Chosen Ciphertext Secure Encryption under Factoring Assumption Revisited*

Qixiang Mei[1,2], Bao Li[1], Xianhui Lu[1], and Dingding Jia[1]

[1] State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing, 100049, China
[2] School of Information, Guangdong Ocean University, Zhanjiang, 524088, China
nupf@163.com, {lb,xhlu,ddjia}@is.ac.cn

Abstract. In Eurocrypt 2009, Hofheinz and Kiltz proposed a practical chosen ciphertext (CCA) secure public key encryption under factoring assumption based on Rabin trapdoor one-way *permutation*.

We show that when the modulus is special such that Z_N^* has semi-smooth order, the instantiation of Hofheinz-Kiltz 09 scheme (HK09) over a much smaller subgroup of quadratic residue group (Semi-smooth Subgroup) is CCA secure as long as this type of modulus is hard to be factored. Since the exponent domain of this instantiation is much smaller than the original one, the efficiency is substantially improved.

In addition, we show how to construct a practical CCA secure encryption scheme from ElGamal trapdoor one-way *function* under factoring assumption. When instantiated over Semi-smooth Subgroup, this scheme has even better decryption efficiency than HK09 instantiation.

Keywords: public key encryption, chosen ciphertext secure, semi-smooth subgroup, factoring assumption.

1 Introduction

Chosen ciphertext security is now widely accepted as the standard security notion for the public key encryption. The first practical CCA secure public key encryption scheme without random oracle was proposed by Cramer and Shoup [6]. Their construction was later generalized to hash proof system [7]. However, the Cramer-Shoup encryption scheme and all its variants [20,16] inherently rely on decisional assumptions, e.g., the Decisional Diffie-Hellman (DDH) assumption, Decisional Composite Residuosity (DCR) assumption, and Decisional Quadratic Residuosity (DQR) assumption. In [24], Peikert and Waters proposed a general framework of constructing CCA secure encryption from the lossy trapdoor function. In [27], Rosen and Segev proposed a general way under the correlated inputs

* Supported by the National Natural Science Foundation of China (No.60862001, 61070171), the National Basic Research Program of China(973 project) (No.2007CB311201) and the Postdoctoral Science Foundation of China (No.20090460565).

D. Catalano et al. (Eds.): PKC 2011, LNCS 6571, pp. 210–227, 2011.

function. However, all the concrete constructions of lossy trapdoor function and correlated inputs function are also based on decisional assumptions.

It is widely believed that computational assumptions are more standard than their decisional versions. Canetti, Halevi and Katz [3] proposed the first practical public key encryption under a computational assumption, namely the Bilinear Diffie-Hellman assumption. In Eurocrypt 2008, Cash, Kiltz and Shoup [5] (CKS08) proposed a practical CCA secure scheme under the Computational Diffie-Hellman (CDH) assumption. Later in the same year, Hanaoka and Karosawa (HK08) proposed a more efficient CCA secure scheme under the CDH assumption [15]. Very recently, Haralambiev et al.[14], further improved the efficiency of CKS08 and HK08.

Even though the CCA secure schemes under CDH assumption are already fairly practical, since the pseudo-random generator for CDH problem can only extract one bit (or $O(\log(\lambda))$ with loose reduction), the encryption/decryption require $O(\lambda)$ (or $O(\lambda/\log(\lambda))$ respectively) modular exponentiations, where λ denotes the security lever parameter.

Hofheinz and Kiltz proposed a practical CCA secure PKE in Eurocrypt 2009 [17]. The Hofheinz-Kiltz 2009 scheme (HK09) [17] is constructed from the Blum-Goldwasser encryption [2], which itself is based on the Rabin encryption scheme [25] and Blum-Blum-Shub (BBS) generator[1]. The noticeable property of HK09 is that it only add a group element in Z_N^* to BG scheme and can be proved under factoring assumption (instead of the related decisional assumption). Since the BBS generator extracts one bit with only one modular multiplication, both the encryption and decryption require only $O(1)$ modular exponentiations.

However, in original HK09, the exponent is chosen from $[(N-1)/4]$. For the secure level of 80, the bits length of N, ℓ_N, needs to be chosen at least as 1024. For higher security, the length needs to be chosen even larger. A natural problem is that can we choose smaller domain of the exponent to improve the efficiency under factoring assumption?

In HK09, its security proof heavily relies on the fact that Rabin encryption is a trapdoor one-way *permutation*. The same technique seems hard to be directly used to construct CCA encryption from another factoring based encryption, ElGamal encryption over composite modulus, since the latter is only a trapdoor one-way *function*. In TCC 2010, Cramer, Hofheinz and Kiltz obtained an efficient CCA encryption from the EIGamal encryption over composite modulus under RSA assumption [4] (For convenience, throughout this paper, we will refer their scheme under RSA assumption as CHK10). However, the security of CHK10 could not be proved under factoring assumption. In addition, since the authors did not give an efficient pseudo-random bits generator, the encryption/decryption require $O(\lambda)$ modular exponentiations. It is interesting to construct practical CCA encryption such that the encryption/decryption only require $O(1)$ modular exponentiations from ElGamal encryption over composite modulus under factoring assumption.

1.1 Contributions

We present a HK09 instantiation over the much smaller subgroup than QR_N for a special modulus, i,e., Z_N^* has semi-smooth order. We prove that as long as this type of modulus is hard to be factored, this instantiation is CCA secure. Compared to the original HK09, the domain of the exponent is much smaller, thus, the efficiency is substantially improved. More precisely, this type of modulus is of the form $N = PQ = (2pp' + 1)(2qq' + 1)$, where p' and q' are primes large enough but much smaller than P and Q respectively, p and q are product of some distinct odd primes smaller than a low bound B. We instantiate HK09 over the unique subgroup G of QR_N with order $p'q'$. For the convenience, we call this subgroup as Semi-smooth Subgroup throughout this paper. In this instantiation, the domain of the exponent is set as $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$, where a useful property for our proof is that a uniform element has almost the same distribution as the uniform element of $[p'q']$. We prove a simple but crucial lemma: computing the square roots resides of uniformly chosen element in G can be reduce to the factoring assumption about this type of modulus.

Another contribution of ours is that we construct a practical CCA secure encryption from ElGamal encryption over composite modulus under factoring assumption. As in HK09, the ciphertext only consists of two group elements in Z_N^*. Taking account of efficiency, we present the instantiation over Semi-smooth Subgroup. The decryption of this instantiation is more efficient than the decryption of HK09 instantiation. First of all, we need an efficient pseudo-random generator for ElGamal encryption over composite modulus to transform it from one-wayness secure encryption to indistinguishability secure one under factoring assumption. This can be achieved since, adapting the proof technique of [23], we are able to prove that, under factoring assumption, $BBS_r(g^{xy})$ is pseudo-random even given (g^x, g^y). To explain how to transform it into CCA secure, we describe our attempts towards the final scheme step by step. The first attempt is directly applying Kiltz 07 [19] to the composite modulus case: the public key is $(g, X' = g^{\rho'}, X = g^{\rho})$, private key is (ρ', ρ); the ciphertext is $(R = g^{\mu}, S = (X'^t X)^{\mu})$; the encapsulated key is $\mathrm{BBS}_r(X'^{\mu})(= \mathrm{BBS}_r(g^{\rho'\mu}))$. This scheme could not be proved CCA secure under factoring assumption using known techniques since the simulator could not answer the DDH oracle and could not compute exponent inversion modulo unknown order. Inspired by a fact proved in [18], i.e., factoring assumption imply the strong Diffie-Hellman assumption over the signed quadratic residue group, QR_N^+, we make our second attempt by instantiating Kiltz 07 over QR_N^+ so that the simulator is able to answer the DDH oracle. But the simulator still could not compute exponent inversion modulo unknown order. Inspired by the method of HK09, we further modify the scheme as follows: the public key is $(g, X' = g^{\rho'}, X = g^{2^{\nu}\rho})$, private key is (ρ', ρ); the ciphertext is $(R = |g^{\mu 2^{\nu}}|, S = |(X'^t X)^{\mu}|)$; the encapsulated key is $\mathrm{BBS}_r(|(X')^{2^{\nu}\mu}|)(= \mathrm{BBS}_r(|R^{\rho'}|) = \mathrm{BBS}_r(|g^{2^{\nu}\rho'\mu}|)$. Recall that, in HK09, they implicitly used the following fact: Given $A, B \in Z_N^*$, $x, y \in Z$, from the equation $A^x = B^y$, any one can efficiently compute $A^{c/y}$, where $c = \gcd(x, y)$. But when we attempt to directly apply the proof method of HK09 to our case, we find

