SUBGROUP MEMBERSHIP PROBLEM AND ITS APPLICATIONS TO INFORMATION SECURITY

Akihiro Yamamura* and Taiichi Saito**

Received December 18, 2001

Dedicated to the 60th birthday of Professor Masami Ito

Abstract. The widely used algorithmic problems, the quadratic residue problem and the decision Diffie-Hellman problem, are characterized as the subgroup membership problem. Several cryptographic schemes are realized assuming the hardness of the subgroup membership problem. We apply the subgroup membership problem to several information security schemes: a probabilistic encryption, a bit commitment and a private information retrieval.

1 Introduction It is well known that the *subgroup membership problem* for a finitely presented group is not decidable in general because Novikov-Boone theorem claims the existence of finitely presented group whose word problem is unsolvable. This implies that for a certain finitely presented group, there exists no procedure to check whether or not an element given as a word is equal to the identity element of the group. The subgroup membership problem is often called the *generalized word problem* in the literature of combinatorial group theory. On the other hand, the word problem is always solvable for the class of finite groups or finitely generated abelian groups.

However, if we consider more practical computation, that is, the bounded probabilistic polynomial time algorithms (or equivalently the computation class **BPP**), the membership problem is not trivial even for the class of finite abelian groups. When we consider a mathematical object, the object is described by finite data. The effectiveness is measured by the asymptotical behavior of algorithm to carry out certain tasks like deciding a mathematical proposition (equivalently calculating a Boolean predicate) and computing functions. In the case of decision problems for finitely presented groups, we consider the class of recursive functions. In specific cases like automatic groups and word hyperbolic groups, the word problem can be solved in polynomial time with respect to the word length. In the case of finite groups, the description of groups has simple structure, and any decision problem is solvable. We are interested in the effectiveness of such algorithmic problems. The behavior of algorithm is related to the size of data structure of a group family.

Several algorithmic problems used in cryptography are characterized as the subgroup membership problem. We note that there exists no known probabilistic polynomial time algorithm for the integer factorization or the discrete logarithm problem for some class of finite cyclic groups. So these problems are not in the class of **BPP**. The quadratic residue (QR for short) problem and the decision Diffie-Hellman (DDH for short) problem have numerous applications in cryptography, and hence, they have been studied in detail. In [17], the similarity of QR and DDH is discussed. We now give more formal approach to generalize and formalize cryptographic hard problems as the *subgroup membership problem*, and show many other algorithmic problems, which are used in public key cryptography, are

2000 *Mathematics Subject Classification.* 11Y16, 68Q17, 94A60.

Key words and phrases. Subgroup Membership Problem, Decision Diffie-Hellman Problem, Quadratic Residue Problem, Probabilistic Encryption, Private Information Retrieval,

characterized as the subgroup membership problem as well. Such a unification of algorithmic problems used in cryptography has not been appeared up to date as far as the authors know.

Widely used assumptions in cryptography are divided into two groups: the algorithmic assumptions related to the integer factoring (and the QR) and the algorithmic assumptions related to the discrete logarithm problem (and the DDH). The first is originated from the RSA cryptosystem [15] and the second from the Diffie-Hellman key exchange protocol [6]. These two look different and are usually discussed separately. The unified approach to the integer factoring problem and the discrete logarithm problem shed light on the fundamental properties of algorithms required to provide the security. Therefore, we can get better understanding of the algorithmic problems by unified treatment of subgroup membership problems. To apply the membership problem to cryptographic schemes such as asymmetric cryptosystems, we require the efficiency of computation for legal participants and the existence of a trapdoor. Once we prove that the subgroup membership problem is applicable to a certain scheme in general, then any primitive based on the subgroup membership problem concerning a specific group is applicable to the scheme in principle. As an example, in this paper, we show that any subgroup membership problem can be employed to construct a computational PIR system by constructing a PIR system using the subgroup membership problem in a general manner.

2 Subgroup Membership Problem

Determining the membership of a given element of a certain group in its subgroup is not always easy. As a matter of fact, the membership problem of a subgroup in a finitely presented group is not recursive in general. To apply the membership problem to cryptographic schemes such as asymmetric cryptosystems, we require the efficiency of computation for legal participants and the existence of a trapdoor. In this section we consider the subgroup membership problem with a trapdoor, and show that several problems widely used in cryptography are characterized as the subgroup membership problem.

Let G be a group, and let H be its subgroup. The membership problem is to decide whether or not a given element g in G belongs to H. Furthermore, we consider a family of finite groups indexed by a parameter and the asymptotic behavior according to computation. In such a case, the subgroup membership is described as a computation problem to decide the membership when given an element, a subgroup and a group indexed by a parameter. A computation problem is hard if no efficient algorithms. The efficiency is characterized by the asymptotic behavior of an algorithm

2.1 Subgroup Membership Assumption

We suppose that every element in G has a binary representation of size k, where k is the security parameter. The membership can be decided within polynomial time in k if a certain information, called a *trapdoor*, is provided. The membership of an element g in G in H can be decided provided the trapdoor, however, the membership cannot be decided with a probability substantially larger than one half without the trapdoor. We now formalize the subgroup membership problem.

Let k be the security parameter. For the input 1^k, a probabilistic polynomial time algorithm $\mathcal{IG}$ outputs the description of a group G, the description of a subgroup H of G and the trapdoor that provides a polynomial time algorithm for the subgroup membership problem of H in G. The algorithm $\mathcal{IG}$ is called the *instance generator*. Every element of G is represented as a binary sequence of length k. Computation of the multiplication in G is performed in polynomial time in k.

The predicate for the membership of a subgroup is denoted by Mem, that is, Mem is

defined as follows.

$$\mathrm{Mem}(G,H,x) = \begin{cases} 1 & \text{if } x \text{ lies in } H \\ 0 & \text{if } x \text{ lies in } S \end{cases} ,$$

where $\mathcal{IG}$ outputs the pair (G,H) for 1^k, x is in G, and $S = G \backslash H$. The *subgroup membership problem* is to compute Mem in polynomial time in k when we inputs 1^k and obtain a pair of groups (G,H) and an element g in G, which is uniformly and randomly chosen from H or G according to the coin toss $b \stackrel{R}{\leftarrow} \{0,1\}$. If there does not exist a probabilistic polynomial time algorithm that computes Mem with a probability substantially larger than $\frac{1}{2}$, then we say that the membership problem is *intractable*. We also assume that one can choose uniformly and randomly an element from both H and G. This is significant to apply to cryptographic schemes.

The following is trivial, however, it is useful for the construction of a PIR system based on the subgroup membership problem.

Proposition 2.1 *Let G be a group, and let H be a subgroup of G. For any g in G and h in H, gh lies in H if and only if g lies in H.* □

Subgroup Membership Assumption I
For every constant c, and every family $\{C_k \mid k \in \mathbb{N}\}$ of circuits of polynomial size in k, there is an integer K such that for all $k > K$ we have

$$\mathbf{Prob}(C_k(G,H,g) = \mathrm{Mem}(G,H,g)) < \frac{1}{2} + \frac{1}{k^c} , \tag{2.2.1}$$

where the probability is taken over $(G,H) \leftarrow \mathcal{IG}(1^k)$, $b \stackrel{R}{\leftarrow} \{0,1\}$, $g \stackrel{R}{\leftarrow} H$ if $b = 1$, $g \stackrel{R}{\leftarrow} S$ if $b = 0$.

