Chosen Ciphertext Secure Encryption over Semi-smooth Subgroup

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

1: State Key Laboratory of Information Security, Beijing, 100049, China
2: Guangdong Ocean University, Zhanjiang, 524088, China

Abstract. In this paper, we propose two public key encryption schemes over the semi-smooth subgroup introduced by Groth (TCC 2005). Both schemes are proved secure against chosen ciphertext attacks under the factoring assumption. Since the domain of exponents is much smaller, both our schemes are significantly more efficient than the scheme presented by Hofheiz and Kiltz in Eurocrypt 2009.

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 [7]. Their construction was later generalized to hash proof system [8]. However, the Cramer-Shoup encryption scheme and all its variants [21, 17] inherently rely on the decisional assumption, e.g., the Decisional Diffie-Hellman (DDH) assumption, Decisional Composite Residuosity (DCR) assumption, and Decisional Quadratic Residuosity (DQR) assumption. In [25], Peirk and Brent Waters proposed a general framework of constructing CCA secure encryption from the lossy trapdoor function. In [28], Rosen and Segev proposed a general way under the correlated inputs function. However, all the concrete constructions of lossy trapdoor function and correlated inputs function are also based on the Decisional assumption.

Canetti, Halevi and Katz [4] proposed the first practical public key encryption under a computational assumption, namely the Bilinear Diffie-Hellman assumption. Cash, Kiltz and Shoup [6] proposed a practical CCA secure scheme under the Computational Diffie-Hellman assumption. Hanaoka and Karosawa proposed a more efficient CCA secure scheme under the CDH assumption [16]. Very recently, Haralambiev et al., further improved the result of CKS08 and HK08 under CDH assumption. Notably, Hofheiz and Kiltz proposed a practical CCA secure PKE under the factoring assumption in 2009 [18]. In [5], Cramer, Hofheiz and Kiltz proposed a framework based on the computational assumption, which yields practical CCA secure scheme under the CDH assumption and the RSA type assumption.

The Hofheiz-Kiltz 2009 scheme (HK09) [18] is constructed from the Blum-Goldwasser encryption [2], which itself is based on the Rabin encryption [26] and Blum-Blum-Shub (BBS) pseudorandomness 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 in the standard factoring assumption (instead of the related decisional assumption). The encryption is about two full exponentiation in Z_N^*, the decryption is about one full exponentiation in Z_N^* plus some other multiplications.

Though very elegant, in the original HK09, the exponent is chosen from $[(N-1)/4]$, which means it has almost the same bits length as N. For the secure lever of 80, the bits length of N, ℓ_N, needs to be chosen at least as 1024. So the exponentiation operation of HK09 is fairly time consuming.

A nature question is to ask how to construct more efficient CCA secure public key encryption scheme(s) under the factoring assumption.

In this paper, we present two efficient schemes over some special subgroup called semi-smooth subgroup of Z_N^*. Both the schemes are proved secure against chosen ciphertext attacks under the factoring assumption. Since the domain of exponents is much smaller, both the schemes are significantly more efficient than HK09.

The semi-smooth subgroup is introduced by Groth in TCC 2005 [14]. This subgroup is defined for some special modulus N that is called semi-smooth. More precisely, this type $N = PQ$ is chosen such that $P = 2p'p_1p_2 \cdots p_{t_p} + 1$, $Q = 2q'q_1q_2 \cdots q_{t_q} + 1$, where $p_1, p_2, \cdots, p_{t_p}$,$q_1, p_2, \cdots, q_{t_q}$ are distinct odd primes smaller than some small bound B. The only subgroup, G, of Z_N^* of order $p'q'$ is called the semi-smooth subgroup. The element of this subgroup can be efficiently sampled without the factors of the modulus.

