Wu CC, Chang CC, Lin IC. New sealed-bid electronic auction with fairness, security and efficiency. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(2): 253–264 Mar. 2008

New Sealed-Bid Electronic Auction with Fairness, Security and Efficiency

Chia-Chi Wu[1] (吴家麒), Chin-Chen Chang[1,2] (张真诚), and Iuon-Chang Lin[3,*] (林詠章)

[1] *Department of Computer Science and Information Engineering, "National Chung Cheng University", Chiayi 621 Taiwan, China*

[2] *Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan, China*

[3] *Department of Management Information Systems, "National Chung Hsing University", Taichung, Taiwan, China*

E-mail: iclin@nchu.edu.tw

Received September 8, 2006; revised December 19, 2007.

Abstract Electronic sealed-bid auction schemes usually have a common drawback, the third party (auction host) can conspire with a malicious bidder to leak all bidding prices before the opening stage. It results in the malicious bidder wining the auction with an optimal bidding price. Recently, Liaw *et al.* proposed an auction protocol for electronic online bidding in which they designed a deposit deduction certification for government procurement. However, it also has above mentioned flaw. Moreover, we further found that there were some extra security drawbacks in their protocol. First, the bidder can forge a bidding receipt to claim that he/she is a valid auction winner. Second, it may suffer from the third party forging attack. Third, their protocol leaked some bidders' private information to the third party, such as the bidder's bank account number and the authorization code. Thus, it cannot protect the bidder's privacy at all. In this paper, we not only point out the drawbacks from the previous scheme but also propose a new electronic auction scheme to overcome the above mentioned drawbacks. Furthermore, the computational complexity can be decreased in our online sealed-bid auction scheme.

Keywords electronic auction, e-commerce, information security

1 Introduction

Due to Internet popularization, common consumers gradually accept electronic transactions and services. Nowadays, the Internet can provide online shopping, online bidding and many financial actions. Electronic auction has become one of the most popular activities of electronic commerce. Some famous auction websites are substantially growing such as Yahoo and eBay in recent years.

Generally, electronic auction can be divided into three transaction types: traditional English auction, Dutch auction and sealed-bid auction. Traditional English auction is a well-known auction type, in which the bidders cast public bids and the bids must be higher than all bids in the previous round. The bidders can continue to bid an upper price until nobody can offer a higher price. On the contrary, Dutch auction is a buyer's market auction such as wholesale procurement, in which the bidders (sellers) have to cast their bids in public and the bids must be lower than all bids in the previous round. In this type, the price will decrease until only one bidder is willing to provide the lowest price. In the sealed-bid auction, each bidder writes down his/her bid price and authorizes it on a sheet, and then seals the sheet and submits it to the auction host. When it is the bidding deadline, the auction host opens all the sealed sheets and determines the winner. Government procurement usually performs a hybrid scheme by sealed-bid auction and Dutch auction.

For security reasons, in the sealed-bid auction, all bids are encrypted in transmission. When it is the bidding deadline, the auction host can open all bids and decide the winner. In order to maintain fairness, all bids must be verified publicly and nobody can open the bids before the deadline. Due to electronic auction it needs to achieve security and efficiency, several electronic auction schemes have been proposed in recent years. Franklin and Reiter proposed a sealed-bid auction protocol[1], which adopts a verifiable signature to prevent malicious bidders from canceling their bids. Kudo[2] proposed a sealed-bid auction method with a

Regular Paper

[*] Corresponding Author

time server. Kikuchi, Hakavy and Tygar[3] proposed an electronic auction scheme to improve the privacy of bids such that only the auctioneer knows who the winner is.

Chang C. C. and Chang Y. F.[4] proposed three anonymous auction protocols which conform to the above-mentioned auction types to ensure that the bidders can bid arbitrarily and anonymously, they apply the deniable authentication scheme to check the validation of source bids and keep the bidders secret.

However, Jiang *et al.*[5] pointed out that Chang C. C. and Chang Y. F.'s schemes have a security weakness, in which the bidder cannot detect the tampered response message from the auctioneer. Therefore, Jiang *et al.* proposed an improved scheme to prevent tampering attacks. Subsequently, Chang C. C. and Chang Y. F. led into an alias method to propose an enhancement[6].

In the following, we summarize the requirements of electronic auction from the researches of Chang C. C. and Chang Y. F.[4], Subramanian[8], and Chen[9].

1) Anonymity: all bidders can keep anonymity in an auction, even if the bid is opened.

2) Public verifiability: all the bidding prices and the winning prices can be verified by anyone.

3) Non-repudiation: the property of non-repudiation is that both the bidder cannot deny having cast his/her bid and the third party T cannot deny that T has received the bid from the bidder.

4) Traceability: the winning bidder can be identified when the auction is finished.

5) Accountability of bidder: the auction cannot be interrupted by any malicious bidders with a dishonest bid without being detected. That is to say, the third party can verify each bid when the bidder casts a bid.

6) Unforgeability: the bidders, the auctioneers and the auction host cannot perform forgery.

7) Fairness: all sealed-bids are opened at the same time, and the third party or the auctioneer cannot collude with a malicious bidder to cheat the other bidders.

8) Privacy: a third party (the auction host) will not get to know details about the bidders' payment information, such as bank accounts, authorization codes and so on.

9) Confidentiality: each bid must keep integrity and confidential before the opening bids.

10) Low overhead cost: the transaction cost must be as low as possible.

Recently, the services of the electronic government have been flourishing, the government procurement gradually adopts electronic auction activity for the sellers or service providers, where the related procurement information of government departments can be published, for example, government procurement related regulations, procurement product name, quantity, opening deadline, and other requirements (delivery date of goods or due date of products guarantee). Generally, government procurement procedures are formal and conscientious. First, the government will establish a professional procurement authority such as a procurement bureau which takes charge of the bidding auction procedures to avoid disputes between bidders and demand departments. Second, to decrease government budget expenses, government procurement adopts invitation providers and public sealed-bid auction activity. However, a delay of delivered goods or construction will damage public benefits. Thus the government procurement must ask all bidders to pay a *deposit* before the bidding stage. When the bidding is finished, the auction host (procurement bureau) will refund *deposits* to bidders except the winners. If the winning bidder breaks the transaction contract, the *deposit* will be confiscated by the government.

Hence, how to build an efficient and secure electronic auction of government procurement is an interesting topic. Liaw *et al.*[7] proposed an electronic auction protocol with a deposit deducting certification to solve the problem of the bidder's deposit payment. However, we found their protocol has three security drawbacks. First, the bidder can forge a bidding receipt to claim that he/she is a valid auction winner. Second, it may suffer from the third party forging attacks. Third, it leaked some bidders' private information to the third party, such as the bidder's bank account number and the authorization code.

In this paper, we shall point out the security drawbacks in Liaw *et al.*'s scheme and propose a new scheme to overcome these drawbacks. Moreover, our scheme decreases the number of asymmetric key encryption/decryption computations to enhance efficiency, thus it is suitable for applying in government procurement electronic auction.

The rest of this paper is organized as follows. In Section 2, we shall briefly review Liaw *et al.*'s scheme and point out the drawbacks of their scheme. In Section 3, we present our proposed scheme. The security analysis of the proposed scheme is described in Section 4. A discussion of how the proposed scheme performs in terms of security and efficiency is provided in Section 5. Finally, a concluding remark is given in Section 6.

2 Related Work

In this section, the processes and the drawbacks of Liaw *et al.*'s auction protocol will be specified and discussed.