The assumption claims that there exists no polynomial size circuit family to compute the predicate Mem. The following is equivalent to the assumption above.

Subgroup membership assumption II
For every constant c, and every family $\{C_k \mid k \in \mathbb{N}\}$ of circuits of polynomial size in k, there is an integer K such that for all $k > K$ we have

$$|\mathbf{P}_H - \mathbf{P}_S| < \frac{1}{k^c} , \tag{2.2.2}$$

where the probabilities $\mathbf{P}_H$ and $\mathbf{P}_S$ are defined as follows;

$$\mathbf{P}_H = \mathbf{Prob}_{(G,H)\leftarrow\mathcal{IG}(1^k)\ ;\ g\stackrel{R}{\leftarrow}H}(C_k(G,H,g) = 1) ,$$

and

$$\mathbf{P}_S = \mathbf{Prob}_{(G,H)\leftarrow\mathcal{IG}(1^k)\ ;\ g\stackrel{R}{\leftarrow}S}(C_k(G,H,g) = 1) .$$

2.2 Examples We exhibit several subgroup membership problems: the DDH problem, the QR problem, the rth residue (RR for short) problem studied by Kurosawa and Tsujii [10], the p-subgroup (PSUB for short) problem introduced by Okamoto and Uchiyama [13] and the decisional composite residue (DCR for short) problem introduced by Paillier [14]. Recall that the assumption that the QR problem is intractable (QR assumption) is employed to prove the semantic security of the Goldwasser-Micali cryptosystem [8], and the

assumption that the DDH problem is intractable (DDH assumption) is employed to prove the semantic security of the ElGamal cryptosystem. These two have many other applications. The assumption that one of problems above is intractable is employed to prove the semantic security of the corresponding cryptosystem [10], [13], [14], respectively. We also note that the security of the cryptosystem introduced by Naccache and Stern [11] depends on the PSUB assumption as well.

Quadratic Residue Problem
Let p, q be prime integers. Set $N = pq$. The primes p and q are trapdoor information for the quadratic residue problem, on the other hand, the number N is public information. Let G be the subgroup of $(\mathbb{Z}/(N))^*$ consisting of the elements whose Jacobi symbol is 1, and let H be the subgroup of G consisting of quadratic residues of G, that is,

$$H = \{x \in G \mid x = y^2 \bmod N \text{ for } y \in (\mathbb{Z}/(N))^*\}.$$

The quadratic residue problem of H in G is to decide whether or not, a given element $g \in G$, g belongs to H. We can effectively determine the membership of g in H provided that the information p and q are available. No polynomial time algorithm is known for the membership of a randomly chosen element of G in H without the information p and q. Hence, if we define an instance generator for the QR problem as a probabilistic algorithm that outputs two primes p and q of size k and a quadratic non-residue h whose Jacobi symbol is 1 for the input 1^k, then the QR problem is considered as the subgroup membership problem. Note that we can obtain a quadratic non-residue h with Jacobi symbol 1 by using p, q, and that it is possible to uniformly and randomly choose elements from H without the trapdoor information provided h is given.

Decision Diffie-Hellman Problem
Let C be a cyclic group of prime order p. The group C may be the multiplication group of a finite field or the group of rational points of an elliptic curve. Let g be a generator of C. The decision Diffie-Hellman problem is to decide whether or not $h_2 = g_2^a$ for the given quadruple (g_1, h_1, g_2, h_2) of elements in C with $h_1 = g_1^a$ for some $1 \leq a \leq p-1$. If so, we say that (g_1, h_1, g_2, h_2) is a Diffie-Hellman quadruple. The integer a is the trapdoor of the decision Diffie-Hellman problem. Knowing the trapdoor a, we can efficiently decide whether or not $h_2 = g_2^a$.

The DDH problem can be characterized as the subgroup membership problem for a certain group as follows We set G to be the direct product $C \times C$. Then the input to the DDH problem is (x, y) where $x, y \in G$, that is, $x = (g_1, h_1)$ and $y = (g_2, h_2)$. It is obvious that (g_1, h_1, g_2, h_2) is a Diffie-Hellman quadruple if and only if y belongs to the subgroup $< x >$ of G generated by x. It follows that the DDH problem for the cyclic group C is equivalent to the subgroup membership problem of the group $H =< x >$, where $x = (g_1, g_1^a)$, in the group

$$G = C \times C =< g_1 > \times < g_1 > .$$

Note that, when a generator x of H is given, it is possible to choose uniformly and randomly elements from H without the trapdoor information.

Rth Residue Problem
The RR problem is a natural extension of the QR problem defined as follows. Let p, q be primes, and let e_1, e_2 be odd integers dividing $p-1$ and $q-1$, respectively, such that e_1 is prime to $q-1$ and e_2 is prime to $p-1$. Set $N = pq$ and $r = e_1 e_2$. The primes p and q

are the trapdoor information for the RR problem, on the other hand, the number N and r are the public information. Let G be the group $(\mathbb{Z}/(N))^*$, and let H be the subgroup consisting of rth residues of G, that is,

$$H = \{x \in G \mid x = y^r \bmod N \text{ for } y \in G\}.$$

The RR problem of H in G is to decide whether or not, a given element $g \in G$, g belongs to H. Thus, the RR is a subgroup membership problem of H in G. We can effectively determine the membership of g in H provided that the information p and q are available. No polynomial time algorithm is known for the membership of a randomly chosen element of G in H without the information p and q. Note that we can obtain an element h such that h^i does not lie in $\{x^r \bmod N : x \in (\mathbb{Z}/(N))^*\}$ for any $1 \leq i \leq r-1$ by using the trapdoor information, and that we can uniformly and randomly choose an element from H provided h is given.

P-Subgroup Problem
Let p, q be primes such that p does not divide $q-1$. Set $N = p^2q$ and let g be a random element in $(\mathbb{Z}/(N))^*$ such that the order of $g^{p-1} \bmod p^2$ is p. The primes p and q are trapdoor information for the PSUB problem, on the other hand, the number N, g, k are public information. Let G be a group defined by

$$G = \{x \mid x = g^m y^N \bmod N \text{ for } m \in \mathbb{Z}/(p) \text{ and } y \in (\mathbb{Z}/(N))^*\},$$

and let H be the subgroup defined by

$$H = \{x \mid x = y^N \bmod N \text{ for } y \in G\}.$$

The PSUB problem of H in G is to decide whether or not, a given element g in G, g belongs to H. Thus, the PSUB is the membership problem of H in G. We can efficiently determine the membership of g in H provided that the information p and q are available. No polynomial time algorithm is known for the membership of a randomly chosen element of G in H without the information p and q. Note that our description of PSUB is slightly different from Okamoto-Uchiyama [13]. Naccache and Stern [11] implicitly used PSUB problem in their scheme. Paillier introduces the *decisional composite residue* (DCR for short). This is a generalization of [13] and also characterized as a subgroup membership problem.

For other plausible applications of the subgroup membership problem, the reader is also referred to [16] in which the DDH assumption is applied to the cryptographic schemes which only known method to construct is to base on the QR assumption.

We summarize the examples in Table 1. We note that the table is not exhaustive at all.

