A preliminary version of this papers appears in the proceedings of PKC 2015. This is the full version, appearing as IACR ePrint Archive Record 2014/376.

How Secure is Deterministic Encryption?

Mihir Bellare[1] Rafael Dowsley[2] Sriram Keelveedhi[3]

February 11, 2015

Abstract

This paper presents three curious findings about deterministic public-key encryption (D-PKE) that further our understanding of its security, in particular because of the contrast with standard, randomized public-key encryption (R-PKE):

- It would appear to be a triviality, for any primitive, that security in the standard model implies security in the random-oracle model, and it is certainly true, and easily proven, for R-PKE. For D-PKE it is not clear and depends on details of the definition. In particular we can show it in the non-uniform case but not in the uniform case.
- The power of selective-opening attacks (SOA) comes from an adversary's ability, upon corrupting a sender, to learn not just the message but also the coins used for encryption. For R-PKE, security is achievable. For D-PKE, where there are no coins, one's first impression may be that SOAs are vacuous and security should be easily achievable. We show instead that SOA-security is impossible, meaning no D-PKE scheme can achieve it.
- For R-PKE, single-user security implies multi-user security, but we show that there are D-PKE schemes secure for a single user and insecure with two users.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `mihir@cs.ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/mihir`. Supported in part by NSF grants CNS-1228890 and CNS-1116800.

[2] Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Am Fasanengarten 5, Geb. 50.34
76131 Karlsruhe, Germany. Email: `rafael.dowsley@kit.edu`. URL: `https://crypto.iti.kit.edu/?id=dowsley`.

[3] Work done while at UCSD, supported in part by NSF grants CNS-1228890 and CNS-1116800. Email: `sriramkr@cs.ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/skeelvee`.

Contents

1 Introduction

Public-key encryption (PKE) schemes are usually randomized, in order to achieve goals like IND-CPA [28]. BBO [5] introduced deterministic PKE (D-PKE), arguing that it offers practical benefits over randomized PKE (R-PKE) in certain applications. These include efficient search on encrypted databases [5] and resilience in the face of the low-quality randomness that pervades systems [6, 40].[1]

BBO [5] provide a definition PRIV of "best possible" security for D-PKE, and ROM constructions achieving it. Equivalent, IND-style formulations appear in [9]. These definitions are unusual, and achieving them in the standard model (SM) is challenging. Emerging as a practically-motivated notion of theoretical depth and interest, D-PKE has attracted significant foundational work as researchers aim to understand the properties and achievability of the basic definitions and variants [9, 18, 19, 32, 27, 38, 10]. We continue this line of work.

Our work. This paper shows that determinism impacts security in beyond-obvious ways. Specifically, we consider three questions. The first is whether security in the standard model implies security in the ROM. The second is whether D-PKE is secure under selective-opening attack. The last is whether single-user security implies multi-user security. Figure 1 summarizes our findings, which are discussed in more depth below. On the practical side, our work indicates that care must be taken in the use of D-PKE. On the theoretical side it indicates further foundational subtleties for D-PKE, and, more broadly, for multi-stage security definitions, in the wake of those already indicated in [39, 41].

Background. In R-PKE, the encryption algorithm Enc takes the public (encryption) key pk, message m and coins r to return a ciphertext $c = \mathsf{Enc}(pk, m; r)$. The basic notion of security is IND-CPA [28, 7]. An adversary is a pair (A_1, A_2) of PT algorithms. The game picks keys (pk, sk) and a challenge bit b. We run A_1 on input pk to get a pair (m_0, m_1) of messages and state information st. The game picks random coins r, computes challenge ciphertext $c = \mathsf{Enc}(pk, m_b; r)$ and runs A_2 on c, st to get a bit b'. Security requires that $2\Pr[b = b'] - 1$ is negligible.

In D-PKE [5], there are no coins, Enc taking pk, m to return $c = \mathsf{Enc}(pk, m)$. Such a scheme cannot achieve IND-CPA. The notion we use is IND [9], an indistinguishability variant of the PRIV notion of [5]. An adversary is a pair (A_1, A_2) of PT algorithms. The game picks keys (pk, sk) and a challenge bit b. We run A_1 (it does not get pk) to get a pair $(\mathbf{m}_0, \mathbf{m}_1)$ of vectors of messages (but no state information). The game computes challenge ciphertext vector $\mathbf{c} = \mathsf{Enc}(pk, \mathbf{m}_b)$, encryption being component-wise, and runs A_2 on $\mathbf{c}, pk$ to get a bit b'. Security requires that $2\Pr[b = b'] - 1$ is negligible. Important restrictions are that (1) A_1 does not get the public key (2) each individual message in the vectors $\mathbf{m}_0, \mathbf{m}_1$ has high min-entropy, meaning is statistically unpredictable, and (3) A_1, A_2 do not communicate directly, meaning no state information is passed from A_1 to A_2. These restrictions are necessary, for without them security is not achievable.

In the ROM [13], both stages of the adversary have access to the random oracle RO, whether for R-PKE or D-PKE. In the latter case, the min-entropy condition is required to hold even given (conditioned on) the RO.

Does SM-security imply ROM-security? That security in the standard model (SM) implies security in the ROM appears to be a triviality or tautology, true for any primitive. To be specific, suppose we have a standard-model R-PKE scheme, meaning the algorithms of the scheme make no calls to RO. Suppose it is IND-CPA in the SM. Then it is IND-CPA in the ROM. Intuitively this seems clear because if the scheme does not use the RO, then giving the adversary access to RO cannot violate security. If we want to prove the claim formally, we could do so by reduction. Given a ROM adversary (A_1, A_2), we build SM adversary (B_1, B_2) with the same advantage by just having B_1 and B_2 simulate the RO. Thus, B_1 maintains a table H, and runs A_1. When the latter makes a query

[1] Weak randomness leads to catastrophic failures in R-PKE including the ability to recover the plaintext from the ciphertext in schemes including GM, El Gamal and Rabin-SAEP [35].

Primitive	**SM ⇒ ROM**	**SOA**	**SU ⇒ MU**
R-PKE	Yes	Yes	Yes
D-PKE	Sometimes	No	No

Figure 1: **Summary of our results:** The first column indicates whether or not security in the standard model (SM) implies security in the ROM, the "sometimes" for D-PKE reflecting that we can show it in the non-uniform case but not in the uniform case. The second column indicates whether or not security against selective-opening attack (SOA) is achievable. The third column indicates whether or not single-user (SU) security implies multi-user (MU) security.

RO(x), adversary B_1 checks if $H[x]$ is defined, and, if not, picks it at random, in either case returning $H[x]$ to A_1 as the answer. When A_1 halts with output (m_0, m_1) and state st_A, adversary B_1 halts with output (m_0, m_1) and state st_B, where the latter consists of st_A plus the populated (defined) part of table H, which has polynomial size. Now B_2, given c, st_B, runs $A_2(c, \mathsf{st}_A)$, continuing to respond to A_2's oracle queries via table H, suitably augmenting it as necessary for new queries. Eventually B_2 returns whatever A_2 returns. It is clear that SM adversary (B_1, B_2) simulates ROM adversary (A_1, A_2) perfectly and has the same advantage.

The claim that SM security implies ROM security, and the simulation argument above to establish it, hardly seem specific to R-PKE. It would appear to be true that SM security trivially implies ROM security for any primitive via such an argument.

But for D-PKE, the argument fails, and whether SM security implies ROM security is not clear. To see why, let us try to mimic the above argument for D-PKE. We can design B_1, simulating A_1, in the same way. The difficulty is that B_1 cannot pass its partial table H to B_2, for no state information is allowed to flow from B_1 to B_2. This leaves B_2 stuck. It could simulate a new RO for A_2, but in the real ROM game, A_1, A_2 see the same RO, not different ones. The question this raises is whether the difficulty is inherent, meaning SM security does not imply ROM security, or whether some alternative argument can show the implication.

We find that the answer depends on details of the definition. Let $\mathrm{IND_u}, \mathrm{IND_{nu}}$ denote, respectively, the uniform and non-uniform renditions of IND. That is, in the first case, the adversaries are TMs while in the second they are families of circuits. We show that SM security implies ROM security for $\mathrm{IND_{nu}}$. Our proof works by starting with ROM adversaries $\mathsf{A}_1, \mathsf{A}_2$, hardwiring a $q(\cdot)$-wise independent hash function h into the circuits of B_1, B_2, and having these circuits use h to simulate RO for A_1, A_2, with $q(\cdot)$ depending on the number of oracle queries of A_1 and A_2. We show that there exists a "good" choice of h under which the simulation is correct. However, in the case of $\mathrm{IND_u}$, we were not able to settle the question. That is, we see no way to prove that SM security implies ROM security (it is not clear how to perform a simulation, and it is not clear there is any other approach to the proof) but nor can we imagine a counter-example (it would need to exploit the fact that the scheme is secure for uniform adversaries but not for non-uniform ones, for otherwise the claim is true).

Intuitively, it is hard for us to imagine how a SM scheme can be insecure in the ROM, meaning how an adversary can exploit the RO when scheme algorithms do not even use it.[2] We found it curious that it was not obvious how to prove this and that it is not clear if it is even true in the uniform case.

These findings show further subtleties for multi-stage security definitions following ones already discovered by [39, 41], making D-PKE a central test case in this subtle and surprising domain.

Is SOA-secure D-PKE achievable? In a selective opening attack (SOA) on a R-PKE scheme, a vector $\mathbf{m}$ of n messages is chosen from some distribution, a vector $\mathbf{r}$ of random and independent coins

[2] One might imagine the adversary gaining an advantage by having B_1 pick messages that depend on the RO in some clever way. The reason this does not appear to help the adversary is that each message is required to have high min-entropy even conditioned on the entire random oracle.

is chosen, and the adversary A is given the ciphertext vector $\mathbf{c} = \mathsf{Enc}(pk, \mathbf{m}; \mathbf{r})$. A responds with a subset I of $\{1, \ldots, n\}$. In the message-only version of the attack, it is returned $\langle \mathbf{m}[i] : i \in I \rangle$; in full SOA, it is returned both $\langle \mathbf{m}[i] : i \in I \rangle$ and $\langle \mathbf{r}[i] : i \in I \rangle$. In either case, to win, it has to compute some non-trivial information about $\langle \mathbf{m}[i] : i \notin I \rangle$. Security for the message-only version is implied by IND-CPA, as shown in [16], and is thus easily achievable. Security for full SOA is not implied by IND-CPA [8]. However, using lossy encryption [31, 37, 11, 16], it is shown in [11, 16] that there exist schemes that provide full SOA under standard assumptions, so full SOA security is achievable, under standard assumptions in the standard model. Subsequently, further schemes have been provided as well [26, 29].

The question of security of D-PKE under SOA has not been considered before, and we initiate an investigation. A vector $\mathbf{m}$ of n messages is again chosen from some distribution, and the adversary A is given the ciphertext vector $\mathbf{c} = \mathsf{Enc}(pk, \mathbf{m})$. A responds with a subset I of $\{1, \ldots, n\}$, is returned $\langle \mathbf{m}[i] : i \in I \rangle$, and, to win, has to compute some non-trivial information about $\langle \mathbf{m}[i] : i \notin I \rangle$. We note that what we have defined is the message-only version. Naturally, there is no "full" SOA here, since there are no coins used, and thus none to expose.

The difficulty of achieving SOA-secure R-PKE lies in exposure of the coins. Since D-PKE has no coins, one's first impression may be that SOA-security for it would be like message-only SOA-security for R-PKE and thus easy to achieve. To the contrary, we show that SOA-secure D-PKE is impossible. That is, there is no D-PKE scheme that is SOA-secure. Given *any* D-PKE scheme, we give an attack violating SOA-security.

The contrast with R-PKE is two-fold. For the latter, SOA is easy in the message-only case, and, with exposure of coins, even if not easy, it is achievable. But for D-PKE, it is simply not achievable. The key element of our proof is to show that for any D-PKE scheme there is an algorithm that can impose and verify an association between a message and ciphertext that is unique with high probability, *even for dishonestly chosen public keys.* We combine this with the technique of BDWY [8] to obtain our impossibility result. We note that for R-PKE the BDWY technique did not show impossibility of (full) SOA for all R-PKE schemes, but for a subclass of them, while we are using the technique to rule out SOA-security for *all* D-PKE schemes.

The problem of SOA-security has been the subject of many works [3, 21, 20, 2, 23, 33, 25, 22, 36, 11, 16, 26, 15, 29, 17, 30]. These have looked at R-PKE, commitment and IBE. We are the first to consider SOA for D-PKE.

