Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements

Mihir Bellare[1], Alexandra Boldyreva[1], and Silvio Micali[2]

[1] Dept. of Computer Science & Engineering, University of California at San Diego,
9500 Gilman Drive, La Jolla, California 92093, USA.
E-mails: {mihir, aboldyre}@cs.ucsd.edu.
URLs: www-cse.ucsd.edu/users/{mihir, aboldyre}.
[2] MIT Laboratory for Computer Science,
545 Technology Square, Cambridge MA 02139, USA.

Abstract. This paper addresses the security of public-key cryptosystems in a "multi-user" setting, namely in the presence of attacks involving the encryption of related messages under different public keys, as exemplified by Håstad's classical attacks on RSA. We prove that security in the single-user setting implies security in the multi-user setting as long as the former is interpreted in the strong sense of "indistinguishability," thereby pin-pointing many schemes guaranteed to be secure against Håstad-type attacks. We then highlight the importance, in practice, of considering and improving the concrete security of the general reduction, and present such improvements for two Diffie-Hellman based schemes, namely El Gamal and Cramer-Shoup.

1 Introduction

TWO SETTINGS. The setting of public-key cryptography is usually presented like this: there is a receiver R, possession of whose public key pk enables anyone to form ciphertexts which the receiver can decrypt using the secret key associated to pk. This *single-user setting* —so called because it considers a single recipient of encrypted data— is the one of formalizations such as indistinguishability and semantic security [9]. Yet it ignores an important dimension of the problem: in the real world there are many users, each with a public key, sending each other encrypted data. Attacks presented in the early days of public-key cryptography had highlighted the presence of security threats in this *multi-user setting* that were not present in the single-user setting, arising from the possibility that a sender might encrypt, under different public keys, plaintexts which although unknown to the attacker, satisfy some known relation to each other.

HÅSTAD'S ATTACKS. An example of the threats posed by encrypting related messages under different public keys is provided by Håstad's well-known attacks on the basic RSA cryptosystem [10].[1] Suppose we have many users where the

[1] As Håstad points out, the simple version of the attack discussed here was discovered by Blum and others before his work. His own paper considers extensions of the attack

public key of user U_i is an RSA modulus N_i and (for efficiency) all users use encryption exponent $e = 3$. Given a single ciphertext $y_i = m^3 \bmod N_i$, the commonly accepted one-wayness of the RSA function implies that it is computationally infeasible for an adversary to recover the plaintext m. However, suppose now that a sender wants to securely transmit the same plaintext m to three different users, and does so by encrypting m under their respective public keys, producing ciphertexts y_1, y_2, y_3 where $y_i = m^3 \bmod N_i$ for $i = 1, 2, 3$. Then an adversary given y_1, y_2, y_3 can recover m. (Using the fact that N_1, N_2, N_3 are relatively prime, y_1, y_2, y_3 can be combined by Chinese remaindering to yield $m^3 \bmod N_1N_2N_3$. But $m^3 < N_1N_2N_3$ so m can now be recovered.)

Several counter-measures have been proposed, e.g. padding the message with random bits. The benefit of such measures is, however, unclear in that although they appear to thwart the specific known attacks, we have no guarantee of security against other similar attacks.

A GENERAL REDUCTION. The first and most basic question to address is whether it is possible to prove security against the kinds of attacks discussed above, and if so how and for which schemes. This question turns out to have a simple answer: the schemes permitting security proofs in the multi-user setting are exactly those permitting security proofs in the single-user setting, as long as we use "strong-enough" notions of security in the two cases. What is "strong-enough"? Merely having the property that it is hard to recover the plaintext from a ciphertext is certainly not: basic RSA has this property, yet Håstad's attacks discussed above show it is not secure in the multi-user setting. Theorem 1 interprets "strong enough" for the single-user setting in the natural way: secure in the sense of indistinguishability of Goldwasser and Micali [9]. As to the multi-user setting, the notion used in the theorem is an appropriate extension of indistinguishability that takes into account the presence of multiple users and the possibility of an adversary seeing encryptions of related messages under different public keys. We prove the general reduction for security both under chosen-plaintext attack and chosen-ciphertext attack, in the sense that security under either type of attack in one setting implies security under the same type of attack in the other setting. (The analogous statement can be shown with regard to non-malleability [7] under chosen-plaintext attack, and a simple way to extend our proof to that setting is to exploit the characterization of [5].)

We view ourselves here as establishing what most theoreticians would have "expected" to be true. The proof is indeed simple, yet validating the prevailing intuition has several important elements and fruits beyond the obvious one of filling a gap in the literature, as we now discuss.

IMMEDIATE CONSEQUENCES. The above-mentioned results directly imply security guarantees in the multi-user setting for all schemes proven to meet the notion of indistinguishability, under the same assumptions that were used to establish indistinguishability. This includes practical schemes secure against chosen-

using lattice reduction [10]. For simplicity we will continue to use the term "Håstad's attack(s)" to refer to this body of cryptanalysis.

plaintext attack [8], against chosen-ciphertext attack [6], and against chosen-ciphertext attack in the random oracle model [4, 12].

These results confirm the value of using strong, well-defined notions of security and help to emphasize this issue in practice. As we have seen, designers attempt to thwart Håstad-type attacks by specific counter-measures. Now we can say that the more productive route is to stick to schemes meeting notions of security such as indistinguishability. Designers are saved the trouble of explicitly considering attacks in the multi-user setting.

THE MODEL. The result requires, as mentioned above, the introduction of a new model and notion. We want to capture the possibility of an adversary seeing encryptions of related messages under different keys when the choice of the relation can be made by the adversary. To do this effectively and elegantly turns out to need some new definitional ideas. Very briefly —see Section 3 for a full discussion and formalization— the formalization introduces the idea of an adversary given (all public keys and) a list of "challenge encryption oracles," one per user, each oracle capable of encrypting one of two given equal-length messages, the choice of which being made according to a bit that *although hidden from the adversary is the same for all oracles.*[2] This obviates the need to *explicitly* consider relations amongst messages. This model is important because its use extends beyond Theorem 1, as we will see below.

