Chosen ciphertext secure public key encryption under DDH assumption with short ciphertext

Xianhui Lu[1], Xuejia Lai[2], Dake He[1]
Email:luxianhui@gmail.com

1: School of Information Science & Technology, SWJTU, Chengdu, China
2: Dept. of Computer Science and Engineering, SJTU, Shanghai, China

Abstract. An efficient variant of the ElGamal public key encryption scheme is proposed which is provably secure against adaptive chosen ciphertext attacks(IND-CCA2) under the decisional Diffie-Hellman(DDH) assumption. Compared to the previously most efficient scheme under DDH assumption by Kurosawa and Desmedt [Crypto 2004] it has one group element shorter ciphertexts, 50% shorter secret keys, 25% shorter public keys and it is 28.6% more efficient in terms of encryption speed, 33.3% more efficient in terms of decryption speed. A new security proof logic is used, which shows directly that the decryption oracle will not help the adversary in the IND-CCA2 game. Compared to the previous security proof, the decryption simulation is not needed in the new logic. This makes the security proof simple and easy to understand.

Keywords: public key encryption, IND-CCA2, DDH

1 Introduction

Security against adaptive chosen ciphertext attacks (IND-CCA2 security) [1–3] is now commonly accepted as the standard security notion for public key encryption schemes. Currently, most of the practical IND-CCA2 secure public key encryption schemes in standard model are variants of ElGamal[4] scheme. Cramer and Shoup[5, 6] proposed the first provably IND-CCA2 secure practical public key encryption scheme based on the decisional Diffie-Hellman(DDH) assumption in the standard model. This was further improved by Kurosawa and Desmedt and yield a more efficient scheme(KD04)[7]. Kiltz proposed a IND-CCA2 secure KEM(key encapsulation mechanism) under the Gap Hashed Diffie-Hellman(GHDH) assumption[8]. Combined with a redundancy-free DEM(data encapsulation mechanism) it will yield a IND-CCA2 secure hybrid encryption scheme(Kiltz07) more efficient than KD04. Okamoto proposed a variant of KD04-KEM that is IND-CCA2 secure[9] under the DDH assumption and πPRF(pseudo-random function with pairwise-independent random sources). Combined with a redundancy-free DEM it will also yield a IND-CCA2 secure hybrid encryption scheme(Okamoto07) more efficient than KD04. M.Abdalla, M.Bellare and P.Rogaway proposed an efficient Diffie-Hellman Integrated Encryption Scheme(DHIES)[10] which is provably IND-CCA2 secure based on oracle Diffie-Hellman(ODH) assumption. Although Kiltz07, Okamoto07 and DHIES are more efficient than KD04, their underlying assumptions are stronger than the DDH assumption. In fact the ODH assumption can be seen as the combination of the gap Diffie-Hellman assumption and a random oracle hash function.

To prove the IND-CCA2 security of a scheme we need to show that the decryption oracle will not give the adversary any useful help. Currently, this is achieved by security reduction: showing that if there is an adversary A who can attack the IND-CCA2 security of a scheme, then an adversary B can be constructed to solve the underlying intractable problem. Here adversary B, also called

the simulator, play the attack game with A and simulate the encryption oracle and the decryption oracle of the scheme perfectly.

1.1 Our Contributions

We propose an efficient variant of the ElGamal public key encryption scheme which is provably secure against adaptive chosen ciphertext attacks under the DDH assumption. Compared to the previously most efficient scheme under DDH assumption by Kurosawa and Desmedt[7] it has one group element shorter ciphertexts, 50% shorter secret keys, 25% shorter public keys and it is 28.6% more efficient in terms of encryption speed, 33.3% more efficient in terms of decryption speed.

