CCA secure ElGamal encryption over an integer group where ICDH assumption holds

Gyu-Chol.Kim*, Jae-Yong.Sin, Yong-Bok.Jong

Department of Information Science and Technology, Kim Chaek University of Technology,
Pyongyang, Democratic People's Republic of Korea
*: Corresponding author, E-mail: kgc841110@star-co.net.kp

Abstract

In order to prove the ElGamal CCA (Chosen Ciphertext Attack) security in the random oracle model, it is necessary to use the group (i.e., ICDH group) where ICDH assumption holds. Until now, only bilinear group where ICDH assumption is equivalent to CDH assumption has been known as the ICDH group. In this paper, we introduce another ICDH group in which ICDH assumption holds under the RSA assumption. Based on this group, we propose the CCA secure ElGamal encryption. And we describe the possibility to speed up decryption by reducing CRT (Chinese Remainder Theorem) exponents in CCA secure ElGamal.

Keywords: ElGamal, CCA security, Interactive Computational Diffie Hellman problem, random oracle, RSA

1. Introduction

After the discovery of Diffie-Hellman key exchange protocol[1], ElGamal[4] and it's variants[7,10,12], which are public key encryption schemes based on Diffie-Hellman problem, have been developed and widely used. Meanwhile, using DH value itself to mask plaintext via multiplication is not recommended in modern ElGamal systems and it is recommended to hash DH value in order to derive the symmetric encryption key which is used to encrypt the plaintext in the semantically secure symmetric encryption (e.g., symmetric authenticated encryption).

CDH(Computational Diffie Hellman), DDH(Decisional Diffie Hellman) and ICDH(Interactive Computational Diffie Hellman) assumptions are basically used to prove the CCA security of modern ElGamal protocols. In the random oracle model, hashed ElGamal is proved to be CCA secure (i.e., to be semantically secure against Chosen Ciphertext Attack) under the ICDH assumption and twin ElGamal is proved to be CCA secure under the CDH assumption (more precisely, under the Interactive Twin Computational Diffie Hellman assumption which is implied by CDH assumption)[10,12,16]. Under the DDH assumption, Cramer-Shoup scheme is proved to be CCA secure without random oracle model[7].

Among the above CCA secure protocols, hashed ElGamal is advantageous in the aspect of optimal ciphertext overhead[12] and encryption/decryption efficiency, but this can be implemented only in ICDH group for the CCA security.

Now, only bilinear group has been known as ICDH group because, in this group, it is proved that the ICDH assumption is equivalent to the CDH assumption.

The aim of the present work is to propose another ICDH group by using RSA assumption. We have proved that breaking generalized ICDH assumption modulo a composite leads to breaking RSA assumption[2]. In other words, we described that by using the attacker who can break the ICDH assumption in G(i.e., by using ICDH attacker), adversary can break RSA assumption.

On the basis of this, we have proposed a group where ICDH assumption holds. We now sketch how ICDH group can be obtained by RSA assumption.

Let G be the multiplicative subgroup of $Z^*_{n(=pq)}$ with generator g of order $\lambda = \frac{(p-1)(q-1)}{2}$ where $p, q, \frac{p-1}{2}$ and $\frac{q-1}{2}$ are prime numbers. Then, CDH and DDH have been believed to be intractable in G[5,6,8].

That is, in group $G,$ it is believed that there is no probabilistic polynomial time CDH algorithm A such that

$$Pr[A(g, g^x, g^y) = g^{xy} \mid x, y \in Z_n] > negligible$$

and there is no probabilistic polynomial time DDH algorithm A' such that

$Pr\{|Pr[A'(g, g^x, g^y, g^{xy}) = 1] - Pr[A'(g, g^x, g^y, g^z) = 1]| > negligible \mid x, y, z \in Z_n\} > negligible.$

Let (n, e) be the RSA public key and d be the RSA private key such that $ed \equiv 1 \; mod \; \lambda.$ Assume that an adversary can obtain the generator g of group G and $g^d(\in G)$(In RSA, this is possible by randomly selecting generator u and setting $g = u^e mod \; n.$ In this case, g is also a generator and $u = g^d mod \; n$ is satisfied). And assume that r be the element of $G.$

Then, $r = g^x$ is satisfied for some $x(\in Z_n)$ and if CDH assumption is broken in $G,$ the adversary can obtain $r^d(= g^{xd})$ from $r(= g^x)$ and $g^d.$

From the fact above, it can be seen that breaking CDH assumption in group G gives the possibility to break the RSA assumption.

Note. Of course, CDH assumption has been already known to be intractable in G[5,8]. In this paper, we reconsidered it in correlation with RSA assumption.

Similarly, we proved that ICDH assumption holds in G under the RSA assumption as follows.

In the ICDH problem, access to "DH-decision oracle" is added to CDH problem. Assume that CDH assumption is not broken, but ICDH assumption is broken in $G.$ Then, the adversary can briefly break RSA assumption by using public key e as follows.

In RSA, the adversary can briefly test whether any triple $(u = g^d, \hat{v}, \hat{w})$ he likes is a DH-triple (i.e., $\hat{v}^d = \hat{w}$ for the triple $(g^d, \hat{v}, \hat{w})$) by using the given public key e(i.e., by checking that $\hat{w}^e = \hat{v}$), without knowledge of any secret key material and so, he never needs to issue queries to the challenger. In other words, the adversary can access the "DH-decision oracle" that recognizes DH-triples of the form $(g^d, \cdot, \cdot)$ offline on his own.