ISN'T SIMULATION ENOUGH? It may appear at first glance that the implication (security in the single-user setting implies security in the multi-user setting for strong-enough notions of security) is true for a trivial reason: an adversary attacking one user can just simulate the other users, itself picking their public keys so that it knows the corresponding secret keys. This doesn't work, and misses the key element of the multi-user setting. Our concern is an adversary that sees ciphertexts of related messages under different keys. Given a challenge ciphertext of an unknown message under a target public key, a simulator cannot produce a ciphertext of a related message under a different public key, even if it knows the secret key corresponding to the second public key, because it does not know the original message. Indeed, our proof does not proceed by this type of simulation.

THE NEED FOR CONCRETE SECURITY IMPROVEMENTS. Perhaps the most important impact of the general reduction of Theorem 1 is the manner in which it leads us to see the practical importance of concrete security issues and improvements for the multi-user setting.

Suppose we have a system of n users in which each user encrypts up to q_e messages. We fix a public-key cryptosystem $\mathcal{PE}$ used by all users. Theorem 1 says that the maximum probability that an adversary with running time t can

[2] An encryption oracle is used in definitions of security for private-key encryption [3] because there the encryption key is secret, meaning not given to the adversary. One might imagine that oracles performing encryption are unnecessary in the public-key case because the adversary knows the public keys: can't it just encrypt on its own? Not when the message in question is a challenge one which it doesn't know, as in our setting.

compromise security in the multi-user setting —this in the sense of our definition discussed above— is at most $q_e n$ times the maximum probability that an adversary with running time closely related to t can compromise security in the standard sense of indistinguishability. Notationally, $\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(t, q_e) \leq q_e n \cdot \mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE},I}(t')$ where $t' \approx t$. (Here I represents any possible information common to all users and should be ignored at a first reading, and the technical term for the "maximum breaking probabilities" represented by the notation is "advantage".) It follows that if any poly-time adversary has negligible success probability in the single-user setting, the same is true in the multi-user setting. This corollary is what we have interpreted above as saying that "the schemes secure in the single-user setting are exactly those secure in the multi-user setting". However, what this theorem highlights is that the advantage in the multi-user setting may be more than that in the single-user setting by a factor of $q_e n$. Security can degrade linearly as we add more users to the system and also as the users encrypt more data. The practical impact of this is considerable, and in the full version of this work [2] we illustrate this with some numerical examples that are omitted here due to lack of space.

We prove in Proposition 1 that there is no general reduction better than ours: if there is any secure scheme, there is also one whose advantage in the two settings provably differs by a factor of $q_e n$. So we can't expect to reduce the security loss in general. But we can still hope that there are *specific* schemes for which the security degrades less quickly as we add more users to the system. These schemes become attractive in practice because for a fixed level of security they have lower computational cost than schemes not permitting such improved reductions. We next point to two popular schemes for which we can provide new security reductions illustrating such improvements.

El Gamal. The El Gamal scheme in a group of prime order can be proven to have the property of indistinguishability under chosen-plaintext attack (in the single-user setting) under the assumption that the decision Diffie-Hellman (DDH) problem is hard. (This simple observation is made for example in [11, 6]). The reduction is essentially tight, meaning that the maximum probability that an adversary of time-complexity t can compromise the security of the El Gamal scheme in the single-user setting is within a constant factor of the probability of solving the DDH problem in comparable time. Theorem 1 then implies that the maximum probability of breaking the El Gamal scheme under chosen-plaintext attack in the presence of n users each encrypting q_e messages is bounded by $2q_e n$ times the probability of solving the DDH problem in comparable time. We show in Theorem 2 that via an improved reduction the factor of $q_e n$ can be essentially eliminated. In other words, the maximum probability of breaking the El Gamal scheme under chosen-plaintext attack, even in the presence of n users each encrypting q_e messages, remains tightly related to the probability of solving the DDH problem in comparable time.

Our reduction exploits a self-reducibility property of the decisional Diffie-Hellman problem due to Stadler and Naor-Reingold [15, 11], and a variant thereof that was also independently noted by Shoup [14]. See Lemma 1.

CRAMER-SHOUP. The Cramer-Shoup scheme [6] is shown to achieve indistinguishability under chosen-ciphertext attack (in the single-user setting) assuming the DDH problem is hard. Their reduction of the security of their scheme to that of the DDH problem is essentially tight. Applying our general result to bound the advantage in the multi-user setting would indicate degradation of security by a factor of $q_e n$. We present in Theorem 3 an improved reduction which (roughly speaking) reduces the factor of $q_e n$ to a factor of q_e only. Thus the maximum probability of breaking the Cramer-Shoup scheme under chosen-ciphertext attack, in the presence of n users, each encrypting q_e messages, is about the same as is proved if there was only one user encrypting q_e messages. (The result is not as strong as for El Gamal because we have not eliminated the factor of q_e, but this is an open problem even when there is only one user.) This new result exploits Lemma 1 and features of the proof of security for the single-user case given in [6].

DISCUSSION AND RELATED WORK. A special case of interest in these results is when $n = 1$. Meaning we are back in the single-user setting, but are looking at an extension of the notion of indistinguishability in which one considers the encryption of up to q_e messages. Our results provide improved security for the El Gamal scheme in this setting.

The questions raised here can also be raised in the private-key setting: what happens there when there are many users? The ideas of the current work are easily transfered. The definitions of [3] for the single-user case can be adapted to the multi-user case using the ideas in Section 3. The analogue of Theorem 1 for the private-key setting is then easily proven.

Baudron, Pointcheval and Stern have independently considered the problem of public-key encryption in the multi-user setting [1]. Their notion of security for the multi-user setting —also proved to be polynomially-equivalent to the standard notion of single-user indistinguishability— is slightly different from ours. They do not consider concrete-security or any specific schemes. (The difference in the notions is that they do not use the idea of encryption oracles; rather, their adversary must output a pair of vectors of plaintexts and get back as challenge a corresponding vector of ciphertexts. This makes their model weaker since the adversary does not have adaptive power. If only polynomial-security is considered, their notion, ours and the single-user one are all equivalent, but when concrete security is considered, our notion is stronger.)

