Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 6567146, 14 pages
http://dx.doi.org/10.1155/2016/6567146

Research Article

An Efficient Electronic English Auction System with a Secure On-Shelf Mechanism and Privacy Preserving

Hong Zhong,[1] Song Li,[1] Ting-Fang Cheng,[2] and Chin-Chen Chang[2]

[1]*School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, China*
[2]*Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan*

Correspondence should be addressed to Chin-Chen Chang; alan3c@gmail.com

Received 16 December 2015; Accepted 24 March 2016

Academic Editor: Isao Echizen

With the rapid development of the Internet, electronic commerce has become more and more popular. As an important element of e-commerce, many Internet companies such as Yahoo! and eBay have launched electronic auction systems. However, like most electronic commerce products, safety is an important issue that should be addressed. Many researchers have proposed secure electronic auction mechanisms, but we found that some of them do not exhibit the property of unlinkability, which leads to the leakage of users' privacy. Considering the importance of privacy preservation, we have designed a new auction mechanism. Through symmetrical key establishment in the registration phase, all messages transmitted over the Internet would be protected and, meanwhile, achieve the property of unlinkability. The security analysis and performance analysis show that our protocol fulfills more security properties and is more efficient for implementation compared with recent works.

1. Introduction

With the development of network technology, more and more people are searching for information on the Internet. In 2013, the number of Internet users of the whole world reached 2.92 billion [1]. Now, people not only read the news and search for information on the Internet but also do business with others. Not surprisingly, e-commerce has grown rapidly in recent decades. As the founder and CEO of China's most famous e-commerce company Alibaba, Jack Ma was ranked as the second richest man in China in 2014. Many famous Internet companies launched electronic auction products several years ago such as Yahoo! and eBay. On these websites, users can play the role of not only auctioneer but also bidder after they register their accounts successfully. They can participate in the auction anytime and anywhere when their devices have access to the Internet.

Auctions can be classified into many types [2]. According to numbers of sellers and buyers, auctions can be classified as forward auctions and reverse auctions; according to the determinant of the winner, auctions can be classified as single attribute auctions and multiattribute auctions; and according to whether the bidding price is made open, we can classify auctions as sealed-bid auctions and open auctions. The sealed-bid auctions can be subdivided into sealed-bid first-price auctions and sealed-bid second-price auctions. In the first mode, all bidders submit their sealed bids to the auctioneer of an auction at the same time, and the auctioneer or arbiter of the auction secretly calculates the highest price. After that, the auctioneer or arbiter announces the winner as the owner with the highest price bid. In a sealed-bid second-price auction (also called Vickrey auction), the highest price bidder wins the auction but only needs to pay the second highest price. The designer of the sealed-bid second price auction thinks that everyone will submit their bidding price rationally in this mode; however, this method confronts a lot of problems in practice, such as bidder collusion. Open auctions can be further classified into English auctions and Dutch auctions. In an English auction, the auctioneer changes the current price dynamically, with the bidding price increasing. This means that, if one bidder's bid price is higher than the current price, then the auctioneer uses this price as the new current price and waits for someone to offer a higher price than the current price. If someone submits a higher price, the current price is changed simultaneously; if no one offers a higher price when the auction is closed, the

owner of the current price bid wins the auction. A Dutch auction is bidding process contrary to the English auction. In a Dutch auction, the auctioneer decreases the current price until one bidder can afford it and is the winner of the auction.

A lot of problems arise when we introduce the auction process in reality on the Internet. An important issue is the information security of the system. As an Internet product is connected with money and goods, users are concerned with the safety of the online auction. Many researchers have proposed their protocols to solve security problems in auction systems. In 1999, based on millionaires' problem, Cachin [3] proposed a private bidding and auction scheme with an oblivious third party. In 2000, Nguyen and Traoré [4] used a group signature to protect bidders' anonymity. However, the huge computational cost and the special authority of group managers still cannot be resolved. In 2001, Omote and Miyaji [5] applied a bulletin board to overcome these problems, but their scheme does not publish the winner's information at the end. In 2003, C.-C. Chang and Y.-F. Chang [6] proposed an anonymous English auction protocol to ensure that bidders can bid arbitrarily. In 2005, Jiang et al. [7] considered that the scheme in [6] cannot protect against man-in-the-middle attacks. In 2006, Y.-F. Chang and C.-C. Chang [8] proposed another anonymous auction scheme to resolve the problem in [6]. In 2005, Suzuki and Yokoo proposed safety problem in the multiattribute auction systems [9]. In 2007, Shih et al. proposed a privacy preserving multi-item auction mechanism with a shared key chain [10]. In 2008, Parkes et al. used a homomorphic cryptograph to achieve bid privacy in multi-item auction [11]. In 2009, Xiong et al. [12] proposed an anonymous auction scheme based on the ring signature. However, like the group signature-based protocol in [4], huge computational cost is an essential problem that should be resolved. In 2012, Xiong et al. [13] proposed another protocol based on revocable ring signature to solve the problem of high computational cost in [12] and added a dispute section.

In 2013, Chang et al. [14] pointed out that [13] is vulnerable to denial of service attacks and designed an ECC-based protocol which provides a secure on-shelf phase. Unfortunately, we found that Chang et al.'s proposal [14] does not provide unlinkability. In [14], users' privacy can be leaked by monitoring the communication of system and linking the captured information. Although some important data such as public key and certificate are encrypted before transmission, unencrypted messages such as on-shelf information can link different packets together. For example, in the on-shelf phase, although attackers cannot calculate the true public key of the auctioneer using an encrypted message transmitted from the auctioneer and the published anonymous public key on website, attackers still can link transmitted packets and website information together through on-shelf information. This is because the on-shelf information is transmitted in plaintext. On the other hand, bidders' privacy is in more serious danger during the bidding phase. At the end of the bidding phase, the agent center (*AC*) publishes the information with the bidder's true public key on its website. Attackers can link this public key with a certain IP address and know what the user bids on.

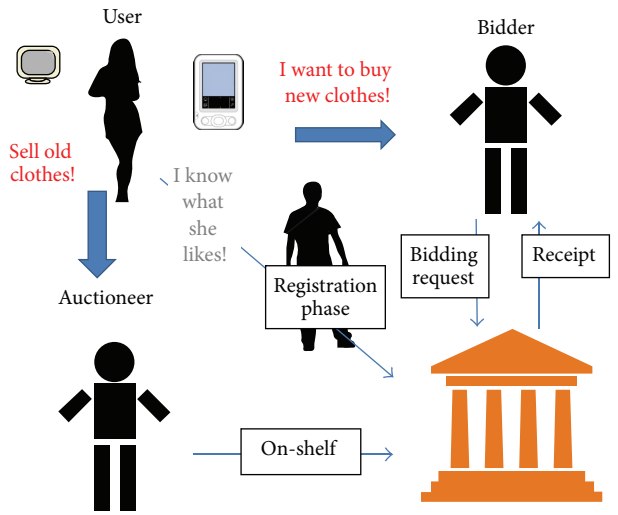

Figure 1: Privacy problem in electronic auction system.

Then, attackers can trace this public key owner's transaction history.

Until now, many researchers have stressed the importance of privacy preservation in online auctions [15–17]. Hence, in order to overcome the privacy problem in [14], in this paper, we provide a new English auction system with privacy preservation. The overview of the auction system architecture is shown in Figure 1. Using the same system mode with Chang et al.'s proposal [14], our system has 5 phases: the registration phase, on-shelf phase, bidding phase, product-claim phase, and dispute phase. We utilize a trusted third party as the *AC* of our system, a well-meaning role that will not initiate attacks. However, the safety of the *AC*'s database cannot be completely guaranteed. We do not consider the case of hackers controlling the *AC*, but hackers may get access to read data in the *AC*. The only data we consider to be completely safe is the private key of *AC*. Besides this, any leaked data cannot do serious harm to the safety of the whole system. In addition, we built our protocol on Elliptic Curve Cryptosystem. According to the analysis in Section 8, our proposal has an efficiency advantage over the original proposal. In our scheme, messages transmitted between any two entities are encrypted with a symmetrical encryption algorithm such as AES. By utilizing this method, monitors cannot link different messages together. Security analysis shows that our proposal provides more security properties than the original one.

The rest of this paper is organized as follows. In Section 2, we introduce the Elliptic Curve Cryptosystem and ECDLP (elliptic curve discrete logarithm problem). Subsequently, we briefly review Chang et al.'s auction protocol [14] in Section 3. In Section 4, we talk about network model, adversary model, and properties that an auction mechanism needed. We describe our proposal in detail and use BAN logic to analyze authentication accuracy of on-shelf phase and bidding phase in Sections 5 and 6, respectively. In Sections 7 and 8, we compare the security and efficiency of our protocol with related schemes.

2. Preliminaries

In this part, we introduce some basic ideas of Elliptic Curve Cryptosystem including the definition of elliptic curve, basic operations on the Elliptic Curve, and ECDLP on ECC that can be used to construct cryptographic algorithms.

2.1. Elliptic Curve Cryptosystem. Elliptic Curve Cryptosystem (ECC) is an asymmetrical cryptosystem. It was independently proposed by Miller [19] and Koblitz [20] in 1985 and 1987. Compared to RSA, ECC can achieve the same security requirement with a shorter key-length [21]. Hence, it has been widely used in many cryptographic schemes recently.

