Robust Encryption, Extended

Rémi Géraud[1(✉)], David Naccache[1(✉)], and Răzvan Roşie[1,2(✉)]

[1] ENS, CNRS, INRIA, PSL Research University, Paris, France
{remi.geraud,david.naccache,razvan.rosie}@ens.fr
[2] University of Luxembourg, Esch-sur-Alzette, Luxembourg

Abstract. Robustness is a notion often tacitly assumed while working with encrypted data. Roughly speaking, it states that a ciphertext cannot be decrypted under different keys. Initially formalized in a public-key context, it has been further extended to key-encapsulation mechanisms, and more recently to pseudorandom functions, message-authentication codes and authenticated encryption. In this work, we motivate the importance of establishing similar guarantees for *functional encryption* schemes, even under adversarially generated keys. Our main security notion is intended to capture the scenario where a ciphertext obtained under a master key (corresponding to Authority 1) is decrypted by functional keys issued under a different master key (Authority 2). Furthermore, we show there exist simple functional encryption schemes where robustness under adversarial key-generation is not achieved. As a secondary and independent result, we formalize robustness for digital signatures – a signature should not verify under multiple keys – and point out that certain signature schemes are not robust when the keys are adversarially generated.

We present simple, generic transforms that turn a scheme into a robust one, while maintaining the original scheme's security. For the case of public-key functional encryption, we look into ciphertext anonymity and provide a transform achieving it.

Keywords: Robustness · Functional encryption · Signatures · Anonymity

1 Introduction

Cryptographic primitives, such as encryption and signature schemes, provide security guarantees under the condition, often left implicit, that they are "used correctly". Fatal examples of cryptographic misuse abound, from weak key generation to nonce-reuse. This reliance on operational security has attracted attackers, who can for instance impose faulty or backdoored random number generators to erode cryptographic protections. At the same time, the social usage of technology leans towards a more open environment than the one in which historic primitives were designed: keys are generated by one party, shared with another, certified by third... These two observations raise new interesting questions, which

M. Matsui (Ed.): CT-RSA 2019, LNCS 11405, pp. 149–168, 2019.
https://doi.org/10.1007/978-3-030-12612-4_8

have only recently been addressed in the cryptographic literature. For instance, if Alice generates keys that she is using, but doesn't share, can an adversary (observing Alice or influencing her in some way) nevertheless generate a *different* set of keys, which would allow decryption (maybe only partial)? Intuitively this should not be the case, but it was not until the seminal work of Abdalla, Bellare and Neven [1], that this situation was formally analysed. They introduced the notion of robustness, which ensures that a ciphertext cannot be decrypted under multiple keys.

Is robustness desirable? Imagine a scenario where users within a network exchange messages by broadcasting them, and further encrypt them with the public key of the recipient to ensure confidentiality. If this is the case, we usually assume that there is only one receiver, by arguing that no other members apart from the intended recipient can decrypt the ciphertext and obtain a valid (non-$\perp$) plaintext. But if the adversary can somehow tamper with the key generation process, she may "craft" keys that behave unexpectedly for some messages, or design alternative keys that give at least some information on some of the messages.

Farshim et al. [12] refined the original definition of robustness, by covering the cases where the keys are adversarially generated, under a master notion called "complete robustness". Mohassel addressed the question in the context of key-encapsulation mechanisms [19]. More recently, Farshim et al. also defined robustness for symmetric primitives [13], motivated by the security of oblivious transfer protocols [9] or message authentication codes. Further extensions of their security notions found applications in novel password-authenticated key-exchange protocols described by Jarecki et al. [17] or (fast) message-franking schemes [16]. The above line of work, however, leaves open several questions. Indeed, to the best of our knowledge there has been no notion of robustness defined for digital signatures [15], functional signatures [7] or functional encryption [6,20]. Yet, some existing schemes seem to be vulnerable to attacks that a proper notion of robustness would prevent.

Consider digital signature schemes (DS), that are used to authenticate electronic documents. The textbook notion, capturing the *existential unforgeability* of a DS ensures that an adversary, interacting with *one* signing oracle, cannot forge a signature (for a message he did not previously query). On the other hand, a real-world scenario is placed in a multi-user context, where it is often assumed (but not necessarily proven) that a signature can *only* be verified under the issuer's key.

Example 1: Consider a practical situation where a clerk has *acquired* a digital signature for daily use, with a third party generating the pairs of keys. Even if the scheme remains unforgeable according to the classical definition, we do not have formal guarantees that two pairs of keys—$(\mathsf{sk}, \mathsf{pk})$ and $(\mathsf{sk}', \mathsf{pk}')$—generated by the third party (potentially *malicious*), cannot be used to produce a signature σ for some *chosen message* M, verifiable under both pk and pk'—something completely undesirable in practice. To be fully explicit with our example, let us suppose one pair of keys $(\mathsf{pk}, \mathsf{sk})$ is given to the clerk and the second pair $(\mathsf{pk}', \mathsf{sk}')$, is issued by the third party and is covertly used by a local/global

security agency. When needed (and if needed), an operator can issue a signature (using sk') for the message: "I attest [...] is true." which can later be verified under pk, thus having baleful consequences for the clerk.

To give a flavour of a signature scheme where such an attack is feasible, consider the one obtained from a toy version of the Boneh–Boyen scheme [4]. The construction is *pairing*-based and can be summarized as follows: (1) key-generation samples two group generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, both of prime order p, and publishes as a public key $(g_1, g_2, g_2^x, e(g_1, g_2))$—for a uniformly sampled x over $\mathbb{Z}_p$—keeping x as a secret key. To sign the message M, one computes $\sigma \leftarrow g_1^{1/(x+M)}$. A robustness attack against this simple signature scheme exploits the randomness in choosing the secret keys, observing that for a different pair $(\mathsf{pk}', \mathsf{sk}')$, one can choose $g_1' \equiv g_1^t \pmod p$ and then can set $x' \equiv t(x+M) - M \pmod p$ such that $\sigma \equiv {g_1'}^{1/(x'+M)}$.

The above example provides the intuition that robustness has practical consequences. As expected, under *correct* key generation, standard unforgeability *does imply* robustness. But it fails in a malicious setting. Fortunately, we can provide a trivial construction that generically transforms any unforgeable signature scheme into a completely robust one (allowing for adversarial, yet well-formed keys). As we prove in Sect. 4.1, the natural idea of including the public key (or a collision-resistant hash of it) in the signature is indeed sufficient.

Speaking roughly about robustness as the property of a ciphertext of not being decryptable under multiple keys, then, when it comes to decryption, an FE scheme trivially does not exhibit this property. The reason resides in the broken symmetry to the way decryption works in symmetric/public-key schemes. Through its purpose, a functional ciphertext can be decrypted under multiple keys [6,20]. In this respect, an adversary holding multiple functional keys (which is not a restriction by itself) will be able to decrypt under multiple keys. Therefore, defining robustness in terms of decryption itself is fallacious. Instead, an appropriate definition should ensure the FE ciphertext can be decrypted only by the intended set of receivers.

Example 2: Consider a simple use case of a functional encryption scheme for the "inner product" function (IP FE) [2,3]. From a technical perspective, suppose the ciphertext is generated by encrypting a plaintext M as $C \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, M; R)$. If msk is somehow corrupted[1] to msk', then is it possible that performing decryption under sk'_y reveals a different plaintext $M' \neq M$? Intuitively, if the functional encryption scheme meets robustness, we expect that no ciphertext can be decrypted under functional keys issued by a *different* master secret keys.

As a concrete scenario, consider a Computer Science (CS) department's registry, which holds the marks obtained by each student in the Crypto course, the final grade being computed as a weighted average of the stored marks (i.e. homework counts 30%, midterm 20% and final 50%). *A priori* established confidentiality rules ask that a clerk should not have access to the marks, but still,

[1] There are several scenarios leading to such corruption, including memory corruption.

it must be possible to compute the final grade. Therefore, considering the set of marks as the vector $\boldsymbol{x}$ and the weights as $\boldsymbol{y}$, one can use an IP FE scheme, to obtain the final grade, its formula mapping to $\boldsymbol{x}^\top \cdot \boldsymbol{y}$. In order to achieve this, for each course: (1) the course leader encrypts the marks; (2) later, the clerk obtains a new key sk_y (depending on the established course weights), and uses it to obtain the final average. A failure to guarantee robustness could result in decryption to succeed, but the final average being incorrect (and possibly under the control of an adversary). To illustrate this, consider the (bounded-norm) IP FE scheme instantiated from ElGamal and introduced in [2]: encrypting a plaintext under $\mathsf{mpk} = (g^{s_1}, \dots, g^{s_n})$—where $\mathsf{msk} = \boldsymbol{s} = (s_1, \dots, s_n)$—is done as follows: $C \leftarrow\!\!\$\ (g^{-r}, g^{r \cdot s_1 + x_1}, \dots, g^{r \cdot s_n + x_n})$, for r sampled uniformly at random in $\mathbb{Z}_p$. If an attacker wishes to obtain the same C, then r remains the same, but it can use different $\boldsymbol{s'}$ and $\boldsymbol{x'}$, implicitly changing the value of msk. As expected, even if $\mathsf{FE.KDer}$ is correct, and the queried key is indeed issued for the vector $\boldsymbol{y}$, the final decrypted result corresponds to $\boldsymbol{x'}^\top \cdot \boldsymbol{y}$ rather than to $\boldsymbol{x}^\top \cdot \boldsymbol{y}$.

