Multiplicative Homomorphic E-Auction with Formally Provable Security

Kun Peng, Matt Henricksen

Multiplicative Homomorphic E-Auction With Formally Provable Security

Kun Peng and Matt Henricksen

Institute for Infocomm Research, Singapore
Email: dr.kun.peng@gmail.com

Abstract. A new method, homomorphic e-auction based on multiplicative homomorphic encryption algorithm like ElGamal encryption is proposed in this paper. Its advantage is obvious and useful in practice: efficient distributed private key generation and thus efficient trust sharing. A long existing problem in homomorphic e-auction, inefficiency of bid validity check, is solved in the new multiplicative homomorphic e-auction scheme in this paper, which employs efficient bid re-formatting to enforce bid validity. Another contribution of the new multiplicative homomorphic e-auction scheme is that it is the first e-auction scheme to provide formal and comprehensive security analysis to achieve formally provable security (especially privacy).

1 Introduction

E-auction is a popular e-commerce application to distribute resources. In e-auction applications, the bids are often sealed for fairness and security. In many sealed-bid e-auction schemes, it is desired to protect privacy of the losing bids, which is called bid privacy. An obvious solution to protect bid privacy in e-auction is secure multiparty computation (called secure evaluation in [30]) as e-auction can be regarded as computation (evaluation) of some secret inputs (the bids) to obtain an output (the auction result). Secure-multiparty-computation-based solution to e-auction includes a few schemes [24, 17, 16, 9, 6, 20]. As analysed in [30], these schemes are not efficient as they employ general multiparty computation techniques designed to evaluate any function. In comparison, special techniques designed to handle e-auction only are usually more efficient. A very popular such method is homomorphic bid opening. With this mechanism, each bidder employs a homomorphic encryption algorithm or a homomorphic secret sharing algorithm to seal their bids, while the auctioneers exploit homomorphism of the encryption algorithm or secret sharing algorithm to open the bids collectively instead of separately so that no losing bid is revealed. As the power of secret reconstruction or decryption is shared by multiple auctioneers such that the number of cooperating auctioneers must be over a threshold to gain the power, bid privacy is retained if the number of malicious auctioneers is not over the threshold.

Well known homomorphic e-auction schemes include [18, 19, 1, 26, 33, 31, 30, 29]. They usually require that each bidder includes a bidding choice for every

biddable price in his bid where every bidding choice must be one of two appointed integers, representing YES and NO respectively. To test whether there is a bid at a price, usually homomorphic bid opening schemes sum up all the bidders' bidding choices at that price. No separate bid choice is revealed and the sum is enough to show whether there is a bid at that price. Together with binary search for the wining price among the biddable prices, this summing-up bid opening mechanism is very efficient in finding the winning bid.

In the beginning, homomorphic e-auction schemes [18, 19, 33, 30] employ Shamir's secret sharing [39] to seal the bids and exploit its homomorphism to implement homomorphic bid opening. As secret sharing and reconstruction have to be repeated multiple times in secret-sharing-based homomorphic e-auction, it is not efficient enough, especially when public verifiability is required and validity of secret sharing needs to be publicly verified. So homomorphic encryption algorithm becomes more popular and gradually replaces secret sharing in homomorphic e-auction [1, 26, 31, 29] as a bid sealing tool.

The advantage of homomorphic encryption algorithm is obvious in homomorphic e-auction. Secret sharing only needs to be performed once to share the private key among the auctioneers. However, there is a difficulty in using homomorphic encryption in e-auction applications: the difficulty in threshold private key sharing for homomorphic algorithm. Most homomorphic e-auction schemes employ an additive homomorphic algorithm to seal the bids, where $D(c_1) + D(c_2) = D(c_1 c_2)$ for any ciphertexts c_1, c_2 and $D()$ denotes the decryption function. It is interesting to note that all the known additive homomorphic algorithms (e.g. [25, 23, 27]) employ factorization problem as a trapdoor. Although there exist some distributed key generation mechanisms for RSA [4, 22, 11], which is also factorization based, they (especially [4, 22]) are inefficient. The distributed key generation technique in [11] improves efficiency to some extent by loosening the requirements on the parameters and using additional security assumptions, but is still inefficient compared to distributed key generation of DL based encryption algorithms [12, 28, 14] like ElGamal. So they cannot provide an efficient solution to distributed key generation for additive homomorphic encryption, not to mention their difficulty in public verification and that the relatively more efficient mechanism among them may not satisfy the parameter requirements with reasonable security assumptions when applied to e-auction. It is easy for a central trusted dealer to generate the private key [13, 2, 10] and then distribute it among the auctioneers. However, it requires too strong a trust and compromises the advantage of threshold trust, so is impractical in applications like e-auction. Although a modified ElGamal encryption in [21, 35] is additive homomorphic and support efficient distributed key generation [12, 28, 14], it is not practical in most cases as it does not support efficient decryption.

Peng *et al* [30] design a special homomorphic e-auction scheme based on Goldwasser-Micali encryption, which is not additive homomorphic. As encryption and decryption operations are very efficient with Goldwasser-Micali encryption, their auction scheme needs few exponentiations with long exponents in computation. However, Goldwasser-Micali encryption depends on hardness of

factorization problem as well, so suffers from lack of efficient distributed key generation as well. Moreover, as the message space of Goldwasser-Micali encryption is only one bit long, bid opening must be repeated multiple times at any price in [30], which leads to two drawbacks in efficiency. Firstly, communicational cost is high. Secondly, a large number of multiplications are needed in computation.

A common efficiency bottleneck in homomorphic e-auction is bid validity check. Validity of homomorphic bid opening depends on two assumptions about validity of the bids. Firstly, each biding choice must represent either YES or NO. An attack called BBC attack is proposed in [31] to compromise correctness of auction when this condition is not satisfied. Secondly, each bidder chooses YES for all the biddable prices no higher than his offer and chooses NO for all the biddable prices higher than his offer. An attack called challenge attack is shown to work when this condition is not satisfied in Section 4. So validity of bids should be publicly proved and verified in secure homomorphic e-auction. Homomorphic e-auction schemes without bid validity check [18, 19, 33, 30, 31] are vulnerable to various attacks. For example, as explained in [30], colluding bidders in [18, 19, 33] can launch BBC attack. In [30, 31], a malicious bidder can launch challenge attack as described in Section 4. Those attacks threatens fairness of the auction, so must be prevented. Some other homomorphic e-auction schemes [1, 26] employ zero knowledge proof to prove validity of bids, but their proof is inefficient.