Note. In hashed ElGamal, the adversary has to access the "DH-decision oracle" online (more precisely, the adversary has to issue the decryption queries to the challenger in the "DH-decision oracle")[12,16].

Consequently, the adversary can obtain $r^d(= g^{xd})$ from $r(= g^x)$ and g^d by using his own "DH-decision oracle" and so, it can be seen that breaking ICDH assumption in group G also gives the possibility to break the RSA assumption. See the proof of Theorem2 for more details.

When modulus n is large enough (e.g., 2048bit), RSA assumption is not broken and so, ICDH assumption holds in group G for the large modulus.

This paper is organized as follows. In Section 2, we consider hashed ElGamal encryption and relationship between RSA security and ICDH assumption in $G.$ In Section 3, we propose the fast and CCA secure hashed ElGamal encryption scheme. In Section 4, we present the some theoretical and experimental results. Finally we conclude this paper in Section 5.

2. Relationship between RSA security and ICDH assumption

To motivate the discussion, we first consider hashed ElGamal encryption scheme in G as follows.

Algorithm 2.1: Key generation for hashed ElGamal in *G*.

Each user creates the public key and the corresponding private key.

Step1. Select a multiplicative cyclic group G of order $\lambda(=\frac{(p-1)(q-1)}{2})$, with generator g where $p\ ,q,\frac{p-1}{2}$ and $\frac{q-1}{2}$ are large primes.

In this case, G becomes a subgroup of $Z_{n(=pq)}^*$. This can be described in detail as follows.

Step1.1. Select the large primes p, q, p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$ and calculate $n = pq$ and $\lambda = lcm(p - 1, q - 1) = 2p'q'$.

Step1.2. Select the generator $g_p$of Z_p^* and generator g_q of Z_q^* and calculate $g(\in Z_n^*)$ that satisfies $g_p = g\ mod\ p$ and $g_q = g\ mod\ q$ as follows.

$$g = \left(\left((g_p - g_q)(q^{-1} mod\ p)\right) mod\ p\right) q + g_q$$

In this case, g becomes a generator of G.

Step2. Select a random integer $x(1 \le x < \lambda, gcd(x, \lambda) = 1)$ and compute the group element $u = g^x$.

This can be described in detail as follows.

Step2.1. Select random integers $x_p(1 < x_p < p - 1)$ and $x_q(1 < x_q < q - 1)$ such that $gcd(x_p, p - 1) = 1$ and $gcd(x_q, q - 1) = 1$. In this case, $x_p \equiv x_q\ mod\ 2$ is satisfied.

Step2.2. Calculate $u_p = g_p^{x_p} mod\ p, u_q = g_q^{x_q} mod\ q$ and

$$u = \left(\left((u_p - u_q)(q^{-1} mod\ p)\right) mod\ p\right) q + u_q.$$

In this case, $u =\ g^x mod\ n, x_p = x\ mod\ (p - 1)$ and $x_q = x\ mod\ (q - 1)$ are satisfied.

Step3. Public key is (g, u, n) and private key is x.

This can be described in detail as follows.

Step3.1. Public key is (g, u, n) and private key is (x, x_p, x_q, p, q).

For the semantic security, encryption and decryption use the symmetric authenticated encryption (E_s, D_s) defined over (K_s, M_s, C_s) and hash function $H(G^2 \rightarrow K_s)$.

Algorithm 2.2: Encryption for hashed ElGamal in *G*.

User encrypts a message $m \in M_s$, where M_s is a message space of (E_s, D_s).

Step1. Obtain authentic public key (g, u, n).

Step2. Select a random integer $y(1 < y < n)$ and compute group elements $v = g^y, w = u^y$ and hash value $k_s = H(v, w)$.

Step3. Encrypt the message m by using symmetric encryption E_s and key k_s.

$$c = E_s(k_s, m)$$

Step4. Send the cipher text $(v \in G, c \in C_s)$. C_s is a cipher text space of (E_s, D_s).

Algorithm 2.3: Decryption for hashed ElGamal in *G*.

User recovers message m from (v, c).

Step1. Compute the group element $w = v^x$ and hash value $k_s = H(v, w)$.

Calculation of w can be done fast by using CRT exponents x_p and x_q as in CRT-RSA[3].

Step1.1. Calculate $v_p = v\ mod\ p$ and $v_q = v\ mod\ q$.

Step1.2. Calculate

$$w_p = v_p^{x_p} mod\, p$$

and

$$w_q = v_q^{x_q} mod\, q.$$

Step1.3. Calculate w as follows.

$$w = \left(\left(\left(w_p - w_q\right)\left(q^{-1} mod\, p\right)\right) mod\, p\right) q + \, w_q$$

Step1.4. Calculate $k_s = H(v, w)$.

Step2. Recover the message m by using symmetric decryption D_s and key k_s.

$$m = D_s(k_s, c)$$

Because CDH and DDH assumptions are satisfied in G[6], following Theorem1 can be obtained referring to Theorem11.4 of [16].