	Related Problem	Group / Subgroup	Applications
DDH	DLP	$C \times C$: Direct Product of Cyclic Groups	ElGamal
	DH	$\langle (g,h) \rangle$: Subgroup Generated by (g,h)	
QR	FACT(pq)	$\{x \in \mathbb{Z}_N^* \mid (\frac{x}{N}) = 1\}$	Goldwasser-Micali [8]
		$\{x^2 \bmod N \mid x \in \mathbb{Z}_N^*\}$	
RR	FACT(pq)	$\mathbb{Z}_N^*$	Kurosawa-Tsujii [10]
		$\{x^r \bmod N \mid x \in \mathbb{Z}_N^*\}$	
PSUB	FACT(p^2q)	$\{x \mid x = g^m y^N \bmod N$ for $m \in \mathbb{Z}/(p),\ y \in (\mathbb{Z}/(N))^*\}$	Okamoto-Uchiyama [13]
		$\{y^N \bmod N \mid y \in \mathbb{Z}_N^*\}$	Naccache-Stern [11]
DCR	FACT(pq)	$\{x \mid x = g^m y^N \bmod N^2$ $m \in \mathbb{Z}/(N), y \in (\mathbb{Z}/(N^2))^*\}$	Paillier [14]
		$\{y^N \bmod N^2 \mid y \in (\mathbb{Z}/(N^2))^*\}$	

Table 1: Subgroup Membership Problems

2.3 Equivalent Problems We examine several algorithmic problems equivalent to the subgroup membership problem. Suppose that $\mathcal{IG}$ is an instance generator of a family of groups, and that $\mathcal{IG}$ outputs (G, H) for the input 1^k. We set $S = G \setminus H$. Suppose that t is an integer bounded above by a polynomial in k. Let K_i be the direct product of $t-1$ H's and S, where all jth position $(j \neq i)$ is occupied by H except for ith position, that is,

$$K_i = H \times H \times \cdots \times \overset{i}{S} \times \cdots \times H$$

for every $i = 1, 2, \ldots, t$. Let L be the union of K_1, K_2, $\cdots$, K_t, that is,

$$L = K_1 \bigcup K_2 \bigcup \cdots \bigcup K_t.$$

Pattern Indistinguishability Assumption
The *pattern indistinguishability assumption* is to assume the following holds: for every constant c, every family $\{C_k \mid k \in \mathbb{N}\}$ of circuits of polynomial size in k and all i, j such that $1 \leq i, j \leq n$ there is an integer K such that for all $k > K$ we have

$$|\mathbf{P}_i - \mathbf{P}_j| < \frac{1}{k^c} \ . \tag{2.2.3}$$

Here the probabilities $\mathbf{P}_i$ and $\mathbf{P}_j$ are defined as follows.

$$\mathbf{P}_i = \mathbf{Prob}_{(G,H) \leftarrow \mathcal{IG}(1^k)\ ;\ (g_1, g_2 \ldots, g_t) \overset{R}{\leftarrow} K_i}(C_k(G, H, i, g_1, g_2 \ldots, g_t) = 1)$$

$$\mathbf{P}_j = \mathbf{Prob}_{(G,H) \leftarrow \mathcal{IG}(1^k)\ ;\ (g_1, g_2 \ldots, g_t) \overset{R}{\leftarrow} K_j}(C_k(G, H, i, g_1, g_2 \ldots, g_t) = 1)$$

General Pattern Indistinguishability Assumption
The *general pattern indistinguishability assumption* is to assume the following holds: for

every constant c, every family $\{C_k \mid k \in \mathbb{N}\}$ of circuits of polynomial size in k and all $(i_1, i_2, \ldots, i_u)$ and $(j_1, j_2, \ldots, j_u)$, there is an integer K such that for all $k > K$ we have

$$|\mathbf{P}_{(i_1,i_2,\ldots,i_u)} - \mathbf{P}_{(j_1,j_2,\ldots,j_u)}| < \frac{1}{k^c} \ . \tag{2.2.4}$$

Here the probabilities $\mathbf{P}_{(i_1,i_2,\ldots,i_u)}$ and $\mathbf{P}_{(j_1,j_2,\ldots,j_u)}$ are defined by

$$\mathbf{P}_{(i_1,i_2,\ldots,i_u)} = \mathbf{Prob}(C_k(G, H, x_1, x_2 \ldots, x_u) = 1) \ ,$$

where the probability is taken over

$$(G, H) \leftarrow \mathcal{IG}(1^k)$$

and

$$(x_1, x_2 \ldots, x_u) \stackrel{R}{\leftarrow} K_{i_1} \times K_{i_2} \times \cdots \times K_{i_u},$$

and

$$\mathbf{P}_{(j_1,j_2,\ldots,j_u)} = \mathbf{Prob}(C_k(G, H, x_1, x_2 \ldots, x_u) = 1) \ ,$$

where the probability is taken over

$$(G, H) \leftarrow \mathcal{IG}(1^k)$$

and

$$(x_1, x_2 \ldots, x_u) \stackrel{R}{\leftarrow} K_{j_1} \times K_{j_2} \times \cdots \times K_{j_u}.$$

Direct Product Indistinguishability Assumption
Let M be an element in $\{0,1\}^l$. Then $M = (b_1, b_2, \ldots, b_l)$, where b_i belongs to $\{0,1\}$. Let $D(M)$ be the direct product $S_1 \times S_2 \times S_3 \times \cdots S_l$, where $S_i = H$ if $b_i = 1$ and $S_i = G \setminus H$ otherwise.

The *direct product indistinguishability assumption* is defined as follows. For every constant c, every family $\{C_k \mid k \in \mathbb{N}\}$ of circuits of polynomial size in k, and all M_1, M_2 in $\{0,1\}^l$, there exists an integer K such that for all $k > K$ we have

$$|\mathbf{P}_1 - \mathbf{P}_2| < \frac{1}{k^c} \ . \tag{2.2.5}$$

Here the probabilities $\mathbf{P}_1$ and $\mathbf{P}_2$ are defined as follows.

$$\mathbf{P}_1 = \mathbf{Prob}_{(G,H)\leftarrow\mathcal{IG}(1^k)\ ;\ (g_1,g_2\ldots,g_l)\stackrel{R}{\leftarrow}D(M_1)}(C_k(G, H, i, g_1, g_2 \ldots, g_l) = 1)$$

$$\mathbf{P}_2 = \mathbf{Prob}_{(G,H)\leftarrow\mathcal{IG}(1^k)\ ;\ (g_1,g_2\ldots,g_l)\stackrel{R}{\leftarrow}D(M_2)}(C_k(G, H, i, g_1, g_2 \ldots, g_l) = 1)$$

Theorem 2.2 *The following are equivalent.*
(1) The subgroup membership assumption I.
(2) The subgroup membership assumption II.
(3) The pattern indistinguishability assumption.
(4) The general pattern indistinguishability assumption.
(5) The direct product indistinguishability assumption.

Proof. We show the equivalence among (1), (2), (3). Clearly (1) implies (3). The proof for the equivalence between (1), (4) and (5) can be shown similarly.