2 Definitions

We specify a concrete-security version of the standard notion of security of a public-key encryption scheme in the sense of indistinguishability. We consider both chosen-plaintext and chosen-ciphertext attacks.

First recall that a *public-key encryption scheme* $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms. The *key generation* algorithm $\mathcal{K}$ is a randomized algorithm that takes as input some global information I and returns a pair (pk, sk) of keys, the public key and matching secret key, respectively; we write $(pk, sk) \stackrel{R}{\leftarrow}$

$\mathcal{K}(I)$. (Here I includes a security parameter, and perhaps other information. For example in a Diffie-Hellman based scheme, I might include a global prime number and generator of a group which all parties use to create their keys.) The *encryption* algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a *plaintext* M to return a *ciphertext* C; we write $C \xleftarrow{R} \mathcal{E}_{pk}(M)$. The *decryption* algorithm $\mathcal{D}$ is a deterministic algorithm that takes the secret key sk and a ciphertext C to return the corresponding plaintext M; we write $M \leftarrow \mathcal{D}_{sk}(C)$. Associated to each public key pk is a *message space* $\mathrm{MsgSp}(pk)$ from which M is allowed to be drawn. We require that $\mathcal{D}_{sk}(\mathcal{E}_{pk}(M)) = M$ for all $M \in \mathrm{MsgSp}(pk)$.

An adversary B runs in two stages. In the "find" stage it takes the public key and outputs two equal length messages m_0, m_1 together with some state information s. In the "guess" stage it gets a challenge ciphertext C formed by encrypting a random one of the two messages, and must say which message was chosen. Below the superscript of "1" indicates that we are in the single-user setting, meaning that although there may be many senders, only one person holds a public key and is the recipient of encrypted information. In the case of a chosen-ciphertext attack the adversary gets an oracle for $\mathcal{D}_{sk}(\cdot)$ and is allowed to invoke it on any point with the restriction of not querying the challenge ciphertext during the guess stage [13].

Definition 1. [Indistinguishability of encryptions] Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let $B_{\mathrm{cpa}}, B_{\mathrm{cca}}$ be adversaries where the latter has access to an oracle. Let I be some initial information string. For $b = 0, 1$ define the experiments

Experiment $\mathbf{Exp}^{\text{1-cpa}}_{\mathcal{PE},I}(B_{\mathrm{cpa}}, b)$	**Experiment** $\mathbf{Exp}^{\text{1-cca}}_{\mathcal{PE},I}(B_{\mathrm{cca}}, b)$
$(pk, sk) \leftarrow \mathcal{K}(I)$	$(pk, sk) \leftarrow \mathcal{K}(I)$
$(m_0, m_1, s) \leftarrow B_{\mathrm{cpa}}(\mathsf{find}, I, pk)$	$(m_0, m_1, s) \leftarrow B^{\mathcal{D}_{sk}(\cdot)}_{\mathrm{cca}}(\mathsf{find}, I, pk)$
$C \xleftarrow{R} \mathcal{E}_{pk}(m_b)$	$C \xleftarrow{R} \mathcal{E}_{pk}(m_b)$
$d \leftarrow B_{\mathrm{cpa}}(\mathsf{guess}, C, s)$	$d \leftarrow B^{\mathcal{D}_{sk}(\cdot)}_{\mathrm{cca}}(\mathsf{guess}, C, s)$
`Return` d	`Return` d

It is mandated that $|m_0| = |m_1|$ above. We require that B_{cca} not make oracle query C in the guess stage. We define the *advantage* of B_{cpa} and B_{cca}, respectively, as follows:

$$\mathsf{Adv}^{\text{1-cpa}}_{\mathcal{PE},I}(B_{\mathrm{cpa}}) = \Pr\left[\,\mathbf{Exp}^{\text{1-cpa}}_{\mathcal{PE},I}(B_{\mathrm{cpa}}, 0) = 0\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{1-cpa}}_{\mathcal{PE},I}(B_{\mathrm{cpa}}, 1) = 0\,\right]$$
$$\mathsf{Adv}^{\text{1-cca}}_{\mathcal{PE},I}(B_{\mathrm{cca}}) = \Pr\left[\,\mathbf{Exp}^{\text{1-cca}}_{\mathcal{PE},I}(B_{\mathrm{cca}}, 0) = 0\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{1-cca}}_{\mathcal{PE},I}(B_{\mathrm{cca}}, 1) = 0\,\right] .$$

We define the *advantage function of the scheme for privacy under chosen-plaintext (resp. chosen-ciphertext) attacks in the single-user setting* as follows. For any t, q_{d}, let

$$\mathsf{Adv}^{\text{1-cpa}}_{\mathcal{PE},I}(t) = \max_{B_{\mathrm{cpa}}} \{\, \mathsf{Adv}^{\text{1-cpa}}_{\mathcal{PE},I}(B_{\mathrm{cpa}}) \,\}$$
$$\mathsf{Adv}^{\text{1-cca}}_{\mathcal{PE},I}(t, q_{\mathrm{d}}) = \max_{B_{\mathrm{cca}}} \{\, \mathsf{Adv}^{\text{1-cca}}_{\mathcal{PE},I}(B_{\mathrm{cca}}) \,\}$$

where the maximum is over all $B_{\text{cpa}}, B_{\text{cca}}$ with "time-complexity" t, and, in the case of B_{cca}, also making at most q_{d} queries to the $\mathcal{D}_{sk}(\cdot)$ oracle. ▮

The "time-complexity" is the worst case execution time of the associated experiment plus the size of the code of the adversary, in some fixed RAM model of computation. (Note that the execution time refers to the entire experiment, not just the adversary. In particular, it includes the time for key generation, challenge generation, and computation of responses to oracle queries if any.) The same convention is used for all other definitions in this paper and will not be explicitly mentioned again. The advantage function is the maximum likelihood of the security of the encryption scheme $\mathcal{PE}$ being compromised by an adversary, using the indicated resources, and with respect to the indicated measure of security.

