Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model

Benoît Libert[1], Kenneth G. Paterson[2], and Elizabeth A. Quaglia[2]

[1] Université catholique de Louvain, ICTEAM Institute, Belgium
[2] Information Security Group, Royal Holloway, University of London, U.K.
`Benoit.Libert@uclouvain.be, {Kenny.Paterson,E.A.Quaglia}@rhul.ac.uk`

Abstract. In this paper we consider *anonymity* in the context of Broadcast Encryption (BE). This issue has received very little attention so far and *all but one* of the currently available BE schemes fail to provide anonymity. Yet, we argue that it is intrinsically desirable to provide anonymity in standard applications of BE and that it can be achieved at a moderate cost. We provide a security definition for Anonymous Broadcast Encryption (ANOBE) and show that it is achievable assuming only the existence of IND-CCA secure public key encryption (PKE). Focusing on reducing the size of ciphertexts, we then give two generic constructions for ANOBE. The first is from any anonymous (key-private) IND-CCA secure PKE scheme, and the second is from any IBE scheme that satisfies a weak security notion in the multi-TA setting. Furthermore, we show how randomness re-use techniques can be deployed in the ANOBE context to reduce computational and communication costs, and how a new cryptographic primitive – anonymous hint systems – can be used to speed up the decryption process in our ANOBE constructions. Finally, we present a slightly modified version of the Kurosawa-Desmedt (KD) PKE scheme (establishing several results about this scheme that may be of independent interest) and use it to instantiate our first main construction, yielding the currently most efficient ANOBE scheme. All of our results are in the standard model, achieving fully collusion-resistant ANOBE schemes secure against *adaptive* IND-CCA adversaries.

Keywords: Broadcast Encryption, Anonymity

1 Introduction

Anonymity. In a world that is increasingly relying on digital technologies, addressing the issue of protecting users' privacy is of crucial importance. This is reflected by the great attention given to *anonymity* in all the main fields of modern cryptography. In the area of Public-Key Encryption (PKE), anonymity is often referred to as key-privacy [6]. This notion captures the property that an eavesdropper is not able to tell under which one of several public keys a ciphertext was created. The analogous concept in the ID-based setting was studied in [1,13]. The benefit of preserving receivers' privacy is relevant in more elaborate systems involving for example Hierarchical IBE [12], Attribute-Based Encryption (ABE) or Predicate Encryption [31], where achieving anonymity guarantees becomes increasingly challenging. Furthermore, in the context of digital signatures, a number of primitives effectively *rely* on anonymity – group signatures [18], anonymous credentials [17] and e-cash [16] are well-known examples of this.

Broadcast Encryption. Broadcast Encryption (BE) addresses the issue of confidentially broadcasting a message to an arbitrary subset drawn from a universe of users. We will call the universe of n users U and the target, or privileged, set S, where $S \subseteq U$. Since its introduction in 1993 by Fiat and Naor [25], various flavours of BE have been introduced: the scheme can be in a symmetric or asymmetric setting; the set of receivers could be static or dynamic; revocation and traitor-tracing algorithms could be integrated into the system, users' keys might or might not be updated and then forward secrecy may be achieved. We refer to some of the relevant work in the area and the references therein [25,38,22,45,9,20,19,29]. One of the fundamental properties of a BE scheme is *collusion resistance* in the sense that no coalition of users in $U \setminus S$ should be able to recover the message. In the literature we can find several schemes that resist collusion attacks mounted by coalitions of at most $t < n$ users; only some schemes are *fully* collusion-resistant, *i.e.* they can tolerate attacks by coalitions of any size. For the purpose of this paper, we will consider systems that are *public-key*, allow *stateless receivers* (users that are not required to update their private keys) and are *fully* collusion-resistant. These are by now standard objectives for a BE scheme in the public-key setting.

Several additional practical aspects need to be taken into consideration, especially in view of the real-life applications of BE: strength of security notions, public and private storage requirements, ciphertext length, and computational costs. The specific nature of the primitive has led researchers to focus in particular on solutions having ciphertexts that are as short as possible. In this respect, the results of [9] and [29] are nearly optimal. However, designing BE schemes for real-life applications to broadcasting should not only involve efficiency and confidentiality issues. In particular, the privacy of users should be protected as much as possible. We believe that, to date, this aspect has not been adequately dealt with. Our study of the literature reveals that anonymity in BE has only been considered in a single paper [5], in the context of encrypted file systems[3]. Surprisingly, almost all subsequent work on BE has ignored the issue of anonymity. Moreover, as we shall explain below, state-of-the-art BE schemes are inherently incapable of providing any kind of anonymity.

Anonymity in Broadcast Encryption. According to commonly accepted definitions [29,10,19], a BE scheme consists of four algorithms: `Setup`, `KeyGen`, `Enc` and `Dec`. Each user in the system can obtain his private key from the `KeyGen` algorithm, and the sender can choose an *arbitrary* target set of users S to which he wishes to broadcast a message. To decrypt, a legitimate user, i.e. a user in S, has to run the decryption algorithm on input the ciphertext, his private key *and* a description of the target set S. This set S is required specifically as an input to `Dec` in the existing definitions of BE. Hence the user needs to somehow know to which set S the message was broadcast, otherwise he cannot decrypt. Unfortunately, solving this problem is not just a matter of removing this requirement from the model, as current schemes explicitly *rely* on S as an input to `Dec` for decryption to work. Thus these schemes cannot provide any anonymity.

This limitation in the existing BE model and schemes clearly causes serious privacy issues: imagine we deploy a BE scheme, as defined above, for television broadcasting. Suppose the privileged set is the set of all users who have paid a subscription to a certain channel. Each customer should have access to that channel using his private key. The problem is that, to decrypt, he will have to know who *else* has paid for the specific subscription! Not only is this requirement very inconvenient for the practical deployment of BE schemes, it is also a severe violation of the individual subscriber's privacy. Ideally, a BE scheme should protect users' privacy by guaranteeing that ciphertexts do not leak any information about the privileged set S.

Current BE schemes such as those in [29,10,19] do not account for the cost of broadcasting a description of S when calculating the size of ciphertexts. In the most general usage scenario intended for BE, where S is dynamic and may be unpredictable from message to message, the ciphertexts in such schemes must effectively include a description of S as part of the ciphertexts themselves.

[3] We observe that [30] addresses the issue of hiding the identity of the *sender* in a broadcast protocol, which is *not* what we intend by anonymous broadcast encryption.

This means that the true ciphertext size in these schemes is linear in n rather than constant-size, as a cursory examination of the schemes might suggest[4]. However, achieving linear-sized ciphertext is already an impressive achievement, since there is a simple counting argument showing that, for a universe of n users in which every possible subset S should be reachable by secure broadcast, ciphertexts must contain at least n bits.

Further Details on Related Work. As mentioned above, the only prior work addressing the issue of anonymity in BE appears to be that of Barth *et al.* [5] (there, it is called *privacy*). In [5], several BE systems used in practice were examined with respect to anonymity. In addition, a generic construction for a BE scheme using a key-private, IND-CCA secure PKE scheme was given, with the scheme achieving anonymity and IND-CCA security against static adversaries. The construction encrypts the message for each intended receiver using the PKE scheme, and then ties together the resulting ciphertexts using a strongly secure one-time signature. Barth *et al.* [5] also provided a technique which can be used to speed-up decryption, but this technique was only analysed in the Random Oracle Model. In [11] the authors provide a private linear broadcast encryption (PLBE) scheme to realise a fully collusion-resistant traitor-tracing scheme. A PLBE, however, is a BE system with limited capabilities (i.e. it cannot address arbitrary sets of users) and hence this work does not provide a solution to the problem considered so far.

In a very recent work [24] that builds on [5] and this paper, the authors have given constructions for anonymous broadcast encryption schemes with compact ciphertexts, but using a much weaker notion of anonymity that does not seem to relate very closely to real-world requirements.

There is much work, both cryptographic and non-cryptographic, on pseudonymous systems. In principle, pseudonyms could be used to enhance the anonymity of BE schemes: now users would not be identifiable directly, since a certificate would link a public key to a pseudonym rather than a real name. However, ciphertexts would still be linkable, in the sense that it would be possible to detect if two ciphertexts were intended for the same set of recipients or not. The approach we take here offers much stronger levels of privacy, removing ciphertext linkability in particular.

Our Contributions. Despite its importance, anonymous broadcast encryption has not received much attention since the initial work of Barth *et al.* [5]. This paper aims to raise the profile of this neglected primitive.

We start by giving a unified security definition for Anonymous Broadcast Encryption (ANOBE). Instead of separating anonymity and confidentiality as in [5], we use a combined security notion for ANOBE which helps to streamline our presentation and proofs. In addition, we strengthen the model to allow the adversary to make *adaptive* corruptions, *with all of our constructions achieving security in this setting.* In contrast, the definition of [5] is static, requiring the adversary to choose whom to corrupt before seeing the public keys in the system. As a first step we show that our enhanced security definition is indeed satisfiable: adaptively secure ANOBE can be built based only on the existence of IND-CCA secure PKE (without requiring the base PKE scheme to have any anonymity properties itself). This construction results in a very efficient (constant) decryption procedure but has ciphertexts whose size is linear in n, the number of users in the universe U.

Our second contribution is to show that the generic construction for ANOBE suggested by Barth *et al.* [5] actually possesses adaptive security, and not merely static security as was established

[4] This does not rule the use of compact encodings of S being transmitted with ciphertexts in more restrictive usage scenarios, for example, only sending the difference in S when the set S changes only slowly from message to message.

in [5]. This construction starts from any weakly robust (in the sense of [2]), key-private PKE scheme with chosen-ciphertext security. In comparison with our first generic construction, this result imposes stronger requirements on the underlying encryption scheme. However, it achieves shorter ciphertexts, with the size being linear in the size of the target set S. We also provide a variant of this construction that replaces the IND-CCA secure PKE component with an identity-based encryption (IBE) scheme having suitable security properties. This alternative further increases the set of components that can be used to obtain ANOBE.

One major drawback of the latter constructions is that decryption takes linear time in the size of the set S. Our third result is a technique allowing for constant decryption cost and which we prove secure in the standard model (*i.e.*, without random oracles) using our enhanced security definition. So far, the only known technique – put forth by Barth *et al.* [5] – enabling constant-time decryption requires the random oracle heuristic in the security analysis. To eliminate the random oracle, we introduce a new primitive, which we call an *anonymous hint system.* In essence, this primitive provides a way for an encrypter to securely tell receivers which ciphertext component is intended for them, allowing them to ignore all but one ciphertext component and so decrypt more efficiently. The hint primitive, for which we provide an implementation based on the Decision-Diffie-Hellman (DDH) assumption, is defined and realized in such a way that its integration with our generic ANOBE constructions maintains compatibility with our proofs of adaptive security.

Our fourth contribution is to show how randomness re-use techniques originally developed for PKE in [34,8,7] can be modified for secure deployment in the ANOBE setting. In particular, we identify a slightly stronger notion of reproducibility that we call *key-less reproducibility.* We show that if our base PKE scheme has this property (in addition to the other properties needed in our generic construction) then it can be used with the same randomness across all ciphertext components in our main ANOBE construction. This not only allows the size of ciphertexts to be reduced further (by eliminating repeated ciphertext elements) but also reduces the sender's computational overhead.

Our final contribution is to establish that the Kurosawa-Desmedt (KD) [36] hybrid encryption scheme can be tweaked to have all the properties that are needed of the base PKE scheme in our constructions. The KD scheme is an ideal starting point since it is one of most efficient PKE schemes with IND-CCA security in the standard model. In results that may be of independent interest, we present KD*, a modified version of the KD scheme, that is *strongly robust* (although weak robustness suffices for our purposes), assuming that its symmetric components satisfy some relatively mild conditions; *anonymous* under the DDH assumption (and, again, under mild assumptions on its symmetric components) and *key-less reproducible.*

Tying everything together and using KD* as the base scheme, we obtain the *currently most efficient instantiation* of an ANOBE scheme, for which ciphertexts contain only 2 group elements and $|S|$ symmetric ciphertexts (plus a signature and a verification key). Decryption can be achieved in constant time by combining this scheme with our DDH-based hint system, with an additional $2|S| + 1$ group elements in the ciphertext.

As can be seen from the details of our constructions, achieving anonymity does not add *any* cost to the encryption process compared to non-anonymous schemes (for example, [9,29]): in our ANOBE schemes, encryption requires a number of group operations that is linear in $|S|$. As for decryption, our speed-up technique allows the legitimate user to recover the message in constant time. Our ciphertext size is linear in $|S|$ (and thus linear in n and of the same order of magnitude as the *true* ciphertext size in existing BE schemes). Thus one interpretation of our results is that anonymity does not "cost" anything in an asymptotic sense. Naturally, the constants matter in practice, and reducing the constant in the ciphertext size for ANOBE to something closer to what can be achieved in the non-anonymous setting is a major open problem. However, we reiterate that reducing the *true* size of ciphertexts below linear in n in either the anonymous or non-anonymous setting is impossible.

2 Anonymous Broadcast Encryption

We define a model of public-key Broadcast Encryption, where algorithms are specified to allow for anonymity (similarly to [5]) and they are general enough to include the identity-based variant of BE introduced in [19].

Definition 1. *Let $U = \{1, ..., n\}$ be the universe of users. A* broadcast encryption (BE) scheme *is defined by four algorithms and has associated message space $\mathcal{MSP}$ and ciphertext space $\mathcal{CSP}$.*

$\texttt{BE.Setup}(\lambda, n)$**:** *This algorithm takes as input the security parameter λ and the number of users in the system n. It outputs a master public key BE-MPK and a master secret key BE-MSK.*

$\texttt{BE.Key-Gen}(\text{BE-MPK}, \text{BE-MSK}, i)$**:** *This algorithm takes as input BE-MPK, BE-MSK and an index $i \in U$ and outputs the private key sk_i for user i.*

$\texttt{BE.Enc}(\text{BE-MPK}, m, S)$**:** *This algorithm takes as input BE-MPK, a message $m \in \mathcal{MSP}$ and a subset $S \subseteq U$, the broadcast target set. It outputs a ciphertext $c \in \mathcal{CSP}$.*

$\texttt{BE.Dec}(\text{BE-MPK}, sk_i, c)$**:** *This algorithm takes as input BE-MPK, a private key sk_i and a ciphertext $c \in \mathcal{CSP}$. It outputs either a message $m \in \mathcal{MSP}$ or a failure symbol $\perp$.*

The correctness property *is that for all $S \subseteq U$ and all $i \in U$ if $c =$ $\texttt{BE.Enc}(BE\text{-}MPK, m, S)$ and sk_i is the private key for $i \in S$ then $\texttt{BE.Dec}(BE\text{-}MPK, sk_i, c) = m$ with overwhelming probability.*

We observe that this definition no longer requires the set S as an input to the decryption algorithm. This is crucial in developing the notion of anonymous broadcast encryption (ANOBE), for which we next provide an appropriate security model for the case of *adaptive* adversaries.

Definition 2. *We define the ANO-IND-CCA security game for BE as follows.*
Setup. *The challenger $\mathcal{C}$ runs $\texttt{BE.Setup}(\lambda, n)$ to generate master public key BE-MPK and master secret key BE-MSK and gives BE-MPK to the adversary $\mathcal{A}$.*
Phase 1. *$\mathcal{A}$ can issue queries to a private key extraction oracle for any index $i \in U$. The oracle will respond by returning $sk_i = \texttt{BE.Key-Gen}(BE\text{-}MPK, BE\text{-}MSK, i)$. $\mathcal{A}$ can also issue decryption queries of the form (c, i), where $i \in U$, and the oracle will return the decryption $\texttt{BE.Dec}(BE\text{-}MPK, sk_i, c)$.*
Challenge. *$\mathcal{A}$ selects two equal-length messages m_0, $m_1 \in \mathcal{MSP}$ and two distinct sets $S_0, S_1 \subseteq U$ of users. We require that S_0 and S_1 be of equal size and also impose the restriction that $\mathcal{A}$ has not issued key queries for any $i \in S_0 \triangle S_1 = (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$. Further, if there exists an $i \in S_0 \cap S_1$ for which $\mathcal{A}$ has queried the key, then we require that $m_0 = m_1$. The adversary $\mathcal{A}$ passes m_0, m_1 and S_0, S_1 to $\mathcal{C}$. The latter picks a random bit $b \in \{0, 1\}$ and computes $c^* = \texttt{BE.Enc}(BE\text{-}MPK, m_b, S_b)$ which is returned to $\mathcal{A}$.*
Phase 2. *$\mathcal{A}$ continues to make queries to the private key extraction oracle with the restrictions that $i \notin S_0 \triangle S_1$ and that, if $i \in S_0 \cap S_1$, then $m_0 = m_1$. $\mathcal{A}$ may continue issuing decryption queries (c, i) with the restriction that if $c = c^*$ then either $i \notin S_0 \triangle S_1$ or $i \in S_0 \cap S_1$ and $m_0 = m_1$.*
Guess. *The adversary outputs its guess b' for b.*

Definition 3. *We say that a BE scheme is* anonymous and semantically secure against chosen-ciphertext attacks *(ANO-IND-CCA) if all polynomial-time adaptive adversaries $\mathcal{A}$ have at most negligible advantage in the above game, where $\mathcal{A}$'s advantage is defined as $Adv_{\mathcal{A},BE}^{ANO\text{-}IND\text{-}CCA}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|$.*

Like the definition of [5], Definition 2 does not require the ANOBE ciphertext to hide the number of receivers. However, specific schemes (such as the one in Section 3.1) can also conceal the cardinality of S.

We will next show that this notion is indeed *feasible* by presenting a generic construction that relies solely on the existence of IND-CCA secure PKE schemes. We will then improve its performance by giving alternative generic constructions whose underlying primitives require additional security properties.

3 Generic Constructions for ANOBE from PKE

3.1 ANOBE from Minimal Assumptions

Since our aim is to provide a formal treatment of anonymous broadcast encryption, we begin by showing that ANOBE *can be achieved*. Indeed, by simply assuming the existence of an IND-CCA secure PKE scheme we can construct an ANOBE scheme as follows.

Let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme with message space $\mathcal{M} = \{0,1\}^m$. Here, algorithm Gen takes as input a security parameter and outputs public parameters par, used by KeyGen to generate a key pair (pk, sk). Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme consisting of a key generation algorithm $\mathcal{G}$, a signing algorithm $\mathcal{S}$ and a verification algorithm $\mathcal{V}$. We assume that the key space of Σ is $\mathcal{K} = \{0,1\}^v$, for some $v \in \mathsf{poly}(\lambda)$. We use π^{pke} and Σ to generically instantiate a BE scheme, with message space $\{0,1\}^{m-v}$. In the description hereafter, we include the symbol ε as a valid but distinguished message in $\{0,1\}^{m-v}$: in other words, all the messages that receivers accept as legal plaintexts are different from ε.

`BE.Setup`(λ, n)**:** Generate $par \leftarrow \mathsf{Gen}(\lambda)$ and, for $i = 1$ to n, generate $(sk_i, pk_i) \leftarrow \mathsf{Keygen}(par)$. The master private key is BE-MSK $= \{sk_i\}_{i=1}^n$ and the master public key consists of

$$\text{BE-MPK} = \Big(par,\ \Sigma,\ \{pk_i\}_{i=1}^n\Big).$$

`BE.Key-Gen`(BE-MPK, BE-MSK, i)**:** parse the master secret key BE-MSK as $\{sk_i\}_{i=1}^n$ and output sk_i.

`BE.Enc`(BE-MPK, M, S)**:** to encrypt a message M for a receiver set $S \subseteq \{1, \ldots, n\}$, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Then, for all indices $j \in \{1, \ldots, n\}$, compute $C_j = \mathsf{Encrypt}(par, pk_j, M||\mathsf{VK})$ if $j \in S$ and $C_j = \mathsf{Encrypt}(par, pk_j, \varepsilon||\mathsf{VK})$ if $j \notin S$. The ANOBE ciphertext consists of $C = \big(C_1, \ldots, C_n, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, (C_1, \ldots, C_n)\big)$.

`BE.Dec`(BE-MPK, sk_i, C)**:** given $C = \big(C_1, \ldots, C_n, \sigma\big)$, compute $M' = \mathsf{Decrypt}(sk_i, C_i)$. If $M' \neq \perp$, parse M' as $M' = M||\mathsf{VK}$ for some bitstrings $M \in \{0,1\}^{m-v}$ and $\mathsf{VK} \in \{0,1\}^v$. Then, if $\mathcal{V}\big(\mathsf{VK}, (C_1, \ldots, C_n), \sigma\big) = 1$ and $M \neq \varepsilon$ return M. Otherwise, output $\perp$.

The correctness of the BE scheme follows directly from the correctness of π^{pke} and Σ. This construction is reminiscent of generic constructions of chosen-ciphertext-secure multiple encryption [23] and it is easily seen to yield a secure ANOBE. A proof of the following theorem is available in Appendix A.

Theorem 1. *Let π^{pke} be an IND-CCA secure PKE scheme and let Σ be a strongly unforgeable one-time signature scheme. The BE scheme constructed above is ANO-IND-CCA secure against adaptive adversaries.*

We have described an ANOBE scheme from minimal assumptions. We note that encryption time is linear in n but decryption is performed in *constant* time, since a user simply selects the ciphertext component to decrypt according to its index. However, the ciphertext size is *linear* in n, as we encrypt to each user in the universe. It is desirable to improve on this and achieve a realization of ANOBE with more compact ciphertexts.

We will next see how to modify this first generic construction, obtaining an ANOBE scheme whose ciphertext size is linear in the size of the *target set* S.

3.2 ANOBE with Adaptive Security from Robust, Anonymous PKE

A simple solution to the broadcast problem is to encrypt the message under the public key of each user in the privileged set. This naive approach, so often discarded in most BE literature due to efficiency reasons, turns out to provide another generic construction for ANOBE, which differs from the previous one as now we deploy a public-key encryption scheme only to encrypt the *message* to the users *in the target set*.

For this approach, the underlying PKE scheme has to be key-private (or IK secure [6]), in that the ciphertext does not leak under which public key it was created. We also require the PKE scheme to be weakly robust, in the sense of [2], not only for correctness but also for consistency in the CCA security proof simulation. This property can be generically achieved [2] for any PKE scheme, by appending some publicly-known redundancy to the message and checking it upon decryption.

This is essentially the construction that was already suggested by Barth, Boneh and Waters [5]. We now prove that it is actually *adaptively* secure, rather than just statically secure, as was established in [5].

Let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme and $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme. We construct an ANOBE scheme, $\text{ANOBE}^{\pi^{\text{pke}},\Sigma}$, as follows.

$\texttt{BE.Setup}(\lambda, n)$**:** Run $\mathsf{Gen}(\lambda, n)$ to obtain public parameters par. For $i = 1$ to n, run $\mathsf{Keygen}(par)$ to generate (sk_i, pk_i). The master private key is $\text{BE-MSK} = \{sk_i\}_{i=1}^n$ and the master public key is

$$\text{BE-MPK} = \Big(par,\ \Sigma,\ \{pk_i\}_{i=1}^n\Big).$$

$\texttt{BE.Key-Gen}(\text{BE-MPK}, \text{BE-MSK}, i)$**:** parse BE-MSK as $\{sk_i\}_{i=1}^n$ and output sk_i.

