J. Cryptol. (2013) 26: 102–118
DOI: 10.1007/s00145-011-9115-0

Journal of
CRYPTOLOGY

Practical Chosen Ciphertext Secure Encryption from Factoring*

Dennis Hofheinz†
Karlsruhe Institute of Technology, Karlsruhe, Germany
Dennis.Hofheinz@kit.edu

Eike Kiltz
Ruhr-Universität Bochum, Bochum, Germany
eike.kiltz@rub.de

Victor Shoup‡
Courant Institute, New York University, New York, USA
shoup@cs.nyu.edu

Communicated by Keneth G. Paterson.

Received 26 December 2010
Online publication 20 December 2011

Abstract. We propose a practical public-key encryption scheme whose security against chosen-ciphertext attacks can be reduced in the standard model to the assumption that factoring is intractable.

Key words. Public-key encryption, Chosen-ciphertext security, Factoring.

1. Introduction

The security of almost any cryptographic primitive (such as public-key encryption or digital signatures) has to rely on the computational hardness of a certain number-theoretic problem. Unfortunately, since there are currently no tools available to rigorously prove lower bounds on the complexity of such problems, one has to base security on (unproven) *cryptographic hardness assumptions*. The only confidence we have in such assumptions is that after a sufficiently large period of time, nobody could successfully refute them. The most established cryptographic hardness assumption is without doubt the so-called *factoring assumption* which states that, given the product of two distinct large primes, it is computationally infeasible to reconstruct the primes. Despite

* This paper was solicited from Eurocrypt 2009.

† Work performed while the author was with the Centrum Wiskunde en Informatica (CWI), Amsterdam and supported by the Dutch Organization for Scientific Research (NWO).

‡ V. Shoup was supported by NSF grant CNS-0716690.

intensive research, no algorithm has been found that can efficiently factor composite numbers.

Main Result In this paper we propose a new public-key encryption scheme that is based on Rabin's trapdoor one-way permutation [42]. We can prove that the security of our scheme against adaptive chosen-ciphertext attacks (CCA security) is equivalent to the factoring assumption. Furthermore, the scheme is practical as its encryption performs only roughly two, and its decryption roughly one, modular exponentiations. This is the first scheme that simultaneously enjoys those two properties.

History The notion of CCA security is due to Rackoff and Simon [43] and is now widely accepted as the standard security notion for public-key encryption schemes. In contrast to security against passive adversaries (security against chosen-plaintext attacks aka semantic security), in a chosen-ciphertext attack the adversary plays an active role by obtaining the decryptions of ciphertexts (or even arbitrary bitstrings) of his choosing. The practical significance of such attacks was demonstrated by Bleichenbacher [4] by means of a CCA attack against schemes following the encryption standard PKCS #1.

Historically, the first scheme that was provably secure against CCA attacks is due to Dolev, Dwork, and Naor [18] (building on an earlier result by Naor and Yung [38]). Their generic construction is based on non-interactive zero-knowledge proofs, and therefore (using the proof systems from [21]) yields a scheme CCA secure under the factoring assumption. However, in practice these schemes are prohibitively impractical. The first practical schemes provably CCA secure under standard cryptographic hardness assumptions were due to Cramer and Shoup [15,16]. However, their framework of "hash proof systems" inherently relies on *decisional assumptions* such as the assumed hardness of deciding if a given integer has a square root modulo a composite number with unknown factorization (DQR assumption), or of deciding if a given tuple is a Diffie–Hellman tuple or not (DDH assumption). Until today, Cramer and Shoup's framework of hash proof systems (with its variations from [11,20,27,31,32,34]) and the recent concept of lossy trapdoor functions [40] yield the only known CCA secure practical encryption schemes based on an assumption related to factoring: the DQR assumption and Paillier's decisional composite residuosity (DCR) assumption. Currently, no practical scheme is known that is CCA secure solely under the factoring assumption (or even under the potentially stronger RSA assumption).

In general, decisional assumptions are a much stronger class of assumptions than computational assumptions. For example, deciding if a given integer has a modular square root or not may be much easier than actually computing a square root (or, equivalently, factoring the modulus). It is noteworthy that there are known ways to achieve CCA security that do not inherently rely on decisional assumptions (e.g., [9,13,14,25]). In particular, the first practical encryption scheme CCA secure under the *Computational* Diffie–Hellman (CDH) assumption was only recently proposed by Cash, Kiltz, and Shoup [14] and improved by Hanaoka and Kurosawa [25], and Haralambiev et al. [26]. On the other hand, Dan Boneh et al. [9] provided a practical encryption scheme CCA secure under the Bilinear Computational Diffie–Hellman (BCDH) assumption.

Random Oracle Schemes In a different line of research, Bellare and Rogaway [2,3] presented practical schemes for which they gave heuristic proofs of CCA security under

standard computational hardness assumptions. Their proofs are in the so-called random oracle model [2] where a hash function is treated as an ideal random function. We stress that although a proof in the random oracle model has a certain value, it is still only a heuristic security argument for any implementation of the scheme. In particular, there exist cryptographic schemes that are provably secure in the random oracle model yet that are insecure with *any* possible standard-model instantiation of the hash function [12].

Details of Our Construction In 1979 Rabin [42] proposed an encryption scheme based on the "modular squaring" trapdoor permutation whose one-wayness is equivalent to the factoring assumption. A semantically secure variant was later proposed by Goldwasser and Micali [23]. Our construction is based on the latter scheme [23] in its more efficient variant by Blum and Goldwasser [5] (which uses the Blum–Blum–Shub pseudorandom generator [7] to obtain an efficient hard-core function with linear output length). The Blum–Goldwasser scheme can easily be shown insecure against a CCA attack. Our main contribution consists of modifying the Blum–Goldwasser scheme such that it is provably CCA secure under the same hardness assumption yet it retains its high efficiency. Surprisingly, it is sufficient to add one additional group element to the ciphertexts that is then used for a consistency check in the decryption algorithm. For the consistency check itself, we also need to add two group elements to the public key. Another important ingrident of our scheme is that we work in the group of "signed quadratic residues" in which the computational problem of computing square roots is as hard as factoring, while the problem of recognizing group elements (i.e., signed quadratic residues) is easy.

Note that Paillier and Villar [39] (building on work of Williams [45]) show that the CCA security of schemes which only include an RSA modulus in the public key cannot be proven (using a black-box reduction) equivalent to factoring. In particular, this applies to the Blum–Goldwasser scheme [5] from which we start, so we have to modify the scheme's public key (and not only the ciphertexts). And indeed, given our modifications, our scheme's CCA security is *equivalent* to the factoring problem.

Proof Details At a more technical level, the additional group elements in the public key can be set up by a simulator such that it is possible to decrypt (without the knowledge of the scheme's secret key) all consistent ciphertexts, except the ciphertext that is used to challenge the adversary. This "all-but-one" simulation technique can be traced back at least to [36] where it was used in the context of pseudorandom functions.[1] In the encryption context, "all-but-one" simulations have been used in identity-based encryption [8] and were already applied to several encryption schemes in [9,10,14,27,29].

The main novelty is that our proof makes direct use of the fact that the underlying primitive is a trapdoor one-way permutation, rather than the Diffie–Hellman problem. Therefore, the scheme's consistency check can be directly implemented by the simulator *without* having access to some external gap-oracle (as in [9,10,29]) or using other extrinsic rejection techniques (such as "hash proof systems" [15,16], "twinning" [14], or

[1] We stress that our use of the term "all-but-one" refers to the ability to generate a secret key that can be used to decrypt all consistent ciphertexts except for an *externally given* ciphertext. This is very different from the techniques of, e.g., [16,18,38]: in these latter frameworks, the first step in the proof consists in *making the challenge ciphertext inconsistent*, and then constructing a secret key that can be used to decrypt *all* consistent ciphertexts. Hence, "all-but-one" really refers to an "artificially punctured" secret key.

authenticated symmetric encryption [27,32][2]). Thus, our proof technique is fundamentally different from all known approaches to obtain CCA security. This also includes the recent class of schemes based on lossy trapdoor functions [40].