(2) implies (1): Suppose that there exists a constant c and that for every K, there is $k \geq K$ such that the circuit C_k does not satisfy (2.2.1). Note that

$$\begin{aligned} \mathbf{Prob}(C_k(G,H,g) &= \mathrm{Mem}(G,H,g)) \\ &= \frac{1}{2}\mathbf{P}_H + \frac{1}{2}(1-\mathbf{P}_S). \end{aligned} \tag{2.2.6}$$

Since (2.2.1) does not hold, we have

$$\frac{1}{2}(\mathbf{P}_H - \mathbf{P}_S + 1) > \frac{1}{2} + \frac{1}{k^c}.$$

Therefore we have

$$|\mathbf{P}_H - \mathbf{P}_S| > \frac{2}{k^c}.$$

(1) implies (2): Suppose that there exists a constant c and that for every k, there is $k \geq K$ such that the circuit C_k does not satisfy (2.2.2). For the circuit C_k, we have

$$\begin{aligned} \mathbf{Prob}(C_k(G,H,g) &= \mathrm{Mem}(G,H,g)) \\ &= \frac{1}{2}\mathbf{P}_H + \frac{1}{2}(1-\mathbf{P}_S) = \frac{1}{2}(1+\mathbf{P}_H - \mathbf{P}_S) > \frac{1}{2} + \frac{1}{k^c}. \end{aligned} \tag{2.2.7}$$

(3) implies (2): Suppose that there exists a constant c and that for every k, there is $k \geq K$ such that the circuit C_k does not satisfy (2.2.3). Construct a circuit C'_k as follows. Given (G,H) and $g \in G$, we choose uniformly and randomly $x_1, x_2, \ldots, x_{t-2}$ form H. We also choose uniformly and randomly y from H. We toss a coin, say, $b \stackrel{R}{\leftarrow} \{0,1\}$. If $b = 0$, then we input $(G, H, x_1, x_2, \ldots, \overset{i}{y}, \ldots, \overset{j}{g}, \ldots, x_{t-2})$, and the circuit C'_k returns the output of C_k. If $b = 1$, then we input $(G, H, x_1, x_2, \ldots, \overset{i}{g}, \ldots, \overset{j}{y}, \ldots, x_{t-2})$, and the circuit C'_k returns the negation of the output of C_k. If $g \in S$, then we have

$$\mathbf{Prob}(C'_k(G,H,g) = 1 : g \leftarrow S) = \frac{1}{2}\mathbf{P}_i + \frac{1}{2}(1-\mathbf{P}_j).$$

If $g \in H$, then we have

$$\mathbf{Prob}(C'_k(G,H,g) = 1 : g \leftarrow H) = \frac{1}{2}\theta + \frac{1}{2}(1-\theta),$$

where

$$\theta = \mathbf{Prob}(C_k(G,H,g_1,g_2,\ldots,g_t))$$

and the probability is taken over $g_1, g_2, \ldots, g_t$ are taken uniformly and randomly from H. It follows that

$$|\mathbf{P}_H - \mathbf{P}_S| > \frac{1}{2}|\mathbf{P}_i - \mathbf{P}_j| > \frac{1}{2k^c}.$$

□

2.4 Probabilistic Encryption Goldwasser and Micali [8] introduce a *semantic secure* probabilistic encryption scheme, whose security is based on the QR assumption. An encryption is called *semantic secure* if the information leaked to a passive enemy is computationally negligible. This concept is a computational version of Shannon's *perfect secrecy*. The concept is significant in modern cryptography.

The subgroup membership problem is applied to a probabilistic encryption. See [16] for a probabilistic encryption based on the decision Diffie-Hellman problem.

Key generation: Bob inputs 1^k to a probabilistic polynomial time algorithm $\mathcal{IG}$, called *instance generator*, and gets a pair (G, H) of groups and the trapdoor for the subgroup membership problem of H in G, where k is the security parameter. Every element of G is represented by a binary sequence of length k. We assume the subgroup membership assumption of H in G. Therefore, Alice can generates elements in both G and H uniformly and randomly. Bob publicizes G and H, but keeps the trapdoor information for the subgroup membership problem of H secret.

Encryption: Suppose Alice encrypts a message $M = b_1 b_2 b_3 \cdots n_l$, where b_i belongs to $\{0, 1\}$ for every $i = 1, 2, 3, \ldots, l$. For every b_i $(1 \leq i \leq l)$, Alice generates random element r_i, where r_i belongs to H if $b_i = 1$, and b_i belongs to $G \setminus H$ otherwise. Then the sequence of group elements $(r_1, r_2, r_3, \ldots, r_l)$ is an encrypted message for M. We note that the encrypted message is a random element in the direct product $S_1 \times S_2 \times S_3 \times \cdots S_l$, where $S_i = H$ if $b_i = 1$, and $S_i = G \setminus H$ otherwise. So the encryption is probabilistic.

Decryption: Bob knows the trapdoor for the subgroup membership problem of H in G. Hence, he can decides whether or not each element r_i belongs to H in polynomial time in the security parameter k.

Security: An encryption scheme is *semantic secure* if any adversary cannot computationally distinguish two ciphertexts of two messages of the same length. This means that no probabilistic polynomial time algorithm can distinguish two ciphertexts C_1 and C_2. It follows that the encryption above is semantic secure if and only if no probabilistic polynomial time algorithm can distinguish two direct products $S_1 \times S_2 \times S_3 \times \cdots S_l$ and $T_1 \times T_2 \times T_3 \times \cdots T_l$, where S_i and T_i depend on the messages M_1 and M_2. Therefore, the encryption is semantic secure if the direct product indistinguishability assumption holds. Thus, the encryption is semantic secure under the subgroup membership assumption for H in G.

Originally, Goldwasser and Micali used QR to construct a probabilistic encryption. Moreover, we can use any subgroup membership problem for a semantic secure encryption.

2.5 Bit Commitment Another possible application of the subgroup membership problem is the bit commitment scheme. We briefly describe a bit commitment scheme based on the subgroup membership problem. See [16] for a bit commitment scheme based on the decision Diffie-Hellman problem.

Key Generation Alice inputs 1^k to an instance generator $\mathcal{IG}$, and gets a pair (G, H) of groups and the trapdoor for the subgroup membership problem of H in G, where k is the security parameter. We assume the subgroup membership assumption of H in G. Alice publicizes G and H, but keeps the trapdoor information for the subgroup membership problem of H secret.

Committing Alice commits her bit b in $\{0, 1\}$. She also generates uniformly and randomly an element r according to her bit b so that r belongs to H if $b = 1$ and r belongs to $G \setminus H$ otherwise.

Verifying Alice confesses her bit b to Bob, and gives the trapdoor for the subgroup membership problem. Bob can verify the membership of the element r.

Thus, we can use any subgroup membership problem to construct a bit commitment protocol. We note that the bit commitment protocol can be used to construct a coin flipping protocol as well.

3 Private Information Retrieval Chor, Goldreich, Kushilevitz and Sudan [3] introduced the *private information retrieval scheme* for remote database access, in which the user can retrieve the data of user's choice without revealing it. Their scheme attains *information theoretic security*, however, the database must be replicated in several locations where the managers are not allowed to communicate each other. The *computational private information retrieval scheme* was introduced by Chor and Gilboa [4]. Their scheme attains more efficient communication than Chor, Goldreich, Kushilevitz and Sudan's model by sacrificing the information theoretic security, nevertheless, their scheme enjoys computational security by assuming the existence of pseudorandom generators. However, their scheme still needs replication of the database. Kushilevitz and Ostrovsky [9] introduced a computational private information retrieval scheme in which only one database is needed. Their scheme depends on the intractability of the quadratic residue problem. More efficiency, polylogarithmic communication complexity, is attained by Cachin, Micali and Stadler [2]. They assume a number theoretic hypothesis, which they call the Φ assumption, and sacrifice one-round communication and then obtain polylogarithmic communication complexity. However, a rigorous proof of the intractability of the Φ assumption or its equivalence to a widely used assumption like the quadratic residue assumption or the integer factorization is not given in [2]. We summarize the known results on private information retrievals in Table 2.

