A Separation between the Random-Oracle Model and the Standard Model for a Hybrid-Encryption Problem

Mihir Bellare* Alexandra Boldyreva† Adriana Palacio‡

February 10, 2003

Abstract

We present a simple, practical, natural RO-model scheme that is proven in the RO model to meet its goal and yet admits *no* standard-model instantiation that meets this goal. The goal in question is *IND-CCA-preserving asymmetric encryption* which formally captures security of the most common practical usage of asymmetric encryption, namely to transport a symmetric key in such a way that symmetric encryption under the latter remains secure. The scheme is an El Gamal variant, called Hash El Gamal, that resembles numerous existing RO-model schemes, and on the surface shows no evidence of its anomalous properties.

We obtain these results as a consequence of a more general one showing that a certain type of IND-CCA-preserving asymmetric encryption is impossible to achieve in the standard model but is achievable (by Hash El Gamal in particular) in the RO model. This helps us better understand the source of the anomalies in Hash El Gamal and also lifts separation results from being about specific, example schemes to being about entire goals.

We believe these results deepen our understanding of the nature and extent of the gap between the standard and RO models, and bring concerns raised by previous work closer to practice by indicating that the problem of RO-model schemes admitting no secure instantiation is a very real one that can and does arise in domains where RO schemes are commonly designed.

Keywords: Random-Oracle Model, asymmetric encryption, hybrid encryption, foundations.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF grant CCR-0098123, NSF grant ANR-0129617 and an IBM Faculty Partnership Development Award.

†Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: aboldyre@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/aboldyre. Supported in part by above-mentioned grants of first author.

‡Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: apalacio@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/apalacio. Supported by a National Science Foundation Graduate Research Fellowship.

Contents

1 Introduction

A random oracle (RO) model scheme is one whose algorithms have oracle access to one or more public random functions. (Its security is evaluated with respect to an adversary with oracle access to the same functions.) An "instantiation" of such a scheme is the standard-model scheme obtained by replacing this function with a polynomial-time computable function having a public description. (Its security is evaluated with respect to an adversary given the description of the same function.) In the random-oracle paradigm, as enunciated by Bellare and Rogaway [2], one first designs and proves secure a scheme in the RO model, and then instantiates it to get a (hopefully still secure) standard-model scheme.

The RO model has proven quite popular and there are now numerous practical schemes designed and proven secure in this model. But the important issue of how such schemes can be securely instantiated, and whether this is even possible, remains less clear.

This paper adds to existing concerns in this regard via new "separation" results. We present a simple, natural RO-model scheme, proven in the RO model to meet an eminently practical goal. Yet we show that *no* instantiation of this scheme results in a standard-model scheme meeting the goal in question.

This is not the first separation result (cf. [6, 17]), but the nature of the scheme and goal in our case bring the concerns raised by the previous results closer to practice and indicate that the problem of RO-model schemes admitting no secure instantiation is a very real one that can and does arise in domains where RO schemes are commonly designed.

We also prove further and stronger results that lift separations from being about specific schemes to being about primitives. Below we begin with some background and then describe our contributions in more detail.

Separation results. A separation result for some cryptographic goal is an example of a RO-model scheme provably meeting the goal in question but admitting *no* instantiation that meets the goal. The first such results are due to Canetti, Goldreich and Halevi [6], the goals in question being IND-CPA-secure asymmetric encryption and digital signatures secure against chosen-message attack. Nielsen [17] followed with a separation result for the goal of non-interactive, non-committing encryption (NCE) [5].

However, the schemes of [6] are somewhat artificial (meaning, not like practical schemes one typically encounters) and also complex through the use of CS proofs [16]. The scheme of [17], on the other hand, is natural but the goal considered (namely, non-interactive NCE) is not as practical as the one we consider.[1]

In recent work that is independent of and concurrent to ours, Goldwasser and Taumann [15] present a separation result about the Fiat-Shamir transform, showing the existence of a 3-move protocol which when collapsed via a hash function yields a signature that is secure in the RO model but possesses no secure instantiation. This result addresses an important practical problem but their example protocol, like those of [6], is a comparitively artificial and complex CS-proof based one.

In our separation results, both the scheme and the goal are natural and practical. Let us begin by describing the goal.

[1] An additional concern is that Nielsen's separation [17] might be the result of having incorrectly lifted the standard-model definition of the NCE goal [5] to the random-oracle model. We recall that the definition of NCE is in Canetti's multi-party computation framework [4], which uses the notion of an "environment." In moving to the RO model, Nielsen denies the random oracle to the environment. But the definition of the RO model is that all parties, including the adversary, have access to the random oracle, and in [4] the environment is an adversary. If one does provide the random oracle to the environment, however, the separation result in [17] vanishes.

IND-CCA-PRESERVING ASYMMETRIC ENCRYPTION. We consider the use of an asymmetric encryption scheme for the purpose for which it is overwhelmingly employed in practice, namely to transport a key that is later used for the symmetric encryption of one or more messages. Such a "hybrid" scheme is secure if symmetric encryption under the transported key remains as secure as if this key had been distributed to the parties magically and out-of-band. Based on this intuition, we present in Section 2 a natural formalization of the IND-CCA security of the *multi-message (mm) hybrid scheme* associated to an asymmetric encryption scheme AS and a symmetric encryption scheme SS.[2]

Good cryptographic-engineering practice suggests that one require the asymmetric encryption scheme to have the property that security of the mm-hybrid relies only on the assumption that the underlying symmetric encryption scheme is itself secure. This leads to our saying that an asymmetric encryption scheme AS is *IND-CCA preserving* if the mm-hybrid associated to AS and symmetric encryption scheme SS is IND-CCA secure for *every* IND-CCA secure SS. The goal we consider is IND-CCA-preserving asymmetric encryption.

Note that any IND-CCA-secure asymmetric encryption scheme is IND-CCA preserving (cf. [10, 19]). However IND-CCA preservation is actually a weaker requirement on an asymmetric encryption scheme than IND-CCA security itself, leading researchers to seek IND-CCA-preserving asymmetric encryption schemes that are more efficient than existing IND-CCA secure ones. These designs tend to be in the RO model. We now consider one such design.

THE HASH EL GAMAL SCHEME AND ITS SECURITY. It is easy to see that the El Gamal encryption scheme [11] is not IND-CCA preserving. An effort to strengthen it to be IND-CCA preserving lead us to a variant that we call the Hash El Gamal scheme. It uses the idea underlying the Fujisaki-Okamoto [12] transformation, namely to encrypt under the original (El Gamal) scheme using coins obtained by applying a random oracle H to the message. Specifically, encryption of a message K under public key (q, g, X) in the Hash El Gamal scheme is given by

$$\mathsf{AE}^{G,H}((q, g, X), K) \;=\; (g^{H(K)}\,,\, G(X^{H(K)}) \oplus K)\,, \tag{1}$$

where G, H are random oracles, $q, 2q+1$ are primes, g is a generator of the order q cyclic subgroup of $\mathbb{Z}^*_{2q+1}$, and the secret key is (q, g, x) where $g^x = X$. Decryption is performed in the natural way as detailed in Figure 1.

The transformation turns out to "work" in the sense that we are able to prove Theorem 3.1, which says that the Hash El Gamal scheme is IND-CCA preserving in the RO model, assuming the CDH problem is hard in the underlying group. (The proof, which is not entirely straightforward, is outlined briefly in Section 3 and detailed in Appendix A.)

However, we follow this with Theorem 4.5 which says that the Hash El Gamal scheme admits no IND-CCA-preserving instantiation. In other words, the standard-model asymmetric encryption scheme obtained by instantiating the RO-model Hash El Gamal is not IND-CCA preserving,[3] regardless of the choice of instantiating functions. (We allow these to be drawn from any family of polynomial-time computable functions.)

DISCUSSION. What we find scary about the above is that, on the surface of it, the Hash El Gamal scheme seems innocuous enough. It does not seem to be making any "peculiar" use of its random

[2] The term "hybrid encryption" in the literature seems to cover a large body of goals and techniques. We are interested in a particular sub-class of these, and to avoid confusion provide a distinguishing name and definition of security. The term mm-hybrid reflects the fact that multiple messages may be symmetrically encrypted under a single transported symmetric key.

[3] This is under the assumption that IND-CCA-secure symmetric encryption schemes exist, since, otherwise, by default, *any* asymmetric encryption scheme is IND-CCA preserving, and, indeed, the entire hybrid encryption problem we are considering is vacuous. This assumption is made implicitly in all results in this paper.

oracle that would lead us to think it is "wrong;" indeed, it uses random oracles in ways they have been used previously, in particular by [12]. The scheme is simple, efficient, and similar to other RO-model schemes out there. The fact that this scheme is in some very real sense "wrong" points to the difficulty of being able to distinguish such RO-model schemes from ones that at least *may* be securely instantiable.

The bulk of papers that use the RO model stop at the point of proving that their scheme is secure in this model, with little or no attention to instantiation, barring perhaps a line saying it should be done using SHA-1 or other cryptographic hash functions as suggested in [2]. (As an exception we note [13].) The above results suggest that after having proven a scheme secure in the RO model, designers should look more closely at the possibility of secure instantiation.

GENERALIZATIONS. Underlying Theorem 4.5 (recall this is the result that says that the Hash El Gamal asymmetric encryption scheme admits no IND-CCA-preserving instantiation) are stronger and more general results from which it follows. These are of independent interest. They provide a deeper indication of the nature and extent of the gap between the standard and RO models, and they help pinpoint the source of the anomalous behavior of the Hash El Gamal scheme.

The first of these results is Theorem 4.4, which says that no RO-model asymmetric encryption scheme possessing a pair of properties that we call *key verifiability* and *ciphertext verifiability* admits an IND-CCA-preserving instantiation. (Key verifiability means there is a way to recognize valid public keys in polynomial time. Ciphertext verifiability means there is a polynomial-time procedure to determine whether a given ciphertext is an encryption of a given message under a given valid public key.) Theorem 4.5 follows once we observe (cf. Proposition 4.1) that the Hash El Gamal scheme has the two properties in question.

This helps to better understand what aspects of the Hash El Gamal scheme lead to its admitting no IND-CCA-preserving instantiation. In particular we see that this is not due to some "peculiar" use of random oracles but rather due to some simply stated properties of the resulting asymmetric encryption scheme itself.

Theorem 4.4 is itself a corollary of a yet stronger and more general result, namely Theorem 4.3. The latter is an "impossibility" result purely in the standard model. It says that, in this model, there simply do not exist asymmetric encryption schemes that are all of the following: IND-CCA preserving, key verifiable and ciphertext verifiable. Theorem 4.5 follows once we observe (cf. Proposition 4.2) that the key verifiability and ciphertext verifiability of a RO-model asymmetric encryption scheme are inherited by its standard-model instantiations.