Our contributions. We begin by motivating and defining the notion of robust signature schemes under honest and adversarial keys, denoted as strong (SROB) and complete (CROB) robustness (Sect. 3.1). A natural question is whether existing schemes already possess a form of robustness: we show that while SROB is indeed typically guaranteed, it is not the case of CROB, thus providing a separation between the two security concepts. Fortunately, there exist a simple generic transform, in the standard model, that turn a SROB signature scheme into a CROB one (Sect. 4.1).

In Sect. 3.2, we define robustness for functional encryption in a multi-authority context. The strongest security notion we propose (FEROB) is intended to capture adversaries able to generate the keys and the randomness used during encryption and key-derivation, while remaining as simple as possible. As regards the generic transforms, we provide them in the public and private-key paradigms Sect. 4.2. The case for private-key FE schemes [8,18] relies on right-injective PRGs and collision-resistant PRFs, concepts that we review in Sect. 2. Finally, in the original spirit of the security notion we consider, we discuss anonymity for the context of functional encryption schemes.

2 Preliminaries

Notations. We denote the security parameter by $\lambda \in \mathbb{N}^*$ and we assume it is implicitly given to all algorithms in the unary representation 1^λ. An algorithm is equivalent to a Turing machine. Algorithms are assumed to be randomized unless stated otherwise; PPT stands for "probabilistic polynomial-time," in the security parameter (rather than the total length of its inputs). Given a randomized algorithm $\mathcal{A}$ we denote the action of running $\mathcal{A}$ on input(s) $(1^\lambda, x_1, \dots)$ with uniform random coins r and assigning the output(s) to $(y_1, \dots)$ by $(y_1, \dots) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, x_1, \dots ; r)$. When $\mathcal{A}$ is given oracle access to some procedure $\mathcal{O}$, we write $\mathcal{A}^{\mathcal{O}}$. For a finite set S, we denote its cardinality by $|S|$ and the action of sampling a uniformly at random element x from X by $x \leftarrow\!\!\$\ X$.

We define $[k] \coloneqq \{1, \ldots, k\}$. A real-valued function $\text{NEGL}(\lambda)$ is negligible if $\text{NEGL}(\lambda) \in \mathcal{O}(\lambda^{-\omega(1)})$. We denote the set of all negligible functions by NEGL. Throughout the paper $\perp$ stands for a special error symbol, while $||$ denotes concatenation. For completeness, we recall below definitions for the more important concepts to be used throughout the paper.

2.1 (Right-Injective) Pseudorandom Generators

Definition 1. *A pseudorandom generator* $\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^{n+\ell}$ *takes as input a random seed* s *of length* n *and outputs a pseudorandom binary string of length* $n+\ell$*. We require a negligible advantage for any* PPT *adversary* $\mathcal{A}$ *against the* PRG *security experiment defined in* Fig. 1*:*

$$\mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{A},\mathsf{PRG}}(\lambda) \coloneqq 2 \cdot \Pr\left[\mathsf{PRG}^{\mathcal{A}}_{\mathsf{PRG}}(\lambda) = 1\right] - 1 \in \text{NEGL}(\lambda).$$

RIGHT-INJECTIVE PRGS. We will make use of length-doubling, right-injective PRGs, where the right-injectivity condition is defined as

$$R_2 = R_2' \implies s = s'$$

for $R_1||R_2 \leftarrow \mathsf{PRG}(s)$ and $R_1'||R_2' \leftarrow \mathsf{PRG}(s')$. Such constructions can be achieved assuming the existence of one-way permutations, as shown by Yao [21].

2.2 (Collision-Resistant) Pseudorandom Functions

The notion of a pseudorandom function (PRF), introduced in the seminal work of Goldreich, Goldwasser, and Micali [14], is a foundational building block in theoretical cryptography. A PRF is a *keyed* functionality guaranteeing the randomness of its output under various assumptions. PRFs found applications in the construction of both symmetric and public-key primitives.

Definition 2. *A* PRF *is a pair of* PPT *algorithms* $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ *such that:*

- $\mathsf{sk} \leftarrow\!\!\$\ \mathsf{PRF.Gen}(1^\lambda)$*: is the randomized procedure that samples a secret key* sk*, given as input the unary version of the security parameter.*
- $y \leftarrow \mathsf{PRF.Eval}(\mathsf{sk}, M)$*: is the deterministic procedure that outputs* y*, corresponding to the evaluation of* M *under* sk*.*

We require the advantage of any PPT *adversary* $\mathcal{A}$ *in the* PRF *security experiment defined in* Fig. 1 *to be negligible:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathcal{A},\mathsf{PRF}}(\lambda) \coloneqq 2 \cdot \Pr\left[\mathrm{PRF}^{\mathcal{A}}_{\mathsf{PRF}}(\lambda)\right] - 1 \in \text{NEGL}(\lambda).$$

COLLISION-RESISTANT PRFS. We make use of collision-resistant PRFs [13]. The collision-resistance property is defined over both the secret-keys and the inputs:

$$\mathsf{PRF.Eval}(\mathsf{sk}, M) = \mathsf{PRF.Eval}(\mathsf{sk}', M') \implies (\mathsf{sk}, M) = (\mathsf{sk}', M').$$

Such constructions can be obtained for instance from key-injective PRFs via the GGM construction - see for instance [10, Appendix C] and length-doubling right-injective PRGs.

$\underline{\mathsf{PRG}^{\mathcal{A}}_{\mathsf{PRG}}(\lambda)}$:
$b \leftarrow_{\$} \{0,1\}$
$s \leftarrow_{\$} \{0,1\}^n$
$y \leftarrow \mathsf{PRG}(s)$
if $b = 0$ then
  $y \leftarrow_{\$} \{0,1\}^{n+l}$
$b' \leftarrow_{\$} \mathcal{A}(y)$
return $b' = b$

$\underline{\mathrm{PRF}^{\mathcal{A}}_{\mathsf{PRF}}(\lambda)}$:
$b \leftarrow_{\$} \{0,1\}$
$\mathrm{L} \leftarrow \emptyset$
$\mathsf{sk} \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$b' \leftarrow_{\$} \mathcal{A}^{\textsc{Eval}}(1^\lambda)$
return $b' = b$

Proc. $\textsc{Eval}(M)$:
if $M \in \mathrm{L}$ then return $\perp$
$y \leftarrow \mathsf{Eval}(\mathsf{sk}, M)$
if $b = 0$ then
  $y \leftarrow_{\$} \{0,1\}^{|y|}$
$\mathrm{L} \leftarrow \mathrm{L} \cup \{M\}$
return y

$\underline{\mathrm{ANON}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)}$:
$b \leftarrow_{\$} \{0,1\}$
$(\mathsf{mpk}_0,\ \mathsf{msk}_0) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$(\mathsf{mpk}_1,\ \mathsf{msk}_1) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$M \leftarrow_{\$} \mathcal{A}^{\mathsf{mpk}_0, \mathsf{mpk}_1}(1^\lambda)$
$C \leftarrow_{\$} \mathsf{Enc}(\mathsf{mpk}_b, M)$
$b' \leftarrow_{\$} \mathcal{A}(1^\lambda, C)$
return $b = b'$

Fig. 1. Experiments defining pseudorandomness for PRGs (left) and PRFs (middle). Anonymity for public-key functional encryption is defined on the right.

2.3 Functional Encryption

Definition 3 (Functional Encryption Scheme - Public-Key Setting). *A functional encryption scheme* FE *in the public-key setting consists of a tuple of* PPT *algorithms* (Setup, Gen, KDer, Enc, Dec) *such that:*

- $\mathsf{pars} \leftarrow_{\$} \mathit{FE}.\mathsf{Setup}(1^\lambda)$*: we assume the existence of a* Setup *algorithm producing a set of public parameters which are implicitly given to all algorithms. When omitted, the output of* $\mathit{FE}.\mathsf{Setup}$ *is* $\emptyset$*.*
- $(\mathsf{msk}, \mathsf{mpk}) \leftarrow_{\$} \mathit{FE}.\mathsf{Gen}(1^\lambda)$ *: takes as input the unary representation of the security parameter* λ *and outputs a pair of master secret/public keys.*
- $\mathsf{sk}_f \leftarrow_{\$} \mathit{FE}.\mathsf{KDer}(\mathsf{msk}, f)$*: given the master secret key and a function* f*, the (possibly randomized) key-derivation procedure outputs a corresponding* sk_f*.*
- $C \leftarrow_{\$} \mathit{FE}.\mathsf{Enc}(\mathsf{mpk}, M)$*: the randomized encryption procedure encrypts the plaintext* M *with respect to* mpk*.*
- $\mathit{FE}.\mathsf{Dec}(\mathsf{sk}_f, C)$*: decrypts the ciphertext* C *using the functional key* sk_f *in order to learn a valid message* $f(M)$ *or a special symbol* $\perp$*, in case the decryption procedure fails.*

A functional encryption scheme is s-IND-FE-CPA*-secure if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the* IND-FE-CPA*-game defined in* Fig. 2 *is negligible:*

$$\mathsf{Adv}^{\text{s-IND-FE-CPA}}_{\mathcal{A},\mathit{FE}}(\lambda) := 2 \cdot \Pr\left[\text{s-IND-FE-CPA}^{\mathcal{A}}_{\mathit{FE}}(\lambda) = 1\right] - 1 \in \textsc{Negl}(\lambda).$$

