An extended abstract of this paper appears in *Advances in Cryptology – ASIACRYPT '01*, Lecture Notes in Computer Science Vol. 2248, C. Boyd ed., Springer-Verlag, 2001. This is full version.

Key-Privacy in Public-Key Encryption

M. Bellare* A. Boldyreva† A. Desai‡ D. Pointcheval§

September 2001

Abstract

We consider a security property of encryption schemes that has been surfacing increasingly often of late. We call it "key-privacy" or "anonymity". It asks that an eavesdropper in possession of a ciphertext not be able to tell which specific key, out of a set of known public keys, is the one under which the ciphertext was created— meaning the receiver is anonymous from the point of view of the adversary. We investigate the anonymity of known encryption schemes. We prove that the El Gamal scheme provides anonymity under chosen-plaintext attack assuming the Decision Diffie-Hellman problem is hard and that the Cramer-Shoup scheme provides anonymity under chosen-ciphertext attack under the same assumption. We also consider anonymity for trapdoor permutations. Known attacks indicate that the RSA trapdoor permutation is not anonymous and neither are the standard encryption schemes based on it. We provide a variant of RSA-OAEP that provides anonymity in the random oracle model assuming RSA is one-way. We also give constructions of anonymous trapdoor permutations, assuming RSA is one-way, which yield anonymous encryption schemes in the standard model.

Keywords: Encryption, key-privacy, anonymity, El Gamal, Cramer-Shoup, RSA, OAEP.

*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 above-mentioned grants of first author.

‡NTT Multimedia Communications Laboratories, 250 Cambridge Avenue, Suite 300, Palo Alto, CA 94306. E-mail: desai@nttmcl.com

§Laboratoire d'Informatique de l'École Normale Supérieure, 45 rue d'Ulm, F – 75230 Paris Cedex 05. E-mail: david.pointcheval@ens.fr URL: http://www.dmi.ens.fr/~pointche/

Contents

1 Introduction

The classical security requirement of an encryption scheme is that it provide privacy of the encrypted data. Popular formalizations— such as indistinguishability (semantic security) [22] or non-malleability [15], under either chosen-plaintext or various kinds of chosen-ciphertext attacks— are directed at capturing various data-privacy requirements. (A comprehensive treatment can be found in [5]).

In this paper we consider a different (additional) security requirement of an encryption scheme which we call *key-privacy* or *anonymity*. It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. This might sound odd, especially in the public-key setting which is our main focus: here the key under which encryption is performed is the public key of the receiver and being public there might not seem to be anything to keep private about it. The privacy refers to the information conveyed to the adversary regarding which specific key, out of a set of known public keys, is the one under which a given ciphertext was created. We call this anonymity because it means that the receiver is anonymous from the point of view of the adversary.

Anonymity of encryption has surfaced in various different places in the past, and found several applications, as we detail later. However, it lacks a comprehensive treatment. Our goal is to provide definitions, and then systematically study popular asymmetric encryption schemes with regard to their meeting these definitions. Below we discuss our contributions and then discuss related work.

1.1 Definitions

We suggest a notion we call "indistinguishability of keys" that formalizes the property of key-privacy in a way that seems to capture the intuition of previous research in a strong sense. In the formalization, the adversary knows two public keys pk_0, pk_1 corresponding to two different entities and gets a ciphertext C formed by encrypting some data under one of these keys. Possession of C should not give the adversary an advantage in determining under which of the two keys C was created. This can be considered under either chosen-plaintext attack or chosen-ciphertext attack, yielding two notions of security, IK-CPA and IK-CCA.

We also introduce the notion of an anonymous trapdoor permutation, which will serve as tool in some of the designs.

1.2 The search for anonymous asymmetric encryption schemes

In a heterogenous public-key environment, encryption will probably fail to be anonymous for trivial reasons. For example, different users might be using different cryptosytems, or, if the same cryptosystem, have keys of different lengths. (If one possible recipient has a RSA public key with a 1024 bit modulus and the other a RSA public key with a 512 bit modulus, the length of the RSA ciphertext will immediately enable an eavesdropper to know for which recipient the ciphertext is intended.) We can however hope for anonymity in a context where all users use the same security parameter or global parameters. We will look at specific systems with this restriction in mind.

Ideally, we would like to be able to prove that popular, existing and practical encryption schemes have the anonymity property (rather than having to design new schemes.) This would be convenient because then existing encryption-using protocols or software would not have to be altered in order for them to have the anonymity guarantees conferred by those of the encryption scheme. Accordingly, we begin by examining existing schemes. We will consider discrete log based schemes such as El Gamal and Cramer-Shoup, and also RSA based schemes such as RSA-OAEP.

It is easy to see that an encryption scheme could meet even the strongest notion of data-privacy— namely indistinguishability under chosen-ciphertext attack— yet not provide key-privacy. (The ciphertext could contain the public key.) Accordingly, existing results about data-privacy of asymmetric encryption schemes are not directly applicable. Existing schemes must be re-analyzed with regard to key-privacy.

In approaching this problem, we had no a priori way to predict whether or not a given asymmetric scheme would have the key-privacy property, and, if it did, whether the proof would be a simple modification of the known data privacy proof, or require new techniques. It is only by doing the work that one can tell what is involved.

We found that the above-mentioned discrete log based schemes did have the key-privacy property, and, moreover, that it was possible to prove this, under the same assumptions as used to prove data-privacy, by following the outline of the proofs of data-privacy with appropriate modifications. This perhaps unexpected strength of the discrete log based world (meaning not only the presence of the added security property in the popular schemes, but the fact that the existing techniques are strong enough to lead to a proof) seems important to highlight. In contrast, folklore attacks already rule out key-privacy for standard RSA-based schemes. Accordingly, we provide variants that have the property. Let us now look at these results in more detail.

1.3 Discrete log based schemes

The El Gamal cryptosystem over a group of prime order provably provides data-privacy under chosen-plaintext attack assuming the DDH (Decision Diffie-Hellman) problem is hard in the group [25, 12, 31, 3]. Let us now consider a system of users all of which work over the same group. (To be concrete, let q be a prime such that $2q+1$ is also prime, let G_q be the order q subgroup of quadratic residues of Z^*_{2q+1} and let $g \in G_q$ be a generator of G_q. Then q, g are system wide parameters based on which all users choose keys.) In this setting we prove that the El Gamal scheme meets the notion of IK-CPA under the same assumption used to establish data-privacy, namely the hardness of the DDH problem in the group. Thus the El Gamal scheme provably provides anonymity. Our proof exploits self-reducibility properties of the DDH problem together with ideas from the proof of data-privacy.

The Cramer-Shoup scheme [12] is proven to provide data-privacy under chosen-ciphertext attack, under the assumption that the DDH problem is hard in the group underlying the scheme. Let us again consider a system of users, all of which work over the same group, and for concreteness let it be the group G_q that we considered above. In this setting we prove that the Cramer-Shoup scheme meets the notion of IK-CCA assuming the DDH problem is hard in G_q. Our proof exploits ideas in [12, 3].

1.4 RSA based schemes

A simple observation that seems to be folkore is that standard RSA encryption does not provide anonymity, even when all modulii in the system have the same length. In all popular schemes, the ciphertext is (or contains) an element $y = x^e \bmod N$ where x is a random member of Z^*_N. Suppose an adversary knows that the ciphertext is created under one of two keys N_0, e_0 or N_1, e_1, and suppose $N_0 \leq N_1$. If $y \geq N_0$ then the adversary bets it was created under N_1, e_1, else it bets it was created under N_0, e_0. It is not hard to see that this attack has non-negligible advantage.

One approach to anonymizing RSA, suggested by Desmedt [14], is to add random multiples of the modulus N to the ciphertext. This seems to overcome the above attack, at least when the data encrypted is random, but results in a doubling of the length of the ciphertext. We look at a few

other approaches.

We consider an RSA-based encryption scheme popular in current practice, namely OAEP [8]. (It is the PKCS v2.0 standard [26], proved secure against chosen-ciphertext attack in the random oracle model [18].) We suggest a variant which we can prove is anonymous. Recall that OAEP is a randomized (invertible) transform that on input a message M picks a random string r and, using some public hash functions, produces a point $x = \mathsf{OAEP}(r, M) \in Z_N^*$ where N, e is the public key of the receiver. The ciphertext is then $y = x^e \bmod N$. Our variant simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $1 \leq y \leq 2^{k-2}$, where k is the length of N. We prove that this scheme meets the notion of IK-CCA in the random oracle model assuming RSA is a one-way function. (Data-privacy under chosen-ciphertext attack is proved under the same assumption as in [18].) The expected number of exponentiations for encryption being two, encryption in our variant is about twice as expensive as for OAEP itself, but this may be tolerable when the encryption exponent is small. The cost of decryption is the same as for OAEP itself, namely one exponentiation with the decryption exponent. As compared to Desmedt's scheme, the size of the ciphertext increases by only one bit rather than doubling. Our proof exploits the framework and techniques of [18, 8].

We then ask a more theoretical, or foundational, question, namely whether there exists an encryption scheme that can be proven to provide key-privacy based only on the assumption that RSA is one-way, meaning without making use of the random oracle model. To answer this we return to the classical techniques based on hardcore bits. We define a notion of anonymity for trapdoor permutations. We note that the above attack implies that RSA is not an anonymous trapdoor permutation, but we then design some trapdoor permutations which are anonymous and one-way as long as RSA is one-way. Appealing to known results about hardcore bits then yields an encryption scheme whose anonymity is proven based solely on the one-wayness of RSA. The computational costs of this approach, however, prohibit its being useful in practice.

1.5 Related work

In recent years, anonymous encryption has arisen in the context of mobile communications. Consider a mobile user A, communicating over a wireless network with some entity B. The latter is sending A ciphertexts encrypted under A's public key. A common case is that B is a base station. A wants to keep her identity private from an eavesdropping adversary. In this case A will be a member of some set of users whose identities and public keys are possibly known to the adversary. The adversary will also be able to see the ciphertexts sent by B to A. If the scheme is anonymous, however, the adversary will be unable to determine A's identity. A particular case of this is anonymous authenticated key exchange, where the communication between roaming user A and base station B is for the purpose of authentication and distribution of a session key based on the parties public keys, but the identity of A should remain unknown to an eavesdropper. Anonymity is targeted in authenticated key exchange protocols such as SKEME [23]. The author notes that a requirement for SKEME to provide anonymous authenticated key exchange is that the public-key encryption scheme used to encrypt under A's public key must have the key-privacy property.

In independent and concurrent work, Camenisch and Lysyanskaya [10] consider anonymous credential systems. Such a sytem enables users to control the dissemination of information about themselves. It is required that it be infeasible to correlate transactions carried out by the same user. The solution to this given in [10] makes use of a *verifiable circular* encryption scheme that needs to have the key-privacy property. They provide a notion similar to ours, but in the context of verifiable encryption. They observe that their variant of the El Gamal scheme is anonymous under chosen-plaintext attack.

Sako [28] considers the problem of achieving bid secrecy and verifiability in auction protocols. Their approach is to express each bid as an encryption of a known message, with the *key* to encrypt it corresponding to the value of the bid. Thus, what needs to be hidden is not the message that is encrypted, but the key used to encrypt it. The bid itself can be identified by finding the corresponding decrypting key that successfully decrypts to the given message. Unlike the previous examples, where the key-privacy property was needed to protect identities, this application shows how that property can be exploited to satisfy a secrecy requirement. Sako also considered a notion similar to ours and gave a variant of the El Gamal scheme that was expected to be secure in that sense.

Formal notions of key-privacy have appeared in the context of symmetric encryption [1, 13, 17]. Abadi and Rogaway [1] prove that popular modes of operation of block ciphers, such as CBC, provide key-privacy if the block cipher is a pseudorandom permutation.

The notion given by Desai [13], like ours, is concerned with the privacy of keys. However, the goal, model and setting in which it is considered differs from ours— the goal there is to capture a security property for block-cipher-based encryption schemes that implies that exhaustive key-search on them is slowed down proportional to the size of the ciphertext. There is, however, a similarity between our definitions (suitably adapted to the symmetric setting) and those of Abadi and Rogaway [1] and Fischlin [17]. Although the exact formalizations differ, it is not hard to see that there is an equivalence between the three for chosen-plaintext attack.