Theorem 4.3 lifts separation results from being about specific, example schemes to being about entire goals. It says that a certain goal (namely IND-CCA-preserving, key-verifiable and ciphertext-verifiable asymmetric encryption) is unachievable in the standard model. Yet, as a consequence of Theorem 3.1 and Proposition 4.1, we know that this goal is achievable in the RO model under standard computational assumptions. This points to a broader gap between the RO and standard models, and says that the RO model can at times lead us not just to "wrong" schemes, but even to "wrong" goals.

RELATED WORK. The large body of work on hybrid encryption in the RO model [7, 8, 12, 18] is an important backdrop for our work, and in particular the Hash El Gamal scheme is based on RO-model schemes and techniques from this literature. We stress, however, that we have no reason to believe that any of these schemes fail to be securely instantiable.

2 Definitions

NOTATION. If S is a randomized algorithm, then $[S(x, y, \ldots)]$ denotes the set of all points having

positive probability of being output by S on inputs $x, y, \ldots$. If n is an integer, then $\langle n \rangle$ is its representation as a binary string. If x is a binary string, then $|x|$ denotes its length.

SYMMETRIC ENCRYPTION. A symmetric encryption scheme $\mathsf{SS} = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ is specified by three polynomial-time algorithms: via $K \xleftarrow{\$} \mathsf{SK}(1^k)$ one can generate a key; via $C \xleftarrow{\$} \mathsf{SE}(K, M)$ one can encrypt a message $M \in \{0,1\}^*$; and via $M \leftarrow \mathsf{SD}(K, C)$ one can decrypt a ciphertext C. We assume (without loss of generality) that $[\mathsf{SK}(1^k)] \subseteq \{0,1\}^k$. Following [1], SS is said to be IND-CCA secure if the function

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) \;=\; 2 \cdot \Pr\left[\; K \xleftarrow{\$} \mathsf{SK}(1^k)\;;\; b \xleftarrow{\$} \{0,1\} \;:\; \boldsymbol{S}^{\mathsf{SE}(K,\mathrm{LR}(\cdot,\cdot,b)),\mathsf{SD}(K,\cdot)}(1^k) = b \;\right] - 1$$

is negligible for all legitimate polynomial-time adversaries $\boldsymbol{S}$, where $\mathrm{LR}(M_0, M_1, b) = M_b$ for all messages M_0, M_1 of equal length, and $\boldsymbol{S}$ is legitimate if it never queries $\mathsf{SD}(K, \cdot)$ with a ciphertext previously returned by $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$. We do not consider symmetric encryption schemes in the RO model (both for simplicity and because public random oracles are of little pragmatic value in this setting), but our results extend to cover them.

ASYMMETRIC ENCRYPTION. An asymmetric encryption scheme $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ is specified by three polynomial-time algorithms: via $(pk, sk) \xleftarrow{\$} \mathsf{AK}(1^k)$ one can generate keys; via $C \xleftarrow{\$} \mathsf{AE}(pk, K)$ one can encrypt a message $K \in \{0,1\}^k$; and via $K \leftarrow \mathsf{AD}(sk, C)$ one can decrypt a ciphertext C. (We denote the message by K because we will set it to a key for a symmetric encryption scheme.) In the RO model, the encryption and decryption algorithms have access to one or more oracles, of appropriate domains and ranges that might depend on the public key. Discussions in this paper might refer to standard notions of security for such schemes like IND-CPA and IND-CCA but, since our results do not require them, we do not recall the formal definitions.

IND-CCA-PRESERVING ASYMMETRIC ENCRYPTION. We provide the formal definitions first and explanations later. A *multi-message hybrid (mm-hybrid) encryption scheme* is simply a pair $(\mathsf{AS}, \mathsf{SS})$ consisting of an asymmetric encryption scheme $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ and a symmetric encryption scheme $\mathsf{SS} = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$. We say that mm-hybrid scheme $(\mathsf{AS}, \mathsf{SS})$ is IND-CCA secure if the function

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k) \;=\; 2 \cdot \Pr\left[\; \mathbf{Exp}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k) = 1 \;\right] - 1$$

is negligible for all legitimate polynomial-time *hybrid adversaries* $\boldsymbol{H}$, where the experiment in question is:

$(pk, sk) \xleftarrow{\$} \mathsf{AK}(1^k)\;;\; K \xleftarrow{\$} \mathsf{SK}(1^k)\;;\; b \xleftarrow{\$} \{0,1\}$
Pick random oracles $G, H, \ldots$ with appropriate domains and ranges
$C_a \xleftarrow{\$} \mathsf{AE}^{G,H,\cdots}(pk, K)$
Run $\boldsymbol{H}$ with inputs pk, C_a and oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$, $\mathsf{SD}(K, \cdot)$, $\mathsf{AD}^{G,H,\cdots}(sk, \cdot)$, $G, H, \ldots$
Let d denote the output of $\boldsymbol{H}$
If $b = d$ then return 1 else return 0.

The adversary is legitimate if it does not query $\mathsf{SD}(K, \cdot)$ on a ciphertext previously returned by $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$, and it does not query $\mathsf{AD}(sk, \cdot)$ on C_a. We say that AS is *IND-CCA preserving* if the mm-hybrid encryption scheme $(\mathsf{AS}, \mathsf{SS})$ is IND-CCA secure for *all* IND-CCA-secure symmetric encryption schemes SS.

Let us now explain the ideas behind these formalisms. Recall that we are modelling the security of the following two-phase scenario: in phase one, the sender picks a key K for symmetric encryption, asymmetrically encrypts it under the receiver's public key to get a ciphertext C_a, and sends C_a to the receiver; in phase two, the sender symmetrically encrypts messages of its choice under K and

$\mathsf{AK}(1^k)$	$\mathsf{AE}^{G,H}((q,g,X),K)$	$\mathsf{AD}^{G,H}((q,g,x),(Y,W))$
$(q,g) \xleftarrow{\$} \mathsf{CG}(1^k)$	$y \leftarrow H(K)$	$T \leftarrow G(Y^x)$
$x \xleftarrow{\$} \mathbb{Z}_q$	$Y \leftarrow g^y$	$K \leftarrow T \oplus W$
$X \leftarrow g^x$	$T \leftarrow G(X^y)$	If $g^{H(K)} = Y$ then
Return $((q,g,X),(q,g,x))$	$W \leftarrow T \oplus K$	Return K
	Return (Y,W)	else Return $\perp$ EndIf

Figure 1: Algorithms of the RO-model asymmetric encryption scheme $\mathsf{HEG}[\mathsf{CG}] = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ associated to cyclic-group generator CG. Here $G\colon \langle g\rangle \to \{0,1\}^k$ and $H\colon \{0,1\}^k \to \mathbb{Z}_q$ are random oracles.

transmits the resulting ciphertexts to the receiver. The definitions above capture the requirement of privacy of the symmetrically encrypted data under a chosen-ciphertext attack. Privacy is formalized in terms of indistinguishability via left-or-right oracles, and the chosen-ciphertext attack is formalized via the adversary's access to decryption oracles for *both* the symmetric and asymmetric schemes. The legitimacy requirement, as usual, disallows decryption queries on challenge ciphertexts since they would lead to trivial adversary victory. The experiment reflects the possibility that AS is a RO-model scheme by picking random oracles for AE and AD. (For maximum generality, the oracles are chosen after key generation since in many practical RO-model schemes, including Hash El Gamal, their domains and ranges depend on pk.) The standard model is the special case where the algorithms of AS do not refer to any oracles, and thus the definition above covers security in both models. The notion of AS being IND-CCA preserving reflects a valuable pragmatic requirement, namely that one may use, in conjunction with AS, any symmetric encryption scheme and be guaranteed security of the mm-hybrid under the minimal assumption that the symmetric scheme itself was secure.

3 The HEG scheme and its security in the RO model

We introduce a variant of the El Gamal encryption scheme [11] that we show is IND-CCA preserving in the RO model under a standard assumption, but which, in Section 4, we will show to admit no IND-CCA-preserving instantiation.

Preliminaries. A *cyclic-group generator* is a randomized, polynomial-time algorithm CG which on input 1^k outputs a pair (q,g), where q is a prime such that $p = 2q+1$ is also a prime, g is a generator of the cyclic, order q subgroup $\langle g\rangle$ of $\mathbb{Z}_p^*$, and $|\langle p\rangle| = k$. Recall that the Computational Diffie-Hellman (CDH) problem is said to be hard for CG if the function

$$\mathsf{Adv}^{\mathrm{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) \;=\; \Pr\left[\, (q,g) \xleftarrow{\$} \mathsf{CG}(1^k)\,;\; x,y \xleftarrow{\$} \mathbb{Z}_q \;:\; \boldsymbol{C}(q,g,g^x,g^y) = g^{xy} \,\right]$$

is negligible for all polynomial-time *cdh adversaries* $\boldsymbol{C}$.

Scheme and result statement. To any cyclic-group generator CG we associate the RO-model asymmetric encryption scheme $\mathsf{HEG}[\mathsf{CG}] = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ whose constituent algorithms are depicted in Figure 1. We call this variant of the El Gamal encryption scheme the *Hash El Gamal* encryption scheme associated to CG. The main result about its security in the RO model is the following:

Theorem 3.1 If the CDH problem is hard for cyclic-group generator CG, then the associated Hash El Gamal asymmetric encryption scheme $\mathsf{HEG}[\mathsf{CG}]$ is IND-CCA preserving in the RO model.

For the definition of what it means to be IND-CCA preserving, we refer the reader to Section 2.

REMARKS. We note that the encryption algorithm AE of $\mathsf{HEG}[\mathsf{CG}]$ is deterministic. For this reason alone, $\mathsf{HEG}[\mathsf{CG}]$, as a stand-alone asymmetric encryption scheme, is not IND-CCA secure or even IND-CPA secure in the RO model. Nonetheless, the above says that it is IND-CCA preserving as long as the CDH problem is hard for CG. This is not a contradiction. Very roughly, the reason $\mathsf{HEG}[\mathsf{CG}]$ can preserve IND-CCA while not itself being even IND-CPA is that the former notion considers the use of the scheme only for the encryption of messages that are symmetric keys, which (as long as the associated symmetric encryption scheme is secure) have relatively high entropy, and the entropy in these messages compensates for the lack of any introduced by AE. Furthermore, the $\mathsf{HEG}[\mathsf{CG}]$ scheme and Theorem 3.1 are in line with previous work [7, 8, 12, 18] where also relatively weak asymmetric components suffice to ensure strong security properties of the hybrid based on them.

The full proof of Theorem 3.1 is quite technical and is in Appendix A. Below we provide an intuitive overview that highlights the main areas of novelty.

PROOF SETUP. Let $\mathsf{AS} = \mathsf{HEG}[\mathsf{CG}]$ and let $\mathsf{AK}, \mathsf{AE}, \mathsf{AD}$ denote its constituent algorithms. Let $\mathsf{SS} = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ be any IND-CCA-secure symmetric encryption scheme. We need to show that $(\mathsf{AS}, \mathsf{SS})$ is an IND-CCA-secure mm-hybrid encryption scheme.

