IND-CCA Secure Encryption Based on Zheng-Seberry Scheme

Murat Ak[a,*], Turgut Hanoymak[b,c], Ali Aydın Selçuk[a]

[a] *Department of Computer Engineering, Bilkent University, 06800, Ankara, Turkey*
[b] *Institute of Applied Mathematics, METU, 06800, Ankara, Turkey*
[c] *Department of Mathematics, Yüzüncü Yıl University, 65080, Van, Turkey*

Abstract

In 1993, Zheng and Seberry proposed three methods for strengthening public key cryptosystems. These methods aimed to obtain schemes that are secure against adaptively chosen ciphertext attacks. One method was improving security by using digital signatures. Zheng and Seberry gave an example scheme that employs this method. However, they were not able to prove IND-CCA security of their cryptosystem. In this paper, we modify this cryptosystem by employing Schnorr signature scheme and prove this new scheme to be IND-CCA secure in the random oracle model.

Keywords: Public key cryptography, Zheng-Seberry encryption scheme, Schnorr signature, provable security, random oracle model

1. Introduction

Since public key cryptography was introduced in 1970s, there has been a considerable amount of research regarding the correct definition of its security. Today as the standard definitions of security, we have IND-CPA (indistinguishability against chosen plaintext attacks) and IND-CCA (indistinguishability against chosen ciphertext attacks, non-adaptive or adaptive) types of security. The former is a rather loose definition compared to the latter, and today, any newly proposed scheme is expected to have a proof

*Corresponding author
Email addresses: muratak@cs.bilkent.edu.tr (Murat Ak), turguthanoymak@gmail.com (Turgut Hanoymak), selcuk@cs.bilkent.edu.tr (Ali Aydın Selçuk)

of IND-CCA security. Therefore, the problem of providing new IND-CCA secure schemes, proving existing schemes to be IND-CCA secure, and modifying existing schemes so that they would then be IND-CCA secure, have been popular research problems for almost the last twenty years.

In 1993, for obtaining IND-CCA secure public key encryption schemes from basic ElGamal encryption [2], Zheng and Seberry [1] proposed three immunization methods based on: (1) one-way hash functions, (2) universal hash families, (3) digital signatures. They illustrated each method by giving example schemes. They also gave formal proofs of IND-CCA security for the first two methods. However, they were not able to prove the security of the scheme that uses digital signatures.

In this paper, we modify this construction by adapting the Schnorr signature scheme and obtain a slightly more efficient encryption scheme. Moreover, we also give a formal proof of IND-CCA security for the new scheme in the random oracle model.

2. Related Work

Zheng and Seberry [1] proposed three methods to make public key encryption schemes secure against adaptive chosen ciphertext attacks. As they state it, the common idea in their methods is to append a tag to each ciphertext that is correlated to the plaintext being encrypted. Apparently, this guarantees that any decryption query that an attacker makes can give no new information about the plaintext, because the attacker already needs to know the plaintext before asking the query since it is needed to find the tag.

The methods given in [1] differ in the ways they design the tag in the ciphertext. In the first method, the tag is a one-way function of the plaintext. The second method uses universal hash families to relate the tag to the plaintext. And the last method uses digital signatures for the same purpose.

For the first two methods, Zheng and Seberry proposed example schemes and they were able to prove IND-CCA security for each one. However, the scheme they gave as an example for the last method was left without a formal security proof for IND-CCA security. They rather proved that if it is IND-CPA secure, it will also be IND-CCA secure. However, they were not able to prove IND-CPA security. The reason was simple: one of the components involved the hash of the plaintext and it was not clear that whether this would give away information about the plaintext.

2.1. Random Oracles

In 1993, Bellare and Rogaway introduced a new model for proving security of encryption schemes: the random oracle model (ROM) [3]. The core idea in ROM is to assume that truly random functions can exist and hash functions can be assumed to produce their result truly randomly. This effectively brought much more efficient schemes into consideration since they can at least be proven in ROM while the schemes proven in the standard model were too expensive by that time – until the Cramer-Shoup (CS) cryptosystem [4]. However, even after the CS cryptosystem, ROM has still been used for proving schemes secure, as it is accepted to provide at least a strong security guarantee although it does not *really* provide certain security.

