Robust Public-Key and Identity-Based Encryption

Michel Abdalla[1] Mihir Bellare[2] Chanathip Namprempre[3] Gregory Neven[4,5]

[1] Departement d'Informatique, École normale supérieure
45 Rue d'Ulm, 75230 Paris Cedex 05, France
Michel.Abdalla@ens.fr
http://www.di.ens.fr/users/mabdalla
[2] Department of Computer Science & Engineering, University of California San Diego
9500 Gilman Drive, La Jolla, California 92093, USA
mihir@cs.ucsd.edu
http://www.cs.ucsd.edu/users/mihir
[3] Electrical Engineering Department, Thammasat University
Klong Luang, Patumtani 12121, Thailand
cnamprem@engr.tu.ac.th
http://www.engr.tu.ac.th/~cnamprem
[4] Department of Electrical Engineering, Katholieke Universiteit Leuven
[5] IBM Zurich Research Laboratory
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
nev@zurich.ibm.com
http://www.neven.org

Abstract

We provide a provable-security treatment of the notion of a "robust" encryption scheme, namely one where the decryption algorithm rejects when the "wrong" secret key is used. We provide formal definitions of robustness under chosen-plaintext and chosen-ciphertext attacks (notions ROB-CPA and ROB-CCA). We find that contrary to what seems intuitive, robustness —at least in combination with privacy and anonymity as required by applications— is actually rarely if ever present, and obvious ways to confer it fail. We however provide general ways to efficiently confer ROB-CCA without sacrificing other security properties, for both public-key and identity-based encryption. We also show that a modified version of the Cramer-Shoup scheme is ROB-CCA. (The original scheme is not even ROB-CPA.) We present applications to auctions, searchable encryption and anonymous wireless communication, including the first PEKS scheme secure against chosen-ciphertext attacks in the standard model. We believe these results are important to clarify and help fill gaps in the literature arising from the implicit use of a robustness property that until now lacked formal definitions.

Keywords: Anonymity, identity-based encryption, searchable encryption.

1 Introduction

Suppose C is a public-key encryption (PKE) ciphertext obtained by encrypting a message M under a public key pk_0. We know that if C is decrypted using the secret key sk_0 corresponding to pk_0, the result would be M. But what if we decrypt C using a secret key sk_1 corresponding to a public key $pk_1 \neq pk_0$? Previous security notions for public-key encryption are silent about this. We will refer to a scheme as "robust" —this is a rough formulation that we will refine and strengthen later— if the result of this decryption is $\perp$, meaning the decryption algorithm rejects. For identity-based encryption (IBE), the notion is analogous; simply read "identity" in place of "public key" above.

Why should one care about robustness? Our interest in it was sparked by finding insecurities in several PKE or IBE-using protocols whose cause is ultimately a lack of robustness in the underlying

encryption scheme. Yet the works in question do not discuss, let alone define robustness. Rather, designers seem to have the intuition that robustness is always present. And in fact this intuition is quite natural. After all, what can you expect if you decrypt under the "wrong" key? As long as the scheme satisfies a strong enough notion of privacy like IND-CCA, one would think that the result, if not already $\perp$, would certainly be "garbage". Throw in some redundancy before encrypting, and we should be done. Robustness, in short, should be "easy".

This paper makes explicit the so-far implicit notion of robustness, defines it formally, and investigates provably achieving it. We find that contrary to the intuition presented above, robustness, at least in combination with privacy and anonymity as required by applications, is actually rarely if ever present, and obvious ways to confer it fail. We however provide ways to efficiently confer it without sacrificing other security properties. As a consequence, we obtain several new applications and results, in areas such as auctions, public-key encryption with keyword search (PEKS) [BDOP04, ABC$^+$05], and anonymous communication. But we believe that "naming" and provably achieving this so-far under-the-covers notion of robustness is important beyond this, from the point of view of clarifying and helping to fill gaps in the literature, and of making encryption more resistant to misuse.

Motivating applications. Before getting into specific applications or results, we believe we can give some general insight into where and why robustness is important. Briefly, we believe that robustness is an essential conjunct of *anonymous* encryption due to what we will call the *anonymous identification* problem. To explain, first recall that there are two main security requirements for encryption. The primary one is data privacy, as captured by notions IND-CPA or IND-CCA [GM84, RS92, DDN00, BDPR98, BF03]. However, we are seeing an increasing number of applications [Sak00, BDOP04, ABC$^+$05] that rely on the *anonymity* of the encryption scheme. The latter asks that a ciphertext does not reveal the public key or identity under which it was created and is captured by notions ANO-CPA and ANO-CCA [BBDP01, ABC$^+$05]. Our thesis, then, is that wherever one needs anonymity, one is likely to need robustness too.

As a canonical example, suppose a wireless base station is broadcasting messages to its subscribers. Each broadcast message has in fact a single intended recipient, but the broadcaster wishes to keep the identity of this recipient hidden from eavesdroppers. Anonymous encryption [BBDP01, ABC$^+$05] will provide the required privacy. But then, how is the intended recipient to know which ciphertext is intended for it? This is what we call the anonymous identification problem. The obvious (anonymity-preserving) solution is to include the public key or identity of the recipient in the plaintext and have recipients check this upon decryption. However, this does not work in general: there are (anonymous) encryption schemes where it is possible for an adversary to create a ciphertext that multiple recipients will accept. Robustness solves the problem. If the encryption scheme is robust, only the intended recipient will, upon decryption, obtain a valid result, meaning one different from $\perp$. (And the solutions discussed above are, in this light, simply attempts to add robustness that did not work, again showing that robustness is easily underestimated.) Thus, robust anonymous encryption enables anonymous broadcast.

Let us briefly discuss some other applications. Here the anonymous identification problem does not show up in a direct or obvious way, but we contend that, under the covers, that is really what is going on. First, we show that robustness of the underlying IBE scheme is sufficient for the PEKS scheme resulting from the IBE-to-PEKS transform of [BDOP04] to provide the consistency that was shown otherwise to be lacking by [ABC$^+$05]. Besides effectively validating the original transform of [BDOP04], this enables us to obtain the first IND-CCA PEKS schemes without random oracles [BR93]. Second, we present an attack on Sako's auction protocol [Sak00] that stems ultimately from the strong lack of robustness of ElGamal encryption. Our constructions fill the gap.

We remark that the above-mentioned applications require that the encryption scheme be not just

robust but also anonymous. Specifically, privacy of the PEKS scheme resulting from the IBE-to-PEKS transform of [BDOP04] requires anonymity of the underlying IBE scheme, as noted in [BDOP04] and formally proved in [ABC+05]. And in Sako's auction protocol [Sak00], a bid is an encryption of a *known* message under a public key related to the bid value, so anonymity of the encryption scheme is required for bid privacy. This supports our thesis that when anonymity is required, robustness often is too.

DEFINITION. The first step in a provable-security treatment of robustness, is of course, to provide a formal definition of robustness. Our definition in Section 3 actually requires something stronger than what we discussed above, namely that it is computationally infeasible for an adversary, given a pair of independently-generated public keys, to produce a ciphertext valid under both of them. In the IBE case, the role of the public keys is played by distinct identities of the adversary's choice. This strengthening is important for the security of the above applications. Our attack against Sako's protocol, for example, is based on a maliciously created ciphertext that decrypts correctly under *any* secret key. A rogue wireless base station could use similar ciphertexts to obtain the list of subscribers that are within reach; and when PEKS is used to filter encrypted email, such ciphertexts enable spammers to bypass all filters. As is typical for encryption, we define a pair of notions ROB-CPA and ROB-CCA.

ACHIEVING ROBUSTNESS. Robustness is trivially achieved by appending to the ciphertext the public key or identity of the intended recipient, and by checking for it upon decryption. This, however, is at the expense of anonymity. As we noted above, many applications require that the encryption scheme be not just robust but also anonymous (ANO-CPA or ANO-CCA). Given that we would also like to have the usual data privacy (IND-CPA or IND-CCA), we conclude that we need encryption schemes that provide the conjunction of the three properties, namely robustness, anonymity, and data privacy all together. We now approach the task of building such schemes, beginning by looking at obvious approaches and existing schemes.

One might think that robustness is implied by strong existing notions like IND-CCA, ANO-CCA or non-malleability [DDN00]. In Section 4, we show by counterexample that even IND-CCA+ANO-CCA (which implies non-malleabiblity [BDPR98, DDN00]) or plaintext awareness [BP04] do not imply robustness.

Conceding that robustness is not already present, one might think it could be obtained by adding redundancy to the encrypted data, in the form of a fixed constant or even the public key or identity, before encrypting. The decryption algorithm rejects if the redundancy is absent. The intuition here is that when one decrypts with the "wrong" secret key, the result is "random" and hence the redundancy would be garbled and absent. We show that this is false by presenting examples of schemes which fail to be robust even after redundancy is added. (Our examples are contrived but this is enough to illustrate that the methods in question do not work in general.) In fact, in Section 4 we show something even stronger and more general, namely that even addition of redundancy computed as any function of the public key and message fails to confer robustness.

Consideration of specific schemes —we restrict attention, for the reasons discussed above, to ones already known to be anonymous— also turns up bad news. There is no reason to expect schemes that are only IND-CPA + ANO-CPA to be robust since the decryption algorithm may never reject, so we focus on schemes that are IND-CCA and ANO-CCA. In the public-key domain, this mainly means the Cramer-Shoup (CS) scheme, shown to be IND-CCA in [CS03] and ANO-CCA in [BBDP01], and the DHIES scheme [ABR01], shown to be IND-CCA in [ABR01] and easily seen to be ANO-CCA. Simple attacks show that neither is even ROB-CPA. (We will however be able to prove ROB-CCA a simple modification of the CS scheme, see below. A similar modification of DHIES can be proved ROB-CCA as well.)

In Appendix B we show that neither of two popular IND-CCA-providing transforms, the Fujisaki-Okamoto (FO) transform [FO99] in the random oracle model and the Canetti-Halevi-Katz (CHK) transform [CHK04] in the standard model, yield robustness. Our counterexample for the FO transform actually shows that robustness is not even implied by the strong notion of plaintext-awareness. We note that the fact that neither of the transforms confers robustness generically does not exclude that they may still do so for certain specific schemes. We show that this is actually the case for the Boneh-Franklin IBE [BF01], which uses the FO transform to obtain IND-CCA security, and that it is *not* the case for the Boyen-Waters IBE [BW06], which uses the CHK transform. That is, the former is robust but the latter is not. The correctness definition of predicate encryption [KSW08] (which is a generalization of IBE) includes a weak notion of robustness that only considers honestly generated ciphertexts; the proposed scheme is *not* robust under our stronger definition.

Our transforms. We provide two general transforms that confer robustness on any given PKE and IBE scheme, respectively, without sacrificing privacy or anonymity. Namely, for any ATK $\in$ {CPA, CCA}, if the starting encryption scheme is IND-ATK and ANO-ATK, then the scheme resulting from our *commit-public-key transform* and *commit-identity transform* is IND-ATK, ANO-ATK and ROB-ATK.

The transforms use as a tool a (non-interactive) commitment scheme. Conferring ROB-CPA requires that the commitment scheme satisfy the usual hiding and binding properties. Conferring ROB-CCA requires that it additionally satisfy a weak form of non-malleability, but we show that the required property, which we call copy resistance, is easily obtained. The transform is efficient: using for example the commitment scheme of [DPP97], its overhead is just some symmetric cryptography (universal and collision-resistant hashing), and otherwise one can use a standard discrete-log-based commitment for an overhead of one exponentiation. Our transform and the above-mentioned instantiations do not use random oracles, but we note that commitment with a random oracle (RO) is trivial, so our transform is particularly efficient in the RO model.

As a result of these transforms we obtain the first IND-CCA+ANO-CCA+ROB-CCA PKE schemes (with or without ROs) and the first IND-CCA + ANO-CCA + ROB-CCA IBE schemes without ROs.

Robustness of $\mathcal{CS}^*$. As we indicated above, the Cramer-Shoup (CS) scheme [CS03] is not robust. We could make it so via our transform, but we can do better. We modify the scheme slightly, removing the pathological case of zero randomness that is the basis of the attack, and prove in Section 6 that the resulting $\mathcal{CS}^*$ scheme is ROB-CCA. Thus, in this case, robustness can be obtained at essentially zero added cost. The result assumes only security —specifically, pre-image resistance— of the underlying hash function; we do not assume DDH is hard. The proof combines ideas from the information-theoretic part of the proof of [CS03] with some new ideas.

Applications. PEKS schemes [BDOP04, ABC$^+$05] allow privacy-preserving filtering of encrypted email. Boneh, Di Crescenzo, Ostrovsky and Persiano [BDOP04] present a transform —we call it bdop-ibe-2-peks— that turns an IBE scheme into a PEKS scheme. The transform is IND-CPA-conferring: if the IBE scheme is ANO-CPA then the PEKS scheme is IND-CPA [BDOP04, ABC$^+$05]. However, Abdalla et al. [ABC$^+$05] noted that the PEKS scheme can lack *consistency*, meaning the filter can turn up false positives. They accordingly presented a modified transform new-ibe-2-peks that was not only IND-CPA-conferring but also provided consistency. We, however, return to the bdop-ibe-2-peks transform and show that the constructed PEKS scheme actually *is* consistent if the underlying IBE scheme is robust. Besides validating the bdop-ibe-2-peks transform from [BDOP04], this yields the first IND-CCA and consistent PEKS scheme without random oracles. This stems from something else we show, namely that the bdop-ibe-2-peks transform is (not only IND-CPA but also) IND-CCA-conferring: if the IBE scheme is ANO-CCA then the PEKS scheme is IND-CCA. (This is

not true of new-ibe-2-peks.) Now an IND-CCA and consistent PEKS scheme without random oracles can be obtained by for example starting from the Boyen-Waters ANO-CCA IBE scheme [BW06], applying our above-mentioned transform to make it robust, and then applying bdop-ibe-2-peks.

We also show how the lack of robustness in the ElGamal encryption scheme leads to lack of *fairness* in Sako's auction protocol [Sak00]. Our attack, presented in Appendix F, enables an adversary, with the help of a colluding auctioneer, to create a minimal winning bid, meaning a ciphertext representing a bid that is one dollar more than the highest encrypted bid. The hole can be plugged by instead using a robust encryption scheme.

As discussed above, another application is to enable broadcast encryption that is anonymous and yet allows recipients to unambiguously identify the ciphertexts intended for them.

2 Definitions

Notation and conventions. If x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. The empty string is denoted ε. By $a_1 \| \ldots \| a_n$, we denote a string encoding of $a_1, \ldots, a_n$ from which $a_1, \ldots, a_n$ are uniquely recoverable. (Usually, concatenation suffices.) By $a_1 \| \ldots \| a_n \leftarrow a$, we mean that a is parsed into its constituents $a_1, \ldots, a_n$. Similarly, if $a = (a_1, \ldots, a_n)$ then $(a_1, \ldots, a_n) \leftarrow a$ means we parse a as shown. Unless otherwise indicated, an algorithm may be randomized. By $y \xleftarrow{\$} A(x_1, x_2, \ldots)$ we denote the operation of running A on inputs $x_1, x_2, \ldots$ and fresh coins and letting y denote the output. We denote by $[A(x_1, x_2, \ldots)]$ the set of all possible outputs of A on inputs $x_1, x_2, \ldots$.

Public-key encryption. A *public-key encryption* (PKE) scheme is a tuple $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of algorithms. The parameter generation algorithm PG takes no input and returns common parameter *pars* that could be, for example, the description of a group common to all users. On input *pars*, the key generation algorithm KG produces a public key *pk* and private key *sk*. On inputs *pars*, *pk*, M, the encryption algorithm Enc produces a ciphertext C encrypting plaintext M. On input *pars*, *pk*, *sk*, C, the deterministic decryption algorithm Dec returns either a plaintext message M or $\perp$ to indicate that it rejects. (The inclusion of *pk* as an explicit input to Dec is somewhat unconventional but convenient in our case and without loss of generality since it can always be put in *sk* by KG.) In Appendix A, we recall the correctness definition and the advantage measures $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{PKE}}(\mathcal{A})$ and $\mathbf{Adv}^{\text{ano-atk}}_{\mathcal{PKE}}(\mathcal{A})$, where atk $\in \{\text{cpa}, \text{cca}\}$, which capture, respectively, privacy (ind) and anonymity (ano) of a PKE scheme $\mathcal{PKE}$ against chosen-plaintext (atk = cpa) and chosen-ciphertext (atk = cca) attacks. The notions are denoted IND-ATK and ANO-ATK for ATK $\in \{\text{CPA}, \text{CCA}\}$.

Identity-based encryption. An *identity-based encryption* (IBE) scheme is a tuple $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ of algorithms. The parameter generation algorithm Setup takes no input and returns common parameters *pars* and a master secret key *msk*. On input *pars*, *msk*, *id*, the extraction algorithm Ext produces a secret key *usk* for the user *id*. On input *pars*, *id*, M, the encryption algorithm Enc encrypts the plaintext M for the user *id*. On input *pars*, *id*, *usk*, C, the deterministic decryption algorithm Dec decrypts the ciphertext C and returns a message M or $\perp$ to indicate that it rejects. In Appendix A, we recall the correctness definition and advantage measures $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{IBE}}(\mathcal{A})$ and $\mathbf{Adv}^{\text{ano-atk}}_{\mathcal{IBE}}(\mathcal{A})$, where atk $\in \{\text{cpa}, \text{cca}\}$, which capture, respectively, privacy and anonymity of an IBE scheme $\mathcal{IBE}$ against chosen-plaintext (atk = cpa) and chosen-ciphertext (atk = cca) attacks. The notions are denoted IND-ATK and ANO-ATK for ATK $\in \{\text{CPA}, \text{CCA}\}$.