In [32, 35], batch zero knowledge proof is employed to improve efficiency of bid validity check. Another batch proof technique is proposed in [8], which can be applied to bid validity check. However, their improvement in efficiency is not great enough to ease the efficiency concern in bid validity check. The most recent attempt to improve efficiency of bid validity check is [29], which allows the auctioneers to efficiently verify validity of the bids. However, it is not universally verifiable and one instance of proof can only convince one verifier. When there are other verifiers the bidders must provide a different proof to each of them. Moreover, it is required in [29] to extend the digital signature algorithm in [3] to distributed signature generation, where the power of signature generation is shared by the auctioneers. However, there is no known method to distribute signature generation for the digital signature algorithm in [3] as it is a special signature scheme to prevent malleability.

In this paper, a new homomorphic e-auction scheme is designed to overcome the two main drawbacks in the existing homomorphic e-auction schemes: lack of efficient distributed key generation and inefficiency of bid validity check. Firstly, a multiplicative homomorphic encryption algorithm, ElGamal encryption, is employed to seal the bids such that distributed key generation of the encryption algorithm can be efficiently implemented. Secondly, simpler operations are employed to replace bid validity proof and verification such that correctness of auction can be guaranteed even at the presence of malicious bidders. The new homomorphic e-auction scheme is designed in two steps. In the first step, efficient distributed key generation and multiplicative homomorphic bid opening are specified in Section 3 such that homomorphic bid opening at any price is always correct with an overwhelmingly large probability. In the second step,

the e-auction scheme is optimised in Section 4 to prevent the known attacks. A special contribution of this paper is that it presents the first formal and comprehensive security analysis for e-auction, while security (especially comprehensive privacy) of the existing e-auction schemes is intuitive only. Especially, privacy of the whole new e-auction scheme including all the published information is comprehensively and formally proved using a simulation-based model.

2 Symbols and Security Model

The most important security properties in e-auction are as follows.

- Correctness: the auction result is determined strictly according to the auction rule, while no bid is ignored or tampered with.
- Robustness: in abnormal situations (e.g. at presence of invalid bid), the auction can still run properly.
- Privacy: no secret information (e.g. the losing bid) except for the auction-result is revealed. More precisely, the auction transcript including all the published information in the auction can be simulated by a party without any secret knowledge but the auction result such that the simulating transcript is indistinguishable from the real auction transcript.
- Universal (public) verifiability: any one can publicly verify that no participant deviates from the auction protocol.

As the bids must be sealed to achieve bid privacy, privacy of e-auction at least depends on security of the employed sealing function (e.g. encryption algorithm) and no stronger privacy (e.g. unconditional privacy) is possible. We will formally illustrates that privacy of our new e-auction scheme only depends on semantic security of the employed encryption algorithm and a threshold trust in sharing the private key. The following symbols and parameters are used in this paper.

- There are m auctioneers $A_1, A_2, \ldots, A_m$ and n bidders $B_1, B_2, \ldots, B_n$.
- There are L biddable prices and they are denoted as $P_1, P_2, \ldots, P_L$ in descending order.
- p and q are large primes such that $p - 1 = 2q$. G is the cyclic subgroup in Z_p^* with order q and g is a generator of G.
- Integer l smaller than m is the trust threshold such that cooperation of at least l auctioneers is necessary to open any bid.
- L_1 and L_2 are security parameters smaller than q. They do not need to be very large on the condition that $2^{(L-1)L_2} < q$ and 2^{-L_1}, 2^{-L_2} are negligible.
- $H()$ is a one-way and collision-resistent hash function.

The formal proof to illustrate privacy of the new e-auction scheme in this paper needs to extend the traditional definition of semantic security [15] in Definition 1 to a new property called semantic security regarding multiple encryptions, which is defined in Definition 2 and illustrated to be a deduction of traditional semantic security in Theorem 1.

Definition 1 *(Traditional semantic security) An encryption algorithm with a random probabilistic encryption function $E()$ is semantically secure in the traditional definition if no polynomial adversary can win the following game with a probability non-negligibly larger than 0.5.*

1. *The adversary chooses two different messages m_0 and m_1 in the message space in any way he likes and sends them to a probabilistic encryption oracle.*
2. *The encryption oracle randomly chooses a bit I and returns $c = E(m_I)$.*
3. *Receiving c, the adversary wins the game if it finds out I.*

Definition 2 *A random probabilistic encryption function is semantically secure regarding multiple encryptions if no polynomial adversary can win the following game with a probability non-negligibly larger than 0.5.*

1. *Messages $m_1, m_2, \ldots, m_N$ with any possible distribution in the message space are given where $N \geq 1$.*
2. *Ciphertext $c_{0,1}, c_{0,2}, \ldots, c_{0,N}$ and $c_{1,1}, c_{1,2}, \ldots, c_{1,N}$ are given such that i is randomly chosen from $\{0, 1\}$ and $c_{i,j} = E(m_j)$ for $j = 1, 2, \ldots, N$ and every $c_{1-i,j}$ may be encryption of any message in the message space.*
3. *The adversary wins the game if it finds out i.*

Theorem 1. *If an encryption algorithm is semantically secure in the traditional definition, then it is semantically secure regarding multiple encryptions.*

Proof: If an encryption algorithm is not semantically secure regarding multiple encryptions, there is a polynomial algorithm A to win the game in Definition 2 with a probability non-negligibly larger than 0.5. A polynomial adversary can use A to break traditional semantic security of the encryption algorithm as follows.

1. Choosing m_0, m_1 and given c the adversary in Definition 1 needs to find I.
2. The adversary randomly chooses $m_2, m_3, \ldots, m_N$ from Z_q and generates two encryptions for each of them: $c_j = E(m_j)$ and $c'_j = E'(m_j)$ for $j = 2, 3, \ldots, N$ where $E'()$ denotes the same encryption algorithm using the same key but different probabilistic randomization from $E()$.
3. The adversary generates $c' = E(m_0)$.
4. The adversary submits messages $m_0, m_2, m_3, \ldots, m_N$ and ciphertexts $c', c'_2, c'_3, \ldots, c'_N$ and $c, c_2, c_3, \ldots, c_N$ to A.
5. A returns i and the adversary outputs $I = i$.