2.2. Works on Zheng-Seberry Schemes

So far, there has been no concrete studies on the signature-based scheme of Zheng and Seberry. However, a similar scheme appeared in a different line of research by Abe [5]. The main result of [5] was a generic method for obtaining IND-CCA secure public-key encryption schemes by using weak primitives. Two schemes, one of which is similar to ours, were discussed as instantiations to this generic scheme. We observed that although the schemes look similar, our scheme outperforms the instantiation of [5] by requiring fewer exponentiations for both encryption and decryption operations. More precisely, our scheme requires two exponentiations for both ecryption and decryption whereas instantiation of [5] requires three for both.

3. Background

In this section, we present the background knowledge. First, we mention the security definitions we use. Then, we discuss the random oracle model and security models. Finally, we recall the signature-based scheme of [1].

3.1. Security of Encryption Schemes

Security of cryptographic schemes are formalized with two components: (1) The goal of the attacker. (2) Abilities of the attacker. Today, it is widely accepted that the goal of the attacker should be to distinguish among two plaintexts given the ciphertext of one of them. Thus, if a scheme is secure against this kind of attack, given a ciphertext, it would be impossible for the attacker to induce any non-trivial information about the plaintext. This is called as *indistinguishability* (IND) or *semantic security*. When it comes to the abilities of the attacker, we have three different types:

- Chosen plaintext attack (CPA): Attacker has no ability to learn the decryption of any ciphertext he chooses.
- Chosen ciphertext attack (CCA1): Attacker can ask for the decryption of a polynomial number of ciphertexts he chooses—but only before he sees the challenge ciphertext.
- Adaptive chosen ciphertext attack (CCA2): Attacker can also ask for decryptions after the challenge—but he is not allowed to ask for the decrpytion of the challenge ciphertext for obvious reasons.

3.2. Security Models

The security of cryptographic schemes are typically modeled in the form of a game between a hypothetical attacker $\mathcal{A}$ and a challenger $\mathcal{C}$ that represents the scheme. Now, suppose that we have a public key encryption scheme S consisting of the usual three algorithms, $\texttt{KeyGen}(1^\lambda)$, $\texttt{Encrypt}(pk, m)$, and $\texttt{Decrypt}(sk, c)$. IND-CPA/CCA security is modeled with Game 1.

Game 1 IND-CPA and IND-CCA1/2 games.

1: $\mathcal{C}$ prepares an instance of a scheme by running $(pk, sk) \leftarrow \texttt{KeyGen}()$.
2: $\mathcal{C}$ passes the public key pk to $\mathcal{A}$.
3: (Only in IND-CCA) $\mathcal{A}$ makes polynomially many decryption queries with ciphertexts and $\mathcal{C}$ responds with the decryptions.
4: $\mathcal{A}$ chooses two messages m_0, m_1 and requests a challenge.
5: $\mathcal{C}$ selects a random bit $b \leftarrow_R \{0, 1\}$ and sends $c^* \leftarrow \texttt{Encrypt}(pk, m_b)$ to $\mathcal{A}$ as the challenge ciphertext.
6: (Only in IND-CCA2) $\mathcal{A}$ makes polynomially many decryption queries with ciphertexts other than c^* and $\mathcal{C}$ responds with the decryptions.
7: $\mathcal{A}$ guesses b' for b and wins if $b' = b$.

In a typical indistinguishability proof, the advantage of the attacker, $\text{Adv}_A \leftarrow |\Pr[b' = b] - 1/2|$, must be proved to be negligible.

3.3. Computational and Gap Diffie-Hellman Assumptions

One of the most famous problems that is used in cryptography is the computational Diffie-Hellman (CDH) problem [6]. Formally, the CDH problem is defined as follows: Let $\mathbb{G}$ be a cyclic group of order q. Given three group elements g, g^a, g^b, where $a, b \in \mathbb{Z}_q$, compute g^{ab}. The CDH problem is

assumed to be hard, meaning that it is computationally intractable, i.e., has no polynomial-time solution.