Similarly we say that it is adaptive IND-FE-CPA*-secure if*

$$\mathsf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A},\mathit{FE}}(\lambda) := 2 \cdot \Pr\left[\text{IND-FE-CPA}^{\mathcal{A}}_{\mathit{FE}}(\lambda) = 1\right] - 1 \in \textsc{Negl}(\lambda).$$

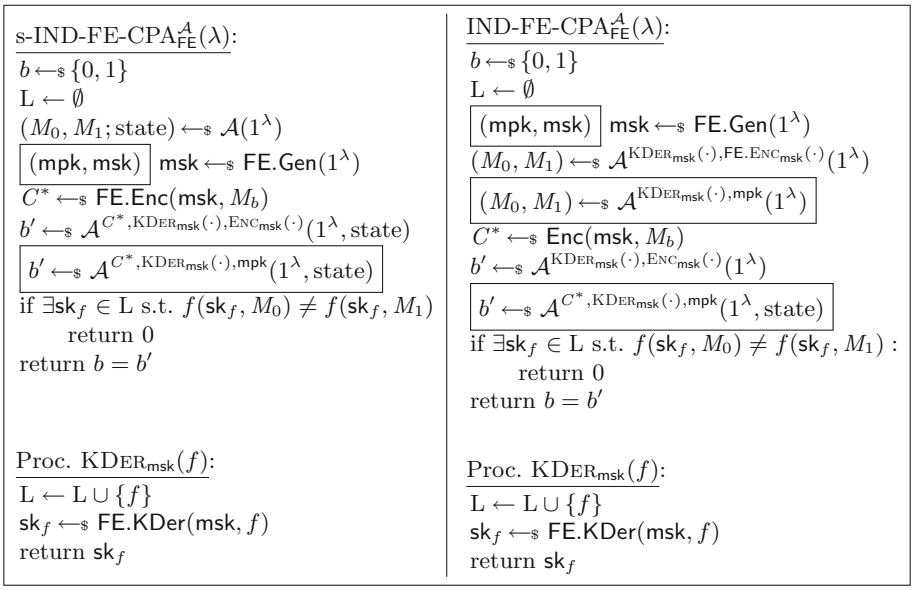

Fig. 2. The selective and adaptive indistinguishability experiments defined for a functional encryption scheme. The difference between the private-key and the public settings are marked in boxed lines of codes, corresponding to the latter notion.

Functional encryption can be defined in a private-key setting: the master secret key msk is used to encrypt the plaintext M, as there is no mpk.

ANONYMITY. We define the classical notion of anonymity to the context of functional encryption and its security experiment in Fig. 1 (right). We point out that usually, in an FE scheme, a central authority answers key-derivation queries from a potential set of users $\mathcal{U}$, therefore it is unnatural to assume that a user does not know from whom it received the functional key. What we want to ensure is that an adversary $\mathcal{A} \notin \mathcal{U}$ cannot tell *which* central authority has issued a ciphertext, without interacting with the key-derivation procedures, otherwise the game becomes trivial. As an easy consequence, anonymity makes sense only in the context of public-key FE, as for a private scheme, the adversary uses encryption oracles to obtain a ciphertext. Thus, anonymity requires that a PPT bounded adversary can tell which mpk was used to encrypt a ciphertext only with negligible probability: $\mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A},\mathsf{FE}}(\lambda) \coloneqq 2 \cdot \Pr\left[\mathrm{ANON}^{\mathcal{A}}_{\mathsf{FE}}(\lambda) = 1\right] - 1 \in \mathrm{NEGL}(\lambda)$.

3 Robustness: Definitions, Implications and Separations

Robustness guarantees hardness in finding ciphertexts (resp. signatures) generated under adversarial, but well-formed keys, decryptable (resp. verifiable) under multiple secret (resp. public) keys. As stated in the introductory part, this property is often tacitly presumed, but almost as often left without a proof. In this

work, we capture two levels of strengths of an adversary: *strong* robustness models the case where the keys are honestly generated and the adversary is agnostic of their actual values, the interaction being interfaced through decryption/signing oracles. A related, stronger notion, dubbed *complete* robustness gives an adversary the ability to generate keys (not necessarily honestly). In this work, we restrict to the cases where the keys are malicious, but well-formed[2].

We commence by presenting the security definition for digital signatures in Sect. 3.1, and then for functional encryption in Sect. 3.2.

3.1 Warm-Up: Robustness for Digital Signatures

The case for digital signatures is treated with respect to two security notions, which we denote strong and complete robustness. The winning condition remains the same in both experiments: that of obtaining a signature/message pair in such a way that it verifies under both public keys. In the SROB experiment, two signing oracles under $\mathsf{sk}_1, \mathsf{sk}_2$ are given to the adversary, while a CROB adversary generates its intrinsic keys for accomplishing essentially the same break.

$\underline{\mathrm{SROB}^{\mathcal{A}}_{\mathsf{DS}}(\lambda)}$:
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(M, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Sign}_{\mathsf{sk}_1}(\cdot), \mathrm{Sign}_{\mathsf{sk}_2}(\cdot)}(1^\lambda, \mathsf{pk}_1, \mathsf{pk}_2)$
if $\mathsf{Ver}(\mathsf{pk}_1, \sigma, M) = 1 \wedge$
$\quad \mathsf{Ver}(\mathsf{pk}_2, \sigma, M) = 1$:
$\quad$ return 1
return 0

$\underline{\mathrm{CROB}^{\mathcal{A}}_{\mathsf{DS}}(\lambda)}$:
$(\mathsf{pk}_1, \mathsf{pk}_2, \sigma, M) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda)$
if $\mathsf{pk}_1 = \mathsf{pk}_2$:
$\quad$ return 0
if $\mathsf{Ver}(\mathsf{pk}_1, \sigma, M) = 1 \wedge$
$\quad \mathsf{Ver}(\mathsf{pk}_2, \sigma, M) = 1$:
$\quad$ return 1
return 0

Fig. 3. Games defining strong robustness SROB (left) and complete robustness CROB (right) for a digital signature scheme DS. We assume a negligible probability of sampling $\mathsf{pk}_1 = \mathsf{pk}_2$ in the SROB game.

Definition 4 (SROB and CROB Security). *Let* DS *be a digital signature scheme. We say* DS *achieves complete robustness if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the CROB game depicted in* Fig. 3 *(right side) is negligible:* $\mathsf{Adv}^{\mathrm{CROB}}_{\mathcal{A},\mathsf{DS}}(\lambda) := \Pr\left[\mathrm{CROB}^{\mathcal{A}}_{\mathsf{DS}}(\lambda) = 1\right]$. SROB*-security is defined similarly, the* $\mathrm{SROB}^{\mathcal{A}}_{\mathsf{DS}}(\lambda)$ *game being defined in* Fig. 3 *(left side).*

Notice the *difference* to the classical unforgeability game where the adversary obtains signatures issued under the *same* secret key. We prove any EUF-scheme is implicitly strong-robust, and show there exist signature schemes that fail to achieve complete robustness (thus providing a separation between the two).

[2] We may assume that malformed keys would be easily recognisable and rejected.

Proposition 1. *Let* DS *be a* CROB*-secure digital signature scheme. Then* DS *is also* SROB*-secure, the advantage of breaking the strong robustness game being bounded as follows:* $\mathsf{Adv}^{\mathrm{SROB}}_{\mathcal{A},\mathsf{DS}}(\lambda) \leq \mathsf{Adv}^{\mathrm{CROB}}_{\mathcal{A}',\mathsf{DS}}(\lambda)$.

Proof (Proposition 1). Suppose DS is not SROB-secure. Let $\mathcal{A}$ be a PPT adversary that wins the SROB game with advantage at most ϵ_{SROB}. We construct a PPT adversary $\mathcal{A}'$ against the CROB game as follows: (1) sample two pairs of keys $(\mathsf{sk}_1, \mathsf{pk}_1), (\mathsf{sk}_2, \mathsf{pk}_2)$ using $\mathsf{Gen}(1^\lambda)$; (2) $\mathcal{A}'$ publishes $\mathsf{pk}_1, \mathsf{pk}_2$ and constructs the signing oracles $\mathsf{Sign}_{\mathsf{sk}_1}(\cdot)$ and $\mathsf{Sign}_{\mathsf{sk}_2}(\cdot)$; (3) $\mathcal{A}'$ runs $\mathcal{A}$ w.r.t. signing oracles and public-keys to obtain (M, σ); (4) $\mathcal{A}'$ constructs the tuple $(\mathsf{pk}_1, \mathsf{pk}_2, \sigma, M)$ and outputs it. We obtain that $\mathsf{Adv}^{\mathrm{SROB}}_{\mathcal{A}',\mathsf{DS}}(\lambda) \leq \mathsf{Adv}^{\mathrm{CROB}}_{\mathcal{A},\mathsf{DS}}(\lambda)$. □

Of interest, is a minimal level of robustness achieved by any digital signature scheme, and as it turns out, SROB is accomplished.

Lemma 1. *Any* EUF*-secure digital signature scheme* DS *is* SROB*-secure. The advantage of breaking the* SROB *game is bounded by the advantage of breaking the* EUF *game:* $\mathsf{Adv}^{\mathrm{SROB}}_{\mathcal{A},\mathsf{DS}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathrm{EUF}}_{\mathcal{A}',\mathsf{DS}}(\lambda)$.