Does SU security imply MU security? The basic IND-CPA notion for R-PKE [28] is a single-user (SU) setting, meaning there is only one public key in the game. In practice, many users, each with their own key pair, could encrypt messages, and these messages may be related. Security of R-PKE in the multi-user (MU) setting was defined in [4, 1]. They showed that SU security implied MU security, meaning any R-PKE scheme that meets the usual SU IND-CPA notion is also MU secure.

It is natural to ask whether the same is true for D-PKE, namely whether SU security, in the form of IND, implies MU security. We define MU security for D-PKE and show that the answer to the question is "no." That is, we present a counter-example, namely a D-PKE scheme that we show meets the standard SU IND definition, but we give an attack showing that it fails to be MU-secure. Indeed, it is insecure even with just two users, meaning when there are two public keys in the picture.

BBO [5] had conjectured that indeed SU security did not in general imply MU security for D-PKE. Our results prove and confirm this conjecture. Brakerski and Segev [19] define MU security of D-PKE in the auxiliary input setting and give a scheme that achieves it for messages that are block sources, but they do not show a separation between the SU and MU settings. Dodis, Lee and Yum [24] give another example of a setting where SU security does not imply MU security, namely optimistic fair exchange.

2 Preliminaries

Notation and conventions. We let $\lambda \in \mathbb{N}$ denote the security parameter. If $n \in \mathbb{N}$ then we let 1^n denote the string of n ones and $[n]$ denote the set $\{1, \ldots, n\}$. If A is a finite set, then let $|A|$ denote its size, and $a \xleftarrow{\$} A$ denote sampling a uniformly at random from A. The empty string is denoted by ε. If a and b are two bit strings, we denote by $a \,\|\, b$ their concatenation. We use boldface letters for vectors. For any vector $\mathbf{x}$, we let $|\mathbf{x}|$ denote the number of its components. We say $\mathbf{x}$ is an n-vector if $|\mathbf{x}| = n$. For $i \in [|\mathbf{x}|]$ we let $\mathbf{x}[i]$ denote the i-th component of $\mathbf{x}$. We let $\mathrm{Maps}(D, R)$ denote the set of all functions $f\colon\ D \to R$.

Algorithms are randomized, unless otherwise specified as being deterministic. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \xleftarrow{\$} A(x_1, \ldots)$ be the resulting of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all y that have positive probability of being output by $A(x_1, \ldots)$. A function $\epsilon\colon\ \mathbb{N} \to \mathbb{R}$ is negligible if for every polynomial p, there exists $\lambda_p \in \mathbb{N}$ such that $\epsilon(\lambda) \leq 1/p(\lambda)$ for all $\lambda \geq \lambda_p$. An algorithm A is uniform if there exists a Turing machine T which halts with the output of A on all inputs. An algorithm A is non-uniform if there exists a sequence of circuits $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ such that C_λ computes $A(1^\lambda, \ldots)$.

Games. Our definitions and proofs use the code-based game-playing framework of [14] with some of the syntax of [39]. A game $G(\lambda)$ (see Figure 2 for an example) consists of a Main procedure, and possibly others, and begins by executing Main, which runs an adversary A after some initialization steps. A is given oracle access to certain game procedures. After A finishes executing, G performs steps with A's output to produce some output itself. We assume that boolean variables are initialized to false, that sets are initialized to $\emptyset$, strings are initialized to ϵ, and that integers are initialized to 0. We denote by $G^A \Rightarrow y$ the event that an execution of G with A outputs y. We abbreviate $G^A \Rightarrow \mathsf{true}$ as G^A.

Functions families. A family of functions HF is a PT, deterministic algorithm that defines for each $\lambda \in \mathbb{N}$ a map $\mathsf{HF}(1^\lambda, \cdot, \cdot) : \{0,1\}^{\mathsf{HF.kl}(\lambda)} \times \{0,1\}^{\mathsf{HF.il}(\lambda)} \to \{0,1\}^{\mathsf{HF.ol}(\lambda)}$. Here $\mathsf{HF.kl}, \mathsf{HF.il}, \mathsf{HF.ol}\colon\ \mathbb{N} \to \mathbb{N}$ are the key, input and output lengths of HF, respectively. We extend HF to vectors (in a component-wise way) via $\mathsf{HF}(1^\lambda, k, \mathbf{x}) = (\mathsf{HF}(1^\lambda, k, \mathbf{x}[1]), \ldots, \mathsf{HF}(1^\lambda, k,)\mathbf{x}[|\mathbf{x}|])$ for all $\lambda \in \mathbb{N}$, all $k \in \{0,1\}^{\mathsf{HF.kl}(\lambda)}$ and all vectors $\mathbf{x}$ over $\{0,1\}^{\mathsf{HF.il}(\lambda)}$.

3 Deterministic PKE

We provide definitions for D-PKE following [5, 9]. We give a unified treatment of the ROM and the SM by regarding the latter as a special case of the former.

D-PKE. A deterministic public key encryption (D-PKE) scheme DE specifies four PT algorithms and related functions as follows. The parameter generator algorithm $\mathsf{DE.Pg}$ takes as input a unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$ and returns the system parameters $\pi \in \{0,1\}^{\mathsf{DE.pl}(\lambda)}$ which are common to all users. The key generation algorithm $\mathsf{DE.Kg}$ takes as input π and outputs a public encryption key $pk \in \{0,1\}^{\mathsf{DE.pkl}(\lambda)}$ and a secret decryption key sk. Given inputs $1^\lambda, \pi, pk$, a message $m \in \{0,1\}^{\mathsf{DE.ml}(\lambda)}$ and access to an oracle $\mathrm{R}\colon\ \{0,1\}^{\mathsf{DE.ROil}(\lambda)} \to \{0,1\}^{\mathsf{DE.ROol}(\lambda)}$, the deterministic encryption algorithm $\mathsf{DE.Enc}$ outputs a ciphertext $c = \mathsf{DE.Enc}^{\mathrm{R}}(1^\lambda, \pi, pk, m)$. Given inputs $1^\lambda, \pi, sk$, a ciphertext c and oracle R, the deterministic decryption algorithm $\mathsf{DE.Dec}$ output either a message $m \in \{0,1\}^{\mathsf{DE.ml}(\lambda)}$, or $\perp$. Here $\mathsf{DE.pl}, \mathsf{DE.pkl}, \mathsf{DE.ml}\colon\ \mathbb{N} \to \mathbb{N}$ are the parameter, public key and message length functions of DE, respectively, while $\mathsf{DE.ROil}, \mathsf{DE.ROol}\colon\ \mathbb{N} \to \mathbb{N}$ are the RO input and output length functions, respectively. Correctness requires that for all $\lambda \in \mathbb{N}$, all $\pi \in [\mathsf{DE.Pg}(1^\lambda)]$,

```
Main $\mathrm{IND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$
$\mathsf{st} \xleftarrow{\$} \mathsf{A.cs}^{\mathrm{RO}}(1^\lambda)$ ; $(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}^{\mathrm{RO}}(1^\lambda, \mathsf{st})$
$\pi \xleftarrow{\$} \mathsf{DE.Pg}(1^\lambda)$ ; $(pk, sk) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$
$b \xleftarrow{\$} \{0,1\}$ ; $\mathbf{c} \leftarrow \mathsf{DE.Enc}^{\mathrm{RO}}(1^\lambda, \pi, pk, \mathbf{m}_b)$
$b' \xleftarrow{\$} \mathsf{A.g}^{\mathrm{RO}}(1^\lambda, \pi, pk, \mathsf{st}, \mathbf{c})$ ; Return $(b = b')$

$\mathrm{RO}(x)$
If $T[x] = \bot$ then $T[x] \xleftarrow{\$} \{0,1\}^{\mathsf{DE.ROol}(\lambda)}$
Return $T[x]$
```

```
Main $\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)$
$(\mathsf{st}, m, \mathrm{R}) \xleftarrow{\$} \mathsf{P}(1^\lambda)$
$(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}^{\mathrm{R}}(1^\lambda, \mathsf{st})$
Return $(\exists\, i, b \,:\, \mathbf{m}_b[i] = m)$
```

Figure 2: The IND game used to define security of D-PKE scheme DE and the PRED game used to define unpredictability of adversary A.

all $[(pk, sk) \in [\mathsf{DE.Kg}(1^\lambda, \pi)]$, all $m \in \{0,1\}^{\mathsf{DE.ml}(\lambda)}$ and all $\mathrm{R} \in \mathrm{Maps}[\mathsf{DE.ROil}(\lambda), \mathsf{DE.ROol}(\lambda)]$ we have $\mathsf{DE.Dec}^{\mathrm{R}}(1^\lambda, \pi, sk, \mathsf{DE.Enc}^{\mathrm{R}}(1^\lambda, \pi, pk, m)) = m$. We extend $\mathsf{DE.Enc}$ to take input vectors of messages by defining $\mathsf{DE.Enc}^{\mathrm{R}}(1^\lambda, \pi, pk, \mathbf{m}) = (\mathsf{DE.Enc}^{\mathrm{R}}(1^\lambda, \pi, pk, \mathbf{m}[1]), \ldots, \mathsf{DE.Enc}^{\mathrm{R}}(1^\lambda, \pi, pk, \mathbf{m}[|\mathbf{m}|]))$, and similarly we let $\mathsf{DE.Dec}^{\mathrm{R}}(1^\lambda, \pi, sk, \mathbf{c}) = (\mathsf{DE.Dec}^{\mathrm{R}}(1^\lambda, \pi, sk, \mathbf{c}[1]), \ldots, \mathsf{DE.Dec}^{\mathrm{R}}(1^\lambda, \pi, sk, \mathbf{c}[|\mathbf{c}|]))$. We say that DE is a standard-model (SM) scheme if $\mathsf{DE.Enc}, \mathsf{DE.Dec}$ make no oracle queries, and in this case we will omit the superscript R to $\mathsf{DE.Enc}, \mathsf{DE.Dec}$.

IND security. We define IND security of a D-PKE scheme DE following BFOR [9]. An IND adversary A specifies a common-state generation algorithm $\mathsf{A.cs}$, a message-generation algorithm $\mathsf{A.msg}$ and a guessing algorithm $\mathsf{A.g}$, all PT. On input 1^λ, algorithm $\mathsf{A.cs}$ generates state information st that will be passed to both $\mathsf{A.msg}$ and $\mathsf{A.g}$. Algorithm $\mathsf{A.msg}$, on input $1^\lambda, \mathsf{st}$ returns a pair $(\mathbf{m}_0, \mathbf{m}_1)$ of vectors of messages with $|\mathbf{m}_0| = |\mathbf{m}_1| = \mathsf{A.nm}(\lambda)$ and $\mathbf{m}_0[i], \mathbf{m}_1[i] \in \{0,1\}^{\mathsf{DE.ml}(\lambda)}$, where $\mathsf{A.nm}\colon \mathbb{N} \to \mathbb{N}$ is the number-of-messages function associated to A. It is required that the strings (messages) $\mathbf{m}_0[1], \ldots, \mathbf{m}_0[|\mathbf{m}_0|]$ are distinct and the strings (messages) $\mathbf{m}_1[1], \ldots, \mathbf{m}_1[|\mathbf{m}_1|]$ are distinct. Also associated to DE are functions $\mathsf{DE.ROil}, \mathsf{DE.ROol}$, the input and output length of the RO that is used by the scheme. We say that A is a standard-model adversary if it makes no oracle queries, and in this case we may omit giving it an oracle.

The $\mathrm{IND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$ game associated with DE and adversary A is described on the left of Figure 2. We define the advantage of A via $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\lambda) = 2 \cdot \Pr[\mathrm{IND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)] - 1$ for all $\lambda \in \mathbb{N}$. If $\mathcal{A}$ is a class (set) of adversaries then we say that DE is $\mathrm{IND}[\mathcal{A}]$-secure if $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\cdot)$ is negligible for all $\mathsf{A} \in \mathcal{A}$. It is convenient to view $\mathrm{IND}[\mathcal{A}]$ as a set, so that $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}]$ iff DE is $\mathrm{IND}[\mathcal{A}]$-secure. With this framework, we can now obtain various variants of the notion by varying and restricting the class $\mathcal{A}$.

First, we must impose the necessary condition that messages being encrypted have high min-entropy. In game PRED of Figure 2, the predictor adversary P begins by specifying st and a guess m as to a message that $\mathsf{A.msg}$ will generate. It also specifies the function $\mathrm{R} \in \mathrm{Maps}[\mathsf{DE.ROil}(\lambda), \mathsf{DE.ROol}(\lambda)]$ that will play the role of the RO. This captures the requirement that high min-entropy is required across all choices of the oracle. We let $\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A},\mathsf{P}}(\lambda) = \Pr[\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)]$ for all $\lambda \in \mathbb{N}$. We say that A is unpredictable if $\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A},\mathsf{P}}(\cdot)$ is negligible for all P. We stress that here P is not restricted to PT but may be computationally unbounded. If A is a standard model adversary then we may omit R in the output of P.