Note that

$$P[I = i] = P[I = i = 1] + P[I = i = 0]$$
$$= P[I = 1]P[i = 1|I = 1] + P[I = 0]P[i = 0|I = 0]$$

where $P[E]$ denotes the probability that event E happens and $P[E_1|E_2]$ denotes the probability that event E_1 happens on the condition that event E_2 happens. As A breaks semantic security regarding multiple encryptions, $P[i = 1|I = 1]$ is non-negligibly larger than 0.5 and denoted as ϵ. When $I = 0$, both

$c, c_2, c_3, \ldots, c_N$ and $c', c'_2, c'_3, \ldots, c'_N$ are encryptions of $m_0, m_2, m_3, \ldots, m_N$, so A outputs 0 or 1 with the same chance and thus $P[i = 0|I = 0] = 0.5$.

So

$$P[I = i] = 0.5\epsilon + 0.5 \times 0.5,$$

which is non-negligibly larger than 0.5. So the adversary can break traditional semantic security of the encryption algorithm in polynomial time by querying A. Since breaking semantic security regarding multiple encryptions implies breaking traditional semantic security, traditional semantic security implies semantic security regarding multiple encryptions. □

3 The Basic Protocol, Efficient Distributed Key Generation and Multiplicative Homomorphic Bid Opening

In this section, we start with a simple question: how to design homomorphic bid opening in first-bid sealed-bid auction, while the private key of the employed bid-sealing encryption algorithm must be generated in a distributed way to avoid too strong trust. Discussions of more complex questions will be given in Section 4 and Section 5. In our design, the auctioneers set up ElGamal encryption with distributed decryption for bid sealing as follows.

1. Each A_j randomly chooses x_j from Z_q and calculates $y_j = g^{x_j} \bmod p$ and publishes $H_j = H(y_j)$.
2. After $H_1, H_2, \ldots, H_m$ have been published, each A_j publishes y_j and any one can verify $H_j = H(y_j)$ for $j = 1, 2, \ldots, m$.
3. The public key of ElGamal encryption is $y = \prod_{j=1}^{m} y_j \bmod p$ and the private key $x = \sum_{j=1}^{m} x_j \bmod q$ needs to be secretly shared among the auctioneers with the trust threshold l.
4. Each A_j builds a polynomial $F_j(X) = \sum_{k=0}^{l-1} \mathbf{a}_{j,k} X^k$ where $\mathbf{a}_{j,0} = x_j$.
5. Each A_j calculates $s_{j,J} = F_j(J) \bmod q$ and secretly sends $s_{j,J}$ to every A_J for $j = 1, 2, \ldots, m$ and $J = 1, 2, , \ldots, m$.
6. Each A_J calculates his private key share $s_J = \sum_{j=1}^{m} s_{j,J} \bmod q$.

Public verification of the key distribution can be easily implemented by publishing $\alpha_{j,k} = g^{\mathbf{a}_{j,k}} \bmod p$ for $j = 1, 2, \ldots, m$ and $k = 0, 1, \ldots, l-1$ and verifying each of the secret sharing operations using the existing publicly verifiable secret sharing techniques (e.g. [5, 38]). Encryption and decryption are as follows.

- Encryption of a message M in G is $E(M) = (g^r \bmod p,\ My^r \bmod p)$ where r is randomly chosen from Z_q.
- A ciphertext $c = (a, b)$ can be decrypted by the cooperation of at least l auctioneers (denoted as $A_1, A_2, \ldots, A_l$ for simplicity of description) as follows.

1. Each auctioneers A_j calculates and publishes $\beta_j = a^{s_j} \bmod p$ for $j = 1, 2, \ldots, l$. If public verification is required, A_j publicly proves $\log_a \beta_j = \log_g \prod_{k=0}^{l-1} \theta_k^{j^k}$ using zero knowledge proof of equality of discrete logarithms [7] where $\theta_k = \prod_{j=1}^{m} \alpha_{j,k} \bmod p$.
2. The final result is $D(c) = b / \prod_{j=1}^{l} \beta_j^{u_j} \bmod p$ where $u_j = \prod_{1 \le v \le l, v \ne j} v/(v-j) \bmod q$.

Note that unlike the encryption algorithms employed in traditional homomorphic e-auction schemes ElGamal encryption is not additive homomorphic. Instead, it is multiplicative homomorphic. More precisely, in ElGamal encryption $D(c_1)D(c_2) = D(c_1c_2)$ for any ciphertexts c_1, c_2. So our homomorphic bid opening is different from the existing homomorphic bid opening mechanisms. It exploits multiplicative homomorphism instead of additive homomorphism of the employed encryption algorithm, so is called multiplicative homomorphic e-auction. One-choice-per-price strategy is employed in our design. The biddable prices are limited in a definite set and each bidder must make a choice (indicating willingness or unwillingness to pay) at every biddable price. If a bidder is willing to pay a price, he chooses an integer standing for "YES" as his choice at that price. If a bidder is unwilling to pay a price, he chooses an integer standing for "NO" as his choice at that price. The bidders seal their bidding vectors (including their choices at all the biddable prices) and publish the sealed bidding vectors. The detailed design of bid sealing is as follows.