Algorithm $\mathcal{A}'_{\mathcal{A}}(\lambda, \mathsf{pk}_1, \mathrm{Sign}_{\mathsf{sk}_1}(\cdot))$:
$(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
build $\mathrm{Sign}_{\mathsf{sk}_2}(\cdot)$
$(M, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Sign}_{\mathsf{sk}_1}(\cdot), \mathrm{Sign}_{\mathsf{sk}_2}(\cdot)}(\mathsf{pk}_1, \mathsf{pk}_2)$
if $M \in \mathsf{Sign}_{\mathsf{sk}_1}(\cdot).\mathrm{SignedMessages}()$
    abort
return (M, σ)

Fig. 4. The reduction $\mathcal{A}'$ in Lemma 1.

Proof (Lemma 1). Let $\mathcal{A}$ be a PPT adversary against the strong robustness game. Let $\mathcal{A}'$ stand for an adversary against the unforgeability of the digital signature. We assume without loss of generality that $\mathcal{A}$: (1) never queries a "winning" message M to the second signing oracle after it has been signed by the first oracle (since it can check it right away) and (2) it never queries a "winning" message M to the first oracle after it has been signed by the second oracle (for the same reason). We present the reduction in Fig. 4 and describe it below:

1. The EUF game proceeds by sampling $(\mathsf{sk}_1, \mathsf{pk}_1)$ and builds a signing oracle $\mathrm{Sign}_{\mathsf{sk}_1}(\cdot)$.
2. The reduction $\mathcal{A}'$ is given pk_1 and oracle access to the $\mathrm{Sign}_{\mathsf{sk}_1}(\cdot)$. $\mathcal{A}'$ samples uniformly at random $(\mathsf{sk}_2, \mathsf{pk}_2)$ via $\mathsf{DS.Gen}$ and constructs a second signing oracle $\mathrm{Sign}_{\mathsf{sk}_2}(\cdot)$.
3. $\mathcal{A}'$ runs $\mathcal{A}$ w.r.t. the two $(\mathsf{pk}_1, \mathsf{pk}_2)$ and the corresponding signing oracles $\mathrm{Sign}_{\mathsf{sk}_1}(\cdot), \mathrm{Sign}_{\mathsf{sk}_2}(\cdot)$. $\mathcal{A}'$ keeps track of the queried messages to each oracle.
4. $\mathcal{A}$ returns a pair (σ, M) which verifies under both public keys with probability ϵ_{SROB}, s.t. M has been queried to either $\mathrm{Sign}_{\mathsf{sk}_1}$ or $\mathrm{Sign}_{\mathsf{sk}_2}$ but not to both.
5. $\mathcal{A}'$ returns (σ, M). If $M \in \mathsf{Sign}_{\mathsf{sk}_1}(\cdot).\mathrm{SignedMessages}()$, $\mathcal{A}'$ aborts and runs $\mathcal{A}$ again. With probability $\frac{1}{2}$, M was not queried before to $\mathsf{Sign}_{\mathsf{sk}_1}(\cdot)$. The tuple (σ, M) wins the EUF game w.r.t. $(\mathsf{pk}_1, \mathsf{sk}_1)$ with probability $\geq \frac{1}{2} \cdot \epsilon_{\mathrm{SROB}}$.

Thus, the reduction (Fig. 4) shows the advantage of breaking SROB is bounded by advantage breaking EUF, which completes the proof. □

We also show a separation between the SROB and CROB, by pointing to a signature scheme that is not CROB secure (but already SROB).

Proposition 2. *There exist DS schemes that are not* CROB*-secure.*

Proof (Proposition 2). We provide a simple counterexample as follows. Consider the digital signature scheme in [5]:

- Gen: selects uniformly at random $g_1 \leftarrow_\$ \mathbb{G}_1, g_2 \leftarrow_\$ \mathbb{G}_2$ and $(x, y) \leftarrow_\$ \mathbb{Z}_p^2$. Set $\mathsf{sk} \leftarrow (x, y)$ and $\mathsf{pk} \leftarrow (g_1, g_2, g_2^x, g_2^y, e(g_1, g_2))$, where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a pairing[3].
- Sign: given a message M, sample $r \leftarrow_\$ \mathbb{Z}_p$ and compute $\sigma \leftarrow g_1^{1/(x+M+yr)}$. Note that with overwhelming probability, $x + M + yr \neq 0 \bmod p$, where p is the order of $\mathbb{G}_1$. The signature is the pair (σ, r).
- Verify: check that $e\left(\sigma, g_2^x \cdot g_2^M \cdot (g_2^y)^r\right) \stackrel{?}{=} e(g_1, g_2)$.

To win the CROB game, an adversary $\mathcal{A}$ proceeds as follows:

1. $\mathcal{A}$ samples a key-pair: $\mathsf{sk} \leftarrow_\$ (x, y); \mathsf{pk} \leftarrow (g_1, g_2, g_2^x, g_2^y, e(g_1, g_2))$ and a message $M \in \mathbb{Z}_p$.
2. $\mathcal{A}$ samples $r \leftarrow_\$ \mathbb{Z}_p$ and computes σ under sk_1. Since g'_1 can be written as g_1^t, $\mathcal{A}$ sets t, x', y' such that $1/(x + M + yr) = t/(x' + M + y'r)$ (equate the exponents to obtain the same σ corresponding to M). This can be done by assigning random values to x', y' and setting $t \leftarrow (x' + M + y'r)/(x + M + yr)$.
3. $\mathcal{A}$ sets $\mathsf{sk}' \leftarrow_\$ (x', y'); \mathsf{pk}' \leftarrow (g'_1, g'_2, g'^{x'}_2, g'^{y'}_2, e(g'_1, g'_2))$, for some uniformly sampled generator $g'_2 \leftarrow_\$ \mathbb{G}_2$.
4. Finally, observe that (σ, r) verifies under $(\mathsf{sk}_1, \mathsf{pk}_1)$ through the correctness of the signature scheme, but also under $(\mathsf{pk}_2, \mathsf{sk}_2)$, since

$$e\left(g_1^{t/(x'+M+y'r)}, g'^{x'}_2 \cdot g'^{M}_2 \cdot (g'^{y'}_2)^r\right) = e(g_1^t, g'_2).$$

   $\mathcal{A}$ halts and returns $(\mathsf{pk}, \mathsf{pk}', (\sigma, r), M)$. Note that $\mathcal{A}$ runs in probabilistic polynomial time. □

3.2 Robustness for Functional Encryption

As discussed in the motivational part of Sect. 1, robustness should be considered as a security notion achieved by a functional encryption scheme. In what follows, we define it for the public/private key settings. We stress about the existence of essentially two major paths one can explore. A first stream of work would study the meaning of robustness in a single-authority context.

[3] See for instance [5] for the definition and usage of a cryptographic pairing.

Multi-Authority Setting. A second path is placed in a multi-authority context—that is, assuming there exist multiple pairs $(\mathsf{msk}, \mathsf{mpk})$. Aiming for a correct definition, one property that should be guaranteed is that a ciphertext should not be decryptable under *two* (*or more*) functional keys issued via *different* master secret keys. Stated differently, if msk_1 produces sk_{f_1} and $\mathsf{msk}_2 \neq \mathsf{msk}_1$ produces sk_{f_2} for two functionalities f_1, f_2, we do not want that C (say encrypted under mpk_1) to be decrypted under sk_{f_2} (it already decrypts under sk_{f_1} with high probability due to the correctness of the scheme). We follow the lines of Definition 4, and propose two new flavours of robustness, corresponding to the cases where the adversary has oracle access to the (encryption, if in a private key setting case), key-derivation and decryption oracles. The security experiments are depicted in Fig. 5. The difference between the two paradigms may seem minor (for our purpose), but in fact having a *public* master key confers a significant advantage when it comes to deriving a generic transform for achieving complete robustness, as detailed in Sect. 4. In what follows, we will explore the multi-authority path, since it naturally maps to our motivational examples.

Intermediate Notions. Intermediate notions considering robustness under adversarially generated keys introduced in [12]—such as full-robustness or mixed robustness—do not generalize well to functional encryption (or attribute-based encryption). The notion we consider, namely FEROB is in fact the generalization of KROB (key-less robustness), as introduced for PKE by Farshim et al. [12].

Definition 5 (SROB and FEROB Security for FE). *Let FE be a functional encryption scheme. We say* FE *achieves functional robustness if the advantage of any* PPT *adversary* $\mathcal{A}$ *against the* FEROB *game defined in* Fig. 5 *(bottom) is negligible:* $\mathsf{Adv}^{\mathrm{FEROB}}_{\mathcal{A},\mathsf{Pub}/\mathsf{Prv}FE}(\lambda) \coloneqq \Pr\left[\mathrm{FEROB}^{\mathcal{A}}_{\mathsf{Pub}/\mathsf{Prv}FE}(\lambda) = 1\right]$. SROB*-security is defined similarly, the* $\mathrm{SROB}^{\mathcal{A}}_{\mathsf{Pub}/\mathsf{Prv}FE}(\lambda)$ *game being defined in* Fig. 5 *(top).*

As stated in the algorithmic description of the security experiment, an adversary against the strongest notion of FEROB attempts to find colliding ciphertexts, which decrypt under two msk-separated keys $\mathsf{sk}_{f_1}, \mathsf{sk}_{f_2}$.