2.1 Review of Liaw *et al.*'s Protocol

In Liaw *et al.*'s auction protocol, the auction protocol was divided into four stages: the advertisement stage, the registration stage, the bidding stage, and the exchange of the product and the payment stage. The four stages are specified as follows.

Advertisement Stage

The auctioneer signs the auction's product information M_1 using his/her private key s_U, and then broadcasts M_1 and its signature on the Internet. M_1 provides the related information of this auction, such as the auction products combination, the auction period, a specific description of products, and their respective amounts for the bidders.

Registration Stage

If bidder B wants to join this auction, B has to register with the third party T first. Throughout the paper, B_{info} denotes *B's* personal information, P_B denotes *B's* public key, r is a random number which is chosen by the bidder B, and $E_{P_c}[X]$ represents a public key cryptosystem with a public key P_c to encrypt a plaintext X. The steps of this stage are illustrated in Fig.1, and are described as follows.

Step 1. Bidder B computes $E_{P_T}[B_{\text{info}}, P_B, r, M_1]$ and delivers them to the third party.

Step 2. The third party T can decrypt this data using its private key, and then checks the random number r.

Step 3. If the bidder's r already has been used in this auction, the third party will ask the bidder to choose a new random number. Otherwise, the third party T will publish the specific auction information M_1, $H(r), H(w), H(x), H(y)$, and $H(z)$ on the auction website, where w, x, y, z are the random numbers chosen by the third party, and $H()$ is a collision-resistance one way hash function[10,11].

Step 4. The third party T uses the bidder B's public key P_B to compute $E_{P_B}[M_1, r, x, B_{id}]$, and then sends them to bidder B.

Step 5. Bidder B can decrypt and verify r and x to authenticate the validity using $H(r)$ and $H(x)$ on *Web*.

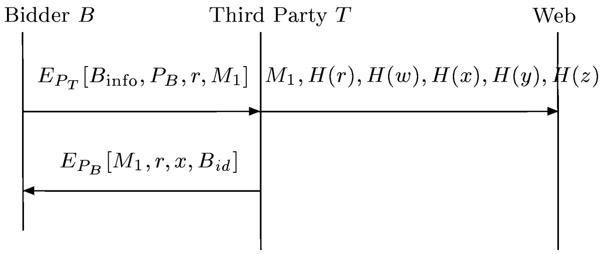
Bidder B
Third Party T
Web
$E_{P_T}[B_{\text{info}}, P_B, r, M_1]$
$M_1, H(r), H(w), H(x), H(y), H(z)$
$E_{P_B}[M_1, r, x, B_{id}]$

Fig.1. Registration stage of Liaw *et al.*'s scheme.

Bidding Stage

Fig.2 illustrates the steps of the bidding stage in Liaw *et al.*'s scheme. In the following, we explain the details of each step.

Step 1. The third party T uses bank A's public key P_A to compute $E_{P_A}[M_1, B_{id}, \textit{payment}, \textit{deposit}, y]$ and sends them to bank A.

Step 2. Bank A decrypts them, and verifies y via $H(y)$ value from *Web*. If it holds, bank A transfers the payment of *deposit* from bidder B's account to the third party's funds.

Step 3. After a successful transfer, bank A issues a deposit deducting certification $Cert_d$ which is signed by bank A. Afterward, bank A computes $E_{P_B}[M_1, B_{id}, Cert_d, y]$ and sends them to bidder B.

Step 4. Bidder B can check y to validate the message, then gets B_{id}, and $Cert_d$, after B decrypts the message.

Step 5. For bidding the auction, bidder B computes $E_{P_T}[M_1, B_{id}, Cert_d, \textit{price}, y, r]$ and sends them to the third party T. The *price* is the bidding price for this auction.

Step 6. The third party T decrypts them and checks $Cert_d$ of bidder B's, *price*, and y. If they are valid, T can validate this bidding message.

Step 7. T generates a sequence bidding number *order* for this bidding, then computes $E_{P_B}[M_1, B_{id}, \textit{order}, \textit{price}, r]$ and sends them to bidder B for proving that T has received the bidding price.

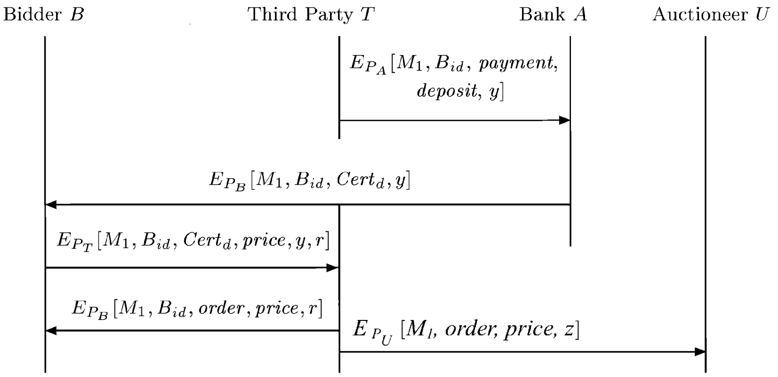
Bidder B
Third Party T
Bank A
Auctioneer U
$E_{P_A}[M_1, B_{id}, \textit{payment}, \textit{deposit}, y]$
$E_{P_B}[M_1, B_{id}, Cert_d, y]$
$E_{P_T}[M_1, B_{id}, Cert_d, \textit{price}, y, r]$
$E_{P_B}[M_1, B_{id}, \textit{order}, \textit{price}, r]$
$E_{P_U}[M_l, \textit{order}, \textit{price}, z]$

Fig.2. Bidding stage of Liaw *et al.*'s scheme.

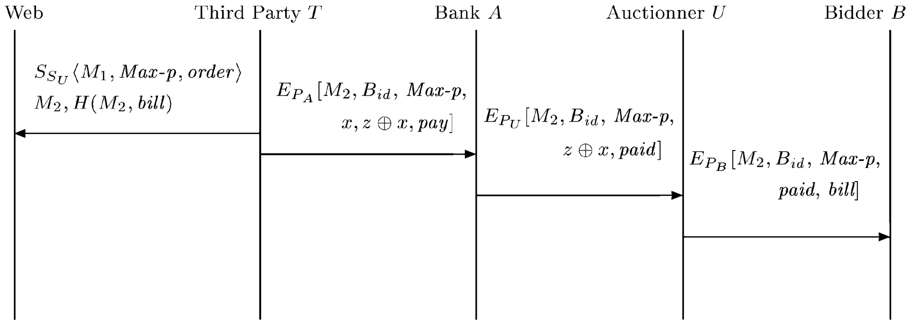
Web
Third Party T
Bank A
Auctionner U
Bidder B
$S_{S_U}\langle M_1, Max\text{-}p, order\rangle$
$M_2, H(M_2, bill)$
$E_{P_A}[M_2, B_{id}, Max\text{-}p, x, z \oplus x, pay]$
$E_{P_U}[M_2, B_{id}, Max\text{-}p, z \oplus x, paid]$
$E_{P_B}[M_2, B_{id}, Max\text{-}p, paid, bill]$

Fig.3. Exchange of the product and the payment stage of Liaw *et al.*'s scheme.

Step 8. When the bidding is completed, the third party T gets a maximum price *Max-p* and then computes $E_{P_U}[M_1$, *order*, *Max-p*, $z]$ and sends them to auctioneer U.

Exchange of the Product and the Payment Stage

The steps of the exchange of the product and payment stage are illustrated in Fig.3. We use $S_{s_c}\langle X\rangle$ to represent a digital signature with a secret key S_c to sign a plaintext X, and *bill* denotes a bill of lading for this auction. We describe the details of each step as follows.