Games. Our definitions and proofs use the language of code-based game-playing [BR06]. Recall that a game —look at Figure 1 for an example— has an **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. A game G is executed with an adversary $\mathcal{A}$ as

proc Initialize
$pars \xleftarrow{\$} \mathsf{PG}$; $i \leftarrow 0$
$(pk_0, sk_0) \xleftarrow{\$} \mathsf{KG}(pars)$; $(pk_1, sk_1) \xleftarrow{\$} \mathsf{KG}(pars)$
Return $(pars, pk_0, pk_1)$

proc Finalize
Return Win

proc Dec(C)
$i \leftarrow i + 1$; $\boxed{\text{If } i \geq 2 \text{ Then return } \bot}$
$M_0 \leftarrow \mathsf{Dec}(pars, pk_0, sk_0, C)$
$M_1 \leftarrow \mathsf{Dec}(pars, pk_1, sk_1, C)$
If $(M_0 \neq \bot) \wedge (M_1 \neq \bot)$ Then Win $\leftarrow$ true
Return (M_0, M_1)

Figure 1: $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is a public-key encryption scheme. Game ROB-CPA$_{\mathcal{PKE}}$ contains the boxed code while Game ROB-CCA$_{\mathcal{PKE}}$ does not.

follows. First, **Initialize** executes and its outputs are the inputs to $\mathcal{A}$. Then $\mathcal{A}$ executes, its oracle queries being answered by the corresponding procedures of G. When $\mathcal{A}$ terminates, its output becomes the input to the **Finalize** procedure. The output of the latter, denoted $\mathrm{G}^{\mathcal{A}}$, is called the output of the game, and we let "$\mathrm{G}^{\mathcal{A}} \Rightarrow y$" denote the event that this game output takes value y. Boolean flags are assumed initialized to false.

3 Our Notions of Robustness

Robustness of PKE. As with other security notions for public-key encryption, robustness can be considered under chosen-plaintext attacks (CPA) or chosen-ciphertext attacks (CCA). Our definition uses the games of Figure 1. The **Initialize** procedure picks parameters *pars* and independent key pairs (pk_0, sk_1) and (pk_1, sk_1), and the values $pars, pk_0, pk_1$ it returns become the input to the adversary. The adversary can query the **Dec** oracle with a ciphertext C, and **Dec** returns the decryptions M_0, M_1 of C under sk_0 and sk_1, respectively. If both their decryptions are valid, then the flag Win is set, indicating that the adversary has won. The difference between Games ROB-CPA and ROB-CCA is that, in the former, *only one* **Dec** query is allowed. (This rule is enforced by the boxed code.) The **Finalize** procedure simply returns the value of the flag Win. For atk $\in \{\mathrm{cpa}, \mathrm{cca}\}$, we define the advantage of adversary $\mathcal{A}$ attacking PKE scheme $\mathcal{PKE}$ as

$$\mathbf{Adv}^{\text{rob-atk}}_{\mathcal{PKE}}(\mathcal{A}) = \Pr\left[\,\text{ROB-ATK}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \mathsf{true}\,\right] .$$

The corresponding notions are denoted ROB-CPA and ROB-CCA.

Discussion. Perhaps the first formulation of robustness one would come to defines the advantage of an adversary $\mathcal{A}$ as the probability that $\mathsf{Dec}(pars, pk_1, sk_1, \mathsf{Enc}(pars, pk_0, M)) \neq \bot$ where the probability is over

$$pars \xleftarrow{\$} \mathsf{PG}\ ;\ (pk_0, sk_0) \xleftarrow{\$} \mathsf{KG}(pars)\ ;\ (pk_1, sk_1) \xleftarrow{\$} \mathsf{KG}(pars)\ ;\ M \xleftarrow{\$} \mathcal{A}(pars, pk_0, pk_1)$$

and the coins of Enc. Our ROB-CPA notion implies this because our adversary could always compute its **Dec** query as an encryption under pk_0 of some message of its choice. Allowing the adversary to directly choose the ciphertext, however, yields a strictly stronger notion (there are schemes that are robust in this weaker sense but not ROB-CPA) and is important for applications. Specifically, our stronger notion of robustness is required for the fairness of Sako's auction protocol [Sak00] and to ensure correct recipient detection in the anonymous broadcast encryption example we discussed in Section 1.

One could consider a formulation in which the adversary gets to pick one or both of the public keys. But if so, there is nothing to really desire security-wise, since what happens under decryption

proc Initialize
$(pars, msk) \xleftarrow{\$} \mathsf{Setup}$; $i \leftarrow 0$
Return $pars$

proc Ext(id)
$\mathsf{USKT}[id] \xleftarrow{\$} \mathsf{Ext}(pars, msk, id)$
Return $\mathsf{USKT}[id]$

proc Finalize
Return Win

proc Dec(C, id_0, id_1)
$i \leftarrow i + 1$; $\boxed{\text{If } i \geq 2 \text{ Then return } \bot}$
If $\mathsf{USKT}[id_0] = \bot$ Then $\mathsf{USKT}[id_0] \xleftarrow{\$} \mathsf{Ext}(pars, msk, id_0)$
If $\mathsf{USKT}[id_1] = \bot$ Then $\mathsf{USKT}[id_1] \xleftarrow{\$} \mathsf{Ext}(pars, msk, id_1)$
$M_0 \leftarrow \mathsf{Dec}(pars, id_0, \mathsf{USKT}[id_0], C)$
$M_1 \leftarrow \mathsf{Dec}(pars, id_1, \mathsf{USKT}[id_1], C)$
If $(M_0 \neq \bot) \wedge (M_1 \neq \bot) \wedge (id_0 \neq id_1)$ Then Win $\leftarrow$ true
Return (M_0, M_1)

Figure 2: $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ is an IBE scheme. Game ROB-CPA$_{\mathcal{IBE}}$ contains the boxed code while Game ROB-CCA$_{\mathcal{IBE}}$ does not.

with an adversarially-chosen key affects only the adversary.

We note that while our constructions achieve the strong notions we have defined, the attacks and negative results show failure to meet even much weaker notions of robustness.

Robustness of IBE. The formalization is analogous to the above with identities playing the role of public keys, except that these identities are under adversarial control. Consider the games of Figure 2. The **Initialize** procedure picks parameters $pars$ and a master secret key msk, and the values it returns become the input to the adversary. The difference between Games ROB-CPA and ROB-CCA is that, in the former, only one **Dec** query is allowed. The adversary wins if the **Finalize** procedure returns true, meaning it makes a **Dec** query C valid under the secret keys of a pair id_0 and id_1 of distinct identities. For atk $\in \{\text{cpa}, \text{cca}\}$, we define the advantage of an adversary $\mathcal{A}$ attacking an IBE scheme $\mathcal{IBE}$ as

$$\mathbf{Adv}^{\text{rob-atk}}_{\mathcal{IBE}}(\mathcal{A}) = \Pr\left[\,\text{ROB-ATK}^{\mathcal{A}}_{\mathcal{IBE}} \Rightarrow \mathsf{true}\,\right].$$

The corresponding notions are again denoted ROB-CPA and ROB-CCA.

4 Adding redundancy before encryption fails

Towards understanding and achieving robustness, it is natural to first ask whether it is implied by existing notions, already present in existing schemes, or easily conferred by obvious transforms. The answer to all these questions is negative. Here, we illustrate what we think is the most interesting of the negative results, namely that adding even sophisticated forms of redundancy before encryption does not provide robustness. The discussion is for the PKE case, the IBE case is analogous. Other negative results, including that neither of the IND-CCA-providing Fujisaki-Okamoto (FO) [FO99] or Canetti-Halevi-Katz (CHK) [CHK04] transforms yield robustness, can be found in Appendix B. Since in fact the FO transform provides the stronger notion of plaintext-awareness, this also shows that even plaintext-awareness does not imply robustness.

Transforms. We are interested in transforming a given PKE scheme $\mathcal{PKE}$ into a robust PKE scheme $\overline{\mathcal{PKE}}$. If we are willing to sacrifice anonymity, this is easy: we simply append the public key to the ciphertext and have the decryption algorithm check for it. However, for the reasons mentioned earlier, we are only interested in transforms that preserve privacy and anonymity. For any ATK $\in \{\text{CPA}, \text{CCA}\}$, we say a transform is ATK-*security preserving* if, for any scheme that is IND-ATK+ANO-ATK, the scheme resulting from the transform is also IND-ATK+ANO-ATK. Then,

$\mathsf{RC}(pk\|M)$	$\mathsf{RV}(pk\|M, r)$
Return ε	Return 1
Return 0^k	Return $(r = 0^k)$
Return pk	Return $(r = pk)$
$K \xleftarrow{\$} \{0,1\}^k$; Return $K\|H(K, pk\|M)$	$K\|h \leftarrow r$; Return $(H(K, pk\|M) = h)$

Figure 3: Examples of redundancy codes, where the data x is of the form $pk\|M$.

we are interested in transforms that are CPA or CCA-preserving.

ADDING REDUNDANCY FAILS. A common perception is that if a ciphertext is encrypted under the "wrong" secret key, the resulting plaintext will be "random". This leads one to think that robustness is either implied by IND-CCA security, or can be easily conferred by adding redundancy before encrypting, and upon decryption rejecting if the redundancy is absent. The redundancy could take the form of a fixed constant, the public key, or even a hash of the message and the public key. We show that nothing like this works. In fact, we show more: that no *redundancy code* that is computed as a function of the public key and message works.

A redundancy code $\mathcal{RED} = (\mathsf{RC}, \mathsf{RV})$ is a pair of algorithms. On input x the redundancy computation algorithm RC returns redundancy r. Given x and claimed redundancy r, the redundancy verification algorithm RV returns 0 or 1. The consistency condition is that for all x we have $\mathsf{RV}(x, \mathsf{RC}(x)) = 1$ with probability one, where the probability is taken over the coins of RC. (We stress that the latter is allowed to be randomized.) Given a PKE scheme $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and a redundancy code $\mathcal{RED} = (\mathsf{RC}, \mathsf{RV})$, the *redundancy-adding transform* associates to them the PKE scheme $\overline{\mathcal{PKE}} = (\mathsf{PG}, \mathsf{KG}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ where $\overline{\mathsf{Enc}}(pars, pk, M) = \mathsf{Enc}(pars, pk, M\|\mathsf{RC}(pk\|M))$, and where $\overline{\mathsf{Dec}}$ parses the decrypted message as $M\|r$ and returns $\perp$ if $\mathsf{RV}(pk\|M, r) = 0$. Note the redundancy-adding transform is both CPA and CCA-security preserving, meaning preserves privacy and anonymity in the sense discussed above.

The first row of Figure 3 shows how $\mathcal{PKE}$ itself is a special case of this transform. The counterexample below therefore also shows that IND-CCA + ANO-CCA is not sufficient to imply ROB-CPA. The second and third rows show redundancy equal to a constant or the public key as examples of redundancy codes. The fourth row shows a randomized code where the redundancy is the hash of the public key and message under a key that is part of the redundancy so that it ends up encrypted. (The hash function could be a MAC or collision resistant.) Obviously, there are many other examples. Yet we now show that for any redundancy code there is an encryption scheme $\mathcal{PKE}$ such that the scheme $\overline{\mathcal{PKE}}$ resulting from the redundancy-adding transform is not even ROB-CPA. We build $\mathcal{PKE} = (\mathsf{PG}^*, \mathsf{KG}^*, \mathsf{Enc}, \mathsf{Dec})$ by modifying a given encryption scheme $\mathcal{PKE}^* = (\mathsf{PG}^*, \mathsf{KG}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$. Let $l(|x|)$ be the number of coins used by RC on input x, and let $\mathsf{RC}(x; \omega)$ denote the result of executing RC on input x with coins $\omega \in \{0,1\}^{l(|x|)}$. Below, M^* is a fixed message in the message space of $\mathcal{PKE}^*$:

Algorithm $\mathsf{Enc}(pars, pk, M)$	Algorithm $\mathsf{Dec}(pars, pk, sk, C)$
$C \xleftarrow{\$} \mathsf{Enc}^*(pars, pk, M)$	$b\|\|C^* \leftarrow C$
Return $1\|\|C$	If $b = 1$ Then return $\mathsf{Dec}^*(pars, pk, sk, C^*)$
	Else return $M^*\|\mathsf{RC}(pk\|M^*; 0^{l(\|pk\|M^*\|)})$

The reason we used $0^{l(|pk\|M^*|)}$ as coins for RC above is that Dec is required to be deterministic. Now, apply the redundancy-adding transform to $\mathcal{PKE}$ and $\mathcal{RED}$ to obtain $\overline{\mathcal{PKE}}$. Observe that when $\overline{\mathsf{Dec}}$ is applied to inputs $pars, pk, sk, 0\|0$, it first computes $\mathsf{Dec}(pars, pk, sk, 0\|0)$, the result of which is $M^*\|r$ where $r = \mathsf{RC}(pk\|M^*; 0^{l(|pk\|M^*|)})$, and then checks whether $\mathsf{RV}(pk\|M^*, r)$ equals 1. But the

consistency of $\mathcal{RED}$ tells us that this check will always be true, so $\overline{\mathsf{Dec}}$ returns M^*. But this is true for any $pars, pk, sk$, so the adversary that, given $pars, pk_0, pk_1$, makes **Dec** query $0\|0$ and halts, wins Game ROB-CPA$_{\overline{\mathcal{PKE}}}$ with probability one. We have shown that $\overline{\mathcal{PKE}}$ is not ROB-CPA, as desired.

5 General transforms that provide ROB-CCA

We present transforms that confer robustness on any encryption scheme while preserving privacy and anonymity. More precisely, for any ATK $\in \{\text{CPA}, \text{CCA}\}$, our transform turns any IND-ATK + ANO-ATK encryption scheme into one that continues to be IND-ATK + ANO-ATK but is additionally ROB-CCA. We present two transforms, one for PKE schemes and the other for IBE schemes. The underlying idea is the same for both transforms, with the only difference being that identities play the role of public keys in the IBE case. Each transform uses a commitment scheme.

COMMITMENT SCHEMES. A non-interactive commitment scheme is a 3-tuple $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$. The parameter generation algorithm CPG returns public parameters *cpars*. The commital algorithm Com takes *cpars* and data x as input and returns a commitment *com* to x along with a decommital key *dec*. The deterministic decommital algorithm Open takes *cpars*, *com*, and *dec* as input and returns x or $\perp$ to indicate that it rejects. The consistency definition and the advantage measures $\mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{A})$ and $\mathbf{Adv}^{\text{bind}}_{\mathcal{CMT}}(\mathcal{A})$, referring to the standard hiding and binding properties, are recalled in Appendix A. We refer to the corresponding notions as HIDE and BIND.

THE COMMIT-PUBLIC-KEY TRANSFORM. The idea is for the ciphertext to include a commitment to the public key. The commitment is *not* encrypted, but the decommital key along with a random string R chosen once as a parameter of the scheme are encrypted. In detail, given an encryption scheme $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and a commitment scheme $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$, the *commit-public-key transform* associates to them the encryption scheme $\overline{\mathcal{PKE}} = (\overline{\mathsf{PG}}, \overline{\mathsf{KG}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ where $\overline{\mathsf{PG}}$ generates parameters *pars* and *cpars* from the base schemes $\mathcal{PKE}$ and $\mathcal{CMT}$, respectively, chooses a random string $R \xleftarrow{\$} \{0,1\}^r$ (where r is a security parameter), and outputs $(pars, cpars, R)$. $\overline{\mathsf{KG}}$ simply outputs a public-secret key pair (pk, sk) output by $\mathsf{KG}(pars)$. The encryption and decryption algorithms are as follows:

Algorithm $\overline{\mathsf{Enc}}(\overline{pars}, pk, \overline{M})$	Algorithm $\overline{\mathsf{Dec}}(\overline{pars}, pk, sk, \overline{C})$
$(pars, cpars, R) \leftarrow \overline{pars}$	$(pars, cpars, R) \leftarrow \overline{pars}$; $(C, com) \leftarrow \overline{C}$
$(com, dec) \xleftarrow{\$} \mathsf{Com}(cpars, pk)$	$(M, dec, R') \leftarrow \mathsf{Dec}(pars, pk, sk, C)$
$C \xleftarrow{\$} \mathsf{Enc}(pars, pk, (\overline{M}, dec, R))$	If $\mathsf{Open}(cpars, com, dec) = pk$ and $R' = R$
Return (C, com)	Then Return M Else Return $\perp$

We first fully state the results then explain each of them in turn. Full proofs can be found in Appendix C; we merely highlight some notable aspects here.

