Augmented Oblivious Polynomial Evaluation Protocol and Its Applications

Huafei Zhu and Feng Bao

Department of Information Security, Institute for Infocomm Research,
A-Star, Singapore 119613
{huafei, baofeng}@i2r.a-star.edu.sg

Abstract. In this paper, we first introduce a new notion called augmented oblivious polynomial evaluation (AOPE), a useful notion to deal with the general oblivious polynomial evaluation protocol. And then we propose a novel implementation of our AOPE protocol. Finally we show that our construction is provably secure within our model. The potential areas of application of this protocol are numerous (two-party computation, bidding protocol, keyword search problem, and so on...).

Keywords: Augmented oblivious polynomial evaluation, homomorphic commitment scheme, homomorphic public key encryption.

1 Introduction

Oblivious polynomial evaluation (OPE) first introduced by Naor and Pinkas [12], is a protocol involving two parties, a sender whose input is a polynomial $f(x) \in F[x]$ of degree m ($m \geq 1$), and a receiver whose input is a value $a \in F$, where F is a finite field. At the end of execution of the protocol the receiver learns $f(a)$ and the sender learns nothing. There are two constructions proposed in [12]. The first construction is based on a conjecture that given a randomly chosen input to the polynomial list reconstruction problem, the value of the polynomial at $x = 0$ is pseudo-random. The second construction is more efficient but based on more stronger security assumption that the value of the polynomial at $x = 0$ is pseudo-random even given some additional hints about the location of the values of the polynomial. Unfortunately, their constructions were shown to be weaker than expected in [2]. Thus it is still an unsolved problem to implement secure and efficient OPE protocol.

1.1 This Work

Let F=Z/rZ, and f be $f(x) = a_0 + a_1x + \cdots + a_mx^m \bmod r$. We provide two observations in order to construct the proposed OPE scheme:

- a general OPE protocol for a finite field F where f has general order can be obtained from OPE protocol for prime field where the order of f is one.

S. De Capitani di Vimercati et al. (Eds.): ESORICS 2005, LNCS 3679, pp. 222–230, 2005.

– an OPE protocol for a finite field F where the order of f is one can be obtained from an OPE protocol over an integer ring where the order of f is one. An OPE protocol for an integer ring where the order of f is one can be obtained when a trusted initializer distributes a public key of RSA system.

We now explain the first observation (the main contribution of this paper). Suppose Alice has a polynomial f at hand. She first randomly selects $a_{j0} \in F$, $a_{j1} \in F$ such that $a_{j,0} + a_{j,1} = a_j + r$ (computed over integer domain Z), for $j = 1, \cdots, m-1$. Here, $f(x) = a_0 + a_1 x + \cdots + a_m x^m \bmod r = a_0 + (a_{1,0} + a_{1,1})x + \cdots + (a_{m-1,0} + a_{m-1,1})x^{m-1} + a_m x^m \bmod r$. $f(x)$ can be further written as $a_0 + a_{1,0}x + (a_{1,1} + a_{2,0}x)x + \cdots + (a_{m-1,1} + a_m x)x^{m-1}$. Bob who holds the secret input α perform OPE protocol, obtains $(a_{j,1} + a_{j+1,0}\alpha)$ for all $1 \leq j \leq m-1$ together with $(a_0 + a_{1,0}\alpha)$, and thus $f(\alpha)$. Notice that the finite field F must be a prime field, so that $a_{j,1} + a_{j+1,0}\alpha$ can be uniformly distributed among F. Throughout the following discussion, we assume that r is a prime number.

We further explain the second observation. Suppose Bob has an integer $0 \neq \alpha < r$, and Alice has $f(x) = ax + b \bmod r$. Bob generates public key and secret keys of Paillier's cryptographic system and sends $E_B(\alpha)$ and public key E_B to Alice. Alice then selects long enough integers u, v satisfying $u = a \bmod r$, and $v = b \bmod r$, and sends $E_B(\alpha)^u \ E_B(1)^v = E_B(u\alpha + v)$ to Bob. Alice further proves to Bob that all performance are correct against the common reference string − a public key of commitment scheme (RSA modulus) provided by the trusted initializer. Bob performs decryption and obtains $u\alpha + v \bmod r = a\alpha + b$ if the proof is correct.