One of the schemes we constructed is a variant of HK09 over semi-smooth subgroup, which we would like to refer it as the HK09 instantiation. The domain of exponent in this variant is replaced with $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$. For secure lever of 80, we can choose the parameters as $\ell_{p'} = \ell_{q'} = 160$, $\ell_N = 1024$, and $B = 2^{15}$. So the bits length of the exponent is 400. Here, we describe the idea in a high level. Recall that, the HK09 proof can be classed in two steps: first, they proved the BBS generator is pseudorandom; then, they black box reduced the CCA security of the encryption to the security of the BBS generator. The BBS generator can be proved pseudorandom if the Quadratic Residuosity assumption in Z_N^* holds by using the the hybrid argument to the hard-core predictor. But the Quadratic Residuosity assumption can be reduced to the factoring assumption in Z_N^*. In original HK09, one of the public key, g, is chosen uniformly from QR_N. We observe that the security proof of the second step in HK09 goes through if the domain of the randomizing encryption exponent, μ, satisfies two conditions: the first one is that the domain should be easily sampled without knowing the order of g or the factors of N; the second one is that if μ is uniformly chosen from this domain, then the distribution of μ mod $ord(g)$ is statistically close to the uniform distribution of $[ord(g)]$. But for the general Blum integer, since g is chosen randomly from QR_N. Different g may have different orders, even the bits size of $ord(g)$ is not necessarily fixed. So we have to use a common large domain, $[(N-1)/4]$. If we instead choose g as the generator of a fixed subgroup of QR_N, we can choose the domain as $[2^{\ell_{ord(g)}+\lambda}]$. It is not difficulty to verify that this domain does satisfy the above two conditions if $\ell_{ord(g)}$ is given. If the Quadratic Residuosity is hard to compute in this subgroup, the resulting scheme is indeed CCA secure. The problem is that, for the general Blum integer, this subgroup is not efficiently sampled without the factors of the modulus. But if the subgroup is not efficiently sampled, we find it difficulty to reduce the Quadratic Residuosity assumption to the more standard factoring assumption. Now, we turn to the case the modulus N is semi-smooth such that Z_N^* has a unique semi-smooth subgroup G. We find that the Quadratic Residuosity assumption does can be reduced to the standard factoring assumption by using the fact that this subgroup can be efficiently sampled without knowing the factors of the modulus N. In particular, in this variant, we choose g uniformly from the semi-smooth subgroup G instead of QR_N, set the domain as $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$. With overwhelming probability, g will be a random generator of G. So the order of g is equal to $p'q'$ and is fixed when N is given. And it is not difficult to prove that this domain is easily sampled and satisfies the second condition, ie., if μ is chosen from this domain, then the distribution of μ mod $p'q'$ is close to the uniform distribution of $[p'q']$. Similarly, we can replace the domain of the private decryption exponent ρ with $[2^{\ell_{p'}+\ell_{q'}+\lambda}]$.

Another scheme we construct is actually based on the ElGamal encryption over semi-smooth group [12]. Compared to the HK09 instantiation, the decryption is more efficient, though the encryption is less efficient. It is well known that the one-wayness of ElGamal encryption over the Composite can be reduced to the factoring assumption [23]. Using the same idea as mentioned above, one-wayness of ElGamal encryption over semi-smooth group can also be reduced to the factoring assumption. We observe that by using the proof technique of [24] it can be proved that $BBS_r(g^{xy})$ is pseudo-random even g^x, g^y are given. So we can turn the one-wayness secure ElGamal encryption over the semi-smooth subgroup into a indistinguishability secure scheme. Surprisingly, we prove that we only need add a group element in G as the check ciphertext to make the scheme CCA secure. The resulting scheme is very related to the HK09 instantiation, though has obvious difference. In the actually proof, we do not black box reduce CCA security of this scheme to the pseudo-random of $BBS_r(g^{xy})$. Instead, we prove the CCA security of this scheme *and* the pseudo-randomness of $BBS_r(g^{xy})$ *simultaneously.*

For easy of presentation, next, we focus on the CCA secure key encapsulation mechanism. Combining it with the CCA secure data encapsulation mechanism, it is easy to obtain the full CCA secure public key encryption [30].

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 indistinguishability 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 the 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 C 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

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

2.2 Target collision resistant hash function

Informally, we say that a function $H : X \to Y$ 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 $H(x') = H(x)$.

