Computationally-Efficient Password Authenticated Key Exchange Based on Quadratic Residues

Muxiang Zhang

Verizon Communications Inc.
40 Sylvan Road, Waltham, MA 02451, USA
muxiang.zhang@verizon.com

Abstract. In this paper, we present a computationally efficient password authenticated key exchange protocol based on quadratic residues. The protocol, called $QR\text{-}CEKE$, is derived from the protocol $QR\text{-}EKE$, a previously published password authenticated key exchange protocol based on quadratic residues. The computational time for the client, however, is significant reduced in the protocol $QR\text{-}CEKE$. In comparison with $QR\text{-}EKE$, the protocol $QR\text{-}CEKE$ is more suitable to an imbalanced computing environment where a low-end client device communicates with a powerful server over a broadband network. Based on number-theoretic techniques, we show that the computationally efficient password authenticated key exchange protocol is secure against *residue attacks*, a special type of off-line dictionary attack against password-authenticated key exchange protocols based on factorization. We also provide a formal security analysis of $QR\text{-}CEKE$ under the factoring assumption and the random oracle model.

1 Introduction

Password-authenticated key exchange protocols allow two entities who share a small password to authenticate each other and agree on a large session key between them. Such protocols are attractive for their simplicity and convenience and have received much interest in the research community. A major challenge in designing password-authenticated key exchange protocols is to deal with the so-called exhaustive guessing or off-line dictionary attack, as passwords are generally drawn from a small space enumerable, *off-line*, by an adversary. In 1992, Bellovin and Merritt [3] presented a family of protocols, known as *Encrypted Key exchange* (EKE), which was shown to be secure against off-line dictionary attack. Following EKE, a number of protocols for password-based authentication and key exchange have been proposed; a comprehensive list of such protocols can be found in Jablon's research link [6]. Over the last decade, many researchers have investigated the feasibility of implementing EKE using different types of public-key cryptosystems such as RSA, ElGamal, and Diffie-Hellman key exchange. Nonetheless, most of the well-known and secure variants of EKE

K. Srinathan, C. Pandu Rangan, M. Yung (Eds.): Indocrypt 2007, LNCS 4859, pp. 312–321, 2007.

are based on Diffie-Hellman key exchange. It seems that EKE works well with Diffie-Hellman key exchange, but presents subtleties one way or the other when implemented with RSA and other public-key cryptographic systems. In their original paper [3], Bellovin and Merritt pointed out that the RSA-based EKE variant is subject to a special type of dictionary attack, called residue attack. In 1997, Lucks [7] proposed an RSA-based password-authenticated key exchange protocol (called OKE) which was claimed to be secure against residue attacks. Later, Mackenzie et al. [8] found that the OKE protocol is still subject to residue attacks. In [8], Mackenzie et al. proposed an RSA-based EKE variant (called SNAPI) and provided a formal security proof in the random oracle model. Unfortunately, the SNAPI protocol has to use a prime public exponent e which is larger than the RSA modulus n. This renders the SNAPI protocol impractical in resource-limited platforms. To avoid using large public exponents, Zhu et al. [28] proposed an "interactive" protocol which is revised from an idea of [3]. The interactive protocol requires a large communication overhead in order to verify the RSA public key. Bao [1] and Zhang [14] also pointed out some weaknesses of Zhu et al.'s password-authenticated key exchange protocol. In 2004, Zhang [12] presented an RSA-based password authenticated key exchange protocol (called $PEKEP$) which can use both large and small primes as RSA public exponents, but without inducing large communication overhead on communication entities. Alternatively, Zhang [13] also presented a password authenticated key exchange protocol (called $QR\text{-}EKE$) based on quadratic resides.