Step 1. When the winner and the selling price *Max-p* are determined, the auctioneer U will sign$[M_1$, *Max-p*, *order*] and publish $S_{S_U}\langle M_1$, *Max-p*, *order*$\rangle$, M_2, and $H(M_2$, *bill*) on the auctioneer's website. Here *order* is the winning bidder order, and M_2 denotes a plaintext of this signature.

Step 2. The third party computes $E_{P_A}[M_2, B_{id}$, *Max-p*, x, $z \oplus x$, *pay*], then sends them to bank A. Here, *pay* is a payment request.

Step 3. Bank A decrypts the message and verifies x. If it is valid, then the bank transfers money from the bidder's account to the auctioneer's account. After transferring the money, bank A computes $E_{P_U}[M_2, B_{id}$, *Max-p*, $z \oplus x$, *paid*] and sends them to auctioneer U.

Step 4. The auctioneer U first decrypts the message, then checks the paid information. If it is correct, U computes $E_{P_B}[M_2, B_{id}$, *Max-p*, *paid*, *bill*] and sends them to the winner.

Step 5. The winner can validate *bill* using computed $H(M2$, *bill*) to compare with the published $H(M_2$, *bill*) on *Web*. Note that *bill* can exchange the auction product, after the winner is paid at the winning price.

2.2 Drawbacks of Liaw *et al.*'s Scheme

According to the above mentioned description, Liaw *et al.* claimed that their scheme is secure and efficient. However, we found out some security drawbacks in their scheme as follows.

1) All bidders face an unfair risk in this scheme, if the third party conspires with a malicious bidder to cheat other bidders, it can allow the malicious bidder to get an optimal selling price. As the third party T can get all bidding prices before the auction time is up, T can notify the malicious bidder to cast the optimal selling price before the end of bidding stage. It is unfair for other bidders.

2) In Step 7 of the bidding stage, an attacker can make a message $E_{P_B}[M_1, B'_{id}$, *order*$'$, *price*$'$, $r]$, for example *price*$'$ and *order*$'$ are equal to *Max-p* and *order* is published on the *Web*, to claim that he/she is a valid auction winner. The main reason for these drawbacks is that the bidding receipt lacks the third party T's signature.

3) To avoid the third party cheating, the auctioneer has to check the winner's bidding receipt in the exchange of the product and payment stage. Nevertheless, Liaw *et al.*'s scheme does not perform this check. In addition, all bidding prices and related sequence numbers must be published on the *Web* for bidders to verify them. Therefore, in our scheme, we bring in these checking procedures.

4) We find that the third party can know the bidder's payment data in the bidding stage, and bank A also knows the bidder's auction message M_1. Thus, they cannot protect the bidder's privacy. Generally, electronic transactions must be separated into information flow and financial flow to protect the user's privacy[15].

3 Proposed Scheme

In this section, we shall improve Liaw *et al.*'s scheme to withstand these drawbacks. In addition, since the symmetric encryption is 1000 times faster than the public key encryption, we shall decrease the public key encryption/decryption computations as much as possible to improve efficiency. We describe our scheme in the following five stages: the advertisement stage, the registration stage, the bidding stage, the opening stage, and the exchange of the product and the payment stage. In the proposed scheme, $E_k(X)$ stands for a symmetric cryptosystem with a secret key k to encrypt a plaintext X, and $D_k(Y)$ stands for the same symmetric cryptosystem with a secret key k to decrypt a ciphertext Y. Moreover, $S_{S_X}\langle Y\rangle$ stands for a digital signature of the plaintext Y using a secret key S_X. Assume that *Web* belongs to the inside network of the third party T and it also is protected by T; hence their communications are secure between T and *Web*. It allows the third party T to publish information, but others only can download information.

3.1 Advertisement Stage

In this stage, auctioneer U computes $S_{S_U}\langle M_1, H(\mathit{bill})\rangle$, and then broadcasts them and their plaintext to everyone in the auction.

3.2 Registration Stage

In this stage, bidder B registers at the third party. The steps of this stage are illustrated in Fig.4 and explained as follows.

Step 1. Bidder B chooses a nonce N_B, and a temporary key K_B, then computes $\mathit{Sign}_B = S_{S_B}\langle B_{\text{info}}, N_B, H(K_B)\rangle$, $E_{K_B}(B_{\text{info}}, N_B, K_B, M_1)$, and $E_{P_T}[K_B]$. B sends $\{\mathit{Sign}_B, E_{K_B}(B_{\text{info}}, N_B, K_B, M_1),\ E_{P_T}[K_B]\}$ to the third party T.

Step 2. The third party T can get K_B using his/her private key to decrypt $E_{P_T}[K_B]$, and then decrypts $E_{K_B}(B_{\text{info}}, N_B, K_B, M_1)$ using K_B. T checks B_{info}, N_B, $H(K_B)$, and verifies the bidder B's signature Sign_B. If they are not valid, T may ask B to retransmit the registration data. Otherwise, T will accept B's registration, and records K_B and generates an identity B_{id} for the bidder B.

Step 3. The third party T will publish the specific auction information $M_1, H(x), H(y), Y = g^k \bmod P$ and a large prime number P on *Web*, where k is a secret number, g is a primitive root of P and its order is $P-1$.

Step 4. The third party T computes $E_{K_B}(M_1, N_B + 1, x, B_{id})$ and sends them to bidder B.

Step 5. Bidder B can compute $D_{K_B}(E_{K_B}(M_1, N_B + 1, x, B_{id}))$, then checks $N_B + 1$ and compares $H(x)$ with the published $H(x)$ on *Web*. If they are equal, the message can be authenticated.

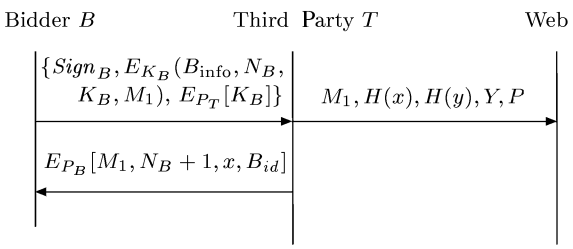
Bidder B
Third Party T
Web
$\{\mathit{Sign}_B, E_{K_B}(B_{\text{info}}, N_B, K_B, M_1),\ E_{P_T}[K_B]\}$
$M_1, H(x), H(y), Y, P$
$E_{P_B}[M_1, N_B + 1, x, B_{id}]$

Fig.4. Registration stage of the proposed scheme.

3.3 Bidding Stage

In this stage, we assume that the bidder must obtain the bank's deposit deduction certification before she/he bids the auction. To protect the bidder's privacy, the third party cannot ask the bank to perform a *deposit* payment transfer. Fig.5 illustrates the steps of this stage. We explain the steps of this stage as follows.