that the simulator does not know one of bases, without of loss generality, we denote it as A. To overcome this problem, we construct another simulator such that he knows both $B' = B^{2^k}$ and $A' = A^{2^k}$ for some suitable $k \in Z^+$. Basing on the underlying fact used in [18] to prove factoring assumption imply strong DH assumption, i.e., if U and V both belong to QR_N^+, then the equation $U = V$ is equivalent to the equation $U^{2^k} = V^{2^k}$ for any $k \in Z^+$, this simulator is able to verify the equivalent equation $A'^x = B'^y$ and then compute the encapsulated key from this equation. In the real proof, instead of black-box reducing the CCA security of the encryption to the pseudo-randomness of the generator, we prove the pseudo-randomness of the generator and the CCA security of the encryption simultaneously.

Both our schemes presented in this paper are actually CCA secure key encapsulation mechanism, from which it is easy to obtain full CCA secure public key encryption [28].

2 Preliminaries

2.1 Key Encapsulation Mechanism

A key encapsulation mechanism consists of three algorithms: Key generation $Gen(1^\lambda)$, Encapsulation $Enc(PK)$, Decapsulation $Dec(SK, C)$.

$Gen(1^\lambda)$: A probabilistic polynomial-time key generation algorithm takes as input a security parameter λ and outputs a public-key PK and secret key SK.

$Enc(PK)$: A probabilistic polynomial-time encryption algorithm takes public-key PK as input, and outputs a pair (K, C), where K is the key and C is a ciphertext.

$Dec(SK, C)$: A decryption algorithm takes a ciphertext C and the secret key SK as input. It returns a key K.

We require that for all (PK, SK) output by $Gen(1^\lambda)$, all (K, C) output by $Enc(PK)$, we have $Dec(SK, C) = K$.

Definition 1. (CCA Secure KEM) *A key encapsulation mechanism is indistinguishable against chosen ciphertext attacks if any PPT adversary* M *has negligible advantage in the game defined between the adversary* M *and the challenger* D *as follows:*

1. When M queries a key generation oracle, D invokes $Gen(1^\lambda)$ to obtain (PK, SK), responds with PK.
2. When M queries a challenge oracle. D invokes $Enc(PK)$ to obtain C^*, K_0, and chooses a random bits string K_1 with the same length as K_0, chooses a random bit b, set $K^* = K_b$, responds with (C^*, K^*).
3. When M makes a sequence of calls to the decryption oracle. For each decryption oracle query, M submits a ciphertext C, and D invokes $Dec(SK, C)$ to obtain K, responds with the K. The only restriction is that the adversary M can not request the decryption of C^*.
4. Finally, the adversary outputs a guess b'.

The adversary's advantage in the above game is

$$\mathrm{Adv}^{\mathrm{CCA}}_{\mathrm{KEM,M}}(\lambda) = |Pr[\mathrm{M}(K_0) = 1] - Pr[\mathrm{M}(K_1) = 1]|$$

2.2 Target Collision Resistant Hash Function

Informally, we say that a function $\mathrm{H} : X \to Y$ is a target-collision resistant (TCR) hash function, if, given a random pre-image $x \in X$, it is hard to find $x' \neq x$ with $\mathrm{H}(x') = \mathrm{H}(x)$.

Definition 2. *Let* $\mathrm{H} : X \to Y$ *be a function. For an adversary* M*, define*

$$\mathrm{Adv}^{TCR}_{\mathrm{H,M}}(\lambda) = Pr[x \leftarrow X, x' \leftarrow \mathrm{M}(x, \mathrm{H}) : x' \neq x \wedge \mathrm{H}(x') = \mathrm{H}(x)]$$

We say that H *is target-collision resistant if for any PPT adversary* M, $\mathrm{Adv}^{TCR}_{\mathrm{H,M}}(\lambda)$ *is negligible.*

2.3 Semi-smooth Subgroup

In [13], the author introduced the definition of semi-smooth subgroup.

Let $\mathrm{IGen}(1^\lambda)$ be a probability polynomial-time algorithm such that on input security parameter λ, randomly chooses two $m(\lambda)$-bit primes P and Q satisfying $P = 2p'p + 1$, $Q = 2q'q + 1$, outputs $N = PQ$, where p' and q' are $m'(\lambda)$-bit primes, both p and q are product of some distinct odd primes smaller than a low bound B. We call such integer N as semi-smooth integer.

Definition 3. *Let* $N = (2p'p + 1)(2q'q + 1)$ *be a random output of* $\mathrm{IGen}(1^\lambda)$*, the unique subgroup* G *of order* $p'q'$ *is called the semi-smooth subgroup of* Z^*_N*.*

Factoring Assumption about Semi-smooth Integer. We assume that there exists no probabilistic polynomial-time algorithm such that given only N, the random output of $\mathrm{IGen}(1^\lambda)$, can factor N with non-negligible probability.

In [13], at secure level of 80, parameters are suggest to be $\ell_{p'} = \ell_{q'} = 160$, $\ell_N = 1024$, and $B = 2^{15}$.

Here, we describe some properties that will be used later.

Property 1. Let h be a uniform element of Z^*_N, $P_B = \prod_{1<p<B,\ \mathrm{p\ is\ prime}} p$, and $g = h^{P_B}$. Then, g is a uniform element of G.

Property 2. With probability $1 - O(2^{-m'(\lambda)})$, a uniform element in G is a generator of G.

Property 3. Any element z of G is a quadratic residue, the unique quadratic residue u such that $u^2 = z$ lies in G.

Property 4. For any element z of G, the unique quadratic residue u such that $z = u^{2^k}$ lies in G for any $k \in Z^+$.

2.4 Signed Quadratic Residues

The signed quadratic residues[18], QR_N^+, are defined as the group $QR_N^+ = \{|x| : x \in QR_N\}$, where $|x|$ is the absolute value when representing elements of Z_N^* as the set $\{-(N-1)/2, \cdots, (N-1)/2\}$, N is a Blum integer. The group operation $\circ$ is defined by $a \circ b = |ab \bmod N|$. For simplification, we denote $|ab|$ instead of $|ab \bmod N|$.

An attractive property is that the membership in QR_N^+ can be efficiently verified since $QR_N^+ = J_N^+ = J_N \bigcap [(N-1)/2]$, where J_N^+ denotes $\{|x| : x \in J_N\}$, and J_N denotes the group of elements with Jacobi symbol 1.