Proposition 5.1 *Let $\mathcal{PKE}$ be a PKE scheme, let $\mathcal{CMT}$ be a commitment scheme, and let $\overline{\mathcal{PKE}}$ be the PKE scheme obtained by applying the commit-public-key transform. For any adversary $\mathcal{A}$ running in time t against $\overline{\mathcal{PKE}}$, there exist adversaries $\mathcal{B}$ and $\mathcal{B}_1$–$\mathcal{B}_4$ such that*

$$\mathbf{Adv}^{\text{rob-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \le \mathbf{Adv}^{\text{bind}}_{\mathcal{CMT}}(\mathcal{B}) + \mathbf{Coll}_{\mathcal{PKE}} \quad (1)$$

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \le \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{PKE}}(\mathcal{B}) \quad (2)$$

$$\mathbf{Adv}^{\text{ano-cpa}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \le \mathbf{Adv}^{\text{ano-cpa}}_{\mathcal{PKE}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{PKE}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_3) \quad (3)$$

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \le 2 \cdot \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE}}(\mathcal{B}_2) \quad (4)$$

$$\mathbf{Adv}^{\text{ano-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \le 2 \cdot \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE}}(\mathcal{B}_2) + 2 \cdot \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_3) + \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{PKE}}(\mathcal{B}_4) + \frac{2|M^*|}{2^r} \quad (5)$$

proc Initialize
$cpars \xleftarrow{\$} \mathsf{CPG}$; Return $cpars$

proc Com(x)
$x^* \leftarrow x$; $(com, dec) \xleftarrow{\$} \mathsf{Com}(cpars, x)$; Return (com, dec)

proc Finalize(com')
Return $(com \neq com' \wedge \mathsf{Open}(cpars, com', dec) = x^*)$

Figure 4: Let $\mathcal{CMT}$ be a commitment scheme. Game $\mathrm{COPY}_{\mathcal{CMT}}$ captures the copy resistance property of the scheme. An adversary $\mathcal{A}$ must call the **Com** oracle exactly once.

---

where $|M^*|$ is the length of the challenge message chosen by $\mathcal{A}$. Furthermore, each adversary $\mathcal{B}$, $\mathcal{B}_1$–$\mathcal{B}_4$ runs in time at most t plus the time for $O(t)$ executions of the algorithms of $\mathcal{PKE}$ and $\mathcal{CMT}$.

Equation (1) says that the transform provides ROB-CCA as long as the commitment scheme is BIND-secure and the base encryption scheme $\mathcal{PKE}$ has low public-key collision probability. Explicitly, the latter can be defined as

$$\mathbf{Coll}_{\mathcal{PKE}} \;=\; \Pr\left[\; pk_0 = pk_1 \;\;:\;\; pars \xleftarrow{\$} \mathsf{PG}\,;\; (pk_0, sk_0) \xleftarrow{\$} \mathsf{KG}(pars)\,;\; (pk_1, sk_1) \xleftarrow{\$} \mathsf{KG}(pars) \;\right] .$$

It is easy to see that $\mathcal{PKE}$ being IND-CPA implies $\mathbf{Coll}_{\mathcal{PKE}}$ is negligible, so asking for low public-key collision probability is in fact not an extra assumption. The reason we made it explicit is that for most schemes it is unconditionally low. For example, for ElGamal, it is $1/|\mathbb{G}|$ where $\mathbb{G}$ is the group being used. Also, it can always be made smaller than or equal to 2^{-k} by adding a random k-bit string to the public key. (We could have had our transform do this and then have not needed to talk about this collision probability, but the current transform has the nice property that it leaves public keys unchanged, so that new certificates are not needed if this transform is used.)

Equation (2) says that, if the base scheme $\mathcal{PKE}$ is IND-CPA, then so is $\overline{\mathcal{PKE}}$, without any assumptions on $\mathcal{CMT}$. Equation (3) says that, if $\mathcal{PKE}$ is ANO-CPA + IND-CPA, then $\overline{\mathcal{PKE}}$ is ANO-CPA. For $\overline{\mathcal{PKE}}$ to be ANO-CPA, the reason we need $\mathcal{PKE}$ to be IND-CPA in addition to ANO-CPA is that the decommitment is encrypted, and, if it is revealed, anonymity is lost.

Equations (4) and (5) specify the security assumptions of the base schemes for the transform to be CCA-preserving, and require a non-standard security assumption from the commitment scheme that we call *copy resistance*. This is captured by defining the advantage of an adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathrm{copy}}_{\mathcal{CMT}}(\mathcal{A}) = \Pr\left[\,\mathrm{COPY}^{\mathcal{A}}_{\mathcal{CMT}} \Rightarrow \mathsf{true}\,\right] ,$$

where $\mathrm{COPY}_{\mathcal{CMT}}$ is the game of Figure 4. Thus, copy resistance requires that given an honestly-generated commitment (com, dec) of an adversarially-chosen message x^*, it is computationally infeasible to find $com' \neq com$ such that (com', dec) opens to x^*.

This property is necessary, because without it the transform may not be CCA-preserving. To see this, consider an IND-CCA adversary $\mathcal{A}$ that receives a challenge ciphertext (C^*, com^*), where C^* is an encryption of (M_b^*, dec^*, R) for some $b \in \{0, 1\}$. If it can come up with a commitment $com' \neq com^*$ such that $\mathsf{Open}(cpars, com', dec^*) = pk$, then (C^*, com') is a valid ciphertext that decrypts to M_b. It could therefore determine b by submitting this cipertext to the decryption oracle. To prevent this attack, it would suffice for the commitment scheme to be non-malleable [DDN00]. Although such schemes exist [FF00, DKOS01, Di 02], we observe that we can get away with a much weaker property, namely copy-resistance.

Finding appropriate commitment schemes. Ignore copy-resistance to begin with. Then there are many suitable commitment schemes. For example, one can use Pedersen commitments [Ped92], where *cpars* is a pair of random generators of a group of G of prime order p. The committal to x is $g^{H(x)}h^{dec}$, where $dec \xleftarrow{\$} Z_p$ is the decommital and H is a collision-resistant hash function with

range $\mathbb{Z}_p$. This is binding if the discrete log problem is hard and provides unconditional privacy. This scheme is simple and efficient for practical use. An even more efficient scheme, using only universal and collision-resistant hashing, is that of [DPP94]. On the theoretical side, commitment schemes of this nature exist given any one-way function [Nao90]. Naor [Nao90] presents his scheme as interactive, with the receiver first sending a random string and the sender computing the commitment as a function of it. But one can put the receiver move into *cpars* and get a non-interactive scheme of the type we want. None of these instantiations involve random oracles.

Getting COPY security is easy, which is the advantage of working with this weak form of non-malleability. Let us say a commitment scheme has the *uniqueness* property if (1) when Com returns (com, dec) on input $(cpars, x)$, the decommital dec is exactly the coins used by Com, and (2) $\mathsf{Open}(cpars, com, x, dec)$ runs Com on inputs $(cpars, x)$ with coins dec, returning 1 if the result is (com, dec) and 0 otherwise. Most schemes in fact have this uniqueness property, and, if not, can always be modified to do so by using the coins of the committal as the decommital. (This retains HIDE and BIND security.) However, any scheme with the uniqueness property is COPY secure because there is only one possible committal com corresponding to data x and decommital dec. In particular, all of the above-mentioned schemes have the uniqueness property. So we get efficient transforms that confer ROB-CCA while being CPA and CCA-preserving, and we also get such transforms assuming only the existence of a one-way function.

Note from the term $2|M^*|/2^r$ in Equation (5) that the preservation of ANO-CCA crucially relies on the presence of the random string R in the encrypted data. To understand where this need comes from, consider what happens when R is not included in the transform, and when the following PKE and commitment scheme are used. First imagine that the decryption algorithm of $\mathcal{PKE}$ returns a fixed string of the form $(\hat{M}, \hat{dec})$ whenever the wrong key is used to decrypt. Moreover, imagine $\mathcal{CMT}$ is such that it is easy to, for given $cpars, x, dec$, find com so that $\mathsf{Open}(cpars, com, dec) = x$. (For example, any commitment scheme where dec are the coins used by the Com algorithm.) Consider then the ANO-CCA adversary $\mathcal{A}$ against the transformed scheme that, after receiving the challenge ciphertext (C^*, com^*), creates a commitment $\hat{com}$ of pk_1 with opening information $\hat{dec}$, and queries $(C^*, \hat{com})$ to be decrypted under pk_0. Let C^* be generated as $\mathsf{Enc}(pars, pk_b, (M^*, dec^*))$ for a hidden bit $b \in \{0, 1\}$. If $b = 0$ this query will return $\perp$ because most likely $\mathsf{Open}(cpars, \hat{com}, dec^*) \neq pk_0$; if $b = 1$ it returns $\hat{M}$, allowing $\mathcal{A}$ to distinguish the value of b.

The inclusion of R in the plaintext thwarts this attack, because the decryption algorithm cannot "default" to a string that contains R. Proving that this measure is sufficient is a bit delicate however. As part of the proof, we introduce an information-theoretic lemma that is used to bound the amount of information about R that the adversary can "slip into" the challenge ciphertext. See Appendix C for details.

THE COMMIT-IDENTITY TRANSFORM. We propose the robustness-conferring *commit-identity transform* for IBE schemes. It is analogous to the commit-public-key transform, but the user's identity plays the role of the public key. It is given in detail in Figure 5. The following proposition states that it provides robustness and is both IND-ATK and ANO-ATK-preserving. The proof is largely analogous to that of Proposition 5.1 and can be found in Appendix D.

Proposition 5.2 *Let $\mathcal{IBE}$ be an IBE scheme with identity space IDSp, let $\mathcal{CMT}$ be a commitment scheme, and let $\overline{\mathcal{IBE}}$ be the IBE scheme obtained by applying the commit-identity transform. Given*

Algorithm $\overline{\mathsf{Setup}}$
$(pars, msk) \xleftarrow{\$} \mathsf{Setup}$; $cpars \xleftarrow{\$} \mathsf{CPG}$
$R \xleftarrow{\$} \{0,1\}^r$; $\overline{pars} \leftarrow (pars, cpars, R)$
Return $(\overline{pars}, msk)$

Algorithm $\overline{\mathsf{Enc}}(\overline{pars}, id, \overline{M})$
$(pars, cpars, R) \leftarrow \overline{pars}$
$(com, dec) \xleftarrow{\$} \mathsf{Com}(cpars, id)$
$C \xleftarrow{\$} \mathsf{Enc}(pars, id, (\overline{M}, dec, R))$
Return (C, com)

Algorithm $\overline{\mathsf{Ext}}(\overline{pars}, msk, id)$
$(pars, cpars, R) \leftarrow \overline{pars}$
$usk \xleftarrow{\$} \mathsf{Ext}(pars, msk, id)$
Return usk

Algorithm $\overline{\mathsf{Dec}}(\overline{pars}, id, usk, \overline{C})$
$(pars, cpars, R) \leftarrow \overline{pars}$; $(C, com) \leftarrow \overline{C}$
$M \leftarrow \mathsf{Dec}(pars, id, usk, C)$; $(\overline{M}, dec, \overline{R}) \leftarrow M$
If $\mathsf{Open}(cpars, com, dec) = id$ and $\overline{R} = R$
Then Return $\overline{M}$ Else Return $\perp$

Figure 5: $\overline{\mathcal{IBE}} = (\overline{\mathsf{Setup}}, \overline{\mathsf{Ext}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ resulting from applying our commit-identity transform to $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$. Above, $r \geq 0$ is an integer.

any adversary $\mathcal{A}$ running in time t against $\overline{\mathcal{IBE}}$, we can construct adversaries $\mathcal{B}$ and $\mathcal{B}_1$–$\mathcal{B}_4$ such that

$$\mathbf{Adv}^{\text{rob-cca}}_{\overline{\mathcal{IBE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{bind}}_{\mathcal{CMT}}(\mathcal{B}) \quad (6)$$

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{IBE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{IBE}}(\mathcal{B}) \quad (7)$$

$$\mathbf{Adv}^{\text{ano-cpa}}_{\overline{\mathcal{IBE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ano-cpa}}_{\mathcal{IBE}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{IBE}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_3) \quad (8)$$

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{IBE}}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE}}(\mathcal{B}_2) \quad (9)$$

$$\begin{aligned}\mathbf{Adv}^{\text{ano-cca}}_{\overline{\mathcal{IBE}}}(\mathcal{A}) \leq\ & 2 \cdot \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBE}}(\mathcal{B}_2) + 2 \cdot \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{B}_3) \\ & + \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_4) + \min\left(\frac{2(q_{\text{id}} + q_{\text{dec}})}{|\mathsf{IDSp}|} + \frac{2|M^*|}{2^r}\ ,\ \frac{2 \cdot |\mathsf{IDSp}| \cdot |\mathsf{M}^*|}{2^r}\right) \quad (10)\end{aligned}$$

*where q_{id} and q_{dec} are $\mathcal{A}$'s number of queries to the **Ext** and **Dec** oracles, respectively, and M^* is the challenge message chosen by $\mathcal{A}$. Furthermore, each adversary $\mathcal{B}$ and $\mathcal{B}_1$–$\mathcal{B}_4$ runs in time at most t plus the time for $O(t)$ executions of the algorithms of $\mathcal{IBE}$ and $\mathcal{CMT}$.*

The minimum in Equation (10) shows that our transform works regardless whether the identity space is small, large, or even infinite. The first bound is tightest when $|\mathsf{IDSp}| \gg \mathsf{q}_{\text{id}} + \mathsf{q}_{\text{dec}}$, the second is tightest when $|\mathsf{IDSp}| \ll 2^{\mathsf{r}}$. For small identity spaces one can always choose r large enough so that the latter is negligible.

6 A ROB-CCA version of Cramer-Shoup

Let $\mathbb{G}$ be a group of prime order p, and H: $\mathsf{Keys}(H) \times \mathbb{G}^3 \to \mathbb{G}$ a family of functions. We assume $\mathbb{G}, p, H$ are fixed and known to all parties. Figure 6 shows the Cramer-Shoup (CS) scheme and the variant $\mathcal{CS}^*$ scheme where $\mathbf{1}$ denotes the identity element of $\mathbb{G}$. The differences are boxed. Recall that the CS scheme was shown to be IND-CCA in [CS03] and ANO-CCA in [BBDP01]. However, for any message $M \in \mathbb{G}$ the ciphertext $(\mathbf{1}, \mathbf{1}, M, \mathbf{1})$ in the CS scheme decrypts to M under *any* $pars, pk$, and sk, meaning in particular that the scheme is not even ROB-CPA. The modified scheme $\mathcal{CS}^*$ —which continues to be IND-CCA and ANO-CCA— removes this pathological case by having Enc choose the randomness u to be non-zero —Enc draws u from $\mathbb{Z}_p^*$ while the CS scheme draws it from $\mathbb{Z}_p$— and then

Algorithm PG
$K \xleftarrow{\$} \mathsf{Keys}(H)$; $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$
$g_2 \leftarrow g_1^w$; Return (g_1, g_2, K)

Algorithm $\mathsf{Enc}((g_1, g_2, K), (e, f, h), M)$
$u \xleftarrow{\$} \mathbb{Z}_p^{\boxed{*}}$
$a_1 \leftarrow g_1^u$; $a_2 \leftarrow g_2^u$; $b \leftarrow h^u$
$c \leftarrow b \cdot M$; $v \leftarrow H(K, (a_1, a_2, c))$
$d \leftarrow e^u f^{uv}$; Return (a_1, a_2, c, d)

Algorithm $\mathsf{KG}(g_1, g_2, K)$
$x_1, x_2, y_1, y_2, z_1, z_2 \xleftarrow{\$} \mathbb{Z}_p$
$e \leftarrow g_1^{x_1} g_2^{x_2}$; $f \leftarrow g_1^{y_1} g_2^{y_2}$; $h \leftarrow g_1^{z_1} g_2^{z_2}$
Return $((e, f, h), (x_1, x_2, y_1, y_2, z_1, z_2))$

Algorithm $\mathsf{Dec}((g_1, g_2, K), (e, f, h), (x_1, x_2, y_1, y_2, z_1, z_2), C)$
$(a_1, a_2, c, d) \leftarrow C$; $v \leftarrow H(K, (a_1, a_2, c))$; $M \leftarrow c \cdot a_1^{-z_1} a_2^{-z_2}$
If $d \neq a_1^{x_1+y_1 v} a_2^{x_2+y_2 v}$ Then $M \leftarrow \perp$
$\boxed{\text{If } a_1 = \mathbf{1} \text{ Then } M \leftarrow \perp}$
Return M

Figure 6: The original CS scheme [CS03] does not contain the boxed code while the variant $\mathcal{CS}^*$ does. Although not shown above, the decryption algorithm in both versions always checks to ensure that the ciphertext $C \in \mathbb{G}^4$. The message space is $\mathbb{G}$.

having Dec reject (a_1, a_2, c, d) if $a_1 = \mathbf{1}$. This thwarts the attack, but is there any other attack? We show that there is not by proving that $\mathcal{CS}^*$ is actually ROB-CCA. Our proof of robustness relies only on the security —specifically, pre-image resistance— of the hash family H: it does not make the DDH assumption. Our proof combines ideas from the information-theoretic part of the proof of [CS03] with some new ideas.

