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‡

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 ElGamal and Cramer-Shoup.

Keywords: Encryption, public-key cryptosystems, ElGamal, Diffie-Hellman, decisional Diffie-Hellman problem.

*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. 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. 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

1 Introduction

This paper addresses the security of public-key (asymmetric) 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 ElGamal 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].[1] 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 $C_i = M^3 \bmod N_i$, 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 C_1, C_2, C_3 where $C_i = M^3 \bmod N_i$ for $i = 1, 2, 3$. Then an adversary given C_1, C_2, C_3 can recover M. (Using the fact that N_1, N_2, N_3 are relatively prime, C_1, C_2, C_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.

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.

[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 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*.[2] 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

[2] 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[3]. We fix a public-key cryptosystem $\mathcal{PE}$ used by all users. Theorem 4.1 says that the probability that any 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 $q_e n$ times the probability that some adversary with running time closely related to t can compromise security in the standard sense of indistinguishability. Notationally, for any adversary A there exists an adversary B such that for any security parameter k and $\mathrm{atk} = \{\mathrm{cpa}, \mathrm{cca}\}$, $\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE},A}(k) \leq q_e n \cdot \mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B}(k)$ where the running time of B is approximately the running time of A. (Here the technical term for the "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 polynomially 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 for any adversary B with some appropriately large running time $\mathsf{Adv}^{\text{su-cpa}}_{\mathcal{PE},B}(|p|) \leq 2^{-60}$. 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 $q_e n \cdot \mathsf{Adv}^{\text{su-cpa}}_{\mathcal{PE},B}(k)$ is ≈ 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 $q_e n \approx 2^{60}$. 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 $q_e n$. So we can't expect to reduce the security loss in general. But we can still hope

[3] We allow n, q_e be polynomials in the security parameter.

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 ElGamal in the multi-user setting

The ElGamal 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, TsYu]). The reduction is essentially tight, and in our language says that for any adversary B, $\mathsf{Adv}^{\text{su-cpa}}_{\mathcal{EG},B}(k)$ —the probability that B can break the ElGamal scheme via a chosen-plaintext attack in the single-user setting— is at most $2\mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k)$ —twice the probability of solving the DDH problem by some adversary D, in the same amount of time. (Here $\mathcal{G}$ specifies g, q, where g is a generator of the group G of a large prime order q, where $|q| = k$, the parameters used in $\mathcal{EG}$.) We thus have a complete and satisfactory picture of the security of the ElGamal scheme in the single-user setting; our concern now is the multi-user setting.

Theorem 4.1 together with the above implies that for any adversary A, $\mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{EG},A}(k)$ —the probability that A can break the ElGamal scheme via a chosen-plaintext attack in the presence of n users each encrypting q_{e} messages— is upper bounded by $2q_{\text{e}}n \cdot \mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k)$ —$2q_{\text{e}}n$ times the probability of solving the DDH problem by some D with the time complexity approximately the same as that of A. We show in Theorem 5.3 that via an improved reduction the factor of $q_{\text{e}}n$ can be essentially eliminated. In other words, the maximum probability of breaking the ElGamal 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 ElGamal scheme provides security against chosen-plaintext attack. Nowadays there is much interest and need for practical schemes provably achieving security against chosen-ciphertext attack. The Cramer-Shoup scheme [CrSh] achieves 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 $q_{\text{e}}n$. We present in Theorem 5.5 an improved reduction which (roughly speaking) reduces the factor of $q_{\text{e}}n$ 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 ElGamal 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 ElGamal 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 transfered. 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

2.1 Notation

$\mathbb{N}$ denotes the set of natural numbers. For $k \in \mathbb{N}$, $\mathbb{Z}_k$ denotes the ring of integers modulo k. We denote by $\{0,1\}^*$ the set of all binary strings of finite length. We will refer to members of $\{0,1\}^*$ as strings. If X is string then $|X|$ denotes its length in bits and if X, Y are strings then $X \| Y$ denotes the concatenation of X and Y. If S is a set then $X \xleftarrow{\$} S$ denotes that X is selected uniformly at random from S. For convinience for any $k \in N$ we will often write $X_1, X_2, \ldots, X_k \xleftarrow{\$} S$ as a shorhand for $X_1 \xleftarrow{\$} S, X_2 \xleftarrow{\$} S, \ldots, X_n \xleftarrow{\$} S$. If $k \in \mathbb{N}$ then 1^k denotes the string consisting of k

consecutive "1" bits. If A is a deterministic (resp. randomized) algorithm and $k \in N$, then the notation $X \leftarrow A(X_1, X_2, \ldots, X_k)$ (resp. $X \xleftarrow{\$} A(X_1, X_2, \ldots, X_k)$) denotes that X is assigned the outcome of the experiment of running A on inputs $X_1, X_2, \ldots, X_k$. When describing algorithms, if X is a variable and Y is a string, then $X \leftarrow Y$ denotes that X is assigned the value of Y. For $k \in N$ we say that an algorithm A runs in poly(k) time if A given inputs whose length is bounded by a polynomial in k always halts in time bounded by a polynomial in k.

2.2 Syntax

We recall the standard definitions of asymmetric encryption schemes. We extend the usual syntax to allow a common key generation algorithm. *Sasha: cite where did it before?* Let $k \in \mathbb{N}$ be a security parameter. An *asymmetric (public-key) encryption scheme* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms:

- The randomized *common-key generation* algorithm $\mathcal{G}$ takes as input 1^k and, in poly(k)-time, returns a *common key* I; we write $I \xleftarrow{\$} \mathcal{G}(k)$.
- The randomized *key generation* algorithm $\mathcal{K}$ takes as input the common key I and, in poly(k)-time, returns a pair (pk, sk) consisting of a public key and a corresponding secret key; we write $(pk, sk) \xleftarrow{\$} \mathcal{K}(I)$.
- The randomized *encryption* algorithm $\mathcal{E}$ takes input the common and the public keys I, pk and a plaintext M and, in poly(k)-time, returns a ciphertext; we write $C \xleftarrow{\$} \mathcal{E}_{I,pk}(M)$.
- The deterministic, *decryption* algorithm $\mathcal{D}$ takes the common and the secret key sk and a ciphertext C to return in poly(k)-time the corresponding plaintext or a special symbol $\perp$ indicating that the ciphertext was invalid; we write $M \leftarrow \mathcal{D}_{I,sk}(C)$ (or $\perp \leftarrow \mathcal{D}_{I,sk}(C)$.)

Associated to each common key I is a *message space* $\mathrm{MsgSp}(I)$ from which M is allowed to be drawn. We require that $\mathcal{D}_{I,sk}(\mathcal{E}_{I,pk}(M)) = M$ for all $M \in \mathrm{MsgSp}(I)$.

Further in the paper we will use the terms "plaintext" and "message" interchangeably. As an example to illustrate the addition of a common-key generation algorithm to the usual syntax, consider a Diffie-Hellman based scheme. Here the common key I could include a description of a group and a generator for this group. Different parties may have different keys, but the algorithms are all in the same group.

2.3 Privacy for asymmetric encryption schemes

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. An adversary B runs in two stages. In the "`find`" stage it takes the common key and 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 $\mathcal{D}_{I,sk}(\cdot)$ 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 in the single user setting] Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let $k \in \mathbb{N}$ be the security parameter. Let $B_{\mathrm{cpa}}, B_{\mathrm{cca}}$ be adversaries where the latter has access to an oracle. For $b \in \{0, 1\}$ define the experiments

Experiment $\mathbf{Exp}^{\text{su-cpa-b}}_{\mathcal{PE},B_{\text{cpa}}}(1^k)$

$I \xleftarrow{\$} \mathcal{G}(1^k)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(I)$
$(M_0, M_1, s) \xleftarrow{\$} B_{\text{cpa}}(\mathsf{find}, I, pk)$
$C \xleftarrow{\$} \mathcal{E}_{I,pk}(M_b)$
$d \xleftarrow{\$} B_{\text{cpa}}(\mathsf{guess}, C, s)$
`Return` d

Experiment $\mathbf{Exp}^{\text{su-cca-b}}_{\mathcal{PE},B_{\text{cca}}}(1^k)$

$I \xleftarrow{\$} \mathcal{G}(1^k)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}(I)$
$(M_0, M_1, s) \xleftarrow{\$} B_{\text{cca}}^{\mathcal{D}_{I,sk}(\cdot)}(\mathsf{find}, I, pk)$
$C \xleftarrow{\$} \mathcal{E}_{I,pk}(M_b)$
$d \xleftarrow{\$} B_{\text{cca}}^{\mathcal{D}_{I,sk}(\cdot)}(\mathsf{guess}, C, s)$
`Return` d

We say that the adversary is *legitimate* if it outputs $M_0, M_1 \in \text{MsgSp}(I)$ such that $|M_0| = |M_1|$ above and when atk = cca, B_{cca} does not make oracle query C in the guess stage. For atk $\in$ {cpa, cca} the *advantages* of the legitimate adversaries are the function of the security parameter defined as follows:

$$\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B_{\text{atk}}}(k) \quad = \quad \Pr\left[\,\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE},B_{\text{atk}}}(1^k) = 0\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE},B_{\text{atk}}}(1^k) = 0\,\right].$$

Sasha: is it ok to have k and 1^k*?* The scheme $\mathcal{PE}$ is said to be *polynomially-secure against chosen-plaintext attack (resp. chosen-ciphertext attack)* in the single-user setting if the function $\mathsf{Adv}^{\text{su-cpa}}_{\mathcal{PE},B_{\text{cpa}}}(\cdot)$ (resp. $\mathsf{Adv}^{\text{su-cca}}_{\mathcal{PE},B_{\text{cca}}}(\cdot)$) is negligible for any legitimate poly(k)-time adversary. ▮

Recall that a function f is called *negligible* if it approaches zero faster than the reciprocal of any polynomial, i.e., for any polynomial p, there exists $n_p \in \mathbb{N}$ such that for all $n \geq n_p$, $f(n) \leq 1/p(n)$. The "time-complexity" is measured as a function of a security parameter and is the worst case execution time of the associated experiment, 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.

3 Security in the multi-user setting

We envision a set of n users. All users use a common, fixed cryptosystem $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. User i has a public key pk_i and holds the matching secret key sk_i. All public and secret keys are generated using the same common key. 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 common key and 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 start with chosen-plaintext attacks.

Some intuition. To get a start on the modelling, consider the following game. We imagine that a message M is chosen at random from some known distribution, and the adversary is provided with $\mathcal{E}_{I,pk_1}(M)$, 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 $\mathcal{E}_{I,pk_j}(\Delta(M))$, 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 the common key and all the public keys. But unlike 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 two messages M_0, M_1 (of equal length) and computes and returns a ciphertext $\mathcal{E}_{I,pk_i}(M_b)$. 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 $\mathcal{E}_{I,pk_1}(M_b)$ and $\mathcal{E}_{I,pk_3}(M_b)$. More generally, if the adversary wanted to obtain the ciphertext of the result of a message modification function Δ 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 $\mathcal{E}_{I,pk_1}(M_b)$ and $\mathcal{E}_{I,pk_3}(\Delta(M_b))$. 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

$$\mathrm{LR}(M_0, M_1, b) \;=\; M_b$$

for all equal-length strings M_0, M_1, and for any $b \in \{0,1\}$. Let $k \in \mathbb{N}$ be the security parameter and let n be a polynomial. Let I be the common key. The adversary A is given n oracles, which we call *LR (left-or-right) encryption oracles*,

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

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 [BDJR] in the symmetric setting.) The oracle $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$, given query M_0, M_1 where $M_0, M_1 \in \mathrm{MsgSp}(I)$ 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 $C \leftarrow \mathcal{E}_{I,pk_i}(M_b)$ and returns C as the answer to the oracle query. The adversary also gets as input the common key and $n(k)$ public keys.