Following [9], our adversaries A are three stage. If $\mathsf{A.cs}$ always returns ε then we say that A has trivial initial state and we may refer to A as a two-stage adversary. In BFOR [9], definitions of security are relative to two-stage adversaries, three-stage ones being introduced in order to facilitate proofs. Accordingly, our definitions of security will also be in terms of two-stage adversaries.

We are now ready to define adversary classes of interest. We consider two dimensions: the model (ROM or SM), and the type of computation (non-uniform or uniform). With two choices in each category, we get 4 classes of adversaries and 4 corresponding notions of security for D-PKE. Proceeding to the details, we let $\mathcal{A}_3$ be the class of all PT, 3-stage, unpredictable adversaries and $\mathcal{A}_2 \subseteq \mathcal{A}_3$ the class of all PT, 2-stage unpredictable adversaries. We let $\mathcal{A}^{\mathrm{rom}}$ denote the class of ROM adversaries, and $\mathcal{A}^{\mathrm{sm}} \subseteq \mathcal{A}^{\mathrm{rom}}$ the class of SM adversaries. We let $\mathcal{A}^{\mathrm{nu}}$ denote the class of non-uniform adversaries, and $\mathcal{A}^{\mathrm{u}} \subseteq \mathcal{A}^{\mathrm{nu}}$ the class of uniform adversaries. Then our 4 classes are $\mathcal{A}_2^{\mathrm{xm-xu}} = \mathcal{A}^{\mathrm{xm}} \cap \mathcal{A}^{\mathrm{xu}} \cap \mathcal{A}_2$ for $\mathrm{xm} \in \{\mathrm{rom}, \mathrm{sm}\}$ and $\mathrm{xu} \in \{\mathrm{nu}, \mathrm{u}\}$. The 4 corresponding notions of D-PKE security are $\mathrm{IND}[\mathcal{A}_2^{\mathrm{xm-xu}}]$ for $\mathrm{xm} \in \{\mathrm{rom}, \mathrm{sm}\}$ and $\mathrm{xu} \in \{\mathrm{nu}, \mathrm{u}\}$.

4 Does SM security imply ROM security?

We now explore if a D-PKE scheme that is IND-secure in the standard model (SM) is IND-secure in the ROM.

Problem and approach. It is easy to show that a SM R-PKE scheme retains its security in the ROM, where the adversary has access to the random oracle, because a SM adversary can simply simulate the random oracle for a ROM adversary. Indeed, that SM security implies ROM security seems to have been viewed as trivial and true for any primitive. We are about to see, however, that for D-PKE the answer is less clear.

We are given a SM D-PKE scheme DE that is secure in the SM, meaning its algorithms make no calls to RO and it is secure against adversaries that make no calls to RO. We ask if DE remains secure in the ROM, meaning when the adversary is allowed to query RO. The reason an adversary A may be able now to do more is that $\mathsf{A.msg}$ may create messages that depend in some clever way on RO and then $\mathsf{A.g}$ could exploit the fact that it has access to the same RO to figure out something about the messages from the ciphertexts. Intuitively, however, it is difficult to see how this could happen because messages are required to have high min-entropy even given RO. However, it is not clear how to prove it, which raises the question of whether it is even true. The difficulty is that no communication is allowed from the message-finding stage of the adversary to the guessing stage, and so a simulating SM adversary has no obvious way to ensure that these two stages have a common view of the random oracle it is simulating.

We will first present Lemma 4.1 showing the claim is true in the 3-stage adversary formulation of the IND games. Namely given a SM D-PKE scheme and given a 3-stage ROM adversary A, we show how to simulate A with a 3-stage SM adversary B so that the latter has the same advantage as A. The proof uses a $q(\cdot)$-wise independent hash function, with the polynomial q depending on A, as the common initial state created by $\mathsf{B.cs}$. The lemma is true both in the uniform and the non-uniform settings. However, recall that IND security is defined with respect to adversaries that have trivial initial state, meaning are two stage. And in our reduction, B will have non-trivial initial state even if A has trivial initial state. So the lemma does not directly show that IND in the SM implies IND in the ROM. In the non-uniform case, however, we can flatten the constructed 3-stage adversary B into an equivalent one with trivial initial state, thereby concluding that if SM D-PKE scheme DE is in $\mathrm{IND}[\mathcal{A}_2^{\mathrm{sm-nu}}]$ then it is also in $\mathrm{IND}[\mathcal{A}_2^{\mathrm{rom-nu}}]$. In the uniform setting we have no obvious way to remove the non-trivial initial state of B, and thus are not able to conclude that DE being in $\mathrm{IND}[\mathcal{A}_2^{\mathrm{sm-u}}]$ implies it is in $\mathrm{IND}[\mathcal{A}_2^{\mathrm{rom-u}}]$. This very basic question (surprisingly) remains open.

q-wise independent function families. We say that a family HF of functions is $q(\cdot)$-wise independent if for all $\lambda \in \mathbb{N}$, all $q(\lambda)$-vectors $\mathbf{x}$ over $\{0,1\}^{\mathsf{HF.il}(\lambda)}$ all of whose entries are distinct, and all $q(\lambda)$-vectors $\mathbf{y}$ over $\{0,1\}^{\mathsf{HF.ol}(\lambda)}$ we have $\Pr[\mathsf{HF}(1^\lambda, k, \mathbf{x}) = \mathbf{y}] = 2^{-q(\lambda)\cdot\mathsf{HF.ol}(\lambda)}$, where the probability is over k chosen at random from $\{0,1\}^{\mathsf{HF.kl}(\lambda)}$.

From SM security to ROM security with 3 stages. The following lemma says that for any SM D-PKE scheme (meaning, the scheme algorithms do not call the RO), a 3-stage ROM adversary

A may be simulated by a 3-stage SM adversary B who achieves the same advantage as A. It does not follow that a 2-stage ROM adversary can be simulated by a 2-stage SM adversary since our constructed adversary B will have non-trivial initial state even if the given adversary A had trivial initial state.

Lemma 4.1 Let DE be a standard-model D-PKE scheme. Let $\mathsf{A} \in \mathcal{A}^{\mathrm{rom}} \cap \mathcal{A}_3$ be a 3-stage, PT ROM adversary. Then there is a 3-stage, PT standard-model adversary $\mathsf{B} \in \mathcal{A}^{\mathrm{sm}} \cap \mathcal{A}_3$ such that

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{B}}(\lambda) = \mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\lambda) \tag{1}$$

for all $\lambda \in \mathbb{N}$. Furthermore, if A is unpredictable then so is B and if A is uniform then so is B.

Proof of Lemma 4.1: Without loss of generality, we assume that there exists a polynomial $q : \mathbb{N} \to \mathbb{N}$ such that for all $\lambda \in \mathbb{N}$, adversary A always makes exactly $q(\lambda)$ RO queries in game $\mathrm{IND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$. Let HF be a $q(\cdot)$-wise independent family of functions with $\mathsf{HF.il} = \mathsf{DE.ROil}$ and $\mathsf{HF.ol} = \mathsf{DE.ROol}$. We define SM adversary B as follows:

```
$\mathsf{B.cs}(1^\lambda)$
    $k \xleftarrow{\$} \{0,1\}^{\mathsf{HF.kl}(\lambda)}$
    $\mathsf{st_A} \xleftarrow{\$} \mathsf{A.cs}^{\mathrm{ROSim}}(1^\lambda)$
    Return $(k, \mathsf{st_A})$
$\mathrm{ROSim}(x)$
    Return $\mathsf{HF}(1^\lambda, k, x)$

$\mathsf{B.msg}(1^\lambda, \mathsf{st_B})$
    $(k, \mathsf{st_A}) \leftarrow \mathsf{st_B}$
    $(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}^{\mathrm{ROSim}}(1^\lambda, \mathsf{st_A})$
    Return $(\mathbf{m}_0, \mathbf{m}_1)$
$\mathrm{ROSim}(x)$
    Return $\mathsf{HF}(1^\lambda, k, x)$

$\mathsf{B.g}(1^\lambda, \pi, pk, \mathsf{st_B}, \mathbf{c})$
    $(k, \mathsf{st_A}) \leftarrow \mathsf{st_B}$
    $b' \xleftarrow{\$} \mathsf{A.g}^{\mathrm{ROSim}}(1^\lambda, \pi, pk, \mathsf{st_A}, \mathbf{c})$
    Return $b'$
$\mathrm{ROSim}(x)$
    Return $\mathsf{HF}(1^\lambda, k, x)$
```

That is, $\mathsf{B.cs}$ picks at random a key defining a member of HF and passes it to $\mathsf{B.msg}, \mathsf{B.g}$. The latter use the function $\mathsf{HF}(1^\lambda, k, \cdot)$ to simulate the RO of A, via the ROSim procedure. Since A makes at most $q(\lambda)$ queries to RO, the $q(\lambda)$-wise independence of the family should result in a perfect simulation of the RO. Also, since both $\mathsf{B.msg}$ and $\mathsf{B.g}$ use the same function, $\mathsf{A.msg}$ and $\mathsf{A.g}$ will see a consistent RO across their two stages. As a result we expect that Equation (1) is true.

Formally proving that Equation (1) is true, however, is not straightforward because the RO queries are adaptive and $q(\cdot)$-wise independence is a non-adaptive condition, so some care must be taken. In Appendix Awe provide an analysis that handles this, and do not discuss it further here.

It is clear that if A is uniform then so is B. Assuming A is unpredictable we now have to show that B is unpredictable. Let $\mathsf{P_B}$ be a predictor adversary for B. We define a predictor adversary $\mathsf{P_A}$ for A as follows. On input 1^λ it runs $\mathsf{P_B}(1^\lambda)$ to get back $(\mathsf{st_B}, m)$. (Since B is SM, $\mathsf{P_B}$ returns a pair, not a triple.) It parses $\mathsf{st_B}$ as $(k, \mathsf{st_A})$ and returns $(\mathsf{st_A}, m, \mathsf{HF}(1^\lambda, k, \cdot))$. Then we have $\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A},\mathsf{P_A}}(\cdot) = \mathbf{Adv}^{\mathsf{pred}}_{\mathsf{B},\mathsf{P_B}}(\cdot)$. But the LHS is negligible by assumption, so the RHS is negligible as well. ▌

We note that alternatively, in place of a family of $q(\cdot)$-wise independent functions, we could have used a PRF, the key being chosen by $\mathsf{B.cs}$ and included in the state so that it is passed to $\mathsf{B.msg}, \mathsf{B.g}$. The latter would use the PRF under this key to simulate the RO for $\mathsf{A.msg}, \mathsf{A.g}$, respectively. O'Neill used this technique [34, Lemma 3.3.2] to partially remove the RO for a restricted class of D-PKE schemes.

SM SECURITY IMPLIES ROM SECURITY IN THE NON-UNIFORM SETTING. The following theorem uses Lemma 4.1 to show that if a D-PKE scheme DE is IND-secure in the standard model with respect to non-uniform adversaries, then it is IND-secure in the ROM with respect to non-uniform adversaries. The proof uses non-uniformity in a crucial way, and hence cannot be adapted to the uniform setting.

Theorem 4.2 Let DE be a SM D-PKE scheme such that $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}_2^{\mathrm{sm-nu}}]$. Then $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}_2^{\mathrm{rom-nu}}]$.