We say that a family H: $\mathsf{Keys}(H) \times \mathsf{Dom}(H) \to \mathsf{Rng}(H)$ of functions is *pre-image resistant* if, given a key K and a *random* range element v^*, it is computationally infeasible to find a pre-image of v^* under $H(K, \cdot)$. The notion is captured formally by the following advantage measure for an adversary $\mathcal{I}$:

$$\mathbf{Adv}^{\text{pre-img}}_H(\mathcal{I}) = \Pr\left[\, H(K, x) = v^* \;\; : \;\; K \xleftarrow{\$} \mathsf{Keys}(H)\,;\; v^* \xleftarrow{\$} \mathsf{Rng}(H)\,;\; x \xleftarrow{\$} \mathcal{I}(K, v^*) \,\right] .$$

Pre-image resistance is not implied by the standard notion of one-wayness, since in the latter the target v^* is the image under $H(K, \cdot)$ of a random domain point, which may not be a random range point. However, it seems like a fairly mild assumption on a practical cryptographic hash function and is implied by the notion of "everywhere pre-image resistance" of [RS04], the difference being that, for the latter, the advantage is the maximum probability over all $v^* \in \mathsf{Rng}(H)$. The following implies that $\mathcal{CS}^*$ is ROB-CCA if H is pre-image resistant:

Theorem 6.1 *Given any adversary $\mathcal{A}$ against $\mathcal{CS}^*$ running in time t and making q **Dec** queries, we can construct an adversary $\mathcal{I}$ such that*

$$\mathbf{Adv}^{\text{rob-cca}}_{\mathcal{CS}^*}(\mathcal{A}) \;\le\; \mathbf{Adv}^{\text{pre-img}}_H(\mathcal{I}) + \frac{2q+1}{p} \,. \tag{11}$$

Furthermore, the running time of $\mathcal{I}$ is $t + q \cdot O(t_{\text{exp}})$ where t_{exp} denotes the time for one exponentiation in $\mathbb{G}$.

A detailed proof of Theorem 6.1 is in Appendix E. Here we sketch some intuition. Let $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$ be the parameters and public keys generated by Game ROB-CCA$_{\mathcal{CS}^*}$, and let $(x_{01}, x_{02}, y_{01}, y_{02}, z_{01}, z_{02})$ and $(x_{11}, x_{12}, y_{11}, y_{12}, z_{11}, z_{12})$ be the corresponding secret keys. Suppose $\mathcal{A}$ makes a **Dec** query (a_1, a_2, c, d). Then the code of the decryption algorithm Dec from Figure 6 tells us that, for this to be a winning query, it must be that

$$d \;=\; a_1^{x_{01}+y_{01}v} a_2^{x_{02}+y_{02}v} \;=\; a_1^{x_{11}+y_{11}v} a_2^{x_{12}+y_{12}v}$$

where $v = H(K, (a_1, a_2, c))$. Letting $u_1 = \log_{g_1}(a_1), u_2 = \log_{g_2}(a_2)$ and $s = \log_{g_1}(d)$, we have

$$s \;=\; u_1(x_{01} + y_{01}v) + wu_2(x_{02} + y_{02}v) \;=\; u_1(x_{11} + y_{11}v) + wu_2(x_{12} + y_{12}v) \tag{12}$$

However, even acknowledging that $\mathcal{A}$ knows little about $x_{b1}, x_{b2}, y_{b1}, y_{b2}$ ($b \in \{0,1\}$) through its **Dec** queries, it is unclear why Equation (12) is prevented by pre-image resistance —or in fact any property short of being a random oracle— of the hash function H. In particular, there seems no way to "plant" a target v^* as the value v of Equation (12) since the adversary controls u_1 and u_2. However, suppose now that $a_2 = a_1^w$. (We will discuss later why we can assume this.) This implies $wu_2 = wu_1$ or $u_2 = u_1$ since $w \neq 0$. Now from Equation (12) we have

$$u_1(x_{01} + y_{01}v) + wu_1(x_{02} + y_{02}v) - u_1(x_{11} + y_{11}v) - wu_1(x_{12} + y_{12}v) \;=\; 0\,.$$

We now see the value of enforcing $a_1 \neq 1$, since this implies $u_1 \neq 0$. After canceling u_1 and re-arranging terms, we have

$$v(y_{01} + wy_{02} - y_{11} - wy_{12}) + (x_{01} + wx_{02} - x_{11} - wx_{12}) \;=\; 0\,. \tag{13}$$

Given that $x_{b1}, x_{b2}, y_{b1}, y_{b2}$ ($b \in \{0,1\}$) and w are chosen by the game, there is at most one solution v (modulo p) to Equation (13). We would like now to design $\mathcal{I}$ so that on input K, v^* it chooses x_{b1}, x_{b2}, y_{b1}, y_{b2} ($b \in \{0,1\}$) so that the solution v to Equation (13) is v^*. Then (a_1, a_2, c) will be a pre-image of v^* which $\mathcal{I}$ can output.

To make all this work, we need to resolve two problems. The first is why we may assume $a_2 = a_1^w$ —which is what enables Equation (13)— given that a_1, a_2 are chosen by $\mathcal{A}$. The second is to properly design $\mathcal{I}$ and show that it can simulate $\mathcal{A}$ correctly with high probability. To solve these problems, we consider, as in [CS03], a modified check under which decryption, rather than rejecting when $d \neq a_1^{x_1+y_1v} a_2^{x_2+y_2v}$, rejects when $a_2 \neq a_1^w$ or $d \neq a_1^{x+yv}$, where $x = x_1 + wx_2$, $y = y_1 + wy_2$, $v = H(K, (a_1, a_2, c))$ and (a_1, a_2, c, d) is the ciphertext being decrypted. In our proof in Appendix E, games G_0–G_2 move us towards this perspective. Then, we fork off two game chains. Games G_3–G_6 are used to show that the modified decryption rule increases the adversary's advantage by at most $2q/p$. Games G_7–G_{11} show how to embed a target value v^* into the components of the secret key without significantly affecting the ability to answer **Dec** queries. Based on the latter, we then construct $\mathcal{I}$ as shown in Appendix E.

7 Applications to searchable encryption

Public-key encryption with keyword search. A *public key encryption with keyword search* (PEKS) scheme [BDOP04] is a tuple $\mathcal{PEKS} = (\mathsf{KG}, \mathsf{PEKS}, \mathsf{Td}, \mathsf{Test})$ of algorithms. Via $(pk, sk) \xleftarrow{\$} \mathsf{KG}$, the key generation algorithm produces a pair of public and private keys. Via $C \xleftarrow{\$} \mathsf{PEKS}(pk, w)$, the encryption algorithm encrypts a keyword w to get a ciphertext under the public key pk. Via $t_w \xleftarrow{\$} \mathsf{Td}(sk, w)$, the trapdoor extraction algorithm computes a trapdoor t_w for keyword w. The deterministic test algorithm $\mathsf{Test}(t_w, C)$ returns 1 if C is an encryption of w and 0 otherwise. In Appendix A, we recall the advantage measures $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{PEKS}}(\mathcal{A})$, where atk $\in \{\text{cpa}, \text{cca}\}$, which capture privacy of PEKS scheme $\mathcal{PEKS}$ against chosen-plaintext (atk = cpa) and chosen-ciphertext (atk = cca) attacks, and the notions are denoted IND-ATK. Furthermore, we also recall the advantage measure $\mathbf{Adv}^{\text{consist}}_{\mathcal{PEKS}}(\mathcal{A})$, which captures the notion CONSIST of computational consistency of PEKS scheme $\mathcal{PEKS}$.

Transforming IBE to PEKS. The bdop-ibe-2-peks transform of [BDOP04] transforms an IBE scheme into a PEKS scheme. Given an IBE scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, the transform associates to it the PEKS scheme $\mathcal{PEKS} = (\mathsf{KG}, \mathsf{PEKS}, \mathsf{Td}, \mathsf{Test})$, where the key-generation algorithm

KG returns $(pk, sk) \xleftarrow{\$} \mathsf{Setup}$; the encryption algorithm $\mathsf{PEKS}(pk, w)$ returns $C \leftarrow \mathsf{Enc}(pk, w, 0^k)$; the trapdoor extraction algorithm $\mathsf{Td}(sk, w)$ returns $t \xleftarrow{\$} \mathsf{Ext}(pk, sk, w)$; the test algorithm $\mathsf{Test}(t, C)$ returns 1 if and only if $\mathsf{Dec}(pk, t, C) = 0^k$. Abdalla et al. [ABC+05] showed that this transform generally does not provide consistency, and presented the consistency-providing new-ibe-2-peks transform as an alternative. We now show that the original bdop-ibe-2-peks transform does yield a consistent PEKS if the underlying IBE scheme is robust. We also show that if the base IBE scheme is ANO-CCA, then the PEKS scheme is IND-CCA, thereby yielding the first IND-CCA-secure PEKS schemes in the standard model, and the first consistent IND-CCA-secure PEKS schemes in the RO model. (Non-consistent IND-CCA-secure PEKS schemes in the RO model are easily derived from [FP07].)

Proposition 7.1 *Let $\mathcal{IBE}$ be an IBE scheme, and let $\mathcal{PEKS}$ be the PEKS scheme associated to it per the* bdop-ibe-2-peks *transform. Given any adversary $\mathcal{A}$ running in time t, we can construct an adversary $\mathcal{B}$ running in time $t + O(t)$ executions of the algorithms of $\mathcal{IBE}$ such that*

$$\mathbf{Adv}^{\text{consist}}_{\mathcal{PEKS}}(\mathcal{A}) \;\le\; \mathbf{Adv}^{\text{rob-cpa}}_{\mathcal{IBE}}(\mathcal{B}) \qquad \textit{and} \qquad \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PEKS}}(\mathcal{A}) \;\le\; \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{B})\ .$$

To see why the first inequality is true, it suffices to consider the adversary $\mathcal{B}$ that on input *pars* runs $(w, w') \xleftarrow{\$} \mathcal{A}(pars)$ and outputs $C \xleftarrow{\$} \mathsf{Enc}(pars, w)$. The proof of the second inequality is an easy adaptation of the proof of the new-ibe-2-peks transform in [ABC+05], where $\mathcal{B}$ answers $\mathcal{A}$'s **Test** queries using its own **Dec** oracle.

Securely combining PKE and PEKS. Searchable encryption by itself is only of limited use since it can only encrypt individual keywords, and since it does not allow decryption. Fuhr and Paillier [FP07] introduce a more flexible variant that allows decryption of the keyword. An even more powerful (and general) primitive can be obtained by combining PEKS with PKE to encrypt non-searchable but recoverable content. For example, one could encrypt the body of an email using a PKE scheme, and append a list of PEKS-encrypted keywords. The straightforward approach of concatenating ciphertexts works fine for CPA security, but is insufficient for a strong, combined IND-CCA security model where the adversary has access to *both* a decryption oracle *and* a testing oracle. Earlier attempts to combine PKE and PEKS [BSNS06, ZI07] do not give the adversary access to the latter. A full IND-CCA-secure PKE/PEKS scheme in the standard model can be obtained by combining the IND-CCA-secure PEKS schemes obtained through our transformation with the techniques of [DK05]. Namely, one can consider label-based [Sho01] variants of the PKE and PEKS primitives, tie the different components of a ciphertext together by using as a common label the verification key of a one-time signature scheme, and append to the ciphertext a signature of all components under the corresponding signing key. We leave details as an exercise to the reader.

Acknowledgments

First and fourth authors were supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT. Second author was supported in part by NSF grants ANR-0129617 and CCR-0208842 and by an IBM Faculty Partnership Development Award. Third author was supported in part by the Thailand Research Fund. Fourth author was supported in part by the Flemish Government under GOA Mefisto 2006/06 and Ambiorix 2005/11, and by the European Commission through the IST Project PRIME.

References

[ABC+05] 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. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222, Santa Barbara, CA, USA, August 14–18, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 4, 15, 20.)

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 3.)

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582, Gold Coast, Australia, December 9–13, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 12, 19.)

[BDOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 4, 14.)

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa Barbara, CA, USA, August 23–27, 1998. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3.)

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 4, 21, 22.)

[BF03] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. (Cited on page 2, 20.)

[BP04] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62, Jeju Island, Korea, December 5–9, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 3, 21.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 2.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology –*

EUROCRYPT 2006, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany. (Cited on page 5, 21.)

[BSNS06] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. On the integration of public key data encryption and public key encryption with keyword search. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *ISC 2006: 9th International Conference on Information Security*, volume 4176 of *Lecture Notes in Computer Science*, pages 217–232, Samos Island, Greece, August 30 – September 2, 2006. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany. (Cited on page 4, 5, 21, 22.)

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 4, 7, 21.)

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. (Cited on page 3, 4, 12, 13, 14.)

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. (Cited on page 2, 3, 10.)

[Di 02] Giovanni Di Crescenzo. Equivocable and extractable commitment schemes. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 74–87, Amalfi, Italy, September 12–13, 2002. Springer-Verlag, Berlin, Germany. (Cited on page 10.)

[DK05] Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 188–209, Cambridge, MA, USA, February 10–12, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[DKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 40–59, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 10.)

[DPP94] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 250–265, Santa Barbara, CA, USA, August 22–26, 1994. Springer-Verlag, Berlin, Germany. (Cited on page 11.)

[DPP97] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *Journal of Cryptology*, 10(3):163–194, 1997. (Cited on page 4.)

[EGM96] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996. (Cited on page 22.)

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 413–431, Santa Barbara, CA, USA, August 20–24, 2000. Springer-Verlag, Berlin, Germany. (Cited on page 10.)

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany. (Cited on page 4, 7, 21.)

[FP07] Thomas Fuhr and Pascal Paillier. Decryptable searchable encryption. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *Provable Security, First International Conference, ProvSec 2007*, volume 4784 of *Lecture Notes in Computer Science*, pages 228–236, Wollongong, Australia, November 1–2, 2007. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany. (Cited on page 23.)

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. (Cited on page 2.)

[Hal05] Shai Halevi. A sufficient condition for key-privacy. Cryptology ePrint Archive, Report 2005/005, 2005. `http://eprint.iacr.org/`. (Cited on page 23.)

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, Lecture Notes in Computer Science, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979. (Cited on page 22.)

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136, Santa Barbara, CA, USA, August 20–24, 1990. Springer-Verlag, Berlin, Germany. (Cited on page 11.)

[Ped92] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, CA, USA, August 11–15, 1992. Springer-Verlag, Berlin, Germany. (Cited on page 10.)

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Santa Barbara, CA, USA, August 11–15, 1992. Springer-Verlag, Berlin, Germany. (Cited on page 2.)

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388, New Delhi, India, February 5–7, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 13.)

[Sak00] Kazue Sako. An auction protocol which hides bids of losers. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 422–432, Melbourne, Victoria, Australia, January 18–20, 2000. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 5, 6, 34.)

[Sho01] Victor Shoup. A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112, 2001. `http://eprint.iacr.org/`. (Cited on page 15.)

[ZI07] Rui Zhang and Hideki Imai. Generic combination of public key encryption with keyword search and public key encryption. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *CANS 07: 6th International Conference on Cryptology and Network Security*, volume 4856 of *Lecture Notes in Computer Science*, pages 159–174, Singapore, December 8–10, 2007. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

A Recall of standard definitions and tools

Correctness and security of PKE schemes. Correctness of PKE schemes requires that, for all $pars \in [\mathsf{PG}]$, all plaintexts M in the underlying message space, and all $(pk, sk) \in [\mathsf{KG}(pars)]$, we have $\mathsf{Dec}(pars, pk, sk, \mathsf{Enc}(pars, pk, M)) = M$ with probability one, where the probability is taken over the coins of Enc. We recall the standard privacy notions IND-CPA, IND-CCA and anonymity notions ANO-CPA, ANO-CCA of [BBDP01].

The games IND-ATK$_{\mathcal{PKE}}$ for ATK $\in \{\mathrm{CPA}, \mathrm{CCA}\}$ begin with the generation of parameters $pars \xleftarrow{\$} \mathsf{PG}$, a key pair $(pk, sk) \xleftarrow{\$} \mathsf{KG}(pars)$, and a random bit $b \xleftarrow{\$} \{0,1\}^*$. The adversary $\mathcal{A}$ is run on input $(pars, pk)$ and has access to an encryption oracle $\mathbf{LR}(M_0^*, M_1^*)$ and, if ATK = CCA, a decryption oracle $\mathbf{Dec}(C) = \mathsf{Dec}(pars, pk, sk, C)$. The encryption oracle can only be queried once and returns $C^* \xleftarrow{\$} \mathsf{Enc}(pars, pk, M_b^*)$. The game returns true if $\mathcal{A}$ outputs $b' = b$, $|M_0^*| = |M_1^*|$, and, if ATK = CCA, it never queried $\mathbf{Dec}(C^*)$. Its advantage is defined as $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{PKE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{IND-ATK}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \mathsf{true}\right] - 1$.

