Deterministic and Efficiently Searchable Encryption

Mihir Bellare[1], Alexandra Boldyreva[2] and Adam O'Neill[2]

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

[2] College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA.
{aboldyre,amoneill}@cc.gatech.edu
http://www.cc.gatech.edu/{∼ aboldyre, amoneill}

Abstract. We present as-strong-as-possible definitions of privacy, and constructions achieving them, for public-key encryption schemes where the encryption algorithm is *deterministic*. We obtain as a consequence database encryption methods that permit fast (i.e. sub-linear, and in fact logarithmic, time) search while provably providing privacy that is as strong as possible subject to this fast search constraint. One of our constructs, called RSA-DOAEP, has the added feature of being length preserving, so that it is the first example of a public-key cipher. We generalize this to obtain a notion of efficiently-searchable encryption schemes which permit more flexible privacy to search-time trade-offs via a technique called bucketization. Our results answer much-asked questions in the database community and provide foundations for work done there.

1 Introduction

The classical notions of privacy for public-key encryption schemes, mainly indistinguishability or semantic security under chosen-plaintext or chosen-ciphertext attack [34, 43, 47, 28, 10], can only be met when the encryption algorithm is randomized. This paper treats the case where the encryption algorithm is deterministic. We begin by discussing the motivating application.

Fast search. Remote data storage in the form of outsourced databases is of increasing interest [49]. Data will be stored in encrypted form. (The database service provider is not trusted.) We are interested in a public key setting, where anyone can add to the database encrypted data which a distinguished "receiver" can retrieve and decrypt. The encryption scheme must permit search (by the receiver) for data retrieval. Public-key encryption with keyword search (PEKS) [15, 1, 17] is a solution that provably provides strong privacy but search takes time linear in the size of the database. Given that databases can be terabytes in size, this is prohibitive. The practical community indicates that they want search on encrypted data to be as efficient as on unencrypted data, where a record containing a given field value can be a retrieved in time logarithmic in the size of the database. (For example, via appropriate tree-based data structures.) Deterministic encryption allows just this. The encrypted fields can be stored in the data structure, and one can find a target ciphertext in time logarithmic in the size of the database. The question is what security one can expect. To answer this, we need a definition of privacy for deterministic encryption.

A DEFINITION. One possibility is to just ask for one-wayness, but we would like to protect partial information about the plaintext to the maximum extent possible. To gauge what this could be, we note two inherent limitations of deterministic encryption. First, no privacy is possible if the plaintext is known to come from a small space. Indeed, knowing that c is the encryption under public key pk of a plaintext x from a set X, the adversary can compute the encryption c_x of x under pk for all $x \in X$, and return as the decryption of c the x satisfying $c_x = c$. We address this by only requiring privacy when the plaintext is drawn from a space of large min-entropy. Second, and more subtle, is that the ciphertext itself is partial information about the plaintext. We address this by only requiring non-leakage of partial information when the plaintext and partial information do not depend on the public key. This is reasonable because in real life public keys are hidden in our software and data does not depend on them. While certainly weaker than the classical notions met by randomized schemes, our resulting notion of privacy for deterministic encryption, which we call PRIV, is still quite strong. The next question is how to achieve this new notion.

CONSTRUCTIONS. Our first construction is generic and natural: Deterministically encrypt plaintext x by applying the encryption algorithm of a randomized scheme but using as coins a hash of (the public key and) x. We show that this "Encrypt-with-Hash" deterministic encryption scheme is PRIV secure in the random oracle (RO) model of [12] assuming the starting randomized scheme is IND-CPA secure. Our second construction is an extension of RSA-OAEP [13, 31]. The padding transform is deterministic but uses three Feistel rounds rather than the two of OAEP. RSA-DOAEP is proven PRIV secure in the RO model assuming RSA is one-way. This construction has the attractive feature of being length-preserving. (The length of the ciphertext equals the length of the plaintext.) This is important when bandwidth is expensive —senders in the database setting could be power-constrained devices— and for securing legacy code.

HISTORICAL CONTEXT. Diffie and Hellman [26] suggested that one encrypt plaintext x by applying to it an injective trapdoor function. A deterministic encryption scheme is just a family of injective trapdoor functions, so our definition is an answer to the question of how much privacy Diffie-Hellman encryption can provide. (We clarify that not all trapdoor functions meet our definition. For example, plain RSA does not.)

In the symmetric setting, deterministic encryption is captured by ciphers including block ciphers. So far there has been no public key analog. Deterministic encryption meeting our definition provides one, and in particular RSA-DOAEP is the first length-preserving public-key cipher.

Efficiently searchable encryption. We introduce the notion of *efficiently searchable encryption* (ESE) schemes. These are schemes permitting fast (i.e. logarithmic time) search. Encryption may be randomized, but there is a deterministic, collision-resistant function of the plaintext that can also be computed from the ciphertext and serves as a tag, permitting the usual (fast) comparison-based search. Deterministic encryption schemes are a special case and the notion of security remains the same. (Our PRIV definition does not actually require encryption to be deterministic.) The benefit of the generalization is to permit schemes with more flexible privacy to search-time trade-offs. Specifically, we analyze a scheme from the database literature that we call "Hash-and-Encrypt." It encrypts the plaintext with a randomized scheme but also includes in the ciphertext a deterministic, collision-resistant hash of the plaintext. (This is an ESE scheme with the hash playing the role of the tag, and so permits fast search.) We prove that this scheme is PRIV secure in the RO model when the underlying encryption scheme is IND-CPA. With this scheme, loss of privacy due to lack of entropy in the message space can be compensated for by increasing the probability δ of hash collisions. (This can be done, for example, by truncating the output of the hash function.) The trade-off is that the receiver gets "false positives" in response to a search query and must spend the time to sift through them to obtain the true answer. This technique is known as *bucketization* in the database literature, but its security was not previously rigorously analyzed.

Discussion. Our schemes only provide privacy for plaintexts that have high min-entropy. (This is inherent in being deterministic or efficiently searchable, not a weakness of our particular constructs.) We do not claim database fields being encrypted have high min-entropy. They might or they might not. The point is that practitioners have indicated that they will not sacrifice search time for privacy. Our claim is to provide the best possible privacy subject to allowing fast search. In some cases, this may very well mean no privacy. But we also comment that bucketization can increase privacy (at the cost of extra processing by the receiver) when the database fields being encrypted do not have high min-entropy.