Proof: Let $\mathsf{A} \in \mathcal{A}_2^{\mathrm{rom-nu}}$ be an unpredictable, non-uniform PT ROM adversary with trivial initial state. By Lemma 4.1, we get an unpredictable, non-uniform PT SM adversary $\mathsf{B} \in \mathcal{A}^{\mathrm{sm}} \cap \mathcal{A}^{\mathrm{nu}} \cap \mathcal{A}_3$ such that $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{B}}(\cdot) = \mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\cdot)$. However, $\mathsf{B.cs}$ is *not* trivial, so the assumption that $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}_2^{\mathrm{sm-nu}}]$ does not allow us to conclude that $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{B}}(\cdot)$, and hence $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\cdot)$, is negligible. We modify B to an unpredictable, trivial initial state, non-uniform SM adversary $\mathsf{C} \in \mathcal{A}_2^{\mathrm{sm-nu}}$ with $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{C}}(\cdot) = \mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{B}}(\cdot)$. Now the assumption that $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}_2^{\mathrm{sm-nu}}]$ means that $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{C}}(\cdot)$ is negligible and hence so is $\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\cdot)$, showing that $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}_2^{\mathrm{rom-nu}}]$ as desired. To obtain C from B, we simply use coin fixing, namely we hardwire a best choice of the key k chosen randomly by $\mathsf{B.cs}(1^\lambda)$ into the circuits $\mathsf{C.msg}(1^\lambda, \cdots)$ and $\mathsf{C.g}(1^\lambda, \cdots)$ while letting $\mathsf{C.cs}$ always return ε. ▌

We note that the issues and difficulties associated with showing that SM security implies ROM security could also be viewed as arising from definitional shortcomings of existing formulations, and addressed definitionally, for example by making the three-stage definition the basic one with respect to which security is measured. Lemma 4.1 directly implies that if DE is a SM D-PKE scheme, then: (1) If $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}^{\mathrm{sm}} \cap \mathcal{A}^{\mathrm{u}} \cap \mathcal{A}_3]$ then $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}^{\mathrm{rom}} \cap \mathcal{A}^{\mathrm{u}} \cap \mathcal{A}_3]$ and (2) If $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}^{\mathrm{sm}} \cap \mathcal{A}^{\mathrm{nu}} \cap \mathcal{A}_3]$ then $\mathsf{DE} \in \mathrm{IND}[\mathcal{A}^{\mathrm{rom}} \cap \mathcal{A}^{\mathrm{nu}} \cap \mathcal{A}_3]$. That is, for 3-stage adversaries, SM security implies ROM security both in the uniform and non-uniform settings. However the question of whether the implication holds for two-stage adversaries and the current definitions would still be interesting.

5 Is SOA security achievable?

We initiate an investigation of SOA security for D-PKE. We provide definitions and then show that the goal is impossible to achieve in the SM, meaning no SM D-PKE scheme achieves it.

What makes this interesting is that the difficulty of achieving SOA security in the R-PKE case arises from the fact that an attacker obtains not only messages but the coins underlying the opened ciphertexts. If it only obtained messages, security is easy to achieve [16]. Since in D-PKE there are no coins, one might think security would be also easy to achieve. But in fact this is not true.

Preliminaries. We let $\perp_n$ denote the vector of length n all of whose entries are $\perp$. For a set $I \subseteq [|\mathbf{x}|]$ we let $\mathbf{x}[I]$ denote the $|\mathbf{x}|$-vector whose i-th component is $\mathbf{x}[i]$ if $i \in I$ and $\perp$ otherwise.

Collision resistance of a function family HF is defined via game $\mathrm{CR}^{\mathsf{X}}_{\mathsf{HF}}(\lambda)$ associated to HF, adversary X and $\lambda \in \mathbb{N}$. The game starts by picking $k \xleftarrow{\$} \{0,1\}^{\mathsf{HF.kl}(\lambda)}$. Then X is run with inputs $1^\lambda, k$ to return $x_0, x_1 \in \{0,1\}^{\mathsf{HF.il}(\lambda)}$. The game returns true if $x_0 \neq x_1$ and $\mathsf{HF}(1^\lambda, k, x_0) = \mathsf{HF}(1^\lambda, k, x_1)$, and false otherwise. The advantage of X is defined as $\mathbf{Adv}^{\mathsf{cr}}_{\mathsf{HF},\mathsf{X}}(\lambda) = \Pr[\mathrm{CR}^{\mathsf{X}}_{\mathsf{HF}}(\lambda)]$ and we say that HF is collision resistant if $\mathbf{Adv}^{\mathsf{cr}}_{\mathsf{HF},\mathsf{X}}(\cdot)$ is negligible for all PT X.

We will use the embedding subroutine described below in the descriptions of some algorithms. Here, Emb receives as input a subset $I \subseteq [n]$, a $|I|$-vector $\mathbf{x}^*$ and a n-vector $\overline{\mathbf{x}}$ and returns the n-vector formed by embedding $\mathbf{x}^*$ in $\overline{\mathbf{x}}$ in the positions specified by I:

```
$\underline{\mathrm{Emb}(I, \mathbf{x}^*, \overline{\mathbf{x}})}$
$j \leftarrow 0$ ; For $i = 1, \ldots, n$ do
    If $i \in I$ then $j \leftarrow j + 1$ ; $\overline{\mathbf{x}}[i] \leftarrow \mathbf{x}^*[j]$
Return $\overline{\mathbf{x}}$
```

Defining SOA security. Providing a meaningful definition of SOA-security for D-PKE takes some care. A definition based on semantic security for relations, as given for R-PKE in [11, 8], is trivially unachievable for D-PKE because a ciphertext is already partial information about a plaintext. Thus we consider semantic security for functions, where the adversary, given ciphertexsts, aims to figure out a function of the message, this function not being given the public key and thus unable to encrypt.

```
MAIN $\mathrm{REAL}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$
$k \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$
$\mathbf{m} \xleftarrow{\$} \mathsf{A.msg}(1^\lambda)$
$\pi \xleftarrow{\$} \mathsf{DE.Pg}(1^\lambda)$
$(pk, sk) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$
$\mathbf{c} \xleftarrow{\$} \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m})$
$\mathsf{st} \xleftarrow{\$} \mathsf{A.cor}^{\textsc{Cor}}(1^\lambda, \pi, pk, k, \mathbf{c})$
$w \xleftarrow{\$} \mathsf{A.g}(1^\lambda, k, I, \mathsf{st}, \mathbf{m}[I])$
Return $(w = \mathsf{A.f}(1^\lambda, \mathbf{m}))$

$\textsc{Cor}(I)$
Return $\mathbf{m}[I]$
```

```
MAIN $\mathrm{IDEAL}^{\mathsf{A,S}}_{\mathsf{DE}}(\lambda)$
$k \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$
$\mathbf{m} \xleftarrow{\$} \mathsf{A.msg}(1^\lambda)$
$\mathsf{st} \xleftarrow{\$} \mathsf{S}^{\textsc{Cor}}(1^\lambda, k)$
$w \xleftarrow{\$} \mathsf{A.g}(1^\lambda, k, I, \mathsf{st}, \mathbf{m}[I])$
Return $(w = \mathsf{A.f}(1^\lambda, \mathbf{m}))$

$\textsc{Cor}(I)$
Return $\mathbf{m}[I]$
```

```
MAIN $\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)$
$(I, \mathsf{st}) \xleftarrow{\$} \mathsf{P}(1^\lambda)$
$\mathbf{m} \xleftarrow{\$} \mathsf{A.msg}(1^\lambda)$
$m \xleftarrow{\$} \mathsf{P}(\mathsf{st}, \mathbf{m}[I])$
Return $(\exists\, i \notin I \,:\, \mathbf{m}[i] = m)$
```

```
MAIN $\mathrm{CCR}^{\mathsf{C}}_{\mathsf{DE},z}(\lambda)$
$(\pi, pk) \xleftarrow{\$} \mathsf{C}(1^\lambda)$
If (not $\mathsf{DE.Vf}(1^\lambda, \pi, pk)$) then return false
$\mathbf{m}_0, \mathbf{m}_1 \xleftarrow{\$} (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$
$\mathbf{c}_0 \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}_0)$
$\mathbf{c}_1 \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}_1)$
For $i = 1, \ldots, z(\lambda)$ do
  If $((\mathbf{c}_0[i] = \mathbf{c}_1[i])$ and $(\mathbf{m}_0[i] \neq \mathbf{m}_1[i]))$
    then return true
Return false
```

Figure 3: The REAL, IDEAL, PRED, and CCR games.

Additionally we must continue to require that messages do not depend on the public key and are unpredictable. Our definition is simulation-based and combines ideas from the basic (non-SOA) definitions of secure D-PKE [5, 9] with ideas from the definitions of SOA-security for R-PKE from [11, 8].

In Figure 3 is the "real" game $\mathrm{REAL}^{\mathsf{A}}_{\mathsf{DE}}$ associated to D-PKE scheme DE and adversary A. PT common state generation algorithm $\mathsf{A.cs}$ is executed on input 1^λ to get a common state k that will be passed to the $\mathsf{A.cor}$ stage of A. (Other stages can get it too, but since our results are negative, not giving it only makes the results stronger.) Then PT message generator $\mathsf{A.msg}$ is executed on input 1^λ to get a $\mathsf{A.nm}(\lambda)$-vector of messages over $\{0,1\}^{\mathsf{DE.ml}(\lambda)}$, where $\mathsf{A.nm}$ is the number-of-messages function associated to A. Then public parameters and keys are generated. (It is important that the messages do not depend on the public parameters or public key of DE for the same reason as with PRIV [5] and IND [9], namely that otherwise security is trivially unachievable.) Then the vector of messages is encrypted, component-wise, to get a vector $\mathbf{c}$ of ciphertexts. The PT corruption algorithm $\mathsf{A.cor}$ gets $1^\lambda, \pi, pk, k, \mathbf{c}$ and an oracle $\textsc{Cor}$ to which it is allowed exactly one query, this consisting of a subset I of $[\mathsf{A.nm}(\lambda)]$, indicating positions at which it wants $\mathbf{m}$ opened. In response it gets $\mathbf{m}[I]$, meaning the values $\mathbf{m}[i]$ for $i \in I$, and returns state information st. The PT guessing algorithm $\mathsf{A.g}$ gets $1^\lambda, k, I, \mathsf{st}, \mathbf{m}[I]$, where I is the $\textsc{Cor}$-query previously made by $\mathsf{A.cor}$ and recorded by the game, and outputs a guess w as to the value of $\mathsf{A.f}(1^\lambda, \mathbf{m})$. Here deterministic PT algorithm $\mathsf{A.f}$, called the information function, represents the information about $\mathbf{m}$ that the adversary is trying to compute. The game returns true iff the guess is correct.

The "ideal" game $\mathrm{IDEAL}^{\mathsf{A,S}}_{\mathsf{DE}}$ of Figure 3 is associated to DE, adversary A and a simulator S. Here, the common state and message vector are chosen as before, but the game neither chooses parameters and public key, nor generates any ciphertexts. The simulator is given no information about $\mathbf{m}$, but has access to oracle $\textsc{Cor}$, to which it is allowed exactly one query, this consisting of a subset I of $[\mathsf{A.nm}(\lambda)]$. In response S gets $\mathbf{m}[I]$ and must then return state information st that should resemble the output of $\mathsf{A.cor}$. The rest is as in the real game.

We need to restrict $\mathsf{A.msg}$ to reflect the inherent weaknesses of D-PKE, analogous to the restrictions made in defining PRIV and IND. Namely we require a message-distinctness condition and a message unpredictability (high min-entropy) condition. Before detailing definitions, we note that the $\mathsf{A.msg}$ in Theorem 5.2 simply outputs uniform, independently distributed messages of super-logarithmic length, so both the conditions will be trivially met, and thus a reader can skip the rest of this paragraph if they wish. Proceeding, since ciphertext equality leaks plaintext equality in D-PKE, we require the following message-distinctness condition: there is a negligible function ν such that $\Pr[\exists\, i, j \,:\, (i \neq j) \wedge (\mathbf{m}[i] = \mathbf{m}[j])] \leq \nu(\lambda)$ where the probability is over $\mathbf{m} \overset{\$}{\leftarrow} \mathsf{A.msg}(1^\lambda)$. Second, we require that A is unpredictable, which we define to mean that $\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A},\mathsf{P}}(\lambda) = \Pr[\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)]$ is negligible for all P (we emphasize that here P is not restricted to be PT), where game $\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}$ is shown on the middle, bottom of Figure 3. The unpredictability condition we define here is very strong, requiring that each component message of $\mathbf{m}$ has high min-entropy even given the others, but this only strengthens our results since they are negative. We let $\mathcal{A}^{\mathrm{soa}}$ denote the class of all PT A that satisfy the message distinctness and unpredictability conditions.

We define the soa-advantage of an adversary A with respect to DE and a simulator S via

$$\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{DE},\mathsf{A},\mathsf{S}}(\lambda) = \Pr\left[\mathrm{REAL}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)\right] - \Pr\left[\mathrm{IDEAL}^{\mathsf{A},\mathsf{S}}_{\mathsf{DE}}(\lambda)\right]$$

for all $\lambda \in \mathbb{N}$. We say that DE is SOA-secure if for all $\mathsf{A} \in \mathcal{A}^{\mathrm{soa}}$, there exists a PT simulator S such that $\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{DE},\mathsf{A},\mathsf{S}}(\cdot)$ is negligible.