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

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

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

3 Semi-smooth Subgroup

In Groth 05 [14], the author introduces 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 probability polynomial-time algorithm such that given only N, the random output of $\mathrm{IGen}(1^\lambda)$, can factoring N with non-negligible probability.

In [14], 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 and lemmas that will be used in the proof of the CCA scheme in section 4 and 5.

Property 1. Element of semi-smooth subgroup can be uniformly and efficiently sampled without the factors of modulus.

Proof. One can choose h uniformly from Z_N^*, set $P_B = \prod_{1<p<B,\ \mathrm{p\ is\ prime}} p$, and $g = h^{P_B}$. 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 overwhelming probability, a randomly sampled element 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 randomly sampled element is a generator of G.

Property 3. Any element v of G is a quadratic residue, the unique quadratic residue u such that $u^2 = v$ 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 $u^2 = v$ has unique solution in QR_N. Furthermore, the order of G, $p'q'$, is odd, then $gcd(2, p'q') = 1$, so $2^{-1}\text{mod } p'q'$ exists. Since v lies in G, then $v^{2^{-1}\text{mod } p'q'}$ lies in G. Finally, since $(v^{2^{-1}\text{mod } p'q'})^2 = v$, then $v^{2^{-1}\text{mod } p'q'}$ which lies in G is the unique solution of the equation $u^2 = v$.

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 $2^{-k}\text{mod } p'q'$ exists, thus $z^{2^{-k}\text{mod } p'q'}$ 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 $(z^{2^{-k}\text{mod } p'q'})^{2^k} = z$, then $u = z^{2^{-k}\text{mod } p'q'}$ is the unique quadratic residue satisfies $z = u^{2^k}$ and lies in G.

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 μ mod$p'q'$ and $(\mu+k)$mod$p'q'$ are statistically close to the uniformly distribution of $[p'q']$, both g^{μ} and $g^{\mu+k}$ are statistically close to the uniformly distribution of G.*

Proof. Write $2^{\ell_{p'}+\ell_{q'}+\lambda}$ as $k_1p'q' + k_2$ over Z, where $0 < k_2 < p'q'$. If μ is uniformly chosen from $[k_1p'q']$, then both μ mod $p'q'$ 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 $[k_1p'q']$ with probability $k_1p'q'/2^{\ell_{p'}+\ell_{q'}+\lambda} = 1 - k_2/2^{\ell_{p'}+\ell_{q'}+\lambda} \geq 1 - O(2^{-\lambda})$.

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

Lemma 2. *Let G be the semi-smooth subgroup of Z_N^*, v is a uniformly chosen element of G, if there exists an adversary A can compute the unique quadratic residue u such that $u^2 = v$ 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 $P'_B = \prod_{2<p<B,\ \text{p is prime}} p$, and $h' = h^2$, $v = h'^{P'_B} (= h^{P_B})$. So v is a uniform element of G. If A can compute u such that $u^2 = v$, 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[13], it is easy to see that given $v = u^2$ 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 $z = u^{2^{\ell_K}}$, $BBS_r(u) = (B_r(u), B_r(u^2), \cdots, B_r(u^{2^{\ell_K-1}}))$, r is a random element with bits-size ℓ_N.*

4 The Instantiation of HK09 over Semi-smooth Subgroup

4.1 Scheme description

$Gen(1^\lambda)$: Run IGen(1^λ) to get the modulus N. Then, Gen chooses a target-collision resistant hash function $H : Z_N \to [2^{\ell_H} - 1]$. 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 $X = g^{\rho 2^\nu}(\nu = \ell_H + \ell_K)$. The public key is $PK = (N, g, X, r, H)$, and the private key is $SK = \rho$.

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

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

Set the ciphertext as $C = (R, S)$. Compute the encapsulation key as $K = BBS_r(g^{\mu 2^{\ell_H}})$.