The security definition of our protocol should be viewed as an enhanced version of Naor and Pinkas' definition [12]. That is, we allow an adversary to corrupt one of participant (either P or V could be malicious within our 2-party client/server model). The proof of security is standard. That is the rewinding of the malicious party is allowed within our model since a malicious party is not allowed to communicate with distinguisher D, i.e., the distinguisher D only gets to see the transcript to protocol execution which is significant difference from the argument of the universally composable property [3]. We also remark that the rewinding of a malicious party is strictly forbidden in Canetti's model [3], however, we do not deal with the security of our AOPE protocol in Canetti's model throughout the paper.

In summary, the main contribution of this paper are following. We introduce an interesting yet useful notion called augmented oblivious polynomial function evaluation (AOPE) in this paper. And we also propose an efficient implementation of the AOPE protocol and show that our construction is provably secure in the common reference string model assuming that a static adversary corrupts P or V.

1.2 Road Map

The rest of paper is organized as follows: in Section 2 building blocks which will be used for constructing of AOPE protocol are sketched; In section 3.1 syntax and

security of AOPE protocol are proposed and in Section 3.2 our implementation and security proof of the implementation are presented. In Section 4, we provide a novel application of our AOPE protocol to realize greater gate protocol. And we conclude our works in Section 5.

2 Building Blocks

We briefly review the following building blocks that will be used throughout the paper.

2.1 Paillier's Public Key Encryption Scheme

Paillier investigated a novel computational problem, called Composite Residuosity Class Problem, and its applications to public key cryptography in [14]. Our construction will heavily rely on this probabilistic encryption scheme which is sketched below.

The public key is a k_1-bit RSA modulus $n = pq$, where p, q are two large safe primes. The plain-text space is Z_n and the cipher-text space is $Z^*_{n^2}$. To encrypt $\alpha \in Z_n$, one chooses $r \in Z^*_n$ uniformly at random and computes the cipher-text as $E_{PK}(a, r) = g^a r^n \bmod n^2$, where $g = (1+n)$ has order n in $Z^*_{n^2}$. The private key is (p, q).

The encryption function is homomorphic, i.e., $E_{PK}(a_1, r_1) \times E_{PK}(a_2, r_2) \bmod n^2 = E_{PK}(a_1 + a_2 \bmod n, r_1 \times r_2 \bmod n)$.

Another interesting result of Paillier's public key encryption scheme is that it can be viewed as a commitment scheme as well since given a cipher-text $c{:=}g^a r^n \bmod n^2$, we first compute $a \in Z_n$ from the following equation $\frac{L(c^\lambda \bmod n^2)}{L((1+n)^\lambda \bmod n^2)} \bmod n$ and then compute r from the equation $r \in Z^*_n {=} c'^{n^{-1} \bmod \lambda} \bmod n$, where $\lambda = lcm(p-1, q-1)$, $n = pq$.

2.2 Fujisaki-Okamoto Commitment Scheme

Let s be a security parameter. The public key is a k_2-bit RSA modulus, where $\underline{P}$, $\underline{Q}$ are two large safe primes. We assume that neither P nor V knows factorization $\underline{N}$. Let g_1 be a generator of QR_N and g_2 be an element of large order of the group generated by g_1 such that both discrete logarithm of g_1 in base g_2 and the discrete logarithm of g_2 in base g_1 are unknown by P and V.

We denote $C(a, r_a) = g_1^a g_2^{r_a} \bmod N$ a commitment to x in base (g_1, g_2), where r_a is randomly selected over $\{0, 2^s N\}$. This commitment scheme first appeared in [10] and reconsidered by Damgård and Fujisaki [7] is statistically secure commitment scheme, i.e.:

-P is unable to commit itself to two values a_1, a_2 such that $a_1 \neq a_2$ in Z by the same commitment unless P can factor N or solves the discrete logarithm of g_1 in base g_2 or the the discrete logarithm of g_2 in base g_1.

-$C(a, r_a)$ statistically reveals no information to V, i.e., there is a simulator which outputs simulated commitment to a which are statistically indistinguishable from true ones.

Notice that this commitment is homomorphic, i.e, $C(a+b, r_a+r_b) = C(a, r_a) \times C(b, r_b)$. This property is useful when P wants to prove that the committed value $a \in [x, y]$. Also notice that Paillier's encryption scheme is only homomorphic with respect to addition RSA modulus. If we want the output of oblivious polynomial evaluation protocol defined over Z (so that V can verify the correctness of its received messages), then k_1 should be chosen large enough compared with the sizes of inputs of participants.