The games ANO-ATK$_{\mathcal{PKE}}$ for ATK $\in \{\mathrm{CPA}, \mathrm{CCA}\}$ first generate parameters $pars \xleftarrow{\$} \mathsf{PG}$, two key pairs $(pk_0, sk_0), (pk_1, sk_1) \xleftarrow{\$} \mathsf{KG}(pars)$, and a random bit $b \xleftarrow{\$} \{0,1\}^*$. The adversary $\mathcal{A}$ is run on input $(pars, pk_0, pk_1)$ and has access to an encryption oracle $\mathbf{LR}(M^*)$ and, if ATK = CCA, a decryption oracle $\mathbf{Dec}(d, C) = \mathsf{Dec}(pars, pk_d, sk_d, C)$. The encryption oracle can only be queried once and returns $C^* \xleftarrow{\$} \mathsf{Enc}(pars, pk_b, M^*)$. The game returns true if $\mathcal{A}$ outputs $b' = b$ and, if ATK = CCA, if it never queried $\mathbf{Dec}(d, C^*)$ for $d \in \{0,1\}$. Its advantage is defined as $\mathbf{Adv}^{\text{ano-atk}}_{\mathcal{PKE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{ANO-ATK}^{\mathcal{A}}_{\mathcal{PKE}} \Rightarrow \mathsf{true}\right] - 1$.

CORRECTNESS AND SECURITY OF IBE SCHEMES. Correctness of IBE schemes requires that, for all $(pars, msk) \in [\mathsf{Setup}]$, all plaintexts M in the underlying message space, all identities id, and all secret keys $usk \in [\mathsf{Ext}(pars, msk, id)]$, we have $\mathsf{Dec}(pars, id, usk, \mathsf{Enc}(pars, id, M)) = M$ with probability one, where the probability is taken over the coins of Enc. We recall here the standard privacy notions IND-ATK [BF03] and anonymity notions ANO-ATK [ABC+05] for ATK $\in \{\mathrm{CPA}, \mathrm{CCA}\}$.

The IND-ATK$_{\mathcal{IBE}}$ game starts with the generation of parameters $(pars, msk) \xleftarrow{\$} \mathsf{Setup}$ and a random bit $b \xleftarrow{\$} \{0,1\}^*$. The adversary $\mathcal{A}$ is run on input $pars$ and has access to an encryption oracle $\mathbf{LR}(id^*, M_0^*, M_1^*)$, a key extraction oracle $\mathbf{Ext}(id) = \mathsf{Ext}(pars, msk, id)$, and, if ATK = CCA, a decryption oracle $\mathbf{Dec}(id, C) = \mathsf{Dec}(pars, id, \mathsf{Ext}(pars, msk, id), C)$. The encryption oracle can only be queried once and returns $C^* \xleftarrow{\$} \mathsf{Enc}(pars, id^*, M_b^*)$. The game returns true if $\mathcal{A}$ outputs $b' = b$ and if $|M_0^*| = |M_1^*|$ while $\mathcal{A}$ never made a query $\mathbf{Ext}(id^*)$ or $\mathbf{Dec}(id^*, C^*)$. Its advantage is defined as $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{IBE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{IND-ATK}^{\mathcal{A}}_{\mathcal{IBE}} \Rightarrow \mathsf{true}\right] - 1$.

The ANO-ATK$_{\mathcal{IBE}}$ game starts with the generation of parameters $(pars, msk) \xleftarrow{\$} \mathsf{Setup}$ and a random bit $b \xleftarrow{\$} \{0,1\}^*$. It runs the adversary $\mathcal{A}(pars)$ and gives it access to an encryption oracle $\mathbf{LR}(id_0^*, id_1^*, M^*)$, a key extraction oracle $\mathbf{Ext}(id) = \mathsf{Ext}(pars, msk, id)$, and, if ATK = CCA, a decryption oracle $\mathbf{Dec}(id, C) = \mathsf{Dec}(pars, id, \mathsf{Ext}(pars, msk, id), C)$. The encryption oracle can only be queried once and returns $C^* \xleftarrow{\$} \mathsf{Enc}(pars, id_b^*, M^*)$. The game returns true if $\mathcal{A}$ outputs $b' = b$ without making a query $\mathbf{Ext}(id_d^*)$ or $\mathbf{Dec}(id_d^*, C^*)$ for any $d \in \{0,1\}$. Its advantage is defined as $\mathbf{Adv}^{\text{ano-atk}}_{\mathcal{IBE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{ANO-ATK}^{\mathcal{A}}_{\mathcal{IBE}} \Rightarrow \mathsf{true}\right] - 1$.

CONSISTENCY AND SECURITY OF PEKS SCHEMES. We consider the consistency definition of [ABC+05] through the following CONSIST$_{\mathcal{PEKS}}$ game. A fresh key pair $(pk, sk) \xleftarrow{\$} \mathsf{KG}$ is generated. On input the public key pk, the adversary $\mathcal{A}$ outputs two keywords w, w'. The game returns true if $w \neq w'$ and $\mathsf{Test}(t', C) = 1$ where $C \xleftarrow{\$} \mathsf{PEKS}(pk, w)$ and $t' \xleftarrow{\$} \mathsf{Td}(sk, w')$. The advantage $\mathbf{Adv}^{\text{consist}}_{\mathcal{PEKS}}(\mathcal{A})$ is defined as the probability that the game returns true.

Privacy is defined through the following IND-ATK$_{\mathcal{PEKS}}$ games for ATK $\in \{\mathrm{CPA}, \mathrm{CCA}\}$. The experiment generates a key pair $(pk, sk) \xleftarrow{\$} \mathsf{KG}$ and a random bit $b \xleftarrow{\$} \{0,1\}$. The adversary $\mathcal{A}$ is given pk as input and has access to oracles $\mathbf{LR}(w_0^*, w_1^*)$, $\mathbf{TD}(w) = \mathsf{Td}(sk, w)$, and, if ATK = CCA, $\mathbf{Test}(w, C) = \mathsf{Test}(\mathsf{Td}(sk, w), C)$. The $\mathbf{LR}$ oracle can only be called once and returns $C^* \xleftarrow{\$} \mathsf{PEKS}(pk, w_b^*)$. The game returns true if the adversary $\mathcal{A}$ outputs $b' = b$ without querying $\mathbf{TD}(w_d^*)$ or, if ATK = CCA, without querying $\mathbf{Test}(w_d^*, C^*)$ for any $d \in \{0,1\}$. Its advantage is defined as $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{PEKS}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{IND-ATK}^{\mathcal{A}}_{\mathcal{PEKS}} \Rightarrow \mathsf{true}\right] - 1$.

CORRECTNESS AND SECURITY OF COMMITMENT SCHEMES. Correctness of a commitment scheme $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$ requires that, for any $x \in \{0,1\}^*$, any $cpars \in [\mathsf{CPG}]$, and any $(com, dec) \in [\mathsf{Com}(cpars, x)]$, we have that $\mathsf{Open}(cpars, com, dec) = x$ with probability one, where the probability is taken over the coins of Com. The standard hiding property of a commitment scheme $\mathcal{CMT}$ is defined through the following game HIDE$_{\mathcal{CMT}}$. The experiment generates parameters $cpars \xleftarrow{\$} \mathsf{CPG}$ and a random bit $b \xleftarrow{\$} \{0,1\}^*$. The adversary $\mathcal{A}$ is run on input $cpars$ and gets to make a single query to an oracle $\mathbf{LR}(x_0, x_1) = \mathsf{Com}(cpars, x_b)$. The game returns true if $\mathcal{A}$ outputs $b' = b$. Its advantage is defined as $\mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{HIDE}^{\mathcal{A}}_{\mathcal{CMT}} \Rightarrow \mathsf{true}\right] - 1$.

The binding property is defined through a game BIND$_{\mathcal{CMT}}$ where $\mathcal{A}$ is run on input $cpars \xleftarrow{\$} \mathsf{CPG}$ and outputs a three-tuple (com, dec_0, dec_1). Let $x_0 \leftarrow \mathsf{Open}(cpars, com, dec_0)$ and $x_1 \leftarrow \mathsf{Open}(cpars, com, dec_1)$. The game returns true if $x_0 \neq x_1$, $x_0 \neq \perp$, and $x_1 \neq \perp$. The advantage $\mathbf{Adv}^{\text{bind}}_{\mathcal{CMT}}(\mathcal{A})$ is the probability that the game returns true.

REMARK ON MESSAGE LENGTHS. We note that the IND-ATK definitions of PKE and IBE schemes

insist that the challenge messages M_0^*, M_1^* be of the same length, while no such restrictions are present for the challenge identities id_0^*, id_1^* in the ANO-ATK definition of IBE, for the challenge keywords w_0^*, w_1^* in the IND-ATK definition of PEKS, or for the data x_0, x_1 in the HIDE definition of commitment schemes. This is a conscious choice,

Game-Playing Lemma. We will use the Fundamental Lemma of game-playing of [BR06], which we now recall. Games $\mathrm{G}_i, \mathrm{G}_j$ are *identical until* bad if their code differs only in statements that follow the setting of bad to true. (For example, games $\mathrm{G}_1, \mathrm{G}_2$ of Figure 7 are identical until bad.)

Lemma A.1 [BR06] *Let $\mathrm{G}_i, \mathrm{G}_j$ be identical until* bad *games, and $\mathcal{A}$ an adversary. Then for any y*

$$\left|\Pr\left[\,\mathrm{G}_i^{\mathcal{A}} \Rightarrow y\,\right] - \Pr\left[\,\mathrm{G}_j^{\mathcal{A}} \Rightarrow y\,\right]\right| \;\leq\; \Pr\left[\,\mathrm{G}_j^{\mathcal{A}} \textit{ sets } \mathsf{bad}\,\right] . \;\blacksquare$$

B More results on robustness of specific transforms and schemes

In this section we show that neither of two popular IND-CCA-providing transforms, the Fujisaki-Okamoto (FO) transform [FO99] in the random oracle model and the Canetti-Halevi-Katz (CHK) transform [CHK04] in the standard model, yield robustness. Since the FO transform even provides the stronger notion of plaintext awareness [BP04], the counterexample below is at the same time a proof that even plaintext awareness does not suffice for robustness. The fact that neither of the transforms confer robustness generically does not exclude that they may still do so for certain specific schemes. We show that this is actually the case for the Boneh-Franklin IBE [BF01], which uses the FO transform to obtain IND-CCA security, and that it is *not* the case for the Boyen-Waters IBE [BW06], which uses the CHK transform.

The Fujisaki-Okamoto transform. Given a public-key encryption scheme $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ the FO transform yields a PKE scheme $\overline{\mathcal{PKE}} = (\mathsf{PG}, \mathsf{KG}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ where a message M is encrypted as

$$\left(\; \mathsf{Enc}(\mathit{pars}, \mathit{pk}, x; H(x, M)) \;\;,\;\; G(x) \oplus M \;\right) ,$$

where $x \xleftarrow{\$} \{0,1\}^k$, where $G(\cdot)$ and $H(\cdot)$ are random oracles, and where $H(x, M)$ is used as the random coins for the Enc algorithm. To decrypt a ciphertext (C_1, C_2), one recovers x by decrypting C_1, recovers $M \leftarrow C_2 \oplus G(x)$, and checks that $\mathsf{Enc}(\mathit{pars}, \mathit{pk}, x; H(x, M)) = C_1$. If this is the case then M is returned, otherwise $\perp$ is returned.

Given a scheme $\mathcal{PKE}^* = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}^*, \mathsf{Dec}^*)$, we show how to build a scheme $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}$ such that $\overline{\mathcal{PKE}}$ obtained by applying the FO transform to $\mathcal{PKE}$ is not ROB-CPA. Namely, for some fixed $x^* \in \{0,1\}^k$ and M^*, let encryption and decryption be given by

Algorithm $\mathsf{Enc}(\mathit{pars}, \mathit{pk}, x; \rho)$	Algorithm $\mathsf{Dec}(\mathit{pars}, \mathit{pk}, \mathit{sk}, b \| C^*)$
If $x = x^*$ and $\rho = H(x^*, M^*)$ then return 0	If $b = 0$ then return x^*
Else return $1 \| \mathsf{Enc}^*(\mathit{pars}, \mathit{pk}, x; \rho)$	Else return $\mathsf{Dec}^*(\mathit{pars}, \mathit{pk}, \mathit{sk}, C^*)$.

It is easy to see that if $\mathcal{PKE}^*$ is one-way (the notion required by the FO transform), then so is $\mathcal{PKE}$, because for an honestly-generated ciphertext the random coins $H(x^*, M^*)$ will hardly ever occur. Moreover, it is also straightforward to show that, if $\mathcal{PKE}^*$ is γ-uniform, then $\mathcal{PKE}$ is γ'-uniform for $\gamma' = \max(\gamma, 1/2^\ell)$, where ℓ is the output length of H (please refer to [FO99] for the definition of γ-uniformity). It is also easy to see that the scheme $\overline{\mathcal{PKE}}$ obtained by applying the FO transform to $\mathcal{PKE}$ is not robust: the ciphertext $\overline{C} = (0\ ,\ G(x^*) \oplus M^*)$ decrypts correctly to M^* under any public key.

The Boneh-Franklin IBE. Boneh and Franklin proposed the first truly practical provably secure IBE scheme in [BF01]. They also propose a variant that uses the FO transform to obtain provable

IND-CCA security in the random oracle model under the bilinear Diffie-Hellman (BDH) assumption; we refer to it as the BF-IBE scheme here. A straightforward modification of the proof can be used to show that BF-IBE is also ANO-CCA in the random oracle model under the same assumption. We now give a proof sketch that BF-IBE is also (unconditionally) ROB-CCA in the random oracle model.

Let e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a non-degenerate bilinear map, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are multiplicative cyclic groups of prime order p [BF01]. Let g be a generator of $\mathbb{G}_1$. The master secret key of the BF-IBE scheme is an exponent $s \xleftarrow{\$} \mathbb{Z}_p^*$, the public parameters contain $S \leftarrow g^s$. For random oracles $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \to \{0,1\}^k$, $H_3 : \{0,1\}^k \times \{0,1\}^\ell \to \mathbb{Z}_p^*$, and $H_4 : \{0,1\}^k \to \{0,1\}^\ell$, the encryption of a message M under identity id is a tuple

$$\left(g^r \ , \ x \oplus H_2(e(S, H_1(id))^r) \ , \ M \oplus H_4(x)\right) \ ,$$

where $x \xleftarrow{\$} \{0,1\}^k$ and $r \leftarrow H_3(x, M)$. To decrypt a ciphertext (C_1, C_2, C_3), the user with identity id and decryption key $usk = H_1(id)^s$ computes $x \leftarrow C_2 \oplus H_2(e(C_1, usk))$, $M \leftarrow C_3 \oplus H_4(x)$, and $r \leftarrow H_3(x, M)$. If $C_1 \neq g^r$ he rejects, otherwise he outputs M.

Let us now consider a ROB-CCA adversary $\mathcal{A}$ that even knows the master secret s (and therefore can derive all keys and decrypt all ciphertexts that it wants). Since H_1 maps into $\mathbb{G}_1^*$, all its outputs are of full order p. The probability that $\mathcal{A}$ finds two identities id_1 and id_2 such that $H_1(id) = H_1(id_2)$ is negligible. Since $S \in \mathbb{G}_1^*$ and the map is non-degenerate, we therefore have that $g_{id_1} = e(S, H_1(id_1))$ and $g_{id_2} = e(S, H_1(id_2))$ are different and of full order p. Since H_3 maps into $\mathbb{Z}_p^*$, we have that $r \neq 0$, and therefore that $g_{id_1}^r$ and $g_{id_2}^r$ are different. If the output of H_2 is large enough to prevent collisions from being found, that also means that $H_2(g_{id_1}^r)$ and $H_2(g_{id_2}^r)$ are different. Decryption under both identities therefore yields two different values $x_1 \neq x_2$, and possibly different messages M_1, M_2. In order for the ciphertext to be valid for both identities, we need that $r = H_3(x_1, M_1) = H_3(x_2, M_2)$, but the probability of this happening is again negligible in the random oracle model. As a result, it follows that the BF-IBE scheme is also ROB-CCA in the random oracle model.

The Canetti-Halevi-Katz transform. The CHK transform turns an IBE scheme and a one-time signature scheme [EGM96, Lam79] into a PKE scheme as follows. For each ciphertext a fresh signature key pair (spk, ssk) is generated. The ciphertext is a tuple (C, spk, σ) where C is the encryption of M to identity spk and σ is a signature of C under ssk. To decrypt, one verifies the signature σ, derives the decryption key for identity spk, and decrypts C.

Given a scheme $\mathcal{IBE}^* = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}^*, \mathsf{Dec}^*)$, consider the scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ where $\mathsf{Enc}(pars, id, M) = 1 \| \mathsf{Enc}^*(pars, id, M)$ and where $\mathsf{Dec}(pars, id, usk, b \| C^*)$ returns $\mathsf{Dec}^*(pars, id, usk, C^*)$ if $b = 1$ and simply returns C^* if $b = 0$. This scheme clearly inherits the privacy and anonymity properties of $\mathcal{IBE}^*$. However, if $\mathcal{IBE}$ is used in the CHK transformation, then one can easily generate a ciphertext $(0 \| M, spk, \sigma)$ that validly decrypts to M under any parameters $pars$ (which in the CHK transform serve as the user's public key).

An extension of the CHK transform turns any IND-CPA secure $\ell+1$-level hierarchical IBE (HIBE) into an IND-CCA secure ℓ-level HIBE. It is easy to see that this transform does not confer robustness either.