Theorem1. If $\boldsymbol{H}: \boldsymbol{G}^{\boldsymbol{2}} \to \boldsymbol{K_s}$ is modeled as a random oracle and symmetric encryption $(\boldsymbol{E_s}, \boldsymbol{D_s})$ is CPA secure (i.e., is semantically secure against Chosen Plaintext Attack), then hashed ElGamal in $\boldsymbol{G}$ is CPA secure.

Theorem1 shows only the CPA security of hashed ElGamal in G. For the CCA security, a stronger assumption is needed.

Assume that the adversary selects arbitrary elements $\hat{v}(\in Z_n^*)$ and $\hat{w}(\in Z_n^*)$, and computes $\hat{k}_s = H(\hat{v}, \hat{w})$ and $\hat{c} = E_s(\, \hat{k}_s, \hat{m})$ for some arbitrary message $\hat{m}(\in M_s)$. Further, assume the adversary gives the ciphertext$(\hat{v}, \hat{c})$ to a “decryption oracle” and obtains the decryption $m = D_s(\, H(\hat{v}, \hat{v}^x), \hat{c})$. Now, it is very likely that $m = \hat{m}$ if and only if $\hat{w} = \hat{v}^x$. See [12] and [16] for more details.

Note. Decryption algorithm does not verify that $\hat{v} \in G$(Of course, such a verification can be easily done, but it requires additional calculation. Furthermore, it could present a more attractive target for the adversary because it gives an oracle to check whether or not $\hat{v} \in G$? for an arbitrary element $\hat{v} \in Z_n^*$) for given ciphertext$(\hat{v}, c)$ (See Algorithm2.3) and so, $\hat{v} \in Z_n^*$ and $\hat{w} \in Z_n^*$ can be used instead of $\hat{v} \in G$ and $\hat{w} \in G$, respectively, in the CCA scenario (more precisely, in the definition of DH-triple$(u, \hat{v}, \hat{w})$).

For $U(= g^x) \in G, V \in Z_n^*$, define the predicate $dh(U, V) := V^x$ and for $U \in G, \hat{V}, \widehat{W} \in Z_n^*$, define the predicate $dhp(U, \hat{V}, \widehat{W}) := (dh(U, \hat{V}) = \widehat{W}?\,)$. (These are little different from the definition of [12, Section1.1] and [16, Section12.4] because $\hat{V}, \widehat{W} \in Z_n^*$ are used instead of $\hat{V}, \widehat{W} \in G$. As mentioned above, factorization of n is unknown and so, adversary cannot distinguish between G and Z_n^*.) Then, in the CCA scenario, the adversary can use the decryption oracle to answer questions (i.e., $\hat{w} = \hat{v}^x$?) of the form $dhp(u = g^x, \hat{v}, \hat{w})$ for elements $\hat{v}(\in Z_n^*)$ and $\hat{w}(\in Z_n^*)$ of the adversary’s choosing.

The adversary cannot efficiently answer such questions on his own(if he can, DDH assumption is broken in G), and so the decryption oracle is leaking some information about that secret key x which could potentially be used to break the encryption scheme.

From the facts above, ICDH assumption which is used in the CCA security of hashed ElGamal over G can be defined as follows.

ICDH assumption: It is difficult to compute $dh(U, V)$, given random $U \in G$ and $V \in G$, along with access to decision oracle for the predicate $dhp(U,\cdot,\cdot)$, which on input $(\hat{V} \in Z_n^*, \widehat{W} \in Z_n^*)$, returns $dhp(U, \hat{V}, \widehat{W})$.

Following Theorem2 shows that if ICDH assumption is broken in G, then it is possible to break RSA assumption.

Theorem2: Assume ICDH assumption is (t, q_{dh}, ε)-broken in group G, where q_{dh} is the number of queries to "DH-decision oracle" and ε is the probability to break the assumption in time t. Then, RSA assumption is $(t, q_{dh}, \varepsilon/8)$-broken when safe primes are used.

Proof. Let B be an attacker which (t, q_{dh}, ε)-breaks ICDH assumption in group G. We present an adversary A which $(t, q_{dh}, \varepsilon/8)$-breaks RSA assumption when modulus n is the product of two safe primes. Let e be the public exponent and d be the private exponent. Adversary A is given as input (n, e, r) where r was chosen at random from Z_n^* and is trying to find $r^d \ mod\ n$.

In RSA, anyone can obtain the pair of elements (h, h^d), where h is an element of Z_n^*, by selecting arbitrary element $u \in Z_n^*$ and setting $h = u^e mod\ n$(i.e., $u = h^d mod\ n$). Besides, anyone can obtain the arbitrary element $v \in Z_n^*$ by multiplying e^{th} power of arbitrary element $s \in Z_n^*$ and r(i.e., $v = s^e r\ mod\ n$).

Assume that h is a generator of G and v is an element of G(i.e., $v = h^a$).

Then, the ICDH attacker B can obtain $v^d = h^{ad}$ from elements $u = h^d$ and $v = h^a$ with success probability ε and running time t, making q_{dh} queries to "DH-decision oracle" that recognizes DH-triples of form $(h^d \in G,\ \cdot \in Z_n^*,\ \cdot \in Z_n^*\ )$.

In this case, "DH-decision oracle" is different from the one of hashed ElGamal.