Let $\boldsymbol{H}$ be a polynomial-time hybrid adversary attacking $(\mathsf{AS}, \mathsf{SS})$. We will construct polynomial-time adversaries $\boldsymbol{S}, \boldsymbol{C}$ such that

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k) \;\le\; \text{poly}(k) \cdot \text{poly}\left(\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)\,,\, \mathsf{Adv}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) \right) + \frac{\text{poly}(k)}{2^k}\ . \tag{2}$$

Since SS is assumed IND-CCA secure and the CDH problem is hard for CG, the advantage functions related to $\boldsymbol{S}, \boldsymbol{C}$ above are negligible, and thus so is the advantage function related to $\boldsymbol{H}$. To complete the proof, we need to specify adversaries $\boldsymbol{S}, \boldsymbol{C}$ for which Equation (2) is true. Below we let GH be the event that there is a time at which g^{xy} is queried to G but K has not been queried to H; HG the event that there is a time at which K is queried to H but g^{xy} has not been queried to G; and $\mathsf{Succ}(\boldsymbol{H})$ the event that $\boldsymbol{H}$ is successful at guessing the value of its challenge bit b. We will construct $\boldsymbol{C}$ so that

$$\Pr[\,\mathsf{GH}\,] \;\le\; \text{poly}(k) \cdot \mathsf{Adv}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) + \frac{\text{poly}(k)}{2^k}\ ,$$

and we will construct $\boldsymbol{S}$ so that

$$\Pr[\,\mathsf{HG} \vee (\mathsf{Succ}(\boldsymbol{H}) \wedge \neg\mathsf{GH} \wedge \neg\mathsf{HG})\,] \;\le\; \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) + \frac{\text{poly}(k)}{2^k}\ . \tag{3}$$

Equation (2) follows.

THE ADVERSARIES. The design of $\boldsymbol{C}$ relies mostly on standard techniques, and so we leave it to Appendix A. We turn to $\boldsymbol{S}$. The latter gets input 1^k and oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b)), \mathsf{SD}(K,\cdot)$, begins with the initializations

$$((q,g,X),(q,g,x)) \xleftarrow{\$} \mathsf{AK}(1^k)\ ;\ y \xleftarrow{\$} \mathbb{Z}_q\ ;\ Y \leftarrow g^y\ ;\ W \xleftarrow{\$} \{0,1\}^k\ ;\ C_a \leftarrow (Y,W)\ , \tag{4}$$

and then runs $\boldsymbol{H}$ on inputs $(q,g,X), C_a$, itself responding to the oracle queries of the latter. Its aim is to do this in such a way that the key K underlying $\boldsymbol{S}$'s oracles plays the role of the quantity of the same name for $\boldsymbol{H}$. Eventually, it will output what $\boldsymbol{H}$ outputs. The difficulty faced by this adversary is that $\boldsymbol{H}$ might query K to H. (Other oracle queries are dealt with in standard ways.) In that case, $\boldsymbol{H}$ expects to be returned y. (And it cannot be fooled since, knowing $Y = g^y$, it can verify whether or not the value returned is y.) The difficulty for $\boldsymbol{S}$ is not that it does not know the right answer (via Equation (4), it actually knows y), but rather that it is not clear how it would

know that a query being made to H equals the key K underlying its oracles, so that it would know *when* to return y as the answer to a query to H.

In order to "detect" when query K is made, we would, ideally, like a test that can be performed on a value L, accepting if $L = K$ and rejecting otherwise. However, it is not hard to see that, in general, such a test does not exist.[4] Instead, we introduce a test that has a weaker property and show that it suffices for us.

Our test $\mathsf{KeyTest}$ takes input L and has access to $\boldsymbol{S}$'s $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ oracle. It returns a pair $(\mathrm{dec}, \mathrm{gs})$ such that: (1) If $L = K$ then $(\mathrm{dec}, \mathrm{gs}) = (1, b)$, meaning in this case it correctly computes the challenge bit b, and (2) If $L \neq K$ then, with overwhelming probability, either $\mathrm{dec} = 0$ (the test is saying $L \neq K$) or $(\mathrm{dec}, \mathrm{gs}) = (1, b)$ (the test is saying it does not know whether or not $L = K$, but it has successfully calculated the challenge bit anyway). With $\mathsf{KeyTest}$ in hand, $\boldsymbol{S}$ can answer a query L made to H as follows. It runs $(\mathrm{dec}, \mathrm{gs}) \xleftarrow{\$} \mathsf{KeyTest}(L)$. If $\mathrm{dec} = 0$, it can safely assume $L \neq K$ and return a random answer, while if $\mathrm{dec} = 1$, it can output gs as its guess to challenge bit b and halt.

A precise description and analysis of $\mathsf{KeyTest}$ are in Appendix A, but we briefly sketch the ideas here. The algorithm has two phases. In the first phase, it repeatedly tests whether or not

$$\mathsf{SD}(L, \mathsf{SE}(K, \mathrm{LR}(T_0, T_0, b))) = T_0 \quad \text{and} \quad \mathsf{SD}(L, \mathsf{SE}(K, \mathrm{LR}(T_1, T_1, b))) = T_1 \ ,$$

where T_0, T_1 are some distinct "test" messages. If any of these checks fails, it knows that $L \neq K$ and returns $(0, 0)$. (However, the checks can succeed with high probability even if $L \neq K$.) In the next phase, it repeatedly computes $\mathsf{SD}(L, \mathsf{SE}(K, \mathrm{LR}(T_0, T_1, b)))$ and, if *all* these computations yield T_{gs} for some bit gs, it returns $(1, \mathrm{gs})$. The analysis shows that, conditional on the first phase not returning $(0, 0)$, the bit gs from the second stage equals b with overwhelming probability.

A subtle point arises with relation to the test. Recall that $\boldsymbol{H}$ is making queries to $\mathsf{SD}(K, \cdot)$. $\boldsymbol{S}$ will answer these via its own oracle of the same name. Now, consider the event that $\boldsymbol{H}$ queries to $\mathsf{SD}(K, \cdot)$ a ciphertext C generated in some execution of $\mathsf{KeyTest}$. If $\boldsymbol{S}$ calls $\mathsf{SD}(K, C)$ to obtain the answer, it would immediately become an illegitimate adversary and thus forgo its advantage, since C is a result of a call to $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ made by $\boldsymbol{S}$ via subroutine $\mathsf{KeyTest}$. There are a few ways around this, and the one we use is to choose the initial "test" messages randomly so that $\boldsymbol{H}$ has low probability of being able to query a ciphertext C generated in some execution of $\mathsf{KeyTest}$.

This is all put together in Appendix A to show that Equation (3) holds.

We note that one might consider an alternative solution to $\boldsymbol{S}$'s problem of wanting to "detect" query K to H. Namely, reply to queries to H at random, then, after $\boldsymbol{H}$ terminates, pick one such query L at random, decrypt a challenge ciphertext via L, and use that to predict the challenge bit. Unfortunately, even though $L = K$ with probability $1/\mathrm{poly}(k)$, the advantage over one-half obtained by $\boldsymbol{S}$ via the strategy just outlined could be negligible because the wrong answers from the wrong random choices could overwhelm the right answer that arises when K is chosen.

4 Impossibility results

In this section we show (cf. Theorem 4.5) that the RO-model Hash El Gamal scheme admits no IND-CCA-preserving instantiation. However, rather than prove this directly, we obtain it as a consequence of stronger and more general results that are of interest in their own right. These are Theorem 4.3 and its corollary Theorem 4.4.

[4] Suppose, for example, that algorithms SE, SD only depend on the first half of the bits of their k-bit key. This is consistent with their being IND-CCA secure (in the sense that, if there exists an IND-CCA-secure symmetric encryption scheme, there also exists one with this property), but now, any test has probability at most $2^{-k/2}$ of being able to differentiate between K and a key $L \neq K$ that agrees with K in its first half.

$\overline{\mathsf{AK}}(1^k)$	$\overline{\mathsf{AE}}((q,g,X,gk,hk),K)$	$\overline{\mathsf{AD}}((q,g,x,gk,hk),(Y,W))$
$((q,g,X),(q,g,x)) \xleftarrow{\$} \mathsf{AK}(1^k)$	$k \leftarrow \lvert\langle 2q+1\rangle\rvert$	$k \leftarrow \lvert\langle 2q+1\rangle\rvert$
$gk \xleftarrow{\$} \{0,1\}^{\mathrm{GKL}(k)}$	$y \leftarrow \overline{H}_{k,(q,g)}(hk,K)$	$T \leftarrow \overline{G}_{k,(q,g)}(gk,Y^x)$
$hk \xleftarrow{\$} \{0,1\}^{\mathrm{HKL}(k)}$	$Y \leftarrow g^y$	$K \leftarrow T \oplus W$
Return $((q,g,X,gk,hk),$	$T \leftarrow \overline{G}_{k,(q,g)}(gk,X^y)$	If $g^{\overline{H}_{k,(q,g)}(hk,K)} = Y$
$\quad (q,g,x,gk,hk))$	$W \leftarrow T \oplus K$	$\quad$ then Return K
	Return (Y,W)	else Return $\perp$ EndIf

Figure 2: Algorithms of the standard-model asymmetric encryption scheme $\overline{\mathsf{HEG}}[\mathsf{CG}] = (\overline{\mathsf{AK}}, \overline{\mathsf{AE}}, \overline{\mathsf{AD}})$ obtained by instantiating RO-model asymmetric encryption scheme $\mathsf{HEG}[\mathsf{CG}]$ via poly-time function families $\overline{G}, \overline{H}$.

Theorem 4.3 is an impossibility result in the standard model. It says that, in this model, there simply do not exist asymmetric encryption schemes that are all of the following: IND-CCA preserving, key verifiable and ciphertext verifiable. (The last two properties are defined below.) Via Proposition 4.2 this yields Theorem 4.4, a general result about the possibility of instantiating certain RO-model schemes. It says that no RO-model key-verifiable and ciphertext-verifiable asymmetric encryption scheme admits an IND-CCA-preserving instantiation. Theorem 4.5 follows once we observe (cf. Proposition 4.1) that the Hash El Gamal scheme has the verifiability properties in question.

Although Theorem 4.5 is of most direct interest due to the results in Section 3 above, we believe that the more general results are deeper indications of the source of the gaps between RO-model security and standard-model security.

Below we begin by detailing what we mean by instantiation of a RO-model asymmetric encryption scheme because our notion is slightly more general than standard ones: in order to cover RO-model schemes (like Hash El Gamal) in which the domains and ranges of the oracles depend on the public key, we allow the families of functions used to instantiate the oracles to have domains and ranges depending on an auxiliary parameter a that, upon instantiation, can be chosen to depend on the public key. Then we will define the above-mentioned verifiability properties and move to the results.