The Boyen-Waters IBE. Boyen and Waters [BW06] proposed a HIBE scheme which is IND-CPA and ANO-CPA in the standard model, and a variant that uses the CHK transform to achieve IND-CCA and ANO-CCA security. Decryption in the IND-CPA secure scheme never rejects, so it is definitely not ROB-CPA. Without going into details here, it is easy to see that the IND-CCA variant is not ROB-CPA either, because any ciphertext that is valid with respect to one identity will also be valid with respect to another identity, since the verification of the one-time signature does not depend on the identity of the recipient. (The natural fix to include the identity in the signed data may ruin anonymity.)

The IND-CCA-secure variant of Gentry's IBE scheme [Gen06] falls to a similar robustness attack as the original Cramer-Shoup scheme, by choosing a random exponent $r = 0$. We did not check whether explicitly forbidding this choice restores robustness, however.

C Proofs of the commit-public-key transform

Proof of Equation (1). Let $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$, and $\overline{\mathcal{PKE}} = (\overline{\mathsf{PG}}, \overline{\mathsf{KG}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$. On input *cpars*, adversary $\mathcal{B}$ lets $pars \xleftarrow{\$} \mathsf{PG}$; $(pk_0, sk_0) \xleftarrow{\$} \mathsf{KG}(pars)$; $(pk_1, sk_1) \xleftarrow{\$} \mathsf{KG}(pars)$; $R \xleftarrow{\$} \{0,1\}^r$. Notice that $\mathcal{B}$ knows sk_0 and sk_1. It runs $\mathcal{A}$ on input $(pars, cpars, R)$, pk_0, pk_1. When $\mathcal{A}$ submits a decryption query C, adversary $\mathcal{B}$ computes $\overline{M_0} = \overline{\mathsf{Dec}}((pars, cpars, R), pk_0, sk_0, C)$ and $\overline{M_1} = \overline{\mathsf{Dec}}((pars, cpars, R), pk_1, sk_1, C)$, which it can do so since it has both sk_0 and sk_1. Then, $\mathcal{B}$ returns $(\overline{M_0}, \overline{M_1})$ to $\mathcal{A}$. If $\mathcal{A}$ wins Game ROB-CCA$_{\overline{\mathcal{PKE}}}$ then there exists a **Dec** query $\overline{C} = (C, com)$ for which neither $\overline{M_0} = \overline{\mathsf{Dec}}((pars, cpars, R), pk_0, sk_0, C)$ nor $\overline{M_1} = \overline{\mathsf{Dec}}((pars, cpars, R), pk_1, sk_1, C)$ equal $\perp$. Upon seeing such a query, $\mathcal{B}$ parses $M_b = \mathsf{Dec}(pars, pk_b, sk_b, C)$ as $(\overline{M_b}, dec_b, R)$ for $b \in \{0,1\}$. It returns (com, dec_0, dec_1). If $pk_0 \neq pk_1$ and if $\mathcal{A}$ wins Game ROB-CCA$_{\overline{\mathcal{PKE}}}$, then $\mathcal{B}$ also wins BIND$_{\mathcal{CMT}}$. We omit details.

Proof of Equation (2). Let $\mathcal{PKE} = (\mathsf{PG}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$, and $\overline{\mathcal{PKE}} = (\overline{\mathsf{PG}}, \overline{\mathsf{KG}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$. Adversary $\mathcal{B}$, on input $(pars, pk)$, lets $cpars \xleftarrow{\$} \mathsf{CPG}$; $R \xleftarrow{\$} \{0,1\}^r$ and runs $\mathcal{A}$ on input $((pars, cpars, R), pk)$. When $\mathcal{A}$ queries (M_0, M_1) to its **LR** oracle, $\mathcal{B}$ lets $(com, dec) \xleftarrow{\$} \mathsf{Com}(cpars, pk)$ and queries $((M_0, dec, R), (M_1, dec, R))$ to its own **LR** oracle, getting back C. It forwards (C, com) to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs.

Proof of Equation (3). Given that we already showed $\overline{\mathcal{PKE}}$ is IND-CPA, it is natural to consider using Halevi's condition for anonymity [Hal05] to prove Equation (3), but this does not seem to help here. Instead we give a direct proof. We consider a sequence of games, all of which have the same **Initialize** and **Finalize** procedures. Specifically, for **Initialize**, each game picks a challenge bit b, generates the parameters $(pars, cpars, R), pk_0, sk_0, pk_1, sk_1$ faithfully, namely according to $\overline{\mathsf{PG}}$ and $\overline{\mathsf{KG}}$, and returns $(pars, cpars, R), pk_0, pk_1$. On input a bit b', the **Finalize** procedure returns true iff $b' = b$. The differences among the games lie in the way **LR** encrypts an input plaintext M^*. Specifically, for each $i, j, k \in \{0,1\}$, we have a game G_{ijk} in which

(1) pk_i is used for encryption by the algorithm Enc

(2) pk_j is committed to by the algorithm Com, and

(3) if $k = 1$ then (M, dec, R) gets encrypted by the algorithm Enc; otherwise, $(0^{|M|}, 0^{|dec|}, 0^r)$ is encrypted by Enc.

We let $P_{ijk} = \Pr\left[\mathrm{G}^{\mathcal{A}}_{ijk} \Rightarrow \mathsf{true}\right]$ for each $i, j, k \in \{0,1\}$. (Here, *dec* is the decommital key obtained from Com executed in the game.) We consider the sequence of games $\mathrm{G}_{001}, \mathrm{G}_{000}, \mathrm{G}_{100}, \mathrm{G}_{110}, \mathrm{G}_{111}$. Now, notice that for any adversary $\mathcal{A}$

$$\mathbf{Adv}^{\text{ano-cpa}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \;=\; P_{001} - P_{111} \;=\; (P_{001} - P_{000}) + (P_{000} - P_{100}) + (P_{100} - P_{110}) + (P_{110} - P_{111})\;.$$

It is easy to construct adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ so that

$$P_{001} - P_{000} \;\leq\; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{PKE}}(\mathcal{B}_2) \tag{14}$$

$$P_{000} - P_{100} \;\leq\; \mathbf{Adv}^{\text{ano-cpa}}_{\mathcal{PKE}}(\mathcal{B}_1) \tag{15}$$

$$P_{100} - P_{110} \;\leq\; \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_3) \tag{16}$$

$$P_{110} - P_{111} \;\leq\; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{PKE}}(\mathcal{B}_2)\;. \tag{17}$$

Equations (14) and (17) hold because the difference in the two games in each equation is whether (M, dec, R) or a string of zeros is encrypted. Equation (15) holds because the difference in the two games is whether the base encryption algorithm Enc is performed under pk_0 or pk_1. Equation (16) holds because the difference in the two games is whether pk_0 or pk_1 is being committed to.

Proof of Equation (4). We prove Equation (4) by considering the following games G_0 and G_1.

1. Game G_0 is exactly as the IND-CCA$_{\overline{\mathcal{PKE}}}$ game, so

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \;=\; 2 \cdot \Pr\left[\, \mathrm{G}_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] - 1 \;.$$

2. Game G_1 differs from G_0 so that when $\mathcal{A}$ makes a query $\mathbf{Dec}((C^*, com))$ with $com \neq com^*$ yet still $\mathsf{Open}(cpars, com, dec) = pk$, then the flag bad is set and $\perp$ is returned. The game-playing lemma says that

$$\Pr\left[\, \mathrm{G}_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] \;\leq\; \Pr\left[\, \mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] + \Pr\left[\, \mathrm{G}_1^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\right] .$$

   The second term is easily bounded by observing that bad only gets set when com and com^* open to the same data pk with the same decommitment dec, thereby violating the copy-resistance of $\mathcal{CMT}$. To bound the first term, observe that an adversary $\mathcal{A}$ winning G_1 is easily transformed into an adversary $\mathcal{B}_2$ winning the IND-CCA$_{\overline{\mathcal{PKE}}}$ game. We have that

$$\begin{aligned} \Pr\left[\, \mathrm{G}_1^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\right] &\leq \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) \;, \\ 2 \cdot \Pr\left[\, \mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] - 1 &\leq \mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{B}_2) \;, \end{aligned}$$

   from which Equation (4) easily follows.

Proof of Equation (5). To prove Equation (5), we need the following information-theoretic lemma that bounds the probability that an algorithm $\mathcal{A}_2$ reproduces a random input R to $\mathcal{A}_1$ when $\mathcal{A}_1$ can only leak a limited amount of information about R to $\mathcal{A}_2$.

Lemma C.1 *Let $r \in \mathbb{N}$ and let $\mathcal{A}_1, \mathcal{A}_2$ be arbitrary (i.e., possibly randomized and computationally unbounded) algorithms, then*

$$P \;=\; \Pr\left[\, R' = R \;|\; R \xleftarrow{\$} \{0,1\}^r \,;\, L \xleftarrow{\$} \mathcal{A}_1(R) \,;\, R' \xleftarrow{\$} \mathcal{A}_2(L) \,\right] \;\leq\; 2^{|L|-r} \;.$$

Proof of Lemma C.1: First let's assume $\mathcal{A}_1, \mathcal{A}_2$ are deterministic. For a fixed length $\ell = |L|$ consider the sets S_L for $L \in \{0,1\}^\ell$ so that

$$S_L \;=\; \{R \in \{0,1\}^r \;:\; \mathcal{A}_1(R) = L\}$$

and let $c_L = |S_L|$. When $\mathcal{A}_2$ gets input L, its best strategy is to return any $R' \in S_L$, in which case the probability that $R' = R$ is $1/c_L$. Let $S = \{L : S_L \neq \emptyset\}$. Then for a random R the probability that $\mathcal{A}_1(R) = L$ is $c_L/2^r$, so that

$$P \;\leq\; \sum_{L \in S} \frac{1}{c_L} \cdot \frac{c_L}{2^r} \;=\; \frac{|S|}{2^r} \;\leq\; \frac{2^\ell}{2^r}$$

as claimed. If $\mathcal{A}_1, \mathcal{A}_2$ are randomized, then let their respective random tapes be denoted by ρ_1, ρ_2. The probability P is taken over the choice of R and the random tapes ρ_1, ρ_2. Let $\overline{\rho}_1, \overline{\rho}_2$ be the random

tapes that maximize P, and let $\overline{\mathcal{A}}_1, \overline{\mathcal{A}}_2$ be the deterministic algorithms obtained by fixing the random tapes of $\mathcal{A}_1, \mathcal{A}_2$ to $\overline{\rho}_1, \overline{\rho}_2$, respectively. Then applying the above result to $\overline{\mathcal{A}}_1, \overline{\mathcal{A}}_2$ yields

$$P \;\le\; \Pr\Big[\, R' = R \mid\; R \xleftarrow{\$} \{0,1\}^r \,;\; L \xleftarrow{\$} \overline{\mathcal{A}}_1(R) \,;\; R' \xleftarrow{\$} \overline{\mathcal{A}}_2(L) \,\Big] \;\le\; 2^{\ell - r} \;.$$

∎

The proof of Equation (5) is composed of the following sequence of games G_0 through G_4:

0. Game G_0 is exactly as Game ANO-CCA$_{\overline{\mathcal{PKE}}}$, so we have

$$\mathbf{Adv}^{\text{ano-cca}}_{\overline{\mathcal{PKE}}}(\mathcal{A}) \;=\; 2 \cdot \Pr\big[\, \mathrm{G}_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] - 1 \;.$$

1. In Game G_1, the oracle $\mathbf{Dec}(d, (C, com))$ returns $\perp$ to $\mathcal{A}$ and sets bad to true whenever $d = b$, $C = C^*$, $com \neq com^*$, and yet $\mathsf{Open}(cpars, com, dec) = pk$ and $R' = R$ (where dec and R' are obtained by decrypting C). It is clear that this happens only when the copy-resistance of $\mathcal{CMT}$ is broken.

$$\begin{aligned} \Pr\big[\, \mathrm{G}_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] \;&\le\; \Pr\big[\, \mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] + \Pr\big[\, \mathrm{G}_1^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\big] \\ &\le\; \Pr\big[\, \mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] + \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) \end{aligned}$$

2. In Game G_2, the procedure $\mathbf{LR}$ computes C^* as $\mathsf{Enc}(pars, pk_b, (0^{|M^*|}, 0^{|dec^*|}, 0^r))$ instead of $\mathsf{Enc}(pars, pk_b, (M^*, dec^*, R))$. It is easy to see that to distinguish between these games one needs to break the IND-CCA security of $\mathcal{PKE}$:

$$\Pr\big[\, \mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] \;\le\; \Pr\big[\, \mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] + \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE}}(\mathcal{B}_2) \;.$$

3. In G_3, $\mathbf{Dec}$ returns $\perp$ to $\mathcal{A}$ and sets bad to true whenever $d = 1 - b$, $C = C^*$, $com \neq com^*$ and yet $\mathsf{Open}(cpars, com, dec) = pk$ and $R' = R$ (where dec and R' are obtained by decrypting C). We have

$$\Pr\big[\, \mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] \;\le\; \Pr\big[\, \mathrm{G}_3^{\mathcal{A}} \Rightarrow \mathsf{true} \,\big] + \Pr\big[\, \mathrm{G}_3^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\big] \;.$$

   To bound the second term above, first note that bad is set exactly when $\mathcal{A}$ manages to find a message M^* and a commitment $com \neq com^*$ such that the encryption C^* of a string of zeros of length $|M^*| + |dec^*| + r$ with respect to public key pk_b decrypts to (M, dec, R) under secret key sk_{1-b}. The ciphertext C^* however is generated in the $\mathbf{LR}$ procedure as $\mathsf{Enc}(pars, pk_b, (0^{|M^*|}, 0^{|dec^*|}, 0^r))$, which is "almost" independent of R: almost, because $\mathcal{A}$ may have slipped some information about R into M^*, and the length of M^* is used in the generation of C^*.

   We instantiate Lemma C.1 with the algorithm $\mathcal{A}_1(R)$ that runs $\mathcal{A}$ on input $\overline{pars} = (pars, cpars, R)$ until it makes its left-or-right query $\mathbf{LR}(M^*)$. The output of $\mathcal{A}_1$ is $L = |M^*|$. Algorithm $\mathcal{A}_2(L)$ generates a ciphertext $C^* \xleftarrow{\$} \mathsf{Enc}(pars, pk_b, (0^L, 0^{|dec^*|}, 0^r))$ and decrypts it again as $(M, dec, R') \leftarrow \mathsf{Dec}(pars, pk_{1-b}, sk_{1-b}, C)$. It is clear that this is exactly what happens in the $\mathbf{LR}$ and $\mathbf{Dec}$ procedures of G_3, and that the probability that bad gets set is bounded by the probability that $R' = R$, so by Lemma C.1 we have

$$\Pr\big[\, \mathrm{G}_3^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\big] \;\le\; 1/2^{r - \log |M^*|} \;=\; |M^*|/2^r \;.$$

4. In G_4, **LR** computes (com^*, dec^*) as $\mathsf{Com}(cpars, 0^{|pk_0|})$. It is easy to see that to distinguish between G_3 and G_4 one needs to break the hiding property of the commitment scheme:

$$\Pr\left[\, G_3^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] \;\leq\; \Pr\left[\, G_4^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] + \mathbf{Adv}^{\text{hide}}_{\mathcal{CMT}}(\mathcal{B}_3)\ .$$

In Game G_4 all queries of the form $\mathbf{Dec}(d, (C^*, com))$ are responded to with $\perp$, and the only information that $\mathcal{A}$ has about the hidden bit b is due to the ciphertext $C^* \xleftarrow{\$} \mathsf{Enc}(pars, pk_b, (0^{|M^*|}, 0^{|dec^*|}, 0^r))$. An adversary $\mathcal{A}$ winning Game G_4 is therefore easily transformed into an adversary $\mathcal{B}_4$ against the ANO-CCA-security of $\mathcal{PKE}$:

$$2 \cdot \Pr\left[\, G_4^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] - 1 \;\leq\; \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{PKE}}(\mathcal{B}_4)\ .$$

Equation (5) is easily obtained by combining the inequalities above.

D Proof sketches of the commit-identity transform

Proofs of Equations (6)–(9). Let $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, $\mathcal{CMT} = (\mathsf{CPG}, \mathsf{Com}, \mathsf{Open})$, and $\overline{\mathcal{IBE}} = (\overline{\mathsf{Setup}}, \overline{\mathsf{Ext}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$. The proof of Equation (6) is similar to that of Equation (1) with the role of a public key played by a user's identity. On input *cpars*, the adversary $\mathcal{B}$ simply generates the rest of the parameters including the master secret key for the scheme $\mathcal{IBE}$ then runs $\mathcal{A}$, answering $\mathcal{A}$'s queries faithfully. If $\mathcal{A}$ wins, there must be a $\overline{\mathbf{Dec}}$ query $\overline{C} = (C, com)$ that decrypts correctly for both id_0 and id_1 such that $id_0 \neq id_1$. This means that $\mathcal{B}$ can output (com, dec_0, dec_1) where $(\overline{M_0}, dec_0, \overline{R})$ and $(\overline{M_1}, dec_1, \overline{R})$ are the decryptions of C under id_0 and id_1, respectively.

The proof of Equation (7) is straightforward and analogous to that of Equation (2). The proof of Equation (8) is almost identical to that of Equation (3), replacing the public keys pk_0, pk_1 with identities id_0^*, id_1^*. We omit details.