2.5 Some Lemmas

Lemma 1. *Let g be a generator of G, μ is chosen uniformly from $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$, k is any integer, then both $\mu \bmod p'q'$ and $(\mu+k) \bmod p'q'$ are statistically close to the uniform distribution of $[p'q']$, both g^μ and $g^{\mu+k}$ are statistically close to the uniform distribution of G.*

Proof. Write $2^{\ell_{p'}+\ell_{q'}+\lambda}$ as $k_1 p'q' + k_2$ over Z, where $0 < k_2 < p'q'$. If μ is uniformly chosen from $[k_1 p'q']$, then both $\mu \bmod p'q'$ and $(\mu + k)$ *mod* $p'q'$ are uniform in $[p'q']$, and then both g^μ and $g^{\mu+k}$ are uniform in G. But a uniformly chosen element from $2^{\ell_{p'}+\ell_{q'}+\lambda}$ belongs to $[k_1 p'q']$ with probability $k_1 p'q'/2^{\ell_{p'}+\ell_{q'}+\lambda} = 1 - k_2/2^{\ell_{p'}+\ell_{q'}+\lambda} \geq 1 - O(2^{-\lambda})$.

The following lemma states that computing the square root residue of random element in semi-smooth subgroup can be reduced to the factoring algorithm for the modulus N.

Lemma 2. *Let G be the semi-smooth subgroup of Z_N^*, z is a uniformly chosen element of G, if there exists an adversary* A *can compute the unique quadratic residue u such that $u^2 = z$ with non-negligible probability, then there exists an adversary* C *can factor N with non-negligible probability.*

Proof. Given N, C chooses h uniformly from Z_N^*, set $P'_B = \prod_{2<p<B,\ \mathrm{p\ is\ prime}} p$, and $h' = h^2$, $z = h'^{P'_B} (= h^{P_B})$. So z is a uniform element of G. If A can compute u such that $u^2 = z$, then C can compute $\widetilde{h}$ such that $\widetilde{h}^2 = h^2$: compute a, b over Z such that $aP'_B + 2b = gcd(P'_B, 2) = 1$, set $\widetilde{h} = u^a h'^b$. If $\widetilde{h} \neq \pm h$, then C outputs $\gcd(\widetilde{h} - h, N)$. With probability $1/2$, $\widetilde{h} \neq \pm h$, and so $\gcd(\widetilde{h} - h, N)$ is a non-trivial factor of N.

From lemma 2 and the Goldreich-Levin lemma[12], it is easy to see that given $z = u^2$ over G, the Goldreich-Levin predicate, $B_r(u)$, is a hard-core. Using the hybrid argument,we have:

Lemma 3. *Let G be the semi-smooth subgroup of Z_N^*, given a uniform element z of G, then $BBS_r(u)$ is indistinguishable from the uniform bits string U from $[2^{\ell_K}]$ under the assumption factoring N is hard, where u is the unique quadratic residue such that $z = u^{2^{\ell_K}}$, $\mathrm{BBS}_r(u) \stackrel{\mathrm{def}}{=} (B_r(u), B_r(u^2), \cdots, B_r(u^{2^{\ell_K - 1}}))$, r is a random element with bits-size ℓ_N.*

Lemma 4. *Given A, B in some subgroup F of Z_N^*, along with x, y in Z, such that $A^x = B^y$, $\gcd(x, y) = z$, $\gcd(y, \mathrm{ord}(F)) = 1$. Then one can efficiently compute $A^{z/y}$, where the inversion in the exponent is computed modulo* $\mathrm{ord}(F)$.

Proof. Since $\gcd(x, y) = z$, using the extended Euclidean algorithm, one can compute a, b over Z such that $ax + by = z$. Let $B' = A^b B^a$. Since A, B belongs to F , so B' belongs to F. It is easy to verify that $A^z = (B')^y$. Since $\gcd(y, \mathrm{ord}(F)) = 1$, so $B' = A^{z/y}$.

Lemma 5. *If $A, B \in QR_N^+$, then $A^2 = B^2 \bmod N \Leftrightarrow A = B$. More generally, $A^{2^k} = B^{2^k} \bmod N$ $(k \in Z^+) \Leftrightarrow A = B$.*

Lemma 6. *If $A, B \in QR_N \cup QR_N^+$, then $|AB| = \big||A||B|\big|$.*

3 The Instantiation of HK09 over Semi-smooth Subgroup

3.1 Scheme Description

$Gen(1^\lambda)$: Run $\mathrm{IGen}(1^\lambda)$ to get the modulus N. Then, Gen chooses a target-collision resistant hash function $\mathrm{H} : Z_N \to [2^{\ell_\mathrm{H}} - 1]$. Next, Gen randomly chooses an element g of G, a bit string r of length ℓ_N, and ρ from $[2^{\ell_{p'} + \ell_{q'} + \lambda}]$. Finally, Gen sets $X = g^{\rho 2^\nu}$ $(\nu = \ell_\mathrm{H} + \ell_\mathrm{K})$. The public key is $PK = (N, g, X, r, \mathrm{H})$, and the private key is $SK = \rho$.

$Enc(PK)$: Enc randomly chooses $\mu \in [2^{\ell_{p'} + \ell_{q'} + \lambda}]$, and computes

$$R = g^{\mu 2^\nu}, \qquad t = \mathrm{H}(R), \qquad S = |(g^t X)^\mu|.$$

Set the ciphertext as $C = (R, S)$. Compute the encapsulation key as $K = \mathrm{BBS}_r(g^{\mu 2^{\ell_\mathrm{H}}})$.

$Dec(SK, C)$: Dec writes C as $C = (R, S)$, verifies both R, S belong to $Z_N^* \times (Z_N^* \bigcap [(N-1)/2])$ and rejects it if not. Then Dec computes $t = \mathrm{H}(R)$, verifies:

$$(S^2)^{2^\nu} = (R^2)^{t + \rho 2^\nu} \tag{1}$$

Reject it if it not. Dec computes $a, b, c \in Z$ such that

$$2^c = gcd(t, 2^\nu) = at + b2^\nu \tag{2}$$

Then Dec computes

$$T = ((S^2)^a \cdot (R^2)^{b - a\rho})^{2^{\ell_\mathrm{H} - c - 1}} \tag{3}$$

and $K = \mathrm{BBS}_r(T)$, outputs K

Correctness: Notice that, in *Dec*, even R, S may not sit in the subgroup G, as long as they pass the verification, they must sit in Z_N^*. So from the proof of the original HK09, the computed T is equal to $(R^2)^{1/(2^{\ell_K+1}) \bmod \mathrm{ord}(QR_N)}$. If $R = g^{\mu 2^{\ell_K+\ell_H}} \in G \subset QR_N$, then, we have $(R^2)^{1/(2^{\ell_K+1}) \bmod \mathrm{ord}(QR_N)} = (R^2)^{1/(2^{\ell_K+1}) \bmod \mathrm{ord}(QR_N) \bmod \mathrm{ord}(G)} = (R^2)^{1/(2^{\ell_K+1}) \bmod \mathrm{ord}(G)} = g^{\mu 2^{\ell_H}}$.

Efficiency comparison with original HK09. The encapsulation and decapsulation of both the original HK09 and this variant need $3\ell_{\exp} + \ell_K + 2.5\ell_H$ and $1.5\ell_{\exp} + 4\ell_K + 6.5\ell_H$ multiplications respectively, where both ℓ_K and ℓ_H can be set as λ. The difference is that, in original HK09, $\ell_{\exp}$ equals to ℓ_N, instead, in this variant, $\ell_{\exp}$ equals to $\ell_{p'} + \ell_{q'} + \lambda$. For 80-bits security, $\ell_N = 1024$, $\ell_{p'} = \ell_{q'} = 160$, $\lambda = 80$. In original HK09, the encapsulation requires 3352 multiplications, the decapsulation requires 2376 multiplications. In this variant, the encapsulation requires 1480 multiplications, the decapsulation requires 1440 multiplications.