We briefly review the general scheme of a private information retrieval (PIR for short) scheme. A computational PIR scheme with a single database is a protocol for two players, a user $\mathcal{U}$ and a database manager $\mathcal{DB}$. Both are able to perform only probabilistic polynomial time computation. The database manager $\mathcal{DB}$ maintains a database, which is a binary sequence $X = x_0x_1x_2\cdots x_{n-1}$. The goal of the protocol is to allow $\mathcal{U}$ to obtain the ith bit x_{i+1} of X without leaking any information on x_i to $\mathcal{DB}$. The protocol runs as follows.

Step 1 $\mathcal{U}$ computes a query Query(i) using his random tape (coin toss), which $\mathcal{U}$ keeps secret. Then he sends Query(i) to $\mathcal{DB}$.

Step 2 $\mathcal{DB}$ receives Query(i). He performs a polynomial-time computation for the input X, Query(i) and his random tape. The computation yields the answer Answer(Query(i)). He sends Answer(Query(i)) back to $\mathcal{U}$.

Step 3 $\mathcal{U}$ receives Answer(Query(i)). He performs a polynomial-time computation using the answer Answer(Query(i)) and his private information (his random tape). The computation yields the ith bit x_{i+1} of the database.

Correctness
For any database sequence X and for any query Query(i) for ith bit of X, $\mathcal{U}$ obtains x_i at the end.

Privacy
$\mathcal{DB}$ cannot distinguish a query for the ith bit and a query for the jth bit for all i and j by a polynomial-time (probabilistic) computation with non-negligible probability. Formally, for all constants c, for all database of length n, for any two $1 \leq i, j \leq n$, and all polynomial-size

family of circuits C_k, there exists an integer K such that for all $k > K$ we have

$$|\mathbf{Prob}(C_k(\text{Query}(i)) = 1) - \mathbf{Prob}(C_k(\text{Query}(j)) = 1)| < \sigma \quad , \tag{3.3.1}$$

where k is the security parameter of the protocol and $\sigma = \frac{1}{(\text{Max}(k,n))^c}$.

Computation
Computations of both $\mathcal{DB}$ and $\mathcal{U}$ are bounded above by a polynomial in the size n of the database and the security parameter k.

Scheme	Round Number	Security Assumption	Communication Complexity	Number of DBs
Chor, Goldreich, Kushilevitz, Sudan [3]	1	Information Theoretical	$O(n^{1/3})$	≥ 2
Ambainis [1]	1	Information Theoretical	$O(n^{1/2k-1})$ for $k(>1)$ DBs	≥ 2
Chor and Gilboa [4]	1	Existence of Pseudo Number Generators	$O(n^c)$ for $c > 0$	≥ 2
Kushilevitz and Ostrovsky [9]	1	Quadratic Residue Problem Assumption	$O(n^c)$ for $c > 0$	1
Ostrovsky and Shoup [12]	Multiple	Reduction to Read only scheme		
Cachin, Micali and Stadler [2]	2	Φ Assumption	Polylogarithmic	1
Proposed Scheme	1	Subgroup Membership Assumption (e.g. DDH assumption)	$O(n^c)$ for $c > 0$	1

Table 2: Several Private Information Retrieval Schemes

3.1 PIR Scheme Based on the Subgroup Membership Problem We show that the subgroup membership problem can be applied to a PIR scheme by modifying Kushilevitz and Ostrovsky's scheme [9]. The proposed scheme has the same communication complexity as Kushilevitz and Ostrovsky's scheme whose security depends on the QR assumption. On the other hand, the security of the private information retrieval scheme proposed in this paper is based on the subgroup membership assumption. Therefore, we can construct a private information retrieval scheme based on any algorithmic problems in Section 2.2, in particular, we can use groups of rational points on elliptic curves or multiplicative groups of finite fields under the corresponding DDH assumption. We should remark that all the private information retrieval schemes proposed so far depend on either the existence of pseudorandom number generators or intractability assumption related to the integer factorization. No private information retrieval scheme based on the DDH has been proposed, yet as far as the authors know. Modifying [9], we construct a PIR scheme based on the subgroup membership problem.

3.2 Basic Idea First of all, we explain the basic idea of the scheme by a simple model. Suppose $\mathcal{DB}$ has the database $X = x_0x_1x_2\cdots x_{n-1}$ and that $\mathcal{U}$ wishes to know the ith bit

x_{i-1}. $\mathcal{U}$ chooses group elements g_0, g_1, g_2, ..., g_{i-1}, ..., g_{n-1} so that g_j in H for $j \neq i-1$ and g_{i-1} in $S = G \setminus H$. Then $\mathcal{U}$ sends them all to $\mathcal{DB}$. $\mathcal{DB}$ computes the group element $g = g_0^{x_0} g_1^{x_1} g_2^{x_2} \cdots g_{i-1}^{x_{i-1}} \cdots g_{n-1}^{x_{n-1}}$ and sends it back to $\mathcal{U}$. $\mathcal{DB}$ cannot get to know which of g_0, g_1, g_2, ..., g_{i-1}, ..., g_{n-1} comes from S if the subgroup membership problem of H in G is intractable. Since $\mathcal{U}$ possesses the trapdoor, he can determine whether or not g lies in H. By Proposition 1, g lies in H if and only if $x_{i-1} = 0$. Therefore, $\mathcal{U}$ can obtain the ith bit x_{i-1}. This simple model illustrates the idea of using the subgroup membership problem, but the communication complexity is still large. We need the trick by [9] to reduce the communication complexity.

3.3 Scheme We now describe the private information retrieval scheme using the subgroup membership problem.

Step 0 The user $\mathcal{U}$ inputs 1^k to the instance generator $\mathcal{IG}$ and then gets a pair (G, H) of groups and the trapdoor for the subgroup membership problem of H in G, where k is the security parameter and every element of G is represented by a binary sequence of length k. We assume the subgroup membership assumption of H in G. The group G is shared by both $\mathcal{DB}$ and $\mathcal{U}$. On the other hand, $\mathcal{U}$ keeps the trapdoor information for the subgroup membership problem of H secret. Computations of both $\mathcal{DB}$ and $\mathcal{U}$ are performed in the group G. Let X be the database managed by $\mathcal{DB}$. We suppose that $X = x_0 x_1 x_2 \cdots x_{n-1}$, where x_i lies in $\{0, 1\}$, and that $n = t^l$, where t, l are positive integers.