Definition 2. We say that $\mathcal{PE}$ is *polynomially-secure against chosen-plaintext attack (resp. chosen-ciphertext attack) in the single-user setting* if $\mathsf{Adv}^{\text{1-cpa}}_{\mathcal{PE},I}(B)$ (resp. $\mathsf{Adv}^{\text{1-cca}}_{\mathcal{PE},I}(B)$) is negligible for any probabilistic, poly-time adversary B.

Here complexity is measured as a function of a security parameter that is contained in the global input I. If I consists of more than a security parameter (as in the El Gamal scheme), we fix a probabilistic generator for this information and the probability includes the choices of this generator.

3 Security in the multi-user setting

We envision a set of n users. All users use a common, fixed cryptosystem $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. User i has a public key pk_i and holds the matching secret key sk_i. It is assumed that each user has an authentic copy of the public keys of all other users.

As with any model for security we need to consider attacks (what the adversary is allowed to do) and success measures (when is the adversary considered successful). The adversary is given the global information I and also the public keys of all users. The main novel concern is that the attack model must capture the possibility of an adversary obtaining encryptions of related messages under different keys. To have a strong notion of security, we will allow the adversary to choose how the messages are related, and under which keys they are encrypted. For simplicity we first address chosen-plaintext attacks only.

Some intuition. To get a start on the modeling, consider the following game. We imagine that a message m is chosen at random from some known distribution, and the adversary is provided with $\mathcal{E}_{pk_1}(m)$, a ciphertext of m under the public key of user 1. The adversary's job is to compute some partial information about m. To do this, it may, for example, like to see an encryption of m under pk_3. We allow it to ask for such an encryption. More generally, it may want to see an encryption of the bitwise complement of m under yet another key, or perhaps the encryption of an even more complex function of m. We could capture this

by allowing the adversary to specify a polynomial-time "message modification function" Δ and a user index j, and obtain in response $\mathcal{E}_{pk_j}(\Delta(m))$, a ciphertext of the result of applying the modification function to the challenge message. After many such queries, the adversary must output a guess of some partial information about m and wins if it can do this with non-trivial advantage. Appropriately generalized, these ideas can be used to produce a semantic-security type notion of security for the multi-user setting, but, as should be evident even from our brief discussion here, it would be relatively complex. We prefer an indistinguishability version because it is simpler and extends more easily to a concrete security setting. It is nonetheless useful to discuss the semantic security setting because here we model the attacks in which we are interested in a direct way that helps provide intuition.

Indistinguishability based approach. The adversary is provided with all the public keys. But unlike in the single-user indistinguishability setting of Section 2, it will not run in two phases, and there will be no single challenge ciphertext. Rather the adversary is provided with n different oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$. Oracle i takes as input any pair m_0, m_1 of messages (of equal length) and computes and returns a ciphertext $\mathcal{E}_{pk_i}(m_b)$. The challenge bit b here (obviously not explicitly given to the adversary) is chosen only once at the beginning of the experiment and *is the same across all oracles and queries*. The adversary's success is measured by its advantage in predicting b.

We suggest that this simple model in fact captures encryption of related messages under different keys; the statement in the italicized text above is crucial in this regard. The possibility of the adversary's choosing the relations between encrypted messages is captured implicitly; we do not have to worry about explicitly specifying message modification functions.

The formal definition. Formally, the *left or right selector* is the map LR defined by $\mathrm{LR}(m_0, m_1, b) = m_b$ for all equal-length strings m_0, m_1, and for any $b \in \{0, 1\}$. The adversary A is given n oracles, which we call *LR (left-or-right) encryption oracles*,

$$\mathcal{E}_{pk_1}(\mathrm{LR}(\cdot, \cdot, b)) \ , \ \ldots \ , \ \mathcal{E}_{pk_n}(\mathrm{LR}(\cdot, \cdot, b))$$

where pk_i is a public key of the encryption scheme and b is a bit whose value is unknown to the adversary. (LR oracles were first defined by [3] in the symmetric setting.) The oracle $\mathcal{E}_{pk_i}(\mathrm{LR}(\cdot, \cdot, b))$, given query (m_0, m_1) where $m_0, m_1 \in \mathrm{MsgSp}(pk_i)$ must have equal length, first sets $m_b \leftarrow \mathrm{LR}(m_0, m_1, b)$, meaning m_b is one of the two query messages, as dictated by bit b. Next the oracle encrypts m_b, setting $C \leftarrow \mathcal{E}_{pk_i}(m_b)$ and returns C as the answer to the oracle query. The adversary also gets as input the public keys and the global information I.

In the case of a chosen-ciphertext attack the adversary is also given a decryption oracle with respect to each of the n public keys. Note we must disallow a query C to $\mathcal{D}_{sk_i}(\cdot)$ if C is an output of oracle $\mathcal{E}_{pk_i}(\mathrm{LR}(\cdot, \cdot, b))$. This is necessary for meaningfulness since if such a query is allowed b is easily computed, and moreover disallowing such queries seems the least limitation we can impose,

meaning the adversary has the maximum meaningful power. Below we indicate the number n of users as a superscript.

Definition 3. Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let A_{cpa}, A_{cca} be adversaries. Both have access to $n \geq 1$ oracles, each of which takes as input any two strings of equal length, and A_{cca} has access to an additional n oracles each of which take a single input. Let I be some initial information string. For $b = 0, 1$ define the experiments:

Experiment $\mathbf{Exp}^{n\text{-cpa}}_{\mathcal{PE},I}(A_{\text{cpa}}, b)$
  `For` $i = 1, \ldots, n$ do $(pk_i, sk_i) \leftarrow \mathcal{K}(I)$ `EndFor`
  $d \leftarrow A_{\text{cpa}}^{\mathcal{E}_{pk_1}(\text{LR}(\cdot,\cdot,b)), \ldots, \mathcal{E}_{pk_n}(\text{LR}(\cdot,\cdot,b))}(I, pk_1, \ldots, pk_n)$; `Return` d

Experiment $\mathbf{Exp}^{n\text{-cca}}_{\mathcal{PE},I}(A_{\text{cca}}, b)$
  `For` $i = 1, \ldots, n$ do $(pk_i, sk_i) \leftarrow \mathcal{K}(I)$ `EndFor`
  $d \leftarrow A_{\text{cca}}^{\mathcal{E}_{pk_1}(\text{LR}(\cdot,\cdot,b)), \ldots, \mathcal{E}_{pk_n}(\text{LR}(\cdot,\cdot,b)), \mathcal{D}_{sk_1}(\cdot), \ldots, \mathcal{D}_{sk_n}(\cdot)}(I, pk_1, \ldots, pk_n)$
  `Return` d