$Dec(SK, C)$: 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 $t = H(R)$, verifies:

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

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

$$2^c = gcd(t, 2^\nu) = at + b2^\nu.$$

Then Dec computes

$$T = ((S^2)^a \cdot (R^2)^{b-a\rho})^{2^{\ell_H - c - 1}}$$

and $K = BBS_r(T)$, outputs K.

Correctness: the correctness proof can be referred to HK09.

Efficiency The encapsulation and decapsulation 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 ℓ_{exp} equals to $\ell_{p'} + \ell_{q'} + \lambda$. For typical parameter, $\ell_N = 1024$, $\ell_{p'} = \ell_{q'} = 160$, $\lambda = 80$. So, the encapsulation requires 1480 multiplications, the decapsulation requires 1440 multiplications.

Remark N, H, r, g can be set as global system parameters that can be shared by many parties. Instead, if N and g are not the global system parameters, then the decapsulation exponent ρ can be chosen from the even smaller domain $[p'q']$, and the decapsulation efficiency can be further improved by using then Chinese Remainder Theorem.

4.2 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 reducing 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 $BBS_r(u)$ 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 $H : Z_N \to [2^{\ell_H} - 1]$, 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

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

The public key is set as $PK = (N, g, X, r, H)$. 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

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

The challenge ciphertext is $C^* = (R^*, S^*)$, the challenge key is V. Note that, as in the IND-CCA2 game, if V is $BBS_r(u)$, 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 $t = H(R)$.

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

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

Reject it if it not. If it holds, then compute $a', b', c' \in Z$ such that

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

Then compute

$$T = ((S^2)^{a'} \cdot (R^2)^{b' - a'\beta})^{2^{\ell_H - c - 1}}$$

and $K = BBS_r(T)$, output K.

The correctness of the decryption simulation for $t \neq t^*$ can be referred to the proof of original HK09.

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

$$(S^2)^{2^\nu} = (R^2)^{\beta 2^\nu} = (R^{*2})^{\beta 2^\nu} = S^{*2}.$$

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 ciphertext.

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

D outputs what A outputs.

D can use A as a oracle to distinguish whether V is $BBS_r(u)$ or a uniform bits string.

5 Another Scheme over Semi-smooth Subgroup Secure under Factoring Assumption

In this section, we will construct another efficient KEM over semi-smooth subgroup and prove it is chosen ciphertext secure under the factoring assumption. This scheme is implicitly instantiated over the signed quadratic residues group [19].

5.1 Signed quadratic residues

The signed quadratic residues, 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.

The following lemmas will be used in the security and correctness proof for the KEM in next subsection.

Lemma 4. *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 $0 \leq u^2 < N/2$ or else $A = -u^2$. Similarly, there exists $v \in Z_N^*$ such that $B = v^2$ if $0 \leq v^2 < N/2$ 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 $0 \leq u^2 < N/2$ 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 5. *If $A, B \in QR_N \cup QR_N^+$, then $|AB| = ||A||B||$.*

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 $|u^2v^2|$.

5.2 Scheme description

$Gen(1^\lambda)$: Run $\mathrm{IGen}(1^\lambda)$ to get the modulus N. Then, choose a target-collision resistant hash function $H : Z_N \to [2^{\ell_H} - 1]$. 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 $X = g^{\rho 2^\nu} (\nu = \ell_H - 1)$ and $X' = g^{\rho'}$. The public key is $PK = (N, g, X, X', r, H)$, and the private key is $SK = (\rho, \rho')$.

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

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

$$K = BBS_r(T) \overset{\text{def}}{=} (B_r(|T|), B_r(|T^2|), \cdots, B_r(|T^{2^{\ell_K - 1}}|)).$$

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 $t = H(R)$, verifies:

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

If it holds, then Dec computes

$$T = |R^{\rho'}|, \quad K = BBS_r(T).$$

The difference between HK09 instantiation and this scheme: in HK09 instantiation presented in section 4, the first ciphertext is $g^{u 2^{\ell_H + \ell_K}}$, but in this scheme, the first ciphertext is $|g^{u 2^{\ell_H - 1}}|$. The encapsulated key of the former is $BBS_r(R^{\frac{1}{2^{\ell_K}}})$, whereas that of the latter is $BBS_r(|R^{\rho'}|)$. The verification part in HK09 instantiation is $S = |(g^t X)^\mu|$, whereas it is $S = |(X'^t X)^\mu|$ in this scheme. The decapsulation for key in this scheme is simpler than in HK09 instantiation.