The proof of Equation (9) is the same as that of Equation (4), but again with identities playing the role of public keys. In the intermediate game that isolates the copy-resistance of $\mathcal{CMT}$, the $\mathbf{Dec}(id, (C, com))$ procedure sets bad if $(C = C^*) \wedge (com \neq com^*) \wedge (id = id^*)$. None of the other differences introduce any particular difficulties in the proof.

Proof of Equation (10). In this proof we essentially prove two different bounds, one of them being obtained through games G_0–G_5, and the other being obtained through games G_0–G_2, G_3'–G_5'. Both bounds hold at any time, but the first is more useful for large identity spaces IDSp, while the second is more useful for small identity spaces. The games work as follows:

0. Game G_0 executes exactly as in Game ANO-CCA$_{\overline{\mathcal{IBE}}}$, so

$$2 \cdot \Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] - 1 \;=\; \mathbf{Adv}^{\text{ano-cca}}_{\overline{\mathcal{IBE}}}(\mathcal{A})$$

1. Game G_1 is like G_0 except for the description of procedure **Dec**. In G_1, **Dec** returns $\perp$ to $\mathcal{A}$ and sets bad to true whenever $id = id_b^*$, $C = C^*$, $com \neq com^*$, and yet $\mathsf{Open}(cpars, com, dec) = id$ and $R' = R$ (where *dec* and R' are obtained by decrypting C). This means that $\mathcal{A}$ managed to construct a second commitment $com \neq com^*$ that with the same opening information dec^* opens to the same content id_b^*, thereby breaking copy-resistance:

$$\begin{aligned} \Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] &\leq \Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right] + \Pr\left[\, G_1^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\right] \\ \Pr\left[\, G_1^{\mathcal{A}} \text{ sets } \mathsf{bad} \,\right] &\leq \mathbf{Adv}^{\text{copy}}_{\mathcal{CMT}}(\mathcal{B}_1)\ . \end{aligned}$$

2. Game G_2 is similar to G_1 except for the description of procedure **LR**. In G_2, **LR** computes C^* as $\mathsf{Enc}(pars, id_b^*, (0^{|M^*|}, 0^{|dec^*|}, 0^r))$. We have

$$\Pr\left[\,G_1^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \;\le\; \Pr\left[\,G_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \mathbf{Adv}^{\text{ind-cca}}_{I\mathcal{BE}}(\mathcal{B}_2)\;.$$

At this point, we introduce a fork in the game sequence to derive the two separate bounds. We first present the branch G_3–G_5 that is most appropriate for large identity spaces, and then the branch G_3'–G_5' for small identity spaces.

3. Game G_3 is like G_2 except that in the procedure **LR** the target ciphertext is encrypted to a random identity $id^* \xleftarrow{\$} \mathsf{IDSp}$ (or an arbitrarily large subset thereof in case IDSp is infinite), rather than to id_b^*. Consider the adversary $\mathcal{B}_3$ who runs $\mathcal{A}$ in an environment like that of G_2/G_3 and submits identities $id_0' = id^*$ and $id_1' = id_b^*$ as the challenge identities and $(0^{|M^*|}, 0^{|dec^*|}, 0^r)$ as the challenge message in its own game ANO-CCA$_{I\mathcal{BE}}$. The challenge ciphertext C^* is used by $\mathcal{B}_3$ as part of $\mathcal{A}$'s challenge ciphertext (C^*, com^*). For most of $\mathcal{A}$'s **Ext** and **Dec** queries $\mathcal{B}_3$ simply relays answers from its own **Ext** and **Dec** oracles, except when $\mathcal{A}$ queries the **Ext** or **Dec** oracle on id^*. Note that $\mathcal{A}$ is allowed to make such a query, but $\mathcal{B}_3$ is not. If this happens, then $\mathcal{B}_3$ outputs 0. If this doesn't happen and eventually $\mathcal{A}$ guesses the bit b correctly, then $\mathcal{B}_3$ outputs 1, otherwise it outputs 0.

   Let $\beta \in \{0,1\}$ be the bit that $\mathcal{B}_3$ is trying to guess, i.e., C^* is encrypted under id_β'. We have

$$\mathbf{Adv}^{\text{ano-cca}}_{I\mathcal{BE}}(\mathcal{B}_3) \;=\; \Pr\left[\,\mathcal{B}_3 \text{ outputs } 1 \mid \beta = 1\,\right] - \Pr\left[\,\mathcal{B}_3 \text{ outputs } 1 \mid \beta = 0\,\right]\;.$$

   If $\beta = 1$, then C^* is encrypted under id_b^*, and $\mathcal{A}$'s execution environment is exactly as in G_2, unless it makes a query $\mathbf{Ext}(id^*)$ or $\mathbf{Dec}(id^*, \cdot)$. Since $\mathcal{A}$'s view is independent of id^*, however, this happens with probability at most $(q_{\text{id}} + q_{\text{dec}})/|\mathsf{IDSp}|$, so that

$$\Pr\left[\,\mathcal{B}_3 \text{ outputs } 1 \mid \beta = 1\,\right] \;\ge\; \Pr\left[\,G_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] - \frac{q_{\text{id}} + q_{\text{dec}}}{|\mathsf{IDSp}|}\;.$$

   If $\beta = 0$, then let E denote the event that $\mathcal{A}$ makes a query $\mathbf{Ext}(id^*)$ or $\mathbf{Dec}(id^*, \cdot)$. Since $\mathcal{A}$ may recover id^* from C^* we cannot upper-bound $\Pr[\,\mathrm{E}\,]$, but we have that

$$\begin{aligned}
&\Pr\left[\,\mathcal{B}_3 \text{ outputs } 1 \mid \beta = 0\,\right] \\
&\quad= \Pr\left[\,\mathcal{B}_{10} \text{ outputs } 1 \mid \beta = 0 \wedge \mathrm{E}\,\right] \cdot \Pr[\,\mathrm{E}\,] + \Pr\left[\,\mathcal{B}_{10} \text{ outputs } 1 \mid \beta = 0 \wedge \neg\mathrm{E}\,\right] \cdot \Pr[\,\neg\mathrm{E}\,] \\
&\quad\le \Pr\left[\,\mathcal{B}_{10} \text{ outputs } 1 \mid \beta = 0 \wedge \mathrm{E}\,\right] + \Pr\left[\,\mathcal{B}_{10} \text{ outputs } 1 \mid \beta = 0 \wedge \neg\mathrm{E}\,\right] \\
&\quad= 0 + \Pr\left[\,G_3^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right]
\end{aligned}$$

   Putting this together yields

$$\Pr\left[\,G_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \;\le\; \Pr\left[\,G_3^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \mathbf{Adv}^{\text{ano-cca}}_{I\mathcal{BE}}(\mathcal{B}_3) + \frac{q_{\text{id}} + q_{\text{dec}}}{|\mathsf{IDSp}|}\;.$$

4. Game G_4 is like G_3 except for the description of procedure **Dec**. In G_3, **Dec** returns $\perp$ to $\mathcal{A}$ and sets bad to true whenever $id = id_{1-b}^*$, $C = C^*$, $com \neq com^*$ and yet $\mathsf{Open}(cpars, com, dec) = id$ and $R' = R$ (where dec and R' are obtained by decrypting C). As in Game G_3 in the proof of Equation (5) however, the ciphertext C^* is generated almost independently of R. The only

input to the encryption $C^* \xleftarrow{\$} \mathsf{Enc}(pars, id^*, (0^{|M^*|}, 0^{|dec^*|}, 0^r)$ that can possibly carry information about R is the length of the message M^*. Using Lemma C.1 we have that

$$\begin{aligned} \Pr\left[\,\mathrm{G}_3^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] &\leq \Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \\ \Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] &\leq \frac{|M^*|}{2^r}\,. \end{aligned}$$

5. Game G_5 returns $\perp$ to all queries $\mathbf{Dec}(id_d^*, (C^*, com))$ for any $d \in \{0,1\}$, which is in fact merely a syntactical change with respect to Game G_4. The only information about the bit b that is passed to $\mathcal{A}$ is through the commitment com^* in the challenge ciphertext, so it is easy to see that

$$\begin{aligned} \Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] &= \Pr\left[\,\mathrm{G}_5^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \\ 2 \cdot \Pr\left[\,\mathrm{G}_5^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] - 1 &\leq \mathbf{Adv}^{\mathrm{hide}}_{\mathcal{CMT}}(\mathcal{B}_4)\,. \end{aligned}$$

Putting all the above equations from games G_0–G_5 together yields

$$\begin{aligned} \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{A}) \leq\ & 2 \cdot \mathbf{Adv}^{\mathrm{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBE}}(\mathcal{B}_2) + 2 \cdot \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{B}_3) \\ & + \mathbf{Adv}^{\mathrm{hide}}_{\mathcal{CMT}}(\mathcal{B}_4) + \frac{2(q_{\mathrm{id}} + q_{\mathrm{dec}})}{|\mathsf{IDSp}|} + \frac{2|M^*|}{2^r} \end{aligned} \tag{18}$$

We now present the second game sequence G'_3–G'_5 that forks off from game G_2.

3. Game G'_3 is like G_2 except that in the procedure **LR** the target ciphertext is always encrypted to id_0^* instead of to id_b^*. It is straightforward to construct an ANO-CCA adversary $\mathcal{B}'_3$ such that

$$\Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \leq \Pr\left[\,\mathrm{G}_3'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{B}'_3)\,.$$

4. Game G'_4 is like G'_3 except that **Dec** returns $\perp$ to $\mathcal{A}$ and sets bad to true whenever $id = id^*_{1-b}$, $C = C^*$, $com \neq com^*$ and yet $\mathsf{Open}(cpars, com, dec) = id$ and $R' = R$ (where dec and R' are obtained by decrypting C). Again, the ciphertext C^* is generated almost independently of R. The inputs that carry information about R in the ciphertext generation $C^* \xleftarrow{\$} \mathsf{Enc}(pars, id_b^*, (0^{|M^*|}, 0^{|dec^*|}, 0^r)$ are id_b^* and $|M^*|$. Using Lemma C.1 we have that

$$\begin{aligned} \Pr\left[\,\mathrm{G}_3'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] &\leq \Pr\left[\,\mathrm{G}_4'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \Pr\left[\,\mathrm{G}_4'^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \\ \Pr\left[\,\mathrm{G}_4'^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] &\leq 2^{\log|\mathsf{IDSp}| + \log|M^*| - \mathsf{r}} = \frac{|\mathsf{IDSp}| \cdot |\mathsf{M}^*|}{2^r}\,. \end{aligned}$$

5. Game G'_5 returns $\perp$ to all queries $\mathbf{Dec}(id_d^*, (C^*, com))$ for any $d \in \{0,1\}$. The only information about the bit b that is passed to $\mathcal{A}$ is through the commitment com^*, so it is easy to see that

$$\begin{aligned} \Pr\left[\,\mathrm{G}_4'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] &= \Pr\left[\,\mathrm{G}_5'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \\ 2 \cdot \Pr\left[\,\mathrm{G}_5'^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] - 1 &\leq \mathbf{Adv}^{\mathrm{hide}}_{\mathcal{CMT}}(\mathcal{B}'_4)\,. \end{aligned}$$

Putting the inequalities obtained from games G_0–G_2,G'_3–G'_5 together yields

$$\begin{aligned} \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{A}) \leq\ & 2 \cdot \mathbf{Adv}^{\mathrm{copy}}_{\mathcal{CMT}}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBE}}(\mathcal{B}_2) + 2 \cdot \mathbf{Adv}^{\text{ano-cca}}_{\mathcal{IBE}}(\mathcal{B}'_3) \\ & + \mathbf{Adv}^{\mathrm{hide}}_{\mathcal{CMT}}(\mathcal{B}'_4) + \frac{2 \cdot |\mathsf{IDSp}| \cdot |\mathsf{M}^*|}{2^r}\,. \end{aligned} \tag{19}$$

Equation (10) is easily obtained from Equations (18) and (19).

E Proof of the robustness of $\mathcal{CS}^*$ (Theorem 6.1)

The proof relies on Games G_0–G_{11} of Figures 7–9 and the adversary $\mathcal{I}$ of Figure 10. See Section 6 for intuition. The games are written in a compact form where we show individual procedures, writing next to each the games in which it occurs. We assume that every **Dec** query (a_1, a_2, c, d) of $\mathcal{A}$ satisfies $a_1 \neq \mathbf{1}$. This is without loss of generality because the decryption algorithm rejects otherwise. This will be crucial below. Similarly, we assume $(a_1, a_2, c, d) \in \mathbb{G}^4$. We now proceed to the analysis. Game G_0 is simply Game ROB-CCA$_{\mathcal{CS}^*}$, so

$$\mathbf{Adv}^{\text{rob-cca}}_{\mathcal{CS}^*}(\mathcal{A}) \;=\; \Pr\left[\,\mathrm{G}_0 \Rightarrow \mathsf{true}\,\right]. \tag{20}$$

Games $\mathrm{G}_1, \mathrm{G}_2$ start to move us to the alternative decryption rule. In G_1, if $a_2 = a_1^w$ and $d = a_1^{x_b + y_b v}$ then $d = a_1^{x_{b1} + y_{b1} v} a_2^{x_{b2} + y_{b2} v}$, so **Dec** in G_1 returns the correct decryption, like in G_0. If $a_2 \neq a_1^w$ or $d \neq a_1^{x_b + y_b v}$ then, if $d \neq a_1^{x_{b1} + y_{b1} v} \cdot a_2^{x_{b2} + y_{b2} v}$, then **Dec** in G_1 returns $\perp$, else it returns $c a_1^{-z_{b1}} a_2^{-z_{b2}}$, so again is correct either way. Thus,

$$\begin{aligned}
\Pr\left[\,\mathrm{G}_0^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] &= \Pr\left[\,\mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] \\
&= \Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \left(\Pr\left[\,\mathrm{G}_1^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] - \Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right]\right) \\
&\leq \Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true}\,\right] + \Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right], \qquad (21)
\end{aligned}$$

where the last line is by Lemma A.1 since $\mathrm{G}_1, \mathrm{G}_2$ are identical until bad. We now fork off two game chains, one to bound each term above.

First, we will bound the second term in the right-hand side of Inequality (21). Our goal is to move the choices of $x_{b1}, x_{b2}, y_{b1}, y_{b2}, z_{b1}, z_{b2}$ $(b = 0, 1)$ and the setting of bad into **Finalize** while still being able to answer **Dec** queries. We will then be able to bound the probability that bad is set by a static analysis. Consider Game G_3. If $a_2 \neq a_1^w$ and $d = a_1^{x_{b1} + y_{b1} v} a_2^{x_{b2} + y_{b2} v}$ then bad is set in G_2. But $a_2 = a_1^w$ and $d \neq a_1^{x_b + y_b v}$ implies $d \neq a_1^{x_{b1} + y_{b1} v} a_2^{x_{b2} + y_{b2} v}$, so bad is not set in G_2. So,

$$\Pr\left[\,\mathrm{G}_2^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \;=\; \Pr\left[\,\mathrm{G}_3^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right]. \tag{22}$$

Since we are only interested in the probability that G_3 sets bad, we have it always return true. The flag bad may be set at line 315, but is not used, so we move the setting of bad into the **Finalize** procedure in G_4. This requires that G_4 do some bookkeeping. We have also done some restructuring, moving some loop invariants out of the loop in **Dec**. We have

$$\Pr\left[\,\mathrm{G}_3^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \;=\; \Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right]. \tag{23}$$

The choice of x_{b1}, x_{b2}, x_b at lines 404, 405 can equivalently be written as first choosing x_b and x_{b2} at random and then setting $x_{b1} = x_b - w x_{b2}$. This is true because w is not equal to 0 modulo p. The same is true for y_{b1}, y_{b2}, y_b. Once this is done, $x_{b1}, x_{b2}, y_{b1}, y_{b2}$ are not used until **Finalize**, so their choice can be delayed. Game G_5 makes these changes, so we have

$$\Pr\left[\,\mathrm{G}_4^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \;=\; \Pr\left[\,\mathrm{G}_5^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right]. \tag{24}$$

Game G_6 simply writes the test of line 524 in terms of the exponents. Note that this game computes discrete logarithms, but it is only used in the analysis and does not have to be efficient. We have

$$\Pr\left[\,\mathrm{G}_5^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \;=\; \Pr\left[\,\mathrm{G}_6^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right]. \tag{25}$$

We claim that

$$\Pr\left[\,\mathrm{G}_6^{\mathcal{A}} \text{ sets } \mathsf{bad}\,\right] \;\leq\; \frac{2q}{p}, \tag{26}$$

(Recall q is the number of **Dec** queries made by $\mathcal{A}$.) We now justify Equation (26). By the time we

proc Initialize Game G_0

000 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$
001 $K \xleftarrow{\$} \mathsf{Keys}(H)$
002 For $b = 0, 1$ do
003 $x_{b1}, x_{b2}, y_{b1}, y_{b2}, z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$
004 $e_b \leftarrow g_1^{x_{b1}} g_2^{x_{b2}}$
005 $f_b \leftarrow g_1^{y_{b1}} g_2^{y_{b2}}$
006 $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
007 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Initialize Games G_1,G_2,G_3,G_4

