Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles

Mihir Bellare[1], Marc Fischlin[2], Adam O'Neill[3], and Thomas Ristenpart[1]

[1] Dept. of Computer Science & Engineering, University of California at San Diego
9500 Gilman Drive, La Jolla, CA 92093-0404, USA
{mihir,tristenp}@cs.ucsd.edu
http://www-cse.ucsd.edu/~{mihir,tristenp}

[2] Dept. of Computer Science, Darmstadt University of Technology
Hochschulstrasse 10, 64289 Darmstadt, Germany
fischlin@informatik.tu-darmstadt.de
http://www.fischlin.de/

[3] College of Computing, Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332, USA
amoneill@cc.gatech.edu
http://www.cc.gatech.edu/~amoneill

Abstract. We strengthen the foundations of deterministic public-key encryption via definitional equivalences and standard-model constructs based on general assumptions. Specifically we consider seven notions of privacy for deterministic encryption, including six forms of semantic security and an indistinguishability notion, and show them all equivalent. We then present a deterministic scheme for the secure encryption of uniformly and independently distributed messages based solely on the existence of trapdoor one-way permutations. We show a generalization of the construction that allows secure deterministic encryption of independent high-entropy messages. Finally we show relations between deterministic and standard (randomized) encryption.

1 Introduction

The foundations of public-key encryption, as laid by Goldwasser and Micali [23] and their successors, involve two central threads. The first is definitional equivalences, which aim not only to increase our confidence that we have the "right" notion of privacy but also to give us definitions that are as easy to use in applications as possible. (Easy-to-use indistinguishability is equivalent to the more intuitive, but also more complex, semantic security [21, 23, 24, 28].) The second (of the two threads) is to obtain schemes achieving the definitions under assumptions as minimal as possible. In this paper we pursue these same two threads for *deterministic* encryption [3], proving definitional equivalences and providing constructions based on general assumptions.

D. Wagner (Ed.): CRYPTO 2008, LNCS 5157, pp. 360–378, 2008.

Deterministic encryption. A public-key encryption scheme is said to be deterministic if its encryption algorithm is deterministic. Deterministic encryption was introduced by Bellare, Boldyreva, and O'Neill [3]. The motivating application they gave is efficiently searchable encryption. Deterministic encryption permits logarithmic time search on encrypted data, while randomized encryption only allows linear time search [13, 27], meaning a search requires scanning the whole database. This difference is crucial for large outsourced databases which cannot afford to slow down search. Of course deterministic encryption cannot achieve the classical notions of security of randomized encryption, but [3] formalize a semantic security style notion PRIV that captures the "best possible" privacy achievable when encryption is deterministic, namely that an adversary provided with encryptions of plaintexts drawn from a message-space of high (super-logarithmic) min-entropy should have negligible advantage in computing any public-key independent *partial information function* of the plaintexts. The authors provide some schemes in the random-oracle (RO) model [5] meeting this definition but leave open the problem of finding standard model schemes.

The PRIV definition captures intuition well but is hard to work with. We would like to find simpler, alternative definitions of privacy for deterministic encryption —restricted forms of semantic security as well as an indistinguishablility style definition— that are equivalent to PRIV. We would also like to find schemes not only in the standard model but based on general assumptions.

Notions considered. We define seven notions of privacy for deterministic encryption inspired by the work of [3, 19]. These include a notion IND in the indistinguishability style and six notions —A-CSS, B-CSS, BB-CSS, A-SSS, B-SSS, BB-SSS— in the semantic-security style. The IND definition —adapted from [19]— asks that the adversary be unable to distinguish encryptions of plaintexts drawn from two, adversary-specified, high-entropy message spaces, and is simple and easy to use. The semantic security notions are organized along two dimensions. The first dimension is the class of partial information functions considered, and we look at three choices, namely arbitrary (A), boolean (B), or balanced boolean (BB). (A boolean function is balanced if the probabilities that it returns 0 or 1 are nearly the same.) The second dimension is whether the formalization is simulation (S) based or comparison (C) based.[1] The PRIV notion of [3] is A-CSS in our taxonomy. Low-end notions —think of BB as the lowest, then B then A and similarly C then S in the other dimension— are simpler and easier to use in applications, while high end ones are more intuitively correct. The question is whether the simplifications come at the price of power.

Definitional equivalences. We show that all seven notions discussed above are equivalent. The results are summarized in Figure 1. These results not only

[1] In the first case, A's success in computing partial information about plaintexts from ciphertexts is measured relative to that of a simulator, while in the second it is measured relative to A's own success when it is given the encryption of plaintexts independent of the challenge ones. The terminology is from [8] who prove equivalence between simulation and comparison based notions of non-malleability.

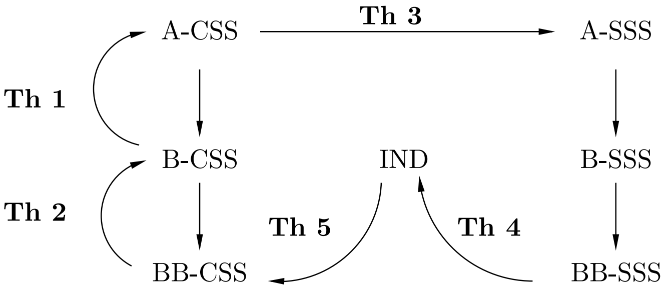

Fig. 1. Notions of security for deterministic encryption schemes and implications showing that all seven notions are equivalent. An arrow $X \to Y$ means that every scheme secure under X is also secure under Y. Unlabeled implications are trivial.

show that semantic security for boolean functions (predicates) is as powerful as semantic security for arbitrary functions, but (perhaps surprisingly) that one can even restrict attention to boolean functions that are balanced, meaning semantic security for balanced boolean functions is as powerful as semantic security for arbitrary functions. We note that balance in this context originates with [19] but they only use it as a tool. We explicitly define and consider the notions BB-CSS and BB-SSS because they bear a natural and intuitive relation to IND and because we feel that the use made of balance by [19] indicates it is important. The proofs of our results rely on new techniques compared to [17, 18, 19].

Definitional framework. We believe that an important and useful contribution of our paper is its definitional framework. Rather than an experiment per notion, we have a few core experiments and then use the approach of [6], capturing different notions via different adversary classes. Advantages of this approach are its easy extendability —for example we can capture the notions of [12] by simply introducing a couple of new adversary classes— and the ability to capture many definitional variants in a way that is unified, concise and yet precise.

A construction for uniform messages. Constructing a non-RO model deterministic encryption scheme meeting our strong notions of security seems like a very challenging problem. We are however able to make progress on certain special cases. We present a deterministic encryption scheme DE1 for the secure encryption of independent, uniformly distributed messages. The scheme is not only without random oracles but based on general trapdoor one-way permutations. To encrypt a random message x one iterates a trapdoor permutation f on x a number of times to get a point y. Let r denote the sequence of Goldreich-Levin [22] hardcore bits obtained in the process. Then one uses a standard IND-CPA scheme —which exists assuming trapdoor one-way permutations— to encrypt y with coins r. The interesting aspect of the scheme, and the source of the difficulty in analyzing it, is its cyclic nature, namely that the coins used for the IND-CPA encryption depend on the plaintext y that is IND-CPA encrypted. The proof manages to show that an adversary who, given y, can distinguish r from random can recover x *even though* this adversary may have partial information about the underlying seed x. The proof exploits in a crucial way that the equivalence between A-CSS and B-CSS holds even for uniformly and independently distributed messages.

Another perspective. A deterministic encryption scheme is (syntactically) the same thing as a family of injective trapdoor functions. Our notions can then be seen as an extension of the usual notion of one-wayness. Our construction is then a family of injective trapdoor functions which hides all (possible) partial information about its (randomly chosen) input. We believe this is a natural and useful strengthening of the usual notion of a trapdoor function that is fully achieved under standard assumptions in our work.

Efficiency. The general assumption notwithstanding, our scheme admits efficient instantiations. For example with squaring as the trapdoor permutation [9] and Blum-Goldwasser [10] as the bare IND-CPA scheme, encryption and decryption come in at about double that of Blum-Goldwasser with no increase in ciphertext size. See Section 5.

A generalization. We generalize our construction to obtain a non-RO model deterministic scheme DE2 for the encryption of independent, high min-entropy (but not necessarily uniform) plaintexts. The assumption used is that one has a trapdoor permutation that is one-way for high min-entropy distributions on its input. This increase in assumption strength is in some sense necessary, since deterministic encryption secure for some distribution trivially provides a one-way injective trapdoor function for that distribution.