Extensions. Our basic PRIV definition, and the above-mentioned results, are all for the CPA (chosen-plaintext attack) case. The definition easily extends to the CCA (chosen-ciphertext attack) case, and we call the resulting notion PRIV-CCA. Our Encrypt-with-Hash deterministic encryption scheme is not just PRIV, but in fact PRIV-CCA, in the RO model even if the underlying randomized encryption scheme is only IND-CPA, as long as the latter has the extra property that no ciphertext is too likely. In Section 6 we detail this and also discuss how RSA-DOAEP and Encrypt-and-Hash fare under CCA.

Open. All our constructs are in the RO model. An important open question is to construct ESE or deterministic encryption schemes meeting our definition in the standard model. We note that in the past also we have seen new notions first emerge only with RO constructions achieving them, but later standard model constructs have been found. This happened for example for IBE [14, 52] and PEKS [17]. Note that the results of [32] rule out a standard model black-box reduction from deterministic public-key encryption to ordinary public-key encryption, but the former could still be built under other assumptions.

Related work. In the symmetric setting, deterministic encryption is both easier to define and to achieve than in the asymmetric setting. Consider the experiment that picks a random challenge bit b and key K and provides the adversary with a left-or-right oracle that, given plaintexts x_0, x_1 returns the encryption of x_b under K. Security asks that the adversary find it hard to guess b as long as its queries $(x_0^1, x_1^1), \ldots, (x_0^q, x_1^q)$ satisfy the condition that $x_0^1, \ldots, x_0^q$ are all distinct and also $x_1^1, \ldots, x_1^q$ are all distinct. To the best of our knowledge, this definition of privacy for deterministic symmetric encryption first appeared in [11]. However, it is met by a PRP and in this sense deterministic symmetric encryption goes back to [42].

Previous searchable encryption schemes provably meeting well-defined notions of privacy include [15, 35, 1, 6, 16, 17] in the public-key setting and [50, 33, 23] in the symmetric setting. However, all these require linear-time search, meaning the entire database must be scanned to answer each query. In the symmetric setting, further assumptions such as the data being known in advance, and then having the user (who is both the "sender" and "reciever" in this setting) pre-compute a specialized index for the server, has been shown to permit efficiency comparable to ours without sacrificing security [24]. Follow-on work to ours [4] treats ESE (as we mean it here) in the symmetric setting, providing the symmetric analog of what we do in our current paper.

Sub-linear time searchable encryption has been much targeted by the database security community [45, 3, 36, 25, 38, 39, 41, 37, 19, 51]. However, they mostly employ weak, non-standard or non-existing primitives and lack definitions or proofs of security. As a notable exception, Kantarcioglu and Clifton [40] recently called for a new direction of research on secure database servers aiming instead for "efficient encrypted database and query processing with *provable* security properties." They also propose a new cryptographic definition that ensures schemes reveal only the number of records accessed on each query, though a scheme meeting the definition requires tamper-resistant trusted hardware on the server.

Definitions that, like ours, restrict security to high min-entropy plaintexts have appeared before, specifically in the contexts of perfectly one-way probabilistic hash functions (POWHFs) [20, 21] and information-theoretically secure one-time symmetric encryption [48, 27]. The first however cannot be met by deterministic schemes, and neither handle the public-key related subtleties we mentioned above. (Namely that we must limit security to plaintexts not depending on the public key.) Also our definition considers the encryption of multiple related messages while those of [20, 21] consider only independent messages.

Use for other applications. We note that one can also use our definitions to analyze other systems-security applications. In particular, a notion of "convergent encryption" is proposed in [2, 29] for the problem of eliminating wasted space in an encrypted file system by combining duplicate files across multiple users. Despite pinpointing the correct intuition for security of their scheme, they are only able to formally show (for lack of a suitable security definition) that it achieves the very weak security notion of one-wayness. One can use our definitions to show that their scheme achieves much stronger security.

2 Notation and Conventions

Unless otherwise indicated, an algorithm may be randomized. An adversary is either an algorithm or a tuple of algorithms. In the latter case, we say it is polynomial time if each constituent algorithm is polynomial time. The abbreviation "PT" stands for "polynomial time" and "PTA" for polynomial time algorithm or adversary. If x is a string then $|x|$ denotes its length in bits. By $x_1 \| \cdots \| x_n$ we denote an encoding of $x_1, \ldots, x_n$ from which $x_1, \ldots, x_n$ are uniquely recoverable. Vectors are denoted in boldface, for example $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of components of $\mathbf{x}$ and $\mathbf{x}[i]$ denotes its ith component ($1 \le i \le |\mathbf{x}|$).

3 Deterministic Encryption and its Security

Asymmetric Encryption. An (asymmetric) encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three PTAs. The key-generation algorithm $\mathcal{K}$ takes input the unary encoding 1^k of the security parameter k to return a public key pk and matching secret key sk. The encryption algorithm $\mathcal{E}$ takes pk and a plaintext x to return a ciphertext. The deterministic decryption algorithm $\mathcal{D}$ takes sk and a ciphertext c to return a plaintext. We require that $\mathcal{D}(sk, c) = x$ for all for all k and all $x \in \mathrm{PtSp}(k)$, where the probability is over the experiment

$$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(1^k) \; ; \; c \stackrel{\$}{\leftarrow} \mathcal{E}(pk, x)$$

and PtSp is a plaintext space associated to Π. Unless otherwise indicated, we assume $\mathrm{PtSp}(k) = \{0,1\}^*$ for all k. We extend $\mathcal{E}$ to vectors via

Algorithm $\mathcal{E}(pk, \mathbf{x})$
  For $i = 1, \ldots, |\mathbf{x}|$ do $\mathbf{y}[i] \stackrel{\$}{\leftarrow} \mathcal{E}(pk, \mathbf{x}[i])$
  Return $\mathbf{y}$

We say that Π is deterministic if $\mathcal{E}$ is deterministic. Although this is an important case of interest below, this is not assumed by the definition, which also applies when $\mathcal{E}$ is randomized.