Instantiating RO-model asymmetric encryption schemes. A *poly-time family of functions* $\overline{F}$ associates to security parameter k and *auxiliary parameter* $a \in \{0,1\}^*$ a map

$$\overline{F}_{k,a}\colon\ \{0,1\}^{\mathrm{FKL}(k)} \times \mathit{Dom}(a) \to \mathit{Rng}(a)$$

where $\mathit{Dom}(a)$ and $\mathit{Rng}(a)$ are sets depending on a. The *key-length* FKL of the scheme is a polynomial in k. We require that there exist a polynomial t such that $\overline{F}_{k,a}(fk,x)$ is computable in $t(k+|x|)$ time for all $k \in \mathbb{N}$, $a \in \{0,1\}^*$, $fk \in \{0,1\}^{\mathrm{FKL}(k)}$ and $x \in \mathit{Dom}(a)$. An *instantiation* of a RO-model asymmetric encryption scheme $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ is a standard-model asymmetric encryption scheme $\overline{\mathsf{AS}} = (\overline{\mathsf{AK}}, \overline{\mathsf{AE}}, \overline{\mathsf{AD}})$ obtained by replacing each random oracle used by $\mathsf{AE}(pk,\cdot)$ or $\mathsf{AD}(sk,\cdot)$ with an instance $\overline{F}_{k,a}(fk,\cdot)$ of an appropriate poly-time family of functions $\overline{F}$ (here a might depend on pk), having first enhanced the public and secret keys to include the key fk specifying this instance.

To illustrate, consider the RO-model Hash El Gamal scheme $\mathsf{HEG}[\mathsf{CG}] = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ associated to cyclic-group generator CG. Let $\overline{G}, \overline{H}$ be poly-time families such that for for each $k \in \mathbb{N}$ and

each $(q, g) \in [\mathsf{CG}(1^k)]$:

$$\overline{G}_{k,(q,g)}\colon \{0,1\}^{\mathrm{GKL}(k)} \times \langle g \rangle \to \{0,1\}^k \quad \text{and} \quad \overline{H}_{k,(q,g)}\colon \{0,1\}^{\mathrm{HKL}(k)} \times \{0,1\}^k \to \mathbb{Z}_q\ .$$

Here the auxiliary parameter is $a = (q, g)$. Then an instantiation of $\mathsf{HEG}[\mathsf{CG}]$ via $\overline{G}, \overline{H}$ is the standard-model asymmetric encryption scheme $\overline{\mathsf{HEG}}[\mathsf{CG}] = (\overline{\mathsf{AK}}, \overline{\mathsf{AE}}, \overline{\mathsf{AD}})$ whose constituent algorithms are depicted in Figure 2.

THE TWO PROPERTIES. We now define the above-mentioned key-verifiability and ciphertext-verifiability properties of asymmetric encryption schemes.

Let $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ be an asymmetric encryption scheme. We say that pk is (AS, k)-valid if there exists sk such that $(pk, sk) \in [\mathsf{AK}(1^k)]$. We say that AS is *key verifiable* if there exists a polynomial-time, possibly randomized algorithm VfPK (called the *key verifier*) and a negligible function ν (called the *error probability* of VfPK) such that $\mathsf{VfPK}(1^k, pk)$ returns 1 with probability at least $1 - \nu(k)$ if pk is (AS, k)-valid, and returns 1 with probability at most $\nu(k)$ otherwise.

Let $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ be an asymmetric encryption scheme. We say that $\mathsf{AS} = (\mathsf{AK}, \mathsf{AE}, \mathsf{AD})$ is *ciphertext verifiable* if there exists a polynomial-time, possibly randomized algorithm VfCtxt (called the *ciphertext verifier*) and a negligible function ν (called the *error probability* of VfCtxt) such that, if VfCtxt is run on inputs $1^k, pk, K, C$, where pk is (AS, k)-valid and $K \in \{0,1\}^k$, then VfCtxt returns 1 with probability at least $1 - \nu(k)$ if $C \in [\mathsf{AE}(pk, K)]$, and returns 1 with probability at most $\nu(k)$ otherwise. If AE or AD access any random oracles, then VfCtxt is given access to the same random oracles.

RESULTS. We show that the Hash El Gamal scheme possesses the two properties defined above, and that if a RO-model scheme possesses these properties then any instantiation inherits them. Then we state the main result of this section and several corollaries.

Proposition 4.1 The RO-model Hash El Gamal scheme $\mathsf{HEG}[\mathsf{CG}]$ associated to a cyclic-group generator CG is both key verifiable and ciphertext verifiable.

Proof of Proposition 4.1: We note that (q, g, X) is $(\mathsf{HEG}[\mathsf{CG}], k)$-valid if and only if $q, 2q+1$ are primes, g is a generator of the order q cyclic subgroup $\langle g \rangle$ of $\mathbb{Z}^*_{2q+1}$, $|\langle 2q+1 \rangle| = k$, and $X \in \langle g \rangle$. The key verifier VfPK, given inputs $1^k, (q, g, X)$, can thus verify that (q, g, X) is $(\mathsf{HEG}[\mathsf{CG}], k)$-valid based on standard facts from computational number theory. We omit the details.

Ciphertext verifiability is a consequence of the fact that the encryption algorithm $\mathsf{AE}^{G,H}(\cdot,\cdot)$ of $\mathsf{HEG}[\mathsf{CG}]$ (cf. Figure 1) is deterministic. Ciphertext verifier VfCtxt, given oracles G, H and inputs $1^k, (q, g, X), K, C$, where (q, g, X) is $(\mathsf{HEG}[\mathsf{CG}], k)$-valid and $K \in \{0,1\}^k$, simply runs $\mathsf{AE}^{G,H}((q, g, X), K)$ and checks whether or not the result is C. ∎

Proposition 4.2 Suppose AS is a RO-model asymmetric encryption scheme that is both key verifiable and ciphertext verifiable. Let $\overline{\mathsf{AS}}$ be any instantiation of AS via poly-time families of functions. Then $\overline{\mathsf{AS}}$ is also both key verifiable and ciphertext verifiable.

Proof of Proposition 4.2: For simplicity, we assume that only one random oracle is queried by the encryption and decryption algorithms of AS. The argument can be easily extended to the case of multiple random oracles. Let VfPK and VfCtxt be a key verifier and a ciphertext verifier for AS, respectively. Let $\overline{F}$ be the poly-time family of functions used in $\overline{\mathsf{AS}}$ to replace the random oracle. Recall that a public key of $\overline{\mathsf{AS}}$ contains a public key pk of AS and also a key fk specifying an instance of $\overline{F}$. We define algorithms $\overline{\mathsf{VfPK}}$ and $\overline{\mathsf{VfCtxt}}$.

On inputs $1^k, s$, $\overline{\mathsf{VfPK}}$ attempts to parse s as a pair (pk, fk). If it fails, it returns 0. Otherwise, it runs $\mathsf{VfPK}(1^k, pk)$. If the result is 0, it returns 0. Otherwise, it verifies that $fk \in \{0,1\}^{\mathrm{FKL}(k)}$. If so,

it returns 1, if not it returns 0. Clearly, $\overline{\mathsf{VfPK}}$ is a key verifier for $\overline{\mathsf{AS}}$.

$\overline{\mathsf{VfCtxt}}$ is identical to VfCtxt except that the random oracle is replaced with the same instance of $\overline{F}$ used in $\overline{\mathsf{AS}}$ to replace the oracle. ▮

The main result of this section is the following:

Theorem 4.3 Let $\overline{\mathsf{AS}}$ be a standard-model asymmetric encryption scheme that is both key verifiable and ciphertext verifiable. Then $\overline{\mathsf{AS}}$ is *not* IND-CCA preserving.

The proof is postponed in favor of deriving corollaries of interest:

Theorem 4.4 Let AS be a RO-model asymmetric encryption scheme that is both key verifiable and ciphertext verifiable. Let $\overline{\mathsf{AS}}$ be *any* instantiation of AS via poly-time families of functions. Then $\overline{\mathsf{AS}}$ is *not* IND-CCA preserving.

Proof of Theorem 4.4: $\overline{\mathsf{AS}}$ is a standard-model asymmetric encryption scheme. Proposition 4.2 implies that it inherits the key verifiability and ciphertext verifiability of AS. Theorem 4.3 then implies that it is not IND-CCA preserving. ▮

Theorem 4.5 Let $\mathsf{HEG}[\mathsf{CG}]$ be the RO-model Hash El Gamal scheme associated to a cyclic-group generator CG. Let $\overline{\mathsf{HEG}}[\mathsf{CG}]$ be *any* instantiation of $\mathsf{HEG}[\mathsf{CG}]$ via poly-time families of functions. Then $\overline{\mathsf{HEG}}[\mathsf{CG}]$ is *not* IND-CCA preserving.

Proof of Theorem 4.5: Proposition 4.1 says that $\mathsf{HEG}[\mathsf{CG}]$ is key verifiable and ciphertext verifiable. The result follows from Theorem 4.4. ▮

It remains only to prove Theorem 4.3.

Proof of Theorem 4.3. We will construct an IND-CCA-secure symmetric encryption scheme SS such that the mm-hybrid encryption scheme $(\overline{\mathsf{AS}}, \mathsf{SS})$ is not IND-CCA secure. This proves the theorem.

Let VfPK and VfCtxt be a key verifier and a ciphertext verifier for $\overline{\mathsf{AS}}$, respectively. Let $\mathsf{SS}' = (\mathsf{SK}', \mathsf{SE}', \mathsf{SD}')$ be any IND-CCA-secure symmetric encryption scheme. (Recall an implicit assumption is that some such scheme exists, since otherwise *all* asymmetric encryptions schemes are by default IND-CCA preserving and the entire problem we are considering is moot.) The construction of SS is in terms of SS' and algorithms VfPK and VfCtxt. We use the notation $\langle(\cdot,\cdot)\rangle$ to denote an injective, polynomial-time computable encoding of pairs of strings as strings such that given $\langle(M_1, M_2)\rangle$, M_1 and M_2 can be recovered in polynomial time). The algorithms constituting $\mathsf{SS} = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ are depicted in Figure 3. To conclude the proof we need only establish the following:

Claim 4.6 Symmetric encryption scheme SS is IND-CCA secure.

Claim 4.7 Multi-message hybrid encryption scheme $(\overline{\mathsf{AS}}, \mathsf{SS})$ is not IND-CCA secure.