An elliptic curve [22, 23] is defined over a finite field F_p by equation $E_p(a,b) : y^2 = x^3 + ax + b$, where p is a large prime and $4a^3 + 27b^2 \neq 0 \bmod p$. The points on this elliptic curve form a cyclic group. Addition in this group is defined as if points $P, Q, R \in E_p(a,b)$ on one line, and then $P + Q + R = O$ (O is infinite point). Given an integer $s \in F_p^*$ and a point $P \in E_p(a,b)$, the multiplication operation $s \cdot P$ over $E_p(a,b)$ is defined as $P + P + \cdots + P$ in s times. If P is symmetrical with P' on the x-axis, then $P + P' = O$. Furthermore, point P is a base point with an order n if and only if $n \cdot P = O$.

2.2. ECDLP. Every cryptosystem has its own difficult problem, such as the integer factorization used in RSA. The most important difficult problem in ECC is the elliptic curve discrete logarithm problem (ECDLP) [24]. Based on ECDLP, we can develop many other difficult problems, such as the computational Diffie-Hellman problem (CDLP) and the elliptic curve factorization problem (ECFP). In our scheme, we will use ECDLP.

Definition 1 (elliptic curve discrete logarithm problem (ECDLP)). Given two points P and Q over $E_p(a,b)$, it is very hard to find an integer $s \in F_p^*$ such that $Q = sP$.

3. Related Work

In this part, we describe Chang et al.'s proposal [14] briefly and analyze the security and privacy preserving in it.

Chang et al.'s protocol includes five phases: registration phase, on-shelf phase, bidding phase, product claiming phase, and dispute phase. A new user should register to an agent center (AC) and AC will issue a certificate for him/her such that he/she can play the role of both auctioneer and bidder. If a registered user wants to initiate an auction, he/she should send some basic information about auction (i.e., product identity, basic price of the auction, and deadline) to AC via the on-shelf phase. Upon checking the legality of user's identity and the information of the new auction, AC publishes an advertisement on its website (BB_{AC}) such that anyone can read it after the auction has started. If someone wants to bid for the product of this auction via the bidding phase, he/she can send bidding information to AC. After AC checked the legality of bidder's identity and the bidding price is higher than current price, AC should change the value of current price to the bidding price and publish the bidder's public key and bidding price on BB_{AC}; else, AC ignores this bid and waits for new bid. After the auction deadline, AC selects the bidder with highest price as the winner, sends a receipt to the winner, and publishes final result on BB_{AC} such that anyone can read and verify it. After getting the receipt, the winner can claim the product from the auctioneer by showing the receipt via the product claiming phase. Then, the auctioneer can verify the correctness of this receipt and send the product to the user. With regard to the dispute phase, if a user takes action illegally, the user trading with him/her can submit dispute information to AC and wait for the arbitration of AC. If illegal action existed, AC can trace the identity of malicious user and punish him/her.

The notations used in Chang et al.'s protocol are as follows and the details of their protocol are described in the following except of the dispute phase:

AC: a trusted agent center.

B_Z: a Bidder Z.

A_Z: an auctioneer Z.

ID_Z: the identity of Z.

SK_{AC}: AC's private key.

PK_Z: Z's public key.

PW_Z: Z's password.

$CERT_Z$: the certificate of PK_Z signed by AC.

BB_{AC}: the Bulletin board of AC.

P: the base point of ECC group over a finite field F_p with order p, where p is a large prime.

$(\cdot)_x$: an operation using x-coordinate in ECC.

$(\cdot)_y$: an operation using y-coordinate in ECC.

$E_k[\cdot]/D_k[\cdot]$:AES-based encryption/decryption with key k.

$f(\cdot)$: a secure one-way hash function.

3.1. Registration Phase. If a user wants to play the role of an auctioneer or a bidder in the auction, he/she should register an account at AC by the following steps.

(1) User chooses ID_u, $PW_u \in Z_p^*$ and a random number $r_u \in Z_p^*$, and computes his/her public key $PK_u = PW_u \cdot P \bmod p$. Then, the user calculates $R_0 = r_u \cdot P \bmod p$, $R_1 = ID_u + (r_u \cdot PK_{AC})_x \bmod p$, and $R_2 = PK_u + r_u \cdot PK_{AC} \bmod p$ and sends (R_0, R_1, R_2) to AC.

(2) AC retrieves user's identity $ID_u = R_1 - (SK_{AC} \cdot R_0)_x \bmod p$ and public key $PK_u = R_2 - SK_{AC} \cdot R_0 \bmod p$. Subsequently, AC chooses a random number $r_{AC} \in Z_q^*$ calculates $R_0' = r_{AC} \cdot P \bmod p$ and $CERT_u = r_{AC} + (PK_u)_x \cdot SK_{AC} \cdot ID_u \bmod p$, and sends them to the user. User checks the correctness of equation $CERT_u \cdot P - R_0' ? \equiv (PK_u)_x \cdot ID_u \cdot PK_{AC} \pmod p$ to verify the validation of this certificate. If valid, then user stores it with R_0'.

3.2. On-Shelf Phase. If a registered user wants to host an auction, then he/she becomes an auctioneer A_j and performs the following.

(1) A_j chooses a random number $r_1 \in Z_p^*$ and a nonce $n_1 \in Z_p^*$ and calculates $O_0 = r_1 \cdot P \bmod p$, $O_1 = (PK_{A_j} + r_1 \cdot PK_{AC}) \bmod p$, $K_{aa} = (PW_{A_j} \cdot PK_{AC} \cdot n_1)_x \bmod p$, and $O_2 = E_{K_{aa}}(R_0' \parallel CERT_{A_j})$. A_j generates auction information $AI = (pid_k \parallel BasicPrice \parallel Deadline)$, that is, the identity of a product, basic price of this auction, and the deadline of this auction. Finally, A_j computes $Sig = (r_1 + AI \cdot PW_{A_j} \cdot n_1) \bmod p$ and sends $(ID_{A_j}, n_1, O_0, O_1, O_2, AI, Sig)$ to AC.

(2) AC retrieves A_j's public key $PK_{A_j} = O_1 - SK_{AC} \cdot O_0$. Then, AC computes the symmetrical key shared with A_j : $K_{aa} = (SK_{AC} \cdot PK_{A_j} \cdot n_1)_x \bmod p$. AC can decrypt O_2 with K_{aa} to extract A_j's certificate. AC should check the correctness of the certificate by verifying if equation $CERT_{A_j} \cdot P - R_0' \equiv (PK_{A_j})_x \cdot ID_{A_j} \cdot PK_{AC} (\bmod p)$ holds and the integrity of AI by verifying if equation $Sig \cdot P - O_0 \equiv PK'_{A_j} \cdot AI (\bmod p)$ holds where $PK'_{A_j} = n_1 \cdot PK_{A_j} \bmod p$. If both are valid, AC generates a unique gid for A_j's product. AC then computes a verifier $V = h(gid \parallel K_{aa})$ and sends (V, gid) to A_j.

(3) After A_j received message from AC, he/she computes and checks if $V = h(gid \parallel K_{aa})$. If the equation holds, A_j responds *Approval* message to AC. Upon receiving *Approval* from A_j, AC publishes $(gid, ID_{A_j}, AI, O_0, PK'_{A_j}, Sig, N_1)$ on BB_{AC} and anyone can verify this message by checking equation $SigP - O_0 \equiv PK'_{A_j} \cdot AI (\bmod p)$.

3.3. Bidding Phase. If someone is interested in this auction, he/she could play the role of a bidder B_i and send bidding message to AC by the following.

(1) B_i chooses a random number $r_2 \in Z_q^*$ and computes $D_0 = r_2 \cdot P \bmod p$, $D_1 = (PK_{B_i} + r_2 \cdot PK_{AC} \bmod p$, $K_{ba} = (PW_{B_i} \cdot PK_{AC} \cdot (D_0)_y)_x \bmod p$, and $D_2 = E_{K_{ba}}(R_0' \parallel CERT_{B_i} \parallel ID_{B_i})$. B_i generates bidding message $BM = (gid \parallel price)$ and signs it as $BSig_{B_i} = (r_2 + BM \cdot PW_{B_i}) \bmod p$. B_i sends $(D_0, D_1, D_2, BM, BSig_{B_i})$ to AC.

(2) Upon receiving these messages, AC retrieves B_i's public key $PK_{B_i} = D_1 - D_0 \cdot SK_{AC} \bmod p$ and computes $K_{ba} = (SK_{AC} \cdot PK_{B_i} \cdot (D_0)_y)_x$. Then, AC decrypts D_2 using K_{ba} to extract B_i's certificate. AC can verify the correctness of $CERT_{B_i}$ by checking equation $CERT_{B_i} \cdot P - R_0' ? \equiv (PK_{B_i})_x \cdot ID_{B_i} \cdot PK_{AC} (\bmod p)$. If it is valid, AC checks the correctness of BM by checking equation $BSig_{B_i} \cdot P - D_0 ? \equiv PK_{B_i} \cdot BM (\bmod p)$. If it is valid too, AC further checks *Deadline* and *BasicPrice* of the auction by using gid to search the corresponding information on its website. If the auction has closed or the *price* is not larger than *BasicPrice*, AC rejects this bid; otherwise, AC generates a unique transaction identity tid for B_i and chooses a random number $r_3 \in Z_q^*$. AC then computes $D_3 = r_3 \cdot P \bmod p$, $BM' = (tid \parallel BM)$, $Rec = r_3 + (BM' \cdot SK_{AC})_x \bmod p$, and $D_4 = E_{K_{ba}}(tid \parallel D_3 \parallel Rec)$. Finally, AC updates *BasicPrice* = *price*, publishes $(tid, price, date)$ on BB_{AC}, stores $(tid, price, date, D_0, BM, BSig_{B_i}, PK_{B_i}, D_2)$ in database, and sends D_4 to bidder.

