv_{\mathcal{A}}^{q_1+2}\right| \leq Adv_{\mathcal{B}}^{\mathbf{SMH}} + Adv_{\mathcal{B}}^{\mathbf{TRF}}$.*

If the adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$, we can build a PPT adversary $\mathcal{B}$ against the SMH security of the pair encoding scheme P or the TRF family $\mathcal{H}$. Denote $\mathbf{K}^*$ by the challenge secret key. Define F as the event that $\mathbf{K}^*$ is generated under the original master key and $\neg F$ be the event that $\mathbf{K}^*$ is generated under the falsified master key. In order to complete the proof of Lemma 7, it suffices to prove Claim 3, and 4.

Claim 3 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $\left|Adv_{\mathcal{A}}^{q_1+1} - Adv_{\mathcal{A}}^{q_1+2}\right| \leq Adv_{\mathcal{B}}^{\mathbf{SMH}}$, conditioned on F occurs.*

Proof We will build a PPT adversary $\mathcal{B}$ against the SMH security of the pair encoding scheme P with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given (g_1, g_2, g_3) and will simulate either $\mathbf{Game}_{q_1+1}$ or $\mathbf{Game}_{q_1+2}$.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = \left(g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)}\right)$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ by generating type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ makes a ciphertext query to its challenger and receives back $D = g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{h}})}$. Then $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and the challenge ciphertext $\mathrm{CT}_{Y^*}^* = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})} \cdot D, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}_{Y^*}^*$, which is a properly distributed and semi-functional ciphertext.

Phase 2. In this phase, upon receiving X_j and a tampering function ϕ_i, $\mathcal{B}$ makes a key query to its challenger and receives back $T_\beta = g_2^{\mathbf{k}(\beta,\hat{\mathbf{r}},\hat{\mathbf{h}})}$. This query can be made since $R(X_j, Y^*) = 0$. $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X_k)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. It computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot T_\beta \cdot \mathbf{R}_3.$$

- If $\beta = 0$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-1 semi-functional secret key under the original master key α. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q_1+1}$.
- If $\beta = 1$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\alpha),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-2 semi-functional secret key under the original master key α. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q_1+2}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$ with negligible probability, otherwise $\mathcal{B}$ can break the SMH security of P.

Claim 4 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv_{\mathcal{A}}^{q_1+1} - Adv_{\mathcal{A}}^{q_1+2}| \leq Adv_{\mathcal{B}}^{\mathbf{TRF}}$, conditioned on $\neg F$ occurs.*

Proof We will build a PPT adversary $\mathcal{B}$ against the family of tampering resilient functions $\mathcal{H}$ with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given (g_1, g_2, g_3) and will simulate either $\mathbf{Game}_{q_1+1}$ or $\mathbf{Game}_{q_1+2}$.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = (g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1)^{\mathrm{H}(\alpha)})$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ by generating type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}, \hat{\mathbf{h}} \xleftarrow{\$} \mathbb{Z}_N^n$ and computes $D \leftarrow g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{h}})}$ and the challenge ciphertext $\mathrm{CT}_{Y^*}^* = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{h})} \cdot D, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}_{Y^*}^*$, which is a properly distributed and semi-functional ciphertext.

Phase 2. In this phase, upon receiving X_j and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X_k)$ and picks $\mathbf{r}, \hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. It computes $T_\beta \leftarrow \begin{cases} g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} & \text{if } \beta = 0 \\ g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\alpha,\hat{\mathbf{r}},\hat{\mathbf{h}})} & \text{if } \beta = 1 \end{cases}$ and the secret key:

$$\mathrm{SK}_X = T_\beta \cdot \mathbf{R}_3.$$

- If $\beta = 0$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-1 semi-functional secret key under the falsified master key $\phi_i(\alpha)$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q_1+1}$.
- If $\beta = 1$,

  $$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{h}})} \cdot \mathbf{R}_3.$$

  SK_X is a properly distributed and type-2 semi-functional secret key under the falsified master

key $\phi_i(\alpha)$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{Game}_{q_1+2}$.