From deterministic to randomized encryption. Another central foundational theme is relations between primitives, meaning determining which primitives imply which others. From this perspective we consider how to build IND-CPA-secure standard (randomized) encryption from PRIV-secure deterministic encryption. The obvious approach would be to use the deterministic encryption scheme as a trapdoor one-way function within some well-known general construction [22]. However, this approach leads to large ciphertexts, and we would hope to achieve better efficiency when using a primitive that provides more than one-wayness. We provide a much more efficient construction using a hybrid encryption-style approach, in which the deterministic scheme encrypts a fresh session key padded with extra randomness and the session key is used to encrypt the message. See [4] for the details.

CCA. Lifting our notions and equivalences to the CCA setting is straightforward; see [4] . Our above-mentioned construction of a randomized encryption scheme from a deterministic one works even in the CCA setting. This means, in particular, that we can generically build witness-recovering IND-CCA encryption schemes [25] from arbitrary CCA-secure deterministic schemes.

(Witness-recovering encryption allows, during decryption, recovery of all randomness used to generate a ciphertext.) CCA-secure witness-recovering encryption is of use in further applications [16], and only very recently was a (not very efficient) standard-model construction produced [25]. Our construction shows that building CCA-secure deterministic schemes is at least as hard as building witness-recovering probabilistic encryption.

Related work. Dodis and Smith's work on entropic security [19] has in common with ours the consideration of privacy for messages of high min-entropy. But there are important differences in the settings, namely that theirs is information-theoretic and symmetric while ours is computational and public-key. Dodis and Smith [19] introduce definitions that in our framework are IND, B-SSS, and BB-SSS, to complement the A-SSS-like information-theoretic notion originally proposed by Russell and Wang [26]. Also, Desrosiers [17] and Desrosiers and Dupuis [18] subsequently treat quantum entropic security, providing notions similar to our framework's B-CSS and A-CSS. These works provide some relations between the notions they define. While some of their techniques and implications lift to our setting, others do not. The salient fact that emerges is that prior work *does not* imply equivalence of all seven notions we consider. In particular, the BB-SSS and BB-CSS notions are not considered in [17, 18] and Dodis and Smith [19] only provide reductions for BB-SSS implying A-SSS that result in inefficient or restricted adversaries. See [4] for more information.

Another setting that deals with high min-entropy messages is that of perfectly one-way hash functions (POWHF), introduced by Canetti [14] and further studied by Canetti, Micciancio, and Reingold [15]. These are randomized hash functions that produce publically-verifiable outputs. Our definitions and equivalences can be adapted to the POWHF setting.

Independent work. In concurrent and independent work, Boldyreva, Fehr, and O'Neill [12] consider a relaxation of PRIV in which message sequences need to not merely have high entropy but each message must have high entropy even given the others. They prove some relations between their notions using techniques of [17, 18, 19] but do not consider as many notions as us and in particular do not consider balance. Their schemes achieve stronger notions of security then our DE1 but at the cost of specific algebraic assumptions as opposed to our general one. Combining their results with ours shows that our DE2 achieves their notion of security while using a general (even though non-standard) assumption.

2 Preliminaries

Notation and Conventions. If x is a string then $|x|$ denotes its length; if x is a number then $|x|$ denotes its absolute value; if S is a set then $|S|$ denotes its size. We denote by λ the empty string. If S is a set then $X \leftarrow_{\$} S$ denotes that X is selected uniformly at random from S. We let $x[i \dots j]$ denote bits i through j of string x, for $1 \le i \le j \le |x|$. By $x_1 \parallel \cdots \parallel x_n$ we denote the concatenation of $x_1, \dots, x_n$. Vectors are denoted in boldface, e.g. $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$

denotes the number of components of $\mathbf{x}$ and $\mathbf{x}[i]$ denotes its i^{th} component for $1 \leq i \leq |\mathbf{x}|$. If $i \geq 1$ is an integer, we use B_i as shorthand for $\{0,1\}^i$. By $\langle a, b\rangle$ we denote the inner product modulo 2 of equal-length strings a, b.

We write $\alpha \leftarrow\!\!\$\; X(x, y, \ldots)$ to denote running X on inputs $(x, y, \ldots)$ with fresh random coins and assigning the result to α. We let $[X(x, y, \ldots)]$ denote the set of possible outputs of X when run on $x, y, \ldots \in \{0,1\}^*$. An algorithm X is *non-uniform* if its first input is 1^k and there is a collection $\{C_k\}_{k\in\mathbb{N}}$ of (randomized) circuits such that C_k computes $X(1^k, \ldots)$. The running time is the circuit size. A function f is called *negligible* if it approaches zero faster than the reciprocal of any polynomial, that is, for any polynomial p, there exists $n_p \in \mathbb{N}$ such that $f(n) \leq 1/p(n)$ for all $n \geq n_p$. "PT" stands for polynomial time. We denote by Λ the algorithm that on any inputs returns λ.

Public-key encryption. A *public-key encryption (PKE)* scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a triple of PT algorithms. The key generation algorithm $\mathcal{K}$ takes input 1^k, where $k \in \mathbb{N}$ is the security parameter, and outputs a public-key, secret-key pair (pk, sk). The encryption algorithm $\mathcal{E}$ takes inputs 1^k, pk, and plaintext $x \in \{0,1\}^*$ and outputs a ciphertext. The deterministic decryption algorithm $\mathcal{D}$ takes inputs 1^k, sk, and ciphertext y and outputs either a plaintext x or $\perp$. We say that Π is *deterministic* if $\mathcal{E}$ is deterministic. If $\mathbf{x}$ is a vector of plaintexts, then we write $\mathbf{y} \leftarrow\!\!\$\; \mathcal{E}(1^k, pk, \mathbf{x})$ to denote component-wise encryption of $\mathbf{x}$, i.e. $\mathbf{y}[i] \leftarrow\!\!\$\; \mathcal{E}(1^k, pk, \mathbf{x}[i])$ for all $1 \leq i \leq |\mathbf{x}|$.

3 Security Notions for Deterministic PKE

We first provide formal definitions and then discuss them.

Semantic security. An *SS-adversary* $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}})$ is a tuple of non-uniform algorithms. A_{c} takes as input a unary encoding 1^k of the security parameter $k \in \mathbb{N}$ and returns a string st representing some state information. A_{m} takes input 1^k and st, and returns a vector of challenge messages $\mathbf{x}$ together with a test string t that represents some information about $\mathbf{x}$. A_{g} takes 1^k, a public key and the component-wise encryption of $\mathbf{x}$ under this key, and tries to compute t. The running time of A is defined as the sum of the running times of $A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}$, so that A is PT if $A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}$ are all PT.

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme, $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}})$ an SS-adversary, and S a simulator (a non-uniform algorithm). Let $k \in \mathbb{N}$. Figure 2 displays the css (comparison-based semantic security) and sss (simulation-based semantic security) experiments. We define the css advantage and sss advantage of A by

$$\mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k) \;=\; 2 \cdot \Pr\left[\,\mathbf{Exp}^{\mathrm{css}}_{\Pi,A}(k) \Rightarrow \mathsf{true}\,\right] - 1\;, \text{ and} \qquad (1)$$

$$\mathbf{Adv}^{\mathrm{sss}}_{\Pi,A,S}(k) = \;2 \cdot \Pr\left[\,\mathbf{Exp}^{\mathrm{sss}}_{\Pi,A,S}(k) \Rightarrow \mathsf{true}\,\right] - 1\;. \qquad (2)$$

Our approach to defining the six notions of semantic security of interest to us is to associate to each a corresponding class of adversaries and ask that the

```
$\mathbf{Exp}^{\mathrm{css}}_{\Pi,A}(k)$
$b \xleftarrow{\$} \{0,1\}$ ; $st \xleftarrow{\$} A_{\mathrm{c}}(1^k)$
$(\mathbf{x}_0, t_0) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st)$
$(\mathbf{x}_1, t_1) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$
$\mathbf{c} \xleftarrow{\$} \mathcal{E}(1^k, pk, \mathbf{x}_b)$
$g \xleftarrow{\$} A_{\mathrm{g}}(1^k, pk, \mathbf{c}, st)$
If $g = t_1$ then $b' \leftarrow 1$
Else $b' \leftarrow 0$
Ret $(b' = b)$
```

```
$\mathbf{Exp}^{\mathrm{sss}}_{\Pi,A,S}(k)$
$b \xleftarrow{\$} \{0,1\}$ ; $st \xleftarrow{\$} A_{\mathrm{c}}(1^k)$
$(\mathbf{x}, t) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$
$\mathbf{c} \xleftarrow{\$} \mathcal{E}(1^k, pk, \mathbf{x})$
If $b = 1$ then
    $g \xleftarrow{\$} A_{\mathrm{g}}(1^k, pk, \mathbf{c}, st)$
Else $g \xleftarrow{\$} S(1^k, pk, st)$
If $g = t$ then $b' \leftarrow 1$
Else $b' \leftarrow 0$
Ret $(b' = b)$
```

```
$\mathbf{Exp}^{\mathrm{ind}}_{\Pi,I}(k)$
$b \xleftarrow{\$} \{0,1\}$ ; $st \xleftarrow{\$} I_{\mathrm{c}}(1^k)$
$\mathbf{x}_b \xleftarrow{\$} I_{\mathrm{m}}(1^k, b, st)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$
$\mathbf{c} \xleftarrow{\$} \mathcal{E}(1^k, pk, \mathbf{x}_b)$
$b' \xleftarrow{\$} I_{\mathrm{g}}(1^k, pk, \mathbf{c}, st)$
Ret $(b' = b)$
```

Fig. 2. Three experiments for defining security of encryption schemes

advantage of any adversary in this class be negligible. We proceed to define the relevant classes.