1. Each bidder B_i chooses his evaluation P_{e_i} from $\{P_1, P_2, \ldots, P_L\}$.
2. Each B_i builds his bidding vector: $\mathbf{b}_i = (\mathbf{b}_{i,1}, \mathbf{b}_{i,2}, \ldots, \mathbf{b}_{i,L})$ where $\mathbf{b}_{i,t} = 1$ for $t < e_i$ and $\mathbf{b}_{i,t}$ is a random integer larger than 1 in G for $t \ge e_i$.
3. Each B_i seals his bidding vector in a ciphertext vector

$$c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,L}) = (E(\mathbf{b}_{i,1}), E(\mathbf{b}_{i,2}), \ldots, E(\mathbf{b}_{i,L})).$$

Note that in our new design the bidders do not need to prove validity of any of his bidding choices as a bidding choice indicating "YES" can be any integer larger than 1. Namely, any integer in the messages space of the employed encryption algorithm is a valid bidding choice (1 for "NO" and otherwise for "YES"). It is illustrated in the following bid opening procedure that homomorphic bid opening can still work with such a tolerating bid format. The bid opening procedure employs binary search, where the biddable prices form a binary tree and the searching route starts at the tree root ($P_{L/2}$) and ends at a tree leaf. On each node of the route it is tested whether there is at least one "YES" choice without revealing the choices at that price. If there is one "YES" choice at that price, the search goes into the sub-tree with higher prices. If there is no "YES" choice at that price, the search goes into the sub-tree with lower prices. At every price P_ρ on the searching route the homomorphic bid opening is as follows.

1. The auctioneers cooperate to choose random integers $T_{\rho,1}, T_{\rho,2}, \ldots, T_{\rho,n}$ in $Z_{2^{L_1}}$ as follows.

(a) Each A_j randomly chooses $T_{\rho,i,j}$ in $Z_{2^{L_1}}$ and publishes $T'_{\rho,i,j} = H(T_{\rho,i,j})$ for $i = 1, 2, \ldots, n$.
(b) After all the $T'_{\rho,i,j}$s are published, each A_j publishes $T_{\rho,i,j}$ for $i = 1, 2, \ldots, n$.
(c) $T_{\rho,i} = \sum_{j=1}^{m} T_{\rho,i,j} \bmod 2^{L_1}$ for $i = 1, 2, \ldots, n$.

2. The auctioneers calculate

$$C_\rho = (a_\rho, b_\rho) = \prod_{i=1}^{n} c_{i,\rho}^{T_{\rho,i}} = (\prod_{i=1}^{n} a_{i,\rho}^{T_{\rho,i}} \bmod p,\ \prod_{i=1}^{n} b_{i,\rho}^{T_{\rho,i}} \bmod p)$$

where $c_{i,\rho}$ is denoted as $(a_{i,\rho}, b_{i,\rho})$.

3. An enough number of auctioneers cooperate to decrypt C_ρ into $d_\rho = D(C_\rho)$.

The binary search goes on until it stops at a leaf of the binary searching tree, which is declared as the winning price. Finally, the winner opens his bid (e.g. by publishing its encryption detail) to claim winning. It is illustrated in Theorem 2 that our homomorphic bid opening mechanism can work although any integer in G is a valid bidding choice.

Theorem 2. *Homomorphic bid opening at any price in the new multiplicative homomorphic e-auction is correct. More precisely, with an overwhelmingly large probability the decryption result at a price is larger than one iff there is at least one bidding choice larger than one at the price.*

Before Theorem 2 is proved, a lemma needs to be proved first.

Lemma 1. *Suppose $b_1, b_2, \ldots, b_N$ are integers in G. If $\prod_{i=1}^{N} b_i^{T_i} = 1 \bmod p$ with a probability larger than 2^{-L_1} for random L_1-bit integers $T_1, T_2, \ldots, T_N$, then $b_i = 1 \bmod p$ for $i = 1, 2, \ldots, N$.*

Proof: Given any integer k in $\{1, 2, \ldots, N\}$, there must exist integers $T_1, T_2, \ldots, T_{k-1}, T_{k+1}, \ldots, T_N$ in $\{0, 1, \ldots, 2^{L_1} - 1\}$ and two different integers T_k and $\hat{T}_k$ in $\{0, 1, \ldots, 2^{L_1} - 1\}$ such that the following two equations are correct.

$$\prod_{i=1}^{N} b_i^{T_i} = 1 \bmod p \tag{1}$$

$$(\prod_{i=1}^{k-1} b_i^{T_i}) b_k^{\hat{T}_k} \prod_{i=k+1}^{N} b_i^{T_i} = 1 \bmod p \tag{2}$$

Otherwise, for any L_1-bit integers $T_1, T_2, \ldots, T_{k-1}, T_{k+1}, \ldots, T_N$ there is at most one L_1-bit integer T_k to satisfy equation $\prod_{i=1}^{N} b_i^{T_i} = 1 \bmod p$, which implies that equation $\prod_{i=1}^{N} b_i^{T_i} = 1 \bmod p$ is satisfied with a probability no larger than 2^{-L_1} (with at most $2^{(N-1)L_1}$ combinations among the 2^{NL_1} possible combinations of $T_1, T_2, \ldots, T_N$) and is a contradiction.

(1) divided by (2) yields

$$b_k^{T_k - \hat{T}_k} = 1 \bmod p.$$

Note that T_k and $\hat{T}_k$ are L_1-bit integers, $2^{L_1} < q$ and q is prime, so $GCD(S_k - \hat{S}_k, q) = 1$. Therefore, $b_k = 1 \bmod p$. Also note that k can be any

integer in $\{1, 2, \ldots, n\}$ and thus $b_i = 1 \bmod p$ for $i = 1, 2, \ldots, n$. □

Proof of Theorem 2: At a price P_ρ the result of multiplicative homomorphic bid opening is

$$d_\rho = D(C_\rho) = D(\prod_{i=1}^{n} c_{i,\rho}^{T_{\rho,i}}) = \prod_{i=1}^{n} D(c_{i,\rho})^{T_{\rho,i}} = \prod_{i=1}^{n} \mathbf{b}_{i,\rho}^{T_{\rho,i}} \bmod p.$$

- When all the bidding choices, $\mathbf{b}_{1,\rho}, \mathbf{b}_{2,\rho}, \ldots, \mathbf{b}_{n,\rho}$ are 1, d_ρ is always 1.
- According to Lemma 1, when any of $b_{1,\rho}, b_{2,\rho}, \ldots, b_{n,\rho}$ modulo p is larger than 1, d_ρ is larger than 1 with an overwhelmingly large probability. □

As traditional semantic security has been reduced to semantic security regarding multiple encryptions in Theorem 1, formal privacy of the new e-auction scheme can be proved in Theorem 3, which reduces distinguishability between the e-auction transcript and a simulating transcript to breaking semantic security regarding multiple encryptions as defined in Definition 2.