In comparison with the RSA based password authenticated key exchange protocol $PEKEP$, the quadratic residue based password authenticated key exchange protocol $QR\text{-}EKE$ has a merit that one of the entities (i.e., client) does not need to perform primality test. In the protocol $QR\text{-}EKE$, the computational time for both entities, i.e., client and server, is $\mathcal{O}(\log_2 n)^3$. In many applications, however, it is highly desirable that the computational time for a low-end client device be much less than $\mathcal{O}(\log_2 n)^3$, in order to support resource-limited computing platforms, such as mobile phones and personal digital assistants. To reduce the computational burden on client devices, we present a computationally efficient password authenticated key exchange protocol in this paper. The protocol, called $QR\text{-}CEKE$, is derived from $QR\text{-}EKE$ by adding two additional flows between the client and the server. The two additional flows increase the communication overhead by $\log_2 n + 2k$ bits, where k is the security parameter (e.g., $k = 160$). With the two additional flows, we show that the probability for an adversary to launch a successful residue attack against $QR\text{-}CEKE$ is less than or equal to 2ε, where ε is a small number (e.g., $0 < \varepsilon \leq 2^{-80}$) selected by the client. In the protocol $QR\text{-}CEKE$, the computational time for the client is $\mathcal{O}(\log_2 \varepsilon^{-1}(\log_2 n)^2)$, which is much less than $\mathcal{O}((\log_2 n)^3)$. When $\varepsilon = 2^{-80}$, $\log_2 n = 1024$, $k = 160$, for example, the computational time for the client in the protocol $QR\text{-}CEKE$ is about 13.5 times less than that in the protocol $QR\text{-}EKE$, while the communication overhead in $QR\text{-}CEKE$ is just about 1 mini-second (1 ms) more than that in $QR\text{-}EKE$ when both protocols are running in a communication network of 1 megabits per second (1 mbps) bandwidth. Hence, the

protocol $QR\text{-}CEKE$ is more suitable to an imbalanced computing environment where a low-end client device communicates with a powerful server over a broadband network. Under the factoring assumption and the random oracle model, we also provide a formal security analysis of the $QR\text{-}CEKE$ protocol.

2 Security Model

Our formal model of security for password-authenticated key exchange protocols is based on that of [2].

Initialization. Let I denote the identities of the protocol participants. Elements of I will often be denoted A and B (Alice and Bob). Each pair of entities, $A, B \in I$, are assigned a password w which is randomly selected from the password space $\mathcal{D}$. The initialization process may also specify a set of cryptographic functions (e.g., hash functions) and establish a number of cryptographic parameters.

Running the Protocol. Mathematically, a protocol Π is a probabilistic polynomial-time algorithm which determines how entities behave in response to received message. For each entity, there may be multiple instances running the protocol in parallel. We denote the i-th instance of entity A as Π_A^i. The adversary $\mathcal{A}$ can make queries to any instance; she has an endless supply of Π_A^i oracles ($A \in I$ and $i \in \mathbb{N}$). In response to each query, an instance updates its internal state and gives its output to the adversary. At any point in time, the instance may accept and possesses a session key sk, a session id sid, and a partner id pid. The query types, as defined in [2], include:

- $\mathsf{Send}(A, i, M)$: This sends message M to instance Π_A^i. The instance executes as specified by the protocol and sends back its response to the adversary.
- $\mathsf{Execute}(A, i, B, j)$: This call carries out an honest execution between two instances Π_A^i and Π_B^j, where $A, B \in I, A \neq B$ and instances Π_A^i and Π_B^j were not used before. At the end of the execution, a transcript is given to the adversary, which logs everything an adversary could see during the execution (for details, see [2]).
- $\mathsf{Reveal}(A, i)$: The session key sk_A^i of Π_A^i is given to the adversary.
- $\mathsf{Test}(A, i)$: The instance Π_A^i generates a random bit b and outputs its session key sk_A^i to the adversary if $b = 1$, or else a random session key if $b = 0$. This query is allowed only once, at any time during the adversary's execution.
- $\mathsf{Oracle}(M)$: This gives the adversary oracle access to a function h, which is selected at random from some probability space Ω. The choice of Ω determines whether we are working in the standard model, or in the random-oracle model (see [2] for further explanations).