In this case, $R(X_k, Y^*) = 1$ (Here $\phi_i(\alpha) \neq \alpha$), SK_X can be used to decrypt the challenge ciphertext and the adversary will get $e(g_1,g_1)^{\mathrm{H}(\phi_i(\alpha))s}$ or $e(g_1,g_1)^{\mathrm{H}(\phi_i(\alpha))s}e(g_2,g_2)^{\hat{\alpha}\hat{s}}$. Because of property of $\mathcal{H}$, $e(g_1,g_1)^{\mathrm{H}(\phi_i(\alpha))s}$ is still a random element in $\mathbb{G}_T$ given $e(g_1,g_1)^{\mathrm{H}(\alpha)s}M_b$, meaning that both secret keys cannot decrypt correctly. Hence, $\mathcal{A}$ can distinguish $\mathbf{Game}_{q_1+1}$ from $\mathbf{Game}_{q_1+2}$ with negligible probability, otherwise $\mathcal{B}$ can break the TRF family $\mathcal{H}$.

Lemma 8 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{G}_{q_1+2}$ from $\mathbf{G}_{q_1+3}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $|Adv_{\mathcal{A}}^{q_1+2} - Adv_{\mathcal{A}}^{q_1+3}| \leq Adv_{\mathcal{B}}^{\mathbf{SD2}}$.*

Proof We will build a PPT adversary $\mathcal{B}$ against Assumption **SD2** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_3, g_1^u g_2^z, g_2^v g_3^\rho, T_\beta\right)$ and will simulate either $\mathbf{G}_{q_1+2}$ or $\mathbf{G}_{q+1+3}$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = (g_1, g_3, g_1^{\mathbf{h}}, e(g_1,g_1)^{\mathrm{H}(\alpha)})$ and the master key $\mathrm{MSK} = \alpha$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$. Let g_2 be an unknown random generator of G_{p_2}. $\mathcal{B}$ implicitly sets $\hat{\mathbf{h}} \mod p_2 = \mathbf{h} \mod p_2$. $\hat{\mathbf{h}}$ is properly distributed and is independent from $\mathbf{h} \mod p_1$ due to the Chinese Remainder Theorem.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ by generating type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r},\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = v\hat{\alpha}'$ is uniform.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = \left(g_1^u g_2^z\right)^{\mathbf{c}(\mathbf{s}',\mathbf{h})} = g_1^{\mathbf{c}(u\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(z\mathbf{s}',\mathbf{h})},$$

$$C_1 = e((g_1^u g_2^z), g_1)^{\mathrm{H}(\alpha)s'} M_b = e(g_1,g_1)^{\mathrm{H}(\alpha)us'} M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$, which is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{s}} = z\mathbf{s}' \mod p_2$ is uniform and independent from $\mathbf{s} = u\mathbf{s}' \mod p_1$ due to the Chinese Remainder Theorem.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key. At the beginning, $\mathcal{B}$ picks $\hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}', \hat{\mathbf{r}}' \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot (T_\beta)^{\mathbf{k}(\mathbf{0},\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3.$$

- If $\beta = 0$,

$$\begin{aligned}\mathrm{SK}_X &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha))\mathbf{r}',\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot (g_1^w g_3^\delta)^{\mathbf{k}(\mathbf{0},\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3 \\ &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(v\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3'.\end{aligned}$$

  SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ and $\hat{\alpha} = v\hat{\alpha}'$ are uniform. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q_1+3}$