Chosen-ciphertext attacks do not seem to have been considered before in the context of key-privacy. In fact, Fischlin [17] observes that giving decryption oracles to the adversary in their setting makes its task trivial. However, in our formalization chosen-ciphertext attacks can be modeled by giving decryption oracles and then putting an appropriate restriction on their use. The restriction is the most natural and is anyway in effect for modeling semantic security against chosen-ciphertext attack. This allows us to make a distinction between those encryption schemes that are anonymous under chosen-ciphertext attack, such as Cramer-Shoup, and those that are not, such as El Gamal— just as there are schemes that are semantically secure under chosen-plaintext attack but not under chosen-ciphertext attack.

2 Notions of Key-Privacy

The notions of security typically considered for encryption schemes are "indistinguishability of encryptions under chosen-plaintext attack" (IE-CPA) [22] and "indistinguishability of encryptions under adaptive chosen-ciphertext attack" (IE-CCA) [27]. It is well-known that these capture strong data-privacy properties. However, they do not guarantee that some partial information about the underlying *key* is not leaked. Indeed, in a public-key encryption scheme, the entire public-key could be made an explicit part of the ciphertext and yet the scheme could meet the above-mentioned data-privacy notions. We want to make a distinction between such schemes and those that do not leak information about the underlying key. As noted earlier, schemes of the latter kind are necessary if the anonymity of receivers is a concern.

We are interested in formalizing the inability of an adversary, given a challenge ciphertext, to learn any information about the underlying plaintext or key. It is not hard to see that the goals of data-privacy and key-privacy are orthogonal. We recognize that existing encryption schemes are likely to have already been investigated with respect to their data-privacy security properties. Hence it is useful, from a practical point of view, to isolate the key-privacy requirements from the data-privacy ones. We do this in the form of two notions: "indistinguishability of keys under chosen-plaintext attack" (IK-CPA) and "indistinguishability of keys under adaptive chosen-ciphertext

attack" (IK-CCA). We begin with a syntax for public-key encryption schemes, divorcing syntax from formal notions of security.

SYNTAX. The syntax of an encryption scheme specifies what algorithms make it up. We augment the usual formalization in order to better model practice, where users may share some fixed "global" information.

A *public-key encryption scheme* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms. The *common-key generation* algorithm $\mathcal{G}$ takes as input some security parameter k and returns some common key I. It can be deterministic or randomized. The *key generation* algorithm $\mathcal{K}$ is a randomized algorithm that takes as input the common key I and returns a pair (pk, sk) of keys, the public key and a matching secret key, respectively; we write $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}(I)$. (Here I may be just a security parameter k, or include some additional information. For example in a Diffie-Hellman based scheme, I might include, in addition to k, a global prime number and generator of a group which all parties use to create their keys.) The *encryption* algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a *plaintext* x to return a *ciphertext* y; we write $y \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(x)$. The *decryption* algorithm $\mathcal{D}$ is a deterministic algorithm that takes the secret key sk and a ciphertext y to return the corresponding plaintext x or a special symbol $\perp$ to indicate that the ciphertext was invalid; we write $x \leftarrow \mathcal{D}_{sk}(y)$ when y is valid and $\perp \leftarrow \mathcal{D}_{sk}(y)$ otherwise. Associated to each public key pk is a *message space* $\mathrm{MsgSp}(pk)$ from which x is allowed to be drawn. We require that $\mathcal{D}_{sk}(\mathcal{E}_{pk}(x)) = y$ for all $x \in \mathrm{MsgSp}(pk)$.

INDISTINGUISHABILITY OF KEYS. We give a notion of key-privacy under chosen-plaintext and chosen-ciphertext attacks. We think of an adversary running in two stages. In the find stage it takes two public keys pk_0 and pk_1 (corresponding to secret keys sk_0 and sk_1, respectively) and outputs a message x together with some state information s. In the guess stage it gets a challenge ciphertext y formed by encrypting at random the messages under one of the two keys, and must say which key was chosen. In the case of a chosen-ciphertext attack the adversary gets oracles for $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$ and is allowed to invoke them on any point with the restriction (on both oracles) of not querying y during the guess stage.

Definition 1 [IK-CPA, IK-CCA] Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $b \in \{0, 1\}$ and $k \in \mathsf{N}$. Let $A_{\mathrm{cpa}}, A_{\mathrm{cca}}$ be adversaries that run in two stages and where A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$. Now, we consider the following experiments:

Experiment $\mathbf{Exp}^{\text{ik-cpa-}b}_{\mathcal{PE},A_{\mathrm{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\text{ik-cca-}b}_{\mathcal{PE},A_{\mathrm{cca}}}(k)$
$I \stackrel{R}{\leftarrow} \mathcal{G}(k)$	$I \stackrel{R}{\leftarrow} \mathcal{G}(k)$
$(pk_0, sk_0) \stackrel{R}{\leftarrow} \mathcal{K}(I);\ (pk_1, sk_1) \stackrel{R}{\leftarrow} \mathcal{K}(I)$	$(pk_0, sk_0) \stackrel{R}{\leftarrow} \mathcal{K}(I);\ (pk_1, sk_1) \stackrel{R}{\leftarrow} \mathcal{K}(I)$
$(x, s) \leftarrow A_{\mathrm{cpa}}(\mathsf{find}, pk_0, pk_1)$	$(x, s) \leftarrow A_{\mathrm{cca}}^{\mathcal{D}_{sk_0}(\cdot), \mathcal{D}_{sk_1}(\cdot)}(\mathsf{find}, pk_0, pk_1)$
$y \leftarrow \mathcal{E}_{pk_b}(x)$	$y \leftarrow \mathcal{E}_{pk_b}(x)$
$d \leftarrow A_{\mathrm{cpa}}(\mathsf{guess}, y, s)$	$d \leftarrow A_{\mathrm{cca}}^{\mathcal{D}_{sk_0}(\cdot), \mathcal{D}_{sk_1}(\cdot)}(\mathsf{guess}, y, s)$
Return d	Return d

Above it is mandated that A_{cca} never queries $\mathcal{D}_{sk_0}(\cdot)$ or $\mathcal{D}_{sk_1}(\cdot)$ on the challenge ciphertext y. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ we define the *advantages* of the adversaries via

$$\mathbf{Adv}^{\text{ik-atk}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) \quad = \quad \Pr[\,\mathbf{Exp}^{\text{ik-atk-1}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ik-atk-0}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1\,]\,.$$

The scheme $\mathcal{PE}$ is said to be *IK-CPA secure* (respectively *IK-CCA secure*) if the function $\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{PE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. ▌

The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation. (Note that the execution time refers to the entire experiment, not just the adversary. In particular, it includes the time for key generation, challenge generation, and computation of responses to oracle queries if any.) The same convention is used for all other definitions in this paper and will not be explicitly mentioned again.

ANONYMOUS ONE-WAY FUNCTIONS. A *family of functions* $F = (K, S, E)$ is specified by three algorithms. The randomized *key-generation* algorithm K takes input the security parameter $k \in \mathsf{N}$ and returns a pair (pk, sk) where pk is a public key, and sk is an associated secret key. (In cases where the family is not trapdoor, the secret key is simply the empty string.) The randomized *sampling* algorithm S takes input pk and returns a random point in a set that we call the domain of pk and denote $\mathrm{Dom}_F(pk)$. We usually omit explicit mention of the sampling algorithm and just write $x \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk)$. The deterministic *evaluation* algorithm E takes input pk and a point $x \in \mathrm{Dom}_F(pk)$ and returns an output we denote by $E_{pk}(x)$. We let $\mathrm{Rng}_F(pk) = \{\, E_{pk}(x) \,:\, x \in \mathrm{Dom}_F(pk) \,\}$ denote the range of the function $E_{pk}(\cdot)$. We say that F is a family of *trapdoor* functions if there exists a deterministic *inversion* algorithm I that takes input sk and a point $y \in \mathrm{Rng}_F(pk)$ and returns a point $x \in \mathrm{Dom}_F(pk)$ such that $E_{pk}(x) = y$. We say that F is a family of *permutations* if $\mathrm{Dom}_F(pk) = \mathrm{Rng}_F(pk)$ and E_{pk} is a permutation on this set.

Definition 2 Let $F = (K, S, E)$ be a family of functions. Let $b \in \{0, 1\}$ and $k \in \mathsf{N}$ be a security parameter. Let A, B be adversaries. Now, we consider the following experiments:

Experiment $\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,B}(k)$	Experiment $\mathbf{Exp}^{\text{ik-fnc-}b}_{F,A}(k)$
$(pk, sk) \stackrel{R}{\leftarrow} K(k)$	$(pk_0, sk_0) \stackrel{R}{\leftarrow} K(k)$
$x_1 \| x_2 \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk)$ where $\lvert x_1\rvert = \theta \cdot \lvert(x_1 \| x_2)\rvert$	$(pk_1, sk_1) \stackrel{R}{\leftarrow} K(k)$
$y \leftarrow E_{pk}(x_1 \| x_2)$	$x \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk_b)$
$x_1' \leftarrow B(pk, y)$ where $\lvert x_1'\rvert = \lvert x_1\rvert$	$y \leftarrow E_{pk_b}(x)$
For any x_2' if $E_{pk}(x_1' \| x_2') = y$ then return 1	$d \leftarrow A(pk_0, pk_1, y)$
Else return 0	Return d

We define the advantages of the adversaries via

$$\begin{aligned} \mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,B}(k) &= \Pr[\,\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,B}(k) = 1\,] \\ \mathbf{Adv}^{\text{ik-fnc}}_{F,A}(k) &= \Pr[\,\mathbf{Exp}^{\text{ik-fnc-1}}_{F,A}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ik-fnc-0}}_{F,A}(k) = 1\,]\,. \end{aligned}$$

The family F is said to be *θ-partial one-way* if the function $\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,B}(\cdot)$ is negligible for any adversary B whose time complexity is polynomial in k. The family F is said to be *anonymous* if the function $\mathbf{Adv}^{\text{ik-fnc}}_{F,A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k. The family F is said to be *perfectly anonymous* if $\mathbf{Adv}^{\text{ik-fnc}}_{F,A}(k) = 0$ for every k and every adversary A. ▌

Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness. As the above indicates, we expect that information-theoretic anonymity is possible for one-way functions, even though not for encryption schemes.

3 Anonymity of DDH based schemes

The DDH based schemes we consider work over a group of prime order. This could be a subgroup of order q of Z_p^* where p, q are primes such that q divides $p-1$. It could also be an elliptic curve group of prime order. For concreteness our description is for the first case. Specifically if q is a prime

such that $2q+1$ is also prime we let G_q be the subgroup of quadratic residues of Z_p^*. It has order q. A *prime-order-group generator* is a probabilistic algorithm that on input the security parameter k returns a pair (q, g) satisfying the following conditions: q is a prime with $2^{k-1} < q < 2^k$; $2q+1$ is a prime; and g is a generator of G_q. (There are numerous possible specific prime-order-group generators.) We will relate the anonymity of the El Gamal and Cramer-Shoup schemes to the hardness of the DDH problem for appropriate prime-order-group generators. Accordingly we next summarize definitions for the latter.

Definition 3 [DDH] Let $\mathcal{G}$ be a prime-order-group generator. Let D be an adversary that on input q, g and three elements $X, Y, T \in G_q$ returns a bit. We consider the following experiments

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

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

We say that the DDH problem is hard for $\mathcal{G}$ if the function $\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(\cdot)$ is negligible for every algorithm D whose time-complexity is polynomial in k. ▌

El Gamal. The El Gamal scheme in a group of prime order is known to meet the notion of indistinguishability under chosen-plaintext attack under the assumption that the decision Diffie-Hellman (DDH) problem is hard. (This is noted in [25, 12] and fully treated in [31]). We want to look at the anonymity of the El Gamal encryption scheme under chosen-plaintext attack.

Let $\mathcal{G}$ be a prime-order-group generator. This is the common key generation algorithm of the associated scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, the rest of whose algorithms are as follows:

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

The message space associated to a public key (q, g, X) is the group G_q itself, with the understanding that all messages from G_q are properly encoded as strings of some common length whenever appropriate. Note that a generator g is the output of the common key generation algorithm, which means we fix g for all keys. We do it only for a simplicity reason and will show that all our results hold also for a case when each key uses a random generator g.

We now analyze the anonymity of the El Gamal scheme under chosen-plaintext attack.

Theorem 3.1 *Let $\mathcal{G}$ be a prime-order-group generator. If the DDH problem is hard for $\mathcal{G}$ then the associated El Gamal scheme $\mathcal{EG}$ is IK-CPA secure. Concretely, for any adversary A there exists a distinguisher D such that for any k*

$$\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k) \;\le\; 2\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) + \frac{1}{2^{k-2}}$$

and the running time of D is that of A plus $O(k^3)$. ▌

The proof of the above is in the full version of this paper [2].