It is mandated that a query to any LR oracle consists of two messages of *equal* length and that for each $i = 1, \ldots, n$ adversary A_{cca} does not query $\mathcal{D}_{sk_i}(\cdot)$ on an output of $\mathcal{E}_{pk_i}(\text{LR}(\cdot, \cdot, b))$ We define the *advantage* of A_{cpa}, and the *advantage* of A_{cca}, respectively, as follows:

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(A_{\text{cpa}}) = \Pr\left[\mathbf{Exp}^{n\text{-cpa}}_{\mathcal{PE},I}(A_{\text{cpa}}, 0) = 0\right] - \Pr\left[\mathbf{Exp}^{n\text{-cpa}}_{\mathcal{PE},I}(A_{\text{cpa}}, 1) = 0\right]$$
$$\mathsf{Adv}^{n\text{-cca}}_{\mathcal{PE},I}(A_{\text{cca}}) = \Pr\left[\mathbf{Exp}^{n\text{-cca}}_{\mathcal{PE},I}(A_{\text{cca}}, 0) = 0\right] - \Pr\left[\mathbf{Exp}^{n\text{-cca}}_{\mathcal{PE},I}(A_{\text{cca}}, 1) = 0\right] .$$

We define the *advantage function of the scheme for privacy under chosen- plaintext (resp. chosen-ciphertext) attacks, in the multi-user setting*, as follows. For any $t, q_{\text{e}}, q_{\text{d}}$ let

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(t, q_{\text{e}}) = \max_{A_{\text{cpa}}} \{\, \mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(A_{\text{cpa}}) \,\}$$
$$\mathsf{Adv}^{n\text{-cca}}_{\mathcal{PE},I}(t, q_{\text{e}}, q_{\text{d}}) = \max_{A_{\text{cca}}} \{\, \mathsf{Adv}^{n\text{-cca}}_{\mathcal{PE},I}(A_{\text{cca}}) \,\}$$

where the maximum is over all $A_{\text{cpa}}, A_{\text{cca}}$ with "time-complexity" t, making at most q_{e} queries to each LR oracle, and, in the case of A_{cca}, also making at most q_{d} queries to each decryption oracle. ▌

The advantage function is the maximum likelihood of the security of the symmetric encryption scheme $\mathcal{PE}$ being compromised by an adversary, using the indicated resources, and with respect to the indicated measure of security.

Remark 1. Notice that when $n = q_{\text{e}} = 1$ in Definition 3, the adversary's capability is limited to seeing a ciphertext of one of two messages of its choice under a single target key. Thus Definition 3 with $n = q_{\text{e}} = 1$ is equivalent to Definition 1. We can view Definition 3 as extending Definition 1 along two dimensions: the number of users and the number of messages encrypted by each user.

Definition 4. We say that $\mathcal{PE}$ is *polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the multi-user setting* if $\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(A)$ (resp. $\mathsf{Adv}^{n\text{-cca}}_{\mathcal{PE},I}(A)$) is negligible for any probabilistic, poly-time adversary A and polynomial n.

Again complexity is measured as a function of a security parameter that is contained in the global input I, and the latter is generated by a fixed probabilistic polynomial-time generation algorithm if necessary.

4 A general reduction and its tightness

Fix a public-key encryption scheme $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The following theorem says that the advantage of an adversary in breaking the scheme in a multi-user setting can be upper bounded by a function of the advantage of an adversary of comparable resources in breaking the scheme in the single-user setting. The factor in the bound is polynomial in the number n of users in the system and the number q_{e} of encryptions performed by each user, and the theorem is true for both chosen-plaintext attacks and chosen-ciphertext attacks. The proof of Theorem 1 is via a simple hybrid argument that is omitted here due to lack of space but can be found in the full version of this paper [2].

Theorem 1. *Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let $n, q_{\mathrm{e}}, q_{\mathrm{d}}$, t be integers and I some initial information string. Then*

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE},I}(t, q_{\mathrm{e}}) \le q_{\mathrm{e}} n \cdot \mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE},I}(t')$$
$$\mathsf{Adv}^{n\text{-cca}}_{\mathcal{PE},I}(t, q_{\mathrm{e}}, q_{\mathrm{d}}) \le q_{\mathrm{e}} n \cdot \mathsf{Adv}^{1\text{-cca}}_{\mathcal{PE},I}(t', q_{\mathrm{d}})$$

where $t' = t + O(\log(q_{\mathrm{e}} n))$. ∎

The relation between the advantages being polynomial, we obviously have the following:

Corollary 1. *Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme that is polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the single-user setting. Then $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is also polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the multi-user setting.* ∎

Tightness of the bound. We present an example that shows that in general the bound of Theorem 1 is essentially tight. Obviously such a statement is vacuous if no secure schemes exist, so first assume one does, and call it $\mathcal{PE}$. We want to modify this into another scheme $\mathcal{PE}'$ for which $\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE}',I}(t, q_{\mathrm{e}})$ is $\Omega(q_{\mathrm{e}} n)$ times $\mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE}',I}(t)$. This will be our counter-example. The following proposition does this, modulo some technicalities. In reading it, think of $\mathcal{PE}$ as being very good, so that $\mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE},I}(t)$ is essentially zero. With that interpretation we indeed have the claimed relation.

Proposition 1. *Given any public-key encryption scheme $\mathcal{PE}$ and integers n, q_e we can design another public-key encryption $\mathcal{PE}'$ such that for any I and large enough t we have*

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{PE}',I}(t, q_e) \geq 0.6 \text{ and } \mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE}',I}(t) \leq \frac{1}{q_e n} + \mathsf{Adv}^{1\text{-cpa}}_{\mathcal{PE},I}(t) \ . \ \blacksquare$$

The proof of Proposition 1 is in [2]. An analogous result holds in the chosen-ciphertext attack case, and we omit it.