Correctness: If R and S is 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: If we choose $\ell_{q_1} = \ell_{p_1} = 160$, $\lambda = 80$, then $\ell_\rho = \ell_{\rho'} = \ell_{exp} = 400$. We assume $\ell_H = 80$. The encapsulation requires $4.5\ell_{exp} + 2.5\ell_H = 2000$ multiplications. The decapsulation requires $1.5 \times 1.2\ell_{exp} + 2.5\ell_H = 920$ multiplications (Notice that $g^{\rho'}$ and g^ρ use the same base g and can be computed with about 1.2 exponentiations).

5.3 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), firstly, the pseudo-randomness of the BBS generator $BBS_r(u)$ is reduced to the factoring assumption, then, the CCA security is black box reduced to the pseudo-randomness of the BBS generator $BBS_r(u)$. But, in this scheme, if we directly reduce the CCA security to the pseudo-randomness of $BBS_r(g^{\mu\rho'})$ (even g^{μ} and $g^{\rho'}$ is given), then the simulator could not answer DDH oracle that 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 $BBS_r(g^{\mu\rho'})$ *simultaneously.* Adapting the proof idea of [24], we firstly reduce the security (both the CCA security of this scheme and the pseudo-randomness of $BBS_r(g^{\mu\rho'})$) 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 $\rho' 2^{\ell_K} \bmod p'q'$, so he could compute $|R^{\rho' 2^{\ell_K}}|$. The distinguisher does not directly verify the equation $|S^{2^{\nu}}| = |R^{\rho' t + \rho 2^{\nu}}|$, instead, he verify a equivalent equation $|S^{2^{\nu+\ell_K}}| = |R^{\rho' 2^{\ell_K} t + \rho 2^{\nu+\ell_K}}|$, which is why we implicitly use the signed group. Then, by using the technique of HK09, the distinguisher is able to efficiently compute $|(R^{\rho' 2^{\ell_K}})^{\frac{1}{2^{\ell_K}}}|$, which equals to the encapsulated key $|R^{\rho'}|$.

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

Reduce to the hard-core distinguisher $D(v^2, N, r, \alpha)$
As in [24], w below is actually determined by v^2 and ξ_1, ξ_2, where ξ_1, ξ_2 is actually chosen by a factoring algorithm. So w is fixed in advance and not determined by D's internal coin tosses. But for the moment, we assume $\overrightarrow{\xi}$ is chosen by D, so w depends on the D's internal coin.

Lemma 6. *If there exists a PPT adversary M such that $Adv^{\mathrm{CCA}}_{\mathrm{KEM,M}}(\lambda)$ equals to $\varepsilon(\lambda)$, then there exists a PPT adversary $D(v^2, N, r, \alpha)$ 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 $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}^{TCR}_{H,M}(\lambda)}{\ell_K}$.*

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

Prepare the public key: Choose a target-collision resistant hash function $H : Z_N \rightarrow [2^{\ell_H} - 1]$. Randomly choose $J = k$ from $\{0, 1, \cdots \ell_K - 1\}$. 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 $s = 2\ell_K - k$, $g = v^{2^s} \bmod N$, and $a_i = (\xi_i + 2^{-\ell_K}) \bmod p'q' \ \ (i = 1, 2)$. Set $X' = g^{a_1} = g^{\xi_1 + 2^{-\ell_K} \bmod p'q'}$ (implicitly define $\rho' = (\xi_1 + 2^{-\ell_K}) \bmod p'q'$). Set $B = g^{a_2} = g^{\xi_2 + 2^{-\ell_K} \bmod p'q'}$ (implicitly define $B = g^{\mu^* 2^{\nu}}$). Set $t^* = H(|B|)$. 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, H).