Lemma 2 (Implications). *Let FE denote a functional encryption scheme. If FE is* FEROB*-secure, then it is also* SROB*-secure.*

Proof (Lemma 2). We prove the implication holds in both the public and private key settings:

Public-Key FE. We take the contrapositive. For a scheme FE, we assume the existence of an adversary $\mathcal{A}$ winning the SROB-game with non-negligible advantage ϵ_{SROB}. A reduction $\mathcal{A}'$ that wins the FEROB game is built as follows: (1) $\mathcal{A}'$ samples uniformly at random $(\mathsf{msk}_1, \mathsf{mpk}_1, \mathsf{msk}_2, \mathsf{mpk}_2)$; (2) the corresponding oracles for key-derivation are built; (3) $\mathcal{A}$ runs with access to the aforementioned oracles, returning $(C, \mathsf{sk}_{f_1}, \mathsf{sk}_{f_2})$. If $\mathcal{A}$ outputs a winning tuple, then $\mathcal{A}'$ wins the FEROB game by releasing the messages and the randomness terms used to construct $(C, \mathsf{sk}_{f_1}, \mathsf{sk}_{f_2})$. Hence, $\mathsf{Adv}^{\mathrm{SROB}}_{\mathcal{A},\mathsf{FE}}(\lambda) \leq \mathsf{Adv}^{\mathrm{FEROB}}_{\mathcal{A}',\mathsf{FE}}(\lambda)$.

$\underline{\text{SROB}^{\mathcal{A}}_{\mathsf{PubFE}}(\lambda)}$:
$\mathrm{L}_1 \leftarrow \emptyset$
$\mathrm{L}_2 \leftarrow \emptyset$
$(\mathsf{mpk}_1, \mathsf{msk}_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(\mathsf{mpk}_2, \mathsf{msk}_2) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(C, \mathsf{sk}_{f_1}, \mathsf{sk}_{f_2}) \leftarrow\!\!\$$
$\quad \leftarrow\!\!\$\ \mathcal{A}\left(\begin{array}{l} \\ \\ \\ \mathrm{KDER}_{\mathsf{msk}_1}(\cdot), \\ \mathrm{KDER}_{\mathsf{msk}_2}(\cdot) \end{array}\right)(\mathsf{mpk}_1, \mathsf{mpk}_2)$
if $\mathsf{sk}_{f_1} \in \mathrm{L}_2 \vee \mathsf{sk}_{f_2} \in \mathrm{L}_1$:
  return 0
if $\mathsf{Dec}(C, \mathsf{sk}_{f_1}) \neq \bot \wedge$
    $\mathsf{Dec}(C, \mathsf{sk}_{f_2}) \neq \bot$:
  return 1
return 0

$\underline{\mathrm{KDER}_{\mathsf{msk}_i}(f)}$:
$\mathsf{sk}_f \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_i, f)$
$\mathrm{L}_i \leftarrow \mathrm{L}_i \cup \{(\mathsf{sk}_f, f)\}$
return sk_f

$\underline{\mathrm{ENC}_{\mathsf{mpk}_i}(M)}$:
$C \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{mpk}_i, M)$
return C

$\underline{\text{SROB}^{\mathcal{A}}_{\mathsf{PrvFE}}(\lambda)}$:
$\mathrm{L}_1 \leftarrow \emptyset$
$\mathrm{L}_2 \leftarrow \emptyset$
$\mathsf{msk}_1 \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$\mathsf{msk}_2 \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$(C, \mathsf{sk}_{f_1}, \mathsf{sk}_{f_2}) \leftarrow\!\!\$$
$\quad \leftarrow\!\!\$\ \mathcal{A}\left(\begin{array}{l} \mathrm{ENC}_{\mathsf{msk}_1}(\cdot), \\ \mathrm{ENC}_{\mathsf{msk}_2}(\cdot), \\ \mathrm{KDER}_{\mathsf{msk}_1}(\cdot), \\ \mathrm{KDER}_{\mathsf{msk}_2}(\cdot) \end{array}\right)(1^\lambda)$
if $\mathsf{sk}_{f_1} \in \mathrm{L}_2 \vee \mathsf{sk}_{f_2} \in \mathrm{L}_1$:
  return 0
if $\mathsf{Dec}(C, \mathsf{sk}_{f_1}) \neq \bot \wedge$
    $\mathsf{Dec}(C, \mathsf{sk}_{f_2}) \neq \bot$:
  return 1
return 0

$\underline{\mathrm{KDER}_{\mathsf{msk}_i}(f)}$:
$\mathsf{sk}_f \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_i, f)$
$\mathrm{L}_i \leftarrow \mathrm{L}_i \cup \{(\mathsf{sk}_f, f)\}$
return sk_f

$\underline{\mathrm{ENC}_{\mathsf{msk}_i}(M)}$:
$C \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{msk}_i, M)$
return C

$\underline{\text{FEROB}^{\mathcal{A}}_{\mathsf{PubFE}}(\lambda)}$:
$(\mathsf{mpk}_1, \mathsf{msk}_1, R_1, M_1, f_1, R_{f_1},$
  $\mathsf{mpk}_2, \mathsf{msk}_2, R_2, M_2, f_2, R_{f_2}) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda)$
$C_1 \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{mpk}_1, M_1; R_1)$
$C_2 \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{mpk}_2, M_2; R_2)$
if $C_1 = C_2 \wedge \mathsf{mpk}_1 \neq \mathsf{mpk}_2$:
  $\mathsf{sk}_{f_1} \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_1, f_1; R_{f_1})$
  $\mathsf{sk}_{f_2} \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_2, f_2; R_{f_2})$
  if $\mathsf{Dec}(C, \mathsf{sk}_{f_1}) \neq \bot \wedge$
      $\mathsf{Dec}(C, \mathsf{sk}_{f_2}) \neq \bot$:
    return 1
return 0

$\underline{\text{FEROB}^{\mathcal{A}}_{\mathsf{PrvFE}}(\lambda)}$:
$(\mathsf{msk}_1, R_1, M_1, f_1, R_{f_1},$
  $\mathsf{msk}_2, R_2, M_2, f_2, R_{f_2}) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda)$
$C_1 \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{msk}_1, M_1; R_1)$
$C_2 \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{msk}_2, M_2; R_2)$
if $C_1 = C_2 \wedge \mathsf{msk}_1 \neq \mathsf{msk}_2$:
  $\mathsf{sk}_{f_1} \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_1, f_1; R_{f_1})$
  $\mathsf{sk}_{f_2} \leftarrow\!\!\$\ \mathsf{KDer}(\mathsf{msk}_2, f_2; R_{f_2})$
  if $\mathsf{Dec}(C, \mathsf{sk}_{f_1}) \neq \bot \wedge$
      $\mathsf{Dec}(C, \mathsf{sk}_{f_2}) \neq \bot$:
    return 1
return 0

Fig. 5. We introduce FEROB and SROB in the context of FE schemes defined both in the public and private key setting. For the SROB games, we give the oracles implementing Enc and KDer procedures, mentioning that each query to the latter oracle adds an entry of the form (f, sk_f) in the corresponding list L_i—where $i \in \{1, 2\}$ stands for the index of the used master keys.

PRIVATE-KEY FE. We take the contrapositive. For a scheme FE, we assume the existence of an adversary $\mathcal{A}$ winning the SROB-game with non-negligible advantage ϵ_{SROB}. A reduction $\mathcal{A}'$ that wins the FEROB game is built as follows:

(1) $\mathcal{A}'$ samples uniformly at random $(\mathsf{msk}_1, \mathsf{msk}_2)$; (2) $\mathcal{A}'$ constructs the encryption and key-derivation oracles under the two keys; (3) $\mathcal{A}'$ runs $\mathcal{A}$ with these oracles, records the random coins used and obtains $(C, \mathsf{sk}_{f_1}, \mathsf{sk}_{f_2})$. Finally $\mathcal{A}'$ wins the FEROB game by issuing the FEROB tuple, using the random coins used to derive the functional keys and the ciphertext and therefore we have: $\mathsf{Adv}^{\mathrm{SROB}}_{\mathcal{A},\mathsf{FE}}(\lambda) \leq \mathsf{Adv}^{\mathrm{FEROB}}_{\mathcal{A}',\mathsf{FE}}(\lambda)$. □

Proposition 3 (Separations). *There exist functional encryption schemes in the public/private-key setting that are not* FEROB*-secure.*

Proof (Proposition 3). As sketched in Sect. 1, a DDH instantiation for the FE scheme of [2] is not FEROB-secure. The adversary is built upon the idea presented in the introduction and is shown in Fig. 6. Given that any public-key functional encryption scheme can be trivially converted into one in the private-key setting simply by making mpk private, we obtain an FE scheme for the inner product functionality in the private-key setting that is not FEROB-secure.

