

A preliminary version of this paper appears in *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.

# Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements

MIHIR BELLARE\*      ALEXANDRA BOLDYREVA†      SILVIO MICALI‡

May 2000

## 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.

**Keywords:** Encryption, public-key cryptosystems, El Gamal, Diffie-Hellman, decision Diffie-Hellman.

---

\*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: [mihir@cs.ucsd.edu](mailto:mihir@cs.ucsd.edu). URL: <http://www-cse.ucsd.edu/users/mihir>. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

†Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: [aboldyre@cs.ucsd.edu](mailto:aboldyre@cs.ucsd.edu). URL: <http://www-cse.ucsd.edu/users/aboldyre>. Supported in part by grants of first author.

‡MIT Laboratory for Computer Science, 545 Technology Square, Cambridge MA 02139, USA.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Background . . . . .	3
1.2	Model and general reduction . . . . .	3
1.3	The need for concrete security improvements . . . . .	5
1.4	Concrete security of El Gamal in the multi-user setting . . . . .	6
1.5	Concrete security of Cramer-Shoup in the multi-user setting . . . . .	6
1.6	Discussion and related work . . . . .	7
<b>2</b>	<b>Definitions</b>	<b>7</b>
<b>3</b>	<b>Security in the multi-user setting</b>	<b>9</b>
<b>4</b>	<b>A general reduction and its tightness</b>	<b>11</b>
<b>5</b>	<b>Improved security for DDH based schemes</b>	<b>14</b>
5.1	El Gamal . . . . .	16
5.2	Cramer-Shoup . . . . .	18
	<b>References</b>	<b>23</b>
<b>A</b>	<b>Proof of Lemma 5.2</b>	<b>24</b>

# 1 Introduction

This paper addresses the security of public-key cryptosystems in the “multi-user” setting, namely in the presence of attacks involving the encryption of related messages under different public keys. We present answers to the basic theoretical questions —namely what does security in this setting mean and for which schemes can we prove security— and then show how these results highlight a new “differentiating measure” between schemes, namely how security behaves as a function of the number of users, making obvious the importance, in practice, of seeking schemes permitting *improved* security reductions in the multi-user setting. Such reductions are presented for the El Gamal and Cramer-Shoup schemes, showing that these schemes provide a better efficiency to security tradeoff than some of their competitors when the effect on security of the presence of many different users is taken into consideration

## 1.1 Background

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 [GoMi]. 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 [Hå].<sup>1</sup> Suppose we have many users where the 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  $m^3 \bmod N_1 N_2 N_3$ . But  $m^3 < N_1 N_2 N_3$  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.

## 1.2 Model and 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.

---

<sup>1</sup> 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 using lattice reduction [Hå]. For simplicity we will continue to use the term “Håstad’s attack(s)” to refer to this body of cryptanalysis.

A GENERAL REDUCTION. The above 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 4.1 interprets “strong enough” for the single-user setting in the natural way: secure in the sense of indistinguishability of Goldwasser and Micali [GoMi]. 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 [DDN] under chosen-plaintext attack, and a simple way to extend our proof to that setting is to exploit the characterization of [BS]. Non-malleability under chosen-ciphertext attack is equivalent to indistinguishability under chosen-ciphertext attack [BDPR] so is already covered.)

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 several practical schemes secure against chosen-plaintext attack [BlGo, ElG], against chosen-ciphertext attack [CrSh], and against chosen-ciphertext attack in the random oracle model [BR, PKCS].

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*.<sup>2</sup> We will explain how this obviates the need to *explicitly* consider relations amongst messages. This model is important because its use extends beyond Theorem 4.1, 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

---

<sup>2</sup> An encryption oracle is used in definitions of security for private-key encryption [BDJR] 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.

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.

### 1.3 The need for concrete security improvements

Perhaps the most important impact of the general reduction of Theorem 4.1 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 4.1 says that the maximum probability that an adversary with running time  $t$  can compromise security in the multi-user setting —this in the sense of our definition discussed above— is at most  $q_en$  times the maximum probability that an adversary with running time closely related to  $t$  can compromise security in the standard sense of indistinguishability. Notationally,  $\text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(t, q_e) \leq q_en \cdot \text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(t')$  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  $q_en$ . 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. Here's an example to illustrate. Assume we have a Diffie-Hellman based scheme modulo a prime  $p$  whose size was chosen based only on consideration of the single-user setting, say to ensure that  $\text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(t') \leq 2^{-60}$ , for some appropriately large  $t'$ . This would be a quite acceptable security guarantee in the single-user setting. Now consider the “real” setting, which is the multi-user one. Say there are up to 200 million users with public keys. (This may not be true today, but we should budget for a large growth in the use of public-key cryptosystems with time.) Let's say we allow  $q_e = 2^{30}$  messages to be encrypted under each key. Then  $q_en \cdot \text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(t')$  is  $\approx 0.2$ , meaning essentially no security guarantee remains in the multi-user setting. To have a breaking probability in the multi-user setting bounded by the original target of  $2^{-60}$  we would have to increase the size of  $p$ . Even a small increase in the size of  $p$  will impact the efficiency of the scheme quite a lot because the cost of encryption is a cubic function of the size of  $p$ . (To get some rough numerical estimates, assume the advantage in the single-user setting is of the form  $t'/O(e^{1.9 \cdot \ln(p)^{1/3} \cdot \ln \ln(p)^{2/3}})$  and  $q_en \approx 2^{60}$ . If we wanted the advantage in the multi-user setting to be the same as that yielded by a 1024 bit prime in the single-user setting, whatever this value might be, then we would have to use a prime of about 1720 bits, meaning the cost of encryption would increase by a factor of about 4.7.)

It is natural to ask whether the gap in advantages exhibited in Theorem 4.1 is real or an artifact of our proof. We prove in Proposition 4.3 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  $q_en$ . 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.

#### 1.4 Concrete security of El Gamal in the multi-user setting

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 [NaRe, CrSh]). The reduction is essentially tight, and in our language says that  $\text{Adv}_{\mathcal{E}\mathcal{G},(q,g)}^{1\text{-cpa}}(t)$ —the maximum probability that an adversary of time-complexity  $t$  can break the El Gamal scheme via a chosen-plaintext attack in the single-user setting—is at most  $2\text{Adv}_{q,g}^{\text{ddh}}(t)$ —twice the maximum probability of solving the DDH problem in the same amount of time. (Here  $g$  is a generator of the group  $G$  of a large prime order  $q$ .) We thus have a complete and satisfactory picture of the security of the El Gamal scheme in the single-user setting; our concern now is the multi-user setting.

Theorem 4.1 together with the above implies that  $\text{Adv}_{\mathcal{E}\mathcal{G},(q,g)}^{n\text{-cpa}}(t, q_e)$ —the maximum probability that an adversary of time-complexity  $t$  can break the El Gamal scheme via a chosen-plaintext attack in the presence of  $n$  users each encrypting  $q_e$  messages—is upper bounded by  $2q_en \cdot \text{Adv}_{q,g}^{\text{ddh}}(t')$ — $2q_e$  times the maximum probability of solving the DDH problem in the time  $t'$ —where  $t' \approx t$ . We show in Theorem 5.3 that via an improved reduction the factor of  $q_en$  can be essentially eliminated. Namely  $\text{Adv}_{\mathcal{E}\mathcal{G},(q,g)}^{n\text{-cpa}}(t, q_e)$  is upper bounded (roughly) by  $2\text{Adv}_{q,g}^{\text{ddh}}(t')$  where  $t'$  equals  $t$  plus an additive term that depends on  $n, q_e$ . 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. As discussed above, this translates into considerable cost savings in practice.

Our reduction exploits a self-reducibility property of the decisional Diffie-Hellman problem due to Stadler and Naor-Reingold [St, NaRe], and a variant thereof that was also independently noted by Shoup [Sh]. See Lemma 5.2.

#### 1.5 Concrete security of Cramer-Shoup in the multi-user setting

The El Gamal scheme provides security against chosen-plaintext attack. Nowadays there is much interest and need also for practical schemes provably achieving security against chosen-ciphertext attack. The Cramer-Shoup scheme [CrSh] 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  $q_en$ . We present in Theorem 5.4 an improved reduction which (roughly speaking) reduces the factor of  $q_en$  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 5.2 and features of the proof of security for the single-user case given in [CrSh].