5 Improved security for DH based schemes

The security of the schemes we consider is based on the hardness of the Decisional Diffie-Hellman (DDH) problem. Accordingly we begin with definitions for latter.

Definition 5. Let G be a group of a large prime order q and let g be a generator of G. Let D be an adversary that on input q, g and three elements $X, Y, K \in G$ returns a bit. We consider the experiments

Experiment $\mathbf{Exp}^{\text{ddh-real}}_{q,g}(D)$	Experiment $\mathbf{Exp}^{\text{ddh-rand}}_{q,g}(D)$
$x \xleftarrow{R} Z_q$; $X \leftarrow g^x$	$x \xleftarrow{R} Z_q$; $X \leftarrow g^x$
$y \xleftarrow{R} Z_q$; $Y \leftarrow g^y$	$y \xleftarrow{R} Z_q$; $Y \leftarrow g^y$
$K \leftarrow g^{xy}$	$K \xleftarrow{R} G$
$d \leftarrow D(q, g, X, Y, K)$	$d \leftarrow D(q, g, X, Y, K)$
`Return` d	`Return` d

The advantage of D in solving the Decisional Diffie-Hellman (DDH) problem with respect to q, g, and the advantage of the DDH with respect to q, g, are defined, respectively, by

$$\mathsf{Adv}^{\text{ddh}}_{q,g}(D) = \Pr\left[\mathbf{Exp}^{\text{ddh-real}}_{q,g}(D) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{q,g}(D) = 1\right]$$
$$\mathsf{Adv}^{\text{ddh}}_{q,g}(t) = \max_D \{\, \mathsf{Adv}^{\text{ddh}}_{q,g}(D) \,\}$$

where the maximum is over all D with "time-complexity" t. $\blacksquare$

The "time-complexity" of D is the maximum of the execution times of the two experiments $\mathbf{Exp}^{\text{ddh-real}}_{q,g}(D)$ and $\mathbf{Exp}^{\text{ddh-rand}}_{q,g}(D)$, plus the size of the code for D, all in our fixed RAM model of computation.

A common case is that G is a subgroup of order q of Z_p^* where p is a prime such that q divides $p-1$. But these days there is much interest in the use of Diffie-Hellman based encryption over elliptic curves, where G would be an appropriate elliptic curve group. Our setting is general enough to encompass both cases.

Our improvements exploit in part some self-reducibility properties of the DDH problem summarized in Lemma 1 below. The case $x \neq 0$ below is noted in [15, Proposition 1] and [11, Lemma 3.2]. The variant with $x = 0$ was noted independently in [14]. Below T^{exp}_q denotes the time needed to perform an exponentiation operation with respect to a base element in G and an exponent in Z_q, in our fixed RAM model of computation. A proof of Lemma 1 is in [2].

Lemma 1. *Let G be a group of a large prime order q and let g be a generator of G. There is a probabilistic algorithm R running in $O(T_q^{\exp})$ time such for any a, b, c, x in Z_q the algorithm takes input q, g, g^a, g^b, g^c, x and returns a triple $g^{a'}, g^{b'}, g^{c'}$ such that the properties represented by the following table are satisfied, where we read the row and column headings as conditions, and the table entries as the properties of the outputs under those conditions:*

	$x = 0$	$x \neq 0$
$c = ab \bmod q$	$a' = a$ b' *is random* $c' = a'b' \bmod q$	a' *is random* b' *is random* $c' = a'b' \bmod q$
$c \neq ab \bmod q$	$a' = a$ b' *is random* c' *is random*	a' *is random* b' *is random* c' *is random*

Here random means distributed uniformly over Z_q independently of anything else. ▌

El Gamal. As indicated above, our reduction of multi-user security to single-user security is tight in general. Here we will obtain a much better result for a specific scheme, namely the El Gamal encryption scheme over a group of prime order, by exploiting Lemma 1. We fix a group G for which the decision Diffie-Hellman problem is hard and let q (a prime) be its size. Let g be a generator of G. The prime q and the generator g comprise the global information I for the El Gamal scheme. The algorithms describing the scheme $\mathcal{EG} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ are depicted below. The message space associated to a public key (q, g, X) is the group G itself, with the understanding that all messages from G are properly encoded as strings of some common length whenever appropriate.

Algorithm $\mathcal{K}(q, g)$	Algorithm $\mathcal{E}_{q,g,X}(M)$	Algorithm $\mathcal{D}_{q,g,x}(Y, W)$
$x \xleftarrow{R} Z_q$	$y \xleftarrow{R} Z_q$	$K \leftarrow Y^x$
$X \leftarrow g^x$	$Y \leftarrow g^y$	$M \leftarrow WK^{-1}$
$pk \leftarrow (q, g, X)$	$K \leftarrow X^y$	`Return` M
$sk \leftarrow (q, g, x)$	$W \leftarrow KM$	
`Return` (pk, sk)	`Return` (Y, W)	

We noted in Section 1 that the hardness of the DDH problem implies that the El Gamal scheme meets the standard notion of indistinguishability of encryptions (cf.[11, 6]), and the reduction is essentially tight: $\mathsf{Adv}^{\text{1-cpa}}_{\mathcal{EG},(q,g)}(t)$ is at most $2\mathsf{Adv}^{\text{ddh}}_{q,g}(t)$. We want to look at the security of the El Gamal scheme in the multi-user setting. Directly applying Theorem 1 in conjunction with the above would tell us that

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{EG},(q,g)}(t, q_{\text{e}}) \leq 2q_{\text{e}}n \cdot \mathsf{Adv}^{\text{ddh}}_{q,g}(t') \tag{1}$$

where $t' = t + O(\log(q_{\text{e}}n))$. This is enough to see that polynomial security of the DDH problem implies polynomial security of El Gamal in the multi-user

setting, but we want to improve the concrete security of this relation and say that the security of the El Gamal scheme in the multi-user setting almost does not degrade with respect to the assumed hardness of the DDH problem. The following theorem states our improvement.