3.2 Security Proof

Theorem 1. *If factoring the modulus N is hard and* H *is target-collision resistant, then the above key encapsulation mechanism is chosen ciphertext secure.*

Proof. To prove this theorem, from lemma 3, it is enough to reduce the CCA security of this scheme to the pseudo-randomness of the BBS generator over the Semi-smooth Subgroup.

Assume there exists an adversary A on KEM's IND-CCA security. We define an adversary D on the pseudo-randomness of the BBS generator. On input (N, z, V), the goal of D is to distinguish whether V is $\mathrm{BBS}_r(u)$ or a uniform bits string with equal length, where u is the unique quadratic residue in G such that $z = u^{2^{\ell_K}}$, z is a uniform element in G.

Prepare the public key. D chooses a target-collision resistant hash function $\mathrm{H} : Z_N \to [2^{\ell_H} - 1]$, a bits string r of length ℓ_N, a random element $g \in G$, as well as $\beta \in [2^{\ell_{p'}+\ell_{q'}+\lambda}]$, sets

$$R^* = z, \qquad t^* = \mathrm{H}(R^*), \qquad X = g^{\beta 2^\nu - t^*}.$$

The public key is set as $PK = (N, g, X, r, \mathrm{H})$. The private key is implicitly defined as $\rho = \beta - t^*/2^\nu \bmod p'q'$.

Prepare the challenge cipertext and key. Next, we assume g is a generator of G. So we can write $R^* = g^{\mu^* 2^\nu}$, though μ^* is unknown to D. D defines

$$S^* = |R^{*\beta}| \quad (= |g^{\mu^* \beta 2^\nu}| = |(g^{t^*} X)^{\mu^*}|).$$

The real corresponding key K^* is defined as

$$K^* = \mathrm{BBS}_r(g^{\mu^* 2^{\ell_H}}) = \mathrm{BBS}_r(R^{*\frac{1}{2^{\ell_K}}}) = \mathrm{BBS}_r(z^{\frac{1}{2^{\ell_K}}}) = \mathrm{BBS}_r(u)$$

The challenge ciphertext is $C^* = (R^*, S^*)$, the challenge key is V. Note that, as in the IND-CCA2 game, if V is $\mathrm{BBS}_r(u)$, then V is a real key , else V is a uniform string.

We claim that the distribution of the public key and the challenge ciphertext C^* is almost identical in simulation and IND-CCA game. Firstly, in public key, g, N, r and H are perfectly simulated. From Property 2, with overwhelming probability, g is a generator of G. From Lemma 1, we know that if g is a generator of G, then X in the real game and in simulation are both statistically close to the uniform element in G. So X is simulated perfectly with overwhelming probability. Similarly, with overwhelming, R^* is also perfectly simulated. Conditioned on X, R^*, g, r, N are simulated perfectly, from the simulation, we know that S^* and K^* are also perfectly simulated. As required.

Answer the decryption queries. When A submit a ciphertext (R, S), D does as following.

Check $(R, S) \in Z_N^* \times (Z_N^* \bigcap [(N-1)/2])$, reject if not. Compute $t = \mathrm{H}(R)$.

For the case $t \neq t^*$. Verify:

$$(S^2)^{2^\nu} = (R^2)^{t-t^*+\beta 2^\nu} \tag{4}$$

Reject it if it not.

Note that the equation (4) is equivalent to

$$(R^2)^{(t-t^*)} = (R^{-2\beta} S^2)^{2^{\ell_\mathrm{H}+\ell_\mathrm{K}}}$$

Since R^2 and $R^{-2\beta}S^2$ belong to QR_N, using lemma 4, the simulator can compute $B' = (R^2)^{2^{c'}/2^{\ell_\mathrm{H}+\ell_\mathrm{K}}}$, where $2^{c'} = \gcd(t-t^*, 2^{\ell_\mathrm{H}+\ell_\mathrm{K}})$, the inversion in the exponent is computed modulo $\mathrm{ord}(QR_N)$. Then Dec can compute $T = (R^2)^{1/2^{\ell_\mathrm{K}+1}} = (B')^{2^{\ell_\mathrm{H}-1-c'}}$ since $\ell_\mathrm{H} - 1 - c' \geq 0$. If $R = g^{\mu 2^{\ell_\mathrm{K}+\ell_\mathrm{H}}}$, then $(R^2)^{1/(2^{\ell_\mathrm{K}+1}) \bmod \mathrm{ord}(QR_N)} = (R^2)^{1/(2^{\ell_\mathrm{K}+1}) \bmod \mathrm{ord}(G)} = g^{\mu 2^{\ell_\mathrm{H}}}$. Concretely, the simulator compute $a', b', c' \in Z$ such that

$$2^{c'} = gcd(t-t^*, 2^\nu) = a'(t-t^*) + b' 2^\nu \tag{5}$$

Then compute

$$T = ((S^2)^{a'} \cdot (R^2)^{b'-a'\beta})^{2^{\ell_\mathrm{H}-c'-1}} \tag{6}$$

D answer with $\mathrm{BBS}_r(T)$.

For the case $t = t^*$. If $R = R^*$ and the ciphertext is valid , it will satisfy

$$(S^2) = (R^2)^{(t-t^*)/2^\nu+\beta} = (R^2)^\beta = S^{*2}.$$

Therefore, $S^2 = S^{*2}$. Furthermore, $(R, S) \neq (R^*, S^*)$ implies that $|S| = S \neq S^* = |S^*|$, so that $S \neq \pm S^*$ and $(S+S^*)(S-S^*) = S^2 - S^{*2} = 0 \bmod N$ yields a non-trivial factor of N.

If $t = t^*$ and $R \neq R^*$, then it will contradict the target-collision resistance of H, so D can safely give up this type of ciphertext.

So with overwhelming probability, D perfectly simulates the CCA game.

D outputs what A outputs.

Therefore, D can use A as an oracle to distinguish whether V is $\mathrm{BBS}_r(u)$ or a uniform bits string.

4 Scheme Based on ElGamal Encryption over Composite Modulus

In this section, we show how to construct a practical CCA secure KEM from ElGamal encryption over composite modulus. Taking account of efficiency, we present the instantiation over semi-smooth subgroup. This scheme implicitly uses the signed quadratic residues group [18].

4.1 Scheme Description

$Gen(1^\lambda)$: Run $\mathrm{IGen}(1^\lambda)$ to get the modulus N. Then, choose a target-collision resistant hash function $\mathrm{H} : Z_N \to [2^{\ell_\mathrm{H}} - 1]$. Next, randomly choose an element g of the semi-smooth subgroup G, a bit string r of length $\ell_{(N-1)/2}$, and $\rho, \rho' \in [2^{\ell_{p'}+\ell_{q'}+\lambda}]$. Finally, set $X = g^{\rho 2^\nu}$ $(\nu = \ell_\mathrm{H} - 1)$ and $X' = g^{\rho'}$. The public key is $PK = (N, g, X, X', r, \mathrm{H})$, and the private key is $SK = (\rho, \rho')$.