## 1.6 Discussion and related work

It is important to confirm —as we did— that the notion of indistinguishability is strong enough to also imply security in the multi-user setting. If security in the polynomial-time framework is the only concern, we can stop here: the two notions are equivalent. But if we wish to use the theoretical results in practice we must be careful which model we use as the basis for selecting the size of security parameters in schemes. The multi-user setting is the “real” one, and thus when we choose a security parameter size it should be with the target of having some guaranteed bound on the probability of the scheme being broken in the multi-user setting, not the single-user one. This means we must have a clear model of security for the multi-user setting and quantitative bounds on adversarial advantage, in this setting, for schemes we want to consider. Once we do this we see that some schemes become preferable to others due to their better security, translating into improved efficiency for a given level of security.

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 improved reductions we have exhibited for Diffie-Hellman based schemes are possible because all users can work over the same group —specified by a public prime  $q$ — yet have different trapdoors. Such improvements are unlikely for RSA or factoring based schemes where the moduli must be different for each user. Thus our improved reductions highlight an advantage of Diffie-Hellman based schemes: they admit better proven-security to cost tradeoffs in the multi-user setting than schemes based on some other assumptions.

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 transferred. The definitions of [BDJR] for the single-user case can be adapted to the multi-user case using the ideas in Section 3. The analogue of Theorem 4.1 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 [BPS]. 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.)

A preliminary version of this paper appears as [BBM].

## 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  $C \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(M)$ . 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*  $\text{MsgSp}(pk)$  from which  $M$  is allowed to be drawn. We require that  $\mathcal{D}_{sk}(\mathcal{E}_{pk}(M)) = M$  for all  $M \in \text{MsgSp}(pk)$ .

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 [RaSi].

**Definition 2.1 [Indistinguishability of encryptions]** Let  $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be a public-key encryption scheme. Let  $B_{\text{cpa}}, B_{\text{cca}}$  be adversaries where the latter has access to an oracle. Let  $I$  be some initial information string. For  $b = 0, 1$  define the experiments

<p><b>Experiment <math>\text{Exp}_{\mathcal{PE}, I}^{1\text{-cpa}}(B_{\text{cpa}}, b)</math></b></p> <p><math>(pk, sk) \leftarrow \mathcal{K}(I)</math></p> <p><math>(m_0, m_1, s) \leftarrow B_{\text{cpa}}(\text{find}, I, pk)</math></p> <p><math>C \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(m_b)</math></p> <p><math>d \leftarrow B_{\text{cpa}}(\text{guess}, C, s)</math></p> <p><b>Return <math>d</math></b></p>	<p><b>Experiment <math>\text{Exp}_{\mathcal{PE}, I}^{1\text{-cca}}(B_{\text{cca}}, b)</math></b></p> <p><math>(pk, sk) \leftarrow \mathcal{K}(I)</math></p> <p><math>(m_0, m_1, s) \leftarrow B_{\text{cca}}^{\mathcal{D}_{sk}(\cdot)}(\text{find}, I, pk)</math></p> <p><math>C \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(m_b)</math></p> <p><math>d \leftarrow B_{\text{cca}}^{\mathcal{D}_{sk}(\cdot)}(\text{guess}, C, s)</math></p> <p><b>Return <math>d</math></b></p>
--	--

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

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

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

$$\begin{aligned} \text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(t) &= \max_{B_{\text{cpa}}} \{ \text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(B_{\text{cpa}}) \} \\ \text{Adv}_{\mathcal{PE}, I}^{1\text{-cca}}(t, q_d) &= \max_{B_{\text{cca}}} \{ \text{Adv}_{\mathcal{PE}, I}^{1\text{-cca}}(B_{\text{cca}}) \} \end{aligned}$$

where the maximum is over all  $B_{\text{cpa}}, B_{\text{cca}}$  with “time-complexity”  $t$ , and, in the case of  $B_{\text{cca}}$ , 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 2.2** We say that  $\mathcal{PE}$  is *polynomially-secure against chosen-plaintext attack* (resp. *chosen-ciphertext attack*) in the *single-user setting* if  $\text{Adv}_{\mathcal{PE},I}^{1\text{-cpa}}(B)$  (resp.  $\text{Adv}_{\mathcal{PE},I}^{1\text{-cca}}(B)$ ) 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”  $\Delta$  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  $\mathcal{O}_1, \dots, \mathcal{O}_n$ . 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. Let us see why. Suppose the adversary wanted to obtain ciphertexts of the same message under two different keys  $pk_1$  and

$pk_3$ . It could make a query  $(m_0, m_1)$  of  $\mathcal{O}_1$ , and the same query of  $\mathcal{O}_3$ . In response it gets  $\mathcal{E}_{pk_1}(m_b)$  and  $\mathcal{E}_{pk_3}(m_b)$ . More generally, if the adversary wanted to obtain the ciphertext of the result of a message modification function  $\Delta$  on some target message, it would first query  $(m_0, m_1)$  of  $\mathcal{O}_1$ , and then  $(\Delta(m_0), \Delta(m_1))$  of  $\mathcal{O}_3$ . In response it gets  $\mathcal{E}_{pk_1}(m_b)$  and  $\mathcal{E}_{pk_3}(\Delta(m_b))$ . The adversary could even use different message modification functions on the two messages. Thus 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. Thus 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

$$\text{LR}(m_0, m_1, b) = m_b$$

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*,

$$\mathcal{E}_{pk_1}(\text{LR}(\cdot, \cdot, b)), \dots, \mathcal{E}_{pk_n}(\text{LR}(\cdot, \cdot, b))$$

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 [BDJR] in the symmetric setting.) The oracle  $\mathcal{E}_{pk_i}(\text{LR}(\cdot, \cdot, b))$ , given query  $(m_0, m_1)$  where  $m_0, m_1 \in \text{MsgSp}(pk_i)$  must have equal length, first sets  $m_b \leftarrow \text{LR}(m_0, m_1, b)$ , 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  $\mathcal{E}_{pk_i}(\text{LR}(\cdot, \cdot, b))$ . 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 3.1** Let  $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be a public-key encryption scheme. Let  $A_{\text{cpa}}, A_{\text{cca}}$  be adversaries. Both have access to  $n \geq 1$  oracles, each of which takes as input any two strings of equal length, and  $A_{\text{cca}}$  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  $\text{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A_{\text{cpa}}, b)$**   
**For**  $i = 1, \dots, n$  **do**  $(pk_i, sk_i) \leftarrow \mathcal{K}(I)$  **EndFor**  
 $d \leftarrow A_{\text{cpa}}^{\mathcal{E}_{pk_1}(\text{LR}(\cdot, \cdot, b)), \dots, \mathcal{E}_{pk_n}(\text{LR}(\cdot, \cdot, b))}(I, pk_1, \dots, pk_n)$ ; **Return**  $d$

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

It is mandated that a query to any LR oracle consists of two messages of *equal* length and that for each  $i = 1, \dots, n$  adversary  $A_{\text{cca}}$  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  $A_{\text{cpa}}$ , and the *advantage* of  $A_{\text{cca}}$ , respectively, as follows:

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

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  $t, q_e, q_d$  let

$$\begin{aligned}\text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(t, q_e) &= \max_{A_{\text{cpa}}} \{ \text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(A_{\text{cpa}}) \} \\ \text{Adv}_{\mathcal{PE}, I}^{n\text{-cca}}(t, q_e, q_d) &= \max_{A_{\text{cca}}} \{ \text{Adv}_{\mathcal{PE}, I}^{n\text{-cca}}(A_{\text{cca}}) \}\end{aligned}$$

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  $A_{\text{cca}}$ , 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 3.2** Notice that when  $n = q_e = 1$  in Definition 3.1, 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 3.1 with  $n = q_e = 1$  is equivalent to Definition 2.1. We can view Definition 3.1 as extending Definition 2.1 along two dimensions: the number of users and the number of messages encrypted by each user.

In analogy to Definition 2.2 we now have

**Definition 3.3** We say that  $\mathcal{PE}$  is *polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the multi-user setting* if  $\text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(A)$  (resp.  $\text{Adv}_{\mathcal{PE}, I}^{n\text{-cca}}(A)$ ) 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.

