J. Cryptology (1996) 9: 101–109

Journal of
CRYPTOLOGY


# A Low Communication Competitive Interactive Proof System for Promised Quadratic Residuosity

Toshiya Itoh and Masafumi Hoshi
Department of Information Processing,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku,
Yokohama 226, Japan

Shigeo Tsujii
Department of Computer Science, Faculty of Science and Engineering,
Chuo University, 1-13-27 Kasuga, Bunkyo-ku,
Tokyo 112, Japan

Communicated by Joan Feigenbaum

Received 29 October 1993 and revised 25 January 1995

**Abstract.** Bellare and Goldwasser showed that if the modulus $N$ is guaranteed to be the product of $O(\log\log|N|)$ distinct odd primes, then quadratic residuosity has a competitive interactive proof system (with reasonably large communication complexity). In this paper we show that if the modulus $N$ is guaranteed to be the product of $O(\log|N|)$ distinct odd primes, then quadratic residuosity has a competitive interactive proof system with low communication complexity.

**Key words.** Competitive interactive proof systems, Communication complexity, Quadratic residuosity, Promise problems, Residue classes.

## 1. Introduction

### 1.1. *Background and Motivation*

Is proving membership harder than deciding membership? This is a basic question in theoretical computer science. It is known that if a language $L$ is $\mathcal{NP}$-complete, then computing a witness $w$ for $x \in L$ is polynomially equivalent to deciding $x \in L$. How about the languages that are not known to be $\mathcal{NP}$-complete? In general, it is believed that this is not the case. Recently, Bellare and Goldwasser [BG] showed that a language $L \in \mathcal{NP} - \mathcal{P}$ exists for which computing a witness $w$ for $x \in L$ is harder than deciding $x \in L$ if deterministic double exponential time is not equal to nondeterministic double exponential time. The language found in [BG] is "uniformly log-sparse" and thus it is somewhat unnatural. On the other hand, several natural languages $L \in \mathcal{NP}$ exist for

which computing a witness $w$ for $x \in L$ may be harder than deciding $x \in L$, e.g., quadratic residuosity (QR), quadratic nonresiduosity (QNR), etc.

What will happen when interaction and randomization are allowed in the process of proving membership? This way of proving membership is formalized by Goldwasser *et al.* [GMR] in the notion of an interactive proof system and independently by Babai [B], who called it an Arthur–Merlin game. Then the following problem comes to mind: Does a prover $P$ in an interactive proof system $\langle P, V \rangle$ for a language $L$ need more power than that needed to decide membership in $L$? This has been studied in several contexts, e.g., (zero-knowledge) arguments [BCC], the complexity of (multiprover) interactive proof systems [LFKN], [S], [BFL], and program checking [BK], [BF]. Bellare and Goldwasser [BG] regarded the problem as the (natural) extension of whether computing a witness $w$ for $x \in L$ is harder than deciding $x \in L$ for a language $L \in \mathcal{NP}$ and defined "competitive" interactive proof systems, i.e., an interactive proof in which the honest prover must be computable by a probabilistic polynomial-time Turing machine with oracle access to $L$.

When considering interactive proofs, asking if proving membership is harder than deciding membership is equivalent to asking if the existence of an interactive proof of membership for a language implies the existence of a competitive interactive proof system for that language. In some cases interaction and randomization alleviate the proving task, but in other cases they may not. To see this more precisely, we first consider quadratic nonresiduosity $\mathrm{QNR} = \{\langle x, N \rangle \mid x \in Z_N^*$ is not a square modulo $N\}$. It is not known whether or not computing a witness $w$ for $x \in \mathrm{QNR}$ is polynomially equivalent to deciding $x \in \mathrm{QNR}$. Indeed it is believed that computing a witness $w$ for $x \in \mathrm{QNR}$ may be harder than deciding $x \in \mathrm{QNR}$. However, the interactive proof system for QNR [GMR] is competitive, i.e., the (honest) prover $P$ needs only the computational ability of deciding $x \in \mathrm{QNR}$ in order to prove membership of $x \in \mathrm{QNR}$ in an interactive and a randomized manner. Next we consider quadratic residuosity $\mathrm{QR} = \{\langle x, N \rangle \mid x \in Z_N^*$ is a square modulo $N\}$. It is also believed that computing a witness $w$ for $x \in \mathrm{QR}$ may be harder than deciding $x \in \mathrm{QR}$. Contrary to QNR, in all known interactive proof systems $\langle P, V \rangle$ for QR (see, e.g., [GMR], [FS], [TW], and [FFS]), the (honest) prover $P$ needs at least the computational ability to compute square roots modulo a composite number $N$ (equivalently the computational ability to factor a composite number $N$).