Theorem 3. *The new multiplicative homomorphic e-auction scheme is private and computationally reveals no bidding information other than the auction result. More precisely, the information revealed in the e-auction scheme can be simulated by a party without any knowledge of any bid but the auction result such that the simulating transcript is indistinguishable from the real transcript of the revealed information on the condition that ElGamal encryption algorithm is semantically secure regarding multiple encryptions.*

Proof: The revealed information in the new multiplicative homomorphic e-auction scheme includes:

- $c_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$;
- at each price p_ρ on the binary searching route C_ρ, d_ρ, $T_{\rho,i,j}$ and $T'_{\rho,i,j}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$ and $T_{\rho,i}$ for $i = 1, 2, \ldots, n$;
- the integers published in the the multiple instances of underlying zero knowledge proof of equality of discrete logarithms, when complete public verifiability is required to include key generation and distributed decryption.

As ZK proof of equality of discrete logarithms in [7] has been formally proved to be zero knowledge, we only need to prove that $c_{i,t}$, C_ρ, d_ρ, $T_{\rho,i}$, $T_{\rho,i,j}$, $T'_{\rho,i,j}$ for p_ρ on the searching route, $1 \le i \le n$, $1 \le t \le L$ and $1 \le j \le m$ can be simulated by a party without any secret knowledge but the auction result. A party without any secret knowledge but the auction result can simulate them as follows.

1. He finds the binary searching route according to the auction result.
2. He randomly chooses $T_{\rho,i,j}$ for p_ρ on the searching route, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$ from $Z_{2^{L_1}}$ and calculates $T'_{\rho,i,j} = H(T_{\rho,i,j})$.
3. He calculates $T_{\rho,i} = \sum_{j=1}^{m} T_{\rho,i,j} \bmod 2^{L_1}$ for p_ρ on the searching route and $i = 1, 2, \ldots, n$.
4. He randomly chooses $\mathbf{b}_{i,t}$ from G for $1 \le i \le n$ and $1 \le t \le L$.
5. He calculates $c_{i,t} = (a_{i,t}, b_{i,t}) = (g^{r'_{i,t}} \bmod p,\ \mathbf{b}_{i,t} y^{r'_{i,t}} \bmod p)$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$ where $r'_{i,t}$ is randomly chosen from Z_q.

6. He calculates $C_\rho = (\prod_{i=1}^{n} a_{i,\rho}^{T_{\rho,i}} \bmod p,\ \prod_{i=1}^{n} b_{i,\rho}^{T_{\rho,i}} \bmod p)$ for p_ρ on the searching route.
7. He calculates $d_\rho = \prod_{i=1}^{n} \mathbf{b}_{i,\rho}^{T_{\rho,i}} \bmod p$ for p_ρ on the searching route.

In this simulating transcript of the revealed information,

- each $T_{\rho,i,j}$ is uniformly distributed in $Z_{2^{L-1}}$;
- each $T_{\rho,i}$ is uniformly distributed in $Z_{2^{L-1}}$;
- each $c_{i,t}$ is distributed in G^2 in a way independent of the secret bids but depending on how $\mathbf{b}_{i,t}$ is chosen in the simulation;
- each d_ρ is uniformly distributed in G;
- each C_ρ is uniformly distributed in G^2;
- $T_{\rho,i} = \sum_{j=1}^{m} T_{\rho,i,j} \bmod 2^{L_1}$ for p_ρ on the searching route and $1 \le i \le n$;
- $T'_{\rho,i,j} = H(T_{\rho,i,j})$ for p_ρ on the searching route ,$1 \le j \le m$ and $1 \le i \le n$;
- $C_\rho = (\prod_{i=1}^{n} a_{i,\rho}^{T_{\rho,i}} \bmod p,\ \prod_{i=1}^{n} b_{i,\rho}^{T_{\rho,i}} \bmod p)$ for p_ρ on the searching route;
- $d_\rho = \prod_{i=1}^{n} \mathbf{b}_{i,\rho}^{T_{\rho,i}} \bmod p$ and $d_\rho = D(C_\rho)$ for p_ρ on the searching route.

In comparison, in the real transcript of the same information,

- each $T_{\rho,i,j}$ is uniformly distributed in $Z_{2^{L-1}}$;
- each $T_{\rho,i}$ is uniformly distributed in $Z_{2^{L-1}}$;
- each $c_{i,t}$ encrypts B_i's choice at p_t;
- each d_ρ is uniformly distributed in G;
- each C_ρ is uniformly distributed in G^2;
- $T_{\rho,i} = \sum_{j=1}^{m} T_{\rho,i,j} \bmod 2^{L_1}$ for p_ρ on the searching route and $1 \le i \le n$;
- $T'_{\rho,i,j} = H(T_{\rho,i,j})$ for p_ρ on the searching route ,$1 \le j \le m$ and $1 \le i \le n$;
- $C_\rho = (\prod_{i=1}^{n} a_{i,\rho}^{T_{\rho,i}} \bmod p,\ \prod_{i=1}^{n} b_{i,\rho}^{T_{\rho,i}} \bmod p)$ for p_ρ on the searching route;
- $d_\rho = \prod_{i=1}^{n} \mathbf{b}_{i,\rho}^{T_{\rho,i}} \bmod p$ and $d_\rho = D(C_\rho)$ for p_ρ on the searching route.

The only difference between the two transcripts lies in distribution of $c_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$. If a polynomial adversary can distinguish the two transcripts of $c_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$, it can be employed to break semantic security of ElGamal encryption regarding multiple encryptions as follows. Since the the adversary can distinguish the two transcripts of $c_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$ without any other information, given additional information $\mathbf{b}_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$, it can still distinguish the two transcripts as the additional information only makes the distinguishing work easier (or at least does not make it harder). So given messages $\mathbf{b}_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$ and two sets of ciphertexts $c_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$, one of which is encryption of $\mathbf{b}_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$, the adversary can distinguish which set of ciphertexts are in the real e-auction transcript and thus encryption of $\mathbf{b}_{i,t}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$. Namely, the adversary breaks semantic security of ElGamal encryption regarding multiple encryptions. □

According to Theorem 1 and Theorem 3, when the number of colluding malicious auctioneers is not over l, compromising privacy of the new e-auction scheme

implies breaking semantic security of ElGamal encryption, which is widely known to be hard assuming hardness of the Decisional Diffie-Hellman problem. So the new e-auction scheme is private on the condition that the DDH problem is hard.