- If $\beta = 1$,

$$\begin{aligned}\mathrm{SK}_X &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}',\mathbf{h})} \cdot (g_2^v g_3^\rho)^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot (g_1^w g_2^\kappa g_3^\delta)^{\mathbf{k}(\mathbf{0},\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3 \\ &= g_1^{\mathbf{k}(\mathrm{H}(\phi_i(\alpha)),\mathbf{r}'+w\hat{\mathbf{r}}',\mathbf{h})} \cdot g_2^{\mathbf{k}(v\hat{\alpha}',\kappa\hat{\mathbf{r}}',\mathbf{h})} \cdot \mathbf{R}_3'.\end{aligned}$$

  SK_X is a properly distributed and type-2 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\mathbf{r}} = \kappa\hat{\mathbf{r}}' \mod p_2$ and $\hat{\mathbf{h}} = \mathbf{h} \mod p_2$ are uniform and independent from $\mathbf{r} = \mathbf{r}' + w\hat{\mathbf{r}}' \mod p_1$ and $\mathbf{h}$. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q+2}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{G}_{q+2}$ from $\mathbf{G}_{q_1+3}$ with negligible probability, otherwise $\mathcal{B}$ can break Assumption **SD2**.

Lemma 9 *For any adversary $\mathcal{A}$ can distinguish $\mathbf{G}_{q_1+3}$ from $\mathbf{G}_{final}$, there exists an adversary $\mathcal{B}$, such that for any security parameter λ, $\left|Adv_{\mathcal{A}}^{q_1+3} - Adv_{\mathcal{A}}^{final}\right| \leq Adv_{\mathcal{B}}^{\mathbf{TRF-SD3}}$.*

Proof We will build a PPT adversary $\mathcal{B}$ against the Assumption **TRF-SD3** with the help of an adversary $\mathcal{A}$. $\mathcal{B}$ is given $D = \left(\mathbf{param}_{\mathbb{G}}, g_1, g_2, g_3, g_1^{H(\alpha)} g_2^u, g_1^{H(\phi_i(\alpha))} g_2^u, g_1^s g_2^v, T_\beta\right)$ and will simulate either $\mathbf{G}_{q_1+3}$ or $\mathbf{G}_{final}$ with D.

Setup. $\mathcal{B}$ chooses $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_N^n, \mathrm{H} \xleftarrow{\$} \mathcal{H}$ and computes the public key $\mathrm{PP} = (g_1, g_3, g_1^{\mathbf{h}}, e(g_1, g_1^{H(\alpha)} g_2^u) = e(g_1,g_1)^{\mathrm{H}(\alpha)})$. Then $\mathcal{B}$ gives PP to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{B}$ answers all key generation queries and tampering queries from $\mathcal{A}$ by generating type-3 semi-functional secret keys under the falsified master key. Upon receiving X and a tampering function ϕ_i, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}, \hat{\alpha}' \xleftarrow{\$}$

$\mathbb{Z}_N$. Then it computes the secret key [7]:

$$\mathrm{SK}_X = \left(g_1^{H(\phi_i(\alpha))} g_2^{u}\right)^{\mathbf{k}(1,\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = u + \hat{\alpha}'$ is uniform.

Challenge. In this phase, $\mathcal{A}$ submits two messages M_0, M_1 and the challenge parameter Y^*. $\mathcal{B}$ flips an unbiased coin $b \xleftarrow{\$} \{0,1\}$ and runs $(\mathbf{c}; w_2) \leftarrow \mathbf{Enc2}(Y^*)$. Then it picks $\mathbf{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ and computes the challenge ciphertext $\mathrm{CT}^*_{Y^*} = (\mathbf{C}_0, C_1)$ as follows:

$$\mathbf{C}_0 = (g_1^s g_2^v)^{\mathbf{c}(\mathbf{s}',\mathbf{h})} = g_1^{\mathbf{c}(s\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(v\mathbf{s}',\mathbf{h})}, C_1 = T_\beta M_b.$$

$\mathcal{B}$ sends $\mathcal{A}$ the challenge ciphertext $\mathrm{CT}^*_{Y^*}$.