$\texttt{BE.Enc}(\text{BE-MPK}, M, S)$**:** to encrypt M for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Then, for each $j = 1$ to ℓ, compute

$$C_j = \mathsf{Encrypt}(par, pk_{i_j}, M||\mathsf{VK}).$$

The ANOBE ciphertext is $C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

$\texttt{BE.Dec}(\text{BE-MPK}, sk_i, C)$**:** parse C as a tuple $\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big)$. If $\mathcal{V}\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big) = 0$, return $\perp$. Otherwise, repeat the following steps for $j = 1$ to ℓ.

1. Compute $M' = \mathsf{Decrypt}(sk_i, C_j)$. If $M' \neq \perp$ and can moreover be parsed as $M' = M||\mathsf{VK}$ for some M of appropriate length, return M.

2. If $j = \ell$ output $\perp$.

The correctness of $\mathrm{ANOBE}^{\pi^{\mathrm{pke}},\Sigma}$ follows directly from the correctness and weak robustness of π^{pke}.

Theorem 2. *$ANOBE^{\pi^{\mathrm{pke}},\Sigma}$ is adaptively ANO-IND-CCA secure assuming that: (i) π^{pke} is key-private and IND-CCA (AI-CCA) secure and weakly robust under chosen-ciphertext attacks (as defined in Appendix F); (ii) Σ is a strongly unforgeable one-time signature scheme.*

In our proof (given in Appendix B) we make use of a sequence of hybrid arguments where ciphertext components are gradually modified at each step and each hybrid argument requires the reduction to guess upfront the identity of an uncorrupted user. We note that Gentry and Waters [29] already briefly mentioned that such an approach could potentially be useful to prove adaptive security but, to the best of our knowledge, no rigorous analysis of this type was previously given in the literature. Moreover, in the constructions that follow, achieving adaptive security represents even more of a challenge since it is a non-trivial task to get this proof technique to suitably interact with the methods we present for speeding up encryption and decryption procedures.

In terms of efficiency, from this construction we will obtain secure ANOBE schemes with typically very small (constant) private key storage requirements and ciphertexts which are $|S|$ times the size of the ciphertext of the underlying PKE scheme. Encryption and decryption have both cost linear in the size of S. If, for example (as suggested in [5]), we use the Cramer-Shoup PKE scheme to instantiate the ANOBE scheme, the private keys will have constant size (namely 5 elements in $\mathbb{Z}_p$), and the resulting ciphertext will consist of roughly $4 \cdot |S|$ group elements. The scheme will be secure in the standard model under the DDH assumption.

If we look at recent efficient instantiations of BE, for example that of Gentry-Waters [29], we have private keys whose size is linear in the number of users, and ciphertexts which consist of n bits plus 3 group elements (if we include the cost of transmitting a description of S as part of the ciphertext). It is clear that in general the solution of [29] is more efficient in terms of ciphertext size. The key point though is that it is not anonymous.

4 Generic Construction for ANOBE from IBE

An IBE scheme I typically consists of four algorithms $(\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$, where Setup and KeyExt are run by a trusted authority (TA). In our construction we will make use of a multi-TA IBE scheme $I' = (\mathsf{CommonSetup}, \mathsf{TASetup}, \mathsf{KeyDer}, \mathsf{Enc}', \mathsf{Dec}')$ as formalised in [40]. We recall from [40] that $\mathsf{CommonSetup}$, on input the security parameter, outputs the system's parameters par and a set of labels of the TAs in the system, and that $\mathsf{TASetup}$, on input par, outputs a master public key mpk and a master secret key msk. This algorithm is randomized and executed independently for each TA in the system. The remaining algorithms are as per a normal IBE scheme. For this primitive we consider the notion of TA anonymity, as defined in [40], which formally models the inability of the adversary to distinguish two ciphertexts corresponding to the same message and identity, but created using different TA master public keys. An example of a TA-anonymous IBE scheme is the multi-TA version of Gentry's IBE scheme [28] developed in [41].

Now, let $I' = (\mathsf{CommonSetup}, \mathsf{TASetup}, \mathsf{KeyDer}, \mathsf{Enc}', \mathsf{Dec}')$ be a weakly robust, in the sense of definition 10 (in Appendix F.2), multi-TA IBE scheme and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme. We will use I' and Σ to generically instantiate a BE scheme in the following way.

$\texttt{BE.Setup}(\lambda, n)$**:** Run $\mathsf{CommonSetup}$ on input of $\lambda \in \mathbb{N}$ to obtain the system's parameters par. Run $\mathsf{TASetup}(par)$ n times to obtain n key-pairs $\{mpk_i, msk_i\}_{i \in U}$. Return the par, Σ and n public keys $\{mpk_i\}_{i \in U}$.

$\texttt{BE.Key-Gen}(par, \lambda, i)$**:** Return msk_i, the secret key corresponding to the public key mpk_i of user i.

$\texttt{BE.Enc}(par, M, S)$**:** Run $\mathcal{G}$ to obtain a verification key VK and the corresponding signing key SK. For each $i \in S$ run $\mathsf{Enc}'(mpk_i, M, \mathsf{VK})$ to obtain ciphertext C_i. The ANOBE ciphertext is

$$C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big),$$

where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

$\texttt{BE.Dec}(par, msk_i, C)$**:** Parse the ciphertext C as $\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big)$. If $\mathcal{V}\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big) = 0$, return $\perp$. Otherwise, compute $sk_{i_{\mathsf{VK}}} = \mathsf{KeyDer}(mpk_i, msk_i, \mathsf{VK})$ and repeat the following steps for $j = 1$ to ℓ.

1. Compute $M = \mathsf{Dec}'(mpk_i, sk_{i_{\mathsf{VK}}}, C_j)$. If $M \neq \perp$, return M.
2. If $j = \ell$ output $\perp$.

The correctness of the BE scheme follows directly from the correctness and a certain weak robustness property (formalized by Definition 10 in Appendix F.2) of the IBE scheme I' used to construct it.

If instantiated with the multi-TA version of Gentry's IBE scheme [28,41] (which can be made weakly robust simply by applying the transform in [2]), this construction yields very short constant size private keys (just one element in $\mathbb{Z}_p^*$) and ciphertexts consisting of roughly $3 \cdot |S|$ group elements ($|S|$ in $\mathbb{G}$ and $2 \cdot |S|$ in $\mathbb{G}_T$) plus a signature and a verification key. Encryption and decryption have both cost linear in the size of S.

Theorem 3. *Let I' be a TA-anonymous, sID-IND-CPA secure IBE scheme and let Σ be a strongly unforgeable one-time signature. Then, the above BE scheme is adaptively ANO-IND-CCA secure.*

We give some intuition for the proof. We observe that, in [41], the authors apply a modified version of the Canetti-Halevi-Katz (CHK) transform [14] using the same primitives as our generic construction to obtain a key-private IND-CCA PKE scheme. We introduce further modifications to build a BE scheme achieving ANO-IND-CCA security. The idea is that, within this transform, we encrypt m for the *same* identity VK under the $|S|$ different public keys. We then sign all ciphertexts and append the verification key VK (note that this signature binds all these ciphertexts together). Upon decryption, a user verifies the signature against VK and, if valid, proceeds to derive the decryption key for identity VK by running the IBE key-extraction algorithm on input his private key. By similar arguments to those in [14] and [41], and by applying techniques analogous to those proving adaptive security in Theorem 2, we can show that adaptive ANO-IND-CCA security is achieved.

5 Efficient Decryption in the Standard Model

The generic constructions for ANOBE presented in Section 3.2 and 4 both suffer from linear time decryption. This arises from the fact that users do not know which ciphertext component is intended for them, and hence will have to perform an average of $|S|/2$ decryptions before recovering the message. Clearly this procedure is quite cumbersome. We now present a technique which achieves *constant* time decryption in the standard model. We make use of a new primitive, called tag-based *anonymous hint systems*, for which we provide a definition, the relevant security models and a concrete instantiation.

5.1 Tag-Based Anonymous Hint Systems

A tag-based anonymous *hint* system can be seen as a tag-based encryption scheme [33] allowing to generate weak forms of encryption under a tag t and a public key pk. The result of the process consists of a *value* U and a *hint* H. The pair (U, H) should be pseudo-random (in particular, hints generated under two distinct public keys should be indistinguishable) when only the public key pk is available. Also, the private key sk makes it possible to check whether a given hint H is valid w.r.t. a tag t. A value-hint pair can be seen as an extractable commitment to a public key. Formally, such a system is defined in terms of the following algorithms.

$\mathtt{Keygen}(\mathtt{cp})$: takes as input a set of common public parameters $\mathtt{cp}$ and outputs a key pair (sk, pk). We assume that $\mathtt{cp}$ specifies a randomness space $\mathcal{R}^h$ and a space $\mathcal{T}^h$ of acceptable tags for the scheme.

$\mathtt{Hint}(\mathtt{cp}, t, pk, r)$**:** is a deterministic algorithm taking as input common public parameters $\mathtt{cp}$, a public key pk, a tag t and random coins $r \in_R \mathcal{R}^h$. It outputs pair (U, H) consisting of a value U and a hint H. It is required that U only depends on the random coins r and not on pk.

$\mathtt{Invert}(\mathtt{cp}, sk, t, U)$**:** is a deterministic "inversion" algorithm taking as input a value U, a tag t and a private key sk. It outputs either a hint H or $\perp$ if U is not in the appropriate domain.

Correctness requires that, for any pair $(sk, pk) \leftarrow \mathtt{Keygen}(\lambda)$ and any possible random coins r, if $(U, H) \leftarrow \mathtt{Hint}(t, pk, r)$, then $\mathtt{Invert}(\mathtt{cp}, sk, t, U) = H$.

Although hint systems bear similarities with tag-KEMs, as formalized by Abe *et al.* [3], the two primitives are different and incomparable. In the tag-KEM syntax, the symmetric "session key" is chosen first and it does not depend on the tag. In hint schemes, the syntax requires to choose a pair (U, H), where U does not depend on pk but the session key H can depend on both pk and the tag (this is what happens in the construction we give). The security definitions are also different since, in Definition 4 hereafter, there is no inversion oracle (that would return H given U and t) but only a verification oracle that determines if (U, H, t) form a valid triple with respect to public keys pk_0 and pk_1.

In certain aspects, hint schemes are reminiscent of extractable hash proof systems [44] but there are several differences. In [44], in addition to the value that we call U, the random coins allowing to compute U are used to compute a witness S such that (U, S) satisfies some relation. From U, the element S is also computable using the private key and the value that we call H (which is termed "hash value" in [44]). At the same time, S should be infeasible to compute without the private key or the random coins used to sample U. Hint schemes are different in that they rather require the hardness of computing H from U without the private key. In addition, tag-based hints require that it be hard to decide if a pair (U, H) is valid for a certain tag $t^\star$ (*i.e.*, to decide if $H = \mathtt{Invert}(\mathtt{cp}, sk, t^\star, U)$) even with access to a decision oracle for tags $t \neq t^\star$.

Definition 4. *A tag-based hint system* $(\mathtt{Keygen}, \mathtt{Hint}, \mathtt{Invert})$ *is* anonymous *if no PPT adversary has non-negligible advantage in the following game:*

1. *On input of common public parameters* $\mathtt{cp}$*, the adversary* $\mathcal{A}$ *chooses a tag* $t^\star$ *and sends it to the challenger.*
2. *The challenger generates pairs* $(sk_0, pk_0) \leftarrow \mathtt{Keygen}(\lambda)$*,* $(sk_1, pk_1) \leftarrow \mathtt{Keygen}(\lambda)$ *and gives* pk_0, pk_1 *to* $\mathcal{A}$*.*

3. *On polynomially-many occasions, $\mathcal{A}$ adaptively invokes a verification oracle on value-hint-tag triples (U, H, t) such that $t \neq t^\star$. The challenger replies by returning bits $(d_0, d_1) \in \{0,1\}^2$ where $d_0 = 1$ if and only if $H = \texttt{Invert}(\texttt{cp}, sk_0, t, U)$ and $d_1 = 1$ if and only if $H = \texttt{Invert}(\texttt{cp}, sk_1, t, U)$.*
4. *When $\mathcal{A}$ decides to enter the challenge phase, the challenger flips a binary coin $b \overset{\$}{\leftarrow} \{0,1\}$ and chooses other random coins $r^\star \overset{\$}{\leftarrow} \mathcal{R}^h$. It outputs $(U^\star, H^\star) = \texttt{Hint}(\texttt{cp}, t^\star, pk_b, r^\star)$.*
5. *$\mathcal{A}$ makes further queries but is not allowed to make queries involving the target tag $t^\star$.*
6. *$\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and wins if $b' = b$.*

As usual, $\mathcal{A}$'s advantage is measured by the distance $\mathbf{Adv}^{\text{anon-hint}}(\mathcal{A}) = |\Pr[b' = b] - 1/2|$.

Definition 5. *A tag-based hint system* $(\texttt{Keygen}, \texttt{Hint}, \texttt{Invert})$ *is strongly* robust *if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the following game, where $\mathcal{A}$'s advantage is its probability of success.*

1. *The challenger chooses public parameters* $\texttt{cp}$ *and generates pairs $(sk_0, pk_0) \leftarrow \texttt{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \texttt{Keygen}(\lambda)$. It gives* $\texttt{cp}$ *and pk_0, pk_1 to $\mathcal{A}$.*
2. *$\mathcal{A}$ invokes a verification oracle on arbitrary value-hint-tag triples (U, H, t). The challenger replies by returning bits $(d_0, d_1) \in \{0,1\}^2$ where $d_0 = 1$ if and only if $H = \texttt{Invert}(\texttt{cp}, sk_0, t, U)$ and $d_1 = 1$ if and only if $H = \texttt{Invert}(\texttt{cp}, sk_1, t, U)$.*
3. *$\mathcal{A}$ outputs a triple $(U^\star, H^\star, t^\star)$ and wins if the latter satisfies $H^\star = \texttt{Invert}(\texttt{cp}, sk_0, t^\star, U^\star) = 1$ and $H^\star = \texttt{Invert}(\texttt{cp}, sk_1, t^\star, U^\star) = 1$.*

Analogously to the PKE case [2], weak robustness for tag-based hint systems is defined by letting the adversary simply make a challenge request in step 3. The challenger then chooses a tag $t^\star$ as well as random coins $r^\star$, generates a value-hint pair $(U^\star, H^\star) = \texttt{Hint}(\texttt{cp}, t^\star, pk_0, r^\star)$ and $\mathcal{A}$ wins if $H^\star = \texttt{Invert}(\texttt{cp}, sk_1, t^\star, U^\star) = 1$. Weak robustness will be sufficient for our purposes but the scheme hereafter is also strongly robust under the discrete logarithm assumption in $\mathbb{G}$.

To show that this newly defined primitive is indeed feasible, we give an example of an anonymous hint system based on the DDH assumption and the CCA-secure public key encryption scheme described in [15].

Let the common public parameters $\texttt{cp} = \{\mathbb{G}, p, g\}$ consist of a group $\mathbb{G}$ of prime order $p > 2^\lambda$ with a generator $g \in_R \mathbb{G}$. We assume that tags are elements of $\mathcal{T}^h = \mathbb{Z}_p^*$ and that the randomness space is $\mathcal{R}^h = \mathbb{Z}_p^*$.

$\texttt{Keygen}(\texttt{cp})$**:** chooses random $x_1, x_2, y_1, y_2 \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $X_i = g^{x_i}$ and $Y_i = g^{y_i}$ for each $i \in \{1,2\}$. The public key is $pk = \big(X_1, X_2, Y_1, Y_2\big)$ and the private key is $sk = (x_1, x_2, y_1, y_2)$.

$\texttt{Hint}(\texttt{cp}, t, pk, r)$**:** given $pk = (\mathbb{G}, p, g, X_1, X_2, Y_1, Y_2)$, return $\perp$ if $r \notin \mathcal{R}^h = \mathbb{Z}_p^*$. Otherwise, compute (U, H) as
$$U = g^r, \qquad H = (V, W) = \big((X_1^t X_2)^r,\ (Y_1^t Y_2)^r\big).$$

$\texttt{Invert}(\texttt{cp}, sk, t, U)$**:** return $\perp$ if $U \notin \mathbb{G}$. Otherwise, parse sk as $(x_1, x_2, y_1, y_2) \in (\mathbb{Z}_p^*)^4$ and output $H = (V, W) = (U^{t \cdot x_1 + x_2}, U^{t \cdot y_1 + y_2})$

The following results are proved in Appendix C and D, respectively.

Lemma 1. *The above tag-based hint scheme is anonymous if the DDH assumption holds in $\mathbb{G}$.*

Lemma 2. *The hint scheme is strongly robust under the discrete logarithm assumption in $\mathbb{G}$.*

We will now use an anonymous hint system to generically obtain ANOBE with efficient decryption.

5.2 ANOBE with Efficient Decryption

Let $\pi^{\text{hint}} = (\texttt{Keygen}, \texttt{Hint}, \texttt{Invert})$ be an anonymous hint system with its set of common public parameters cp. Let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme and $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme.

$\texttt{BE.Setup}(\lambda, n)$: Obtain $(par) \leftarrow \mathsf{Gen}(\lambda)$ and, for $i = 1$ to n, and generate encryption key pairs $(\tilde{sk}_i, \tilde{pk}_i) \leftarrow \pi^{\text{pke}}.\mathsf{Keygen}(par)$ and hint key pairs $(sk_i^h, pk_i^h) \leftarrow \pi^{\text{hint}}.\texttt{Keygen}(\mathsf{cp})$. The master public key consists of

$$\text{BE-MPK} = \Big(\mathsf{cp}, par,\ \{(\tilde{pk}_i, pk_i^h)\}_{i=1}^n,\ \Sigma\Big)$$

and the master private key is $\text{BE-MSK} = \{\tilde{sk}_i, sk_i^h\}_{i=1}^n$.

$\texttt{BE.Key-Gen}(\text{BE-MPK}, \text{BE-MSK}, i)$: parse BE-MSK as $\{\tilde{sk}_i, sk_i^h\}_{i=1}^n$ and output $sk_i = (\tilde{sk}_i, sk_i^h)$.

$\texttt{BE.Enc}(\text{BE-MPK}, M, S)$: to encrypt a message M for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Then, choose $r \stackrel{\$}{\leftarrow} \mathcal{R}^h$ and compute $(U, H_j) = \pi^{\text{hint}}.\texttt{Hint}(\mathsf{cp}, \mathsf{VK}, pk_{i_j}^h, r)$ for $j = 1$ to ℓ (recall that the first output U of $\texttt{Hint}$ does not depend on the public key). Then, for each $j \in \{1, \ldots, \ell\}$, compute a ciphertext $C_j = \pi^{\text{pke}}.\mathsf{Encrypt}(par, \tilde{pk}_{i_j}, M||\mathsf{VK})$. Choose a random permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ and set the final ciphertext as

$$C = \big(\mathsf{VK}, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma\big),$$

where $\sigma = \mathcal{S}\big(\mathsf{SK}, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)})\big)$.

$\texttt{BE.Dec}(\text{BE-MPK}, sk_i, C)$: given $sk_i = (\tilde{sk}_i, sk_i^h)$ and $C = \big(\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma\big)$, return $\perp$ if $\mathcal{V}\big(\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma\big) = 0$ or if U is not in the appropriate space. Otherwise, compute $H = \pi^{\text{hint}}.\texttt{Invert}(\mathsf{cp}, sk_i^h, \mathsf{VK}, U)$. If $H \neq H_j$ for all $j \in \{1, \ldots, \ell\}$, return $\perp$. Otherwise, let j be the smallest index such that $H = H_j$ and compute $M' = \pi^{\text{pke}}.\mathsf{Decrypt}(\tilde{sk}_i, C_j)$. If M' can be parsed as $M' = M||\mathsf{VK}$ for some M of appropriate length, return M. Otherwise, output $\perp$.

The correctness of this scheme follows directly from the correctness and weak robustness of its component schemes π^{hint} and π^{pke}.

The proof of the following theorem is deferred to Appendix E.

Theorem 4. *The above construction is adaptively ANO-IND-CCA secure assuming that (i) π^{hint} is anonymous; (ii) π^{pke} is AI-CCA secure and weakly robust under chosen-ciphertext attacks; (iii) Σ is a strongly unforgeable one-time signature.*

In [5], a technique to speed up decryption was presented. The scheme of [5] can be seen as using a hint scheme where tags are empty strings and pairs (U, H_j) consist of $U = g^r$ and $H_j = H(X_{i_j}^r)$, where H is a random oracle and $X_{i_j} \in \mathbb{G}$ is the public key of the hint scheme. In the present context, it is tempting to believe that simple hints of the form $X_{i_j}^r$ suffice to achieve efficient decryption in the standard model. Indeed, one step of the proof consists of a DDH-based transition from one hybrid game to another and, during that specific transition, the simulator $\mathcal{B}$ could simply handle all decryption queries using the private keys $\{\tilde{sk}_i\}_{i=1}^n$ in the underlying encryption scheme since it knows them all. For reasons that will become apparent in the proof of a key lemma for Theorem 4 below, this does not suffice. The reason is that, the adversary can issue decryption queries where $(g, U = g^r, X_{i_j}, H_{i_j} = X_{i_j}^{r'})$ does *not* form a Diffie-Hellman tuple. In this case, the answer of the simulator would differ from that of the real decryption procedure in the chosen-ciphertext scenario: more precisely, the simulation could accept a ciphertext that would be rejected by a real decryption.

In [5], this problem was addressed using a random oracle and the Gap Diffie-Hellman assumption [39]: each hint was of the form $H_j = H(X_{i_j}^r)$, where H is the random oracle. By invoking the DDH-oracle at each random oracle query, the simulator was able to figure out which ciphertext components had to be decrypted so as to perfectly emulate the real decryption algorithm. Here, we address this issue in the standard model using the tag-based anonymous hint primitive.

From a practical standpoint, it is convenient to instantiate the above construction by combining our DDH-based hint scheme with an encryption scheme based on the same assumption such as the Cramer-Shoup cryptosystem. Interestingly both schemes can be instantiated using the same DDH-hard cyclic group. Considering efficiency, it is moreover possible to recycle the group element g^r of the hint system and simultaneously use it as part of a Cramer-Shoup ciphertext. In the security proof, everything goes through with these optimizations although we omit the details here.

6 Reducing the Size of the Ciphertext with Randomness Re-Use

This section considers *randomness re-use* as a technique to optimize ANOBE constructions. Randomness re-use [7,4] is a powerful tool that provides computational and bandwidth savings. In [7], Bellare *et al.* introduce a property, called *reproducibility*, providing a condition under which randomness re-use is secure. We define the notion of *key-less reproducibility*, which is better suited for the anonymity setting.