The definitions and results here are all in the standard model. Our impossibility result does not rule out achieving an appropriate (programmable) ROM version of our definition of SOA-security for D-PKE. In

Discussion. In the definition of SOA-security for R-PKE from [11, 8], the game returns the boolean output of a relation $\mathsf{A.rel}(1^\lambda, \mathbf{m}, k, I, w)$. Such a definition is not achievable for D-PKE even in the absence of a SOA, meaning even if $\mathsf{A.cor}$ sets $I = \emptyset$. This follows from Lemma 5.1. We have $\mathsf{A.g}$ set $w = (\pi, pk, \mathbf{c})$. Now have $\mathsf{A.rel}$ return true only if $I = \emptyset$, $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}) = \mathbf{c}$ and $\mathsf{DE.Vf}(1^\lambda, \pi, pk) = \mathsf{true}$. So the real game returns true with probability one. But the simulator will be unable to produce $w = (\pi, pk, \mathbf{c})$ passing this test by Lemma 5.1. Accordingly, following the definition of simulation-based semantic security for D-PKE from [9], we have restricted attention to relations $\mathsf{A.rel}$ that check $w = \mathsf{A.f}(1^\lambda, \mathbf{m})$ for some deterministic function $\mathsf{A.f}$. Since $\mathsf{A.f}$ does not take input w there is no way to pass it parameters and public key and thus it cannot encrypt. The issue here is that in D-PKE, a ciphertext constitutes partial information about a plaintext [5], and thus security should only ask for non-leakage of partial information that does not depend on the public key. This is captured by function-based semantic security but not by relation-based semantic security. We note that the impossibility result we have just discussed represents an interesting application of Lemma 5.1 even outside the context of SOA.

In our definition, the simulator creates st that is then passed to $\mathsf{A.g}$ who creates the guess w as to the value of $\mathsf{A.f}(1^\lambda, \mathbf{m})$. Why not ask the simulator to directly create the guess w? This gives the simulator too much power. Indeed, it can always create the correct w, by corrupting the set $I = [\mathsf{A.nm}(\lambda)]$ of all instances, learning $\mathbf{m}$ and setting $w = \mathsf{A.f}(1^\lambda, \mathbf{m})$. Some check is needed to ensure that the I selected by the simulator is distributed similarly to the one selected by $\mathsf{A.cor}$. In the R-PKE case, this is handled by I being an input to $\mathsf{A.rel}$, but we cannot give I to $\mathsf{A.f}$ since I could encode π, pk. In our formulation, $\mathsf{A.g}$ "filters" the output st of the simulator and may check that I satisfies certain conditions. This may seem restrictive, but it is not clear what is the alternative, and in any case positive results, when possible [11, 16], have always created simulators that do run the adversary to obtain the final guess, and indeed it is difficult to imagine any other way for a simulator to operate. So a negative result under our definition remains strong and meaningful.

As a sanity check we note that in the absence of an SOA, our definition collapses to existing ones. More precisely let $\mathcal{A}_0^{\mathrm{soa}}$ denote the class of adversaries $\mathsf{A} \in \mathcal{A}^{\mathrm{soa}}$ whose COR query is always the empty set. Then a D-PKE scheme DE is secure with respect to adversaries in $\mathcal{A}_0^{\mathrm{soa}}$ under our soa definition if and only if it is IND secure. This is because without corruptions our definition collapses to the sss (simulation-based semantic security) definition of [9].

APPROACH. BDWY [8] show that if CR hash functions exist then any R-PKE scheme satisfying a certain binding property they define is not SOA-secure. Roughly, binding says that encryption remains injective even on dishonestly-chosen public keys. Not all R-PKE schemes satisfy this binding property, but many common ones do, and the BDWY result shows in particular that IND-CPA does not imply SOA for R-PKE. In the D-PKE case, rather than ask for schemes that are binding, we introduce a verification algorithm that, given a dishonestly-generated public key, tests the extent to which the encryption induced by this key is an injective function. If it is far from injective, verification will catch it, and otherwise we have some sort of binding. We then show that such a verification algorithm exists for *every* D-PKE scheme. Adapting the technique of BDWY we can then use this to show that no D-PKE scheme is SOA-secure.

INJECTIVITY VERIFICATION. Let DE be a D-PKE scheme. A *verification algorithm* $\mathsf{DE.Vf}$ for DE is a PT algorithm that takes as input $1^\lambda, \pi, pk$ and returns a boolean value. Here, π and pk play the role of parameters and a public key but are to be thought of as adversarially chosen and not necessarily ones that would actually arise in honest parameter and key generation. Informally, $\mathsf{DE.Vf}$ checks if the provided π, pk induce an almost injective function on valid DE messages. We impose a requirement we call completeness, which says that for all $\lambda \in \mathbb{N}$, all $\pi \in [\mathsf{DE.Pg}(1^\lambda)]$ and all $(pk, sk) \in [\mathsf{DE.Kg}(1^\lambda, \pi)]$ we have $\mathsf{DE.Vf}(1^\lambda, \pi, pk) = \mathsf{true}$. That is, if the parameters and key are honestly chosen then the verifier accepts. To formalize the requirement for adversarially chosen π, pk, consider the game described in Figure 3, and define the ciphertext collision resistance advantage of an adversary C via $\mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) = \Pr\left[\mathrm{CCR}^{\mathsf{C}}_{\mathsf{DE},z}(\lambda)\right]$. Here adversary C picks π, pk, so the encryption function induced by them, unlike that induced by an honestly-generated π, pk, may not be injective. The advantage of the adversary is the probability that it can get some non-injectivity to surface via collisions. The following lemma says that it is possible to design a verification algorithm that makes it hard for any adversary to defeat CCR.

Lemma 5.1 Let DE be a D-PKE scheme and $z\colon \mathbb{N} \to \mathbb{N}$. Define the verification algorithm $\mathsf{DE.Vf}$ as follows:

```
$\underline{\mathsf{DE.Vf}(1^\lambda, \pi, pk)}$
If $(|\pi| \neq \mathsf{DE.pl}(\lambda)$ or $|pk| \neq \mathsf{DE.pkl}(\lambda))$ then return $\mathsf{false}$
For $t = 1, \ldots, z(\lambda)$ do
    $\mathbf{m}'_0[t] \xleftarrow{\$} \{0,1\}^{\mathsf{DE.ml}(\lambda)}$ ; $\mathbf{m}'_1[t] \xleftarrow{\$} \{0,1\}^{\mathsf{DE.ml}(\lambda)}$
    If $((\mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}'_0[t]) = \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}'_1[t])) \wedge (\mathbf{m}'_0[t] \neq \mathbf{m}'_1[t]))$ then return $\mathsf{false}$
Return $\mathsf{true}$
```

Then $\mathsf{DE.Vf}$ is PT and complete. Also for any (not necessarily PT) adversary C we have $\mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) \leq \frac{1}{4}$ for all $\lambda \in \mathbb{N}$.

Proof of Lemma 5.1: For any $\lambda \in \mathbb{N}$, any $\pi \in \{0,1\}^{\mathsf{DE.pl}(\lambda)}$ and any $pk \in \{0,1\}^{\mathsf{DE.pkl}(\lambda)}$ let $\mathbf{CP}_{\mathsf{DE}}(1^\lambda, \pi, pk)$ equal

$$\Pr[\exists t \in [z(\lambda)] \;:\; \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}_0[t]) = \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}_1[t]) \text{ and } \mathbf{m}_0[t] \neq \mathbf{m}_1[t]]$$

where the probability is over $\mathbf{m}_0, \mathbf{m}_1 \xleftarrow{\$} (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$. In game CCR the probability that the test performed using $\mathsf{DE.Vf}$ is passed is $1 - \mathbf{CP}_{\mathsf{DE}}(1^\lambda, \pi, pk)$. If such test is passed, the probability

```
Algorithm $\mathsf{A.cs}(1^\lambda)$
$k \xleftarrow{\$} \{0,1\}^{\mathsf{HF.kl}(\lambda)}$ ; Return $k$

Algorithm $\mathsf{A.msg}(1^\lambda)$
$\mathbf{m} \xleftarrow{\$} (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{n(\lambda)}$ ; Return $\mathbf{m}$

Algorithm $\mathsf{A.cor}^{\textsc{Cor}}(1^\lambda, \pi, pk, k, \mathbf{c})$
$b[1] \ldots b[z(\lambda)] \leftarrow \mathsf{HF}(1^\lambda, k, \mathbf{c}) \| \pi \| pk$
$I \leftarrow \{\, 2j - 1 + b[j] \;:\; 1 \le j \le z(\lambda) \,\}$
$\overline{\mathbf{m}} \leftarrow \textsc{Cor}(I)$
$\mathsf{st} \leftarrow (\pi, pk, \mathbf{c})$
Return $\mathsf{st}$
```

```
Algorithm $\mathsf{A.g}(1^\lambda, k, I, \mathsf{st}, \overline{\mathbf{m}})$
$(\pi, pk, \mathbf{c}) \leftarrow \mathsf{st}$
If $(|\pi| \ne \mathsf{DE.pl}(\lambda)$ or $|pk| \ne \mathsf{DE.pkl}(\lambda))$ then return 0
If $|\mathbf{c}| \ne n(\lambda)$ then return 0
If (not $\mathsf{DE.Vf}(1^\lambda, \pi, pk)$) then return 0
$b[1] \ldots b[z(\lambda)] \leftarrow \mathsf{HF}(1^\lambda, k, \mathbf{c}) \| \pi \| pk$
If $(I \ne \{\, 2j - 1 + b[j] \;:\; 1 \le j \le z(\lambda) \,\})$ then return 0
For all $i \in I$ do
    If $(\mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}[i]) \ne \mathbf{c}[i])$ then return 0
Return 1

Algorithm $\mathsf{A.f}(1^\lambda, \mathbf{m})$
Return 1
```

Figure 4: Adversary A for the proof of Theorem 5.2.

that some ciphertext collision appears (thus making the game CCR return true) is upper bounded by $\mathbf{CP}_{\mathsf{DE}}(1^\lambda, \pi, pk)$. Since passing the verification algorithm's test and having some ciphertext collision is the only combination in which game CCR returns true, for any adversary C, we get

$$\mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) \le \max_{\pi \in \{0,1\}^{\mathsf{DE.pl}(\lambda)}} \; \max_{pk \in \{0,1\}^{\mathsf{DE.pkl}(\lambda)}} \left(\left(1 - \mathbf{CP}_{\mathsf{DE}}(1^\lambda, \pi, pk)\right) \mathbf{CP}_{\mathsf{DE}}(1^\lambda, \pi, pk) \right) \le \frac{1}{4}$$

where the last inequality is from the maximum of the quadratic function. ▌

Impossibility of SOA security. In order to prove that a given D-PKE scheme DE is not SOA-secure we need to prove the existence of an adversary $\mathsf{A} \in \mathcal{A}^{\mathrm{soa}}$ such that for *every* PT simulator S, the function $\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{DE},\mathsf{A},\mathsf{S}}(\cdot)$ is *not* negligible. We assume a collision-resistant hash function HF in the following.

Theorem 5.2 Let DE be a D-PKE scheme such that $2^{-\mathsf{DE.ml}(\cdot)}$ is negligible. Assume the existence of a collision-resistant family of functions. Then, there exists a PT adversary $\mathsf{A} \in \mathcal{A}^{\mathrm{soa}}$ such that, for all PT simulators S there exists a function ν that is not negligible and is such that $\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{DE},\mathsf{A},\mathsf{S}}(\lambda) \ge \nu(\lambda)$ for all $\lambda \in \mathbb{N}$. Furthermore, message sampler A.msg returns a vector of uniformly and independently distributed messages. ▌

The proof follows the template of the proof from [8] but makes crucial use of Lemma 5.1. We use a variant of the reset lemma of [12]. We recall the lemma, in the form stated in [8], in Appendix B.

Proof of Theorem 5.2: Let HF be a collision-resistant family of functions. Let $z(\cdot) = \mathsf{HF.ol}(\cdot) + \mathsf{DE.pkl}(\cdot) + \mathsf{DE.pl}(\cdot)$. Let $n(\cdot) = 2z(\cdot)$. Let A be the adversary defined in Figure 4. We should emphasize that the hash function here is not being applied element-wise, but to the ciphertext vector as a whole. Here, DE.Vf is the verification algorithm provided by Lemma 5.1 for DE. We first note that $\mathsf{A} \in \mathcal{A}^{\mathrm{soa}}$. Indeed, A is unpredictable due to the assumption that $2^{-\mathsf{DE.ml}(\cdot)}$ is negligible and the fact that messages in the message vector are independently and uniformly distributed. It also satisfies the distinctness condition since $2^{-\mathsf{DE.ml}(\cdot)}$ is negligible and $n(\cdot)$ is a polynomial. Next we note that

$$\Pr\left[\,\mathrm{REAL}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)\,\right] = 1 \tag{2}$$

for all $\lambda \in \mathbb{N}$. This follows from the description of A and the completeness of the verifier. We will build adversaries X and C such that

$$\Pr\left[\,\mathrm{IDEAL}^{\mathsf{A},\mathsf{S}}_{\mathsf{DE}}(\lambda)\,\right] \le 2^{-\mathsf{DE.ml}(\lambda) z(\lambda)} + \sqrt{\mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) + \mathbf{Adv}^{\mathsf{cr}}_{\mathsf{HF},\mathsf{X}}(\lambda)} \tag{3}$$

for all $\lambda \in \mathbb{N}$. But by the assumption that HF is CR and by Lemma 5.1, we have that the above probability is not negligibly close to 1 and hence

$$\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{DE},\mathsf{A},\mathsf{S}}(\cdot) = 1 - \Pr\left[\,\mathrm{IDEAL}^{\mathsf{A},\mathsf{S}}_{\mathsf{DE}}(\cdot)\,\right] \tag{4}$$

is a function that is not negligible.