FEROB adversary $\mathcal{A}^{\mathrm{FEROB}}_{\mathsf{FE}}(\lambda)$:

1. $(g^{\boldsymbol{s}}, \boldsymbol{s}, r, \boldsymbol{x}, \boldsymbol{y}, \emptyset$
   $g^{\boldsymbol{s'}}, \boldsymbol{s'}, r, \boldsymbol{x'}, \boldsymbol{y}, \emptyset) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
   such that $r \cdot s_i + x_i = r \cdot s'_i + x'_i$ and $\boldsymbol{s} \neq \boldsymbol{s'}$
2. observe that $\mathsf{Enc}(g^{\boldsymbol{s}}, \boldsymbol{x}) = (g^{-r}, g^{r \cdot s_1 + x_1} \ldots, g^{r \cdot s_n + x_n}) =$
   $(g^{-r}, g^{r \cdot s'_1 + x'_1} \ldots, g^{r \cdot s'_n + x'_n}) = \mathsf{Enc}(g^{\boldsymbol{s'}}, \boldsymbol{x'})$
3. $\mathsf{sk}_y \leftarrow \boldsymbol{s}^\top \cdot \boldsymbol{y}$
4. $\mathsf{sk}'_y \leftarrow \boldsymbol{s'}^\top \cdot \boldsymbol{y}$
5. $\mathsf{Dec}(C, \mathsf{sk}_y) = \boldsymbol{y}^\top \cdot \boldsymbol{x} \neq \perp$
6. $\mathsf{Dec}(C, \mathsf{sk}'_y) = \boldsymbol{y}^\top \cdot \boldsymbol{x'} \neq \perp$

Fig. 6. A FEROB adversary against the DDH instantiation of the bounded-norm inner product scheme in [2].

□

4 Achieving Robustness via Generic Transforms

4.1 Robust Digital Signatures

We put forward a generic transform similar in spirit to the original work of Abdalla, Bellare, and Neven [1] in the context of digital signatures. For a digital signature scheme, we benefit from the fact that pk acts as an "immutable" value to which one can easily commit to, while signing a message. Thus, checking if a message verifies under another public key implicitly breaks the binding property of the commitment scheme. For simplicity, we use a hash instead of a commitment scheme.

$\overline{\mathsf{Gen}}(1^\lambda)$:
$(\mathsf{sk}, \mathsf{pk}) \leftarrow_\$ \mathsf{DS.Gen}(1^\lambda)$
$\overline{\mathsf{pk}} \leftarrow \mathsf{pk}$
$\overline{\mathsf{sk}} \leftarrow \mathsf{sk}$
return $(\overline{\mathsf{sk}}, \overline{\mathsf{pk}})$

$\overline{\mathsf{Sign}}(\overline{\mathsf{sk}}, M)$:
$\mathsf{sk} \leftarrow \overline{\mathsf{sk}}$
$\sigma_1 \leftarrow_\$ \mathsf{DS.Sign}(\mathsf{sk}, M)$
$\sigma_2 \leftarrow \mathsf{H}(\mathsf{pk})$
$\overline{\sigma} \leftarrow (\sigma_1, \sigma_2)$
return $\overline{\sigma}$

$\overline{\mathsf{Ver}}(\overline{\mathsf{pk}}, \overline{\sigma}, M)$:
$\mathsf{pk} \leftarrow \overline{\mathsf{pk}}$
$(\sigma_1, \sigma_2) \leftarrow \overline{\sigma}$
return $\mathsf{DS.Ver}(\mathsf{pk}, \sigma_1) = 1 \wedge \sigma_2 \stackrel{?}{=} \mathsf{H}(\mathsf{pk})$

$\overline{\mathsf{Setup}}(1^\lambda)$:
$K \leftarrow \mathsf{H.Gen}(1^\lambda)$; $\mathsf{H} \leftarrow \mathsf{H}_K$; return H

Fig. 7. A generic transform that turns any digital signature scheme DS into one that is, in addition, CROB-secure. The (publicly available) collision-resistant hash function H can be based on claw-free permutations in the standard model, as shown in the seminal work of Damgård [11]. It is used as a commitment to the public-key.

Lemma 3. *Let* DS *be an* EUF*-secure digital signature scheme. Let* H *denote a collision-resistant hash function. The digital signature* $\overline{\mathsf{DS}}$ *obtained through the transform depicted in Fig. 7 is* CROB*-secure.*

Proof (Lemma 3). We prove both the unforgeability and the complete robustness of the newly obtained construction:

Unforgeability. Assume the existence of a PPT adversary $\mathcal{A}$ against $\overline{\mathsf{DS}}$. We build an adversary $\mathcal{A}'$ against the EUF of the underlying DS. The unforgeability experiment EUF for DS samples $(\mathsf{pk}, \mathsf{sk})$ and constructs a signing oracle under sk, which is given to $\mathcal{A}'$. $\mathcal{A}'$ is given a collision resistant hash function H and builds its own signing oracle $\overline{\mathsf{Sign}}$; when queried, $\overline{\mathsf{Sign}}$ returns the output of Sign concatenated to the value of $\mathsf{H}(\mathsf{pk})$. When $\mathcal{A}$ replies with $(\overline{\sigma}, M)$, it must be the case that $\mathsf{Ver}(\mathsf{pk}, \sigma, M)$ passes, which breaks EUF for DS. Thus we conclude that: $\mathsf{Adv}^{\mathrm{EUF}}_{\mathcal{A},\overline{\mathsf{DS}}}(\lambda) \leq \mathsf{Adv}^{\mathrm{EUF}}_{\mathcal{A}',\mathsf{DS}}(\lambda)$.

CROB. To show robustness, we rely on the collision-resistance of H. The CROB game in Fig. 3 specifies that the adversary $\mathcal{A}$ against the CROB game finds $\mathsf{pk}_1 \neq \mathsf{pk}_2$ such that $\overline{\mathsf{Ver}}$ passes. The latter implies $\mathsf{H}(\mathsf{pk}_1) = \mathsf{H}(\mathsf{pk}_2)$, trivially breaking the collision-resistance of H, giving us: $\mathsf{Adv}^{\mathrm{CROB}}_{\mathcal{A},\overline{\mathsf{DS}}}(\lambda) \leq \mathsf{Adv}^{\mathrm{CR}}_{\mathcal{A}',\mathsf{H}}(\lambda)$. □

4.2 Achieving Robustness for Functional Encryption

The ABN Transform [1] Adapted to Public-Key FE. As for the case of digital signatures, one can reuse the elegant idea rooted in the *binding* property of a commitment scheme. Concretely, we start from a FE scheme, encrypt the plaintext, and post-process the resulting ciphertext through the use of a public-key encryption scheme. The transform consists in committing to the two public keys (corresponding to FE and PK) and encrypting the resulting decommitment together with the output of $\mathsf{FE.Enc}$ under pk. For decryption, in addition to the functional key, the secret key sk[4] is needed to recover the decommitment from

[4] sk is common to all users querying a sk_f.

the "middle" part of the ciphertext. A key difference to the ABN transform is rooted in the innate nature of FE: we cannot encrypt the plaintext under pk, as this would break indistinguishability. For space reasons, we defer such a construction to the full version of this work.

Simple Robustness Transforms in the Public-Key Setting. A simpler idea makes use of a collision-resistant hash function and simply appends the hash of $\mathsf{mpk}||C$ to the already existing ciphertext.

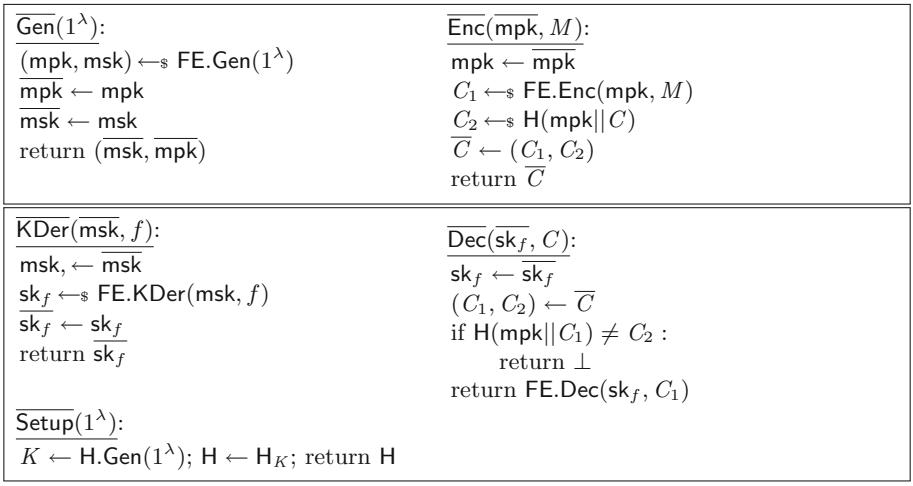

Fig. 8. Generic transform that turns an FE scheme into a FEROB scheme $\overline{\mathsf{FE}}$.

Lemma 4. *Let* FE *be an* IND-FE-CPA*-secure functional encryption scheme in the public setting and let* H *denote a collision-resistant hash function. The functional encryption scheme* $\overline{\mathsf{FE}}$ *obtained through the transform depicted in* Fig. 8 *is* FEROB*-secure, while preserving the* IND-FE-CPA*-security.*

Proof (Lemma 4). Robustness. To show the transform achieves FEROB, we argue that if an adversary concludes with $(\overline{\mathsf{mpk}_1}, R_1, M_1, \overline{\mathsf{mpk}_2}, R_2, M_2, \ldots)$ such that $\overline{\mathsf{FE}}.\mathsf{Enc}(\overline{\mathsf{mpk}_1}, M_1; R_1) = \overline{\mathsf{FE}}.\mathsf{Enc}(\overline{\mathsf{mpk}_2}, M_2; R_2)$, then the adversary is essentially able to find two tuples such that $\mathsf{H}(\mathsf{mpk}_1||\mathsf{FE.Enc}(\mathsf{mpk}_1, M_1; R_1)) = \mathsf{H}(\mathsf{mpk}_2||\mathsf{FE.Enc}(\mathsf{mpk}_2, M_2; R_2))$ which cannot happen with non-negligible probability down to the collision-resistance of H.