(3) After B_i received the message from AC, B_i can extract the transaction information $(tid \parallel D_3 \parallel Rec)$ from D_4. Then, B_i checks the validation of receipt Rec by verifying $Rec \cdot P - D_3 ? \equiv (tid \parallel BM) \cdot PK_{AC} \bmod p$. If the equation holds, B_i stores Rec and D_3.

(4) After deadline, AC sets the winner as the bidder with the highest price. AC then publishes the winner's information on BB_{AC} including several items $(tid, price, date, D_0, bid, BSig_{B_i}, PK_{B_i}, D_2)$ such that anyone can verify its validity by checking $BSig_{B_i} \cdot P - D_0 ? \equiv PK_{B_i} \cdot BM (\bmod p)$.

3.4. Product Claiming Phase. After the auction deadline, the winner can claim the product from the auctioneer by showing his/her receipt and performing the following steps. After the auctioneer checked the correctness of receipt, the auctioneer should send the product to the winner.

(1) B_i selects a random number r_c and a nonce n_2 from Z_q^*, computes $C_0 = r_c \cdot P \bmod p$, $C_1 = (Rec + r_c \cdot n_2 \cdot PK'_{A_j}) \bmod p$, and $C_2 = (D_3 + r_c \cdot n_2 \cdot PK'_{A_j}) \bmod p$, and sends $(gid, C_1, C_2, C_0, n_2)$ to A_j.

(2) Once receiving B_i's claiming request, A_j retrieves $Rec = (C_1 - (C_0 \cdot n_2 \cdot n_1 \cdot PW_{A_j})_x) \bmod p$ and $D_3 = (C_2 - C_0 \cdot n_2 \cdot n_1 \cdot PW_{A_j}) \bmod p$. Then, A_j verifies $Rec \cdot P - D_3 ? \equiv (tid \parallel gid \parallel price) \cdot PK_{AC} \bmod p$. If the verification holds, A_j is convinced that B_i is the actual winner and sends the product to B_i.

3.5. Security Defect of Chang et al.'s Protocol. According to our analysis, we found that Chang et al.'s protocol meets most of the security requirements of an auction protocol except for unlinkability. In Chang et al.'s protocol, although some important data such as users' public keys and certificates are encrypted before being transferred, unencrypted messages such as the on-shelf information still can be linked together with different packets. Figures 2 and 3 show the linkage between transferred messages and published messages on AC's website. The solid border represents the message transferred on the Internet, and the dotted border represents the message published on AC's website.

From Figures 2 and 3, we can see that these messages can be linked together through certain parameters. For example, in the on-shelf phase (Figure 2), although attackers cannot calculate the public key or certificate of an auctioneer A_j from

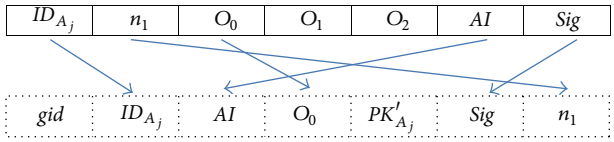

Figure 2: Linkage of messages in on-shelf phase.

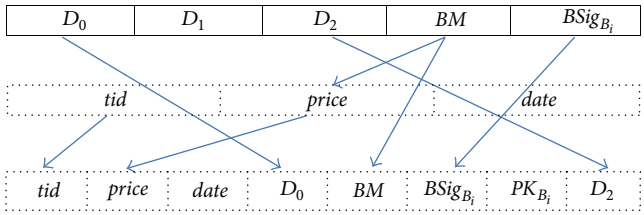

Figure 3: Linkage of messages in bidding phase.

the encrypted message (solid border) based on the definition of ECDLP and the security assumption of underlying AES algorithms, attackers can still find the relevance of transferred messages (solid message) and published messages (dotted message) to analyze the specific user's privacy. As shown in Figure 2, we find that ID_{A_j}, n_1, AI, O_0, and Sig are linkable. Obviously, an adversary can link the captured message on the Internet with the published message if they have the same parameters ID_{A_j}, n_1, AI, O_0, andSig. Furthermore, the adversary can decide that if these messages come from the specific IP address. If so, the adversary obtained this IP address owner's identity ID_{A_j}.

On the other hand, as mentioned previously, bidders' privacy is in more serious danger in the bidding phase. At the end of the bidding phase, the *AC* publishes the information with the winner's true public key on *AC*'s website. As shown in Figure 3, similar to Figure 2, attackers have the ability to determine the IP addresses of captured messages on the Internet and to link the specific transferred encrypted bidding information with the published message (the winner's information). Consequently, attackers would find the relevance of the winner's public key and the specific IP address. If this user continues to anticipate other auctions, his/her transaction history will be traced. For example, an adversary used Sniffer (a network tool to capture the packets transferred in local network) to capture packets transferred in the local network of his/her office environment and found that some packets are transferred to the auction server (*AC*). After analyzing the important parts of these packets and comparing them with the messages published on *AC*'s website, the adversary can know what the specific IP address owner had bidden for or sold out. Obviously, the adversary can easily know who uses the IP address in this office if he/she wishes.

According to the aforementioned analyses, we can see that the main problem that results in privacy disclosure is that some published data is transferred in plaintext before such that an adversary can easily link them together. Hence, in our proposal, we establish a shared symmetrical key between each user and the *AC* to encrypt the message transferred on the Internet to avoid linkability. By using symmetrical encryptions and fewer ECC operations, our proposal can not only improve the security but also reduce the system cost. In the registration phase, we connect a user's identity and the corresponding symmetrical key so that the *AC* can compute the shared symmetrical key easily with the user's identity. Besides, each user's identity is encrypted with the *AC*'s public key so it cannot be revealed without the knowledge of *AC*'s private key. In our proposal, we fulfill the verifiability of all messages published on the *AC*'s website; however, Chang et al.'s proposal only partially achieved this feature.

4. System and Adversary Model

In this section, we describe the definitions of our network model, adversary model, and requirements as follows before introducing our proposed scheme.

4.1. Network Model. We consider a network composed of an agent center (*AC*) and users. The *AC* is a trusted third party that undertakes most tasks of system running, such as user registration, generation of users' certificates, and product on-shelf. With regard to the application of the trusted third party, it is widespread in e-commerce systems [25–28]. The *AC* is also an arbiter of an auction and has the right to determine who the winner is or punish the illegal user's identity. Hence, we think that the *AC* has large capability of computation and storage. The *AC* should maintain a database on its machine to store users' data. Herein, we do not consider this database to be completely secure. Hackers may have methods to access it, but such actions will not destroy the security of the whole system. Furthermore, the *AC*'s private key should be kept in a trusted place such as a bank security box.

On the other hand, users use their devices such as computers or mobile devices to communicate with the *AC* through the Internet. When a user wants to connect with the *AC*, he/she should type in a password or insert a smart card with a password into the machine. The client in the user's machine erases the password after the user goes offline.

4.2. Adversary Model. In this paper, we assume that an adversary can launch a passive attack to monitor the communication channel of the system. The adversary has knowledge of the format of a packet, so he/she can analyze what content is included in the packet. We also think that the adversary has the ability to access the database stored in the *AC* and can read the *AC*'s website as the other users. In addition, some registered users may attack the system jointly.

4.3. Security Requirements. In order to provide a secure auction scheme, the following security properties are critical [29].

Anonymity. User's identification should be kept secret as the system is running. No one can obtain the user's identification from messages transmitted or published.

Easy Revocation. A user's right should be easily and correctly revoked in the auction system by the *AC*.

Fairness. All bidders should have the ability to verify whether their bids have indeed been included in the auction.

Nonrepudiation. Users including bidders and auctioneers cannot deny the actions that have been taken before. For example, if an auctioneer puts a product on-shelf, he/she cannot deny that he/she did it and off-shelf this product arbitrarily.

Onetime Registration. Any registered user can host or participate in any auction without reregistration at the *AC*.

Traceability. If some illegal actions are taken such as a nonpaying bid (*NPB*), the *AC* should have the ability to identify and publish the dishonest user's true identity anytime.

Unforgeability. No one can forge a valid message or impersonate any legal users to do illegal things.

Unlinkability. No one can link different messages together to trace a specific user's transaction history even if he/she does not know the true identification of the user.

Verifiability. All winning bids and product information published on a bulletin board should be publicly verifiable without revealing the bidders' identities.

5. Proposed Scheme

In this section, we propose a secure English auction system with privacy preservation. Our proposal consists of six phases: the system setup phase, registration phase, on-shelf phase, bidding phase, product claiming phase, and dispute phase. There are two kinds of participants in this system: the *AC* and the users. In the system setup phase, the *AC* inputs a security parameter and generates a set of system parameters. The *AC* then requests a certificate from an acknowledged certificate authority and publishes these data on its website. Anyone can get these data from the *AC*'s website. If a new user wants to register an account, he/she can connect with the *AC* during the registration phase. After registration, the user can either host or participate in an auction. If he/she wants to host an auction, he/she can on-shelf his/her products in the on-shelf phase; if he/she wants to bid for some on-shelved products, he/she can send the bidding information to the *AC* during the bidding phase. The *AC* should publish every bidder's bidding information except for the bidder's identity on its website such that anyone can read it. If a bidder wins the auction, he/she will receive a receipt and a session key shared with the auctioneer from the *AC*. Then, the winning bidder can obtain his/her product from the auctioneer during the product claiming phase. In our proposal, the *AC* has the capability of tracing the whole transaction and identifying the participants in an auction if necessary. If a dispute has occurred, anyone participating in an auction can submit the dispute request to the *AC* in the dispute phase. The notations used in our proposal are as follows and the details of our proposal are described in the following subsections:

B_i: a bidder i.

A_j: an auctioneer j.

AC: a trusted agent center.