Phase 2. In this phase, $\mathcal{B}$ answers all key generation queries from $\mathcal{A}$ with the falsified master key. At the beginning, $\mathcal{B}$ picks $\hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. Upon receiving X, $\mathcal{B}$ runs $(\mathbf{k}; m_2) \leftarrow \mathbf{Enc1}(X)$ and picks $\mathbf{r}, \mathbf{R}_3 \xleftarrow{\$} \mathbb{Z}_{p_3}^{m_1}$. Then it computes the secret key:

$$\mathrm{SK}_X = \left(g_1^{\mathrm{H}(\phi_i(\alpha))} g_2^{u}\right)^{\mathbf{k}(1,\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot \mathbf{R}_3.$$

SK_X is a properly distributed and type-3 semi-functional secret key under the falsified master key $\phi_i(\alpha)$ where $\hat{\alpha} = u + \hat{\alpha}'$ is uniform.

- If $\beta = 0$,

  $$\mathbf{C}_0 = g_1^{\mathbf{c}(s\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(v\mathbf{s}',\mathbf{h})}, C_1 = e(g_1, g_1)^{\mathrm{H}(\alpha)s} M_b$$

  $\mathrm{CT}^*_{Y^*}$ is a properly distributed and semi-functional ciphertext where $\hat{\mathbf{s}} = v\mathbf{s}' \mod p_2$ is uniform and independent from $\mathbf{s} = s\mathbf{s}' \mod p_1$ due to the Chinese Remainder Theorem. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{q_1+3}$.
- If $\beta = 1$,

  $$\mathbf{C}_0 = g_1^{\mathbf{c}(s\mathbf{s}',\mathbf{h})} g_2^{\mathbf{c}(v\mathbf{s}',\mathbf{h})}, C_1 = T_1 M_b$$

  $\mathrm{CT}^*_{Y^*}$ is a properly distributed, semi-functional and random ciphertext. In this case, $\mathcal{B}$ has properly simulated $\mathbf{G}_{final}$.

Although $\mathcal{A}$ is allowed to make key generation queries for X where $R(X, Y^*) = 1$, he receives secret keys under the falsified master key $\phi_i(\alpha)$ and ϕ_i cannot be the identity function. If $\mathcal{A}$ attempts to decrypt, he can only get $e(g_1, g_1)^{\mathrm{H}(\phi_i(\alpha))s}$. Due to property of the tampering resilient function, $\mathrm{H}(\alpha)$ is independent from $\mathrm{H}(\phi_i(\alpha))$ when $\phi_i(\alpha) \neq \alpha$. That is, these secret keys are useless for $\mathcal{A}$. Hence, $\mathcal{A}$ can distinguish $\mathbf{G}_{q+3}$ from $\mathbf{G}_{final}$ with negligible probability, otherwise $\mathcal{B}$ can break the Assumption **TRF-SD3**.

[7] Since the tampering functions ϕ_i are given before the security game, the challenger can generate properly distributed secret keys under the falsified master key with $g_1^{H(\phi_i(\alpha))} g_2^{u}$ received from the Assumption **TRF-SD3**.

Acknowledgements
Not applicable.

Authors' contributions
YL proposed the tampering attacks on PE, showed how to design secure PE against such attacks and drafted the manuscript. RZ and YZ participated in problem discussions and improvements of the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported in part by National Natural Science Foundation of China (No. 61632020, 61472416, 61772520), National key research and development program of China (No. 2017YFB0802705), Key Research Project of Zhejiang Province (No. 2017C01062), Fundamental Theory and Cutting-edge Technology Research Program of Institute of Information Engineering, CAS (No. Y7Z0321102).