Indistinguishability. The proof follows easily down to the indistinguishability of the underlying scheme FE: during the challenge phase, the reduction will be given the C^* corresponding to M_b (chosen by $\mathcal{A}$); after appending $\mathsf{H}(C^*||\mathsf{mpk})$, the adversary will be given $\overline{C^*}$. Also, that the reduction can answer all the functional key-derivation queries the adversary makes. Hence the advantage in winning the IND-FE-CPA game against $\overline{\mathsf{FE}}$ is bounded by the advantage of winning the IND-FE-CPA game against FE.

FEROB Transform in the Private-Key FE Setting. In this part, we provide a similar generic transform for turning any FE scheme into one that is FEROB-secure, in the private-key framework.

$\overline{\mathsf{Gen}}(1^\lambda)$:
$R \leftarrow\!\!\$\ \{0,1\}^\lambda$
$R_1 || R_2 \leftarrow \mathsf{PRG.Eval}(R)$
$\mathsf{msk} \leftarrow \mathsf{FE.Enc}(1^\lambda; R_1)$
$\mathsf{sk} \leftarrow R_2$
$\overline{\mathsf{msk}} \leftarrow (\mathsf{msk}, \mathsf{sk})$
return $\overline{\mathsf{msk}}$

$\overline{\mathsf{Enc}}(\overline{\mathsf{msk}}, M)$:
$(\mathsf{msk}, \mathsf{sk}) \leftarrow \overline{\mathsf{msk}}$
$C_1 \leftarrow\!\!\$\ \mathsf{FE.Enc}(\mathsf{msk}, M)$
$C_2 \leftarrow\!\!\$\ \mathsf{PRF.Eval}(\mathsf{sk}, C_1)$
$\overline{C} \leftarrow (C_1, C_2)$
return $\overline{C}$

$\overline{\mathsf{KDer}}(\overline{\mathsf{msk}}, f)$:
$(\mathsf{msk}, \mathsf{sk}) \leftarrow \overline{\mathsf{msk}}$
$\mathsf{sk}_f \leftarrow\!\!\$\ \mathsf{FE.KDer}(\mathsf{msk}, f)$
$\overline{\mathsf{sk}_f} \leftarrow (\mathsf{sk}_f, \mathsf{sk})$
return $\overline{\mathsf{sk}_f}$

$\overline{\mathsf{Dec}}(\overline{\mathsf{sk}_f}, C)$:
$(\mathsf{sk}_f, \mathsf{sk}) \leftarrow \overline{\mathsf{sk}_f}$
$(C_1, C_2) \leftarrow \overline{C}$
if $\mathsf{PRF.Eval}(\mathsf{sk}, C_1) \neq C_2$:
  return $\perp$
return $\mathsf{FE.Dec}(\mathsf{sk}_f, C_1)$

Fig. 9. A generic transform that turns a FE scheme in the private-key setting into a FEROB-secure scheme $\overline{\mathsf{FE}}$.

Lemma 5. *Let* FE *be an* IND-FE-CPA *functional encryption scheme in the private-key setting. Let* PRG *denote a right-injective length doubling pseudorandom generator from* $\{0,1\}^{|1^\lambda|}$ *to* $\{0,1\}^{2\cdot|1^\lambda|}$ *and* PRF *a collision-resistant* PRF. *The functional encryption scheme* $\overline{\mathsf{FE}}$ *obtained through the transform depicted in* Fig. 9 *is* FEROB*-secure, while preserving* IND-FE-CPA*-security.*

Proof (Lemma 5). ROBUSTNESS. Assuming the FEROB adversary $\mathcal{A}$ outputs $(\overline{\mathsf{msk}_1}, R_1, M_1, f_1, R_{f_1}, \overline{\mathsf{msk}_2}, R_2, M_2, f_2, R_{f_2})$ such that $\overline{\mathsf{FE}}.\mathsf{Enc}(\overline{\mathsf{msk}_1}, M_1; R_1) = \overline{\mathsf{FE}}.\mathsf{Enc}(\overline{\mathsf{msk}_2}, M_2; R_2)$, we argue that:

- $C_2 = \mathsf{PRF.Eval}(\mathsf{sk}_1, C_1) = \mathsf{PRF.Eval}(\mathsf{sk}_2, C_1)$. Down to the collision-resistance (over both keys and inputs) property of the PRF, it results that $\mathsf{sk}_1 = \mathsf{sk}_2$.
- the $\overline{\mathsf{Gen}}$ function makes use of a right injective pseudorandom generator. Since the right half is exactly $\mathsf{sk}_1 (= \mathsf{sk}_2)$, through the injectivity property, it must be the case that the seed R used to feed the PRG is the same.
- since the randomness R is the same for both cases, it results that the random coins used by $\mathsf{FE.Gen}$ are the same, implying that $\mathsf{msk}_1 = \mathsf{msk}_2$.
- finally, we obtain that $\overline{\mathsf{msk}_1} = \overline{\mathsf{msk}_2}$, which is not allowed in the robustness game.

Therefore, the advantage of breaking the FEROB game is bounded by the union bound applied on the collision-resistance of the PRF and right-injectivity of the PRG: $\mathsf{Adv}^{\mathrm{FEROB}}_{\mathcal{A},\overline{\mathsf{FE}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{INJ}}_{\mathcal{A}',\mathsf{PRG}}(\lambda) + \mathsf{Adv}^{\mathrm{CR}}_{\mathcal{A}'',\mathsf{PRF}}(\lambda)$.

IND-FE-CPA-SECURITY. The reduction proceeds via one game hop:

- Game_0: is the game, where the adversary runs against the scheme depicted in Fig. 9—the output of the PRG is the expected one.

– Game_1: based on the pseudorandomness property of the PRG, we change the output to a truly random string, ensuring independence between msk and sk. The distance to Game_0 is bounded by the pseudorandomness advantage against PRG. We now show the advantage of an adversary winning the IND-FE-CPA experiment against $\overline{\mathsf{FE}}$ in this setting is negligible.

Assume the existence of a PPT adversary $\mathcal{A}$ against the IND-FE-CPA of $\overline{\mathsf{FE}}$. We build an adversary $\mathcal{A}'$ against the IND-FE-CPA of the underlying FE scheme. The IND-FE-CPA experiment samples a bit b', the key msk and constructs a key-derivation oracle KDer under msk, which is given to $\mathcal{A}'$. The reduction then proceeds as follows:

1. $\mathcal{A}'$ chooses uniformly at random sk to key the PRF utility.
2. $\mathcal{A}'$ builds the $\overline{\mathsf{FE}}.\mathsf{Enc}$ oracle and the $\overline{\mathsf{FE}}.\mathsf{KDer}$ oracle by querying the given $\mathsf{FE}.\mathsf{Enc}, \mathsf{FE}.\mathsf{KDer}$. The PRF is evaluated under sk.
3. $\mathcal{A}'$ runs $\mathcal{A}$, obtains a tuple (M_0, M_1) and gets back the encryption of $M_{b'}$ (say C^*) by querying $\mathsf{FE}.\mathsf{Enc}(\mathsf{msk}, M_{b'})$. $\mathcal{A}'$ computes the corresponding $\overline{C^*}$, which is passed to $\mathcal{A}$.
4. finally, $\mathcal{A}$ returns a bit b, which constitutes the output of $\mathcal{A}'$.

Analysis of the Reduction. The correctness of the reduction follows trivially. Thus we conclude that in Game_1, the probability of winning is:

$$\Pr[\text{Game}_1^{\mathcal{A}}(\lambda) \Rightarrow 1] \leq \mathsf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A}',\mathsf{FE}}(\lambda).$$

For the analysis, we also include the fact that the transition between Game_0 and Game_1 is bounded as follows:

$$\Pr[\text{Game}_0^{\mathcal{A}}(\lambda) \Rightarrow 1] - \Pr[\text{Game}_1^{\mathcal{A}}(\lambda) \Rightarrow 1] \leq \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{A}'',\mathsf{PRG}}(\lambda).$$

We apply the Union Bound and conclude:

$$\mathsf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A},\overline{\mathsf{FE}}}(\lambda) \leq \mathsf{Adv}^{\text{IND-FE-CPA}}_{\mathcal{A}',\mathsf{FE}}(\lambda) + \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{A}'',\mathsf{PRG}}(\lambda).$$

□

5 Anonymity and Robustness

Interestingly, FEROB does not imply anonymity as defined in Fig. 1 (right) for the public-key case. And based on FEROB ⇒ SROB, it follows that SROB does not imply anonymity in a generic fashion. Therefore, we have the following separation:

Proposition 4. *There exist* FEROB *transforms for public-key functional encryption that do not ensure anonymity (as defined in* Fig. 1*).*

Proof (Proposition 4). We consider the scheme in Fig. 8 and observe that the anonymity game can be easily won as follows: an adversary, given two

master public keys and the ciphertext $\overline{C} \leftarrow (C_1, C_2)$, decides the issuer by checking whether $\mathsf{H}(C_1||\mathsf{mpk}_1) \stackrel{?}{=} C_2$ or $\mathsf{H}(C_1||\mathsf{mpk}_2) \stackrel{?}{=} C_2$, via the publicly available H. □

Finally, we give a generic construction of an anonymous FEROB scheme. Reaching both anonymity and robustness for FE is non-trivial: on one hand, we expect the ciphertext to be "robust" w.r.t. a sole authority (mpk), but the "link" should not be detectable when included in the ciphertext (anonymity). Therefore, we attempt to embed such a link in the functional key. Our solution ensures FEROB through the means of a collision-resistant PRF with keys K generated on the fly. An independent functional key to compute the PRF value is issued via a second FE supporting general circuits, while the PRF key K is encrypted under the additional mpk.