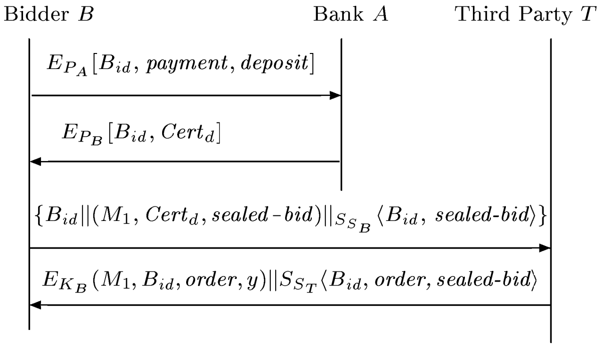
Bidder B
Bank A
Third Party T
$E_{P_A}[B_{id}, \mathit{payment}, \mathit{deposit}]$
$E_{P_B}[B_{id}, \mathit{Cert}_d]$
$\{B_{id}\|(M_1, \mathit{Cert}_d, \mathit{sealed\text{-}bid})\|S_{S_B}\langle B_{id},\ \mathit{sealed\text{-}bid}\rangle\}$
$E_{K_B}(M_1, B_{id}, \mathit{order}, y)\|S_{S_T}\langle B_{id}, \mathit{order}, \mathit{sealed\text{-}bid}\rangle$

Fig.5. Bidding stage of the proposed scheme.

Step 1. Bidder B computes $E_{P_A}[B_{id}$, *payment*, *deposit*$]$ and sends them to the bank A.

Step 2. Bank A decrypts it, and then verifies B's *payment*. If it passes, bank A will transfer a deposit payment from B's account to the third party T's account. If it is a successful transfer, bank A signs a deposit deducting certification Cert_d, then computes $E_{P_B}[B_{id}, \mathit{Cert}_d]$ and sends them to bidder B. Otherwise, bank A will ask bidder B to retransmit the essential information.

Step 3. Bidder B can decrypt $E_{P_B}[B_{id}, \mathit{Cert}_d]$ to get Cert_d. If B wants to cast a bid price, B computes *sealed-bid* $= (B_{id}\|$ *bidding price*$)\times Y^r \bmod P$ and $S_{S_B}\langle B_{id}$, *sealed-bid*$\rangle$, where r is a random number which is selected by the bidder B. Then, B computes $E_{K_B}(M_1, \mathit{Cert}_d$, *sealed-bid*$)$ and sends $\{B_{id}\|\ (M_1, \mathit{Cert}_d$, *sealed-bid*$)\|S_{S_B}\langle B_{id}$, *sealed-bid*$\rangle\}$ to T.

Step 4. The third party T checks Cert_d, and verifies $(B_{id}$, *sealed-bid*$)$ signature using B's public key P_B.

Step 5. If they are valid, T generates an order number *order* and makes a signature $S_{S_T}\langle B_{id}$, *order*, *sealed-bid*$\rangle$. Then T computes $E_{K_B}(M_1, B_{id}$, *order*, $y)$, and delivers $\{E_{K_B}(M_1, B_{id}$, *order*, $y)\|S_{S_T}\langle B_{id}$, *order*, *sealed-bid*$\rangle\}$ to bidder B.

Step 6. Bidder B computes $D_{K_B}(E_{K_B}(M_1, B_{id}$, *order*, $y))$, then verifies $H(y)$ and $S_{S_T}\langle B_{id}$, *order*, *sealed-bid*$\rangle$. If they are valid, B stores $S_{S_T}\langle B_{id}$, *order*, *sealed-bid*$\rangle$ as a bidding receipt.

3.4 Opening Stage

Since sealed-bids must be publicly opened and compared, our scheme adds the opening stage. Fig.6 illustrates the data flow of this stage, and we explain these steps as follows.

Step 1. The third party T collects all *order*'s, and the related *sealed-bid*'s to sign a signature, $S_{S_T}\langle \underset{i}{\forall}\, order_i \| \underset{i}{\forall}\, sealed\text{-}bid_i \| Time\rangle$, where *Time* is a timestamp.

Step 2. After bidder B sees the web publishing message, B verifies the signature. If it is valid, B computes $R = g^r \bmod P$, and $E_{K_B}(order, R)$, and sends $\{B_{id}\|E_{K_B}(order, R)\}$ to the third party T.

Step 3. The third party T decrypts the data, then computes each $(B_{id}\|$ *bidding price*$)$ = *sealed-bid* $\times(R^k)^{-1} \bmod P$. If B_{id} is recovered, then T gets the bidder B's bidding price. Otherwise, T sends a fault message to ask B to retransmit data. If the bidder does not deliver the related data within the deadline, then T takes the bidder to waive his/her bidding right.

Step 4. T can compare these bidding prices and get a maximum price *Max-p*. T publishes all *order*'s, the related $(R^k)^{-1} \bmod P$, the wining order *W-order* and *Max-p* on *Web*.

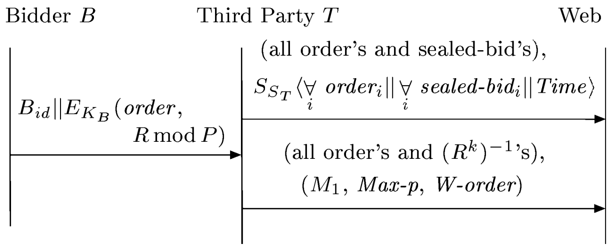
Bidder B
Third Party T
Web
$B_{id}\|E_{K_B}(order,$
$R \bmod P)$
(all order's and sealed-bid's),
$S_{S_T}\langle \underset{i}{\forall}\, order_i \| \underset{i}{\forall}\, sealed\text{-}bid_i \| Time\rangle$
(all order's and $(R^k)^{-1}$'s),
$(M_1,\ Max\text{-}p,\ W\text{-}order)$

Fig.6. Opening stage of the proposed scheme.

3.5 Exchange of the Product and the Payment Stage

We present the exchange of the product and payment stage of our scheme in Fig.7. We describe the steps as follows.

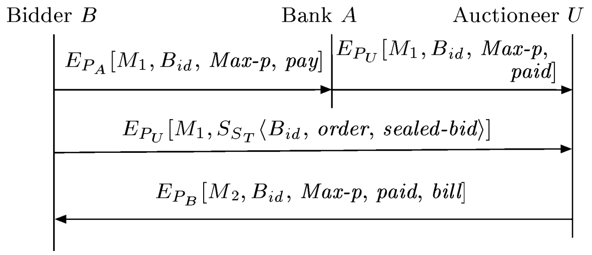
Bidder B
Bank A
Auctioneer U
$E_{P_A}[M_1, B_{id},\ Max\text{-}p,\ pay]$
$E_{P_U}[M_1, B_{id},\ Max\text{-}p,\ paid]$
$E_{P_U}[M_1, S_{S_T}\langle B_{id},\ order,\ sealed\text{-}bid\rangle]$
$E_{P_B}[M_2, B_{id},\ Max\text{-}p,\ paid,\ bill]$

Fig.7. Exchange of the product and the payment stage of our scheme.

Step 1. The winner B uses bank A's public key P_A to compute $E_{P_A}[M_1, B_{id},$ *Max-p*, *pay*] and then sends them to the bank A for paying the auction. Here *pay* is a payment request information.

Step 2. According to the decrypted message, bank A transfers money from the B's account to the auctioneer's account. After transferring, bank A computes $E_{P_U}[M_1, B_{id},$ *Max-p*, *paid*] and sends them to auctioneer U.

Step 3. The winner B computes $E_{P_U}[M_1, S_{S_T}\langle B_{id},$ *order*, *sealed-bid*$\rangle]$ and sends the message to auctioneer U.

Step 4. When auctioneer U collects Step 2 and Step 3 messages, it then performs a tripartite check for the winner's bid receipt validation, and *order* of the bid receipt and payment status are as follows.

1) U decrypts message of Step 2, and verifies the bid receipt's signature using the third party T's public key.

2) If the signature is valid, then U checks whether the *order* of Step 2 is equal to the wining *W-order*. If they are equal, U can confirm the bidder B is the winner.