A new security proof logic is used, which shows directly that the decryption oracle will not help the adversary in the IND-CCA2 game. In the new security proof method ciphertexts submitted by the adversary are classified into two categories: ciphertexts that constructed by the adversary independent of the challenge ciphertext(independent-ciphertexts) and ciphertexts that constructed based on the challenge ciphertext(extended-ciphertexts). We show that the decryption will reject all extended-ciphertexts. For independent-ciphertexts, we show that the decryption oracle will not leak the distribution information of $eKey$(the key of the one-time symmetric key encryption scheme). Since one-time symmetric key encryption scheme(SKE) is secure against passive attacks, the decryption oracle will not leak information about the distribution of b(the target of IND-CCA2 game) except with negligible probability. Compared to the previous security proof, we do not need to construct the decryption simulation in the new logic. This makes the security proof simple and easy to understand.

2 Definitions

In this section we describe the definitions of public key encryption, one-time symmetric encryption scheme, one-time message authentication code, target collision resistant hash function and decisional Diffie-Hellman assumption.

In describing probabilistic processes, we write $x \xleftarrow{R} X$ to denote the action of assigning to the variable x a value sampled according to the distribution X. If S is a finite set, we simply write $s \xleftarrow{R} S$ to denote assignment to s of an element sampled from uniform distribution on S. If A is a probabilistic algorithm and x an input, then $A(x)$ denotes the output distribution of A on input x. Thus, we write $y \xleftarrow{R} A(x)$ to denote of running algorithm A on input x and assigning the output to the variable y.

2.1 Public Key Encryption

A public key encryption scheme consists the following algorithms:

- PKE.KeyGen(l): A probabilistic polynomial-time key generation algorithm takes as input a security parameter l and outputs a public key/secret key pair (PK,SK). We write (PK,SK) $\leftarrow$ PKE.KeyGen(l)
- PKE.Encrypt(PK,m): A probabilistic polynomial-time encryption algorithm takes as input a public key PK and a message m, and outputs a ciphertext C. We write $C \leftarrow$ PKE.Encrypt(PK, m)

– PKE.Decrypt(SK,C): A decryption algorithm takes as input a ciphertext C and secret key SK, and outputs a plaintext m or a reject symbol $\perp$. We write $m \leftarrow \text{PKE.Decrypt}(\text{SK}, C)$.

We require that for all PK,SK output by PKE.KeyGen(l), all $m \in \{0,1\}^*$, and all C output by PKE.Encrypt(PK,m) we have $\text{PKE.Decrypt}(\text{SK}, C) = m$.

A public key encryption scheme is secure against adaptive chosen ciphertext attacks if the advantage of any adversary in the following game is negligible in the security parameter l:

1. The adversary queries a key generation oracle. The key generation oracle computes (PK,SK) $\leftarrow$ PKE.KeyGen(l) and responds with PK.
2. The adversary makes a sequence of calls to the decryption oracle. For each decryption oracle query the adversary submits a ciphertext C, and the decryption oracle responds with $\text{PKE.Decrypt}(\text{SK}, C)$.
3. The adversary submits two messages m_0, m_1 with $|m_0| = |m_1|$. On input m_0, m_1 the encryption oracle computes:
$$b \overset{R}{\leftarrow} \{0,1\}; C^* \leftarrow \text{PKE.Encrypt}(\text{PK}, m_b)$$
and responds with C^*.
4. The adversary continues to make calls to the decryption oracle except that it may not request the decryption of C^*.
5. Finally, the adversary outputs a guess b'.

We say the adversary succeeds if $b' = b$, and denote the probability of this event by $\Pr_{\text{A}}[\text{Succ}]$. The adversary's advantage is defined as $\text{AdvCCA}_{\text{PKE,A}} = |\Pr_{\text{A}}[\text{Succ}] - 1/2|$.