An SS-adversary $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}})$ is *legitimate* if there exists a function $v(\cdot)$, called the number of messages, and a collection $\{\mathbf{y}_k\}_{k\in\mathbb{N}}$ of *reference* message-vectors such that the following three conditions hold. First, $|\mathbf{x}| = v(k)$ for all $(\mathbf{x}, t) \in [A_{\mathrm{m}}(1^k, st)]$ and all $st \in \{0,1\}^*$. Second, $|\mathbf{x}[i]| = |\mathbf{y}_k[i]|$ for all $(\mathbf{x}, t) \in [A_{\mathrm{m}}(1^k, st)]$, all $st \in \{0,1\}^*$, and all $1 \le i \le v(k)$. Third, the function

$$\nu(k) = \Pr\left[\, \mathrm{eq}(\mathbf{x}, \mathbf{y}_k) = 0 \;:\; st \xleftarrow{\$} A_{\mathrm{c}}(1^k) \;;\; (\mathbf{x}, t) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st) \,\right]$$

is negligible, where

$$\mathrm{eq}(\mathbf{x}, \mathbf{y}_k) = \begin{cases} 1 \text{ if } \forall i, j:\ \mathbf{x}[i] = \mathbf{x}[j] \text{ iff } \mathbf{y}_k[i] = \mathbf{y}_k[j] \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

(The third condition reflects that every deterministic scheme leaks plaintext equality.) Let $\mathcal{A}_{\mathrm{SS}}$ be the set of all legitimate, PT SS-adversaries. We say that A has *trivial state function* if $A_{\mathrm{c}} = \Lambda$. Let $\mathcal{A}_{\lambda}$ be the set of all SS-adversaries with trivial state functions.

Without loss of generality (through suitable padding) we can assume there is a function $\ell(\cdot)$ such that the output of $A_{\mathrm{g}}(1^k, \cdot, \cdot)$ and any test string t output by $A_{\mathrm{m}}(1^k, \cdot)$ always have length $\ell(k)$. We call ℓ the *information length* of A. An SS-adversary $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}) \in \mathcal{A}_{\mathrm{SS}}$ is *boolean* if it has information length $\ell(\cdot) = 1$. Let $\mathcal{A}_{\mathrm{B}} \subseteq \mathcal{A}_{\mathrm{SS}}$ be the class of all boolean SS-adversaries. A boolean SS-adversary $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}) \in \mathcal{A}_{\mathrm{B}}$ is δ-balanced if for every st we have

$$\left| \Pr\left[\, t = 0 \;:\; (\mathbf{x}, t) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st) \,\right] - \frac{1}{2} \right| \le \delta \;. \tag{4}$$

When $\delta = 0$ we say that A is *perfectly balanced.* We say that A is *balanced* if it is δ-balanced for some $\delta < 1/2$. Let $\mathcal{A}^{\delta}_{\mathrm{BB}} \subseteq \mathcal{A}_{\mathrm{B}}$ be the class of all δ-balanced boolean SS-adversaries. An SS-adversary $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}) \in \mathcal{A}_{\mathrm{SS}}$ has min-entropy μ if

$$\Pr\left[\, \mathbf{x}[i] = x \;:\; (\mathbf{x}, t) \xleftarrow{\$} A_{\mathrm{m}}(1^k, st) \,\right] \;\le\; 2^{-\mu(k)}$$

for all $k \in \mathbb{N}$, all $1 \leq i \leq v(k)$, all $x \in \{0,1\}^*$, and all $st \in \{0,1\}^*$. Let $\mathcal{A}_{\mathrm{ME}}^{\mu} \subseteq \mathcal{A}_{\mathrm{SS}}$ be the class of all SS-adversaries with min-entropy μ. We say that A has *high min-entropy* if it is in $\mathcal{A}_{\mathrm{ME}}^{\mu}$ for some $\mu(k) \in \omega(\log k)$. Let $\mathcal{A}_{\mathrm{HE}} \subseteq \mathcal{A}_{\mathrm{SS}}$ be the class of all SS-adversaries that have high min-entropy.

Let Π be a PKE scheme. We say that Π is A-CSS secure if $\mathbf{Adv}_{\Pi,A}^{\mathrm{css}}(\cdot)$ is negligible for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda}$; Π is B-CSS-secure if $\mathbf{Adv}_{\Pi,A}^{\mathrm{css}}(\cdot)$ is negligible for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{B}}$; and Π is BB-CSS-secure if there exists $\delta < 1/2$ such that $\mathbf{Adv}_{\Pi,A}^{\mathrm{css}}(\cdot)$ is negligible for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{BB}}^{\delta}$.

Similarly, we say that Π is A-SSS-secure if for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda}$ there exists a PT simulator S such that $\mathbf{Adv}_{\Pi,A,S}^{\mathrm{sss}}(\cdot)$ is negligible; Π is B-SSS-secure if for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{B}}$ there exists a PT simulator S such that $\mathbf{Adv}_{\Pi,A,S}^{\mathrm{sss}}(\cdot)$ is negligible; and Π is BB-SSS-secure if there exists $\delta < 1/2$ such that for all $A \in \mathcal{A}_{\mathrm{HE}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{BB}}^{\delta}$ there exists a PT simulator S such that $\mathbf{Adv}_{\Pi,A,S}^{\mathrm{sss}}(\cdot)$ is negligible.

The *message space* of an SS-adversary $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}})$ is the algorithm A_{d} that on input $1^k, st$ lets $(\mathbf{x}, t) \leftarrow\!\!\$\; A_{\mathrm{m}}(1^k, st)$ and returns $\mathbf{x}$. An SS-adversary is said to produce independent messages if the coordinates of $\mathbf{x}$ are independently distributed when $\mathbf{x} \leftarrow\!\!\$\; A_{\mathrm{d}}(1^k, st)$ for all k, st. Let $\mathcal{A}_{\times}$ be the class of all SS-adversaries which produce independent messages.

For each $d \in \{0,1\}$, we let $\mathbf{Exp}_{\Pi,A}^{\mathrm{css\text{-}d}}(k)$ be the same as $\mathbf{Exp}_{\Pi,A}^{\mathrm{css}}(k)$ except that the first line sets $b \leftarrow d$ rather than picking b at random. We similarly define $\mathbf{Exp}_{\Pi,A,S}^{\mathrm{sss\text{-}d}}(k)$. A standard argument gives

$$\mathbf{Adv}_{\Pi,A}^{\mathrm{css}}(k) = \Pr\left[\,\mathbf{Exp}_{\Pi,A}^{\mathrm{css\text{-}1}}(k) \Rightarrow \mathsf{true}\,\right] - \Pr\left[\,\mathbf{Exp}_{\Pi,A}^{\mathrm{css\text{-}0}}(k) \Rightarrow \mathsf{false}\,\right] \quad \text{and} \quad (5)$$

$$\mathbf{Adv}_{\Pi,A,S}^{\mathrm{sss}}(k) = \Pr\left[\,\mathbf{Exp}_{\Pi,A,S}^{\mathrm{sss\text{-}1}}(k) \Rightarrow \mathsf{true}\,\right] - \Pr\left[\,\mathbf{Exp}_{\Pi,A,S}^{\mathrm{sss\text{-}0}}(k) \Rightarrow \mathsf{false}\,\right] . \quad (6)$$

Indistinguishability. An *IND-adversary* $I = (I_{\mathrm{c}}, I_{\mathrm{m}}, I_{\mathrm{g}})$ is a tuple of non-uniform algorithms. I_{c} takes as input 1^k and returns a string st representing some state information. I_{m} takes input 1^k, a bit b, and st, and returns a vector of messages $\mathbf{x}$. I_{g} takes 1^k, a public key, the component-wise encryption of $\mathbf{x}$ under this key, and st and tries to compute the bit b. The running time of I is defined as the sum of the running times of $I_{\mathrm{c}}, I_{\mathrm{m}}, I_{\mathrm{g}}$, so that I is PT if $I_{\mathrm{c}}, I_{\mathrm{m}}, I_{\mathrm{g}}$ are all PT.