Decisional Diffie-Hellman (DDH) problem is the decisional counterpart of the CDH problem: Again, let $\mathbb{G}$ be a cyclic group of order q. Given a generator g, and three group elements g^a, g^b, g^c, where $a, b, c \in \mathbb{Z}_q$, decide whether $g^c = g^{ab}$. DDH assumption states that the DDH problem is hard.

We will use another Diffie-Hellman variant, the Gap Diffie-Hellman (GDH) problem, introduced by Okamoto and Pointcheval [7]: Let $\mathbb{G}$ be cyclic group of order q with a generator g. Given g^a, g^b and a DDH oracle, compute g^{ab}. Basically, GDH problem is the same as CDH problem, except this time, an oracle to decide DDH problem is given.

Here we define a variant of Computational Diffie-Hellman problem, which we call Π:

Definition 1 (Problem Π). Given (g^a, g^b, A, B), for randomly chosen values $(a, b, A, B) \in \mathbb{Z}_q$, compute $g^{a(b+c)}$, where $c = Bb + A$.

Proposition 1. *The problem Π is equivalent to the CDH problem.*

Proof. We prove that Π is equivalent to the CDH problem by showing that these problems can be reduced to each other.

The problem Π can be reduced to the CDH problem as follows: Given a Π instance (g^a, g^b, A, B) and a CDH solver, we first calculate $g^{b+c} = g^b(g^b)^B g^A$, then ask (g^a, g^{b+c}) to the CDH solver which outputs $g^{a(b+c)}$, which is exactly the result of the Π instance we are looking for.

The CDH problem can be reduced to Π as follows: Given a CDH problem instance (g^a, g^b) and a Π oracle, we first choose random $A, B \in \mathbb{Z}_q$ where $B \neq -1$ and we ask (g^a, g^b, A, B) to the Π solver and we get $y = g^{a(b+c)}$. Note that y can be rewritten as $y = g^{a(b+Bb+A)} = g^{ab} g^{abB} g^A$. Then, we can calculate g^{ab} as $g^{ab} = (y/g^A)^{(1+B)^{-1}}$ and return g^{ab}, which is exactly the result of the CDH problem we are looking for. (Note that, in fact, we could have also chosen $A = 0$ and $B = 0$ so that the result of the Π query would immediately be the answer of the CDH instance.) □

In the IND-CCA2 security proof of our modified scheme, we will assume that GDH problem is hard. We will also show that our scheme is IND-CPA secure under the CDH assumption.

3.4. Random Oracle Model

ROM is a model for proving security of encryption schemes. In the proofs within this model, we assume the existence of an hypothetical box called *random oracle* which can produce truly random outputs every time it is queried with a different input. In practice, cryptographic hash functions are used instead of random oracles. This model has the advantage that the inherent weaknesses of hash functions are not considered within the security proof. This makes it possible to use hash functions in encryption schemes more effectively and obtain much more efficient cryptosystems compared to the ones that can be proved in the standard model.

3.5. Zheng-Seberry Scheme with Digital Signature Method

In this section, we will briefly recall the original Zheng-Seberry encryption scheme that adapts ElGamal signature, called C_{sig}.

Suppose Bob, B, wants to send an n-bit message m in secret to Alice, A, with A's public key $y_A = g^{x_A}$ where x_A is the secret key of A. Let $\mathbb{G}$ be a cyclic group of prime order q and let g be a generator of $\mathbb{G}$. Let H be a cryptographic hash function and G be a cryptographically strong pseudorandom string generator. We denote the set of numbers $\{i, i+1, \ldots, j\}$ by $[i, j]$ and the substring of a from position i to position j by $a_{[i,\ldots,j]}$. Encryption and decryption algorithms are as follows:

Algorithm 1 C_{sig}.`Encrypt`(y_A, m)

1: $x \in_R [1, q]$ and $k \in_R [1, q-1]$.
2: $r = {y_A}^{x+k}$ and $z = G(r)$.
3: $c_1 = g^x$ and $c_2 = g^k$.
4: $c_3 = (H(m) - xr)/k \bmod q$.
5: $c_4 = z \oplus m$.
6: $C = (c_1, c_2, c_3, c_4)$.