Efficiency The resulting encryption scheme (which is actually a key encapsulation mechanism, see [16]) is very efficient: encryption needs roughly two, and decryption roughly one, modular exponentiations; the public key contains the modulus plus two group elements. (The modulus and one element can be viewed as system parameters shared among all parties.) To the best of our knowledge this is much more efficient than all known CCA-secure schemes based on an assumption *related* to factoring, even the ones based on a decisional assumption.

Follow-up Work Cramer et al. [17] explain our construction as a *hard algebraic set system*; Wee [44] explains our construction as an *extractable hash proof system*. Both works give abstractions of the "all-but-one" decryption in our simulation. In particular, both [17] and [44] provide an abstraction of the extra group elements in our ciphertext that enable us to set up decryption keys that can be used to decrypt all ciphertexts, except the challenge ciphertext. Mei et al. [35] propose a variant of our scheme with improved efficiency.

2. Preliminaries

2.1. *Notation*

We write $[N] = \{1, \ldots, N\}$. For group elements g, h, we denote by $\mathrm{dlog}_g h$ the discrete logarithm of h to the base g, i.e., the smallest $i \geq 0$ with $h = g^i$. A probabilistic polynomial-time (PPT) algorithm is a randomized algorithm which runs in strict polynomial time. If A is a probabilistic algorithm, we write $y \leftarrow A(x)$ to denote that the random variable y is defined as the output of A when run on input x and with fresh random coins. On the other hand, if S is a set, then $s \leftarrow S$ defines s as being uniformly and independently sampled from S. By k we denote the security parameter, which indicates the "amount of security" we desire. Typically, an adversarial advantage should be bounded by 2^{-k}, and a typical value for k is 80.

2.2. *Key Encapsulation Mechanisms*

Instead of a public-key encryption scheme we consider the conceptually simpler KEM framework. It is well known that an IND-CCA secure KEM combined with a (one-time) IND-CCA secure symmetric cipher (DEM) yields an IND-CCA secure public-key encryption scheme [16]. Efficient *one-time* IND-CCA secure DEMs can be constructed even without computational assumptions by using an encrypt-then-MAC paradigm [16] (or, alternatively, using computational assumptions such as strong pseudorandom permutations [41]).

A *key encapsulation mechanism* (*KEM*) $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms. Via $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, the key generation algorithm produces public/secret keys for security parameter $k \in \mathbb{N}$; via $(K, C) \leftarrow \mathsf{Enc}(pk)$, the encapsulation

[2] As opposed to generic CCA-secure symmetric encryption, a potentially weaker primitive.

algorithm creates a symmetric key[3] $K \in \{0,1\}^{\ell_K}$ together with a ciphertext C; via $K \leftarrow \mathsf{Dec}(sk, C)$, the possessor of secret key sk decrypts ciphertext C to get back a key K which is an element in $\{0,1\}^{\ell_K}$ or a special reject symbol $\perp$. For correctness, we require that for all possible $k \in \mathbb{N}$, and all $(K, C) \leftarrow \mathsf{Enc}(pk)$, we have $\Pr[\mathsf{Dec}(sk, C) = K] = 1$, where the probability is taken over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, and the coins of all the algorithms in the expression above.

The common requirement for a KEM is indistinguishability against chosen-ciphertext attacks (IND-CCA) [16], where an adversary is allowed to adaptively query a decapsulation oracle with ciphertexts to obtain the corresponding key. We are using the slightly simpler but equivalent one-phase definition from [30]. Formally:

Definition 1 (IND-CCA Security of a KEM). Let $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a KEM. For any PPT algorithm A, we define the following experiments $\mathsf{Exp}^{\text{CCA-real}}_{\mathsf{KEM},\mathsf{A}}$ and $\mathsf{Exp}^{\text{CCA-rand}}_{\mathsf{KEM},\mathsf{A}}$:

Experiment $\mathsf{Exp}^{\text{CCA-real}}_{\mathsf{KEM},\mathsf{A}}(k)$	**Experiment** $\mathsf{Exp}^{\text{CCA-rand}}_{\mathsf{KEM},\mathsf{A}}(k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
	$R \leftarrow \{0,1\}^{\ell_K}$
$(K^*, C^*) \leftarrow \mathsf{Enc}(pk)$	$(K^*, C^*) \leftarrow \mathsf{Enc}(pk)$
Return $\mathsf{A}^{\mathsf{Dec}(sk,\cdot)}(pk, K^*, C^*)$	Return $\mathsf{A}^{\mathsf{Dec}(sk,\cdot)}(pk, R, C^*)$

In the above experiments, the decryption oracle $\mathsf{Dec}(sk, \cdot)$, when queried with a ciphertext $C \neq C^*$, returns $K \leftarrow \mathsf{Dec}(sk, C)$. ($\mathsf{Dec}(sk, \cdot)$ ignores queries $C = C^*$.) We define A's advantage in breaking KEM's IND-CCA security as

$$\mathsf{Adv}^{\text{CCA}}_{\mathsf{KEM},\mathsf{A}}(k) := \frac{1}{2}\left|\Pr\left[\mathsf{Exp}^{\text{CCA-real}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{CCA-rand}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right]\right|.$$

A $(t_{\mathsf{KEM}}, \epsilon_{\mathsf{KEM}})$-*breaks* KEM*'s* IND-CCA *security* (*short:* A $(t_{\mathsf{KEM}}, \epsilon_{\mathsf{KEM}})$-*breaks* KEM) if A runs in time at most $t_{\mathsf{KEM}} = t_{\mathsf{KEM}}(k)$ and we have $\mathsf{Adv}^{\text{CCA}}_{\mathsf{KEM},\mathsf{A}}(k) \geq \epsilon_{\mathsf{KEM}}(k)$. We say that KEM *has indistinguishable ciphertexts under chosen-ciphertext attacks* (*short:* KEM *is* IND-CCA *secure*) if for all PPT A, the function $\mathsf{Adv}^{\text{CCA}}_{\mathsf{KEM},\mathsf{A}}(k)$ is negligible in k.

2.3. *Target-Collision Resistant Hashing*

Informally, we say that a function $\mathsf{T} : X \to Y$ is a target-collision resistant (TCR) hash function (aka universal one-way hash function [37]), if, given a random preimage $x \in X$, it is hard to find $x' \neq x$ with $\mathsf{T}(x') = \mathsf{T}(x)$.

Definition 2 (TCR Hash Function). Let $\mathsf{T} : X \to Y$ be a function. For an algorithm B, define

$$\mathsf{Adv}^{\text{TCR}}_{\mathsf{T},\mathsf{B}}(k) := \Pr\left[x \leftarrow X, x' \leftarrow \mathsf{B}(x) : x' \neq x \wedge \mathsf{T}(x') = \mathsf{T}(x)\right].$$

We say that B $(t_{\mathsf{T}}, \epsilon_{\mathsf{T}})$-*breaks* T*'s TCR property* (*short:* B $(t_{\mathsf{T}}, \epsilon_{\mathsf{T}})$-*breaks* T) iff B's running time is at most $t_{\mathsf{T}}(k)$ and $\mathsf{Adv}^{\text{TCR}}_{\mathsf{T},\mathsf{B}}(k) \geq \epsilon_{\mathsf{T}}(k)$. We say that T is *target-collision resistant* if for all PPT B, the function $\mathsf{Adv}^{\text{TCR}}_{\mathsf{T},\mathsf{B}}(k)$ is negligible in k.

[3] For simplicity we assume that the KEM's keyspace are bitstrings of length ℓ_K.