In the case of a chosen-ciphertext attack the adversary is also given a decryption oracle with respect to each of the $n(k)$ public keys. Note we must disallow a query C to $\mathcal{D}_{I,sk_i}(\cdot)$ if C is an output of oracle $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$. 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. *Sasha: is it clear? maybe add it as a second parameter?*

Definition 3.1 [Indistinguishability of encryptions in the multi-user setting] Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let k be the security parameter and let n be a polynomial. Let $A_{\rm cpa}$, $A_{\rm cca}$ be adversaries. Both have access to $n(k) \geq 1$ oracles, each of which takes as input any two strings of equal length, and $A_{\rm cca}$ has access to an additional $n(k)$ oracles each of which take a single input. For $b \in \{0,1\}$ define the experiments:

Experiment $\mathbf{Exp}^{\text{n-mu-cpa-b}}_{\mathcal{PE},A_{\rm cpa}}(1^k)$
$I \xleftarrow{\$} \mathcal{G}(1^k)$
For $i = 1, \ldots, n(k)$ do $(pk_i, sk_i) \xleftarrow{\$} \mathcal{K}(I)$ EndFor
$d \xleftarrow{\$} A_{\rm cpa}^{\mathcal{E}_{I,pk_1}(\mathrm{LR}(\cdot,\cdot,b)),\ldots,\mathcal{E}_{I,pk_{n(k)}}(\mathrm{LR}(\cdot,\cdot,b))}\left(I, pk_1, \ldots, pk_{n(k)}\right)$
Return d

Experiment $\mathbf{Exp}^{\text{n-mu-cca-b}}_{\mathcal{PE},A_{\rm cca}}(1^k)$
$I \xleftarrow{\$} \mathcal{G}(1^k)$
For $i = 1, \ldots, n(k)$ do $(pk_i, sk_i) \xleftarrow{\$} \mathcal{K}(I)$ EndFor
$d \xleftarrow{\$} A_{\rm cca}^{\mathcal{E}_{I,pk_1}(\mathrm{LR}(\cdot,\cdot,b)),\ldots,\mathcal{E}_{I,pk_{n(k)}}(\mathrm{LR}(\cdot,\cdot,b)),\mathcal{D}_{I,sk_1}(\cdot),\ldots,\mathcal{D}_{I,sk_{n(k)}}(\cdot)}\left(I, pk_1, \ldots, pk_{n(k)}\right)$
Return d

It is mandated that a query of a legitimate adversary to any LR oracle consists of two messages in $\mathrm{MsgSp}(I)$ and of *equal* length and that for each $i = 1, \ldots, n(k)$ adversary $A_{\rm cca}$ does not query $\mathcal{D}_{I,sk_i}(\cdot)$ on an output of $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$, $k \in \mathbb{N}$ and polynomial n we define the *advantages* of the legitimate adversaries as the function of the security parameter as follows:

$$\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE},A_{\rm atk}}(k) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{n-mu-atk-0}}_{\mathcal{PE},B_{\rm atk}}(1^k) = 0\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{n-mu-atk-1}}_{\mathcal{PE},B_{\rm atk}}(1^k) = 0\,\right].$$

The scheme $\mathcal{PE}$ is said to be *polynomially-secure against chosen-plaintext attack (resp. chosen-ciphertext attack) in the multi-user setting* if the function $\mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{PE},A_{\rm cpa}}(\cdot)$ (resp. $\mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{PE},A_{\rm cca}}(\cdot)$) is negligible for any $\mathrm{poly}(k)$-time legitimate adversary. ∎

Remark 3.2 Notice that when $n = 1$ and only one LR encryption query is allowed 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 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.

4 A general reduction and its tightness

4.1 A general reduction

Fix a public-key encryption scheme $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. 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 $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let n, q_e be polynomials. Then for for any $k \in \mathbb{N}, \text{atk} \in \{\text{cpa}, \text{cca}\}$ and for any adversary A_{atk} making at most $q_e(k)$ LR encryption oracle queries there exist an adversary B_{atk} such that

$$\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE},A_{\text{atk}}}(k) \quad \leq \quad q_e(k)n(k) \cdot \mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B_{\text{atk}}}(k),$$

where the running time of B_{atk} is that of A_{atk} plus $O(\log(q_e n(k)))$. If $\text{atk} = \text{cca}$ then B_{cca} does at most the number of decryption oracle queries A_{cca} does. ▌

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

Corollary 4.2 Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme that is polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the single-user setting. Then $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is also polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the multi-user setting. ▌

Proof of Theorem 4.1: We provide a proof common for $\text{atk} \in \{\text{cpa}, \text{cca}\}$ specifying the differences pertaining to the case when $\text{atk} = \text{cca}$.

Let A_{atk} be a legitimate adversary attacking the encryption scheme $\mathcal{PE}$ in the multi-user setting. Assume it makes at most $q_e(k)$ queries to any of its $n(k)$ oracles, has time-complexity at most t and if $\text{atk} = \text{cca}$ makes $q_d(k)$ decryption oracle queries. We will design a legitimate adversary B_{atk} attacking the same scheme in the single-user setting so that

$$\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B_{\text{atk}}}(k) \quad \geq \quad \frac{1}{q_e(k)n(k)} \cdot \mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE},A_{\text{atk}}}(k)\ . \tag{1}$$

Furthermore, B_{atk} will have running time at most $t + O(\log(q_e n(k)))$. This would imply the statement of the theorem. So it remains to design B_{atk}. We now proceed to the full proof.

We begin by describing some hybrid experiments associated to A_{atk} for $\text{atk} = \{\text{cpa}, \text{cca}\}$. It is convenient to parameterize the hybrids via a single integer l ranging from 0 to $q_e(k)n(k)$ which counts the number of oracle queries replied to by encrypting the left message.

Experiment $\mathbf{ExpH}^{l}_{\mathcal{PE},A_{\text{atk}}}(1^k)$ $\quad [\,0 \leq l \leq q_e(k)n(k)]$

$I \xleftarrow{\$} \mathcal{G}(1^k)$

`For` $i = 1, \ldots, n(k)$ do $(pk_i, sk_i) \xleftarrow{\$} \mathcal{K}(I)$ `EndFor`

$ctr \leftarrow 0$

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

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

Run $A_{\text{atk}}\left(I, pk_1, \ldots, pk_{n(k)}\right)$ as follows:

When A_{atk} makes a query M_0, M_1 to oracle $\mathcal{E}_{I,pk_i}(\text{LR}(\cdot,\cdot,b))$:

`If` $i < c$ `then` $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_0)$; `return` C to A_{atk} `EndIf`

`If` $i > c$ `then` $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_1)$; `return` C to A_{atk} `EndIf`

`If` $i = c$ `then`

$ctr \leftarrow ctr + 1$

`If` $ctr \leq r$ `then` $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_0)$; `return` C to A_{atk} `EndIf`

If $ctr > r$ then $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_1)$; return C to A_{atk} EndIf

EndIf

If atk = cca and A_{cca} makes a decryption oracle query C' to oracle $\mathcal{D}_{I,sk_i}(\cdot)$:

$M \leftarrow \mathcal{D}_{I,sk_i}(C)$; return M to A_{cca}

EndIf

Eventually A_{atk} halts outputting a bit d

Return d

Let $P_l \stackrel{\text{def}}{=} \Pr\left[\mathbf{ExpH}^{l}_{\mathcal{PE},A_{\text{atk}}}(1^k) = 0\right]$ denote the probability that experiment $\mathbf{ExpH}^{l}_{\mathcal{PE},A_{\text{atk}}}(1^k)$ returns 0, for $l = 0, 1, \ldots, q_{\text{e}}(k)n(k)$. Now we claim that

$$\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE},A_{\text{atk}}}(k) \;=\; P_{q_{\text{e}}(k)n(k)} - P_0 \;. \tag{2}$$

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

$$\Pr\left[\mathbf{Exp}^{\text{n-mu-atk-0}}_{\mathcal{PE},A_{\text{atk}}}(1^k) = 0\right] \;=\; P_{q_{\text{e}}(k)n(k)} \quad \text{and} \quad \Pr\left[\mathbf{Exp}^{\text{n-mu-atk-1}}_{\mathcal{PE},A_{\text{atk}}}(1^k) = 0\right] \;=\; P_0 \;,$$

and after subtraction Equation (2) follows. The two equations above are justified as follows. In experiment $\mathbf{ExpH}^{q_{\text{e}}(k)n(k)}_{\mathcal{PE},A_{\text{atk}}}(1^k)$ we have $l = q_{\text{e}}(k)n(k)$ so $c = n(k)$ and $r = q_{\text{e}}(k)$. 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_{atk}'s "view" is the same as in experiment $\mathbf{Exp}^{\text{n-mu-atk-0}}_{\mathcal{PE},A_{\text{atk}}}(k)$. On the other hand in experiment $\mathbf{ExpH}^{0}_{\mathcal{PE},A_{\text{atk}}}(1^k)$ 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_{atk}'s "view" is the same as in experiment $\mathbf{Exp}^{\text{n-mu-atk-1}}_{\mathcal{PE},A_{\text{atk}}}(k)$.

Now we turn to the description of B_{atk}. 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 a common and a public keys I, pk. However, we will not explicitly separate the operation of B_{atk} into the find and guess stages. Instead, think of B_{atk} supplied with the oracle $\mathcal{E}_{I,pk}(\text{LR}(\cdot,\cdot,b))$ and allowed to make only a single query of this oracle. If atk = cca then B_{cca} is also given oracle $\mathcal{D}_{I,sk}(\cdot)$. To capture this and to simplify the notation we write below $B^{\mathcal{D}(\cdot)}_{\text{atk}}$, where $\mathcal{D}(\cdot) = \varepsilon$ if atk = cpa, and $\mathcal{D}(\cdot) = \mathcal{D}_{I,sk}(\cdot)$ in case when atk = cca. Finally B_{atk} must try to guess the challenge bit b.

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

$l \xleftarrow{\$} \{1, \ldots, q_{\text{e}}(k)n(k)\}$

Let r, c be such that $l = (c-1)q_{\text{e}}(k) + r$ and $1 \le r \le q_{\text{e}}(k)$ and $1 \le c \le n(k)$

For $i \in \{1, \ldots, c-1, c+1, \ldots, n(k)\}$ do $(pk_i, sk_i) \xleftarrow{\$} \mathcal{K}(I)$ EndFor

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

Run $A_{\text{atk}}\left(I, pk_1, \ldots, pk_{n(k)}\right)$ as follows:

When A_{atk} makes a query M_0, M_1 to oracle $\mathcal{E}_{I,pk_i}(\text{LR}(\cdot,\cdot,b))$:

If $i < c$ then $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_0)$; return C to A_{atk} EndIf

If $i > c$ then $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_1)$; return C to A_{atk} EndIf

If $i = c$ then

$ctr \leftarrow ctr + 1$

If $ctr < r$ then $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_0)$; return C to A_{atk} EndIf

If $ctr > r$ then $C \xleftarrow{\$} \mathcal{E}_{I,pk_i}(M_1)$; return C to A_{atk} EndIf

If $ctr = r$ then

Let $C \xleftarrow{\$} \mathcal{E}_{I,pk}(\text{LR}(M_0, M_1, b))$ [Let M_0, M_1 be the single allowed oracle query]

return C to A_{atk}

EndIf
EndIf
If atk = cca and $A_{\rm cca}$ makes a decryption oracle query C' to oracle $\mathcal{D}_{I,sk_i}(\cdot)$:
If $i = c$ then $M \leftarrow \mathcal{D}(C')$; return M to $A_{\rm atk}$ EndIf
If $i \neq c$ then $M \leftarrow \mathcal{D}_{I,sk_i}(C')$; return M to $A_{\rm atk}$ EndIf
EndIf
Eventually $A_{\rm atk}$ halts outputting a bit d
Return d

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

$$\begin{aligned}
\Pr\left[\, \mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE},B_{\rm atk}}(1^k) = 0 \,\right] &= \frac{1}{q_{\rm e}(k)n(k)} \cdot \sum_{l=1}^{q_{\rm e}(k)n(k)} P_l \ , \\
\Pr\left[\, \mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE},B_{\rm atk}}(1^k) = 0 \,\right] &= \frac{1}{q_{\rm e}(k)n(k)} \cdot \sum_{l=1}^{q_{\rm e}(k)n(k)} P_{l-1} \ . \qquad (3)
\end{aligned}$$

Subtracting and exploiting the collapse of the sums we get

$$\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B_{\rm atk}}(k) \;=\; \frac{1}{q_{\rm e}(k)n(k)} \cdot \sum_{l=1}^{q_{\rm e}(k)n(k)} P_l - P_{l-1} \;=\; \frac{1}{q_{\rm e}(k)n(k)} \cdot [P_{q_{\rm e}(k)n(k)} - P_0] \ .$$

Equation (1) follows by applying Equation (2), so it remains to justify Equations (3). Each value of l in $\{1, \ldots, q_{\rm e}(k)n(k)\}$ is equally likely for $B_{\rm atk}$ so let's focus on one choice of l. It is always true that all decryption oracle queries are answered truthfully, using the correct secret key and all queries to the first $c-1$ oracles, and the first $r-1$ queries to the c-th LR oracle, are answered by encrypting the left message, while all queries to $c+1, \ldots, n(k)$-th LR oracles, and the last $q_{\rm e}(k) - r$ queries to the c-th LR oracle, are answered by encrypting the right message. When we run experiment $\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE},B}(1^k)$, the r-th query to the c-th LR oracle is answered by encrypting the left message, so the experiment is the same as $\mathbf{ExpH}^{l}_{\mathcal{PE},A_{\rm atk}}(1^k)$. When we run experiment $\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE},B_{\rm atk}}(1^k)$, the r-th query to the c-th LR oracle is answered by encrypting the right message, so the experiment is the same as $\mathbf{ExpH}^{l-1}_{\mathcal{PE},A_{\rm atk}}(1^k)$.