Algorithm 2 C_{sig}.`Decrypt`$(x_A, c_1, c_2, c_3, c_4)$

1: $r' = (c_1 c_2)^{x_A}$
2: $z' = G(r')_{[1\ldots n]}$
3: $m' = z' \oplus c_4$
4: if $g^{H(m')} = c_1^{r'} c_2^{c_3}$ then output m', otherwise $\perp$.

The scheme C_{sig} has not been proven IND-CPA secure against chosen plaintext attacks so far. This is mostly because m appears both c_3 and c_4 and it is not clear that whether c_3 can be used to induce information about m to break the scheme. In the next section, we will modify this scheme to get a more efficient one and prove that this new scheme is IND-CCA2 secure.

3.6. Schnorr Signature

Now, we briefly discuss Schnorr signature [8]. Let $\mathbb{G}$ be a multiplicative, prime order group in which the discrete logarithm (DL) problem is hard, and let g be a generator. Let H be a cryptographic hash function, x and $y = g^x$ be the private and public keys, respectively.

Algorithm 3 $S.\texttt{Sign}(g, H, x, m)$

1: Choose a random $k \in \mathbb{Z}_q^*$ and set $r = g^k$.
2: Compute $e = H(m \parallel r)$ and $s = k - xe \bmod q$.
3: Return (e, s) as the signature pair.

Algorithm 4 $S.\texttt{Verify}(g, H, e, s, y)$

1: Compute $r_v = g^s y^e$ and $e_v = H(m \parallel r_v)$.
2: If $e_v = e$ holds, then the signature is verified.

Correctness of the verification algorithm can be shown as follows: $r_v = g^s y^e = g^{(k-ex)} g^{ex} = g^k = r$ and then, $e_v = H(m \parallel r_v) = H(m \parallel r) = e$.

Recall that under the discrete logarithm assumption, the Schnorr signature scheme [8] is existentially unforgeable under chosen message attacks. The proof is similar to that of hashed RSA signature scheme. It is also efficient and generates short signatures.

4. An IND-CCA2 Secure Encryption Scheme

We modify C_{sig} by using the Schnorr signature instead of ElGamal. We call our new modified scheme C_{msig}. This new scheme is indeed IND-CCA2 secure as we will prove below. Also, it turns out to be a more efficient scheme, since no inversion operation is needed as in C_{sig}.

Let $\mathbb{G}$ be a group with prime order q and g be a generator of $\mathbb{G}$. Let x_A be A's private key and $y_A = g^{x_A}$ be its public key. B needs to send an n bit

Algorithm 5 C_{msig}.`Encrypt`(y_A, p, g, m)

1: $x \in_R [1, q]$ and $k \in_R [1, q]$.
2: $r = {y_A}^{x+k}$ and $z = G(r)$.
3: $c_1 = g^x$ and $c_2 = H(m \parallel c_1 \parallel r)$.
4: $c_3 = k - xc_2 \bmod q$.
5: $c_4 = z \oplus m$.
6: $C = (c_1, c_2, c_3, c_4)$.

Algorithm 6 C_{msig}.`Decrypt`$(x_A, p, g, c_1, c_2, c_3, c_4)$

1: $r' = (g^{c_3} {c_1}^{c_2+1})^{x_A}$
2: $m' = c_4 \oplus G(r')$
3: if $(H(m' \parallel c_1 \parallel r') = c_2$ then output m', otherwise $\perp$.

message $m \in \mathcal{M}$ to A. Let $G : \mathbb{G} \to \mathcal{M}$ and $H : \mathcal{M} \times \mathbb{G}^2 \to \{1, \ldots, q\}$ be cryptographic hash functions. Encryption and decryption works as follows:

Correctness of the scheme can be checked easily as follows:

$$r' = (g^{c_3} {c_1}^{c_2+1})^{x_A} = (g^{k-xc_2} g^{xc_2+x})^{x_A} = (g^k g^x)^{x_A} = {y_A}^{x+k} = r$$

$$m' = c_4 \oplus G(r') = c_4 \oplus G(r) = m$$

$$H(m' \| c_1 \| r') = H(m \| c_1 \| r) = c_2$$

Regarding the time complexity of this scheme, note that compared to the original ZS scheme, the inversion operation is no more needed and we need fewer exponentiations for both encryption and decryption.