κ: the system security parameter chosen by *AC*.

F_q: finite filed of order q.

E/F_q: an elliptic curve based on q order finite field.

G_q: a cyclic group of the elliptic curve.

P: the base point of ECC group.

$(\cdot)_x$: an operation using x-coordinate in ECC.

SK_{AC}: *AC*'s private key.

PK_{AC}: *AC*'s public key.

$Cert_{AC}$: the certificate of PK_{AC} signed by the certificate authority.

id_U: the identity of a user U.

pw_U: U's password.

N_n: a nonce.

P_U: U's partial key used for computing symmetrical key shared with *AC*.

K_{AU}: the symmetrical key of *AC* and a user.

K_{AB}: the symmetrical key of *AC* and a bidder.

K_{AA}: the symmetrical key of *AC* and an auctioneer.

$E_K(\cdot)$: an AES-based encryption with key K.

$D_K(\cdot)$: an AES-based decryption with key K.

$H(\cdot)$: a secure one-way hash function.

BB_{AC}: the bulletin board of *AC*.

5.1. System Setup Phase. Before the system is running, the *AC* chooses $SK_{AC} \in_R Z_q^*$ as the private key. The *AC* then inputs a security parameter $\kappa \in Z_q^*$ and generates a set of system parameters $\Omega = \{F_q, E/F_q, G_q, P, PK_{AC}, Cert_{AC}, H(\cdot)\}$ through the steps below:

(1) Choose a κ-bit prime q.

(2) Determine the tuple $\{F_q, E/F_q, G_q, P\}$; F_q is a finite filed modular q, E/F_q is an elliptic curve defined over F_q, G_q is a cyclic group defined over F_q, and P is the generator of G_q.

(3) Compute the *AC*'s public key $PK_{AC} = SK_{AC} \cdot P \bmod q$.

(4) Choose a cryptographic hash function $H(\cdot)$.

(5) Request a certificate $Cert_{AC}$ of PK_{AC} from the certificate authority.

Then, the *AC* publishes Ω on its online BB_{AC} and keeps SK_{AC} secret.

5.2. Registration Phase. If a user wants to host or participate in an auction, he/she must perform the following steps to register an account at the *AC*.

(1) The user obtains system parameters on BB_{AC} and checks the validation of the *AC*'s public key by $Cert_{AC}$. If PK_{AC} is invalid, he/she terminates the subsequent operations; otherwise, he/she chooses his/her password $pw_U \in Z_q^*$ and identity $id_U \in Z_q^*$ and generates a random number r_0 and a nonce N_1. Afterwards, he/she computes $R_0 = r_0 \cdot P \bmod q$, $P_U = pw_U \cdot P \bmod q$, $C = (r_0 \cdot PK_{AC} + P_U) \bmod q$, and $C_0 = ((r_0 \cdot PK_{AC})_x + M) \bmod q$ and sends (C, C_0, R_0) to *AC*, where $M = (id_U, N_1)$.

(2) Upon the receipt of the message sent from the user, the *AC* can extract P_U and M by computing $P_U = C - SK_{AC} \cdot R_0 \bmod q$ and $= (id_U, N_1) = C_0 - (SK_{AC} \cdot R_0)_x \bmod q$. The *AC* then checks the freshness of N_1. If N_1 is valid, the *AC* chooses a nonce N_2 and computes $RESPONSE = ((success, N_2) + (SK_{AC} \cdot P_U)_x) \bmod q$. The *AC* then sends *RESPONSE* to the user and stores (id_U, v_U) in its database, where $v_U = P_U \oplus H(SK_{AC})$.

(3) After receiving *RESPONSE*, the user retrieves $(success, N_2) = (RESPONSE - (PK_{AC} \cdot pw_U)_x) \bmod q$ and checks if *success* is contained in it. If so, the user checks the freshness of N_2. If both are valid, the user ascertains that he/she has registered successfully and then can either host or participate in an auction.

Note that each registered user has a partial key P_U and can share a key $K_{AU} \equiv (pw_U \cdot PK_{AC})_x \equiv (SK_{AC} \cdot P_U)_x (\bmod q)$ with the *AC*. For convenience, in the following, we use K_{AA} to imply the key shared between the *AC* and a user who plays as an auctioneer A_j, and A_j's partial key is P_{A_j}. In contrast, the keys of a user who plays as a bidder B_i are K_{AB} and P_{B_i}.

5.3. On-Shelf Phase. If a registered user wants to host an auction to sell some products, he/she becomes an auctioneer A_j and performs the following steps.

(1) A_j generates on-shelf information $\gamma = (ID_{\text{product}} \parallel Basic\_price \parallel Deadline)$ firstly, where ID_{product} refers to the identity of a product, *Basic_price* indicates the basic price of this auction (every bidder's bidding price should not be less than this value), and *Deadline* represents the deadline of this auction. Then, A_j computes $K_{AA} = (pw_{A_j} \cdot PK_{AC})_x \bmod q$. After that, A_j randomly chooses a nonce N_3 and calculates $e_1 = E_{K_{AA}}(\gamma, N_3, id_{A_j})$, $h_1 = H(e_1, K_{AA})$, $C_1 = (h_1 \cdot PK_{AC})_x + id_{A_j} \bmod q$, and $R_1 = h_1 \cdot P \bmod q$. Then, A_j sends (C_1, R_1, e_1) to the *AC*.

(2) The *AC* retrieves id_{A_j} from C_1 by computing $id_{A_j} = C_1 - (R_1 \cdot SK_{AC})_x \bmod q$. Then, the *AC* searches its database to extract the corresponding P_{A_j} ($P_{A_j} = H(SK_{AC}) \oplus v_{A_j}$) of id_{A_j} and calculates $K_{AA} = (SK_{AC} \cdot P_{A_j})_x \bmod q$. The *AC* extracts (γ, N_3, id_{A_j}) by decrypting operation $D_{K_{AA}}(e_1)$. The *AC* checks whether the identity id_{A_j} in e_1 is equal to that retrieved from C_1. If they are not equal, the *AC* may confront a replay attack, and the *AC* should terminate sequent operations immediately; otherwise, the *AC* checks the freshness of N_3. If both are valid, the *AC* generates a unique on-shelf identity $ID_{\text{on-shelf}}$ for A_j and sends $E_{K_{AA}}(ID_{\text{on-shelf}}, id_{A_j}, N_4)$ to A_j, where N_4 is a nonce.

(3) After receiving the message from the *AC*, A_j retrieves $(ID_{\text{on-shelf}}, id_{A_j}, N_4)$ by decryption with key K_{AA}. A_j checks the integrity of the message by checking if a correct id_{A_j} is in it and checks the freshness of this message by checking N_4. If both are valid, the auctioneer then sends *APPROVAL* to *AC*.

(4) Upon receiving *APPROVAL* from A_j, the *AC* chooses a random number $r_1 \in_R Z_q^*$ and computes $R_2 = r_1 \cdot P \bmod q$ and $S_1 = r_1^{-1} \cdot (M_1 - SK_{AC} \cdot (R_2)_x) \bmod q$, where $M_1 = (\gamma, ID_{\text{on-shelf}}, id_{A_j})$. Waiting τ minutes ($\tau \in_R (0, t)$), the *AC* publishes (M_1, R_2, S_1) on BB_{AC}. Once the *AC* publishes the on-shelf information, the auction has been held, and anyone can verify the validation of the auction information by checking the correctness of equation $S_1 \cdot R_2 + (R_2)_x \cdot PK_{AC}? \equiv M_1 \cdot P \bmod q$.

5.4. Bidding Phase. If a bidder B_i wants to bid for an on-shelved product, he/she should send the bidding information to the *AC*, including the corresponding identity $ID_{\text{on-shelf}}$ of the product and a bidding price *price*. We denote $(ID_{\text{on-shelf}} \parallel price)$ as η here.

(1) B_i computes $K_{AB} = pw_{B_i} \cdot PK_{AC} \bmod q$ firstly. Then, B_i computes $e_2 = E_{K_{AB}}(\eta, N_5, id_{B_i})$, $h_2 = H(e_2, K_{AB})$, $C_2 = ((h_2 \cdot PK_{AC})_x + id_{B_i}) \bmod q$, and $R_3 = h_2 \cdot P \bmod q$, where N_5 is a nonce. Then, B_i sends (C_2, R_3, e_2) to the *AC*.

(2) After receiving the bid sent from B_i, the *AC* can retrieve id_{B_i} from C_2 by computing $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$. Then, the *AC* searches the id_{B_i} in the database. If the id_{B_i} does not exist, the *AC* terminates the sequent operation; otherwise, the *AC* retrieves the corresponding P_{B_i} by computing $P_{B_i} = H(SK_{AC}) \oplus v_{B_i}$ and computes the symmetrical key shared with B_i as $K_{AB} = SK_{AC} \cdot P_{B_i} \bmod q$. The *AC* decrypts e_2 with K_{AB} to extract (η, N_5, id_{B_i}). The *AC* can check the integrity of this bid by seeing if id_{B_i} is equal to the one retrieved from C_2. If they are not equal, the *AC* may confront a replay attack, and the *AC* should terminate the sequent operations; otherwise, the *AC* checks the freshness of N_5. If both are valid, the *AC* uses $ID_{\text{on-shelf}}$ in η to search the corresponding auction information in BB_{AC}.