Proof of Claim 4.6: Let us first provide some intuition. Note that on input M, encryption algorithm $\mathsf{SE}(K_1'||K_2, \cdot)$ uses the encryption algorithm SE' of an IND-CCA-secure scheme to compute $C' \xleftarrow{\$} \mathsf{SE}'(K_1', M)$ and outputs $C'||0$ or $C'||1$ depending on whether M has some "special" form or not. The ciphertext ends with 0 if M parses as a pair (M_1, M_2) such that algorithms

$\mathsf{SK}(1^k)$	$\mathsf{SE}(K, M)$	$\mathsf{SD}(K, C)$

$K' \xleftarrow{\$} \mathsf{SK}'(1^{\lceil k/2 \rceil})$
$K_2 \xleftarrow{\$} \{0,1\}^{\lfloor k/2 \rfloor}$
Return $K'\|K_2$
	$k \leftarrow \|K\|$
	Let K' be the first $\lceil k/2 \rceil$ bits of K, and let K_2 be the rest
	$C' \leftarrow \mathsf{SE}'(K', M)$
	Parse M as $\langle (M_1, M_2) \rangle$
	If the parsing fails then Return $C'\|1$ EndIf
	$p \xleftarrow{\$} \mathsf{VfPK}(k, M_1)$
	$c \xleftarrow{\$} \mathsf{VfCtxt}(k, M_1, K, M_2)$
	If ($p = 1$ and $c = 1$) then Return $C'\|0$
	else Return $C'\|1$ EndIf
		$k \leftarrow \|K\|$
		Let K' be the first $\lceil k/2 \rceil$ bits of K, and let K_2 be the rest
		Parse C as $C'\|d$, where $d \in \{0,1\}$
		$M' \leftarrow \mathsf{SD}'(K', C')$
		Parse M' as $\langle (M_1, M_2) \rangle$
		If the parsing fails then If $d = 1$ then Return M' else Return $\perp$ EndIf
		$p \xleftarrow{\$} \mathsf{VfPK}(k, M_1)$
		$c \xleftarrow{\$} \mathsf{VfCtxt}(k, M_1, K, M_2)$
		If ($d = 0$ and $p = 1$ and $c = 1$) then Return M' EndIf
		If ($d = 1$ and ($p \neq 1$ or $c \neq 1$)) then Return M'
		Return $\perp$ EndIf

Figure 3: Algorithms of the symmetric encryption scheme $\mathsf{SS} = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ for the proof of Theorem 4.3. Above, $\langle (M_1, M_2) \rangle$ denotes an encoding of the pair of strings (M_1, M_2) as a string.

$\mathsf{VfPK}, \mathsf{VfCtxt}$ indicate that M_1 is $(\overline{\mathsf{AS}}, k)$-valid and $M_2 \in [\overline{\mathsf{AE}}(M_1, K_1'\|K_2)]$. The decryption algorithm $\mathsf{SD}(K_1'\|K_2, \cdot)$ on input $C'\|d$, where d is a bit, computes $M' \leftarrow \mathsf{SD}'(K_1', C')$ and returns M' only if either M' is of the special form and $d = 0$, or M' is not of this form and $d = 1$. Therefore, an obvious strategy for an adversary against SS is to query its oracle $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$ on a pair of messages such that one of them is of this special form and the other is not. Using the unique decryptability of $\overline{\mathsf{AE}}$ and the fact that K_2 is chosen at random, independently from the adversary's view, we show that it cannot find such queries except with negligible probability. Moreover, we show that any strategy for the adversary can be employed by an attacker against scheme SS' to win its game. Details follow.

Let $\boldsymbol{S}$ be a legitimate polynomial-time adversary attacking SS. We will construct a legitimate polynomial-time adversary $\boldsymbol{S}'$ such that

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) \;\leq\; \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS}',\boldsymbol{S}'}(\lceil k/2 \rceil) + O(Q(k)) \cdot \nu(k) + \frac{O(Q(k))}{2^{\lfloor k/2 \rfloor}}\,, \tag{5}$$

where Q is a polynomial upper bounding the total number of queries made by $\boldsymbol{S}$ to its different oracles, and ν is a negligible function related to the error probabilities of algorithms VfPK and VfCtxt. Since SS' is assumed IND-CCA secure, the advantage function associated to $\boldsymbol{S}'$ above is negligible, and thus so is the advantage function associated to $\boldsymbol{S}$. To complete the proof we need to specify adversary $\boldsymbol{S}'$ and prove Equation (5).

Adversary $\boldsymbol{S}'$ is given input $1^{\lceil k/2 \rceil}$ and has access to oracles $\mathsf{SE}'(K_1', \mathrm{LR}(\cdot, \cdot, b))$ and $\mathsf{SD}'(K_1', \cdot)$. Its goal is to guess the bit b. It runs $\boldsymbol{S}$ on input 1^k. In this process, $\boldsymbol{S}$ will query its two oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$ and $\mathsf{SD}(K, \cdot)$. To answer a query to the first of these oracles, $\boldsymbol{S}'$ forwards the query to its oracle $\mathsf{SE}'(K_1', \mathrm{LR}(\cdot, \cdot, b))$, appends 1 to the oracle's reply and returns the result to $\boldsymbol{S}$. To answer a query to the second oracle, $\boldsymbol{S}'$ checks the last bit of the query. If it is 0, $\boldsymbol{S}'$ returns $\perp$ to $\boldsymbol{S}$. Otherwise, it removes the last bit, forwards the result to its oracle $\mathsf{SD}'(K_1', \cdot)$, and returns the answer to $\boldsymbol{S}$. When $\boldsymbol{S}$ outputs its guess b', $\boldsymbol{S}'$ returns b'.

We now analyze $\boldsymbol{S}'$. Consider the experiment in which $\boldsymbol{S}'$ attacks SS'. We define the following events.

$\mathsf{Succ}(\boldsymbol{S}')$	:	$\boldsymbol{S}'$ is successful, meaning its output equals the challenge bit b.
BadE	:	$\boldsymbol{S}$ makes a query to oracle $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ in which one of the messages can be parsed as $\langle (M_1, M_2) \rangle$ such that M_1 is $(\overline{\mathsf{AS}}, k)$-valid and $M_2 \in [\overline{\mathsf{AE}}(M_1, K)]$
BadD	:	$\boldsymbol{S}$ makes a query to oracle $\mathsf{SD}(K, \cdot)$ that can be parsed as $C'\|\|d$, where d is a bit, such that $\mathsf{SD}'(K'_1, C') = \langle (M_1, M_2) \rangle$, where M_1 is $(\overline{\mathsf{AS}}, k)$-valid and $M_2 \in [\overline{\mathsf{AE}}(M_1, K)]$

For the experiment in which $\boldsymbol{S}$ attacks SS, we define the following events.

$\mathsf{Succ}(\boldsymbol{S})$	:	$\boldsymbol{S}$ is successful, meaning its output equals the challenge bit b.
Crct	:	Every time algorithms VfPK and VfCtxt are invoked, they return the correct value

We claim that if events BadE and BadD do not occur, then $\boldsymbol{S}'$ simulates perfectly the environment provided to $\boldsymbol{S}$ in its attack against SS when algorithms VfPK and VfCtxt never err. First, note that answers to queries to oracle $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ can only be off by the last bit. In the absence of the "bad" events, each ciphertext returned to $\boldsymbol{S}$ as a reply to a query to oracle $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ has 1 as the last bit. This is also the case in $\boldsymbol{S}$'s real attack when algorithms VfPK and VfCtxt are always correct. If $\boldsymbol{S}$ queries $\mathsf{SD}(K, \cdot)$ with a ciphertext $C'||0$, assuming events BadE and BadD do not occur, $\boldsymbol{S}'$ gives $\boldsymbol{S}$ the response it would get in the real attack when algorithms VfPK and VfCtxt are always correct, namely $\perp$. Since $\boldsymbol{S}$ is legitimate, if it queries oracle $\mathsf{SD}(K, \cdot)$ with a ciphertext $C'||1$, then C' must not have previously been returned by oracle $\mathsf{SE}'(K'_1, \mathrm{LR}(\cdot,\cdot,b))$. Thus $\boldsymbol{S}'$ can legitimately make query C' to its oracle $\mathsf{SD}'(K'_1, \cdot)$. If M is the response, then, assuming that events BadE and BadD do not occur, the answer $\boldsymbol{S}$ expects when algorithms VfPK and VfCtxt are always correct is exactly M. Therefore,

$$\begin{aligned}
\Pr\left[\, \mathsf{Succ}(\boldsymbol{S}') \,\right] &\geq \Pr\left[\, \mathsf{Succ}(\boldsymbol{S}') \mid \neg\mathsf{BadE} \wedge \neg\mathsf{BadD} \,\right] - \Pr\left[\, \mathsf{BadE} \vee \mathsf{BadD} \,\right] \\
&\geq \Pr\left[\, \mathsf{Succ}(\boldsymbol{S}) \mid \mathsf{Crct} \,\right] - \Pr\left[\, \mathsf{BadE} \vee \mathsf{BadD} \,\right] \\
&\geq \Pr\left[\, \mathsf{Succ}(\boldsymbol{S}) \,\right] - \Pr\left[\, \neg\mathsf{Crct} \,\right] - \Pr\left[\, \mathsf{BadE} \vee \mathsf{BadD} \,\right].
\end{aligned}$$

We now provide upper bounds for the probabilities of events $\neg\mathsf{Crct}$ and $\mathsf{BadE} \vee \mathsf{BadD}$. Let $q_e(k)$ and $q_d(k)$ be the number of queries $\boldsymbol{S}$ makes to oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ and $\mathsf{SD}(K, \cdot)$, respectively, on input 1^k. Let ν_1 be the error probability of key verifier VfPK, and ν_2 the error probability of ciphertext verifier VfCtxt. Then

$$\Pr\left[\, \overline{\mathsf{Crct}} \,\right] \leq q_e(k) \cdot (\nu_1(k) + \nu_2(k)) + q_d(k) \cdot (\nu_1(k) + \nu_2(k)) = Q(k) \cdot \nu(k)\,,$$

where $Q(k) = q_e(k) + q_d(k)$ and $\nu(k) = \nu_1(k) + \nu_2(k)$.

