國立交通大學

資訊科學與工程研究所

碩士論文

行動商務的代理人英式拍賣機制之研究

A Study of Agent-based English Auction Protocols for Mobile Commerce

研 究 生：陳鈺婷

指導教授：黃世昆　教授

中華民國 一百年七月

行動商務的代理人英式拍賣機制之研究
A Study of Agent-based English Auction Protocols for Mobile Commerce

研　究　生：陳鈺婷　　　Student：Yu-Ting Chen

指導教授：黃世昆　　　Advisor：Shih-Kun Huang

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer and Information Science

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

行動商務的代理人英式拍賣機制之研究

學生：陳鈺婷　　　　　　　　　　指導教授：黃世昆　老師

國立交通大學資訊科學與工程研究所碩士班

摘要

由於網路科技的迅速發展以及個人行動電話普遍化的趨勢，使得以行動裝置進行網路拍賣的商機需求，具有相當可期的潛力，因此，在滿足行動裝置需求與限制的條件下，本論文將行動代理人技術與英式拍賣機制的運作予以整合，使競標者透過代理人的方式參與拍賣與競標，並使用英式拍賣機制建置網路拍賣環境，以提供使用者一個安全、公平及有效率的網路拍賣環境。在此機制之下，包含四個參與者，包括註冊單位、代理人管理者、拍賣所管理者、競標者，註冊單位負責註冊與驗證競標者的身分；代理人管理者負責管理或控制所有相關的代理人與核發交易公開金鑰給競標者；拍賣所管理者負責提供拍賣的場所，並維護與主持整個拍賣的運作；競標者則是參與拍賣與出價的買方。該機制符合電子拍賣協定中的安全性：匿名性、可追蹤性、不可陷害性、不可偽造性、不可否認性、公平性、可公開驗證性、在不同拍賣中無關聯性、同一拍賣中有關聯性、投標有效率、單次註冊、容易註銷。同時，為了因應網路環境，需考量競標資訊傳遞過程所耗損的時間成本，因此，本論文以佈告欄的方式供各個管理者公佈競標資訊，並應用橢圓曲線密碼系統，利用其短金鑰、低運算量等特性，力求提升產生金鑰與出價的速度、驗證的效率，並且減少行動裝置的計算量與伺服器的負載量，從而增進網路拍賣系統的便利性。

關鍵字：行動代理人、英式拍賣、橢圓曲線密碼系統、匿名性、公開驗證性

A Study of Agent-based English Auction Protocols for Mobile Commerce

Student：Yu-Ting Chen Advisor：Dr. Shih-Kun Huang

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

Rapid development of the Internet and the extensive use of mobile phones have led to increased potentiality for the application of mobile devices in online auctions. Keeping the needs and constraints of mobile devices in mind, this paper proposes a secure, fair, and effective online auction environment based on the English Auction protocol by integrating its operation with mobile agent technology that allows bidders to participate in online auctions through mobile agents. The protocol consists of four participants: Registration Manager, Agent House, Auction House, and Bidder. The Registration Manager is responsible for Bidders' identity registration and verification. The Agent House manages and controls all related agents and assigns the public transaction keys to Bidders. The Auction House provides a place for auction, and maintains and hosts all necessary operations for an online auction to be taken place. The Bidder can be defined as the buyer who is interested in purchasing items from the auction. The proposed scheme conforms the following security characteristics which satisfying the requirements of an online auction protocol: anonymity, traceability, non-framing, unforgeability, non-repudiation, fairness, public verifiability, unlinkability among various auction rounds, linkability within a single auction round, bidding efficiency, one-time registration, and easy revocation. Given the complex nature of the Internet environment, the consideration is also given to time costs of bidders' data transmission. Thus, this paper makes use of the bulletin board approach for managers to publish the bidding information. Application of Elliptic Curve Cryptosystem for its small key size and low computation amount is exploited to improve the speed of key generation and bidding, and verification efficiency. By cutting down on computation cost for mobile devices and load volumes on the part of servers, convenience of conducting online auctions is enhanced as well.

Keywords: Mobile Agent, English Auction, Elliptic Curve Cryptosystem, Anonymity, Public Verification

Contents

List of Tables

List of Figures

1. Introduction

Over the past few years, with the technical development of the Internet continuing to become more mature in these years, and the prevalence of World Wide Web, the Internet which not only crosses the geographical boundaries but also the applications of Internet have developed toward diversification has become the largest information communication network and media marketing. At the same time, it has also changed business types. Therefore, many kinds of online transactions and auctions have come out. Due to the instantaneity and the interactivity of the Internet, online transactions and auctions present better advantages than traditional businesses do, such as offering the latest business information with less cost or 24-hour services. In this case, the proportion of people using online auction is gradually higher than traditional auction. Moreover, the functions and the effectiveness from online auction are more powerful than those from traditional auction. However, security problems accompany the development of online auction has become one of the important issues for e-business. Due to the improvement of network capability and the growing of the number of smart phones, more and more people perform various Internet activities via mobile devices. One of these activities is online auctions. In order to satisfy the demand for mobile commerce, it is necessary to develop techniques for mobile commerce which satisfy the requirements and the limitations of the mobile devices.

1.1 Background and Motivation

With the popularization of the World Wide Web and its prompt adaptability to trends, traditional auction systems and business transactions have gradually transferred themselves to network platform transactions. Not only does online auction solve general problems like market price discrepancy due to asymmetric information, it makes up for the time, space, and location constraints faced by traditional auctions, and provides for transactions to be conducted in conditions much freer and more public, allowing information much more transparency, and thus much fairer, and equal trading opportunities for the interested [1]. Therefore, online English auction has successfully come to replace the offline traditional auction and is more powerful and capable than the offline traditional auction.

With the rapid development of mobile phones, the demand for mobile commerce has increased as a consequence. Based on market demand considerations, mobile service providers have begun launching mobile commerce service. In order to meet auction security demands in an environment of heterogeneous networks, it is necessary to develop an auction mechanism that satisfies both mobile and constraint needs. Therefore, this paper will present both research and analysis on current auction mechanisms, and by employing mobile agent's mobility and autonomy features, present a safe, fair, and efficient online auction environment.