Finally we should justify the claim about the running time of $B_{\rm atk}$. 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 $n(k)$ key pairs, so that these actions of $B_{\rm atk}$ are not an overhead. Taking the conventions into account the only overhead for $B_{\rm atk}$ is to pick the random number l and execute some conditional statements. ∎

4.2 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. Accordingly we assume that there exists an asymmetric encryption scheme $\mathcal{PE}$ which is polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attacks in the single-user setting. Namely, for atk = cpa (resp. for atk = cca) $\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},D_{\rm atk}}(\cdot)$ is negligible for every poly(k)-time legitimate adversary $D_{\rm atk}$. We want to modify this scheme into another scheme $\mathcal{PE}'$ and present a poly(k)-time legitimate adversary $A_{\rm atk}$ such that for any poly(k)-time legitimate adversary $B_{\rm atk}$, $\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE}',A_{\rm atk}}(k)$ is $\Omega(q_{\rm e}(k)n(k))$ times $\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE}',B_{\rm atk}}(k)$. The following proposition does this, modulo some technicalities.

Proposition 4.3 Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attacks in the single-user setting. Fix any $k \in \mathbb{N}$ and any polynomials $q_{\rm e}, q_{\rm d}, n$. Then there exists another public-key encryption scheme $\mathcal{PE}'$ and

1. for atk $= \{\text{cpa}, \text{cca}\}$ there exists a poly(k)-time legitimate adversary A such that

$$\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE}', A_{\rm atk}}(k) \geq 0.6,$$

2. for atk $= \{\text{cpa}, \text{cca}\}$ and for any poly(k)-time legitimate adversary B there exists a poly(k)-time legitimate adversary D such that

$$\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE}', B_{\rm atk}}(k) \leq \frac{1}{q_{\rm e}(k)n(k)} + \mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE}, D_{\rm atk}}(k) \ .$$

Proof: We first define $\mathcal{PE}' = (\mathcal{G}, \mathcal{K}, \mathcal{E}', \mathcal{D}')$. Its algorithms $\mathcal{G}, \mathcal{K}$ are the same as the corresponding algorithms of $\mathcal{PE}$. The encryption algorithm $\mathcal{E}'$ takes inputs the common and the public keys and a message M and with probability $1/q_{\rm e}(k)n(k)$ returns $0\|M$, and with probability $1 - 1/q_{\rm e}(k)n(k)$ runs $\mathcal{E}$ on the same inputs and returns $1\|C$, where C is the output of $\mathcal{E}$. The decryption algorithm $\mathcal{D}'$ takes inputs the common and the secret keys and a ciphertext C, parses C as $b\|C'$. If $b = 0$ then $\mathcal{D}'$ outputs C', otherwise it runs $\mathcal{D}$ on the same inputs and returns its output.

We now justify the first claim of the proposition by presenting a poly(k)-time legitimate adversary $A_{\rm atk}$. $A_{\rm atk}$ simply queries M_0, M_1, where $M_0 \neq M_1$, $q_{\rm e}(k)n(k)$ times to any of its LR encryption oracles. If it gets a response $0\|M_0$, it returns 0, if it ever gets a response $0\|M_1$, it returns 1, otherwise it aborts.

We now analyze $A_{\rm atk}$. The adversary can guess the challenge bit correctly unless it aborts. The probability that $A_{\rm atk}$ aborts after making $q_{\rm e}(k)n(k)$ queries is $(1 - 1/q_{\rm e}(k)n(k))^{q_{\rm e}(k)n(k)}$. Hence we have

$$\begin{aligned}
\mathsf{Adv}^{\text{n-mu-atk}}_{\mathcal{PE}', A_{\rm atk}}(k) &= \Pr\left[\, \mathbf{Exp}^{\text{n-mu-atk-0}}_{\mathcal{PE}', A_{\rm atk}}(1^k) = 0 \,\right] - \Pr\left[\, \mathbf{Exp}^{\text{n-mu-atk-1}}_{\mathcal{PE}', A_{\rm atk}}(1^k) = 0 \,\right] \\
&= \left(1 - \left(1 - \frac{1}{q_{\rm e}(k)n(k)}\right)^{q_{\rm e}(k)n(k)}\right) - 0 \geq 1 - \frac{1}{e} \geq 0.6.
\end{aligned}$$

The adversary is obviously legitimate and does not make any decryption oracle queries and therefore the claim is true for atk $= \{\text{cpa}, \text{cca}\}$. Clearly $A_{\rm atk}$ runs in poly-(k) time.

We now justify the second claim of the proposition. Let $B_{\rm atk}$ be a poly(k)-time legitimate adversary attacking privacy of $\mathcal{PE}'$ against chosen-plaintext (resp. chosen-ciphertext) attacks in the single-user setting. We present a poly(k)-time adversary $D_{\rm atk}$ attacking privacy of the original scheme $\mathcal{PE}$ against chosen-plaintext (resp. chosen-ciphertext) attacks in the single-user setting.

$D_{\rm atk}$ is given a common and a public key pair I, pk. If atk $=$ cca then $D_{\rm cca}$ is given access to the decryption oracle $\mathcal{D}_{I,sk}(\cdot)$. With probability $1/q_{\rm e}(k)n(k)$ it aborts. Otherwise, it runs $B_{\rm atk}$ on I, pk. When $B_{\rm atk}$ makes a LR encryption oracle query M_0, M_1, $D_{\rm atk}$ outputs it as its own query, receives a challenge ciphertext C and forwards $1\|C$ back to $B_{\rm atk}$. If atk $=$ cca and $B_{\rm cca}$ makes a decryption oracle query C', $D_{\rm cca}$ parses C as $b\|C'$. If $b = 0$ then $D_{\rm cca}$ forwards C' back to $B_{\rm cca}$, otherwise $D_{\rm cca}$ queries C' to its own decryption oracle $\mathcal{D}$ and forwards its reply to $B_{\rm cca}$. Finally $D_{\rm atk}$ outputs the bit output by $B_{\rm atk}$.

Analyzing $D_{\rm atk}$ we have

$$\begin{aligned}
\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},D_{\rm atk}}(k) &= \Pr\left[\,\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE},D_{\rm atk}}(1^k)=0\,\right]-\Pr\left[\,\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE},D_{\rm atk}}(1^k)=0\,\right]\\
&\geq \left(1-\frac{1}{q_{\rm e}(k)n(k)}\right)\cdot\left(\Pr\left[\,\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE}',B_{\rm atk}}(1^k)=0\,\right]-\Pr\left[\,\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE}',B_{\rm atk}}(1^k)=0\,\right]\right)\\
&= \left(1-\frac{1}{q_{\rm e}(k)n(k)}\right)\cdot\mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE},B_{\rm atk}}(k)\geq \mathsf{Adv}^{\text{su-atk}}_{\mathcal{PE}',B_{\rm atk}}(k)-\frac{1}{q_{\rm e}(k)n(k)}\ .
\end{aligned}$$

The above is justified as follows. If $D_{\rm atk}$ does not abort (this happens with probability $1-1/q_{\rm e}(k)n(k)$), and it is in $\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE},D_{\rm atk}}(1^k)$ then the view of $B_{\rm atk}$ in the simulated experiment is exactly as its view in $\mathbf{Exp}^{\text{su-atk-0}}_{\mathcal{PE}',B_{\rm atk}}(1^k)$. Similarly, if $D_{\rm atk}$ does not abort and it is in $\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE},D_{\rm atk}}(1^k)$ then the view of $B_{\rm atk}$ in the simulated experiment is exactly as its view in $\mathbf{Exp}^{\text{su-atk-1}}_{\mathcal{PE}',B_{\rm atk}}(1^k)$. Note that $D_{\rm cca}$ is legitimate, makes at most as many decryption oracle queries as $B_{\rm atk}$ does and it clearly runs in poly(k) time. ▌

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 for appropriate prime-order-group generators. Accordingly we begin with definitions for the latter.

A *prime-order-group generator* is a probabilistic poly(k)-time algorithm that on input 1^k, where $k\in\mathbb{N}$ is the security parameter, returns a tuple $(1^k,\tilde{\mathbb{G}},q,g)$, where q is a prime with $2^{k-1}<q<2^k$, $\tilde{\mathbb{G}}$ is a description of a group $\mathbb{G}$ of order q, and g is a generator of $\mathbb{G}$. There can be numerous such prime-order-group generators. We will not specify a particular one but will use it as a parameter to the computational problems we consider. The description of a group should specify the algorithms for group operations (multiplication and inverse), the algorithm for testing group membership, and also the random group element sampling algorithm. All of these talgorithms are assumed to be polynomial in k. Here and further in the paper we assume that the group elements are uniquely encoded as strings. We let $\hat{1}$ denote the identity element of $\mathbb{G}$. Let $T^{\rm exp}_{\mathcal{G}}(k)$ denote the worst time needed to perform an exponentiation operation with respect to a base element in $\mathbb{G}$ and an exponent in $\mathbb{Z}_q$, for any $\tilde{\mathbb{G}},q,g$ output of $\mathcal{G}(1^k)$. This operation is assumed to be polynomial in k.

Definition 5.1 [DDH] Let $\mathcal{G}$ be a prime-order-group generator. Let D be an adversary that on input $\tilde{\mathbb{G}},q,g$ and three elements $X,Y,T\in\mathbb{G}$ returns a bit. We consider the following experiments

Experiment $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(1^k)$	Experiment $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k)$
$(1^k,\tilde{\mathbb{G}},q,g)\xleftarrow{\$}\mathcal{G}(1^k)$	$(1^k,\tilde{\mathbb{G}},q,g)\xleftarrow{\$}\mathcal{G}(k)$
$x\xleftarrow{\$}\mathbb{Z}_q$; $X\leftarrow g^x$; $y\xleftarrow{\$}\mathbb{Z}_q$; $Y\leftarrow g^y$	$x\xleftarrow{\$}\mathbb{Z}_q$; $X\leftarrow g^x$; $y\xleftarrow{\$}\mathbb{Z}_q$; $Y\leftarrow g^y$
$T\leftarrow g^{xy}$	$T\xleftarrow{\$}\mathbb{G}$
$d\xleftarrow{\$}D(1^k,\tilde{\mathbb{G}},q,g,X,Y,T)$	$d\xleftarrow{\$}D(1^k,\tilde{\mathbb{G}},q,g,X,Y,T)$
`Return` d	`Return` d

The advantage of D in solving the Decisional Diffie-Hellman (DDH) problem for $\mathcal{G}$ is the function of the security parameter defined by

$$\mathsf{Adv}^{\rm ddh}_{\mathcal{G},D}(k)\ =\ \Pr\left[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(1^k)=1\,\right]-\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k)=1\,\right]\ .$$

We say that the DDH problem is hard for $\mathcal{G}$ if the function $\mathsf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(\cdot)$ is negligible for every poly(k)-time adversary D. ▮

We will refer to (g, X, Y, T) as to a *valid* Diffie-Hellman tuple if $\log_g X = \log_Y T$, and as to a *random* Diffie-Hellman tuple otherwise.

A common case is that $\mathbb{G}$ is a subgroup of order q of $\mathbb{Z}_p^*$ where p is a prime such that q divides $p-1$. Another example is when $\mathbb{G}$ is an appropriate elliptic curve group. Our setting is general enough to encompass both these 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]. A proof of Lemma 5.2 is in Appendix A.

