Concrete Security Treatment of Public-Key Cryptosystem against Adaptive Chosen Cipher-Text Attack Using PQK Public-Key Cryptosystem

Tamer M. Barakat

***Abstract*---**We propose a new public key cryptosystem which named PQK that based on the decisional Diffie-Hellman problem. The scheme is provably secure against adaptive chosen cipher-text attack under the hardness assumption of the decisional Diffie-Hellman problem. Compared with the RSA public key scheme, our scheme has nice features: (1) our scheme is provably secure against adaptive chosen cipher-text attack under the intractability paradigm, (2) the PQK is secure against other attacks such as common modulus attack and low exponent attack which the RSA is still suffered from these attacks, and (3) it is faster than, in the decryption process, the RSA cryptosystem.

***Keywords*---**RSA Cryptosystem, PQK Cryptosystem, Adaptive Chosen Ciphertext Attack, Low Exponent Attack, Common Modulus Attack.

I. Introduction

THE RSA cryptosystem is one of the most practical public key cryptosystems and is used throughout the world [1]. Let n be a public key, which is the product of two appropriate primes, e be an encryption key, and d be a decryption key. The algorithms of encryption and decryption consist of exponentiation to the e^{th} and d^{th} powers modulo n, respectively. We can make e small, but must consider low exponent attacks [2] [3]. The encryption process takes less computation and is fast. On the other hand, the decryption key d must have more than one fourth the number of bits of the public key n to preclude Wiener's attack [4] and its extension [5]. Therefore, the cost of the decryption process is dominant for the RSA cryptosystem.

If a cryptosystem has more than one block of plaintexts, where each block is as large as the public-key n, we call it a multi-block cryptosystem.

A lot of multi-block RSA-type cryptosystems have been proposed [6] [7] [8]. Their advantage is that they allow us to encrypt data larger than the public-key at a time, and we can prove their security is equivalent to the original RSA cryptosystem or factoring. However, these algorithms are very slow and the attacks against the RSA cryptosystem are also applicable to them (See, for example, [9] [10]).

We cannot find significant advantage over using the original RSA cryptosystem for each block. The RSA cryptosystem suffers from two main problems; the decryption process is very slow and it is insecure against adaptive chosen ciphertext attack, low exponent attack and common modulus attack.

Tsuyshi Takagi [11] presented a new public-key cryptosystem with fast decryption named n^k cryptosystem which is constructed over $Z/\ n^k Z$, where n is the modulus and k is a positive integer. To implement the n^k cryptosystems, we used only ordinary and elementary mathematical techniques such as computation of greatest common divisors, so that it is easy to implement. Moreover, the decryption time of the first block is dominant, because after the first block we only calculate the modular multiplication of the encryption exponent and an extended Euclidean algorithm to decrypt blocks after the first one. Therefore the n^k cryptosystem is faster in the decryption process compared with the previously reported RSA-type cryptosystems [12]. If a message is several times longer than a public-key n, we can encrypt this message fast without additionally using a symmetry-key cryptosystem. This cryptosystem solve the slowness of decryption process, but it still suffers from the mentioned attacks. Unfortunately, this scheme is also suffers from all attacks which the RSA scheme faced.

In this paper, we present and analyze a new public key cryptosystem which named PQK cryptosystem that is provably secure against adaptive chosen ciphertext attack (as defined byRackoff and Simon [13]). The scheme is quite practical, requiring just a fewexponentiations over a group. Moreover, the proof of security relies onlyon a standard intractability assumption, namely, the hardness of the Diffie-Hellman decision problem in the underlying group.

The hardness of the Diffie-Hellman decision problem is essentially equivalent to the semantic security of the basic El Gamal encryption scheme [11]. Thus, with just a bit more computation, we get security against adaptive chosenciphertext attack, whereas the basic El Gamal scheme is completely insecure against adaptive chosen ciphertext attack. Actually, the basic schemewe describe also requires a universal one-way hash function. In a typical implementation,this can be efficiently constructed without extra assumptions; however, we also present a hash-free variant as well.

Moreover, This paper will prove that the PQK cryptosystem has a modification of the encryption algorithm to enhance security against standard attacks such as the common modulus

Manuscript received on December 26, 2013, review completed on January 10, 2014 and revised on January 20, 2014.

Tamer M. Barakat is with Electrical Engineering Department, Faculty of Engineering, Fayoum University, Fayoum, Egypt E-Mail: tmb00@fayoum.edu.eg

Digital Object Identifier: NCE122013007.

attack and the low exponent attack are not applicable on the PQK cryptosystem.

Therefore, the public encryption key becomes a composite number and has the relation $\overline{e} = ea$, for an integer $a \geq 1$.

A. Chosen Cipher-text Security

The notion of semantic security (defined by Goldwasser and Micali [14]) captures the notion of security of a public key cryptosystem against chosen plaintext attack. It is now generally accepted that this is a basic requirement of a good cryptosystem. However, it also known that other, stronger attacks are possible, and moreover, security against these types of attacks is necessary to ensure the security of many higher-level protocols built on top of the cryptosystem.