(3) After finding the corresponding auction, the *AC* compares *price* in η with *Basic_price* of the auction. If *price* is not larger than *Basic_price*, the *AC* ignores

this bid and returns to the waiting state; otherwise, the AC updates $Basic\_price = price$ and generates a unique identity ID_{trans} for this transaction. Then, the AC generates a nonce N_6 and computes $k_{dis} = H(SK_{AC} \parallel N_6)$ and $dis = E_{k_{dis}}(C_2, R_3)$. Let $(ID_{\text{trans}}, date, price, dis, N_6)$ be M_2; the AC then generates $r_2 \in_R Z_q^*$ and computes $R_4 = r_2 \cdot P \bmod q$ and $S_2 = r_2^{-1} \cdot (M_2 - SK_{AC} \cdot (R_4)_x) \bmod q$. Waiting τ minutes, $\tau \in_R (0, t)$, the AC publishes (M_2, R_4, S_2) on BB_{AC} such that anyone can verify it by checking if equation $S_2 \cdot R_4 + (R_4)_x \cdot PK_{AC} \equiv M_2 \cdot P \bmod q$ holds.

(4) The AC then finds out the corresponding P_{A_j} of the auctioneer's id_{A_j} of this auction in the database. The AC computes the symmetrical key shared with A_j : $K_{AA} = SK_{AC} \cdot P_{A_j} \bmod q$. Then, the AC generates a session key shared between B_i and A_j as $K_{\text{session}} = H(ID_{\text{trans}}, price, date, (K_{AA})_x)$. The AC generates a receipt for B_i as $RECEIPT = (K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$ and sends $E_{K_{AB}}(RECEIPT, K_{\text{session}}, id_{B_i}, N_7, ID_{\text{trans}})$ to B_i, where N_7 is a nonce.

(5) Upon receiving the message from the AC, B_i decrypts it with K_{AB} and checks the id_{B_i} and N_7 in it. If id_{B_i} is correct and the nonce is fresh, B_i stores $(RECEIPT, K_{\text{session}}, ID_{\text{trans}})$.

(6) After the *DeadLine* arrives, the AC closes the auction and determines the bidder with the highest price as the winner. The AC determines from the corresponding winning bid information on BB_{AC} whose *price* of the bid is equal to $Basic\_price$. Let $M_3 = (ID_{\text{trans}}, price, date, H(SK_{AC} \parallel N_6) \cdot P, dis, N_6)$; the AC chooses $r_3 \in_R Z_q^*$ and computes $R_5 = r_3 \cdot P \bmod q$ and $S_3 = r_3^{-1} \cdot (M_3 - SK_{AC} \cdot (R_5)_x) \bmod q$. Then, the AC publishes the winning information (M_3, R_5, S_3) on BB_{AC}. Anyone can verify the winner's information by checking $S_3 \cdot R_5 + (R_5)_x \cdot PK_{AC}? \equiv M_3 \cdot P (\bmod q)$.

5.5. Product Claiming Phase. After the AC publishes the auction result, every bidder can check whether he/she is a winner by checking whether his/her transaction identity ID_{trans} is equal to the winner's ID_{trans}. If a bidder B_i wins the auction, then he/she can claim the product from the auctioneer A_j using the following.

B_i computes $E_{K_{\text{session}}}(RECEIPT \cdot P \bmod q, N_8)$ and sends it to A_j, where N_8 is a nonce. Upon receiving the claiming request, A_j computes $K_{AA} = pw_{A_j} \cdot PK_{AC}$ firstly. Then, he/she browses the BB_{AC} and computes the session key $K_{\text{session}} = H(ID_{\text{trans}}, price, date, (K_{AA})_x)$ shared with the winner and extracts $(RECEIPT \cdot P \bmod q, N_8)$ by decrypting the ciphertext sent from B_i using K_{session}. Then, A_j checks the freshness of N_8. If N_8 is fresh, A_j verifies the validity of $RECEIPT$ by checking $RECEIPT \cdot P - (K_{AA})_x \cdot P? \equiv M_2 \cdot H(SK_{AC} \parallel N_6) \cdot P \bmod q$. Note that M_2 and $H(SK_{AC} \parallel N_6) \cdot P$ can be found in the winner's information on BB_{AC}. If the equation holds, A_j is convinced that B_i is the winner and sends the product to him/her.

5.6. Dispute Phase. In our proposal, the AC has the capability of tracing the whole transaction and identifying the participants in an auction if necessary. If a dispute has occurred, anyone who participates in an auction can submit a dispute request to the AC in this phase. In the following, we consider two situations to explain the dispute phase.

Situation 1 (auctioneer appeals to AC). If a nonpaying auction happened, which means that the winner B_i did not contact the auctioneer A_j after he/she won the auction, A_j can appeal to the AC and the AC will trace the transaction to deal with this dispute.

A_j browses BB_{AC} and finds the winner's ID_{trans} firstly. Then, A_j selects a nonce N_9 and computes $K_{AA} = pw_{A_j} \cdot PK_{AC}$, $e_3 = E_{K_{AA}}(ID_{\text{trans}}, id_{A_j}, N_9)$, $h_3 = H(e_3, K_{AA})$, $C_3 = (h_3 \cdot PK_{AC})_x + id_{A_j} \bmod q$, and $R_6 = h_3 \cdot P \bmod q$. Subsequently, A_j sends (C_3, R_6, e_3) to AC. Once receiving the dispute request from A_j, the AC retrieves $id_{A_j} = C_3 - (R_6 \cdot SK_{AC})_x \bmod q$ and checks the validation of id_{A_j} in the database. If it is valid, then the AC computes $K_{AA} = (H(SK_{AC}) \oplus v_{A_j}) \cdot SK_{AC} \bmod q$ and decrypts e_3 to extract $ID_{\text{trans}}, id_{A_j}$, and N_9. The AC checks the freshness of N_9. If the nonce is fresh, the AC checks if id_{A_j} in C_3 equals id_{A_j} in e_3. If they are not equal, the AC terminates operation immediately; otherwise, the AC searches the ID_{trans} on BB_{AC} and traces the transaction history to determine whether the problem really exists. If the AC ascertains that the problem claimed by A_j is true, the AC then finds out the corresponding dis and N_6 among the winner's information from BB_{AC}. Afterwards, the AC computes $k_{dis} = H(SK_{AC} \parallel N_6)$ and extracts (C_2, R_3) from dis. Finally, the AC can retrieve the dishonest winner's identity as $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$ and publish it.

Situation 2 (bidder appeals to AC). If the auctioneer A_j of an auction refuses to send the product to the winner B_i, B_i also can appeal to the AC.

Firstly, B_i computes $K_{AB} = pw_{B_i} \cdot PK_{AC} \bmod q$, $e_4 = E_{K_{AB}}(RECEIPT \parallel ID_{\text{trans}} \parallel N_{10} \parallel id_{B_i})$, $h_4 = H(e_4, K_{AB})$, $C_4 = (h_4 \cdot PK_{AC})_x + id_{B_i} \bmod q$, and $R_7 = h_4 \cdot P \bmod q$, where N_{10} is a nonce. B_i then sends (C_4, R_7, e_4) to the AC. Upon receiving the dispute request from B_i, the AC retrieves $id_{B_i} = C_4 - (R_7 \cdot SK_{AC})_x \bmod q$ and checks if id_{B_i} exists in the database. If it exists, the AC then computes $K_{AB} = (H(SK_{AC}) \oplus v_{B_i}) \cdot SK_{AC} \bmod q$ and decrypts e_4 to extract $RECEIPT, ID_{\text{trans}}, N_{10}$, and id_{B_i}. The AC checks the freshness of N_{10}. If the nonce is fresh, the AC checks if id_{B_i} in C_4 equals id_{B_i} in e_4. If they are not equal, the AC terminates the operation immediately; otherwise, the AC then finds the corresponding N_6 and id_{A_j} of ID_{trans} on BB_{AC} and the corresponding v_{A_j} of id_{A_j} in the database. Then, the AC calculates $P_{A_j} = H(SK_{AC}) \oplus v_{A_j}$ and $K_{AA} = P_{A_j} \cdot SK_{AC} \bmod q$. The AC checks the validation of RECEIPT by verifying $RECEIPT ? = (K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$. Note that M_2 and N_6 can be found in the winner's information on BB_{AC}. If RECEIPT is valid, the AC traces the transaction history to determine whether the problem really exists. If the AC

ascertains that the problem claimed by B_i is true, the AC publishes the identity of the dishonest auctioneer A_j.

6. BAN Logic Analysis

In this part, we will use BAN logic to analyze the authentication accuracy. Burrows-Abadi-Needham logic (also known as the BAN logic) is a set of rules for defining and analyzing information exchange protocols. Specifically, BAN logic helps its users to determine whether exchanged information is trustworthy, secured against eavesdropping, or both [30]. The notations of BAN logic are as follows:

$P \mid\equiv X$: P believes X.

$P \triangleleft X$: P sees X.

$P \mid\sim X$: P once said X.

$\mid \xrightarrow{PK} P$: P has PKas a public key.

$\#(X)$: the formula X is fresh.

$P \mid\Rightarrow X$: Phas jurisdiction over X.

$P \overset{K}{\longleftrightarrow} Q$: P and Q have a shared key K for communication.

$\{X\}_K$: the formula X encrypted under key K.

We mainly focus on the proof of on-shelf phase and bidding phase.

6.1. On-Shelf Phase. We idealized the on-shelf phase as below:

(I1) $A_j \rightarrow AC : \{id_{A_j}\}_{PK_{AC}}, \{\gamma, N_3, id_{A_j}\}_{K_{AA}}$.

(I2) $AC \rightarrow A_j : \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}_{K_{AA}}$.

In this phase, we want to ensure that AC believes in the on-shelf information γ and A_j believes in the $ID_{\text{on-shelf}}$ transferred from AC, so we can conclude the two goals below:

(G1) $AC \mid\equiv \gamma$.