5. Security Analysis

In this section, we prove that our modified scheme C_{msig} is IND-CCA2 secure in the ROM, under the GDH assumption. First, recall that in ROM, we have the following crucial random oracle property:

ROP. The output of a random oracle H to an input a stays truly random unless $H(a)$ is explicitly queried.

Theorem 2. *The encryption scheme C_{msig} is IND-CCA2 secure in the random oracle model if the GDH assumption holds in the underlying group.*

PROOF. The proof consists of two parts: (1) proving that the scheme is IND-CPA secure, (2) proving that the scheme is plaintext aware. It is a well-known fact that these two properties imply IND-CCA2 security [10]. We will prove IND-CPA security under the CDH assumption which already implies GDH assumption since CDH is a simpler assumption. However, as we will mention after the proof, a similar and even better proof that assumes GDH is also possible.

Lemma 1. *The encryption scheme C_{msig} is IND-CPA secure in the random oracle model under the CDH assumption.*

PROOF. Assume that there exists an attacker A_{CPA} that can win the IND-CPA game against C_{msig} with a non-negligible probability ε. Then, there exists a solver, S_Π, that can solve the problem Π with a non-negligible probability ε'. This solver S_Π makes use of A_{CPA} to solve an arbitrary Π instance (g^a, g^b, A, B) as follows:

First, although S_Π does not know a, it implicitly sets x_A to a, by sending $y_A = g^a$ as the public key to A_{CPA}. Then, A_{CPA} makes random oracle queries which S_Π responds by choosing a random value and returning it for any new query. The solver S_Π maintains a list of queries denoted by

$$\ell_H = ((\mu_1 \| u_1 \| v_1), H_1), ((\mu_2 \| u_2 \| v_2), H_2), \ldots, ((\mu_{\alpha_H} \| u_{\alpha_H} \| v_{\alpha_H}), H_{\alpha_H}) \text{ and}$$

$$\ell_G = (r_1, G_1), (r_2, G_2), \ldots, (r_{\alpha_G}, G_{\alpha_G})$$

After the first query phase, A_{CPA} will output two messages m_0 and m_1. At this point, S_Π chooses a random bit d and prepares a challenge $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ that is indistinguishable for A_{CPA} from a real encryption of m_d. S_Π first sets $c_1^* = g^b$, implicitly setting $x^* = b$. It also sets $k^* = Bb + A$ implicitly. Then, it sets $c_2^* = B$ and adds a special incomplete record, $(m_b \| c_1^* \| ?), B)$, to ℓ_H. It also sets $c_3^* = A$. Finally, S_Π randomly chooses c_4^* and adds a special record $(?, c_4^* \oplus m_d)$ to ℓ_G. S_Π sends this challenge c^* to A_{CPA} and whenever A_{CPA} makes a random oracle query in the new query phase, S_Π responds as in the first query phase. After the second query phase is over, A_{CPA} returns a guess d' for d. Finally, S_Π arbitrarily chooses one of the query inputs v_i or r_i from lists ℓ_H and ℓ_G and returns it.

Let us define the followings events regarding this game:

`AskH` is the event where A_{CPA} makes the query $H(m_d \| g^b \| y_A^{x^*+k^*})$.

`AskG` is the event where A_{CPA} makes the query $G(y_A^{x^*+k^*})$.

Now, note that $\Pr[d = d' \mid \neg\texttt{AskH} \wedge \neg\texttt{AskG}] = 1/2$. This is simply because unless exactly these queries are explicitly done, the challenge ciphertext contains no information about m_d at all due to the random oracle property ROP. On the other hand, let $\Pr[d = d' \mid \texttt{AskH} \vee \texttt{AskG}] = \beta$. Now, note that since we assumed $|\Pr[d = d'] - 1/2|$ to be non-negligible so must be $\Pr[\texttt{AskH} \vee \texttt{AskG}]$. Because otherwise, $|\Pr[d = d'] - 1/2|$ would immediately be negligible. But then, since S_Π succeeds with probability $\Pr[\texttt{AskH} \vee \texttt{AskG}]/(\ell_H + \ell_G)$, which would also be non-negligible because the queries can only be polynomially many. Given problem Π is hard, such an attacker A_{CPA} cannot exist. Therefore, C_{msig} is IND-CPA secure. □