CRAMER-SHOUP. The El Gamal scheme provides data privacy and anonymity against chosen-plaintext attack. We now consider the Cramer-Shoup scheme [12] in order to obtain the same security properties under chosen-ciphertext attack. The scheme uses collision-resistant hash functions so we begin by recalling what we need.

A family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ is defined by a probabilistic generator algorithm $\mathcal{GH}$ —which takes as input the security parameter k and returns a key 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 $\mathcal{EH}_K(M) \in \{0,1\}^{k-1}$.

Definition 4 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 $\mathbf{Exp}^{\mathrm{cr}}_{\mathcal{H},C}(k)$
$K \xleftarrow{R} \mathcal{GH}(k)$; $(x_0, x_1) \leftarrow C(K)$
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

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

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

Let $\overline{\mathcal{G}}$ be a prime-order-group generator. The common key generation algorithm of the associated Cramer-Shoup scheme $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is:

$$\text{Algorithm } \mathcal{G}(k):\ (q, g_1) \xleftarrow{R} \overline{\mathcal{G}};\ g_2 \xleftarrow{R} G_q;\ K \xleftarrow{R} \mathcal{GH}(k);\ \ \text{Return } (q, g_1, g_2, K).$$

The rest of algorithms are specified as follows:

Algorithm $\mathcal{K}(q, g_1, g_2, K)$	Algorithm $\mathcal{E}_{pk}(M)$	Algorithm $\mathcal{D}_{sk}(u_1, u_2, e, v)$
$g_1 \leftarrow g$	$r \xleftarrow{R} Z_q$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
$x_1, x_2, y_1, y_2, z \xleftarrow{R} Z_q$	$u_1 \leftarrow g_1^r$; $u_2 \leftarrow g_2^r$	If ${u_1}^{x_1+y_1\alpha}{u_2}^{x_2+y_2\alpha} = v$
$c \leftarrow g_1^{x_1} g_2^{x_2}$; $d \leftarrow g_1^{y_1} g_2^{y_2}$	$e \leftarrow h^r M$	then $M \leftarrow e/{u_1}^z$
$h \leftarrow g_1^z$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$	else $M \leftarrow \perp$
$pk \leftarrow (g_1, g_2, c, d, h, K)$	$v \leftarrow c^r d^{r\alpha}$	Return M
$sk \leftarrow (x_1, x_2, y_1, y_2, z)$	Return (u_1, u_2, e, v)	
Return (pk, sk)		

The message space is the group G_q. Note that the range of the hash function $\mathcal{EH}_K$ is $\{0,1\}^{k-1}$ which we identify with $\{0, \ldots, 2^{k-1}\}$. Since $q > 2^{k-1}$ this is a subset of Z_q. Again for simplicity we assume that g_1, g_2 are fixed for all keys but we will show that our results hold even if g_1, g_2 are chosen at random for all keys.

We now analyze the anonymity of $\mathcal{CS}$ under chosen-ciphertext attack.

Theorem 3.2 *Let $\overline{\mathcal{G}}$ be a prime-order-group generator and let $\mathcal{CS}$ be the associated Cramer-Shoup scheme. If the DDH problem is hard for $\overline{\mathcal{G}}$ then $\mathcal{CS}$ is anonymous in the sense of IK-CCA. Concretely, for any adversary A attacking the anonymity of $\mathcal{CS}$ under a chosen-ciphertext attack*

and making in total $q_d(\cdot)$ decryption oracle queries, there exists a distinguisher D_A for DDH and an adversary C attacking the collision-resistance of $\mathcal{H}$ such that

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k) \;\le\; 2\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D_A}(k) + 2\mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k) + \frac{q_d(k)+2}{2^{k-3}}.$$

and the running time of D_A and C is that of A plus $O(k^3)$. ∎

The proof of the above is in the full version of this paper [2].

4 Anonymity of RSA based schemes

The attack on RSA mentioned in Section 1 implies that the RSA family of trapdoor permutations is not anonymous. This means that all traditional RSA-based encryption schemes are not anonymous. We provide several ways to implement anonymous RSA-based encryption. First we take a direct approach, specifying an anonymous RSA-OAEP variant based on repetition and proving it secure in the random oracle model. Then we show how to construct anonymous trapdoor permutation families based on RSA and derive anonymous RSA-based encryption schemes from them. In particular, the latter leads to anonymous encryption schemes whose proofs of security are in the standard rather than the random oracle model. We begin with a description of the RSA family of trapdoor permutations we will use in this section. See Section 2 for notions of security for families of trapdoor permutations.

Example 4.1 The specifications of the *standard RSA family* of trapdoor permutations $\mathsf{RSA} = (K, S, E)$ is as follows. The key generation algorithm takes as input a security parameter k and picks random, distinct primes p, q in the range $2^{k/2-1} < p, q < 2^{k/2}$. (If k is odd, increment it by 1 before picking the primes.) It sets $N = pq$. It picks $e, d \in Z^*_{\varphi(N)}$ such that $ed \equiv 1 \pmod{\varphi(N)}$ where $\varphi(N) = (p-1)(q-1)$. The public key is N, e and the secret key is N, d. The sets $\text{Dom}_{\mathsf{RSA}}(N, e)$ and $\text{Rng}_{\mathsf{RSA}}(N, e)$ are both equal to Z^*_N. The evaluation algorithm is $E_{N,e}(x) = x^e \bmod N$ and the inversion algorithm is $I_{N,d}(y) = y^d \bmod N$. The sampling algorithm returns a random point in Z^*_N. ∎

The anonymity attack on RSA carries over to most encryption schemes based on it, including the most popular one, $\mathsf{OAEP}[\mathsf{RSA}]$. We next describe a variant of $\mathsf{OAEP}[\mathsf{RSA}]$ that preserves its data-privacy properties but is in addition anonymous.

Anonymous variant of RSA-OAEP. The original scheme and our variant are described in the random-oracle (RO) model [7]. All the notions of security, defined earlier, can be "lifted" to the RO setting in a straightforward manner. To modify the definitions, begin the experiment defining advantage by choosing random functions G and H, each from the set of all functions from some appropriate domain to appropriate range. Then provide a G-oracle and H-oracle to the adversaries, and allow that $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ may depend on G and H (which we write as $\mathcal{E}^{G,H}_{pk}$ and $\mathcal{D}^{G,H}_{sk}$).

The idea behind our variant is to repeat the standard encryption procedure under $\mathsf{OAEP}[\mathsf{RSA}]$, until the ciphertext falls in some "safe" range. We refer to our scheme as $\mathsf{RAEP}[\mathsf{RSA}]$ (for *repeated* asymmetric encryption with padding). More concretely, for $\mathsf{RSA} = (K, S, E)$, our scheme $\mathsf{RAEP}[\mathsf{RSA}] = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common key generator algorithm $\mathcal{G}$ takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k) + k_1(k) < k$ for all $k > 1$. This defines an associated plaintext-length function $n(k) = k - k_0(k) - k_1(k)$. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1 and runs the key-generation algorithm of the RSA family, namely K on k to get a public key (N, e) and secret key (N, d) (see Example 4.1). The public key for the

scheme pk is $(N,e),k,k_0,k_1$ and the secret key sk is $(N,d),k,k_0,k_1$. The other algorithms are depicted below. The oracles G and H which $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ reference below have input/output lengths of $G:\ \{0,1\}^{k_0} \mapsto \{0,1\}^{n+k_1}$ and $H:\ \{0,1\}^{n+k_1} \mapsto \{0,1\}^{k_0}$.

Algorithm $\mathcal{E}_{pk}^{G,H}(x)$
  $ctr = -1$
  Repeat
    $ctr \leftarrow ctr + 1$
    $r \xleftarrow{R} \{0,1\}^{k_0}$
    $s \leftarrow (x \| 0^{k_1}) \oplus G(r)$
    $t \leftarrow r \oplus H(s)$
    $v \leftarrow (s\|t)^e \bmod N$
  Until $(v < 2^{k-2}) \vee (ctr = k_1)$
  If $ctr = k_1$ then $y \leftarrow 10^{k_0+k_1} \| x$
  Else $y \leftarrow 0 \| v$
  Return y

Algorithm $\mathcal{D}_{sk}^{G,H}(y)$
  Parse y as $b\|v$ where b is a bit
  If $b = 1$ then parse v as $w\|x$ where $|x| = n$
    If $w = 0^{k_0+k_1}$ then $z \leftarrow x$
    Else (if $w \neq 0^{k_0+k_1}$) $z \leftarrow \perp$
  Else (if $b = 0$)
    $(s\|t) \leftarrow v^d \bmod N:\ |s| = k_1 + n;\ |t| = k_0$
    $r \leftarrow t \oplus H(s)$
    $(x\|p) \leftarrow s \oplus G(r):\ |x| = n;\ |p| = k_1$
    If $p = 0^{k_1}$ then $z \leftarrow x$
    Else $z \leftarrow \perp$
  Return z

Note that the valid ciphertexts under $\mathsf{OAEP}[\mathsf{RSA}]$ are (uniformly) distributed in $\mathrm{Rng}_{\mathsf{RSA}}(N,e)$, which is Z_N^*. Under $\mathsf{RAEP}[\mathsf{RSA}]$, valid ciphertexts take the form $0\|v$ where $v \in (Z_N^* \cap [1, 2^{k-2}])$. The expected running time of this scheme is approximately twice that of $\mathsf{OAEP}[\mathsf{RSA}]$ (and k_1 times more, in the worst case). The ciphertext is longer by one bit. However, unlike $\mathsf{OAEP}[\mathsf{RSA}]$, this scheme turns out to be IK-CCA secure. The (data-privacy) security of $\mathsf{OAEP}[\mathsf{RSA}]$ under CCA has already been established [18]. It is not hard to see that this result holds for $\mathsf{RAEP}[\mathsf{RSA}]$ as well. We omit the (simple) proof of this, noting only that the security (relative to $\mathsf{OAEP}[\mathsf{RSA}]$) degrades roughly by the probability that after k_1 repetitions, the ciphertext was still not in the desired range (and consequently, the plaintext had to be sent in the clear). Given this, we turn to determining its security in the IK-CCA sense. We show that if the RSA family of trapdoor permutations is *partial* one-way then $\Pi = \mathsf{RAEP}[\mathsf{RSA}]$ is anonymous.

Theorem 4.2 *If the* RSA *family of trapdoor permutations is partial one-way then* $\Pi = \mathsf{RAEP}[\mathsf{RSA}]$ *is anonymous. Concretely, for any adversary A attacking the anonymity of Π under a chosen-ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary M_A for the* RSA *family, such that for any $k, k_0(k), k_1(k)$ and $\theta = \frac{k-k_0(k)}{k}$,*

$$\begin{aligned}\mathbf{Adv}^{\text{ik-cca}}_{\Pi,A}(k) \ &\leq\ 32q_{\mathrm{hash}} \cdot ((1-\epsilon_1)\cdot(1-\epsilon_2)\cdot(1-\epsilon_3))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M_A}(k) + \\ &\quad\ q_{\mathrm{gen}} \cdot (1-\epsilon_3)^{-1} \cdot 2^{-k+2}\end{aligned}$$

where

$$\begin{aligned}\epsilon_1 \ &=\ \left(\frac{3}{4}\right)^{k/2-1}; \qquad \epsilon_2 \ =\ \frac{1}{2^{k/2-3}-1};\\ \epsilon_3 \ &=\ \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}},\end{aligned}$$

and the running time of M_A is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$. ∎

The proof of the above is in the full version of this paper [2]. Note that for typical parameters $k_0(k), k_1(k)$, and number of allowed queries $q_{\mathrm{gen}}, q_{\mathrm{hash}}$ and q_{dec}, the values of ϵ_1, ϵ_2 and ϵ_3 are very small. This means that if there exists an adversary that is successful in breaking $\mathsf{RAEP}[\mathsf{RSA}]$ in

the IK-CCA sense, then there exists an partial inverting adversary for the RSA family of trapdoor permutations that has a comparable advantage and running time. The partial one-wayness of RSA has been shown to be equivalent to the one-wayness of RSA, if a constant fraction of the most significant bits of the pre-image can be recovered [18]. Hence our result can be translated to one in terms of the one-wayness of RSA.