2.2 One-time symmetric key encryption scheme

Now we review the definition of one-time symmetric key encryption scheme[6]. A one-time symmetric key encryption scheme SKE consists of two algorithms:

– SKE.Encrypt(k, m): The deterministic, polynomial-time encryption algorithm takes as input a key k, and a message m, and outputs a ciphertext χ. We write $\chi \leftarrow SKE.Encrypt(k, m)$
– SKE.Decrypt(k, χ): The deterministic, polynomial-time decryption algorithm takes as input a key k, and a ciphertext χ, and outputs a message m or the special symbol $reject$. We write $m \leftarrow SKE.Decrypt(k, \chi)$

We require that for all $kLen \in N$, for all $k \in \{0,1\}^{kLen}$,$kLen$ denotes the length of the key of SKE, and for all $m \in \{0,1\}^*$,we have:

$$SKE.Decrypt(k, SKE.Encrypt(k, m)) = m.$$

A SKE scheme is secure against passive attacks if the advantage of any probabilistic, polynomial-time adversary A in the following game is negligible in the security parameter kLen:

1. The challenger randomly generates an appropriately sized key $k \in \{0,1\}^{kLen}$.
2. A queries an encryption oracle with two messages m_0, m_1 , $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $\chi^* \leftarrow SKE.Encrypt(k, m_b)$.
3. Finally, A outputs a guess b' .

The adversary's advantage in the above game is defined as $AdvPA_{SKE,A}(kLen) = |\Pr[b = b'] - 1/2|$. If a SKE is secure against passive attack we say it is IND-PA secure.

2.3 One-time message authentication code

Now we review the definition of one-time message authentication code[6]. A one-time message authentication code MAC consists of two algorithms:

- MAC.Gen(l): The parameter generate algorithm takes as input l, and outputs the definitions of key space K_V and the verification message space M_V. We write $(K_V, M_V) \leftarrow MAC.Gen(l)$
- MAC.Ver(k_v, α): The verification algorithm takes as input a key $k_v \in K_V$, and a message $\alpha \in \{0,1\}^*$, and outputs a tag $\tau \in M_V$. We write $\tau \leftarrow MAC.Ver(k_v, \alpha)$

We define the forgery attack game as follows:

1. MAC.Gen(l) outputs K_V, M_V.
2. The adversary chose a bit string $\alpha^* \in \{0,1\}^*$, and submits this to the verify oracle. The verify oracle generate a random key $k_v \in K_V$ then responds $\tau^* \leftarrow MAC.Ver(k_v, \alpha^*)$ to A.
3. A outputs a list:
$$((\alpha_1, \tau_1), \cdots, (\alpha_n, \tau_n))$$

We say that A has produced a forgery if for some $1 \leq i \leq n$ and $\alpha_i \neq \alpha^*$, we have $\tau_i \leftarrow MAC.Ver(k_v, \alpha_i)$, where $n \leq N(l)$, $N()$ is a polynomial function. Define $AdvForge_{MAC}$ to be the probability that A produces a forgery in the above game.

A MAC is secure against forgery attacks if the advantage of any PPT adversary A grows negligibly in l.

2.4 Target collision resistant hash function

A (t, ϵ) target collision resistant hash function (TCR) family is a collection $\mathcal{F}$ of functions $f_K : \{0,1\}^n \rightarrow \{0,1\}^m$ indexed by a key $K \in \mathcal{K}$ (where $\mathcal{K}$ denotes the key space), and such that any attack algorithm A running in time t has success probability at most ϵ in the following game:

- Key Sampling: A uniformly random key $K \in \mathcal{K}$ is chosen (but not yet revealed to A).
- A Commits: A runs (with no input) and outputs a hash function input $s_1 \in \{0,1\}^n$.
- Key Revealed: The key K is given to A.
- A Collides: A continues running and outputs a second hash function input $s_2 \in \{0,1\}^n$.

We say that A succeeds in the above game if it finds a valid collision for f_K, i.e. if $s_1 \neq s_2$ but $f_K(s_1) = f_K(s_2)$. We define the advantage of A as $AdvTCR = |Pr[f_K(s_1) = f_K(s_2) : s_1 \neq s_2] - 1/2|$. We say H is target collision resistant hash function if $AdvTCR$ is negligible.