Today, online auction protocols are applied over the Internet include open auction and sealed-bid auction. Open auction can be subdivided into two types: English Auction and Dutch Auction [1]. The English Auction is to have all participant bidders to place their bid prices on the basis of the reserve price that is preliminarily set by a host. After everything is in place, the host will start the bidding process. As the bid price increases, the person with the highest price will win after the auction time ends. In Dutch Auction, the bidders in the auction will place their bids for lower prices. The auction will be closed when a bidder who is willing to pay the final price [2]. Bidders in English Auction can observe the bidding behavior of their competitors during the entire auction process and make immediate adjustments to how he/she would place the bid. Therefore, it can be highly competitive under this kind of protocol, because the protocol would force the bid price to increase if the goods are desirable. Thus, we can say the English Auction protocol is efficient, because a good protocol can promote the auctioned goods to be sold to someone who is willing to pay at a higher price [3]. As a result of it, the expected return on the goods that are used in English Auction protocol is usually higher than other protocols. So, most auction-based websites, such as eBay and Yahoo! Auctions, use English Auction to operate. Therefore, this paper would primarily focus on how to apply the English Auction protocol for mobile commerce.

1.2 Objective

For the auction model on a mobile-based environment, Kuo-Hsuan Huang proposed a mobile auction agent model (MoAAM) [4], which allows the bidders to participate in online auctions through mobile agents. Huang's scheme employs modular exponentiation operations, but it would increase the processing time for key generation, bidding, and verification. Thus, the paper is proposed to add the concept of Elliptic Curve Cryptosystem (ECC) onto MoAAM, because ECC is able to provide low computation amount and small key size. It would aid to increase the speed of generating keys, bidding, and verification. In terms of reduction of computation on mobile device and workload of the connected server, the proposed scheme will make online auction system become more convenient for users. In order to maintain a fair and secured auction, the following features for security are as below [5]:

(1) Anonymity: During the course of an auction, no one is able to recognize the other bidders' identity.

(2) Traceability: Winner's real identity can be recognized at the end of the auction.

(3) No framing: The identities of all bidders remain independent. No one can falsely claim to be any other bidder who participated in the auction.

(4) Non-forgeability: Nobody is able to forge another one's valid bid price.

(5) Non-repudiation: The winning bidder is unable to deny his/her bid price after the

winning bidder has been announced.

(6) Fairness: All bidding must be conducted in an open and fair manner.

(7) Public verifiability: Anyone can verify the identity and bid prices of the participated bidders.

(8) Unlinkability among various auction rounds: Nobody will know the same bidder’s identity that among different rounds of auction.

(9) Linkability within a single auction round: The bidders can repeatedly place new bid price within a single auction round and can be recognized by other bidders.

(10) Efficient bidding: In order to make the bidding become efficient, times for computation must be minimized.

(11) One-time Registration: The bidder only needs to register once and then he/she can participate in all auctions that are opened.

(12) Easy Revocation: Registration manager can easily revoke someone’s right to bid.

1.3 Thesis Outline

The rest of this paper is organized as follows. Section 2 contains a review of related work on English Auction protocol. In Section 3, we will introduce some backgrounds about proposed scheme, including mathematical theories, principles of cryptography, and mobile agent, etc. Our proposed scheme, which is about how to apply ECC onto MoAAM, is shown in Section 4. In Section 5, a security analysis would be performed to examine our proposed scheme. The final conclusion and recommendations for further studies are given in Section 6.

2. Related Work

In English auction protocol research, Omote and Miyayu [6] were the first to propose the concept of adding the bulletin board for verification in 2001 to construct an English auction protocol that satisfies various security concerns in the auction to reduce computation and server load during the auction. Their method was based on the concept proposed by Nguyen and Traore [7, 8], who utilized group signature technology in English Auction protocol to raise the security level for the bidders. However, for a security reason, Omate and Miyaji's scheme would not publish any bidders' information in order to avoid the possibility of security breach on bidders' privacy. However, this could violate the purpose of anonymity, fairness, and unlinkability among auction rounds, etc., as are required by the English Auction protocol.

Later, Lee et al. [5] made improvements on Omote and Miyaji's method. It allowed bidders' identities and information to be published, yet maintained the feature of unlinkability among auction rounds, i.e. bidders' identities cannot be identified through released information of previous auction rounds.

In 2003, Chang et al. [9] proposed a much simpler and more effective method for anonymity in English Auction. However, Jiang et al. [10] pointed out that Chang et al.'s method was not secure enough to protect bidders' privacy and rights, as bidders have no way to verify whether the shared keys they possess belong to the same auctioneer during the auction. Subsequently, Chang et al. utilized an alias to resolve the situation [11].

2.1 Omote and Miyaju's scheme

In 2001, Omete and Miyaji [6] proposed the use of bulletin board approach for verifying information of bidders to improve the efficient of Group Signature for English Auction protocol proposed by Nguyen and Traore [7, 8]. In the scheme of Omete and Miyaji, there are mainly three entities, Bidder, Registration Manager (RM), and Auction Manager (AM). During the auction, RM manages the correspondence of bidder identity to public key. AM manages a bulletin board, and maintains the operations and hosts the auctions. When a certain bidder is identified after a winner decision procedure or later disputes, AM has only to request RM to identify the bidder to complete the entire auction. Omete and Miyaji claimed that their scheme can satisfy the safety features for English Auction, including: (1) Anonymity, (2) Traceability, (3) No framing, (4) Unforgeability, (5) Fairness, (6) Verifiability, (7) Unlikability among different auctions, (8) Linkability in an auction, (9) Efficiency of bidding, (10) One-time registration, and (11) Easy revocation. The participants and the parameters in the scheme proposed by Omote and Miyaji are explained as below:

【Entity】

Registration Manager (RM) : being responsible to manage and store the correspondence of bidder identity to public key, and send the identity of the bidder to the vendor when a bidder wins out.

Auction Manager (AM) : being responsible to maintain the operations and host the auctions.

Bidder (B) : being participant in an auction that AM holds.