First, in order to determine whether or not any triple $(u = h^d \in G, \hat{v} \in Z_n^*, \widehat{w} \in Z_n^*)$ is DH-triple(i.e., $\hat{v}^d = \widehat{w}$?), the ICDH attacker B checks that $\widehat{w}^e = \hat{v}$ using RSA public exponent e on his own without making queries to the challenger, because modular inverse of private key (i.e., $e = d^{-1} mod\ \lambda$) is published in RSA, unlike hashed ElGamal. In other words, "DH-decision oracle" can be done off line(This creates more favorable conditions to B than in hashed ElGamal's DH-decision oracle) by B and so, A need not simulate "DH-decision oracle" to answer B's query.

Second, the computational cost per iteration of "DH-decision oracle" query is comparable to hashed ElGamal.

In RSA, small public exponents are commonly used (i.e., RSA assumption still holds for small public exponents such as 3 and 65537) and so, for given $(u, \hat{v}, \widehat{w})$, calculation of $\tilde{v} = \widehat{w}^e$ for the test $(\tilde{v} = \hat{v}?)$ is much faster(This also creates favorable conditions to B) than the calculation of "DH-decision oracle" of hashed ElGamal in G (i.e., calculation of $\hat{k} = H(\hat{v}, \widehat{w}), \hat{c} = E_s(\hat{k}, \widehat{m}), \widetilde{w} = \hat{v}^x, k = H(\hat{v}, \widetilde{w})$ and $m^* = D_s(k, \hat{c})$ for the test $(\widehat{m} = m^*?)$) because $log_n x \approx 1$. Even though full sized public exponent $e(log_n e \approx 1\ )$ is used[9] in RSA, computation of $\widehat{w}^e$ is comparable to the computation of $\hat{v}^x$ of decryption oracle in hashed ElGamal.

Of course, the generator and element of G are unknown to B. Hence, adversary A must select $h(= u^e)$ and $v(= s^e r\ mod\ n)$ as a generator and an element of G, respectively, and run the ICDH attacker B on input $(u(= h^d), v)$ in order to get v^d.

Meanwhile, many elements of Z_n^* can become the generator or element of G. Hence, when adversary A selects h and v as random elements of Z_n^*(this is accomplished by anyone in RSA as mentioned above), h becomes a generator and v becomes an element of G with high probability.

Let $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$. From the property of Euler function, the probability that random element $h \in Z_n$ becomes a generator of G is as follows.

$$Pr[genertor(h, G) = 1 | h \in Z_n] = \frac{(p'-1)(q'-1)}{(2p'+1)(2q'+1)} = \frac{p'q' - (p'+q') + 1}{4p'q' + 2(p'+q') + 1} \approx \frac{p'q'}{4p'q'} = \frac{1}{4} (1)$$

Order of G is $2p'q'$ and so, the probability that random element $v \in Z_n$ is included in G is as follows.

$$Pr[element(v,G)=1|v\in Z_n]=\frac{2p'q'}{(2p'+1)(2q'+1)}=\frac{2p'q'}{4p'q'+2(p'+q')+1}\approx\frac{2p'q'}{4p'q'}=\frac{1}{2}\ (2)$$

From Equation (1) and (2), the probability that h is a generator of G and v is included in G for arbitrarily selected h and v is as follows.

$$Pr[genertor(h,G)=1,element(v,G)=1\ |h\in Z_n,v\in Z_n]\approx\frac{1}{4}\cdot\frac{1}{2}=\frac{1}{8}\qquad(3)$$

Hence, with probability at least 1/8, A can select h and v as a generator and element of G, respectively, and give B the challenge instance $(u=h^d, v=h^a)$. If and when B outputs $v^d mod\ n$, A outputs $r^d mod\ n=\ v^d s^{-1}\ mod\ n$.

From all facts above, it can be seen that if ICDH assumption is (t,q_{dh},ε)-broken in G, then it is possible to $(t,q_{dh},\varepsilon/8)$-break RSA assumption.**(end of proof)**

Even though safe primes p and q are used, RSA assumption have been believed not to be broken(regardless of whether public exponent e is small or large) and so, ICDH assumption holds in G from Theorem2.

From the above fact, referring to Theorem12.4 of [16], following Theorem3 can be obtained.

Theorem3. If $H: G^2\rightarrow K_s$ is modeled as a random oracle and symmetric encryption (E_s, D_s) is CCA secure, then hashed ElGamal in G is CCA secure.

In the aspect of encryption and decryption efficiency, hashed ElGamal in G is advantageous than other CCA secure ElGamal protocols such as twin ElGamal and Cramer-Shoup scheme because it requires less exponentiation(Table I).

TABLE I
COMPARISON BETWEEN PROPOSED SCHEME AND OTHER CCA-SECURE ELGAMAL PROTOCOLS IN EFFICIENCY

	Hashed ElGamal in G	Twin ElGamal	Cramer Shoup
Public Key	$g,u=g^x$	$g,u_1=g^{x_1},u_2=g^{x_2}$	$g_1,g_2,u_1=g_1^{x_1}g_2^{x_2},u_2=g_1^{y_1}g_2^{y_2},u_3=g_1^z$
Private Key	x	x_1,x_2	x_1,x_2,y_1,y_2,z
Exponentiations in Encryption	$T=g^r,u^r$	$T=g^r,u_1^r,u_2^r$	$T_1=g_1^r,T_2=g_2^r,u_3^r,u_1^r u_2^{r\alpha}$
Exponentiations in Decryption	T^x	T^{x_1},T^{x_2}	$T_1^z,T_1^{x_1+y_1\alpha}T_2^{x_2+y_2\alpha}$

Note.g, $g_1$and g_2 are generators of multiplicative cyclic group.