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme, $I = (I_{\mathrm{c}}, I_{\mathrm{m}}, I_{\mathrm{g}})$ an IND-adversary and $k \in \mathbb{N}$. Figure 2 displays the ind experiment. We define the ind advantage of I by

$$\mathbf{Adv}_{\Pi,I}^{\mathrm{ind}}(k) \;=\; 2 \cdot \Pr\left[\,\mathbf{Exp}_{\Pi,I}^{\mathrm{ind}}(k) \Rightarrow \mathsf{true}\,\right] - 1 \;. \quad (7)$$

We next define classes of IND-adversaries. An IND-adversary $I = (I_{\mathrm{c}}, I_{\mathrm{m}}, I_{\mathrm{g}})$ is *legitimate* if there exists a function $v(\cdot)$, called the number of messages, and a collection $\{\mathbf{y}_k\}_{k \in \mathbb{N}}$ of *reference* message-vectors such that the following three conditions hold. First, $|\mathbf{x}| = v(k)$ for all $(\mathbf{x}, t) \in [I_{\mathrm{m}}(1^k, b, st)]$, all $b \in \{0,1\}$, and all $st \in \{0,1\}^*$. Second, $|\mathbf{x}[i]| = |\mathbf{y}_k[i]|$ for all $(\mathbf{x}, t) \in [I_{\mathrm{m}}(1^k, b, st)]$, all $b \in \{0,1\}$, all $st \in \{0,1\}^*$, and all $1 \leq i \leq v(k)$. Third, the function

$$\nu(k) = \Pr\left[\,\mathrm{eq}(\mathbf{x}, \mathbf{y}_k) = 0 \;:\; st \leftarrow\!\!\$\; I_{\mathrm{c}}(1^k) \;;\; b \leftarrow\!\!\$\; \{0,1\} \;;\; (\mathbf{x}, t) \leftarrow\!\!\$\; I_{\mathrm{m}}(1^k, b, st)\,\right]$$

is negligible, where $\mathrm{eq}(\mathbf{x}, \mathbf{y}_k)$ was defined by (3). Let $\mathcal{I}$ be the set of all legitimate, polynomial time IND-adversaries. We say that I has *trivial state function* if $I_\mathrm{c} = \Lambda$. Let $\mathcal{I}_\lambda \subseteq \mathcal{I}$ be the set of all IND-adversaries with trivial state functions. An IND-adversary $I = (I_\mathrm{c}, I_\mathrm{m}, I_\mathrm{g}) \in \mathcal{I}$ has *min-entropy* μ if

$$\Pr\left[\, \mathbf{x}[i] = x \;:\; \mathbf{x} \leftarrow\!\$\; I_\mathrm{m}(1^k, b, st) \,\right] \;\leq\; 2^{-\mu(k)}$$

for all $k \in \mathbb{N}$, all $b \in \{0,1\}$, all $1 \leq i \leq v(k)$, all $x \in \{0,1\}^*$, and all $st \in \{0,1\}^*$. Let $\mathcal{I}^\mu_\mathrm{ME} \subseteq \mathcal{I}$ be the class of all IND-adversaries with min-entropy μ. We say I has *high min-entropy* if it is in $\mathcal{I}^\mu_\mathrm{ME}$ for some $\mu(k) \in \omega(\log k)$. Let $\mathcal{I}_\mathrm{HE}$ be the class of all IND-adversaries that have high min-entropy. We say that Π is IND-secure if $\mathbf{Adv}^{\mathrm{ind}}_{\Pi,I}(\cdot)$ is negligible for all $I \in \mathcal{I}_\mathrm{HE} \cap \mathcal{I}_\lambda$.

For each $d \in \{0,1\}$, we let $\mathbf{Exp}^{\mathrm{ind\text{-}d}}_{\Pi,I}(k)$ be the same as $\mathbf{Exp}^{\mathrm{ind}}_{\Pi,I}(k)$ except that the first line sets $b \leftarrow d$ rather than picking b at random. A standard argument gives

$$\mathbf{Adv}^{\mathrm{ind}}_{\Pi,A}(k) \;=\; \Pr\left[\mathbf{Exp}^{\mathrm{ind\text{-}1}}_{\Pi,I}(k) \Rightarrow \mathsf{true}\right] - \Pr\left[\mathbf{Exp}^{\mathrm{ind\text{-}0}}_{\Pi,I}(k) \Rightarrow \mathsf{false}\right] . \qquad (8)$$

Discussion. A-CSS is exactly the PRIV definition of [3]. As discussed in [3], it is important that A_m does not take input the public key, and this carries over to I_m. In the classical setting a standard hybrid argument [2] shows that the security of encrypting one message implies the security of encrypting multiple messages. In the deterministic encryption setting this is not true in general, which is why $A_\mathrm{m}, I_\mathrm{m}$ output vectors of messages.

Following [3], message spaces are not explicit but rather implicitly defined by their PT sampling algorithms A_m and I_m. As a consequence, message spaces are PT sampleable.

Following [3], the partial information function is not explicit. Think of t as its value on $\mathbf{x}$. This is more general because t is allowed to depend on coins underlying the generation of $\mathbf{x}$ rather than merely on $\mathbf{x}$ itself. (This is stronger than merely allowing the function to be randomized, which is standard.) It allows us in particular to capture "history." However, we show in [4] that this formulation is equivalent to one where the partial information is computed as a function of the message. Note that the (implicit or explicit) partial information functions are PT.

Our security definitions quantify only over adversaries with trivial state functions. We do this for compatibility with [3, 19]. So why introduce the common state function at all? The reason is that it is useful in proofs. Indeed, [19] use such a function implicitly in many places. We believe making it explicit increases clarity. In the end we can always hardwire a "best" state and thereby end up with an adversary in $\mathcal{A}_\lambda$.

4 Relating the Security Notions

In this section we justify the implications summarized by Figure 1. The implications given by the unlabeled arrows are trivial and can be justified by the fact that $X \rightarrow Y$ whenever the adversary class corresponding to Y is a subset of the one corresponding to X. We focus on the implications: A-CSS ⇒ A-SSS; BB-SSS ⇒ IND; IND ⇒ BB-CSS; BB-CSS ⇒ B-CSS; and B-CSS ⇒ A-CSS.

Theorem 1. [B-CSS ⇒ A-CSS] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. Let $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda}$ be an SS-adversary having information length $\ell(\cdot)$. Then there exists a boolean SS-adversary $A' = (A'_{\mathrm{c}}, A'_{\mathrm{m}}, A'_{\mathrm{g}}) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{B}}$ such that for all $k \in \mathbb{N}$*

$$\mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k) \le 2 \cdot \mathbf{Adv}^{\mathrm{css}}_{\Pi,A'}(k) \ . \tag{9}$$

A' has the same message space as A and its running time is that of A plus $\mathcal{O}(\ell)$. □

Proof. The proof is from [19] and repeated here in order to provide intuition for Theorem 2. Below we write ℓ for $\ell(k)$. Then let

```
alg. $A^*_{\mathrm{c}}(1^k)$:
$r \leftarrow\!\!\$\ \{0,1\}^{\ell}$
$s \leftarrow\!\!\$\ \{0,1\}$
Ret $(r, s)$
```

```
alg. $A^*_{\mathrm{m}}(1^k, (r,s))$:
$(\mathbf{x}, t) \leftarrow\!\!\$\ A_{\mathrm{m}}(1^k, \lambda)$
Ret $(\mathbf{x}, \langle r, t\rangle \oplus s))$
```

```
alg. $A^*_{\mathrm{g}}(1^k, pk, \mathbf{c}, (r,s))$:
$g \leftarrow\!\!\$\ A_{\mathrm{g}}(1^k, pk, \mathbf{c}, \lambda)$
Ret $\langle r, g\rangle \oplus s$
```

Then $A^* = (A^*_{\mathrm{c}}, A^*_{\mathrm{m}}, A^*_{\mathrm{g}})$ is certainly boolean, and

$$P_{A^*}(k) = P_A(k) + \frac{1}{2}\left[1 - P_A(k)\right]$$
$$Q_{A^*}(k) = Q_A(k) + \frac{1}{2}\left[1 - Q_A(k)\right]$$

where $P_X(k)$= Pr $\left[\,\mathbf{Exp}^{\text{css-1}}_{\Pi,X}(k) \Rightarrow \mathsf{true}\,\right]$ and $Q_X(k)$= Pr $\left[\,\mathbf{Exp}^{\text{css-0}}_{\Pi,X}(k) \Rightarrow \mathsf{false}\,\right]$. Subtracting, we get $\mathbf{Adv}^{\mathrm{css}}_{\Pi,A^*}(k) = \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k)$. We are not done yet because A^* does not have trivial state function. Let A' be obtained from A^* by hardwiring in a "best" choice of r, s and we are done. ∎

Now we wish to show that BB-CSS ⇒ B-CSS. Note that if the adversary A' constructed in the proof of Theorem 1 were balanced, we would be done. But, A' need not be balanced. Dodis and Smith [19] give a partial solution to this problem, showing that it is in fact possible to find an r that, when hardwired into A^*, results in a balanced adversary, as long as $p \le \epsilon^2/4$, where p is the maximum probability of any t being output by A_{m} and $\epsilon = \mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(\cdot)$.