$Enc(PK)$: Enc randomly chooses $\mu \in [2^{\ell_{p'}+\ell_{q'}+\lambda}]$, and computes

$$R = |g^{\mu 2^\nu}|, \qquad t = \mathrm{H}(R), \qquad S = |(X'^t X)^\mu|, \qquad T = |X'^{\mu 2^\nu}|,$$

$$K = \mathrm{BBS}_r^+(T) \stackrel{\mathrm{def}}{=} (B_r(|T|), B_r(|T^2|), \cdots, B_r(|T^{2^{\ell_\mathrm{K}-1}}|)).$$

Set the ciphertext as $C = (R, S)$ and the encapsulation key as K.

$Dec(SK, C)$: Dec writes C as $C = (R, S)$, verifies both R and S belong to $QR_N^+ = J_N \bigcap [(N-1)/2]$. If it holds, then Dec computes $t = \mathrm{H}(R)$, verifies:

$$|S^{2^\nu}| = |R^{\rho' t + \rho 2^\nu}|$$

If it holds, then Dec computes

$$T = |R^{\rho'}|, \quad K = \mathrm{BBS}_r^+(T).$$

Correctness: If R and S are computed according to the encapsulation, then both R and S belong to G^+. Since $G^+ \subseteq QR_N^+$, so $R, S \in QR_N^+$. From lemma 5, we know that $|AB| = ||A||B||$ as long as $A, B \in QR_N \cup QR_N^+$, then

$$|S^{2^\nu}| = ||(X'^t X)^\mu|^{2^\nu}| = |g^{(\rho' t + \rho 2^\nu)\mu 2^\nu}| = |R^{\rho' t + \rho 2^\nu}|.$$

The fact that $|R^{\rho'}|$ equals to $|X'^{\mu 2^\nu}|$ follows from:

$$|X'^{\mu 2^{\nu}}| = |g^{\rho' \mu 2^{\nu}}| = ||g^{\mu 2^{\nu}}|^{\rho'}| = |R^{\rho'}|$$

Efficiency: The ciphertext of this KEM consists of two group element (Notice that for the known CCA secure KEM schemes based on the ElGamal encryption over prime modulus under CDH assumption, the ciphertexts consist of at least three group elements). If we choose $\ell_{q_1} = \ell_{p_1} = 160$, $\lambda = 80$, then $\ell_\rho = \ell_{\rho'} = \ell_{\exp} = 400$. We assume $\ell_{\mathrm{H}} = 80$. As in original HK09, we assume one regular exponentiation with an exponent of length ℓ requires 1.5ℓ modular multiplications and that one squaring takes the same time as one multiplication. Notice that we can compute $g^{\rho'}$ and g^{ρ} with about 1.2 exponentiations since they share the same base g. The encapsulation requires $4.5\ell_{\exp} + 2.5\ell_{\mathrm{H}} + \ell_{\mathrm{K}} = 2080$ multiplications. The decapsulation requires $1.5 \times 1.2\ell_{\exp} + 2.5\ell_{\mathrm{H}} + \ell_{\mathrm{K}} = 1000$ multiplications.

4.2 Security Proof

Theorem 2. *If factoring the modulus N is hard and* H *is target-collision resistant, then the above key encapsulation mechanism is chosen ciphertext secure.*

High level of the proof: In HK09 instantiation (and the original HK09), the proof consists of two steps: firstly, the pseudo-randomness of the BBS generator $\mathrm{BBS}_r(u)$ is reduced to the factoring assumption; then, the CCA security is black box reduced to the pseudo-randomness of the BBS generator $\mathrm{BBS}_r(u)$. But, in this scheme, if we directly reduce the CCA security to the pseudo-randomness of $\mathrm{BBS}_r^+(g^{\mu\rho'})$ (even g^{μ} and $g^{\rho'}$ is given), then the simulator could not answer DDH oracle that is needed for the verification and could not compute the inversion modulo the unknown order $p'q'$ which is needed to compute the encapsulated key. Instead, we prove the CCA security of this scheme *and* the pseudo-randomness of $\mathrm{BBS}_r^+(g^{\mu\rho'})$ *simultaneously.* Adapting the proof idea of [23], we firstly reduce the security (both the CCA security of this scheme and the pseudo-randomness of $\mathrm{BBS}_r^+(g^{\mu\rho'})$) to a hardcore distinguisher; next, we reduce the hardcore distinguisher to a hardcore predictor; finally, we reduce the hardcore predictor to a factoring algorithm. In the first step, the distinguisher could compute $\rho' 2^{\ell_{\mathrm{K}}} \bmod p'q'$, so he could compute $|R^{\rho' 2^{\ell_{\mathrm{K}}}}|$. The distinguisher does not directly verify the equation $|S^{2^{\nu}}| = |R^{\rho' t + \rho 2^{\nu}}|$, instead, he verify a equivalent equation $S^{2^{\nu+\ell_{\mathrm{K}}}} = R^{\rho' t 2^{\ell_{\mathrm{K}}} + \rho 2^{\nu+\ell_{\mathrm{K}}}}$ (Before this, he should verify both R and S belong to QR_N^+, from lemma 5, we know that the two equations are equivalent. Note that, in [18], this technique has been used for proving the factoring assumption implies the strong DH assumption over QR_N^+). Then, by using lemma 4, the distinguisher is able to efficiently compute the encapsulated key from the latter equation.

Proof. The theorem is the consequence of the following three lemmas.

Reduce to the hard-core distinguisher $\mathrm{D}(v^2, N, r, \alpha)$

Lemma 7. *If there exists a PPT adversary* M *such that* $\mathrm{Adv}^{\mathrm{CCA}}_{\mathrm{KEM,M}}(\lambda)$ *equals to* $\varepsilon(\lambda)$*, then there exists a PPT adversary* $\mathrm{D}(v^2, N, r, \alpha)$ *that distinguishes whether* α *is equal to* $B^+_r(|uw|)$ *or a random bit* b *with advantage* $\varepsilon'(\lambda)$*, where* v^2 *is a uniformly chosen element of* G*,* u *is the unique square root residue of* v^2*,* w *is determined by* v^2 *and* D*'s internal coin tosses, and* $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)}{\ell_{\mathrm{K}}}$.

Proof. On input (v^2, N, r, α), D works as follows.

Prepare the public key: Choose a target-collision resistant hash function $\mathrm{H} : Z_N \rightarrow [2^{\ell_{\mathrm{H}}} - 1]$. Randomly choose $J = k$ from $\{0, 1, \cdots \ell_{\mathrm{K}} - 1\}$. Select at random bits string $(b_0, b_1, \cdots, b_{k-1})$. Randomly and independently select 2 elements $\xi_i (i = 1, 2)$ from $[2^{\ell_{p'} + \ell_{q'} + \lambda}]$, denote $\overrightarrow{\xi} = (\xi_1, \xi_2)$. Set $s = 2\ell_{\mathrm{K}} - k$, $g = v^{2^s} \bmod N$, and $a_i = (\xi_i + 2^{-\ell_{\mathrm{K}}}) \bmod p'q' \quad (i = 1, 2)$. Set $X' = g^{a_1} = g^{\xi_1 + 2^{-\ell_{\mathrm{K}}} \bmod p'q'}$ (implicitly define $\rho' = (\xi_1 + 2^{-\ell_{\mathrm{K}}}) \bmod p'q'$). Set $B = g^{a_2} = g^{\xi_2 + 2^{-\ell_{\mathrm{K}}} \bmod p'q'}$ (implicitly define $B = g^{\mu^* 2^\nu}$). Set $t^* = \mathrm{H}(|B|)$. Randomly choose $\beta \in [2^{\ell_{p'} + \ell_{q'} + \lambda}]$, and set $X = g^{\beta 2^\nu} X'^{-t^*}$ (implicitly define $\rho = (\beta - \rho' t^* / 2^\nu) \bmod p'q'$). The public key is set as $(N, g, X', X, r, \mathrm{H})$.