3) U verifies the payment information *paid*. If it passes, auctioneer U can make sure that bidder B has paid the selling price for this auction. Otherwise, the bidder will not pass these checks; auctioneer U must reject the bidder B.

Step 5. After auctioneer U confirms the tripartite check, U computes $E_{P_B}[M_2, B_{id},$ *price*, *paid*, *bill*] and sends them to B.

Step 6. Winner B can verify *bill* using $H(bill)$ to compare with the broadcasted value in the advertisement stage.

4 Security Analysis

In the following, we first define the protocol participants, oracle states and oracle queries, respectively. Then we evaluate the security of the sealed-bid and the E-auction protocol, and construct the probability $Pr[S_0]$ from $Pr[S_1]$, $Pr[S_2]$ and $Pr[S_3]$.

Protocol Participants:

Each E-auction protocol includes one auctioneer, one third party and many bidders. An E-auction bidder may have many instances, called *oracles*, involved in distinct concurrent executions of the E-auction protocol. We denotes instance i of a bidder B as Π_B^i.

Oracle States:

An oracle has two oracle states in the E-auction protocol, they are accepting and terminating. An oracle can be accepted when it has enough information to compute the encrypted key $\theta = g^{k\cdot r_i}$ for a sealed-bid. An oracle Π_B^i can be accepted once at any time in a single execution. We denote the fact that Π_B^i accepts a encrypted key r_i by $\mathrm{ACC}(\Pi_B^i) = \mathrm{TRUE}$. Note that even if Π_B^i accepted an encrypted key r_i, it may not terminate because an oracle normally terminates when it sends/receives the last message of the E-auction protocol. Furthermore, it also terminates when receiving an invalid message or missing an excepted message. Once an oracle terminates, it will not send or receive any message. We denote the fact that an oracle Π_B^i terminated by TERM $(\mathit{\Pi}_B^i)$ = TRUE.

Oracle Queries:

Let A be an adversary attacking the E-auction protocol. The following oracle queries model the capabilities of A.

- Send$(\mathit{\Pi}_B^i, M)$: this query models A controlling all communications in the E-auction protocol. The adversary A sends a message M to an oracle $\mathit{\Pi}_B^i$, then $\mathit{\Pi}_B^i$ performs necessary computations according to the E-auction protocol, and sends back the response message. The adversary A can initiate an execution of the E-auction protocol by sending a query Send$(\mathit{\Pi}_B^i$,“*start*”) to a user oracle $\mathit{\Pi}_B^i$.

- H(M): this query allows an adversary A accessing to the maptofield oracle H, H : $\{0,1\}^\alpha \leftarrow \{0,1\}^\infty$. If the input string M has not been asked, it generates a random number μ and returns μ to A. Otherwise, it looks up the *H-table* to find the record (M, μ') and returns the number μ' to A. *H*-table is a record set used to store all previous H(M) queries.

- Test$(\mathit{\Pi}_B^i)$: this query measures the semantic security of the encrypted key r_i. Upon receiving this query, the oracle $\mathit{\Pi}_B^i$ flips an unbiased coin b. If $b = 1$, it returns the encryption key r_i to A. Otherwise, it returns a random string to A.

The Sealed-Bid Security

In an execution of the E-auction protocol, we say the adversary A breaks the sealed-bid security of the E-auction protocol if A asks a single Test query to an oracle $\mathit{\Pi}_B^i$, and correctly guesses the bit b, which is selected by $\mathit{\Pi}_B^i$ in the Test query. When A terminates, it outputs a bit b'. We say that A *wins* the game if $b = b'$. Since A can trivially win with probability 1/2, we define A's advantage by $Adv_P^{act}(A) = 2 \times \Pr[b = b'] - 1$. The E-auction protocol is E-auction-secure if $Adv_P^{act}(A)$ is negligible.

Security Proof of the E-Auction Protocol

We present a theorem to evaluate the sealed-bid security of the E-auction protocol. The theorem shows that if the discrete logarithms problem is hard, then an adversary who has the ability to perform the oracle queries (Send, H) does not have any advantage to obtain an encrypted key from the E-auction protocol. The theorem is described as follows:

Theorem 4.1. *Let Adv_P^{act} be the advantage that an adversary A breaks the sealed-bid security of the E-auction protocol within time t. Let β be the bit string size of modular P of all sealed-bids. Assume A breaks the sealed-bid security of the E-auction protocol by running q_{se} Send queries, and $q_h H$ queries. Then we have*:

$$Adv_P^{act}(t, q_{se}, q_h) \leqslant q_{se} \cdot Adv_{\hat{e}}^{\omega dh}(\omega) + q_h \cdot 2^{-\alpha}.$$

Proof. Let S_0 be the event that the adversary A breaks the sealed-bid security of the E-auction protocol. Let S_1 be the event that A breaks the encrypted key θ security of the E-auction protocol, S_2 be the event that A breaks the sealed-bid security of the E-auction protocol by forging the legitimate bidder's signature, and S_3 be the event that A breaks the H(*bill*) security of the E-auction protocol. Then, we have:

$$Pr[S_0] = Pr[S_1] + Pr[S_2] + Pr[S_3].$$

In the following, we construct the probability $Pr[S_0]$ from $Pr[S_1], Pr[S_2]$ and $Pr[S_3]$.

The Probability $Pr[S_1]$: a probabilistic algorithm ω is said to (t, ϵ)-break *CDH* in a group $G_{g,p}$ if on input (g, p, q) and (g^a, g^b) and after running in at most t steps, ω computes the Diffie-Hellman function, $DH_{g,p}(g^a, g^b) = g^{ab}$, with probability at least ϵ, where the probability is over the coins of ω and (a, b) chosen uniformly from $Z_q \times Z_q$. We say that group $G_{g,p}$ is a (t, ϵ)-*CDH* group if no algorithm (t, ϵ)-breaks *CDH* in $G_{g,p}$.

ω receives (g, P, g^k, g^r) as an input, then ω selects a random number $i \in [1, q_{se}]$. Then, ω starts running A as a subroutine and answers the oracle queries made by A. If A asks Send$(\mathit{\Pi}_A^*$ “*start*” at the i-th Send query, ω sets $m_i = \text{Sealed-bid}/(Y)^{-i} \bmod P$ and returns the first message flow of the E-auction protocol to A, where $Y = g^k \bmod P$. Later, when A asks Send$(\mathit{\Pi}_B^*, (B_{id}, m_i))$, ω checks whether message m_i includes B_{id}. If it holds, ω can correctly break $\theta = g^{ki} \bmod P$.