Let Π_A^i and Π_B^i, $A \neq B$, be a pair of instances. We say that Π_A^i and Π_B^i are *partnered* if both instances have accepted and hold the same session id sid and the same session key sk. Here, we define the sid of Π_A^i (or Π_B^i) as the concatenation of all the messages sent and received by Π_A^i (or Π_B^i). We say that

Π_A^i is *fresh* if: i) it has accepted; and ii) a Reveal query has not been called either on Π_A^i or on its partner (if there is one). With these notions, we now define the advantage of the adversary $\mathcal{A}$ in attacking the protocol. Let Succ denote the event that $\mathcal{A}$ asks a single Test query on a fresh instance, outputs a bit b', and $b' = b$, where b is the bit selected during the Test query. The advantage of the adversary $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{ake} = 2Pr(\mathsf{Succ}) - 1$.

As suggested in [5], we use the Send query type to count the number of on-line guesses performed by the adversary. We only count *one* Send query for each entity instance, that is, if the adversary sends two Send queries to an entity instance, it should still count as a single password guess. Based on this idea, we have the following definition of secure password-authenticated key exchange protocol, which is the same as in [5].

Definition 1. *A protocol Π is called a secure password-authenticated key exchange protocol if for every polynomial-time adversary $\mathcal{A}$ that makes at most Q_{send} ($Q_{send} \leq |\mathcal{D}|$) queries of* Send *type to different instances, the following two conditions are satisfied:*

(1) *Except with negligible probability, each oracle call* Execute(A, i, B, j) *produces a pair of partnered instances Π_A^i and Π_B^j.*
(2) $\mathsf{Adv}_{\mathcal{A}}^{ake} \leq Q_{send}/|\mathcal{D}| + \epsilon$*, where $|\mathcal{D}|$ denotes the size of the password space and ϵ is a negligible function of security parameters.*

3 Computationally-Efficient Password Authenticated Key Exchange Based on Quadratic Residues

Define hash functions $H_1, H_2, H_3 : \{0,1\}^* \rightarrow \{0,1\}^k$ and $H : \{0,1\}^* \rightarrow \mathbb{Z}_n$, where k is a security parameter, e.g., $k = 160$. The protocol $QR\text{-}CEKE$, which is described in Fig. 1, is based on $QR\text{-}EKE$, but the number of squaring operations performed by Bob is much less than $\lfloor \log_2 n \rfloor$. In the protocol $QR\text{-}CEKE$, Bob selects a small number ε, $0 < \varepsilon \leq 2^{-80}$, which determines the probability of a successful residue attack against the protocol $QR\text{-}CEKE$. Alice starts the protocol $QR\text{-}CEKE$ by sending a Blum integer n and two random numbers $\rho, r_A \in_R \{0,1\}^k$ to Bob. Bob verifies if n is an odd integer. If n is not odd, Bob rejects. Else, Bob computers $m = \lceil \log_2 \varepsilon^{-1} \rceil$. Then Bob selects a random number $\varrho \in_R \{0,1\}^k$ such that $\gamma = H(n, e, \rho, \varrho, A, B, m)$ satisfying $\gcd(\gamma, n) = 1$ and $\left(\frac{\gamma}{n}\right) = 1$. Bob sends ϱ and m to Alice. After receiving ϱ and m, Alice checks if $\gamma = H(n, e, \rho, \varrho, A, B, m)$ is a quadratic residue. If γ is not a quadratic residue, then $-\gamma$ must be a quadratic residue since n is a Blum integer. Next, Alice computes an integer u satisfying $u^{2^m} = \pm\gamma \mod n$, and sends u back to Bob. Subsequently, Bob verifies if Alice has made the right computation, i.e., $u^{2^m} = \pm\gamma \mod n$. If not, Bob rejects. Else, Alice and Bob executes the rest of the protocol as in $QR\text{-}EKE$.