(D can efficiently compute $X' = g^{a_1} = g^{\xi_1 + 2^{-\ell_{\mathrm{K}}} \bmod p'q'} = v^{2^{(\ell_{\mathrm{K}} - k)}} v^{2^{(2\ell_{\mathrm{K}} - k)} \xi_1}$ since $\ell_{\mathrm{K}} > k$ and v^2 is given. Similar, D can be efficiently computed too. It is easy to see that other elements of the public key can be efficiently simulated by D).

Prepare the challenge ciphertext and key: The challenge ciphertext is set as:

$$R^* = |B| \quad (= |g^{\mu^* 2^\nu}|); \qquad S^* = |R^{*\beta}| \quad (= |(X'^{t^*} X)^{\mu^*}|)$$

And the challenge key is set as

$$K^* = (b_0, b_1, \cdots, b_{k-1}, \alpha, B_r(|g^{2^{k+1} a_1 a_2}|), \cdots, B_r(|g^{2^{\ell_{\mathrm{K}} - 1} a_1 a_2}|))$$

Define w**:** We define $w = (v^{2^{2\ell_{\mathrm{K}}}})^{\xi_1 \xi_2} (v^{2^{\ell_{\mathrm{K}}}})^{\xi_1 + \xi_2}$. Given the values of ξ_1, ξ_2 and v^2, D is able to efficiently compute w. It is easy to see that, w is a quadratic residue in G (recall that $v^2 \in G$)and is determined by v^2 and D's internal coin tosses.

Claim 1. Let a_1, a_2, g, u, w be defined as above respectively, then $g^{2^k a_1 a_2} = uw$.

Proof.

$$g^{2^k a_1 a_2} = g^{2^k (\xi_1 + 2^{-\ell_{\mathrm{K}}})(\xi_2 + 2^{-\ell_{\mathrm{K}}})} = g^{2^k (\xi_1 \xi_2 + (\xi_1 + \xi_2) 2^{-\ell_{\mathrm{K}}} + 2^{-2\ell_{\mathrm{K}}})}$$
$$= v^{2^{2\ell_{\mathrm{K}}} (\xi_1 \xi_2 + (\xi_1 + \xi_2) 2^{-\ell_{\mathrm{K}}} + 2^{-2\ell_{\mathrm{K}}})} = uw.$$

Claim 2. D is able to compute $g^{2^{k+j} a_1 a_2}$ for $j = 1, \ldots \ell_{\mathrm{K}} - k - 1$.

Proof. From Claim 1, we have $g^{2^k a_1 a_2} = uw$, so each $g^{2^{k+j} a_1 a_2}$ equals to $(uw)^{2^j}$ for $j = 1, \ldots \ell_{\mathrm{K}} - k - 1$. Furthermore, since D knows v^2 and w^2, so he is able to compute $(uw)^{2^j}$ for $j = 1, \ldots \ell_{\mathrm{K}} - k - 1$, as required.

From claim 1 and 2, it is easy to see that, D is able to efficiently prepare the challenge ciphertext and key.

Answer the decryption queries: For the query ciphetext (R, S), D verifies both R and S belong to QR_N^+. If it holds, D computes $t = \mathrm{H}(R)$.

If $t \neq t^*$, D verifies if

$$(S^{2^{\ell_\mathrm{K}}})^{2^\nu} = (R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{t-t^*}(R^{2^{\ell_\mathrm{K}}})^{\beta 2^\nu} \tag{7}$$

(Note that the right side equals to

$$(R^{(2^{\ell_\mathrm{K}}\rho')})^{t-t^*}(R^{2^{\ell_\mathrm{K}}})^{\beta 2^\nu} = (R^{2^{\ell_\mathrm{K}}})^{\rho' t-\rho' t^*+\beta 2^\nu} = (R^{2^{\ell_\mathrm{K}}})^{\rho' t+\rho 2^\nu}$$

From Lemma 5, we know that verifying $|S^{2^\nu}| = |R^{\rho' t+\rho 2^\nu}|$ is equivalent to verifying $(S^{2^{\ell_\mathrm{K}}})^{2^\nu} = (R^{2^{\ell_\mathrm{K}}})^{\rho' t+\rho 2^\nu}$, as required).

Equation (7) is equivalent to $(R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{t-t^*} = (SR^{-\beta})^{2^{\ell_\mathrm{K}+\nu}}$. Since $R^{(2^{\ell_\mathrm{K}}\xi_1+1)}$, $SR^{-\beta}$ belong to QR_N^+, using lemma 4, D is able to compute $(R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{2^{c'}/2^{\ell_\mathrm{K}+\nu}}$, where $2^{c'} = \gcd(t-t^*, 2^{\nu+\ell_\mathrm{K}})$, the inversion in the exponent is computed modulo $\mathrm{ord}(QR_N^+)(= \mathrm{ord}(QR_N))$. Furthermore, since both t and t^* are smaller than 2^{ℓ_H}, then $c' \leq \ell_\mathrm{H} - 1 = \nu$. Therefore, D is able to compute

$$T = |((R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{2^{c'}/2^{\ell_\mathrm{K}+\nu}})^{2^{\nu-c'}}|(= |(R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{1/2^{\ell_\mathrm{K}}}| = |R^{\rho'}|)$$

Concretely, if (7) holds, D computes $a', b', c' \in Z$ such that:

$$2^{c'} = \gcd(t-t^*, 2^{\nu+\ell_\mathrm{K}}) = a'(t-t^*) + b' 2^{\nu+\ell_\mathrm{K}}$$

Then D computes

$$T = |((SR^{-\beta})^{a'} R^{b'(2^{\ell_\mathrm{K}}\xi_1+1)})^{2^{\nu-c'}}|$$

Response the oracle with $\mathrm{BBS}_r^+(T)$.

If $t = t^*$, D rejects the query ciphetext (R, S). (If $\mathrm{H}(R) = t = t^* = \mathrm{H}(R^*)$ and $R \neq R^*$, then M has broken the target-collision resistance of H. If $t = t^*$ and $R = R^*$, and the ciphertext is valid, then we have

$$S = |S| = |((R^{(2^{\ell_\mathrm{K}}\xi_1+1)})^{t-t^*}(R^{2^{\ell_\mathrm{K}}})^{\beta 2^\nu})^{1/(2^{\nu+\ell_\mathrm{K}})}| = |R^\beta| = |(R^*)^\beta| = S^*$$

which means that $(R, S) = (R^*, S^*)$, so this query will be rejected, as required).

When M outputs a bit, D outputs the same bit.

The running time of D**:** It is easy to see that D can run in polynomial time.

The success-probability of D**.** To find the success probability of D, we prove that the distribution of the public key, challenge ciphetext, and the decryption in the simulated game is statistically close to that in the real game.

Since v^2 is a uniformly chosen element of G, and squaring is a permutation, so the above defined g is a uniformly distributed element in G. Thus, g is perfectly simulated. Obviously, N, r and H are perfectly simulated. From property 2, we know that with probability $1 - O(2^{-m'(\lambda)}) \geq 1 - O(2^{-\lambda})$, g is a generator. From

Lemma 1, we know that with probability $1 - O(2^{-\lambda})$, X' in simulation and in real game are both statistically close to the uniformly distributed element in G. So, with probability $1 - O(2^{-\lambda})$, X' is perfectly simulated. With the same analysis, with probability $1 - O(2^{-\lambda})$, X and R^* are perfectly simulated.