**Theorem 4.1** Let  $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be a public-key encryption scheme. Let  $n, q_e, q_d, t$  be integers and  $I$  some initial information string. Then

$$\begin{aligned}\text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(t, q_e) &\leq q_e n \cdot \text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(t') \\ \text{Adv}_{\mathcal{PE}, I}^{n\text{-cca}}(t, q_e, q_d) &\leq q_e n \cdot \text{Adv}_{\mathcal{PE}, I}^{1\text{-cca}}(t', q_d)\end{aligned}$$

where  $t' = t + O(\log(q_e n))$ . ■

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

**Corollary 4.2** 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. ■

**Proof of Theorem 4.1:** We first consider the case of chosen-plaintext attacks only and then briefly indicate how to extend the argument to the case of chosen-ciphertext attacks.

Let  $A$  be an adversary attacking the encryption scheme  $\mathcal{PE}$  in the multi-user setting. Assume it makes at most  $q_e$  queries to any of its  $n$  oracles and has time-complexity at most  $t$ . We will design an adversary  $B_A$  attacking the same scheme in the single-user setting so that

$$\text{Adv}_{\mathcal{PE}, I}^{1\text{-cpa}}(B_A) \geq \frac{1}{nq_e} \cdot \text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(A). \quad (1)$$

Furthermore,  $B_A$  will have running time at most  $t'$ . The theorem follows by taking maximums. So it remains to design  $B_A$ . Refer to Section 4 for a discussion of the intuition behind the design of  $B_A$ . We now proceed to the full proof.

We begin by describing some hybrid experiments associated to  $A$ . It is convenient to parameterize the hybrids via a single integer  $l$  ranging from 0 to  $nq_e$  and counting the number of oracle queries replied to by encrypting the left message.

**Experiment  $\text{ExpH}_l$**  [ $0 \leq l \leq nq_e$ ]

For  $i = 1, \dots, n$  do  $(pk_i, sk_i) \leftarrow \mathcal{K}(I)$  EndFor

$ctr \leftarrow 0$

If  $l = 0$  then  $(r, c) \leftarrow (0, 0)$  EndIf

If  $l > 0$  then let  $r, c$  be such that  $l = (c - 1)q_e + r$  and  $1 \leq r \leq q_e$  and  $1 \leq c \leq n$  EndIf

Run  $A$  replying to oracle queries as follows:

$A \rightarrow (i, m_0, m_1)$  [ $1 \leq i \leq n$ ]

If  $i < c$  then  $C \leftarrow \mathcal{E}_{pk_i}(m_0)$ ;  $A \leftarrow C$  EndIf

If  $i > c$  then  $C \leftarrow \mathcal{E}_{pk_i}(m_1)$ ;  $A \leftarrow C$  EndIf

If  $i = c$  then

$ctr \leftarrow ctr + 1$

If  $ctr \leq r$  then  $C \leftarrow \mathcal{E}_{pk_i}(m_0)$ ;  $A \leftarrow C$  EndIf

If  $ctr > r$  then  $C \leftarrow \mathcal{E}_{pk_i}(m_1)$ ;  $A \leftarrow C$  EndIf

EndIf

Eventually  $A$  halts outputting a bit  $d$

Return  $d$

Let  $P_l \stackrel{\text{def}}{=} \Pr[\mathbf{ExpH}_l = 0]$  denote the probability that experiment  $\mathbf{ExpH}_l$  returns 0, for  $l = 0, 1, \dots, nq_e$ . Now we claim that

$$\text{Adv}_{\mathcal{PE}, I}^{n\text{-cpa}}(A) = P_{nq_e} - P_0. \quad (2)$$

This is justified as follows. Referring to Definition 3.1 for the terminology, we claim that

$$\Pr[\mathbf{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A, 0) = 0] = P_{nq_e} \quad \text{and} \quad \Pr[\mathbf{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A, 1) = 0] = P_0,$$

and after subtraction Equation (2) follows. The two equations above are justified as follows. In experiment  $\mathbf{ExpH}_{nq_e}$  we have  $l = nq_e$  so  $c = n$  and  $r = q_e$ . It follows that the response to any query  $(m_0, m_1)$  to any oracle is provided by encrypting  $m_0$ , so that the  $A$ 's "view" is the same as in experiment  $\mathbf{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A, 0)$ . On the other hand in experiment  $\mathbf{ExpH}_0$  we have  $l = 0$  so  $c = r = 0$ .

It follows that the response to any query  $(m_0, m_1)$  to any oracle is provided by encrypting  $m_1$ , so that  $A$ 's "view" is the same as in experiment  $\mathbf{Exp}_{\mathcal{P}\mathcal{E},I}^{n\text{-cpa}}(A, 1)$ .

Now we turn to the description of  $B_A$ . This is made more convenient by visualizing its operation in a way a little different from, but clearly equivalent to, that described in Definition 2.1. It takes as input  $I$  and a public key  $pk$ . However we will not explicitly separate the operation of  $B_A$  into the find and guess stages. Instead, think of  $B_A$  supplied with the oracle  $\mathcal{E}_{pk}(\text{LR}(\cdot, \cdot, b))$  and allowed to make only a single query of this oracle. After getting the response it must try to guess the challenge bit  $b$ .

**Adversary**  $B_A^{\mathcal{E}_{pk}(\text{LR}(\cdot, \cdot, b))}$  [Only one query to the LR oracle is allowed]

$l \xleftarrow{R} \{1, \dots, nq_e\}$

Let  $r, c$  be such that  $l = (c-1)q_e + r$  and  $1 \leq r \leq q_e$  and  $1 \leq c \leq n$

**For**  $i \in \{1, \dots, c-1, c+1, \dots, n\}$  **do**  $(pk_i, sk_i) \leftarrow \mathcal{K}(I)$  **EndFor**

$pk_c \leftarrow pk$ ;  $ctr \leftarrow 0$

Run  $A$  replying to oracle queries as follows:

$A \rightarrow (i, m_0, m_1)$  [ $1 \leq i \leq n$ ]

**If**  $i < c$  **then**  $C \leftarrow \mathcal{E}_{pk_i}(m_0)$ ;  $A \leftarrow C$  **EndIf**

**If**  $i > c$  **then**  $C \leftarrow \mathcal{E}_{pk_i}(m_1)$ ;  $A \leftarrow C$  **EndIf**

**If**  $i = c$  **then**

$ctr \leftarrow ctr + 1$

**If**  $ctr < r$  **then**  $C \leftarrow \mathcal{E}_{pk_i}(m_0)$ ;  $A \leftarrow C$  **EndIf**

**If**  $ctr > r$  **then**  $C \leftarrow \mathcal{E}_{pk_i}(m_1)$ ;  $A \leftarrow C$  **EndIf**

**If**  $ctr = r$  **then**

Let  $C \leftarrow \mathcal{E}_{pk}(\text{LR}(m_0, m_1, b))$  [Let  $m_0, m_1$  be the single allowed oracle query]

$A \leftarrow C$  [Return  $C$  to  $A$  as the response to its query]

**EndIf**

**EndIf**

Eventually  $A$  halts outputting a bit  $d$

**Return**  $d$

We now justify Equation (1). Referring to Definition 2.1 for terminology we claim that

$$\Pr \left[ \mathbf{Exp}_{\mathcal{P}\mathcal{E},I}^{1\text{-cpa}}(B, 0) = 0 \right] = \frac{1}{nq_e} \cdot \sum_{l=1}^{nq_e} P_l \quad \text{and} \quad \Pr \left[ \mathbf{Exp}_{\mathcal{P}\mathcal{E},I}^{1\text{-cpa}}(B, 1) = 0 \right] = \frac{1}{nq_e} \cdot \sum_{l=1}^{nq_e} P_{l-1}. \quad (3)$$

Subtracting and exploiting the collapse of the sums we get

$$\text{Adv}_{\mathcal{P}\mathcal{E},I}^{1\text{-cpa}}(B_A) = \frac{1}{nq_e} \cdot \sum_{l=1}^{nq_e} P_l - P_{l-1} = \frac{1}{nq_e} \cdot [P_{nq_e} - P_0].$$

Equation (1) follows by applying Equation (2), so it remains to justify Equations (3). Each value of  $l$  in  $\{1, \dots, nq_e\}$  is equally likely for  $B_A$  so let's focus on one choice of  $l$ . It is always true that all queries to the first  $c-1$  users, and the first  $r-1$  queries to the  $c$ -th user, are answered by encrypting the left message, while all queries to users  $c+1, \dots, n$ , and the last  $q_e - r$  queries to the  $c$ -th user, are answered by encrypting the right message. When we run experiment  $\mathbf{Exp}_{\mathcal{P}\mathcal{E},I}^{1\text{-cpa}}(B_A, 0)$ , the  $r$ -th query to the  $c$ -th user is answered by encrypting the left message, so the experiment is the same as  $\mathbf{ExpH}_l$ . When we run experiment  $\mathbf{Exp}_{\mathcal{P}\mathcal{E},I}^{1\text{-cpa}}(B_A, 1)$ , the  $r$ -th query to the  $c$ -th user is answered by encrypting the right message, so the experiment is the same as  $\mathbf{ExpH}_{l-1}$ .