Beigel and Feigenbaum [BF] were the first to obtain a negative result about the power of the prover in an interactive proof system for an $\mathcal{NP}$ language. Specifically, they showed that an *incoherent* language $L \in \mathcal{NP}$ exists if nondeterministic triple exponential time is not included in bounded probabilistic triple exponential time. This implies that interaction and randomization do not necessarily alleviate the proving task, because an incoherent language does not possess a competitive interactive proof system (see [Y], [BK], and [BG]). Subsequently, Bellare and Goldwasser [BG] showed that a coherent language $L \in \mathcal{NP}$ exists that does not have a competitive interactive proof system if nondeterministic double exponential time is not included in bounded probabilistic double exponential time. These results only guarantee the existence of a language that does not have a competitive interactive proof system under the given complexity assumptions, but do not imply that QR does not have a competitive interactive proof system. Thus is it possible to construct a competitive interactive proof system for QR?

This has not been solved yet. As a start toward solving this open problem affirmatively,

Bellare and Goldwasser [BG] investigated QR in a *promised* form. In this setting, Bellare and Goldwasser introduced a notion of *representative* of $Z_N^*$ [BG, Definition 6.4] and showed that a competitive interactive proof system for promised QR exists, i.e., the QR problem when the modulus $N$ is guaranteed to be the product of $k = O(\log\log|N|)$ distinct odd primes. The basic idea behind the result above is to use the fact that there are $2^k = O(\log|N|)$ distinct residue classes under a relation appropriately defined on $Z_N^*$ and to reduce a quadratic residuosity test to a collection of quadratic nonresiduosity tests. Then the protocol following this idea requires about $2^{2k}$ quadratic nonresiduosity tests and thus the communication complexity of the resulting protocol is comparatively large (but still polynomial)—in the protocol, the prover $P$ sends the verifier $V$ about $2^k(|N| + 2^k)$ bits and the verifier $V$ sends the prover $P$ about $2^{2k}|N|$ bits.

### 1.2. *Results*

In this paper we consider how to reduce the communication complexity of a competitive interactive proof system for promised QR and how to relax the constraint on $k$ from $O(\log\log|N|)$ to $O(\log|N|)$. For this purpose, we first introduce the notion of *dominant* of $Z_N^*$, which plays a role very similar to a basis in a linear space over $GF(2)$. Then we investigate several properties of a dominant vector of $Z_N^*$ and show that promised QR with the constraint that $k = O(\log|N|)$ has a competitive interactive proof system in which the prover $P$ sends the verifier $V$ about $k|N|$ bits and the verifier $V$ sends the prover $P$ about $4|N|$ bits. The basic idea behind the result here is to use the fact that if the modulus $N$ is guaranteed to be the product of $k = O(\log|N|)$ distinct odd primes, then there are sufficiently many (samplable) vectors $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ over $Z_N^*$ to specify $2^k$ residue classes uniquely under a relation appropriately defined on $Z_N^*$. The idea here is inspired by the one due to Bellare and Goldwasser [BG], but its use enables us to avoid $2^{2k}$ invocations of quadratic nonresiduosity tests. Thus the resulting protocol based on this idea considerably reduces the communication complexity.

## 2. Preliminaries

See [GMR] for definitions of interactive protocols and interactive proof systems. The definition of interactive proof systems in this paper is slightly different from that of [GMR], but it is known that both are equivalent.

**Definition 2.1** [BG]. An interactive proof system $\langle P, V\rangle$ for a language $L$ is said to be *competitive* if the honest prover $P$ is a probabilistic polynomial-time oracle Turing machine such that, for any $x \in L$, $\text{Prob}\{\langle P^L, V\rangle \text{ accepts } x\} \geq 2/3$.

It is known that competitive interactive proof systems exist for the languages "quadratic nonresiduosity [GMR]," "graph nonisomorphism [GMW]," and "graph isomorphism," however, the language "quadratic residuosity" is believed not to have a competitive interactive proof system.

The following definition of a promise problem is equivalent to the ones in [ESY] and [GS].