Therefore, the statistical distance between distribution of the public key in the simulated game and that in the real game is $O(2^{-\lambda})$.

Note that, conditioned on the public key is simulated perfectly, the challenge ciphertext is perfectly simulated, and the decryption oracle is simulated perfectly except the case that M finds a target collision, which occurs with negligible probability $\mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)$.

For convenience, we denote some hybrid experiments $H^J(J = 0, \cdots, \ell_{\mathrm{K}})$ the same as the real game except the way the challenge key is responded with: the first J bits are chosen randomly, while the other $\ell_{\mathrm{K}} - J$ bits are computed as in K_0. So in the experiment H^0, the distribution of the key that the adversary sees is the same as K_0, whereas in the experiment $H^{\ell_{\mathrm{K}}}$, the distribution of the key is the same as K_1.

From Claim 1, we know that, if $\alpha = B_r^+(|uw|)$, then the distribution that M sees is the simulated H^J , while if α is a random bit b, then the distribution that M sees is the simulated H^{J+1}. We denote the simulated H^k as H_S^k for each $k \in \{0, 1, \cdots, \ell_{\mathrm{K}}\}$, and still denote real H^k as H^k. So the advantage of D is:

$$
\begin{aligned}
&|Pr[\mathrm{D}(B_r^+(|uw|)) = 1] - Pr[\mathrm{D}(b) = 1]| \\
&= \tfrac{1}{\ell_{\mathrm{K}}} |\textstyle\sum_{j=0}^{\ell_{\mathrm{K}}-1} \{Pr[\mathrm{D}(B_r^+(|uw|)) = 1 | J = j] - Pr[\mathrm{D}(b) = 1 | J = j]\}| \\
&= \tfrac{1}{\ell_{\mathrm{K}}} |\textstyle\sum_{j=0}^{\ell_{\mathrm{K}}-1} \{Pr[\mathrm{M}(H_S^j) = 1] - Pr[\mathrm{M}(H_S^{j+1}) = 1]\}| \\
&= \tfrac{1}{\ell_{\mathrm{K}}} |Pr[\mathrm{M}(H_S^0) = 1] - Pr[\mathrm{M}(H_S^{\ell_{\mathrm{K}}}) = 1]| \\
&\geq \tfrac{1}{\ell_{\mathrm{K}}} \{|Pr[\mathrm{M}(H^0) = 1] - Pr[\mathrm{M}(H^{\ell_{\mathrm{K}}}) = 1]| - O(2^{-\lambda}) - \mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)\} \\
&= \tfrac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)}{\ell_{\mathrm{K}}}
\end{aligned}
$$

This completes the proof of Lemma 7.

Reduce to the hard-core predictor $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$

Since D defined in Lemma 7 chooses ξ_1, ξ_2 itself and w depends on v^2 and ξ_1, ξ_2 , then the value of w potentially changes each time D is invoked. Furthermore, D is not a predictor for $B_r^+(|uw|)$ but rather a distinguisher. So D is not suitable to be used as an oracle for the Goldreich-Levin reconstruction algorithm [12]. The first problem can be solved by fixing the value ξ_1, ξ_2 in advance. The second problem can be addressed by reducing the hard-core distinguisher to a suitable hard-core predictor. On input $< r >$, the hard-core predictor $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$ is defined as follows:

1. Uniformly choose random bits α and β .
2. Invoke D on input $< v^2, N, r, \alpha >$, and feed it with ξ_1, ξ_2 .
3. If D outputs 1, then output α, else if D outputs 0, then output β.

Note that now, the value of w does not change with the invoking of $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$. So it is possible to use $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$ as an oracle to reconstruct $|uw|$.

Lemma 8. *If there exists a PPT adversary* M *such that* $\mathrm{Adv}^{\mathrm{CCA}}_{\mathrm{KEM,M}}(\lambda)$ *equals to* $\varepsilon(\lambda)$*, then there exists a PPT hard-core predictor,* $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$*, with the probability* $\varepsilon'(\lambda)/2$ *(over the choice of* N, v^2*, and* ξ_1, ξ_2*), can predict the value of* $B_r^+(|uw|)$ *with advantage* $\varepsilon'(\lambda)/4$*, where* u *is the unique square root residue of* v^2*,* w *is determined by* v^2 *and* ξ_1, ξ_2*, and* $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)}{\ell_{\mathrm{K}}}$.

Proof. By Lemma 7, M has $\varepsilon'(\lambda)$-advantage in distinguishing $B_r^+(|uw|)$ from a random bit b. Then for at least $\varepsilon'(\lambda)/2$ fraction of the choices of N, v^2, and ξ_1, ξ_2, M has $\varepsilon'(\lambda)/2$-advantage in distinguishing $B_r^+(|uw|)$ from the random bit b . So it is straightforward that $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$ can predict the value of $B_r^+(|uw|)$ with advantage $\varepsilon'(\lambda)/4$.

Reduce to the factoring algorithm $\mathrm{A}(N)$

Lemma 9. *If there exists a PPT adversary* M *such that* $\mathrm{Adv}^{\mathrm{CCA}}_{\mathrm{KEM,M}}(\lambda)$ *equals to* $\varepsilon(\lambda)$*, then there exists a PPT algorithm* A *factoring* N *with success probability* $\Omega(\varepsilon'(\lambda)^2)$*, where* $\varepsilon'(\lambda)$ *equals to* $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}^{\mathrm{TCR}}_{\mathrm{H,M}}(\lambda)}{\ell_{\mathrm{K}}}$.

Proof. From lemma 2, it is enough to prove that there exists another adversary $\mathrm{A}'(N, v^2)$, using M as oracle, is able to compute the unique quadratic residue u such that $u^2 = v^2$ with probability $\Omega(\varepsilon'(\lambda)^2)$, where v^2 is a uniform element of G.

On input (N, v^2), A′ is defined as follows:

1. Choose ξ_1, ξ_2 uniformly from $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$.
2. Compute $w = (v^{2^{2\ell_{\mathrm{K}}}})^{\xi_1\xi_2}(v^{2^{\ell_{\mathrm{K}}}})^{\xi_1+\xi_2}$.
3. Invoke the Goldreich-Levin reconstruction algorithm, $R(1^\lambda)$:
   (a) Whenever asked for $B_{r_i}(z)$, invoke $\mathrm{D}'_{N,\xi_1,\xi_2,v^2}$ on input $< r_i >$, and give its output as an answer. (Recall that $\mathrm{D}'_{N,\xi_1,\xi_1,v^2}$ invokes M and answers its queries).
   (b) Denote by z the output of R.
4. Compute $u' = zw^{-1}$. Given that R outputs the correct value, i.e., $z = |uw|$, then $z = uw$ or $z = -uw$.

The successful probability of A′**:** Since with the probability $\varepsilon'(\lambda)/2, \mathrm{D}'_{N,\xi_1,\xi_2,v^2}$ predicts the value of $B_r^+(|uw|)$ with advantage $\varepsilon'(\lambda)/4$, then by Goldreich-Levin theorem [12], we have that R retrives the value of $|uw|$ with probability at least $\Omega(\varepsilon'(n)^2)$. Given that R outputs the correct value, with probability $1/2$, $u' = u$. Therefore, A′ compute u with probability $\Omega(\varepsilon'(n)^2)$, as required.