Lemma 5.2 Let $\mathcal{G}$ be a prime order group generator and let $k \in \mathbb{N}$ be the security parameter. Then there is a probabilistic algorithm R such that for any $(\tilde{\mathbb{G}}, q, g)$, an output of $\mathcal{G}(1^k)$, and any a, b, c, x in $\mathbb{Z}_q$, it takes input q, g, g^a, g^b, g^c, x, returns a triple $g^{a'}, g^{b'}, g^{c'}$ and runs in $O(T^{\exp}_{\mathcal{G}}(k))$ time such that the properties represented by the following table are always 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 $\mathbb{Z}_q$ 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 $\mathbb{Z}_q$.

5.1 ElGamal

As indicated in Section 4.2, 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 ElGamal encryption scheme over a group of prime order, by exploiting Lemma 5.2. Let $\mathcal{G}$ be a prime-order-group generator. This is the common key generation algorithm of the ElGamal scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, the rest of the algorithms are as follows:

Algorithm $\mathcal{K}((1^k, \tilde{\mathbb{G}}, q, g))$	Algorithm $\mathcal{E}_{(1^k,\tilde{\mathbb{G}},q,g),X}(M)$	Algorithm $\mathcal{D}_{(1^k,\tilde{\mathbb{G}},q,g),x}(Y, W)$
$x \xleftarrow{\$} \mathbb{Z}_q$; $X \leftarrow g^x$	$r \xleftarrow{\$} \mathbb{Z}_q$; $Y \leftarrow g^r$	$T \leftarrow Y^x$
$pk \leftarrow X$; $sk \leftarrow x$	$T \leftarrow X^r$; $W \leftarrow TM$	$M \leftarrow WT^{-1}$
Return (pk, sk)	Return (Y, W)	Return M

The message space associated to a common key $(1^k, \tilde{\mathbb{G}}, q, g)$ is the group $\mathbb{G}$ itself. Note that a generator g is the output of the common key generation algorithm, which means we fix g for all keys.

We noted in Section 1.4 that the hardness of the DDH problem implies that the ElGamal scheme meets the standard notion of indistinguishability of encryptions (cf.[NaRe, CrSh, TsYu]), and the reduction is essentially tight: for any poly(k)-time adversary B there is a poly(k)-time adversary D such that $\mathsf{Adv}^{\text{su-cpa}}_{\mathcal{EG},B}(k)$ is at most $2\mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k)$. We want to look at the security of the ElGamal scheme in the multi-user setting. Directly applying Theorem 4.1 in conjunction with the above would tell us that for any poly(k)-time adversary A there is a poly(k)-time adversary D such that for any $k \in \mathbb{N}$ and a polynomial n

$$\mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{EG},A}(k) \leq 2q_{\text{e}}(k)n(k) \cdot \mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) \ , \tag{4}$$

where D's running time is that of A plus $O(\log(q_{\text{e}}(k)n(k)))$. This is enough to see that polynomial security of the DDH problem implies polynomial security of ElGamal in the multi-user setting, but we want to improve the concrete security of this relation and say that the security of the ElGamal 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 $\mathcal{G}$ be a prime-order-group generator and let $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the associated ElGamal encryption scheme, Then for any adversary A there exists an adversary D such that for any $k \in \mathbb{N}$ and any polynomial n

$$\mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{EG},A}(k) \ \leq \ 2 \cdot \mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) + \frac{1}{2^{k-1}} \ ,$$

where the running time of D is that of A plus $O(q_{\text{e}}(k)n(k) \cdot T^{\text{exp}}_{\mathcal{G}}(k))$. ▌

The last term is negligible in practice, 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 a legitimate adversary attacking privacy of the ElGamal public-key encryption scheme $\mathcal{EG}$ against chosen-plaintext attacks in the multi-user setting (cf. Definition 3.1). Suppose it makes at most $q_{\text{e}}(k)$ queries to each of its $n(k)$ oracles and has time-complexity at most t. We will design an adversary D for the Decisional Diffie-Hellman problem (cf. Definition 5.1) so that D has running time is that of A plus $O(q_{\text{e}}(k)n(k) \cdot T^{\text{exp}}_{\mathcal{G}}(k))$ and

$$\mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) \ \geq \ \frac{1}{2} \cdot \mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{EG},A}(k) - \frac{1}{2^k} \ . \tag{5}$$

This implies the statement of the theorem. So it remains to specify D. The code for D is presented in Figure 1. It has input $1^k, \tilde{\mathbb{G}}, q, g$, and also three elements $X, Y, T \in \mathbb{G}$. It will use adversary A as a subroutine. D will provide for A input keys $I, pk_1, \ldots, pk_{n(k)}$ and will simulate $n(k)$ LR oracles, $\mathcal{E}_{I,pk_i}(\text{LR}(\cdot,\cdot,b))$ for $i = 1, \ldots, n(k)$. 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. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(1^k)$. In this case, (g, X, Y, T) given to D above is a valid Diffie-Hellman tuple and thus satisfy $T = g^{xy}$ where $X = g^x$ and $Y = g^y$. Lemma 5.2 tells us, first, that $X'_1[1], \ldots, X'_{n(k)}[1]$ are all uniformly and independently distributed over $\mathbb{G}$, because here R was called with $x = 1 \neq 0$. Thus they have the proper distribution of public keys for the ElGamal cryptosystem. Applying the same lemma again, we see that the reply to the $A's$ query M_0, M_1 to LR oracle $\mathcal{E}_{I,pk_i}(\text{LR}(\cdot,\cdot,b))$ is distributed exactly like an ElGamal encryption of M_b under public key $X'_i[1]$, for all $i = 1, \ldots, n(k)$. We use this to see that

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

Adversary $D(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$
$I \leftarrow (1^k, \tilde{\mathbb{G}}, q, g)$
$b \xleftarrow{\$} \{0,1\}$
`For` $i = 1, \ldots, n(k)$ `do`
$(X'_i[1], Y'_i[1], T'_i[1]) \xleftarrow{\$} R(q, g, X, Y, T, 1)$; $pk_i \leftarrow X'_i[1]$; $ctr_i \leftarrow 0$
`For` $j = 2, \ldots, q_e$ `do` $(X'_i[j], Y'_i[j], T'_i[j]) \xleftarrow{\$} R(q, g, X'_i[1], Y'_i[1], T'_i[1], 0)$ `EndFor`
`EndFor`
Run $A\left(I, pk_1, \ldots, pk_{n(k)}\right)$ as follows:
When A makes a query M_0, M_1 to oracle $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$ $[1 \le i \le n]$:
$ctr_i \leftarrow ctr_i + 1$; $W_i \leftarrow T'_i[ctr_i] \cdot M_b$
`Return` $(Y_i[ctr_i], W_i[ctr_i])$ to A
Eventually A halts
`If it output a bit` d and $b = d$ `then return 1 else return 0 EndIf`

Figure 1: *Adversary D in proof of Theorem 5.3, where R is the algorithm of Lemma 5.2.*

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k)$. In this case, the D's inputs X, Y, T above are all uniformly distributed over $\mathbb{G}$. Lemma 5.2 again tells us, first, that $X'_1[1], \ldots, X'_{n(k)}[1]$ are all uniformly and independently distributed over $\mathbb{G}$, so that they again have the proper distribution of public keys for the ElGamal cryptosystem, because R was called with $x = 1 \neq 0$. Let NR denote the event when $\log_g X = \log_Y T$ in $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k)$. Note that $\Pr[\,\mathsf{NR}\,] \le 2^{-(k-1)}$ since X, Y, T are random elements in $\mathbb{G}$, the order of $\mathbb{G}$ is q and $2^{k-1} < q < 2^k$. Now assume $\neg\mathsf{NR}$ (this event happens with probability $(1 - \Pr[\,\mathsf{NR}\,])$, and let us apply the same lemma again. It tells us that for $i = 1, \ldots, n(k)$ the values $T'_i[ctr_i]$ are distributed uniformly at random in $\mathbb{G}$ independently of anything else. Hence the same is true of $W_i[ctr_i]$, for $i = 1, \ldots, n(k)$. This means that the replies to A's queries gives A no information about b, in an information-theoretic sense. So

$$\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k) = 1\,\right] \;\le\; \frac{1}{2}\left(1 - \Pr[\,\mathsf{NR}\,]\right) + \Pr[\,\mathsf{NR}\,] \;=\; \frac{1}{2} + \frac{\Pr[\,\mathsf{NR}\,]}{2} \;\le\; \frac{1}{2} + \frac{1}{2^k}\,. \qquad (7)$$

Subtracting Equations 6 and 7 we get

$$\begin{aligned}\mathsf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) &= \Pr\left[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(1^k) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(1^k) = 1\,\right] \\ &\ge \frac{1}{2} \cdot \mathsf{Adv}^{\text{n-mu-cpa}}_{\mathcal{EG},A}(k) - \frac{1}{2^k}\,,\end{aligned}$$

which is Equation (5).

It remains to justify the claim about the time-complexity of D. The overhead for D is essentially that of invoking the algorithm R a total of $q_e(k)n(k)$ times, and that's the added cost in D's running time. ∎

5.2 Cramer-Shoup

Now we consider another specific scheme, namely the public-key encryption scheme proposed by Cramer and Shoup [CrSh], which is secure against chosen-ciphertext attacks in the single-user setting. We are interested in the security of this scheme (against chosen-ciphertext attack) in the multi-user setting. Let us first recall the basic scheme.

The scheme uses a family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ defined by a probabilistic generator algorithm $\mathcal{GH}$ —which takes as input 1^k, where $k \in \mathbb{N}$ is a security parameter and returns a key

Algorithm $\mathcal{G}(1^k)$
$(1^k, \tilde{\mathbb{G}}, q, g) \xleftarrow{\$} \overline{\mathcal{G}}(1^k)$
$g_1 \leftarrow g$
$K \xleftarrow{\$} \mathcal{GH}(k)$
$I \leftarrow (1^k, \tilde{\mathbb{G}}, q, g_1, K)$
Return I

Algorithm $\mathcal{E}_{I,pk}(M)$
Parse I as $(\tilde{\mathbb{G}}, q, g_1, K)$
Parse pk as (g_2, c, d, h)
$r \xleftarrow{\$} \mathbb{Z}_q$
$u_1 \leftarrow g_1^r$; $u_2 \leftarrow g_2^r$
$e \leftarrow h^r M$
$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
$v \leftarrow c^r d^{r\alpha}$
Return (u_1, u_2, e, v)

Algorithm $\mathcal{K}((1^k, \tilde{\mathbb{G}}, q, g_1, K))$
$g_2 \xleftarrow{\$} \mathbb{G}/\{\hat{1}\}$
$x_1, x_2, y_1, y_2, z \xleftarrow{\$} \mathbb{Z}_q$
$c \leftarrow g_1^{x_1} g_2^{x_2}$; $d \leftarrow g_1^{y_1} g_2^{y_2}$; $h \leftarrow g_1^z$
$pk \leftarrow (g_2, c, d, h)$
$sk \leftarrow (x_1, x_2, y_1, y_2, z)$
Return (pk, sk)

Algorithm $\mathcal{D}_{I,sk}((u_1, u_2, e, v))$
Parse sk as $(x_1, x_2, y_1, y_2, z, K)$
$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
If $u_1, u_2, e, v \in \mathbb{G}$ and ${u_1}^{x_1+y_1\alpha} {u_2}^{x_2+y_2\alpha} = v$
then $M \leftarrow e({u_1}^z)^{-1}$
else $M \leftarrow \perp$
EndIf
Return M

Figure 2: Cramer-Shoup scheme

K— and a deterministic evaluation algorithm $\mathcal{EH}$ —which takes as input the key K and a string $M \in \{0,1\}^*$ and returns a string $\mathcal{EH}_K(M) \in \{0,1\}^{k-1}$. Without loss of generality we assume that $K \in \{0,1\}^k$. Let $\overline{\mathcal{G}}$ be a prime-order-group generator. The algorithms of the associated Cramer-Shoup scheme $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ are depicted in Figure 2. The message space associated to a common key $(1^k, \tilde{\mathbb{G}}, q, g)$ is the group $\mathbb{G}$ itself.