ENCRYPTION BASED ON ANONYMOUS TRAPDOOR PERMUTATIONS. Given that the standard RSA family is not anonymous, we seek families that are. We describe some simple RSA-derived anonymous families.

Construction 1 We define a family $F = (K, S, E)$ as follows. The key generation algorithm is the same as in the standard RSA family of Example 4.1. Let (N, e) be a public key and k the corresponding security parameter. We set $\mathrm{Dom}_F(N, e) = \mathrm{Rng}_F(N, e) = \{0,1\}^k$. Viewing Z_N^* as a subset of $\{0,1\}^k$ we define

$$E_{N,e}(x) \;=\; \begin{cases} x^e \bmod N & \text{if } x \in Z_N^* \\ x & \text{otherwise} \end{cases}$$

for any $x \in \{0,1\}^k$. This is a permutation on $\{0,1\}^k$. The sampling algorithm S on input N, e simply returns a random k-bit string. It is easy to see that this family is trapdoor. ▌

As we will see, the family F is perfectly anonymous. But it is not one-way. However, it is weakly one-way. Thus, standard transformations of weak to strong one-way functions can be applied. Most of these preserve anonymity. To be concrete, let us use one.

Construction 2 Let $\overline{F} = (K, \overline{S}, \overline{E})$ be obtained from F of Construction 1 by Yao's cross-product construction [32]. In detail, the key-generation algorithm is unchanged and for any key N, e we set $\mathrm{Dom}_{\overline{F}}(N, e) = \mathrm{Rng}_{\overline{F}}(N, e) = \{0,1\}^{k^2}$. Parsing a point from this domain as a sequence of k-bit strings we set $\overline{E}_{N,e}(x_1, \ldots, x_k) = (E_{N,e}(x_1), \ldots, E_{N,e}(x_k))$. The sampling algorithm is obvious and it is easy to see the family is trapdoor. ▌

Proposition 4.3 *The family $\overline{F}$ of Construction 2 is a perfectly anonymous family of trapdoor, one-way permutations, under the assumption that the standard RSA family is one-way.* ▌

The proof of one-wayness is a direct consequence of the known results on the security of the cross-product construction. (A proof of Yao's result can be found for example in [19].) The anonymity is easy to see. Regardless of the key, the adversary simply gets a random string of length k^2, and can have no advantage in determining the key based on it.

The drawback of the construction is that the cross product construction is costly, increasing both the computational and the space requirements. There are alternative amplification methods that do not increase space requirements, but we know of none that do not increase the computational cost.

Standard methods of trapdoor permutation based encryption yield anonymous schemes provided the underlying trapdoor permutation is anonymous. This means any encryption method based on hardcore bits [21].

Acknowledgements

The UCSD authors are supported in part by Bellare's 1996 Packard Foundation Fellowship in Science and Engineering.

References

[1] M. Abadi and P. Rogaway, "Reconciling two views of cryptography (The computational soundness of formal encryption)," *Proceedings of the First IFIP International Conference on Theoretical Computer Science*, LNCS Vol. 1872, Springer-Verlag, 2000.

[2] M. Bellare, A. Boldyreva, A. Desai and D. Pointcheval, "Key-privacy in public-key encryption," Full version of this paper available via the authors.

[3] M. Bellare, A. Boldyreva and S. Micali, "Public-key encryption in a multi-user setting: security proofs and improvements," *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.

[4] 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.

[5] 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.

[6] M. Bellare, J. Kilian and P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology – CRYPTO '94*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[7] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993.

[8] M. Bellare and P. Rogaway, "Optimal asymmetric encryption – How to encrypt with RSA," *Advances in Cryptology – EUROCRYPT '95*, Lecture Notes in Computer Science Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.

[9] 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.

[10] J. Camenisch and A. Lysyanskaya, "Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation," *Advances in Cryptology – EUROCRYPT '01*, Lecture Notes in Computer Science Vol. 2045, B. Pfitzmann ed., Springer-Verlag, 2001.

[11] D. Coppersmith, "Finding a small root of a bivariate integer equation; factoring with high bits known," *Advances in Cryptology – EUROCRYPT '96*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[12] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[13] A. Desai, "The security of all-or-nothing encryption: protecting against exhaustive key search," *Advances in Cryptology – CRYPTO '00*, Lecture Notes in Computer Science Vol. 1880, M. Bellare ed., Springer-Verlag, 2000.

[14] Y. Desmedt, "Securing traceability of ciphertexts: Towards a secure software escrow scheme," *Advances in Cryptology – EUROCRYPT '95*, Lecture Notes in Computer Science Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.

[15] D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," *SIAM J. of Computing*, to appear. Preliminary version in *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991.

[16] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol 31, 1985, pp. 469–472.

[17] M. Fischlin, "Pseudorandom Function Tribe Ensembles based on one-way permutations: Improvements and applications," *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science Vol. 1592, J. Stern ed., Springer-Verlag, 1999.

[18] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern, "RSA-OAEP is Secure under the RSA Assumption," *Advances in Cryptology – CRYPTO '01*, Lecture Notes in Computer Science Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.

[19] O. Goldreich, Foundations of Cryptography (Volume 1 - Basic Tools) July 1, 2000.

[20] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.

[21] O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions," *Proceedings of the 21st Annual Symposium on the Theory of Computing*, ACM, 1989.

[22] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. of Computer and System Sciences*, Vol. 28, April 1984, pp. 270–299.

[23] H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet," *Proceedings of the 1996 Internet Society Symposium on Network and Distributed System Security*, 1996.

[24] National Bureau of Standards, NBS FIPS PUB 81, "DES modes of operation," U.S Department of Commerce, 1980.

[25] 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.

[26] RSA Labs, "PKCS-1," `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/`.

[27] 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.

[28] K. Sako, "An auction protocol which hides bids of losers," *Proceedings of the Third International workshop on practice and theory in Public Key Cryptography (PKC 2000)*, LNCS Vol. 1751, H. Imai and Y. Zheng eds., Springer-Verlag, 2000.

[29] V. Shoup, "On formal models for secure key exchange, " Technical report. Theory of Cryptography Library: 1999 Records.

[30] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology – EUROCRYPT '96*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[31] 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)*, LNCS Vol. 1431, H. Imai and Y. Zheng eds., Springer-Verlag, 1998.

[32] A. Yao, "Theory and applications of trapdoor functions, " *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982.

A Proof of Theorem 3.1

Let A be an adversary attacking $\mathcal{EG}$ in the IK-CPA sense (cf. Definition 1). We will design a distinguisher D for the DDH problem (cf. Definition 3) so that

$$\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) \;\geq\; \frac{1}{2}\cdot \mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k) - \frac{1}{2^{k-1}}\;. \tag{1}$$

The statement of Theorem 3.1 follows. So it remains to specify D. D has input q, g, and also three elements $X, Y, T \in G_q$. It will use the adversary A as a subroutine. D first computes another Diffie-Hellman triple which has the same property and distribution as its own challenge triple using DDH random self-reducibility [30, 25, 29, 3]. This means that if its challenge is a real Diffie-Hellman triple so is its computed triple. Otherwise, it is a triple of random values in G_q. Using its challenge and computed triples, the distinguisher computes two public keys. D will provide for A as input for its find stage these two public keys. At the end of the find stage A outputs a message M and some state information s. As an input for a guess stage A gets from D a challenge ciphertext, which is an encryption of the message M under one of the public keys. The code for D is in Figure 1.

Adversary $D(q, g, X, Y, T)$
$b \xleftarrow{R} \{0,1\}$
$u \xleftarrow{R} Z_q$; $v \xleftarrow{R} Z_q$; $w \xleftarrow{R} Z_q$
$X_0 \leftarrow X$; $Y_0 \leftarrow Y$; $T_0 \leftarrow T$;
$X_1 \leftarrow X_0 \cdot g^u$; $Y_1 \leftarrow (Y_0)^w \cdot g^v$; $T_1 \leftarrow T^w \cdot X^v \cdot Y^{uw} \cdot g^{uv}$
$pk_0 \leftarrow (q, g, X_0)$; $pk_1 \leftarrow (q, g, X_1)$
$(M, s) \leftarrow A(\text{find}, pk_0, pk_1)$
$d \leftarrow A(\text{guess}, (Y_b, T_b \cdot M), s)$
If $b = d$ then return 1 else return 0

Figure 1: Adversary D for the proof of Theorem 3.1

We now proceed to analyze D. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above satisfy $T = g^{xy}$ where $X = g^x$ and $Y = g^y$ for some x, y in Z_q. We claim that the triple (X_1, Y_1, T_1) computed by D is also a valid Diffie-Hellman triple and X_1, Y_1, T_1 are all uniformly and independently distributed over G_q. This is because $X_1 = g^{x+u}, Y_1 = g^{wy+v}, T_1 = g^{(x+u)(wy+v)}$ and u, v, w are random elements in Z_q. Thus X_0, X_1 have the proper distribution of public keys for the El Gamal cryptosystem. Also, the challenge ciphertext is distributed exactly like an El Gamal encryption of M under public key pk_b. We use it to see that for any k

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

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above are all uniformly distributed over G_q. Clearly, $X_0, Y_0, T_0, X_1, Y_1, T_1$ are all uniformly and independently distributed over G_q. Again, we have a proper distribution public keys for the El Gamal cryptosystem. But now Y_b, T_b are random elements in G_q and are independent of anything else. This means that the challenge ciphertext gives A no information about b, in an information-theoretic sense. We have

$$\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k) = 1\,] \ \le\ \frac{1}{2} + \frac{1}{2^{k-1}}\ . \tag{3}$$

The last term accounts for the maximum probability that random inputs to D happen to have the distribution of a valid Diffie-Hellman triple. For any q this probability is less then $\frac{1}{2^{k-1}}$ since $2^{k-1} < q < 2^k$. Subtracting Equations 2 and 3 we get

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

which is Equation (1). It remains to justify the claim about the time-complexity of D. The overhead for D is essentially that of performing 5 exponentiation operations with respect to a base element in G_q and an exponent in Z_q and 5 multiplication operations of the elements in G_q, which we can bound by $O(k^3)$, and that's the added cost in time of D.

We now show that with a small modification this proof will hold for a case when a generator g is not an output of a common key generation algorithm but chosen at random for each key by a key generation algorithm. Then the fourth line in the algorithm for an adversary D in Figure 1

will change to

$$g_0 \leftarrow g \; ; \; r \stackrel{R}{\leftarrow} Z_q \; ; \; g_1 \leftarrow g_0^r \; ; \; X_1 \leftarrow X_0 \cdot g^u \; ; \; Y_1 \leftarrow (Y_0)^w \cdot g^{vr} \; ; \; T_1 \leftarrow T^w \cdot X^v \cdot Y^{uw} \cdot g^{uv}$$

and the fifth line will change correspondingly to

$$pk_0 \leftarrow (q, g_0, X_0) \; ; \; pk_1 \leftarrow (q, g_1, X_1).$$

B Proof of Theorem 3.2

We specify a strategy for D_A in Figure 2. Similarly to the proof of Theorem 3.1 D_A computes two pairs of public and secret keys using random self-reducibility of DDH, but now $g_2 = X$ is the same for two public keys. The adversary provides two public keys for A as input for its find stage. At the end of the find stage A outputs a message M and some state information s. As an input for the guess stage A gets from D_A a challenge ciphertext, which is an encryption of the message M under one of the public keys. The code for D_A appears in Figure 2.

As we noted in the proof of Theorem 3.1 here it is also possible for D_A to create two public keys using self-reducibility of DDH such that g_1, g_2 are not fixed and the proof with minor modifications will also hold for a case when public generation algorithms picks both generators at random for each key.

Lemma B.1 *For any k we have*

$$\Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D_A}(k) = 1\,] \;=\; \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k)\;. \;\blacksquare$$

Lemma B.2 *There exists a polynomial time adversary C such that for every k*

$$\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D_A}(k) = 1\,] \;\leq\; \frac{1}{2} + \frac{q_d(k)+2}{2^{k-2}} + \mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k)$$

where q_d is the number of decryption oracle queries made by A. $\blacksquare$

Proof of Theorem 3.2: This follows from Lemma B.1 and Lemma B.2. $\blacksquare$

It remains to prove the above two lemmas. The proof of Lemma B.1 is in Section B.1 and the proof of Lemma B.2 is in Section B.2.

B.1 Proof of Lemma B.1

We analyze D_A. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D_A}(k)$. To prove the claim of the lemma we show that under D_A's simulation the view of the adversary A is exactly as in the actual experiment. This means that the two public keys and challenge ciphertext given to A have the right distribution and that decryption queries are answered as in an actual experiment.