The probability that ω correctly guesses the moment at which A breaks the θ security is equivalent to the probability that correctly guesses the value i, denoted by δ. Thus, we have

$$\delta \geqslant \frac{1}{q_{se}}.$$

We know that the probability that ω breaks CDH assumption[16] is equivalent to the probability that A breaks the θ security multiplied by the probability that ω correctly guesses the moment at which A breaks the sealed bid security.

$$Adv_{\hat{e}}^{cdh}(\omega) = \varepsilon' = Pr[S_1] \cdot \frac{1}{q_{se}}.$$

Therefore,

$$Pr[S_1] \leqslant q_{se} \cdot Adv_{\hat{e}}^{cdh}(\omega). \tag{1}$$

Let T_r be the time to generate a random number in $GF(P)$. Let T_e be the time to perform a long exponentiation in $G_{g,P}$. In each execution of the E-auction protocol, ω will generate a random number r and compute one long exponentiation in $G_{g,P}$. Thus, the running time of ω is:

$$t' \leqslant q_{se} \cdot T_r + q_{se} \cdot T_e + t.$$

The Probability $Pr[S_2]$: in 1993, Bellare and Rogaway proposed a random oracle model[17] of signature. They defined a digital signature scheme is a triple $(\mathit{\Gamma}, \mathrm{Sign}, \mathrm{Verify})$ of polynomial algorithms. (PK, SK) are a pair of matching public and secret keys. The signature $\sigma \leftarrow \mathrm{Sign}^R(SK, m)$ is a algorithm for signing m; and the verify algorithm $\mathrm{Verify}^R(PK, m, \sigma) \in \{0, 1\}$ to check whether σ is the signature of m. It must be the case that $\mathrm{Verify}^R(PK, m, \sigma) = 1$ for all $\sigma \in [\mathrm{Sign}^R(SK, m)]$. An adversary has a polynomial-time algorithm F with access to R and a signing oracle. The output of F is a pair (m, σ) such that m was not queried of the signing oracle. They defined and proposed a formal proof about the signature scheme is secure if for every adversary F the function

$$\begin{aligned}\epsilon(k) = Pr[R \leftarrow 2^{\infty}; (PK, SK) \leftarrow \mathit{\Gamma}(1^k); (m, \sigma)\\ \leftarrow F^{R, \mathrm{Sign}^R(SK, \cdot)}(PK) : \mathrm{Verify}^R(PK, m, \sigma) = 1]\end{aligned} \tag{2}$$

is negligible, that $\mathit{\Gamma}$ is a trapdoor permutation generator. Therefore, we can directly quote from their result to infer $Pr[S_2]$ is negligible.

The Probability $Pr[S_3]$: since the one-way function H is regarded as a random oracle, we define H: $\{0, 1\}^{\alpha} \leftarrow \{0, 1\}^{\infty}$. If A knows the $H(\mathit{bill})$ corresponded to the i-th Send query, A must ask an $H(\mathit{bill}_t)$ query which is recorded in the *H-table* such that $H(\mathit{bill}) = H(\mathit{bill}_t)$. Thus, A successfully derives $\mathit{bill} = \mathit{bill}_t$ with a probability at most $q_h \cdot 2^{-\alpha}$. Therefore,

$$Pr[S_3] \leqslant q_h \cdot 2^{-\alpha}. \tag{3}$$

By (1)∼(3), we have

$$\begin{aligned}Adv_P^{act}(A) &= Pr[S_0]\\ &= Pr[S_1] + Pr[S_2] + Pr[S_3]\\ &\leqslant q_{se} \cdot Adv_{\hat{e}}^{cdh}(\omega) + q_h \cdot 2^{-\alpha}.\end{aligned}$$

□

5 Discussions

First, we shall show whether the proposed scheme satisfies the mentioned electronic auction requirements in Subsection 5.1. Then, we shall demonstrate extra security properties in Subsection 5.2. In addition, the efficiency of the proposed scheme is discussed in Subsection 5.3.

5.1 Requirements Evaluation

1) Anonymity

All bidders cast bids are encrypted with a session key according to Step 1 of the registration stage, no one can gain access to other bidder's information, except the third party T. In addition, only the third party T stores the bidder's information, therefore, it can maintain anonymity in the auction, even after the bid is opened.

2) Public Verifiability

All biding prices can be publicly verified to prevent the third party T cheating bidders and performing a conspiracy with a malicious bidder. In our scheme, all *order*'s and *sealed-bid*'s are signed before the opening stage, since the third party T signs a signature S_{S_T} ⟨*all order's and sealed-bid's, Time*⟩ and publishes them on *Web*, that can provide public verifiability. In addition, all bidding prices can be publicly recovered, because the third party T publishes each decrypted parameter $(R^k)^{-1}$ which can derive at the original bidding price using $(B_{id} \| \mathit{bidding\ price}) = \mathit{sealed\text{-}bid} \times (R^k)^{-1} \bmod P$. Therefore, we can make sure that our scheme provides a public verifiability for each bidding price.

3) Non-Repudiation

(1) The property of non-repudiation is that neither the bidder can deny having cast a bid nor the third party T can deny having gained access to the bid information. In our scheme, no bidder can deny having cast a bid, because only bidder B can use the secret key S_B to sign the bid information $S_{S_B}\langle B_{id}$, *sealed-bid*⟩. Therefore, we can make sure that any bidder cannot deny having signed the bid information.

(2) The third party T cannot deny that he/she has got the bid information from the bidder, because the third party T must sign a bid receipt $S_{S_T}\langle B_{id}$, *order*, *sealed-bid*⟩, and reply to the bidded bidder, after T has received the bid information. Only the third party T has the secret key S_T to compute $S_{S_T}\langle B_{id}$, *order*, *sealed-bid*⟩. Thus, the third party T cannot deny having received the bid information.

4) Traceability

After T compares all the recovered bidding prices and gets a maximum price *Max-p*, T then publishes the winning *W-order* and *Max-p* on the *Web*. Since each bidder has got a bid receipt, he/she can check whether the order number of his/her bid receipt is equal to *W-order*. Furthermore, because all *order*'s, *sealed-bid*'s, and decrypted parameters are published on the website which provides anyone to compare each *order* and the related bidding price, so the winner can be easily traced when the auction is complete.

5) Accountability of the Bidder

The third party T can verify each bid by checking its signature $S_{S_B}\langle B_{id}$, *sealed-bid*$\rangle$ in the bidding stage. In the following, T can recover each bidding price through $(B_{id} \parallel \textit{bidding price}) = \textit{sealed-bid} \times (R^k)^{-1} \bmod P$ in the opening stage. We can make sure that T verifies each bid to satisfy accountability.

6) Unforgeability

An attacker may try to forge a sealed-bid, a bid receipt and a bill of lading *bill*. We will show that all attacks fail in our scheme.

(1) Without bidder B's secret key S_B, no one can forge bidder B to construct the sealed-bid signature $S_{S_B}\langle B_{id}$, *sealed-bid*$\rangle$. Thus, only the bidder B can cast the sealed-bid of his own.

(2) No one can forge a valid bid receipt $S_{S_T}\langle B_{id}$, *order, sealed-bid*$\rangle$ of the bid, because only the third party T can sign it in the bidding stage. However this drawback exists in Liaw *et al.*'s scheme. In their scheme, the third party T has not signed the bid receipt, hence, the bid receipt can easily be forged.

(3) The bill of lading *bill* cannot be forged, since the auctioneer U broadcasted the message $S_{S_U}\langle M_1, H(\textit{bill})\rangle$ in the advertisement stage. Only U has the secret key S_U to generate this signature and *bill* is protected by a one-way hash function.

7) Fairness

To achieve the auction fairness, our scheme can prevent both the third party T cheating and bidders cheating. Moreover, T also cannot conspire with any bidder to get a cheating bidding price. We show that all attacks fail as follows.

(1) Since all *order*'s, the related *sealed-bid*'s, and their signature $S_{S_T}\langle$*all order's and sealed-bid's, Time*$\rangle$ are published by the third party T when the opening stage is beginning, the third party T cannot cheat bidders to alter published data. In addition, all sealed-bids can be opened by the third party T and each auction bidder, because they can compute each $(B_{id} \parallel \textit{bidding price}) = \textit{sealed-bid} \times (R^k)^{-1} \bmod P$ to recover each order's bidding price. Hence, the third party T cannot cheat the auction bidders to change the winner.