【Notation】

p, q : two large primes, satisfying $q \mid p-1$;

g : an element $g \in Z_p$ with order q;

I : the number of bidders;

i : the index of bidders ($i = 1, \cdots, I$);

B_i : bidder i；

x_i : the secret key of B_i ($x_i \in Z_q$);

y_i : the public key of B_i ($\mathrm{y_i = g^{x_i} \ (mod\, p)}$);

r_i : AM’s random number for B_i ($r_i \in Z_q$);

t_i : a random number of B_i ($t_i \in Z_q$);

T_i : an auction key for B_i;

k : the index of auctions ($k \geq 1$);

X_{AM} : AM’s secret key ($X_{\mathrm{AM}} \in Z_q$);

Y_{AM} : AM’s public key ($Y_{AM} = g^{X_{\mathrm{AM}}} \bmod p, p \in Z_q$);

Enc : $Enc(key, data)$ is a secret key encryption function by using a secret key, *key*;

Enc^j : $Enc^j(key, data)$ is *j*-times encryption by using the same key, i.e. $Enc^j(key, data) = Enc(key, Enc^{j-1}(key, \cdots))$.

The operation of the scheme includes: (1) Initialization, (2) Bidder Registration, (3) AM’s Setup, (4) Bidding, (5) Verifiability, and (6) Winner Announcement. The different stages are described as follows:

【Initialization】

The system parameter settings of the RM and AM are as below:

RM ➪ publishes *p*, *q* and *g* on his bulletin board.

AM ➪ selects a private key $X_{\mathrm{AM}} \in Z_q$, and calculates the corresponding public key

$Y_{AM} = g^{X_{\mathrm{AM}}} \bmod p, p \in Z_q$ and publishes Y_{AM}.

【Bidder Registration】

When a new bidder B*ᵢ* joins in the auction, he/she must follow the following steps in requesting registration from RM:

Step 1: Select a private key $x_i \in Z_q$ and calculates its corresponding registration key y_i:

$$y_i = g^{x_i} \bmod p$$

Step 2: Select a random number $t_i \in Z_q$, named ticket.

Step 3: Send $\{ y_i, t_i \}$ to RM as the registration key, registers his/her identity and proves that he/she knows the discrete logarithm x_i of y_i to the base *g* by showing V_1:

$$V_1 = SK[(\alpha): y_i = g^{\alpha}](m_R)$$

Step 4: After RM authenticates the validity of V_1, RM publishes bidder's registration key $\{ y_i, t_i \}$ on his bulletin board and keeps bidder's name and the corresponding registration in its own database.

【Auction Setup by AM】

Assume that the set of registered bidders is B*ᵢ* ($i = 1, 2, \ldots, I$). When an auction is requested, AM follows the following steps to set up the auction. The auction below is assumed to be at the *k*-th auction:

Step 1: AM calculates a shared secret key $y_i^{x_{AM}} \bmod p$ with each bidder B*ᵢ* ($i = 1, 2, \ldots, I$) by using Diffie-Hellman key-distribution.

Step 2: AM generates the random numbers $r_i \in Z_q(\{r_1, r_2, \ldots, r_I\})$ for each bidder published on RM's bulletin board and keeps the numbers $\{r_1, r_2, \ldots, r_I\}$ secret.

Step 3: AM encrypts t_i to $Enc^k(y_i^{X_{\mathrm{AM}}}, t_i) = Enc(y_i^{X_{\mathrm{AM}}}, Enc^{k-1}(y_i^{X_{\mathrm{AM}}}, t_i))$ in the *k*-time *Enc* encryption function by using a shared key $y_i^{X_{\mathrm{AM}}}$.

Step 4: AM calculates the following auction key T_i for B*ᵢ* using B*ᵢ*'s public key y_i published on RM's bulletin board.

Step 5: AM publishes the shuffled auction key T_i of all bidders on his bulletin board.

【Bidding】

To participate in the *k*-th auction, B*ᵢ* must complete the following steps:

Step 1: Using AM's public key Y_{AM} to calculate $y_i^{X_{\mathrm{AM}}}$ as follows:

$$y_i^{X_{\mathrm{AM}}} = Y_{\mathrm{AM}}^{x_i} \bmod p$$

Step 2: Calculating auction certificate T_i as follows:

$$T_i = (Enc^k(Y_{\mathrm{AM}}^{x_i}, t_i), y_i^{r_i}, g^{r_i})$$

T_i must be verified that they are matched with the information posted on AM's bulletin board.

Step 3: Generates the signature of knowledge V_2 for bid m_i as follows:

$$V_2 = SK[\alpha : y_i^{r_i} = (g^{r_i})^{\alpha}](m_i)$$

Step 4: Finally, send the following bid information $\{m_i, y_i^{r_i}, g^{r_i}, V_2\}$ to AM, thus completing the bidding procedure.

【Verifiability】

After B*ᵢ* publishes the bid information $\{m_i, y_i^{r_i}, g^{r_i}, V_2\}$, anyone can verify them as shown below:

Step 1: Anybody can confirm that a bidder knows surely the discrete logarithm x_i of y_i by checking the validity of the signature of knowledge V_2. Otherwise, AM would remove the illegal bid information from his bulletin board.

Step 2: Anybody can accept that the signer is one of the bidders if the values $y_i^{r_i}$ and g^{r_i} in V_2 are published on AM's bulletin board. If they are, the bidder which owns the bid information is a legitimate bidder.

【Winner Announcement】

At the end of the bidding, AM on obtaining the information on the highest bid, forwards the r_i^{-1} of B*ᵢ* to RM. Then RM uses $y_i^{r_i}$ and r_i^{-1} to calculate y_i, and saves the comparison result in the database, for confirming the bidder's identity and then informing the vendor of the winner's identity.

Although the scheme of Omote and Miyaji satisfies the security requirements of English Auction, the real identity of the winner cannot be published for verification. In the winner announcement stage, RM secretly informs the vendor of winner's identity. Therefore, other bidders and AM cannot verify the legality of winner. If RM announces the winner's identity, AM can get his/her real identity from the public key that could violate the purpose of anonymity, fairness, and unlinkability among auction rounds [5].

2.2 The scheme by Lee et al.

Lee et al. [5] improved the security problem of Omote and Miyaji's scheme [6] that the identity of the winner cannot be published. And they proposed the essential requirements of the public auction. The scheme of Lee et al. is as follows:

【Entity】

Registration Manager (RM) : being in charge of the registration process and has secret database to keep bidder's identity information and the corresponding secret parameter. After a winner decision procedure, RM and AM together post the winning bidder information on the winner announcement bulletin board.

Auction Manager (AM) : being responsible to manage and host the auction. After a winner decision procedure, RM and AM together post the winning bidder information on the winner announcement bulletin board.