3. The Group of Signed Quadratic Residues

3.1. *Factoring Assumption*

A prime number P is called a *safe prime* iff $P = 2p+1$ for a prime p. We assume a PPT algorithm IGen that, on input of a security parameter k in unary, generates two random safe primes $P = 2p+1$ and $Q = 2q+1$ with $\mathsf{bitlength}(p) = \mathsf{bitlength}(q) = \ell_{\mathsf{N}}(k)/2-1$. We assume that p and q are odd, such that P and Q are congruent 3 modulo 4 and $N = PQ$ is a Blum integer. IGen returns N along with P and Q. Here $\ell_{\mathsf{N}}(k)$ denotes a function that represents, for any given security parameter k, the recommended (bit-)size of the composite modulus N. For simplicity, we will assume that $\ell_{\mathsf{N}}(k) \geq 2k$, so that $\ell_{\mathsf{N}}(k)/2 \geq k$. For the rest of the paper, we assume that N is generated by the factoring instance generator IGen.

Definition 3 (Factoring Assumption). For an algorithm F, we define its *factoring advantage* as

$$\mathsf{Adv}^{\mathsf{fac}}_{\mathsf{IGen},\mathsf{F}}(k) := \mathsf{Pr}\big[(N, P, Q) \leftarrow \mathsf{IGen}\big(1^k\big) : \mathsf{F}(N) = \{P, Q\}\big].$$

We say that F $(t_{\mathsf{fac}}, \epsilon_{\mathsf{fac}})$-*factors composite integers* if F runs in time t_{fac} and $\mathsf{Adv}^{\mathsf{fac}}_{\mathsf{IGen},\mathsf{F}}(k) \geq \epsilon(k)$. The *factoring assumption* (with respect to IGen) states that $\mathsf{Adv}^{\mathsf{fac}}_{\mathsf{IGen},\mathsf{F}}(k)$ is negligible in k for every PPT F.

The best algorithms currently known for factoring $N = PQ$ of length $\ell_{\mathsf{N}} = \mathsf{bitlength}(N) = \log N$ have (heuristic) running time

$$L_N\big(1/3, (64/9)^{1/3}\big) = e^{1.92\ell_{\mathsf{N}}{}^{1/3+o(1)}(\log \ell_{\mathsf{N}})^{2/3}}.$$

(See, e.g., [33].) Therefore, if we want k bits of security, we need to choose the function $\ell_{\mathsf{N}}(k)$ such that the above term is lower-bounded by 2^k. As an example, one commonly uses $\ell_{\mathsf{N}}(80) = 1024$.

3.2. *Quadratic Residues*

The group $\mathbb{Z}_N^*$ consists of all elements of $\mathbb{Z}_N$ that have an inverse modulo N. $\mathbb{Z}_N^*$ has order $\phi(N) = (P-1)(Q-1)$, where $\phi(N)$ is Euler's totient function. By $\mathbb{J}_N$ we denote the subgroup of all elements from $\mathbb{Z}_N^*$ with Jacobi symbol 1. $\mathbb{J}_N$ has index 2 in $\mathbb{Z}_N^*$ and has order $(P-1)(Q-1)/2$. Since N is Blum, $-1 \in \mathbb{J}_N$. The set $\mathbb{QR}_N \subseteq \mathbb{Z}_N^*$ of *quadratic residues modulo N* is defined as $\mathbb{QR}_N := \{x \in \mathbb{Z}_N^* : \exists y \in \mathbb{Z}_N^*$ with $y^2 = x \bmod N\}$. Since $\mathbb{Z}_N^* \cong \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_{pq}$, $\mathbb{QR}_N$ is a cyclic group of order pq. Note that this implies that a uniformly chosen element of $\mathbb{QR}_N$ is a generator (of $\mathbb{QR}_N$) with overwhelming probability. Computations in $\mathbb{QR}_N$ are computations modulo N. If it is implied by context, we omit writing explicitly "$\bmod N$" for calculations modulo N. Note that $\mathbb{QR}_N$ is a subgroup of $\mathbb{J}_N$ with index 2 and has order $(P-1)(Q-1)/4$. We remark that distinguishing random $\mathbb{QR}_N$-elements from random $\mathbb{J}_N$-elements is generally believed to be a hard problem (the quadratic residuosity problem).

3.3. *Signed Quadratic Residues*

For $x \in \mathbb{Z}_N$ we define $|x|$ as the absolute value of x, where x is represented as a signed integer in the set $\{-(N-1)/2, \ldots, (N-1)/2\}$. We define the group of signed quadratic residues as

$$\mathbb{QR}_N^+ := \{|x| : x \in \mathbb{QR}_N\},$$

where the group operation $\circ$ in $\mathbb{QR}_N^+$ is defined through $|x| \circ |y| := |xy|$. Henceforth, all computations will take place in $\mathbb{QR}_N^+$, and hence we will omit the absolute values from the notation and simply write xy or $x \cdot y$ for $x \circ y$. Note that taking the absolute value is a surjective homomorphism from $\mathbb{QR}_N$ to $\mathbb{QR}_N^+$ with trivial kernel. (This is since N is a Blum integer and hence $-1 \notin \mathbb{QR}_N$.) The following basic facts have already been noted in earlier works such as [1,19,24].

Lemma 1. *Let N be a Blum integer. Then*:

1. $(\mathbb{QR}_N^+, \circ)$ *is a group of order* $\phi(N)/4$.
2. *If we let* $\mathbb{J}_N^+ := \{|x| : x \in \mathbb{J}_N\}$, *then* $\mathbb{J}_N^+ = \mathbb{QR}_N^+$. *In particular,* $\mathbb{QR}_N^+$ *is efficiently recognizable* (*given only* N).
3. *If* $\mathbb{QR}_N$ *is cyclic, so is* $\mathbb{QR}_N^+$.

Proof. First, note that $|\cdot| : (\mathbb{Z}_N, \cdot) \to (\mathbb{Z}_N^+, \circ)$ is a group homomorphism so $(\mathbb{QR}_N^+, \circ)$ is a group. Since $-1 \notin \mathbb{QR}_N$, the map $\mathbb{QR}_N \to \mathbb{QR}_N^+$ has kernel $\{1\}$, and so $\mathrm{ord}(\mathbb{QR}_N^+) = \mathrm{ord}(\mathbb{QR}_N) = \phi(N)/4$. On the other hand, the map $\mathbb{J}_N \to \mathbb{J}_N^+$ has kernel $\{\pm 1\}$, and so $\mathrm{ord}(\mathbb{J}_N^+) = \mathrm{ord}(\mathbb{J}_N)/2 = \phi(N)/4$. Since $\mathbb{QR}_N \subseteq \mathbb{J}_N$, we have $\mathbb{QR}_N^+ \subseteq \mathbb{J}_N^+$, so $\mathrm{ord}(\mathbb{QR}_N^+) = \mathrm{ord}(\mathbb{J}_N^+)$ implies $\mathbb{QR}_N^+ = \mathbb{J}_N^+$. Elements in $\mathbb{QR}_N^+$ can be efficiently recognized since $\mathbb{QR}_N^+ = \mathbb{J}_N^+ = \mathbb{J}_N \cap [(N-1)/2]$. If $\mathbb{QR}_N$ is cyclic, a generator g of $\mathbb{QR}_N$ is mapped to a generator $|g|$ of $\mathbb{QR}_N^+$, so $\mathbb{QR}_N^+$ is a cyclic group. □

4. Chosen-Ciphertext Security from Factoring

4.1. *The Scheme*

In this section, we will present our KEM construction.[4] We will make use of two building blocks: a target-collision resistant hash function, and the Blum–Blum–Shub (BBS) pseudorandom number generator [7].