Furthermore, composite number is used as modulus number and so, CRT can be used to speed up decryption of hashed ElGamal in G. However, in decryption, this scheme is not fast compared to twin ElGamal and Cramer-Shoup scheme on small prime order subgroup of Z_p^* where p is prime number. Hence, we proposed fast variant of hashed ElGamal as follows.

3. Proposed scheme

As in rebalanced RSA, it is possible to increase the decryption speed in hashed ElGamal by reducing the CRT exponents $x_p(=x\ mod\ (p-1))$ and $x_q(=x\ mod\ (q-1))$ instead of private exponent x. In this case, the key generation is same as the one of hashed ElGamal (i.e., Algorithm2.1) except for the Step2.1, which can be described as follows.

Step2.1. Select two random R-bit$\left(0<R<\frac{N}{2},\ 2^{N-1}<n<2^N\right)$ integers x_p and x_q such that $log_n x\approx 1,\ gcd(x_p,p-1)=1$ and $gcd(x_q,q-1)=1$.

Let $I(\subset G)$ is a set of g^x such that $2^{R-1} < x_p, x_q < 2^R,\ log_n x \approx 1, 0 < x < \lambda$ and $gcd(x, \lambda) = 1$ where g is a generator of G with order λ.

For $U(= g^x) \in I, V \in Z_n^*$, define the predicate $Rdh(U, V) := V^x$ and for $U \in I, \hat{V}, \widehat{W} \in Z_n^*$, define the predicate $Rdhp(U, \hat{V}, \widehat{W}) := (Rdh(U, \hat{V}) = \widehat{W}?)$.

Then, RCDH (Restricted CDH with small CRT exponents) and RICDH (Restricted ICDH with small CRT exponents) assumption can be defined as follows.

RCDH assumption: It is difficult to compute $Rdh(U, V)$, given random $U \in I$ and $V \in G$.

RICDH assumption: It is difficult to compute $Rdh(U, V)$, given random $U \in I$ and $V \in G$, along with access to decision oracle for the predicate $Rdhp(U,\cdot,\cdot)$, which on input ($\hat{V} \in Z_n^*, \widehat{W} \in Z_n^*$), returns $Rdhp(U, \hat{V}, \widehat{W})$.

In Section 1 and 2, we proved that under the RSA assumption, CDH and ICDH assumption hold in G. In other words, we proved that CDH and ICDH assumptions can be reduced to RSA assumption in G.

Similarly, under the assumption that RSA assumption holds in rebalanced RSA [9], it would be possible to prove that CDH and ICDH assumptions still hold even if CRT exponents x_p and x_q are reduced in hashed ElGamal over G(i.e., it would be possible to prove that RCDH and RICDH assumptions can be reduced to RSA assumption in rebalanced RSA). See Appendix A for more details.

The important point here is to obtain the minimal value of R at which RCDH (or RICDH) assumption holds.

First, we considered security parameters that RSA assumption is not broken in rebalanced RSA and on the basis of this, set security parameters N and R for RCDH (RICDH) assumption in different security levels.

For this purpose, we analyzed the known attacks to rebalanced RSA. To the best of our knowledge when $\alpha(= log_n e) \approx 1$, the attacks which have been known to be applicable to the rebalanced RSA are only the BS(2002)'s attack[9], JM(2007)'s attack[11] and TLP(2019)'s attack [14,15]. Hence, we considered above three attacks and proposed the security parameter choices of rebalanced RSA($d_p, d_q < n^\delta$) in different security levels(Table II).

TABLE II

RECOMMENDED SECURITY PARAMETER CHOICES OF REBALANCED RSA IN DIFFERENT SECURITY LEVELS

k	N	R	BS's attack ($R^{1/2} logR < 2^k$)	JM's attack ($\delta < 0.073$)	TLP's attack ($\delta < 0.122$)
80	1024	160	160	75	125
112	2048	250	224	150	250
128	3072	375	256	225	375
192	7680	937	384	561	937
256	15360	1874	512	1122	1874

Up to now, it has been believed that RSA assumption is not broken in rebalanced RSA which has security parameters N and R of Table II. Hence, it is trivial that RCDH (RICDH) assumption with security parameters of Table II is also not broken under the assumption that RICDH (RCDH) assumption can be reduced to RSA assumption in rebalanced RSA.

Second, we considered the practical small CRT exponent attack to break RCDH assumption as follows.

From $u_p = u\, mod\, p = g^{x_p} mod\, p,\ u - u_p = u - g^{x_p} mod\, p = jp$ is satisfied and so, $gcd(u - g^{x_p} mod\, p, pq) = p$ is satisfied.