2.5 Decisional Diffie-Hellman assumption

Now we review the definition of decisional Diffie-Hellman assumption[10]. Let G be a group of large prime order q, A be an adversary, consider the following two experiments:

experiments $\text{Exp}_{G,A}^{ddh-real}$: $a, b \overset{R}{\leftarrow} Z_q^*; W \leftarrow g^{ab}; \gamma \leftarrow A(g^a, g^b, W)$; return γ

experiments $\text{Exp}_{H,A}^{ddh-rand}$: $a, b, c \overset{R}{\leftarrow} Z_q^*; W \leftarrow g^c; \gamma \leftarrow A(g^a, g^b, W)$; return γ

Now define the advantage of the A as:

$$Adv_{G,A}^{ddh} = |\Pr[\text{Exp}_{G,A}^{ddh-real} = 1] - \Pr[\text{Exp}_{G,A}^{ddh-rand} = 1]|$$

When it is the $\text{Exp}_{G,A}^{ddh-rand}$ experiment we say (g, g^a, g^b, W) comes from the random distribution R, otherwise we say (g, g^a, g^b, W) comes from the DDH distribution D.

3 New Scheme

We describe our new scheme as follow:

- KeyGen(1^k): Assume that G is a group of order q where q is large prime number.

$$g \xleftarrow{R} G; x, y \xleftarrow{R} Z_q^*; c \leftarrow g^x; d \leftarrow g^y$$

$$PK \leftarrow (g, c, d, H, TCR, SKE, MAC); SK \leftarrow (x, y)$$

  Here TCR is a target collision resistant hash function, $H : G \rightarrow \{0,1\}^{eLen+mLen}$ is a hash function that $H(v)$ is uniformly distributed over $\{0,1\}^{(eLen+mLen)}$ if v is uniformly distributed over G($eLen$ is the length of SKE key, $mLen$ is the length of MAC key), SKE is a IND-PA secure one-time symmetric key encryption scheme, MAC is an one-time message authentication code secure against forgery attacks.
- Encrypt(PK, m): The encryption algorithm works as follow:

$$r \xleftarrow{R} Z_q^*; u \leftarrow g^r; a \leftarrow TCR(u); k \leftarrow H(c^r d^{ra})$$

$$encKey \leftarrow k[1 \cdots eLen]; macKey \leftarrow k[eLen + 1 \cdots eLen + mLen];$$

$$e \leftarrow SKE.Encrypt(encKey, m), v \leftarrow MAC.Ver(macKey, e); C \leftarrow (u, e, v)$$

- Decrypt(SK, C): The decryption algorithm works as follow:

$$a \leftarrow TCR(u); k \leftarrow H(u^{x+ay})$$

$$encKey \leftarrow k[1 \cdots eLen]; macKey \leftarrow k[eLen + 1 \cdots eLen + mLen];$$

$$if\ v = MAC.Ver(macKey, e)\ return\ m \leftarrow SKE.Decrypt(encKey, e);$$

$$else\ return\ \perp$$

Now we prove that the scheme above is IND-CCA2 secure under the DDH assumption:

Theorem 1. *The scheme above is secure against adaptive chosen cipher-text attack assuming that (1) the decisional Diffie-Hellman problem is hard in group G, (2) SKE is secure against passive attack, (3)MAC is secure against forgery attack, (4)TCR is a target collision resistant hash function.*

To prove the theorem, we show that the decryption oracle will not leak any information about the distribution of the hidden bit b except with negligible probability. Ciphertexts submitted by the adversary are classified into two categories: ciphertexts that constructed by the adversary independent of the challenge ciphertext(independent-ciphertexts) and ciphertexts that constructed based on the challenge ciphertext(extended-ciphertexts). We show that the decryption will reject all extended-ciphertexts. For independent-ciphertexts, we show that the decryption oracle will not leak the distribution information of $eKey$. Since SKE is IND-PA secure, the decryption oracle will not leak information about the distribution of b except with negligible probability.