Concretely, for a Blum integer $N = PQ$ and $u \in \mathbb{Z}_N$, we establish the following notation: $\mathsf{LSB}_N(u) = u \bmod 2$ the least significant bit of u, where u is interpreted as a signed integer with $-(N-1)/2 \le u \le (N-1)/2$. Furthermore, let

$$\mathsf{BBS}_N(u) = \left(\mathsf{LSB}_N(u), \mathsf{LSB}_N\left(u^2\right), \ldots, \mathsf{LSB}_N\left(u^{2^{\ell_K - 1}}\right)\right) \in \{0,1\}^{\ell_K}$$

[4] Compared to the construction from the conference version [28], we work in the group of signed quadratic residues. Since the signed quadratic residues are efficiently recognizable, no extra protection against trivial malleability attacks (such as sign-flipping attacks) need to be added to our schemes.

denote the BBS generator applied to $u \in \mathbb{QR}_N^+$ and modulo N.[5] We stress that we use the BBS generator in the group $\mathbb{QR}_N^+$; this does not affect its security [19].

Furthermore, for N as above, let $\mathsf{T} : \mathbb{QR}_N^+ \to \{1, \ldots, 2^{\ell_\mathsf{T}} - 1\}$ be a target-collision resistant hash function.

The Scheme We are ready to define the following key encapsulation mechanism $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

Key Generation. $\mathsf{Gen}(1^k)$ chooses uniformly at random

- a modulus $N = PQ = (2p+1)(2q+1)$ (using $\mathsf{IGen}(1^k)$, cf. Sect. 3.1),
- a signed quadratic residue $g \in \mathbb{QR}_N^+$,
- an exponent $\alpha \in [(N-1)/4]$.

Gen then sets $X = g^{\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}$ and outputs a public key pk and a secret key sk, where

$$pk = (N, g, X), \qquad sk = (N, g, \alpha).$$

Encapsulation. $\mathsf{Enc}(pk)$ chooses uniformly $r \in [(N-1)/4]$, sets

$$R = g^{r 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}, \qquad t = \mathsf{T}(R) \in \left\{1, \ldots, 2^{\ell_\mathsf{T}} - 1\right\}, \qquad S = \left(g^t X\right)^r$$

and outputs the key $K = \mathsf{BBS}_N(g^{r 2^{\ell_\mathsf{T}}}) \in \{0,1\}^{\ell_\mathsf{K}}$ and the ciphertext $C = (R, S) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+$.

Decapsulation. $\mathsf{Dec}(sk, (R, S))$ verifies that $(R, S) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+$ and rejects if not. Then, Dec computes $t = \mathsf{T}(R) \in \{1, \ldots, 2^{\ell_\mathsf{T}} - 1\}$, checks whether

$$S^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \stackrel{?}{=} R^{t+\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \tag{1}$$

holds, and rejects if not. If (1) holds, Dec computes $a, b, c \in \mathbb{Z}$ such that

$$2^c = \gcd\left(t, 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}\right) = at + b 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}. \tag{2}$$

Note that $c < \ell_\mathsf{T}$ since $0 < t < 2^{\ell_\mathsf{T}}$. Then, Dec derives

$$T = \left(S^a \cdot R^{b-a\alpha}\right)^{2^{\ell_\mathsf{T}-c}} \tag{3}$$

and from this $K = \mathsf{BBS}_N(T) \in \{0,1\}^{\ell_\mathsf{K}}$, which is the output.

We remark that decapsulation (or, rather, generation of the secret keys) does not require knowledge about the factorization of N. Indeed, the modulus N as well as the generator g can be viewed as global system parameter shared by many parties. Then pk only contains the value $X \in \mathbb{QR}_N^+$ and sk only contains $\alpha \in [(N-1)/4]$.

Our scheme uses an RSA modulus N that consists of safe primes. In Sect. 6 we show how to avoid this assumption and allow N to be an arbitrary Blum integer.

[5] For efficiency, and at the price of a worse reduction, one can even simultaneously extract $\lceil \log_2 \log_2 N \rceil$ bits of each u^{2^i} instead of only the least significant bit [1,19]. However, our analysis treats the original BBS generator for simplicity.

Correctness The correctness of the scheme might not be obvious, so we prove it here. Fix a public key pk and a secret key sk as produced by $\mathsf{Gen}(1^k)$, and assume that (R, S) is a ciphertext for a key K as generated by $\mathsf{Enc}(pk)$. We have to show that $\mathsf{Dec}(sk, (R, S))$ outputs K. First, it is clear that $(R, S) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+$. Also,

$$S^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} = \left(\left(g^t X\right)^r\right)^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} = g^{(t+\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}})r2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \overset{(*)}{=} R^{t+\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}$$

(where $(*)$ uses $R = g^{r2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}$), so (1) holds. Hence, (R, S) is not rejected by Dec. Now (1) implies

$$S = R^{\frac{t+\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}} = R^{\frac{t}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}+\alpha}, \tag{4}$$

where the division in the exponent is computed modulo $pq = |\mathbb{QR}_N| = |\mathbb{QR}_N^+|$. This gives

$$\begin{aligned} T &\overset{(3)}{=} \left(S^a \cdot R^{b-a\alpha}\right)^{2^{\ell_\mathsf{T}-c}} = \left(\left(SR^{-\alpha}\right)^a \cdot R^b\right)^{2^{\ell_\mathsf{T}-c}} \overset{(4)}{=} \left(\left(R^{\frac{t}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}}\right)^a \cdot R^b\right)^{2^{\ell_\mathsf{T}-c}} \\ &= \left(R^{\frac{at+b2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}}\right)^{2^{\ell_\mathsf{T}-c}} \overset{(2)}{=} R^{\frac{2^c}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}\cdot 2^{\ell_\mathsf{T}-c}} = R^{\frac{1}{2^{\ell_\mathsf{K}}}} \overset{(*)}{=} g^{r2^{\ell_\mathsf{T}}}, \end{aligned} \tag{5}$$

where, again, $(*)$ uses $R = g^{r2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}$. But (5) shows that Dec outputs $\mathsf{BBS}_N(T) = \mathsf{BBS}_N(g^{r2^{\ell_\mathsf{T}}}) = K$ as desired.

Theorem 1 (IND-CCA Security of KEM). *Assume* T *is a target-collision resistant hash function and the factoring assumption holds. Then* KEM *is* IND-CCA *secure in the sense of Definition* 1.

The proof of Theorem 1 will be given in Sect. 5.

Efficiency We claim that, with some trivial optimizations, encapsulation uses roughly two exponentiations, and decapsulation roughly one exponentiation. Namely, encapsulation can first compute $A = g^r$ and $B = X^r$, which are two full exponentiations. Then, the remaining computations require only multiplications or exponentiations with very small exponents: $K = \mathsf{BBS}_N(A^{2^{\ell_\mathsf{T}}})$, $R = A^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}$, and $S = A^t B$. (In fact, R is a by-product of computing K.) Similarly, decapsulation can first compute $D = S/R^\alpha$, which requires one full exponentiation. From D, (1) can be checked with $D^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \overset{?}{=} R^t$, which requires only two exponentiations with very small exponents. The key K can then be computed as $\mathsf{BBS}_N(T)$ for $T = (R^b D^a)^{2^{\ell_\mathsf{T}-c}}$, which requires three exponentiations with small exponents (note that the bit-length of a and b is at most $\ell_\mathsf{K} + \ell_\mathsf{T}$).

For concreteness let us assume that one regular exponentiation with an exponent of length ℓ requires $1.5 \cdot \ell$ modular multiplications and that one squaring takes the same time as one multiplication. Let us further assume that $\ell_\mathsf{N} := \mathsf{bitlength}(N) = 1024$ and $\ell_\mathsf{K} = \ell_\mathsf{T} = 80$. Then encapsulation requires $3\ell_\mathsf{N} + \ell_\mathsf{K} + 2.5\ell_\mathsf{T} = 3352$ multiplications; decapsulation requires $1.5\ell_\mathsf{N} + 4\ell_\mathsf{K} + 6.5\ell_\mathsf{T} = 2376$ multiplications.