Bidder (B) : being participant in an auction that AM holds.

【Notation】

p, q : two large primes, satisfying $q \mid p-1$;

g : an element $g \in Z_p$ with order q;

B_i : bidder *i*;

x_i : the secret key of B_i ($x_i \in Z_q$);

y_i : the public key of B_i ($\mathrm{y_i = g^{x_i} \ (mod\ p)}$);

t_i : a random number of B_i ($t_i \in \{0,1\}^*$);

$RK_{i,k}$: a round key for B_i in the *k*-th round of auction;

T_i : a ticket identifier for B_i;

k : the index of auctions ($k \geq 1$);

X_{AM} : AM's secret key ($X_{\mathrm{AM}} \in Z_q$);

Y_{AM} : AM's public key ($y_{AM} = g^{X_{\mathrm{AM}}} \bmod p, p \in Z_q$);

$h(x)$: a one-way hash function, satisfying $h^k(x) = h(x, h^{k\text{-}1}(x))$.

The operation of the scheme includes six stages: (1) Initialization, (2) Bidder Registration, (3) Round key Setup, (4) Auction Ticket Preparation, (5) Bidding, and (6) Winner Announcement. The different stages are as described below:

【Initialization】

The system parameter settings of the RM and AM cooperatively set up the system parameters in this stage.

RM executes the following procedure:

Step 1: Set up two read-only bulletin board, and post identities and public keys of all bidders on registration bulletin board and the round keys of all bidders on round key bulletin board. RM is the only one can write and update the bulletin boards.

Step 2: Publish p, q, g and $h(x)$ on his bulletin boards.

Step 3: Together with AM, set up a read-only winning bidder bulletin board and post the winning bidder's information which used to verify one's identity. Only RM and AM have the authority to write and update the bulletin board.

AM executes the following procedure:

Step 1: Set up a read-only auction ticket bulletin board, which provides the auction verification information of all bidders. AM is the only one can write and update the bulletin board.

Step 2: Randomly select an integer $X_{\text{AM}} \in Z_q$ as the private key and use it to calculate the corresponding public key Y_{AM} as follows:

$$Y_{\text{AM}} = g^{X_{\text{AM}}} \bmod p \text{, where } p \in Z_q$$

Step3: Publish Y_{AM}.

Step 4: Together with RM, set up a read-only winning bidder bulletin board and post the winning bidder's information which used to verify one's identity. Only RM and AM have the authority to write and update the bulletin board.

【Bidder Registration】

When a new bidder B$_i$ joins in the auction, he/she must follow the following steps in requesting registration from RM:

Step 1: Select a private key $x_i \in Z_q$ and calculate its corresponding registration key y_i as follows:

$$y_i = g^{x_i} \bmod p$$

Step 2: Select a random number $t_i \in \{0,1\}^*$ and keep it secretly.

Step 3: Send $\{\text{B}_i, y_i, t_i\}$ to RM secretly and prove his/her knowledge of the private key x_i in zero-knowledge.

Step 4: If RM accepts B$_i$'s registration, RM publishes $\{\text{B}_i, y_i\}$ on his registration bulletin board and keeps $\{\text{B}_i, t_i\}$ secretly in his secure database.

【Round key Setup】

RM calculates *n* round key $RK_{i,k}$ for all *n* bidders using y_i and t_i in the *k*-th round of auction as follows:

$$RK_{i,k} = y_i^{h^k(t_i)} \bmod p$$

Then RM shuffles and publishes them on his round key bulletin board. But anybody except RM and B*ᵢ* does not know the correspondence between y_i and $RK_{i,k}$.

【Auction Ticket Preparation】

AM gets the list of all the round keys $RK_{i,k}$ of *n* valid bidders B$_i$ ($i = 1, 2, \ldots, I$) from RM's round key bulletin board. Then AM executes the following steps to complete the setup of the auction:

Step 1: Selects the random numbers $r_i \in Z_q(\{r_1, r_2, \ldots, r_I\})$ for each bidder B$_i$ ($i = 1, 2, \ldots, I$).

Step 2: Calculate the auction keys $\{(RK_{i,k})^{r_i}, g^{r_i}\}$ for each bidder B$_i$ ($i = 1, 2, \ldots, I$).

Step 3: Calculate the ticket identifiers T_i for each bidder B$_i$ ($i = 1, 2, \ldots, I$) as follows:

$$T_i = h((RK_{i,k})^{X_{\mathrm{AM}}} \bmod p$$

$\{T_i, (RK_{i,k})^{r_i}, g^{r_i}\}$ is the auction ticket that AM grants B$_i$ the authorization to participate the *k*-th round of auction.

Step 4: Shuffle and publish the auction tickets $\{T_i, (RK_{i,k})^{r_i}, g^{r_i}\}$ on the auction ticket bulletin board.

Step 5: Keep $\{T_i, r_i\}$ secretly in his database.

【Bidding】

To participate in the k-th round of auction, B$_i$ must complete the following steps:

Step 1: Calculate the round key of the k-th round $RK_{i,k}$ as follows:

$$RK_{i,k} = y_i^{h^k(t_i)} \bmod p$$

And verify that the round key matches with the one that is posted on RM's round key bulletin board. If the round key is not listed, he/she can complain to RM.

Step 2: Calculate the ticket identifier T_i as follows:

$$T_i = h(Y_{\mathrm{AM}}^{h^k(t_i)})^{x_i}$$

T_i must be verified that it matches with the information posted on AM's auction ticket bulletin board. If B$_i$'s ticket identifier is listed in auction ticket bulletin board, he/she can get auction ticket $\{T_i, (RK_{i,k})^{r_i}, g^{r_i}\}$ which granted by AM. Otherwise, B$_i$ can complain to AM.

Step 3: B$_i$ checks the validity of the auction ticket $\{T_i, (RK_{i,k})^{r_i}, g^{r_i}\}$ as below:

$$(g^{r_i})^{h_k(t_i)x_i} \bmod p \overset{?}{=} (RK_{i,k})^{r_i} \bmod p$$

If it does not hold, B$_i$ can complain to AM.