We will remove this restriction by proceeding as follows. Let A be a given SS-adversary, which from Theorem 1 we can assume is boolean (but not balanced). We again construct an A^* with non-trivial state, but this will consist of n independently chosen keys $\mathbf{K}[1], \ldots, \mathbf{K}[n]$ for a family of pairwise independent hash functions H. Then $A^*_{\mathrm{m}}(1^k, \mathbf{K})$ first runs $(\mathbf{x}, t) \leftarrow\!\!\$\ A_{\mathrm{m}}(1^k, \lambda)$ and then returns $(\mathbf{x}, H(\mathbf{K}[i], t))$ for random $i \in \{1, \ldots, n\}$, while $A^*_{\mathrm{g}}(1^k, pk, \mathbf{c}, \mathbf{K})$ picks its own independent random j and returns $H(\mathbf{K}[j], A_{\mathrm{g}}(1^k, pk, \mathbf{c}, \lambda))$. Our analysis will show that for a suitable choice of n there exists a choice of the vector $\mathbf{K}$ which, when hardwired into A^*, yields an adversary A' having all the claimed properties. The theorem is below and the proof is in the full version [4].

Theorem 2. [BB-CSS ⇒ B-CSS] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. Let $A = (A_c, A_m, A_g) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{B}}$ be a boolean SS-adversary. Let $\epsilon(\cdot) = \mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(\cdot) > 0$ and let $\delta = 1/4$. Then there exists an SS-adversary $A' = (A'_c, A'_m, A'_g) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}^{\delta}_{\mathrm{BB}}$ such that for all $k \in \mathbb{N}$*

$$\mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k) \leq 4n(k) \cdot \mathbf{Adv}^{\mathrm{css}}_{\Pi,A'}(k) ,$$

where $n(k) = \max\{485 , \lceil 64 \cdot \ln(1/\epsilon(k)) + 64 \ln 4 \rceil\}$. A' has the same message space as A and its running time is that of A plus $\mathcal{O}(\log(1/\epsilon(k)) + k)$. □

Below are theorem statements for the other three implications. Proofs are given in the full version [4].

Theorem 3. [A-CSS ⇒ A-SSS] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. Let $A = (A_c, A_m, A_g) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda}$ be an SS-adversary outputting at most v messages. Then there exists a simulator S such that for all $k \in \mathbb{N}$*

$$\mathbf{Adv}^{\mathrm{sss}}_{\Pi,A,S}(k) \leq \mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k) .$$

The running time of S is that of A plus the time to perform v encryptions. □

Theorem 4. [BB-SSS ⇒ IND] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. Let $I = (I_c, I_m, I_g) \in \mathcal{I}^{\mu}_{\mathrm{ME}} \cap \mathcal{I}_{\lambda}$ be an IND-adversary. Let $\delta = 0$. Then there exists an SS-adversary $A = (A_c, A_m, A_g) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}^{\delta}_{\mathrm{BB}}$ such that for any simulator S and all $k \in \mathbb{N}$*

$$\mathbf{Adv}^{\mathrm{ind}}_{\Pi,I}(k) \leq 2 \cdot \mathbf{Adv}^{\mathrm{sss}}_{\Pi,A,S}(k) .$$

The running time of A is that of I. □

Theorem 5. [IND ⇒ BB-CSS] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. Let $0 \leq \delta < 1/2$ and let $A = (A_c, A_m, A_g) \in \mathcal{A}^{\mu}_{\mathrm{ME}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}^{\delta}_{\mathrm{BB}}$ be an SS-adversary. Then there exists an ind-adversary $I = (I_c, I_m, I_g) \in \mathcal{I}^{\nu}_{\mathrm{ME}} \cap \mathcal{I}_{\lambda}$ such that for all $k \in \mathbb{N}$*

$$\mathbf{Adv}^{\mathrm{css}}_{\Pi,A}(k) \leq 2 \cdot \mathbf{Adv}^{\mathrm{ind}}_{\Pi,I}(k) + 2^{-k} .$$

I has min-entropy $\nu(k) = \mu(k) - 1 + \log(1 - 2\delta)$ and its running time is that of A plus the time for $\lceil -(\log(2/(1 + 2\delta)))^{-1} \rceil (k + 3) + 1$ executions of A_m. □

5 Deterministic Encryption from Trapdoor Permutations

We construct a deterministic encryption scheme, without ROs, that meets our definitions in the case that the messages being encrypted are uniformly and independently distributed. It is based on the existence of trapdoor permutations. In [4] we generalize the construction to independently distributed messages of high min-entropy μ, but under the (stronger and non-standard) assumption of the existence of trapdoor permutations that are one-way under all input distributions of min entropy μ.

Primitives. A family of trapdoor permutations $\mathcal{TP} = (G, F, \overline{F})$ is a triple of PT algorithms, with the last two being deterministic. On input 1^k, the key

```
alg. $\mathcal{K}(1^k)$:
$(\phi, \tau) \leftarrow_{\$} G(1^k)$
$s \leftarrow_{\$} \{0,1\}^k$
$(\overline{pk}, \overline{sk}) \leftarrow_{\$} \overline{\mathcal{K}}(1^k)$
$pk \leftarrow (\phi, \overline{pk}, s)$
$sk \leftarrow (\tau, \overline{sk})$
Ret $(pk, sk)$
```

```
alg. $\mathcal{E}(1^k, pk, x)$:
$(\phi, \overline{pk}, s) \leftarrow pk$
$y \leftarrow F_{\phi}^{n(k)}(x)$
$\omega \leftarrow \mathcal{G}(1^k, 1^{n(k)}, \phi, x, s)$
$c \leftarrow \overline{\mathcal{E}}(1^k, \overline{pk}, y\,;\,\omega)$
Ret $c$
```

```
alg. $\mathcal{D}(1^k, sk, c)$:
$(\tau, \overline{sk}) \leftarrow sk$
$y \leftarrow \overline{\mathcal{D}}(1^k, \overline{sk}, c)$
$x \leftarrow \overline{F}_{\tau}^{n(k)}(y)$
Ret $x$
```

Fig. 3. Algorithms defining our deterministic encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$

generation algorithm G returns a pair (ϕ, τ) of strings such that $F_{\phi}(\cdot) = F(\phi, \cdot)$ is a permutation on $\{0,1\}^k$ and $\overline{F}_{\tau}(\cdot) = \overline{F}(\tau, \cdot)$ is its inverse. If $f\colon\ \{0,1\}^k \rightarrow \{0,1\}^k$ then $f^i\colon\ \{0,1\}^k \rightarrow \{0,1\}^k$ is defined inductively by $f^0(x) = x$ and $f^{i+1}(x) = f(f^i(x))$ for $i \geq 0$ and $x \in \{0,1\}^k$. The Blum-Micali-Yao [11, 28], Goldreich-Levin [22] generator $\mathcal{G}_{\mathcal{TP}}$ takes input $1^k, 1^n, \phi$ and $x, s \in B_k$ and returns

$$\left\langle F_{\phi}^{0}(x), s \right\rangle \parallel \left\langle F_{\phi}^{1}(x), s \right\rangle \parallel \cdots \parallel \langle F_{\phi}^{n-1}(x), s \rangle \ .$$

To discuss the security of our scheme, we say that an SS-adversary is uniform if for every k and every st the components of $\mathbf{x}$ are uniformly and independently distributed over $\{0,1\}^k$ when $(\mathbf{x}, t) \leftarrow_{\$} A_{\mathrm{m}}(1^k, st)$. We let $\mathcal{A}_{\mathrm{UN}}$ be the class of all uniform SS-adversaries. If $f : B_k \rightarrow B_k$ then $f(\mathbf{x})$ denotes the vector whose i^{th} component is $f(\mathbf{x}[i])$. We let $\mathcal{G}_{\mathcal{TP}}(1^k, 1^n, \phi, \mathbf{x}, s)$ be the vector whose i^{th} component is $\mathcal{G}_{\mathcal{TP}}(1^k, 1^n, \phi, \mathbf{x}[i], s)$.