(2) If a malicious bidder sends a wrong number $R' \neq g^r \bmod P$ to make a fake bidding price which is different from the original, it will fail and be discovered by the third party T, because the original sealed-bid is generated from $\textit{sealed-bid} = (B_{id} \parallel \textit{bidding price}) \times Y^r \bmod P$.

We can derive $(B_{id} \parallel \textit{bidding price}) = \textit{sealed-bid} \times (R^k)^{-1} \bmod P$.

If the third party computes

$$\textit{sealed-bid} \times ((R')^k)^{-1} \bmod P = (B_{id} \parallel \textit{bidding price}) \times Y^r \times ((R')^k)^{-1} \bmod P,$$

then the result will not recover the valid B_{id} value. Therefore, the third party can easily find the bidder cheating.

(3) Without each bidder's decrypted parameter $(R^k)^{-1} \bmod P$, the third party T cannot open the sealed-bid, before the opening stage. In addition, all bidders will not provide their decrypted parameters, before the third party T publishes all *order*'s, the related *sealed-bid*'s, and their signature $S_{S_T}\langle$*all order's and sealed-bid's, Time*$\rangle$. T can neither get all bidding prices to derive maximum price *Max-p* in the bidding stage, nor alters published messages; therefore, T's conspiracy with any bidder to cast a cheating bidding price will fail.

8) Privacy

In our scheme, we separate the auction transaction into the data flow and financial flow such that the third party does not understand the bidder's payment information. The bidder sends $E_{P_A}[B_{id}$, *payment*, *deposit*$]$ to request the bank A to transfer a *deposit* to the third party T. Afterward, the bidder can receive a message $E_{P_B}[B_{id}, \textit{Cert}_d]$, where $\textit{Cert}_d$ is a deposit deducting certification which can be verified by the third party T in the bidding stage. Hence, the bidder's payment information will not be sent to the third party or the auctioneer to preserve the bidder's privacy.

9) Confidentiality

(1) Each bid can keep secret in the bidding stage. The attacker cannot derive the bidding price without the decrypted parameter $(R^k)^{-1} \bmod P$, since $\textit{sealed-bid} = (B_{id} \parallel \textit{bidding price}) \times Y^r \bmod P$. Its security is based on brute force search to find only Y^r in $GF(P)$. In addition, each bidder generates the bid signature $S_{S_B}\langle B_{id}$, *sealed-bid*$\rangle$, that can be verified by the third party T to keep integrity.

(2) Since T computes $\{E_{K_B}(M_1, B_{id}, \textit{order}, y) \parallel S_{S_T}\langle B_{id}, \textit{order, sealed-bid}\rangle\}$ and sends them to the bidder, each *order* is encrypted by the session key K_B and signed by the third party T's secret key S_T, that can achieve confidentiality of each *order*.

10) Low Overhead Cost

In our scheme, the bidder uses a symmetric key encryption instead of a public key encryption to decrease the bidding computation and lightens the third party's and the bidder's burden. We will analyze the efficiency in Subsection 5.3.

5.2 Extra Security Considerations

In our scheme, some extra security can successfully defend against the possible attacks. We explain them as follows.

1) Eavesdropping Attack

If attacker Alice attempts to eavesdrop on the network for her profits, she will gain nothing. Because the session K_B is encrypted and signed by the bidder, it uses the PKCS#7 standard[18] to encapsulate content.

2) Replay Attack

If attacker Alice grabbed and interrupted a valid bidder's registration information $\{Sign_B = S_{S_B}\ \langle B_{\rm info}, N_B, H(K_B)\rangle,\ \ E_{K_B}(B_{\rm info}, N_B, K_B, M_1), E_{P_T}[K_B]\}$, and then retransmits to the third party T to impersonate the bidder B for registration that will not work. Alice cannot derive B_{id} from $E_{K_B}(M_1, N_B + 1, x, B_{id})$ of the response message by T, because the session key K_B is unknown for Alice.

In the same way, Alice cannot employ a replay attack to impersonate the third party T or the auctioneer U without the session key K_B or the auctioneer's secret key S_U.

3) Impersonation Attack

(1) No one can impersonate bidder B, because the bidder signed his/her session K_B and his/her bidding price in the registration stage and the bidding stage respectively. When the bidder asks bank A to transfer money for the *deposit* or *Max-p*, only the valid bidder can provide his/her own payment information and authorization code.

(2) The auctioneer U cannot be impersonated, since only U has the secret key S_U to sign the $H(bill)$ in the broadcasted message $S_{S_U}\langle M_1, H(bill)\rangle$. Moreover, only U can issue a legal *bill* for the winner, who can compute $H(bill)$ to compare with the hash value in the advertisement stage. It is clear to make sure the validation of *bill*.

(3) In our scheme, the third party T signed the receipt $S_{S_B}\langle B_{id},\ sealed\text{-}bid\rangle$ of the bid in the bidding stage, and all published sealed-bids $S_{S_T}\langle$*all order's and sealed-bid's, Time*$\rangle$. Hence, no one can impersonate the third party T to sign this data.

4) Web Data Security

In our scheme, only the third party T can publish information on *Web* to guarantee data security of *Web*; however Liaw *et al.*'s protocol permits the third party T and the auctioneer U to bulletin information on *Web*. Therefore, *Web* must increase extra authentication procedures; otherwise it may suffer from some forged attacks.

5.3 Efficiency

According to Table 1[19,20], we can estimate an account of a symmetric key encryption (DES functions) are about 1000 times faster than an asymmetric key encryption (RSA) in speed. Therefore, our scheme adopts symmetric encryption/decryption for maintaining message confidentiality instead of public key cryptosystem computations in the bidding stage. Moreover, to solve the cheating problem, we take into account the opening stage, such that all bidding prices can be decrypted in the opening stage to enhance fairness.

Table 1. Comparisons of the Computation Speed

Operation	Number of Operations per Second
Public Key Signature (1024 Bits RSA)	2
Symmetric Key Encryption (DES)	2 000
One Way Hash Function (MD5/SHA)	20 000

Table 2. Numbers of Different Computation Comparisons of Related Work

Stage	Liaw *et al.*'s Scheme	Chang and Chang's Scheme	Hwang's Scheme	Proposed Scheme
Advertisement Stage	$nT_{\rm exp}$	0	$3nT_{\rm exp}$	$nT_{\rm exp} + nT_{\rm h}$
Registration Stage	$2nT_{\rm exp} + 5nT_{\rm h}$	$12nT_{\rm exp}$	0	$2nT_{\rm exp} + 2nT_{\rm sym}$ $+3nT_{\rm h}$
Bidding Stage	$5nT_{\rm exp}$	$3nT_{\rm exp} + n \cdot T_{\rm h}$	$10n \cdot T_{\rm exp} + 4n \cdot T_{\rm Xor}$	$4nT_{\rm exp} + nT_{\rm mm}$
Opening Stage	0	$2nT_{\rm exp} + nT_{\rm h}/2$	0	$nT_{\rm exp} + 2n \cdot T_{\rm sym}$ $+nT_{\rm mm}$
Exchange of the Product and the Payment Stage	$4T_{\rm exp} + 2T_{\rm Xor}$	0	$5nT_{\rm exp}$	$3T_{\rm exp} + T_{\rm sym}$
Total	$(8n + 4)T_{\rm exp} + 5nT_{\rm h}$ $+2T_{\rm Xor}$	$17nT_{\rm exp} + 3nT_{\rm h}/2$	$18nT_{\rm exp} + 4n \cdot T_{\rm Xor}$	$(8n + 3)T_{\rm exp} + 4nT_{\rm h}$ $+(4n + 1)T_{\rm sym} + nT_{\rm mm}$

Note: n – the number of bids in an auction; $T_{\rm exp}$ – the time that computes an exponential operation; $T_{\rm h}$ – the time that computes a one way hash function; $T_{\rm sym}$ – the time that computes a symmetric key encryption; $T_{\rm mm}$ – the time that computes a modular multiplication; $T_{\rm Xor}$ – the time that calculates an eXclusive OR operation.