4 The Final Protocol, Optimisation to Achieve Robustness

Although the multiplicative homomorphic bid opening mechanism in Section 3 can work at any price, achieving correctness in the auction still needs an assumption: each bidder submits a 1 as his bidding choice at any price higher than his evaluation and an integer larger than 1 as his bidding choice at any price no higher than his evaluation. More precisely, although it is not needed to assume validity of every single bidding choice in multiplicative homomorphic e-auction (as any bidding choice in the message place of the employed encryption algorithm is valid), it is still necessary to assume that in each bid vector 1s are at the higher prices and larger integers are at the lower prices. If this assumption is not satisfied, the auction scheme is still vulnerable to some attacks. For example, a malicious bidder can submit YES at higher prices and submit NO at lower prices to launch a challenge attack (mentioned in Section 1), which enables him to dispute the auction result like attacking [30, 31] as follows.

1. A malicious bidder includes in his bid vector a larger integer at price P_γ and 1s at lower prices.
2. In the binary search for winning bid, multiplicative homomorphic bid opening is performed at one price lower than P_γ and returns a decryption result 1. So the binary search goes on to lower prices and finally stops at a winning price lower than P_γ.
3. After the auction result is published, the malicious bidder can choose to challenge validity of the result by publishing his bidding choice at P_γ if he likes (e.g. if his colluding co-bidder does not win or he is not satisfied with the auction result for other reasons). His challenge is effective as he does submit YES at a price higher than the winning bid.

This attack obviously violates robustness of auction. Can the malicious bidder be denied of his winning or penalized or kicked out? It is a complex question. Note that in an open-cry auction, a bidder is usually allowed to keep silent at a lower price but bid at a higher price later. Then why is the bid invalid in sealed-bid auction while it is accepted in the bidding phase? If the challenge is acceptable, the malicious bidder compromises fairness of the auction and takes advantage of the other bidders. Even if the challenge can be clarified and rejected, the clarification needs to open the bidding choices separately and compromises privacy of the auction.

Our countermeasure to this attack is simple: before bid opening the auctioneers re-format the encrypted bids such that in each bid if there is a bidding choice larger than 1 all the other bidding choices at lower prices in the same bid vector are larger than 1 with an overwhelmingly large probability. It is described in details as follows.

1. The bidders still seal and submit their bids as in Section 3. Namely each bidder B_i builds his bidding vector: $\mathbf{b}_i = (\mathbf{b}_{i,1}, \mathbf{b}_{i,2}, \ldots, \mathbf{b}_{i,L})$ and seals it in a ciphertext vector $c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,L})$
2. The auctioneers cooperate to choose random integers $S_{i,t}$ in $Z_{2^{L_2}}$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L-1$ just like they choose $T_{i,t}$ in Section 3.
3. The auctioneers calculate

$$c'_{i,t} = {c'}_{i,t-1}^{S_{i,t-1}} c_{i,t} = ({a'}_{i,t-1}^{S_{i,t-1}} a_{i,t} \bmod p,\ {b'}_{i,t-1}^{S_{i,t-1}} b_{i,t} \bmod p)$$
$$\text{for } i = 1, 2, \ldots, n \text{ and } t = 2, 3, \ldots, L$$

where $c'_{i,1} = c_{i,1}$, $c_{i,t}$ is denoted as $(a_{i,t}, b_{i,t})$ and $c'_{i,t}$ is denoted as $(a'_{i,t}, b'_{i,t})$.

After the re-formating, the encrypted bidding choices become

$$c'_{i,t} = ((\prod_{K=1}^{t-1} a_{i,K}^{\prod_{J=K}^{t-1} S_{i,J}}) a_{i,t} \bmod p, (\prod_{K=1}^{t-1} b_{i,K}^{\prod_{J=K}^{t-1} S_{i,J}}) b_{i,t} \bmod p) \quad (3)$$
$$\text{for } i = 1, 2, \ldots, n \text{ and } t = 2, 3, \ldots, L.$$

In this new bid format, every $c'_{i,t}$ encrypts an integer larger than 1 with an overwhelmingly large probability as illustrated in Theorem 4 if any of the bidding choices at higher prices in the same bid vector is larger than 1.

Theorem 4. *If any of $D(c_{i,1}), D(c_{i,2}), \ldots, D(c_{i,t-1})$ is larger than 1 for any t in $\{2, 3, \ldots, n\}$, then $D(c'_{i,t}) > 1$ with a probability no smaller than $1 - 2^{-L_2}$.*

Before Theorem 4 is proved, a lemma is proved first.

Lemma 2. *Suppose $b_1, b_2, \ldots, b_N$ are integers in G. If $b_N \prod_{i=1}^{N-1} b_i^{S_i} = 1 \bmod p$ with a probability larger than 2^{-L_2} where $S_1, S_2, \ldots, S_{N-1}$ are random integers at least L_2 bits long and smaller than q, then $b_i = 1 \bmod p$ for $i = 1, 2, \ldots, N-1$.*

Proof: Given any integer k in $\{1, 2, \ldots, N-1\}$, there must exist one instance of integers $S_1, S_2, \ldots, S_{k-1}, S_{k+1}, \ldots, S_{N-1}$ among all their possible choices and two different instances for the choice of S_k, denoted as S_k and $\hat{S}_k$, such that the following two equations are correct.

$$b_N \prod_{i=1}^{N-1} b_i^{S_i} = 1 \bmod p \quad (4)$$
$$b_N (\prod_{i=1}^{k-1} b_i^{S_i}) b_k^{\hat{S}_k} \prod_{i=k+1}^{N-1} b_i^{S_i} = 1 \bmod p \quad (5)$$

Otherwise, for any possible choice of integers $S_1, S_2, \ldots, S_{k-1}, S_{k+1}, \ldots, S_N$ there is at most one choice for integer S_k among all its possible choices to satisfy equation $b_N \prod_{i=1}^{N-1} b_i^{S_i} = 1 \bmod p$, which implies equation $b_N \prod_{i=1}^{N-1} b_i^{S_i} = 1 \bmod p$ is satisfied with a probability no larger than 2^{-L_2} (as the number of possible choices for S_k is at least 2^{L_2}) and is a contradiction.