$\underline{\overline{\mathsf{Gen}}(1^\lambda)}$:
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow\!\!\$\ \mathsf{FE.Gen}(1^\lambda)$
$(\mathsf{mpk}', \mathsf{msk}') \leftarrow\!\!\$\ \mathsf{FE}'.\mathsf{Gen}(1^\lambda)$
$\overline{\mathsf{mpk}} \leftarrow (\mathsf{mpk}, \mathsf{mpk}')$
$\overline{\mathsf{msk}} \leftarrow (\mathsf{msk}, \mathsf{msk}')$
return $(\overline{\mathsf{msk}}, \overline{\mathsf{mpk}})$

$\underline{\overline{\mathsf{Enc}}(\overline{\mathsf{mpk}}, M)}$:
$(\mathsf{msk}, \mathsf{msk}') \leftarrow \overline{\mathsf{msk}}$
$(\mathsf{mpk}, \mathsf{mpk}') \leftarrow \overline{\mathsf{mpk}}$
$C_1 \leftarrow\!\!\$\ \mathsf{FE.Enc}(\mathsf{mpk}, M)$
$K \leftarrow\!\!\$\ \mathcal{K}$
$C_2 \leftarrow \mathsf{PRF}(K, \mathsf{mpk})$
$C_3 \leftarrow\!\!\$\ \mathsf{FE}'.\mathsf{Enc}(\mathsf{mpk}', K)$
$\overline{C} \leftarrow (C_1, C_2, C_3)$
return $\overline{C}$

$\underline{\overline{\mathsf{KDer}}(\overline{\mathsf{msk}}, f)}$:
$\mathsf{msk} \leftarrow \overline{\mathsf{msk}}$
$\mathsf{sk}_f \leftarrow\!\!\$\ \mathsf{FE.KDer}(\mathsf{msk}, f)$
$\mathsf{sk}_g \leftarrow\!\!\$\ \mathsf{FE}'.\mathsf{KDer}(\mathsf{msk}', \mathcal{C}_{\mathsf{PRF}(\cdot,\mathsf{mpk})})$
$\overline{\mathsf{sk}_f} \leftarrow (\mathsf{sk}_f, \mathsf{sk}_g)$
return $\overline{\mathsf{sk}_f}$

$\underline{\overline{\mathsf{Dec}}(\overline{\mathsf{sk}_f}, C)}$:
$(\mathsf{sk}_f, \mathsf{sk}_g) \leftarrow \overline{\mathsf{sk}_f}$
$(C_1, C_2, C_3) \leftarrow \overline{C}$
if $\mathsf{FE.Dec}(\mathsf{sk}_g, C_3) \neq C_2$:
  return $\perp$
return $\mathsf{FE.Dec}(\mathsf{sk}_f, C_1)$

Fig. 10. A generic transform that converts an FE scheme into a FEROB scheme $\overline{\mathsf{FE}}$, *without* ensuring anonymity. Here $\mathcal{C}_{\mathsf{PRF}}$ denotes the circuit that computes the PRF value, where mpk is hard-coded in the circuit.

Theorem 1. *Let* FE' *be an* ANON*-secure functional encryption scheme supporting (at least) one functional-key for general circuits and* PRF *denote a collision-resistant* PRF*. Given an* ANON, IND-FE-CPA*-secure scheme* FE*, the functional encryption scheme obtained from the transform in* Fig. 10 *is* FEROB*-secure while preserving the original scheme's security guarantees.*

Proof (Theorem 1). ROBUSTNESS. FEROB follows from the collision resistance of the PRF: if an adversary $\mathcal{A}$ is able to find $(K, C_1), (K', C_1)$ such that $\mathsf{PRF}(K, C_1) = \mathsf{PRF}(K', C_1)$, then $\mathcal{A}$ wins the collision resistance game against the PRF.

INDISTINGUISHABILITY. Follows from the IND-FE-CPA-security of the underlying scheme. For any adversary $\mathcal{A}$ against the IND-FE-CPA-security of the scheme $\overline{\mathsf{FE}}$ in Fig. 10, we build the reduction $\mathcal{A}'$ that wins the IND-FE-CPA game against FE. When $\mathcal{A}$ sends the challenge tuple (M_0, M_1), $\mathcal{A}'$ obtains C_1 from IND-FE-CPA challenger, samples its own Ks, $\mathsf{msk}', \mathsf{mpk}'$ and computes C_2, C_3, which are forwarded to $\mathcal{A}$. Whenever $\mathcal{A}$ makes a functional key query for f, then $\mathcal{A}'$ forwards two functional queries for f and for $\mathcal{C}_{\mathsf{PRF}(\cdot,\mathsf{mpk})}$, a circuit that is designed to compute C_2 (the PRF value) over the encrypted K. Thus, whenever $\mathcal{A}$ returns b, $\mathcal{A}'$ returns the same bit and wins under the same advantage.

ANONYMITY. Follows from the anonymity of the underlying FE scheme. We use a hybrid argument. We start from a setting corresponding to $b = 0$ in the $\mathrm{ANON}^{\mathcal{A}}_{\overline{\mathsf{FE}}}$ game (Game_0).

- Game_1**:** in Game_1, we change C_3 from $\mathsf{FE}'.\mathsf{Enc}(\mathsf{mpk}_0, K)$ to $\mathsf{FE}'.\mathsf{Enc}(\mathsf{mpk}_1, K)$, based on the ANON property of FE', the hop between the two games being bounded by $\mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A},\mathsf{FE}'}(\lambda)$.
- Game_2**:** we change C_1 from $\mathsf{FE}.\mathsf{Enc}(\mathsf{mpk}_0, M)$ to $\mathsf{FE}.\mathsf{Enc}(\mathsf{mpk}_1, M)$, based on the anonymity of the underlying FE scheme, the distance to the previous game being bounded by $\mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A},\mathsf{FE}}(\lambda)$. Implicitly, in Game_2, the reduction updates the value of the PRF from $\mathsf{PRF}(K, \mathsf{FE}.\mathsf{Enc}(\mathsf{mpk}_0, C_1))$ to $\mathsf{PRF}(K, \mathsf{FE}.\mathsf{Enc}(\mathsf{mpk}_1, C_1))$.

Finally observe that Game_2 maps to the setting where $b = 1$ in the anonymity game for the $\overline{\mathsf{FE}}$ scheme. Therefore, $\mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A},\overline{\mathsf{FE}}} \leq \mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A}_1,\mathsf{FE}'}(\lambda) + \mathsf{Adv}^{\mathrm{ANON}}_{\mathcal{A}_2,\mathsf{FE}}(\lambda)$. □

Acknowledgements. The authors thank to anonymous reviewers for valuable comments. Roşie was supported by EU Horizon 2020 research and innovation programme under grant agreements No H2020-ERC-2017-ADG-787390 CLOUDMAP and No H2020-MSCA-ITN-2014-643161 ECRYPT-NET.

References

1. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_28
2. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 733–751. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_33
3. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 333–362. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_12
4. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4

5. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptol. **21**(2), 149–177 (2008)
6. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_16
7. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 501–519. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_29
8. Brakerski, Z., Komargodski, I., Segev, G.: Multi-input functional encryption in the private-key setting: stronger security from weaker assumptions. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 852–880. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_30
9. Chou, T., Orlandi, C.: The simplest protocol for oblivious transfer. In: Lauter, K., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 40–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22174-8_3
10. Cohen, A., Holmgren, J., Nishimaki, R., Vaikuntanathan, V., Wichs, D.: Watermarking cryptographic capabilities. In: Wichs, D., Mansour, Y. (eds.) 48th ACM STOC, pp. 1115–1127. ACM Press, June 2016
11. Damgård, I.B.: Collision free hash functions and public key signature schemes. In: Chaum, D., Price, W.L. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 203–216. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-39118-5_19
12. Farshim, P., Libert, B., Paterson, K.G., Quaglia, E.A.: Robust encryption, revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 352–368. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_22
13. Farshim, P., Orlandi, C., Roşie, R.: Security of symmetric primitives under incorrect usage of keys. IACR Trans. Symm. Cryptol. **2017**(1), 449–473 (2017)
14. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
15. Goldwasser, S., Micali, S., Rivest, R.L.: A "Paradoxical" solution to the signature problem. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, p. 467. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_37
16. Grubbs, P., Lu, J., Ristenpart, T.: Message franking via committing authenticated encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 66–97. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_3
17. Jarecki, S., Krawczyk, H., Xu, J.: OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 456–486. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_15
18. Komargodski, I., Segev, G.: From minicrypt to obfustopia via private-key functional encryption. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 122–151. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_5
19. Mohassel, P.: A closer look at anonymity and robustness in encryption schemes. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 501–518. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_29
20. O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010). http://eprint.iacr.org/2010/556
21. Yao, A.C.-C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd FOCS, pp. 80–91. IEEE Computer Society Press, November 1982