In the IND-CCA2 game, the challenger runs the PKE.KeyGen(1) program and gets $PK = (g, c, d, TCR, H, SKE, MAC), SK = (x, y)$. Then the challenger gives PK to the adversary. In decryption query phase, when receives the decryption request $C_i = (u_i, e_i, v_i)$ the challenger responses with $PKE.Decrypt(SK, C_i)$. In encryption oracle query phase, the challenger computes as follow:

$$b \xleftarrow{R} \{0,1\}; r \xleftarrow{R} Z_q^*; u \leftarrow g^r; a \leftarrow TCR(u); k \leftarrow H(c^r d^{ra})$$

$$encKey \leftarrow k[1 \cdots eLen]; macKey \leftarrow k[eLen+1 \cdots eLen+mLen];$$

$$e \leftarrow SKE.Encrypt(encKey, m_b), v \leftarrow MAC.Ver(macKey, e); C \leftarrow (u, e, v)$$

Then the challenger send C to the adversary. The theorem now follows immediately from the following two lemmas.

Lemma 1. *The decryption oracle will not leak any information about the distribution of the hidden bit b except with negligible probability.*

When receive a decryption query $C_i = (u_i, e_i, v_i)$, there are four cases we need to consider. Case1, Case2 and Case3 are the situations of extended-ciphertexts. Case 4 is the situation of independent-ciphertexts.

Case 1: $u_i = u, e_i = e, v_i \neq v$. It is clear that $v_i \neq MAC.Ver(macKey_i, e_i)$, and C_i will be rejected. In this case the decryption oracle will not leak any information about the distribution of the hidden bit b.

Case 2: $u_i = u, e_i \neq e$. Since MAC is secure against forgery attacks, we get that $v_i \neq MAC.Ver(macKey^*, e_i)$ except with negligible probability. Thus, C_i will be rejected in this case. It yield that the decryption oracle will not leak any information about the distribution of the hidden bit b except with negligible probability.

Case 3: $u_i \neq u$ and u_i is constructed based on u. Let $u_i = g^{r_i}, r_i = tr, a_i = TCR(u_i)$, we get $c^{r_i} d^{r_i a_i} = c^{tr} d^{tra_i} = (c^r d^{ra})^t d^{tr(a_i - a)}$. Since TCR is a target collision resistant hash function we have that $a_i \neq a$. Suppose the adversary can get $H(c^r d^{ra})$ from C, the information the adversary can get includes $(g, c, d, g^r, H(c^r d^{ra})$. Now we show that if the adversary A can distinguish d^r from random value we can construct an adversary B to solve the DDH problem:

Given a DDH challenge $(g_1, g_2, u_1, u_2) \in G^4$, B selects random number $w \in Z_q^*$, sets $g \leftarrow g_1, d \leftarrow g_2, a \leftarrow TCR(u_1), c \leftarrow g^w d^{-a}$. Then B gives $(g, c, d, u_1, H(u_1^w), u_2)$ to A. Let $u_1 = g_1^r$, we have $u_1^w = (g^w d^{-a})^r d^{ra} = c^r d^{ra}$. So if A can tell whether $u_2 = d^r$ then B can tell whether $u_2 = d^r = g_2^r$.

Now we have $c^{r_i} d^{r_i a_i} = (c^r d^{ra})^t d^{tr(a_i - a)}$ is independent from the adversary's view. So $macKey_i = H(c^{r_i} d^{r_i a_i})[eLen+1 \cdots eLen+mLen]$ is independent from the adversary's view. Then the probability of $v_i = MAC.Ver(macKey_i, e_i)$ is negligible, and C_i will be rejected except with negligible probability. It yields that the decryption oracle will not leak any information about the distribution of the hidden bit b except with negligible probability.