For obviously illustrating computation complexity comparisons, we survey Chang and Chang's scheme[4], Hwang *et al.*'s scheme[21] and Liaw *et al.*'s scheme[7] to compare the number of computation with that of our scheme. We show the complexity comparisons in Table 2.

Liaw *et al.*'s scheme does not take into consideration the opening stage; however we think that it is an essential stage to prevent cheating. According to Table 2, we can observe that our scheme's total numbers of exponential computations are less than those of the rest of three schemes, even if it is included in the opening stage. The third party increases a modular multiplication to get the bidding price for each bid in the opening stage, but it is necessary to prevent a conspiracy attack between the third party and a malicious bidder.

Our scheme has more symmetric key computations than the others, but the symmetric key computation uses shift and substitution computations, which can be 1 000 times faster than the exponential computation. Hence, symmetric key computations can be negligible in entire E-auction protocol. Our scheme computations are still lighter than those of the others.

6 Conclusions

In this paper, we proposed a secure sealed-bid electronic auction scheme with fairness, security and efficiency. Meanwhile, we also proposed a formal proof with random oracles. It can satisfy the previously mentioned auction requirements; meanwhile, the computation load of our scheme is better than that of the other three famous E-auction schemes. Furthermore, we do not only overcome some security drawbacks in Liaw *et al.*'s scheme, but also provide fairness. It also can be implemented in government procurement auction, only to modify in the last stage in the auction. Therefore, we can say that our scheme provides better functionality and efficiency than before. However, it is still hard to propose a formal proof in standard model according to our scheme. Therefore, our future work is to integrate ID-based encryption[22] and propose a formal proof available to achieve semantic security (i.e., security against chosen-plaintext attacks) in the standard model.

Acknowledgement The authors wish to thank the anonymous referees for their suggestions for improving this paper.

References

[1] Franklin M K, Reiter M K. The design and implementation of a secure auction service. *IEEE Transactions on Software Engineering*, May 1996, 22(5): 302–312.

[2] Kudo M. Secure electronic sealed-bid auction protocol with public key cryptography. *IEICE Transactions on Fundamentals*, Jan. 1998, E81-A(1): 20–27.

[3] Kikuchi H, Hakavy M, Tygar D. Multi-round anonymous auction protocols. *IEICE Transactions on Information and Systems*, Apr. 1999, E82-D(4): 769–777.

[4] Chang C C, Chang Y F. Efficient anonymous auction protocols with freewheeling bids. *Computers & Security*, 2003, 22(8): 728–734.

[5] Jiang R, Pan L, Li J H. An improvement on efficient anonymous auction protocols. *Computers & Security*, 2005, 24(2): 169–174.

[6] Chang C C, Chang Y F. Enhance anonymous auction protocols with freewheeling bids. In *Proc. the 20th International Conference on Advanced Information Networking and Applications* (*AINA 2006*), Vienna, Austria, Vol. 1, Apr. 2006, pp.353–358.

[7] Liaw H T, Juang W S, Lin C K. An electronic online bidding auction protocol with both security and efficiency. *Applied Mathematics and Computation*, 2006, 174(2): 1487–1497.

[8] Subramanian S. Design and verification of a secure electronic auction protocol. In *Proc. IEEE 17th Symposium on Reliable Distributed Systems*, Washington DC, USA, 1998, pp.204–210.

[9] Chen T S. An English auction scheme in the online transaction environment. *Computers & Security*, 2004, 23(5): 389–399.

[10] Rivest R. The MD5 message-digest algorithm. RFC 1321, Internet Activities Board, Internet Privacy Task Force, 1992.

[11] NIST FIPS PUB 180-1. Secure hash standard. National Institute of Standards and Technology, Apr. 1995, Available at http://www.itl.nist.gov/fipspubs/fip180-1.htm

[12] Rivest R, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of ACM*, 1978, 21(2): 120–126.

[13] NBA FIPS PUB 46-1. Data encryption standard. National Bureau of Standard, U.S. Department of Commerce, Jan. 1988.

[14] NIST FIPS PUB 197. Advanced data encryption standard. National Institute of Standards and Technology, Nov. 2001. Available at http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[15] Turban E. Electronic Commerce 2002: A Managerial Perspective. Second edition, Prentice Hall, 2002.

[16] Maurer U M. Towards the equivalence of breaking the diffiehellman protocol and computing discrete logarithms. In *Proc. Advanced in Cryptology-CRYPTO'94*, Santa Barbara, USA, Desmedt Y (ed.), *Lecture Notes in Computer Science* 839, Berlin: Springer-Verlag, 1994, pp.271–281.

[17] Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. the First ACM Conference on Computer and Communications Security*, ACM, Nov. 1993, http://www.cs.ucdavis.edu/research/tech-reports/1995/CSE-95-16.pdf.

[18] RSA Laboratories. PKCS #7: Cryptographic message syntax standard. USA, 1997. Available at http://www.rsasecurity.com/rsalabs/node.asp?id=2129.

[19] O'Mahony D, Pierce M, Tewari H. Electronic Payment Systems. Artech House, 1997.

[20] Schneier B. Applied Cryptography. Second edition, New York: John Wiley & Sons, 1996.

[21] Hwang M S, Lu E J L, Lin I C. Adding timestamps to the electronic auction protocol. *Data & Knowledge Engineering*, 2002, 40: 155–162.

[22] Waters B. Efficient identity-based encryption without random oracles. In *Proc. Advanced in Cryptology-EUROCRYPTO 2005*, Aarhus, Denmark, *Lecture Notes in Computer Science* 3494, Springer-Verlag, 2005, pp.114–27.

Chia-Chi Wu was born in 1967 in Taoyuan. He is currently a Ph.D. candidate in computer science and information engineering in "National Chung Cheng University". His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Chin-Chen Chang received his B.S. degree in applied mathematics in 1977 and his M.S. degree in computer and decision sciences in 1979, both from the "National Tsing Hua University". He received his Ph.D. degree in computer engineering in 1982 from the "National Chiao Tung University". During the academic years of 1980∼1983, he was on the faculty of the Department of Computer Engineering at the "National Chiao Tung University". From 1983∼1989, he was on the faculty of the Institute of Applied Mathematics, "National Chung Hsing University". From 1989 to 2004, he has been a professor in the Institute of Computer Science and Information Engineering at "National Chung Cheng University". Since 2005, he has worked as a professor in the Department of Information Engineering and Computer Science at Feng Chia University. Dr. Chang is a fellow of IEEE, a fellow of IEE. His research interests include computer cryptography, data engineering, and image compression.

Iuon-Chang Lin was born on 1974 in Taipei. He received the B.S. degree in computer and information sciences from "Tung Hai University", in 1998; the M.S. degree in information management from Chaoyang University of Technology, in 2000. He received his Ph.D. degree in computer science and information engineering in March 2004 from "National Chung Cheng University". From 2004 to 2005, he was an assistant professor of the Department of Information Management, "National Kaohsiung University of Applied Sciences". He is currently an assistant professor of the Department of Management Information Systems, National Chung Hsing University. His current research interests include, but not limited to, electronic commerce, information security, cryptography, neural networks, and electronic image processing.