Before we start analyzing the scheme let us first recall the definition of collision resistance of hash function families, since it will be used in our analysis. *Sasha: Shoup uses a weaker notion-target CR, change to Shoup's definition (assumption)?*

Definition 5.4 [Collision resistance] Let $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ be a family of hash functions and let C be an adversary that on input a key K returns two strings. Now, we consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{cr}}_{\mathcal{H},C}(1^k)$
$K \xleftarrow{\$} \mathcal{GH}(1^k)$; $(X_0, X_1) \leftarrow C(K)$
If $(X_0 \neq X_1)$ and $\mathcal{EH}_K(X_0) = \mathcal{EH}_K(X_1)$ then return 1 else return 0

We define the *advantage* of adversary C via

$$\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) = \Pr\left[\, \mathbf{Exp}^{\mathrm{cr}}_{\mathcal{H},C}(1^k) = 1 \,\right] .$$

We say that the family of hash functions $\mathcal{H}$ is *collision-resistant* if $\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(\cdot)$ is negligible for every poly(k)-time algorithm C. ▌

Cramer and Shoup state the concrete security of their reduction, which is essentially tight. In our language: Let $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the Cramer-Shoup encryption scheme associated to a prime-order-group generator $\overline{\mathcal{G}}$, and a family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$. Then for any $k \in \mathbb{N}$ and for any poly(k)-time adversary B, which makes $q_{\mathrm{d}}(k)$ decryption oracle queries, there exist a poly(k)-time adversaries C and D such that

$$\mathsf{Adv}^{\text{su-cca}}_{\mathcal{CS},B}(k) \leq \mathsf{Adv}^{\mathrm{ddh}}_{\overline{\mathcal{G}},D}(k) + \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) + \frac{4(q_{\mathrm{d}}(k)+4)}{2^k} . \tag{8}$$

Moving to the multi-user setting, Theorem 4.1 in combination with the above tells us that for any $k \in \mathbb{N}$ and a polynomial n and for any adversary A, which makes $q_{\rm e}(k)$ LR encryption oracle queries and $q_{\rm d}(k)$ decryption oracle queries, there exist adversaries C and D such that

$$\mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{CS},A}(k) \leq q_{\rm e}(k)n(k) \cdot \mathsf{Adv}^{\rm ddh}_{\overline{\mathcal{G}},D}(k) + q_{\rm e}(k)n(k) \cdot \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) + \frac{4q_{\rm e}(k)n(k) \cdot (q_{\rm d}(k)+4)}{2^k} \ ,$$

where the running time of C, D is that of A plus $\log(q_{\rm e}(k)n(k))$. Our improvement is the following.

Theorem 5.5 Let $\overline{\mathcal{G}}$ be a prime-order-group generator, $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ be a family of hash functions, $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the associated Cramer-Shoup encryption scheme. Then for any $k \in \mathbb{N}$ and a polynomial n and for any adversary A, which makes at most $q_{\rm e}(k)$ queries to each of its LR encryption oracles and at most $q_{\rm d}(k)$ queries to each of its decryption oracles, there exists an adversary C and an adversary D such that

$$\mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{CS},A}(k) \quad \leq \quad 2q_{\rm e}(k) \cdot \mathsf{Adv}^{\rm ddh}_{\overline{\mathcal{G}},D}(k) + 2q_{\rm e}(k) \cdot \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) + \frac{8q_{\rm e}(k)n(k) \cdot (q_{\rm d}(k)+1)}{2^k} \ ,$$

where the running time of D is that of A plus $O(q_{\rm e}(k)n(k) \cdot T^{\rm exp}_{\mathcal{G}}(k))$ and the running time of C is that of A. ▌

So comparing with Equation (8) we see that we have almost the same proven security for n users as for one user when each encrypts $q_{\rm e}(k)$ 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 $q_{\rm e}(k)$ 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 $q_{\rm e}(k)$ even when there is only a single user encrypting $q_{\rm e}(k)$ 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 5.5: We will focus on proving the result in the case when $q_{\rm e} = 1$. Namely we want to show that for any $k \in \mathbb{N}$ and a polynomial n and for any adversary A, which makes only one query to each of LR encryption oracles and $q_{\rm d}(k)$ queries to each of its decryption oracles, there exists an adversary C and an adversary D such that

$$\mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{CS},A}(k) \quad \leq \quad 2\mathsf{Adv}^{\rm ddh}_{\overline{\mathcal{G}},D}(k) + 2 \cdot \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) + \frac{8n(k) \cdot (q_{\rm d}+1)}{2^k} \ . \tag{9}$$

Given this, the statement of the theorem follows via a hybrid argument reducing the case of $q_{\rm e}(k)$ 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 $q_{\rm e}(k)$. 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 $q_{\rm e} = 1$. In this case we have to show how we are avoiding the appearance of a factor equal to the number $n(k)$ of users in the bound on the advantage.

Let A be a legitimate adversary attacking the Cramer-Shoup public-key encryption scheme $\mathcal{CS}$ in the multi-user setting. We are assuming that it makes at most one query to each of its $n(k)$ LR encryption oracles, at most $q_{\rm d}$ queries to each of its $n(k)$ decryption oracles and has time-complexity at most t. We will design an adversary D for the Decisional Diffie-Hellman problem. The adversary D takes as input $(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$. In order to figure out whether (g, X, Y, T) is a valid Diffie-Hellman tuple, D will use A as a subroutine. D will provide for A inputs $I, pk_1, \ldots, pk_n(k)$ and will simulate for A the the $n(k)$ LR encryption oracles, $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$ and the $n(k)$ decryption oracles, $\mathcal{D}_{I,sk_i}(\cdot)$ for $i = 1, \ldots, n(k)$. We let R denote the algorithm of Lemma 5.2.

Adversary $D(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$

$K \xleftarrow{\$} \mathcal{GH}(1^k)$
$g_1 \leftarrow g$
$I \leftarrow (1^k, \tilde{\mathbb{G}}, q, g_1, K)$
`For` $i = 1, \ldots, n(k)$ `do`
  $(g_{2,i}, u_{1,i}, u_{2,i}) \leftarrow R(q, g_1, X, Y, T, 1)$
  $x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}, z_{1,i}, z_{2,i} \xleftarrow{\$} \mathbb{Z}_q$
  $c_i \leftarrow g_1^{x_{1,i}}(g_{2,i})^{x_{2,i}}$; $d_i \leftarrow g_1^{y_{1,i}}(g_{2,i})^{y_{2,i}}$; $h_i \leftarrow g_1^{z_{1,i}}(g_{2,i})^{z_{2,i}}$
  $pk_i \leftarrow (g_{2,i}, c_i, d_i, h_i)$
`EndFor`
$b \xleftarrow{\$} \{0,1\}$
Run $A\left(I, pk_1, \ldots, pk_{n(k)}\right)$ replying to its LR encryption oracle queries as follows:
  When A makes a query M_0, M_1 to oracle $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$:
    [$1 \le i \le n(k)$, only one query to each oracle is allowed]
    $e_i \leftarrow (u_{1,i})^{z_{1,i}}(u_{2,i})^{z_{2,i}} M_b$
    $\alpha_i \leftarrow \mathcal{EH}_K(u_{1,i}, u_{2,i}, e_i)$
    $v_i \leftarrow (u_{1,i})^{x_{1,i}+y_{1,i}\alpha_i}(u_{2,i})^{x_{2,i}+y_{2,i}\alpha_i}$
    `Return` $(u_{1,i}, u_{2,i}, e_i, v_i)$ to A
And replying to A's decryption queries as follows:
  When A makes a query C to its decryption oracle $\mathcal{D}_{I,sk_i}(\cdot)$ [$1 \le i \le n(k)$]
    parse C as (u_1', u_2', e', v')
    $\alpha \leftarrow \mathcal{EH}_K(u_1', u_2', e')$
    `If` $u_1', u_2', e', v' \in \mathbb{G}$ `and` $(u_1')^{x_{1,i}+y_{1,i}\alpha}(u_2')^{x_{2,i}+y_{2,i}\alpha} = v'$
      `then` $M \leftarrow e'((u_1')^{z_{1,i}}(u_2')^{z_{2,i}})^{-1}$
      $M \leftarrow \perp$
    `EndIf`
    `Return` M to A
Eventually A halts
`If it output a bit` d and $b = d$ `then return 1 else return 0 EndIf`

We derived this algorithm by adapting the simulation from *Sasha: first version of* [CrSh] to the multi-user case and then weaving in the Diffie-Hellman self-reducibility algorithms to improve the quality of the reduction.

Lemma 5.6 For any $k \in \mathbb{N}$ we have

$$\Pr\left[\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D}(1^k) = 1\right] \;=\; \frac{1}{2} + \frac{1}{2} \cdot \mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{CS},A}(k)\ ,$$

∎

Lemma 5.7 There exists an adversary C such that for every $k \in \mathbb{N}$

$$\Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1\right] \;\le\; \frac{1}{2} + \frac{q_{\mathrm{d}}(k)n(k)+1}{2^k} + \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k)\ ,$$

where $q_{\mathrm{d}}(k)$ is the number of decryption oracle queries made by A, and the running time of C is that of A. ∎

Proof of Theorem 5.5: This follows from Lemma 5.6, Lemma 5.7 and the observation that the running time of D is that of A plus the time for several group operations and exponentiations, $n(k)$ invocations of R algorithm and taking into account the hybrid argument on $q_{\mathrm{e}}(k)$ LR encryption oracle queries. ∎

It remains to prove the above two lemmas. The proof of Lemma 5.6 is in Section 5.2.1 and the proof of Lemma 5.7 is in Section 5.2.2.

5.2.1 Proof of Lemma 5.6

First consider $\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D}(1^k)$. We claim that in this case depending on the random bit b the adversary D picks, the simulated view of the adversary A is exactly as in the actual experiment $\mathbf{Exp}^{\text{n-mu-cca-b}}_{\mathcal{PE},A}(1^k)$. This means that we have to show that the common key and the public keys given to A have the right distribution and that the LR encryption and decryption oracle queries are answered correctly.

Obviously, the common key I has the right distribution. It follows from Lemma 5.2 that the generated $g_{2,1},\ldots,g_{2,n}$ have the right distribution. Therefore it is immediate that $c_1,d_1,\ldots,c_n,d_n$ have the right distribution. To show that all public keys generated by D have the right distribution it remains to show that $h_1,\ldots,h_{n(k)}$ have the right distribution. According to the key generation algorithm $h=g_1^z$ for a random $z\in\mathbb{Z}_q$. D computes $h_i \leftarrow g_1^{z_{1,i}}(g_{2,i})^{z_{2,i}}$, where $z_{1,i},z_{2,i}$ are random elements in $\mathbb{Z}_q$ for $i\in 1,\ldots,n(k)$. Let us denote $\omega_i=\log_{g_1}(g_{2,i})$. Then for $i\in 1,\ldots,n(k)$ we can rewrite h_i as $g_1^{z_{1,i}+\omega_i z_{2,i}}=g_1^{\bar{z}_i}$, where $\bar{z}_i$ denotes $z_{1,i}+\omega_i z_{2,i}$ and corresponds to z in the real algorithm. We can see that $z,\bar{z}_i$ have the same distribution of random elements in $\mathbb{Z}_q$.

Now we show that all the challenge ciphertexts $(u_{1,i},u_{2,i},e_i,v_i)$ have the right distribution, for $i\in 1,\ldots,n(k)$. The encryption algorithm computes $u_1\leftarrow g_1^r, u_2\leftarrow (g_2^r, e\leftarrow h^r M(=g_1^{rz}M)$ for random $r\in\mathbb{Z}_q$. Since (g,X,Y,T) given to D form a valid Diffie-Hellman tupple and due to Lemma 5.2 one can see that the computed $u_{1,i},u_{2,i}$ are of the right form $g_1^{r_i},(g_{2,i})^{r_i}$, for some random r_i in $\mathbb{Z}_q$. D computes each e_i differently: $e_i\leftarrow (u_{1,i})^{z_{1,i}}(u_{2,i})^{z_{2,i}}M_b$, for $b\in\{0,1\}$. We can rewrite e_i as $g^{r_i z_{1,i}+r_i\omega_i z_{2,i}}M_b=h^{r_i(z_{1,i}+\omega_i z_{2,i})}M=h^{r_i\bar{z}_i}M$. Thus rz in the real encryption algorithm corresponds to $r_i\bar{z}_i$. This shows that $e_1,\ldots,e_{n(k)}$ computed by D have the right distribution, since $r,z,r_i,\bar{z}_i$ are all random elements in $\mathbb{Z}_q$. The encryption algorithm computes $v\leftarrow c^r d^{r\alpha}$. In the simulation $v_i\leftarrow (u_{1,i})^{x_{1,i}+y_{1,i}\alpha}(u_{2,i})^{x_{2,i}+y_{2,i}\alpha}(=g_1^{r_i x_{1,i}+r_i y_{1,i}\alpha}g^{r_i x_{2,i}+r_i y_{2,i}\alpha}=(g^{x_{1,i}}g^{x_{2,i}})^{r_i}(g_1^{y_{1,i}}g_2^{y_{2,i}})^{r_i\alpha}=c_i^{r_i}d_i^{r_i\alpha})$. This is the right form, since r_i corresponds to r in a real experiment, they are all random elements in $\mathbb{Z}_q$ and α is properly computed.