Definition 6. *Let* $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *be a PKE scheme. Let* $\mathcal{M}$ *and* $\mathcal{R}$ *be the message and randomness space of* π^{pke}*. Let* R *be an algorithm that takes as input the public parameters, a ciphertext, another random message and a key pair* (sk, pk)*, and outputs a ciphertext. Consider the experiment:*

```
$\mathbf{Exp}^{\mathsf{KLR}}_{\pi^{\text{pke}},R}(\lambda)$
  $(par) \xleftarrow{\$} \mathsf{Gen}(\lambda)$
  $(pk, sk) \xleftarrow{\$} \mathsf{Keygen}(par)$
  $m \xleftarrow{\$} \mathcal{M};\ r \xleftarrow{\$} \mathcal{R}$
  $c = \mathsf{Encrypt}(pk, m; r)$
  $(pk', sk') \xleftarrow{\$} \mathsf{Keygen}(par)$
  $m' \xleftarrow{\$} \mathcal{M}$
  return $1$ if $\mathsf{Encrypt}(par, pk', m'; r) = R(par, c, m', pk', sk')$ and $0$ otherwise.
```

π^{pke} *is* key-less reproducible *if, for any* λ*, there is a PPT algorithm* R *such that the above experiment outputs 1 with probability 1.*

We note that this definition differs from the one in [7] since the algorithm R does not take pk (the public key under which c was created) as an input. Indeed, this is a crucial difference which allows extending the notion of reproducibility to the context where anonymity is required. We now reconsider the generic construction for ANOBE presented in Section 3.2.

Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme, and let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a key-less reproducible PKE scheme. We call $\text{ANOBE}_{rr}^{\pi^{\text{pke}},\Sigma}$ the scheme constructed from Σ and π^{pke} as follows.

`BE.Setup`, `BE.Key-Gen`, `BE.Dec` are as in Section 3.2.

`BE.Enc`(BE-MPK, M, S): to encrypt M for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Choose $r \overset{\$}{\leftarrow} \mathcal{R}$, where $\mathcal{R}$ is the randomness space of π^{pke}_{par}. Then, for each $j = 1$ to ℓ, compute $C_j = \mathsf{Encrypt}(par, pk_{i_j}, M||\mathsf{VK}; r)$. The final BE ciphertext consists of $C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \rightarrow \{1, \ldots, \ell\}$ is a random permutation.

Theorem 5. *Let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an AI-CCA secure, weakly robust and key-less reproducible PKE scheme. Let Σ be a strongly unforgeable one-time signature scheme. Then $ANOBE_{rr}^{\pi^{\text{pke}}, \Sigma}$ is adaptively ANO-IND-CCA secure.*

The proof for Theorem 5 is analogous to that of Theorem 2, the only difference being the use of algorithm R in the simulation.
Proof sketch. The proof follows precisely the proof of Theorem 2 up until the BE challenge ciphertext is generated. The modifications are in the following steps and apply to both Lemma 4 and Lemma 5.

1. For $j = 1$ to $k-1$, $\mathcal{B}$ sets $C_j = R(par, C^\star, M_1||\mathsf{VK}^\star, pk_{\rho_j}, sk_{\rho_j})$.
2. For $j = k+1$ to ℓ, $\mathcal{B}$ computes $C_j = R(par, C^\star, M_0||\mathsf{VK}^\star, pk_{\theta_j}, sk_{\theta_j})$.
3. Finally, set $C_k = C^\star$.

We observe that $\mathcal{B}$ knows all the necessary secret keys since it generated them on its own at the beginning of the simulation. The proof then continues as in Theorem 2.

We note that there is no further loss in the security reduction since the key-less reproducibility property of π^{pke} implies that $\mathsf{Encrypt}(par, pk', m'; r) = R(par, \mathsf{Encrypt}(par, pk, m; r), m', pk', sk')$ with probability 1. □

We have shown that the key-less reproducibility of a PKE scheme guarantees that randomness can be re-used securely. We can exploit this property to compress the ANOBE ciphertexts and, depending on the concrete instantiation, significantly increase the efficiency of the scheme. More precisely, given an $\text{ANOBE}_{rr}^{\pi^{\text{pke}}, \Sigma}$ ciphertext $C = (\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma)$, let ccc denote the *common ciphertext components* that may arise in $C_{\tau(1)}, \ldots, C_{\tau(\ell)}$ from sharing randomness across PKE components, *i.e.*,

$$C_{\tau(1)} = (\mathsf{ccc}, \tilde{c}_{\tau(1)}), \ldots, C_{\tau(\ell)} = (\mathsf{ccc}, \tilde{c}_{\tau(\ell)}).$$

The compressed ANOBE ciphertext will be $\tilde{C} = (\mathsf{VK}, \mathsf{ccc}, \tilde{c}_{\tau(1)}, \ldots, \tilde{c}_{\tau(\ell)}, \sigma)$. Upon receipt, the user simply reconstitutes the original ciphertext C and runs `BE.Dec` as usual. We explore instantiations of this idea next.

6.1 An Efficient Instantiation of ANOBE from Kurosawa-Desmedt

This section presents an ANOBE scheme based on a randomness re-using variant of the Kurosawa-Desmedt encryption scheme KD* [36] (which is described in Appendix G). KD* is an ideal candidate for our purposes since it is AI-CCA secure (the proof is in Appendix G) and key-less reproducible (Lemma 10). Moreover, as shown in Appendix G, it can easily be made strongly robust [2] under mild assumptions on the involved symmetric components: namely, the hash function H must be pre-image resistant; the key derivation function has to be collision-resistant and the symmetric encryption scheme must be *key-binding* (as defined in [37]) as well as a secure authenticated encryption scheme (as already required by its proof of IND-CCA security [21]).

Concrete instantiation. The following scheme is obtained from KD* by optimizing our generic construction and removing redundant ciphertext components from the overall ciphertext: namely, (g_1^r, g_2^r) only has to appear once in the ANOBE ciphertext and this change is easily seen not to affect the security analysis.

$\texttt{BE.Setup}(\lambda, n)$**:** chooses a group $\mathbb{G}$ of prime order $p > 2^\lambda$ with generators $g_1, g_2 \xleftarrow{\$} \mathbb{Z}_p$ as well as a universal one-way hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. It also selects a key derivation function $\mathsf{KDF} : \mathbb{G} \to \{0,1\}^k$, for some $k \in \mathsf{poly}(\lambda)$, a strongly unforgeable one-time signature scheme $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a symmetric authenticated encryption scheme $\Pi^{\text{sym-enc}} = (\mathsf{E}, \mathsf{D})$. Then, for $i = 1$ to n, choose $x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2} \xleftarrow{\$} \mathbb{Z}_p$ and compute elements $c_i = g_1^{x_{i,1}} g_2^{x_{i,2}}$, $d_i = g_1^{y_{i,1}} g_2^{y_{i,2}}$. The master public key consists of

$$\text{BE-MPK} = \Big(\mathbb{G},\ g_1,\ g_2, \{c_i, d_i\}_{i=1}^n,\ H,\ \mathsf{KDF},\ \Pi^{\text{sym-enc}},\ \Sigma\Big)$$

and the master private key is $\text{BE-MSK} = \big(\{x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2}\}_{i=1}^n\big)$.

$\texttt{BE.Key-Gen}(\text{BE-MPK}, \text{BE-MSK}, i)$**:** parse BE-MSK as $\big(\{x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2}\}_{i=1}^n\big)$ and output user i's private key $sk_i = (x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2})$.

$\texttt{BE.Enc}(\text{BE-MPK}, m, S)$**:** to encrypt m for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$, pick $r \xleftarrow{\$} \mathbb{Z}_p^*$ and compute the common part of the ciphertext $(u_1, u_2) = (g_1^r, g_2^r) \in \mathbb{G}^2$. Then, for each $j = 1$ to ℓ, compute $\alpha = H(u_1, u_2)$ as well as

$$v_j = \big(c_{i_j} \cdot d_{i_j}^{\alpha}\big)^r, \qquad K_j = \mathsf{KDF}(v_j), \qquad e_j = \mathsf{E}_{K_j}(m||\mathsf{VK}).$$

The ciphertext is $C = \big(\mathsf{VK}, u_1, u_2, e_{\tau(1)}, \ldots, e_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, (u_1, u_2, e_{\tau(1)}, \ldots, e_{\tau(\ell)})\big)$ is a signature and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

$\texttt{BE.Dec}(sk_i, C)$**:** given $sk_i = (x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2})$ and the BE ciphertext $C = \big(\mathsf{VK}, u_1, u_2, e_1, \ldots, e_\ell, \sigma\big)$, return $\perp$ if $\mathcal{V}\big(\mathsf{VK}, (u_1, u_2, e_1, \ldots, e_\ell), \sigma\big) = 0$. Otherwise, compute $\alpha = H(u_1, u_2)$. For $j = 1$ to ℓ, do the following.

1. Compute $v_j = u_1^{x_{i,1} + \alpha \cdot y_{i,1}} \cdot u_2^{x_{i,2} + \alpha \cdot y_{i,2}}$ and $K_j = \mathsf{KDF}(v_j) \in \{0,1\}^k$. If $M = \mathsf{D}_{K_j}(e_j) \neq \perp$ and if M can be parsed as $m||\mathsf{VK}$ for some message m, return m.
2. If $i = \ell$, return $\perp$.

In Appendix G we show that KD* has all the properties required for our ANOBE construction to be secure: it is AI-CCA (Theorem 7), weakly robust (Theorem 6) and key-less reproducible (Lemma 10). The following result is a direct consequence of Theorem 5 and Theorem 7, Theorem 6 and Lemma 10.

Corollary 1. *The above BE construction is adaptively ANO-IND-CCA secure.*

Optimizations using labels. We note that the above concrete ANOBE system can be further optimized by using a variant of KD* that supports labels. A label L is essentially a public string that can be non-malleably bound to the ciphertext. To do this in this setting, the simplest solution is to compute the hash value α as $\alpha = H(u_1, u_2, L)$. With this modification, Cramer-Shoup and Kurosawa-Desmedt cryptosystems are easily seen to satisfy label-augmented definitions of chosen-ciphertext security [42] and anonymity under chosen-ciphertext attacks (see, e.g., [32]). The only change in the security proofs is that the hash function H must be assumed collision-resistant (and not only target collision-resistant).

Using labels, the above construction is easily modified in such a way that, instead of encrypting ℓ times a concatenation $M||\mathsf{VK}$, we can only encrypt M alone and settle for including VK among the inputs of H and compute $\alpha = H(u_1, u_2, \mathsf{VK})$. By doing so, we obtain significantly shorter ciphertexts.

Efficiency comparison. It is interesting to compare the above scheme with the one of Section 3.2 when the latter is instantiated with the Kurosawa-Desmedt cryptosystem. We assume that labels are used in all schemes so as to avoid encrypting VK with the plaintext and also consider that a multi-exponentiation with two base elements has roughly the same cost as a single-base exponentiation. At the sender's end, the scheme of Section 3.2 requires $3 \cdot |S|$ exponentiations which is essentially three times as expensive as in the above scheme (where only $|S|$ exponentiations are needed).

From a bandwidth point of view, the above construction requires to transmit $|S|$ symmetric authenticated encryptions (which typically consist of a symmetric encryption and a MAC) in addition to two group elements and a one-time verification key. The saving provided by randomness-re-using techniques is thus about $2 \cdot |S|$ group elements, which, in practice, would be about 50% of the ciphertext size without randomness re-use. Assuming that encrypted messages are at least as short as a group element (which seems reasonable if we encrypt a symmetric key and if group elements are in an elliptic curve subgroup), the ciphertext overhead is about $|S| \cdot (\ell_{\mathbb{G}} + \ell_{MAC})$, where $\ell_{\mathbb{G}}$ and ℓ_{MAC} denote the bitlength of group elements and MAC tags, respectively.

7 Conclusions and Open Problems

We have seen that in the context of broadcast encryption the main focus of research to date has been on reducing ciphertext size. Achieving this has entailed sacrificing *all* anonymity properties. Yet we have argued that anonymity is a *fundamental property* to strive for in broadcast encryption. With the aim of highlighting the importance of this overlooked feature, we have formally defined the notion of anonymous broadcast encryption (ANOBE) and given several constructions for this primitive. We have also shown how these constructions can be improved via anonymous hint systems (to optimize decryption performance) and randomness re-use (to reduce the ciphertext size and the computational costs of encryption).

Much work still needs to be done in this area, from improving the efficiency of ANOBE schemes to considering all the additional properties that can be found in standard BE, such as traitor tracing, revocation, dynamism of users joining the system, and realising them in the anonymous setting. There is still a gap between the sizes of ciphertexts in state-of-the-art BE schemes and our ANOBE schemes. This gap is hidden in the constants in an asymptotic evaluation of ciphertext size (when the true size of ciphertexts is measured) but is nevertheless significant in practice. A major challenge, then, is to further reduce the size of ciphertexts in ANOBE, whilst maintaining its full anonymity properties.

Acknowledgements

The work in this paper was supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The work in this paper was sponsored in part by the US Army Research Laboratory and the UK Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The first author acknowledges the Belgian Fund for Scientific Research (F.R.S.-F.N.R.S.) for his "Collaborateur scientifique" fellowship. The second author was supported by an EPSRC Leadership Fellowship, EP/H005455/1.

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.
2. Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2010.
3. Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of kurosawa-desmedt kem. In *Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 128–146. Springer, 2005.
4. Manuel Barbosa and Pooya Farshim. Randomness reuse: Extensions and improvements. In Steven D. Galbraith, editor, *IMA Int. Conf.*, volume 4887 of *Lecture Notes in Computer Science*, pages 257–276. Springer, 2007.
5. Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Aviel D. Rubin, editors, *Financial Cryptography 2006*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer, 2006.
6. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001.
7. Mihir Bellare, Alexandra Boldyreva, Kaoru Kurosawa, and Jessica Staddon. Multirecipient encryption schemes: How to save on bandwidth and computation without sacrificing security. *IEEE Trans. on Information Theory*, 53(11):3927–3943, 2007.
8. Mihir Bellare, Alexandra Boldyreva, and Jessica Staddon. Randomness re-use in multi-recipient encryption schemes. In *PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 85–99. Springer, 2003.
9. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.
10. Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 455–470. Springer, 2008.
11. Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Vaudenay [43], pages 573–592.
12. Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.
13. Jan Camenisch, Markulf Kohlweiss, Alfredo Rial, and Caroline Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography 2009*, volume 5443 of *Lecture Notes in Computer Science*, pages 196–214. Springer, 2009.
14. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.
15. David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In *Eurocrypt 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 127–145. Springer, 2008.

16. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO 1982*, pages 199–203, 1982.
17. David Chaum. Security without identification: Transaction systems to make Big Brother obsolete. *Commun. ACM 1985*, 28(10):1030–1044, 1985.
18. David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.
19. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kurosawa [35], pages 200–215.
20. Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer, 2007.
21. Yvo Desmedt, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. A new and improved paradigm for hybrid encryption secure against chosen-ciphertext attack. *J. of Cryptology*, 23(1):91–120, January 2010.
22. Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In *Digital Rights Management Workshop 2002*, pages 61–80, 2002.
23. Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 188–209. Springer, 2005.
24. Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In *Public Key Cryptography 2012 (PKC 2012)*, Lecture Notes in Computer Science. Springer, 2012.
25. Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.
26. Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In *EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 1999.
27. Rosario Gennaro and Victor Shoup. A note on an encryption scheme of Kurosawa and Desmedt. Cryptology ePrint Archive: Report 2004/194, 2004.
28. Craig Gentry. Practical identity-based encryption without random oracles. In Vaudenay [43], pages 445–464.
29. Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.
30. Jens Groth. Efficient maximal privacy in boardroom voting and anonymous broadcast. In Ari Juels, editor, *Financial Cryptography 2004*, volume 3110 of *Lecture Notes in Computer Science*, pages 90–104. Springer, 2004.
31. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.
32. Aggelos Kiayias, Yannis Tsiounis, and Moti Yung. Group encryption. In Kurosawa [35], pages 181–199.
33. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *Theory of Cryptograhy Conference 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.
34. Kaoru Kurosawa. Multi-recipient public-key encryption with shortened ciphertext. In *PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 2002.
35. Kaoru Kurosawa, editor. *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*. Springer, 2007.
36. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *Crypto 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2004.
37. Payman Mohassel. A closer look at anonymity and robustness in encryption schemes. In *Asiacrypt 2010*, volume 6477 of *LNCS*, pages 501–518. Springer, 2010.
38. Moni Naor, Dalit Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *Crypto 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.
39. Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In *PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2001.
40. Kenneth G. Paterson and Sriramkrishnan Srinivasan. Security and anonymity of identity-based encryption with multiple trusted authorities. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 354–375. Springer, 2008.

41. Kenneth G. Paterson and Sriramkrishnan Srinivasan. Building key-private public-key encryption schemes. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 2009*, volume 5594 of *Lecture Notes in Computer Science*, pages 276–292. Springer, 2009.
42. Victor Shoup. A proposal for an iso standard for public key encryption (version 2.1). Manuscript, 2001.
43. Serge Vaudenay, editor. *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.
44. Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 314–332. Springer, 2010.
45. Danfeng Yao, Nelly Fazio, Yevgeniy Dodis, and Anna Lysyanskaya. ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *ACM Conference on Computer and Communications Security 2004*, pages 354–363. ACM, 2004.

A Proof of Theorem 1

We now give a proof of Theorem 1.

Proof. Recall that, since $|S_0| = |S_1| = \ell$, we have $|S_0 \backslash S_1| = |S_1 \backslash S_0| = |S_b| - |S_0 \cap S_1|$ for each $b \in \{0,1\}$. We consider a sequence of games starting with Game 0 where the adversary is given an encryption of M_0 for S_0. In the last game, the adversary obtains an encryption of M_1 under S_1.

Game 0_{real}: is the real game when the challenger's bit is set to $b = 0$. The ANOBE adversary $\mathcal{A}$ is given public parameters BE−MPK consisting of n public key encryption keys $\{pk_i\}_{i=1}^{n}$. For each $i \in \{1,\ldots,n\}$, user i's private key is sk_i. In the first stage, $\mathcal{A}$ adaptively chooses indices $i \in \{1,\ldots,n\}$ and obtains the corresponding sk_i. The adversary may also query the decryption oracle by sending requests (C,i) which are answered using the relevant private key sk_i. In the challenge step, $\mathcal{A}$ chooses messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1,\ldots,n\}$ of equal size $|S_0| = |S_1| = \ell$. The challenger generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and returns the challenge ciphertext $C^\star = \big(C_1,\ldots,C_n,\sigma\big)$ where $\sigma = \mathcal{S}(\mathsf{SK}^\star,(C_1,\ldots,C_n))$ and, for each $j \in \{1,\ldots,n\}$, C_j is computed as $C_j = \mathsf{Encrypt}(pk_j, M_0||\mathsf{VK}^\star)$ if $j \in S_0$ and $C_j = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$ if $j \notin S_0$. In the second phase, $\mathcal{A}$ is allowed to make more corruption queries for indices i such that $i \in \{1,\ldots,n\}\backslash(S_0 \triangle S_1)$ and is granted further access to the decryption oracle under the usual restriction. Upon termination, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and we define E_0^{real} to be the event that $b' = 0$.

Game 0: is as Game 0_{real} with the difference that the challenger now rejects all post-challenge decryption queries $\big(C = (C_1,\ldots,C_n,\sigma), i\big)$ for which $C_i = C_i^\star$ (*i.e.*, the i-component of C coincides with that of the challenge ciphertext). Clearly, the only situation where the challenger rejects a ciphertext that would not have been rejected in Game 0_{real} is when $\mathcal{A}$ breaks the security of the one-time signature. It is easy to see since $C_i = C_i^\star$ decrypts to a message whose last v bits form the challenge verification key $\mathsf{VK}^\star$ as in the challenge phase. We call E_0 the event that $\mathcal{A}$ outputs $b' = 0$ in Game 0.

To describe subsequent games, it is convenient to represent the sets S_0 and S_1 as n-bit words $s_{01}\ldots s_{0n} \in \{0,1\}^n$ and $s_{11}\ldots s_{1n} \in \{0,1\}^n$ such that, for each $b \in \{0,1\}$ and $j \in \{1,\ldots,n\}$, $s_{bj} = 1$ if and only if $j \in S_b$.

Game k $(1 \leq k \leq n)$: From the two adversarially-chosen sets $S_0, S_1 \subset \{1,\ldots,n\}$ and their respective n-bit words $s_{01}\ldots s_{0n}$ and $s_{11}\ldots s_{1n}$, the challenger $\mathcal{B}$ generates the challenge ciphertext as follows.

1. If $s_{0j} = s_{1j} = 1$, set $C_j = \mathsf{Encrypt}(pk_j, M_1||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \mathsf{Encrypt}(pk_j, M_0||\mathsf{VK}^\star)$ if $j > k$. If $s_{0j} = s_{1j} = 0$ set $C_j = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$.
2. If $s_{0j} = 1$ and $s_{1j} = 0$, set $C_j = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \mathsf{Encrypt}(pk_j, M_0||\mathsf{VK}^\star)$ if $j > k$.
3. If $s_{0j} = 0$ and $s_{1j} = 1$, set $C_j = \mathsf{Encrypt}(pk_j, M_1||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$ if $j > k$.

The adversary is then returned $C^\star = \big(C_1, \ldots, C_n, \sigma\big)$ and the second phase is handled as in previous games. We call E_k the event of $\mathcal{A}$ outputting $b' = 0$ at the end of Game k.

Game n_{real}: is identical to Game n with the difference that, when handling decryption queries, the challenger no longer returns $\perp$ in decryption queries $(C = (C_1, \ldots, C_n, \sigma), i)$ such that that $C_i = C_i^\star$. Game n_{real} thus coincides with the real game when the challenger's bit equals $b = 1$. We let E_n^{real} be the event that $\mathcal{A}$ outputs the bit $b' = 0$ at the end of Game ℓ_{real}.

Game 0_{real} and Game 0 are clearly indistinguishable if the one-time signature is strongly unforgeable and the same argument can be made about Game ℓ and Game ℓ_{real}.

We thus have $|\Pr[E_0^{real}] - \Pr[E_0]| = |\Pr[E_n^{real}] - \Pr[E_n]| \leq \mathbf{Adv}_{\mathrm{OTS}}^{\mathrm{suf}}(\mathcal{A})$. As for other game transitions, they are justified by lemmas 3 which demonstrates that, if Game k and Game $k-1$ can be distinguished for some $k \in \{1, \ldots, n\}$, there must exist either an IND-CCA adversary $\mathcal{B}$ against the underlying encryption scheme. Putting the above altogether, we find

$$|\Pr[E_0^{real}] - \Pr[E_n^{real}]| \leq 2 \cdot \mathbf{Adv}_{\mathrm{OTS}}^{\mathrm{suf}}(\mathcal{A}) + n \cdot \mathbf{Adv}^{\text{ind-cca}}(\mathcal{B}).$$

□

Lemma 3. *For any $k \in \{1, \ldots, n\}$, Game k is indistinguishable from Game $k-1$ if the public key encryption scheme is IND-CCA. More precisely, we have*

$$|\Pr[E_k] - \Pr[E_{k-1}]| \leq \mathbf{Adv}^{\text{ind-cca}}(\mathcal{B}).$$

Proof. Towards a contradiction, let us assume that an adversary $\mathcal{A}$ can distinguish Game k and Game $k-1$. We show that it implies a chosen-ciphertext adversary against the cryptosystem.

We first recall that, in the challenge phase, the adversarially-chosen messages M_0, M_1 and sets S_0, S_1 must be such that either

- $S_0 = S_1$ and $M_0 \neq M_1$, in which case the adversary cannot corrupt any user in $S_0 = S_1$ (and, of course, we must have $|S_0| = |S_1| \geq 1$).
- $S_0 \neq S_1$, in which case the adversary is disallowed to corrupt anyone in $S_0 \triangle S_1$.