The construction. We fix a (randomized) encryption scheme $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. Assume that $\overline{\mathcal{E}}(1^k, \cdot, \cdot)$ draws its coins from $\{0,1\}^{n(k)}$, and write $\overline{\mathcal{E}}(1^k, pk, x\,;\,\omega)$ for the execution of $\overline{\mathcal{E}}$ on inputs $1^k, pk, x$ and coins ω. Let $\mathcal{TP} = (G, F, \overline{F})$ be a family of trapdoor permutations and $\mathcal{G}_{\mathcal{TP}}$ the associated generator. Our deterministic encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined as shown in Figure 3. We refer to it as DE1.

Intuition. A weird aspect of our scheme is that one is encrypting, under the standard scheme $\overline{\mathcal{E}}$, a message y under coins ω that are related to y. The challenge is to show that this works assuming $\mathcal{TP}$ is one-way and $\overline{\Pi}$ is IND-CPA. So let $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{g}}) \in \mathcal{A}_{\mathrm{UN}} \cap \mathcal{A}_{\lambda}$ be an adversary with associated information length $\ell(\cdot)$ and number of messages $v(\cdot)$ that is successful in violating the A-CSS security of Π. It is not hard to see that the assumed security of $\overline{\Pi}$ allows us to reduce our task to showing that it is hard for a PT adversary D to have a non-negligible advantage in computing the challenge bit b in the following distinguishing game. The game generates $\phi, \tau, \overline{pk}, \overline{sk}, s$ as done by $\mathcal{K}(1^k)$ and lets $(\mathbf{x}, t) \leftarrow_{\$} A_{\mathrm{m}}(1^k, \lambda)$. It lets

$$\boldsymbol{\omega}_1 \leftarrow \mathcal{G}_{\mathcal{TP}}(1^k, 1^{n(k)}, \phi, \mathbf{x}, s) \quad \text{and} \quad \boldsymbol{\omega}_0 \leftarrow_{\$} B_{n(k)}^{v(k)} \ ,$$

picks a random challenge bit b, and provides the adversary D with ϕ, s, $F_{\phi}^{n(k)}(\mathbf{x})$, $\boldsymbol{\omega}_b$, and t. Now, D's task would be merely the standard (and known to be hard) one of breaking the pseudorandomness of $\mathcal{G}_{\mathcal{TP}}$ (meaning, we would be done) but for one catch, namely that D has "help" information t about the seed(s) $\mathbf{x}$. If we could somehow remove it we would be done, but this seems hard to do directly. Instead, we first produce from D an adversary I' that solves (although still with help) a computational (rather than decision) problem, namely that of inverting F_ϕ: given ϕ, $F_\phi(x)$, and $\ell(\cdot)$ bits of information about x, our adversary computes x. This is obtained by noting that the Goldreich-Levin [22] and Blum-Micali-Yao [11, 28] proof of pseudorandomness of $\mathcal{G}_{\mathcal{TP}}$ based on the one-wayness of $\mathcal{TP}$ generalizes to say that $\mathcal{G}_{\mathcal{TP}}$ remains pseudorandom in the presence of $\ell(\cdot)$ bits of help information about the seed assuming $\mathcal{TP}$ is one-way in the presence of $\ell(\cdot)$ bits of help information about the input. Now we need to turn I' into an adversary succeeding at the same task, but without help. We appeal to Theorem 1, which allows us to assume our starting adversary A was boolean, meaning $\ell(\cdot) = 1$. In this case it is easy to dispense with the help provided to I because we can try both values of it and lower our success probability by at most a factor of 2.

We remark that we have made crucial use of the fact that the adversary constructed by Theorem 1 has the same message space as the original one. This means that if the latter is in $\mathcal{A}_{\mathrm{UN}}$ then so is the former, so that B-CSS for uniform adversaries implies A-CSS for uniform adversaries. We now proceed to the full proof.

OWPs and PRGs with help. For our proof, we will need to extend the usual frameworks of one-wayness and pseudorandomness to adversaries with "help." An inversion adversary $J = (J_{\mathrm{c}}, J_{\mathrm{p}}, J_{\mathrm{s}})$ is a triple of non-uniform algorithms. If $\mathcal{TP} = (G, F, \overline{F})$ is a family of trapdoor permutations we let

$$\mathbf{Adv}^{\mathrm{owf}}_{\mathcal{TP},J}(k) = \Pr\left[\,\mathbf{Exp}^{\mathrm{owf}}_{\mathcal{TP},J}(k) \Rightarrow \mathsf{true}\,\right]$$

where the experiment is shown in Figure 4. The running time of J is defined as the sum of the running times of J_{c} and J_{s}, so that J is PT if $J_{\mathrm{c}}, J_{\mathrm{s}}$ are PT. (J_{p} is not required to be PT.) We say that J has help-length $\ell(\cdot)$ if the output of $J_{\mathrm{p}}(1^k, \cdot, \cdot, \cdot)$ is always of length $\ell(k)$. We say that J is unaided if it has help length $\ell(\cdot) = 0$. We let $\mathcal{J}_\ell$ denote the class of all PT inversion adversaries with help length $\ell(\cdot)$. We say $\mathcal{TP}$ is one-way for help-length $\ell(\cdot)$ if $\mathbf{Adv}^{\mathrm{owf}}_{\mathcal{TP},J}(\cdot)$ is negligible for all $J \in \mathcal{J}_\ell$. We say that $\mathcal{TP}$ is one-way if it is one-way for help-length $\ell(\cdot) = 0$. The following, although trivial, will be very useful.

Proposition 1. *Let $\mathcal{TP}$ be a family of trapdoor permutations and J an inversion adversary with help-length $\ell(\cdot)$. Then there is an inversion adversary J' with help-length 0 such that*

$$\mathbf{Adv}^{\mathrm{owf}}_{\mathcal{TP},J}(k) \le 2^{\ell(k)} \cdot \mathbf{Adv}^{\mathrm{owf}}_{\mathcal{TP},J'}(k)$$

for all k, and the running time of J' is that of J plus $\mathcal{O}(\ell)$. □

```
$\mathbf{Exp}^{\mathrm{owf}}_{\mathcal{TP},J}(k)$
$(\phi,\tau) \leftarrow\!\$\; G(1^k)$ ; $st \leftarrow\!\$\; J_{\mathrm{c}}(1^k,\phi)$
$x \leftarrow\!\$\; \{0,1\}^k$ ; $t \leftarrow\!\$\; J_{\mathrm{p}}(1^k,x,\phi,st)$
$y \leftarrow F_\phi(x)$ ; $x' \leftarrow\!\$\; J_{\mathrm{s}}(1^k,\phi,st,y,t)$
Ret $(x = x')$
```

```
$\mathbf{Exp}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(k)$
$(\phi,\tau) \leftarrow\!\$\; G(1^k)$ ; $st \leftarrow\!\$\; D_{\mathrm{c}}(1^k,\phi)$
$\mathbf{x} \leftarrow\!\$\; B_k^{v(k)}$ ; $s \leftarrow\!\$\; \{0,1\}^k$ ; $d \leftarrow\!\$\; \{0,1\}$
$t \leftarrow\!\$\; D_{\mathrm{p}}(1^k,\mathbf{x},\phi,st)$
$\boldsymbol{\omega}_1 \leftarrow \mathcal{G}_{\mathcal{TP}}(1^k,1^{n(k)},\phi,\mathbf{x},s)$
$\boldsymbol{\omega}_0 \leftarrow\!\$\; B_{n(k)}^{v(k)}$
$d' \leftarrow\!\$\; D_{\mathrm{g}}(1^k,\phi,st,F_\phi^{n(k)}(\mathbf{x}),\boldsymbol{\omega}_d,s,t)$
Ret $(d = d')$
```

Fig. 4. (Left) Experiment defining one-wayness of $\mathcal{TP} = (G, F, \overline{F})$. **(Right)** Experiment defining pseudorandomness of $\mathcal{G}_{\mathcal{TP}}$.