A chosen ciphertext attack is one in which the adversary has access to a "decryption oracle," allowing the adversary to decrypt ciphertexts of his choice. Typically, one distinguishes between a weak form of this attack, known as a lunch-time attack (defined by Naor and Yung [15]), and the strongest possible form, known as an adaptive chosen ciphertext attack (defined by Rackoff and Simon [16]). In a lunch-time attack, the adversary queries the decryption oracle some number of times, after which, he obtains the target ciphertext that he wishes to cryptanalyze, and is not allowed query the decryption oracle further. In an adaptive attack, the adversary may continue to query the decryption oracle after obtaining the target ciphertext; subject only to the (obviously necessary) restriction that queries to the oracle may not be identical to the target ciphertext.

Security against adaptive chosen ciphertext attack also implies non- malleability (defined by Dolev, Dwork and Naor [18]), meaning that an adversary can't take an encryption of some plaintext and "message" it into an encryption of a different plaintext that is related in some interesting way to the original plaintext.

The rest of the paper is organized as follows. Section 2 describes the basic scheme of the PQK cryptosystem. Proof of correctness of the PQK cryptosystem presents in Section 3. Then we describe the proof of security of PQK cryptosystem against adaptive chosen ciphertext attack in Section 4. PQK cryptosystem immunity against other attacks presents in Section 5. Finally, we provide some concluding remarks in Section 6

B. Other Related Work

Provably Secure Schemes. For many years, no public key system was shown to be secure under a chosen ciphertext attack. Naor and Yung [15] presented the first scheme provable secure against lunch-time attacks. Subsequently, Dolev, Dwork, and Naor [19] presented a scheme that is provably secure against adaptive chosen ciphertext attack.

Unfortunately, all of the known schemes provably secure under standard intractability assumptions are completely impractical (albeit polynomial time), as they rely on general and expensive constructions for non-interactive zero-knowledge proofs.

Practical Schemes.Damgard [18] proposed a practical scheme that he conjectured to be secure against lunch-time attacks; however, this scheme is not known to be provably secure, and is in fact demonstrably insecure against adaptive chosen ciphertext attack. Zheng and seberry [20] propose practical schemes that are conjectured to be secure against chosen ciphertext attack, but again, no proof based on standard intractability assumptions is known.

Lim and Lee [21] also proposed practical schemes that were later broken by Frankel and Yung [22].

In a different direction, Bellare and Rogaway [24] have presented practical schemes that are provably secure against adaptive chosen ciphertext attack in an idealized model of computation where hash function is represented by a random oracle.

II. The PQK Scheme

Notation 1: Z is an integer ring. Z_n is a residue ring $Z/\ (pq)^k Z$ and its complete residue class is $\{0,1,2,...,n-1\}$. Z_n^x is a reduced residue group modulo n is the Least Common Multiple of m_1 and m_2. $GCD(m_1, m_2)$ is the Greatest Common Divisor of m_1 and m_2 .We also assume that we have a group G Which plaintext are elements of group G. we also assume that H is a hash function that hashes long strings to elements of Z_n.

- ***Key Generation.*** The key generation algorithm runs as follows:

- Generate two random primes p, q and let $n = pq^k$

- Compute $L = LCM(p-1, q-1)$ and find e, d which satisfies $ed \equiv 1 \pmod{L}$ and $GCD(e, p) = 1$, where e is the prime public encryption key before our modification and d is the corresponding secret decryption key.

- Let $\overline{e}$ be the modified public encryption key which is a composite number, where $\overline{e} = ea$, for an integer $a \geq 1$. Then $\overline{e}$, n are public keys and d, p and q are the secret keys.

- Random elements $g_1, g_2 \in G$ are chosen and random elements

$x_1, x_2, y_1, y_2, z_1, z_2, \in Z_n$ are also chosen.

- Compute the group elements

$b = g_1^{x_1} g_2^{x_2}$, $f = g_1^{y_1} g_2^{y_2}$, $h = g_1^{z_1} g_2^{z_2}$

Then public key is $(g_1, g_2, b, f, h, e, n)$ and the private key is $(x_1, x_2, y_1, y_2, z_1, z_2, d)$.

- ***Encryption.*** The encryption algorithm runs as follows: let $M_0 \in Z_n^x$ and $M_1, \ldots\ldots, M_{k-1} \in Z_n$ be the plaintext. We chose $r \in Z_n$ at random. Then it computes:

$$u_1 = g_1^r, u_2 = g_2^r, \alpha = H(u_1, u_2\ C), and\ v = b^r f^{r\alpha}$$

$$C \equiv h^r \left(M_0 + nM_1 + ... + n^{k-1} M_{k-1}\right)^{\bar{e}} \pmod{pq^k} \quad (1)$$

The ciphertext is (u_1, u_2, C, v)

- ***Decryption.***Given a ciphertext (u_1, u_2, C, v), the decryption algorithm runs as follows. It first computes $\alpha = H(u_1, u_2\ C)$, and then tests if $u_1^{x_1} u_2^{x_2} \left(u_1^{y_1} u_2^{y_2}\right)^{\alpha} = v.$

If this condition does not hold, the decryption algorithm output “reject”; otherwise, it outputs:

First, we decrypt the first block M_0

$$: M_0 \equiv C^d \Big/ \left(u_1^{z_1} u_2^{z_2}\right) \quad (2)$$

Then, for the remaining blocks $M_1, M_2, ...M_{k-1}$, we can decrypt by solving the linear equation modulo n with the fast decryption that described in [23]

- ***Details of Decryption***

Assume that, we have already decrypted M_0 by the decryption method in equation (2), and we write down the process to find M_1, M_2, ..., M_{k-1} as follows.