**Definition 2.2** [BG]. A *promise problem* is a pair of disjoint sets $\langle A, B\rangle$. A *promise oracle* for a promise problem $\langle A, B\rangle$ is an oracle that given a query $q \in A \cup B$, returns 1 if $q \in A$ and returns 0 if $q \in B$.

**Definition 2.3** [BG]. An interactive protocol $\langle P, V\rangle$ is said to be a *competitive* interactive proof system for the *promise problem* $\langle A, B\rangle$ if

- *completeness*: for any $x \in A$ and any promise oracle $O$ for $\langle A, B\rangle$,

$$\text{Prob}\{\langle P^O, V\rangle \text{ accepts } x\} \geq 2/3,$$

  where $P$ is a probabilistic polynomial-time oracle Turing machine;
- *soundness*: for any $x \in B$ and any powerful dishonest prover $P^*$,

$$\text{Prob}\{\langle P^*, V\rangle \text{ accepts } x\} \leq 1/3,$$

where the probabilities are taken over all of the possible coin tosses of both $P$ and $V$.

The problem that we are interested in is when the modulus $N$ is guaranteed to be the product of $k \geq 1$ distinct odd primes.

**Definition 2.4** [BG]. *Promised* $\mathsf{QR}_k$ is the pair of disjoint sets $\langle \text{QR}_k, \text{QNR}_k\rangle$, where $\text{QR}_k = \{\langle x, N\rangle \in \text{QR} \mid N \text{ is the product of } k \text{ distinct odd primes}\}$, $\text{QNR}_k = \{\langle x, N\rangle \in \text{QNR} \mid N \text{ is the product of } k \text{ distinct odd primes}\}$, and $k \geq 1$.

## 3. Main Results

We begin by showing several technical lemmas.

### 3.1. *Technical Lemmas*

Let $N = p_1 p_2 \cdots p_k$, where $p_1, p_2, \ldots, p_k$ are distinct odd primes. For any $x \in Z_N^*$, let $Q_N(x) = 0$ if $x$ is a square modulo $N$ and let $Q_N(x) = 1$ otherwise. For any $x, y \in Z_N^*$, define a binary relation $\simeq$ on $Z_N^*$ by $x \simeq y$ if and only if $Q_{p_i}(x) = Q_{p_i}(y)$ for each $i$ $(1 \leq i \leq k)$. It is easy to see that the relation $\simeq$ on $Z_N^*$ is an equivalence relation on $Z_N^*$. The equivalence class $R_N(x)$ of $x \in Z_N^*$ under the relation $\simeq$ is

$$R_N(x) = \{y \in Z_N^* \mid x \simeq y\}$$

and is called the residue class of $x \in Z_N^*$. Note that $Z_N^*$ can be partitioned into $2^k$ (disjoint) residue classes, each of which is of the same size. It is well known that $Q_{p_i}(xy) \equiv Q_{p_i}(x) + Q_{p_i}(y) \pmod 2$. It is also well known that $xy$ is a square modulo $N$ if and only if $x \simeq y$.

**Definition 3.1.** Let $N$ be the product of $k$ distinct odd primes, where $N = p_1 p_2 \cdots p_k$ and $p_1 < p_2 < \cdots < p_k$. The vector $\mathbf{C}_N(x) = (Q_{p_1}(x), Q_{p_2}(x), \ldots, Q_{p_k}(x))$ is called *the vector associated with* $x \in Z_N^*$.

It is obvious that, for any $x \in Z_N^*$, $x \in \mathrm{QR}_k$ if and only if $\mathbf{C}_N(x) = \mathbf{0}$. Let $\mathbf{v} + \mathbf{w}$ be componentwise addition modulo 2 and, for an integer $e \geq 0$ and $\mathbf{v} = (v_1, v_2, \ldots, v_k)$, define $e\mathbf{v} = (ev_1, ev_2, \ldots, ev_k)$, where $ev_i$ is the modulo 2 sum of $e$ copies of $v_i$ (i.e., $v_i$ if $e$ is odd and 0 if $e$ is even). The following lemmas show basic properties of the vector $\mathbf{C}_N(x)$ associated with $x \in Z_N^*$.