(D is able to efficiently compute $X' = g^{a_1} = g^{\xi_1 + 2^{-\ell_K} \bmod p'q'} (= (v^2)^{\ell_K - k} (v^2)^{\xi_1 (2\ell_K - k)})$ since $\ell_K > k$ and v^2 is given. Similar, B 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:

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

And the challenge key is set as

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

Define w: We define $w = (v^{2^{2\ell_K}})^{\xi_1\xi_2}(v^{2^{\ell_K}})^{\xi_1+\xi_2}$. 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, then $g^{2^k a_1 a_2} = uw$.

Proof.

$$\begin{aligned} g^{2^k a_1 a_2} &= g^{2^k(\xi_1+2^{-\ell_K})(\xi_2+2^{-\ell_K})} = g^{2^k(\xi_1\xi_2+(\xi_1+\xi_2)2^{-\ell_K}+2^{-2\ell_K})} \\ &= v^{2^{2\ell_K}(\xi_1\xi_2+(\xi_1+\xi_2)2^{-\ell_K}+2^{-2\ell_K})} = uw. \end{aligned}$$

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

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 $j = 1, \ldots \ell_K - k - 1$. Furthermore, since D knows v^2 and w^2, so he is able to compute $(uw)^{2^j}$ for $j = 1, \ldots \ell_K - k - 1$, as required.

From claim 1 and 2, we 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 $t = H(R)$. If $t \neq t^*$, D verifies if

$$(S^{2^{\ell_K}})^{2^\nu} = (R^{(2^{\ell_K}\xi_1+1)})^{t-t^*}(R^{2^{\ell_K}})^{\beta 2^\nu}$$

(Note that the right side equals to

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

From Lemma 4, we know that verifying $|S^{2^\nu}| = |R^{\rho' t + \rho 2^\nu}|$ is equivalent to verifying $(S^{2^{\ell_K}})^{2^\nu} = (R^{2^{\ell_K}})^{\rho' t + \rho 2^\nu}$, as required.)

If it holds, D computes $a', b', c' \in Z$ such that:

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

(Note that since both t and t^* are smaller than 2^{ℓ_H}, then $c' \leq \ell_H - 1 = \nu$)
Then D computes

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

$$\begin{aligned} (T &= |\{((SR^{-\beta})^{a'} R^{b'(2^{\ell_K}\xi_1+1)})^{2^{\nu+\ell_K}}\}^{2^{-\ell_K-c'}}| = |\{(SR^{-\beta})^{a' 2^{\nu+\ell_K}}(R^{(2^{\ell_K}\xi_1+1)})^{b' 2^{\nu+\ell_K}}\}^{2^{-\ell_K-c'}}| \\ &= |\{(R^{(2^{\ell_K}\xi_1+1)})^{a'(t-t^*)+b' 2^{\nu+\ell_K}}\}^{2^{-\ell_K-c'}}| = |\{(R^{(2^{\ell_K}\xi_1+1)})\}^{2^{-\ell_K}}| = |R^{\rho'}|) \end{aligned}$$

(The third step follows from $(SR^{-\beta})^{2^{\nu+\ell_K}} = (R^{(2^{\ell_K}\xi_1+1)})^{t-t^*}$, the fourth step follows from $2^{c'} = a'(t - t^*) + b' 2^{\ell+\ell_K}$, the last step follows from $\rho' = (\xi_1 + 2^{-\ell_K}) \bmod p'q'$. Correctness follows.)

If $t = t^*$, D rejects the query ciphetext (R, S). (If $H(R) = t = t^* = H(R^*)$ 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

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

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 element of G, and squaring is 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 $\mathrm{Adv}_{H,M}^{TCR}(\lambda)$.

For convenience, we denote some hybrid experiments $H^J (J = 0, \cdots, \ell_K)$ 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 $\ell_K - J$ 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^{ℓ_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 $k \in \{0, 1, \cdots, \ell_K\}$, and still denote real H^k as H^k. So the advantage of D is :

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

This completes Lemma 6.