Privacy Adversaries. A privacy adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ is a pair of PTAs. We clarify that $A_{\mathrm{m}}, A_{\mathrm{g}}$ share neither coins nor state. A_{m} takes input 1^k but *not* the public key, and returns a plaintext vector $\mathbf{x}$ together with some side information t. A_{g} takes $1^k, pk$ and an encryption $\mathbf{x}$ and tries to compute t.

The adversary must also obey the following rules. First, there must exist functions $v(\cdot), n(\cdot)$ such that $|\mathbf{x}| = v(k)$ and $|\mathbf{x}[i]| = n(k)$ for k, all $(\mathbf{x}, t)$ output by $A_{\mathrm{m}}(1^k)$, and all $1 \le i \le v(k)$. Second, all plaintext vectors must have the same *equality pattern*, meaning for all $1 \le i, j \le v(k)$ there is a symbol $\diamond \in \{=, \neq\}$ such that $\mathbf{x}[i] \diamond \mathbf{x}[j]$ for all $(\mathbf{x}, t)$ output by $A_{\mathrm{m}}(1^k)$. We say that A has *min-entropy* $\mu(\cdot)$ if

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

for all $1 \le i \le v(k)$ and all $x \in \{0,1\}^*$. We say that A has *high* min-entropy if $\mu(k) \in \omega(\log(k))$.

The definition below is for chosen-plaintext attacks (CPA). In Section 6 we extend the definition to take chosen-ciphertext attacks (CCA) into account.

THE DEFINITION. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and A be a privacy adversary as above. We associate to A, Π the following:

Experiment $\mathbf{Exp}^{\text{priv-0}}_{\Pi,A}(k)$	**Experiment** $\mathbf{Exp}^{\text{priv-1}}_{\Pi,A}(k)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$	$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$
$(\mathbf{x}_1, t_1) \xleftarrow{\$} A_{\mathrm{m}}(1^k)$	$(\mathbf{x}_0, t_0) \xleftarrow{\$} A_{\mathrm{m}}(1^k)$; $(\mathbf{x}_1, t_1) \xleftarrow{\$} A_{\mathrm{m}}(1^k)$
$\mathbf{c} \xleftarrow{\$} \mathcal{E}(pk, \mathbf{x}_1)$	$\mathbf{c} \xleftarrow{\$} \mathcal{E}(pk, \mathbf{x}_0)$
$g \xleftarrow{\$} A_{\mathrm{g}}(1^k, pk, \mathbf{c})$	$g \xleftarrow{\$} A_{\mathrm{g}}(1^k, pk, \mathbf{c})$
If $g = t_1$ then return 1	If $g = t_1$ then return 1
Else return 0	Else return 0

The *advantage* of a privacy adversary A against Π is

$$\mathbf{Adv}^{\text{priv}}_{\Pi,A}(k) = \Pr\left[\, \mathbf{Exp}^{\text{priv-0}}_{\Pi,A}(k) = 1 \,\right] - \Pr\left[\, \mathbf{Exp}^{\text{priv-1}}_{\Pi,A}(k) = 1 \,\right] .$$

We say that Π is PRIV secure if $\mathbf{Adv}^{\text{priv}}_{\Pi,A}(\cdot)$ is negligible for every PTA A with high min-entropy.

As usual, in the random oracle (RO) model [12], all algorithms and adversaries are given access to the RO(s). In particular, both A_{m} and A_{g} get this access. Let us now discuss some noteworthy aspects of the new definition.

ACCESS TO THE PUBLIC KEY. If A_{m} were given pk, the definition would be unachievable for deterministic Π. Indeed, $A_{\mathrm{m}}(1^k)$ could output $(\mathbf{x}, t)$ where $\mathbf{x}[1]$ is chosen at random from $\{0,1\}^k$, $|\mathbf{x}| = 1$, and $t = \mathcal{E}(pk, \mathbf{x})$. Then $A_{\mathrm{g}}(1^k, pk, c)$ could return c, and A would have min-entropy 2^{-k} but

$$\mathbf{Adv}^{\text{priv}}_{\Pi,A}(k) \;\ge\; 1 - 2^{-k} .$$

Intuitively, the ciphertext is non-trivial information about the plaintext, showing that any deterministic scheme leaks information about the plaintext that depends on the public key. Our definition asks that information unrelated to the public key not leak. Note that this also means that we provide security only for messages unrelated to the public key, which is acceptable in practice, because normal data is unlikely to depend on any public key. In real life, public keys are abstractions hidden in our software, not strings we look at.

VECTORS OF MESSAGES. The classical definitions explicitly only model the encryption of a single plaintext, but a simple hybrid argument from [8] shows that security when multiple plaintexts are encrypted follows. This hybrid argument fails in our setting. One can give examples showing that the version of our definition in which $|\mathbf{x}|$ is restricted to be 1 does not imply the stated version. (See the full paper [9] for details.) This is why we have explicitly considered the encryption of multiple messages.

The High Min-Entropy Requirement. In the absence of the high-entropy restriction on A, it is clear that the definition would be unachievable for deterministic Π. To see this, consider $A_{\mathrm{m}}(1^k)$ that outputs $(0,0)$ with probability $1/2$ and $(1,1)$ with probability $1/2$. Then $A_{\mathrm{g}}(1^k, pk, c)$ could return 0 if $\mathcal{E}(pk, 0) = c$ and 1 otherwise, giving A an advantage of $1/2$. This reflects the fact that trial encryption of candidate messages is always a possible attack when encryption is deterministic.

Security for Multiple Users. The classical notions of privacy, as well as ours, only model a single user (SU) setting, where there is just one receiver and thus just one public key. An extension of the classical notions to cover multiple users, each with their own public key, is made in [8, 7], and these works go on to show that SU security implies multi-user (MU) security in this case. We leave it open to appropriately extend our definition to the MU setting and then answer the following questions: does SU security imply MU security, and do our schemes achieve MU security? But we conjecture that the answer to the first question is "no" while the answer to the second is "yes."

4 Secure Deterministic Encryption Schemes

We propose two constructions of deterministic schemes that we prove secure under our definition.