Hence, in order to break RCDH assumption by factorizing modulus n, adversary tries to find i that satisfy

$$gcd(u - g^i mod\, n, n) \neq 1$$

for all available i because CRT exponents are small. And referring to [9, Section4] and [17, Proposition2], he can success in time $O(r^{1/2} logr)$ when $r = min(x_p, x_q)$.

Following Table III shows recommended security parameter choices to be secure from small CRT exponent attack (noted as square root attack) in different security levels.

TABLE III
RECOMMENDED SECURITY PARAMETER CHOICES TO BE SECURE FROM SQUARE ROOT ATTACK IN DIFFERENT SECURITY LEVELS

k	N	R
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Finally, we proved that best known attack to break RCDH (or RICDH) assumptions in G is the square root attack above (i.e., RCDH and RICDH assumptions hold for security parameters of Table III.).

Let B be an attack which is better than square root attack in breaking RCDH(or RICDH) assumption. Then, as in Section 1 and 2, it is possible to present an attack A which breaks RSA assumption in rebalanced RSA by using B.

Table IV shows the rebalanced RSA schemes of different security levels which can be broken by attack A and possibility to break these schemes by known realistic attacks.

The triple indicates the bit length of public exponent e, bit length of private exponent d and bit length of CRT exponent d_p(or d_q).

TABLE IV
APPLICABILITY OF KNOWN REALISTIC ATTACKS TO REBALANCED RSA BROKEN BY ADVERSARY A IN DIFFERENT SECURITY LEVELS

k	Rebalanced RSA	BS's attack ($r^{1/2} logr < 2^k$)	JM's attack ($\delta < 0.073$)	TLP's attack ($\delta < 0.122$)
80	(1024, 1024, 160)	X	X	X
112	(2048, 2048, 224)	X	X	O
128	(3072, 3072, 256)	X	X	O
192	(7680, 7680, 384)	X	O	O
256	(15360, 15360, 512)	X	O	O

Note. "O" means possibility and "X" means impossibility

As shown in Table IV, for 80-bit security level, no attack better than square root attack has been known thus far. For 112-bit and 128-bit security levels, no attack had been introduced until TLP's attack was proposed in 2019. For 192-bit and 256-bit security levels, except for JM's attack and TLP's attack, no attack has been known thus far. Furthermore, JM's attack and TLP's attack are all lattice based attacks and are not attack A.

The facts above contravene the assumption that attack B which is better than square root attack exists. (If B existed, then for 80-bit security level, attack A would be introduced as a small CRT exponent attack stronger than BS's attack. Similarly, if B existed, for 112-bit and 128-bit security levels, attack A would be introduced as a small CRT exponent attack stronger than BS's attack before the proposal of TLP's attack(2019). The same can be said for 192-bit and 256-bit security levels.)

From all the facts above, it can be seen that best known attack to break RCDH (or RICDH) assumptions in G is the square root attack.

Note. In RSA assumption, the modular inverse of private exponent (i.e., public exponent) is given to adversary. However, in RCDH(RICDH) assumption, modular inverse of private exponent (i.e., $e' = x^{-1} mod\ \lambda$) is unknown and it is difficult to find e' because $log_n e' \approx 1$ is usually satisfied[9, Section4]. Hence, JM's attack and TLP's attack which are possible only when public exponent is known cannot be used to break RCDH(RICDH) assumption in G. This gives the possibility that RCDH (RICDH) assumption still holds even if RSA assumption is broken in rebalanced RSA. In other words, breaking RSA assumption by JM's attack and TLP's attack does not affect RCDH (RICDH) assumption in G.

4. Performance analysis

In Table V, we show the comparison of decryption speedup between rebalanced RSA and proposed scheme in different security levels. We get a speedup of $(N/2)/R$ over standard RSA with CRT in rebalanced RSA(R of Table II) and proposed scheme(R of Table III), respectively, because modular exponentiation takes time linear in the exponent's bit-length. Note that decryption time for standard RSA with CRT is obtained by two full exponentiations modulo $(N/2)$-bit numbers.

TABLE V

COMPARISON OF DECRYPTION SPEEDUP BETWEEN REBALANCED RSA AND PROPOSED SCHEME IN DIFFERENT SECURITY LEVELS

k	N	$(N/2)/R$ (Rebalanced RSA)	$(N/2)/R$ (Propose scheme)
80	1024	$3.2 = \frac{512}{160}$	$3.2 = \frac{512}{160}$
112	2048	$4.1 \approx \frac{1024}{250}$	$4.6 \approx \frac{1024}{224}$
128	3072	$4.1 \approx \frac{1536}{375}$	$6 = \frac{1536}{256}$
192	7680	$4.1 \approx \frac{3840}{937}$	$10 = \frac{3840}{384}$
256	15360	$4.1 \approx \frac{7680}{1874}$	$15 = \frac{7680}{512}$

As shown in Table V, the larger modulus is used, the faster proposed scheme becomes than rebalanced RSA in decryption.

Moreover, unlike rebalanced RSA, decryption can be sped up without losing of encryption speed in proposed scheme. Note that encryption time for rebalanced RSA is obtained by one full exponentiation modulo N-bit numbers because public exponent e is full sized [9, 13].

Table VI shows the practical decryption time comparison between proposed scheme and other CCA secure public key protocols for 2048bits modulus.