Step 1 $\mathcal{U}$ computes a query Query(i) for his desired bit x_{i-1}, where $1 \leq i \leq n$, in the following manner. First, $\mathcal{U}$ computes the t-adic expansion of i. Let $i = \alpha_0$. Then the t-adic expansion of i is $\beta_l \beta_{l-1} \cdots \beta_2 \beta_1$, where

$$
\begin{aligned}
\alpha_0 &= \alpha_1 t + \beta_1 \quad && 0 \leq \alpha_0 \leq t^{l-1} - 1, \text{ and } \quad 0 \leq \beta_1 \leq t-1 \\
\alpha_1 &= \alpha_2 t + \beta_2 \quad && 0 \leq \alpha_1 \leq t^{l-2} - 1, \text{ and } \quad 0 \leq \beta_2 \leq t-1 \\
\alpha_2 &= \alpha_3 t + \beta_3 \quad && 0 \leq \alpha_2 \leq t^{l-3} - 1, \text{ and } \quad 0 \leq \beta_3 \leq t-1 \\
& && \cdots\cdots \\
\alpha_{l-2} &= \alpha_{l-1} t + \beta_{l-1} \quad && 0 \leq \alpha_{l-2} \leq t-1, \text{ and } \quad 0 \leq \beta_{l-1} \leq t-1 \\
& && 0 \leq \alpha_{l-1} = \beta_l \leq t-1 \text{ and } \alpha_l = 0 \ .
\end{aligned}
\tag{3.3.2}
$$

For each u $(1 \leq u \leq l)$, $\mathcal{U}$ chooses uniformly and randomly $t-1$ elements $g_{(u,0)}$, $g_{(u,1)}$, ..., $g_{(u,\beta_u-1)}$, $g_{(u,\beta_u+1)}$, ..., $g_{(u,t-1)}$ from H. He also chooses uniformly and randomly $g_{(u,\beta_u)}$ from $S = G \setminus H$. $\mathcal{U}$ defines $Q(u)$ by

$$
(g_{(u,0)}, g_{(u,1)}, \cdots, g_{(u,\beta_u-1)}, g_{(u,\beta_u)}, g_{(u,\beta_u+1)}, \cdots, g_{(u,t-1)}) \ , \tag{3.3.3}
$$

that is, $Q(u)$ is a sequence of group elements of G such that the β_uth component is uniformly and randomly chosen from $S = G \setminus H$ and the others are uniformly and randomly chosen from H. Then, $Q(1)$, $Q(2)$, ..., $Q(l)$ comprise a query (denoted by Query(i)) for the ith bit x_{i-1} of X, and $\mathcal{U}$ sends Query(i) to $\mathcal{DB}$. Since each $Q(u)$ consists of t group elements from G, $Q(u)$ is represented by $k \times t$ bits. Thus, Query(i) consists of $k \times t \times l$ bits.

Step 2 Receiving Query(i), $\mathcal{DB}$ constructs child databases recursively from the original

database X. We regard X as the $t^{l-1} \times t$ binary matrix

$$D(0,\lambda) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{t-1} \\ x_t & x_{t+1} & x_{t+2} & \cdots & x_{2t-1} \\ & & \cdots & & \\ x_{t^l-t} & x_{t^l-t+1} & \cdots & \cdots & x_{t^l-1} \end{pmatrix},$$

where λ denotes the empty sequence in $\{0,1,2,\ldots,k-1\}^*$. We note that the target bit x_{i-1} is the (α_1,β_1) entry of $D(0,\lambda)$ (α_1 and β_1 are obtained in (3.3.2)). Denote it by Target($D(0,\lambda)$).

We recursively define child databases $D(u,s)$, where $1 \leq u \leq l$ and s belongs to $\{0,1,2,\ldots,k-1\}^u$. Suppose that we have defined the databases $D(u,s)$ and their target bits Target($D(u,s)$) and s in $\{0,1,2,\ldots,k-1\}^u$ for $0 \leq u < l-1$. Then we define the databases $D(u+1,s0)$, $D(u+1,s1)$, ..., $D(u+1,s(k-1))$.

The database $D(u,s)$ is a binary sequence of length t^{l-u}. We regard $D(u,s)$ as a $t^{l-u-1} \times t$ binary matrix. Suppose that

$$D(u,s) = \begin{pmatrix} y_0 & y_1 & y_2 & \cdots & y_{t-1} \\ y_t & y_{t+1} & y_{t+2} & \cdots & y_{2t-1} \\ & & \cdots & & \\ y_{t^{l-u}-t} & y_{t^{l-u}-t+1} & \cdots & \cdots & y_{t^{l-u}-1} \end{pmatrix}.$$

We now construct k child databases, $D(u+1,s0)$, $D(u+1,s1)$, ..., $D(u+1,s(k-1))$.

Recall that $Q(u)$ consists of t group elements $g_{(u,0)}, g_{(u,1)}, \ldots, g_{(u,t-1)}$ in G (defined in (3.3.3)). We define a group element g_v for each row $v = 0,1,2,\ldots,t^{l-u-1}-1$ as follows. We set

$$f_{(v,w)} = \begin{cases} g_{(u,w)} & \text{if} \quad D(u,s)(v,w) = 1 \\ 1 & \text{if} \quad D(u,s)(v,w) = 0 \end{cases}, \tag{3.3.4}$$

where $D(u,s)(v,w)$ denotes the (v,w) entry of $D(u,s)$. Then we set

$$f_{D(u,s),v} = \prod_{w=0,1,2,\ldots,t-1} f_{(v,w)} \tag{3.3.5}$$

for each row $v = 0,1,2,\ldots,t^{l-u-1}-1$. Note that the group element $f_{D(u,s),v}$ ($0 \leq v \leq t^{l-u-1}-1$) is of size k, and that $f_{D(u,s),v}$ belongs to H if and only if $D(u,s)(v,\beta_u) = 0$ by Proposition 2.1. The rth child database $D(u+1,sr)$ ($0 \leq r \leq k-1$) is defined to be the sequence consisting of $g_0(r), g_1(r), \ldots, g_{t^{l-u-1}-1}(r)$, where $g_v(r)$ denotes the rth bit of the representation of $f_{D(u,s),v}$. Hence, we have the following matrix equation:

$$\begin{pmatrix} f_{D(u,s),0} \\ f_{D(u,s),1} \\ \cdots \\ f_{D(u,s),t^{l-u-1}-1} \end{pmatrix} = \begin{pmatrix} D(u+1,s0) & D(u+1,s1) & \cdots & D(u+1,s(k-1)) \end{pmatrix} \tag{3.3.6}$$

where each $f_{D(u,s),v}$ is a row vector and each $D(u+1,sr)$ is a column vector. Thus, $D(u+1,sr)$ is a binary sequence of length t^{l-u-1}. We regard it as a $t^{l-u-2} \times t$ binary matrix. Then the target bit for it (denoted by Target($D(u+1,sr)$)) is defined to be the $(\alpha_{u+1},\beta_{u+1})$ entry of $D(u+1,sr)$ for every r in $\{0,1,\ldots,k-1\}$ (α_{u+1} and β_{u+1} are obtained in (3.3.2)).

Step 3 In the last stage of constructing child databases, $\mathcal{DB}$ obtains k^{t-1} databases $D(l-1,s)$ (s lies in $\{1,2,\ldots,k\}^{t-1}$). Note that each $D(l-1,s)$ contains t bits. We regard $D(l-1,s)$ as a $1\times t$ matrix. For each $D(l-1,s)$, we define a group element $A(s)$ as follows. First, we define

$$f_{(0,w)} = \begin{cases} g_{(u,w)} & \text{if} \quad D(l-1,s)(0,w)=1 \\ 1 & \text{if} \quad D(l-1,s)(0,w)=0 \end{cases}.$$

Then, we set

$$f_{D(l-1,s),0} = \prod_{w=0,1,2,\ldots,t-1} f_{(0,w)} = A(s).$$

The group element $A(s)$ is of size k for every s in $\{0,1,2,\ldots,k-1\}^{t-1}$. Then the group elements $A(s)$ (s lies in $\{0,1,\ldots,k-1\}^{t-1}$) form the answer Answer(Query(i)) to the query Query(i), and $\mathcal{DB}$ sends Answer(Query(i)) to $\mathcal{U}$.