We observe that if M_1 is $(\overline{\mathsf{AS}}, k)$-valid, then for any $M_2 \in \{0,1\}^*$, there exists a unique $K' \in [\mathsf{SK}(1^k)]$ such that $M_2 \in [\overline{\mathsf{AE}}(M_1, K')]$. Recall that the key for oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ and $\mathsf{SD}(K, \cdot)$ is $K = K'_1||K_2$, where K_2 is chosen uniformly at random from $\{0,1\}^{\lfloor k/2 \rfloor}$ and is independent from $\boldsymbol{S}$'s view. Therefore, for any query made by $\boldsymbol{S}$ to oracle $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$, the probability that one of the messages in the query parses as $\langle (M_1, M_2) \rangle$ such that M_1 is $(\overline{\mathsf{AS}}, k)$-valid and $M_2 \in [\overline{\mathsf{AE}}(M_1, K)]$ is at most $2/2^{\lfloor k/2 \rfloor}$. Similarly, for any query $C'||d$, where d is a bit, made by $\boldsymbol{S}$ to oracle $\mathsf{SD}(K, \cdot)$, the probability that $\mathsf{SD}'(K'_1, C') = M'$, where M' parses as $\langle (M_1, M_2) \rangle$, M_1 is $(\overline{\mathsf{AS}}, k)$-valid and $M_2 \in [\overline{\mathsf{AE}}(M_1, K)]$ is at most $1/2^{\lfloor k/2 \rfloor}$. Therefore,

$$\Pr\left[\, \mathsf{BadE} \vee \mathsf{BadD} \,\right] \;\leq\; \frac{2q_e(k) + q_d(k)}{2^{\lfloor k/2 \rfloor}} \;\leq\; \frac{2 \cdot Q(k)}{2^{\lfloor k/2 \rfloor}}\,.$$

Hence

$$\begin{aligned}\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS}',\boldsymbol{S}'}(\lceil k/2\rceil) &= 2\cdot\Pr\left[\,\mathsf{Succ}(\boldsymbol{S}')\,\right]-1 \geq 2\cdot\left(\Pr\left[\,\mathsf{Succ}(\boldsymbol{S})\,\right]-Q(k)\cdot\nu(k)-\frac{O(Q(k))}{2^{\lfloor k/2\rfloor}}\right)-1\\ &= \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)-O(Q(k))\cdot\nu(k)-\frac{O(Q(k))}{2^{\lfloor k/2\rfloor}}\,.\end{aligned}$$

Rearranging terms gives Equation (5). ▮

Proof of Claim 4.7: We define a hybrid adversary $\boldsymbol{H}$ attacking $(\overline{\mathsf{AS}},\mathsf{SS})$. $\boldsymbol{H}$ is given inputs pk, C_a and has access to oracles $\mathsf{SE}(K,\mathrm{LR}(\cdot,\cdot,b))$, $\mathsf{SD}(K,\cdot)$, and $\mathsf{AD}(sk,\cdot)$. Its goal is to guess the challenge bit b. By the definition of experiment $\mathbf{Exp}^{\text{ind-cca}}_{\overline{\mathsf{AS}},\mathsf{SS},\boldsymbol{H}}(k)$, pk is $(\overline{\mathsf{AS}},k)$-valid and $C_a \in [\overline{\mathsf{AE}}(pk,K)]$. Therefore, $\langle(pk,C_a)\rangle$ is a message which, when encrypted with $\mathsf{SE}(K,\cdot)$, yields a ciphertext that with overwhelming probability has last bit 0. (The last bit will be 0, if algorithms VfPK and VfCtxt output the correct value.) We observe that for any string C chosen at random from $\{0,1\}^{|C_a|}\setminus\{C_a\}$, the probability that $K=\mathsf{AD}(sk,C)$ is at most 2^{-k}, i.e., the probability that $C\in[\overline{\mathsf{AE}}(pk,K)]$ is at most 2^{-k}. Hence $\langle(pk,C)\rangle$ is a message which, when encrypted with $\mathsf{SE}(K,\cdot)$, yields a ciphertext that with overwhelming probability has last bit 1. (If $C\notin[\overline{\mathsf{AE}}(pk,K)]$ and algorithms VfPK and VfCtxt output the correct value, then the last bit will be 1.) Thus, adversary $\boldsymbol{H}$ can construct two messages for which it can guess with high probability the last bit of the corresponding ciphertext. Using this information it can then guess the challenge bit. Details follow.

Adversary $\boldsymbol{H}$ chooses C at random from $\{0,1\}^{|C_a|}\setminus\{C_a\}$, makes a query $\langle(pk,C_a)\rangle,\langle(pk,C)\rangle$ to oracle $\mathsf{SE}(K,\mathrm{LR}(\cdot,\cdot,b))$, parses the response as $C'||d$, where d is a bit, and returns d. The running time of $\boldsymbol{H}$ is clearly polynomial in k. We claim that $\mathsf{Adv}^{\text{ind-cca}}_{\overline{\mathsf{AS}},\mathsf{SS},\boldsymbol{H}}(k)\ \geq\ 1-2^{-k}-\nu(k)$, where ν is a negligible function related to the error probabilities of algorithms VfPK and VfCtxt. To prove this, we consider the following events.

$\mathsf{Succ}(\boldsymbol{H})$: $\boldsymbol{H}$ is successful, meaning its output equals the challenge bit b.
Crct : Every time algorithms VfPK and VfCtxt are invoked, they return the correct value

Assume that event Crct occurs. If challenge bit b is 0, then the response to $\boldsymbol{H}$'s query is a ciphertext that has last bit 0. If bit b is 1, then with probability at least $1-2^{-k}$, the response is a ciphertext that has last bit 1. Thus

$$\Pr\left[\,\mathsf{Succ}(\boldsymbol{H})\,\right]\ \geq\ \Pr\left[\,\mathsf{Succ}(\boldsymbol{H})\ \mid\ \mathsf{Crct}\,\right]-\Pr\left[\,\neg\mathsf{Crct}\,\right]\ \geq\ \frac{1}{2}\cdot\left(1-\frac{1}{2^k}\right)+\frac{1}{2}-\Pr\left[\,\neg\mathsf{Crct}\,\right]$$

If ν_1 is the error probability of key verifier VfPK, and ν_2 is the error probability of ciphertext verifier VfCtxt, then $\Pr\left[\,\neg\mathsf{Crct}\,\right]\ \leq\ \nu_1(k)+\nu_2(k)$. Hence

$$\mathsf{Adv}^{\text{ind-cca}}_{\overline{\mathsf{AS}},\mathsf{SS},\boldsymbol{H}}(k)\ =\ 2\cdot\Pr\left[\,\mathsf{Succ}(\boldsymbol{H})\,\right]-1\ \geq\ 1-2^{-k}-2\cdot(\nu_1(k)+\nu_2(k))\ =\ 1-2^{-k}-\nu(k)\,,$$

where $\nu(k)=2\cdot(\nu_1(k)+\nu_2(k))$. ▮

Notice that the adversary constructed in the proof of Claim 4.7 does not make any queries to its oracles $\mathsf{SD}(K,\cdot)$ and $\mathsf{AD}(sk,\cdot)$. The proof thus shows that $\overline{\mathsf{AS}}$ is not even IND-CPA preserving.

An interesting question at this point may be why the proof of Theorem 4.3 fails for the RO-model Hash El Gamal scheme $\mathsf{HEG}[\mathsf{CG}]$ associated to a cyclic-group generator CG —it must, since otherwise Theorem 3.1 would be contradicted— but succeeds for any instantiation of this scheme. The answer is that symmetric encryption scheme SS, depicted in Figure 3 runs a ciphertext verifier VfCtxt for the asymmetric encryption scheme in question. In the case of the RO-model scheme

HEG[CG], any ciphertext verifier must query random oracles G and H. But SS does not have access to these oracles, and so cannot run such a ciphertext verifier. The adversary of course does have access to G, H, but has no way to "pass" these objects to the encryption algorithm of the symmetric encryption scheme. On the other hand, in the instantiated scheme, the keys describing the functions instantiating the random oracles may be passed by the adversary to the encryption algorithm of SS in the form of a message containing the public key, giving SS the ability to run the ciphertext verifier.

This might lead one to ask why SS does not have oracle access to G, H. The answer that we are not considering symmetric encryption schemes in the random-oracle model is not the right one, since we did that only to simplify notation, and our results should and do extend to the case where the symmetric encryption schemes too might be in the RO model. The right answer is that even if SS were a RO-model scheme, in a context such as $\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k)$ in which it is executed in conjunction with a RO-model asymmetric encryption scheme AS, its random oracles would be chosen independently of those used by AS. This means that the symmetric encryption scheme above would continue to be unable to run a ciphertext verifier that depended on oracles used by AS.

The correctness of the principle of independently choosing random oracles of different schemes in a common context may be clarified via an analogy. Think of running two schemes, such as an encryption scheme and a signature scheme, in a common context. This context should pick keys for the schemes independently, not use the same key for both schemes, since each scheme is designed and analyzed under the assumption that its key is used by it alone. Similarly, a RO-model scheme is designed and analyzed under the assumption that its random oracles are used by it alone, and thus in a context involving several such schemes, each would get its own random oracles. (The adversary of course gets *all* the random oracles.)

References

[1] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[2] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, *First ACM Conference on Computer and Communications Security*, ACM, 1993.

[3] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of Cryptology*, 13(1), 2000.

[4] R. Canetti, "Universally composable security," *Proceedings of the 42nd Symposium on Foundations of Computer Science*, IEEE, 2001.

[5] R. Canetti, U. Feige, O. Goldreich and M. Naor, "Adaptively secure multi-party computation," *Proceedings of the 28th Annual Symposium on the Theory of Computing*, ACM, 1996.

[6] R. Canetti, O. Goldreich, S. Halevi, "The random oracle methodology, revisited," *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, 1998.

[7] J.-S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, C. Tymen, "GEM: A Generic Chosen-Ciphertext Secure Encryption Method", *Topics in Cryptology – CT-RSA '02*, Lecture Notes in Computer Science Vol. 2271 , B. Preneel ed., Springer-Verlag, 2002.

[8] J.-S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, C. Tymen, "Optimal Chosen-Ciphertext Secure Encryption of Arbitrary-Length Messages," *Proceedings of the Fifth International workshop on practice and theory in Public Key Cryptography (PKC'02)*, Lecture Notes in Computer Science Vol. 1431, D. Naccache and P. Paillier eds., Springer-Verlag, 2002.

[9] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[10] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," IACR ePrint archive Record 2001/108, 2001, `http://eprint.iacr.org/`.

[11] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol 31, 1985.

[12] E. Fujisaki, T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[13] R. Gennaro, S. Halevi and T. Rabin, "Secure Hash-and-Sign Signatures without the Random Oracle," *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science Vol. 1592 , J. Stern ed., Springer-Verlag, 1999.

[14] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.

[15] S. Goldwasser and Y. Taumann, "On the (in)security of the Fiat-Shamir paradigm," IACR ePrint archive Record 2003/034, 2003, `http://eprint.iacr.org/`.

[16] S. Micali, "Computationally sound proofs," *SIAM Journal on Computing*, Vol. 30, No. 4, 2000, pp. 1253-1298. Preliminary version in *FOCS'94*.

[17] J. B. Nielsen "Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case," *Advances in Cryptology – CRYPTO '02*, Lecture Notes in Computer Science Vol. 2442 , M. Yung ed., Springer-Verlag, 2002.

[18] T. Okamoto and D. Pointcheval "REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform," *Topics in Cryptology – CT-RSA '01*, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed., Springer-Verlag, 2001.

[19] V. Shoup, "A proposal for an ISO standard for public key encryption", IACR ePrint archive Record 2001/112, 2001, `http://eprint.iacr.org/`.