We remark that, by adding the prime factors P and Q to the secret key, we can further improve the scheme's efficiency. For example, using Chinese Remainder Theorem will speed up decapsulation by a factor between 3 and 4.

5. Proof of Security

We split up the proof of Theorem 1 into two parts:

- We first recall that the BBS generator is pseudorandom if factoring Blum integers is hard. This holds even if the modulus N and the 2^{ℓ_K}th power $u^{2^{\ell_K}}$ of the BBS seed u are published, as is the case in our KEM. (Theorem 2.)
- We then prove that KEM is IND-CCA secure under the assumption that the BBS generator is pseudorandom and the employed hash function is target-collision resistant. This reduction is the heart of our proof. (Theorem 3.)

Combining both parts yields Theorem 1.

We start by recalling that the BBS generator is pseudorandom, in the following sense.

Definition 4 (PRNG Experiment for BBS Generator). For an algorithm D, define

$$\mathsf{Adv}_{\mathsf{D}}^{\mathsf{BBS}}(k) = \Pr\big[\mathsf{D}\big(N, z, \mathsf{BBS}_N(u)\big) = 1\big] - \Pr\big[\mathsf{D}(N, z, U_{\{0,1\}^{\ell_K}}) = 1\big],$$

where

- $N \in \mathbb{N}$ is distributed as $\mathsf{IGen}(1^k)$,
- $u \in \mathbb{QR}_N^+$ is uniformly chosen, and $z = u^{2^{\ell_K}}$,
- $U_{\{0,1\}^{\ell_K}} \in \{0,1\}^{\ell_K}$ is independently and uniformly chosen.

We say that D (t, ϵ)-*breaks* BBS if D's running time is at most $t = t(k)$ and $\mathsf{Adv}_{\mathsf{D}}^{\mathsf{BBS}}(k) \geq \epsilon = \epsilon(k)$.

Concretely, any BBS-distinguisher can be used to factor Blum integers. This result has already been used in the Blum–Goldwasser scheme [5].

Theorem 2 (BBS-Distinguisher $\Rightarrow$ Factoring Algorithm [1,6,7,19]). *For every algorithm* D *that* $(t_{\mathsf{BBS}}, \epsilon_{\mathsf{BBS}})$*-breaks* BBS, *there is an algorithm* F *that* $(t_{\mathsf{fac}}, \epsilon_{\mathsf{fac}})$*-factors Blum integers, where*

$$t_{\mathsf{fac}} \approx {\ell_K}^4 t_{\mathsf{BBS}}/\epsilon_{\mathsf{BBS}}^2, \qquad \epsilon_{\mathsf{fac}} = \epsilon_{\mathsf{BBS}}/\ell_K.$$

Proof. Let D be an algorithm that $(t_{\mathsf{BBS}}, \epsilon_{\mathsf{BBS}})$-breaks BBS. That is, D distinguishes pseudorandom from truly random bitstrings. First, using a (by now standard) hybrid argument, [6] show that then D can be converted to an algorithm that distinguishes individual pseudorandom bits from truly random bits. Concretely, D gives rise to an algorithm D′ that $(t_{\mathsf{LSB}}, \epsilon_{\mathsf{LSB}})$-distinguishes tuples $(N, u^2, \mathsf{LSB}(u))$ from tuples $(N, u^2, U_{\{0,1\}})$, where $u \in \mathbb{QR}_N^+$ and $U_{\{0,1\}} \in \{0,1\}$ are uniformly chosen, $t_{\mathsf{LSB}} \approx t_{\mathsf{BBS}}$, and $\epsilon_{\mathsf{LSB}} = \epsilon_{\mathsf{BBS}}/\ell_K$.

Next, we can use that the least significant bit is a hard-core bit of the squaring function modulo N. In particular, any algorithm that, given N and u^2, successfully distinguishes $\mathsf{LSB}(u)$ from a random bit, can be used to recover a square root of u^2. This in turn yields a non-trivial factor of N with significant probability. Concretely, building on [1],

Fischlin and Schnorr [19] show how to transform D' into an algorithm F that $(t_{\mathsf{fac}}, \epsilon_{\mathsf{fac}})$-factors Blum integers, where $t_{\mathsf{fac}} \approx {\ell_{\mathsf{K}}}^2 t_{\mathsf{LSB}}/\epsilon_{\mathsf{LSB}}^2 \approx {\ell_{\mathsf{K}}}^4 t_{\mathsf{BBS}}/\epsilon_{\mathsf{BBS}}^2$ and $\epsilon_{\mathsf{fac}} = \epsilon_{\mathsf{LSB}} = \epsilon_{\mathsf{BBS}}/\ell_{\mathsf{K}}$. (We use the quantitative interpretation [36, Theorem 6.1] of the results from [19] here.) The claim follows. □

The following theorem contains the heart of our proof, namely, a simulation that shows that any successful IND-CCA adversary on KEM implies a successful BBS-distinguisher (and hence, using Theorem 2, can be used to factor Blum integers).

Theorem 3 (IND-CCA Adversary ⇒ BBS-Distinguisher). *For every adversary* A *that* $(t_{\mathsf{KEM}}, \epsilon_{\mathsf{KEM}})$*-breaks* KEM*'s* IND-CCA *property, there exists an algorithm* D *that* $(t_{\mathsf{BBS}}, \epsilon_{\mathsf{BBS}})$*-breaks* BBS *and an adversary* B *that* $(t_{\mathsf{T}}, \epsilon_{\mathsf{T}})$*-breaks* T*, such that*

$$t_{\mathsf{BBS}} \approx t_{\mathsf{T}} \approx t_{\mathsf{KEM}}, \qquad \epsilon_{\mathsf{BBS}} + \epsilon_{\mathsf{T}} + 2^{-k+3} \geq \epsilon_{\mathsf{KEM}}.$$

Proof (Setting up the Variables for Simulation). Assume an adversary A on KEM's IND-CCA security. We define a BBS-distinguisher D, which acts on input (N, z, V) as follows. D first uniformly selects a signed quadratic residue $g \in \mathbb{QR}_N^+$, as well as exponent $\beta \in [(N-1)/4]$, and sets

$$R^* = z, \qquad t^* = \mathsf{T}\big(R^*\big) \in \big\{1, \ldots, 2^{\ell_{\mathsf{T}}} - 1\big\}, \qquad X = g^{\beta 2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}} - t^*}.$$

The public key used in the simulation is $pk = (N, g, X)$. It will be convenient to write $X = g^{\alpha 2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}}$ as in Gen, for $\alpha = \beta - t^*/2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}$ unknown to D. (Here and in the following, a division of exponents is computed modulo pq, the order of $\mathbb{QR}_N^+$.) Furthermore, in the following, we will silently assume that g generates $\mathbb{QR}_N^+$, which is very likely, but not guaranteed. A rigorous justification that takes into account error probabilities follows below.

Preparation of Challenge Ciphertext and Key To complete the definition of the challenge ciphertext $C^* = (R^*, S^*)$, write $R^* = g^{r^* 2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}}$. Since we assumed that g is a generator, this is possible, but of course r^* is unknown. D defines

$$S^* = R^{*\beta} \quad \big(= g^{r^* \beta 2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}} = \big(g^{t^*} X\big)^{r^*}\big) \tag{6}$$

as Enc would have computed. The (real) corresponding key K^* is defined as

$$K^* = \mathsf{BBS}_N\big(g^{2^{\ell_{\mathsf{T}}} r^*}\big) = \mathsf{BBS}_N\big(R^{*\frac{1}{2^{\ell_{\mathsf{K}}}}}\big) = \mathsf{BBS}_N\big(z^{\frac{1}{2^{\ell_{\mathsf{K}}}}}\big) = \mathsf{BBS}_N(u). \tag{7}$$

D then invokes A with public key $pk = (N, g, X)$, challenge ciphertext $C^* = (R^*, S^*)$, and challenge key V. Note that V is either the real challenge key $\mathsf{BBS}_N(u)$, or it is a uniform string.