Step 4 $\mathcal{U}$ receives Answer(Query(i)) consisting of $A(s)$, where s belongs to $\{0,1,\ldots,k-1\}^{t-1}$. $\mathcal{U}$ can retrieve the target bit $x_i = \mathrm{Target}(D_{(0,\lambda)})$ in polynomial time in k,n. In fact, the following holds in general.

Theorem 3.1 *For every database $D_{(u,s)}$, where $0\le u\le l-2$ and s in $\{1,2,\ldots,k\}^u$, $\mathcal{U}$ can compute* $\mathrm{Target}(D_{(u,s)})$ *in polynomial time in n,k if* $\mathrm{Target}(D_{(u+1,s0)})$, $\mathrm{Target}(D_{(u+1,s1)})$, $\ldots$, $\mathrm{Target}(D_{(u+1,s(k-1))})$ *are given.*

Proof. Suppose that we have the information

$$\mathrm{Target}(D_{(u+1,s0)}),\ \mathrm{Target}(D_{(u+1,s1)}),\ \ldots,\ \mathrm{Target}(D_{(u+1,s(k-1))})\ .$$

Recall that $\mathcal{U}$ knows the trapdoor for the subgroup membership problem of the subgroup H and the secret information that $g_{(u,\beta_u)}$ lies in $S=G\setminus H$ and

$$g_{(u,0)},\ g_{(u,1)},\ \cdots,\ g_{(u,\beta_u-1)},\ g_{(u,\beta_u+1)},\ \cdots,\ g_{(u,t-1)}\ \in\ H,$$

where

$$Q(u) = (g_{(u,0)}, g_{(u,1)}, \ldots, g_{(u,\beta_u-1)}, g_{(u,\beta_u)}, g_{(u,\beta_u+1)}, \ldots, g_{(u,t-1)}).$$

Note that the number β_u is a private information for $\mathcal{U}$. Recall that $\mathrm{Target}(D_{(u,s)})$ is the (α_u,β_u) entry of the database $D_{(u,s)}$. By the computation of $\mathcal{DB}$ in (3.3.5), we have

$$f_{D(u,s),\beta_u} = \prod_{w=0,1,2,\ldots,t-1} f_{(\beta_u,w)}.$$

By Proposition 2.1 and (3.3.4), $f_{D(u,s),\beta_u}$ belongs to H if and only if (α_u,β_u) entry is 0. Moreover, $f_{D(u,s),\alpha_u}$ is the α_uth row of the matrix

$$\begin{pmatrix} D(u+1,s0) & D(u+1,s1) & D(u+1,s2) & \cdots & D(u+1,s(k-1)) \end{pmatrix}$$

by (3.3.6). Note that α_uth bit in the database $D(u+1,sr)$ is the $(\alpha_{u+1},\beta_{u+1})$ entry of the matrix $D(u+1,sr)$ for every $r=0,1,\ldots,k-1$. On the other hand, the $(\alpha_{u+1},\beta_{u+1})$ entry of $D(u+1,sr)$ is $\mathrm{Target}(D_{(u+1,sr)})$. Since $\mathcal{U}$ knows $\mathrm{Target}(D_{(u+1,s0)})$, $\mathrm{Target}(D_{(u+1,s1)})$, $\ldots$, $\mathrm{Target}(D_{(u+1,s(k-1))})$, he can retrieve $f_{D(u,s),\alpha_u}$. After retrieving $f_{D(u,s),\alpha_u}$, $\mathcal{U}$ checks whether or not $f_{D(u,s),\alpha_u}$ is in H. Therefore, $\mathcal{U}$ can retrieve $\mathrm{Target}(D_{(u,s)})$ in polynomial time. □

3.4 Privacy In the proposed scheme, the query Query(i) consists of $Q(1)$, $Q(2)$, ..., $Q(l)$, and each $Q(u)$ consists of

$$(g_{(u,0)},\ g_{(u,1)},\ \cdots,\ g_{(u,\beta_u-1)},\ g_{(u,\beta_u)},\ g_{(u,\beta_u+1)},\ \cdots,\ g_{(u,t-1)})\ ,$$

where one of the components is chosen uniformly and randomly from $S = G \setminus H$ and the others are chosen uniformly and randomly from H. The privacy is assured by the inequality

$$|\mathbf{Prob}(C_k(\mathrm{Query}(i)) = 1) - \mathbf{Prob}(C_k(\mathrm{Query}(j)) = 1)| < \sigma\ ,$$

where $\sigma = \frac{1}{(\mathrm{Max}(k,n))^c}$, given in (3.3.1). This is exactly the general pattern indistinguishability assumption in (2.2.4) if n is bounded by a polynomial in k. Hence, the privacy of the proposed scheme is guaranteed by the subgroup membership assumption by Theorem 2.2.

3.5 Communication Complexity In the first step, $\mathcal{U}$ sends

$$\mathrm{Query}(i) = (Q(1), Q(2), \ldots, Q(l)).$$

Each $Q(u)$ consists of t group elements in G. Since every element in G is represented by a binary sequence of length k, the total bits sent in this stage is $l \times t \times k$. In the second step, $\mathcal{DB}$ sends Answer(Query(i)) consisting of k^{l-1} group elements in G. Therefore, the total bits sent in this stage is $k^{l-1} \times k = k^l$. Consequently, the communication complexity is $ltk + k^l = ln^{\frac{1}{l}}k + k^l$. Suppose that $k = n^c$ and $l = O(\frac{\log n}{\log k})$. Then we have $l = \sqrt{\frac{\log n}{\log k}}$, and $k^l = (2^{\log k})^l = 2^{l \log k} = 2^{\sqrt{\log n \log k}} = 2^{\sqrt{\log n^c \log n}} = n^{\sqrt{c}}$. On the other hand, we have $ltk + k^l = k^l(lk + 1) < k^l k^l = (k^l)^2$. Hence, we have $ltk + k^l = (n^{\sqrt{c}})^2$. It follows that the communication complexity is $O(n^c)$.