Theorem 2. *Let G be a group of a large prime order q and let g be a generator of the group G. Let $\mathcal{EG} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the El Gamal public-key encryption scheme associated to these parameters as described above. Let $n, q_{\rm e}, t$ be integers. Then*

$$\mathsf{Adv}^{n\text{-cpa}}_{\mathcal{EG},(q,g)}(t, q_{\rm e}) \;\le\; 2 \cdot \mathsf{Adv}^{\rm ddh}_{q,g}(t') + \frac{1}{q}$$

where $t' = t + O(q_{\rm e} n \cdot T^{\rm exp}_q)$. ▮

The $1/q$ term is negligible in practice since q is large, so the theorem is saying that the security of the encryption scheme is within a constant factor of that of the DDH problem, even where there are many users and the time-complexities are comparable.

Proof of Theorem 2: Let A be an adversary attacking the El Gamal public-key encryption scheme $\mathcal{EG}$ in the multi-user setting (cf. Definition 3). Suppose it makes at most $q_{\rm e}$ queries to each of its n oracles and has time-complexity at most t. We will design an adversary D_A for the Decisional Diffie-Hellman problem (cf. Definition 5) so that D_A has running time at most t' and

$$\mathsf{Adv}^{\rm ddh}_{q,g}(D_A) \;\ge\; \frac{1}{2} \cdot \mathsf{Adv}^{n\text{-cpa}}_{\mathcal{EG},(q,g)}(A) - \frac{1}{2q} \;. \tag{2}$$

The statement of theorem follows by taking maximums. So it remains to specify D_A. The code for D_A is presented in Figure 1. It has input q, g, and also three elements $X, Y, K \in G$. It will use adversary A as a subroutine. D_A will provide for A as input public keys $pk_1, \ldots, pk_n$ and global information q, g and will simulate for A the n LR oracles, $\mathcal{E}_{pk_i}(\mathrm{LR}(\cdot,\cdot,b))$ for $i = 1, \ldots, n$. We use the notation $A \to (i, m_0, m_1)$ to indicate that A is making query (m_0, m_1) to its i-th LR oracle, where $1 \le i \le n$ and $|m_0| = |m_1|$. We use the notation $A \leftarrow C$ to indicate that we are returning ciphertext C to A as the response to this LR oracle query. We are letting R denote the algorithm of Lemma 1.

An analysis of this algorithm —which is omitted here due to lack of space but can be found in [2]— shows that

$$\Pr\left[\, \mathbf{Exp}^{\rm ddh\text{-}real}_{q,g}(D) = 1 \,\right] = \frac{1}{2} + \frac{1}{2} \cdot \mathsf{Adv}^{n\text{-cpa}}_{\mathcal{EG},(q,g)}(A) \;. \tag{3}$$

and

$$\Pr\left[\, \mathbf{Exp}^{\rm ddh\text{-}rand}_{q,g}(D) = 1 \,\right] \;\le\; \frac{1}{2} \cdot \left(1 - \frac{1}{q}\right) + \frac{1}{q} \;=\; \frac{1}{2} + \frac{1}{2q} \;. \tag{4}$$

Subtracting Equations (3) and (4) we get Equation (2). □

Cramer-Shoup. Now we consider another specific scheme, namely the practical public-key cryptosystem proposed by Cramer and Shoup, which is secure against chosen-ciphertext attack in the single-user setting as shown in [6]. We

Adversary $D_A(q, g, X, Y, K)$
  $b \xleftarrow{R} \{0,1\}$
  `For` $i = 1, \ldots, n$ `do`
    $(X'_i[1], Y'_i[1], K'_i[1]) \leftarrow R(q, g, X, Y, K, 1)$; $pk_i \leftarrow (q, g, X'_i[1])$; $ctr_i \leftarrow 0$
    `For` $j = 2, \ldots, q_e$ `do`
      $(X'_i[j], Y'_i[j], K'_i[j]) \leftarrow R(q, g, X'_i[1], Y'_i[1], K'_i[1], 0)$
    `EndFor`
  `EndFor`
  Run A replying to oracle queries as follows:
    $A \rightarrow (i, m_0, m_1)$ $[1 \le i \le n$ and $m_0, m_1 \in G]$
    $ctr_i \leftarrow ctr_i + 1$; $W_i \leftarrow K'_i[ctr_i] \cdot m_b$
    $A \leftarrow (Y'_i[ctr_i], W_i[ctr_i])$
  Eventually A halts and outputs a bit d
  `If` $b = d$ `then return 1 else return 0`

Fig. 1. *Distinguisher D_A in proof of Theorem 2, where R is the algorithm of Lemma 1.*

are interested in the security of this scheme (against chosen-ciphertext attack) in the multi-user setting. Let us define the basic scheme. Let G be a group of a large prime order q and let g be a generator of G. The prime q and the generator g comprise the global information I for the scheme. Let $\mathcal{H}$ be a family of collision-resistant hash functions, each member of which maps strings of arbitrary length to the elements of Z_q. The message space is the group G. The algorithms describing the scheme $\mathcal{CS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ are defined as follows:

Algorithm $\mathcal{K}(q, g)$
  $g_1 \leftarrow g$; $g_2 \xleftarrow{R} G$
  $H \xleftarrow{R} \mathcal{H}$
  $x_1, x_2, y_1, y_2, z \xleftarrow{R} Z_q$
  $c \leftarrow g_1^{x_1} g_2^{x_2}$
  $d \leftarrow g_1^{y_1} g_2^{y_2}$
  $h \leftarrow g_1^{z}$
  $pk \leftarrow (g_1, g_2, c, d, h)$
  $sk \leftarrow (x_1, x_2, y_1, y_2, z)$
  `Return` (pk, sk)

Algorithm $\mathcal{E}_{pk}(M)$
  $r \xleftarrow{R} Z_q$
  $u_1 \leftarrow g_1^r$
  $u_2 \leftarrow g_2^r$
  $e \leftarrow h^r M$
  $\alpha \leftarrow H(u_1, u_2, e)$
  $v \leftarrow c^r d^{r\alpha}$
  $C \leftarrow (u_1, u_2, e, v)$
  `Return` C