Consider that the encryption function (1) is the polynomial of the variables $X_0, X_1, ..., X_{k-1}$ such that:

$$E(X_0, X_1, ..., X_{k-1}) =$$

$$h^r (X_0 + nX_1 + n^2 X_2 + ... + n^k X_{k-1})^{ea}$$

Expand the polynomial $E(X_0, X_1, X_2, \dots, X_{k-1})$ by the polynomial theorem:

$$h^r \Big(\sum_{\substack{0 \le S_0, S_1, \cdots S_{K-1} \le ey \\ S_0 + S_1 + \cdots + S_{K-1} = ey}} \frac{ea!}{S_0! S_1! \cdots S_{k-1}!} X_0^{S_0} (nX_1)^{S_1} \cdots (n^{k-1} X_{k-1})^{S_{K-1}} \Big)$$

And let:

$$\Gamma_i := \left\{ (S_0, S_1, ...S_i) \,\middle|\, \begin{array}{l} S_1 + 2S_2 + ... + iS_i = i, \\ S_0 + S_1 + ... + S_i = ea, 0 \le S_0, S_1, ...S_i \le ea \end{array} \right\}$$

Where $(0 \le i \le k-1)$ and $a \ge 1$.

Let $D_i(X_0, X_1, ..., X_i)$ be the coefficient of n^i $(0 \le i \le k-1)$, we can find $D_i(X_0, X_1, ..., X_i)$ by calculating:

$$D_i(X_0, X_1, ..., X_i) =$$

$$h^r \Big(\sum_{(S_0, S_1, \cdots, S_i) \in \Gamma_i} \frac{e\,a!}{S_0! S_1! \cdots\cdots S_i!} X_0^{S_0} X_1^{S_1} \cdots\cdots X_i^{S_i} \Big)$$

Here, we write them down with small i as follows:

$$D_0(X_0) = h^r X_0^{ea} = h^r M_0^{ea}$$

$$D_1(X_0, X_1) = h^r eaM_0^{ey-1} M_1,$$

$$D_2(X_0, X_1, X_2) = h^r (eac_2 M_0^{ea-2} M_1^2 + eaM_0^{ea-1} M_2),$$

$$\vdots$$

$$D_{k-1}(X_0, X_1, ..., X_{k-1}) = \{polynomial\ of\ M_0, M_1, ..., M_{k-1}\}$$

Where $c_2, c_3, ...$ are constants.

Note that: the only term that includes X_i in D_i is $h^r (eaX_0^{ea-1} X_i)$.

We define:

$$D_i'(X_0, X_1, ..., X_{i-1}) = D_i(X_0, X_1, ., X_i) - h^r (eaX_0^{ea-1} X_i)$$

Therefore, the terms $D_0, D_1, ..., D_{i-1}, D_i'$ are the polynomial of $X_0, X_1, ..., X_{i-1}$.

From this relation, we can decrypt $M_1, M_2, ..., M_{k-1}$. Indeed, $M_1, M_2, ..., M_{k-1}$ are calculated as follows:

by setting $i = 1$,

the relations $D_1' = (X_0) = 0\ and\ D_0\ (X_0) = h^r X_0^{ea}$. So, the solution of the linear equation:

$$eM_0^{ea-1}\ x \equiv B_1 \big/ h^r \pmod{n},$$

$where\ \ B_1 \equiv C - (D_0 M_0) \pmod{pq}^2$, is M_1, then we can decrypt $M_2, M_3, ..., M_{k-1}$ by solving the general linear equation:

$$ea\, M_0^{ea-1}\ x \equiv B_i \big/ h^r \pmod{n},$$

$$B_i \equiv C - \sum_{j=0}^{i-1} \begin{array}{l} D_j(M_0, M_1, ..., M_j) - \\ D_i'(M_0, M_1, ..., M_{i-1}) \end{array} \pmod{n^{i+1}}$$

Inductively, we can decrypt all plaintexts $M_1, M_2, ..., M_{k-1}$.

III. Proof of Correctness for PQK Construction

In this section, we prove the correctness for our construction which discuss how can successfully recovered the original message M after encrypt it using thePQK cryptosystem.