Note that in the protocol $QR\text{-}CEKE$, Bob only verifies that the integer n received from Alice is an odd number; he does not verify that n is the product of two distinct primes p and q and $p \equiv q \equiv 3 \pmod 4$. This may foster the

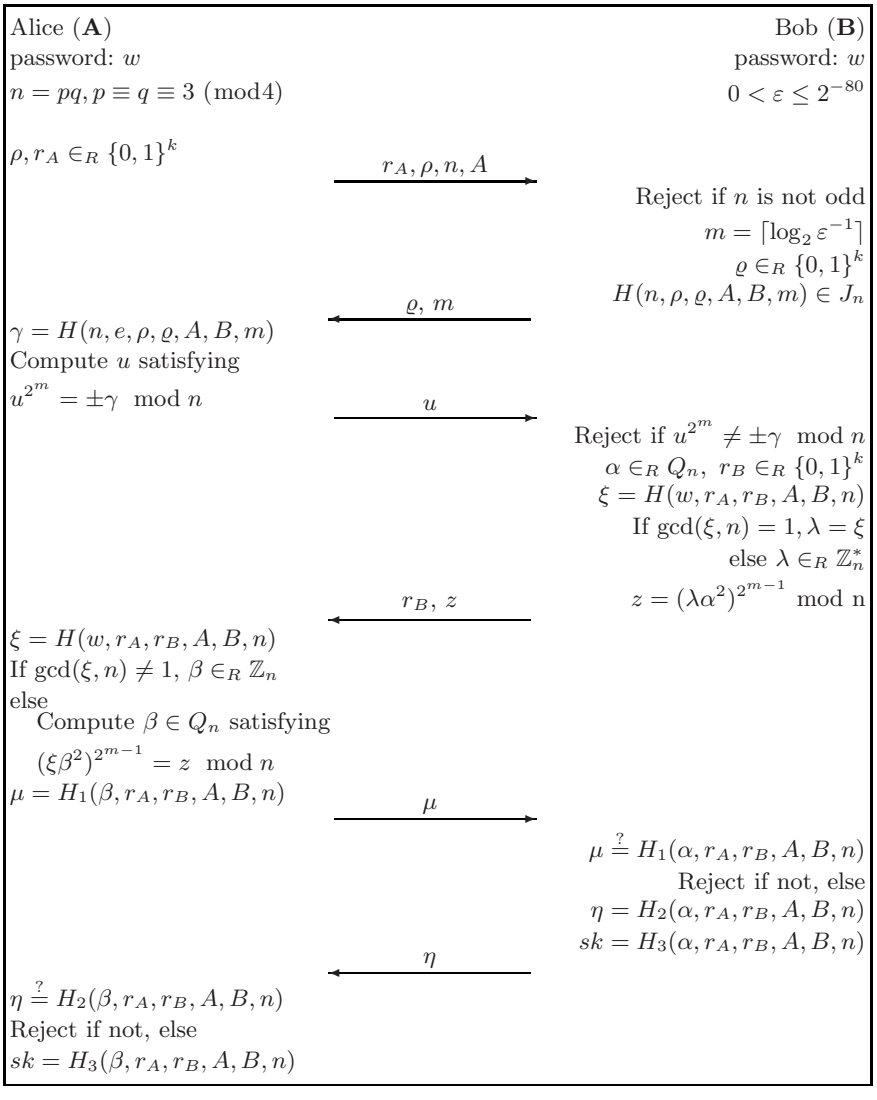

Fig. 1. The Protocol QR-CEKE

so-called residue attack as described in [3]. In such an attack, an adversary, say, *Eva*, selects a password π_0 at random from $\mathcal{D}$ and an odd integer n which may not necessarily be a Blum integer. Then Eva impersonates as Alice and starts the protocol by sending r_E, n, A to Bob. After receiving r_B and z from Bob, Eva Computes μ and sends it back to Bob. If Bob accepts, then Eva has a successful guess of Alice's password. If Bob rejects, on the other hand, Eva excludes her guess (i.e., π_0) from the password space $\mathcal{D}$. Furthermore, Eva may exclude more passwords by repeating, *off-line*, the following three steps:

1) Eva selects a password π from $\mathcal{D}$.
2) Eva computes $\gamma = H(\pi, r_E, r_B, A, B, n)$.
3) Eva tests if $\gcd(\gamma, n) = 1$. If not, Eva returns to step 1; otherwise, Bob verifies if the congruence $(\gamma x^2)^{2^t} \equiv z \pmod{n}$ has a solution in Q_n. If the congruence has a solution, Eva returns to step 1. If the congruence has no solution in Q_n, then Eva is ensured that π is not the password of Alice. Next Eva excludes π from $\mathcal{D}$ and returns to step 1.