Proof. Let $J = (J_{\mathrm{c}}, J_{\mathrm{p}}, J_{\mathrm{s}})$ and $J' = (J_{\mathrm{c}}, \Lambda, J'_{\mathrm{s}})$ where $J'_{\mathrm{s}}(1^k, \phi, st, y, \lambda)$ lets $t \leftarrow\!\$\; \{0,1\}^{\ell(k)}$ and returns $J_{\mathrm{s}}(1^k, \phi, st, y, t)$. ▮

A PRG adversary $D = (D_{\mathrm{c}}, D_{\mathrm{p}}, D_{\mathrm{g}})$ is a triple of non-uniform algorithms. If $\mathcal{TP} = (G, F, \overline{F})$ is a family of trapdoor permutations and $\mathcal{G}_{\mathcal{TP}}$ is the corresponding generator we let

$$\mathbf{Adv}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(k) = 2 \cdot \Pr\left[\,\mathbf{Exp}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(k) \Rightarrow \mathsf{true}\,\right] - 1$$

where the experiment is shown in Figure 4 and $v(\cdot), n(\cdot)\colon\ \mathbb{N} \to \mathbb{N}$. The running time of D is defined as the sum of the running times of D_{c} and D_{g}, so that D is PT if $D_{\mathrm{c}}, D_{\mathrm{g}}$ are PT. (D_{p} is not required to be PT.) We say that D has help-length $\ell(\cdot)$ if the output of $D_{\mathrm{p}}(1^k, \cdot, \cdot, \cdot)$ is always of length $\ell(k)$. We let $\mathcal{D}_\ell$ denote the class of all PT PRG-adversaries with help length $\ell(\cdot)$. We say $\mathcal{G}_{\mathcal{TP}}$ is pseudorandom for help-length $\ell(\cdot)$ if $\mathbf{Adv}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(\cdot)$ is negligible for all $D \in \mathcal{D}_\ell$ and all polynomials v, n. We say that $\mathcal{G}_{\mathcal{TP}}$ is pseudorandom if it is pseudorandom for help-length $\ell(\cdot) = 0$. We remark that it is important that D_{p} does not get s as input, meaning the help information is only about x. The following says that if $\mathcal{TP}$ is one-way for help-length $\ell(\cdot)$ then $\mathcal{G}_{\mathcal{TP}}$ is pseudorandom for help-length $\ell(\cdot)$. The case $\ell(\cdot) = 0$ is the standard result [11, 22, 28] saying that $\mathcal{G}_{\mathcal{TP}}$ is pseudorandom if $\mathcal{TP}$ is one-way. The proof of the following is in the full version [4].

Lemma 1. *Let $\mathcal{TP} = (G, F, \overline{F})$ be a family of trapdoor permutations. Let $v(\cdot)$, $n(\cdot)$ be polynomials. Let D be a PRG-adversary with help-length $\ell(\cdot)$ and let $\epsilon(\cdot) = \mathbf{Adv}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(\cdot) > 0$. Then there is an inversion adversary J with help-length $\ell(\cdot)$ such that*

$$\epsilon(k) \leq 4n(k)v(k) \cdot \mathbf{Adv}^{\mathrm{owf}}_{\mathcal{TP},J}(k)$$

and the running time of J is

$$T_J = \mathcal{O}(k^3 n^4 v^4 \epsilon^{-4}) + \mathcal{O}(T_D + nvT_F)k^2 n^2 v^2 \epsilon^{-2}\ ,$$

where T_X is the running time of X. □

IND-CPA. Associate to (randomized) encryption scheme $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ and adversary B the experiment $\mathbf{Exp}^{\text{ind-cpa}}_{\overline{\Pi},B}(k)$ defined by

$$b \leftarrow_{\$} \{0,1\}\ ;\ (\overline{pk}, \overline{sk}) \leftarrow_{\$} \overline{\mathcal{K}}(1^k)\ ;\ b' \leftarrow_{\$} B^{\overline{\mathcal{E}}_{\overline{pk}}(\mathrm{LR}(\cdot,\cdot,b))}(\overline{pk})\ ;\ \mathrm{Ret}\ (b = b')$$

where $\mathrm{LR}(M_0, M_1, b) = M_b$. B is an IND-CPA adversary if all its oracle queries consist of equal length strings. Let

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\Pi},B}(k) = 2 \cdot \Pr\left[\, \mathbf{Exp}^{\text{ind-cpa}}_{\overline{\Pi},B}(k) \Rightarrow \mathsf{true} \,\right] - 1\ .$$

We say that $\overline{\Pi}$ is IND-CPA secure if $\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\Pi},B}(\cdot)$ is negligible for all PT IND-CPA adversaries B.

Security of our scheme. The following says that our scheme is B-CSS secure for uniform adversaries assuming $\mathcal{TP}$ is one-way and $\overline{\Pi}$ is IND-CPA secure. By Theorem 1 it is A-CSS secure for uniform adversaries under the same assumptions and a constant factor loss in security. Since the existence of one-way trapdoor permutations implies the existence of IND-CPA secure encryption schemes we obtain the results under the sole assumption of the existence of one-way trapdoor permutations.

Theorem 6. *Let $\mathcal{TP} = (G, F, \overline{F})$ be a family of trapdoor permutations and $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ an encryption scheme. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the associated deterministic encryption scheme as per our construction above. Let $A = (A_{\mathrm{c}}, A_{\mathrm{m}}, A_{\mathrm{c}}) \in \mathcal{A}_{\mathrm{B}} \cap \mathcal{A}_{\lambda} \cap \mathcal{A}_{\mathrm{UN}}$ be an SS-adversary against Π with advantage $\epsilon(\cdot) = \mathbf{Adv}^{\text{css}}_{\Pi,A}(\cdot) > 0$ and number of messages $v(\cdot)$. Then there is an unaided inversion adversary J and an IND-CPA adversary B such that for all $k \in \mathbb{N}$*

$$\epsilon(k) \leq 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\overline{\Pi},B}(k) + 16n(k)v(k) \cdot \mathbf{Adv}^{\text{owf}}_{\mathcal{TP},J}(k)\ . \tag{10}$$

The running time of B is that of A plus $\mathcal{O}(nT_F + T_{\mathcal{G}})$ and it makes $v(k)$ oracle queries. The running time of J is

$$\mathcal{O}(k^3 n^4 v^4 \epsilon^{-4}) + \mathcal{O}(T_A + T_{\overline{\mathcal{E}}} + T_{\overline{\mathcal{K}}} + nvT_F) \cdot k^2 n^2 v^2 \epsilon^{-2} \tag{11}$$

where T_X is the running time of X. □

Proof. Consider the experiments of Figure 5. There $\overline{\mathcal{E}}(1^k, \overline{pk}, \mathbf{y}\,;\, \boldsymbol{\omega})$ is the vector whose i^{th} component is $\overline{\mathcal{E}}(1^k, \overline{pk}, \mathbf{y}[i]\ ;\ \boldsymbol{\omega}[i])$. Let

$$P_a = \Pr\left[\, \mathbf{Exp}^{d\text{-}a}_{\Pi,A}(k) \Rightarrow \mathsf{true} \,\right]$$

for $a \in \{0,1\}$. Then

$$\mathbf{Adv}^{\text{css}}_{\Pi,A}(k) = 2P_1 - 1 = 2(P_1 - P_0) + (2P_0 - 1)\ .$$

Adversary B is shown in Figure 5, and we omit the (easy) analysis establishing that

$$2P_0 - 1 \leq \mathbf{Adv}^{\text{ind-cpa}}_{\overline{\Pi},B}(k)\ .$$

```
$\mathbf{Exp}^{d\text{-}1}_{\Pi,A}(k)$ / $\boxed{\mathbf{Exp}^{d\text{-}0}_{\Pi,A}(k)}$
$b \leftarrow\$ \{0,1\}$
$(\mathbf{x}_0, t_0), (\mathbf{x}_1, t_1) \leftarrow\$ A_{\mathrm{m}}(1^k, \lambda)$
$(\phi, \tau) \leftarrow\$ G(1^k)$ ; $s \leftarrow\$ \{0,1\}^k$
$(\overline{pk}, \overline{sk}) \leftarrow\$ \overline{\mathcal{K}}(1^k)$ ; $pk \leftarrow (\phi, \overline{pk}, s)$
$\boldsymbol{\omega} \leftarrow \mathcal{G}_{\mathcal{TP}}(1^k, 1^{n(k)}, \phi, \mathbf{x}_b, s)$
$\boxed{\boldsymbol{\omega} \leftarrow\$ B^{v(k)}_{n(k)}}$
$\mathbf{y} \leftarrow F^{n(k)}_{\phi}(\mathbf{x}_b)$ ; $\mathbf{c} \leftarrow \overline{\mathcal{E}}(1^k, \overline{pk}, \mathbf{y} \,;\, \boldsymbol{\omega})$
$g \leftarrow\$ A_{\mathrm{g}}(1^k, pk, \mathbf{c}, \lambda)$
If $g = t_1$ then $b' \leftarrow 1$ else $b' \leftarrow 0$
Ret $(b = b')$
```

```
adversary $B^{\overline{\mathcal{E}}_{\overline{pk}}(\mathrm{LR}(\cdot,\cdot,b))}(\overline{pk})$:
$(\mathbf{x}_0, t_0), (\mathbf{x}_1, t_1) \leftarrow\$ A_{\mathrm{m}}(1^k, \lambda)$
$(\phi, \tau) \leftarrow\$ G(1^k)$ ; $s \leftarrow\$ \{0,1\}^k$
$pk \leftarrow (\phi, \overline{pk}, s)$
$\mathbf{y}_0 \leftarrow F^{n(k)}_{\phi}(\mathbf{x}_0)$ ; $\mathbf{y}_1 \leftarrow F^{n(k)}_{\phi}(\mathbf{x}_1)$
For $i = 1, \ldots, v(k)$ do
    $\mathbf{c}[i] \leftarrow\$ \overline{\mathcal{E}}_{\overline{pk}}(\mathrm{LR}(\mathbf{y}_0[i], \mathbf{y}_1[i], b))$
$g \leftarrow\$ A_{\mathrm{g}}(1^k, pk, \mathbf{c}, \lambda)$
If $g = t_1$ then Ret 1 else Ret 0
```