- Proof of correctness

From the construction of PQK cryptosystem we can see that:

$$M = (M_0 + nM_1 + n^2 M_2 ..... + n^{k-1} M_{k-1}) \quad (\bmod pq)^k$$

Then, the value of M is correct if and only if the values of $M_0, M_1, ..., M_{k-1}$ are correct values. Therefore, we consider another assumption:

$$M = (M'_0 + nM'_1 + .n^2 M'_2 ... + n^{k-1} M'_{k-1}) \quad (\bmod pq)^k \quad (3)$$

Hence, to achieve the main purpose it must be used our analysis to prove that,

$$M'_0 = M_0,\ M'_1 = M_1, ..., M'_{k-1} = M_{k-1} \ .$$

Proof that $M'_0 = M_0$

From equation (3) let $k = 1$ and $i = 0$. Then, $M = M'_0 \quad (\bmod n)$

The general linear equation that find $M_1, M_2, ..., M_{k-1}$ is given by:

$$eaM_0^{ea-1} x \equiv B_i / h^r \ (\bmod pq),$$

$$B_i \equiv C - \sum_{j=0}^{i-1} D_j (M_0, M_1, ...M_j) - D'_i (M_0, M_1, ...M_{i-1}) \ (\bmod n^{i+1})$$

$$B_0 = C - [D_0(M_0) - eaM_0^{ea-1} M_0] = C - h^r (M_0^{ea} + eaM_0^{ea})$$

$$C = h^r M_0^{ea}, then B_0 = h^r eaM_0^{ea}$$

$$M'_0 = x = \frac{B_0}{h^r eaM_0^{ea-1}} = \frac{h^r eaM_0^{ea} M_0}{h^r eaM_0^{ea}} = M_0$$

Hence, the value of M_0 is correct.

Proof that $M'_1 = M_1$

Then, $k = 2, i = 1$ and $j = 0$.

From equation (3) $M = (M'_0 + nM'_1) \quad (\bmod pq)^2$

The linear equation that find M_1 is

$$eaM_0^{ea-1} x \equiv B_1 / h^r \ (\bmod n),$$

$$where\ B_1 \equiv C - D_0(M_0) - D'_1(M_0, M_1) \quad (\bmod pq)^2$$

Then, $x = M'_1 = \dfrac{B_1}{h^r eaM_0^{ea-1}}$

Now, we can compute each value of B_1 as follows:

The value of D'_1 can be calculated as follows:

$$D'_1(X_0) = D_1(X_0, X_1) - h^r\ eaX_0^{ea-1} X_1$$

$$D'_1(M_0) = h^r (eaM_0^{ea-1} M_1 - eaM_0^{ea-1} M_1)$$

$$then, D'_1(M_0, M_1) = 0.$$

The encryption function of M is given by the following equation:

$$C \equiv h^r \left(M_0 + nM_1 + n^2 M_2 + ... + n^{k-1} M^{k-1}\right)^{ea} \left(\bmod pq\right)^k$$

In this case:

$$C \equiv h^r (M'_0 + nM'_1)^{ea} \qquad (\bmod pq)^2$$

From the polynomial theorem we can get:

$$\sum_{(S_0, S_1, \cdots\cdots, S_i) \in \Gamma_i} \frac{ea!}{S_0! S_1! \cdots\cdots S_i!} X_0^{S_0} (nX_1^{S_1}) \cdots\cdots (n^i X_i^{S_i})$$

$$\text{Where } \Gamma_i := \left\{ (S_0, S_1, ...S_i) \middle| \begin{array}{l} S_1 + 2S_2 + ... + iS_i = i, \\ S_0 + S_1 + ... + S_i = ea, 0 \le S_0, S_1, ... S_i \le ea \end{array} \right\}$$

Then, $C = h^r (M_0^{ea} + e\,anM_0^{ea-1} M_1)$

Hence,

$$B_1 = h^r (e\,aM_0^{ea-1} M_1 + M_0^{ea} - M_0^{ea}) = h^r eaM_0^{ea-1} M_1$$

Then, $M'_1 = \dfrac{B_1}{h\ eaM_0^{ea-1}} = \dfrac{h^r eaM_0^{ea-1} M_1}{h^r eaM_0^{ea-1}}$

Then, $M'_1 = M_1$

Hence, the value of M_1 is correct.

By the same manner, we can prove that the values of $M_2, ..., M_{k-1}$ are the correct values. So that our construction is correct to recover the original message after encrypt it using the proposed cryptosystem.

IV. Proof of Security

In this section, we prove the following theorem.

Theorem 1*The above cryptosystem is secure against adaptive chosen ciphertext attack assuming that (1) the hash function H is collision resistant, and (2) the Diffie- Hellman decision is hard in the group G.*

Beforegoing into the proof, we recall the meaning of the technical terms in the above theorem.

Security against adaptive chosen ciphertext attack. Security is defined via the following game played by the adversary:

First, the key generation makes arbitrary queries to a "decryption oracle," decrypting ciphertexts of his choice.

Next theadversary chooses two messages, m_0, m_1, and sends these to an "encryption oracle." The decryption oracle chooses a bit $b \in \{0, 1\}$ at random, and encrypt m_b . The corresponding ciphertext is given to the adversary (the internal coin tosses of the encryption oracle, in particular b , are not in the adversary's view).

After receiving the ciphertext from the encryption oracle, the adversay continues to query the decryption oracle, subject to the restriction that the query must be different than the output if the encryption oracle.