Finally we should justify the claim that the running time of  $B_A$  is  $t^l$ . Here we take advantage of our convention that the time-complexity of the adversary refers to the execution time of the entire underlying experiment rather than just the adversary itself. Thus  $t$  includes the time to select  $n$

key pairs, so that these actions of  $B_A$  are not an overhead. Taking the conventions into account the only overhead for  $B_A$  is to pick the random number  $l$  and execute some conditional statements.

We provide a brief sketch of how to extend the proof to the case of chosen-ciphertext attacks. The definition of the hybrid experiments is the same with regard to how left-or-right encryption oracle queries are answered. Decryption queries are however answered truthfully, using the correct secret key. The adversary  $B_A$  is given also the decryption oracle  $\mathcal{D}_{sk}(\cdot)$  where  $sk$  is the secret key corresponding to its input public key  $pk$ . It proceeds as before. The novel elements is to provide answers to decryption oracle queries. When the query is to  $\mathcal{D}_{sk_i}(\cdot)$  for  $i \neq c$ , algorithm  $B_A$  can easily provide the answer since it is in possession of  $sk_i$ . When  $i = c$  it provides the answer by invoking its own given decryption oracle. The analysis proceeds as before. ■

**TIGHTNESS OF THE BOUND.** We present an example that shows that in general the bound of Theorem 4.1 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  $\text{Adv}_{\mathcal{PE}',I}^{n\text{-cpa}}(t, q_e)$  is  $\Omega(q_e n)$  times  $\text{Adv}_{\mathcal{PE}',I}^{1\text{-cpa}}(t)$ . 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  $\text{Adv}_{\mathcal{PE},I}^{1\text{-cpa}}(t)$  is essentially zero. With that interpretation we indeed have the claimed relation.

**Proposition 4.3** 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

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

**Proof of Proposition 4.3:** Let  $\mathcal{E}$  be the encryption algorithm of the given scheme  $\mathcal{PE}$ . Modify the encryption algorithm so that with probability  $1/nq_e$  it returns the message to be encrypted in the clear, together with some flag that indicates it is “misbehaving”. The new scheme  $\mathcal{PE}'$  has the same key generation algorithm as the old scheme, this new, modified encryption algorithm, and a decryption algorithm obtained by appropriately modifying that of the old scheme. (The new decryption algorithm is just like the old one unless it sees the flag in the ciphertext it is provided, in which case it returns the message that accompanies the flag.) Now, in the multi-user setting, each of the  $nq_e$  encryption queries can result in the mis-behavior with probability  $1/nq_e$ , so the adversary can win the game except with probability  $(1 - 1/nq_e)^{nq_e} \approx 1/e$ . Thus, the probability that the adversary wins is about  $1 - 1/e \geq 0.6$ . On the other hand in the single user setting, the probability of getting the encryption function to mis-behave is at most  $1/nq_e$ . If the pathological event does not occur the adversary is faced with the task of breaking an instance of the old scheme. This intuition can be turned into a formal argument by providing an explicit reduction. We omit the details. An analogous result holds in the chosen-ciphertext attack case, and we omit it. ■

## 5 Improved security for DDH 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 5.1** 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 $\mathbf{Exp}_{q,g}^{\text{ddh-real}}(D)$	Experiment $\mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D)$
$x \stackrel{R}{\leftarrow} Z_q; X \leftarrow g^x$	$x \stackrel{R}{\leftarrow} Z_q; X \leftarrow g^x$
$y \stackrel{R}{\leftarrow} Z_q; Y \leftarrow g^y$	$y \stackrel{R}{\leftarrow} Z_q; Y \leftarrow g^y$
$K \leftarrow g^{xy}$	$K \stackrel{R}{\leftarrow} G$
$d \leftarrow D(q, g, X, Y, K)$	$d \leftarrow D(q, g, X, Y, K)$
<b>Return</b> $d$	<b>Return</b> $d$

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

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

where the maximum is over all  $D$  with “time-complexity”  $t$ . ■

The “time-complexity” of  $D$  is the maximum of the execution times of the two experiments  $\mathbf{Exp}_{q,g}^{\text{ddh-real}}(D)$  and  $\mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D)$ , 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 5.2 below. The case  $x \neq 0$  below is noted in [St, Proposition 1] and [NaRe, Lemma 3.2]. The variant with  $x = 0$  was noted independently in [Sh]. Below  $T_q^{\text{exp}}$  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 5.2 is in Appendix A.

**Lemma 5.2** 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  $O(T_q^{\text{exp}})$  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. ■

For example when  $x = 0$  and  $c \neq ab$ , the lemma says that  $a' = a$  but  $b', c'$  are randomly and independently distributed over  $Z_q$ .

## 5.1 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 5.2. 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 $\mathcal{K}(q, g)$ $x \xleftarrow{R} Z_q$ $X \leftarrow g^x$ $pk \leftarrow (q, g, X)$ $sk \leftarrow (q, g, x)$ <b>Return</b> $(pk, sk)$	Algorithm $\mathcal{E}_{q,g,X}(M)$ $y \xleftarrow{R} Z_q$ $Y \leftarrow g^y$ $K \leftarrow X^y$ $W \leftarrow KM$ <b>Return</b> $(Y, W)$	Algorithm $\mathcal{D}_{q,g,x}(Y, W)$ $K \leftarrow Y^x$ $M \leftarrow WK^{-1}$ <b>Return</b> $M$
---	---	--

We noted in Section 1.4 that the hardness of the DDH problem implies that the El Gamal scheme meets the standard notion of indistinguishability of encryptions (cf. [NaRe, CrSh]), and the reduction is essentially tight:  $\text{Adv}_{\mathcal{EG},(q,g)}^{1\text{-cpa}}(t)$  is at most  $2\text{Adv}_{q,g}^{\text{ddh}}(t)$ . We want to look at the security of the El Gamal scheme in the multi-user setting. Directly applying Theorem 4.1 in conjunction with the above would tell us that

$$\text{Adv}_{\mathcal{EG},(q,g)}^{n\text{-cpa}}(t, q_e) \leq 2q_e n \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') \quad (4)$$

where  $t' = t + O(\log(q_e n))$ . 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 5.3** 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  $n, q_e, t$  be integers. Then

$$\text{Adv}_{\mathcal{EG},(q,g)}^{n\text{-cpa}}(t, q_e) \leq 2 \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') + \frac{1}{q}$$

where  $t' = t + O(q_e n \cdot T_q^{\text{exp}})$ . ■

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 5.3:** Let  $A$  be an adversary attacking the El Gamal public-key encryption scheme  $\mathcal{EG}$  in the multi-user setting (cf. Definition 3.1). Suppose it makes at most  $q_e$  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 5.1) so that  $D_A$  has running time at most  $t'$  and

$$\text{Adv}_{q,g}^{\text{ddh}}(D_A) \geq \frac{1}{2} \cdot \text{Adv}_{\mathcal{EG},(q,g)}^{n\text{-cpa}}(A) - \frac{1}{2q}. \quad (5)$$

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**  $D_A(q, g, X, Y, K)$   
 $b \stackrel{R}{\leftarrow} \{0, 1\}$   
**For**  $i = 1, \dots, n$  **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**  $j = 2, \dots, q_e$  **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:  
 $A \rightarrow (i, m_0, m_1)$  [ $1 \leq i \leq n$  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**

Figure 1: *Distinguisher*  $D_A$  in proof of Theorem 5.3, where  $R$  is the algorithm of Lemma 5.2.

adversary  $A$  as a subroutine.  $D_A$  will provide for  $A$  as input public keys  $pk_1, \dots, pk_n$  and global information  $q, g$  and will simulate for  $A$  the  $n$  LR oracles,  $\mathcal{E}_{pk_i}(\text{LR}(\cdot, \cdot, b))$  for  $i = 1, \dots, n$ . We use the notation  $A \rightarrow (i, m_0, m_1)$  to indicate that  $A$  is making query  $(m_0, m_1)$  to its  $i$ -th LR oracle, where  $1 \leq i \leq n$  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. The security improvement over that provided by the naive hybrid argument is achieved by using the self-reducibility properties of the DDH problem in several ways. We are letting  $R$  denote the algorithm of Lemma 5.2.