If we consider the n-bit words $s_{01} \ldots s_{0n} \in \{0,1\}^n$ and $s_{11} \ldots s_{1n} \in \{0,1\}$ associated with S_0 and S_1, Game k is identical to Game $k-1$ if $s_{0k} = s_{1k} = 0$ (since C_k is an encryption of $\perp$ in both games) and we thus assume that $s_{0k} = s_{1k} = 1$ or $s_{0k} \neq s_{1k}$. Moreover, if $s_{0k} = s_{1k} = 1$ (in other words, if $k \in S_0 \cap S_1$), the adversary can only corrupt sk_k in the situation where $M_0 = M_1$, in which case Game k and Game $k-1$ are also identical. In the following, we can thus only consider the situation $s_{0k} \neq s_{1k}$ (*i.e.*, $k \in S_0 \triangle S_1$), in which the adversary cannot legally query sk_k.

Our IND-CCA adversary $\mathcal{B}$ receives a public key $pk^\star$ from its challenger and to prepare BE-MPK for $\mathcal{A}$, it has to generate n encryption keys $pk_1, \ldots, pk_n$. To do this, $\mathcal{B}$ defines $pk_k = pk^\star$. Then, $\mathcal{B}$ runs the key generation algorithm of π^{pke} itself and generates $n-1$ key pairs $(sk_i, pk_i) \leftarrow \mathsf{Keygen}(1^\lambda)$ for each $i \in \{1, \ldots, n\} \backslash \{k\}$. It finally hands the master public key BE-MPK $= \big(\{pk_i\}_{i=1}^n, \Sigma\big)$ to $\mathcal{A}$.

At any time, $\mathcal{A}$ can corrupt an arbitrary user $i \in \{1, \ldots, n\}$ depending on the previously collected information. At each corruption query, $\mathcal{B}$ can consistently answer the query since it knows secret keys $\{sk_i\}_{i \neq k}$ (recall that, according to the rules, should be denied access to sk_k). When $\mathcal{A}$ queries the decryption of a ciphertext $(C = (C_1, \ldots, C_n, \sigma), i)$, we assume that $i = k$ (*i.e.*, the query involves the challenge key $pk_k = pk^\star$) since $\mathcal{B}$ can always decrypt by itself otherwise. To simulate the behavior of the decryption algorithm without knowing $sk_k = sk^\star$, $\mathcal{B}$ invokes its own decryption oracle on C_k. If the IND-CCA challenger's response is not $\perp$ and can be parsed as $M||\mathsf{VK}$, for some message $M \in \{0,1\}^{m-v}$ and some bitstring $\mathsf{VK} \in \{0,1\}^v$, $\mathcal{B}$ returns M to $\mathcal{A}$ if $\mathcal{V}(\mathsf{VK}, (C_1, \ldots, C_n), \sigma) = 1$ and $M \neq \varepsilon$. In any other situation, $\mathcal{B}$ returns $\perp$, meaning that C_k fails to decrypt properly under sk_k.

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size. At this step, $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and constructs two messages M_0', M_1' as follows.

- If $s_{0k} = 1$ and $s_{1k} = 0$, it sets $M_0' = M_0||\mathsf{VK}^\star$ and $M_1' = \varepsilon||\mathsf{VK}^\star$.
- If $s_{0k} = 0$ and $s_{1k} = 1$, it sets $M_0' = \varepsilon||\mathsf{VK}^\star$ and $M_1' = M_1||\mathsf{VK}^\star$.

The two messages M_0' and M_1' are sent to $\mathcal{B}$'s IND-CCA challenger which returns a challenge ciphertext $C^\star = \mathsf{Encrypt}(pk^\star, M_b')$, for some internally flipped random bit $b \in_R \{0,1\}$. The ANOBE challenge ciphertext is generated by setting $C_k^\star = C^\star$ and defining remaining ciphertext components as follows, for $j = 1$ to n.

1. If $s_{0j} = s_{1j} = 1$, set $C_j^\star = \mathsf{Encrypt}(pk_j, M_1||\mathsf{VK}^\star)$ if $j \leq k-1$ and $C_j^\star = \mathsf{Encrypt}(pk_j, M_0||\mathsf{VK}^\star)$ if $j > k$. If $s_{0j} = s_{1j} = 0$ set $C_j = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$.
2. If $s_{0j} = 1$ and $s_{1j} = 0$, set $C_j^\star = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$ if $j \leq k-1$ and $C_j^\star = \mathsf{Encrypt}(pk_j, M_0||\mathsf{VK}^\star)$ if $j > k$.
3. If $s_{0j} = 0$ and $s_{1j} = 1$, set $C_j^\star = \mathsf{Encrypt}(pk_j, M_1||\mathsf{VK}^\star)$ if $j \leq k-1$ and $C_j^\star = \mathsf{Encrypt}(pk_j, \varepsilon||\mathsf{VK}^\star)$ if $j > k$.

The ANOBE adversary $\mathcal{A}$ is given $C = (C_1^\star, \ldots, C_n^\star, \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}^\star, (C_1^\star, \ldots, C_n^\star))$.

In the second phase, $\mathcal{A}$ makes another series of adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ (and *a fortiori* such that $i \neq k$) and $\mathcal{B}$ deals with them as in the first phase. When $\mathcal{A}$ makes a decryption query (C, i), $\mathcal{B}$ parses the ciphertext C as $C = (C_1, \ldots, C_n, \sigma)$ and handles the query using $\{sk_i\}_{i \neq k}$ if $i \neq k$. If $i = k$, $\mathcal{B}$ returns $\perp$ if $C_k = C_k^\star$. If $C_k \neq C_k^\star$, $\mathcal{B}$ can query C_k for decryption to its IND-CCA challenger and proceed as in pre-challenge decryption queries.

At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and $\mathcal{B}$ outputs the same result. It is easy to see that $\mathcal{B}$'s advantage as an IND-CCA adversary is exactly the difference between $\mathcal{A}$'s probabilities of outputting 0 in Game k and Game $k-1$. Indeed, if $\mathcal{B}$'s challenger chooses $b = 0$ (and encrypts M_0' in the challenge phase), $\mathcal{B}$ is playing Game $k-1$. If $b = 1$, $\mathcal{B}$ is rather playing Game k. □

B Proof of Theorem 2

We now give a proof of Theorem 2.

Proof. Recall that, since $|S_0| = |S_1| = \ell$, we have $|S_0 \backslash S_1| = |S_1 \backslash S_0| = |S_b| - |S_0 \cap S_1|$ for each $b \in \{0,1\}$. We consider a sequence of games where the adversary is given an encryption of M_0 for S_0 in Game 0 while, in the last game, the adversary obtains an encryption of M_1 under S_1.

Game 0_{real}: corresponds to the real game when the challenger's bit is $b = 0$. Namely, the ANOBE adversary $\mathcal{A}$ is given public parameters BE-MPK containing par and n public keys $\{pk_i\}_{i=1}^n$. For each $i \in \{1, \ldots, n\}$, user i's private key is sk_i. In the first phase, the adversary $\mathcal{A}$ adaptively chooses indices $i \in \{1, \ldots, n\}$ and obtains the corresponding sk_i. The adversary may also invoke the decryption oracle by making queries (C, i) which are handled using the relevant private key sk_i. In the challenge phase, the adversary $\mathcal{A}$ comes up with messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size $|S_0| = |S_1| = \ell$ with $S_0 \neq S_1$. The challenger generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$, parses S_0 as $\{\theta_1, \ldots, \theta_\ell\}$ and returns the challenge ciphertext $C^\star = \big(\mathsf{VK}^\star, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$ where $C_j = \mathsf{Encrypt}(par, pk_{\theta_j}, M_0||\mathsf{VK}^\star)$ for $j = 1$ to ℓ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation. In the second phase, $\mathcal{A}$ is allowed to make more decryption queries (under the usual restriction) and key queries for arbitrary indices i such that $i \in \{1, \ldots, n\} \backslash (S_0 \triangle S_1)$. Eventually, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$ and we define E_0^{real} to be the event that $b' = 0$.

Game 0: is as Game 0_{real} but the challenger now rejects all post-challenge decryption queries (C, i) where C contains the same verification key $\mathsf{VK}^\star$ as in the challenge phase. We call E_0 the event that $\mathcal{A}$ outputs $b' = 0$ in Game 0.

Game k ($1 \leq k \leq \ell$): From the two adversarially-chosen sets $S_0, S_1 \subset \{1, \ldots, n\}$, the challenger $\mathcal{B}$ defines the value $\phi = |S_0 \cap S_1|$ and then considers two ordered sets $S_0' = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$, $S_1' = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ that are obtained by ordering S_0 and S_1 in such a way that $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$ and $\theta_j \neq \rho_j$ if $j \in \{\phi+1, \ldots, \ell\}$. Then, $\mathcal{B}$ generates the challenge ciphertext as follows.

1. For $j = 1$ to ϕ, set $C_j = \mathsf{Encrypt}(par, pk_{\theta_j}, M_1||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \mathsf{Encrypt}(par, pk_{\theta_j}, M_0||\mathsf{VK}^\star)$ if $j > k$.
2. For $j = \phi + 1$ to ℓ, set $C_j = \mathsf{Encrypt}(par, pk_{\rho_j}, M_1||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \mathsf{Encrypt}(par, pk_{\theta_j}, M_0||\mathsf{VK}^\star)$ if $j > k$.

The adversary is then returned $C^\star = \big(\mathsf{VK}^\star, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, for a randomly chosen permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$, and the second phase is handled as in previous games. We call E_k the event of $\mathcal{A}$ outputting $b' = 0$ at the end of Game k.

Game ℓ_{real}: is identical to Game ℓ with the difference that, when handling decryption queries, the challenger no longer rejects ciphertexts that contain the verification key $\mathsf{VK}^\star$. Game ℓ_{real} actually proceeds like the real game when the challenger's bit is $b = 1$. We let E_ℓ^{real} be the event that $\mathcal{A}$ outputs the bit $b' = 0$ at the end of Game ℓ_{real}.

Game 0_{real} and Game 0 are clearly indistinguishable if the one-time signature is strongly unforgeable and the same argument can be made about Game ℓ and Game ℓ_{real}.

We thus have $|\Pr[E_0^{real}] - \Pr[E_0]| = |\Pr[E_\ell^{real}] - \Pr[E_\ell]| \leq \mathbf{Adv}_{\mathrm{OTS}}^{\mathrm{suf}}(\mathcal{A})$. As for other game transitions, they are justified by lemmas 4 and 5 that separately consider the situations where $k \leq \phi$ and $k > \phi$. More precisely, we have that, if Game k and Game $k - 1$ can be distinguished for some $k \in \{1, \ldots, \ell\}$, lemmas 4 and 5 show that there exists either a AI-CCA adversary $\mathcal{B}_2$ or a WROB-CCA adversary $\mathcal{B}_3$ (see appendix F for definitions of these two notions) against the encryption scheme. Putting the above arguments altogether, we obtain

$$\begin{aligned}|\Pr[E_0^{real}] - \Pr[E_\ell^{real}]| &\leq 2 \cdot \mathbf{Adv}_{\mathrm{OTS}}^{\mathrm{suf}}(\mathcal{A}) + n^2 \cdot \ell \cdot \Big(\mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}_3)\Big) \\ &\leq 2 \cdot \mathbf{Adv}_{\mathrm{OTS}}^{\mathrm{suf}}(\mathcal{A}) + n^3 \cdot \Big(\mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}_3)\Big).\end{aligned}$$

□

Lemma 4. *For each $k \in \{1, \ldots, \phi\}$, Game k is indistinguishable from Game $k-1$ if the underlying encryption scheme is IND-CCA. More precisely, we have*

$$|\Pr[E_k] - \Pr[E_{k-1}]| \leq n \cdot \mathbf{Adv}^{\text{ind-cca}}(\mathcal{B}).$$

Proof. Assuming that an attacker $\mathcal{A}$ can distinguish Game k and Game $k-1$, we build a chosen-ciphertext adversary against the public key encryption scheme. For each $k \in \{1, \ldots, \phi\}$, we observe that Game k and Game $k-1$ are identical when $M_0 = M_1$ and we thus assume $M_0 \neq M_1$, so that the adversary cannot corrupt any user in $S_0 \cap S_1$.

Our IND-CCA adversary $\mathcal{B}$ obtains par and a public key $pk^\star$ from its challenger and it has to prepare a master public key BE-MPK comprising n encryption keys $pk_1, \ldots, pk_n$ for the ANOBE adversary $\mathcal{A}$. To this end, picks $i^\star \stackrel{\$}{\leftarrow} \{1, \ldots, n\}$ at random and defines $pk_{i^\star} = pk^\star$. Then, $\mathcal{B}$ runs Keygen and generates $n-1$ key pairs (sk_i, pk_i) on its own for each $i \in \{1, \ldots, n\} \backslash \{i^\star\}$. It finally gives the master public key BE-MPK $= \big(par, \Sigma, \{pk_i\}_{i=1}^n\big)$ to $\mathcal{A}$.

At any time, $\mathcal{A}$ is allowed to corrupt an arbitrary user $i \in \{1, \ldots, n\}$ depending on the information it gathered so far. At each corruption query, $\mathcal{B}$ aborts and fails in the event that $\mathcal{A}$ chooses to corrupt user $i^\star$. Otherwise, $\mathcal{B}$ is necessarily able to consistently answer the query since it knows secret keys $\{sk_i\}_{i \neq i^\star}$. When the adversary $\mathcal{A}$ makes a decryption query $(C = (\mathsf{VK}, C_1, \ldots, C_\ell, \sigma), i)$, we assume that the query involves the challenge key $pk^\star$ since $\mathcal{B}$ can always decrypt itself using sk_i otherwise. To simulate the decryption algorithm without knowing the challenge private key $sk^\star$, $\mathcal{B}$ proceeds as follows. For $j = 1$ to ℓ, it resorts to its IND-CCA challenger and asks it for the decryption of C_j. If the IND-CCA challenger's response differs from $\perp$ and can be parsed as $M||\mathsf{VK}$, for some message M of appropriate length, $\mathcal{B}$ returns M to $\mathcal{A}$. If the counter j reaches its maximal value ℓ and no decryption query provided a result of the form $M||\mathsf{VK}$, $\mathcal{B}$ returns $\perp$ to indicate that the ciphertext fails to decrypt properly.

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size. At this step, $\mathcal{B}$ re-orders S_0, S_1 as $S_0' = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$, $S_1' = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ where $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$. If $\theta_k \neq i^\star$, $\mathcal{B}$ aborts and declares failure and we denote by Good the event that $\theta_k = i^\star$.

If the event Good occurs, $\mathcal{B}$ chooses a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and sends the messages $(M_0||\mathsf{VK}^\star)$, $(M_1||\mathsf{VK}^\star)$ to its IND-CCA challenger. The latter replies by generating a challenge ciphertext $C^\star = \mathsf{Encrypt}(par, pk^\star, M_b||\mathsf{VK}^\star)$, for some internally flipped random bit $b \stackrel{\$}{\leftarrow} \{0,1\}$. The ANOBE challenge ciphertext is then generated as follows.

1. For $j = 1$ to $k-1$, $\mathcal{B}$ sets $C_j = \mathsf{Encrypt}\big(par, pk_{\theta_j}, (M_1||\mathsf{VK}^\star)\big)$.
2. For $j = k+1$ to ℓ, $\mathcal{B}$ sets $C_j = \mathsf{Encrypt}\big(par, pk_{\theta_j}, (M_0||\mathsf{VK}^\star)\big)$.
3. Finally, set $C_k = C^\star$.

The adversary $\mathcal{A}$ then receives $C = (\mathsf{VK}^\star, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}^\star, C_{\tau(1)}, \ldots, C_{\tau(\ell)})$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

In the second phase, $\mathcal{A}$ makes another series of adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ and $\mathcal{B}$ deals with them as in the first phase. Whenever $\mathcal{A}$ makes a decryption query (C, i), $\mathcal{B}$ parses the ciphertext C as $C = (\mathsf{VK}, C_1, \ldots, C_\ell, \sigma)$ and outputs $\perp$ if $\mathsf{VK} = \mathsf{VK}^\star$ or if σ is invalid. Otherwise, if $i \neq i^\star$, $\mathcal{B}$ simply runs the legal decryption procedure on its own since it knows sk_i. If $i = i^\star$, $\mathcal{B}$ appeals to its IND-CCA challenger and the decryption oracle it is given access to. Namely, ciphertexts $\{C_1, \ldots, C_\ell\}$ are handled by repeating the following steps for $j = 1$ to ℓ.

- If $C_j = C^\star$, $\mathcal{B}$ considers that C_j decrypts to $\perp$ under $sk^\star$ (which is legitimate since $C^\star$ would decrypt to $M_b||\mathsf{VK}^\star$ and $\mathsf{VK} \neq \mathsf{VK}^\star$) and does not make use of its decryption oracle.
- If $C_j \neq C^\star$, $\mathcal{B}$ queries the decryption of C_j. If the result can be parsed as $M||\mathsf{VK}$ for some plaintext M of appropriate length, $\mathcal{B}$ outputs M.

If the counter j reaches ℓ and no decryption query resulted in a plaintext of the form $M||\mathsf{VK}$, $\mathcal{B}$ returns $\perp$.

Eventually, the adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and $\mathcal{B}$ outputs the same result. If $\mathcal{B}$ did not abort, its advantage as an IND-CCA adversary is as large as the difference between $\mathcal{A}$'s probabilities of outputting 0 in Game k and Game $k-1$. Indeed, if $\mathcal{B}$'s challenger chooses $b = 0$, then $\mathcal{B}$ is clearly playing Game $k-1$ whereas, if $b = 1$, $\mathcal{B}$ is playing Game k.

Now, let us assess $\mathcal{B}$'s probability not to abort. First, since $M_0 \neq M_1$ by hypothesis, $\mathcal{A}$ is not allowed to corrupt any user in $S_0 \cap S_1 = \{\theta_1, \ldots, \theta_\phi\}$. Since $\theta_k \in S_0 \cap S_1$, a sufficient condition for $\mathcal{B}$ not to be asked for the unknown private key $sk_{i^\star}$ is to be lucky when drawing $i^\star \xleftarrow{\$} \{1, \ldots, n\}$ and have event Good occurring. This is the case with probability $\Pr[\mathsf{Good}] = 1/n$ since the choice of $i^\star$ is completely independent of $\mathcal{A}$'s view. □

Lemma 5. *For each $k \in \{\phi+1, \ldots, \ell\}$, Game k is indistinguishable from Game $k-1$ if the underlying encryption acheme is AI-CCA secure and weakly robust. More precisely, for any ANOBE adversary distinguishing the two games, there exists either an AI-CCA adversary $\mathcal{B}$ or a WROB-CCA adversary $\mathcal{B}'$ (as defined in appendix F.2) such that*

$$|\Pr[E_k] - \Pr[E_{k-1}]| \leq n^2 \cdot \Big(\mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}')\Big).$$

Proof. We prove that, if an ANOBE attacker $\mathcal{A}$ is able to distinguish Game k and Game $k-1$ for some $k \in \{\phi+1, \ldots, \ell\}$, we can either translate $\mathcal{A}$ into an AI-CCA adversary $\mathcal{B}$ against the encryption scheme or break its WROB-CCA property.

The AI-CCA adversary $\mathcal{B}$ takes as input par and two public keys $pk_0^\star$, $pk_1^\star$ from its AI-CCA challenger and we call $sk_0^\star$ and $sk_1^\star$ the underlying private keys. Algorithm $\mathcal{B}$ has to generate a master public key BE-MPK containing n public key keys $pk_1, \ldots, pk_n$. To this end, $\mathcal{B}$ picks two distinct indices $i_0^\star, i_1^\star \xleftarrow{\$} \{1, \ldots, n\}$ and defines $pk_{i_0^\star} = pk_0^\star$ and $pk_{i_1^\star} = pk_1^\star$. Then, $\mathcal{B}$ runs Keygen and generates $n-2$ key pairs (sk_i, pk_i) for each $i \in \{1, \ldots, n\} \backslash \{i_0^\star, i_1^\star\}$. The master public key BE-MPK $= \big(par, \Sigma, \{pk_i\}_{i=1}^n\big)$ is provided as input to $\mathcal{A}$.

Throughout the game, $\mathcal{A}$ can adaptively corrupt any user $i \in \{1, \ldots, n\}$. At each corruption query, $\mathcal{B}$ aborts if the queried index i falls in $\{i_0^\star, i_1^\star\}$. Otherwise, $\mathcal{B}$ necessarily knows the queried secret key sk_i and hands it to $\mathcal{A}$. For each decryption query $(C = (\mathsf{VK}, C_1, \ldots, C_\ell, \sigma), i)$ made by $\mathcal{A}$, $\mathcal{B}$ can handle the query on its own whenever $i \notin \{i_0^\star, i_1^\star\}$. If $i = i_0^\star$ (resp. $i = i_1^\star$), $\mathcal{B}$ queries its own decryption oracle up to ℓ times and successively asks for the decryption of $C_1, \ldots, C_\ell$ under $sk_0^\star$ (resp. $sk_1^\star$). At the first answer that differs from $\perp$ and can be parsed as $M||\mathsf{VK}$, for some M of the right length, $\mathcal{B}$ returns M. If $\mathcal{B}$ fails to obtain a decryption result of the form $M||\mathsf{VK}$, for some M, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, meaning that C does not properly decrypt under $sk_0^\star$ (resp. $sk_1^\star$).

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size ℓ. These sets are re-ordered as $S_0' = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$ and $S_1' = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ where $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$. If $\theta_k \neq i_0^\star$ or $\rho_k \neq i_1^\star$, $\mathcal{B}$ aborts. We denote by Good the event $(\theta_k = i_0^\star) \wedge (\rho_k = i_1^\star)$, which implies $pk_{\theta_k} = pk_0^\star$ and $\tilde{pk}_{\rho_k} = \tilde{pk}_1^\star$.

If Good occurs, $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and sends the messages $(M_0||\mathsf{VK}^\star)$, $(M_1||\mathsf{VK}^\star)$ to its AI-CCA challenger. The latter returns a challenge ciphertext $C^\star = \mathsf{Encrypt}(par, pk_b, M_b||\mathsf{VK}^\star)$, for some internally flipped random bit $b \stackrel{\$}{\leftarrow} \{0,1\}$. The ANOBE adversary's challenge ciphertext is then obtained as follows.

1. For $j = 1$ to $k-1$, $\mathcal{B}$ sets $C_j = \mathsf{Encrypt}\big(par, pk_{\rho_j}, (M_1||\mathsf{VK}^\star)\big)$.
2. For $j = k+1$ to ℓ, $\mathcal{B}$ computes $C_j = \mathsf{Encrypt}\big(par, pk_{\theta_j}, (M_0||\mathsf{VK}^\star)\big)$.
3. Finally, set $C_k = C^\star$.