At the end of the game, the adversary outputs $b' \in \{0, 1\}$, which is supported to be the adversary's guess of the value b . If the probability that $b' = b$ is $\frac{1}{2} + \varepsilon$, then the adversary's *advantage* is defined to be ε .

The cryptosystem is said to be secure against adaptive chosen ciphertext attack if the advantage of any polynomial-time adversary is negligible.

Collision resistant hash functions. A family of hash functions is collision resistant if given a random hash function H in the family, it is infeasible to find a collision, i.e., two strings $x \neq y$ such that $H(x) = H(y)$.

*The Diffie-Hellman decision problem.*Let G be a group of prime order order q, and consider the following two distributions:

The distribution **R** of quadruples (g_1, g_2, u_1, u_2), where g_1, g_2, u_2, u_2 are chosen at random.

The distribution **D** of quadruples (g_1, g_2, g_1^r, g_2^r), where $g_1, g_2 \in G$ are chosen at random, and $r \in Z_q$ is chosen at random.

An algorithm that solves the Difie-Hellman decision problem is a statistical test that can distinguish the two distributions. That is, given a quadruple coming from one of the two distributions, it should output 0 or 1, and there should be a non-negligible difference between (a) the probability that it outputs a 1 given an input from**R**, and (b) the probability that it outputs a 1 given an input from **D**. the Diffie-Hellman decision problem is hard if there is no such polynomial-time statistical test.

- ***Proof of Theorem***

To prove the theorem, we will assume that there is an adversary that can break the cryptosystem, and show how to use this adversary to construct a statistical test for the Diffie-Hellman decision problem.

For the statistical test, we are given (g_1, g_2, u_1, u_2) coming from either the distribution **R** or **D**. at a high level, our construction works as follows. We build a simulator that simulates the joint distribution consisting of adversary's view in its attack on the cryptosystem, and the bit b generated by the decryption oracle (which is not a part of the adversary's view). It will be clear that from if the input happens to come from **D**, the simulation of this joint distribution is perfect, and so the adversary has a non-negligible advantage. We then show that if the input happens to come from **R**, then the adversary's view is essentially independent of b , and therefore the adversary's advantage isnegligible.

We now give the details of the simulator. The input to the simulator is (g_1, g_2, u_1, u_2). The simulator runs the key generation algorithm and it chooses:

$$x_1, x_2, y_1, y_2, z_1, z_2, \in Z_n$$

The public key that the adversary sees is $(g_1, g_2, b, f, h, e, n)$. The simulator knows the corresponding private key $(x_1, x_2, y_1, y_2, z_1, z_2, d)$. The simulator answers decryption quires as in the actual attack, which it can do since it knows the private key.

We now describe the simulation of the encryption oracle. Given m_0, m_1, the simulator chooses $b \in \{0, 1\}$ at random, and computes

$$C \equiv u_1^{z_1} u_2^{z_2} (M_0)^{\bar{e}} \pmod{n},$$
$$\alpha = H(u_1, u_2\ C), \mathit{and}\ v = u_1^{x_1} u_2^{x_2} \left(u_1^{y_1} u_1^{y_2}\right)^{\alpha},$$

And outputs

$$(u_1, u_2, C, v)$$

That completes the description of the simulator.

First, consider the joint distribution of the adversary's view and the bit b when the input comes from the distribution **D**. say $u_1 = g_1^r$ and $u_2 = g_2^r$. The is clear that $u_1^{x_1} u_2^{x_2} = b^r$, $u_1^{y_1} u_2^{y_2} = f^r$ and $u_1^{z_1} u_2^{z_2} = h^r$. From this is clear that the joint distribution of the adversary's view and b is identical to that in the actual attack.

Second, consider that the distribution of the adversary's view and the bit b when the input comes from **R**. we want to show that the adversary's view and b are essentially independent.

Notation 2: let $\log(.)$ denote the logarithm to the base g_1 , and let $w = \log g_2$. Let $u_1 = g_1^{r_1}$ and $u_2 = g_1^{wr_2}$ where $r_1 \neq r_2$. Also let us define (u_1', u_2', C', v') to be a "valid ciphertext" if there exists $r' \in Z_n$ such that $u_1' = g_1^{r_1''}$ and $u_2' = g_2^{r_1''}$. Otherwise, we will say it is an "invalid ciphertext."

Claim 1. *If the decryption oracle rejects all invalid ciphertexts during the attack, then b is independently distributed from the adversary's view.*

To see this, consider the pair (z_1, z_2). At the beginning of the attack, this is a random point on the line $z_1 + wz_2 = \log h$ (this is the information about (z_1, z_2) leaked by the public key). Moreover, if the decryption oracle only decrypts valid ciphertext (u_1', u_2', C', v'), then the adversary obtains only linearly dependant relations $r'z_1 + r'wz_2 = r' \log h$ (Since $(u_1')^{z_1} (u_2')^{z_2} = g_1^{r'z_1} g_2^{r'z_2} = h^{r'}$). Thus, no information about (z_1, z_2) is leaked.