(4) divided by (5) yields

$$b_k^{S_k - \hat{S}_k} = 1 \bmod p.$$

Note that S_k and $\hat{S}_k$ are smaller than q and q is prime, so $GCD(S_k - \hat{S}_k, q) = 1$. Therefore, $b_k = 1 \bmod p$. Also note that k can be any integer in $\{1, 2, \ldots, N-1\}$ and thus $b_i = 1 \bmod p$ for $i = 1, 2, \ldots, N-1$. □

Proof of Theorem 4: (3) and multiplicative homomorphism of ElGamal encryption imply

$$D(c'_{i,t}) = (\prod_{K=1}^{t-1} D(c_{i,K})^{\prod_{J=K}^{t-1} S_{i,J}}) D(c_{i,t}) \bmod p$$
$$\text{for } i = 1, 2, \ldots, n \text{ and } t = 2, 3, \ldots, L.$$

Note that any $\prod_{J=K}^{t-1} S_{i,J}$ is smaller than q as $2^{(L-1)L_2} < q$. If $D(c'_{i,t}) > 1$ is not guaranteed with a probability no smaller than $1 - 2^{-L_2}$ for any t, then $D(c'_{i,t}) = 1$ with a probability larger than 2^{-L_2}. So according to Lemma 2, all of $D(c_{i,1}), D(c_{i,2}), \ldots, D(c_{i,t-1})$ are 1s, which is contradictory to the fact that at least one of $D(c_{i,1}), D(c_{i,2}), \ldots, D(c_{i,t-1})$ is larger than 1. Therefore, $D(c'_{i,t}) > 1$ must be guaranteed with a probability no smaller than $1 - 2^{-L_2}$. □

As the re-formatted encrypted bids are enforced to contain consistent bidding choices with an overwhelmingly large probability, multiplicative homomorphic bid opening and binary search can be performed on them and challenge attack can be prevented. Moreover, as the bidding choices except those at the top price in all the bids are already randomized, multiplicative homomorphic bid opening can actually be simplified and the randomization through raising the encrypted bidding choices to the power of $T_{i,t}$ in Section 3 can be removed unless multiplicative homomorphic bid opening is performed at the top price. The simplified bid opening operation is as follows.

1. The auctioneers reformat the encrypted bids as detailed earlier in this section and obtain the re-formatted encrypted bids $c'_{i,t} = (a'_{i,t}, b'_{i,t})$ for $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, L$.
2. The auctioneers perform binary search for the winning bid. If the binary search goes to the top price P_1, the multiplicative homomorphic bid opening at P_1 is the same as described in Section 3. Otherwise, the multiplicative homomorphic bid opening at any price P_ρ is as follows.
   (a) The auctioneers calculate
   $$C_\rho = \prod_{i=1}^{n} c'_{i,\rho} = (\prod_{i=1}^{n} a'_{i,\rho} \bmod p, \prod_{i=1}^{n} b'_{i,\rho} \bmod p).$$
   (b) An enough number of them cooperate to decrypt C_ρ into $d_\rho = D(C_\rho)$ as described in Section 3.
   (c) If $d_\rho > 1$, the search goes into the sub-tree with prices higher than P_ρ. If $d_\rho = 1$, the search goes into the sub-tree with prices lower than P_ρ.
3. The binary search goes on until it stops at a leaf of the binary searching tree, which is declared as the winning price. Finally, the winner opens his bid to claim winning.

Table 1. Security comparison of homomorphic e-auction schemes

Auction schemes	Bid opening power sharing	Bid validity check	Universal verifiability	Vulnerability or problem	Correctness & privacy
[18]	nL instances of secret sharing	no	yes	BBC attack and challenge attack	intuitive
[19]	nL instances of secret sharing	no	yes	BBC attack and challenge attack	intuitive
[33]	nL instances of secret sharing	no	yes	BBC attack and challenge attack	intuitive
[1]	sharing 1 key only but no efficient distributed key generation	yes	yes	no	intuitive
[26]	sharing 1 key only but no efficient distributed key generation	yes	yes	no	intuitive
[30]	sharing 1 key only but no efficient distributed key generation	no	yes	challenge attack	intuitive
[31]	nL instances of secret sharing	no	yes	challenge attack	intuitive
[29]	sharing 1 key only but no efficient distributed key generation	yes	no	unknown how to generate Boneh signature [3] in a distributed way	intuitive
New	sharing 1 key only efficient distributed key generation	enforcing validity by bid re-formatting	yes	no	formally proved

Table 2. Efficiency comparison of secure homomorphic e-auction schemes

Auction	Bidder		Auctioneer	
schemes	multiplication	example	multiplication	example
$[1, 26]$	$\geq 12291.5L$	50346140	$\geq 12292nL + (10752 + 2n)\log_2 L$	50348957633
[29]	$1536(2L + 8)$ per verifier	12595200 per verifier	$1536(0.2nL + 20\log_2 L + 16n)$	1283297280
New	$3072L$	12582912	$3L_2n(L - 1) + 4608\log_2 L$	368605296

In comparison with the original e-auction scheme in Section 3, the optimisation in this section changes the way the bidding choices are randomized. The new randomization operation not only enables multiplicative homomorphic bid opening but also enforces validity of the bids. The optimised e-auction scheme is correct and private as illustrated in Theorem 2 and Theorem 3 (since the change in randomization operation does not affect their applicability to the e-auction scheme) and its robustness is guaranteed by Theorem 4.

5 Comparison and Conclusion

Security and efficiency of the new e-auction scheme is compared with the existing e-auction schemes with bid privacy in this section. As explained in Section 1, secure-multiparty-computation-based e-auction schemes [24, 17, 16, 9, 6, 20] and e-auction schemes employing downward search [37, 40, 41, 36, 34] are less efficient and so we focus our comparison on homomorphic e-auction. Firstly, comparison of security properties is given in Table 1. Then, efficiency comparison of secure homomorphic e-auction schemes (with bid validity check and invulnerable to known attacks) is given in Table 2, where the number of multiplications is counted and for simplicity an exponentiation with a **L**-bit exponent is counted

as 1.5**L** multiplications. A full-length exponent in cryptographic operations in G or Z_p is supposed to be 1024 bits long. Note that like the analysis in the existing e-auction schemes, cost of preparation work (e.g. distributed key generation) is not included in Table 2. If it is taken into account, advantage of our new scheme will be greater as it is the only homomorphic e-auction scheme with efficient distributed key generation. In the example in Table 2, $n = 1000$ and $L = 4096$, while $L_2 = 30$ such that 2^{-L_2} is smaller than one out of one billion.