Fig. 5. (Left) Experiments used in the proof of Theorem 6. The experiment d-0 includes the boxed statement while d-1 does not. **(Right)** IND-CPA adversary for proof of Theorem 6.

```
alg. $D_{\mathrm{p}}(1^k, \mathbf{x}, \phi, \lambda)$:
Repeat
    $(\mathbf{x}', t') \leftarrow\$ A_{\mathrm{m}}(1^k, \lambda)$
Until $(\mathbf{x}' = \mathbf{x})$
$t \leftarrow t'$
Ret $t$
```

```
alg. $D_{\mathrm{g}}(1^k, \phi, \lambda, \mathbf{y}, \boldsymbol{\omega}, s, t)$:
$c \leftarrow\$ \{0,1\}$ ; $\mathbf{y}_1 \leftarrow \mathbf{y}$ ; $t_1 \leftarrow t$ ; $\boldsymbol{\omega}_1 \leftarrow \boldsymbol{\omega}$
$(\mathbf{x}_0, t_0) \leftarrow\$ A_{\mathrm{m}}(1^k, \lambda)$
$(\overline{pk}, \overline{sk}) \leftarrow\$ \overline{\mathcal{K}}(1^k)$ ; $pk \leftarrow (\phi, \overline{pk}, s)$
$\boldsymbol{\omega}_0 \leftarrow \mathcal{G}_{\mathcal{TP}}(1^k, 1^{n(k)}, \phi, \mathbf{x}_0, s)$ ; $\mathbf{y}_0 \leftarrow F^{n(k)}_{\phi}(\mathbf{x}_0)$
$\mathbf{c} \leftarrow \overline{\mathcal{E}}(1^k, \overline{pk}, \mathbf{y}_c \,;\, \boldsymbol{\omega}_c)$
$g \leftarrow\$ A_{\mathrm{g}}(1^k, pk, \mathbf{c}, \lambda)$
If $(g = t_c)$ then $c' \leftarrow 1$ else $c' \leftarrow 0$
Ret $c \oplus c' \oplus 1$
```

Fig. 6. PRG adversary for proof of Theorem 6

Next we define PRG-adversary $D = (\Lambda, D_{\mathrm{p}}, D_{\mathrm{g}})$ with help length $\ell(\cdot)$ as shown in Figure 6 and claim that

$$P_1 - P_0 \leq 2 \cdot \mathbf{Adv}^{\mathrm{prg}\text{-}v}_{\mathcal{TP},D,n}(k) \,. \tag{12}$$

Let J' be the inversion adversary obtained from D by Lemma 1. It also has help-length $\ell(\cdot)$. Now apply Proposition 1 to get inversion adversary J with help-length 0. In [4] we justify (12), (10) and (11) to conclude the proof. ∎

Instantiations. DE1 admits quite efficient instantiations. Say we want to encrypt a 1024 bit (random) message. Let the trapdoor one-way permutation be squaring modulo a 1024-bit composite number N [9] that is part of the public key. Then the PRG requires n squarings, where n is the number of bits of randomness required by the (randomized) encryption scheme $\overline{\Pi}$. Let $\overline{\Pi}$ be the

Blum-Goldwasser scheme [10], also using a 1024-bit modulus. (This modulus, also part of the public key, must be different from N.) Then encryption cost of $\mathsf{DE1}$ is that of Blum-Goldwasser (1024 squarings) plus $n = 1024$ squarings for the PRG to get coins for $\overline{\Pi}$. (We assume here, and below, an efficient mapping from bits to group elements, otherwise n increases by a small amount.) Decryption time also doubles, coming in at about 4 exponentiations modulo 512 bit numbers (less than one 1024 bit exponentiation!) using Chinese remainders. The ciphertext size is that of Blum-Goldwasser, namely 2048 bits, and security rests solely on factoring. Alternatively, let $\overline{\Pi}$ be El Gamal hybrid encryption using a 160-bit group. (A universal hash of the DH key is used to one-time symmetrically encrypt the data.) Encryption time for $\mathsf{DE1}$ is that of hybrid El Gamal plus the time for $n = 320$ squarings modulo N, decryption time is 2 exponentiations modulo 512 bit numbers plus one 160-bit exponentiation. and the ciphertext size is only 1344 bits. The security assumption is now factoring + DDH.

Discussion. One might ask why we did not work with IND rather than with CSS notions. The reason is that it is unclear how to meaningfully capture the case of uniformly and independently distributed messages with IND. We could certainly say that an IND-adversary $I = (I_{\rm c}, I_{\rm m}, I_{\rm g})$ is uniform if for every k and every st, b the components of $\mathbf{x}$ are uniformly distributed over $\{0,1\}^k$ when $\mathbf{x} \leftarrow_{\$} I_{\rm m}(1^k, b, st)$. But such an adversary would always have zero advantage.

Acknowledgments

Mihir Bellare was supported in part by NSF grants CNS 0524765 and CNS 0627779 and a gift from Intel Corporation. Marc Fischlin was supported in part by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG). Adam O'Neill was supported in part by Alexandra Boldyreva's NSF CAREER award 0545659. Thomas Ristenpart was supported in part by the above-mentioned grants of the first author.

References

1. Bellare, M.: The Goldreich-Levin Theorem (manuscript), `http://www-cse.ucsd.edu/users/mihir/papers/gl.pdf`
2. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
3. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
4. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. Full version of this paper. IACR ePrint archive (2008) `http://eprint.iacr.org/`
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Conference on Computer and Communications Security – CCS 1993, pp. 62–73. ACM, New York (1993)

6. Bellare, M., Rogaway, P.: Robust computational secret sharing and a unified account of classical secret-sharing goals. In: Conference on Computer and Communications Security – CCS 2007, pp. 172–184. ACM, New York (2007)
7. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
8. Bellare, M., Sahai, A.: Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 519–536. Springer, Heidelberg (1999)
9. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on Computing 15, 364–383 (1986)
10. Blum, M., Goldwasser, S.: An efficient probabilistic public-key encryption scheme which hides all partial information. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 289–302. Springer, Heidelberg (1984)
11. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM Journal on Computing 13, 850–864 (1984)
12. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
13. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
14. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)
15. Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions (Preliminary version). In: Symposium on the Theory of Computation – STOC 1998, pp. 131–141 (1998)
16. Damgaard, I., Hofheinz, D., Kiltz, E., Thorbek, R.: Public-key encryption with non-interactive opening. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 239–255. Springer, Heidelberg (2008)
17. Desrosiers, S.: Entropic security in quantum cryptography. arXiv e-Print quant-ph/0703046 (2007), http://arxiv.org/abs/quant-ph/0703046
18. Desrosiers, S., Dupuis, F.: Quantum entropic security and approximate quantum encryption. arXiv e-Print quant-ph/0707.0691 (2007), http://arxiv.org/abs/0707.0691
19. Dodis, Y., Smith, A.: Entropic security and the encryption of high entropy messages. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 556–577. Springer, Heidelberg (2005)
20. El Gamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
21. Goldreich, O.: A uniform complexity treatment of encryption and zero-knowledge. Journal of Cryptology 6, 21–53 (1993)
22. Goldreich, O., Levin, L.: A hard-core predicate for all one-way functions. In: Symposium on the Theory of Computation – STOC 1989, pp. 25–32. ACM, New York (1989)
23. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and Systems Sciences 28(2), 412–426 (1984)
24. Micali, S., Rackoff, C., Sloan, R.: The notion of security for probabilistic cryptosystems. SIAM Journal on Computing 17(2), 412–426 (1988)

25. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Symposium on the Theory of Computing – STOC 2008, pp. 187–196. ACM, New York (2008)
26. Russell, A., Wang, H.: How to fool an unbounded adversary with a short key. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 133–148. Springer, Heidelberg (2002)
27. Song, D., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Symposium on Security and Privacy, pp. 44–55. IEEE, Los Alamitos (2000)
28. Yao, A.: Theory and applications of trapdoor functions. In: Symposium on Foundations of Computer Science – FOCS 1982, pp. 80–91. IEEE, Los Alamitos (1982)