Consider now the output of the simulated encryption oracle; we have:

$$\begin{pmatrix} \log\ h \\ \log\ C^d \big/ M_b \end{pmatrix} = \begin{pmatrix} 1 & w \\ r_1 & wr_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.$$

Since the matrix in the above equation is nonsingular, for each choice of $b \in \{0, 1\}$, there exists exactly one solution (z_1, z_2). This is implies that the distribution of b is independent of the adversary's view.

Claim 2. Assuming the adversary does not find a collision in H, then the decryption oracle will reject all invalid ciphertexts during the attack.

To prove this claim, we study the distribution of $(x_1, x_2, y_1, y_2) \in Z_n$ as seen by the adversary. From the adversary's view, this is essentially a random point on the line formed by interesting the hyper planes:

$$x_1 + wx_2 = \log b$$
$$y_1 + wy_2 = \log f$$
$$r_1 x_1 + wr_2 x_2 + (\alpha\ r_1) y_1 + (\alpha\ wr_2) x_2 = \log v\,,$$

The first two equations come from the public key, and the third comes from the output of the decryption oracle.

Also note that decrypting a valid ciphertext leaks no information about the point (x_1, x_2, y_1, y_2).

The above considerations imply that it suffices to consider what happens when the adversary presents a single invalid ciphertext $(u_1', u_2', C', v') \neq (u_1, u_2, C, v)$ to the decryption oracle.

First, assume that $(u_1', u_2', C') = (u_1, u_2, C)$. In this case, the hash value is the same, but $v' \neq v$ implies that the decryption oracle will certainly reject.

Second, assume that $(u_1', u_2', C') \neq (u_1, u_2, C)$. Let $\alpha' = H(u_1', u_2', C')$ and $\alpha = H(u_1, u_2, C)$. We are assuming, by collision intractability, that $\alpha' \neq \alpha$.

Let $u_1' = g_1^{r_1'}$ and $u_2' = g_1^{wr_2'}$, where $r_1' \neq r_2'$ (since the ciphertext is invalid). The decryption oracle will not reject if and only if $v' = v''$, where:

$$v'' = (u_1')^{x_1}\ (u_2')^{x_2}\ \left((u_1')^{y_1}\ (u_2')^{y_2}\right)^{\alpha'}$$

Consider the following matrix:

$$\lambda = \begin{pmatrix} 1 & w & 0 & 0 \\ 0 & 0 & 1 & w \\ r_1 & wr_2 & \alpha\, r_1 & \alpha\, wr_2 \\ r_1' & wr_2' & \alpha'\, r_1' & \alpha'\, wr_2' \end{pmatrix}.$$

It will to show that λ is nonsingular, because even when the adversary sees the first three entries of the vector:

$$\begin{pmatrix} \log b \\ \log f \\ \log v \\ \log v'' \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix},$$

The last entry of this vector will be independent of the adversary's view. But then the probability that $\log v' = \log v''$ is negligible, and when equality does not hold, the decryption oracle will reject.

To finish the proof, we only need to show that λ is nonsingular. We can easily verify that

$$\det(\lambda) = w^2 (r_2 - r_1)(r_2' - r_1')(\alpha - \alpha') \neq 0.$$

That completes the proof of security.

V. The PQK Cryptosystem Immunity to Other Attacks

In this section, we explain the effectiveness of low exponent attack and common modulus attack against the PQK cryptosystem.

- **Low Exponent Attack**

A low public exponent is desirable to reduce encryption time. However, there is a powerful attack on low public exponent for RSA cryptosystem based on a theorem due to Coppersmith [25]. Hastad reported an attack, named low exponent attack based on Coppersmith's theorem, which detects a low public exponent e. This attack is effective for $N \geq e$, where N is a number of parities that receive the same message M at the same time with common public exponent e; on the other hand the public key for the PQK cryptosystem has the relation: $\overline{e} = e\,a$, for an integer $a \geq 1$. Therefore, if the attacker can achieve Coppersmith's constraint such that $N \geq ea$, then, our modified cryptosystem will be broken.

Here, we discuss Hastad's attack. For simplicity, suppose that the same message M has to be sent to three different users and all corresponding public exponents are equal to 3. Therefore, the modified public key becomes $\overline{e} = e\,a$ = 3*2, where a =2.

Let the original message $M = 4$ (assuming that the message has one block). Then, we calculate the modulus n for each message as follows:

For message #1:

$$p_1 = 3,\ q_1 = 5,\ n = 15\ and\ k = 2 \Rightarrow n_1^k = 2025.$$

For message #2:

$$p_2 = 3,\ q_2 = 7, n_2 = 21 \ and\ k = 2 \Rightarrow n_2^k = 441.$$

For message #3:

$$p_3 = 5,\ q_3 = 7, n_3 = 35 \ and\ k = 2 \Rightarrow n_3^k = 1225.$$

Assuming that, $g_1 = 2, g_2 = 3, z_1 = 2, z_2 = 3, and\ r = 2$. Then, $h^r = 1296$