Reduce to the hard-core predictor D'_{N,ξ_1,ξ_2,v^2}

Since D defined in Lemma 6 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 [13]. 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 D'_{N,ξ_1,ξ_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 D'_{N,ξ_1,ξ_2,v^2}. So it is possible to use D'_{N,ξ_1,ξ_2,v^2} as an oracle to reconstruct $|uw|$.

Lemma 7. *If there exists a PPT adversary M such that $Adv_{\mathrm{KEM,M}}^{\mathrm{CCA}}(\lambda)$ equals to $\varepsilon(\lambda)$, then there exists a PPT hard-core predictor, D'_{N,ξ_1,ξ_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 $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}_{H,M}^{TCR}(\lambda)}{\ell_K}$.*

Proof. By Lemma 6, 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 D'_{N,ξ_1,ξ_2,v^2} can predict the value of $B_r(|uw|)$ with advantage $\varepsilon'(\lambda)/4$.

Reduce to the factoring algorithm $A(N)$

Lemma 8. *If there exists a PPT adversary M such that $Adv_{\mathrm{KEM,M}}^{\mathrm{CCA}}(\lambda)$ 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 $\varepsilon'(\lambda) = \frac{\varepsilon(\lambda) - O(2^{-\lambda}) - \mathrm{Adv}_{H,M}^{TCR}(\lambda)}{\ell_K}$.*

Proof. On input $< N >$, A is defined as follows:

1. (a) Choose h uniformly at random from Z_N^*, set $P'_B = \prod_{2<p<B,\ \mathrm{p\ is\ prime}} p$, and $h' = h^2$,$v = h^{P'_B}$, and compute $v^2 \bmod N$ (So v^2 is a uniformly random element of G).
   (b) Choose each ξ_1, ξ_2 uniformly from $[2^{\ell_{p'} + \ell_{q'} + \lambda}]$.
2. Compute $w = (v^{2^{2\ell_K}})^{\xi_1 \xi_2} (v^{2^{\ell_K}})^{\xi_1 + \xi_2}$.
3. Invoke the Goldreich-Levin reconstruction algorithm, $R(1^\lambda)$:
   (a) Whenever asked for $B_{r_i}(z)$, invoke D'_{N,ξ_1,ξ_2,v^2} on input $< r_i >$, and give its output as an answer. (recall that D'_{N,ξ_1,ξ_1,v^2} invokes M and answers its queries).
   (b) Denote by z the output of R.
4. Compute $|u| = |z|w^{-1}||$. Given that R outputs the correct value (i.e., $z = |uw|$) then $|u^2| = |v^2| \bmod N$. Compute a, b over Z such that $aP'_B + 2b = gcd(P'_B, 2) = 1$, set $\widetilde{h} = |u^a h'^b|(= ||u|^a h'^b|)$. If $\widetilde{h} \neq \pm h$, then A outputs $\gcd(\widetilde{h} - h, N)$.

The successful probability of A: Since with probability $\varepsilon'(\lambda)/2$, D'_{N,ξ_1,ξ_2,v^2} predicts the value of $B_r(|uw|)$ with advantage $\varepsilon'(\lambda)/4$, then by Goldreich-Levin theorem [13], we have that R retrives the value of $|uw|$ with probability at least $\Omega(\varepsilon'(n)^2)$. Note that both $|uw|$ and $|w|$ belong to G^+, therefore $|u|$ must also belongs to G^+. If $|u^2| = |v^2| \bmod N$, then $|u^2| = |h'^{P'_B}| \bmod N$. Thus $\widetilde{h} = |u^a h'^b|$ belongs to QR_N^+ and satisfies $|\widetilde{h}^2| = |h'| = |h^2|$. So $\widetilde{h}^2 = h^2$ or $\widetilde{h}^2 = -h^2$. But since N is a Blum integer, then -1 is not quadratic residue. It easy to see that $-h^2$ is not quadratic residue. So it must be that $\widetilde{h}^2 = h^2$. But with probability 1/2, $\widetilde{h} \neq \pm h$, thus $\gcd(\widetilde{h} - h, N)$ is a non-trivial factor of N. Therefore, A factors N 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 efficient probabilistic public-key encryption scheme which hides all partial information. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 289-302. Springer, Heidelberg (1985)
3. Boyen,X., Mei,Q., Waters, B. :Direct Chosen Ciphertext Security from Identity-Based Techniques.In 12th ACM Conference on Computer and Communications Security (CCS 2005), pages 320-329, ACM Press, 2005
4. Canetti,R., Halevi,S., and Katz,J.: Chosen-Ciphertext Security from Identity-Based Encryption[C]. Advances in Cryptology Eurocrypt 2004.Berlin:Springer-Verlag,2004: 207- 222.