We now proceed to analyze  $D_A$ . First consider  $\mathbf{Exp}_{q,g}^{\text{ddh-real}}(D_A)$ . In this case, the inputs  $X, Y, K$  to  $D_A$  above satisfy  $K = g^{xy}$  where  $X = g^x$  and  $Y = g^y$ . Lemma 5.2 tells us, first, that  $X'_1[1], \dots, X'_n[1]$  are all uniformly and independently distributed over  $G$ , because here  $R$  was called with  $x = 1 \neq 0$ . Thus they have the proper distribution of public keys for the El Gamal cryptosystem. Applying the same lemma again, we see that the reply to LR oracle query  $(i, m_0, m_1)$  of  $A$  is distributed exactly like an El Gamal encryption of  $m_b$  under public key  $(q, g, X'_i[1])$ , for all  $i = 1, \dots, n$ . We use this to see that

$$\begin{aligned} \Pr \left[ \mathbf{Exp}_{q,g}^{\text{ddh-real}}(D) = 1 \right] &= \frac{1}{2} \cdot \Pr \left[ \mathbf{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A, 0) = 0 \right] + \frac{1}{2} \cdot \left( 1 - \Pr \left[ \mathbf{Exp}_{\mathcal{PE}, I}^{n\text{-cpa}}(A, 1) = 0 \right] \right) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \text{Adv}_{\mathcal{EG}, (q,g)}^{n\text{-cpa}}(A). \end{aligned} \quad (6)$$

Now consider  $\mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D_A)$ . In this case, the inputs  $X, Y, K$  to  $D_A$  above are all uniformly distributed over  $G$ . Lemma 5.2 again tells us, first, that  $X'_1[1], \dots, X'_n[1]$  are all uniformly and independently distributed over  $G$ , so that they again have the proper distribution of public keys for the El Gamal cryptosystem, because  $R$  was called with  $x = 1 \neq 0$ . Now letting  $x, y, z$  be such that  $X = g^x, Y = g^y, K = g^z$ , with probability at least  $1 - 1/q$  it is true that  $z \neq xy \pmod{q}$ . Assuming this is true, we apply the same lemma again. It tells us that the value  $K'_i[ctr_i]$  is distributed uniformly at random in  $G$  independently of anything else. Hence the same is true of  $W_i[ctr_i]$ . This means that the reply to query  $(i, m_0, m_1)$  of  $A$  gives  $A$  no information about  $b$ , in an information-theoretic sense. So

$$\Pr \left[ \mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D) = 1 \right] \leq \frac{1}{2} \cdot \left( 1 - \frac{1}{q} \right) + \frac{1}{q} = \frac{1}{2} + \frac{1}{2q}. \quad (7)$$

The  $1/q$  term accounts for the probability that  $z = xy \pmod{q}$ . Subtracting Equations 6 and 7 we get

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

$$\geq \frac{1}{2} \cdot \text{Adv}_{\mathcal{EG},(q,g)}^{n\text{-cpa}}(A) - \frac{1}{2q},$$

which is Equation (5).

It remains to justify the claim about the time-complexity of  $D_A$ . The overhead for  $D_A$  is essentially that of invoking the algorithm  $R$  a total of  $nq_e$  times, and that's the added cost in  $t'$ . ■

## 5.2 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 [CrSh]. We 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:

<p>Algorithm <math>\mathcal{K}(q, g)</math></p> <p><math>g_1 \leftarrow g ; g_2 \stackrel{R}{\leftarrow} G</math></p> <p><math>H \stackrel{R}{\leftarrow} \mathcal{H}</math></p> <p><math>x_1, x_2, y_1, y_2, z \stackrel{R}{\leftarrow} Z_q</math></p> <p><math>c \leftarrow g_1^{x_1} g_2^{x_2}</math></p> <p><math>d \leftarrow g_1^{y_1} g_2^{y_2}</math></p> <p><math>h \leftarrow g_1^z</math></p> <p><math>pk \leftarrow (g_1, g_2, c, d, h)</math></p> <p><math>sk \leftarrow (x_1, x_2, y_1, y_2, z)</math></p> <p><b>Return</b> <math>(pk, sk)</math></p>	<p>Algorithm <math>\mathcal{E}_{pk}(M)</math></p> <p><math>r \stackrel{R}{\leftarrow} Z_q</math></p> <p><math>u_1 \leftarrow g_1^r</math></p> <p><math>u_2 \leftarrow g_2^r</math></p> <p><math>e \leftarrow h^r M</math></p> <p><math>\alpha \leftarrow H(u_1, u_2, e)</math></p> <p><math>v \leftarrow c^r d^{r\alpha}</math></p> <p><math>C \leftarrow (u_1, u_2, e, v)</math></p> <p><b>Return</b> <math>C</math></p>	<p>Algorithm <math>\mathcal{D}_{sk}(C)</math></p> <p>parse <math>C</math> as <math>(u_1, u_2, e, v)</math></p> <p><math>\alpha \leftarrow H(u_1, u_2, e)</math></p> <p><b>If</b> <math>u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} = v</math></p> <p><b>then</b> <math>M \leftarrow e/u_1^z</math></p> <p><b>else reject</b></p> <p><b>Return</b> <math>M</math></p>
---	---	--

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

$$\text{Adv}_{\mathcal{CS},(q,g)}^{1\text{-cca}}(t, q_d) \leq 2 \cdot \text{Adv}_{q,g}^{\text{ddh}}(t) + 2 \cdot \text{Adv}_{\mathcal{H}}^{cr}(t) + \frac{2(4q_d + 1)}{q}. \quad (8)$$

as long as  $q_d \leq q/2$ . 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 2.1 has the details.) We are using  $\text{Adv}_{\mathcal{H}}^{cr}(t)$  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 (8) is negligible because  $q$  is much bigger than  $q_d$  in practice, which is why we view this reduction as tight. Moving to the multi-user setting, Theorem 4.1 in combination with the above tells us that

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

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

**Theorem 5.4** 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  $n, q_e, q_d, t$  be integers with  $q_d \leq q/2$ . Then

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

where  $t' = t + O(n \cdot T_q^{\text{exp}})$ . ■

Note that the last term is negligible for any reasonable values of  $n, q_e, q_d$  due to the fact that  $q$  is large. So comparing with Equation (8) we see that we have essentially the same proven security for  $n$  users or one user when each encrypts  $q_e$  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  $q_e$  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  $q_e$  even when there is only a single user encrypting  $q_e$  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 5.2 and modify the simulation algorithm from [CrSh].

**Proof of Theorem 5.4:** We will focus on proving the result in the case that  $q_e = 1$ . Namely we want to show that

$$\text{Adv}_{\mathcal{CS},(q,g)}^{n\text{-cca}}(t, 1, q_d) \leq 2 \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') + 2 \cdot \text{Adv}_{\mathcal{H}}^{\text{cr}}(t') + \frac{2(4nq_d + n)}{q}. \quad (9)$$

Given this, the statement of the theorem follows via a hybrid argument reducing the case of  $q_e$  LR encryption queries per user to the case of a single LR encryption query per user while allowing the advantage to grow by a factor of  $q_e$ . Since this hybrid argument is simple and standard—it can be done along the lines of the proof of Theorem 4.1—we omit it and proceed to the crux of the proof which is the case  $q_e = 1$ . In this case we have to show how we are avoiding the appearance of a factor equal to the number  $n$  of users in the bound on the advantage.