The input to D_A has the form $q, g, g^{r_1}, g^{r_2}, g^{r_1 r_2}$. We can read this also as $q, g_1, g_2, u_{1,0}, u_{2,0}$, where $u_{1,0} = g_1^{r_2}$ and $u_{2,0} = g_2^{r_2}$. We use the same reasoning as we used in the proof of Theorem 3.1 to show that both triples, the challenge triple $g_2, u_{1,0}, u_{2,0}$ and the computed triple $g_2, u_{1,1}, u_{2,1}$ are valid Diffie-Hellman triples and $u_{1,0}, u_{2,0}, u_{1,1}, u_{2,1}$ are all independently distributed. Therefore, (c_0, c_1, d_0, d_1) have the right distribution of public keys since they are computed exactly like in the actual experiment. To show that two public keys computed by D_A have the right distribution it remains to show that h_0, h_1 have the right distribution. In the real encryption algorithm $h = g_1^z$ for

Adversary $D_A(q, g, X, Y, T)$

$K \stackrel{R}{\leftarrow} \mathcal{GH}(k)$

$g_1 \leftarrow g$; $g_2 \leftarrow X$; $u_{1,0} \leftarrow Y$; $u_{2,0} \leftarrow T$

$w_0 \stackrel{R}{\leftarrow} Z_q$; $w_1 \stackrel{R}{\leftarrow} Z_q$

$u_{1,1} \leftarrow Y^{w_0} \cdot g_1^{w_1}$; $u_{2,1} \leftarrow T^{w_0} \cdot g_2^{w_1}$

$x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1} \stackrel{R}{\leftarrow} Z_q$

$c_0 \leftarrow g_1^{x_{1,0}} g_2^{x_{2,0}}$; $d_0 \leftarrow g_1^{y_{1,0}} g_2^{y_{2,0}}$; $h_0 \leftarrow g_1^{z_{1,0}} g_2^{z_{2,0}}$

$c_1 \leftarrow g_1^{x_{1,1}} g_2^{x_{2,1}}$; $d_1 \leftarrow g_1^{y_{1,1}} g_2^{y_{2,1}}$; $h_1 \leftarrow g_1^{z_{1,1}} g_2^{z_{2,1}}$

$sk_0 \leftarrow (x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0})$

$sk_1 \leftarrow (x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1})$

$pk_0 \leftarrow (g_1, g_2, c_0, d_0, h_0, K)$

$pk_1 \leftarrow (g_1, g_2, c_1, d_1, h_1, K)$

$b \stackrel{R}{\leftarrow} \{0, 1\}$

Run A

$(M, s) \leftarrow A(\text{find}, pk_0, pk_1)$

$e \leftarrow (u_{1,b})^{z_{1,b}} (u_{2,b})^{z_{2,b}} M$

$\alpha \leftarrow \mathcal{EH}_K(u_{1,b}, u_{2,b}, e)$

$v \leftarrow (u_{1,b})^{x_{1,b}+y_{1,b}\alpha} (u_{2,b})^{x_{2,b}+y_{2,b}\alpha}$

$d \leftarrow A(\text{guess}, u_{1,b}, u_{2,b}, e, v; s)$

replying to A's decryption queries at any stage as follows:

$A \stackrel{\mathcal{D}_{sk_i}}{\rightarrow} \bar{C}$ // *This denotes that A makes a query* $\bar{C}$ *to* $\mathcal{D}_{sk_i}$ *for* $i \in \{0, 1\}$

parse $\bar{C}$ as $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$

$\bar{\alpha} \leftarrow \mathcal{EH}_K(\bar{u}_1, \bar{u}_2, \bar{e})$

If $(\bar{u}_1)^{x_{1,i}+y_{1,i}\bar{\alpha}} (\bar{u}_2)^{x_{2,i}+y_{2,i}\bar{\alpha}} = \bar{v}$ then $m \leftarrow \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}}$ else $m \leftarrow \perp$

A gets m

If $b = d$ then return 1 (real) else return 0 (random)

Figure 2: Adversary D_A for the proof of Theorem 3.2

---

a random $z \in Z_q$. D_A computes $h_b = g_1^{z_{1,b}} g_2^{z_{2,b}}$ for $b \in 0, 1$ and random elements $z_{1,b}, z_{2,b} \in Z_q$. Let us denote $\omega = \log_{g_1} g_2$. Then we can rewrite h_b as $g_1^{z_{1,b}+\omega z_{2,b}} = g_1^{\bar{z}_b}$, where $\bar{z}_b$ denotes $z_{1,b} + \omega z_{2,b}$ and corresponds to z in the real algorithm. We can see that $z, \bar{z}_b$ have the same distribution.

Now we show that the challenge ciphertext $(u_{1,b}, u_{2,b}, e, v)$ has the right distribution. Clearly, $(u_{1,b}, u_{2,b})$ are of the right form. The encryption algorithm computes $e = h^r M = g_1^{rz} M$ for random $r, z \in Z_q$. D_A computes e differntly: $e = (u_{1,b})^{z_{1,b}} (u_{2,b})^{z_{2,b}} M$. We can rewrite it as $e = g_1^{r_2 z_{1,b} + r_2 \omega z_{2,b}} M = h^{r_2(z_{1,b}+\omega z_{2,b})} M = h^{r_2 \bar{z}_b} M$. Thus rz in the real encryption algorithm corresponds to $r_2 \bar{z}_b$. This shows that e computed by D_A has the right distribution, since $r, z, r_2, \bar{z}_b$ are all random elements in Z_q. The encryption algorithm computes $v = c^r d^{r\alpha}$. In the simulation $v = (u_{1,b})^{x_{1,b}+y_{1,b}\alpha} (u_{2,b})^{x_{2,b}+y_{2,b}\alpha} = g_1^{r_2 x_{1,b} + r_2 y_{1,b}\alpha} g_2^{r_2 x_{2,b} + r_2 y_{2,b}\alpha} = (g_1^{x_{1,b}} g_2^{x_{2,b}})^{r_2} (g_1^{y_{1,b}} g_2^{y_{2,b}})^{r_2 \alpha} = c_b^{r_2} d_b^{r_2 \alpha}$. This is a right form, since r_2 corresponds to r in a real experiment and they are both random elements in Z_q and α is properly computed.

To complete the proof we show that the decryption oracle queries $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ are answered as they should. This is true because the condition of a valid ciphertext is computed as in the actual experiment and the plaintext is computed as $M = \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}} = \bar{e}/h_i^{r_2}$ for $i \in \{0, 1\}$ if the query was

made to $\mathcal{D}_{sk_i}$, which is as in the actual decryption algorithm, because r, r_2 have the same uniform distribution in Z_q. So we have

$$\begin{aligned}\Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D_A}(k)=1\,] &= \frac{1}{2}\cdot\Pr[\,\mathbf{Exp}^{\text{ik-cca-1}}_{\mathcal{CS},A}(k)=1\,]+\frac{1}{2}\cdot\Big(1-\Pr[\,\mathbf{Exp}^{\text{ik-cca-0}}_{\mathcal{CS},A}(k)=1\,]\Big)\\ &= \frac{1}{2}+\frac{1}{2}\cdot\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k).\end{aligned}$$

B.2 Proof of Lemma B.2

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D_A}(k)$. In this case, the inputs X, Y, T to D_A above and therefore $u_{1,0}, u_{2,0}$ are uniformly distributed over G_q. We can view the input (q, g, X, Y, T) as $(q, g_1, g_2, u_{1,0}, u_{2,0})$ where $u_{1,0} = g_1^{r_1}$, $u_2 = g_2^{r_2} = g_1^{\omega r_2}$, where r_1, r_2 are random elements in Z_q. When the adversary A makes a query $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ to a decryption oracle $\mathcal{D}_{sk_i}$, for $i \in \{0,1\}$ we say that the ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ is *invalid* if $\log_{g_1}\bar{u}_1 \neq \log_{g_2}\bar{u}_2$. Note, that the challenge ciphertext A gets is invalid. Let us define events associated to D_A.

- NR is true if $r_2 = r_1$ or $g_2 = 1$.
- Inv is true if during its execution the adversary A submits an invalid ciphertext to a decryption oracle $\mathcal{D}_{sk_0}$ or $\mathcal{D}_{sk_1}$ and does not get $\perp$

Lemma B.3 $\Pr[\,\mathsf{NR}\,] \leq 1/2^{k-2}$. ∎

Lemma B.4 *We have*

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

Lemma B.5 *There exists a polynomial-time adversary C such that for any k*

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

Proof of Lemma B.2: By conditioning we get

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

Applying Lemmas B.4, B.3 and B.5 to the above statement we get the claim of Lemma B.2. ∎

The proof of Lemmas B.3, B.4 and B.5 are in Sections B.2.1, B.2.2, B.3, respectively.

B.2.1 Proof of Lemma B.3

The claim is true since r_1, r_2 are random elements in Z_q and $2^{k-1} < q < 2^k$.

B.2.2 Proof of Lemma B.4

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 $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D_A}(k)$. We will denote an element of S as

$$\vec{s} \;=\; (x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1}, g_1, g_2, u_{1,0}, u_{2,0}, u_{1,1}, u_{2,1}, b)$$

and define the sample space as

$$S \;=\; \{\vec{s} : \vec{s} \in Z_q^{12} \times G_q^6 \times \{0,1\}\}$$

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 $\mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D_A}(k)$ when the random choices in that experiment are those given in $\vec{s}$. For simplicity we assume the adversary is deterministic. (The argument can simply be made for each choice of its coins.) The view then includes the inputs the adversary receives in its two stages, and the answers to all its oracle queries. The adversary's output is a deterministic function of its view.

Claim B.6 *Fix a specific view $\hat{V}$ of the adversary A simulated by D_A. Assume that the events $\neg\mathsf{NR} \wedge \neg\mathsf{Inv}$ hold for this view. Then*

$$\Pr\Big[\,\mathsf{View} = \hat{V} \mid b = 0\,\Big] = \Pr\Big[\,\mathsf{View} = \hat{V} \mid b = 1\,\Big]$$

This claim states that any view of the adversary A is equally likely given the bit b. We conclude the proof of Lemma B.4 given this claim.

Proof of Lemma B.4: Claim B.6 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 $\frac{1}{2}$. Thus the proof of Lemma B.4 follows since the distinguisher D_A outputs 1 only when A guesses the bit b correctly. ∎

It remains to prove the above claim.

Proof of Claim B.6: For simplicity of the analysis we will exclude the key $\hat{K}$ defining the hash function, which is fixed and a part of the two public keys, from the fixed view of the adversary we consider, because it is clearly independent from the bit b. We do not consider the answers of the decryption oracles to the valid ciphertext queries as a part of the view of the adversary because we show below that this does not give the adversary any additional information about the hidden bit b. We have

$$\hat{V} \;=\; (\hat{g}_1, \hat{g}_2, \hat{c}_0, \hat{d}_0, \hat{h}_0, \hat{g}_2, \hat{c}_1, \hat{d}_1, \hat{h}_1, \hat{u}_1, \hat{u}_2, \hat{e}, \hat{v})$$

Next for $i \in \{0,1\}$ define the event $E_i \subseteq S$ as the set of all $\vec{s} \in S$ such that $\vec{s}$ gives rise to $b = i$ and $\mathsf{View}(\vec{s}) = \hat{V}$ and $\neg\mathsf{NR}$ is true when the random choices in the experiment are $\vec{s}$. Then

$$\Pr[\,V = \hat{V} \wedge b = 0\,] = \frac{|E_0|}{|S|} = \frac{|E_0|}{2q^{19}}\,. \tag{4}$$