The adversary $\mathcal{A}$ receives $C = (\mathsf{VK}^\star, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}^\star, (C_{\tau(1)}, \ldots, C_{\tau(\ell)}))$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

In the second phase, $\mathcal{A}$ makes further adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ and $\mathcal{B}$ handles them as previously. Decryption queries are handled as in the first phase with one difference: if $\mathcal{A}$ makes a decryption query $(C = (\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma), i)$ for which we simultaneously have $i \in \{i_0, i_1\}$, $\mathsf{VK} \neq \mathsf{VK}^\star$ and $C_j = C^\star$ for some $j \in \{1, \ldots, \ell\}$, $\mathcal{B}$ considers that C_j decrypts to $\perp$ under sk_i without invoking its own decryption oracles on C_j. Since $\mathsf{VK} \neq \mathsf{VK}^\star$, it is clear that $C^\star$ cannot correctly decrypt to a message ending with VK under the private key $sk_b^\star$. Still, we have to rule out the possibility to have $\mathsf{Decrypt}(sk_{1-b}^\star, C^\star) = M||\mathsf{VK}$, for some plaintext M, since this could render $\mathcal{A}$'s view inconsistent. If this event were to happen with non-negligible probability, algorithm $\mathcal{B}$ could be turned into a weak robustness (more precisely, WROB-CCA) adversary $\mathcal{B}'$. The latter would simply generate the ANOBE challenge ciphertext by computing $C_1, \ldots, C_\ell$ itself and waiting for $\mathcal{A}$ to make a decryption query $C = (\mathsf{VK}, C_1, \ldots, C_\ell, \sigma)$ for which there exists $j \in \{1, \ldots, \ell\}$ such that C_j correctly decrypts under both sk_b and sk_{1-b}.

When $\mathcal{A}$ halts, it outputs a result $b' \in \{0,1\}$ and $\mathcal{B}$ outputs b' as well. If $\mathcal{B}$ did not abort, its AI-CCA advantage is as large as the gap between $\mathcal{A}$'s probabilities of outputting 0 in Game k and Game $k-1$. Indeed, if $\mathcal{B}$'s AI-CCA challenger sets its challenge bit as $b = 0$, $\mathcal{B}$ is playing Game $k-1$ with $\mathcal{A}$ whereas, if the AI-CCA challenger sets $b = 1$, $\mathcal{B}$ is playing Game k.

Now, let us assess $\mathcal{B}$'s probability not to abort. Recall that the adversary $\mathcal{A}$ cannot legally corrupt any user in $S_0 \triangle S_1 = \{\theta_{\phi+1}, \ldots, \theta_\ell, \rho_{\phi+1}, \ldots, \rho_\ell\}$. For this reason, a sufficient condition for $\mathcal{A}$ not to query the private keys sk_{θ_k} or sk_{ρ_k} is to have Good occurring. Since event Good comes about with probability $\Pr[\mathsf{Good}] = 1/n(n-1) > 1/n^2$, the claimed result follows. □

C Proof of Lemma 1

We give a proof of Lemma 1.

Proof. The proof proceeds with a sequence of games where the first game is the real game and the last one is a game where the challenger's bit $b \in \{0,1\}$ is unconditionally hidden. In Game i, we call S_i the event that $b' = b$.

Game 0: is the real attack game. The adversary begins by choosing a tag $t^\star$ and obtains two public keys $pk_0 = (X_{0,1}, X_{0,2}, Y_{0,1}, Y_{0,2})$, $pk_1 = (X_{1,1}, X_{1,2}, Y_{1,1}, Y_{1,2})$ from the challenger that keeps the private keys $sk_0 = (x_{0,1}, x_{0,2}, y_{0,1}, y_{0,2})$, $sk_1 = (x_{1,1}, x_{1,2}, y_{1,1}, y_{1,2})$ to itself. The adversary $\mathcal{A}$ then makes verification queries for inputs $(U, H = (V, W), t)$ such that $t \neq t^\star$. At each query, the challenger replies by outputting two bits $(d_0, d_1) \in \{0,1\}^2$ where $d_0 = \big(H = \mathtt{Invert}(\mathsf{cp}, sk_0, t, U)\big)$ and $d_1 = \big(H = \mathtt{Invert}(\mathsf{cp}, sk_1, t, U)\big)$. In the challenge phase, the challenger flips a fair binary coin $b \stackrel{\$}{\leftarrow} \{0,1\}$ and generates the challenge as $\big(U^\star, (V^\star, W^\star)\big) = \big(g^r, ((X_{b,1}^t X_{b,2})^r, (Y_{b,1}^t Y_{b,2})^r)\big)$, for some $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, which is sent to the adversary $\mathcal{A}$. After a second series of queries, $\mathcal{A}$ outputs a bit

$b' \in \{0,1\}$ and we call S_0 the event that $b' = b$.

Game 1: is identical to Game 0 with the following two differences.

- The challenger's bit $b \xleftarrow{\$} \{0,1\}$ is chosen at the beginning of the game.
- In the adversary's challenge $(U^\star, V^\star, W^\star)$, $W^\star$ is replaced by a random element of $\mathbb{G}$.

The first change is purely conceptual and we argue that, under the DDH assumption, a computationally bounded adversary cannot notice the second one.

To prove this, we show a DDH distinguisher $\mathcal{B}$ that bridges between Game 0 and Game 1. Algorithm $\mathcal{B}$ takes as input a tuple $(g, X = g^x, Y = g^y, T)$, where $x, y \in_R \mathbb{Z}_p^*$, and aims at deciding whether $T = g^{xy}$ or $T \in_R \mathbb{G}^*$. At the outset of the game, $\mathcal{B}$ picks $\theta_1, \theta_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and defines $\tilde{X} = g^{\theta_1} X^{\theta_2}$. When the challenge bit $b \xleftarrow{\$} \{0,1\}$ is chosen, $\mathcal{B}$ honestly generates pk_{1-b} by choosing $x_{1-b,1}, x_{1-b,2}, y_{1-b,1}, y_{1-b,2} \xleftarrow{\$} \mathbb{Z}_p^*$ and setting $X_{1-b,1} = g^{x_{1-b,1}}$, $X_{1-b,2} = g^{x_{1-b,2}}$, $Y_{1-b,1} = g^{y_{1-b,1}}$ and $Y_{1-b,2} = g^{y_{1-b,2}}$. As for pk_b, $\mathcal{B}$ chooses $\alpha, \beta_1, \beta_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $X_{b,1} = \tilde{X}$, $X_{b,2} = \tilde{X}^{-t^\star} g^{\beta_1}$, $Y_{b,1} = g^{\beta_2} X^{\alpha}$ and $Y_{b,2} = g^{-\beta_2 t^\star}$. The adversary is given the two public keys $(X_{0,1}, X_{0,1}, Y_{0,1}, Y_{0,2})$ and $(X_{1,1}, X_{1,2}, Y_{1,1}, Y_{1,2})$ and we note that they are both uniformly distributed in $\mathbb{G}^4$ as required.

When the adversary $\mathcal{A}$ makes a verification query $(U, (V, W), t)$, with $t \neq t^\star$, $\mathcal{B}$ can simply run algorithm $\texttt{Invert}(\texttt{cp}, sk_{1-b}, t, U)$ since it knows sk_{1-b}. When it comes to simulate the evaluation of the bit $d_b = \big((V,W) = \texttt{Invert}(\texttt{cp}, sk_b, t, U)\big)$, it computes

$$Z_1 = (V/U^{\beta_1})^{1/(t-t^\star)}, \qquad Z_2 = (W/U^{\beta_2(t-t^\star)})^{1/\alpha\cdot t}$$

and answers that $d_b = 1$ (*i.e.*, that $(V,W) = \texttt{Invert}(\texttt{cp}, sk_b, t, U) = 1$) if and only if $Z_1 = U^{\theta_1} \cdot Z_2^{\theta_2}$. To see why this test works, we note that, if (U, V, W) is a valid hint for pk_b and t, it must be the case that $(Z_1, Z_2) = (\tilde{X}^r, X^r)$, where $r = \log_g(U)$, so that the test is satisfied. If $\big(U, (V, W)\big)$ is not a valid value-hint pair w.r.t. (pk_b, t), we must have $(U, V, W) = (g^r, (X_{b,1}^t X_{b,2})^{r+r'}, (Y_{b,1}^t Y_{b,2})^{r+r''})$, where it holds that either $r' \neq 0$ or $r'' \neq 0$, and this implies that $Z_1 = \tilde{X}^{r+r_1}$ and $Z_2 = X^{r+r_2}$ where either $r_1 \neq 0$ or $r_2 \neq 0$. If $r_2 = 0$ and $r_1 \neq 0$, the equality $Z_1 = U^{\theta_1} Z_2^{\theta_2}$ is never satisfied and we thus assume $r_2 \neq 0$. In this case, we can only have $Z_1 = U^{\theta_1} Z_2^{\theta_2}$ by pure chance since, due to the dependence on $\theta_2 \in \mathbb{Z}_p^*$, the value $U^{\theta_1} Z_2^{\theta_2} = \tilde{X}^r \cdot X^{r_2 \cdot \theta_2}$ is independent of $\mathcal{A}$'s view since it is the product of an information-theoretically fixed term $\tilde{X}^r$ with a completely undetermined value $X^{r_2 \cdot \theta_2}$. The same arguments as in [15] show that $\mathcal{B}$'s probability to incorrectly answer a verification query is at most q/p if q is the number of queries.

In the challenge phase, $\mathcal{B}$ constructs the challenge value-hint pair $(U^\star, (V^\star, W^\star))$ as

$$U^\star = Y, \qquad V^\star = Y^{\beta_1}, \qquad W^\star = T^{\alpha \cdot t^\star}$$

It is easy to see that, if $T = g^{xy}$, $\mathcal{A}$'s view is the same as in Game 0 (except with probability q/p) whereas, if $T \in_R \mathbb{G}^*$, $\mathcal{B}$ is playing Game 1 with $\mathcal{A}$ since $W^\star$ is uniformly distributed in $\mathbb{G}^*$. Combining this observation with the above arguments, we find $|\Pr[S_1] - \Pr[S_0]| \leq \mathbf{Adv}^{\mathrm{DDH}}(\mathcal{B}) + q/p$.

Game 2: is identical to Game 1 but, in the challenge phase $V^\star$ and $W^\star$ are both chosen uniformly in $\mathbb{G}^\star$ and independently of $U^\star$. To argue that $\mathcal{A}$ cannot see the difference as long as the DDH assumption holds, we proceed as in the previous transition.

Namely, the DDH distinguisher $\mathcal{B}$ takes as input $(g, X = g^x, Y = g^y, T)$, where $x, y \in_R \mathbb{Z}_p^*$, and aims to decide if $T = g^{xy}$. At the beginning of the game, $\mathcal{B}$ picks $\theta_1, \theta_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and defines $\tilde{X} = g^{\theta_1} X^{\theta_2}$. When $b \stackrel{\$}{\leftarrow} \{0,1\}$ is chosen, $\mathcal{B}$ honestly generates pk_{1-b} as in the transition from Game 0 to Game 1. Then, it computes pk_b by picking $\alpha, \beta_1, \beta_2, \beta_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $X_{b,1} = g^{\beta_1} X^{\alpha}$, $X_{b,2} = g^{-\beta_1 t^\star}$, $Y_{b,1} = g^{\beta_2} \tilde{X}^{\beta_3}$ and $Y_{b,2} = \tilde{X}$. The adversary is given $pk_0 = (X_{0,1}, X_{0,1}, Y_{0,1}, Y_{0,2})$ and $pk_1 = (X_{1,1}, X_{1,2}, Y_{1,1}, Y_{1,2})$ which are both uniformly distributed in $\mathbb{G}^4$ as required.

Whenever the adversary $\mathcal{A}$ sends a verification query $\big(U, (V, W), t\big)$, $\mathcal{B}$ aborts in the unlikely event that $\beta_3 \cdot t = p - 1$ (since β_3 is chosen at random independently of $\mathcal{A}$'s view, this happens with probability at most q/p during the game). Otherwise, it can compute

$$Z_1 = (V/U^{\beta_1(t-t^\star)})^{1/\alpha \cdot t}, \qquad Z_2 = (W/U^{\beta_2 \cdot t})^{1/(\beta_3 \cdot t+1)}$$

If $Z_1 = U^{\theta_1} \cdot Z_2^{\theta_2}$, then $\mathcal{B}$ replies that $d_b = 1$ (meaning that $(V, W) = \texttt{Invert}(\texttt{cp}, sk_b, t, U) = 1$). Otherwise, the second output bit d_b of the verification algorithm is declared to be 0. In addition, $\mathcal{B}$ can run $\texttt{Invert}(\texttt{cp}, sk_{1-b}, t, U)$ normally since it knows sk_{1-b}. The above test is easily seen to work (with overwhelming probability) for the same reasons as in the transition from Game 0 to Game 1. The only situation where $\mathcal{B}$ fails to answer verification queries in the same way as in Game 2 is when $\beta_3 \cdot t = p - 1$ at some verification query. When taking into account the tiny probability of this event, we find that $|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}^{\mathrm{DDH}}(\mathcal{B}) + q/p$.

When it comes to construct the challenge $\big(U^\star, (V^\star, W^\star)\big)$ for the adversary, $\mathcal{B}$ chooses $W^\star \stackrel{\$}{\leftarrow} \mathbb{G}^*$ at random and generates $(U^\star, V^\star)$ as

$$U^\star = Y, \qquad V^\star = T^{\alpha \cdot t^\star}.$$

It is easy to see that, if $T = g^{xy}$, the challenger $\mathcal{B}$ is playing Game 1 with the adversary. If $T \in_R \mathbb{G}^\star$, $\mathcal{A}$ and $\mathcal{B}$ are playing Game 2.

In Game 2, the challenge $\big(U^\star, (V^\star, W^\star)\big)$ is just a sequence of three independent random group elements that carries no information about the bit $b \in \{0,1\}$. Hence, we have $\Pr[S_2] = 1/2$.

By combining the above arguments, we obtain

$$\mathbf{Adv}^{\text{anon-hint}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\mathrm{DDH}}(\mathcal{B}) + 2 \cdot \frac{q}{p} < 2 \cdot \Big(\mathbf{Adv}^{\mathrm{DDH}}(\mathcal{B}) + \frac{q}{2^\lambda}\Big). \tag{C.1}$$

□

D Proof of Lemma 2

We give a proof of Lemma 2.

Proof. Assuming the existence of a strong robustness adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$ that receives as input $(\mathbb{G}, p, g, X)$ and computes $x = \log_g(X) \in \mathbb{Z}_p$ with overwhelming probability.

To generate the public keys pk_0 and pk_1, $\mathcal{B}$ begins by defining $\tilde{X} = g^{\theta_1} X^{\theta_2}$ for randomly chosen $\theta_1, \theta_2 \in \mathbb{Z}_p^*$. Then, $\mathcal{B}$ picks $\alpha_{i,1}, \alpha_{i,2}, \beta_{i,1}, \beta_{i,2}, \gamma_{i,1}, \gamma_{i,2}, \delta_{i,1}, \delta_{i,2} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, for $i \in \{0,1\}$, and computes

$$X_{i,1} = g^{\alpha_{i,1}} X^{\beta_{i,1}} \qquad X_{i,2} = g^{\alpha_{i,2}} X^{\beta_{i,2}}$$
$$Y_{i,1} = g^{\gamma_{i,1}} \tilde{X}^{\delta_{i,1}} \qquad Y_{i,2} = g^{\gamma_{i,2}} \tilde{X}^{\delta_{i,2}}.$$

The adversary is given $pk_0 = (X_{0,1}, X_{0,2}, Y_{0,1}, Y_{0,2})$ and $pk_1 = (X_{1,1}, X_{1,2}, Y_{1,1}, Y_{1,2})$ and starts making verification queries.

At each verification query $\big(U,(V,W),t\big)$, $\mathcal{B}$ aborts if $\beta_{i,1}\cdot t+\beta_{i,2}=0$ or $\delta_{i,1}\cdot t+\delta_{i,2}=0$ for some $i\in\{0,1\}$ (since $\mathcal{A}$ has no information on $(\beta_{i,1},\beta_{i,2},\delta_{i,1},\delta_{i,2})$, this only happens with probability q/p throughout the game). Otherwise, $\mathcal{B}$ replies by computing

$$Z_{i,1}=(V/U^{\alpha_{i,1}t+\alpha_{i,2}})^{1/(\beta_{i,1}\cdot t+\beta_{i,2})} \qquad \text{and} \qquad Z_{i,2}=(W/U^{\gamma_{i,1}t+\gamma_{i,2}})^{1/(\delta_{i,1}\cdot t+\delta_{i,2})}$$

for each $i\in\{0,1\}$. If it turns out that $Z_{i,2}=U^{\theta_1}Z_{i,1}^{\theta_2}$ for some $i\in\{0,1\}$, the reduction $\mathcal{B}$ replies that $(V,W)=\texttt{Invert}(\texttt{cp},sk_i,t,U)=1$. We note that, if $\big(U,(V,W)\big)$ is valid for pk_i, we must have $Z_{1,i}=X^r$ and $Z_{i,2}=\tilde{X}^r$, where $r=\log_g(U)$, for the same reason as in the proof of lemma 1. The same arguments show that $\mathcal{B}$'s probability to incorrectly answer a verification query is at most q/p.

We note that the adversary is successful if some verification query $\big(U^\star,(V^\star,W^\star),t^\star\big)$ results in the output $(1,1)$ (namely, $U^\star$ is inverted to $(V^\star,W^\star)$ under both private keys sk_0 and sk_1). The same arguments as in the proof of lemma 1 guarantee that, except with probability $2/p$, the equalities

$$U^\star=g^r, \qquad V^\star=(X_{0,1}^{t^\star}X_{0,2})^r=(X_{1,1}^{t^\star}X_{1,2})^r, \qquad W^\star=(Y_{0,1}^{t^\star}Y_{0,2})^r=(Y_{1,1}^{t^\star}Y_{1,2})^r,$$

hold for some $r\in\mathbb{Z}_p^*$. Since $r\neq 0$, this implies $X_{0,1}^{t^\star}X_{0,2}=X_{1,1}^{t^\star}X_{1,2}$ and $Y_{0,1}^{t^\star}Y_{0,2}=Y_{1,1}^{t^\star}Y_{1,2}$. In particular, we thus have $g^{\alpha_{0,1}t^\star+\alpha_{0,2}}X^{\beta_{0,1}t^\star+\beta_{0,2}}=g^{\alpha_{1,1}t^\star+\alpha_{1,2}}X^{\beta_{1,1}t^\star+\beta_{1,2}}$ and thus

$$g^{t^\star\cdot(\alpha_{0,1}-\alpha_{1,1})+(\alpha_{0,2}-\alpha_{1,2})}=X^{t^\star\cdot(\beta_{1,1}-\beta_{0,1})+(\beta_{1,2}-\beta_{0,2})} \tag{D.1}$$

The probability to have $t^\star=(\alpha_{1,2}-\alpha_{0,2})/(\alpha_{0,1}-\alpha_{1,1})=(\beta_{0,2}-\beta_{1,2})/(\beta_{1,1}-\beta_{0,1})$ is negligible since $\{(\alpha_{i,1},\alpha_{i,2})\}_{i=0,1}$ (as well as $\{(\beta_{i,1},\beta_{i,2})\}_{i=0,1}$) are uniformly chosen in $(\mathbb{Z}_p^*)^2$ and independent of $\mathcal{A}$'s view. It comes that the equality (D.1) allows $\mathcal{B}$ to compute

$$x=\log_g(X)=\frac{t^\star\cdot(\alpha_{0,1}-\alpha_{1,1})+(\alpha_{0,2}-\alpha_{1,2})}{t^\star\cdot(\beta_{1,1}-\beta_{0,1})+(\beta_{1,2}-\beta_{0,2})}.$$

□

E Proof of Theorem 4

We give a proof for Theorem 4.

Proof. Recall that, since $|S_0|=|S_1|=\ell$, it always holds that $|S_0\backslash S_1|=|S_1\backslash S_0|=|S_b|-|S_0\cap S_1|$ for each $b\in\{0,1\}$. We consider a sequence of games where the adversary is given an encryption of M_0 for S_0 in Game 0 while, in the last game, the adversary obtains an encryption of M_1 under S_1.

Game 0_{real}: corresponds to the real game when the challenger's bit is $b=0$. Namely, the adversary $\mathcal{A}$ is given public parameters BE-MPK containing n tuples $\{(\tilde{pk}_i,pk_i^h\}_{i=1}^n$. For each $i\in\{1,\dots,n\}$, user i's private key is a pair $sk_i=(\tilde{sk}_i,sk_i^h)$. In the first phase, the adversary $\mathcal{A}$ adaptively chooses indices $i\in\{1,\dots,n\}$ and obtains the corresponding sk_i. The adversary may also invoke the decryption oracle by making queries (C,i) which are handled using the relevant private key sk_i. At the challenge phase $\mathcal{A}$ chooses messages M_0,M_1 and two subsets $S_0,S_1\subset\{1,\dots,n\}$ of size $|S_0|=|S_1|=\ell$ such that $S_0\neq S_1$. The challenger generates a one-time signature key pair $(\mathsf{SK}^\star,\mathsf{VK}^\star)\leftarrow\mathcal{G}(\lambda)$, picks a random exponent $r\xleftarrow{\$}\mathcal{R}^h$, parses S_0 as $\{\theta_1,\dots,\theta_\ell\}$ and returns the challenge ciphertext $C^\star=\big(\mathsf{VK}^\star,U,(H_{\tau(1)},C_{\tau(1)}),\dots,(H_{\tau(\ell)},C_{\tau(\ell)}),\sigma\big)$ where

$(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\theta_j}, r)$, $C_j = \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)\big)$ for $j = 1$ to ℓ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation. In phase 2, $\mathcal{A}$ makes further decryption queries (with the usual restriction) and key queries for arbitrary indices i such that $i \in \{1, \ldots, n\} \backslash (S_0 \triangle S_1)$. Eventually, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$ and we define E_0^{real} to be the event that $b' = 0$.

Game 0**:** is as Game 0_{real} but the challenger now rejects all post challenge decryption queries (C, i) where C contains the same verification key $\mathsf{VK}^\star$ as in the challenge phase. We call E_0 the event that $\mathcal{A}$ outputs $b' = 0$ in Game 0.

Game $0'$**:** is defined to be identical to Game 0 for convenience.

Game k $(1 \leq k \leq \ell)$**:** is identical to **Game** $k - 1$' but, in the challenge phase is processed differently. From the two adversarially-chosen sets $S_0, S_1 \subset \{1, \ldots, n\}$, the challenger $\mathcal{B}$ defines the value $\phi = |S_0 \cap S_1|$ and then considers two ordered sets $S'_0 = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$, $S'_1 = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ that are obtained by ordering S_0 and S_1 in such a way that $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$ and $\theta_j \neq \rho_j$ if $j \in \{\phi + 1, \ldots, \ell\}$. Then, $\mathcal{B}$ picks $r \xleftarrow{\$} \mathcal{R}^h$ and constructs the challenge ciphertext as follows.

1. For $j = 1$ to ϕ,
   a. Compute $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\theta_j}, r)$.
   b. Set $C_j = \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_1||\mathsf{VK}^\star)$ if $j \leq k$ and $C_j = \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)$ if $j > k$.
2. For $j = \phi + 1$ to ℓ,
   - If $j < k$, set $(H_j, C_j) = \big(H_j, \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\rho_j}, M_1||\mathsf{VK}^\star)\big)$, where H_j is obtained by computing $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\rho_j}, r)$.
   - If $j > k$, compute a value-hint pair $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\theta_j}, r)$ and set
   $$(H_j, C_j) = \Big(H_j, \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)\Big).$$
   - If $j = k$, compute $(U, H_k) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\rho_k}, r)$ and set
   $$(H_k, C_k) = \big(H_k, \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_k}, M_0||\mathsf{VK}^\star)\big).$$

The adversary is then returned $C^\star = \big(\mathsf{VK}^\star, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma\big)$, for a randomly chosen permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$, and the second phase is handled as in previous games. We call E_k the event of $\mathcal{A}$ outputting $b' = 0$ at the end of Game k.