Remark. Note that if we use the GDH assumption, S_Π could have checked whether particular queries include the critical value $(y_A^{x^*+k^*})$ and return the correct answer directly. In this case, we could proceed with the same proof above and at the end, we could say that S_Π succeeds with probability $\Pr[\texttt{AskH} \vee \texttt{AskG}]$, which even gets rid of the polynomial fraction.

Below, in Lemma 2, we are going to show that C_{msig} is plaintext aware. This security notion was first introduced by Bellare and Rogaway [9]. According to this initial definition given by [9], an encryption scheme is said to be *plaintext aware* (PA) if it is computationally hard for any polynomial time adversary to generate a valid ciphertext without knowing the corresponding plaintext with non-negligible probability. This guarantees that before the adversary learns any ciphertext, she must already be aware of its plaintext. This, in turn, means that the decryption oracle in an IND-CCA game would not be useful for the adversary. This is the intuition why we get IND-CCA2 security when we add PA to IND-CPA security. However, the model of [9] did not capture attacks where the adversary can obtain ciphertexts via eavesdropping. In [10], such attacks are also captured by giving the adversary access to an encryption oracle. But, of course, the adversary is prohibited to return the result of an encryption oracle query. With this refinement done, PA, together with IND-CPA security, correctly implies IND-CCA security. So, the plaintext awareness definition we adapt is as follows, borrowed from [10]:

Definition 2 (Plaintext Awareness). Let $PE = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let A be an attacker that is given access to random oracle(s) H and an encryption oracle $\hat{\mathcal{E}}_{pk}^H$, can output valid ciphertexts with a non-negligible probability ε. We denote this by $y \leftarrow A^{H,\hat{\mathcal{E}}_{pk}^H}(pk)$. Let K be an

algorithm (called *knowledge extractor*) which takes input-output records of H, ℓ_H, outputs of $\hat{\mathcal{E}}^H_{pk}$, C, and the output of A, y, as input, and outputs $D(y)$ or in case of an invalid ciphertext, it outputs $\perp$ with success probability λ. We denote this by $m = D(y) \leftarrow K(\ell_H, C, y, pk)$.

We say that E is plaintext aware if for all attackers A, there exists a K that has a non-negligible success probability λ.

A PA attacker A takes the public key pk, has access to random oracle H and an encryption oracle $\hat{\mathcal{E}}^H_{pk}$ and aims to produce a valid ciphertext without knowing its plaintext. We need to show that any such attacker has a negligible success probability. In PA proofs, an algorithm called *knowledge extractor* is constructed to show that the attacker can easily find the decryption of any ciphertext it produced without knowing the private key. In a sense, this means that the attacker *can readily know* the plaintext of any ciphertext it produces, which means that decryption oracle would have no use for it.

Lemma 2. *The encryption scheme C_{msig} is plaintext aware in the random oracle model if the GDH assumption holds in the underlying group.*

PROOF. Let P be a PA attacker that has access to public key y_A, oracles H and G, as well as an encryption oracle $\varepsilon^{H,G}_{y_A}$. We define S_P as the event where a ciphertext produced by a single run of P is valid and denote the probability of S_P with $\Pr[S_P]$. We will construct a knowledge extractor K, which can find the decryption of any ciphertext created by P, by observing its queries. We define S_K as the event where K succeeds. We will show that if $\Pr[S_P]$ is non-negligible, then so is $\Pr[S_K]$.