We next compute $|E_0|$. This is the number of solutions to the following system of 13 equations in 19 unknowns– $b, x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1}, g_1, q_2, u_{1,0}, u_{2,0}, u_{1,1}, u_{2,1}$:

$$
\begin{aligned}
b &= 0 && (5)\\
g_1 &= \hat{g}_1 && (6)\\
g_2 &= \hat{g}_2 && (7)\\
x_{1,0} + \hat{\omega} x_{2,0} &= \log_{\hat{g}_1} \hat{c}_0 && (8)\\
y_{1,0} + \hat{\omega} y_{2,0} &= \log_{\hat{g}_1} \hat{d}_0 && (9)\\
z_{1,0} + \hat{\omega} z_{2,0} &= \log_{\hat{g}_1} \hat{h}_0 && (10)\\
x_{1,1} + \hat{\omega} x_{2,1} &= \log_{\hat{g}_1} \hat{c}_1 && (11)\\
y_{1,1} + \hat{\omega} y_{2,1} &= \log_{\hat{g}_1} \hat{d}_1 && (12)\\
z_{1,1} + \hat{\omega} z_{2,1} &= \log_{\hat{g}_1} \hat{h}_1 && (13)\\
u_{1,0} &= \hat{u}_{1,0} && (14)\\
u_{2,0} &= \hat{u}_{2,0} && (15)\\
\hat{r}_1 z_{1,0} + \hat{r}_2 \hat{\omega} z_{2,0} &= \log_{\hat{g}_1} \frac{\hat{e}}{M} && (16)\\
\hat{r}_1 x_{1,0} + \hat{r}_1 \hat{\alpha} y_{1,0} + \hat{r}_2 \hat{\omega} x_{2,0} + \hat{r}_2 \hat{\omega} \hat{\alpha} y_{2,0} &= \log_{\hat{g}_1} \hat{v} && (17)
\end{aligned}
$$

Above $\hat{\omega} = \log_{\hat{g}_1} \hat{g}_2$, $\hat{r}_1 = \log_{\hat{g}_1} \hat{u}_{1,0}$, $\hat{r}_2 = \log_{\hat{g}_2} \hat{u}_{2,0}$, $\hat{\alpha} = \mathcal{EH}_{\hat{K}}(\hat{u}_{1,0}, \hat{u}_{2,0}, \hat{e})$. The variables with a hat, and M, denote the known constants whereas the variables without a hat denote 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 the valid ciphertext (u_1, u_2, e, v) is submitted to $\mathcal{D}_{sk_0}$. Suppose $\log_{g_1} u_1 = \log_{g_2} u_2 = r'$. Let $\alpha = \mathcal{EH}_K(u_1, u_2, e)$. Let m be the answer of a decryption oracle. Consider the equations corresponding to the ciphertext:

$$
\begin{aligned}
r' z_{1,0} + \omega r' z_{2,0} &= \log_{g_1} \frac{e}{m} && (18)\\
r' x_{1,0} + \omega r' x_{2,0} + r' \alpha y_{1,0} + \omega r' \alpha y_{2,0} &= \log_{g_1} v && (19)
\end{aligned}
$$

Note that Equation (18) is Equation (10) multiplied by r' and Equation (19) is Equation (8) plus $r'\alpha$ times Equation (9). 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.

We now rewrite equations 5-17 in a matrix form $F_{13\times 19} \times X_{19} = B_{13}$ in Figure 3. Here the matrix A is from equations 8, 9, 17, the matrix B is from 10, 16, the matrix C comes from 11, 12, 13 and the matrix D is from 6, 7, 14, 15. We prove that the matrix $F_{13\times 19}$ has the full rank and therefore the number of solutions of the corresponding system of equations is $q^{19-13} = q^6$. In order to prove that the matrix F has the full rank we prove that matrices A, B, C, D have full rank.

Let $\overset{\text{cond}}{\rightarrow}$ denotes the Gauss elimination algorithm where cond is a condition needed to apply it.

$$
A = \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \hat{r}_1 & \hat{r}_2\hat{\omega} & \hat{r}_1\hat{\alpha} & \hat{r}_2\hat{\omega}\hat{\alpha} \end{bmatrix} \overset{\text{cond}}{\rightarrow} \ldots \overset{\text{cond}}{\rightarrow} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & (\hat{r}_2 - \hat{r}_1)\hat{\omega}\hat{\alpha} \end{bmatrix}
$$

$$\begin{bmatrix} 1 & & & & & 0 & 0 \\ & A_{3\times 4} & & & 0 & 0 & 0 \\ & & B_{2\times 2} & & & 0 & 0 \\ & 0 & & C_{3\times 6} & & 0 & 0 \\ & & & & D_{4\times 4} & 0 & 0 \end{bmatrix} \times \begin{bmatrix} b \\ x_{1,0} \\ x_{2,0} \\ y_{1,0} \\ y_{2,0} \\ z_{1,0} \\ z_{2,0} \\ x_{1,1} \\ x_{2,1} \\ y_{1,1} \\ y_{2,1} \\ z_{1,1} \\ z_{2,1} \\ g_1 \\ q_2 \\ u_{1,0} \\ u_{2,0} \\ u_{1,1} \\ u_{2,1} \end{bmatrix} = \begin{bmatrix} 0 \\ \log_{\hat{g}_1} \hat{c}_0 \\ \log_{\hat{g}_1} \hat{d}_0 \\ \log_{\hat{g}_1} \hat{v} \\ \log_{\hat{g}_1} \hat{h}_0 \\ \log_{\hat{g}_1} \frac{\hat{e}}{M} \\ \log_{\hat{g}_1} \hat{c}_1 \\ \log_{\hat{g}_1} \hat{d}_1 \\ \log_{\hat{g}_1} \hat{h}_1 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{u}_{1,0} \\ \hat{u}_{2,0} \end{bmatrix}$$

Figure 3: The system of equations 5-17 in the matrix form.

---

where cond is that $\hat{r}_1 \neq \hat{r}_2, \hat{\omega} \neq 0$. If it holds then A has full rank since it contains a singular matrix.

$$\det(B) \;=\; \det \begin{bmatrix} 1 & \hat{\omega} \\ \hat{r}_1 & \hat{\omega}\hat{r}_2 \end{bmatrix} = \hat{\omega}(\hat{r}_2 - \hat{r}_1)$$

If $\hat{r}_1 \neq \hat{r}_2$ and $\hat{\omega} \neq 0$ then $\det(B) \neq 0$ and B has the full rank.

$$C \;=\; \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \hat{\omega} \end{bmatrix}$$

The matrix C has full rank since it contains a singular matrix.

$$\det(D) = 1 \neq 0$$

Obviously D has full rank.

Note that $\neg NR$ means that $\hat{r}_1 \neq \hat{r}_2$ and $\hat{\omega} \neq 0$. Therefore matrix F has the full rank and the number of solutions to the system of equations from Figure 3 is q^6 which is $|E_0|$.

Note that $|E_1|$ is the number of solutions of the system of equations $b = 1$, (6)-(15) and

$$\hat{r}_1 z_{1,1} + \hat{r}_2 \hat{\omega} z_{2,1} \;=\; \log_{\hat{g}_1} \frac{\hat{e}}{M} \tag{20}$$

$$\hat{r}_1 x_{1,1} + \hat{r}_1 \hat{\alpha} y_{1,1} + \hat{r}_2 \hat{\omega} x_{2,1} + \hat{r}_2 \hat{\omega} \hat{\alpha} y_{2,1} \;=\; \log_{\hat{g}_1} \hat{v} \tag{21}$$

where $\hat{\omega} = \log_{\hat{g}_1} \hat{g}_2$, $\hat{r}_1 = \log_{\hat{g}_1} \hat{u}_{1,1}$, $\hat{r}_2 = \log_{\hat{g}_2} \hat{u}_{2,1}$, $\hat{\alpha} = \mathcal{EH}_{\hat{K}}(\hat{u}_{1,1}, \hat{u}_{2,1}, \hat{e})$, both assuming $\neg\mathsf{NR}$.

We now claim that by symmetry of View and 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

$$\Pr[\,\mathsf{View} = \hat{V} \wedge b = 1\,] = \Pr[\,\mathsf{View} = \hat{V} \wedge b = 0\,]. \tag{22}$$

Equation (22) clearly implies Claim B.6 ∎

B.3 Proof of Lemma B.5

Assume the adversary A submits an invalid ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ to any of its decryption oracles $\mathcal{D}_{sk_i}$. By the rules of Definition 1 $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v}) \neq (u_{1,b}, u_{2,b}, e, v)$, where the latter denotes the challenge ciphertext. Let $\bar{\alpha} = \mathcal{EH}_K(\bar{u}_1, \bar{u}_2, \bar{e}), \alpha_b = \mathcal{EH}_K(u_{1,b}, u_{2,b}, e)$. Consider the following three special cases:

Case 1. $(\bar{u}_1, \bar{u}_2, \bar{e}) = (u_{1,b}, u_{2,b}, e)$.

Case 2. $(\bar{u}_1, \bar{u}_2, \bar{e}) \neq (u_{1,b}, u_{2,b}, e)$ and $\bar{\alpha} = \alpha_b$.

Case 3. $(\bar{u}_1, \bar{u}_2, \bar{e}) \neq (u_{1,b}, u_{2,b}, e)$ and $\bar{\alpha} \neq \alpha_b$.

We claim that there exists a polynomial time adversary C such that

$$\begin{aligned}
\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,] &= \Pr[\,\mathsf{Inv} \mid \text{Case 1} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\,\text{Case 1}\,] \\
&+ \Pr[\,\mathsf{Inv} \mid \text{Case 2} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\,\text{Case 2}\,] + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\,\text{Case 3}\,] \\
&\leq 0 + \Pr[\,\text{Case 2}\,] + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,] \\
&\leq 0 + \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H},C}(k) + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,]
\end{aligned} \tag{23}$$

The Equation (23) is justified by the following. In Case 1 $\bar{v} \neq v$ and the decryption oracle will reject. In Case 2 we can construct the adversary C which attacks the collision-resistance of $\mathcal{H}$ as the experiment from Definition 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 plus $O(k^3)$ because of modular exponentiations necessary for encryption keys generation, providing A with a challenge ciphertext and answering its decryption oracle queries.

We now bound $\Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,]$.

A ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ submitted to the $\mathcal{D}_{sk_i}$ for $i \in \{0, 1\}$ is accepted when

$$(\bar{u}_1)^{x_{1,0}+y_{1,0}\bar{\alpha}}(\bar{u}_2)^{x_{2,0}+y_{2,0}\bar{\alpha}} = \bar{v} \qquad \text{for } i = 0 \tag{24}$$

$$(\bar{u}_1)^{x_{1,1}+y_{1,1}\bar{\alpha}}(\bar{u}_2)^{x_{2,1}+y_{2,1}\bar{\alpha}} = \bar{v} \qquad \text{for } i = 1 \tag{25}$$

Let us define the following events:

- $\mathsf{Inv_{i,j}}$ is true if the adversary A during its i^{th} query submits an invalid ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ subject to conditions from Case 3 to a decryption oracle $\mathcal{D}_{sk_j}$ for $i \in \{1, \ldots, q_d\},\ j \in \{0, 1\}$ and does not get $\perp$.
- E_i^{inv} is a set $\{\vec{s} : \vec{s} \in S$ and $\vec{s}$ gives rise to a corresponding Equation (24) or Equation (25), $\neg NR\}$ and conditions from Case 3.

Let us first consider the simulation of $\mathcal{D}_{sk_0}$. To submit a ciphertext which will not be rejected the adversary should come up with the coefficients for Equation (24) which is consistent with its view, which with equal probability can contain a hidden bit $b = 0$ and $b = 1$. Therefore

$$\begin{aligned}\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] &= \frac{1}{2}\Pr\left[\,E_0^{\text{inv}} \mid E_0\,\right] + \frac{1}{2}\Pr\left[\,E_0^{\text{inv}} \mid E_1\,\right] \leq \frac{\Pr[\,E_0^{\text{inv}} \wedge E_0\,]}{2\Pr[\,E_0\,]} + \frac{\Pr[\,E_0^{\text{inv}} \wedge E_1\,]}{2\Pr[\,E_1\,]} \\ &= \frac{|E_0^{\text{inv}} \wedge E_0| \cdot |S|}{2|E_0| \cdot |S|} + \frac{|E_0^{\text{inv}} \wedge E_1| \cdot |S|}{2|E_1| \cdot |S|} = \frac{|E_0^{\text{inv}} \wedge E_0|}{2q^6} + \frac{|E_0^{\text{inv}} \wedge E_1|}{2q^6} \qquad (26)\end{aligned}$$

where $|E_0^{\text{inv}} \wedge E_0|$ is the number of solutions to the system of Equations (6)-(17) and 24 assuming $\neg\mathsf{NR}$, $|E_0^{\text{inv}} \wedge E_1|$ is the number of solutions to the system of Equations (6)-(15), 20, 21 and 25 assuming $\neg\mathsf{NR}$.