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 11 September 2018 Accepted: 4 July 2019
Published online: 27 August 2019

References

Agrawal D, Archambeault B, Rao JR, Rohatgi P (2003) The EM Side—Channel(s). In: Kaliski BS, Koċ ċK, Paar C (eds). Lecture Notes in Computer Science. CHES, Springer Berlin Heidelberg, Berlin. pp 29–45

Agrawal S, Chase M (2016) A Study of Pair Encodings: Predicate Encryption in Prime Order Groups. In: Kushilevitz E, Malkin T (eds). Lecture Notes in Computer Science. TCC, Springer Berlin Heidelberg, Berlin. pp 259–288

Attrapadung N (2014) Dual System Encryption via Doubly Selective Security: Framework, Fully Secure Functional Encryption for Regular Languages, and More(Nguyen PQ, Oswald E, eds.). EUROCRYPT, Springer Berlin Heidelberg, Berlin

Attrapadung N (2016) Dual System Encryption Framework in Prime-Order Groups via Computational Pair Encodings(Cheon JH, Takagi T, eds.). ASIACRYPT, Springer Berlin Heidelberg, Berlin

Bellare M, Cash D (2010) Pseudorandom Functions and Permutations Provably Secure against Related-Key Attacks. In: Rabin T (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin. pp 666–684

Bellare M, Cash D, Miller R (2011) Cryptography Secure against Related-Key Attacks and Tampering(Lee DH, Wang X, eds.). ASIACRYPT, Springer Berlin Heidelberg, Berlin

Bellare M, Kohno T (2003) A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications(Biham E, ed.). EUROCRYPT, Springer Berlin Heidelberg, Berlin

Bellare M, Meiklejohn S, Thomson S (2014) Key-Versatile Signatures and Applications: RKA, KDM and Joint Enc/Sig(Nguyen PQ, Oswald E, eds.). EUROCRYPT, Springer Berlin Heidelberg, Berlin

Bellare M, Paterson KG, Thomson S (2012) RKA Security beyond the Linear Barrier: IBE, Encryption and Signatures(Wang X, Sako K, eds.). ASIACRYPT, Springer Berlin Heidelberg, Berlin

Biham E, Shamir A (1997) Differential fault analysis of secret key cryptosystems. In: Kaliski BS (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin. pp 513–525

Boneh D, DeMillo RA, Lipton RJ (1997) On the Importance of Checking Cryptographic Protocols for Faults. In: Fumy W (ed). Lecture Notes in Computer Science. EUROCRYPT, Springer Berlin Heidelberg, Berlin. pp 37–51

Boneh D, Franklin M (2001) Identity-Based Encryption from the Weil Pairing. In: Kilian J (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin. pp 213–229

Chen J, Gay R, Wee H (2015) Improved Dual System ABE in Prime-Order Groups via Predicate Encodings(Oswald E, Fischlin M, eds.). EUROCRYPT, Springer Berlin Heidelberg, Berlin