3.6 Small Example For good understanding of the scheme, we illustrate with a small example. Suppose that the database is given by $X = x_0x_1x_2x_3x_4x_5x_6x_7x_8 = 110010101$. The size of the database is $9 = 3^2$ in this example. Let $t = 3$. The X is identified with the $t \times t$ matrix

$$D(0,\lambda) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Suppose that the user $\mathcal{U}$ wants to read x_7. He computes 3-adic expansion of 7 as in (3.3.2). Then we have $7 = 2 \times 3 + 1$, $2 = 0 \times 3 + 2$. Hence, we have $\alpha_0 = 7$, $\alpha_1 = 2$, $\alpha_2 = 0$, $\beta_1 = 1$, $\beta_2 = 2$. Then $\mathcal{U}$ chooses uniformly and randomly 3 group elements $g_{(0,0)}, g_{(0,1)}, g_{(0,2)}$, where $g_{(0,0)}$ and $g_{(0,2)}$ belong to H and $g_{(0,1)}$ belongs to $S = G \setminus H$ since $\beta_1 = 1$. Next, $\mathcal{U}$ chooses uniformly and randomly 3 group elements $g_{(1,0)}, g_{(1,1)}, g_{(1,2)}$, where $g_{(1,0)}$ and $g_{(1,1)}$ belong to H and $g_{(1,2)}$ belongs to $S = G \setminus H$ since $\beta_2 = 2$. The query Query(7) consists of $Q(1) = (g_{(0,0)}, g_{(0,1)}, g_{(0,2)})$ and $Q(2) = (g_{(1,0)}, g_{(1,1)}, g_{(1,2)})$. It is sent to $\mathcal{DB}$ by $\mathcal{U}$. Let us assume that every element of G is represented by a binary sequence of length 4. $\mathcal{DB}$ receives Query(7) and then performs the following computation. Using (3.3.4), he sets

$$f_{(0,0)} = g_{(0,0)}, f_{(0,1)} = g_{(0,1)}, f_{(0,2)} = 1, f_{(1,0)} = 1,$$

$$f_{(1,1)} = g_{(2,1)}, f_{(1,2)} = 1, f_{(2,0)} = g_{(2,0)}, f_{(2,1)} = 1, f_{(2,2)} = g_{(2,2)}$$

corresponding to the database. Then, using (3.3.5), he computes

$$f_{D(0,\lambda),0} = f_{(0,0)}f_{(0,1)}f_{(0,2)} = g_{(0,0)}g_{(0,1)},$$

$$f_{D(0,\lambda),1} = f_{(1,0)}f_{(1,1)}f_{(1,2)} = g_{(0,1)},$$

$$f_{D(0,\lambda),2} = f_{(2,0)} f_{(2,1)} f_{(2,2)} = g_{(0,0)} g_{(0,2)}.$$

Suppose that $f_{D(0,\lambda),0}, f_{D(0,\lambda),1}, f_{D(0,\lambda),2}$ are represented by $0110, 1010, 1101$, respectively. It is helpful to see it in the matrix form as follows.

$$\begin{pmatrix} f_{D(0,\lambda),0} \\ f_{D(0,\lambda),1} \\ f_{D(0,\lambda),2} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

$\mathcal{DB}$ constructs four child databases $D_{1,0}, D_{1,1}, D_{1,2}, D_{1,3}$, where

$$D(1,0) = (011)^T, D(1,1) = (101)^T, D(1,2) = (110)^T, D(1,3) = (001)^T.$$

Note that we have

$$\begin{pmatrix} f_{D(0,\lambda),0} \\ f_{D(0,\lambda),1} \\ f_{D(0,\lambda),2} \end{pmatrix} = \begin{pmatrix} D(1,0) & D(1,1) & D(1,2) & \cdots & D(1,3) \end{pmatrix}.$$

For each database, using $Q(2) = (g_{(1,0)}, g_{(1,1)}, g_{(1,2)})$, $\mathcal{DB}$ compute a group element. For $D(1,0) = (011)^T$, he computes $A(0) = g_{(1,1)} g_{(1,2)}$. For $D(1,1) = (101)^T$, he computes $A(1) = g_{(1,0)} g_{(1,2)}$. For $D(1,2) = (110)^T$, he computes $A(2) = g_{(1,0)} g_{(1,1)}$. For $D(1,3) = (001)^T$, he computes $A(3) = g_{(1,2)}$, and sends $(A(0), A(1), A(2), A(3))$ as Answer(Query(7)). Receiving Answer(Query(7)), $\mathcal{U}$ checks the memberships of $A(0)$, $A(1)$, $A(2)$ and $A(3)$ in H. Since $\mathcal{U}$ keeps the trapdoor for the subgroup membership problem for H, he can check the memberships of these elements in polynomial time. He finds that $A(0), A(1), A(3) \in H$ and $A(2) \in S$ and concludes that $f_{D(0,\lambda),2} = 1101$. Checking the membership of $f_{D(0,\lambda),2}$ in H, he finds that $x_7 = 0$.

References

[1] A.Ambainis, Upper Bound on the Communication Complexity of Private Information Retrieval, Automata, Languages and Programming. Lecture Notes in Computer Science, Vol. 1256. Springer-Verlag, (1997) 401–407

[2] C.Cachin, S.Micali and M.Stadler, Computationally Private Information Retrieval with Polylogarithmic Communication, Advances in Cryptology. Lecture Notes in Computer Science, Vol. 1592. Springer-Verlag, (1999) 402–414

[3] B.Chor, O.Goldreich, E.Kushilevitz and M.Sudan, Private Information Retrieval, IEEE Symposium on Foundations of Computer Science. (1995) 41–50

[4] B.Chor and N.Gilboa, Computationally Private Information Retrieval ACM Symposium on Theory of Computing. (1997) 304–313

[5] R.Cramer and V.Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, Advances in Cryptology. *Proc. of Crypto'98*, Lecture Notes in Computer Science, Vol. 1462. Springer-Verlag, (1998) 13–25

[6] W.Diffie and M.E.Hellman, New directions in cryptography, IEEE Transactions on Information Theory, **22** (1976) 644-654

[7] T.ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory, **31** (1985) 469–472

[8] S.Goldwasser and S.Micali, Probabilistic Encryption, J. Computer and System Science **28** (1984) 270–299

[9] E.Kushilevitz and R.Ostrovsky, Replication Is not Needed: Single Database, Computationally-private Information Retrieval, IEEE Symposium on Foundations of Computer Science. (1997) 364–373

[10] K.Kurosawa and S.Tsujii, A General Method to Construct Public Key Residue Cryptosystems, Transactions of the IEICE **E-73**, (1990) 1068–1072

[11] D.Naccache and J.Stern, A New Public-key Cryptosystem, Advances in Cryptology. Lecture Notes in Computer Science, Vol. 1233. Springer-Verlag, (1997) 27–36

[12] R.Ostrofsky and V.Shoup, Private Information Storage, ACM Symposium on Theory of Computing. (1997) 294–303

[13] T.Okamoto,T and S.Uchiyama, A New Public-key Cryptosystem as Secure as Factoring, Advances in Cryptology. Lecture Notes in Computer Science, Vol. 1403. Springer-Verlag, (1998) 308–318

[14] P.Paillier, Public-key Cryptosystems Based on Composite Degree Residuosity Classes, Advances in Cryptology. Lecture Notes in Computer Science, Vol. 1592. Springer-Verlag, (1999) 223–238

[15] R.L.Rivest, A.Shamir and L.Adlemma, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, **21** (1978) 120-126

[16] T.Saito, T.Koshiba and A.Yamamura, The Decision Diffie-Hellman assumption and the Quadratic Residuosity Assumption, IEICE Transactions on Fundamentals of Electronics (1) **E84-A**, (2001) 165–171

[17] A.Yamamura and T.Saito, Private Information Retrieval Based on the Subgroup Membership Problem, Information Security and Privacy, Lecture Notes in Computer Science, Vol. 2119. Springer-Verlag, (2001) 206–220

* Communications Research Laboratory,
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

** NTT Laboratories,
1-1, Hikarinooka, Yokosuka, Kanagawa 239-0847, Japan