Theorem 1. *Let n, $n > 1$, be an odd integer with prime-power factorization $n = p_1^{a_1} p_2^{a_2} \dots p_r^{a_r}$. Let m be a positive integer. If there exists a prime power, say $p_i^{a_i}$, of the factorization of n such that $2^m \mid \phi(p_i^{a_i})$, then for an integer γ randomly selected from J_n, the probability that γ is an 2^m-th power residue of n is less than or equal to 2^{-m+1}.*

Proof. Let $n_i = p_i^{a_i}$ be a prime power of the factorization of n such that $2^m \mid \phi(n_i)$. Since n is odd, n_i possesses a primitive root. Let g be a primitive root of n_i. For an integer γ randomly selected from J_n, let $\mathrm{ind}_g\gamma$ denote the index of γ to the base g modulo n_i. Then γ is an 2^m-th power residue of n_i if and only if the congruence $x^{2^m} \equiv \gamma \pmod{n_i}$ has a solution, or equivalently, if and only if

$$g^{2^m \mathrm{ind}_g x - \mathrm{ind}_g \gamma} \equiv 1 \pmod{n_i},$$

which is equivalent to

$$2^m \mathrm{ind}_g x \equiv \mathrm{ind}_g \gamma \pmod{\phi(n_i)}.$$

Since $2^m \mid \phi(n_i)$, γ is an 2^m-th power residue of n_i if and only if $2^m \mid \mathrm{ind}_g\gamma$.

Let $n_i' = n/n_i$, then n_i and n_i' are relatively prime. For any integer $\beta \in \mathbb{Z}_n^*$, it is clear that $\beta \mod n_i$ and $\beta \mod n_i'$ are integers of $\mathbb{Z}_{n_i}^*$ and $\mathbb{Z}_{n_i'}^*$, respectively. On the other hand, for two integers $\alpha_1 \in \mathbb{Z}_{n_i}^*$ and $\alpha_2 \in \mathbb{Z}_{n_i'}^*$, by the Chinese Remainder Theorem, there is an unique integer $\alpha \in \mathbb{Z}_n^*$, such that $\alpha \equiv \alpha_1 \pmod{n_i}$, and $\alpha \equiv \alpha_2 \pmod{n_i'}$. So, the number of integers $\alpha \in \mathbb{Z}_n^*$ which satisfy the congruence $\alpha \equiv \alpha_1 \pmod{n_i}$ is $\phi(n_i')$. If γ is randomly selected from J_n, then for any integer s, $0 \le s \le \phi(n_i) - 1$, we have

$$Pr(g^s = \gamma \mod n_i) \le \frac{\phi(n_i')}{|J_n|} \le \frac{2}{\phi(n_i)}.$$

Note that in last inequality described above, we make use of the fact $|J_n| \ge \phi(n)/2$. Thus, we have $Pr(\mathrm{ind}_g\gamma = s) \le 2/\phi(n_i)$. Therefore,

$$\begin{aligned} Pr(2^m \mid \mathrm{ind}_g\gamma) &= \sum_{2^m \mid s,\, 0 \le s < \phi(n_i)} Pr(\mathrm{ind}_g\gamma = s) \\ &\le 2\phi(n_i) 2^{-m}/\phi(n_i) \\ &= 2^{-m+1} \end{aligned}$$

which indicates that, for an integer γ randomly selected from J_n, the probability that γ is an 2^m-th power residue of n_i is less than or equal to 2^{-m+1}. So, the probability that γ is an 2^m-th power residue of n does not exceed 2^{-m+1}. □

Theorem 1 demonstrates that, if there exits a prime-power $p_i^{a_i}$ of the factorization of n such that $2^m \mid \phi(p_i^{a_i})$, then for a random number $\gamma \in J_n$, the probability that Alice can take square roots of γ or $-\gamma$ repetitively m times is less than or equal to 2^{-m+1}.