Game k**'** $(1 \leq k \leq \ell)$**:** is identical to Game k but the challenge ciphertext $C^\star$ is now generated as follows.

1. For $j = 1$ to ϕ, compute (H_j, C_j) as in Game k
2. For $j = \phi + 1$ to ℓ, construct (H_j, C_j) by computing $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\rho_j}, r)$ and setting
$$(H_j, C_j) = \big(H_j, \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\rho_j}, M_1||\mathsf{VK}^\star)\big)$$
if $j \leq k$ (namely, C_j is an encryption of M_1 under $\tilde{pk}_{\rho_j}$ instead of M_0 under $\tilde{pk}_{\theta_j}$). If $j > k$, the pair (H_j, C_j) is computed as $(H_j, C_j) = \big(H_j, \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)\big)$ where H_j is obtained as part of $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk^h_{\theta_j}, r)$

We denote by E'_k the event that the adversary outputs $b' = 0$ in Game k'.

Game ℓ_{real}: is identical to Game ℓ' with the difference that, when handling decryption queries, the challenger no longer rejects ciphertexts that contain the verification key $\mathsf{VK}^\star$. Game ℓ_{real} actually proceeds like the real game when the challenger's bit is $b = 1$. We let E_ℓ^{real} be the event that $\mathcal{A}$ outputs the bit $b' = 0$ at the end of Game ℓ_{real}.

Game 0_{real} and Game 0 are clearly indistinguishable if the one-time signature is strongly unforgeable and the same argument can be made about Game ℓ and Game ℓ_{real}.

We thus have $|\Pr[E_0^{real}] - \Pr[E_0]| = |\Pr[E_\ell^{real}] - \Pr[E_\ell]| \leq \mathbf{Adv}^{\text{suf}}_{\text{OTS}}(\mathcal{A})$. As for other game transitions, they are justified by lemmas 6, 7 and 8 that separately consider the situations where $k \leq \phi$ and $k > \phi$. Specifically, we show that, if Game k and Game $k-1$' can be distinguished, there exists an anonymity adversary $\mathcal{B}_1$ against the hint system. Likewise, if Game k and Game k' can be told apart, lemmas 7 and 8 show that there exists either a AI-CCA adversary $\mathcal{B}_2$ or a WROB-CCA adversary $\mathcal{B}_2$ against the encryption scheme. By combining these results altogether, we find

$$\begin{aligned}
|\Pr[E_0^{real}] - \Pr[E_\ell^{real}]| &\leq 2 \cdot \mathbf{Adv}^{\text{suf}}_{\text{OTS}}(\mathcal{A}) + n^2 \cdot \ell \cdot \mathbf{Adv}^{\text{anon-hint}}(\mathcal{B}_1) \\
&\qquad + n^2 \cdot \ell \cdot \Big(\mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}_3)\Big) \\
&\leq 2 \cdot \mathbf{Adv}^{\text{suf}}_{\text{OTS}}(\mathcal{A}) \\
&\qquad + n^3 \cdot \Big(\mathbf{Adv}^{\text{anon-hint}}(\mathcal{B}_1) + \mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}_3)\Big).
\end{aligned}$$

□

Lemma 6. *For each $k \in \{1, \ldots, \ell\}$, Game k is indistinguishable from Game $k-1$' if the tag-based hint scheme is anonymous. More precisely, we have*

$$|\Pr[E_k] - \Pr[E'_{k-1}]| \leq n^2 \cdot \mathbf{Adv}^{\text{anon-hint}}(\mathcal{B}). \tag{E.1}$$

Proof. We show that, if an adversary $\mathcal{A}$ can distinguish Game k and Game $k-1$', we can construct an anonymity adversary $\mathcal{B}$ against the tag-based anonymous hint scheme. We note that Game k and Game $k-1$' are identical when $k \leq \phi = |S_0 \cap S_1|$ and we thus assume $k > \phi$.

On input of common public parameters cp for the hint scheme, our adversary $\mathcal{B}$ initially generates a one-time key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and sends $\mathsf{VK}^\star$ as a target tag to its challenger. The latter replies by sending public parameters $(\mathbb{G}, g)$ as well as two distinct tag-based hint public keys pk_0^h, pk_1^h. Then, $\mathcal{B}$ has to prepare a master public key BE-MPK for the ANOBE adversary $\mathcal{A}$. To do this, $\mathcal{B}$ randomly picks two distinct indices $i_0, i_1 \stackrel{\$}{\leftarrow} \{1, \ldots, n\}$ and sets $pk_{i_0}^h = pk_0^h$ and $pk_{i_1}^h = pk_1^h$. Then, our adversary $\mathcal{B}$ generates a set of n key pairs $(\tilde{sk}_i, \tilde{pk}_i) \leftarrow \pi^{\text{pke}}.\mathsf{Keygen}(1^\lambda)$ for the underlying public key encryption scheme. It then defines $pk_{i_0} = (\tilde{pk}_{i_0}, pk_{i_0}^h)$ and $pk_{i_1} = (\tilde{pk}_{i_1}, pk_{i_1}^h)$. For all indices $i \in \{1, \ldots, n\} \backslash \{i_0, i_1\}$, $\mathcal{B}$ generates $(sk_i^h, pk_i^h) \leftarrow \pi^{\text{hint}}.\mathsf{Keygen}(\mathsf{cp})$ itself and defines $pk_i = (\tilde{pk}_i, pk_i^h)$. It finally hands $\mathcal{A}$ the master public key consisting of BE-MPK $= \big(\mathsf{cp}, \{pk_1, \ldots, pk_n\}, \varSigma\big)$, where $pk_i = (\tilde{pk}_i, pk_i^h)$.

Throughout the game, $\mathcal{A}$ is allowed to corrupt any user $i \in \{1, \ldots, n\}$ depending on the previously collected information. At each corruption query, $\mathcal{B}$ aborts if the queried index i unluckily falls into the forbidden set $\{i_0, i_1\}$. Otherwise, $\mathcal{B}$ necessarily knows the underlying secret key $sk_i = (\tilde{sk}_i, sk_i^h)$ and returns it to the ANOBE adversary $\mathcal{A}$.

When the adversary $\mathcal{A}$ makes a decryption query, $\big(C = (\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma), i\big)$, for some index $i \in \{1, \ldots, n\}$, $\mathcal{B}$ returns $\perp$ if $\mathsf{VK} = \mathsf{VK}^\star$. Otherwise, we note that $\mathcal{B}$ can always answer the query on its own whenever $i \notin \{i_0, i_1\}$. If $i \in \{i_0, i_1\}$, $\mathcal{B}$ invokes its tag-based anonymity challenger[5] up to ℓ times: for $j = 1$ to ℓ, it considers H_j and makes verification queries (U, H_j, VK) until the challenger indicates that $H_j = \mathtt{Invert}(\mathsf{cp}, sk_\beta, \mathsf{VK}, U) = 1$ for some $j \in \{1, \ldots, \ell\}$ and some $\beta \in \{0, 1\}$. When the smallest such j is found, $\mathcal{B}$ uses the corresponding private key $\tilde{sk}_{i_\beta}$ to decrypt C_j and sends the result to $\mathcal{A}$. If the challenger always replies 0 at each verification query, $\mathcal{B}$ simply returns $\perp$ to $\mathcal{A}$, meaning that user i was not a legitimate receiver of C.

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size. At this point, $\mathcal{B}$ re-orders S_0, S_1 as $S'_0 = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$, $S'_1 = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ where $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$. If $\theta_k \neq i_0$ or $\rho_k \neq i_1$, the simulator $\mathcal{B}$ aborts. We call Good the event $(\theta_k = i_0) \wedge (\rho_k = i_1)$, which implies $pk_{\theta_k} = (\tilde{pk}_{\theta_k}, pk_0^h)$ and $pk_{\rho_k} = (\tilde{pk}_{\rho_k}, pk_1^h)$.

If Good occurs, $\mathcal{B}$ sends its challenge request to its own challenger and receives in response a challenge hint $(U^\star, H^\star) = \pi^{\text{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_b^h, r^\star)$, for some random bit $b \xleftarrow{\$} \{0, 1\}$ and some $r^\star \xleftarrow{\$} \mathcal{R}^h$. The ANOBE challenge ciphertext is produced according
to the following steps.

1. For $j = 1$ to ϕ, $\mathcal{B}$ sets $(H_j, C_j) = \big(\pi^{\text{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_{\theta_j}^h, \mathsf{VK}^\star, U^\star),\ \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_1 || \mathsf{VK}^\star)\big)$ using $sk_{\theta_j}^h$, which is available if Good occurs occurs.
2. For $j = \phi + 1$ to $k - 1$, $\mathcal{B}$ can build (H_j, C_j) as
$$(H_j, C_j) = \big(\pi^{\text{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_{\rho_j}^h, \mathsf{VK}^\star, U^\star),\ \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\rho_j}, M_1 || \mathsf{VK}^\star)\big)$$
in the same way since it knows $sk_{\rho_j}^h$ in any occurrence of Good.
3. For $j = k+1$ to ℓ, $\mathcal{B}$ sets $(H_j, C_j) = \big(\pi^{\text{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_{\theta_j}^h, \mathsf{VK}^\star, U^\star), \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0 || \mathsf{VK}^\star)\big)$, which is possible since $sk_{\theta_j}^h$ is available for the same reason as in previous cases.
4. Finally, set $(H_k, C_k) = \big(H^\star, \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_k}, M_0 || \mathsf{VK}^\star)\big)$.

The adversary $\mathcal{A}$ then receives the ciphertext $C = \big(\mathsf{VK}^\star, U^\star, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma\big)$, for some random permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ and where σ is a one-time signature on the whole bundle.

In the second phase, $\mathcal{A}$ makes further adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ and $\mathcal{B}$ handles them as in the first phase. Also, $\mathcal{B}$ can still consistently answer decryption queries by appealing to its tag-based hint challenger and since it knows $\{\tilde{sk}_i\}_{i=1}^n$.

The game ends with $\mathcal{A}$ outputting a bit $b' \in \{0, 1\}$ which is also $\mathcal{B}$'s result. If $\mathcal{B}$ did not abort, its anonymity advantage against the tag-based hint system is as large as the gap between $\mathcal{A}$'s probabilities of outputting 1 in Game k and Game $k - 1$'. Indeed, if $\mathcal{B}$'s challenger chooses $b = 0$, then $\mathcal{B}$ is playing Game $k - 1$' with $\mathcal{A}$ whereas, if $b = 1$, $\mathcal{B}$ is playing Game k.

When analyzing $\mathcal{B}$'s probability not to abort, we first recall that $\mathcal{A}$'s corruption queries cannot involve any index i in $S_0 \triangle S_1 = \{pk_{\theta_{\phi+1}}, \ldots, pk_{\theta_\ell}, pk_{\rho_{\phi+1}}, \ldots, pk_{\rho_\ell}\}$. Therefore, a sufficient condition for $\mathcal{B}$ not to be faced with an embarrassing query is to have Good occurring. The probability of this desirable event is $\Pr[\mathsf{Good}] = 1/n(n-1) > 1/n^2$. The announced bound (E.1) then follows from the bound given by (C.1). $\square$

[5] This is necessary to make sure that decryption queries are handled as in the real scheme.

Lemma 7. *For each $k \in \{1, \ldots, \phi\}$, Game k' is indistinguishable from Game k if the underlying encryption scheme is IND-CCA. More precisely, we have*

$$|\Pr[E'_k] - \Pr[E_k]| \leq n \cdot \mathbf{Adv}^{\text{ind-cca}}(\mathcal{B}).$$

Proof. We prove that, if the adversary $\mathcal{A}$ can distinguish Game k and Game k', there is a chosen-ciphertext adversary against the underlying public key encryption scheme. We note that, for each $k \in \{1, \ldots, \phi\}$, Game k and Game k' are identical when $M_0 = M_1$ and we thus assume $M_0 \neq M_1$. This implies that the adversary cannot corrupt users in $S_0 \cap S_1$.

Our IND-CCA adversary $\mathcal{B}$ receives a public key $\tilde{pk}^\star$ from its challenger and it has to prepare a master public key BE-MPK comprising n public keys $\tilde{pk}_1, \ldots, \tilde{pk}_n$ for the ANOBE adversary $\mathcal{A}$. To this end, $\mathcal{B}$ picks $i^\star \xleftarrow{\$} \{1, \ldots, n\}$ at random and defines $\tilde{pk}_{i^\star} = \tilde{pk}^\star$. Then, $\mathcal{B}$ generates $n-1$ public key encryption key pairs $(\tilde{sk}_i, \tilde{pk}_i) \leftarrow \pi^{\text{pke}}.\mathsf{Keygen}(1^\lambda)$ itself for each $i \in \{1, \ldots, n\} \backslash \{i^\star\}$. It finally chooses public parameters cp for the hint scheme and generates $(sk_i^h, pk_i^h) \leftarrow \pi^{\text{hint}}.\mathtt{Keygen}(\mathsf{cp})$ for $i = 1$ to n. The master public key BE-MPK $= \big(\mathsf{cp},\ \{(\tilde{pk}_i, pk_i^h)\}_{i=1}^n, \Sigma\big)$ is given to $\mathcal{A}$.

At any time, $\mathcal{A}$ is allowed to corrupt an arbitrary user $i \in \{1, \ldots, n\}$ depending on the information it gathered so far. At each corruption query, $\mathcal{B}$ aborts and fails in the event that $\mathcal{A}$ chooses to corrupt user $i^\star$. Otherwise, $\mathcal{B}$ is necessarily able to consistently answer the query since it knows secret keys $\{\tilde{sk}_i, sk_i^h\}_{i \neq i^\star}$. When $\mathcal{A}$ makes a decryption query $(C = (\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma), i)$, we assume that the query involves the challenge key $\tilde{pk}^\star$ since $\mathcal{B}$ can always run the legal decryption procedure otherwise. To simulate the decryption algorithm without knowing $\tilde{sk}^\star$, $\mathcal{B}$ appeals to its own IND-CCA challenger. It first uses the i-th hint private key sk_i^h to compute $H = \pi^{\text{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_i, \mathsf{VK}, U)$. If $H \neq H_j$ for each $j \in \{1, \ldots, \ell\}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ considers the smallest index j such that $H = H_j$ and sends the decryption query C_j to its own challenger. If the latter's response can be parsed as $M||\mathsf{VK}$, for some M of appropriate length, $\mathcal{B}$ returns M to $\mathcal{A}$. Otherwise, $\mathcal{B}$ returns $\perp$ to indicate that the ciphertext fails to decrypt properly.

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and two subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size. At this step, $\mathcal{B}$ re-orders S_0, S_1 as $S'_0 = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$, $S'_1 = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ where $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$. If $\theta_k \neq i^\star$, $\mathcal{B}$ aborts and declares failure and we denote by Good the event that $\theta_k = i^\star$.

If the event Good occurs, $\mathcal{B}$ chooses a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and sends the two messages $(M_0||\mathsf{VK}^\star)$, $(M_1||\mathsf{VK}^\star)$ to its IND-CCA challenger. The latter returns a challenge ciphertext $C^\star = \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}^\star, M_b||\mathsf{VK}^\star)$, for some hidden random bit $b \in_R \{0, 1\}$. The ANOBE challenge ciphertext is then generated by choosing $r \xleftarrow{\$} \mathcal{R}^h$ and defining ciphertext components as follows.

1. For $j = 1$ to $k-1$, $\mathcal{B}$ computes $(U, H_j) = \pi^{\text{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\theta_j}^h, r)$ and sets (H_j, C_j) as
$$(H_j, C_j) = \big(H_j,\ \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_1||\mathsf{VK}^\star)\big).$$
2. For $j = k+1$ to ℓ, $\mathcal{B}$ computes $(U, H_j) = \pi^{\text{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\theta_j}^h, r)$ and sets (H_j, C_j) as
$$(H_j, C_j) = \big(H_j,\ \pi^{\text{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)\big).$$
3. Finally, set $(H_k, C_k) = (H_k, C^\star)$, where H_k is obtained as $(U, H_k) = \pi^{\text{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\theta_k}^h, r)$.

The adversary $\mathcal{A}$ then receives the ciphertext $C = (\mathsf{VK}^\star, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}^\star, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}))$ and for some permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$.

In the second phase, $\mathcal{A}$ makes further adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ and $\mathcal{B}$ handles them as in the first phase. Whenever $\mathcal{A}$ makes a decryption query (C, i), $\mathcal{B}$ parses the ciphertext C as $C = (\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma)$ and outputs $\perp$ if $\mathsf{VK} = \mathsf{VK}^\star$ or if σ is invalid. Otherwise, if $i \neq i^\star$, $\mathcal{B}$ can run the decryption algorithm by itself since it knows the private key $\tilde{sk}_i$. If $i = i^\star$, $\mathcal{B}$ appeals to the decryption oracle it is given access to. It first uses $sk^h_{i^\star}$ to compute $H = \pi^{\text{hint}}.\mathtt{Invert}(\mathsf{cp}, sk^h_{i^\star}, \mathsf{VK}, U)$ and determine which ciphertext should be decrypted among $\{C_1, \ldots, C_\ell\}$. If $H \notin \{H_1, \ldots, H_\ell\}$, $\mathcal{B}$ returns $\perp$. Assuming that $H = H_j$ for some $j \in \{1, \ldots, \ell\}$, $\mathcal{B}$ returns $\perp$ if $C_j = C^\star$ as, given that $\mathsf{VK} \neq \mathsf{VK}^\star$, the real ANOBE decryption necessarily gives $\perp$ since $C^\star$ decrypts to $M_b||\mathsf{VK}^\star$. Otherwise, upon receiving the challenger's response $M||\mathsf{VK}'$, $\mathcal{B}$ returns M if $\mathsf{VK} = \mathsf{VK}'$. Otherwise, it returns $\perp$.

At the end of the game, $\mathcal{A}$ outputs $b' \in \{0, 1\}$ and $\mathcal{B}$ produces the same result. We claim that, if $\mathcal{B}$ did not abort, its IND-CCA advantage is as large as the difference between $\mathcal{A}$'s probabilities of outputting 0 in Game k and Game k'. Indeed, if $\mathcal{B}$'s challenger chooses $b = 0$, then $\mathcal{B}$ is playing Game k whereas, if $b = 1$, $\mathcal{B}$ is playing Game k'.

When it comes to evaluate $\mathcal{B}$'s probability not to abort, we note that, whenever $M_0 \neq M_1$, $\mathcal{A}$ is not allowed to corrupt any user in $S_0 \cap S_1 = \{\theta_1, \ldots, \theta_\phi\}$. Since $\theta_k \in S_0 \cap S_1$, a sufficient condition for $\mathcal{B}$ not to be asked for $\tilde{sk}_{i^\star}$ is to have Good occurring. This is the case with probability $\Pr[\mathsf{Good}] = 1/n$ since the index $i^\star$ is chosen at random independently of $\mathcal{A}$'s view. □

Lemma 8. *For each $k \in \{\phi+1, \ldots, \ell\}$, Game k' is indistinguishable from Game k if the encryption scheme is AI-CCA secure and weakly robust. More precisely, for any ANOBE adversary distinguishing the two games, there exists either an AI-CCA adversary $\mathcal{B}$ or a WROB-CCA adversary $\mathcal{B}'$ such that*

$$|\Pr[E_k] - \Pr[E'_k]| \leq n^2 \cdot \Big(\mathbf{Adv}^{\text{ai-cca}}(\mathcal{B}) + \mathbf{Adv}^{\text{wrob-cca}}(\mathcal{B}')\Big).$$

Proof. We show that, if the ANOBE adversary $\mathcal{A}$ is able to distinguish Game k and Game k', we can either construct an AI-CCA adversary $\mathcal{B}$ against the public key encryption scheme or break its weak robustness property.

Our AI-CCA adversary $\mathcal{B}$ is given as input two public keys $\tilde{pk}^\star_0$, $\tilde{pk}^\star_1$ from its AI-CCA challenger and we denote by $\tilde{sk}^\star_0$ and $\tilde{sk}^\star_1$ the corresponding private keys. Algorithm $\mathcal{B}$ has to generate a master public key BE-MPK containing n public key encryption keys $\tilde{pk}_1, \ldots, \tilde{pk}_n$. To do this, $\mathcal{B}$ randomly picks two distinct indices $i^\star_0, i^\star_1 \xleftarrow{\$} \{1, \ldots, n\}$ and sets $\tilde{pk}_{i^\star_0} = \tilde{pk}^\star_0$ and $\tilde{pk}_{i^\star_1} = \tilde{pk}^\star_1$. Then, $\mathcal{B}$ generates $n-2$ key pairs $(\tilde{sk}_i, \tilde{pk}_i) \leftarrow \pi^{\text{pke}}.\mathsf{Keygen}(1^\lambda)$ for each $i \in \{1, \ldots, n\} \backslash \{i^\star_0, i^\star_1\}$. Finally, $\mathcal{B}$ also chooses public parameters cp for the hint scheme and generates $(sk^h_i, pk^h_i) \leftarrow \pi^{\text{hint}}.\mathsf{Keygen}(\mathsf{cp})$ for $i = 1$ to n. The master public key consisting of $\big(\mathsf{cp}, (\tilde{pk}_0, pk^h_0), \ldots, (\tilde{pk}_n, pk^h_n), \Sigma\big)$ is given as input to $\mathcal{A}$.

During the game, the adversary $\mathcal{A}$ is allowed to adaptively corrupt any user $i \in \{1, \ldots, n\}$. At each corruption query, $\mathcal{B}$ aborts in the event that the queried i is such that $i \in \{i^\star_0, i^\star_1\}$. Otherwise, $\mathcal{B}$ necessarily knows the queried secret key $(\tilde{sk}_i, sk^h_i)$ and returns it to $\mathcal{A}$.

For each decryption query $(C = (\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma), i)$ made by $\mathcal{A}$, $\mathcal{B}$ can reply on its own whenever $i \notin \{i^\star_0, i^\star_1\}$. If $i = i^\star_0$ (resp. $i = i^\star_1$), $\mathcal{B}$ uses sk^h_i to determine which ciphertext should be decrypted among $C_1, \ldots, C_\ell$. Namely, if $H_j \neq \pi^{\text{hint}}.\mathsf{Invert}(sk^h_i, \mathsf{VK}, U)$ for $j = 1$ to ℓ, $\mathcal{B}$ returns $\perp$. If it turns out that $H_j = \pi^{\text{hint}}.\mathsf{Invert}(sk^h_i, \mathsf{VK}, U)$ for some $j \in \{1, \ldots, \ell\}$, $\mathcal{B}$ queries its own decryption oracle and asks it for the decryption of C_j under $\tilde{sk}_0^\star$ (resp. $\tilde{sk}_1^\star$).