**Lemma 3.2.** *Let $N$ be the product of $k$ distinct odd primes. Then $\mathbf{C}_N(xy) \equiv \mathbf{C}_N(x) + \mathbf{C}_N(y) \pmod 2$ for all $x, y \in Z_N^*$.*

**Proof.** Assume that $N = p_1 p_2 \cdots p_k$, where $p_1, p_2, \ldots, p_k$ are distinct odd primes. The proof follows from the fact that $Q_{p_i}(xy) \equiv Q_{p_i}(x) + Q_{p_i}(y) \pmod 2$ $(1 \leq i \leq k)$. □

**Lemma 3.3.** *Let $N$ be the product of $k$ distinct odd primes. Then $\mathbf{C}_N(x^e) \equiv e\mathbf{C}_N(x) \pmod 2$ for any integer $e \geq 0$.*

**Proof.** Follows from Lemma 3.2. □

The following notion of "dominant" is one of the most important ones in our main result. It corresponds to the notion of representative [BG, Definition 6.4] and plays a role similar to a basis in a linear space over $GF(2)$.

**Definition 3.4.** Let $N$ be the product of $k$ distinct odd primes. A vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ over $Z_N^*$ is said to be *dominant* of $Z_N^*$ if $\mathbf{C}_N(y_1), \mathbf{C}_N(y_2), \ldots, \mathbf{C}_N(y_k) \in \{0, 1\}^k$ are linearly independent over $GF(2)$.

Let $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ be a vector over $Z_N^*$ and let $\mathbf{e} = (e_1, e_2, \ldots, e_k)$ be a vector over $GF(2)$. For simplicity here, we use $\mathbf{y} \uparrow \mathbf{e}$ to denote $y_1^{e_1} y_2^{e_2} \cdots y_k^{e_k} \pmod N$.

**Lemma 3.5.** *Let $N$ be the product of $k$ distinct odd primes. A vector $\mathbf{y}$ is dominant of $Z_N^*$ if and only if $\mathbf{y} \uparrow \mathbf{e}$ is not a square modulo $N$ for every nonzero vector $\mathbf{e} \in \{0, 1\}^k$.*

**Proof.** ($\Leftarrow$) We show this by contradiction. We assume that the vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ is not dominant of $Z_N^*$. Then a nonzero vector $\mathbf{e} = (e_1, e_2, \ldots, e_k) \in \{0, 1\}^k$ exists such that $e_1\mathbf{C}_N(y_1) + e_2\mathbf{C}_N(y_2) + \cdots + e_k\mathbf{C}_N(y_k) \equiv \mathbf{0} \pmod 2$ and this implies that $\mathbf{y} \uparrow \mathbf{e}$ is a square modulo $N$.

($\Rightarrow$) Assume that $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ is dominant of $Z_N^*$, i.e., $C_N(y_1), C_N(y_2), \ldots, C_N(y_k)$ are linearly independent over $GF(2)$. Then, for any $\mathbf{e} = (e_1, e_2, \ldots, e_k) \in \{0, 1\}^k$, $e_1\mathbf{C}_N(y_1) + e_2\mathbf{C}_N(y_2) + \cdots + e_k\mathbf{C}_N(y_k) \equiv \mathbf{0} \pmod 2$ if and only if $\mathbf{e} = \mathbf{0}$. Thus $\mathbf{y} \uparrow \mathbf{e}$ is not a square modulo $N$ for every nonzero vector $\mathbf{e} \in \{0, 1\}^k$. □

In the following lemma we show that if $k = O(\log |N|)$, then a dominant vector $\mathbf{y}$ of $Z_N^*$ can be efficiently sampled by a probabilistic polynomial-time oracle Turing machine with access to the promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$.

**Lemma 3.6.** *If $k = O(\log |N|)$, then there is a probabilistic polynomial-time oracle Turing machine $D$ with access to the promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ that on input $\langle x, N \rangle \in \mathrm{QR}_k \cup \mathrm{QNR}_k$ outputs either a dominant vector $\mathbf{y}$ of $Z_N^*$ with probability at least $3/4$ or $\perp$ with probability at most $1/4$.*

**Proof.** From the assumption that $k = O(\log |N|)$, it follows that $2^k \leq |N|^c$ for some integer $c > 0$. The machine $D$ randomly chooses $h = 2|N|^c$ vectors $\mathbf{y}_j = (y_1^j, y_2^j, \ldots, y_k^j)$ over $Z_N^*$ $(1 \leq j \leq h)$. For each $\mathbf{y}_j$ $(1 \leq j \leq h)$ and each $\mathbf{e} \in \{0,1\}^k - \{0^k\}$, the machine $D$ computes $q_j^{\mathbf{e}} \equiv \mathbf{y}_j \mid \mathbf{e} \pmod N$, and queries the promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ with $q_j^{\mathbf{e}}$ to get the answer $a_j^{\mathbf{e}} \in \{0,1\}$. If an index $j$ $(1 \leq j \leq h)$ exists such that $a_j^{\mathbf{e}} = 0$ for every $\mathbf{e} \in \{0,1\}^k - \{0^k\}$, then $D$ outputs $\mathbf{y} = \mathbf{y}_j$ as a *dominant* vector of $Z_N^*$; otherwise $D$ outputs $\perp$. It is obvious from Lemma 3.5 that if a vector $\mathbf{y}$ is sampled by $D$, then it is always dominant of $Z_N^*$.