Theorem 2. *Let n, $n > 1$, be an odd integer with prime-power factorization $n = p_1^{a_1} p_2^{a_2} \dots p_r^{a_r}$. Let m be a positive integer such that for any prime-power $p_i^{a_i}$ of the factorization of n, $2^{m+1} \nmid \phi(p_i^{a_i}), 1 \le i \le r$. If z is an 2^m-th power residue of n, then for any $\lambda \in \mathbb{Z}_n^*$, the congruence $(\lambda x^2)^{2^m} \equiv z \pmod{n}$ has a solution in Q_n.*

Proof. To prove that $(\lambda x^2)^{2^m} \equiv z \pmod{n}$ has a solution in Q_n, we only need to prove that, for each prime power $p_i^{a_i}$ of the factorization of n, the following congruence

$$(\lambda x^2)^{2^m} \equiv z \pmod{p_i^{a_i}} \tag{1}$$

has a solution in $Q_{p_i^{a_i}}$.

Let $n_i = p_i^{a_i}, 1 \le i \le r$. Then $\phi(n_i) = p_i^{a_i - 1}(p_i - 1)$. Since n is odd, p_i is an odd prime. Thus, the integer n_i possesses a primitive root. Let g be a primitive root of n_i, that is, $g^{\phi(n_i)} = 1 \bmod n_i$, and for any $0 \le i, j \le \phi(n_i) - 1, i \ne j$, $g^i \ne g^j \bmod n_i$. Let $\gcd(2^m, \phi(n_i)) = 2^c, 1 \le c \le m$. We consider the following two cases:

(1) If $c = 1$, then $d = \phi(n_i)/2$ must be an odd integer. For any integer $a \in \mathbb{Z}_{n_i}^*$, $a^{2d} \equiv 1 \pmod{n_i}$, which implies that $a^d \equiv 1$ or $-1 \pmod{n_i}$. We claim that $a^d \equiv -1 \pmod{n_i}$ if and only if a is a quadratic non-residue of n_i. If $a^d \equiv -1 \pmod{n_i}$, it is obvious that $a \in \bar{Q}_{n_i}$. On the other hand, if $a \in \bar{Q}_{n_i}$, then there exists an odd integer s such that $a = g^s \bmod n_i$ since g is the primitive root of n_i. As the order of g is $2d$, not d, we have $a^d = (g^d)^s = -1 \bmod n_i$. Similarly, we can also prove that, if $a \in Q_{n_i}$, then the congruence $x^2 \equiv a \pmod{n_i}$ has two solutions, with one solution in Q_{n_i} and another in $\bar{Q}_{n_i}$. Hence, for any $\gamma \in \mathbb{Z}_n^*$, there exists a solution x_j, $0 \le j \le 1$, such that $x_j \gamma \in Q_{n_i}$, that is, $(x_j \gamma)^d = 1 \pmod{n_i}$. Hence, congruence (3) has a solution in Q_{n_i}.

(2) Next, we consider the case that $2 \le c \le m$. Since z is a 2^m-th power residue modulo n, the congruence $x^{2^m} \equiv z \pmod{n}$ has solutions in $\mathbb{Z}_n^*$. By the Chinese Remainder Theorem, the following congruence

$$y^{2^m} \equiv z \pmod{n_i} \tag{2}$$

has solutions in $\mathbb{Z}_{n_i}^*$. Let $\mathrm{ind}_g z$ denote the index of z to the base g modulo n_i and let $y \in \mathbb{Z}_{n_i}^*$ be a solution of (4). Then, $g^{2^m \mathrm{ind}_g y - \mathrm{ind}_g z} \equiv 1 \pmod{n_i}$. Since the order of g modulo n_i is $\phi(n_i)$, it follows that

$$2^m \mathrm{ind}_g y \equiv \mathrm{ind}_g z \pmod{\phi(n_i)} \tag{3}$$

Also since $\gcd(2^m, \phi(n_i)) = 2^c$, equation (5) has exactly 2^c incongruent solutions modulo $\phi(n_i)$ when taking $\mathrm{ind}_g y$ as variable. This indicates that equation (4)

has exactly 2^c incongruent solutions modulo n_i. Let y_0 be one of the solutions of equation (4), then the 2^c incongruent solutions of (5) are given by

$$\mathrm{ind}_g y = \mathrm{ind}_g y_0 + j\phi(n_i)/2^c \mod \phi(n_i), \quad 0 \le j \le 2^c - 1.$$