(G2) $A_j \mid\equiv ID_{\text{on-shelf}}$.

We need to assume that AC believes the nonce N_3 is fresh and A_j believes the nonce N_4 is fresh; AC believes that A_j does not send fake γ and id_{A_j}, so if A_j believes γ and id_{A_j} are true then AC will believe them too; K_{AA} is computed by AC self, so A_j believes that K_{AA} is shared between him/her and AC; and A_j believes that AC does not send fake $ID_{\text{on-shelf}}$, so if AC believes $ID_{\text{on-shelf}}$ is true then A_j will believe them too. All 6 assumptions are listed as below:

(A1) $AC \mid\equiv \#(N_3)$.

(A2) $AC \mid\equiv A_j \mid\Rightarrow \gamma$.

(A3) $AC \mid\equiv A_j \mid\Rightarrow id_{A_j}$.

(A4) $A_j \mid\equiv (AC \overset{K_{AA}}{\longleftrightarrow} A_j)$.

(A5) $A_j \mid\equiv \#(N_4)$.

(A6) $A_j \mid\equiv AC \Rightarrow ID_{\text{on-shelf}}$.

Proof. For goal (G1), our deduction is shown as the following formulas:

(F1) $AC \triangleleft \{id_{A_j}\}_{PK_{AC}}$.

(F2) $AC \triangleleft id_{A_j}$ (message decryption rule).

(F3) $AC \mid\equiv id_{A_j}$(our hypothesis, we will prove that it is true later).

(F4) $AC \mid\equiv (AC \xleftrightarrow{K_{AA}=SK_{AC}\cdot P_{A_j}} A_j)$ ((F3), the computing property of believing in operator).

(F5) $AC \triangleleft \{\gamma, N_3, id_{A_j}\}_{K_{AA}}$.

(F6) $AC \mid\equiv A_j \mid\sim \{\gamma, N_3, id_{A_j}\}$ ((F4), (F5), message meaning rule).

(F7) $AC \mid\equiv \#(\{\gamma, N_3, id_{A_j}\})$ ((A1), freshness propagation rule).

(F8) $AC \mid\equiv A_j \mid\equiv \{\gamma, N_3, id_{A_j}\}$ ((F6), (F7), nonce verification rule).

(F9) $AC \mid\equiv A_j \mid\equiv id_{A_j}$ ((F8), believing rule).

(F10) $AC \mid\equiv id_{A_j}$ ((F9), (A3), jurisdiction rule, our hypothesis in (F3) is proved).

(F11) $AC \mid\equiv A_j \mid\equiv \gamma$ ((F8), believing rule).

(F12) $AC \mid\equiv \gamma$ ((F11), (A3), jurisdiction rule).

According to formula (F12), the proof of (G1) is completed.

For goal (G2), our deduction is shown as following formulas:

(F13) $A_j \triangleleft \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}_{K_{AA}}$.

(F14) $A_j \mid\equiv AC \mid\sim \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}$ ((F13), (A4), message meaning rule).

(F15) $A_j \mid\equiv \#(\{ID_{\text{on-shelf}}, N_4, id_{A_j}\})$ ((A5), freshness propagation rule).

(F16) $A_j \mid\equiv AC \mid\equiv \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}$ ((F14), (F15), nonce verification rule).

(F17) $A_j \mid\equiv AC \mid\equiv ID_{\text{on-shelf}}$ ((F16), believing rule).

(F18) $A_j \mid\equiv ID_{\text{on-shelf}}$ ((F16), (A6), jurisdiction rule).

According to formula (F18), the proof of (G2) is completed. □

6.2. Bidding Phase. Our bidding phase can be idealized as below:

(I3) $B_i \rightarrow AC : \{id_{B_i}\}_{PK_{AC}}, \{\eta, N_5, id_{B_i}\}_{K_{BA}}$.

(I4) $AC \rightarrow B_i : \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}_{K_{AB}}$.

In bidding phase, we want to ensure that AC can be convinced that bidding information η transferred from B_i is true; additionally, B_i should be convinced that $RECEIPT$ and K_{session} transferred from AC are true. So the three goals below can be concluded:

(G3) $AC \mid\equiv \eta$.

(G4) $B_i \mid\equiv RECEIPT$.

(G5) $B_i \mid\equiv K_{\text{session}}$.

We need to assume that AC believes N_5 is fresh and B_i believes N_7 is fresh; if B_i believes η and id_{B_i} are true, AC believes in them too because they are generated by B_i; similarly, if AC believes in $RECEIPT$ and K_{session} then B_i believes in them too; because K_{AB} is generated by B_i, B_i believes that he/she shared K_{AB} with AC. All assumptions are listed as below:

(A7) $AC \mid\equiv \#(N_5)$.

(A8) $AC \mid\equiv B_i \mid\Rightarrow \eta$.

(A9) $AC \mid\equiv B_i \mid\Rightarrow id_{B_i}$.

(A10) $B_i \mid\equiv (AC \overset{K_{AB}}{\longleftrightarrow} B_i)$.

(A11) $B_i \mid\equiv \#(N_7)$.

(A12) $B_i \mid\equiv AC \Rightarrow RECEIPT$.

(A13) $B_i \mid\equiv AC \Rightarrow K_{\text{session}}$.

Proof. For goal (G3), our deduction is shown as the following formulas:

(F19) $AC \triangleleft \{id_{B_i}\}_{PK_{AC}}$.

(F20) $AC \triangleleft id_{B_i}$ (message decryption rule).

(F21) $AC \mid\equiv id_{B_i}$ (our hypothesis, we will prove that it is true later).

(F22) $AC \mid\equiv (AC \overset{K_{AB}=SK_{AC}\cdot P_{B_i}}{\longleftrightarrow} B_i)$ ((F21), operator believes in property).

(F23) $AC \triangleleft \{\eta, N_5, id_{B_i}\}_{K_{BA}}$.

(F24) $AC \mid\equiv B_i \mid\sim \{\eta, N_5, id_{B_i}\}$ ((F22), (F23), message meaning rule).

(F25) $AC \mid\equiv \#(\{\eta, N_5, id_{B_i}\})$ ((A7), freshness propagation rule).

(F26) $AC \mid\equiv B_i \mid\equiv \{\eta, N_5, id_{B_i}\}$ ((F24), (F25), nonce verification rule).

(F27) $AC \mid\equiv B_i \mid\equiv id_{B_i}$ ((F26), believing rule).

(F28) $AC \mid\equiv id_{B_i}$ ((F27), (A9), jurisdiction rule, our hypothesis in (F21) is proved).

(F29) $AC \mid\equiv B_i \mid\equiv \eta$ ((F26), believing rule).

(F30) $AC \mid\equiv \eta$ ((F29), (A8), jurisdiction rule).

According to formula (F30), the proof of (G3) is completed.

For goal (G4), our deduction is shown as the following formulas.

(F31) $B_i \triangleleft \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}_{K_{AB}}$.

(F32) $B_i \mid\equiv AC \mid\sim \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}$ ((F31), (A10), message meaning rule).

(F33) $B_i \mid\equiv \#(\{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\})$ ((A11), freshness propagation rule).

(F34) $B_i \mid\equiv AC \mid\equiv \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}$ ((F32), (F33), nonce verification rule).

(F35) $B_i \mid\equiv AC \mid\equiv RECEIPT$ ((F34), believing rule).

(F36) $B_i \mid\equiv RECEIPT$ ((F34), (A12), jurisdiction rule).

According to formula (F36), the proof of (G4) is completed. Finally, the goal of (G5) is deduced by the following formulas:

(F37) $B_i \mid\equiv AC \mid\equiv K_{\text{session}}$ ((F34), believing rule).

(F38) $B_i \mid\equiv RECEIPT$ ((F37), (A13), jurisdiction rule).

The proof of (G5) is completed. □

According to our proofs, we can see that our protocol completes the mutual authentication between users and AC. In on-shelf phase, AC can be sure that on-shelf information is sent from legal user and auctioneer can be sure that $ID_{\text{on-shelf}}$ is transmitted from AC; in bidding phase, AC can be sure that bidding information is sent from legal user and bidder can be sure that receipt and session key are generated and sent by AC. As the results, our protocol achieves authentication accuracy.

7. Security Analysis

In this part, we analyze the security of our proposal. We assume that an adversary can eavesdrop on public communications on the Internet and read the information on BB_{AC}; however, he/she cannot read ciphertexts without getting the encrypting keys. Of course, the adversary cannot calculate the keys from ciphertexts. All roles in the system except for the AC can conspire to do something illegally for their benefit. In the following, we provide different scenarios to explain our security defense.

(1) Impersonation Attack. It is impossible for an adversary to impersonate a user. When a user wants to communicate with the AC, he/she should send his/her identity ciphertext (C_n, R_n, e_n) to the AC. The AC retrieves and checks the user's identity from this message. We can see that the adversary cannot generate a legal (C_n, R_n, e_n) to pass this verification without the knowledge of the AC's private key SK_{AC} and the key K_{AU} shared between the AC and the user. Even if the adversary eavesdrops on the other users' communications and stores their (C_n, R_n, e_n), he/she still cannot use these to do anything because the AC will check if the identity included in C_n equals that in e_n. The only method for the adversary is to replay the whole (C_n, R_n, e_n); however, this will not work because the AC can check the freshness of the nonce in e_n.

(2) Collusion Attack. In some cases, bidders and auctioneers may conspire to break the secret of the AC. For example, a bidder may send $RECEIPT$ instead of $RECEIPT \cdot P$ to an auctioneer. However, with this knowledge, they can only compute $H(SK_{AC} \parallel N_6) = (RECEIPT - (K_{AA})_x)/M_2$, where K_{AA} can be computed by the auctioneer and M_2 can be found in BB_{AC}. Even obtaining $H(SK_{AC} \parallel N_6)$, they cannot compute SK_{AC} because $H(\cdot)$ is a one-way hash function.