2.3 Proof of Knowledge of Encryptions

Given a cipher-text Enc(x) which is computed from Paillier's encryption scheme, the prover should provide a proof that he knows x and x lies in a given interval I specified in the protocol. There is efficient protocol presented by Damgård and Jurik already in [6]. The basic idea is the following: given Enc(x), the prover provides a commitment $C(x, r_x)$ which is computed from Fujisaki-Okamoto commitment scheme, proves that the commitment contains the same number as the encryption, and then uses Baudot's protocol [1] to prove that $m \in I$. More precisely,

-Let T be the maximum bit length of x. The prover chooses at random u, an integer of length $T + 2k$, where k is a security parameter. He sends a=Enc(u), b=$C(u)$ to the verifier;

-The verifier chooses a l-bit challenge e;

-The prover opens the encryption $a(\text{Enc}(x)^e)$ mod N^2 and the commitment $bC(x)^e$ mod N, to reveal in both cases the number $z = u + ex$. The verifier checks the opening were correct.

The protocol can be made non-interactive in the standard way using a hash function and the Fiat-Shamir paradigm. It is also statistically zero-knowledge in the random oracle mode.

3 AOPE Protocol

3.1 Syntax and Security Definition

An augmented OPE protocol consists of three participants: system initiator I, P and V. The auxiliary information of individual participant is first generated as follows:

- on input k_2, I outputs an instance of Fujisaki and Okamoto's commitment scheme. The public key PK_c of the commitment scheme is called common reference string which will be used by each participant in the system;
- on input m, r and k_2, P outputs a polynomial $f(x)$ of degree m defined over Z/rZ, together with a commitment $C(f) := ((C(a_0 + r), \cdots, C(a_m + r))$ of $f(x)$; Notice that the commitment of $f(x)$ is defined as $(C(a_0 + r), \cdots, C(a_m + r))$ but not $(C(a_0), \cdots, C(a_m))$;
- on input k_1, V outputs an instance of Paillier's encryption scheme for V. The public key/ secret key pair is denoted by (PK_v, SK_v);

To define the functionality of AOPE protocol, an imaginary TTP is introduced. And then TTP involves in the following performance:

- P runs with TTP and proves to TTP that a chosen polynomial $f(x)$ is correct; Otherwise, a random $f(x)$ defined over Z/rZ is assigned;
- V runs with TTP and proves to TTP that a chosen value α is correct; otherwise, a random α is assigned by TTP from Z/rZ;
- The output of P is a null string while the output of V is $f(\alpha)$;

Definition 1: An AOPE protocol is said secure for a sender P if for any malicious chooser V, there exists a simulator sim_V that plays the role of V in the ideal world such that for any polynomial time distinguisher D, the view of V in real conversation is computationally indistinguishable from that simulated by sim_V.

Definition 2: An AOPE protocol is said secure for a chooser V if for any malicious prover P, there exists a simulator sim_P that plays the role of P in the ideal world such that for any polynomial time distinguisher D, the view of P in real conversation is computationally indistinguishable from that simulated by sim_P.

Definition 3: We say that an AOPE protocol is secure for any static and probabilistic polynomial time (PPT) adversary if it is secure for both the sender and the chooser.

3.2 The Construction

The following implementation of AOPE consists of two phases: initializer setup and oblivious polynomial evaluation.

- In initial reprocessing phase: a sender P chooses a polynomial $f(x) = a_0 + a_1x + \cdots + a_mx^m \bmod r$ at hand over a finite field Z/rZ, where r is a prime number, $0 \neq a_j \in Z/rZ$. P then commits the chosen polynomial $f(x)$ using the common reference string − the public key PK_c of a specified Fujisaki and Okamoto's commitment scheme (or its improved version by Damgård and Fujisaki [7]), allowing P or V to commit to an integer a of sufficiently long size and prove efficiently in zero-knowledge that a belongs to some interval using the technique of Baudot [1] (thus, our model is within the common reference string model). At the end of processing phase, P publishes commitments $C(f)$ of the chosen polynomial $f(x)$.
- Inputs of participants: the input of P is the polynomial $f(x)$, and the correspondent commitment $C(f)$: $=(C(a_0 + r), \cdots, C(a_m + r))$ of polynomial $f(x)$. The input of V is the commitment $C(f)$ and a value $x \in Z/rZ$.
- Oblivious polynomial evaluation phase: in this phase, we assume that V has available a homomorphic public key encryption scheme E (e.g., Paillier's public key encryption scheme [14]). Then P and V involve the following steps:
  - V sends the public key PK_v to P, and also sends the encryption $E(\alpha)$ together with the proof that α is chosen in a correct interval by means

of Baudot's protocol [1], where $\alpha \in Z/rZ$ is chosen secretly by V which can be viewed as an non-zero value within the interval $\{0,1\}^{\log(r)}$, $E(\alpha)$: $=E_{PK_v}(\alpha)$;