To complete the proof we show that the decryption oracle queries (u_1',u_2',e',v') are answered as they should. This is true because the condition of a valid ciphertext is computed as in the actual algorithm, and according to the decryption algorithm the plaintext is computed as $M\leftarrow eu_1{}^{-z}$. For a query made to $\mathcal{D}_{sk_i}(\cdot)$, D computes $M\leftarrow e'(u_1')^{-z_{1,i}}(u_2')^{-z_{2,i}}$, which can be viewed as $M\leftarrow eu_1^{-\bar{z}_i}$, where $\bar{z}_i$ denotes $z_{1,i}+\omega_i z_{2,i}$. This is correct decryption. So we have

$$\begin{aligned}\Pr\left[\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D}(1^k)=1\right] &= \frac{1}{2}\Pr\left[\mathbf{Exp}^{\text{n-mu-cca-1}}_{\mathcal{PE},A}(1^k)=1\right]+\frac{1}{2}\left(1-\Pr\left[\mathbf{Exp}^{\text{n-mu-cca-0}}_{\mathcal{PE},A}(1^k)=1\right]\right)\\ &= \frac{1}{2}+\frac{1}{2}\cdot\mathsf{Adv}^{\text{n-mu-cca}}_{\mathcal{CS},A}(k)\ .\end{aligned}$$

5.2.2 Proof of Lemma 5.7

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k)$. In this case, the inputs X, Y, T to D and, therefore, $u_{1,1}, u_{2,1}, \ldots$, $u_{1,n(k)}, u_{2,n(k)}$ are uniformly distributed over $\mathbb{G}$. We say that a ciphertext (u', u', e', v') corresponding to the public key $(g_{2,i}, c_i, d_i, h_i)$ is invalid if $u', u', e', v' \notin \mathbb{G}$ or $\log_{g_1} u'_1 \neq \log_{g_{2,i}} u'_2$. Note, that the challenge ciphertexts A gets are invalid. Let us define events associated to D in $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k)$.

- NR is true if $\log_{g_1} X = \log_Y T$.
- Inv is true if D does not return $\perp$ to A when A submits an invalid ciphertext to any of its decryption oracles.

Lemma 5.8 $\Pr[\,\mathsf{NR}\,] \leq 1/2^{k-1}$. ∎

Lemma 5.9

$$\begin{aligned}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right] &= \frac{1}{2}\,,\\ \Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 1 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right] &= \frac{1}{2}\,.\ ∎\end{aligned}$$

Lemma 5.10 For any $k \in \mathbb{N}$ and any adversary A which makes at most $q_{\mathrm{d}}(k)$ decryption oracle queries there exists an adversary C such that

$$\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,] \leq \frac{4q_d(k)n(k)}{2^k} + \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k)\,,$$

where the running time of C is that of A. ∎

Proof of Lemma 5.7: By conditioning we get

$$\begin{aligned}&\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1\,\right]\\ &= \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 0\,\right] + \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 1\,\right]\\ &\leq \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right]\\ &+ \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 1 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right] + \Pr[\,\mathsf{NR}\,] + \Pr[\,\mathsf{Inv}\,]\\ &\leq \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right]\\ &+ \frac{1}{2}\Pr\left[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k) = 1 \mid b = 1 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\right] + 2\Pr[\,\mathsf{NR}\,] + \Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,]\ .\end{aligned}$$

Applying Lemmas 5.9, 5.8 and 5.10 to the above statement we get the claim of Lemma 5.7. ∎

The proof of Lemmas 5.8, 5.9 and 5.10 are in Sections 5.2.3, 5.2.4, 5.2.5, respectively.

5.2.3 Proof of Lemma 5.8

The claim is true since X, Y, T are random elements in $\mathbb{G}$, the order of $\mathbb{G}$ is q and $2^{k-1} < q < 2^k$.

5.2.4 Proof of Lemma 5.9

We first define the sample space S which is going to be used in our analysis. It consists of the values chosen at random in $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k)$. We will denote an element of S as

$$\begin{aligned}\vec{s} \;=\; & (K, b, g_1, g_{2,1}, u_{1,1}, u_{2,1}, \ldots, g_{2,n(k)}, u_{1,n(k)}, u_{2,n(k)}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1}, \ldots, \\ & x_{1,n(k)}, x_{2,n(k)}, y_{1,n(k)}, y_{2,n(k)}, z_{1,n(k)}, z_{2,n(k)}) \;,\end{aligned}$$

and define the sample space as

$$S \;=\; \left\{ \vec{s} : \vec{s} \in \{0,1\}^k \times \{0,1\} \times \mathbb{G}^{3n(k)+1} \times \mathbb{Z}_q^{6n(k)} \right\} \;.$$

Sasha: explain notation? Note that we treat $u_{1,1}, u_{2,1}, \ldots, u_{1,n(k)}, u_{2,n(k)}$ as being chosen at random because of Lemma 5.2 and the condition that $\neg\mathsf{NR}$ is true. We let View be the function which has domain S and associates to any $\vec{s} \in S$ the view of the adversary A in the experiment $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k)$ when the random choices in that experiment are those given in $\vec{s}$. For simplicity we assume A is deterministic (otherwise, the argument can simply be made for each choice of its coins.) The view then includes the inputs the adversary receives and the answers to all its oracle queries. The adversary's output is a deterministic function of its view.

Claim 5.11 Fix a specific view $\hat{V}$ of the adversary A simulated by D given input $(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$ in $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(1^k)$ such that the events $\neg\mathsf{NR} \wedge \neg\mathsf{Inv}$ hold. Then

$$\Pr\left[\mathsf{View} = \hat{V} \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv} \right] = \Pr\left[\mathsf{View} = \hat{V} \mid b = 1 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv} \right] \;.$$

This claim states in the absence of events $\mathsf{NR}, \mathsf{Inv}$, any view of the adversary A is equally likely given the bit b. We conclude the proof of Lemma 5.9 given this claim.

Proof of Lemma 5.9: Claim 5.11 means that, if $\neg\mathsf{NR} \wedge \neg\mathsf{Inv}$ is true, then A's view is independent of the hidden bit b. Therefore A can output its guess of b correctly only with the probability $1/2$. Thus the proof of Lemma 5.9 follows since the distinguisher D outputs 1 only when A guesses the bit b correctly. ∎

It remains to prove the above claim.

Proof of Claim 5.11: We do not consider the answers of the decryption oracles as a part of the view of the adversary because $\neg\mathsf{Inv}$ means that no invalid ciphertexts are accepted and we show below that the answers to the valid ciphertext queries do not give the adversary any additional information about the hidden bit b. Fix a view

$$\hat{V} \;=\; \left(\hat{K}, \hat{g}_1, \hat{g}_{2,1}, \hat{c}_1, \hat{d}_1, \hat{h}_1, \hat{u}_{1,1}, \hat{u}_{2,1}, \hat{e}_1, \hat{v}_1, \ldots, \hat{g}_{2,n(k)}, \hat{c}_{n(k)}, \hat{d}_{n(k)}, \hat{h}_{n(k)}, \hat{u}_{1,n(k)}, \hat{u}_{2,n(k)}, \hat{e}_{n(k)}, \hat{v}_{n(k)} \right) \;.$$

Next for $d \in \{0,1\}$ define $E_d \subseteq S$ as the set of all $\vec{s} \in S$ such that $\vec{s}$ contains $b = d$, $\mathsf{View}(\vec{s}) = \hat{V}$ given $\neg\mathsf{NR} \wedge \neg\mathsf{Inv}$ is true when the random choices in the experiment are $\vec{s}$. Then

$$\Pr\left[V = \hat{V} \wedge b = 0 \mid \neg\mathsf{NR} \wedge \neg\mathsf{Inv} \right] = \frac{|E_0|}{|S|} \;. \tag{10}$$

We next compute $|E_0|$. This is the number of solutions to the following system of equations (the equations are derived using the algorithm for D):

$$
\begin{aligned}
K &= \hat{K} && (11)\\
b &= 0 && (12)\\
g_1 &= \hat{g}_1 && (13)\\
g_{2,i} &= \hat{g}_{2,i} && (14)\\
x_{1,i} + \hat{\omega}_i x_{2,i} &= \log_{\hat{g}_1} \hat{c}_i && (15)\\
y_{1,i} + \hat{\omega}_i y_{2,i} &= \log_{\hat{g}_1} \hat{d}_i && (16)\\
z_{1,i} + \hat{\omega}_i z_{2,i} &= \log_{\hat{g}_1} \hat{h}_i && (17)\\
u_{1,i} &= \hat{u}_{1,i} && (18)\\
u_{2,i} &= \hat{u}_{2,i} && (19)\\
\hat{r}_{1,i} z_{1,i} + \hat{r}_{2,i}\hat{\omega}_i z_{2,i} &= \log_{\hat{g}_1}(\hat{e}_i M_0^{-1}) && (20)\\
\hat{r}_{1,i} x_{1,0} + \hat{r}_{1,i}\hat{\alpha}_i y_{1,i} + \hat{r}_{2_i}\hat{\omega}_i x_{2,i} + \hat{r}_{2_i}\hat{\omega}_i\hat{\alpha}_i y_{2,i} &= \log_{\hat{g}_1} \hat{v}_i\ , && (21)
\end{aligned}
$$

where each equation containing subscript i stands for $n(k)$ equations corresponding to $i = 1, \ldots, n(k)$ (we will refer to j's equation out of these $n(k)$ ones using the subscript j, (e.g. $(14)_2$ stands for $g_{2,2} = \hat{g}_{2,2}$), and $\hat{\omega}_i = \log_{\hat{g}_1} \hat{g}_{2,i}$, $\hat{r}_{1,i} = \log_{\hat{g}_{1,i}} \hat{u}_{1,i}$, $\hat{r}_{2,i} = \log_{\hat{g}_{2,i}} \hat{u}_{2,i}, \hat{\alpha}_i = \mathcal{EH}_{\hat{K}}(\hat{u}_{1,i}, \hat{u}_{2,i}, \hat{e}_i)$. Above variables with a hat, and M_0, denote the known constants or the values completely determined by the other constants, whereas the variables without a hat denote unknowns. Therefore, there are $3+8n(k)$ equations and $3+9n(k)$ unknowns. As we noted above we should have added to this system the equations corresponding to valid ciphertexts submitted to the decryption oracles. Assume for example that a valid ciphertext (u'_1, u'_2, e', v') is submitted to $\mathcal{D}_{sk_j}(\cdot)$ for any $1 \leq j \leq n(k)$. Let $r' = \log_{g_1} u'_1 = \log_{g_{2,j}} u'_2$. Let $\alpha' = \mathcal{EH}_K(u'_1, u'_2, e')$. Let M' be the answer of the decryption oracle. Consider the equations corresponding to decrypting this ciphertext:

$$
\begin{aligned}
\hat{r}' z_{1,j} + \hat{\omega}_i \hat{r}' z_{2,j} &= \log_{\hat{g}_1}(\hat{e}'(\hat{M}')^{-1}) && (22)\\
\hat{r}' x_{1,j} + \hat{\omega}_i \hat{r}' x_{2,j} + \hat{r}'\hat{\alpha}' y_{1,j} + \hat{\omega}_i \hat{r}'\hat{\alpha}' y_{2,j} &= \log_{\hat{g}_1} \hat{v}' && (23)
\end{aligned}
$$

Note that Equation (22) is Equation $(17)_j$ multiplied by $\hat{r}'$ and Equation (23) is Equation $(15)_j$ plus $\hat{r}'\hat{\alpha}'$ times Equation $(16)_j$. Since the equations corresponding to valid decryption oracle queries are linearly dependent with the equations corresponding to the view we for simplicity do not consider the former later in our analysis.