Let $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ be a randomly chosen vector over $Z_N^*$ and let $\mathbf{y}_m = (y_1, y_2, \ldots, y_m)$ for each $m$ $(1 \leq m \leq k-1)$. Then $\mathbf{y}$ is dominant of $Z_N^*$ if and only if $y_1 \in \mathrm{QNR}_k$ and, for each $m$ $(1 \leq m \leq k-1)$, $y_{m+1} \not\simeq \mathbf{y}_m \uparrow \mathbf{e}_m$ for every $\mathbf{e}_m \in \{0,1\}^m$. Recall that $Z_N^*$ can be partitioned into $2^k$ (disjoint) residue classes of the same size. Then the probability $P_{\mathrm{dom}}$ that a randomly chosen vector $\mathbf{y}$ over $Z_N^*$ is dominant of $Z_N^*$ is given by

$$P_{\mathrm{dom}} = \frac{1}{\|Z_N^*\|^k} \prod_{j=0}^{k-1} \left( \|Z_N^*\| - 2^j \cdot \frac{\|Z_N^*\|}{2^k} \right) = \prod_{i=1}^{k} \left(1 - 2^{-i}\right) > \left(\tfrac{1}{2}\right)^k \geq \frac{1}{|N|^c},$$

where $\|A\|$ denotes the cardinality of a (finite) set $A$. Thus the probability $P_{\mathrm{samp}}$ that the machine $D$ samples a dominant vector $\mathbf{y}$ of $Z_N^*$ is bounded by

$$P_{\mathrm{samp}} = 1 - (1 - P_{\mathrm{dom}})^h > 1 - \left(1 - \frac{1}{|N|^c}\right)^{2|N|^c} \geq 1 - e^{-2} > \tfrac{3}{4},$$

because $(1 - x^{-1})^x \leq e^{-1}$ for any $x \geq 1$. Since $D$ queries the promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ at most $h2^k = 2|N|^{2c}$ times, it runs in probabilistic polynomial (in $|N|$) time. □

The lemma below is used to reduce the communication complexity of a competitive interactive proof system for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$.

**Lemma 3.7.** *Let $N$ be the product of $k$ distinct odd primes and let $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ be dominant of $Z_N^*$. Then, for any $x \in Z_N^*$, there is a* unique *vector $\mathbf{e} = (e_1, e_2, \ldots, e_k)$ over $GF(2)$ such that $x \simeq \mathbf{y} \uparrow \mathbf{e}$.*

**Proof.** Since $\mathbf{y}$ is dominant of $Z_N^*$, $\mathbf{C}_N(y_1), \mathbf{C}_N(y_2), \ldots, \mathbf{C}_N(y_k)$ are linearly independent over $GF(2)$. This implies that, for any $x \in Z_N^*$, there is a *unique* vector $\mathbf{e} = (e_1, e_2, \ldots, e_k)$ over $GF(2)$ such that $\mathbf{C}_N(x) \equiv e_1\mathbf{C}_N(y_1) + e_2\mathbf{C}_N(y_2) + \cdots + e_k\mathbf{C}_N(y_k) \pmod 2$. By Lemmas 3.2 and 3.3, we then have that $\mathbf{C}_N(x) = \mathbf{C}_N(\mathbf{y} \uparrow \mathbf{e})$. Thus from the definition of the equivalence relation $\simeq$ on $Z_N^*$, it follows that $x \simeq \mathbf{y} \uparrow \mathbf{e}$. □

The following lemma shows that if $k = O(\log |N|)$, then for a dominant vector $\mathbf{y}$ of $Z_N^*$ and $z \in Z_N^*$ it is easy to find the (unique) vector $\mathbf{f} \in \{0, 1\}^k$ such that $z \simeq \mathbf{y} \uparrow \mathbf{f}$.