Case 4: $u_i \neq u$ and u_i is independent of u. Let $u_i = g^{r_i}$, we have that $k_i = H(c^{r_i} d^{r_i a_i})$ is independent of $k = H(c^r d^{ra})$. Since SKE is IND-PA secure the decryption oracle will not leak the information of b directly. Now we show that the decryption oracle will not help the adversary to solve the DDH problem of $(g, cd^a, g^r, (cd^a)^r)$. Thus it will not leak the information of b indirectly. Suppose the adversary can get $k_i = H(c^{r_i} d^{r_i a_i})$ from (C_i, m_i). We show that if the adversary A can distinguish $(cd^a)^{r_i}$ from random value conditioned on k_i, then we can construct an adversary B to solve the DDH problem:

Given a DDH challenge $(g_1, g_2, u_1, u_2) \in G^4$, B selects random number $w \in Z_q^*$, sets $g \leftarrow g_1, a_i \leftarrow TCR(u_1), c \leftarrow g^w d^{-a_i}, d \leftarrow (g_2 g^{-w})^{1/(a-a_i)}$. Then B gives $(g, c, d, u_1, H(u_1^w), u_2)$ to A. Let $u_1 = g_1^{r_i}$, we have $u_1^w = (g^w d^{-a_i})^{r_i} d^{r_i a_i} = c^{r_i} d^{r_i a_i}$. So if A can tell whether $u_2 = (cd^a)^{r_i}$ then B can tell whether $u_2 = (cd^a)^{r_i} = g_2^{r_i}$.

We have that the output of the decryption oracle m_i will not leak the distribution information of $(cd^a)^{r_i}$. It means that the decryption oracle will not help the adversary to solve the DDH problem of $(g, cd^a, g^r, (cd^a)^r)$. So the decryption oracle will not leak the distribution information of b indirectly. Finally, it yields that the decryption oracle will not leak any information about the distribution of the hidden bit b in this case.

Lemma 2. *Without the help of the decryption oracle the adversary can not distinguish b from 0 to 1 except with negligible probability.*

We show that if the adversary can distinguish $k = H((cd^a)^r)$ from random value, then we can construct an adversary B to solve the DDH problem:

Given a DDH challenge $(g_1, g_2, u_1, u_2) \in G^4$, B selects random number $w \in Z_q^*$, sets $g \leftarrow g_1, d \leftarrow g_2, c \leftarrow g_2^w, a \leftarrow TCR(u_1)$,. Then B gives $(g, c, d, u_1, H(u_2^{w+a})$ to A. Let $u_1 = g_1^r, u_2 = g_2^{r'}$, we have $u_2^{w+a} = c^{r'} d^{ar'}$. So if A can tell whether $H(u_2^{w+a}) = H(c^r d^{ar})$ then B can tell whether $u_2 = g_2^r$.

We have that, without the help of the decryption oracle, $encKey = H((cd^a)^r)[1 \cdots eLen]$ will be independent from the adversary's view. Since the SKE is IND-PA secure, the adversary can not distinguish b from 0 to 1 except with negligible probability.

That's finish the proof of theorem 1.

Remark: Note that a new proof logic was used in the security proof above, which shows directly that the decryption oracle will not help the adversary. This is different from the typical reduction proof, and our proof dose not need the decryption simulation.

4 Efficiency Analysis

The efficiency of KD04, Kiltz07, Okamoto07, DHIES and our scheme is listed in table 1.

Table 1. Efficiency comparison

	Encryption(exp)	Decryption(exp)	Cipher-text overhead(bit)	Assumption	Public key	Secret key
KD04	3.5(2exp+1mexp)	1.5(0exp+1mexp)	$2\|p\| + \|t\|$	DDH	$4\|p\|$	$4\|q\|$
Kiltz07	3.5(2exp+1mexp)	1.5(0exp+1mexp)	$2\|p\|$	GHDH	$3\|p\|$	$2\|q\|$
Okamoto07	3.5(2exp+1mexp)	1.5(0exp+1mexp)	$2\|p\|$	DDH+πPRF	$4\|p\|$	$4\|q\|$
DHIES	2(2exp+0mexp)	1(1exp+0mexp)	$\|p\| + \|t\|$	ODH	$2\|p\|$	$1\|q\|$
NEW	2.5 (1exp+1mexp)	1(1exp+0mexp)	$\|p\| + \|t\|$	DDH	$3\|p\|$	$2\|q\|$