On the Distribution of Simulated Public Key and Challenge Ciphertext We claim that the distribution of public key pk and challenge ciphertext C^* is almost identical in simulation and IND-CCA experiment. Concretely, we postpone the straightforward but somewhat tedious proof of the following lemma until after the description of our simulation.

Lemma 2. *There exists an event* $\mathsf{bad}_{\mathsf{key}}$ *such that, conditioned on* $\neg\mathsf{bad}_{\mathsf{key}}$, *public key pk and challenge ciphertext* C^* *are identically distributed in simulation and* IND-CCA *experiment. Also,* $\neg\mathsf{bad}_{\mathsf{key}}$ *implies that g is a generator. We have*

$$\Pr[\mathsf{bad}_{\mathsf{key}}] \leq 2^{-k+3} \tag{8}$$

both in the simulation and in the IND-CCA *experiment.*

Thus, conditioned on $\neg\mathsf{bad}_{\mathsf{key}}$, D perfectly simulates A's input as in the real IND-CCA experiment if $V = \mathsf{BBS}_N(u) = \mathsf{BBS}_N(z^{1/2^{\ell_\mathsf{K}}})$, and as in the ideal IND-CCA experiment if V is random.

How to Handle A*'s Decryption Queries* It remains to describe how D handles decryption queries of A as in the IND-CCA experiment. So say that A submits a ciphertext (R, S) for decryption. We may assume that $(R, S) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+$ since $\mathbb{QR}_N^+$ is efficiently recognizable. Let $t = \mathsf{T}(R) \in \{1, \ldots, 2^{\ell_\mathsf{T}} - 1\}$. We call a ciphertext *consistent* iff the original decryption algorithm would not have rejected it. Hence, by (1), a ciphertext is consistent iff

$$S^{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \stackrel{?}{=} R^{t-t^*+\beta 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \quad \big(= R^{t+\alpha 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}\big). \tag{9}$$

By our setup of variables, D can check (9) by itself, and hence detect and reject inconsistent ciphertexts.

How to Decrypt Consistent Ciphertexts Now assume that C is consistent and $t \neq t^*$. Then, (4) and (5) follow (except for the deduction $(*)$) just as in the correctness proof, and we get

$$T = R^{\frac{1}{2^{\ell_\mathsf{K}}}} \tag{10}$$

for the raw key T that would have been computed by Dec. We will now show how D can compute T. Namely, D computes $a', b', c' \in \mathbb{Z}$ such that

$$2^{c'} = \gcd\big(t - t^*, 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}\big) = a'\big(t - t^*\big) + b' 2^{\ell_\mathsf{K}+\ell_\mathsf{T}}. \tag{11}$$

Since $1 \leq t, t^* < 2^{\ell_\mathsf{T}}$ and $t \neq t^*$, we have $c' < \ell_\mathsf{T}$. Similarly to (4) and (5), (9) implies

$$S = R^{\frac{t-t^*}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}+\beta}, \tag{12}$$

from (9), and from this,

$$\begin{aligned}\big(S^{a'} \cdot R^{b'-a'\beta}\big)^{2^{\ell_\mathsf{T}-c'}} &= \big(\big(SR^{-\beta}\big)^{a'} \cdot R^{b'}\big)^{2^{\ell_\mathsf{T}-c'}} \stackrel{(12)}{=} \Big(\big(R^{\frac{t-t^*}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}}\big)^{a'} \cdot R^{b'}\Big)^{2^{\ell_\mathsf{T}-c'}} \\ &= \Big(R^{\frac{a'(t-t^*)+b'2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}}}\Big)^{2^{\ell_\mathsf{T}-c'}} \stackrel{(11)}{=} R^{\frac{2^{c'}}{2^{\ell_\mathsf{K}+\ell_\mathsf{T}}} \cdot 2^{\ell_\mathsf{T}-c'}} = R^{\frac{1}{2^{\ell_\mathsf{K}}}} \stackrel{(10)}{=} T.\end{aligned} \tag{13}$$

Note that from T, the final decryption key can be computed as $K = \mathsf{BBS}_N(T)$. Hence, using (13), D can correctly decrypt every consistent ciphertext with $t \neq t^*$.

The Case $t = t^*$ Let us turn to the case that $t = t^*$ and the ciphertext is consistent. Then, if $R = R^*$ holds, we have

$$S \overset{(9)}{=} R^{\frac{t-t^*}{2^{\ell_K+\ell_T}}+\beta} \overset{(*)}{=} R^{*\beta} = S^*, \tag{14}$$

where in $(*)$ we use $R = R^*$ and $t = t^*$. Since A is not allowed to query $(R, S) = (R^*, S^*)$ for decryption, we may hence assume that $R \neq R^*$.

But if $\mathsf{T}(R) = t = t^* = \mathsf{T}(R^*)$ and $R \neq R^*$, then A has broken the target-collision resistance of T. Formally, let $\mathsf{bad}_{\mathsf{TCR}}$ denote the event that $t = t^*$ and $R \neq R^*$. If $\mathsf{bad}_{\mathsf{TCR}}$ occurs, D can safely give up, since

$$\Pr[\mathsf{bad}_{\mathsf{TCR}}] \leq \mathsf{Adv}^{\mathsf{TCR}}_{\mathsf{T},\mathsf{B}}(k) \tag{15}$$

for a suitable PPT adversary B on T that simulates D and A.

Summary of the Decryption Procedure We summarize the decryption cases:

- inconsistent (R, S) (consistency check (9)$\Leftrightarrow$(1) not passed): reject,
- consistent (R, S) and $t \neq t^*$: decrypt using (13),
- consistent (R, S), $t = t^*$, and $R = R^*$: cannot happen since then $(R, S) = (R^*, S^*)$ by (14),
- consistent (R, S), $t = t^*$, and $R \neq R^*$: give up simulation (A has found a T-collision).

Hence, also decryption is faithfully simulated unless $\mathsf{bad}_{\mathsf{TCR}}$ occurs.

Finishing the Proof We conclude that, unless $\mathsf{bad}_{\mathsf{TCR}}$ or $\mathsf{bad}_{\mathsf{key}}$ occurs, D perfectly simulates the real IND-CCA experiment upon input $V = \mathsf{BBS}_N(u)$, and the ideal IND-CCA experiment if V is random. If we let D output whatever the simulated experiment outputs, we obtain:

$$\begin{aligned}
&\left|\Pr\left[\mathsf{D}\left(N, z, \mathsf{BBS}_N(u)\right) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right]\right| \leq \Pr[\mathsf{bad}_{\mathsf{TCR}}] + \Pr[\mathsf{bad}_{\mathsf{key}}],\\
&\left|\Pr\left[\mathsf{D}(N, z, U_{\{0,1\}^{\ell_K}}) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right]\right| \leq \Pr[\mathsf{bad}_{\mathsf{TCR}}] + \Pr[\mathsf{bad}_{\mathsf{key}}].
\end{aligned} \tag{16}$$

Using (8) and (15), Theorem 3 follows from (16).

It remains to prove Lemma 2.

Proof of Lemma 2 Observe that pk and C^* are distributed slightly differently in the IND-CCA experiment (i.e., as generated by Gen and Enc) and in the simulation:

- $R^* = g^{r^* 2^{\ell_K+\ell_T}}$ for uniform (hidden) $r^* \in [(N-1)/4]$ in the experiment, while $R^* \in \mathbb{QR}^+_N$ is a uniform group element in the simulation.
- $X = g^{\alpha 2^{\ell_K+\ell_T}}$ for uniform (hidden) $\alpha \in [(N-1)/4]$ in the experiment, while $X = g^{\beta 2^{\ell_K+\ell_T} - t^*}$ for uniform (hidden) $\beta \in [(N-1)/4]$ in the simulation.