Let  $A$  be an adversary attacking the Cramer-Shoup public-key encryption scheme  $\mathcal{CS}$  in a multi-user setting. We are assuming that it makes at most one query to each of its  $n$  LR encryption oracles, at most  $q_d$  queries to each of its  $n$  decryption oracles and has time-complexity at most  $t$ . We will design an adversary  $D_A$  for the Decisional Diffie-Hellman problem. The adversary  $D_A$  takes as input  $(q, g_1, g_2, u_1, u_2)$ , where  $(g_1, g_2, u_1, u_2)$  have the form  $(g, g^x, g^y, g^{xy})$  respectively or they are all random elements in  $G$ . To figure out which,  $D_A$  will use adversary  $A$  as a subroutine.  $D_A$  will provide for  $A$  as input public keys  $pk_1, \dots, pk_n$  and global information  $q, g$  and will simulate for  $A$  the  $n$  LR encryption oracles,  $\mathcal{E}_{pk_i}(\text{LR}(\cdot, \cdot, b))$  and the  $n$  decryption oracles,  $\mathcal{D}_{sk_i}(\cdot)$  for  $i = 1, \dots, n$ . We let  $R$  denote the algorithm of Lemma 5.2.

**Adversary  $D_A(q, g_1, g_2, u_1, u_2)$**

$b \xleftarrow{R} \{0, 1\}$

$H \xleftarrow{R} \mathcal{H}$

**For**  $i = 1, \dots, n$  **do**

$(g_{2,i}, u_{1,i}, u_{2,i}) \leftarrow R(q, g_1, g_2, u_1, u_2, 1)$

$x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}, z_{1,i}, z_{2,i} \xleftarrow{R} Z_q$

$c_i \leftarrow g_1^{x_{1,i}} (g_{2,i})^{x_{2,i}}; d_i \leftarrow g_1^{y_{1,i}} (g_{2,i})^{y_{2,i}}; h_i \leftarrow g_1^{z_{1,i}} (g_{2,i})^{z_{2,i}}$

$pk_i \leftarrow (g_1, g_{2,i}, c_i, d_i, h_i)$

**EndFor**

Run  $A$  replying to its LR encryption oracle queries as follows:

$A \xrightarrow{\mathcal{E}} (i, m_0, m_1) \quad [1 \leq i \leq n \text{ and } m_0, m_1 \in G]$   
 $e_i \leftarrow (u_{1,i})^{z_{1,i}} (u_{2,i})^{z_{2,i}} m_b$   
 $\alpha_i \leftarrow H(u_{1,i}, u_{2,i}, e_i)$   
 $v_i \leftarrow (u_{1,i})^{x_{1,i} + y_{1,i} \alpha_i} (u_{2,i})^{x_{2,i} + y_{2,i} \alpha_i}$   
 $A \leftarrow (u_{1,i}, u_{2,i}, e_i, v_i)$

And replying to  $A$ 's decryption queries as follows:

$A \xrightarrow{\mathcal{D}} (i, C) \quad [1 \leq i \leq n]$   
 parse  $C$  as  $(u'_1, u'_2, e', v')$   
 $\alpha \leftarrow H(u'_1, u'_2, e')$   
**If**  $(u'_1)^{x_{1,i} + y_{1,i} \alpha} (u'_2)^{x_{2,i} + y_{2,i} \alpha} = v'$  **then**  $M \leftarrow e' / u_1^{z_{1,i}} u_2^{z_{2,i}}$  **else reject**  
 $A \leftarrow M$

Eventually  $A$  halts and outputs a bit  $d$   
**If**  $b = d$  **then return 1 else return 0**

We derived this algorithm by adapting the simulation from [CrSh] to the multi-user case and then weaving in the Diffie-Hellman self-reducibility algorithms to improve the quality of the reduction. We apply the algorithm  $R$  of Lemma 5.2 to  $D_A$ 's input challenge values  $n$  times to produce  $n$  triples  $g_{2,i}, u_{1,i}, u_{2,i}$ . The resulting  $g_{2,i}$  values will be a part of a corresponding public key. Then we create  $n$  pairs of public and secret keys. Using these keys we are able to answer all encryption queries of  $A$  and also all its decryption queries.

We now analyze  $D_A$ . First consider  $\mathbf{Exp}_{q,g}^{\text{ddh-real}}(D_A)$ . In this case the input to  $D_A$  has the form  $q, g, g^x, g^y, g^{xy}$ . We can read this also as  $q, g_1, g_2, u_1, u_2$ , where  $u_1 = g_1^y$  and  $u_2 = g_2^y$ . Then the algorithm of Lemma 5.2 produces  $n$  pairs of  $u_{1,i}, u_{2,i}$  having the same properties. As [CrSh, Lemma 1] shows, in this case the algorithm produces valid ciphertexts for  $A$  and the outputs of the decryption oracle have the right distribution. Also [CrSh, Lemma 1] states that under this kind of simulation the adversary's view is almost indistinguishable from that in the actual attack, except that a decryption oracle may accept with small probability an invalid ciphertext. The proof of [CrSh, Lemma 1, Claim] holds for the setting with  $n$  users, since their reasonings can be applied to each user independently. Specifically, let  $P_0$  denote the probability that an invalid ciphertext is accepted by the simulator in the case  $b = 0$ , and  $P_1$  the same when  $b = 1$ . Then

$$\begin{aligned}
& \Pr \left[ \mathbf{Exp}_{q,g}^{\text{ddh-real}}(D) = 1 \right] \\
& \geq \frac{1}{2} \cdot \left( \Pr \left[ \mathbf{Exp}_{\mathcal{CS},(q,g)}^{n\text{-cpa}}(A, 0) = 0 \right] - P_0 \right) + \frac{1}{2} \cdot \left( 1 - \Pr \left[ \mathbf{Exp}_{\mathcal{CS},(q,g)}^{n\text{-cpa}}(A, 1) = 0 \right] - P_1 \right) \\
& = \frac{1}{2} + \frac{1}{2} \cdot \text{Adv}_{\mathcal{CS},(q,g)}^{n\text{-cpa}}(A) - \frac{1}{2}(P_0 + P_1). \tag{10}
\end{aligned}$$

Now consider  $\mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D_A)$ . In this case, the input values  $u_1, u_2$  to  $D_A$  above are uniformly distributed over  $G$ . Then the algorithm  $R$  outputs  $n$  values  $u_{1,i}, u_{2,i}$ , which are also random. We show that the proof of [CrSh, Lemma 2] can be generalized for the case with  $n$  users each making at most one LR encryption oracle query. First we consider the [CrSh, Lemma 2, Claim 1]. We want to show that if the decryption oracle rejects all invalid ciphertexts, then the the distribution of the challenge bit  $b$  is independent from the adversary's view, where the adversary now is in a multi-user single-query setting. Consider the point  $Q = (z_{1,1}, z_{2,1}, \dots, z_{1,n}, z_{2,n}) \in \mathbb{Z}_q^{2n}$ . Let  $w_i = \log_{g_{1,i}} g_{2,i}$  and then  $u_{1,i} = g_1^{r_{1,i}}, u_{2,i} = g_1^{w r_{2,i}}$ . From the public keys the adversary gets  $n$  linear relations ( $i = 1, \dots, n$ ):

$$z_{1,i} + w z_{2,i} = \log_{g_1} h_i$$

From the outputs of the LR encryption oracles the adversary gets  $n$  relations ( $i = 1, \dots, n$ ):

$$e_i = (u_{1,i})^{z_{1,i}} (u_{2,i})^{z_{2,i}} m_b$$

Now for each guess of the challenge bit the adversary can consider the following  $n$  relations:

$$r_{1,i} z_{1,i} + w r_{2,i} z_{2,i} = \log_{g_1} \epsilon_{b'_i}$$

where  $\epsilon_i = \frac{\epsilon_{b'_i}}{m_{b'}}$ .