5. 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. TCC 2010: 146-164
6. 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)
7. Cramer, R., Shoup, V.:A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. Advances in Cryptology -CRYPTO'98, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed.,Springer-Verlag, 1998.
8. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption, against adaptive chosen ciphertext attack. Advances in Cryptology - Eurocrypt'02, Lecture Notes in Computer Science Vol. 2332, Springer-Verlag, 2002.
9. 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)
10. Dolev,D., Dwork, C. and Naor, M. : Non-malleable cryptography. In ACM, editor, Proceedings of the 23rd ACM Symposium on Theory of Computing, pp. 542-552. IEEE Computer Society Press, 1991.
11. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22: 644-654 (1976)
12. 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)
13. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC, pp. 25C32. ACM Press, New York (1989)
14. Groth, J.: Cryptography in Subgroups of Z_n^*. In: Kilian, J.(ed.) Theory of Cryptography Conference 2005. LNCS, 3378, pp.50-65. Springer, Heidelberg (2005)
15. Haralambiev, K. Jager,T. Kiltz, E., Shoup,V.: Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. Public Key Cryptography 2010.
16. Hanaoka, G., Kurosawa, K.: Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS,5350, pp. 308-325. Springer, Heidelberg (2008)
17. Hofheinz, D., Kiltz,E.: Secure Hybrid Encryption from Weakened Key Encapsulation. Advances in Cryptology – CRYPTO 2007, pp. 553–571 LNCS 4622 (2007).Springer-Verlag.
18. Hofheinz, D., Kiltz,E.: Practical Chosen Ciphertext Secure Encryption from Factoring.In: Joux, A. (ed.) EUROCRYPT 2009. LNCS,5479,pp. 313-332.
19. Hofheinz, D., Kiltz,E.: The group of signed quadratic residues and applications. In: Halevi,S.(ed.)CRYPTO 2009. LNCS,5677, pp. 637-653.Springer, Heidelberg (2009)
20. Kiltz,E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Okamoto, T., Wang, X.(eds.) Public Key Cryptography 2007. LNCS, 4450, pp.282-297. Springer, Heidelberg (2007)
21. Kurosawa,K., Desmedt,Y.: A New Paradigm of Hybrid Encryption Scheme. CRYPTO 2004: 426-442
22. Yehuda Lindell: A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. EUROCRYPT 2003: 241-254
23. McCurley, K.: A Key Distribution System Equivalent to Factoring. Journal of Cryptology 1(2): 95-105 (1988)
24. Naor, M., Reingold, O., Rosen, A.: Pseudo-random functions and factoring. SIAM Journal on Computing 31(5), 1383-1404 (2002)
25. Peikert,C., Waters,B. : Lossy trapdoor functions and their applications. STOC 2008: 187-196
26. 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)
27. Rackoff, C., Simon, R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum,J.(eds.) CRYPTO 1991. LNCS,576, PP. 433-444. Springer, Heidelberg (1992)
28. Rosen,A., Segev,G. : Chosen-Ciphertext Security via Correlated Products. TCC 2009: 419-436.
29. Sahai, A.: Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. 40th IEEE Symposium on Foundations of Computer Science(FOCS), IEEE, pp. 543-553, 1999.
30. Shoup, V. :Using Hash Functions as a Hedge against Chosen Ciphertext Attack,EUROCRYPT 2000, pp.275-288 (2000)