- Upon receiving $E(\alpha)$ and the correspondent proof that $E(\alpha)$ is the encryption of a correct value, P verifies the correctness of proof. If the proof is correct, P proves to V that P knows that the commitment is correct (each a_j lies in a correct interval), the knowledge of de-commitment of $C(f)$;
- P then randomly selects $a_{j,0} \in Z$, $a_{j,1} \in Z$ such that $a_{j,0} + a_{j,1} = a_j + r$, for $j = 1, \cdots, m-1$ (all computations are defined over the integer domain Z). $f(x)$ can now be written as $a_0 + a_{1,0}x + (a_{1,1} + a_{2,0}x)x + \cdots + (a_{m-1,1} + a_m x)x^{m-1}$: $=(l_0(x), \cdots, l_{m-1}(x)) \bullet (1, x, \cdots, x^{m-1})$, where $l_0(x)$: $=a_0 + a_{1,0}x$, $l_j(x)$:$= (a_{j,1} + a_{j+1,0}x)$ and $l_{m-1}(x)$: $=(a_{m-1,1} + a_m x)$; Finally, P recommits $f(x)$ as following: $C(l_0)= (C(a_0, r_0), C(a_{1,0}, r_{1,0})$; $C(l_j)= (C(a_{j,1}, r_{j,1}), C(a_{j+1,0}, r_{j+1,0}))$; $C(l_{m-1})= (C(a_{m-1,1}, r_{m-1,1}), C(a_m, r_m))$.
- P then sends $(C(l_0), \cdots, C(l_{m-1}))$ to Bob, together with a sequence of proof such that $C(a_{j,0}) \times C(a_{j,1}) = C(a_j)$ by means of proof of the equality of two commitments;
- Given $E(\alpha)$ and for each $l_j(x)$, P further computes β_j: $=E(1)^{a_{j,1}} \times E(\alpha)^{a_{j+1,0}}$, together with a proof that: (1) P knows the decomposition of β_j correspondent to the bases of $E(1)$ and $E(\alpha)$; and (2) each exponent equals to correspondent commitment of $C(l_j)$. Notice that the correctness of $E(1)$ can be verified by V since V has the secret key of E; That is given the Fujisaki-Okamoto's commitments: $C(a)$ and $C(b)$ and the Paillier encryption $E(x)$ one can directly prove that $E(ax + b) = E(x)^a\ E(1)^b$, i.e., one combines the proof of knowledge of the values committed to $C(a)$ and $C(b)$, and the proof that one knows a and b such that $E(ax + b) = E(x)^a\ E(1)^b$. There are tons of examples in the literature where this is done (see for instance, the works already done by Chaum and Pedersen [4], and by Camenisch and Shoup [5]).
- Once V received the correct value of β_j, it obtains the exact value of $f(x)$ thereafter.

This ends the description of our implementation.

3.3 The Proof of Security

In this section, we are able to show the following interesting statements:

Lemma 1: for each malicious V, there exists a simulator sim_V that plays the role of V in the ideal process for carrying out the functionality of AOPE protocol such that for any polynomial time distinguisher D, the view of V in real conversation is computationally indistinguishable from that simulated by sim_V.

Proof: sim_V first generates system parameters − the public key of the underlying commitment scheme and the correspondent secret key. The public key of the commitment scheme is defined as a common reference string while the secret

key of commitment scheme will be used as trapdoor information which is known only by sim_V. Notice that the action of V in our implementation is to generate an encryption $E(\alpha)$ together with the proof that α is chosen in a correct interval by means of Baudot's protocol [1]; Thus, sim_V simply rewinds the malicious V to obtain the correct α with over-whelming probability; Once sim_V has α, it forwards α to TTP. And TTP replies sim_V with $f(\alpha)$. Now sim_V further rewinds the random tape of V so that sim_V obtains the malicious $V's$ private random string that is used in the real implementation and thus simulation of transcripts can be generated as that in the real world protocol.