In table1, NEW is our new scheme, KD04 is the scheme in [7], Kiltz07 is the hybrid scheme constructed by the KEM in [8] and a redundancy-free DEM, Okamoto07 is the hybrid scheme

constructed by the KEM in [9] and a redundancy-free DEM, DHIES is the scheme in [10]. When tabulating computational efficiency hash function and block cipher evaluations are ignored, multi-exponentiation ($mexp$) is counted as 1.5 exponentiations (exp). Ciphertext overhead represents the difference between the ciphertext length and the message length, and $|p|$ is the length of an element of G, $|q|$ is the length of an element of Z_q^*, $|t|$ is the length of the one-time message authentication code.

It is clear that the new scheme is more efficient than all previously schemes except DHIES. Yet DHIES is provably secure under the ODH assumption which can be seen as the combination of the Gap Diffie-Hellman assumption and a random oracle hash function. While, our scheme can be proved to be IND-CCA2 secure under the DDH assumption. Compared to the previously most efficient scheme under DDH assumption by Kurosawa and Desmedt[7] it has one group element shorter ciphertexts, 50% shorter secret keys, 25% shorter public keys and it is 28.6% more efficient in terms of encryption speed, 33.3% more efficient in terms of decryption speed.

5 Conclusion

We proposed a new variant of the ElGamal public key encryption scheme. The new scheme is nearly as efficient as the most efficient variant of ElGamal, DHIES. However, the new scheme is provably IND-CCA2 secure under the DDH assumption. Compared to the previously most efficient scheme under DDH assumption by Kurosawa and Desmedt[7] it more efficient in terms of computation, bandwidth and key size. In the security proof, a simple and direct proof logic was used, which shows directly that the decryption oracle will not leak the distribution information of b. Compared to the current security reduction logic, the decryption simulation is not needed in the new proof logic.

References

1. C. Rackoff and D. Simon, Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack, Adv. in Cryptology - Crypto 1991, LNCS vol. 576, Springer- Vrlag , pp. 433-444, 1991;
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, Relations Among Notions of Security for Public-Key Encryption Schemes, Adv. in Cryptology - Crypto 1998, LNCS vol. 1462, Springer-Verlag, pp. 26-45, 1998;
3. D. Dolev, C. Dwork, and M. Naor, Non-Malleable Cryptography, SIAM J. Computing, 30(2): 391-437, 2000;
4. T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469C472, 1985.
5. R. Cramer and V. Shoup, A Practical Public Key Cryptosystem Provably Secure Against Chosen Ciphertext Attack, Adv. in Cryptology - Crypto 1998, LNCS vol. 1462, Springer- Verlag , pp. 13-25, 1998;
6. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing, 33(1):167-226, 2003.
7. K. Kurosawa and Y. Desmedt, A New Paradigm of Hybrid Encryption Scheme, Adv. in Cryptology - Crypto 2004, LNCS vol. 3152, Springer-Verlag, pp. 426-442, 2004;
8. Eike Kiltz. Chosen-Ciphertext Secure Key Encapsulation based on Hashed Gap Decisional Diffie-Hellman. Proceedings of the 10th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2007, pp. 282–297 LNCS 4450 (2007).Springer-Verlag. Full version available on Cryptology ePrint Archive: Report 2007/036
9. Tatsuaki Okamoto. Authenticated Key Exchange and Key Encapsulation in the Standard Model. Advances in Cryptology C ASIACRYPT 2007, pp. 474–484 LNCS 4833 (2007).Springer-Berlin / Heidelberg.
10. M. Abdalla, M. Bellare and P. Rogaway. DHIES: An encryption scheme based on the Diffie-Hellman Problem Extended abstract, entitled The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES, was in Topics in Cryptology - CT-RSA 01, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed, Springer-Verlag, 2001