Let $\bar{u}_1 = {g_1}^{\bar{r}_1}, \bar{u}_2 = g_2^{\bar{r}_2} = g_1^{\omega\bar{r}_2}$. Adding Equation (24) to the system of Equations (6)-(17) will add a fourth row $(\bar{r}_1\ \bar{r}_2\hat{\omega}\ \bar{r}_1\bar{\alpha}\ \bar{r}_2\hat{\omega}\bar{\alpha})$ to the matrix A and a fifth element $\bar{v}$ to the column D from Figure 3.

$$\det(A) = \det\begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \hat{r}_1 & \hat{r}_2\hat{\omega} & \hat{r}_1\hat{\alpha} & \hat{r}_2\hat{\omega}\hat{\alpha} \\ \bar{r}_1 & \bar{r}_2\hat{\omega} & \bar{r}_1\bar{\alpha} & \bar{r}_2\hat{\omega}\bar{\alpha} \end{bmatrix} = \hat{\omega}(\bar{r}_2 - \bar{r}_1)(\hat{r}_2 - \hat{r}_1) \neq 0$$

This is because q is prime, $\neg\mathsf{NR}$ implies that $\hat{\omega} \neq 0, (\hat{r}_2 - \hat{r}_1) \neq 0$, $(\hat{r}_2 - \hat{r}_1) \neq 0$ because of the condition of the invalid ciphertext. We will have $\det(FF^T) \neq 0$ and the number of the solutions of the system of equations is $q^{19-14} = q^5$, which is $|E_0^{\text{inv}} \wedge E_0|$.

For calculating $|E_0^{\text{inv}} \wedge E_1|$ we do similar modifications to the system of equations, but in this case the modified matrix A will contain just three rows since the challenge ciphertext corresponds to pk_1 and the corresponding equation will contribute to a matrix C. We get

$$A = \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \bar{r}_1 & \bar{r}_2\hat{\omega} & \bar{r}_1\bar{\alpha} & \bar{r}_2\hat{\omega}\bar{\alpha} \end{bmatrix}$$

We showed in the proof of Claim B.6 that A has full rank. Thus F has full rank and $|E_0^{\text{inv}} \wedge E_0| = q^5$. We combine these results with Equation (26) and get

$$\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] \leq \frac{1}{q} \qquad (27)$$

By symmetry and the random choice of b we claim that $\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] = \Pr[\,\mathsf{Inv}_{1,1} \mid \neg\mathsf{NR}\,]$. Each time the adversary submits an invalid ciphertext and it gets rejected this reduces the set of the next possible decryption oracle queries at most by one. Therefore we have

$$\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR} \wedge \mathsf{Case3}\,] \leq \sum_{i=1}^{q_d(k)} \Pr[\,\mathsf{Inv}_{\mathsf{i},0} \mid \neg\mathsf{NR}\,] \leq \sum_{i=1}^{q_d(k)} \frac{1}{q-i+1} \leq \frac{q_d(k)}{q-q_d(k)+1} \leq \frac{2q_d(k)}{q} \leq \frac{q_d(k)}{2^{k-2}}$$

C Proof of Theorem 4.2

C.1 The (partial) inverting algorithm

We first define the behavior of an RSA partial inverting algorithm M_A using an IK-CCA adversary A. M_A is given $pk = (N, e)$ and a string $y \in Z_N^*$ where $|y| = k = n + k_0 + k_1$. Let $sk = (N, d)$

be the corresponding secret key. It is trying to find the $(n + k_1)$ leading bits of the e-th root of y modulo N.

(1) M_A first checks if $y \in [1, 2^{k-2}]$. If is isn't then it outputs Fail and halts; else it continues.

(2) M_A then runs the RSA key generator K with security parameter k to obtain $pk' = (N', e')$ and $sk' = (N', d')$. Then it picks a bit $b \xleftarrow{R} \{0, 1\}$, sets $pk_b \leftarrow (N, e)$ and $pk_{\bar{b}} \leftarrow (N', e')$. If the above y does not furthermore satisfy $y \in (Z^*_{N_0} \cap Z^*_{N_1})$, it outputs Fail and halts; else it continues.

(3) M_A initializes four lists, called its G-list, H-list, Y_0-list and Y_1-list to empty. It then runs A, simulating the two stages of A as indicated in the next two steps.

  (3.1) M_A simulates the find-stage of A by running A on input $(\mathsf{find}, pk_0, pk_1)$. M_A provides A with fair random coins and simulates A's oracles G, H and $\mathcal{D}^{G,H}_{sk_0}, \mathcal{D}^{G,H}_{sk_1}$ as described below. Let (x, s) be the output with which A halts.

  (3.2) Now M_A starts simulating the guess stage of A. It runs A on input (guess, y, s), responding to oracle queries as described below.

(4) Eventually A halts. M_A chooses a random element on the H-list, and outputs it as its guess for the leading part of the e-th root of y modulo N.

M_A simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle call g of G, then for each h on the H-list, M_A builds $z = h\|(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0, 1\}$, M_A checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus x\|0^{k_1}$. Otherwise, M_A outputs a random value G_g of length $n + k_1$. In both cases, g is added to the G-list. Then, for all h, M_A checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and Y_1-list respectively.

- When A makes an oracle call h of H, M_A provides A with a random string H_h of length k_0, and adds h to the H-list. Then for each g on the G-list, M_A builds $z = h\|(g \oplus H_h)$ and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M_A checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ to the Y_0-list and $y_{h,g,1}$ to the Y_1-list.

- When for $i \in \{0, 1\}$, A makes an oracle call y' of $\mathcal{D}^{G,H}_{sk_i}$, M_A checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $y' = y_{h,g,i}$. If there is, then it returns the first n-bits of $h \oplus G_g$ to A; else if $y' \notin (Y_0\text{-list} \cup Y_1\text{-list})$ it returns $\perp$ (indicating that y' is an "invalid" ciphertext).

C.2 Analysis

The intuition is that A in the above experiment is trying to predict b and M_A is trying to make the distribution provided to A look like what it would expect were it running under the experiment defining its success in the IK-CCA sense. Unfortunately, M_A does not provide A with a simulation which is quite perfect. A difference occurs if:

- M_A fails in the two first steps of the simulation;

- The simulation of the random oracles is not consistent;

- The simulation of the decryption oracle is not correct.

One can check that the running time of M_A is essentially that of A plus the time simulate the random oracles. (The simulation of the decryption oracles are very efficient.) The random oracles simulation is rather costly since for each call to G, one has to check all the elements in the H-list. And for any call to H, one has to check all the elements in the G-list. This increases the computational time by $q_{\text{gen}} \cdot q_{\text{hash}} \cdot O(k^3)$. We now proceed to the analysis of M_A's success probability.

We consider the probability space given by the above experiment when it continues beyond its first step. We can think of $(N, e), y$ as being drawn at random according to $((N, e), (N, d)) \leftarrow K(k)$; $y \leftarrow Z_N^* \cap [1, 2^{k-2}]$.

Let $w_0 = y^{d_0} \bmod N_0$ and write it as $w_0 = s_0 \,\|\, t_0$ where $|s_0| = n + k_1$ and $|t_0| = k_0$. Let r_0 be the random variable $t_0 \oplus H(s_0)$. Similarly, let $w_1 = y^{d_1} \bmod N_1$ and write it as $w_1 = s_1 \,\|\, t_1$ where $|s_1| = n + k_1$ and $|t_1| = k_0$. Let r_1 be the random variable $t_1 \oplus H(s_1)$. We consider the following events.

- FBad is true if:
  - A G-oracle query r_0 was made in the find stage, and $G_{r_0} \neq s_0 \oplus (x \| 0^{k_1})$, or
  - A G-oracle query r_1 was made in the find stage, and $G_{r_1} \neq s_1 \oplus (x \| 0^{k_1})$.
- GBad is true if:
  - A G-oracle query r_0 was made in the guess stage, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (x \| 0^{k_1})$, or
  - A G-oracle query r_1 was made in the guess stage, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (x \| 0^{k_1})$.
- DBad is true if:
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage, and $\Pr_0[\,\cdot\,]$ denote the probability distribution in the simulated game. We introduce the following additional events:

- YBad is true if $y \notin (Z_{N_0}^* \cap Z_{N_1}^*)$.
- FAskS is true if H-oracle query s_0 or s_1 was made in the find stage.
- AskR is true if, at the end of the guess stage, r_0 or r_1 is on the G-list.
- AskS is true if, at the end of the guess stage, s_0 or s_1 is on the H-list.

Let $\Pr_1[\,\cdot\,]$ denote the probability distribution in the simulated game provided $\neg$YBad.
We first lower bound $\Pr_1[\,\mathsf{AskS}\,]$.

$$\begin{aligned}
\Pr_1[\,\mathsf{AskS}\,] &\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \wedge \neg\mathsf{DBad}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot (\Pr_1[\,\neg\mathsf{DBad} \wedge \mathsf{AskS}\,] + \Pr_1[\,\neg\mathsf{DBad} \wedge \neg\mathsf{AskS}\,]) \\
&\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \wedge \neg\mathsf{AskS}\,]
\end{aligned}$$

$$
\begin{aligned}
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \,|\, \neg\mathsf{AskS}\,] \cdot \Pr_1[\,\neg\mathsf{AskS}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \,|\, \neg\mathsf{AskS}\,] \cdot (1 - \Pr_1[\,\mathsf{AskS}\,]) \\
&\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \,|\, \neg\mathsf{AskS}\,] - \Pr_1[\,\mathsf{AskS}\,] \\
&= (1/2) \cdot \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \,|\, \neg\mathsf{AskS}\,]
\end{aligned}
$$

We next lower bound each of the terms on the right above. Let $\Pr_2[\,\cdot\,]$ denote the probability distribution in the simulated game, provided $\neg\mathsf{DBad}$ and $\neg\mathsf{YBad}$.

Lemma C.1 *The probability that the events* AskR *and* AskS *are simultaneously true, assuming* $\neg\mathsf{DBad}$ *and* $\neg\mathsf{YBad}$ *is:*

$$
\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS}\,] \quad \geq \quad \frac{\varepsilon}{2} \cdot \left(1 - 2q_{\mathrm{gen}} \cdot 2^{-k_0} - 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}\right) - 2q_{\mathrm{gen}} \cdot 2^{-k}.
$$

Proof: We have

$$
\begin{aligned}
\Pr[A = b] \quad &= \quad \Pr[A = b \,|\, \mathsf{AskR} \wedge \mathsf{AskS}] \cdot \Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \\
&\qquad \Pr[A = b \,|\, \mathsf{AskR} \wedge \neg\mathsf{AskS}] \cdot \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] + \\
&\qquad \Pr[A = b \,|\, \neg\mathsf{AskR}] \cdot \Pr[\neg\mathsf{AskR}] \\
&\leq \quad \Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] + \Pr[A = b \,|\, \neg\mathsf{AskR}]
\end{aligned}
$$

Now given the way the message is masked by $G(r)$, we have that A cannot gain any advantage in the real game, without having asked r_0 or r_1 to G. Thus $\Pr[A = b \,|\, \neg\mathsf{AskR}] = 1/2$. Let ε denote the advantage of A. Then,

$$
\Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] \geq \varepsilon/2.
$$

Since the simulated game is perfect as long as G is true, we have:

$$
\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \mathsf{G}\,] + \Pr_2[\,\mathsf{AskR} \wedge \neg\mathsf{AskS} \,|\, \mathsf{G}\,] \;\geq\; \varepsilon/2.
$$

To bound the second term above, we consider the event

$$
(\mathsf{AskR} \wedge \neg\mathsf{AskS}) \wedge \mathsf{G} = (\mathsf{AskR} \wedge \neg\mathsf{AskS}) \wedge \neg(\mathsf{FBad} \vee \mathsf{GBad} \vee \mathsf{DBad}).
$$

This is the event that r_0 or r_1 has been asked to G, asking neither s_0 nor s_1 to H. Moreover, since $\neg(\mathsf{FBad} \vee \mathsf{GBad})$ holds, we have that the response must be $s_0 \oplus x0^{k_1}$ for a G query of r_0 and $s_1 \oplus x0^{k_1}$ for a G query of r_1. The probability of such an event is:

$$
\leq\; q_{\mathrm{gen}} \cdot 2^{-k_0} \cdot \left(2^{-n-k_1} + 2^{-n-k_1}\right) \leq 2q_{\mathrm{gen}} \cdot 2^{-k}.
$$

Therefore,

$$
\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \mathsf{G}\,] \geq \frac{\varepsilon}{2} - 2q_{\mathrm{gen}} \cdot \frac{2^{-k}}{\Pr_2[\,\mathsf{G}\,]},
$$

and,

$$
\begin{aligned}
\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS}\,] \quad &\geq \quad \Pr_2[\,(\mathsf{AskR} \wedge \mathsf{AskS}) \wedge \mathsf{G}\,] \geq \Pr_2[\,(\mathsf{AskR} \wedge \mathsf{AskS}) \,|\, \mathsf{G}\,] \cdot \Pr_2[\,\mathsf{G}\,] \\
&\geq \quad \frac{\varepsilon}{2} \cdot \Pr_2[\,\mathsf{G}\,] - 2q_{\mathrm{gen}} \cdot 2^{-k}.
\end{aligned}
$$

It remains to lower bound $\Pr_2[\mathsf{G}]$.

$$\begin{aligned}\Pr_2[\neg\mathsf{G}] &= \Pr_2[\mathsf{FBad} \vee \mathsf{GBad}] \\ &\leq \Pr_2[\mathsf{FBad} \vee \mathsf{GBad} \mid \neg\mathsf{FAskS}] + \Pr_2[\mathsf{FAskS}].\end{aligned}$$

If $\neg\mathsf{FAskS}$ holds and FBad or GBad occurs, then it means that A asked r_0 or r_1 to G without having asked s_0 and s_1 to H. Hence: $\Pr_2[\mathsf{FBad} \vee \mathsf{GBad} \mid \neg\mathsf{FAskS}] \leq 2q_{\mathrm{gen}} \cdot 2^{-k_0}$.