In the challenge phase, $\mathcal{A}$ outputs messages M_0, M_1 and subsets $S_0, S_1 \subset \{1, \ldots, n\}$ of equal size ℓ. These sets are re-ordered as $S_0' = \{\theta_1, \ldots, \theta_\phi, \theta_{\phi+1}, \ldots, \theta_\ell\}$ and $S_1' = \{\rho_1, \ldots, \rho_\phi, \rho_{\phi+1}, \ldots, \rho_\ell\}$ where $\theta_j = \rho_j$ for each $j \in \{1, \ldots, \phi\}$. If $\theta_k \neq i_0^\star$ or $\rho_k \neq i_1^\star$, $\mathcal{B}$ aborts. We denote by Good the event $(\theta_k = i_0^\star) \wedge (\rho_k = i_1^\star)$, which implies $\tilde{pk}_{\theta_k} = \tilde{pk}_0^\star$ and $\tilde{pk}_{\rho_k} = \tilde{pk}_1^\star$.

If the event Good occurs, $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star) \leftarrow \mathcal{G}(\lambda)$ and sends the messages $(M_0||\mathsf{VK}^\star, M_1||\mathsf{VK}^\star)$ to its AI-CCA challenger. The latter flips a random coin $b \xleftarrow{\$} \{0,1\}$ and returns a challenge ciphertext $C^\star = \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_b, M_b||\mathsf{VK}^\star)$. The adversary's challenge ciphertext is then obtained by choosing $r \xleftarrow{\$} \mathcal{R}^h$ and computing ciphertext components as follows.

1. For $j = 1$ to $k-1$, $\mathcal{B}$ sets $(H_j, C_j) = \big(H_j,\ \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\rho_j}, M_1||\mathsf{VK}^\star)\big)$, where H_j is obtained as part of $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\rho_j}^h, r)$
2. For $j = k+1$ to ℓ, $\mathcal{B}$ computes $(H_j, C_j) = \big(H_j,\ \pi^{\mathrm{pke}}.\mathsf{Encrypt}(\tilde{pk}_{\theta_j}, M_0||\mathsf{VK}^\star)\big)$, where H_j is obtained as per $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\theta_j}^h, r)$
3. Finally, set $(H_k, C_k) = (H_k, C^\star)$, where $(U, H_k) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}^\star, pk_{\rho_k}^h, r)$.

The adversary $\mathcal{A}$ is then given the challenge $C = (\mathsf{VK}^\star, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}^\star, (U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)})))$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

In the second phase, $\mathcal{A}$ makes further adaptive corruption queries for indices $i \notin S_0 \triangle S_1$ and $\mathcal{B}$ handles them as previously. Decryption queries are also dealt with in the same way with one difference: namely, if $\mathcal{A}$ makes a decryption query $(C = (\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma), i)$ for which we simultaneously have $i \in \{i_0^\star, i_1^\star\}$ and

$$\mathsf{VK} \neq \mathsf{VK}^\star, \qquad H_j = \pi^{\mathrm{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_i^h, \mathsf{VK}, U), \qquad C_j = C^\star,$$

for some $j \in \{1, \ldots, \ell\}$, our adversary $\mathcal{B}$ returns $\perp$. Since $\mathsf{VK} \neq \mathsf{VK}^\star$, it is easy to see that $C^\star$ cannot correctly decrypt to a message ending with VK under the private key $\tilde{sk}_b^\star$. Still, we have to consider the probability to have $\mathsf{Decrypt}(\tilde{sk}_{1-b}^\star, C^\star) = M||\mathsf{VK}$ for some plaintext M, which would render $\mathcal{A}$'s view inconsistent. However, it is easy to see that $\mathcal{B}$ could be turned into a weak robustness (more precisely, WROB-CCA) adversary $\mathcal{B}'$ if the latter event occurs with non-negligible probability.

When $\mathcal{A}$ terminates, it outputs a bit $b' \in \{0,1\}$ and $\mathcal{B}$ outputs b' as well. If $\mathcal{B}$ did not abort, its AI-CCA advantage is easily seen to be as large as the difference between $\mathcal{A}$'s probabilities of outputting 0 in Game k and Game k'. Indeed, if $\mathcal{B}$'s AI-CCA challenger sets its challenge bit as $b = 0$, $\mathcal{B}$ is playing Game k with $\mathcal{A}$ whereas, in the situation $b = 1$, $\mathcal{B}$ is playing Game k'.

When it comes to assess $\mathcal{B}$'s probability not to abort, we recall that $\mathcal{A}$ cannot corrupt any user in $S_0 \triangle S_1 = \{\theta_{\phi+1}, \ldots, \theta_\ell, \rho_{\phi+1}, \ldots, \rho_\ell\}$. Hence, a sufficient condition for the adversary $\mathcal{A}$ not to query the private keys $\tilde{sk}_{\theta_k}$ or $\tilde{sk}_{\rho_k}$ is to have Good occurring. Since it does so with probability $\Pr[\mathsf{Good}] = 1/n(n-1) > 1/n^2$, the announced result follows. $\square$

We note that, if we assume that the underlying encryption scheme supports labels (like, e.g., the Cramer-Shoup cryptosystem), shorter ciphertexts can be obtained by including the verification key VK in the label of each ciphertext (instead of appending it to the plaintext). In this case, the proof of lemma 8 does not have to assume that the encryption scheme is weakly robust.

F Security Definitions for Public Key Encryption

F.1 Definition of AI-CCA Security for Encryption Schemes

Definition 7. *A public key encryption scheme* $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is AI-CCA secure if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger generates two key pairs* $(sk_0, pk_0) \leftarrow \mathsf{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \mathsf{Keygen}(\lambda)$ *and gives* pk_0, pk_1 *to the adversary* $\mathcal{A}$.
2. *The adversary* $\mathcal{A}$ *queries decryption oracles* $\mathcal{O}^{sk_0}(.)$, $\mathcal{O}^{sk_1}(.)$. *Each query consists of an adversarially-chosen ciphertext* C.
3. *In the challenge phase,* $\mathcal{A}$ *outputs equal-length messages* m_0, m_1. *The challenger then flips a coin* $b \xleftarrow{\$} \{0,1\}$ *and returns* $C^\star = \mathsf{Encrypt}(pk_b, m_b)$.
4. $\mathcal{A}$ *makes further queries to decryption oracles* $\mathcal{O}^{sk_0}(.)$, $\mathcal{O}^{sk_1}(.)$ *but is disallowed to query* $C^\star$ *to* $\mathcal{O}^{sk_0}(.)$ *and* $\mathcal{O}^{sk_1}(.)$.
5. *The adversary* $\mathcal{A}$ *outputs a bit* $b' \in \{0,1\}$ *and wins if* $b' = b$.

As always, $\mathcal{A}$*'s advantage is defined as* $\mathbf{Adv}^{\text{ai-cca}}(\mathcal{A}) = |\Pr[b' = b] - \frac{1}{2}|$.

F.2 Definitions of Robustness

This section recalls the definitions of robust public key encryption given by Abdalla, Bellare and Neven [2].

Definition 8. *[2] A public key encryption scheme* $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is strongly robust under chosen-ciphertext attacks (SROB-CCA) if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger generates two distinct key pairs* $(sk_0, pk_0) \leftarrow \mathsf{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \mathsf{Keygen}(\lambda)$ *and sends* pk_0, pk_1 *to the adversary* $\mathcal{A}$.
2. *On a polynomial number of occasions,* $\mathcal{A}$ *invokes decryption oracles* $\mathcal{O}^{sk_0}(.)$, $\mathcal{O}^{sk_1}(.)$ *for arbitrary ciphertexts of its choice. The adversary* $\mathcal{A}$ *wins if it manages to make a decryption query* C *for which* $\mathcal{O}^{sk_0}(C) \neq \perp$ *and* $\mathcal{O}^{sk_1}(C) \neq \perp$.

The advantage of $\mathcal{A}$ *is defined to be its probability of success, taken over all random coins.*

For our purposes, the following weaker form of robustness suffices.

Definition 9. *[2] A public key encryption scheme* $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is weakly robust under chosen-ciphertext attacks (WROB-CCA) if no PPT adversary has non-negligible advantage in the game hereafter:*

1. *The challenger generates key pairs* $(sk_0, pk_0) \leftarrow \mathsf{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \mathsf{Keygen}(\lambda)$ *and sends public keys* pk_0, pk_1 *to the adversary* $\mathcal{A}$.
2. *The adversary* $\mathcal{A}$ *adaptively invokes decryption oracles* $\mathcal{O}^{sk_0}(.)$, $\mathcal{O}^{sk_1}(.)$ *for arbitrary ciphertexts.*
3. *Eventually,* $\mathcal{A}$ *halts and outputs a message* m. *Then, the challenger computes* $C = \mathsf{Encrypt}(pk_0, m)$ *and the adversary* $\mathcal{A}$ *wins if* $\mathsf{Decrypt}(sk_1, C) \neq \perp$.

Again, $\mathcal{A}$*'s advantage is its success probability, taken over all random coins.*

Finally, to guarantee the correctness of our construction of ANOBE from TA-anonymous IBE, the following definition of weak robustness must be satisfied by the underlying IBE system.

Definition 10. *An multi-TA IBE scheme* $(\mathsf{CommonSetup}, \mathsf{TASetup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ *is fixed-ID weakly robust under chosen-plaintext attacks (FID-WROB-CPA) if no PPT adversary has non-negligible advantage in the game below:*

1. *The challenger generates two master key pairs* $(msk_0, mpk_0) \leftarrow \mathsf{TASetup}(\lambda)$, $(msk_1, mpk_1) \leftarrow \mathsf{TASetup}(\lambda)$ *and gives both master public keys* mpk_0, mpk_1 *to the adversary* $\mathcal{A}$.
2. *The adversary* $\mathcal{A}$ *adaptively interacts with key extraction oracles* $\mathcal{O}^{msk_0}(.)$, $\mathcal{O}^{msk_1}(.)$ *for arbitrary identities of its choice.*
3. *At the end of the game,* $\mathcal{A}$ *halts and outputs a plaintext* M. *At this point, the challenger chooses an identity* ID, *computes a ciphertext* $C = \mathsf{Enc}(mpk_0, M, \mathsf{ID})$ *and the adversary* $\mathcal{A}$ *wins if* $\mathsf{Dec}(mpk_1, sk_{\mathsf{ID}}, C) \neq \perp$, *where* $sk_{\mathsf{ID}} = \mathsf{KeyDer}(mpk_1, msk_1, \mathsf{ID})$.

$\mathcal{A}$'s advantage is defined analogously to definition 9.

In [2], Abdalla *et al.* showed a transformation providing weak robustness in any public-key or identity-based encryption scheme. Moreover, if the underlying scheme is AI-CPA (resp. AI-CCA), so is the resulting construction.

This transformation is a keyed redundancy-based transformation. If the common public parameters contain a sufficiently long random string $K_{wrob} \in \{0,1\}^k$, the idea is simply to encrypt a concatenation of K and the actual message: in other words, a message M is encrypted as per $C = \mathsf{Encrypt}(par, pk, M||K_{wrob})$. Upon decryption, it must be checked that, if $M' = \mathsf{Decrypt}(sk, C)$, the obtained M' can be parsed as $M' = M||K_{wrob}$.

In the IBE setting, the same transformation is easily seen to provide FID-WROB-CPA if the IBE system provides selective semantic security. Indeed, the proof of [2][Theorem 4.1] easily goes through with simple changes.

Lemma 9. *In the identity-based setting, the weak-robustness-conferring transformation of [2] provides FID-WROB-CPA security when applied to an IND-sID-CPA secure IBE scheme.*

Proof. The proof considers two games, called Game 0 and Game 1. In the latter, we argue that the adversary $\mathcal{A}$ can only win with negligible probability.

Game 0: is almost exactly the game of Definition 10. The only difference is that the identity ID is chosen at the outset of the game instead of step 3. Still, since ID is independent of $\mathcal{A}$'s view until step 3, this is difference is invisible to $\mathcal{A}$. We define E_0 to be the event that the adversary $\mathcal{A}$ wins.

Game 1: proceeds as follows. The challenger generates (msk_0, mpk_0) and (msk_1, mpk_1) and gives (mpk_0, mpk_1) to $\mathcal{A}$. It responds to $\mathcal{A}$'s key extraction queries as in Game 0. In the last phase, however, it computes C as $C = \mathsf{Enc}(mpk_0, 0^{|M|}||0^{|K_{wrob}|}, \mathsf{ID})$. We define E_1 as the event that $\mathsf{Dec}(mpk_1, sk_{\mathsf{ID}}, C) \neq \perp$ with $sk_{\mathsf{ID}} = \mathsf{KeyDer}(mpk_1, msk_1, \mathsf{ID})$.

It is very simple to build an IND-sID-CPA adversary $\mathcal{A}'$ against the IBE scheme such that $|\Pr[E_0] - \Pr[E_1]| \leq \mathbf{Adv}^{\text{ind-sid-cpa}}(\mathcal{A}')$. Namely, $\mathcal{A}'$ chooses a target identity ID and receives mpk_0 from its challenger. It generates (msk_1, mpk_1) itself and gives (mpk_0, mpk_1) to $\mathcal{A}$. Using msk_1 and its IND-sID-CPA challenger, it can answer all key extraction queries made by $\mathcal{A}$. In step 3 of the FID-WROB-CPA game, $\mathcal{A}$ outputs a message M. At that point $\mathcal{A}'$ defines two messages

$M_0' = M||K_{wrob}$ and $M_1' = 0^{|M|}||0^{|K_{wrob}|}$ which it sends to its challenger and obtains a challenge ciphertext $C^\star$. Using $sk_{\mathsf{ID}} = \mathsf{Keyder}(mpk_1, msk_1, \mathsf{ID})$, $\mathcal{A}'$ decrypts $C^\star$ and outputs 1 if the result is not $\perp$.

In Game 1, the same arguments as in [2][Theorem 4.1] show that C is statistically independent of C. Indeed, if t is a bound on $\mathcal{A}$'s running time, the only information about K_{wrob} that can be carried by C is $|M|$. Since the adversarially-chosen M contains at most t bits of information about K_{wrob}, Lemma D.1 of [2] implies that $\Pr[E_1] = 2^{\lceil \log t \rceil - k}$, which is negligible for a sufficiently large k. Note that, unlike [2][Theorem 4.1], we do not have to worry about a possible correlation between ID and K_{wrob} since ID is honestly chosen by the challenger. $\square$

G The Kurosawa-Desmedt Encryption Scheme

The following description of the KD cryptosystem [36] assumes common public parameters consisting of a group $\mathbb{G}$ of prime order $p > 2^\lambda$, with generators $g_1, g_2 \in_R \mathbb{G}$. They also include the description of a universal one-way hash function $H : \{0,1\}^* \to \mathbb{Z}_p$, a key derivation function $\mathsf{KDF} : \mathbb{G} \to \{0,1\}^k$, for some integer $k \in \mathsf{poly}(\lambda)$, a symmetric *authenticated* encryption scheme $\Pi^{\text{sym-enc}} = (\mathsf{E}, \mathsf{D})$ of key length k

Keygen(λ, cp)**:** given common public parameters $\mathsf{cp} = (\mathbb{G}, g_1, g_2, H, \Pi^{\text{sym-enc}})$, choose random exponents $x_1, x_2, y_1, y_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and compute

$$c = g_1^{x_1} g_2^{x_2}, \qquad\qquad d = g_1^{y_1} g_2^{y_2}$$

The public key is $pk = (c, d)$ and the private key is $sk = (x_1, x_2, y_1, y_2)$.

Encrypt(pk, m)**:** to encrypt a message $m \in \mathbb{G}$,

1. Pick $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and compute

$$u_1 = g_1^r, \qquad u_2 = g_2^r, \qquad v = (c \cdot d^\alpha)^r,$$

where $\alpha = H(u_1, u_2) \in \mathbb{Z}_p$.

2. Compute $K = \mathsf{KDF}(v) \in \{0,1\}^k$, $e = \mathsf{E}_K(m)$.

The ciphertext is $C = (u_1, u_2, e)$.

Decrypt(sk, C)**:** parse the ciphertext C as (u_1, u_2, e). Compute $\alpha = H(u_1, u_2)$, $v = u_1^{x_1 + \alpha \cdot y_1} \cdot u_2^{x_2 + \alpha \cdot y_2}$ and $K = \mathsf{KDF}(v) \in \{0,1\}^k$. Then, return $m = \mathsf{D}_K(e)$ (which may be $\perp$ if the e fails to properly decrypt under the key K).

The above algorithms describe the original Kurosawa-Desmedt encryption scheme. Following [2], we denote by KD* the modified KD scheme where the encryption exponent $r = 0$ is explicitly disallowed: namely, the sender chooses $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ (instead of $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p$) at encryption and the receiver outputs $\perp$ upon receiving a ciphertext (u_1, u_2, e) such that $u_1 = 1_{\mathbb{G}}$.

To prove the strong robustness of KD*, we will need a symmetric authenticated encryption scheme satisfying the following definition.

Definition 11 ([26]). *A symmetric encryption scheme* (E, D) *is* **key-binding** *if, for any message* m*, any key* k *and any randomness* r*, there exists no key* k' *such that* $k' \neq k$ *and* $\mathsf{D}_{k'}(\mathsf{E}_k(m)) \neq \perp$.

Key-binding symmetric encryption schemes are relatively simple to construct (see [26] for example).

Theorem 6. *The KD* scheme is SROB-CCA assuming that (i)* KDF *is a secure key derivation function and is additionally collision-resistant; (ii) the hash function* $H : \{0,1\}^* \to \mathbb{Z}_p$ *is pre-image resistant; (iii)* $\Pi^{\text{sym-enc}}$ *is a key binding symmetric authenticated encryption scheme.*

Proof. The proof uses a sequence of games where, in Game i, we denote by S_i the event that the challenger outputs 1 (meaning that the adversary wins the SROB-CCA game). The sequence proceeds as follows.

Game 0: is the real strong robustness game. Namely, the challenger $\mathcal{B}$ chooses a prime order group $\mathbb{G}$ with generators $g_1, g_2 \xleftarrow{\$} \mathbb{G}$. It also generates two public keys $pk_0 = (c_0, d_0)$ and $pk_1 = (c_1, d_1)$, where

$$c_0 = g_1^{x_{0,1}} g_2^{x_{0,2}}, \qquad d_0 = g_1^{y_{0,1}} g_2^{y_{0,2}}$$
$$c_1 = g_1^{x_{1,1}} g_2^{x_{1,2}}, \qquad d_1 = g_1^{y_{1,1}} g_2^{y_{1,2}}$$

and $sk_0 = (x_{0,1}, x_{0,2}, y_{0,1}, y_{0,2})$ and $sk_1 = (x_{1,1}, x_{1,2}, y_{1,1}, y_{1,2})$ are the underlying private keys. During the game, the adversary $\mathcal{A}$ is allowed to simultaneously invoke both decryption oracles $\mathcal{O}_{sk_0}(.)$, $\mathcal{O}_{sk_1}(.)$ on ciphertexts of its choice. For each adversarially-generated ciphertext $C = (u_1, u_2, e)$, the challenger executes the actions of $\mathcal{O}_{sk_0}(.)$ and $\mathcal{O}_{sk_1}(.)$. If either decryption gives the result $\perp$, the challenger simply returns the outputs of both oracles to $\mathcal{A}$. In the event that $\mathcal{A}$ makes a decryption query C gives $\mathcal{O}_{sk_0}(C) = M_0 \neq \perp$ and $\mathcal{O}_{sk_1}(C) = M_1 \neq \perp$, the challenger returns M_0, M_1 to $\mathcal{A}$, halts and outputs 1. If $\mathcal{A}$ terminates without any occurrence of the latter event, the challenger $\mathcal{B}$ outputs 0. We call S_0 the event of $\mathcal{B}$ outputting 1 at the end of its interaction with $\mathcal{A}$.

Game 1: is identical to Game 0 with the difference that the challenger $\mathcal{B}$ makes uses of the discrete logarithm $\omega = \log_{g_1}(g_2) \in \mathbb{Z}_p$ when handling decryption queries. Namely, when $\mathcal{A}$ comes up with a ciphertext $C = (u_1, u_2, e)$, $\mathcal{B}$ returns $(M_0, M_1) = (\perp, \perp)$ if $u_2 \neq u_1^{\omega}$. Clearly, Game 1 and Game 0 are identical until, on behalf of $\mathcal{O}_{sk_1}(.)$ or $\mathcal{O}_{sk_2}(.)$, $\mathcal{B}$ outputs $\perp$ for a ciphertext that would correctly decrypt in Game 0. If we call F_1 the latter event, we clearly have $|\Pr[S_1] - \Pr[S_0]| \leq \Pr[F_1]$. The same "plug and pray" argument as in [27,3,21] shows that event F_1 implies either a distinguisher $\mathcal{B}'$ for the key derivation function or an adversary $\mathcal{B}''$ against the (weak) ciphertext integrity of the symmetric encryption scheme (see section H for definitions of these properties) . Since algorithms $\mathcal{B}'$ and $\mathcal{B}''$ have to guess upfront which decryption query will involve the accepted invalid ciphertext, we actually have $\Pr[F_1] \leq q \cdot (\mathbf{Adv}_{\mathcal{B}'}^{\mathsf{PRF\text{-}KDF}}(\lambda) + \mathbf{Adv}_{\mathcal{B}''}^{\mathsf{CT\text{-}INT}}(\lambda))$.

Game 2: is as Game 1 but we introduce a new failure event F_2 which causes the challenger $\mathcal{B}$ to halt and output 0 if it occurs. This event F_2 consists in the adversary $\mathcal{A}$ invoking the decryption oracles on a ciphertext $C = (u_1, u_2, e)$ such that $u_2 = u_1^{\omega}$, $\mathsf{KDF}(v_0) = \mathsf{KDF}(v_1)$ and $v_0 \neq v_1$, where

$$v_0 = u_1^{x_{0,1}+\alpha \cdot y_{0,1}} u_2^{x_{0,2}+\alpha \cdot y_{0,2}}, \qquad v_1 = u_1^{x_{1,1}+\alpha \cdot y_{1,1}} u_2^{x_{1,2}+\alpha \cdot y_{1,2}} \tag{G.1}$$

with $\alpha = H(u_1, u_2) \in \mathbb{Z}_p$. Obviously, event F_2 implies an algorithm $\mathcal{B}'''$ finding a collision on the key derivation function and we can write $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq \mathbf{Adv}_{\mathcal{B}'''}^{\mathsf{CR\text{-}KDF}}(\lambda)$.