**Lemma 3.8.** *Let $N$ be the product of $k$ distinct odd primes. Let $\mathbf{y}$ be dominant of $Z_N^*$ and let $z \in Z_N^*$. If $k = O(\log |N|)$, then there is a deterministic polynomial-time algorithm FIND with access to a promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ that on input $\langle \mathbf{y}, z, N \rangle$ always outputs the (unique) vector $\mathbf{f} \in \{0, 1\}^k$ such that $z \simeq \mathbf{y} \uparrow \mathbf{f}$.*

**Proof.** The algorithm is the following:

**Algorithm FIND**

*Input*: $\langle \mathbf{y}, z, N \rangle$, where $\mathbf{y}$ is dominant of $Z_N^*$ and $z \in Z_N^*$.
*Output*: $\mathbf{f} \in \{0, 1\}^k$ such that $z \simeq \mathbf{y} \uparrow \mathbf{f}$.

Step 1: Compute $q_{\mathbf{e}} \equiv (\mathbf{y} \uparrow \mathbf{e}) \times z \pmod N$ for each $\mathbf{e} \in \{0, 1\}^k$.
Step 2: Query $q_{\mathbf{e}}$ to the promise oracle for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ to get the answer $a_{\mathbf{e}} \in \{0, 1\}$ for each $\mathbf{e} \in \{0, 1\}^k$.
Step 3: If $a_{\mathbf{e}} = 1$ for some $\mathbf{e} \in \{0, 1\}^k$, then output $\mathbf{f} = \mathbf{e}$; otherwise output $\perp$.

The correctness of the algorithm above follows from previous lemmas. □

### 3.2. *A Low Communication Competitive Interactive Proof for Promised* QR

We now describe the protocol for a competitive interactive proof system for $\langle \mathrm{QR}_k, \mathrm{QNR}_k \rangle$ with low communication complexity.

**Protocol PQR** (A Competitive Interactive Proof System for Promised QR)

common input: $\langle x, N \rangle \in \mathrm{QR}_k \cup \mathrm{QNR}_k$, where $k = O(\log |N|)$.

P1: $P$ runs the machine $D$ to obtain either a dominant vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ of $Z_N^*$ or $\perp$.
$P \rightarrow V$: The vector $\mathbf{y}$ obtained in step P1.
V1-1: If $V$ receives $\perp$ from $P$, then $V$ halts and rejects $\langle x, N \rangle$; otherwise $V$ continues.
V1-2: $V$ chooses $\mathbf{a}_j \in_{\mathrm{R}} \{0, 1\}^k$ and $r_j \in_{\mathrm{R}} Z_N^*$ and computes $z_j \equiv (\mathbf{y} \uparrow \mathbf{a}_j) \times r_j^2 \pmod N$ for each $j$ $(0 \le j \le 1)$.
$V \rightarrow P$: $z_0, z_1 \in Z_N^*$.
P2: $P$ computes $\alpha_j \in \{0, 1\}^k$ such that $z_j \simeq \mathbf{y} \uparrow \alpha_j$ for each $j$ $(0 \le j \le 1)$.
$P \rightarrow V$: $\alpha_0, \alpha_1 \in \{0, 1\}^k$.
V2-1: If either $\alpha_0 \neq \mathbf{a}_0$ or $\alpha_1 \neq \mathbf{a}_1$, then $V$ halts and rejects $\langle x, N \rangle$; otherwise $V$ continues.
V2-2: $V$ chooses $e_j \in_{\mathrm{R}} \{0, 1\}$, $\mathbf{b}_j \in_{\mathrm{R}} \{0, 1\}^k$, and $s_j \in_{\mathrm{R}} Z_N^*$ and computes $w_j \equiv x^{e_j} \times (\mathbf{y} \uparrow \mathbf{b}_j) \times s_j^2 \pmod N$ for each $j$ $(0 \le j \le 1)$.
$V \rightarrow P$: $w_0, w_1 \in Z_N^*$.
P3: $P$ computes $\beta_j \in \{0, 1\}^k$ such that $w_j \simeq \mathbf{y} \uparrow \beta_j$ for each $j$ $(0 \le j \le 1)$.

$P \rightarrow V$: $\beta_0, \beta_1 \in \{0,1\}^k$.

V3: If either $\beta_0 \neq \mathbf{b}_0$ or $\beta_1 \neq \mathbf{b}_1$, then $V$ halts and rejects $\langle x, N\rangle$; otherwise $V$ halts and accepts $\langle x, N\rangle$.