In the find stage, y and hence s_0 and s_1 are not in A's view. Since s_0 and s_1 are uniformly distributed in $\{0,1\}^{n+k_1}$, we have: $\Pr_2[\mathsf{FAskS}] \leq 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}$.

Thus,

$$\Pr_2[\mathsf{G}] \geq 1 - 2q_{\mathrm{gen}} \cdot 2^{-k_0} - 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}.$$

This completes the proof of Lemma C.1. ∎

Next we show that the event DBad is unlikely.

Lemma C.2 *The probability that* DBad *is true, provided* $\neg\mathsf{AskS}$*, is upper bounded as:*

$$\Pr_1[\mathsf{DBad} \mid \neg\mathsf{AskS}] \leq q_{\mathrm{dec}} \cdot \left(2 \cdot 2^{-k_1} + (2q_{\mathrm{gen}} + 1) \cdot 2^{-k_0}\right).$$

Proof: We first upper-bound the probability of the event DBad being true after *only one decryption query*, provided $\neg\mathsf{AskS}$. Let DBad_1 be the event that DBad is true after one decryption query.

Let $\mathcal{D}_{sk_i}$ (where $i \in \{0,1\}$) be the oracle to which the first decryption query is made and denote this query as y'. Let $w' = y'^{d_i} \bmod N_i$ and write it as $w' = s' \,\|\, t'$ where $|s'| = n + k_1$ and $|t'| = k_0$. Let r' be the random variable $t' \oplus H(s')$.

For the ciphertext y', let us denote by AskG the event that the query r' has been asked to G, and by AskH the event that the query s' has been asked to H.

Note that M_A's simulation fails if it rejects a valid ciphertext. Now a failure may occur if $r' = r_i$ or $s' = s_i$, or there will at least be an inconsistency in the simulation of the random oracles. This is because the oracle answers to r_i and s_i are only implicitly defined, and thus not available in the lists. In order to bound this failure probability, we define the following events:

- BadR is true if $r' = r_i$;
- BadS is true if $s' = s_i$.

We now consider the probability of event DBad_1, provided $\neg\mathsf{AskS}$:

$$\begin{aligned}\Pr_1[\mathsf{DBad}_1 \mid \neg\mathsf{AskS}] &= \Pr_1[\mathsf{DBad}_1 \wedge (\mathsf{BadR} \vee \mathsf{BadS}) \mid \neg\mathsf{AskS}] + \\ &\quad \Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH}) \mid \neg\mathsf{AskS}] + \\ &\quad \Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \mathsf{AskH}) \mid \neg\mathsf{AskS}].\end{aligned}$$

Note that if a ciphertext has been correctly built by A (r' has been asked to G and s' to H), then M_A will output the correct answer. Thus

$$\Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \mathsf{AskH}) \mid \neg\mathsf{AskS}] = 0.$$

In order to bound the second probability, observe that

$$\Pr_1[\,\neg(\mathsf{AskG} \wedge \mathsf{AskH})\,] \;=\; \Pr_1[\,\neg\mathsf{AskG}\,] + \Pr_1[\,\neg\mathsf{AskH} \wedge \mathsf{AskG})\,].$$

Thus,

$$\begin{aligned}
& \Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH})\,] \\
= \;& \Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg\mathsf{AskG}\,] + \\
& \Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \neg\mathsf{AskH})\,] \\
\leq \;& \Pr_1[\,\mathsf{DBad}_1 \wedge \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}\,] + \Pr_1[\,\mathsf{DBad}_1 \wedge \neg\mathsf{BadS} \wedge (\mathsf{AskG} \wedge \neg\mathsf{AskH})\,] \\
\leq \;& \Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}\,] + \Pr_1[\,\mathsf{AskG} \wedge \neg\mathsf{BadS} \wedge \neg\mathsf{AskH}\,] \\
\leq \;& \Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}\,] + \Pr_1[\,\mathsf{AskG} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskH}\,].
\end{aligned}$$

Given $\neg\mathsf{BadR}$ and $\neg\mathsf{AskG}$, $G(r')$ is unpredictable, and hence the probability that the k_1 least significant bits of $s' \oplus G(r')$ are all 0 is at most 2^{-k_1}. On the other hand, the probability of having asked $G(r')$, without any information about $H(s')$ (since $H(s')$ has not been explicitly asked and $s' \neq s_i$) is at most $q_{\mathrm{gen}} \cdot 2^{-k_0}$. Thus

$$\Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH})\,] \;\leq\; 2^{-k_1} + q_{\mathrm{gen}} \cdot 2^{-k_0}.$$

Moreover, since this event is independent of AskS,

$$\Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{AskS}\,] \;\leq\; 2^{-k_1} + q_{\mathrm{gen}} \cdot 2^{-k_0}.$$

Next we bound $\Pr_1[\,\mathsf{DBad}_1 \wedge (\mathsf{BadR} \vee \mathsf{BadS}) \,|\, \neg\mathsf{AskS}\,]$ as

$$\begin{aligned}
= \;& \Pr_1[\,\mathsf{DBad}_1 \wedge \mathsf{BadS} \,|\, \neg\mathsf{AskS}\,] + \Pr_1[\,\mathsf{DBad}_1 \wedge \mathsf{BadR} \wedge \neg\mathsf{BadS} \,|\, \neg\mathsf{AskS}\,] \\
\leq \;& \Pr_1[\,\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] + \Pr_1[\,\mathsf{BadR} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskS}\,].
\end{aligned}$$

It is easy to see that $\Pr_1[\,\mathsf{BadR} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskS}\,] \;\leq\; 2^{-k_0}$. ($H(s')$ being unpredictable and independent of $H(s_i)$ implies that r' is unpredictable and independent of r_i.)

We bound $\Pr_1[\,\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,]$ as:

$$\begin{aligned}
= \;& \Pr_1[\,\mathsf{DBad}_1 \wedge \mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] + \Pr_1[\,\mathsf{DBad}_1 \wedge \neg\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] \\
\leq \;& \Pr_1[\,\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] + \Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{AskG} \wedge \mathsf{BadS} \wedge \neg\mathsf{AskS}\,].
\end{aligned}$$

Now $\Pr_1[\,\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] \;\leq\; q_{\mathrm{gen}} \cdot 2^{-k_0}$. (If s_i has not been asked to H and $s' = s_i$ then $H(s')$ is unpredictable.)

We can bound the second term as: $\Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{AskG} \wedge \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] \;\leq\; 2^{-k_1}$. This is the probability that the redundancy holds (and hence M_A incorrectly rejected y') given that $H(s')$ is unpredictable and r' has not been asked to G. OAEP is a permutation and hence $s' = s_i$ (and $y' \neq y$) implies that $r' \neq r_i$, and that $G(r')$ is unpredictable. Thus

$$\Pr_1[\,\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}\,] \;\leq\; 2^{-k_1} + q_{\mathrm{gen}} \cdot 2^{-k_0}.$$

Putting this all together we bound the probability that DBad_1 is true, provided $\neg\mathsf{AskS}$:

$$\Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{AskS}\,] \;\leq\; 2 \cdot 2^{-k_1} + (2q_{\mathrm{gen}} + 1) \cdot 2^{-k_0}.$$

It follows that after q_{dec} decryption queries the probability that they were all correctly answered is:

$$1 - \Pr_1[\,\mathsf{DBad} \mid \neg\mathsf{AskS}\,] \geq \left(1 - \frac{2}{2^{k_1}} - \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)^{q_{\mathrm{dec}}} \geq 1 - q_{\mathrm{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right).$$

This completes the proof of Lemma C.2. ∎

Continuing, using the results of Lemmas C.1 and C.2, we have

$$\begin{aligned}\Pr_1[\,\mathsf{AskS}\,] &\geq \frac{1}{2} \cdot \left(\frac{\varepsilon}{2} \cdot \left(1 - \frac{2q_{\mathrm{gen}}}{2^{k_0}} - \frac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right) - \frac{2q_{\mathrm{gen}}}{2^{k_0}}\right) \cdot \left(1 - q_{\mathrm{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right) \\ &\geq \frac{\varepsilon}{4} \cdot \left(1 - \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} - \frac{2q_{\mathrm{dec}}}{2^{k_1}} - \frac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right) - \frac{q_{\mathrm{gen}}}{2^k}.\end{aligned}$$

Assuming that $y \in [1, 2^{k-2}]$ and $\neg\mathsf{YBad}$, we have by the random choice of b and symmetry, that the probability of M_A outputting s is at least $\frac{1}{2q_{\mathrm{hash}}} \cdot \Pr_1[\,\mathsf{AskS}\,]$.

We next bound the probabilities that $\neg\mathsf{YBad}$ is true and that y is in the good range.

Lemma C.3

$$\begin{aligned}\Pr_0[\,\mathsf{YBad}\,] &\leq \frac{1}{2^{k/2-3}-1} \\ \Pr_0\left[\,y \notin [1, 2^{k-2}]\,\right] &\leq \frac{3}{4} + \left(\frac{3}{4}\right)^{k/2-1}.\end{aligned}$$

Proof: We assume wlog that $N_1 \geq N_0$. We have

$$\begin{aligned}\Pr_0[\,\mathsf{YBad}\,] &= \Pr_0\left[\,b \xleftarrow{R} \{0,1\};\ y \xleftarrow{R} (Z^*_{N_b} \cap [1, 2^{k-2}]) :\ y \notin (Z^*_{N_0} \cap Z^*_{N_1})\,\right] \\ &\leq \Pr[\,b \xleftarrow{R} \{0,1\};\ y \in (Z^*_{N_1} \cap [1, 2^{k-2}]) :\ y \notin Z^*_{N_0}\,] \\ &\leq \frac{N_0 - \varphi(N_0)}{|Z^*_{N_1} \cap [1, 2^{k-2}]|} \leq \frac{N_0 - \varphi(N_0)}{2^{k-2} - (N_1 - \varphi(N_1))} \leq \frac{2 \cdot 2^{k/2}}{2^{k-2} - 2 \cdot 2^{k/2}}\end{aligned}$$

We use the bounds on the primes, $2^{k/2} - 1 < p_0, q_0, p_1, q_1 < 2^{k/2}$, to obtain the last inequality. Using these bounds we also have

$$\Pr_0\left[\,y \notin [1, 2^{k-2}]\,\right] = \Pr[\,y \xleftarrow{R} Z^*_N :\ y \notin [1, 2^{k-2}]\,] \leq \frac{N - 2^{k-2} - 1}{\varphi(N)} \leq \frac{3}{4} + \left(\frac{3}{4}\right)^{k/2-1}.$$

This completes the proof of Lemma C.3. ∎

We have that

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M_A}(k) \geq \left(1 - \Pr_0\left[\,y \notin [1, 2^{k-2}]\,\right]\right) \cdot (1 - \Pr_0[\,\mathsf{YBad}\,]) \cdot \left(\frac{\Pr_1[\,\mathsf{AskS}\,]}{2q_{\mathrm{hash}}}\right)$$

Substituting our bounds for the above probabilities and re-arranging the terms, we get the claimed result.