For any $\gamma \in \mathbb{Z}_n^*$, we have

$$\mathrm{ind}_g y - \mathrm{ind}_g \gamma = \mathrm{ind}_g y_0 - \mathrm{ind}_g \gamma + j\phi(n_i)/2^c \mod \phi(n_i), \quad 0 \le j \le 2^c - 1.$$

Without loss of generality, let's assume that $\mathrm{ind}_g y_0 - \mathrm{ind}_g \gamma \ge 0$; otherwise we consider $\mathrm{ind}_g \gamma - \mathrm{ind}_g y$. Under the condition that $2^{m+1} \nmid \phi(n_i)$ it is clear that $\phi(n_i)/2^c$ is an odd integer. Hence, there exist an integer $j, 0 \le j \le 3 \le 2^c - 1$, such that

$$\mathrm{ind}_g y_0 - \mathrm{ind}_g \gamma + j\phi(n_i)/2^c \equiv 0 \pmod 4,$$

which implies that there exists an integer $y \in \mathbb{Z}_{n_i}^*$ such that $y^{2^m} \equiv z \pmod{n_i}$ and $y\gamma^{-1}$ is a 4-th power residue of n_i. Therefore, the congruence (3) has a solution in Q_{n_i}, which proves the theorem. □

Based on Theorem 1 and Theorem 2, we can conclude that the probability for an adversary to launch a successful residue attack against the protocol *QR-CEKE* is less than or equal to 2ε.

In the protocol *QR-CEKE*, Alice proves to Bob in an interactive manner (via flow 2 and flow 3) that for every prime-power $p_i^{a_i}$ of the factorization of n, $2^m \nmid \phi(p_i^{a_i})$. The interactive procedure increases the communication overhead on Alice and Bob by $\log_2 n + 2k$ bits. When $\log_2 n = 1024$ and $k = 160$, for example, the communication overhead induced by the interactive procedure is about 1 mini-second (1 ms) over a broadband network of 1 megabits per second (1 mbps) bandwidth. In *QR-CEKE*, the computational burden on Bob includes two modulo exponentiations, i.e., $u^{2^m} \mod n$ and $(\lambda a^2)^{2^{m-1}} \mod n$, where $m = \lceil \log_2 \varepsilon^{-1} \rceil$. The computation time for the two modulo exponentiations is $\mathcal{O}((\log_2 \varepsilon^{-1})(\log_2 n)^2)$. When $\varepsilon^{-1} \ll n$, the computational load on Bob is greatly reduced in *QR-CEKE* in comparison with that in *QR-EKE* (or in SNAPI).

4 Formal Security Analysis

In this section, we analyze the security of *QR-CEKE* within the formal model of security given in Section 2. Our analysis is based on the random-oracle model. In this model, a hash function is modeled as an oracle which returns a random number for each new query. If the same query is asked twice, identical answers are returned by the oracle. In our analysis, we also assume the intractability of the Factoring problem.

Factoring Assumption: Let *GE* be a probabilistic polynomial-time algorithm that on input 1^ℓ returns a product of two distinct primes p and q of length $\ell/2$

satisfying $p \equiv \ q \equiv 3 \pmod 4$. For any probabilistic polynomial-time algorithm $\mathcal{C}$ of running time t, the following probability

$$\mathsf{Adv}_{\mathcal{C}}^{fac}(t) = Pr(\mathcal{C}(n) = (p,q), pq = n : n \leftarrow GE(1^{\ell}))$$

is negligible. In the following, we use $\mathsf{Adv}^{fac}(t)$ to denote $\max_{\mathcal{C}}\{\mathsf{Adv}_{\mathcal{C}}^{fac}(t)\}$, where the maximum is taken over all polynomial-time algorithms of running time t.