It may seem strange that security fails for $\mathsf{A.f}$ that always returns 1, because this function does not leak anything about $\mathbf{m}$. What we are saying is that it is not possible to prove even this simple, intuitive claim, meaning to give a simulator for an adversary relative to this simple information function.

We proceed to prove Equation (3). Given any S, we divide it in two parts, S_1 and S_2. S_1 is the execution until the point at which the subset that will be corrupted is chosen, and S_2 is the rest of the execution. We assume without loss of generality that S_1 forwards the coins to S_2, so S_2 is deterministic. This means we can view S as operating as follows:

```
Simulator $\mathsf{S}^{\textsc{Cor}}(1^\lambda, k)$
$(\mathsf{st}^*, I) \xleftarrow{\$} \mathsf{S}_1(1^\lambda, k)$ ; $\overline{\mathbf{m}} \leftarrow \textsc{Cor}(I)$ ; $\mathsf{st} \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}})$ ; Return $\mathsf{st}$
```

We now provide some intuition about why we expect the simulator to fail. We consider an experiment where we run $\mathsf{A.cs}(1^\lambda)$ to get k, run $\mathsf{S}_1(1^\lambda, k)$ to get (st^*, I), pick two, random vectors $\overline{\mathbf{m}}_0, \overline{\mathbf{m}}_1$ that are $\perp$ on positions not in I, and then run S_2 twice, getting $\mathsf{st}_0 \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}}_0)$ and $\mathsf{st}_1 \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}}_1)$. Parse st_b as $(\pi_b, pk_b, \mathbf{c}_b)$ for $b = 0, 1$. If $\mathsf{st}_0 \neq \mathsf{st}_1$ then, because I is the same in both cases, we have $(\pi_0, pk_0) = (\pi_1, pk_1)$ and thus $\mathbf{c}_0 \neq \mathbf{c}_1$, leading to a collision for $\mathsf{HF}(1^\lambda, k, \cdot)$. So assume $\mathsf{st}_0 = \mathsf{st}_1 = (\pi, pk, \mathbf{c})$. If both runs make the game return true then by definition of $\mathsf{A.g}$ we have $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}_0[I]) = \mathbf{c}[I]$ and $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}_1[I]) = \mathbf{c}[I]$. This is highly unlikely if the function $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \cdot)$ is injective. So the only way the simulator can hope to succeed is pick π, pk so that this function is highly non-injective. But $\mathsf{A.g}$ is running the verifier so if the simulator tries this, $\mathsf{A.g}$ is likely to return 0 by Lemma 5.1. We proceed to formalize the above intuition and establish Equation (3) via the reset lemma (Lemma B.1).

Define P_1, P_2 as follows, where $\mathbf{m}^* \in (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$:

```
Algorithm $P_1(1^\lambda)$
$k \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$ ; $(\mathsf{st}^*, I) \xleftarrow{\$} \mathsf{S}_1(1^\lambda, k)$
$\overline{\mathsf{st}} \leftarrow (1^\lambda, k, I, \mathsf{st}^*)$ ; Return $\overline{\mathsf{st}}$
```

```
Algorithm $P_2(\overline{\mathsf{st}}, \mathbf{m}^*)$
$(1^\lambda, k, I, \mathsf{st}^*) \leftarrow \overline{\mathsf{st}}$ ; $\overline{\mathbf{m}} \leftarrow \mathrm{Emb}(I, \mathbf{m}^*, \perp_{n(\lambda)})$
$\mathsf{st} \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}})$ ; $w \xleftarrow{\$} \mathsf{A.g}(1^\lambda, k, I, \mathsf{st}, \overline{\mathbf{m}})$
Return $(w = 1)$
```

Letting $V_\lambda = (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$, and using Lemma B.1 we get

$$\Pr\left[\,\mathrm{IDEAL}^{\mathsf{A},\mathsf{S}}_{\mathsf{DE}}(\lambda)\,\right] \;=\; \mathbf{AP}_1(P_1, P_2, V, \lambda) \;\leq\; 2^{-\mathsf{DE.ml}(\lambda) z(\lambda)} + \sqrt{\mathbf{AP}_2(P_1, P_2, V, \lambda)}.$$

Consider CR adversary X and CCR adversary C defined as follows:

```
Adversary $\mathsf{X}(1^\lambda, k)$
$(\mathsf{st}^*, I) \xleftarrow{\$} \mathsf{S}_1(1^\lambda, k)$
For $b = 0, 1$ do
    $\mathbf{m}^*_b \xleftarrow{\$} (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$
    $\overline{\mathbf{m}}_b \leftarrow \mathrm{Emb}(I, \mathbf{m}^*_b, \perp_{n(\lambda)})$
    $\mathsf{st}_b \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}}_b)$
    $(\pi_b, pk_b, \mathbf{c}_b) \leftarrow \mathsf{st}_b$
Return $(\mathbf{c}_0, \mathbf{c}_1)$
```

```
Adversary $\mathsf{C}(1^\lambda)$
$k \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$ ; $(\mathsf{st}^*, I) \xleftarrow{\$} \mathsf{S}_1(1^\lambda, k)$
For $j = \mathsf{HF.ol}(\lambda) + 1, \ldots, \mathsf{HF.ol}(\lambda) + \mathsf{DE.pl}(\lambda)$ do
    $i \leftarrow j - \mathsf{HF.ol}(\lambda)$
    If $(2j - 1 \in I)$ then $\pi[i] \leftarrow 0$ else $\pi[i] \leftarrow 1$
For $j = \mathsf{HF.ol}(\lambda) + \mathsf{DE.pl}(\lambda) + 1, \ldots, z(\lambda)$ do
    $i \leftarrow j - \mathsf{HF.ol}(\lambda) - \mathsf{DE.pl}(\lambda)$
    If $(2j - 1 \in I)$ then $pk[i] \leftarrow 0$ else $pk[i] \leftarrow 1$
Return $(\pi, pk)$
```

Here C reconstructs π, pk bit-by-bit from I and returns them. To conclude the proof we claim that

$$\mathbf{AP}_2(P_1, P_2, V, \lambda) \leq \mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) + \mathbf{Adv}^{\mathsf{cr}}_{\mathsf{HF},\mathsf{X}}(\lambda) \; . \tag{5}$$

To justify this, let us write out the double-execution experiment referred to in Lemma B.1:

```
$k \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$ ; $(\mathsf{st}^*, I) \xleftarrow{\$} \mathsf{S}_1(1^\lambda, k)$
For $b = 0, 1$ do
    $\mathbf{m}^*_b \xleftarrow{\$} (\{0,1\}^{\mathsf{DE.ml}(\lambda)})^{z(\lambda)}$ ; $\overline{\mathbf{m}}_b \leftarrow \mathrm{Emb}(I, \mathbf{m}^*_b, \bot_{n(\lambda)})$
    $\mathsf{st}_b \leftarrow \mathsf{S}_2(1^\lambda, \mathsf{st}^*, \overline{\mathbf{m}}_b)$ ; $(\pi_b, pk_b, \mathbf{c}_b) \leftarrow \mathsf{st}_b$ ; $w_b \xleftarrow{\$} \mathsf{A.g}(1^\lambda, k, I, \mathsf{st}_b, \overline{\mathbf{m}}_b)$
```

In this experiment, let E be the event that $\mathsf{st}_0 = \mathsf{st}_1$ and S the event that $w_0 = w_1 = 1$. Then we claim that

$$\Pr[S \wedge \neg E] \leq \mathbf{Adv}^{\mathsf{cr}}_{\mathsf{HF},\mathsf{X}}(\lambda) \quad \text{and} \quad \Pr[S \wedge E \wedge (\mathbf{m}^*_0 \neq \mathbf{m}^*_1)] \leq \mathbf{Adv}^{\mathsf{ccr}}_{\mathsf{DE},z,\mathsf{C}}(\lambda) \; , \tag{6}$$

from which Equation (5) follows. We now justify the two inequalities in Equation (6). Suppose S is true. Since the set I is the same in both executions, it must be that $(\pi_0, pk_0) = (\pi_1, pk_1)$. So $\mathsf{st}_0 \neq \mathsf{st}_1$ implies $\mathbf{c}_0 \neq \mathbf{c}_1$. But then X finds a collision. On the other hand, suppose S is true and $\mathsf{st}_0 = \mathsf{st}_1$, and let $(\pi, pk, \mathbf{c})$ denote this common value. Then by definition of $\mathsf{A.g}$ we have $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}_b[I]) = \mathbf{c}[I]$ for both $b = 0$ and $b = 1$, and thus $\mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}_0[I]) = \mathsf{DE.Enc}(1^\lambda, \pi, pk, \overline{\mathbf{m}}_1[I])$. But $\mathbf{m}^*_0 \neq \mathbf{m}^*_1$ so there is some $i \in I$ such that $\overline{\mathbf{m}}_0[i] \neq \overline{\mathbf{m}}_1[i]$. This means that C wins the CCR game. ▌

Indistinguishability-based SOA. Theorem 5.2 rules out SOA-secure D-PKE under a simulation-style definition. A natural question is whether SOA-secure D-PKE may be achieved under a weaker definition, in particular an indistinguishability style one. Indeed, for R-PKE, SOA-security definitions in both styles have been made and investigated, and the indistinguishability style is easier to achieve [11, 16, 17, 30]. The difficulty is that for D-PKE it is not clear how to give a meaningful indistinguishability style definition of SOA-security. For R-PKE, the indistinguishability definition involves conditional re-sampling of the un-opened messages. In the D-PKE case we cannot provide the un-opened messages in the distinguishing test, since the adversary could easily win by re-encrypting to check versus the ciphertexts. It is not clear to us what could be done instead. Additionally, even for R-PKE, re-sampling is rarely polynomial time so either we consider security for a very limited set of distributions or we have a non-polynomial time game, and both choices have problems. Defining some achievable notion of SOA-secure D-PKE is an interesting open problem.

6 Does SU security imply MU security?

We now define mIND, the multi-key version of IND security, and show a separation between the two notions by showing the existence of a D-PKE scheme that is IND-secure but not mIND-secure.

mIND security. Let DE be a D-PKE scheme. An mIND adversary A specifies a common-state generation algorithm $\mathsf{A.cs}$, a message-generation algorithm $\mathsf{A.msg}$ and a guessing algorithm $\mathsf{A.g}$, all PT. On input 1^λ, algorithm $\mathsf{A.cs}$ generates state information st that will be passed to both $\mathsf{A.msg}$ and $\mathsf{A.g}$. Algorithm $\mathsf{A.msg}$, on input $1^\lambda, \mathsf{st}$ returns a pair $(\mathbf{m}_0, \mathbf{m}_1)$ of $\mathsf{A.nu}(\lambda)$ by $\mathsf{A.nm}(\lambda)$ matrices over $\{0,1\}^{\mathsf{DE.ml}(\lambda)}$, where $\mathsf{A.nu}$ is the number-of-users function associated to A and $\mathsf{A.nm}$ is the number-of-messages function associated to A. It is required that for each b, i the strings $\mathbf{m}_b[i,1], \ldots, \mathbf{m}_b[i, \mathsf{A.nm}(\lambda)]$, which are the messages encrypted under the public key $\mathbf{pk}[i]$ of user i, be distinct. (However, messages may repeat across columns, meaning the same message may be encrypted under different public keys.)