Damgård I, Faust S, Mukherjee P, Venturi D (2013) Bounded Tamper Resilience: How to Go beyond the Algebraic Barrier(Sako K, Sarkar P, eds.). ASIACRYPT, Springer Berlin Heidelberg, Berlin
Damgård I, Faust S, Mukherjee P, Venturi D (2015) The Chaining Lemma and Its Application(Lehmann A, Wolf S, eds.). ICITS, Springer International Publishing, Berlin
Faonio A, Venturi D (2016) Efficient Public-Key Cryptography with Bounded Leakage and Tamper Resilience. In: Cheon JH, Takagi T (eds). Lecture Notes in Computer Science. PKC, Springer Berlin Heidelberg, Berlin. pp 877–907
Faust S, Mukherjee P, Venturi D, Wichs D (2014) Efficient Non-malleable Codes and Key-Derivation for Poly-size Tampering Circuits. In: Nguyen PQ, Oswald E (eds). Lecture Notes in Computer Science. EUROCRYPT, Springer Berlin Heidelberg, Berlin. pp 111–128
Fujisaki E, Xagawa K (2015) Efficient RKA-Secure KEM and IBE Schemes Against Invertible Functions(Lauter K, Rodríguez-Henríquez F, eds.). LATINCRYPT, Springer International Publishing, Berlin
Fujisaki E, Xagawa K (2016) Public-Key Cryptosystems Resilient to Continuous Tampering and Leakage of Arbitrary Functions(Cheon JH, Takagi T, eds.). ASIACRYPT, Springer Berlin Heidelberg, Berlin
Gennaro R, Lysyanskaya A, Malkin T, Micali S, Rabin T (2004) Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In: Naor M (ed). Lecture Notes in Computer Science. TCC, Springer Berlin Heidelberg, Berlin. pp 258–277
Goyal V, O'Neill A, Rao V (2011) Correlated-Input Secure Hash Functions. In: Ishai Y (ed). Lecture Notes in Computer Science. TCC, Springer Berlin Heidelberg, Berlin. pp 182–200
Kalai YT, Kanukurthi B, Sahai A (2011) Cryptography with Tamperable and Leaky Memory. In: Rogaway P (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin. pp 373–390
Katz J, Sahai A, Waters B (2008) Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart N (ed). Lecture Notes in Computer Science. EUROCRYPT, Springer Berlin Heidelberg, Berlin. pp 146–162
Kocher P (1996) Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz N (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin Vol. 1109. pp 104–113
Kocher P, Jaffe J, Jun B (1999) Differential Power Analysis. In: Wiener M (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin Vol. 1666. pp 388–397
Lewko A, Okamoto T, Sahai A, Takashima K, Waters B (2010) Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert H (ed). Lecture Notes in Computer Science. EUROCRYPT, Springer Berlin Heidelberg, Berlin Vol. 6110. pp 62–91
Lewko A, Waters B (2010) New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio D (ed). Lecture Notes in Computer Science. TCC, Springer Berlin Heidelberg, Berlin Vol. 5978. pp 455–479
Liu FH, Lysyanskaya A (2012) Tamper and Leakage Resilience in the Split-State Model. In: Safavi-Naini R, Canetti R (eds). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin. pp 517–532
Matsuda T, Schuldt JCN (2018) Related Randomness Security for Public Key Encryption, Revisited. In: Abdalla M, Dahab R (eds). Lecture Notes in Computer Science. PKC, Springer Berlin Heidelberg, Berlin. pp 280–311
Okamoto T, Takashima K (2009) Hierarchical Predicate Encryption for Inner-Products(Matsui M, ed.). ASIACRYPT, Springer Berlin Heidelberg, Berlin
Okamoto T, Takashima K (2010) Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In: Rabin T (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin Vol. 6223. pp 191–208
Okamoto T, Takashima K (2012) Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In: Pointcheval D, Johansson T (eds). Lecture Notes in Computer Science. EUROCRYPT, Springer Berlin Heidelberg, Berlin. pp 591–608
Qin B, Liu S, Yuen TH, Deng RH, Chen K (2015) Continuous Non-malleable Key Derivation and Its Application to Related-Key Security. In: Katz J (ed). Lecture Notes in Computer Science. PKC, Springer Berlin Heidelberg, Berlin. pp 557–578
Waters B (2005) Efficient Identity-Based Encryption Without Random Oracles(Cramer R, ed.). EUROCRYPT, Springer Berlin Heidelberg, Berlin
Waters B (2009) Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi S (ed). Lecture Notes in Computer Science. CRYPTO, Springer Berlin Heidelberg, Berlin Vol. 5677. pp 619–636
Wee H (2012) Public Key Encryption against Related Key Attacks. In: Fischlin M, Buchmann J, Manulis M (eds). Lecture Notes in Computer Science. PKC, Springer Berlin Heidelberg, Berlin. pp 262–279
Wee H (2014) Dual System Encryption via Predicate Encodings. In: Lindell Y (ed). Lecture Notes in Computer Science. TCC, Springer Berlin Heidelberg, Berlin. pp 616–637

Publisher's Note