Therefore, the corresponding ciphertexts are:

$$C_1 = 1296\,(4)^6 \quad \text{mod}\, 2025 \quad = 891,$$
$$C_2 = 1296\,(4)^6 \quad \text{mod}\, 441 \quad = 99, \quad and$$
$$C_3 = 1296\,(4)^6 \quad \text{mod}\, 1225 \ = 491.$$

Applying the Chinese Remainder Theorem (CRT) to C1, C2, and C3 as follows:

$$M^6 = C_1 + C_2 + C_3 \quad (\text{mod } N)$$
$$M = 3 \quad (\text{ mod } 496125\,)$$

Therefore, the attacker gets the value of message equal 2 whereas the original message is equal 4. Then, the attacker gets a hard problem, he always gets the wrong value of the message M . this means that he did not know if the value of message M is true or false. This problem takes place because Coppersmith's constraint is not satisfied. To successfully mount this attack he must get at least six messages to satisfy Coppersmith's constraint, but this way will increase the effort of the attacker to recover the correct value of the original message. Then, when using the PQK cryptosystem with small public exponent, the attacker will meet a hard problem when attempt to break this system using low exponent attack. The following tables summarize the numbers of messages required to mount a successful Hastad's attack.

Table 1. The number of messages required to mount a successful Hastad's attack for PQK cryptosystem before our modification. Note that: the system is not defined for even values of e.

TABLE 1

e	3	5	7	9	11	13
# of messages	4	6	8	10	12	14

Table 2. The number of messages required to mount a successful Hastad's attack for PQK cryptosystem after our modification. Note that: (*) means that the attack is not applicable which means that the attacker meets a hard computation to get the correct value of the original message.

TABLE 2

e	$\bar{e}$ (a =2)	# of messages
3	6	7
5	10	11
31	62	*

From Table 1 we observe that, at $e = 3$, the attacker just needs four messages to successfully break the PQK cryptosystem, whereas at $\bar{e} = 6$ in Table 2 the attacker will need at least seven messages to break that cryptosystem.

- Common Modulus Attack

Simmon pointed out in [26] that the use of a common RSA modulus is dangerous, indeed, if a message M is sent to two users that have comprised public encryption keys, then the message can be recovered. On the other hand, the PQK cryptosystem presented by Tsuyoshi Takagi also suffer from this attack because the public encryption key still prime number. Here, we discuss the effectiveness of the common modulus attack on the security of the modified PQK cryptosystem. Suppose that we need to send a message M to two users that have the ciphertexts given by (Assuming that the message consists of one block):

$C_1 \equiv h^r M^{\bar{e}_1}\ (\text{mod}\, n)$ and $C_1 \equiv h^r M^{\bar{e}_2}\ (\text{mod}\, n)$, where $\bar{e}_1, \bar{e}_2$ are the public encryption keys for common modulus $(n = pq^k)$ for the PQK cryptosystem.

The attacker attempts to break this system using one of the following methods:

Method #1:

From our modification, we can see that $\gcd(\bar{e}_{1,}\ \bar{e}_2) \neq 1$ because these values become a composite numbers. So, when the attacker attempt to use extended Euclidean algorithm to find o, p such that $o\bar{e}_1 + p\bar{e}_2 = 1$, where $o\ and\ p$ are non-negative integers, he fail to satisfy this equation because $\bar{e}_1, \bar{e}_2$ are composite numbers, i.e. $o\bar{e}_1 + p\bar{e}_2 \neq 1$.

Therefore, he cannot use the following relation to recover the original message:

$$M \neq M^{o\bar{e}_1 + p\bar{e}_2} \not\equiv C_1^o / h^r \ C_2^p / h^r \ (\text{mod}\, n)$$

Method #2:

The attacker succeeds to factorize the composite public key $\bar{e}_1, \bar{e}_2$ to $\bar{e}_1 = e_1 * a \quad and\ \bar{e}_2 = e_2 * a$ respectively.

Then, he begins his attempt to recover the original message as follows:

Let $o\bar{e}_1 + p\bar{e}_2 = 1$

Then, $o e_1\, a + p\, e_2\, a = 1$

$(o\,a)e_1 + (p\,a)e_2 = 1$

$o' e_1 + p' e_2 = 1 \Rightarrow$ This relation can be obtained from extended Euclidean algorithm.

Therefore, $M = M^{o' e_1 + p' e_2} \Rightarrow M = [M^{e_1}]^{o'} * [M^{e_1}]^{p'}$ (8)

But, $M^{e_1} \neq C_1 / h^r$ as well as $M^{e_2} \neq C_2 / h^r$.

So, the attacker cannot use the equation to recover the original message.

Hence, the common modulus attack seems infeasible for the PQK cryptosystem.

VI. Conclusions

We have devolped a new practical public-key cryptosystem based on the notion called decisionalDiffie-Hellman problem.

We concluded that the PQK cryptosystem is provably secure against adaptive chosen ciphertext attack since the hash function H is collision resistant and the Diffie-Hellman decision problem is seemed hard. Also we proved that the PQK cryptosystem is secure against common modulus attack because the public encryption key became a composite number.