However, conditioned on the following event $\mathsf{good}_{\mathsf{key}}$:

(in the experiment:) g is a generator, and $r^*, \alpha \leq |\mathbb{QR}_N^+|$,
(in the simulation:) g is a generator, and $\beta \leq |\mathbb{QR}_N^+|$,

pk and C^* are distributed identically in experiment and simulation: $\mathsf{good}_{\mathsf{key}}$ implies that N, g, X, and R^* are uniformly and independently chosen over their respective domains, and S^* follows deterministically from pk and R^* according to (7). Hence we only need to bound the probability of $\mathsf{bad}_{\mathsf{key}} := \neg\mathsf{good}_{\mathsf{key}}$. Since $|\mathbb{QR}_N^+| = |\mathbb{QR}_N| = pq$ and we assumed that p and q are $\ell_{\mathsf{N}}/2$-bit primes (for $\ell_{\mathsf{N}}/2 \geq k$), a uniform $\mathbb{QR}_N$-element is a generator except with probability $(p+q-1)/pq \leq 2^{-n/2+2}$. Furthermore, $(N-1)/4$ is a close approximation of the group order $|\mathbb{QR}_N^+| = pq = (N-1)/4 - (p+q)/2$, so that, e.g., $r^* \leq |\mathbb{QR}_N^+|$ except with probability $2(p+q)/(N-1) \leq 2^{-\ell_{\mathsf{N}}/2+1}$. Hence,

$$\begin{aligned}\mathsf{Pr}[\mathsf{bad}_{\mathsf{key}}] &\leq \max\left\{2^{-\ell_{\mathsf{N}}/2+2} + 2\cdot 2^{-\ell_{\mathsf{N}}/2+1}, 2^{-\ell_{\mathsf{N}}/2+2} + 2^{-\ell_{\mathsf{N}}/2+1}\right\} \\ &= 2^{-\ell_{\mathsf{N}}/2+3} \\ &\overset{n/2\geq k}{\leq} 2^{-k+3}\end{aligned}$$

both in the experiment and in the simulation. □

6. Avoiding Safe Primes

In our KEM, we assume that $N = PQ$ is composed of two safe primes (i.e., primes of the form $P = 2p+1$ for prime p). We can drop this assumption and allow arbitrary Blum integers N, if we employ a Goldreich–Levin [22] based pseudorandom generator instead of the Blum–Blum–Shub generator. Namely, all we actually need to prove that KEM is IND-CCA is that

$$\left(N, g, g^{r^{2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}}}, \mathsf{Ext}_{pk}\left(g^{r^{2^{\ell_{\mathsf{T}}}}}\right)\right) \overset{\mathrm{c}}{\approx} \left(N, g, g^{r^{2^{\ell_{\mathsf{K}}+\ell_{\mathsf{T}}}}}, U_{\{0,1\}^{\ell_{\mathsf{K}}}}\right), \tag{17}$$

where $\overset{\mathrm{c}}{\approx}$ denotes computational indistinguishability, N is a Blum integer, $g \in \mathbb{QR}_N^+$, $r \in [N/4]$, and $U_{\{0,1\}^{\ell_{\mathsf{K}}}} \in \{0,1\}^{\ell_{\mathsf{K}}}$ are uniform, and Ext is a suitable randomness extractor. In our original description of KEM, we have $\mathsf{Ext}_{pk}(u) = \mathsf{BBS}_N(u)$. In that case, we only know that the hardness of factoring N implies (17) if $u = g^{r^{2^{\ell_{\mathsf{T}}}}}$ is a *uniform* element of $\mathbb{QR}_N^+$ (which is the case when $N = PQ$ for safe primes P, Q, since then g is a generator with high probability). But if g is not a generator at least with high probability, then u may not be uniformly distributed.

Now suppose we set

$$\mathsf{Ext}_{pk}(u) = \left(\mathsf{GL}_s(u), \mathsf{GL}_s\left(u^2\right), \ldots, \mathsf{GL}_s\left(u^{2^{\ell_{\mathsf{K}}-1}}\right)\right) \in \{0,1\}^{\ell_{\mathsf{K}}}$$

for the Goldreich–Levin predicate GL_s that maps u to the bitwise inner product of s and u. Then a hybrid argument and the hard-core property of GL_s show that (17) is implied by the hardness of computing u with $u^2 = v \mod N$ from (N, g, v) (with

$v = g^r$). But any algorithm B that computes such a u from (N, g, v) can be used to factor N. Namely, given N, choose uniformly $h \in \mathbb{Z}_N$ and $\tilde{r} \in [N/4]$, and set $g = h^2$ and $v = g^{2\tilde{r}+1}$. (Observe that v is almost uniformly distributed over $\langle g \rangle$, since N is a Blum integer.) Then, invoke $B(N, g, v)$ to obtain a square root u of v. We can then compute a square root of g as $\tilde{h} = u^a g^b$ (for $a, b \in \mathbb{Z}$ with $a(2\tilde{r}+1) + 2b = \gcd(2\tilde{r}+1, 2) = 1$). With probability $1/2$, then $\gcd(h - \tilde{h}, N)$ yields a non-trivial factor of N. Hence (17) is implied by the hardness of factoring arbitrary Blum integers, and our KEM (instantiated with the Goldreich–Levin predicate) is IND-CCA secure. The price to pay is that we need to place a seed $s \in \{0,1\}^{\ell_N}$ for the Goldreich–Levin hard-core function in the public key. (However, note that s can be made a global system parameter, like N and g.)

Acknowledgements

We would like to thank Ronald Cramer and Ivan Damgård for interesting discussions.

References

[1] W. Alexi, B. Chor, O. Goldreich, C.-P. Schnorr, RSA and Rabin functions: certain parts are as hard as the whole. *SIAM J. Comput.* **17**(2), 194–209 (1988)

[2] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in *ACM CCS 93: 1st Conference on Computer and Communications Security*, ed. by V. Ashby (ACM, New York, 1993), pp. 62–73

[3] M. Bellare, P. Rogaway, Optimal asymmetric encryption, in *Advances in Cryptology—EUROCRYPT'94*, ed. by A. De Santis. Lecture Notes in Computer Science, vol. 950 (Springer, Berlin, 1994), pp. 92–111

[4] D. Bleichenbacher, Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1, in *Advances in Cryptology—CRYPTO'98*, ed. by H. Krawczyk. Lecture Notes in Computer Science, vol. 1462 (Springer, Berlin, 1998), pp. 1–12

[5] M. Blum, S. Goldwasser, An efficient probabilistic public-key encryption scheme which hides all partial information, in *Advances in Cryptology—CRYPTO'84*, ed. by G.R. Blakley, D. Chaum. Lecture Notes in Computer Science, vol. 196 (Springer, Berlin, 1985), pp. 289–302

[6] M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Comput.* **13**(4), 850–864 (1984)

[7] L. Blum, M. Blum, M. Shub, A simple unpredictable pseudo-random number generator. *SIAM J. Comput.* **15**(2), 364–383 (1986)

[8] D. Boneh, X. Boyen, Efficient selective-ID secure identity based encryption without random oracles, in *Advances in Cryptology—EUROCRYPT 2004*, ed. by C. Cachin, J. Camenisch. Lecture Notes in Computer Science, vol. 3027 (Springer, Berlin, 2004), pp. 223–238

[9] D. Boneh, R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.* **36**(5), 1301–1328 (2007)

[10] X. Boyen, Q. Mei, B. Waters, Direct chosen ciphertext security from identity-based techniques, in *ACM CCS 05: 12th Conference on Computer and Communications Security*, ed. by V. Atluri, C. Meadows, A. Juels (ACM, New York, 2005), pp. 320–329