4.1 Encrypt-with-Hash

We first propose a generic deterministic encryption scheme that replaces the coins used by a standard encryption scheme with the hash of the message. More formally, let $\mathsf{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be any public-key encryption scheme. Say that $\mathcal{E}(pk, x)$ draws its coins from a set $\mathsf{Coins}_{pk}(|x|)$. We write $\mathcal{E}(pk, x; R)$ for the output of $\mathcal{E}$ on inputs pk, x and coins R. Let $H\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a hash function with the property that $H(pk, x) \in \mathsf{Coins}_{pk}(|x|)$ for all pk and all $x \in \{0,1\}^*$. The RO-model "*Encrypt-with-Hash*" deterministic encryption scheme $\mathsf{EwH} = (\mathcal{DK}, \mathcal{DE}, \mathcal{DD})$ is defined via

Alg $\mathcal{DK}(1^k)$	**Alg** $\mathcal{DE}^H(pk, x)$	**Alg** $\mathcal{DD}^H((sk, pk), y)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$	$R \leftarrow H(pk, x)$	$x \leftarrow \mathcal{D}(sk, y)$
Return $(pk, (sk, pk))$	$y \leftarrow \mathcal{E}(pk, x; R)$	$R \leftarrow H(pk, x)$
	Return y	If $\mathcal{E}(pk, x; R) = y$ then
		Return x
		Else Return $\perp$

The *max public-key probability* $\mathsf{mpk}(\cdot)$ of AE is defined as follows: for every k we let $\mathsf{mpk}(k)$ be the maximum taken over all $w \in \{0,1\}^*$ of the quantity

$$\Pr\left[\,(pk, sk) \xleftarrow{\$} \mathcal{K} \;:\; pk = w\,\right] .$$

The following then implies that the construction achieves PRIV-security if the starting encryption scheme is IND-CPA.

Theorem 1. *Suppose there is a privacy adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ against* EwH *with min-entropy μ, which outputs vectors of size v and makes at most q_{h} queries to its hash oracle. Then there exists an IND-CPA adversary B against* AE *such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EwH},A} \;\le\; \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathsf{AE},B} + \frac{2q_{\mathrm{h}}v}{2^{\mu}} + 2q_{\mathrm{h}}\mathsf{mpk}\;, \qquad (1)$$

where mpk *is the max public-key probability of* AE. *Furthermore, B makes v queries to its LR-oracle and its running-time is at most that of A plus $O(1)$.*

The proof is in [9].

We stress that $\mathsf{mpk}(\cdot)$ is negligible for any IND-CPA scheme, so its being small here is *not* an extra assumption. The reason we make the term explicit is that for most schemes it is easy to analyze directly and is unconditionally exponentially-small in the security parameter, which provides more precise security guarantees. For example, in ElGamal [30], the public key contains a value g^x, where x is a random exponent in the secret key. In this case, the max public-key probability is $1/|G|$, where $|G|$ is the order of the corresponding group. Also note that in the theorem (and in the rest of the paper), we use the definition of IND-CPA (or -CCA) that allows an adversary to make as many queries as it likes to its LR-oracle. This is known to be equivalent (with loss in security by a factor less than or equal to the total number of LR-queries made) to allowing only one such query [8]. We also clarify that (1) is a relationship between functions of k, so we are saying it holds for all $k \in \mathbb{N}$. For simplicity of notation we omit k here and further in the paper.

4.2 RSA-DOAEP, A Length-Preserving Deterministic Scheme

It is sometimes important to minimize the number of bits transmitted over the network. We devise an efficient deterministic encryption scheme that is optimal in this regard, namely is length-preserving. (That is, the length of the ciphertext equals the length of the plaintext.) Length-preserving schemes can also be needed for securing legacy code. Ours is the first such construction shown secure under a definition of security substantially stronger than one-wayness, and in particular is the first construction of an asymmetric cipher.

The Scheme. The construction is based on RSA-OAEP [13, 31]. But in place of the randomness in this scheme we use part of the message, and we add an extra round to the underlying transform. Formally, our scheme is parameterized by integers $k_0, k_1 > 0$ satisfying $n > 2k_0$ and $n \ge k_1$. The plaintext space $\mathrm{PtSp}(k)$ consists of all strings of length at least $\max(k_1, 2k_0+1)$. We assume here for simplicity that all messages to encrypt have a fixed length $n = n(k)$. Let $\mathcal{F}$ be an RSA trapdoor-permutation generator with modulus length $|N| = k_1$. The key-generation algorithm of the associated RO-model deterministic encryption scheme $\mathsf{RSA\text{-}DOAEP}$ ("D" for "deterministic") on input 1^k runs $\mathcal{F}$ on the same input and returns (N, e) as the public key and (N, d) as the secret key. Let $s[i \dots j]$ denote bits i through j of a string s, for $1 \le i \le j \le |s|$. The encryption and decryption algorithms have oracle access to functions $H_1, H_2 \colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^{k_0}$ and $R \colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^{n-k_0}$, and are defined as follows:

Algorithm $\mathcal{E}^{H_1,H_2,R}((N,e),x)$	**Algorithm** $\mathcal{D}^{H_1,H_2,R}((N,d),y)$
$x_l \leftarrow x[1 \dots k_0]$	$x_1 \leftarrow y[1 \dots n-k_1]$
$x_r \leftarrow x[k_0+1 \dots n]$	$y_1 \leftarrow y[n-k_1+1 \dots n]$
$s_0 \leftarrow H_1((N,e),x_r) \oplus x_l$	$x \leftarrow x_1 \| (y_1^d \bmod N)$
$t_0 \leftarrow R((N,e),s_0) \oplus x_r$	$s_1 \leftarrow x[1 \dots k_0]$
$s_1 \leftarrow H_2((N,e),t_0) \oplus s_0$	$t_0 \leftarrow x[k_0+1 \dots n]$
$x_1 \leftarrow (s_1 \| t_0)[1 \dots n-k_1]$	$s_0 \leftarrow H_2((N,e),t_0) \oplus s_1$
$x_2 \leftarrow (s_1 \| t_0)[n-k_1+1 \dots n]$	$x_r \leftarrow R((N,e),s_0) \oplus t_0$
$y \leftarrow x_1 \| (x_2^e \bmod N)$	$x_l \leftarrow H_1((N,e),x_r) \oplus s_0$
Return y	Return $x_l \| x_r$

Security. The following implies that the construction achieves PRIV-security if RSA is one-way.