Step 4: Prepare the bid information $\{T_i, m_i, V_i\}$ as follows and post them on the bidding bulletin board:

$$m_i = (auction\_ID \,||\, bid\_value)$$

$$V_i = SK[\alpha_i : (RK_{i,k})^{r_i} = (g^{r_i})^{\alpha_i}](m_i), \text{ where } \alpha_i = h^k(t_i)^{x_i}$$

【Winner Announcement】

Assume that a bid m_i of bidder B$_i$ is the highest bid at the end of the bidding stage. AM and RM jointly publish the winner's related information on the winner announcement bulletin board for others to verify winner's identity. The steps are as follows:

Step 1: AM announces the winner's bid information $\{T_i, m_i, V_i\}$ on the winner announcement bulletin board.

Step 2: AM posts $\{T_i, r_i, RK_{i,k}\}$ on the winner announcement bulletin board which allows anyone to confirm the correlation between $RK_{i,k}$ and $(RK_{i,k})^{r_i}$.

Step 3: RM posts $\{RK_{i,k}, h^k(t_i), y_i\}$ on the winner announcement bulletin board which allows anyone to confirm the correspondence between $RK_{i,k} = y_i^{h^k(t_i)} \bmod p$ and y_i. It shows that B$_i$ is the winner.

Step 4: Anyone verifies that B$_i$ is the winner using the published values r_i and $h^k(t_i)$.

Lee et al.'s scheme solves the security concerns of Omote and Miyaji's scheme that the winner's identity cannot be published. However, if there are n bidders to participate in the auction, AM not only publishes 3n the amount of information on auction ticket bulletin board, but also selects a random secret number for each bidder to employ modular exponentiation operations. It would increase the computation amount.

2.3 The scheme by Chang et al.

Chang et al. [9] proposed a simple and efficient method to ensure that the bidders can bid arbitrarily and anonymously in auction. Later, Jiang et al. [10] claimed that the initiation of their scheme would result in a security drawback because the bidder does not authenticate the auctioneer in the initiation. Subsequently, Chang et al. utilized an alias to resolve the situation [11]. The scheme of Chang et al. is as follows:

【Entity】

Certification Authority (CA), : being responsible to issue each bidder and P a certificate containing the public key and the signature of CA for this certificate.

Auctioneer (P) : being responsible to manage and host the auction.

Bidder (U) : being participant in an auction that P holds.

【Notation】

$E2_{PK}(m)$: an asymmetric encryption function with the public key PK to encrypt the message m;

$S2_{SK}(m)$: an asymmetric decryption function with the private key SK to decrypt the message m;

$E1_{K}(m)$: a symmetric encryption function with the secret key K to encrypt the message m;

P : the just auctioneer;

U_i : the bidder i;

ID_{U} : the bidder U's unique identity;

$PK_{\mathrm{p}}, SK_{\mathrm{p}}$: the auctioneer P's public key and private key;

$PK_{\mathrm{U}}, SK_{\mathrm{U}}$: the bidder U's public key and private key;

n, g : the public system parameters, where g and n are two public primes as in Diffie-Hellman protocol;

$H()$: the collision-free one-way hash function;

$||$: the concatenation symbol.

The operation of the scheme is as described below:

【Initialization】

CA issues each of the bidders and the just auctioneer P a certificate containing the corresponding public key and the signature of CA for this certificate. The following steps are performed such that P and U$_i$ share a secret *k*:

Step 1: U$_i$ randomly chooses a large number *a* and computes X and X' as below:

$$X = g^a \bmod n$$

$$X' = S2_{SK_{\mathrm{U}}}(X)$$

Then U$_i$ sends ID_{U}, X and X' to P.

Step 2: P first verifies X by checking if $X = E2_{PK_{\mathrm{U}}}(X')$. If X is indeed sent from U$_i$, P chooses a random large number *b* and computes Y, Y', *k* and *W* as below:

$$Y = g^b \bmod n$$

$$Y' = S2_{SK_{\mathrm{P}}}(Y)$$

$$k = X^b \bmod n = g^{ab} \bmod n$$

$$W = E1_k(AID_{\mathrm{U}} \,||\, H(ID_{\mathrm{U}} \,||\, X \,||\, Y))$$

Then, P sends Y, Y' and *W* to U$_i$, where AID_{U} is U$_i$'s alias of his/her real identity.

Step 3: Upon getting Y, Y' and *W*, U$_i$ checks whether Y is valid by the following equation:

$$Y = E2_{PK_{\mathrm{P}}}(\mathrm{Y}')$$

If it does not hold, U$_i$ may inform P the information; otherwise, U$_i$ computes k' which used to decrypt W to get AID_{U} as follows:

$$k' = Y^a \bmod n = g^{ab} \bmod n$$

U$_i$ verifies AID_{U} by checking if $H(ID_{\mathrm{U}} \| X \| Y)$ is contained in the decryption result. If it does not hold, U$_i$ may ask P to retransmit the essential information; otherwise, U$_i$ makes sure that P is legal and k' is indeed the shared secret. Then U$_i$ computes Z as follows:

$$Z = E1_k(AID_{\mathrm{U}} \| H(Y \| Y' \| W))$$

And U$_i$ sends ID_{U} and Z to P.

Step 4: After getting ID_{U} and Z, P uses k to decrypt Z and checking if AID_{U} and $H(Y \| Y' \| W)$ are in the decryption result. If both of them are contained, P makes sure U$_i$ is legal and has already gotten the shared secret; otherwise, P may resend essential information to have them share the secret k.

【Traditional English Auction】

After the Initialization stage, the bidder U$_i$ and the auctioneer P share a secret k, and they have authenticated each other. If the bidder U$_i$ wants to bid, he/she performs the following steps:

Step 1: U$_i$ computes $S = S2_{SK_U}(B \| T)$, where B is U$_i$'s bid and T is the current timestamp.

Step 2: U$_i$ computes $D = E1_{k'}(S)$.

Step 3: U$_i$ computes C and casts ($AID_{\mathrm{U}}, B, T, D, C$), where $C = H(B \| T \| k')$.

P will set up a timer. When a bidder bids, P resets the timer and verifies the bid by computing $C' = H(B \| T \| k)$ for AID_{U}. If C' is not equal to C, P announces that (B, T, D, C) is invalid; or, P uses k to decrypt D to get $S' = S2_{SK_U}(B \| T)$ and checks if $(B \| T) = E2_{PK_U}(S')$. If it holds, the bid B is valid; otherwise, P announces that (B, T, D, C) is invalid. To prevent the malicious user from linking the bidder and the corresponding bids, AID_{U} can be changed to be $AID_{\mathrm{U}'}$ whenever U$_i$ bids with AID_{U}. This approach depends on the policies.