Experiments are carried for 2048bits modulus on 3.6GHz Core i7-7700 desktop using Open SSL. For each scheme, we ran the decryption algorithm 1000 times in order to obtain an average decryption time. Timings are approximate and should be treated as a relative guideline. As shown in

Table VI, proposed scheme is approximately 2.26, 4.29 and 1.06 times faster than Twin ElGamal, Cramer-Shoup scheme and rebalanced RSA, respectively, in decryption.

TABLE VI

PRACTICAL DECRYPTION TIME COMPARISON BETWEEN PROPOSED SCHEME AND OTHER CCA-SECURE PUBLIC KEY PROTOCOLS (2048-BIT MODULUS)

	Proposed Scheme (KEM/DEM)	Twin ElGamal (KEM/DEM)	Cramer Shoup (KEM/DEM)	Rebalanced RSA (OAEP)
Decryption Exponent (CRT Exponent)	2048 bits (224 bits)	224 bits $(-)$	224 bits $(-)$	2048 bits (250 bits)
Number of Exponentiation in Decryption	1	2	5	1
Number of Multiplication in Decryption (Modular Size)	$2 \times 224 \times 1.5 + 2 = 674$ (1024 bits)	$224 \times 1.75 + 1 = 393$ (2048 bits)	$224 \times 1.5 + 2 + 224 \times 1.75 = 730$ (2048 bits)	$2 \times 250 \times 1.5 + 2 = 752$ (1024 bits)
Decryption time	3.5 ms	7.9 ms	15ms	3.7 ms

5. Conclusions

We proved that under the RSA assumption, ICDH assumption holds in the multiplicative cyclic group of order $\frac{(p-1)(q-1)}{2}$ with composite modulus $n(=pq)$, where $p, q, \frac{p-1}{2}$ and $\frac{q-1}{2}$ are prime numbers.

Our proof gives the possibility to propose the CCA secure hashed ElGamal encryption in the group above and random oracle model.

We also sped up decryption by reducing CRT exponents in CCA secure hashed ElGamal. In this case, our scheme has the fastest decryption among all CCA secure public key encryption schemes (e.g., RSA-OAEP, Twin ElGamal, Cramer-Shoup, ⋯) which are implemented in integer group. In encryption, our scheme needs only two exponentiations, which are optimal for Diffie-Hellman based CCA secure encryption schemes. Meanwhile, the exponentiations for the ElGamal encryption are independent of the plaintext and so, these exponentiations can be sped up by using fixed-base exponentiation algorithms based on precomputation and selecting random exponents which have low Hamming weights. Hence, our results could be applied to the applications which need fast processing in both encryption and decryption..

REFERENCES

1. W.Diffie, M.E.Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22(1976) 644-654.
2. R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public – key cryptosystems, Communications of ACM 21(2)(1978) 120-126.
3. J. J. Quisquater , C. Couvreur, Fast Decipherment Algorithm for RSA Public-Key Cryptosystem, IEEE Electronics Letters 18(1982) 905-907.
4. T.ElGamal, A public key cryptosystem and signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31(1985) 469-472.

5. A.Menezes , P.van Orschot , and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996, pp.287, 617-618.
6. D.Boneh, The decision Diffie–Hellman problem, ANTSIII, Springer LNCS 1423(1998) 48–63.
7. R.Crammer, V.Shoup, A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack, CRYPTO’98, LNCS 1462(1998), 13-25.
8. E.Biham, D.Boneh, O.Reingold, Breaking generalized Diffie–Hellman modulo a composite is no easier than factoring, Information Processing Letters 70(1998), 83-87.
9. D.Boneh , H.Shacham., Fast variants of RSA, CryptoBytes (The Technical Newsletter of RSA Laboratories)5(1) (2002) 1–9.
10. R.Gennaro, H.Krawczyk, T.Rabin, Secure Hashed Diffie–Hellman over Non–DDH Groups, EUROCRYPT’04, LNCS 3027(2004), 361-381.
11. E. Jochemsz, A. May, A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$, In A. Menezes, editor, volume 4622 of Lecture Notes in Computer Science, Springer, 2007, pp 395-411.
12. D.Cash, E.Kiltz and V.Shoup, The twin Diffie-Hellman problem and applications, EUROCRYPT’08, LNCS 4965(2008), 127-145.
13. H.M.Sun, M.E.Wu, M.Jason.Hinek, Trading decryption for speeding encryption in Rebalanced-RSA, The Journal of Systems and Software 82 (2009), 1503-1512.
14. A.Takayasu, Y.Lu, L.Peng, Small CRT-exponent RSA revisited, EUROCRYPT2017, LNCS10211 (2017), 130-159
15. A.Takayasu, Y.Lu, L.Peng, Small CRT-exponent RSA revisited, Journal of Cryptology, 32(4) (2019), 1337-1382 (full version of [14]).
16. D.Boneh, V.Shoup, A Graduate Course in Applied Cryptography (2020), version 0.5, Stanford University, https://crypto.stanford.edu/~dabo/cryptobook/
17. G.C.Kim, S.C.Li, Decryption speed up of ElGamal with composite modulus, PLOS One, October 5, 2020, https://doi.org/10.1371/journal.pone.0240248, 1-16

Appendix A. Possibility to use small CRT exponents in CDH and ICDH assumptions

It is trivial to prove that RCDH assumption can be reduced to RSA assumption in rebalanced RSA. Hence, we considered the case of RICDH assumption as follows.