On the other hand, we showed that it is possible to use the PQK cryptosystem with a composite small encryption key $\bar{e}$ provided Coppersmith's condition $N < \bar{e}$, where N is the number of parties that receive the same message at the same time. This condition is used to defend against the low exponent attack. Therefore, the Low exponent Attack seems infeasible. From this paper we suggested the modulus n to be for example 1024-bit for the 341-bit primes p and q , in order to make both the elliptic curve method and the number field sieve infeasible. So, this modulus is secure against the fast factoring algorithms.

Acknowledgment

The author would like to thank the anonymous reviewers of theCiiT for their valuable comments.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, \A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21, (1978), pp.120-126.

[2] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, \Low-exponent RSA with related messages," Advances in Cryptology { EUROCRYPT '96, LNCS 1070, (1996), pp.1-9.

[3] D. Coppersmith, \Finding a small root of a univariate modular equation," Advances in Cryptology { EUROCRYPT '96, LNCS 1070, (1996), pp.155{165.

[4] M. J. Wiener, \Cryptanalysis of short RSA secret exponents," IEEE Transactions on Information Theory, IT-36, (1990), pp.553-558.

[5] E. R. Verheul and H. C. A. van Tilborg, \Cryptanalysis of `less short' RSA secret exponents," Applicable Algebra in Engineering, Communication and Computing, 8, (1997), pp.425-435.

[6] N. Demytko, \A new elliptic curve based analogue of RSA," Advances in Cryptology { EUROCRYPT '93, LNCS 765, (1994), pp.40-49.

[7] K. Koyama, U. M. Maurer, T. Okamoto, and S. A. Vanstone, \New public-key schemes based on elliptic curves over the ring Zn," Advances in Cryptology { CRYPTO '91, LNCS 576, (1992), pp.252-266.

[8] K. Koyama, \Fast RSA-type schemes based on singular cubic curves," Advances in Cryptology { EUROCRYPT '95, LNCS 921, (1995), pp.329-340.

[9] B. S. Kaliski Jr., \A chosen message attack on Demytko's elliptic curve cryptosystem," Journal of Cryptology, 10, (1997), pp.71-72.

[10] T. Takagi and S. Naito, \The multi-variable modular polynomial and its applications to cryptography," 7th International Symposium on Algorithm and Computation, ISAAC'96, LNCS 1178, (1996), pp.386-396.

[11] T. Takagi. New public-key cryptosystem with fast decryption. Advances in Cryptology (PhD Thesis) - LNCS 1294, Germany, 2001.

[12] C. Racko_ and D. Simon.Noninteractive zero-knowledge proof of knowledgeand chosen ciphertext attack. In Advances in Cryptology{Crypto'91, pages 433{444, 1991.

[13] N. Demytko. A new elliptic curve based analogue of RSA. Advances in Cryptology {EUROCRYPT '93, LNCS 765, (1994), pp.40-49.

[14] T. El Gamal. A public key cryptanalysis and signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory, 31:469-472, 1985.

[15] S. Goldwasser and S. Micali.Probabilistic encryption. Journal of computer and system Scinces, 28:270-299, 1984.

[16] M. Maor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In 22nd annual ACM symposium on technology of computing, pages 427-437, 1990.

[17] C. Rackoff and D. Simon.Noninteractive zero-knoledge proof of knowledge and chosen ciphertext attack. In advances in cryptography-crypto'91, pages 433-444, 1991.

[18] D. Dolv, C. Dwork, amd M. Naor. Non-malleable cryptography. In 23rd annual ACM symposium on theory of computing, pages 542-552, 1991.

[19] I. Damgard. Towords practical public key cryptosystems secure against chosen ciphertext attacks. In advances in cryptology-crypto' 91, pages 445-456, 1991.

[20] Y. Zheng and J. Seberry. Practical approaches to attaining security against adaptively chosen ciphertext attacks. In advances in cryptology-crypto'92, pages 292-304, 1992.

[21] C. H. Lim and P. J. Lee. Another method for attaining security against adaptively chosen ciphertext attacks. In advances in cryptology-crypto'93, pages 420-434, 1993.

[22] Y. Frankel and M. Yung.Cryptanalysis of immunized LL public key systems. In advances in cryptology-crypto '95, pages 287-296, 1995.

[23] T. Takagi, \Fast RSA-type cryptosystem using n-adic expansion," Advancesin Cryptology { CRYPTO '97, LNCS 1294, (1997), pp.372{384.

[24] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In first ACM conferences on computer and communications security, 1993.

[25] D.Coppersmith. Small solutions to polynomials equations and low exponent RSA vulnerabilities. 1996

[26] G. L. Simmons. A 'weak' privacy protocol using the RSA crypto algorithm. Cryptology 7 1993, pp. 180-182.

Tamer Barakat received his BSc in communications and computers engineering from Helwan University, Cairo; Egypt in 2000. Received his MSc in Cryptography and Network security systems from Helwan University in 2004 and received his PhD in Cryptography and Network security systems from Cairo University in 2008. Currently, working as a lecturer, post doctor researcher and also joining several network security projects in Egypt. His main work is Cryptography and network security. More specially, he is working on the design of efficient and secure cryptographic algorithms, in particular, security in the wireless sensor networks.