If the countdown of the timer is zero and no bidder casts his/her bid, P closes the auction and resolves the winner anonymously by the result of comparison while verifying the bid.

3. Preliminaries

This section presents relevant mathematical concepts and principles of cryptography which this paper utilizes.

3.1 Mathematical theories

A lot of mathematical theories are utilized in cryptography, such as Number Theory, Complexity Theory, and so on. They are the indispensable tools used to design cryptography systems and protocols. The required mathematics for cryptography is shown as below [12].

3.1.1 One-way Hash Function

The most basic concept for cryptographic applications is the One-way Function. A One-way Function is a function that is easy to compute on every input, but hard to invert given the image of a random input. Not being one-to-one is not considered sufficient of a function for it to be called one-way. And a One-way Hash Function is a function that transforms one arbitrary-length input to a shorter and fixed-length output. Assume M is a plaintext of arbitrary-length and $h()$ is a One-way Hash Function. $h()$ should have the following properties:

(1) Easy to compute: Given one arbitrary-length input M, it is easy to compute a fixed-length output $h(M)$.

(2) Preimage-resistance: If we only knows the output $h(M)$, it should be unfeasible to compute the input M.

(3) Collision-resistance: Given $h()$ and a "suitably chosen" M, it should be hard to find

another input M' such that $h(M') = h(M)$.

3.1.2 Discrete Logarithm Problem

Let *p* and *q* be two large prime numbers which satisfy $q \mid p-1$, and one integer *g* be a generator with order *q* in Z_p (i.e. $1 < g < p-1$). Given one integer *x*, the maximum number of multiplication operations are required to calculate $y = g^x \bmod p$ is $\lfloor \log_2 x \rfloor + w(x) - 1$, where $w(x)$ represents the number of 1 when *x* is expressed in binary form. However, to calculate *x* with the known numbers, *p*, *g*, and *y*, the fastest solution presently requires $[L(p) = \exp\{(\ln p \ln(\ln p))^{1/2}\}]$ times of operations. Such problems which are encountered in the above calculations are regarded as Discrete Logarithm Problem. For example, when $p = 512$, the required operation time is $L(p) \cong 2^{256} = 10^{77}$ that it is rather impossible to calculate *x* with *y*.

3.1.3 Integer Factorization Problem

There are two known large prime numbers *p* and *q*. To calculate the product $N = pq$, only one multiplication operation is required. However, it is not nearly so easy to calculate the actual factors *p* and *q* from only a knowledge of the product N. This is called Integer Factorization Problem (IFP), which cannot be solved in the polynomial time.

3.2 Principles of cryptography

Due to the rapid growth of digital communication and electronic data exchange, information security has become a crucial issue in business, administration and so on. Cryptography provides essential techniques for securing information and protecting data to ensure confidentiality, integrity, authenticity, and non-repudiation of information. So it has become one of the main tools for privacy, trust, access control, electronic payments, corporate security, and countless other fields. Aiming at the required cryptography for cryptosystems [12], it is further explained as follows.

3.2.1 Public-Key Cryptosystem

The Public-Key Cryptosystem is also known as Asymmetric Cryptography. It is based on the use of two different keys, where the key used to encrypt a message is not the same as the key used to decrypt it. One is secret key, which used to decode or sign the documents, and known only by its owner. The other one is public key, which used to encode and verify a signature, and known to everyone. The Public-Key Cryptosystem proposed by Diffie and Hellman in 1976 is one of such cryptosystem [12, 13]. It presents the features of simply security analysis and being able to solve the problems of key distribution and management in Symmetric Cryptography, but takes time for encryption and decryption operations.

3.2.2 RSA Cryptosystem

RSA is developed by Rivest, Shamir and Adleman. It makes use of an expression with exponentials. It is the first algorithm known to be suitable for signing as well as encryption, and is one of the first great advances in Public-Key Cryptography. RSA is one specific method of Public-Key Cryptosystem utilizing two prime numbers as the key for encryption and decryption. The operations of RSA are listed as below:

(1) Randomly select two large prime numbers p and q, and calculate N as below:

$$N = pq$$

(2) Compute the least common multiple of two large prime numbers

$\phi(N) = (p-1)(q-1)$.

(3) Compute public key e, where e satisfies $GCD(e, \phi(N)) = 1$.

(4) Compute secret key d, where d satisfies $ed = 1 \bmod \phi(N)$.

(5) Publish (e, N), but keep secret key d secretly. Since the security of RSA is based on that calculating the prime factors p and q after publishing N is infeasible, the large prime numbers p and q should be carefully selected so that the factorization of them becomes impossible.

(6) The encryption and the decryption are presented as below, where M is the plaintext and C is the ciphertext after encryption:

【Encryption】 $C = M^e \bmod N$

【Decryption】 $M = C^d \bmod N$

The difficulty in factorizing large prime numbers determines the reliability of RSA algorithm. In other words, the more difficulty the integer factorization presents, the more reliable RSA algorithm is.

3.2.3 ElGamal Digital Signature

To put digital signatures on a document aims to show the integrity and the non-repudiation. Integrity refers to preventing the document from being tampered in the transmission process. And the receiver can use the digital signature to verify whether the document is tampered or not. Since digital signature is calculated by the signer using his/her secret key, it could prevent the signer from denying the signature that he/she signed afterwards. This is considered to protect the receiver, as non-repudiation. In numerous researchs, many schemes were proposed for digital signature, such as DSA, RSA, Schnorr, ElGamal [12, 14], and ECDSS. The signature scheme utilized in this paper is similar to ElGamal, which is further introduced as follows:

(1) Choose a large prime number *p*, where $p-1$ has a large prime factor, and a primitive root *g*, where $g \in Z_p^*$.

(2) The signer chooses one integer *x* as his/her secret key satisfying $1 < x < p-1$.

(3) The signer uses the following equation to compute his/her public key *y* and publishes *y*:

$$y = g^x \bmod p$$

(4) The signer randomly chooses an integer k satisfying $(k, p-1) = 1$.