**Correctness of PQR.** We show that when $k = O(\log |N|)$, Protocol PQR is a competitive interactive proof system for $\langle \mathrm{QR}_k, \mathrm{QNR}_k\rangle$.

(*Completeness*) Assume that $\langle x, N\rangle \in \mathrm{QR}_k$. It follows from Lemma 3.6 that, in step V1-1, $V$ receives a dominant vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ of $Z_N^*$ from $P$ with probability at least 3/4.

Assume that $\mathbf{y}$ is dominant of $Z_N^*$. Then $P$ can execute step P2 by running FIND and this implies that $V$ never rejects $\langle x, N\rangle$ in step V2-1. In addtion, $P$ can also execute step P3 by running FIND and thus $V$ will not reject $\langle x, N\rangle$ in step V3, because $x$ being a square modulo $N$ implies $\mathbf{C}_N(x) = \mathbf{0}$. It follows from Lemmas 3.6 and 3.8 that $P$ runs in probabilistic polynomial (in $|N|$) time.

(*Soundness*) Assume that $\langle x, N\rangle \in \mathrm{QNR}_k$. If $V$ receives $\perp$ from $P$ in step V1-1, then $V$ always halts and rejects $\langle x, N\rangle \in \mathrm{QNR}_k$. Then we consider the case that a dishonest prover $P^*$ sends $V$ a vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ over $Z_N^*$.

Assume that the vector $\mathbf{y}$ is not dominant of $Z_N^*$. For each $z_j \in Z_N^*$ $(0 \leq j \leq 1)$ in step V1-2, there are $2^t$ possible $\alpha_j \in \{0,1\}^k$ for some $t$ $(1 \leq t \leq k)$ such that $z_j \simeq \mathbf{y} \uparrow \alpha_j$. This implies that if $\mathbf{y}$ is not dominant of $Z_N^*$, then, with probability at most $2^{-2t} \leq 1/4$, any all powerful $P^*$ can find a vector $\alpha_j \in \{0,1\}^k$ such that $\alpha_j = \alpha_j$ for each $j$ $(0 \leq j \leq 1)$ in step P2. Thus if $\mathbf{y}$ is not dominant of $Z_N^*$, then $V$ halts and rejects $\langle x, N\rangle \in \mathrm{QNR}_k$ in step V2-1 with probability at least 3/4.

Assume that $\mathbf{y}$ is dominant of $Z_N^*$. Since $\langle x, N\rangle \in \mathrm{QNR}_k$, a unique nonzero vector $\mathbf{e} \in \{0,1\}^k$ exists such that $x \simeq \mathbf{y} \uparrow \mathbf{e}$. For each $j$ $(0 \leq j \leq 1)$, $\beta_j^0, \beta_j^1 \in \{0,1\}^k$ exist such that $w_j \simeq \mathbf{y} \uparrow \beta_j^0$ and $w_j \simeq x \times (\mathbf{y} \uparrow \beta_j^1)$. Indeed, for each $i, j$ $(0 \leq i, j \leq 1)$, $\beta_j^i \equiv \mathbf{b}_j + \{(e_j + i) \times \mathbf{e}\} \pmod 2$. This implies that any dishonest prover $P^*$ cannot guess better than at random the value of $e_j \in \{0,1\}$ for each $j$ $(0 \leq j \leq 1)$ even if it is infinitely powerful. Thus if $\mathbf{y}$ is dominant of $Z_N^*$, then, with probability at most 1/4, any all powerful $P^*$ can find a vector $\beta_j \in \{0,1\}^k$ such that $\beta_j = \mathbf{b}_j$ for each $j$ $(0 \leq j \leq 1)$ in step P3. This implies that $V$ halts and rejects $\langle x, N\rangle \in \mathrm{QNR}_k$ in step V3 with probability at least 3/4.

Thus for any $\langle x, N\rangle \in \mathrm{QNR}_k$, any all powerful dishonest prover $P^*$ can cause the honest verifier $V$ to accept $\langle x, N\rangle \in \mathrm{QNR}_k$ with probability at most 1/4. □

Recently, Goldreich (private communication, 1993) observed that the algorithm given in Proposition 6.6 of [BG] to find a representative vector can be modified to yield an algorithm that works also for $k = O(\log |N|)$. Thus using this in the protocol [BG], a competitive interactive proof system is obtained for $\langle \mathrm{QR}_k, \mathrm{QNR}_k\rangle$ that works for $k = O(\log |N|)$, but its communication complexity is still large.