The comparisons clearly demonstrate that the new e-auction scheme is secure and efficient. It satisfies all the security properties and is the only e-auction scheme to achieve formally provable security. It is much more efficient than any secure homomorphic e-auction scheme, which already employs a relatively more efficient solution to secure e-auction. Extending our technique to more complex auction rules is an open question.

References

1. M Abe and K Suzuki. M+1-st price auction using homomorphic encryption. In *PKC '02*, pages 115–124.
2. O Baudron, P Fouque, D Pointcheval, G Poupard and J Stern. Practical multi-candidate election system. In *ACM PODC '01*, pages 274–283.
3. D Boneh and X Boyen. Short signatures without random oracles. In *Eurocrypt '04*, pages 56–73.
4. D Boneh and M Franklin. Efficient generation of shared RSA keys. In *Crypto '97*, pages 425–439.
5. F Boudot and J Traore. Efficient public verifiable secret sharing schemes with fast or delayed recovery. In *ICICS '99*, pages 87–102.
6. C Cachin. Efficient private bidding and auctions with an oblivious third party. In *ACM CCS '99*, pages 120–127.
7. D Chaum and T Pedersen. Wallet databases with observers. In *CRYPTO '92*, pages 89–105.
8. K Chida and G Yamamoto. Batch processing for proofs of partial knowledge and its applications. In *IEICE TRANS FUNDAMENTALS, JAN 2008*, pages 150–159.
9. R Cramer, I Damgård and J Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT '01*, pages 280–299.
10. I Damgård and M Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *PKC '01*, pages 119–136.
11. I Damgård and M Koprowski. Practical threshold rsa signatures without a trusted dealer. In *EUROCRYPT '01*, pages 152–165.
12. P Feldman. A practical scheme for non-interactive verifiable secret sharing. In *FOCS '87*, pages 427–437.
13. P Fouque, G Poupard and J Stern. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography 2000*, pages 90–104.
14. R Gennaro, S Jarecki, H Krawczyk and T Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT '99*, pages 123–139.
15. S Goldwasser and S Micali. Probabilistic encryption. In *Journal of Computer Security, Vol. 28, No 2, 1984*, pages 270–299.
16. M Jakobsson and A Juels. Mix and match: Secure function evaluation via ciphertexts. In *ASIACRYPT '00*, pages 143–161.

17. A Juels and M Szydlo. A two-server, sealed-bid auction protocol. In *FC '02*, pages 72–86.
18. H Kikuchi, M Harkavy and J Tygar. Multi-round anonymous auction. In *IEEE WDRES 1998*, pages 62–69.
19. H Kikuchi, S Hotta, K Abe and S Nakanishi. Distributed auction servers resolving winner and winning bid without revealing privacy of bids. In *IEEE Workshop on Next Generation Internet 2000*, pages 307–312.
20. K Kurosawa and W Ogata. Bit-slice auction circuit. In *ESORICS '02*, pages 24–38.
21. B Lee and K Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. In *JW-ISC 2000*, pages 101–108.
22. P MacKenzie Y Frankel and M Yung. Robust efficient distributed rsa-key generation. In *STOC '98*, page 320.
23. D Naccache and J Stern. A new public key cryptosystem based on higher residues. In *ACM Computer Science Conference 1998*, pages 160–174.
24. M Naor, B Pinkas and R Sumner. Privacy perserving auctions and mechanism design. In *ACM Conference on Electronic Commerce 1999*, pages 129–139.
25. T Okamoto and S Uchiyama. A new public-key encyptosystem as secure as factoring. In *CRYPTO '98*, pages 308–318.
26. K Omote and A Miyaji. A second-price sealed-bid auction with the discriminant of the p-th root. In *FC '02*, pages 57–71.
27. P Paillier. Public key cryptosystem based on composite degree residuosity classes. In *EUROCRYPT '99*, pages 223–238.
28. T Pedersen. A threshold cryptosystem without a trusted party. In *EUROCRYPT '91*, pages 522–526.
29. K Peng and F Bao. Efficiency improvement of homomorphic e-auction. In *TRUSTBUS '10*, pages 238–249.
30. K Peng, C Boyd and E Dawson. A multiplicative homomorphic sealed-bid auction based on Goldwasser-Micali encryption. In *ISC '05*, pages 374–388.
31. K Peng, C Boyd and E Dawson. Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing. In *Mycrypt '05*, pages 84–98.
32. K Peng, C Boyd and E Dawson. Batch verification of validity of bids in homomorphic e-auction. *Computer Communications 29 (2006) 2798-2805*, 2006.
33. K Peng, C Boyd, E Dawson and K Viswanathan. Robust, privacy protecting and publicly verifiable sealed-bid auction. In *ICICS '02*, pages 147–159.
34. K Peng, C Boyd, E Dawson and K Viswanathan. Non-interactive auction scheme with strong privacy. In *ICISC '02*, pages 407–420.
35. K Peng and E Dawson. Efficient Bid Validity Check in ElGamal-Based Sealed-Bid E-auction. In *ISPEC '07*, pages 209-224.
36. K Sako. An auction scheme which hides the bids of losers. In *PKC '00*, pages 422–432.
37. K Sakurai and S Miyazaki. A bulletin-board based digital auction scheme with bidding down strategy. In *IWCTE 1999*, pages 180–187.
38. B Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *CRYPTO '99*, pages 149–164.
39. A Shamir. How to share a secret. *Communication of the ACM 1979*, 22(11): Pages 612–613.
40. K Suzuki, K Kobayashi and H Morita. Efficient sealed-bid auction using hash chain. In *ICISC '00*, pages 183–191.
41. Y Watanabe and H Imai. Reducing the round complexity of a sealed-bid auction protocol with an off-line ttp. In *STOC '00*, pages 80–86.