(5) Compute the signature (r, s) of plaintext *m*:

$$r = g^k \bmod p$$

$$m = xr + ks \bmod p-1 \quad \text{or} \quad s = k^{-1}(m - xr) \bmod p-1$$

(6) The receiver verifies the legality after receiving the signature (r, s) as below:

$$g^m = y^r r^s \bmod p$$

If the equation holds, (r, s) is the legal signature of plaintext *m*, and vice versa.

The security of ElGamal bases on the difficulty of Discrete Logarithm Problem that the security depends on *p* and *g*. Inappropriate selections of *p* and *g* would therefore result in signature being forged.

3.2.4 Diffie-Hellman key exchange

Diffie-Hellman key exchange scheme allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel

[15]. The security of this type schemes is based on the difficulty of Discrete Logarithm Problem. The operations of Diffie-Hellman key exchange are shown as below:

(1) A selects one integer *x* as his/her secret key, which satisfy $1 < x < p-1$ where *p* is a prime number. Then A uses the following equation to compute his/her public key *X* and publish *X*:

$$X \equiv g^{x} \bmod p$$

(2) B selects one integer *y* as his/her secret key, which satisfy $1 < y < p-1$ where *p* is a prime number. Then B uses the following equation to compute his/her public key *Y* and publish *Y*:

$$Y \equiv g^{y} \bmod p$$

(3) Both A and B have the published public keys and their own secret keys to calculate the shared secret key K_{ab} as below:

$$K_{ab} \equiv Y^{x} \bmod p \quad \text{or} \quad K_{ba} \equiv X^{y} \bmod p$$

(4) The last step is verification as follows:

$$K_{ab} \equiv Y^{x} \equiv (g^{y})^{x} \equiv (g^{x})^{y} \equiv X^{y} \equiv K_{ba} \bmod p$$

There are two disadvantages to Diffie-Hellman scheme. First, the session key could merely be used between the two parties. When there are *n* users in a system, and one of them wants to communicate with any of the other users, the user will need to have *n*-1 session keys and the system will have to maintain $x(x-1)/2$ session keys. Second, the identities of the presently communicated objects are unknown; i.e., A and B would not recognize each other's actual identity.

3.2.5 Elliptic Curve Cryptosystem

In 1985, Elliptic Curve Cryptography (ECC) was proposed by Neal Koblitz [16] and Victor Miller [17]. The ECC was able to improve the existing cryptogram systems in terms of having smaller system parameters, smaller public-key certificates, lower bandwidth usage, faster implementations, lower power requirement, and smaller hardware processor requirements [18]. Therefore, using the Elliptic Curve Cryptography to building up a cryptosystem is commendable by the reasons of high security and efficiency [19]. The mathematic settings of Elliptic Curve Cryptosystem can be as described below [19, 20].

First, elliptic curves can be divided into two families: prime curves and binary curves. Prime curves *(Z_p)* are good to used in software application, because it doesn't require having the extended bit-fiddling operation, which is needed by binary curve. Binary curves ($GF(2^n)$) are best for hardware application as it require a few logic gates to build a powerful cryptosystems. Second, the variable and coefficients of the elliptic curves are limited to the elements of the finite field. Because of this limitation, it would increase the efficiency of ECC computing operation.

In the finite field Z_p, defined modulo a prime p, an elliptic curve is represented as $E_p(a,b): y^2 = x^3 + ax + b \,(\text{mod } p)$, where $(a,b) \in Z_p$ and $4a^3 + 27b^2 \text{ mod } p \neq 0$. The condition, $4a^3 + 27b^2 \text{ mod } p \neq 0$, is necessary to ensure that $y^2 = x^3 + ax + b \,(\text{mod } p)$ has no repeated factors, which means that a finite abelian group can be defined based on the set $E_p(a,b)$ [21]. Included in the definition of an elliptic curve, a point at infinity denoted as O is also called the zero point. The point at infinity O is the third point of intersection of any straight line with the curve, so that there are points including (x, y), (x, $-y$), and O on the straight line.

For points on an elliptic curve, we define a certain addition, denote “+”. The addition rules are given below.

(1) $O + P = P$ and $P + O = P$, where O serves as the additive identity.

(2) $-O = O$.

(3) $P + (-P) = (-P) + P = O$, where $-P$ is the negative point of P.

(4) $(P + Q) + R = P + (Q + R)$.

(5) $P + Q = Q + P$.

For any two points $P=(x_p, y_p)$ and $Q=(x_q, y_q)$ over $E_p(a,b)$, the elliptic curve addition operation, which is denoted as $P+Q=R=(x_r, y_r)$, satisfies the following rules.

$$\begin{matrix} x_r = (\lambda^2 - x_p - x_q) \bmod p \\ y_r = (\lambda(x_p - x_r) - y_p) \bmod p \end{matrix}, \text{ where } \lambda = \begin{cases} \left(\dfrac{y_q - y_p}{x_q - x_p}\right) \bmod p, \text{if } P \neq Q \\ \left(\dfrac{3x_p^2 + a}{2y_p}\right) \bmod p, \text{if } P = Q \end{cases}$$

【Example】

To give an equation of the form denoted as $E_{23}(1,4): y^2 = x^3 + 1x + 4 \bmod 23$, $a=1, b=4 \in Z_p$, and $4a^3 + 27b^2 = 22 \bmod 23 \neq 0$, points over the elliptic curve $E_{23}(1,4)$ show in Table 1 [22].

Table 1: Points over the elliptic curve $E_{23}(1,4)$

(0,2)	(0,21)	(1,11)	(1,12)	(4, 7)	(4,16)	(7,3)
(7,20)	(8,8)	(8,15)	(9,11)	(9,12)	(10,5)	(10,18)
(11,9)	(11,14)	(13,11)	(13,12)	(14,5)	(14,18)	(15,6)
(15,17)	(17,9)	(17,14)	(18,9)	(18,14)	(22,5)	(22,18)

Let $P=(7,3)$ and $Q=(8,15)$ in $E_{23}(1,4)$. When $P \neq Q$, we must derive λ before calculating $P + Q$, as follows:

$$\lambda = \left(\frac{15-3}{8-7}\right) \bmod 23 \equiv 12 \bmod 23 \equiv 12$$

So, when $\lambda = 12$, x_r and y_r can be derived as shown below:

$$x_r = (12^2 - 7 - 8) \bmod 23 \equiv 129 \bmod 23 \equiv 14$$
$$y_r = (12(7 - 14) - 3) \bmod 23 \equiv -87 \bmod 23 \equiv 5$$

Thus, $P + Q = R = (14, 5)$.