The $\mathrm{mIND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$ game associated with DE and adversary A is described on the left of Figure 5. We define the advantage of A via $\mathbf{Adv}^{\mathsf{mind}}_{\mathsf{DE},\mathsf{A}}(\lambda) = 2 \cdot \Pr[\mathrm{mIND}^{\mathsf{A}}_{\mathsf{DE}}(1^\lambda)] - 1$ for all $\lambda \in \mathbb{N}$. We let

```
Main $\mathrm{mIND}^{\mathsf{A}}_{\mathsf{DE}}(\lambda)$
$\mathsf{st} \xleftarrow{\$} \mathsf{A.cs}(1^\lambda)$ ; $(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}(1^\lambda, \mathsf{st})$
$\pi \xleftarrow{\$} \mathsf{DE.Pg}(1^\lambda)$ ; $b \xleftarrow{\$} \{0,1\}$
For $i = 1$ to $\mathsf{A.nu}(\lambda)$ do
  $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$
  For $j = 1$ to $\mathsf{A.nm}(\lambda)$ do
    $\mathbf{c}[i,j] \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, \mathbf{pk}[i], \mathbf{m}_b[i,j])$
$b' \xleftarrow{\$} \mathsf{A.g}(1^\lambda, \pi, \mathbf{pk}, \mathsf{st}, \mathbf{c})$
Return $(b = b')$
```

```
Main $\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)$
$(\mathsf{st}, m) \xleftarrow{\$} \mathsf{P}(1^\lambda)$
$(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}(1^\lambda, \mathsf{st})$
Return $(\exists\, i, j, b : \mathbf{m}_b[i,j] = m)$
```

Figure 5: The mIND game used to define multi-user security of D-PKE scheme DE and the PRED game used to define unpredictability of adversary A.

```
$\overline{\mathsf{DE}}.\mathsf{Pg}(1^\lambda)$
$\pi \xleftarrow{\$} \mathsf{DE.Pg}(1^\lambda)$ ; $(pk^*, sk^*) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$
Return $(\pi, pk^*)$

$\overline{\mathsf{DE}}.\mathsf{Kg}(1^\lambda, (\pi, pk^*))$
$(pk, sk) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$ ; Return $(pk, sk)$
```

```
$\overline{\mathsf{DE}}.\mathsf{Enc}(1^\lambda, (\pi, pk^*), pk, m)$
$c \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, pk, m)$ ; $c^* \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, pk^*, m)$
Return $(c, c^*)$

$\overline{\mathsf{DE}}.\mathsf{Dec}(1^\lambda, (\pi, pk^*), sk, (c, c^*))$
$m \leftarrow \mathsf{DE.Dec}(1^\lambda, \pi, sk, c)$ ; Return $m$
```

Figure 6: D-PKE scheme $\overline{\mathsf{DE}}$ constructed from D-PKE scheme DE.

$\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A,P}}(\lambda) = \Pr[\mathrm{PRED}^{\mathsf{P}}_{\mathsf{A}}(\lambda)]$ for all $\lambda \in \mathbb{N}$, where game PRED is in the middle in Figure 5. We say that A is unpredictable if $\mathbf{Adv}^{\mathsf{pred}}_{\mathsf{A,P}}(\cdot)$ is negligible for all P. If $\mathcal{A}$ is a class (set) of adversaries then we say that DE is mIND[$\mathcal{A}$]-secure if $\mathbf{Adv}^{\mathsf{mind}}_{\mathsf{DE,A}}(\cdot)$ is negligible for all $\mathsf{A} \in \mathcal{A}$. It is convenient to view mIND[$\mathcal{A}$] as a set, so that $\mathsf{DE} \in$ mIND[$\mathcal{A}$] iff DE is mIND[$\mathcal{A}$]-secure. If A.cs always returns ε then we say that A has trivial initial state and we may refer to A as a two-stage adversary. Let $\mathcal{A}^{\mathrm{m}}_2$ be the class of all PT, 2-stage unpredictable uniform adversaries, and for any polynomial $n\colon \mathbb{N} \to \mathbb{N}$ let $\mathcal{A}^{\mathrm{m}}_{2,n}$ be the class of all $\mathsf{A} \in \mathcal{A}^{\mathrm{m}}_2$ for which $\mathsf{A.nu} = n$. Then security for n users is captured by mIND[$\mathcal{A}^{\mathrm{m}}_{2,n}$] and security for any number of users is captured by mIND[$\mathcal{A}^{\mathrm{m}}_2$].

In the case of IND we had four variants, depending on whether adversaries were uniform or non-uniform and whether we were in the SM or the ROM. For simplicity, we address mIND in the uniform, SM case. The separation extends to the other three cases. Thus, below, the understanding is that $\mathrm{IND}, \mathrm{mIND}_n, \mathrm{mIND}$ refer, respectively, to $\mathrm{IND}[\mathcal{A}^{\mathrm{sm-u}}_2], \mathrm{mIND}[\mathcal{A}^{\mathrm{m}}_{2,n}]$ and $\mathrm{mIND}[\mathcal{A}^{\mathrm{m}}_2]$.

Separation result. Our separation is based on the minimal assumption that some IND-secure D-PKE scheme exists, and is established by a somewhat curious case analysis.

Theorem 6.1 Assume there exists an IND-secure D-PKE scheme. Then there exists a D-PKE scheme that is (1) IND-secure but (2) not mIND_2-secure.

Proof of Theorem 6.1: We establish the theorem by considering two cases. *Case 1:* There does not exist a D-PKE scheme that is mIND_2-secure. The assumption in the theorem statement says there exists a D-PKE scheme DE that is IND-secure. But the assumption made for Case 1 says that no D-PKE scheme is mIND_2-secure. So in particular DE is not mIND_2-secure. This establishes the theorem trivially in this case. *Case 2:* There exists a D-PKE scheme that is mIND_2-secure. Let DE be a D-PKE scheme that is mIND_2-secure. We construct from it a D-PKE scheme $\overline{\mathsf{DE}}$ that is (1) IND-secure but (2) not mIND_2-secure. This establishes the theorem in Case 2. Since either Case 1 or Case 2 must be true, we have established the theorem overall.

The D-PKE scheme $\overline{\mathsf{DE}}$ is shown in Figure 6. The parameters of the new scheme include a public key pk^* for the old scheme. The new encryption of a message m under public key pk consists of two encryptions of m under the old scheme, one with pk and the other with pk^*. Intuitively, (2) is true because if users 1, 2 encrypt messages m_1, m_2 then the second components of their ciphertexts are equal iff $m_1 = m_2$, allowing an adversary to detect whether or not $m_1 = m_2$. On the other hand, (1) is true because pk^* can be viewed as a key of a dummy second user in the old scheme. Encryption in the new scheme is then tantamount to encryption of m under two independent keys of the old scheme, which is secure by the assumed mIND$_2$-security of the old scheme. We now proceed to the details.

We first establish (2), that $\overline{\mathsf{DE}}$ is not mIND$_2$-secure, via the following adversary $\mathsf{A} \in \mathcal{A}^{\mathrm{m}}_{2,2}$. Let $\mathsf{A.cs}(1^\lambda)$ return ε. Let $\mathsf{A.msg}(1^\lambda, \varepsilon)$ return 2 by 1 matrices $(\mathbf{m}_0, \mathbf{m}_1)$ defined via

$$\mathbf{m}_0[1,1], \mathbf{m}_0[2,1], \mathbf{m}_1[1,1] \xleftarrow{\$} \{0,1\}^{\mathsf{DE.ml}(\lambda)} \; ; \; \mathbf{m}_1[2,1] \leftarrow \mathbf{m}_1[1,1] \; .$$

Let $\mathsf{A.g}(1^\lambda, (\pi, pk^*), \mathbf{pk}, \varepsilon, \overline{\mathbf{c}})$ parse $(\mathbf{c}[i,1], \mathbf{c}^*[i,1]) \leftarrow \overline{\mathbf{c}}[i,1]$ for $i = 1, 2$. If $\mathbf{c}^*[1,1] = \mathbf{c}^*[2,1]$ then it returns 1 else it returns 0. Then $\mathbf{Adv}^{\mathsf{mind}}_{\overline{\mathsf{DE}},\mathsf{A}}(\lambda) \geq 1 - 2^{-\mathsf{DE.ml}(\lambda)}$.

To establish (1), that $\overline{\mathsf{DE}}$ is IND-secure, let $\overline{\mathsf{A}} \in \mathcal{A}_2$. We will provide $\mathsf{A} \in \mathcal{A}^{\mathrm{m}}_{2,2}$ such that

$$\mathbf{Adv}^{\mathsf{ind}}_{\overline{\mathsf{DE}},\overline{\mathsf{A}}}(\lambda) \leq \mathbf{Adv}^{\mathsf{mind}}_{\mathsf{DE},\mathsf{A}}(\lambda) \tag{7}$$

for all $\lambda \in \mathbb{N}$. Then (1) follows from the assumption that DE is mIND$_2$-secure. Let $\mathsf{A.cs} = \overline{\mathsf{A}}\mathsf{.cs}$ return ε. Let $\mathsf{A.nm} = \overline{\mathsf{A}}\mathsf{.nm}$. Let $\mathsf{A.nu} = 2$. Define $\mathsf{A.msg}$ and $\mathsf{A.g}$ as follows:

```
$\mathsf{A.msg}(1^\lambda, \varepsilon)$
$(\overline{\mathbf{m}}_0, \overline{\mathbf{m}}_1) \xleftarrow{\$} \overline{\mathsf{A}}\mathsf{.msg}(1^\lambda, \varepsilon)$
For $j = 1, \ldots, \mathsf{A.nm}(\lambda)$ do
  $\mathbf{m}_0[1,j] \leftarrow \overline{\mathbf{m}}_0[j]$ ; $\mathbf{m}_0[2,j] \leftarrow \overline{\mathbf{m}}_0[j]$
  $\mathbf{m}_1[1,j] \leftarrow \overline{\mathbf{m}}_1[j]$ ; $\mathbf{m}_1[2,j] \leftarrow \overline{\mathbf{m}}_1[j]$
Return $(\mathbf{m}_0, \mathbf{m}_1)$
```

```
$\mathsf{A.g}(1^\lambda, \pi, \mathbf{pk}, \varepsilon, \mathbf{c})$
For $j = 1, \ldots, \mathsf{A.nm}(\lambda)$ do
  $\overline{\mathbf{c}}[j] \leftarrow (\mathbf{c}[1,j], \mathbf{c}[2,j])$
$b' \xleftarrow{\$} \overline{\mathsf{A}}\mathsf{.g}(1^\lambda, (\pi, \mathbf{pk}[2]), \mathbf{pk}[1], \varepsilon, \overline{\mathbf{c}})$
Return $b'$
```

Then Equation (7) follows. ▌

We remark that the proof of Theorem 6.1 is non-constructive. It proves the existence of a scheme that is IND-secure but not mIND$_2$-secure but does not put in our hands a concrete, specific example of such a scheme. This is because, although either Case 1 or Case 2 in the proof must be true, we do not know which. We also remark that our proof makes crucial use of the system parameters. Whether or not single and multi-user security are equivalent for D-PKE in the absence of system parameters is an interesting open question.

Acknowledgments

We thank Bjorn Tackmann, Peter Gazi and Adam O'Neill for valuable discussions.

References

[1] O. Baudron, D. Pointcheval, and J. Stern. Extended notions of security for multicast public key cryptosystems. In *Automata, Languages and Programming*, pages 499–511. Springer, 2000. (Cited on page 5.)

[2] D. Beaver. Plug and play encryption. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 75–89. Springer, Aug. 1997. (Cited on page 5.)

[3] D. Beaver and S. Haber. Cryptographic protocols provably secure against dynamic adversaries. In R. A. Rueppel, editor, *EUROCRYPT'92*, volume 658 of *LNCS*, pages 307–323. Springer, May 1992. (Cited on page 5.)

[4] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, May 2000. (Cited on page 5.)

[5] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Aug. 2007. (Cited on page 3, 5, 6, 11, 12.)

[6] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Dec. 2009. (Cited on page 3.)

[7] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Aug. 1998. (Cited on page 3.)

[8] M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard security does not imply security against selective-opening. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 645–662. Springer, Apr. 2012. (Cited on page 5, 10, 11, 12, 13, 14, 22.)

[9] M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 360–378. Springer, Aug. 2008. (Cited on page 3, 6, 7, 11, 12, 13.)

[10] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via uces. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version in CRYPTO 2013. (Cited on page 3.)

[11] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Apr. 2009. (Cited on page 5, 10, 11, 12, 16.)

[12] M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Aug. 2002. (Cited on page 14, 22.)

[13] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. (Cited on page 3.)

[14] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. (Cited on page 6.)

[15] M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 235–252. Springer, Mar. 2011. (Cited on page 5.)

[16] M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101, 2009. `http://eprint.iacr.org/2009/101`. (Cited on page 5, 10, 12, 16.)