Game 3: is identical to Game 2 but we add yet another failure event F_3 that gets the challenger $\mathcal{B}$ to stop and output 0. We call F_3 the event that $\mathcal{A}$ outputs a ciphertext $C = (u_1, u_2, e)$ such that $u_2 = u_1^{\omega}$ and $v_0 = v_1$, where v_0 and v_1 are given by (G.1). Since $u_2 = u_1^{\omega}$ (in other words,

$\log_{g_1}(u_1) = \log_{g_2}(u_2)$), the condition $v_0 = v_1$ implies $c_0 d_0^{\alpha} = c_1 d_1^{\alpha}$, where $\alpha = H(u_1, u_2) \in \mathbb{Z}_p$. The same reasoning as in [2][Theorem 5.1] shows that this implies an algorithm $\mathcal{B}^{\text{pre-img}}$ finding pre-images for random elements in the range of H. Indeed, the condition $c_0 d_0^{\alpha} = c_1 d_1^{\alpha}$ implies the equality

$$\alpha = \frac{(x_{1,1} - x_{0,1}) + \omega \cdot (x_{1,2} - x_{0,2})}{(y_{0,1} - y_{1,1}) + \omega \cdot (y_{0,2} - y_{1,2})}. \tag{G.2}$$

Suppose that algorithm $\mathcal{B}^{\text{pre-img}}$ receives as input a random $\alpha^{\star} \in \mathbb{Z}_p$ in the range of H with the task of finding a pre-image for it. Assuming that F_3 occurs with noticeable probability, $\mathcal{B}^{\text{pre-img}}$ can prepare the two public keys $pk_0 = (c_0, d_0)$ and $pk_1 = (c_1, d_1)$ in such a way that the underlying private keys $sk_0 = (x_{0,1}, x_{0,2}, y_{0,1}, y_{0,2})$, $sk_1 = (x_{1,1}, x_{1,2}, y_{1,1}, y_{1,2})$ satisfy (G.2) when the target $\alpha^{\star}$ is the left-hand-side member. This will force $\mathcal{A}$ to output a ciphertext (u_1, u_2, e) such that $\alpha^{\star} = H(u_1, u_2)$ and thereby break the pre-image resistance of H.

Since $\mathcal{B}^{\text{pre-img}}$'s challenger chooses $\alpha^{\star} \in \mathbb{Z}_p$ uniformly and independently of $\mathcal{A}$'s view, the above choice of sk_0 and sk_1 is easily seen not to affect the distribution of pk_0 and pk_1. We can thus write $|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3] \leq \mathbf{Adv}^{\mathsf{pre\text{-}img}}_{\mathcal{B}^{\text{pre-img}}}(\lambda)$.

In Game 3, we claim that $\Pr[S_3] = 0$ as we have ruled out all the possibilities for $\mathcal{A}$ to win the SROB-CCA game. Indeed, the only possibility for $\mathcal{A}$ to output $C = (u_1, u_2, e)$ such that $\mathcal{O}_{sk_1}(C) \neq \perp$ and $\mathcal{O}_{sk_2}(C) \neq \perp$ would be to have $\mathsf{D}_{k_0}(e) \neq \perp$ and $\mathsf{D}_{k_1}(e) \neq \perp$, where $k_0 = \mathsf{KDF}(v_0)$, $k_1 = \mathsf{KDF}(v_1)$ and $v_i = u_1^{x_{i,1}+\alpha \cdot y_{i,1}} u_2^{x_{i,2}+\alpha \cdot y_{i,2}}$ for $i = 0, 1$. Since $k_0 \neq k_1$ unless one of the events F_2 or F_3 occurs, we cannot simultaneously have $\mathsf{D}_{k_0}(e) \neq \perp$ and $\mathsf{D}_{k_1}(e) \neq \perp$ as long as $\Pi^{\text{sym-enc}}$ is a key binding symmetric encryption scheme in the sense of definition 11. $\square$

We now give a proof that KD* is key-private under chosen-ciphertext attacks. The proof makes use of standard techniques along the lines of [6][Theorem 6] and [21]. Although the result seems pretty straightforward, we have not been able to find a proof for it in the literature.

Theorem 7. *The KD* scheme is AI-CCA assuming that: (i) the DDH assumption holds in $\mathbb{G}$; (ii) H is a universal one-way hash function; (iii)* KDF *is a secure key-derivation function; (iv) $\Pi^{\text{sym-enc}}$ is a secure symmetric authenticated encryption scheme.*

Proof. The proof uses a sequence of games where, in Game i, S_i stands for the event that the adversary successfully guesses the challenger's bit $b \in \{0, 1\}$. The proof makes use of Halevi's sufficient condition for key-privacy. Namely, in the challenge phase, the adversary's chosen message M is ignored and the challenger $\mathcal{B}$ rather computes an encryption of a random message $M^{\star}$. The sequence of games is very similar to the one of [21] and proceeds as follows.

Game 0: is the real attack game. More precisely, the challenger $\mathcal{B}$ chooses a prime order group $\mathbb{G}$ with generators $g_1, g_2 \xleftarrow{\$} \mathbb{G}$. It also generates two public keys $pk_0 = (c_0, d_0)$ and $pk_1 = (c_1, d_1)$, where

$$c_0 = g_1^{x_{0,1}} g_2^{x_{0,2}}, \qquad d_0 = g_1^{y_{0,1}} g_2^{y_{0,2}}$$
$$c_1 = g_1^{x_{1,1}} g_2^{x_{1,2}}, \qquad d_1 = g_1^{y_{1,1}} g_2^{y_{1,2}}$$

and $sk_0 = (x_{0,1}, x_{0,2}, y_{0,1}, y_{0,2})$ and $sk_1 = (x_{1,1}, x_{1,2}, y_{1,1}, y_{1,2})$ are the underlying private keys. During the game, the adversary $\mathcal{A}$ is allowed to invoke both decryption oracles $\mathcal{O}_{sk_0}(.)$, $\mathcal{O}_{sk_1}(.)$ on arbitrary ciphertext C of its choice. For each query $C = (u_1, u_2, e)$ made to oracle $\mathcal{O}_{sk_i}(.)$, for $i \in \{0,1\}$, the challenger runs the decryption process using sk_i and hands the result of the decryption process to $\mathcal{A}$. In the challenge phase, the adversary $\mathcal{A}$ comes up with a plaintext M. At this step, the challenger ignores M, flips a random coin $b^\star$, chooses a random plaintext $M^\star$, picks $r^\star \xleftarrow{\$} \mathbb{Z}_p^*$ and computes

$$u_1^\star = g_1^{r^\star}, \qquad u_2^\star = g_2^{r^\star}, \qquad v^\star = (c_{b^\star} d_{b^\star}^{\alpha^\star})^{r^\star}, \qquad e^\star = \mathsf{E}_{K^\star}(M^\star),$$

where $K^\star = \mathsf{KDF}(v^\star)$ and $\alpha^\star = H(u_1^\star, u_2^\star)$. We let $C^\star = (u_1^\star, u_2^\star, e^\star)$ be the challenge ciphertext sent to $\mathcal{A}$.

In the second phase, $\mathcal{A}$ makes further decryption queries C such that $C \neq C^\star$ and eventually outputs a bit $b' \in \{0,1\}$. We call S_0 the event that $b' = b^\star$.

Game 1: we modify the way to generate the common parameters (g_1, g_2), the public keys pk_0, pk_1 and the challenge ciphertext. Namely, the challenger $\mathcal{B}$ picks random values $x, y \in \mathbb{Z}_p^*$, $w_0, w_1 \xleftarrow{\$} \mathbb{Z}_p$ such that $w_0 + x \cdot w_1 \neq 0$ and defines $X = g_1^x$, $Y = g_1^y$, $T = g_1^{xy}$ as well as

$$\begin{aligned} g_2 &= Y & u_{0,1}^\star &= X & u_{0,2}^\star &= T \\ u_{1,1}^\star &= g_1^{w_0} \cdot X^{w_1} & u_{1,2}^\star &= Y^{w_0} \cdot T^{w_1} \end{aligned} \qquad \text{(G.3)}$$

Decryption queries are handled using sk_0 and sk_1 as in Game 0.

In the challenge phase, the challenger flips a coin $b^\star \xleftarrow{\$} \{0,1\}$, picks a random message $M^\star$ and computes

$$v_{b^\star}^\star = {u_{b^\star,1}^\star}^{x_{b^\star,1} + \alpha^\star \cdot y_{b^\star,1}} \cdot {u_{b^\star,2^\star}^\star}^{x_{b^\star,2} + \alpha^\star \cdot y_{b^\star,2}} \qquad \text{and} \qquad K^\star = \mathsf{KDF}\big(v_{b^\star}^\star\big), \qquad \text{(G.4)}$$

where $\alpha^\star = H(u_{b^\star,1}, u_{b^\star,2^\star})$. Everything else is calculated as in Game 0 and the challenge ciphertext is defined to be

$$C^\star = (u_{b^\star,1}, u_{b^\star,2^\star}, e^\star),$$

where $e^\star = E_{K^\star}(M^\star)$. Clearly, this change is just conceptual since $K^\star$ has the same distribution as in Game 0. Hence, $\Pr[S_1] = \Pr[S_0]$.

Game 2: is identical to Game 2 with the difference that the challenger $\mathcal{B}$ rejects all pre-challenge decryption queries $C = (u_1, u_2, e)$ such that $u_1 \in \{u_{0,1}^\star, u_{1,1}^\star\}$ or $u_2 \in \{u_{0,2}^\star, u_{1,2}^\star\}$. We call F_2 the event that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game 1.

Since elements $(u_{0,1}^\star, u_{0,2}^\star, u_{1,1}^\star, u_{1,2}^\star)$ are independent of $\mathcal{A}$'s view until the challenge phase, the probability of F_2 is at most $\Pr[F_2] \leq 4 \cdot q/p$, where q is the number of decryption queries. It comes that $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq 4 \cdot q/p$.

Game 3: in this game, we modify the distribution of the challenge ciphertext and define T as a random element of $\mathbb{G}$ instead of setting $T = g_1^{xy}$ as in Game 1 and 2. All other calculations (including the way to compute g_2 and $u_{0,1}^\star, u_{0,2}^\star, u_{1,1}^\star, u_{1,2}^\star$ in (G.3)) remain unchanged. Clearly, Game 3 is indistinguishable from Game 2 if the DDH assumption holds and $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\mathrm{DDH}}(\mathcal{B})$.

Game 4: is identical to Game 3 with the difference that the challenger $\mathcal{B}$ now rejects all post-challenge decryption queries $C = (u_1, u_2, e)$ such that $(u_1, u_2) \neq (u_1^\star, u_2^\star)$ but $\alpha = H(u_1, u_2) = H(u_1^\star, u_2^\star) = \alpha^\star$. If we call F_4 the probability to reject a ciphertext that would not have been rejected in Game 3, there exists a PPT collision-finding algorithm $\mathcal{B}^{\mathrm{CR}}$ such that $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4] \leq \mathbf{Adv}^{\mathrm{UOWHF}}(\mathcal{B}^{\mathrm{UOWHF}})$ and the two games are indistinguishable if H is a universal one-way hash function.

Game 5: in this game, the simulator $\mathcal{B}$ makes explicit use of the value $y = \log_{g_1}(g_2) \in \mathbb{Z}_p^*$, which can henceforth be used since we are done with the DDH assumption. In this game, $\mathcal{B}$ rejects all decryption queries $C = (u_1, u_2, e)$ such that $u_2 \neq u_1^y$. If F_5 denotes the probability of rejecting a ciphertext that would not have been rejected in Game 4, the same arguments as in [21] show that there exist PPT algorithms $\mathcal{B}'$ and $\mathcal{B}''$ such that $\Pr[F_5] \leq q \cdot (\mathbf{Adv}_{\mathcal{B}'}^{\mathsf{PRF\text{-}KDF}}(\lambda) + \mathbf{Adv}_{\mathcal{B}''}^{\mathsf{CT\text{-}INT}}(\lambda))$. If $b^\star \in \{0,1\}$ denotes the challenger $\mathcal{B}$'s random bit, the proof of [21] already implies that $\mathcal{O}_{sk_{b^\star}}(.)$ rejects pairs $C = (u_1, u_2, e)$ such that $u_2 \neq u_1^y$ with overwhelming probability in Game 4, unless the security of KDF or the ciphertext authentication property of $\Pi^{\text{sym-enc}}$ is broken. So, we only consider queries to oracle $\mathcal{O}_{sk_{1-b^\star}}(.)$ here.

For simplicity, we assume $b^\star = 0$ (the case $b^\star = 1$ can be handled in a completely similar way) and consider the probability that an invalid ciphertext ($C = (\tilde{u}_1, \tilde{u}_2, \tilde{e}) = (g_1^r, g_2^{r'}, \tilde{e})$) (namely, with $r \neq r'$) be accepted by $\mathcal{O}_{sk_1}(.)$. When such a decryption query is processed, the challenger $\mathcal{B}$ computes $\tilde{\alpha} = H(\tilde{u}_1, \tilde{u}_2)$ and $\tilde{v} = \tilde{u}_1^{x_{1,1}+\tilde{\alpha}\cdot y_{1,1}} \cdot \tilde{u}_2^{x_{1,2}+\tilde{\alpha}\cdot y_{1,2}} = g_1^{r\cdot(x_{1,1}+\tilde{\alpha}\cdot y_{1,1})+y\cdot r'\cdot(x_{1,2}+\tilde{\alpha}\cdot y_{1,2})}$ and attempts to decrypt $\tilde{e}$ using the symmetric key $\tilde{K} = \mathsf{KDF}(\tilde{v})$. The same "plug and pray" argument as in [21] show that the symmetric decryption must fail unless $\mathcal{A}$ was able to break either the security of KDF or to forge a valid symmetric encryption for $\Pi^{\text{sym-enc}}$ on a random-looking key $\tilde{K}$.

Indeed, if we consider what $\mathcal{A}$ knows about $sk_0 = (x_{0,1}, x_{0,2}, y_{0,1}, y_{0,2})$, $sk_1 = (x_{1,1}, x_{1,2}, y_{1,1}, y_{1,2})$ and about the value $\log_{g_1}(\tilde{v}) = r\cdot(x_{1,1}+\tilde{\alpha}\cdot y_{1,1})+y\cdot r'\cdot(x_{1,2}+\tilde{\alpha}\cdot y_{1,2})$, we observe that pk_0 reveals the right-hand-side member of the first two equations in the linear system (G.5). The challenge ciphertext is of the form $C^\star = (g_1^x, g_2^{x'}, e^\star)$, where $x, x' \in_R \mathbb{Z}_p^*$ and $e^\star = \mathsf{KDF}\big(g_1^{x\cdot(x_{0,1}+\alpha^\star y_{0,1})+x'\cdot y\cdot(x_{0,2}+\alpha^\star y_{0,2})}\big)$, which potentially leaks the right-hand-side member of the third equation. The fourth and fifth equations correspond to the information revealed by pk_1 but we easily check that $\mathcal{A}$ has no information about the RHS member of the last equation.

$$\begin{pmatrix} 1 & y & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & y & 0 & 0 & 0 & 0 \\ x & yx' & \alpha^\star x & \alpha^\star ry & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & y \\ 0 & 0 & 0 & 0 & r & yr' & \tilde{\alpha}r & \alpha' yr' \end{pmatrix} \begin{pmatrix} x_{0,1} \\ x_{0,2} \\ y_{0,1} \\ y_{0,2} \\ x_{1,1} \\ x_{1,2} \\ y_{1,1} \\ y_{1,2} \end{pmatrix} = \begin{pmatrix} \log_{g_1}(c_0) \\ \log_{g_1}(d_0) \\ \log_{g_1}(v^\star) \\ \log_{g_1}(c_1) \\ \log_{g_1}(d_1) \\ \log_{g_1}(\tilde{v}) \end{pmatrix} \qquad \text{(G.5)}$$

Indeed, as long as $r \neq r'$, the matrix in (G.5) is easily seen to be of full rank. Since the last row is linearly independent of other rows, $\log_{g_1}(\tilde{v})$ is independent of $\mathcal{A}$'s view. So, the only way for $\mathcal{A}$ to create a ciphertext that is rejected in Game 5 but would not have been in Game 4 is to imply a KDF-distinguisher $\mathcal{B}'$ or an algorithm $\mathcal{B}''$ forging a symmetric authenticated encryption: we can write

$$|\Pr[S_5] - \Pr[S_4]| \leq q \cdot (\mathbf{Adv}_{\mathcal{B}'}^{\mathsf{PRF\text{-}KDF}}(\lambda) + \mathbf{Adv}_{\mathcal{B}''}^{\mathsf{CT\text{-}INT}}(\lambda)).$$

In Game 5, we claim that $\Pr[S_5] = 1/2$. Indeed, since $T = g^{xy+\delta}$ for some $\delta \neq 0$, if we consider the value $v_{b^\star}^\star$ for both $b^\star = 0$ and $b^\star = 1$, we find that

$$v_0^\star = (c_0 \cdot d_0^{\alpha^\star})^x \cdot g_1^{\delta\cdot(x_{0,2}+\alpha_0^\star y_{0,2})}$$
$$v_1^\star = (c_1 \cdot d_1^{\alpha^\star})^{w_0+w_1 x} \cdot g_1^{\delta\cdot w_1\cdot(x_{1,2}+\alpha_1^\star y_{1,2})}$$

where $\alpha_0^\star = H(u_{0,1}^\star, u_{0,2}^\star)$ and $\alpha_1^\star = H(u_{1,1}^\star, u_{1,2}^\star)$. It is easy to see that the distributions of $v_0^\star$ and $v_1^\star$ are statistically indistinguishable from $\mathcal{A}$'s view since, given that all invalid ciphertexts are rejected by both oracles, $\mathcal{A}$ has no information about $(x_{0,2}, y_{0,2}, x_{1,2}, y_{1,2})$ whatsoever. $\square$

Finally, we prove the following lemma.

Lemma 10. *The KD* scheme is key-less reproducible.*

Proof. The proof is quite simple and it is similar to the proof of standard reproducibility (given in [8,7]) for the Cramer-Shoup cryptosystem. For any public parameters $par = (\mathbb{G}, g_1, g_2, H)$, any given public key $pk = (c, d) = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2})$ and any ciphertext $C = (u_1, u_2, e) = (g_1^r, g_2^r, \mathsf{E}_K(m))$, with $K = \mathsf{KDF}\big((c \cdot d^\alpha)^r\big)$ and $\alpha = H(u_1, u_2)$, the reproducibility algorithm R proceeds as follows.

```
$R\big(par, C, m', pk' = (c', d'), sk'\big)$:
    Parse $C$ as $(u_1, u_2, e)$ and return $\perp$ if $(u_1, u_2) \notin \mathbb{G}^2$.
    Parse $sk'$ as $(x_1', x_2', y_1', y_2') \in (\mathbb{Z}_p)^4$ and return $\perp$ if $(c', d') \neq \big(g_1^{x_1'} g_2^{x_2'}, g_1^{y_1'} g_2^{y_2'}\big)$.
    Compute $v' = u_1^{x_1' + \alpha \cdot y_1'} \cdot u_2^{x_2' + \alpha \cdot y_2'} \in \mathbb{G}$ and $K' = \mathsf{KDF}(v')$.
    Compute $e' = \mathsf{E}_{K'}(m')$ and return $(u_1, u_2, e')$.
```

It is straightforward that this algorithm satisfies the definition of key-less reproducibility. $\square$

H Definitions for Authenticated Encryption Schemes and Key Derivation Functions

A symmetric encryption scheme is specified by a pair (E, D), where E is the encryption algorithm and D is the decryption procedure, and a key space $\mathcal{K}(\lambda)$ where $\lambda \in \mathbb{N}$ is a security parameter. The security of authenticated symmetric encryption is defined by means of two games that capture the ciphertext indistinguishability and ciphertext (one-time) integrity properties.

Definition 12. *A symmetric encryption scheme is secure in the sense of authenticated encryption if any PPT adversary has negligible advantage in the following games.*

1. **The IND-SYM game.** For any PPT algorithm $\mathcal{A}$, the model considers the following game, where $\lambda \in \mathbb{N}$ is a security parameter:

```
$\boxed{\mathbf{Game}_{\mathcal{A}}^{\mathsf{IND\text{-}SYM}}(\lambda)}$
$K \xleftarrow{\$} \mathcal{K}(\lambda)$
$(m_0, m_1, s) \leftarrow \mathcal{A}(\mathsf{find}, \lambda)$
$d^\star \xleftarrow{\$} \{0,1\}$
$c^\star \leftarrow \mathsf{E}_K(m_{d^\star})$
$d \leftarrow \mathcal{A}(\mathsf{guess}, s, c^\star)$
return $1$ if $d = d^\star$ and $0$ otherwise.
```

   $\mathcal{A}$'s advantage is $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}SYM}}(\lambda) = |\Pr[\mathbf{Game}_{\mathcal{A}}^{\mathsf{IND\text{-}SYM}} = 1] - 1/2|$.

2. **The CT-INT game.** Let $\mathcal{A}$ be a PPT algorithm. We consider the following game, where $\lambda \in \mathbb{N}$ is a security parameter:

```
$\boxed{\mathbf{Game}_{\mathcal{A}}^{\mathsf{CT\text{-}INT}}(\lambda)}$
$K \xleftarrow{\$} \mathcal{K}(\lambda)$
$(m, s) \leftarrow \mathcal{A}(\mathsf{find}, \lambda)$
$c \leftarrow \mathsf{E}_K(m)$
$c' \leftarrow \mathcal{A}(\mathsf{create}, \lambda, c)$
return $1$ if $c' \neq c$ and $\mathsf{D}_K(c') \neq \perp$
       $0$ otherwise.
```

$\mathcal{A}$'s advantage is now defined as $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{CT\text{-}INT}}(\lambda) = \Pr[\mathbf{Game}_{\mathcal{A}}^{\mathsf{CT\text{-}INT}} = 1]$.

The notion of weak ciphertext integrity is defined in the same way but the adversary is not allowed to see an encryption c under the challenge key K.

One ingredient to construct hybrid encryption schemes is a Key Derivation Function (KDF) that allows deriving a key for the symmetric encryption scheme by hashing a group element. Given a symmetric encryption scheme (E, D) with key space $\mathcal{K}$ and a group generator $\mathcal{G}$, a KDF is specified by a pair $(\mathsf{Kg}, \mathsf{Hash})$, where Kg is the key generation algorithm for the KDF and Hash is the evaluation algorithm:

– given $\lambda \in \mathbb{N}$ a security parameter, Kg outputs a bit string dk_λ called a *derivation key*;
– given a derivation key dk_λ and a group element $X \in \mathbb{G}$ (where $\mathbb{G}$ is prime order group of cardinality $p > 2^\lambda$), Hash outputs an element in the key space $\mathcal{K}(\lambda)$.

Definition 13. *Let* $(\mathsf{Kg}, \mathsf{Hash})$ *be a Key Derivation Function. For any* $0/1$*-valued PPT algorithm* $\mathcal{A}$*, we the following game where* $\lambda \in \mathbb{N}$ *is a security parameter:*

```
$\boxed{\mathbf{Game}_{\mathcal{A}}^{\mathsf{PRF\text{-}KDF}}(\lambda)}$
$\mathbb{G} \xleftarrow{\$} \mathcal{G}(\lambda)$
$\mathsf{dk} \xleftarrow{\$} \mathsf{Kg}(\lambda)$
$X \xleftarrow{\$} \mathbb{G}$; $K_0 \leftarrow \mathsf{Hash}(\mathsf{dk}, X)$
$K_1 \xleftarrow{\$} \mathcal{K}(\lambda)$
$d^\star \xleftarrow{\$} \{0, 1\}$
$d \leftarrow \mathcal{A}(\mathsf{dk}, \mathbb{G}, K_{d^\star})$
return $1$ if $d = d^\star$ and $0$ otherwise.
```

$\mathcal{A}$'s advantage is $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{PRF\text{-}KDF}}(\lambda) = |\Pr[\mathbf{Game}_{\mathcal{A}}^{\mathsf{PRF\text{-}KDF}}(\lambda) = 1] - 1/2|$.

The KDF $(\mathsf{Kg}, \mathsf{Hash})$ *is secure if for all PPT algorithms* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *defined by the following experiment is a negligible function of* λ*.*