For $l, m \in \mathbb{N}$ let I_l denote the identity matrix of size l, $A_{l\times m}$ denotes a matrix A of size $l \times m$. For $i \in \{1, \ldots n(k)\}$ let XY_i stand for $(x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, \ldots x_{1,n(k)}, x_{2,n(k)}, y_{1,n(k)}, y_{2,n(k)})^T$ and Z_i stand for $(z_{1,1}, z_{2,1}, \ldots, z_{1,n(k)}, z_{2,n(k)})^T$ and let each other term X containing the subscript i stand for $(X_1, \ldots, X_{n(k)})^T$. We now rewrite Equations (12)-(21) in a matrix form $D_{(3+8n(k))\times(3+9n(k))} \times X_{(3+9n(k))\times 1} = E_{1\times(3+8n(k))}$ in Figure 3. Here the matrix I is from Equations (11), (13), (14), (18), (19), the matrix A is from Equations (15), (16), (21) and the matrix B is from Equations (17), (20). We prove that matrix $D_{(3+8n(k))\times(3+9n(k))}$ has full rank and therefore the number of solutions of the corresponding system of equations is $q^{4n(k)-3n(k)} = q^{n(k)}$. This is because of the only non-square matrix $A_{3n(k)\times 4n(k)}$ which corresponds to the unknowns in $\mathbb{Z}_q$. In order to prove that the matrix D has the full rank we prove that matrices A, B have full rank (obviosly, I has full rank).

$$\begin{pmatrix} I_{3+3n(k)} & & 0 \\ & A_{3n(k)\times 4n(k)} & \\ 0 & & B_{2n(k)\times 2n(k)} \end{pmatrix} \times \begin{pmatrix} K \\ b \\ g_1 \\ q_{2,i} \\ u_{1,i} \\ u_{2,i} \\ XY_i \\ Z_i \end{pmatrix} = \begin{pmatrix} \hat{K} \\ 0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{u}_{1,i} \\ \hat{u}_{2,i} \\ \log_{\hat{g}_1} \hat{c}_i \\ \log_{\hat{g}_1} \hat{d}_i \\ \log_{\hat{g}_1} \hat{v}_i \\ \log_{\hat{g}_1} \hat{h}_i \\ \log_{\hat{g}_1} (\hat{e}_i (M_0)^{-1}) \end{pmatrix}$$

Figure 3: The system of Equations (12)-(21) in the matrix form.

First consider $A_{3n(k)\times 4n(k)}$:

$$\begin{pmatrix} 1 & \hat{w}_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \hat{w}_1 & \dots & 0 & 0 & 0 & 0 \\ \hat{r}_{1,1} & \hat{w}_1\hat{r}_{2,1} & \hat{\alpha}_1\hat{r}_{1,1} & \hat{\alpha}_1\hat{w}_1\hat{r}_{1,1} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & \hat{w}_{n(k)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & \hat{w}_{n(k)} \\ 0 & 0 & 0 & 0 & \dots & \hat{r}_{1,n(k)} & \hat{w}_{n(k)}\hat{r}_{2,n(k)} & \hat{\alpha}_{n(k)}\hat{r}_{1,n(k)} & \hat{\alpha}_{n(k)}\hat{w}_{n(k)}\hat{r}_{2,n(k)} \end{pmatrix}.$$

Let $\xrightarrow{\texttt{cnd}}$ denotes the Gauss elimination algorithm where `cnd` is a condition needed to apply it. Let `Cond` denotes the condition that $\hat{r}_{1,i} \neq \hat{r}_{2,i}, \hat{\omega}_i \neq 0$ for every $1 \leq i \leq n(k)$. Then

$$A_{3n(k)\times 4n(k)} \xrightarrow{\texttt{Cond}} \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & (\hat{r}_{2,1} - \hat{r}_{1,1})\hat{\omega}_1\hat{\alpha}_1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & (\hat{r}_{2,n(k)} - \hat{r}_{1,n(k)})\hat{\omega}_{n(k)}\hat{\alpha}_{n(k)} \end{pmatrix}.$$

It is easy to see that if `Cond` holds then A has full rank since it contains a non-singular matrix.

Now consider

$$B_{2n(k)\times 2n(k)} = \begin{pmatrix} 1 & \hat{w}_1 & \dots & 0 & 0 \\ \hat{r}_{1,1} & \hat{w}_1\hat{r}_{2,1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \hat{w}_{n(k)} \\ 0 & 0 & \dots & \hat{r}_{1,n(k)} & \hat{w}_{n(k)}\hat{r}_{2,n(k)} \end{pmatrix}.$$

We compute its determinant:

$$\det\left(B_{2n(k)\times 2n(k)}\right) = \hat{w}_1 \cdot \ldots \cdot \hat{w}_{n(k)} (\hat{r}_{2,1} - \hat{r}_{1,1}) \cdot \ldots \cdot (\hat{r}_{2,n(k)} - \hat{r}_{1,n(k)}) .$$

If Cond holds then $\det(B) \neq 0$ and B has the full rank.

Now note that $\neg NR$ implies that $\hat{r}_{1,i} \neq \hat{r}_{2,i}, \hat{\omega}_i \neq 0$, and according to the key generation algorithm $\hat{g}_{2,i} \neq \hat{1}$, hence $\hat{\omega}_i \neq 0$, for every $1 \leq i \leq n(k)$. Therefore Cond holds and hence matrix $D_{(3+8n(k))\times(3+9n(k))}$ has full rank and the number of solutions to the system of equations from Figure 3 and, therefore $|E_0|$, is $q^{n(k)}$.

Note that $|E_1|$ is the number of solutions of the system of Equations (11), $b = 1$, (13)-(19), (21) and

$$\hat{r}_{1,i} z_{1,i} + \hat{r}_{2,i} \hat{\omega}_i z_{2,i} \quad = \quad \log_{\hat{g}_1}(\hat{e}_i M_1^{-1}) \; . \tag{24}$$

We now claim that by symmetry of View and of the systems of equations corresponding to E_0 and E_1 with respect to a randomly chosen bit b we get $|E_1| = |E_0|$ and therefore

$$\Pr\left[V = \hat{V} \wedge b = 0 \mid \neg \mathsf{NR} \wedge \neg \mathsf{Inv} \right] = \Pr\left[V = \hat{V} \wedge b = 1 \mid \neg \mathsf{NR} \wedge \neg \mathsf{Inv} \right] \; . \tag{25}$$

Equation (25) clearly implies Claim 5.11

Sasha: Explain why the proof does not hold for $q_{\mathrm{e}} > 1$?

∎

5.2.5 Proof of Lemma 5.10

Let us start by defining several events.

Let Inv' denote the event when A submits an invalid ciphertext to any of its decryption oracles $\mathcal{D}_{I,sk_i}(\cdot)$ for any $1 \leq i \leq n(k)$ *before* it queried the corresponding encryption oracle $\mathcal{E}_{I,pk}(\mathrm{LR}(\cdot,\cdot,b))$, and D does not return $\perp$ to A.

For $1 \leq i \leq n(k), 1 \leq j \leq q_{\mathrm{d}}(k)$ let $\mathsf{Inv}'_{\mathsf{i,j}}$ denote the event when A submits an invalid ciphertext as its j-th to decryption oracle $\mathcal{D}_{I,sk_i}(\cdot)$, this happens before it queried the corresponding encryption oracle $\mathcal{E}_{I,pk}(\mathrm{LR}(\cdot,\cdot,b))$, and D for the first time accepts an invalid ciphertext.

Let Inv'' denote the event when A submits an invalid ciphertext to any of its decryption oracles $\mathcal{D}_{I,sk_i}(\cdot)$ for any $1 \leq i \leq n(k)$ *after* it queried the corresponding encryption oracle $\mathcal{E}_{I,pk}(\mathrm{LR}(\cdot,\cdot,b))$, and D does not return $\perp$ to A.

For $1 \leq i \leq n(k), 1 \leq j \leq q_{\mathrm{d}}(k)$ let $\mathsf{Inv}''_{\mathsf{i,j}}$ denote the event when A submits an invalid ciphertext as its j-th query to decryption oracle $\mathcal{D}_{I,sk_i}(\cdot)$ after it queried the corresponding encryption oracle $\mathcal{E}_{I,pk}(\mathrm{LR}(\cdot,\cdot,b))$, and D for the first time accepts an invalid ciphertext.

Let (u'_1, u'_2, e', v') denote the invalid ciphertext in Inv' or Inv'' and let $(u_{1,i}, u_{2,i}, e_i, v_i)$ denote the corresponding challenge ciphertext in Inv''. Let $\alpha' = \mathcal{EH}_K(u'_1, u'_2, e'), \alpha_i = \mathcal{EH}_K(u_{1,i}, u_{2,i}, e_i)$. Since A is legitimate, $(u'_1, u'_2, e', v') \neq (u_{1,i}, u_{2,i}, e_i, v_i)$. Consider the following three special subcases of Inv'':

- Case 1. $(u'_1, u'_2, e') = (u_{1,i}, u_{2,i}, e_i)$.
- Case 2. $(u'_1, u'_2, e') \neq (u_{1,i}, u_{2,i}, e_i)$ and $\alpha' = \alpha_i$.
- Case 3. $(u'_1, u'_2, e') \neq (u_{1,i}, u_{2,i}, e_i)$ and $\alpha' \neq \alpha_i$.

We have:

$$\begin{aligned}
\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,] &\leq \Pr\left[\,\mathsf{Inv}' \mid \neg\mathsf{NR}\,\right] + \Pr\left[\,\mathsf{Inv}'' \mid \neg\mathsf{NR}\,\right] \\
&\leq \Pr\left[\,\mathsf{Inv}' \mid \neg\mathsf{NR}\,\right] \\
&+ \Pr\left[\,\mathsf{Inv}'' \mid \mathtt{Case\ 1} \wedge \neg\mathsf{NR}\,\right] + \Pr[\,\mathtt{Case\ 2}\,] + \Pr\left[\,\mathsf{Inv}'' \mid \mathtt{Case\ 3} \wedge \neg\mathsf{NR}\,\right] \\
&\leq q_{\mathrm{d}}(k)n(k) \cdot \max_{i,j}(\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR}\,\right]) \\
&+ \Pr\left[\,\mathsf{Inv}'' \mid \mathtt{Case\ 1} \wedge \neg\mathsf{NR}\,\right] + \Pr[\,\mathtt{Case\ 2}\,] \\
&+ q_{\mathrm{d}}(k)n(k) \cdot \max_{i,j}\left(\Pr\left[\,\mathsf{Inv}''_{\mathsf{i,j}} \mid \mathtt{Case\ 3} \wedge \neg\mathsf{NR}\,\right]\right)\ , \qquad (26)
\end{aligned}$$

where $1 \leq i \leq n(k), 1 \leq j \leq q_{\mathrm{d}}(k)$. The above equations are obtained simply by conditioning *Sasha: probably more details?.*

Claim 5.12 For any $1 \leq i \leq n(k), 1 \leq j \leq q_{\mathrm{d}}(k)$

$$\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR}\,\right] \leq \frac{1}{2^{k-1}}\ .$$

Claim 5.13

$$\Pr\left[\,\mathsf{Inv}'' \mid \mathtt{Case\ 1} \wedge \neg\mathsf{NR}\,\right] = 0\ .$$

Claim 5.14 There exists an adversary C such that

$$\Pr[\,\mathtt{Case\ 2}\,] \leq \mathsf{Adv}^{cr}_{\mathcal{H},C}(k)\ ,$$

where the running time of C will be that of A.

Claim 5.15 For any $1 \leq i \leq n(k), 1 \leq j \leq q_{\mathrm{d}}(k)$

$$\Pr\left[\,\mathsf{Inv}''_{\mathsf{i,j}} \mid \mathtt{Case\ 3} \wedge \neg\mathsf{NR}\,\right] \leq \frac{1}{2^{k-1}}\ .$$

Equation (26), Claim 5.12, Claim 5.13, Claim 5.14, Claim 5.15 imply the statement of Lemma 5.10. It remains to prove the above claims.