## 4. Concluding Remarks

On common input $\langle x, N\rangle \in \mathrm{QR}_k \cup \mathrm{QNR}_k$ to our protocol, the prover sends the verifier about $k|N|$ bits and the verifier sends the prover about $4|N|$ bits, where $k = O(\log |N|)$. The constraint on $k$ seems to be indispensable in the protocol by Bellare and Goldwasser

[BG] and in the one presented here, but can it be further relaxed? It is obvious that, for any composite number $N$, $k = O(|N|)$. So one might ask:

- If $k = O(|N|)$, then does a competitive interactive proof system for promised $\mathrm{QNR}_k\langle \mathrm{QR}_k, \mathrm{QNR}_k\rangle$ exist?

In our protocol presented here, the constraint that $k = O(\log |N|)$ seems to be essential especially in the proof of completeness. Thus to solve the problem above affirmatively, we may need to use completely different approaches.

The promise requirement on QR plays an essential role in the protocol by Bellare and Goldwasser [BG] and in the one presented here. Thus it is still open whether QR (not promised QR) has a competitive interactive proof system.

## Acknowledgments

The authors wish to thank the referees and the editor for their many helpful suggestions to improve the quality and the readability of the paper.

## References

[B] Babai, L., Trading Group Theory for Randomness, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pp. 421–429 (1985).

[BCC] Brassard, G., Chaum, D., and Crépeau, C., Minimum Disclosure Proofs of Knowledge, *J. Comput. System Sci.*, Vol. 37, No. 2, pp. 156–189 (1988).

[BF] Beigel, R., and Feigenbaum, J., On Being Incoherent Without Being Very Hard, *Comput. Complexity*, Vol. 2, No. 1, pp. 1–17 (1992).

[BFL] Babai, L., Fortnow, L., and Lund, C., Nondeterministic Exponential Time Has Two-Prover Interactive Protocols, *Comput. Complexity*, Vol. 1, No. 1, pp. 3–40 (1991).

[BG] Bellare, M., and Goldwasser, S., The Complexity of Decision Versus Search, *SIAM J. Comput.*, Vol. 23, No. 1, pp. 97–119 (1994).

[BK] Blum, M., and Kannan, S., Designing Programs that Check Their Work, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pp. 86–97 (1989).

[ESY] Even, S., Selman, A., and Yacobi, Y., The Complexity of Promise Problems with Applications to Public-Key Cryptography, *Inform. and Control*, Vol. 61, pp. 159–173 (1984).

[FFS] Feige, U., Fiat, A., and Shamir, A., Zero-Knowledge Proofs of Identity, *J. Cryptology*, Vol. 1, pp. 77–94 (1988).

[FS] Fiat, A., and Shamir, A., How To Prove Yourself: Practical Solutions to Identification and Signature Problems, *Proceedings of Crypto '86*, Lecture Notes in Computer Science, Vol. 263, Springer-Verlag, Berlin, pp. 186–194 (1987).

[GMR] Goldwasser, S., Micali, S., and Rackoff, C., The Knowledge Complexity of Interactive Proof Systems, *SIAM J. Comput.*, Vol. 18, No. 1, pp. 186–208 (1989).

[GMW] Goldreich, O., Micali, S., and Wigderson, A., Proofs that Yield Nothing but Their Validity or All Languages in $\mathcal{NP}$ Have Zero-Knowledge Interactive Proof Systems, *J. Assoc. Comput. Mach.*, Vol. 38, No. 1, pp. 691–729 (1991).

[GS] Grollmann, J., and Selman, A., Complexity Measures for Public-Key Cryptosystems, *SIAM J. Comput.*, Vol. 17, No. 2, pp. 309–335 (1988).

[LFKN] Lund, C., Fortnow, L., Karloff, H., and Nisan, N., Algebraic Method for Interactive Proof Systems, *J. Assoc. Comput. Mach.*, Vol. 39, No. 4, pp. 859–868 (1992).

[S] Shamir, A., $\mathcal{IP} = \mathcal{PSPACE}$, *J. Assoc. Comput. Mach.*, Vol. 39, No. 4, pp. 869–877 (1992).

[TW] Tompa, M., and Woll, H., Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information, *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pp. 472–482 (1987).

[Y] Yao, A., Coherent Functions and Program Checkers, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pp. 84–94 (1990).