Algorithm $\mathcal{D}_{sk}(C)$
  parse C as (u_1, u_2, e, v)
  $\alpha \leftarrow H(u_1, u_2, e)$
  `If` ${u_1}^{x_1+y_1\alpha} {u_2}^{x_2+y_2\alpha} = v$
  `then` $M \leftarrow e/{u_1}^{z}$
  `else reject`
  `Return` M

Although Cramer and Shoup do not explicitly state the concrete security of their reduction, it can be gleaned from the proof in [6, Section 4]. Their reduction is essentially tight. In our language:

$$\mathsf{Adv}^{\text{1-cca}}_{\mathcal{CS},(q,g)}(t, q_{\mathrm{d}}) \le 2 \cdot \mathsf{Adv}^{\mathrm{ddh}}_{q,g}(t) + 2 \cdot \mathsf{Adv}^{cr}_{\mathcal{H}}(t) + \frac{2(4q_{\mathrm{d}} + 1)}{q} \,. \tag{5}$$

as long as $q_{\mathrm{d}} \le q/2$. The first term represents the advantage of the scheme in the single-user setting under chosen-ciphertext attack. Note that in this attack mode a new parameter is present, namely the number q_{d} of decryption queries made

by the adversary, and hence the advantage is a function of this in addition to the time t. (Definition 1 has the details.) We are using $\mathsf{Adv}^{cr}_{\mathcal{H}}(t)$ to represent the maximum possible probability that an adversary with time t can find collisions in a random member H of the family $\mathcal{H}$. The last term in Equation (5) is negligible because q is much bigger than $q_{\rm d}$ in practice, which is why we view this reduction as tight. Moving to the multi-user setting, Theorem 1 in combination with the above tells us that

$$\mathsf{Adv}^{n\text{-cca}}_{\mathcal{CS},(q,g)}(t, q_{\rm e}, q_{\rm d}) \leq 2 \cdot q_{\rm e} n \cdot \mathsf{Adv}^{\rm ddh}_{q,g}(t') + 2 \cdot q_{\rm e} n \cdot \mathsf{Adv}^{cr}_{\mathcal{H}}(t') + \frac{2q_{\rm e} n \cdot (4q_{\rm d} + 1)}{q}$$

where $t' = t + (\log(q_{\rm e} n))$. The first term represents the advantage of the scheme in the multi-user setting under chosen-ciphertext attack, with n users, $q_{\rm e}$ encryption queries per user, and $q_{\rm d}$ decryption queries per user. Our improvement is the following.

Theorem 3. *Let G be a group of a large prime order q. Let $\mathcal{H}$ be a family of collision-resistant hash function, each member of which maps from $\{0,1\}^*$ into Z_q. Let g be a generator of G. Let $\mathcal{CS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the Cramer-Shoup public-key encryption scheme associated to these parameters as defined above. Let $n, q_{\rm e}, q_{\rm d}, t$ be integers with $q_{\rm d} \leq q/2$. Then*

$$\mathsf{Adv}^{n\text{-cca}}_{\mathcal{CS},(q,g)}(t, q_{\rm e}, q_{\rm d}) \leq 2q_{\rm e} \cdot \mathsf{Adv}^{\rm ddh}_{q,g}(t') + 2q_{\rm e} \cdot \mathsf{Adv}^{cr}_{\mathcal{H}}(t') + \frac{2(4q_{\rm e} n q_{\rm d} + q_{\rm e} n)}{q}$$

where $t' = t + O(n \cdot T^{\rm exp}_q)$. ∎

Note that the last term is negligible for any reasonable values of $n, q_{\rm e}, q_{\rm d}$ due to the fact that q is large. So comparing with Equation (5) we see that we have essentially the same proven security for n users or one user when each encrypts $q_{\rm e}$ messages.

The reduction we got for Cramer-Shoup is not as tight as the one we got for El Gamal. We did not avoid the factor of $q_{\rm e}$ in a degradation of security of Cramer-Shoup for the multi-user setting. However it is still an open problem to avoid the factor of $q_{\rm e}$ even when there is only a single user encrypting $q_{\rm e}$ messages, so our result can be viewed as the optimal extension to the multi-user setting of the *known* results in the single-user setting.

To obtain this result we use Lemma 1 and modify the simulation algorithm from [6]. We provide a full proof and discuss the difficulties in improving the quality of the reduction in [2].

Acknowledgments

We thank Victor Shoup for information about the concrete security of the reduction in [6] and for pointing out to us the difficulties in attempting to improve the quality of the Cramer-Shoup reduction (in the single-user setting) as a function of the number of encryption queries. We also thank the Eurocrypt 2000 referees for their comments.

Mihir Bellare and Alexandra Boldyreva are supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

References

1. O. Baudron, D. Pointcheval and J. Stern, "Extended notions of security for multicast public key cryptosystems," Manuscript.
2. M. Bellare, A. Boldyreva, and S. Micali, "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements," Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir`.
3. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," Proceedings *of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.
4. M. Bellare, P. Rogaway, "Optimal asymmetric encryption – How to encrypt with RSA," *Advances in Cryptology – Eurocrypt* 94 *Proceedings*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.
5. M. Bellare and A. Sahai, "Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization," *Advances in Cryptology – Crypto* 99 *Proceedings*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.
6. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
7. D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," Proceedings *of the* 23rd *Annual Symposium on Theory of Computing*, ACM, 1991.
8. T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol 31, 1985, pp. 469–472.
9. S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.
10. J. Håstad, "Solving simultaneous modular equations of low degree," *SIAM J. on Computing* Vol. 17, No. 2, April 1988.
11. M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," Proceedings *of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.
12. RSA Laboratories, "PKCS-1," `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/`.
13. C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," *Advances in Cryptology – Crypto* 91 *Proceedings*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
14. V. Shoup, "On formal models for secure key exchange, " Theory of Cryptography Library Record 99-12, `http://philby.ucsd.edu/cryptolib/`.
15. M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology – Eurocrypt* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

16. Y. Tsiounis and M. Yung, "On the security of El Gamal based encryption," *Proceedings of the First International workshop on practice and theory in Public Key Cryptography (PKC'98)*, Lecture Notes in Computer Science Vol. 1431, H. Imai and Y. Zheng eds., Springer-Verlag, 1998.