[17] F. Böhl, D. Hofheinz, and D. Kraschewski. On definitions of selective opening security. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 522–539. Springer, May 2012. (Cited on page 5, 16.)

[18] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Aug. 2008. (Cited on page 3.)

[19] Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer, Aug. 2011. (Cited on page 3, 5.)

[20] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 90–104. Springer, Aug. 1997. (Cited on page 5.)

[21] R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996. (Cited on page 5.)

[22] R. Canetti, S. Halevi, and J. Katz. Adaptively-secure, non-interactive public-key encryption. In J. Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. Springer, Feb. 2005. (Cited on page 5.)

[23] I. Damgård and J. B. Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 432–450. Springer, Aug. 2000. (Cited on page 5.)

[24] Y. Dodis, P. J. Lee, and D. H. Yum. Optimistic fair exchange in a multi-user setting. In T. Okamoto and X. Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 118–133. Springer, Apr. 2007. (Cited on page 5.)

[25] C. Dwork, M. Naor, O. Reingold, and L. J. Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003. (Cited on page 5.)

[26] S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 381–402. Springer, May 2010. (Cited on page 5.)

[27] B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 582–599. Springer, Mar. 2012. (Cited on page 3.)

[28] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. (Cited on page 3, 5.)

[29] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88. Springer, Dec. 2011. (Cited on page 5.)

[30] D. Hofheinz and A. Rupp. Standard versus selective opening security: Separation and equivalence results. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 591–615. Springer, Feb. 2014. (Cited on page 5, 16.)

[31] G. Kol and M. Naor. Cryptography and game theory: Designing protocols for exchanging information. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 320–339. Springer, Mar. 2008. (Cited on page 5.)

[32] I. Mironov, O. Pandey, O. Reingold, and G. Segev. Incremental deterministic public-key encryption. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 628–644. Springer, Apr. 2012. (Cited on page 3.)

[33] J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, Aug. 2002. (Cited on page 5.)

[34] A. O'Neill. *Stronger security notions for trapdoor functions and applications*. PhD thesis, Georgia Institute of Technology, 2012. (Cited on page 9.)

[35] K. Ouafi and S. Vaudenay. Smashing SQUASH-0. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 300–312. Springer, Apr. 2009. (Cited on page 3.)

[36] S. Panjwani. Tackling adaptive corruptions in multicast encryption protocols. In S. P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 21–40. Springer, Feb. 2007. (Cited on page 5.)

[37] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Aug. 2008. (Cited on page 5.)

[38] A. Raghunathan, G. Segev, and S. P. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 93–110. Springer, May 2013. (Cited on page 3.)

[39] T. Ristenpart, H. Shacham, and T. Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer, May 2011. (Cited on page 3, 4, 6.)

[40] T. Ristenpart and S. Yilek. When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography. In *NDSS 2010*. The Internet Society, Feb. / Mar. 2010. (Cited on page 3.)

[41] D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In R. D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, Jan. 2013. (Cited on page 3, 4.)

A Adaptive q-wise independence and Proof of Equation (1)

The definition of $q(\cdot)$-wise independence is with respect to fixed inputs and outputs, which means that a $q(\cdot)$-wise independent family provides a RO for $q(\cdot)$ non-adaptive queries. Here we provide a lemma showing that a $q(\cdot)$-wise independent family can replace a RO for $q(\cdot)$ adaptive adversary queries. Then we apply the lemma to provide a proof of Equation (1), completing the proof of Lemma 4.1.

We begin with the lemma. Consider the games of Figure 7 associated to a family of functions HF. The adversary W here is computationally unbounded but may query its oracle adaptively $q(\lambda)$ times. The following says that its probability of returning 1 is the same in both games.

Lemma A.1 Let HF be a $q(\cdot)$-wise independent family of functions. Let W be an adversary who makes $q(\cdot)$ oracle queries and outputs a bit. Let $\lambda \in \mathbb{N}$. Then $\Pr[\mathrm{G}_1^{\mathsf{W}}(\lambda)] = \Pr[\mathrm{G}_0^{\mathsf{W}}(\lambda)]$.

Proof: Write $q, \mathsf{kl}, \mathsf{il}, \mathsf{ol}$ for $q(\lambda), \mathsf{HF.kl}(\lambda), \mathsf{HF.il}(\lambda), \mathsf{HF.ol}(\lambda)$, respectively. We may assume wlog that W is deterministic, since we could prove the lemma for each choice of coins of W. We also assume wlog that W's queries are always distinct. For $y_1, \ldots, y_q \in \{0,1\}^{\mathsf{ol}}$ let $W(y_1 \ldots y_q) \in \{0,1\}$ denote the output of W when it receives responses $y_1, \ldots, y_q$ to its oracle queries. Also for $y_1, \ldots, y_q \in \{0,1\}^{\mathsf{ol}}$ and $b \in \{0,1\}$ let $P_b(y_1 \ldots y_q)$ denote the probability that the responses to W's oracle queries in game $\mathrm{G}_b^{\mathsf{W}}(\lambda)$ are $y_1, \ldots, y_q$. Then $P_0(y_1 \ldots y_q) = 2^{-q \cdot \mathsf{ol}}$. We claim that $P_1(y_1 \ldots y_q)$ is also equal to $2^{-q \cdot \mathsf{ol}}$. If so, we have

$$\Pr[\mathrm{G}_1^{\mathsf{W}}(\lambda)] = \sum_{y_1,\ldots,y_q} W(y_1 \ldots y_q) \cdot P_1(y_1 \ldots y_q) = \sum_{y_1,\ldots,y_q} W(y_1 \ldots y_q) \cdot P_0(y_1 \ldots y_q) = \Pr[\mathrm{G}_0^{\mathsf{W}}(\lambda)] \, ,$$

establishing the lemma. We now prove the above claim. Let Q denote the next query function of W. This means that $Q(\varepsilon)$ is the first oracle query of W, and $Q(y_1 \ldots y_{i-1})$ is W's i-th query if it received responses $y_1, \ldots, y_{i-1} \in \{0,1\}^{\mathsf{ol}}$ to its previous $i-1$ queries, for $i = 2, \ldots, q$. Let $\mathsf{KS}(\varepsilon) = \{0,1\}^{\mathsf{kl}}$. For each $i = 1, \ldots, q$ and $y_1, \ldots, y_i \in \{0,1\}^{\mathsf{ol}}$, define

$$\mathsf{KS}(y_1 \ldots y_i) = \{ \, k \in \mathsf{KS}(y_1 \ldots y_{i-1}) \, : \, \mathsf{HF}(1^\lambda, k, Q(y_1 \ldots y_{i-1})) = y_i \, \} \, .$$

The assumption that HF is q-wise independent means that

$$\forall \, y_1, \ldots, y_q \in \{0,1\}^{\mathsf{ol}} \; : \; \frac{|\mathsf{KS}(y_1 \ldots y_q)|}{|\mathsf{KS}(\varepsilon)|} \; = \; 2^{-q \cdot \mathsf{ol}} \, .$$

We will use this below. For $i = 1, \ldots, q$ define Y_i: $\{0,1\}^{\mathsf{kl}} \to \{0,1\}^{\mathsf{ol}}$ by

$$\mathsf{Y}_i(k) = \mathsf{HF}(1^\lambda, k, Q(\mathsf{Y}_1(k) \ldots \mathsf{Y}_{i-1}(k)))$$

for all $k \in \{0,1\}^{\mathsf{kl}}$. This is the response to W's i-th query in game $\mathrm{G}_1^{\mathsf{W}}(\lambda)$ when the chosen key is k.

```
$\underline{\text{MAIN } \mathrm{G}_0^{\mathsf{W}}(\lambda)}$
$w \xleftarrow{\$} \mathsf{W}^{\mathrm{RO}}(1^\lambda)$ ; Return $(w = 1)$

$\underline{\mathrm{RO}(x)}$
If $T[x] = \bot$ then $T[x] \xleftarrow{\$} \{0,1\}^{\mathsf{HF.ol}(\lambda)}$
Return $T[x]$
```

```
$\underline{\text{MAIN } \mathrm{G}_1^{\mathsf{W}}(\lambda)}$
$k \xleftarrow{\$} \{0,1\}^{\mathsf{HF.kl}(\lambda)}$ ; $w \xleftarrow{\$} \mathsf{W}^{\mathrm{RO}}(1^\lambda)$
Return $(w = 1)$

$\underline{\mathrm{RO}(x)}$
Return $\mathsf{HF}(1^\lambda, k, x)$
```

Figure 7: Games for Lemma A.1.

Regard $\mathsf{Y}_1, \ldots, \mathsf{Y}_q$ as random variables over the choice of $k \xleftarrow{\$} \{0,1\}^{\mathsf{kl}}$. Then

$$\begin{aligned}
P_1(y_1 \ldots y_q) &= \prod_{i=1}^{q} \Pr\left[\, \mathsf{Y}_i = y_i \;\mid\; \mathsf{Y}_1 = y_1, \ldots, \mathsf{Y}_{i-1} = y_{i-1} \,\right] \\
&= \prod_{i=1}^{q} \frac{|\mathsf{KS}(y_1 \ldots y_i)|}{|\mathsf{KS}(y_1 \ldots y_{i-1})|} \\
&= \frac{|\mathsf{KS}(y_1 \ldots y_q)|}{|\mathsf{KS}(\varepsilon)|} \;=\; 2^{-q \cdot \mathsf{ol}}
\end{aligned}$$

as claimed. ▌

Proof of Equation (1): We refer to B constructed from A as in the proof of Lemma 4.1. We design an adversary W such that

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{B}}(\lambda) = \Pr[\mathrm{G}_1^{\mathsf{W}}(\lambda)] \quad \text{and} \quad \mathbf{Adv}^{\mathsf{ind}}_{\mathsf{DE},\mathsf{A}}(\lambda) = \Pr[\mathrm{G}_0^{\mathsf{W}}(\lambda)]\ .$$

Then Equation (1) follows from Lemma A.1. Adversary W is as follows:

```
$\underline{\mathsf{W}^{\mathrm{RO}}(1^\lambda)}$
$\mathsf{st_A} \xleftarrow{\$} \mathsf{A.cs}^{\mathrm{RO}}(1^\lambda)$ ; $(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A.msg}^{\mathrm{RO}}(1^\lambda, \mathsf{st_A})$
$\pi \xleftarrow{\$} \mathsf{DE.Pg}(1^\lambda)$ ; $(pk, sk) \xleftarrow{\$} \mathsf{DE.Kg}(1^\lambda, \pi)$ ; $b \xleftarrow{\$} \{0,1\}$
$\mathbf{c} \leftarrow \mathsf{DE.Enc}(1^\lambda, \pi, pk, \mathbf{m}_b)$ ; $b' \xleftarrow{\$} \mathsf{A.g}^{\mathrm{RO}}(1^\lambda, \pi, pk, \mathsf{st_A}, \mathbf{c})$
If $(b = b')$ then return 1 else return 0
```

This concludes the proof of Equation (1). ▌

B Reset Lemma

We recall the reset lemma that appears as Lemma 3 from [8]. It is an adaptation of the original reset lemma of [12].

Lemma B.1 Let $V = \{V_\lambda\}_{\lambda \in \mathbb{N}}$ be a collection of non-empty sets. Let P_1, P_2 be algorithms, the second with boolean output. Define the *single-execution acceptance probability* $\mathbf{AP}_1(P_1, P_2, V, \lambda)$ as $\Pr[d = \mathsf{true}]$ in the *single execution experiment* $\overline{\mathsf{st}} \xleftarrow{\$} P_1(1^\lambda)$; $\mathbf{m}^* \xleftarrow{\$} V_\lambda$; $d \xleftarrow{\$} P_2(\overline{\mathsf{st}}, \mathbf{m}^*)$. Define the *double-execution acceptance probability* $\mathbf{AP}_2(P_1, P_2, V, \lambda)$ as $\Pr[d_0 = d_1 = \mathsf{true} \wedge \mathbf{m}_0^* \neq \mathbf{m}_1^*]$ in the *double execution experiment* $\overline{\mathsf{st}} \xleftarrow{\$} P_1(1^\lambda)$; $\mathbf{m}_0^*, \mathbf{m}_1^* \xleftarrow{\$} V_\lambda$; $d_0 \xleftarrow{\$} P_2(\overline{\mathsf{st}}, \mathbf{m}_0^*)$; $d_1 \xleftarrow{\$} P_2(\overline{\mathsf{st}}, \mathbf{m}_1^*)$. Then $\mathbf{AP}_1(P_1, P_2, V, \lambda) \;\leq\; 1/|V_\lambda| + \sqrt{\mathbf{AP}_2(P_1, P_2, V, \lambda)}$ for all $\lambda \in \mathbb{N}$.