(3) Forging Receipt. Firstly, we consider the case that a malicious bidder wants to forge a receipt to impersonate the winner of an auction by himself/herself. Because $RECEIPT =$

$(K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$, the malicious bidder obviously cannot forge a valid receipt without knowing the *AC*'s private key SK_{AC} and the key K_{AA} shared between the *AC* and the auctioneer. Secondly, it is also impossible for the winner and auctioneer to conspire and create the receipt of another auction. As mentioned in the Collusion Attack, they only can compute $H(SK_{AC} \parallel N_6)$. However, for different bidders, N_6's are different, so $H(SK_{AC} \parallel N_6)$ are different in different auctions. Obviously, they cannot create a useful receipt of the other auction with $H(SK_{AC} \parallel N_6)$ by collusion. Besides, it is meaningless for the winner and auctioneer of an auction to forge a receipt of the auction.

(4) Denial of Service Attack. If an adversary wants to mount a deny of service attack on a bidder (B_i) such that he/she cannot claim the product in the product claiming phase when he/she wins an auction, the adversary should generate a fake receipt and send it to the bidder during the bidding phase. However, the receipt transmitted to B_i needs to be encrypted with B_i's identity by the symmetrical key shared between B_i and the *AC* as $E_{K_{AB}}(RECEIPT, K_{\text{session}}, id_{B_i}, N_7, ID_{\text{trans}})$. After receiving the ciphertext, the bidder should decrypt it and check if his identity id_{B_i} is correct. So it is impossible for an adversary to deceive a bidder to accept a fake receipt.

(5) Publishing Fake Information. In our proposal, all published information, including new auction information in the on-shelf phase and bidding information and winner information in the bidding phase are signed by the private key of *AC*, so anyone can verify the published information with *AC*'s public key. Let us consider a situation: if a bidder wants to win an auction, he/she may try to publish a fake bidding message with a very high bidding price on BB_{AC} so that no one else wants to compete with him/her. However, he/she cannot be successful without knowing the private key of the *AC*, because every user can check the signature of the *AC* on this message.

(6) Privacy Preserving. In our scheme, an adversary cannot link any information together to analyze a user's privacy. Only auctioneers' identities need to be published in BB_{AC} for enhancing users' willingness to bid. All packets transmitted in the public channel are encrypted, so the adversary cannot know whether these packets can be linked to certain information published on BB_{AC} even if they captured these packets. It is noteworthy that, in our schemes, we use a delay operation in bidding and on-shelf phases for increasing difficulty of linkage of different messages. For example, if the adversary detects that Alice sends an on-shelf message to the *AC* and the *AC* publishes an auction advertisement on the website including the auctioneer's identity immediately, the adversary would ascertain that Alice is the auctioneer and obtain Alice's identity. If we delay publishing the auction message several minutes later (i.e., the *AC* publishes it with other auction information together), then the adversary cannot distinguish a specific user from them. The larger the τ we use, the stronger the privacy preserving function is. However, we should balance the privacy preserving and user experience. Moreover, in the bidding phase, a bidder's identity can be retrieved by $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$, where C_2 and R_3 are involved in the information published on BB_{AC}. However, these parameters are protected by *AC*'s private key as $dis = E_{k_{dis}}(C_2, R_3)$ before being published on BB_{AC}, where $k_{dis} = H(SK_{AC} \parallel N_6)$, such that no one can extract id_{B_i} except for the *AC*. In the product claiming phase, an auctioneer cannot know the winner's identity from $RECEIPT \cdot P$ but can only know the validity of the receipt. As indicated in the results, our scheme indeed achieves privacy preserving.

(7) Accessing to the Database of the AC. In the definition of a network model (Section 4), we have mentioned that we do not consider the database of the *AC* to be completely secure in our proposal. An adversary may access the messages stored in it by certain methods. However, accessing the database would not destroy the security of the whole system. Furthermore, the *AC*'s private key should be kept in a trusted place, such as a bank. There are mainly four data sets maintained in the *AC*: the first is (id_U, v_U) generated in the registration phase, the second is $(ID_{\text{product}}, Basic\_price, Deadline, ID_{\text{on-shelf}}, id_{A_j}, R_2, S_1)$ generated in the on-shelf phase, and the final are $(ID_{\text{trans}}, date, price, dis, N_6, R_4, S_2)$ and $(ID_{\text{trans}}, price, date, H(SK_{AC} \parallel N_6) \cdot P, dis, N_6)$ generated in the bidding phase. Only the first data set is unpublished, so we only consider the safety of the first data set here. If a hacker reads the content of the first data set, he/she only can obtain all users' identities and their partial keys (P_Us) which have been protected by *AC*'s private key as the form $v_U = P_U \oplus H(SK_{AC})$. With only protected partial keys, the hacker cannot compute the symmetrical key K_{AU} ($K_{AU} = pw_u \cdot PK_{AC} \bmod q = SK_{AC} \cdot P_U \bmod q$) shared between users and the *AC*, because the private key of the *AC* is kept secret. Furthermore, he/she cannot retrieve a user's password from the partial key based on ECDLP. So we can see that accessing the database does not destroy the safety of the whole system.

(8) Password Guessing Attack. In our scheme, users' passwords are needed to protect system security. However, because the user's password is not very long and there is a certain routine mode, attackers may try to mount the password guessing attack to the password-based communication schemes. To prevent this attack, in our scheme, users' passwords will not be transmitted or stored in plaintext. As mentioned in the registration phase (Section 5.2), the password related information P_U is transformed to ciphertext $C = (r_0 \cdot PK_{AC} + P_U) \bmod q$ before transferring and is stored as the format of $v_U = P_U \oplus H(SK_{AC})$ in *AC*'s database. Obviously, even an attacker collects ciphertext or v_U by eavesdropping the whole communication channel or intruding *AC*'s database; he/she can not obtain any user's password by password guessing attack without the knowledge of *AC*'s secret key SK_{AC}.

Table 1 ("○" means fully achieved, "△" means partially achieved, and "×" means not achieved) shows the security comparison among our proposal and three other schemes: Chang et al.'s [14], Xiong et al.'s [13], and Chung et al.'s [18]. As shown in Table 1, our scheme fulfills the most secure properties of an electronic auction system. In the analysis of "*privacy preserving,*" our scheme can achieve the

Table 1: Comparison in security properties of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Anonymity	○	○	○	○
Easy revocation	○	○	△	△
Fairness	○	○	○	○
Nonrepudiation	○	○	○	○
Onetime registration	○	○	○	○
Traceability	○	○	○	○
Unforgeability	○	○	○	○
Unlinkability	○	×	○	○
Verifiability	○	△	×	○
On-shelf mechanism	○	○	×	×
Without secure channel	○	○	×	×
Password guessing attack defending	○	○	○	○

Table 2: Comparison in computational cost of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	8PM + H	9PM	4PM + 2H	5PM + 3H
On-shelf phase	7PM + 2E + 2D + 2H	10PM + 2E + 2D + 2H	×	×
Bidding phase	$(10n + 2)$PM + $3n$E + $2n$D + $5n$H	$12n$PM + $2n$E + $2n$D	$(4n + 1)$PM + 3BM + 2H	$(8n + 2)$PM + $(3n + 1)$H
Product claiming phase	2PM + E + D + 1H	6PM	×	×
Dispute phase (bidder/auctioneer)	6PM + E + 1D + 3H/6PM + E + 2D + 3H	3PM + 2H/3PM + E + D	$0.5n$BM/×	×

properties of anonymity and unlinkability; in the analysis of "*Publishing Fake Information,*" our scheme can achieve the properties of fairness and verifiability; and in the analyses of "*Impersonation Attack*, *Forging Receipt*, and *Denial of Service Attack,*" our scheme can achieve the property of unforgeability. Similar to [13, 14, 18], the bidding information should be published on an open website, so the bidder can check if his/her bid appeared on it to ensure fairness. Based on the digital signature, bidders cannot deny their bids so nonrepudiation is achieved in four proposals. By providing the dispute mechanism, our scheme achieves the properties of easy revocation, nonrepudiation, and traceability. Furthermore, our scheme overcomes the security problem (leakage of unlinkability) and meanwhile inherits the advantage of without a secure channel in Chang et al.'s scheme [14]. It is noteworthy that, compared to [14], our protocol needs to store a little of the verification table in the server side. However, as we analyzed before, leakage of these verification tables will not influence system security, so we say that our protocol achieves partial property of no verification table.

With regard to verifiability, in the bidding phase of Chang et al.'s proposal [14], only the winner's bidding information will be signed by the *AC* and can be publicly verified, which may lead attackers to modify any new bidding information with a higher price such that no one else wants to bid for it further. So verifiability is not fully achieved in [14]. As [14] analyzed, the designed revocation function in [13, 18] only partially achieved easy revocation. Our protocol and [14] have a secure on-shelf phase compared to [13, 18]. Our protocol does not need a secure channel to achieve verification in any phase similar to [14]. However, in [13, 18], a secure channel is needed to assist in the completion of the whole protocol.

8. Performance Analysis

In this section, we compare the cost of our scheme with the other three schemes [13, 14, 18] with regard to computational cost, traffic, and communication rounds.

Table 2 shows the comparison of computational cost, where "PM" means a point multiplication operation of ECC, "BM" means a bilinear mapping which can be implemented on ECC such as Weil pairing, "E" means a symmetrical encryption, "D" means a symmetrical decryption, "H" means a hash function, and "n" means the number of bidders in an auction.