[11] J. Camenisch, V. Shoup, Practical verifiable encryption and decryption of discrete logarithms, in *Advances in Cryptology—CRYPTO 2003*, ed. by D. Boneh. Lecture Notes in Computer Science, vol. 2729 (Springer, Berlin, 2003), pp. 126–144

[12] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited. *J. ACM* **51**(4), 557–594 (2004)

[13] D. Cash, E. Kiltz, V. Shoup, The twin Diffie–Hellman problem and applications, in *Advances in Cryptology—EUROCRYPT 2008*, ed. by N.P. Smart. Lecture Notes in Computer Science, vol. 4965 (Springer, Berlin, 2008), pp. 127–145

[14] D. Cash, E. Kiltz, V. Shoup, The twin Diffie–Hellman problem and applications. *J. Cryptol.* **22**(4), 470–504 (2009)
[15] R. Cramer, V. Shoup, Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, in *Advances in Cryptology—EUROCRYPT 2002*, ed. by L.R. Knudsen. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 45–64
[16] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.* **33**(1), 167–226 (2003)
[17] R. Cramer, D. Hofheinz, E. Kiltz, A twist on the Naor–Yung paradigm and its application to efficient CCA-secure encryption from hard search problems, in *TCC 2010: 7th Theory of Cryptography Conference*, ed. by D. Micciancio. Lecture Notes in Computer Science, vol. 5978 (Springer, Berlin, 2010), pp. 146–164
[18] D. Dolev, C. Dwork, M. Naor, Nonmalleable cryptography. *SIAM J. Comput.* **30**(2), 391–437 (2000)
[19] R. Fischlin, C.-P. Schnorr, Stronger security proofs for RSA and Rabin bits. *J. Cryptol.* **13**(2), 221–244 (2000)
[20] R. Gennaro, Y. Lindell, A framework for password-based authenticated key exchange. *ACM Trans. Inf. Syst. Secur.* **9**(2), 181–234 (2006)
[21] O. Goldreich, Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: the state of the art. Manuscript. Online available at http://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/nizk-tdp.ps, 2009
[22] O. Goldreich, L.A. Levin, A hard-core predicate for all one-way functions, in *21st Annual ACM Symposium on Theory of Computing* (ACM, New York, 1989), pp. 25–32
[23] S. Goldwasser, S. Micali, Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984)
[24] S. Goldwasser, S. Micali, R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.* **17**(2), 281–308 (1988)
[25] G. Hanaoka, K. Kurosawa, Efficient chosen ciphertext secure public key encryption under the computational Diffie–Hellman assumption, in *Advances in Cryptology—ASIACRYPT 2008*, ed. by J. Pieprzyk. Lecture Notes in Computer Science, vol. 5350 (Springer, Berlin, 2008), pp. 308–325
[26] K. Haralambiev, T. Jager, E. Kiltz, V. Shoup, Simple and efficient public-key encryption from computational Diffie–Hellman in the standard model, in *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, ed. by P.Q. Nguyen, D. Pointcheval. Lecture Notes in Computer Science, vol. 6056 (Springer, Berlin, 2010), pp. 1–18
[27] D. Hofheinz, E. Kiltz, Secure hybrid encryption from weakened key encapsulation, in *Advances in Cryptology—CRYPTO 2007*, ed. by A. Menezes. Lecture Notes in Computer Science, vol. 4622 (Springer, Berlin, 2007), pp. 553–571
[28] D. Hofheinz, E. Kiltz, Practical chosen ciphertext secure encryption from factoring, in *Advances in Cryptology—EUROCRYPT 2009*, ed. by A. Joux. Lecture Notes in Computer Science, vol. 5479 (Springer, Berlin, 2009), pp. 313–332
[29] E. Kiltz, Chosen-ciphertext security from tag-based encryption, in *TCC 2006: 3rd Theory of Cryptography Conference*, ed. by S. Halevi, T. Rabin. Lecture Notes in Computer Science, vol. 3876 (Springer, Berlin, 2006), pp. 581–600
[30] E. Kiltz, Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie–Hellman, in *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, ed. by T. Okamoto, X. Wang. Lecture Notes in Computer Science, vol. 4450 (Springer, Berlin, 2007), pp. 282–297
[31] E. Kiltz, K. Pietrzak, M. Stam, M. Yung, A new randomness extraction paradigm for hybrid encryption, in *Advances in Cryptology—EUROCRYPT 2009*, ed. by A. Joux. Lecture Notes in Computer Science, vol. 5479 (Springer, Berlin, 2009), pp. 590–609
[32] K. Kurosawa, Y. Desmedt, A new paradigm of hybrid encryption scheme, in *Advances in Cryptology—CRYPTO 2004*, ed. by M. Franklin. Lecture Notes in Computer Science, vol. 3152 (Springer, Berlin, 2004), pp. 426–442
[33] A.K. Lenstra, H.W. Lenstra Jr. (eds.), *The Development of the Number Field Sieve*. Lecture Notes in Mathematics, vol. 1554 (Springer, Berlin, 1993)
[34] S. Lucks, A variant of the Cramer–Shoup cryptosystem for groups of unknown order, in *Advances in Cryptology—ASIACRYPT 2002*, ed. by Y. Zheng. Lecture Notes in Computer Science, vol. 2501 (Springer, Berlin, 2002), pp. 27–45

[35] Q. Mei, B. Li, X. Lu, D. Jia, Chosen ciphertext secure encryption under factoring assumption revisited, in *PKC 2011: 14th International Workshop on Theory and Practice in Public Key Cryptography*, ed. by D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi. Lecture Notes in Computer Science, vol. 6571 (Springer, Berlin, 2011), pp. 210–227

[36] M. Naor, O. Reingold, A. Rosen, Pseudo-random functions and factoring. *SIAM J. Comput.* **31**(5), 1383–1404 (2002)

[37] M. Naor, M. Yung, Universal one-way hash functions and their cryptographic applications, in *21st Annual ACM Symposium on Theory of Computing* (ACM, New York, 1989), pp. 33–43

[38] M. Naor, M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks, in *22nd Annual ACM Symposium on Theory of Computing* (ACM, New York, 1990)

[39] P. Paillier, J.L. Villar, Trading one-wayness against chosen-ciphertext security in factoring-based encryption, in *Advances in Cryptology—ASIACRYPT 2006*, ed. by X. Lai, K. Chen. Lecture Notes in Computer Science, vol. 4284 (Springer, Berlin, 2006), pp. 252–266

[40] C. Peikert, B. Waters, Lossy trapdoor functions and their applications, in *40th Annual ACM Symposium on Theory of Computing*, ed. by R.E. Ladner, C. Dwork (ACM, New York, 2008), pp. 187–196

[41] D.H. Phan, D. Pointcheval, About the security of ciphers (semantic security and pseudo-random permutations), in *SAC 2004: 11th Annual International Workshop on Selected Areas in Cryptography*, ed. by H. Handschuh, A. Hasan. Lecture Notes in Computer Science, vol. 3357 (Springer, Berlin, 2004), pp. 182–197

[42] M.O. Rabin, Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, January 1979

[43] C. Rackoff, D.R. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in *Advances in Cryptology—CRYPTO'91*, ed. by J. Feigenbaum. Lecture Notes in Computer Science, vol. 576 (Springer, Berlin, 1992), pp. 433–444

[44] H. Wee, Efficient chosen-ciphertext security via extractable hash proofs, in *Advances in Cryptology—CRYPTO 2010*, ed. by T. Rabin. Lecture Notes in Computer Science, vol. 6223 (Springer, Berlin, 2010), pp. 314–332

[45] H.C. Williams, A modification of the RSA public-key encryption procedure. *IEEE Trans. Inf. Theory* **26**(6), 726–729 (1980)