Theorem 2. *Suppose there exists a privacy adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ against* $\mathsf{RSA\text{-}DOAEP}$ *with min-entropy μ that makes at most $q_{\mathrm{h_i}}$ queries to oracle H_i for $i \in \{1,2\}$ and q_{r} to oracle R, and outputs vectors of size v with components of length n. Let* mpk *be the max public-key probability of* $\mathsf{RSA\text{-}DOAEP}$*. We consider two cases:*

- *Case 1: $n < k_0 + k_1$. Then there is an inverter I against $\mathcal{F}$ such that*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{RSA\text{-}DOAEP},A} \;\le\; & q_{\mathrm{h}_2} v \cdot \sqrt{\mathbf{Adv}^{\mathrm{owf}}_{\mathcal{F},I} + 2^{4k_0 - 2k_1 + 10}} \\ & - \; 2^{2k_0 - k_1 + 5} + \frac{2q_{\mathrm{r}} v}{2^{k_0}} + \frac{2q_{\mathrm{h}_1} q_{\mathrm{r}} v}{2^{\mu}} + 2(q_{\mathrm{h}_1} + q_{\mathrm{h}_2} + q_{\mathrm{r}})\mathsf{mpk}\,.\end{aligned}$$

  Furthermore the running-time of I is at most that of A plus $O(q_{\mathrm{h}_2} \log(q_{\mathrm{h}_2}) + k_1^3)$.

- *Case 2: $n \ge k_0 + k_1$. Then there is an inverter I against RSA $\mathcal{F}$ such that*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{RSA\text{-}DOAEP},A} \;\le\; & v \cdot \mathbf{Adv}^{\mathrm{owf}}_{\mathcal{F},I} \\ & + \; \frac{2q_{\mathrm{r}} v}{2^{k_0}} + \frac{2q_{\mathrm{h}_1} q_{\mathrm{r}} v}{2^{\mu}} + 2(q_{\mathrm{h}_1} + q_{\mathrm{h}_2} + q_{\mathrm{r}})\mathsf{mpk}\,.\end{aligned}$$

  Furthermore, the running-time of I is at most that of A plus $O(q_{\mathrm{h}_2} \log(q_{\mathrm{h}_2}))$.

The proof is in [9].

In practice, we will have, e.g. $k_1 = 1024$, and then one can set the parameter k_0 to, say, 160 bits to effectively maximize security regardless of which case of the theorem applies (i.e. independent of the length of the particular plaintext to encrypt). Thus, typically, letting n be the length to whose restriction the message space gives the smallest adversarial min-entropy, the relation between $n - 160$ and 1024 then determines which case of the theorem applies. We note that the weaker security guarantee in Case 1 is analogous to the state-of-the-art for RSA-OAEP itself [31, 46].

Encrypting Long Messages. Typically, to encrypt long messages efficiently using an asymmetric scheme in practice, one employs hybrid encryption. This methodology in particular applies to the Encrypt-with-Hash construction, in which the starting scheme can be a hybrid one. However, with RSA-DOAEP, we do not provide an explicit way to securely utilize hybrid encryption while keeping encryption deterministic, and, in any case, if using some form of hybrid encryption, RSA-DOAEP would no longer be length-preserving (since an encrypted symmetric key would need to be included with the ciphertext). We would therefore like to stress that one can efficiently encrypt long messages using RSA-DOAEP *without* making use of hybrid encryption. Intuitively, this is possible because, somewhat similarly to the randomized case [18], the underlying Feistel-network in the scheme acts as a kind of "all-or-nothing transform" (AONT), such that unless an adversary with large min-entropy inverts the RSA image in a ciphertext then it cannot recover any information about a (long) message, for the practical parameter settings given above.

5 Efficiently Searchable Encryption (ESE)

We now turn to the aforementioned application of outsourced databases, where data is sent to a remote server. The database server is untrusted. The data in each field in the database is encrypted separately under the public key of a receiver, who needs to be able to query the server to retrieve the encrypted records containing particular data. Since databases are often large, a linear scan by the server on each query is prohibitive. Deterministic encryption provides a possible solution to the problem. A query, i.e. a ciphertext, specifies the exact piece of data the server needs to locate, so the server can answer it just like for unencrypted data, and hence search-time stays sub-linear (or logarithmic) in the database size. In general though, encryption permitting efficient search does not to be deterministic *per se*. Accordingly, we first define a new primitive that we call *efficiently searchable encryption (ESE)*, which more generally permits this "efficient searchability."

The New Primitive. The basic idea is to associate a "tag" to a plaintext, which can be computed both by the client to form a particular query *and* by the server from a ciphertext that encrypts it, so that it can index the data appropriately in standard (e.g. tree-based) data structures and search according to the tags. These functionalities are captured, respectively, by the functions F, G below.

Let $\mathsf{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme with associated plaintext space $\mathrm{PtSp}(k)$. We say AE is a *δ-efficiently searchable encryption (-ESE)* scheme where $\delta(\cdot) < 1$ if there exist PTAs F, G such that for every k we have

1. Perfect Consistency: For every $x_1 \in \mathrm{PtSp}(k)$, the probability that $F(pk, x_1) = G(pk, c)$ equals one, where the probability is over

$$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) \; ; \; c \xleftarrow{\$} \mathcal{E}(pk, x_1) \; .$$

2. Computational Soundness: For every PTA $\mathcal{M}$ that on input 1^k outputs a pair of distinct messages in $\mathrm{PtSp}(k)$, the probability that $F(pk, x_0) = G(pk, \mathcal{E}(pk, x_1))$ is at most $\delta(k)$, where the probability is over

$$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k) \; ; \; (x_0, x_1) \xleftarrow{\$} \mathcal{M}(1^k) \; .$$

We refer to the output of F, G as the *tag* of the message or a corresponding ciphertext.

Above, consistency ensures that the server can locate at least the desired ciphertexts on a query, because their tags and those of the plaintexts used in forming the query will be the same. Soundness limits the number of false-positives that are located as well, by bounding the number of other plaintexts that may have the same tag, and precludes degeneracy, where the whole database is returned on every query. With flexible trade-offs are desirable here, δ may be quite large. This is why $\mathcal{M}$ is not given input pk; if it were, the soundness condition would not make sense for large δ, since $\mathcal{M}$ could compute tags of messages itself and then output two of the many it finds to agree on their tags. As in our definition of privacy, one way to view this restriction is as saying that, in practice, the data is not picked as depending on any public key.