As we know, a bilinear map is an expensive operation; the more it is used, the more computation cost is needed. Compared with point multiplication, the hash function only consumes negligible computation cost. Symmetrical cryptographic operation is also cheaper than point multiplication. From the comparison in Table 2, we can see that our proposal needs less computation cost compared with Chang et al.'s proposal [14] in each phase, except for the dispute phase.

Table 3: Comparison in traffic of the three schemes (B: bytes).

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	82 B	192 B	108 B	128 B
On-shelf phase	144 B	266 B	×	×
Bidding phase	$176n$ B	$290n$ B	$(96n + 120)$ B	$(119n + 248)$ B
Product claiming phase	48 B	149 B	×	×
Dispute phase (auctioneer/bidder)	112 B/128 B	37 B/85 B	48 B/×	×

Table 4: Comparison in communication rounds of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	2	2	1	1
On-shelf phase	3	3	×	×
Bidding phase	$2n$	$2n$	1	$n + 2$
Product claiming phase	1	1	×	×
Dispute phase (auctioneer/bidder)	1/1	1/1	1	×

However, in reality, the first four phases are more frequently executed.

To make the traffic comparison, we determine the elliptic curve operation as NistP 192 [24], hash function as SHA-256 [31], and symmetrical encryption/decryption as AES-256 [32]. The length of the user's identity is 4 bytes, product identity is 5 bytes, transaction identify is 5 bytes, *GID* in [14] is 5 bytes, *price* is 5 bytes, and *Deadline* is 4 bytes. Note that we ignore the length of nonce transmitted in communications. As shown in Table 3, we can see that our protocol has a traffic advantage compared with related works [13, 14, 18], except for dispute phase of Chang et al.'s scheme [14]. The main reason for this is that our scheme finished the key establishment in the registration phase for saving the traffic cost of public keys and certificates. Comparing with Chang et al.'s proposal, though our dispute phase needs more traffic, this phase is not often used compared with the other four phases. As a result, we believe that our proposal has traffic advantages still. Finally, we compare the communication rounds of our scheme and the related schemes in Table 4. As shown in this table, our protocol needs the same communication rounds as those in Chang et al.'s protocol [14].

9. Conclusions

In this paper, we pointed out that Chang et al.'s protocol does not fulfill the unlinkability such that users' privacy will be leaked by linking different messages together. Considering the importance of privacy preservation, we proposed a novel electronic English auction system. By using symmetrical encryptions and fewer ECC operations, our protocol can not only improve the security but also reduce the system cost. In the registration phase, we connect a user's identity and the corresponding symmetrical key so that the *AC* can compute the shared symmetrical key easily with the user's identity. Furthermore, each user's identity is encrypted with the *AC*'s public key so it cannot be revealed without the knowledge of the *AC*'s private key. In the bidding phase, we improve the efficiency of generating bidding receipts. In our proposal, we fulfill the verifiability of all messages published on the *AC*'s website; however, Chang et al.'s proposal only achieved this feature partially. We used BAN logic to prove that our protocol indeed realizes mutual authentication. The security analysis and performance analysis show that our protocol fulfills more security properties and is more efficient for implementation compared with recent works.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Statista.com, *Number of Worldwide Internet Users from 2000 to 2014 (in millions)*, Statista, New York, NY, USA, 2015, http://www.statista.com/statistics/273018/number-of-internet-users-worldwide/.

[2] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein, "Auctions and bidding: a guide for computer scientists," *ACM Computing Surveys*, vol. 43, no. 2, article 10, 2011.

[3] C. Cachin, "Efficient private bidding and auctions with an oblivious third party," in *Proceedings of the 6th ACM Conference on Computer and Communication Security (CCS '99)*, pp. 120–127, Singapore, November 1999.

[4] K. Q. Nguyen and J. Traoré, "An online public auction protocol protecting bidder privacy," in *Proceedings of the 5th Australasian Conference on Information Security and Privacy (ACISP '00), Brisbane, Australia, July 2000*, vol. 1841 of *Lecture Notes in Computer Science*, pp. 427–442, Springer, 2000.

[5] K. Omote and A. Miyaji, "A practical English auction with one-time registration," in *Information Security and Privacy: 6th*

Australasian Conference, ACISP 2001 Sydney, Australia, July 11–13, 2001 Proceedings, vol. 2119 of *Lecture Notes in Computer Science*, pp. 221–234, Springer, Berlin, Germany, 2001.

[6] C.-C. Chang and Y.-F. Chang, "Efficient anonymous auction protocols with freewheeling bids," *Computers & Security*, vol. 22, no. 8, pp. 728–734, 2003.

[7] R. Jiang, L. Pan, and J.-H. Li, "An improvement on efficient anonymous auction protocols," *Computers & Security*, vol. 24, no. 2, pp. 169–174, 2005.

[8] Y.-F. Chang and C.-C. Chang, "Enhanced anonymous auction protocols with freewheeling bids," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 353–358, April 2006.

[9] K. Suzuki and M. Yokoo, "Secure multi-attribute procurement auction," in *Information Security Applications: 6th International Workshop, WISA 2005, Jeju Island, Korea, August 22–24, 2005, Revised Selected Papers*, vol. 3786 of *Lecture Notes in Computer Science*, pp. 306–317, Springer, Berlin, Germany, 2005.

[10] D.-H. Shih, C.-H. Cheng, and J.-C. Shen, "A secure protocol of reverse discriminatory auction with bid privacy," in *Proceedings of the 6th International Conference on the Management of Mobile Business (ICMB '07)*, 52 pages, Toronto, Canada, July 2007.

[11] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. Thorpe, "Practical secrecy-preserving, verifiably correct and trustworthy auctions," *Electronic Commerce Research and Applications*, vol. 7, no. 3, pp. 294–312, 2008.

[12] H. Xiong, Z. Qin, and F. Li, "An anonymous sealed-bid electronic auction based on ring signature," *International Journal of Network Security*, vol. 8, no. 3, pp. 235–242, 2009.

[13] H. Xiong, Z. Chen, and F. Li, "Bidder-anonymous English auction protocol based on revocable ring signature," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7062–7066, 2012.

[14] C. C. Chang, T. F. Cheng, and W. Y. Chen, "A novel electronic english auction system with a secure on-shelf mechanism," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 657–668, 2013.

[15] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan, "Robust, privacy protecting and publicly verifiable sealed-bid auction," in *Information and Communications Security: 4th International Conference, ICICS 2002 Singapore, December 9–12, 2002 Proceedings*, vol. 2513 of *Lecture Notes in Computer Science*, pp. 147–159, Springer, Berlin, Germany, 2002.

[16] K. Sako, "An auction protocol which hides bids of losers," in *Public Key Cryptography: Third International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2000, Melbourne, Victoria, Australia, January 18–20, 2000. Proceedings*, vol. 1751 of *Lecture Notes in Computer Science*, pp. 422–432, Springer, Berlin, Germany, 2000.

[17] H. Kikuchi, M. Hakavy, and D. Tygar, "Multi-round anonymous auction protocols," *IEICE Transactions on Information and Systems*, vol. 82, no. 4, pp. 769–777, 1999.

[18] Y.-F. Chung, Y.-T. Chen, T.-L. Chen, and T.-S. Chen, "An agent-based English auction protocol using Elliptic Curve Cryptosystem for mobile commerce," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9900–9907, 2011.

[19] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85 Proceedings*, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, Berlin, Germany, 1985.

[20] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[21] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems—CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11–13, 2004. Proceedings*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 119–132, Springer, Berlin, Germany, 2004.

[22] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, New York, NY, USA, 2008.

[23] J. W. Bos, A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3–7, 2014, Revised Selected Papers*, vol. 8437 of *Lecture Notes in Computer Science*, pp. 157–175, Springer, Berlin, Germany, 2014.

[24] Z. Tan, Z. Liu, and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP," in *MM Research Preprints*, no. 21, pp. 212–217, MMRC, AMSS, Chinese Academy of Sciences, Beijing, China, 2002.

[25] H.-T. Liaw, W.-S. Juang, and C.-K. Lin, "An electronic online bidding auction protocol with both security and efficiency," *Applied Mathematics and Computation*, vol. 174, no. 2, pp. 1487–1497, 2006.

[26] C.-C. Chang and T.-F. Cheng, "An efficient proxy raffle protocol with anonymity-preserving," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 772–778, 2009.

[27] E. Pinker, A. Seidmann, and Y. Vakrat, "Using bid data for the management of sequential, multi-unit, online auctions with uniformly distributed bidder valuations," *European Journal of Operational Research*, vol. 202, no. 2, pp. 574–583, 2010.

[28] J.-S. Lee and K.-S. Lin, "An innovative electronic group-buying system for mobile commerce," *Electronic Commerce Research and Applications*, vol. 12, no. 1, pp. 1–13, 2013.

[29] M.-J. Li, J. S. T. Juan, and J. H. C. Tsai, "Practical electronic auction scheme with strong anonymity and bidding privacy," *Information Sciences*, vol. 181, no. 12, pp. 2576–2586, 2011.

[30] M. Burrows, M. Abadi, and R. Needham, "Authentication: a practical study in belief and action," in *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 325–342, March 1988.

[31] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Proceedings of the 10th Annual International Workshop on Selected Areas in Cryptography (SAC '03), Ottawa, Canada, August 2003*, vol. 3006 of *Lecture Notes in Computer Science*, pp. 175–193, Springer, 2003.

[32] F. P. Miller, A. F. Vandome, and J. McBrewster, *Advanced Encryption Standard*, Alphascript, Beau Bassin, Mauritius, 2009.