Assuming the adversary decrypts only valid ciphertexts, the decryptions do not give it any additional information about the  $Q$ . We claim that if  $r_{1,i} \neq r_{2,i}$  for  $i = 1, \dots, n$  then the point  $Q$  is randomly distributed from the adversary's view. This can be shown by observing each of the two systems of linear equations that the adversary has (for  $b' = 0, 1$ ):

$$\begin{bmatrix} 1 & w_1 & 0 & 0 & \dots & 0 & 0 \\ r_{1,1} & w_1 r_{2,1} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & w_2 & \dots & 0 & 0 \\ 0 & 0 & r_{1,2} & w_2 r_{2,2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & w_n \\ 0 & 0 & 0 & 0 & \dots & r_{1,n} & w_n r_{2,n} \end{bmatrix} \cdot \begin{bmatrix} z_{1,1} \\ z_{2,1} \\ z_{1,2} \\ z_{2,2} \\ \dots \\ z_{1,n} \\ z_{2,n} \end{bmatrix} = \begin{bmatrix} \log_{g_1} h_1 \\ \log_{g_1} \epsilon_{b',1} \\ \log_{g_1} h_2 \\ \log_{g_1} \epsilon_{b',2} \\ \dots \\ \log_{g_1} h_n \\ \log_{g_1} \epsilon_{b',n} \end{bmatrix}$$

Let  $A_{2n} Z_{2n} = B_{2n}$  denote this system correspondingly. Here the subscript shows the number of lines in the matrices. Consider now the determinant

$$\begin{aligned} \det(A_{2n}) &= w_1 r_{2,1} \cdot \det(A_{2(n-1)}) - w_1 r_{1,1} \cdot \det(A_{2(n-1)}) \\ &= \det(A_{2(n-1)}) \cdot w_1 (r_{2,1} - r_{1,1}) = w_1 \cdot \dots \cdot w_n (r_{2,1} - r_{1,1}) \dots (r_{2,n} - r_{1,n}) \end{aligned}$$

If  $r_{2,1} \neq r_{1,1}, \dots, r_{2,n} \neq r_{1,n}$ , then  $\det(A_{2n}) \neq 0$ , meaning that for any adversary's guess about which message was encrypted there exists a single solution for a vector  $Z_{2n}$  and the adversary does not have any additional information to check its guess.

Now we show that [CrSh, Lemma 2, Claim 2] holds for a multi-user single-query setting too. Claim states that the decryption oracle rejects all invalid ciphertexts, except with negligible probability. We consider the distribution of the point  $P = (x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, \dots, x_{1,n}, x_{2,n}, y_{1,n}, y_{2,n}) \in Z_q^{4n}$ . From the public keys the adversary gets the following linear relations:

$$x_{1,i} + w_i x_{2,i} = \log_{g_1} c_i \tag{11}$$

$$y_{1,i} + w_i y_{2,i} = \log_{g_1} d_i \tag{12}$$

Here  $i = 1 \dots n$ . And from the outputs of the LR encryption oracles the adversary gets the other  $n$  linear equations,  $i = 1, \dots, n$ :

$$r_{1,i} x_{1,i} + w_i r_{2,i} x_{2,i} + \alpha_i r_{1,i} y_{1,i} + \alpha_i w_i r_{2,i} y_{2,i} = \log_{g_1} v_i \tag{13}$$

So, the adversary has total  $3n$  linear relations (Equations 11,12,13) for  $4n$  unknowns. Assume now that the adversary submits an invalid ciphertexts to all of its decryption oracles:  $(u'_{1,i}, u'_{2,i}, e'_i, v'_i) \neq (u_{1,i}, u_{2,i}, e_i, v_i)$ , for every  $1 \leq i \leq n$ , where  $\log_{g_1} u'_{1,i} = r'_{1,i}$ ,  $\log_{g_1} u'_{2,i} = w_i r'_{2,i}$  and  $r'_{1,i} \neq r_{1,i}$ . Let  $\alpha'_i = H(u'_{1,i}, u'_{2,i}, e'_i)$ . Consider the following three cases:

Case 1.  $(u'_{1,i}, u'_{2,i}, e'_i) = (u_{1,i}, u_{2,i}, e_i)$  for some  $1 \leq i \leq n$ . In this case  $v'_i \neq v_i$  and the decryption oracle will reject.

Case 2.  $(u'_{1,i}, u'_{2,i}, e'_i) \neq (u_{1,i}, u_{2,i}, e_i)$  and  $\alpha'_i \neq \alpha_i$  for all  $1 \leq i \leq n$ . In this case the decryption oracle will reject unless the the relations corresponding to these invalid ciphertexts

$$r'_{1,i}x_{1,i} + w_i r_{2,i}x_{2,i} + \alpha'_i r'_{1,i}y_{1,i} + \alpha'_i w_i r'_{2,i}y_{2,i} = \log_{g_1} v'_i \quad (14)$$

happen to be linear dependent with the Equations 11,12,13. We show that Equations 11,12,13,14 are linearly independent by observing that

$$\det \begin{bmatrix} 1 & w_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & w_1 & \dots & 0 & 0 & 0 & 0 \\ r_{1,1} & w_1 r_{2,1} & \alpha_1 r_1 & \alpha_1 w_1 r_2 & \dots & 0 & 0 & 0 & 0 \\ r'_{1,1} & w_1 r'_{2,1} & \alpha'_1 r'_{1,1} & \alpha'_1 w_1 r'_{1,2} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & w_n & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & w_n \\ 0 & 0 & 0 & 0 & \dots & r_1 & w_n r_2 & \alpha_n r_1 & \alpha_n w_n r_2 \\ 0 & 0 & 0 & 0 & \dots & r'_{1,n} & w_n r'_{2,n} & \alpha'_n r'_{1,n} & \alpha'_n w_n r'^n_2 \end{bmatrix}$$

$$= w_1 \cdot \dots \cdot w_n (r_{2,1} - r_{1,1})(r'_{2,1} - r'_{1,1})(\alpha_1 - \alpha'_1) \cdot \dots \cdot (r_{2,n} - r_{1,n})(r'_{2,n} - r'^n_1)(\alpha^n - \alpha'^n) \neq 0,$$

if  $r_{1,i} \neq r_{2,i}, r'_{1,i} \neq r'_{2,i}, \alpha_i \neq \alpha'_i$  for every  $1 \leq i \leq n$ .

Case 3.  $(u'_1, u'_2, e') \neq (u_{1,i}, u_{2,i}, e_i)$  and  $\alpha' = \alpha_i$  for some  $1 \leq i \leq n$ . As [CrSh] shows, this case would imply that the family of hash functions is not universal one-way, which is a contradiction.

Summarizing these results we can say that the distribution of the hidden bit  $b$  is independent from the adversary's view except if one of the following events occurs: a decryption oracle accepts an invalid ciphertext, a collision is found for hash function  $H$ , or  $R$  outputs  $u_{1,i}, u_{2,i}$  of the form  $g_1^{y_i}, g_2^{y_i}$  for some  $y_i$  and  $i = 1, \dots, n$ . Let  $P_2$  denote the probability of the last event. Since the latter event can happen at any of  $n$  times the algorithm is called with probability  $1/q$ , we have  $P_2 = n/q$ . Let  $P'_0$  denote the probability that an invalid ciphertext is accepted by the simulator in the case  $b = 0$ , and  $P'_1$  the same when  $b = 1$ . Then

$$\begin{aligned} \Pr \left[ \mathbf{Exp}_{q,g}^{\text{ddh-rand}}(D) = 1 \right] &\leq \frac{1}{2} \cdot (0 + P'_0 + P_2 + \text{Adv}_{\mathcal{H}}^{cr}(t')) + \frac{1}{2} \cdot (1 - 0 - P'_1 + P_2 + \text{Adv}_{\mathcal{H}}^{cr}(t')) \\ &= \frac{1}{2} + \frac{1}{2}(P'_0 + P'_1) + P_2 + \text{Adv}_{\mathcal{H}}^{cr}(t'). \end{aligned} \quad (15)$$

According to [CrSh], the probability that a decryption oracle can accept an invalid ciphertext is at most

$$\sum_{i=1}^{q_d} \frac{1}{q-i+1} \leq \frac{q_d}{q-q_d+1} \leq \frac{2q_d}{q},$$

where the last bound assumes  $q_d \leq q/2$ . For the multi-user setting we multiply this bound by the number of users  $n$ . Thus for  $i = 0, 1$  we have

$$P_i \leq \frac{2nq_d}{q} \text{ and } P'_i \leq \frac{2nq_d}{q}.$$

Subtracting Equations (10) and (15) we get

$$\begin{aligned} \text{Adv}_{\mathcal{CS},(q,g)}^{n\text{-cca}}(t, 1, q_d) &\leq 2 \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') + (P_0 + P_1 + P'_0 + P'_1) + P_2 + \text{Adv}_{\mathcal{H}}^{\text{cr}}(t') \\ &\leq 2 \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') + \frac{8nq_d + 2q_en}{q} + 2\text{Adv}_{\mathcal{H}}^{\text{cr}}(t'). \end{aligned}$$

And for a general case when  $q_e$  LR encryption oracle queries are allowed we multiply the previous result by  $q_e$  using the hybrid argument.

$$\text{Adv}_{\mathcal{CS},(q,g)}^{n\text{-cca}}(t, q_e, q_d) \leq 2q_e \cdot \text{Adv}_{q,g}^{\text{ddh}}(t') + \frac{8q_enq_d + 2q_en}{q} + 2q_e \cdot \text{Adv}_{\mathcal{H}}^{\text{cr}}(t').$$

The claim about the time-complexity of  $D_A$  follows from the fact that the only overhead for  $D_A$  is the time of running the algorithm  $R$  a total of  $n$  times, where  $t' = t + O(n \cdot T_q^{\text{exp}})$ . ■

## Acknowledgments

We thank Victor Shoup for information about the concrete security of the reduction in [CrSh] 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.