Lemma 2: for each malicious prover P, there exists a simulator sim_P that plays the role of P in the ideal process for carrying out the functionality of AOPE protocol such that for any polynomial time distinguisher D, the view of P in real conversation is computationally indistinguishable from that simulated by sim_P.

Proof: In this time sim_P first generates system parameters as the real protocol described above. The public key of the underlying commitment scheme is (N, g_1, g_2), the secret key (trapdoor of the commitment scheme) is (P, Q, w), where $N = PQ$, $g_2 = g_1^w$, $g_1 \in QR_N$ is a common reference string. Then by applying the standard rewinding technique, sim_P can extract a_i and $a_{i,0}$ and $a_{i,1}$ such that $a_i = a_{i,0} + a_{i,1} \bmod r$ from its proof. Once a_i, $a_{i,0}$ and $a_{i,1}$ are all known, sim_P forward these values to TTP. TTP replies sim_P a value α. The rest simulation of the protocol is then trivial since P proves no knowledge to V further except for the equation that $\beta = f(\alpha)$. Since there is a simulator sim_1 for proving that β_0 equals $a_0 + a_{1,0}\alpha$, and sim_j for proving that β_j equals $a_{j-1,0} + a_{j,1}\alpha$ if the sub-protocols are constructed from the idea used by Chaum and Pedersen [4], and by Camenisch and Shoup [5] already. It follows that sim_P can be defined as the concatenation of $sim_1 \mid\mid \cdots \mid\mid sim_m$. It is easy to see that the view generated by sim_P is computationally indistinguishable from that generated by P in the real world protocol.

Combining Lemma 1 and Lemma 2, we have the main statement below.

Theorem: The AOPE protocol is provably secure if the underlying Fujisaki-Okamoto's commitment scheme is informational hiding and computational biding as well Paillier's encryption scheme is semantically secure in the common reference string model.

4 Applications

There are two main types of applications in which OPE protocol is useful: first when it is required to enable the receiver to obliviously obtain a value from a m-wise independent space. The second application is when it is desired to preserve anonymity in cryptographic protocols which require a user to get a value of a polynomial held by the sender without revealing the choice of the sender. Thus our AOPE protocol can be applied to the scenario of keyword search protocol [11] and [8] and the privacy preserving auctions [13] as well as private matching

and set intersection [9]. We now further suggest the following implementation of the greater gate protocol by applying our AOPE protocol:

Greater gate protocol can be abstracted as the following problem: on input two commitments $C(a)$ and $C(b)$ (Alice holds a privately while Bob holds b privately). The output of is $>$ or $\leq$. This protocol is useful for real world applications. We now apply our AOPE protocol to deal with this problem as a supportive example to demonstrate the power of our AOPE protocol.

- On input a, Alice chooses two non-zero random strings s_a, r_a from $\{0,1\}^k$ and provides the commitment $C(a)$ of a and the commitments $C(s_a)$ for s_a and $C(r_a)$ for r_a using Fujisaki-Okamoto's commitment scheme. Similarly, on input b, Bob chooses two non-zero random strings s_b, r_b from $\{0,1\}^k$ and provides the commitment $C(b)$ of b and the commitments of s_b and r_b using the same Fujisaki-Okamoto's commitment scheme.
- Alice and Bob then involve the processing of the following computation: δ: $=(a-b)(r_a+r_b)$. Notice that δ can be rewritten as $\delta = (ar_a+s_a) + (ar_b+s_b)$ $-(r_ab + s_a)$ - $(br_b + s_b)$ (in the rest of paper, all computations are defined over the integer domain);
  - Alice computes α_a: $=(ar_a + s_a)$ while Bob computes α_b:= $(br_b + s_b)$ locally.
  - on input $C(a)$, $C(r_b)$ and $C(s_b)$, Alice and Bob run AOPE protocol together so that Alice obtains β_a: $=(ar_b + s_b)$ while Bob knows nothing;
  - on input $C(r_a)$, $C(s_a)$ and $C(b)$, Alice and Bob run AOPE protocol together so that Bob obtains β_b:= $(r_ab + s_a)$ while Alice knows nothing;
- Once given β_a, Alice can compute $\gamma_a = \alpha_a + \beta_a$; And at the same time Bob can compute $\gamma_b = \alpha_b + \beta_b$;
- Alice sends γ_a to Bob while Bob sends γ_b to Alice;