100 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$
101 $K \xleftarrow{\$} \mathsf{Keys}(H)$
102 For $b = 0, 1$ do
103 $x_{b1}, x_{b2}, y_{b1}, y_{b2}, z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$
104 $x_b \leftarrow x_{b1} + w x_{b2}$; $y_b \leftarrow y_{b_1} + w y_{b2}$
105 $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
106 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Finalize Games G_0,G_1,G_2

020 Return Win

proc Finalize Game G_3

320 Return true

proc Finalize Game G_4

420 For $b = 0, 1$ do
421 For all $(a_1, a_2, c, d, v) \in S$ do
422 If $d = a_1^{x_{b1}+y_{b1}v} \cdot a_2^{x_{b2}+y_{b2}v}$ Then
423 bad $\leftarrow$ true
424 Return true

proc Dec$((a_1, a_2, c, d))$ Game G_0

010 $v \leftarrow H(K, (a_1, a_2, c))$
011 For $b = 0, 1$ do
012 $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
013 If $d \neq a_1^{x_{b1}+y_{b1}v} \cdot a_2^{x_{b2}+y_{b2}v}$ Then $M_b \leftarrow \perp$
014 If $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$ Then Win $\leftarrow$ true
015 Return (M_0, M_1)

proc Dec$((a_1, a_2, c, d))$ Games $\boxed{\mathrm{G}_1}$,G_2

110 $v \leftarrow H(K, (a_1, a_2, c))$
111 For $b = 0, 1$ do
112 $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
113 If $(a_2 \neq a_1^w \vee d \neq a_1^{x_b+y_b v})$ Then
114 $M_b \leftarrow \perp$
115 If $d = a_1^{x_{b1}+y_{b1}v} \cdot a_2^{x_{b2}+y_{b2}v}$ Then
116 bad $\leftarrow$ true ; $\boxed{M_b \leftarrow c a_1^{-z_{b1}} a_2^{-z_{b2}}}$
117 If $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$ Then Win $\leftarrow$ true
118 Return (M_0, M_1)

proc Dec$((a_1, a_2, c, d))$ Game G_3

310 $v \leftarrow H(K, (a_1, a_2, c))$
311 For $b = 0, 1$ do
312 $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
313 If $(a_2 \neq a_1^w)$ Then
314 $M_b \leftarrow \perp$
315 If $d = a_1^{x_{b1}+y_{b1}v} \cdot a_2^{x_{b2}+y_{b2}v}$ Then bad $\leftarrow$ true
316 Return (M_0, M_1)

proc Dec$((a_1, a_2, c, d))$ Game G_4

410 $v \leftarrow H(K, (a_1, a_2, c))$
411 For $b = 0, 1$ do $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
412 If $(a_2 \neq a_1^w)$ Then
413 $S \leftarrow S \cup \{(a_1, a_2, c, d, v)\}$; $M_0, M_1 \leftarrow \perp$
414 Return (M_0, M_1)

Figure 7: Games $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3$, and G_4 for proof of Theorem 6.1. G_1 includes the boxed code at line 116 but G_2 does not.

proc Initialize Games G_5,G_6

500 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$
501 $K \xleftarrow{\$} \mathsf{Keys}(H)$; $S \leftarrow \emptyset$
502 For $b = 0, 1$ do
503 $x_b, y_b, z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$
504 $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
505 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Dec$((a_1, a_2, c, d))$ Games G_5,G_6

510 $v \leftarrow H(K, (a_1, a_2, c))$
511 For $b = 0, 1$ do $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
512 If $(a_2 \neq a_1^w)$ Then
513 $S \leftarrow S \cup \{(a_1, a_2, c, d, v)\}$; $M_0, M_1 \leftarrow \perp$
514 Return (M_0, M_1)

proc Finalize Game G_5

520 For $b = 0, 1$ do
521 $x_{b2}, y_{b2} \xleftarrow{\$} \mathbb{Z}_p$
522 $x_{b1} \leftarrow x_b - w x_{b2}$; $y_{b1} \leftarrow y_b - w y_{b2}$
523 For all $(a_1, a_2, c, d, v) \in S$ do
524 If $d = a_1^{x_{b1} + y_{b1} v} \cdot a_2^{x_{b2} + y_{b2} v}$ Then $\mathsf{bad} \leftarrow \mathsf{true}$
525 Return true

proc Finalize Game G_6

620 For $b = 0, 1$ do
621 $x_{b2}, y_{b2} \xleftarrow{\$} \mathbb{Z}_p$
622 $x_{b1} \leftarrow x_b - w x_{b2}$; $y_{b1} \leftarrow y_b - w y_{b2}$
623 For all $(a_1, a_2, c, d, v) \in S$ do
624 $u_1 \leftarrow \log_{g_1}(a_1)$; $u_2 \leftarrow \log_{g_2}(a_2)$
625 $s \leftarrow \log_{g_1}(d)$; $t_b \leftarrow s - u_1 x_b + u_1 y_b v$
626 $\alpha \leftarrow w(u_2 - u_1)$; $\beta \leftarrow wv(u_2 - u_1)$
627 If $t_b = \alpha x_{b2} + \beta y_{b2}$ Then $\mathsf{bad} \leftarrow \mathsf{true}$
628 Return true

Figure 8: Games G_5 and G_6 for proof of Theorem 6.1.

reach **Finalize** in G_6, we can consider the adversary coins, all random choices of **Initialize**, and all random choices of **Dec** to be fixed. We will take probability only over the choice of x_{b2}, y_{b2} made at line 621. Consider a particular $(a_1, a_2, c, d, v) \in S$. This is now fixed, and so are the quantities $u_1, u_2, s, t_0, t_1, \alpha$ and β as computed at lines 624–626. So we want to bound the probability that bad is set at line 627 when we regard t_b, α, β as fixed and take the probability over the random choices of x_{b2}, y_{b2}. The crucial fact is that $u_2 \neq u_1$ because $(a_1, a_2, c, d, v) \in S$, and lines 612, 613 only put a tuple in S if $a_2 \neq a_1^w$. So α and β are not 0 modulo p, and the probability that $t_b = \alpha x_{b2} + \beta y_{b2}$ is thus $1/p$. The size of S is at most q so line 627 is executed at most $2q$ times. Equation (26) follows from the union bound.

We now return to Equation (21) to bound the first term. Game G_7 removes from G_2 code that does not affect outcome of the game. Once this is done, $x_{b1}, y_{b1}, x_{b2}, y_{b2}$ are used only to define x_b, y_b, so G_7 picks only the latter. So we have

$$\Pr\left[\mathrm{G}_2^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \;=\; \Pr\left[\mathrm{G}_7^{\mathcal{A}} \Rightarrow \mathsf{true}\right] . \tag{27}$$

Game G_8 is the same as G_7 barring setting a flag that does not affect the game outcome, so

$$\begin{aligned}
\Pr\left[\mathrm{G}_7^{\mathcal{A}} \Rightarrow \mathsf{true}\right] &= \Pr\left[\mathrm{G}_8^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \\
&= \Pr\left[\mathrm{G}_9^{\mathcal{A}} \Rightarrow \mathsf{true}\right] + \Pr\left[\mathrm{G}_8^{\mathcal{A}} \Rightarrow \mathsf{true}\right] - \Pr\left[\mathrm{G}_9^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \\
&\leq \Pr\left[\mathrm{G}_9^{\mathcal{A}} \Rightarrow \mathsf{true}\right] + \Pr\left[\mathrm{G}_8^{\mathcal{A}} \text{ sets } \mathsf{bad}\right] \qquad (28) \\
&\leq \Pr\left[\mathrm{G}_9^{\mathcal{A}} \Rightarrow \mathsf{true}\right] + \frac{1}{p} . \qquad (29)
\end{aligned}$$

Equation (28) is by Lemma A.1 since G_8, G_9 are identical until bad. The probability that G_8 sets bad is the probability that $y_1 = y_0$ at line 805, and this is $1/p$ since y is chosen at random from $\mathbb{Z}_p$, justifying Equation (29). The distribution of y_1 in G_9 is always uniform over $\mathbb{Z}_q - \{y_0\}$, and the setting

proc Initialize Game G_7

700 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$
701 $K \xleftarrow{\$} \mathsf{Keys}(H)$
702 For $b = 0, 1$ do
703 $x_b, y_b, z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$
704 $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
705 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Dec$((a_1, a_2, c, d))$ Games G_7–G_{11}

710 $v \leftarrow H(K, (a_1, a_2, c))$
711 For $b = 0, 1$ do
712 $M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
713 If $(a_2 \neq a_1^w \vee d \neq a_1^{x_b + y_b v})$ Then $M_b \leftarrow \perp$
714 If $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$ Then $\textsc{Win} \leftarrow \mathsf{true}$
715 Return (M_0, M_1)

proc Finalize Games G_7–G_{11}

720 Return $\textsc{Win}$

proc Initialize Game G_8/$\boxed{G_9}$

800 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$; $K \xleftarrow{\$} \mathsf{Keys}(H)$
801 For $b = 0, 1$ do
802 $x_b, y_b, z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$
803 $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
804 If $y_1 = y_0$ Then
805 $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{y_1 \xleftarrow{\$} \mathbb{Z}_q - \{y_0\}}$
806 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Initialize Game G_{10}

1000 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$; $K \xleftarrow{\$} \mathsf{Keys}(H)$
1001 $x_0, y_0, x_1 \xleftarrow{\$} \mathbb{Z}_q$; $y_1 \xleftarrow{\$} \mathbb{Z}_q - \{y_0\}$
1002 For $b = 0, 1$ do
1003 $z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$; $e_b \leftarrow g_1^{x_b}$
1004 $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
1005 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

proc Initialize Game G_{11}

1100 $g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$; $K \xleftarrow{\$} \mathsf{Keys}(H)$; $v^* \xleftarrow{\$} \mathbb{Z}_q$
1101 $x_0, y_0 \xleftarrow{\$} \mathbb{Z}_q$; $y_1 \xleftarrow{\$} \mathbb{Z}_q - \{y_0\}$; $x_1 \leftarrow x_0 - (y_1 - y_0)v^*$
1102 For $b = 0, 1$ do $z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$; $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
1103 Return $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$

Figure 9: Games G_7–G_{11} for proof of Theorem 6.1. G_9 includes the boxed code at line 805 but G_8 does not.

Adversary $\mathcal{I}(K, v^*)$
$g_1 \xleftarrow{\$} \mathbb{G}^*$; $w \xleftarrow{\$} \mathbb{Z}_p^*$; $g_2 \leftarrow g_1^w$; $x_0, y_0 \xleftarrow{\$} \mathbb{Z}_p$; $y_1 \xleftarrow{\$} \mathbb{Z}_p - \{y_0\}$; $x_1 \leftarrow x_0 - (y_1 - y_0)v^*$
For $b = 0, 1$ do
$z_{b1}, z_{b2} \xleftarrow{\$} \mathbb{Z}_p$; $e_b \leftarrow g_1^{x_b}$; $f_b \leftarrow g_1^{y_b}$; $h_b \leftarrow g_1^{z_{b1}} g_2^{z_{b2}}$
Run $\mathcal{A}$ on $(g_1, g_2, K), (e_0, f_0, h_0), (e_1, f_1, h_1)$
On query $\mathbf{Dec}((a_1, a_2, c, d))$
$v \leftarrow H(K, (a_1, a_2, c))$
For $b = 0, 1$ do
$M_b \leftarrow c \cdot a_1^{-z_{b1}} a_2^{-z_{b2}}$
If $(a_2 \neq a_1^w \vee d \neq a_1^{x_b + y_b v})$ Then $M_b \leftarrow \perp$
If $(M_0 \neq \perp) \wedge (M_1 \neq \perp)$ Then $(a_1^*, a_2^*, c^*) \leftarrow (a_1, a_2, c)$
Return (M_0, M_1) to $\mathcal{A}$
Until $\mathcal{A}$ halts
Return (a_1^*, a_2^*, c^*)

Figure 10: Adversary $\mathcal{I}$ for proof of Theorem 6.1.

of bad at line 805 does not affect the game outcome, so

$$\Pr\left[\mathrm{G}_9^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \;=\; \Pr\left[\mathrm{G}_{10}^{\mathcal{A}} \Rightarrow \mathsf{true}\right] . \tag{30}$$

Game G_{11} picks x_b, y_b differently from G_{10}, but since $y_1 - y_0 \neq 0$, the two ways induce the same distribution on x_0, x_1, y_0, y_1. Thus,

$$\Pr\left[\mathrm{G}_{10}^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \;=\; \Pr\left[\mathrm{G}_{11}^{\mathcal{A}} \Rightarrow \mathsf{true}\right] . \tag{31}$$

We now claim that

$$\Pr\left[\mathrm{G}_{11}^{\mathcal{A}} \Rightarrow \mathsf{true}\right] \;\leq\; \mathbf{Adv}_H^{\text{pre-img}}(\mathcal{I}) \tag{32}$$

where $\mathcal{I}$ is depicted in Figure 10. To justify this, say that the $\mathcal{A}$ makes a **Dec** query (a_1, a_2, c, d) which returns (M_0, M_1) with $M_0 \neq \perp$ and $M_1 \neq \perp$. This means we must have

$$d \;=\; a_1^{x_0 + y_0 v} \;=\; a_1^{x_1 + y_1 v} , \tag{33}$$

where $v = H(K, (a_1, a_2, c))$. Let $u_1 = \log_{g_1}(a_1)$ and $s = \log_{g_1}(d)$. Now, the above implies $u_1(x_0 + y_0 v) = u_1(x_1 + y_1 v)$. But (a_1, a_2, c, d) is a **Dec** query, and we know that $a_1 \neq \mathbf{1}$, so $u_1 \neq 0$. (This is a crucial point. Recall the reason we can without loss of generality assume $a_1 \neq \mathbf{1}$ is that the decryption algorithm of $\mathcal{CS}^*$ rejects otherwise.) Dividing u_1 out, we get $x_0 + y_0 v = x_1 + y_1 v$. Rearranging terms, we get $(y_1 - y_0)v = x_0 - x_1$. However, we know that $y_1 \neq y_0$, so $v = (y_1 - y_0)^{-1}(x_0 - x_1)$. However, this is exactly the value v^* due to the way $\mathcal{I}$ and Game G_{11} define x_0, y_0, x_1, y_1. Thus, we have $H(K, (a_1, a_2, c)) = v^*$, meaning $\mathcal{I}$ will be successful. Putting together Equations (20)–(27), (29)–(32) concludes the proof of Theorem 6.1.

F Application to auctions

Robustness of ElGamal. The parameters of the ElGamal encryption scheme consist of the description of a group $\mathbb{G}$ of prime order p with generator g. The secret key of a user is $x \xleftarrow{\$} \mathbb{Z}_p$, the corresponding public key is $X = g^x$. The encryption of a message M is the pair $(g^r, X^r \cdot M)$ for $r \xleftarrow{\$} \mathbb{Z}_p$. A ciphertext (R, S) is decrypted as $M \leftarrow R/S^x$. Since the decryption algorithm never returns $\perp$, the

ElGamal scheme is obviously not robust. Stronger even, the ciphertext $(1, M)$ decrypts to M under any secret key. It is this strong failure of robustness that opens the way to attacks on applications like Sako's auction protocol [Sak00].

THE PROTOCOL. Sako's auction protocol [Sak00] is important because it is the first truly practical one to hide the bids of losers. Let $1, \ldots, N$ be the range of possible bidding prices. In an initialization step, the auctioneer generates N ElGamal key pairs $(x_1, X_1), \ldots, (x_N, X_N)$, and publishes $g, X_1, \ldots, X_N$ and a fixed message $M \in \mathbb{G}$. A bidder places a bid of value $v \in \{1, \ldots, N\}$ by encrypting M under X_v and posting the ciphertext. Note that the privacy of the bids is guaranteed by the anonymity of ElGamal encryption. The authority opens bids $C_1 = (R_1, S_1), \ldots, C_n = (R_n, S_n)$ by decrypting all bids under secret keys $x_N, \ldots, x_1$, until the highest index w where one or more bids decrypt to M. The auctioneer announces the identity of the winner(s), the price of the item w, and the secret key x_w. All auctioneers can then check that $S_i/R_i^{x_w} = M$ for all winners i.

AN ATTACK. Our attack permits a dishonest bidder and a colluding auctioneer to break the fairness of the protocol. (Security against colluding auctioneers was not considered in [Sak00], so we do not disprove their results, but it is a property that one may expect the protocol to have.) Namely, a cheating bidder can place a bid $(1, M)$. If w is the highest honest bid, then the auctioneer can agree to open the corrupted bid to with x_{w+1}, thereby winning the auction for the cheating bidder at one dollar more than the second-highest bidder.

Sako came close to preventing this attack with an "incompatible encryption" property that avoids choosing $r = 0$ at encryption. A dishonest bidder however may deviate from this encryption rule; the problem is that the decryption algorithm does not reject ciphertexts (R, S) when $R = 1$. The attack is easily prevented by using any of our robust encryption schemes, so that decryption under any other secret key than the intended one results in $\perp$ being returned. Note that for this application we really need the strong robustness notion with adversarially generated ciphertexts.

It is worth noting that, to enforce that all bids are independent of each other even in the presence of a colluding auctioneer, all bidders would also need to commit to their sealed bids (using a non-malleable commitment scheme) during a first round of communication and only open their commitments once all commitments made public.