## References

- [BPS] O. BAUDRON, D. POINTCHEVAL AND J. STERN, “Extended notions of security for multicast public key cryptosystems,” Manuscript.
- [BBM] M. BELLARE, A. BOLDYREVA, AND S. MICALI, “Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements,” Preliminary version of this paper, *Advances in Cryptology – Eurocrypt 2000 Proceedings*, Lecture Notes in Computer Science Vol. ??, B. Preneel ed., Springer-Verlag, 2000.
- [BDJR] 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.
- [BDPR] M. BELLARE, A. DESAI, D. POINTCHEVAL AND P. ROGAWAY, “Relations among notions of security for public-key encryption schemes,” *Advances in Cryptology – CRYPTO ’98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
- [BR] M. BELLARE, P. ROGAWAY, “Optimal asymmetric encryption – How to encrypt with RSA,” *Advances in Cryptology – EUROCRYPT ’94*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.
- [BS] M. BELLARE AND A. SAHAI, “Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization,” *Advances in Cryptology – CRYPTO ’99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.
- [BIGo] M. BLUM AND S. GOLDWASSER, “An efficient probabilistic public-key encryption scheme which hides all partial information,” *Advances in Cryptology – CRYPTO ’84*, Lecture Notes in Computer Science Vol. 196, R. Blakely ed., Springer-Verlag, 1984.
- [CrSh] R. CRAMER AND V. SHOUP, “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack,” *Advances in Cryptology – CRYPTO ’98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
- [DDN] D. DOLEV, C. DWORK, AND M. NAOR, “Non-malleable cryptography,” *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991.

- [ElG] T. ELGAMAL, “A public key cryptosystem and signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol 31, 1985, pp. 469–472.
- [GoMi] S. GOLDWASSER AND S. MICALI, “Probabilistic encryption,” *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.
- [Hå] J. HÅSTAD, “Solving simultaneous modular equations of low degree,” *SIAM J. on Computing* Vol. 17, No. 2, April 1988.
- [NaRe] 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.
- [PKCS] RSA LABORATORIES, “PKCS-1,” <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/>.
- [RaSi] C. RACKOFF AND D. SIMON, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” *Advances in Cryptology – CRYPTO ’91*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
- [Sh] V. SHOUP, “On formal models for secure key exchange,” *Theory of Cryptography Library Record 99-12*, <http://philby.ucsd.edu/cryptolib/>.
- [St] M. STADLER, “Publicly verifiable secret sharing,” *Advances in Cryptology – EUROCRYPT ’96*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.
- [TsYu] 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.

## A Proof of Lemma 5.2

**Proof:** The algorithm  $R$  works as follows:

```

Algorithm  $R(q, g, g^a, g^b, g^c, x)$ 
If  $x = 0$  then  $s_1 \leftarrow 0$  else  $s_1 \xleftarrow{R} Z_q$  EndIf
   $s_2, r \xleftarrow{R} Z_q$ 
   $g^{a'} \leftarrow g^a \cdot g^{s_1}$ 
   $g^{b'} \leftarrow (g^b)^r \cdot g^{s_2}$ 
   $g^{c'} \leftarrow (g^c)^r \cdot (g^a)^{s_2} \cdot (g^b)^{r \cdot s_1} \cdot g^{s_1 s_2}$ 
Return  $(g^{a'}, g^{b'}, g^{c'})$ 

```

When  $x \neq 0$  the above is exactly the algorithm of [NaRe].

For the analysis we let  $e = c - ab \pmod q$ . Then we have

$$\begin{cases} a' &= a + s_1 \pmod q \\ b' &= br + s_2 \pmod q \\ c' &= cr + as_2 + brs_1 + s_1s_2 \pmod q = a'b' + er \pmod q \end{cases} \quad (16)$$

Now we want to verify the claims in the table. If  $c = ab \pmod q$  then we have  $e = 0$  and hence from Equation (16) we have  $c' = a'b' + er = a'b' \pmod q$ . Also if  $x = 0$  then from Equation (16) we have  $a' = a + s_1 = a \pmod q$ . This gives us all the claims with respect to fixed quantities. It remains to verify the claims about the random quantities.

To argue the randomness of the claimed outputs we fix  $a, b, c, x$ . Let  $A', B', C'$  denote the random variables with range  $Z_q$  whose values are  $a', b', c'$  respectively, the probability being over the choices of  $s_2, r$  and (if  $x \neq 0$ ) also  $s_1$ . We want to claim certain things about their distribution depending on  $a, b, c, x$ . To do this fix target quantities  $a', b', c' \in Z_q$  subject to any known restrictions already imposed by the choices of  $a, b, c, x$ .

First consider  $x = e = 0$ . We know from the above that  $a' = a$  and  $c' = a'b'$ . To complete the claims for this case we want to check that

$$\Pr [ B' = b' \mid A' = a, C' = ab' ] = \frac{1}{q},$$

the probability being over the choices of  $s_2, r$ . The above is true because with  $b, b'$  fixed, for any fixed choice of  $r$  there is a unique choice of  $s_2$  such that  $b' = br + s_2 \pmod q$ .

Now consider  $x = 0$  and  $e \neq 0$ . We know from the above that  $a' = a$ . To complete the claims for this case we want to check that

$$\Pr [ B' = b', C' = c' \mid A' = a ] = \frac{1}{q^2}, \quad (17)$$

the probability being over the choices of  $s_2, r$ . From Equation (16), the choices of  $s_2, r$  that result in  $B' = b'$  and  $C' = c'$  given that  $A' = a$  are exactly the solutions to the matrix equation

$$\begin{bmatrix} 1 & b \\ 0 & e \end{bmatrix} \cdot \begin{bmatrix} s_2 \\ r \end{bmatrix} = \begin{bmatrix} b' \\ c' - a'b' \end{bmatrix}$$

The determinant of the above matrix is  $e$  which by assumption is non-zero, meaning the solution is unique. So when  $s_2, r$  are chosen at random there the probability is exactly  $1/q^2$  that they are a solution, which implies Equation (17).

When  $x \neq 0$  the claims follow from [NaRe, Lemma 3.2]. We provide proofs for completeness and because our proof approach is slightly different.

First consider  $x \neq 0$  and  $e = 0$ . We know from the above that  $c' = a'b'$ . To complete the claims for this case we want to check that

$$\Pr [ A' = a', B' = b' \mid C' = a'b' ] = \frac{1}{q^2}, \quad (18)$$

the probability being over the choices of  $s_1, s_2, r$ . Fix a choice of  $r$ . Then from Equation (16), there are unique choices of  $s_1, s_2$  so that the equations hold. So the probability that the equations hold, over  $s_1, s_2, r$ , is  $(q/q)(1/q^2) = 1/q^2$ , which implies Equation (18).

Finally consider  $x \neq 0$  and  $e \neq 0$ . To complete the claims for this case we want to check that

$$\Pr [ A' = a', B' = b', C' = c' ] = \frac{1}{q^3}, \quad (19)$$

the probability being over the choices of  $s_1, s_2, r$ . From Equation (16), the choices of  $s_1, s_2, r$  that result in  $A' = a', B' = b'$  and  $C' = c'$  are exactly the solutions to the matrix equation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & e \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \\ r \end{bmatrix} = \begin{bmatrix} a' - a \\ b' \\ c' - a'b' \end{bmatrix}.$$

The determinant of the above matrix is

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & e \end{vmatrix} = 1 \cdot \begin{vmatrix} 1 & b \\ 0 & e \end{vmatrix} = e \pmod q.$$

Since  $e \neq 0$  the solution  $(s_1, s_2, r)$  is unique. So when  $s_1, s_2, r$  are chosen at random then the probability that they are a solution is exactly  $1/q^3$ , which implies Equation (19). ■