Under the above assumptions, we have the following Theorem 4.

Theorem 3. *Let $\mathcal{A}$ be an adversary which runs in time t and makes Q_{send}, $Q_{send} \leq |\mathcal{D}|$, queries of type* Send *to different instances. Then the adversary's advantage in attacking the protocol $QR\text{-}CEKE$ is bounded by*

$$\mathsf{Adv}_{\mathcal{A}}^{ake} \leq \frac{Q_{send}}{|\mathcal{D}|} + 4\varepsilon + (Q_{execute} + 5Q_{send})\mathsf{Adv}^{fac}(\mathcal{O}(t)) + + \frac{Q_{send}}{2^{k-1}} + \frac{(Q_{execute} + 2Q_{send})Q_{oh}}{\phi(n)},$$

where $Q_{execute}$ denotes the number of queries of type Execute *and Q_{oh} denotes the number of random oracle calls.*

5 Conclusion

In this paper, we present a computationally efficient password authenticated key exchange protocol based on quadratic residues. The protocol $QR\text{-}CEKE$ is derived from the protocol $QR\text{-}EKE$, a previously published password authenticated key exchange protocol based on quadratic residues. However, the computational time for the client is significant reduced in the protocol $QR\text{-}CEKE$. In comparison with $QR\text{-}EKE$, the protocol $QR\text{-}CEKE$ is more suitable to an imbalanced computing environment where a low-end client device communicates with a powerful server over a broadband network. Based on number-theoretic techniques, we show that the computationally efficient password authenticated key exchange protocol is secure against *residue attacks*, a special type of off-line dictionary attack against password-authenticated key exchange protocols based on RSA and quadratic residues. We also provide a formal security analysis of $QR\text{-}CEKE$ under the factoring assumption and the random oracle model.

References

1. Bao, F.: Security analysis of a password authenticated key exchange protocol. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, pp. 208–217. Springer, Heidelberg (2003)
2. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attack. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)

3. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Bellovin, S.M., Merritt, M. (eds.) Proc. of the IEEE Symposium on Research in Security and Privacy, Oakland, pp. 72–84 (May 1992)
4. Catalano, D., Pointcheval, D., Pornin, T.: IPAKE: Isomorphisms for Password-based Authenticated Key Exchange. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, Springer, Heidelberg (to appear, 2004)
5. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: Biham, E. (ed.) EUROCRPYT 2003. LNCS, vol. 2656, pp. 524–542. Springer, Heidelberg (2003)
6. Jablon, D.: `http://www.integritysciences.com`
7. Lucks, S.: Open key exchange: How to defeat dictionary attacks without encrypting public keys. In: Christianson, B., Lomas, M. (eds.) Proc. Security Protocol Workshop. LNCS, vol. 1361, pp. 79–90. Springer, Heidelberg (1997)
8. MacKenzie, P., Patel, S., Swaminathan, R.: Password-authenticated key exchange based on RSA. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000)
9. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
10. Patel, S.: Number theoretic attacks on secure password schemes. In: IEEE Symposium on Security and Privacy, Oakland, California (May 5-7, 1997)
11. Zhu, F., Wong, D., Chan, A., Ye, R.: RSA-based password authenticated key exchange for imbalanced wireless networks. In: Chan, A.H., Gligor, V.D. (eds.) ISC 2002. LNCS, vol. 2433, pp. 150–161. Springer, Heidelberg (2002)
12. Zhang, M.: New approaches to password authenticated key exchange based on RSA. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 230–244. Springer, Heidelberg (2004)
13. Zhang, M.: Password Authenticated Key exchange using quadratic residues. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 248–262. Springer, Heidelberg (2004)
14. Zhang, M.: Further analysis of password authenticated key exchange protocol based on RSA for imbalanced wireless networks. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 12–24. Springer, Heidelberg (2004)