Security of ESE. The rule that a privacy adversary output vectors with the same equality pattern has a natural interpretation in the context of ESE. Intuitively, this means that, in the outsourced database application, all the server should learn about the data is which records contain the same attribute values/keywords and how many times each one occurs (called the *occurrence profile/distribution* of the data).

As shown in the full version [9], any deterministic encryption scheme is 0-efficiently searchable under our definition. We will see how under our PRIV definition, relaxing the soundness of a different ESE scheme (i.e. increasing δ) via "bucketization" (cf. [44, 22]), with each plaintext being randomly assigned a tag and some number of them corresponding to each tag (i.e. each "bucket"), though requiring the receiver to do more work to filter out false-positive results can mitigate the power of a dictionary attack by the server when min-entropy of the data is low. While one might want to try to use such bucketization to hide the occurrence profile of the data as well, or else to have the bucket distribution depend on that of the input, as we explain in the full version [9], neither of these are likely to be possible in practice. So we stick to the PRIV definition in analyzing ESE.

We next analyze a simple probabilistic ESE construction occurring in the database literature.

5.1 Encrypt-and-Hash ESE

This scheme represents an approach suggested in the database literature, in which the tag of a message is its hash. Let $\mathsf{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be any pubic-key encryption scheme and $H\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^{l_{\mathrm{h}}}$ for some $l_{\mathrm{h}} > 0$ be a hash function. The RO-model "*Encrypt-and-Hash*" encryption scheme $\mathsf{EaH} = (\mathcal{HK}, \mathcal{HE}, \mathcal{HD})$ is defined via

Alg $\mathcal{HK}(1^k)$	**Alg** $\mathcal{HE}^H(pk, x)$	**Alg** $\mathcal{HD}^H((sk, pk), y\|h)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(1^k)$	$h \leftarrow H(pk, x)$	$x \leftarrow \mathcal{D}(sk, y)$
Return $(pk, (sk, pk))$	$y \leftarrow \mathcal{E}(pk, x)$	$h' \leftarrow H(pk, x)$
	Return $y\|h$	If $h' = h$ then
		Return x
		Else Return $\perp$

Then EaH is efficiently searchable under our definition. (See the full version [9] for details.) The following implies the construction is PRIV secure if the underlying encryption scheme is IND-CPA, independent of l_{h}, the proof of which appears in [9].

Theorem 3. *Suppose there is a privacy adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ against EaH that outputs vectors of size v and makes at most q_{h} queries to its hash oracle. Then there exists an IND-CPA adversary B against AE such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EaH},A} \;\leq\; \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathsf{AE},B} + \frac{2q_{\mathrm{h}}v}{2^{\mu}} + 2q_{\mathrm{h}}\mathsf{mpk}\ ,$$

where mpk is the max public-key probability of AE. Furthermore, B makes v queries to its LR-oracle and its running-time is at most that of A plus $O(1)$.

The above tells us that the construction achieves security when min-entropy of the data is high enough to preclude a dictionary attack by the adversary against the scheme. What about when min-entropy of the data is not high? In this case, the construct allows for bucketization as previously described. To obtain a γ-ESE scheme (assuming that γ is power of two), one can simply set l_{h} to be $\log\gamma$. The particular RO chosen for an instance of the scheme then determines the plaintext-to-tag mapping. Intuitively, if the number of plaintexts corresponding to any given tag is not too low, the scheme can still provide reasonable security because the adversary will not be able to distinguish ciphertexts of plaintexts with equal tags. The following captures the security gain from using this technique in a quantatitive way. The parameter j below represents a lower bound on the minimum bucket-size (i.e. the minimum number of plaintexts associated to any given tag) according to the choice of the RO, which we hope to be as large as possible with a given hash-length l_{h} for security.

Theorem 4. *Suppose there is a privacy adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ against EaH with min-entropy μ, which outputs vectors of length v and makes at most q_{h} queries to H. Then there exists an IND-CPA adversary B against AE such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EaH},A} \;\leq\; \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathsf{AE},B} + \frac{2q_{\mathrm{h}}v}{2^{\mu}j} + 2q_{\mathrm{h}}\mathsf{mpk}\ ,$$

for any $0 \leq q_{\mathrm{h}} v \leq 2^{\mu}$ *(larger* $q_{\mathrm{h}} v$ *cannot increase* A*'s advantage) and any* $j > 0$ *with probability at least* $1 - 1/(\exp(2^{\mu} - 2^{l_{\mathrm{h}}}(l_{\mathrm{h}} + (j-1)\ln l_{\mathrm{h}})))$ *over the choice of* H*. Furthermore,* B *makes* v *queries to its LR-oracle and its running-time is at most that of* A *plus* $O(1)$*.*

The proof is in [9].

Thus when j above is such that $j \ll (2^{\mu - l_{\mathrm{h}}} - l_{\mathrm{h}})/\ln(l_{\mathrm{h}}) + 1$, the given bound on $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EaH},A}$ holds with probability extremely close to one. This means that the analysis is only meaningful when μ is large enough relative to l_{h} (say by at least a few bits) for the right-hand side of this inequality to be sigificantly greater than 1, reflecting the fact that, if μ and l_{h} are the same size, bucketization is unlikely to have much effect on security because the minimum-bucket size is likely to be very small (again, with probability taken over the choice of the RO in the scheme). Precise bounds can be obtained for a specific application by plugging in the appropriate values and checking at what value a greater-or-equal minimum bucket-size j becomes overwhelmingly likely, in which case one can use the bound with such a j. We provide an example in the full paper [9]. Note that the trade-off as the hash length is decreased is query-efficiency. On each query, all records whose specified attributes values are in the same buckets as those of the desired result are returned to the user, who can complete the query itself by filtering out false-positives as needed.

We remark that one cannot use a POWHF [20, 21] to compute the tags in place of the RO in the construction, because POWHFs are randomized and this will violate the consistency requirement of ESE.

6 CCA and Other Extensions

Our definition, and so far our security proofs, are for the CPA case. Here we discuss extensions to the CCA case and then other extensions such as to hash functions rather than encryption schemes.