Following Theorem4 shows that if RICDH assumption is broken in G, then it is possible to break RSA signature schemes such as RSA-FDH and RSA-PSS when safe primes and small CRT exponents are used(i.e., it is possible to break rebalanced RSA-FDH and rebalanced RSA-PSS with safe primes).

Theorem4: Assume RICDH assumption is (t, q_{dh}, ε) -broken at R such that $0 < R < \frac{N}{2}, 2^{R-1} < x_p, x_q < 2^R$ and $log_n x \approx 1$ in group G, where q_{dh} is the number of queries to “DH-decision oracle” and ε is the probability to break the assumption in time t. Then, even without chosen message attack, the adversary can $(t, q_{dh}, \varepsilon/8)$ -break the RSA signature schemes(RSA-FDH and RSA-PSS) which use the safe primes and small CRT exponents such that $2^{R-1} < d_p, d_q < 2^R$ and $log_n d \approx 1$.

Proof. Let B be an attacker which (t, q_{dh}, ε)-breaks RICDH assumption at R such that $2^{R-1} < x_p, x_q < 2^R$ in group G. We present an adversary A which $(t, q_{dh}, \varepsilon/8)$-breaks signature schemes such as RSA-FDH and RSA-PSS in rebalanced RSA with safe primes where $2^{R-1} < d_p, d_q < 2^R$. Adversary A is given as input public key (n, e) and is trying to make existential forgery(m', s').

In RSA, since public key e is known, anyone can obtain the pair of elements (h, h^d), where h is an element of Z_n^*, by selecting arbitrary element $u \in Z_n^*$ and setting $h = u^e mod\, n$ (i.e., $u = h^d mod\, n$).

Meanwhile, in RSA-FDH and RSA-PSS, from the property of random oracle, padding result becomes a random element of Z_n^*(RSA-FDH and RSA-PSS are known to be secure under random oracle assumption.).

Hence, under the random oracle assumption, anyone can obtain the arbitrary element $v \in Z_n^*$ by padding arbitrary message m'(i.e., $v = P(m')$ for padding function P) in RSA-FDH and RSA-PSS.

Assume that h is a generator of G and v is an element of G(i.e., $v = h^a$). Since $2^{R-1} < d_p, d_q < 2^R$ is satisfied, by the assumption that RICDH is $(\boldsymbol{t}, \boldsymbol{q_{dh}}, \boldsymbol{\varepsilon})$-broken at R, the RICDH attacker B can obtain $s' = P(m')^d = v^d = h^{ad}$, which is a valid signature for message m', from elements $u = h^d$ and $v = P(m') = h^a$ in rebalanced RSA-FDH and rebalanced RSA-PSS with success probability ε and running time t, making q_{dh} queries to "DH-decision oracle" that recognizes DH-triples of form $(h^d \in G,\ \cdot \in Z_n^*,\ \cdot \in Z_n^*)$.

In this case, "DH-decision oracle" can be done offline by B as in Section2. In other words, the ICDH attacker B determines whether or not any triple $(u = h^d \in G, \hat{v} \in Z_n^*, \hat{w} \in Z_n^*)$ is DH-triple(i.e., $\hat{v}^d = \hat{w}$?) by checking that $\hat{w}^e = \hat{v}$ on his own, because he knows the modular inverse of d(i.e., public key e).

Of course, the generator and element of G are unknown to B. Hence, A must select $h(= u^e)$ and v as a generator and an element of G, respectively, and run B on input $(u(= h^d), v)$ in order to get v^d.

Meanwhile, many elements of Z_n^* can become the generator or element of G. Hence, when A selects $h(= u^e mod\, n)$ and $v(= P(m')$ or $c)$ as random elements of Z_n^*(As mentioned before, this is accomplished by anyone in RSA), h becomes a generator and v becomes an element of G with probability at least 1/8(mentioned in Section 2).

Hence, with probability at least 1/8, A can select h and v as a generator and element of G, respectively, and give B the challenge instance $(u = h^d, v = h^a)$. If and when B outputs v^d, A makes the existential forgery $(m', s' = v^d mod\, n : v = P(m'))$.

From all facts above, it can be seen that if RICDH assumption($2^{R-1} < x_p, x_q < 2^R$) is (t, q_{dh}, ε)-broken at R in G, then it is possible to $(t, q_{dh}, \varepsilon/8)$-break RSA signature schemes such as RSA-FDH and RSA-PSS under random oracle assumption when small CRT exponents such that $2^{R-1} < d_p, d_q < 2^R$ and safe primes are used. **(end of proof)**

Even though safe primes p and q are used, FDH and PSS have been believed to be secure at R of Table II in rebalanced RSA$\left(2^{R-1} < d_p, d_q < 2^R\right)$. Hence, from Theorem4, it can be seen that RICDH assumption($2^{R-1} < x_p, x_q < 2^R$) holds at R of Table II in G.

Note. RICDH assumption can be proved using RSA assumption as in ICDH assumption (i.e., as in Theorem2). However, rebalanced RSA is practically used for the fast signature generation and so, we only proved RICDH assumption using signature security. Similarly, ICDH assumption can be proved by using signature security.

.