Let x_A and $y_A = g^{x_A}$ be the private and public keys, let $c = (c_1,\ c_2,\ c_3,\ c_4)$ be a ciphertext produced by P and let the queries of P be as follows:

$$\ell_H = ((\mu_1 \| u_1 \| v_1), H_1), ((\mu_2 \| u_2 \| v_2), H_2), \ldots, ((\mu_{\alpha_H} \| u_{\alpha_H} \| v_{\alpha_H}), H_{\alpha_H}),$$

$$\ell_G = (r_1, G_1), (r_2, G_2), \ldots, (r_{\alpha_G}, G_{\alpha_G}) \text{ and}$$

$$\ell_E = (m_1, C_1), (m_2, C_2), \ldots, (m_{\alpha_E}, C_{\alpha_E}).$$

On input $(y_A, c, \ell_G, \ell_H, \ell_E)$ where $c = (c_1,\ c_2,\ c_3,\ c_4)$, K works as follows:

1. It searches for c_2 in the list ℓ_H and upon finding $c_2 = H_i$, retrieves μ_i.

2. K calculates $\mu_i \oplus c_4$ and similar to the above reasoning, this value must appear as a G_j. Otherwise, that means no G query has returned $\mu_i \oplus c_4$ yet, therefore the probability that c_4 is consistent with c_2 is $1/|\mathcal{M}|$ due to the random oracle property ROP.
3. Then, K compares the oracle queries r_j and v_i. If they are equal, it concludes $r_j = v_i = {y_A}^{x+k}$.
4. K also checks whether $c_1 = u_i$.
5. Finally, K computes g^k from $g^{c_3}(c_1)^{c_2}$, and with the help of the DDH oracle, it checks whether $(g, g^x g^k, g^{x_A}, r_j)$ is a DDH tuple.
6. If all these checks succeed, K returns μ_i, otherwise it returns $\perp$.

First note that, once the second component, c_2, of a ciphertext is fixed, it implicitly determines all other components. Therefore, as long as the output of the attacker contains c_2 from any previously eavesdropped or obtained ciphertexts, all other components also have to be the same. However, it is not allowed to output a previously observed ciphertext. Therefore, we conclude that the attacker has to obtain a newly created c_2 and this would require a query to H necessarily. In such a case, K successfully finds an H_i that equals c_2 and as long as other components are consistent with c_2, it returns μ_i which has to be the correct message.

If the attacker creates an invalid ciphertext, on the other hand, K can detect its invalidity with the checks at steps (2) through (5) and return $\perp$.

Since the attacker's only success chances other than already knowing the plaintext are guessing either c_2 or c_4 exactly, we can write:

$$\Pr[S_P] \leq \Pr[S_K] + 1/q + 1/|\mathcal{M}|$$

which means that the K we constructed has a success probability close to that of the PA attacker P. □

The proof of the main theorem is therefore complete. Recall again that GDH assumption already implies the CDH assumption, and PA together with IND-CPA security implies IND-CCA2 security.

6. Conclusion

We modified the Zheng-Seberry encryption based on digital signatures by employing the Schnorr signature and we obtained a new IND-CCA2 secure encryption. We prove the security of this new scheme in the random oracle

model by proving it is IND-CPA secure and also plaintext aware. The new scheme is also faster than the original Zheng-Seberry encryption with digital signatures since it requires fewer number of exponentiations in encryption and decryption algorithms and no inversions are needed. As for future work, one might be interested in building a similar encryption scheme that will be proven secure in the standard model, or one that will be proven plaintext aware assuming CDH instead of GDH.

References

[1] Y. Zheng, J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *IEEE J. on Selected Areas in Comm.* 11(5): 715–724 (1993).

[2] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logaritms. *IEEE Trans. on Inf. Theory* 31(4): 469–472 (1985).

[3] M. Bellare, P. Rogaway. Random oracles are practical: A Paradigm for designing efficient protocols. *ACM CCS'93*.

[4] R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *CRYPTO'98*.

[5] M. Abe. Combining Encryption and Proof of Knowledge in the Random Oracle Model. *Computer J.* 47(1): 58–70 (2004). *CRYPTO'98*.

[6] W. Diffie, M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory* 22(6): 644–654, (1976).

[7] T. Okamoto, D. Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. *PKC'01*.

[8] C. P. Schnorr. Efficient identification and signatures for smart cards. *CRYPTO'89*.

[9] M. Bellare, P. Rogaway. Optimal Asymmetric Encryption: How to encrypt with RSA. *EUROCRYPT'94*.

[10] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway. Relations among notions of security for public-key encryption schemes. *CRYPTO'98*.