The rest work of Alice is to show Bob that γ_a is computed from $\alpha_a + \beta_a$ while Bob's task is to show Alice that γ_b is computed from $\alpha_b + \beta_b$. To convince Bob γ_a is computed from $\alpha_a + \beta_a$, Alice processes the following protocol with Bob. That is,

- Alice computes $E_A(\alpha)$ and proves to Bob that both the encryption $E_A(a)$ and the commitment $C(a)$ hide the same non-zero value, where E_A stands for Alice's encryption scheme (it is specified by an instance of Paillier's encryption with sufficiently long public key N_A) by means of Damgård and Jurik approach;
- Bob then sends $E_A(1)$ to Alice; Since Alice has private key correspondent to E_A, it follows that the correctness of Bob's encryption can be verified.
- Alice then proves to Bob that $E_A(\alpha_a) = E_A(a)^{r_a}\ E_A(1)^{s_a}$ by means of the standard technique used already by Chaum and Pedersen [4], and by Camenisch and Shoup [5];
- Since Alice and Bob can run AOPE protocol together so that Alice obtains $E_A(\beta_a)$, thus Alice can compute the encryption of $E_A(\gamma_a)$: $=E_A(\alpha_a)\ E_A(\beta_a)$. Alice further proves to Bob that γ_a and t_a are exact decryption of $E_A(\gamma_a)$, where t_a is a random string used to generate the cipher-text $E_A(\gamma_a)$. Notice

that this is possible since Paillier's encryption can be viewed as alternative commitment scheme (both message and a random string can be extracted from the cipher-text with the help of private key) due to the observation stated in Section 2.

5 Conclusion

In this paper, a new notion called augmented oblivious polynomial evaluation is introduced and formalized and then a novel yet efficient implementation of the primitive is proposed which has been proved secure with our model assuming that the underlying Fujisaki-Okamoto commitment scheme is unconditional hiding and computational binding, together with Paillier's encryption is semantic secure. Like its sibling notion, our protocol has numerous applications as well.

References

1. Fabrice Boudot: Efficient Proofs that a Committed Number Lies in an Interval. Proc. of EUROCRYPT 2000: 431-444, Springer Verlag.
2. Daniel Bleichenbacher1 and Phong Q. Nguyen. Noisy Polynomial Interpolation and Noisy Chinese Remaindering. EUROCRYPT 2000, LNCS 1807, pp. 53-69, 2000.
3. R. Canetti: Universally Composable Security: A New Paradigm for Cryptographic Protocols. FOCS 2001: 136-145
4. David Chaum, Torben P. Pedersen: Wallet Databases with Observers. CRYPTO 1992: 89-105.
5. Jan Camenisch, Victor Shoup: Practical Verifiable Encryption and Decryption of Discrete Logarithms. CRYPTO 2003: 126-144
6. Ivan Damgård, Mads Jurik: Client/Server Tradeoffs for Online Elections. Proc. of Public Key Cryptography 2002: 125-140. Springer Verlag.
7. Ivan Damgård, Eiichiro Fujisaki: A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order. Proc. of ASIACRYPT 2002: 125-142, Springer Verlag.
8. Michael J. Freedman1, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword Search and Oblivious Pseudorandom Functions, 2nd Theory of Cryptography Conference (TCC'05).
9. Michael J. Freedman, Kobbi Nissim, Benny Pinkas: Efficient Private Matching and Set Intersection. EUROCRYPT 2004: 1-19
10. E. Fujisaki, T. Okamoto. Statistically zero knowledge protocols to prove modular polynomial relations. Crypto'97. 16-30, 1997.
11. Wakaha Ogata and Kaoru Kurosawa, Oblivious keyword search. Journal of Complexity, Vol.20, pp.356-371, 2004.
12. Moni Naor, Benny Pinkas: Oblivious Transfer and Polynomial Evaluation. STOC 1999: 245-254
13. Moni Naor, Benny Pinkas, Reuban Sumner: Privacy preserving auctions and mechanism design. ACM Conference on Electronic Commerce 1999: 129-139.
14. Pascal Paillier: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Proc. of EUROCRYPT 1999: 223-238, Springer Verlag.