PRIV-CCA. Extend $\mathbf{Exp}^{\text{priv-b}}_{\Pi,A}(k)$ to give A_{g} oracle access to $\mathcal{D}(sk,\cdot)$, for $b \in \{0,1\}$, which it can query on any string not appearing as a component of $\mathbf{c}$. Note that A_{m} does *not* get this decryption oracle. Let

$$\mathbf{Adv}^{\text{priv-cca}}_{\Pi,A}(k) = \Pr\left[\,\mathbf{Exp}^{\text{priv-0}}_{\Pi,A}(k) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{priv-1}}_{\Pi,A}(k) = 1\,\right] .$$

We say that Π is *PRIV-CCA secure* if $\mathbf{Adv}^{\text{priv-cca}}_{\Pi,A}(\cdot)$ is negligible for every PTA A with high min-entropy.

ENCRYPT-WITH-HASH. Deterministic encryption scheme EwH is PRIV-CCA secure even if the starting encryption scheme is only IND-*CPA* but meets an extra condition, namely that no ciphertext occurs with too high a probability. More precisely, the *max-ciphertext probability* $\mathsf{mc}(\cdot)$ of $\mathsf{AE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ is defined as follows: we let $\mathsf{mc}(k)$ be the maximum taken over all $x \in \mathrm{PtSp}(k)$ of the quantity

$$\Pr\left[\,(pk,sk) \xleftarrow{\$} \mathcal{K}\,;\; c_1, c_2 \xleftarrow{\$} \mathcal{E}(pk,x) \;:\; c_1 = c_2\,\right] .$$

Then Theorem 1 extends as follows.

Theorem 5. *Suppose there is a PRIV-CCA adversary $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ against* EwH *with min-entropy μ, which outputs vectors of size v with components of length n and makes at most q_{h} queries to its hash oracle and at most q_{d} queries to its decryption oracle. Let* mpk *and* mc *be max public-key and max-ciphertext probabilities of* AE*, respectively. Then there exists an IND-CPA adversary B against* AE *such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EwH},A} \le \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathsf{AE},B} + \frac{2q_{\mathrm{h}}v}{2^{\mu}} + 2q_{\mathrm{h}}\mathsf{mpk} + 2q_{\mathrm{d}}\mathsf{mc} .$$

Furthermore, B makes v queries to its LR-oracle and its running-time is at most that of A plus $O(q_{\mathrm{h}}(T_{\mathcal{E}} + q_{\mathrm{d}}))$, where $T_{\mathcal{E}}$ is the maximum time for one computation of $\mathcal{E}$ on a message of length n.

The proof is given in [9].

The requirement that $\mathsf{mc}(\cdot)$ be small is quite mild. Most practical encryption schemes have negligible max-ciphertext probability. Furthermore, any IND-CPA scheme can be easily modified to have low max-ciphertext probability if does not already. In the full paper [9], we detail all this and also show by example that not all IND-CPA schemes have low max-ciphertext probability.

RSA-DOAEP. Although $\mathsf{RSA\text{-}DOAEP}$ as a stand-alone is demonstrably PRIV-CCA insecure, when properly combined in an "encrypt-then-sign" fashion with a secure digital signature scheme it achieves CCA security in the natural "outsider security" model analogous to that in [5]. This may come at no additional cost, for example in the database-security application we discussed, which also requires authenticity anyway.

Extensions to Other Primitives. It is straightforward to adapt our PRIV definition to a more general primitive that we call a (public-key) *hiding scheme*, which we define as a pair $\mathsf{HIDE} = (\mathsf{Kg}, \mathsf{F})$ of algorithms, where Kg outputs a key K and F takes K and an input x to return an output we call the ciphertext. Note that every public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ has an associated hiding scheme where Kg runs $\mathcal{K}$ to receive output (pk, sk) and then returns pk, and $\mathsf{F}(K, \cdot)$ is the same as $\mathcal{E}(pk, \cdot)$. In general, though, a hiding scheme is not required to be invertible, covering for example the case of hash functions.

Encrypt-and-Hash. In contrast to Encrypt-with-Hash, PRIV-CCA security of ESE scheme EaH requires IND-CCA security of the starting encryption scheme AE in general, in which case the analogous statements to Theorem 3 and Theorem 4 holds when considering PRIV-CCA adversaries against EaH. These are stated and proven in [9].

In fact, the basic construction generalizes to using any deterministic hiding scheme $\mathsf{HIDE} = (\mathsf{Kg}, \mathsf{F})$ as defined above in place of the RO, where we replace a query $H(pk, x)$ in the scheme by $\mathsf{F}(K, (pk, x))$. Theorem 3 then generalizes as follows.

Theorem 6. *Suppose there is a privacy adversary* $A = (A_{\mathrm{m}}, A_{\mathrm{g}})$ *against* EaH *that outputs vectors of size* v. *Let* mpk *be the max public-key probability of* AE. *Then there exists an IND-CPA adversary* B *against* AE *and a privacy adversary* B' *against* HIDE *such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{EaH},A} \leq \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathsf{AE},B} + \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{HIDE},B'} .$$

Furthermore, B *makes* v *queries to its LR-oracle,* B' *outputs vectors of length* v *with components of length* n, *and the running-times of* B, B' *are at most that of* A *plus* $O(1)$.

Again, the proof is in [9]. Note that in the RO model it is easy to construct a PRIV secure deterministic hiding scheme $(\mathsf{Kg}, \mathsf{F})$, simply by setting Kg to output nothing and F on input x to return $H(x)$, where H is a RO. In this case, we recover the original construction.

7 Acknowledgments

We would like to thank Brian Cooper and Cynthia Dwork for helpful discussions, and Alex Dent and Ulrich Kühn for feedback on an early draft of this paper. Thanks also to Diana Smetters and Dan Wallach for pointing us to the work of [2, 29], and to Nisheeth Vishnoi for help in formulating and proving Theorem 4. Finally, we thank the anonymous reviewers of Crypto 2007 for their comments and suggestions. Mihir Bellare was supported by NSF grants CNS-0524765, CNS-0627779, and a gift from Intel Corporation. Alexandra Boldyreva was supported in part by NSF CAREER award 0545659. Adam O'Neill was supported in part by the above-mentioned grant of the second author.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In V. Shoup, editor, *Crypto '05*, volume 3621 of *LNCS*. Springer, 2005.
2. A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In *Symposium on Operating System Design and Implementation (OSDI '02)*. Springer, 2002.
3. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *SIGMOD '04*. ACM, 2004.
4. G. Amanatidis, A. Boldyreva, and A. O'Neill. New security models and provably-secure schemes for basic query support in outsourced databases. In *Working Conference on Data and Applications Security (DBSec '07)*, LNCS. Springer, 2007.
5. J.-H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In L. Knudsen, editor, *EUROCRYPT '02*, volume 2332 of *LNCS*. Springer, 2002.
6. J. Baek, R. Safavi-Naini, and W. Susilo. Public key encryption with keyword search revisited. *Cryptology ePrint Archive, Report 2005/151*, 2005.