A Proof of Theorem 3.1

We have explained the ideas behind this proof in Section 3. Here we provide the full adversary constructions and analyses.

Proof setup. Let $\boldsymbol{H}$ be a polynomial-time hybrid adversary attacking $(\mathsf{AS}, \mathsf{SS})$. We will construct polynomial-time adversaries $\boldsymbol{S}, \boldsymbol{C}$ such that

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k) \;\leq\; \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) + O(Q(k)) \cdot \mathsf{Adv}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) + \frac{O(Q(k)^2)}{2^k}\,. \tag{6}$$

where $Q(k)$ is a polynomial upper bounding the number of queries made by $\boldsymbol{H}$ to the G and H oracles. (This includes queries made directly by $\boldsymbol{H}$ and those made indirectly as a consquence of $\boldsymbol{H}$'s queries to its $\mathsf{AD}^{G,H}((q,g,x),\cdot)$ oracle.) Since SS is assumed IND-CCA secure and the CDH problem is hard for CG the advantage functions related to $\boldsymbol{S}, \boldsymbol{C}$ above are negligible, and thus so is the advantage function related to $\boldsymbol{H}$. To complete the proof we need to specify the adversaries $\boldsymbol{S}, \boldsymbol{C}$ and prove Equation (6).

Description of $\boldsymbol{S}$. Adversary $\boldsymbol{S}$ is given input 1^k and has access to oracles $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ and $\mathsf{SD}(K,\cdot)$. Its goal is to guess the bit b. It begins with the following initializations:

Subroutine GSim(Z)
  If GT[Z] is not defined then GT[Z] $\xleftarrow{\$}$ $\{0,1\}^k$ EndIf
  Return GT[Z]

Subroutine HSim(L)
  If HT[L] is defined then return it as the answer EndIf
  $(\mathrm{dec}, \mathrm{gs}) \xleftarrow{\$} \mathsf{KeyTest}(L)$; HT[L] $\xleftarrow{\$}$ $\mathbb{Z}_q$
  If dec $= 0$ then return HT[L] as the answer EndIf
  If dec $= 1$ then output gs (as a guess to the value of challenge bit b) and halt EndIf

---

Subroutine $\mathsf{KeyTest}(L)$
  dec $\leftarrow 1$
  For $i = 1, \ldots, k$ do
    $C_0^i[L] \xleftarrow{\$} \mathsf{SE}(K, \mathrm{LR}(T_0, T_0, b))$; If $\mathsf{SD}(L, C_0^i[L]) \neq T_0$ then dec $\leftarrow 0$ EndIf
    $C_1^i[L] \xleftarrow{\$} \mathsf{SE}(K, \mathrm{LR}(T_1, T_1, b))$; If $\mathsf{SD}(L, C_1^i[L]) \neq T_1$ then dec $\leftarrow 0$ EndIf
  EndFor
  If dec $= 0$ return $(0,0)$ EndIf
  For $i = 1, \ldots, k$ do $C^i[L] \xleftarrow{\$} \mathsf{SE}(K, \mathrm{LR}(T_0, T_1, b))$; $T^i \leftarrow \mathsf{SD}(L, C^i[L])$ EndFor
  If $T^1 = T^2 = \cdots = T^k = T_0$ then return $(1,0)$ EndIf
  If $T^1 = T^2 = \cdots = T^k = T_1$ then return $(1,1)$ EndIf
  Return $(0,0)$

Figure 4: Subroutines defined by $\boldsymbol{S}$ and used to simulate $\boldsymbol{H}$'s oracles.

$((q,g,X),(q,g,x)) \xleftarrow{\$} \mathsf{AK}(1^k)$; $y \xleftarrow{\$} \mathbb{Z}_q$; $Y \leftarrow g^y$; $W \xleftarrow{\$} \{0,1\}^k$; $C_a \leftarrow (Y, W)$;
$T_0 \xleftarrow{\$} \{0,1\}^k$; $T_1 \xleftarrow{\$} \{0,1\}^k - \{T_0\}$.

Then it runs $\boldsymbol{H}$ on inputs public key (q,g,X) and ciphertext C_a. In the process $\boldsymbol{H}$ will query its five oracles

$$G\ ,\ H\ ,\ \mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))\ ,\ \mathsf{SD}(K,\cdot)\ ,\ \mathsf{AD}^{G,H}((q,g,x),\cdot)\ . \tag{7}$$

$\boldsymbol{S}$ will answer these queries. To that end, it defines the subroutines shown in Figure 4. It answers a query Z to G by running GSim(Z) and returning the answer to $\boldsymbol{H}$. It answers a query L to H by running HSim(Z) and returning the answer to $\boldsymbol{H}$. It answers queries to the $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ oracle via its own oracle of the same name. It answers each query C to the $\mathsf{SD}(K,\cdot)$ oracle using its own decryption oracle, unless there exist i, j and L such that L was queried to H and either $C = C_j^i[L]$ or $C = C^i[L]$. In that case $\boldsymbol{S}$ aborts. Since $\boldsymbol{S}$ possesses the secret key (q,g,x) it can answer queries to $\mathsf{AD}^{G,H}((q,g,x),\cdot)$ by performing the computation of the decryption algorithm, replacing calls that the latter makes to G or H by calls to the relevant subroutines just mentioned. If $\boldsymbol{H}$ runs to completion ($\boldsymbol{S}$ can output its guess as to the value of b, and halt, before this) then $\boldsymbol{S}$ outputs whatever $\boldsymbol{H}$ outputs.

Description of $\boldsymbol{C}$. Adversary $\boldsymbol{C}$ is given inputs q, g, X, Y, where $X, Y \in \langle g \rangle$ have been chosen uniformly at random. Its goal is to compute g^{xy} where $g^x = X$ and $g^y = Y$. Let $k \leftarrow |\langle 2q+1 \rangle|$. $\boldsymbol{C}$ begins with the following initializations:

$$K \xleftarrow{\$} \mathsf{SK}(1^k)\ ;\ b \xleftarrow{\$} \{0,1\}\ ;\ W \xleftarrow{\$} \{0,1\}^k\ ;\ C_a \leftarrow (Y, W)\ .$$

Then it runs $\boldsymbol{H}$ on inputs public key (q,g,X) and ciphertext C_a. In the process $\boldsymbol{H}$ will query the five oracles listed in Equation (7). $\boldsymbol{C}$ will answer these queries. To that end, it defines the

Subroutine GSim(Z)
If GT[Z] is not defined then GT[Z] $\xleftarrow{\$}$ $\{0,1\}^k$ EndIf
Return GT[Z]

Subroutine HSim(L)
If HT[L] is not defined then HT[Z] $\xleftarrow{\$}$ $\mathbb{Z}_q$ EndIf
Return HT[L]

Subroutine ADSim(Y', W')
If there is no L such that $g^{\mathrm{HT}[L]} = Y'$ then return $\perp$ EndIf
Let L be such that $g^{\mathrm{HT}[L]} = Y'$
$Z' \leftarrow X^{\mathrm{HT}[L]}$; $T' \leftarrow \mathrm{GSim}(Z')$; $K' \leftarrow T' \oplus W'$; Return K'

Figure 5: Subroutines defined by $\boldsymbol{C}$ and used to simulate $\boldsymbol{H}$'s oracles.

---

subroutines shown in Figure 5. It answers a query Z to G by running GSim(Z) and returning the answer to $\boldsymbol{H}$. It answers a query L to H by running HSim(Z) and returning the answer to $\boldsymbol{H}$. Since it possesses K and b it can answer queries to the $\mathsf{SE}(K, \mathrm{LR}(\cdot,\cdot,b))$ or $\mathsf{SD}(K,\cdot)$ oracles by simply performing the relevant computation and returning the answer. It answers a query (Y', W') to $\mathsf{AD}^{G,H}((q,g,x),\cdot)$ by running ADSim(Y', W') and returning the answer. When $\boldsymbol{H}$ has terminated, $\boldsymbol{C}$ picks Z at random from the set $\{\, Z \,:\, \mathrm{GT}[Z] \text{ is defined} \,\}$ and outputs Z.

Analysis. For the analysis, define the following experiments:

$$\begin{array}{rcl} \mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) & : & K \xleftarrow{\$} \mathsf{SK}(1^k)\ ;\ b \xleftarrow{\$} \{0,1\}\ ;\ d \xleftarrow{\$} \boldsymbol{S}^{\mathsf{SE}(K,\mathrm{LR}(\cdot,\cdot,b)),\,\mathsf{SD}(K,\cdot)}(1^k)\ ; \\ & & \text{If } d = b \text{ then return 1 else return 0} \\ \mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) & : & (q,g) \xleftarrow{\$} \mathsf{CG}(1^k)\ ;\ x,y \xleftarrow{\$} \mathbb{Z}_q\ ;\ Z \leftarrow \boldsymbol{C}(q,g,g^x,g^y) \\ & & \text{If } Z = g^{xy} \text{ then return 1 else return 0} \end{array}$$

We let $\Pr_{\boldsymbol{S}}[\,\cdot\,]$ and $\Pr_{\boldsymbol{C}}[\,\cdot\,]$ denote the probabilities in the above experiments, respectively, and we let $\Pr_{\boldsymbol{H}}[\,\cdot\,]$ denote the probability in experiment $\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k)$.

Let $((q,g,X),(q,g,x)) \in [\mathsf{AK}(1^k)]$ and $K \in [\mathsf{SK}(1^k)]$. We define the following events relating to $\boldsymbol{H}$'s execution on inputs public key (q,g,X) and ciphertext $C_a = (Y,W)$ where $g^y = Y$. These events are defined in any of the three experiments we are considering:

- GH : There exists a time at which g^{xy} is queried to G but K has not been queried to H
- HG : There exists a time at which K has been queried to H but g^{xy} has not been queried to G
- $\mathsf{Succ}(\boldsymbol{H})$: $\boldsymbol{H}$ is successful, meaning its output equals the challenge bit b.