References

1. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on Computing 15(2), 364–383 (1986)
2. Blum, M., Goldwasser, S.: An probabilistic public key encryption scheme which hides all partial information. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 289–299. Springer, Heidelberg (1985)

3. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
4. Cramer, R., Hofheinz, D., Kiltz, E.: A Twist on the Naor-Yung Paradigm and Its Application to Efficient CCA-Secure Encryption from Hard Search Problems. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 146–164. Springer, Heidelberg (2010)
5. Cash, D.M., Kiltz, E., Shoup, V.: The twin diffie-hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
6. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
7. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
8. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003)
9. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: Proceedings of the 23rd ACM Symposium on Theory of Computing, pp. 542–552. IEEE Computer Society Press, Los Alamitos (1991)
10. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22, 644–654 (1976)
11. ElGama, T.: A public key cryptosystem and a signature scheme based on discrete loga- rithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)
12. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp. 25–32. ACM Press, New York (1989)
13. Groth, J.: Cryptography in subgroups of Z_n^*. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 50–65. Springer, Heidelberg (2005)
14. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)
15. Hanaoka, G., Kurosawa, K.: Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
16. Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
17. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
18. Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009)
19. Kiltz, E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)

20. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
21. Yehuda Lindell, A.: Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 241–254. Springer, Heidelberg (2003)
22. McCurley, K.: A Key Distribution System Equivalent to Factoring. Journal of Cryptology 1(2), 95–105 (1988)
23. Naor, M., Reingold, O., Rosen, A.: Pseudo-random functions and factoring. SIAM Journal on Computing 31(5), 1383–1404 (2002)
24. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing 2008, pp. 187–196. ACM, New York (2008)
25. Rabin, M.O.: Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology (January 1979)
26. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
27. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
28. Shoup, V.: Using Hash Functions as a Hedge against Chosen Ciphertext Attack. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 275–288. Springer, Heidelberg (2000)

A Proof of Some Properties and Lemmas

Property 1. Let h be a uniform element of Z_N^*, $P_B = \prod_{1<p<B,\ \mathrm{p\ is\ prime}} p$, and $g = h^{P_B}$. Then, g is a uniform element of G.

Proof. Notice that order of h is one of the factors of $2pqp'q'$, and $2pq|P_B$, $\gcd(p'q', P_B) = 1$. Then the order of g must be one of the factors of $p'q'$. Thus g lies in the unique subgroup of order $p'q'$, G. On the other hand, for every element g of G, there must exists an element h belongs to Z_N^*, such that $g = h^{P_B}$ (Reason: since $\gcd(p'q', P_B) = 1$, then there exists $a, b \in Z$ such that $aP_B + bp'q' = 1$. Then $g = g^{aP_B+bp'q'} = (g^a)^{P_B}$). Therefore, $G = \{g|g = h^{P_B},\ h \in Z_N^*\}$. Observe that the mapping $f(x) = x^{P_B}$ is a $4pq$ to 1 mapping from Z_N^* to G, that is, for every z in G, there exists exactly $4pq$ solutions in Z_N^* such that $z = x^{P_B}$(Reason: We firstly consider the set $X_I = \{x|x \in Z_N^*,\ x^{P_B} = 1\}$. Let the number of X_I,$|X_I|$, be m. Let y' be an element of G, x' be an element of Z_N^* such that $y' = (x')^{P_B}$. For every element x of X_I, it must be that $y' = (x'x)^{P_B}$. For every x does not belong to X_I, it must be that $y' \neq (x'x)^{P_B}$. So it must be that for every z of G, the equation $z = x^{P_B}$ has exactly m solutions in Z_N^*. Since the number of G ,$|G|$, equals to $p'q'$. So it must be $mp'q' = 4pqp'q'$. So m equals to $4p'q'$). When x is chosen uniformly from Z_N^*, $z = x^{P_B}$ is uniformly distributed in G. So g is a uniformly random element of G.

Property 2. With probability $1 - O(2^{-m'(\lambda)})$, a uniform element in G is a generator of G.

Proof. The order of G is $p'q'$, there are $(p'-1)(q'-1)$ elements of order $p'q'$. So with probability $1-(p'-1)(q'-1)/p'q' = 1 - O(2^{-m'(\lambda)})$,a uniform element in G is a generator of G.

Property 3. Any element z of G is a quadratic residue, the unique quadratic residue u such that $u^2 = z$ lies in G.

Proof. From property 1 and 2, with overwhelming probability, $g = h^{P_B} = (h^{P'_B})^2$ is a generator of G, where $P'_B = \prod_{2<p<B,\ \text{p is prime}} p$. Obviously, g is a quadratic residue. So any element of $G =< g >$ is a quadratic residue. Since N is a Blum integer, then the equation $u^2 = z$ has unique solution in QR_N. Furthermore, the order of G, $p'q'$, is odd, then $gcd(2, p'q') = 1$, so $2^{-1} \bmod p'q'$ exists. Since z lies in G, then $z^{2^{-1} \bmod p'q'}$ lies in G. Finally, since $(z^{2^{-1} \bmod p'q'})^2 = z$, then $z^{2^{-1} \bmod p'q'}$ which lies in G is the unique solution of the equation $u^2 = z$.

Property 4. For any element z of G, then the unique quadratic residue u such that $z = u^{2^k}$ lies in G for any $k \in Z^+$.

Proof. Since $gcd(2, p'q') = 1$ and so $gcd(2^k, p'q') = 1$, then $2^{-k} \bmod p'q'$ exists, thus $z^{2^{-k} \bmod p'q'}$ lies in G and is a quadratic residue. Since N is a Blum integer, then squaring in quadratic residue group, QR_N, is a permutation. Then $u^2 = z$ has unique solution in QR_N. By induction, $z = u^{2^k}$ has unique solution in QR_N. Since $(z^{2^{-k} \bmod p'q'})^{2^k} = z$, then $u = z^{2^{-k} \bmod p'q'}$ is the unique quadratic residue satisfies $z = u^{2^k}$ and lies in G.

Lemma 5. If $A, B \in QR_N^+$, then $A^2 = B^2 \bmod N \Leftrightarrow A = B$. More generally, $A^{2^k} = B^{2^k} \bmod N$ $(k \in Z^+) \Leftrightarrow A = B$.

Proof. The necessity is obvious. We only prove the sufficiency.

Since $A \in QR_N^+$, then there exists $u \in Z_N^*$ such that $A = u^2$ if $0 \le u^2 < N/2$ or else $A = -u^2$. Similarly, there exists $v \in Z_N^*$ such that $B = v^2$ if $0 \le v^2 < N/2$ or else $B = -v^2$. Now

$$A^2 = B^2 \bmod N \Rightarrow u^4 = v^4 \bmod N$$

From the uniqueness of square quadratic root (recall that N is a Blum integer), we have $u^2 = v^2 \bmod N$.

So if $0 \le u^2 < N/2$ then $A = u^2 = v^2 = B$; else if $-N/2 < u^2 < 0$ then $A = -u^2 = -v^2 = B$.

The general case can be proved by induction.

Lemma 6. If $A, B \in QR_N \cup QR_N^+$, then $|AB| = \big||A||B|\big|$.

Proof. If $A \in QR_N \cup QR_N^+$, then exists u such that $A = u^2$ or $A = -u^2$. Similarly, if $B \in QR_N \cup QR_N^+$, then exists v such that $B = v^2$ or $B = -v^2$. On one hand, we have $|AB| = |u^2v^2|$ or $|AB| = |-u^2v^2|$. So, we have $|AB| = |\pm u^2v^2| = |u^2v^2|$. On the other hand, we also have $||A||B|| = | \pm u^2v^2| = |u^2v^2|$ since $|A|$ equals to u^2 or $-u^2$ and $|B|$ equals to v^2 or $-v^2$. The Lemma follows since both $|AB|$ and $||A||B||$ equal to $|u^2v^2|$.