7. O. Baudron, D. Pointcheval, and J. Stern. Extended notions of security for multicast public key cryptosystems. In *ICALP '00*, volume 1853 of *LNCS*. Springer, 2000.
8. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT '00*, volume 1807 of *LNCS*. Springer, 2000.
9. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. Full Version of this paper, 2007. `http://www.cc.gatech.edu/~aboldyre/publications.html`.
10. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *FOCS '97*, pages 394–403, 1997.
11. M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in SSH: provably fixing the SSH binary packet protocol. In *Conference on Computer and Communications Security (CCS '02)*. ACM, 2002.
12. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Conference on Computer and Communications Security (CCS '93)*. ACM, 1993.
13. M. Bellare and P. Rogaway. Optimal asymmetric encryption – how to encrypt with RSA. In A. De Santis, editor, *Eurocrypt '94*, volume 950. Springer, 1995.
14. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In M. Franklin, editor, *Crypto '04*, volume 3027 of *LNCS*. Springer, 2004.
15. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *EUROCRYPT '04*, volume 3027 of *LNCS*. Springer, 2004.
16. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data, 2007.
17. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *CRYPTO '06*, volume 4117 of *LNCS*. Springer, 2006.
18. V. Boyko. On the security properties of OAEP as an all-or-nothing transform. In M. J. Wiener, editor, *CRYPTO '99*, volume 1666 of *LNCS*. Springer, 1999.
19. R. Brinkman, L. Feng, J. M. Doumen, P. H. Hartel, and W. Jonker. Efficient tree search in encrypted data. Technical Report TR-CTIT-04-15, Enschede, March 2004.
20. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. Kaliski, editor, *CRYPTO '97*, volume 1294 of *LNCS*. Springer, 1997.
21. R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions. In *STOC '98*. ACM, 1998.
22. A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Modeling and assessing inference exposure in encrypted databases. *ACM Trans. Inf. Syst. Secur.*, 8(1):119–152, 2005.
23. Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In J. Ioannidis, A. D. Keromytis, and M. Yung, editors, *ACNS '05*, volume 3531 of *LNCS*, 2005.
24. R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *Conference on Computer and Communications Security (CCS '06)*. ACM, 2006.

25. E. Damiani, S. De Capitani Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing confidentiality and efficiency in untrusted relational DBMSs. In S.Jajodia, V. Atluri, and T. Jaeger, editors, *Conference on Computer and Communications Security (CCS '03)*. ACM, 2003.
26. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):1130–1140, 1976.
27. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In J. Kilian, editor, *TCC '05*, volume 3378 of *LNCS*. Springer, 2005.
28. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2), 2000.
29. J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *Conference on Distributed Computing Systems (ICDCS'02)*, 2002.
30. T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, Vol. 31, 1985.
31. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. In J. Kilian, editor, *CRYPTO '01*, volume 2139 of *LNCS*. Springer, 2001.
32. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS '01*. IEEE, 2001.
33. E.-J. Goh. Secure indexes. Cryptology ePrint Archive, Report 2003/216, 2003. `http://eprint.iacr.org/2003/216/`.
34. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2), 1984.
35. P. Golle, J. Staddon, and B. Waters. Secure conjunctive keyword search over encrypted data. In M. Jakobsson, M. Yung, and J. Zhou, editors, *Applied Cryptography and Network Security Conference (ACNS '04)*, volume 3089 of *LNCS*. Springer, 2004.
36. H. Hacigümüs, B. Iyer, C. Li, and S. Mehrotra. Executing SQL over encrypted data in the database-service-provider model. In *Conference on Management of data (SIGMOD '02)*. ACM, 2002.
37. H. Hacigümüs, B. R. Iyer, and S. Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In Y. Lee, J. Li, K.-Y. Whang, and D. Lee, editors, *DASFAA '04*, volume 2973 of *LNCS*. Springer, 2004.
38. B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In M. A. Nascimento, M. Tamer Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, editors, *VLDB '04*. Morgan Kaufmann, 2004.
39. B. R. Iyer, S. Mehrotra, E. Mykletun, G. Tsudik, and Y. Wu. A framework for efficient storage security in RDBMS. In E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, editors, *EDBT '04*, volume 2992 of *LNCS*. Springer, 2004.
40. M. Kantracioglu and C. Clifton. Security issues in querying encrypted data. In *Working Conference on Data and Applications Security (DBSec '05)*, LNCS. Springer, 2005.
41. J. Li and E. Omiecinski. Efficiency and security trade-off in supporting range queries on encrypted databases. In *Working Conference on Data and Applications Security (DBSec '05)*, LNCS. Springer, 2005.
42. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2), 1988.
43. S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, 1988.

44. E. Mykletun and G. Tsudik. Aggregation queries in the database-as-a-service model. In *Working Conference on Data and Applications Security (DBSec '06)*, LNCS. Springer, 2006.
45. G. Özsoyoglu, D. A. Singer, and S. S. Chung. Anti-tamper databases: Querying encrypted databases. In *Working Conference on Data and Applications Security (DBSec '03)*, LNCS. Springer, 2003.
46. D. Pointcheval. How to encrypt properly with RSA. *RSA Laboratories' CryptoBytes*, 5(1), Winter/Spring 2002.
47. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO '91*, volume 576 of *LNCS*. Springer, 1992.
48. A. Russell and H. Wang. How to fool an unbounded adversary with a short key. *IEEE Transactions on Information Theory*, 52(3):1130–1140, 2006.
49. Arsenal Digital Solutions. Top 10 reasons to outsource remote data protection. `http://www.arsenaldigital.com/services/remote_data_protection.htm`.
50. D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Symposium on Security and Privacy*. IEEE, 2000.
51. H. Wang and L.V.S. Lakshmanan. Efficient secure query evaluation over encrypted XML databases. In *VLDB '06*. VLDB Endowment, 2006.
52. B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*. Springer, 2005.