We clarify that the queries referred to above include both direct and indirect queries of $\boldsymbol{H}$, but, in the case of $\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k)$, they do *not* include the queries to G and H made by the computation $C_a \leftarrow \mathsf{AE}^{G,H}((q,g,X),\cdot)$ that initializes the experiment. (We are only considering queries to G, H resulting from the execution of $\boldsymbol{H}$.) The main claims related to the analysis are:

$$\Pr_{\boldsymbol{H}}[\,\mathsf{HG} \vee (\mathsf{Succ}(\boldsymbol{H}) \wedge \neg\mathsf{HG} \wedge \neg\mathsf{GH})\,] \;\le\; \Pr_{\boldsymbol{S}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) = 1\,\right] + \frac{O(Q(k))}{2^k} \tag{8}$$

$$\Pr_{\boldsymbol{H}}[\,\mathsf{GH}\,] \;\le\; Q(k) \cdot \Pr_{\boldsymbol{C}}\left[\,\mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) = 1\,\right] + \frac{O(Q(k)^2)}{2^k}\,. \tag{9}$$

Let us see how these enable us to conclude the proof, and then return to prove them. We have:

$$\begin{aligned}
&\frac{1}{2}\cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k)+\frac{1}{2}\\
&= \Pr_{\boldsymbol{H}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{AS},\mathsf{SS},\boldsymbol{H}}(k)=1\,\right]\\
&= \Pr_{\boldsymbol{H}}[\,\mathsf{Succ}(\boldsymbol{H})\,]\\
&= \Pr_{\boldsymbol{H}}[\,(\mathsf{Succ}(\boldsymbol{H})\wedge\mathsf{HG})\vee(\mathsf{Succ}(\boldsymbol{H})\wedge\neg\mathsf{HG}\wedge\neg\mathsf{GH})\,]+\Pr_{\boldsymbol{H}}[\,\mathsf{Succ}(\boldsymbol{H})\wedge\mathsf{GH}\,]\\
&\le \Pr_{\boldsymbol{H}}[\,\mathsf{HG}\vee(\mathsf{Succ}(\boldsymbol{H})\wedge\neg\mathsf{HG}\wedge\neg\mathsf{GH})\,]+\Pr_{\boldsymbol{H}}[\,\mathsf{GH}\,]\\
&\le \Pr_{\boldsymbol{S}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)=1\,\right]+\frac{O(Q(k))}{2^k}+Q(k)\cdot\Pr_{\boldsymbol{C}}\left[\,\mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k)=1\,\right]+\frac{O(Q(k)^2)}{2^k}\\
&= \frac{1}{2}\cdot\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)+\frac{1}{2}+Q(k)\cdot\mathsf{Adv}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k)+\frac{O(Q(k)^2)}{2^k}\ .
\end{aligned}$$

Re-arranging terms and simplifying we get Equation (6). To complete the proof, we must establish Equations (8) and (9).

Proof of Equation (8). An important ingredient in this proof is the following lemma that characterizes what Subroutine $\mathsf{KeyTest}$ accomplishes:

Lemma A.1 If $L=K$ then $\mathsf{KeyTest}(L)$ returns $(1,b)$, while if $L\neq K$ then

$$\Pr\left[\,(\text{dec},\text{gs})\xleftarrow{\$}\mathsf{KeyTest}(L)\ :\ (\text{dec},\text{gs})=(1,1-b)\,\right]\le 4^{-k}\ .\ \blacksquare$$

In other words, if $L\neq K$, then with high probability either the test indicates this by returning $\text{dec}=0$ or it successfully computes the value of the challenge bit b. Above, the probability is over the coin tosses made by the $\mathsf{SE}(K,\mathrm{LR}(\cdot,\cdot,b))$ oracle called in $\mathsf{KeyTest}$, with K,b fixed.

Proof of Lemma A.1: The fact that $\mathsf{KeyTest}(L)$ returns $(1,b)$ when $L=K$ is a consequence merely of the unique decrytability of SS, namely the fact that for all $K\in[\mathsf{SK}(1^k)]$ and all $M\in\{0,1\}^*$ we have $\mathsf{SD}(K,\mathsf{SE}(K,M))=M$ with probability one, the probability being over the coin tosses of SE.

Now assume $L\neq K$. Let $\Pr[\,\cdot\,]$ denote the probability taken over the coin tosses of $\mathsf{SE}(K,\cdot)$, with K fixed. Let

$$P_0\ =\ \Pr[\,\mathsf{SD}(L,\mathsf{SE}(K,T_0))=T_0\,]\quad\text{and}\quad P_1\ =\ \Pr[\,\mathsf{SD}(L,\mathsf{SE}(K,T_1))=T_1\,]\ .$$

The probability that $\text{dec}=1$ at the end of the first For loop in subroutine $\mathsf{KeyTest}$ is $P_0^kP_1^k$ and the probability that $T^1=\cdots=T^k=T_{1-b}$ is at most $(1-P_b)^k$. So we have

$$\begin{aligned}
\Pr\left[\,(\text{dec},\text{gs})\xleftarrow{\$}\mathsf{KeyTest}(L)\ :\ (\text{dec},\text{gs})=(1,1-b)\,\right] &= P_0^kP_1^k\cdot(1-P_b)^k\\
&\le P_b^k\cdot(1-P_b)^k\\
&= [P_b(1-P_b)]^k\\
&\le 4^{-k}\ .
\end{aligned}$$

The last line is true because the function $f\colon[0,1]\to\mathbb{R}$ defined by $f(x)=x(1-x)$ attains its maximum at $x=1/2$ and the value of this maximum is $1/4$. This concludes the proof. $\blacksquare$

Returning to the proof of Equation (8), we define the following events in $\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)$:

$\mathsf{FailTest}$: There exists $L \neq K$ such that L was queried to H and $\mathsf{KeyTest}(L)$ returned $(1, 1-b)$ in subroutine $\mathrm{HSim}(L)$

$\mathsf{Illegit}$: There exist i, j and L such that L was queried to H and either $C_j^i[L]$ or $C^i[L]$ was queried by $\boldsymbol{H}$ to $\mathsf{SD}(K, \cdot)$.

We obtain Equation (8) as shown below. Justifications follow the formulas:

$$\begin{aligned}
&\Pr_{\boldsymbol{H}}[\,\mathsf{HG} \vee (\mathsf{Succ}(\boldsymbol{H}) \wedge \neg\mathsf{HG} \wedge \neg\mathsf{GH})\,] \\
&\quad\leq \Pr_{\boldsymbol{S}}[\,\mathsf{HG} \vee (\mathsf{Succ}(\boldsymbol{H}) \wedge \neg\mathsf{HG} \wedge \neg\mathsf{GH}) \mid \neg\mathsf{FailTest}\,] + \Pr_{\boldsymbol{S}}[\,\mathsf{FailTest}\,] \qquad (10)\\
&\quad\leq \Pr_{\boldsymbol{S}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) = 1\,\right] + \Pr_{\boldsymbol{S}}[\,\mathsf{Illegit}\,] + \Pr_{\boldsymbol{S}}[\,\mathsf{FailTest}\,] \qquad (11)\\
&\quad\leq \Pr_{\boldsymbol{S}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) = 1\,\right] + \Pr_{\boldsymbol{S}}[\,\mathsf{Illegit} \mid \neg\mathsf{FailTest}\,] + 2 \cdot \Pr_{\boldsymbol{S}}[\,\mathsf{FailTest}\,] \\
&\quad\leq \Pr_{\boldsymbol{S}}\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k) = 1\,\right] + \frac{O(Q(k))}{2^k}\,. \qquad (12)
\end{aligned}$$

To justify Equation (10), observe that if event $\mathsf{FailTest}$ does not happen, then the simulation of $\boldsymbol{H}$ done by $\boldsymbol{S}$ is correct. (If HG occurs, then prior to this g^{xy} was not a query to G, so the simulation of the G oracle is correct. If $\neg\mathsf{HG} \wedge \neg\mathsf{GH}$ occurs then also g^{xy} was not a query to G so the simulation of the G oracle is correct. If $\mathsf{FailTest}$ does not occur then the replies to queries to H are correct.)

To justify Equation (11) first note that if event HG occurs, then the $L = K$ case of Lemma A.1 tells us that $\boldsymbol{S}$ halts with correct output. On the other hand, if neither HG nor GH occur, then $\boldsymbol{S}$ halts with correct output as long as $\boldsymbol{H}$ does. But $\mathbf{Exp}^{\text{ind-cca}}_{\mathsf{SS},\boldsymbol{S}}(k)$ can still fail to return 1 because $\boldsymbol{S}$ aborted due to the occurence of $\mathsf{Illegit}$. (When the latter occurs, $\boldsymbol{S}$ aborts to avoid calling its oracle $\mathsf{SD}(K, \cdot)$ on a ciphertext returned by its $\mathsf{SE}(K, \mathrm{LR}(\cdot, \cdot, b))$ oracle.)

To justify Equation (12) first note that Lemma A.1 together with the fact that the total number of queries is at most $Q(k)$ implies that $\Pr_{\boldsymbol{S}}[\,\mathsf{FailTest}\,] \leq Q(k)/4^k$. Next we observe that if $\mathsf{FailTest}$ does not occur then $\boldsymbol{H}$ gets no information about T_0, T_1 other than that they are random distinct k-bit strings. The unique decryptability of SS then tells us that $\Pr_{\boldsymbol{S}}[\,\mathsf{Illegit} \mid \neg\mathsf{FailTest}\,]$ is bounded above by the probability of guessing either T_0 or T_1 in $Q(k)$ tries, and this is $O(Q(k)/2^k)$.

Proof of Equation (9). We define the following event in $\mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k)$:

$\mathsf{FailDec}$: There exist times $t_1 < t_2$ and Y', W', L such that all the following hold:
- query (Y', W') was made to $\mathsf{AD}^{G,H}((q, g, x), \cdot)$ at time t_1 and $\mathrm{ADSim}(Y', W')$ returned $\perp$
- query L was made to H at time t_2
- $g^{\mathrm{HT}[L]} = Y'$.

The answers provided by $\mathrm{ADSim}(\cdot, \cdot)$ are correct exactly when this event does not occur. Furthermore, if there is a time at which query g^{xy} to G occurs and GH is true then query K to H has not occured at this time, and thus the answers to queries to H have been correct. Thus

$$\Pr_{\boldsymbol{C}}\left[\,\mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) = 1\,\right] \;\geq\; \frac{\Pr_{\boldsymbol{H}}[\,\mathsf{GH}\,] - \Pr_{\boldsymbol{C}}[\,\mathsf{FailDec}\,]}{Q(k)}\,.$$

Re-arranging, we get

$$\Pr_{\boldsymbol{H}}[\,\mathsf{GH}\,] \;\leq\; Q(k) \cdot \Pr_{\boldsymbol{C}}\left[\,\mathbf{Exp}^{\text{cdh}}_{\mathsf{CG},\boldsymbol{C}}(k) = 1\,\right] + \Pr_{\boldsymbol{C}}[\,\mathsf{FailDec}\,]\,. \qquad (13)$$

At any point in time, a query L to H has probability at most ℓ/q of making $\mathsf{FailDec}$ happen, where ℓ is the number of queries that have been made to $\mathsf{AD}^{G,H}((q,g,x),\cdot)$ at this time. Recall that $k = |\langle 2q+1\rangle|$ and thus $q \geq 2^{k-2}$. Putting these observations together we get

$$\Pr_{\boldsymbol{C}}[\,\mathsf{FailDec}\,] \;\leq\; \frac{Q(k)^2}{q} \leq \frac{Q(k)^2}{2^{k-2}} \;=\; \frac{O(Q(k)^2)}{2^k}\,.$$

Putting this together with Equation (13) completes the proof of Equation (9).