Proof of Claim 5.12: Consider $\mathsf{Inv}'_{\mathsf{i,j}}$. By conditioning

$$\begin{aligned}
\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR}\,\right] &= \frac{1}{2} \cdot \Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR} \wedge \mathsf{b} = 0\,\right] + \frac{1}{2} \cdot \Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR} \wedge \mathsf{b} = 1\,\right] \\
&= \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \neg\mathsf{NR} \wedge \mathsf{b} = 0\,\right]}{2 \cdot \Pr[\,\neg\mathsf{NR} \wedge b = 0\,]} + \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \neg\mathsf{NR} \wedge \mathsf{b} = 1\,\right]}{2 \cdot \Pr[\,\neg\mathsf{NR} \wedge b = 1\,]} \\
&= \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \mathsf{b} = 0 \mid \neg\mathsf{NR}\,\right]\Pr[\,\neg\mathsf{NR}\,]}{2 \cdot \Pr[\,b = 0 \mid \neg\mathsf{NR}\,]\Pr[\,\neg\mathsf{NR}\,]} + \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \mathsf{b} = 1 \mid \neg\mathsf{NR}\,\right]\Pr[\,\neg\mathsf{NR}\,]}{2 \cdot \Pr[\,b = 1 \mid \neg\mathsf{NR}\,]\Pr[\,\neg\mathsf{NR}\,]} \\
&= \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \mathsf{b} = 0 \mid \neg\mathsf{NR}\,\right]}{2 \cdot \Pr[\,b = 0 \mid \neg\mathsf{NR}\,]} + \frac{\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \wedge \mathsf{b} = 1 \mid \neg\mathsf{NR}\,\right]}{2 \cdot \Pr[\,b = 1 \mid \neg\mathsf{NR}\,]}\ .
\end{aligned}$$

Fix an A's view $\hat{V}_{i,j}$ when it made the invalid ciphertext query in question.

$$\begin{aligned}
\hat{V}_{i,j} = \big(&\hat{K}, \hat{g}_1, \hat{g}_{2,1}, \hat{c}_1, \hat{d}_1, \hat{h}_1, \ldots, \hat{g}_{2,n(k)}, \hat{c}_{n(k)}, \hat{d}_{n(k)}, \hat{h}_{n(k)}, \hat{u}_{1,1}, \hat{u}_{2,1}, \hat{e}_1, \hat{v}_1, \hat{u}_{1,i-1}, \hat{u}_{2,i-1}, \hat{e}_{i-1}, \hat{v}_{i-1}, \ldots, \\
&\hat{u}_{1,i+1}, \hat{u}_{2,i+1}, \hat{e}_{i+1}, \hat{v}_{i+1}, \ldots, \hat{u}_{1,n(k)}, \hat{u}_{2,n(k)}, \hat{e}_{n(k)}, \hat{v}_{n(k)}\big)\ .
\end{aligned}$$

The view does not include the challenge ciphertext returned by $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$ since in $\mathsf{Inv}'_{\mathsf{i,j}}$ this oracle has not been queried and the answers to the valid ciphertexts queried to decryption oracles since it can be shown similarly to the proof of Claim 5.11 that this does not give A any additional information. Let $E^{inv'}_{i,j}$ be the set of $\vec{s}$ which give rise to $\hat{V}_{i,j}$ and, given $\neg\mathsf{NR}$, $\mathsf{Inv}'_{\mathsf{i,j}}$ is true when are the random choices are determined by $\vec{s}$. Then

$$\begin{aligned}\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR}\,\right] &= \frac{|E^{inv'}_{i,j}|\cdot|S|}{2\cdot|E_0|\cdot|S|} + \frac{|E^{inv'}_{i,j}|\cdot|S|}{2\cdot|E_1|\cdot|S|} \\ &= \frac{|E^{inv'}_{i,j}|}{2\cdot|E_0|} + \frac{|E^{inv'}_{i,j}|}{2\cdot|E_1|} = \frac{|E^{inv'}_{i,j}|}{|E_0|}\ .\end{aligned}$$

The last equation is due to the fact that $|E_0| = |E_1|$. To calculate the latter fraction it is enough to divide the number of solutions to the system of Equations $(15)_i$, $(16)_i$ and

$$(u'_1)^{x_{1,i}+y_{1,i}\alpha'}(u'_2)^{x_{2,i}+y_{2,i}\alpha'} = v'\ , \tag{27}$$

the equation corresponding to the condition of an acceptance of the ciphertext (u'_1, u'_2, e', v') submitted to the $\mathcal{D}_{I,sk_i}(\cdot)$, by the number of solutions to the system of Equations $(15)_i$, $(15)_i$. It is enough to consider Equations $(15)_i$, $(16)_i$ since only the values c_i, d_i in $\hat{V}$ depend on the unknowns $x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}$ used in Equation (27).

Let $r'_1 = \log_{g_1} u'_1, r'_2 = \log_{g_{2,i}} u'_2$. We first consider the matrix of coefficients from Equations $(15)_i$, $(16)_i$ and (27):

$$\begin{pmatrix} 1 & \hat{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega}_i \\ r'_1 & r'_2\hat{\omega}_i & r'_1\alpha' & r'_2\hat{\omega}_i\alpha' \end{pmatrix}\ .$$

Similarly to the proof of Claim 5.11 it is easy to show that the above matrix has full rank if $\hat{\omega}_i(r'_2 - r'_1) \neq 0$. This condition holds since $\hat{\omega}_i \neq 0$ by the key generation algorithm and $r'_2 \neq r'_1$ by the assumption that the ciphertext is invalid. There are 4 unknowns and 3 equations, hence the number of the solutions is q.

We now consider the matrix of coefficients from Equations $(15)_i$ and $(16)_i$ only:

$$\begin{pmatrix} 1 & \hat{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega}_i \end{pmatrix}\ .$$

Obviously, it has full rank. There are 4 unknowns and 2 equations, hence the number of the solutions is q^2.

Therefore,

$$\Pr\left[\,\mathsf{Inv}'_{\mathsf{i,j}} \mid \neg\mathsf{NR}\,\right] = \frac{q}{q^2} = \frac{1}{q} \leq \frac{1}{2^{k-1}}\ .$$

∎

Proof of Claim 5.13: In Case 1 $v' \neq v$ and D will return $\perp$ to A. ∎

Proof of Claim 5.14: If event `Case 2` happens, we can construct the adversary C which attacks the collision-resistance *Sasha: shoup's?* of $\mathcal{H}$ as the experiment from Definition 5.4 describes. C will simply run the adversary A providing it with a challenge key K and simulating all other

parameters by picking them at random. The advantage function of C will be at least the probability of A of finding such triples as described in Case 2. The running time of C will be that of A. ∎

Proof of Claim 5.15: The proof is similar to the proof of Claim 5.12. Let $\hat{V}_{i,j}$ be the view of A when it made the invalid ciphertext query in question.

$$\hat{V}_{i,j} \quad = \quad (\hat{K}, \hat{g}_1, \hat{g}_{2,1}, \hat{c}_1, \hat{d}_1, \hat{h}_1, \ldots, \hat{g}_{2,n(k)}, \hat{c}_{n(k)}, \hat{d}_{n(k)}, \hat{h}_{n(k)}, \hat{u}_{1,1}, \hat{u}_{2,1}, \hat{e}_1, \hat{v}_1, \ldots, \hat{u}_{1,n(k)}, \hat{u}_{2,n(k)}, \hat{e}_{n(k)}, \hat{v}_{n(k)}) \ .$$

The view now includes the challenge ciphertext returned by $\mathcal{E}_{I,pk_i}(\mathrm{LR}(\cdot,\cdot,b))$ and as before it does not include the answers to the valid ciphertexts queried to decryption oracles. Let $E_{i,j}^{inv''}$ be the set of $\vec{s}$ which give rise to $\hat{V}$ and given $\mathtt{Case\ 3} \wedge \neg\mathsf{NR}$, $\mathsf{Inv}''_{\mathsf{i,j}}$ is true when are the random choices are determined by $\vec{s}$. Let E be the set of $\vec{s}$ which given $\mathtt{Case\ 3} \wedge \neg\mathsf{NR}$ give rise to $\hat{V}$. Then, similarly to the proof of Claim 5.12,

$$\Pr\left[\ \mathsf{Inv}'_{\mathsf{i,j}} \mid \mathtt{Case\ 3} \wedge \neg\mathsf{NR}\ \right] = \frac{|E_{i,j}^{inv''}|}{|E_0|} \ .$$

To calculate this fraction it is enough to divide the number of solutions to the system of Equations $(15)_i$, $(16)_i$, (27) and

$$\hat{r}_{1,i} z_{1,i} + \hat{r}_{2,i} \hat{\omega}_i z_{2,i} \quad = \quad \log_{\hat{g}_1}(\hat{e}_i M_b^{-1}) \tag{28}$$

by the number of solutions to the system of equations $(15)_i$, $(16)_i$ and (28).

Let $r'_1 = \log_{g_1} u'_1, r'_2 = \log_{g_{2,i}} u'_2$. We first calculate the determinant of the matrix of coefficients from Equations $(15)_i$, $(16)_i$, (28) and (27):

$$\begin{vmatrix} 1 & \hat{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega}_i \\ \hat{r}_1 & \hat{r}_2\hat{\omega}_i & \hat{r}_1\hat{\alpha} & \hat{r}_2\hat{\omega}_i\hat{\alpha}_i \\ r'_1 & r'_2\hat{\omega}_i & r'_1\alpha' & r'_2\hat{\omega}_i\alpha' \end{vmatrix} = \hat{\omega}_i^2 (r'_2 - r'_1)(\hat{r}_2 - \hat{r}_1)(\alpha' - \alpha_i) \neq 0 \ ,$$

since $\hat{\omega}_i \neq 0$ by the key generation algorithm, $r'_2 \neq r'_1$ by the assumption that the ciphertext is invalid, $\hat{r}_2 \neq \hat{r}_1$ since NR is true and $\alpha' \neq \alpha$ due to the Case 3 condition. Thus the system has a single solution.

We now consider the matrix of coefficients from Equations $(15)_i$, $(16)_i$ and (28) only:

$$\begin{pmatrix} 1 & \hat{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega}_i \\ \hat{r}_1 & \hat{r}_2\hat{\omega}_i & \hat{r}_1\hat{\alpha}_i & \hat{r}_2\hat{\omega}_i\hat{\alpha}_i \end{pmatrix} \ .$$

It has full rank if $\hat{\omega}_i(\hat{r}_2 - \hat{r}_1) \neq 0$. As we showed before this condition is true. There are 4 unknowns and 3 equations, hence the number of the solutions is q.

Therefore,

$$\Pr\left[\ \mathsf{Inv}''_{\mathsf{i,j}} \mid \mathtt{Case3} \wedge \neg\mathsf{NR}\ \right] = \frac{1}{q} \leq \frac{1}{2^{k-1}} \ .$$

∎

Acknowledgments

Sasha: any changes? 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," *In Proceedings of ICALP '00 U. Montanari, J. D. P. Rolim and E. Welzl Eds.*, LNCS 1853, Springer-Verlag, 2000.

[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. 1807, 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.

[BlGo] 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, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," a manuscript, 2001, to appear in *SIAM Journal of Computing*. A preliminary version of this paper: "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," appeared in *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 ← 0 else s_1 ←$ Z_q EndIf
    s_2, r ←$ Z_q
    g^{a'} ← g^a · g^{s_1}
    g^{b'} ← (g^b)^r · g^{s_2}
    g^{c'} ← (g^c)^r · (g^a)^{s_2} · (g^b)^{r·s_1} · 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 $e = c - ab \bmod q$. Then we have

$$\left\{ \begin{array}{lcll} a' & = & a + s_1 \bmod q & \\ b' & = & br + s_2 \bmod q & \\ c' & = & cr + as_2 + brs_1 + s_1 s_2 \bmod q & = \quad a'b' + er \bmod q \end{array} \right. \tag{29}$$

Now we want to verify the claims in the table. If $c = ab \bmod q$ then we have $e = 0$ and hence from Equation (29) we have $c' = a'b' + er = a'b' \bmod q$. Also if $x = 0$ then from Equation (29) we have $a' = a + s_1 = a \bmod q$. 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 $\mathbb{Z}_q$ 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 $a', b', c' \in \mathbb{Z}_q$ 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

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

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 $b' = br + s_2 \bmod q$.

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

$$\Pr\left[\, B' = b', C' = c' \;\mid\; A' = a \,\right] \;=\; \frac{1}{q^2}\,, \tag{30}$$

the probability being over the choices of s_2, r. From Equation (29), 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

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

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 (30).

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

$$\Pr\left[A' = a', B' = b' \mid C' = a'b' \right] = \frac{1}{q^2}, \tag{31}$$

the probability being over the choices of s_1, s_2, r. Fix a choice of r. Then from Equation (29), 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 (31).

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

$$\Pr\left[A' = a', B' = b', C' = c' \right] = \frac{1}{q^3}, \tag{32}$$

the probability being over the choices of s_1, s_2, r. From Equation (29), 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

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

The determinant of the above matrix is

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

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 (32). ∎