To calculate $2P$, $P = (7, 3)$, we must first derive λ as follows:

$$\lambda = \left(\frac{3 \times 7^2 + 1}{2 \times 3}\right) \bmod 23 \equiv \left(\frac{148}{6}\right) \bmod 23 \equiv 17$$

So, when $\lambda = 17$, x_r and y_r can be derived as shown below:

$$x_r = (17^2 - 7 - 7) \bmod 23 \equiv 257 \bmod 23 \equiv 22$$
$$y_r = (17(7 - 22) - 3) \bmod 23 \equiv -258 \bmod 23 \equiv 18$$

Thus, $P + P = 2P = (22, 18)$.

We can see point multiplication on the elliptic curve. But, the point multiplication does not actually mean that one point multiplies by another. In fact, we have to use the equation, $Q = k \times P$, in order to obtain a point on the curve. By assuming k is a natural number and Q and P are points which are on E, Q can be defined as $P + P + \ldots + P$ in k times. The security of ECC in the finite field is based on double-and-add algorithm, $Q = k \times P$. Therefore, it is difficult to compute the result of k, even if the numbers of Q and P are given. This is the conundrum of Elliptic Curve Cryptography and is also known as Elliptic Curve Discrete Logarithm Problem (ECDLP) [23].

3.3 Mobile agent

We will introduce the basic concept of mobile agent and the architecture of the auction model for mobile agent.

3.3.1 Basic concept of mobile agent

Recently, the application of mobile technology in network data transfer has received considerable attention. In information technology, the mobile agent is a highly autonomous and mobile software that users can capitalize to perform tasks in heterogeneous network environments. Since the mobile agent does not require constant online network connections, it also demonstrates significant improvement in network performance.

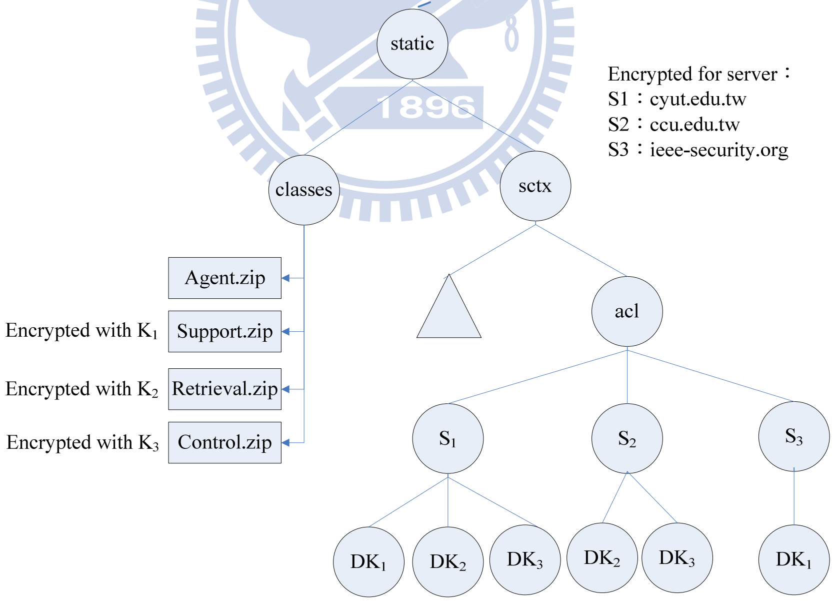

Figure 1: Tree-based structure of key management and access control

The advantages of mobile agent are including (1) Low network loads, (2) High network latency resistance, (3) Encapsulation of protocols, (4) Asynchrony and autonomy, (5) Dynamical adaptation, and (6) Natural heterogeneity. The advantages of assigning tasks to mobile agent cannot be overemphasized. However, because it is entrusted with the user's private key and agent code at the time of task assignment, data transfer management regarding agents' access management and control becomes particularly important.

In related studies, Akl and Taylor proposed a tree-structured key management scheme [24]. Subsequently, Volker and Mehrdad [25] integrated the mobile agent concept into the tree-structured key management mechanism and proposed a system architecture as shown in Figure 1.

Although this scheme can effectively solve data transmission insecurity, efficiency can still be improved on. In Figure 2, repetitious key storage in different agent codes not only results in memory space consumption, but also is large costs on execution performance to key computation.

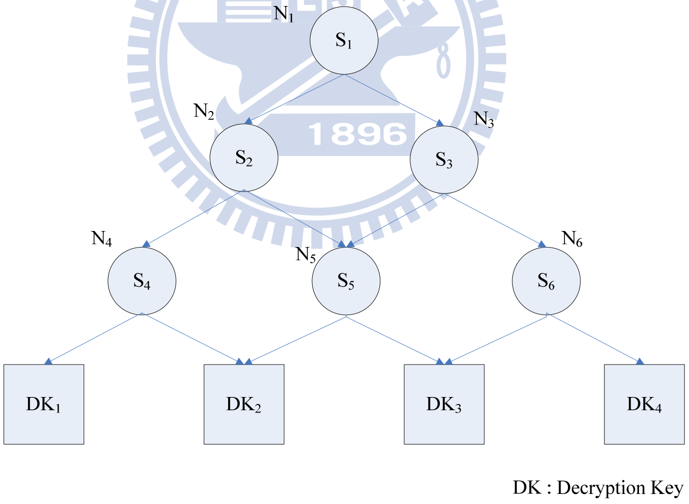

Figure 2: Hierarchical structure of key management and access control

Therefore, Huang et al. [21] proposed two new schemes in mobile agent application. The first method applied the tree-structure of Akl and Taylor into agent management, integrating the keys of lower successive tiers as one. The server can thus use its own key and through mathematical computation obtain successive private keys that can restore confidential documents. This scheme's security is based on the difficulty of Elliptic Curve Discrete

Logarithm Problem. The second scheme, in its attempt to improve mobile agent's computation efficiency, applies the cryptosystem based on the difficulty of Discrete Logarithm Problem to reduce public parameter size without compromising security. Their approach uses hierarchical structure to manage mobile agent's access control to users' keys, at the same time protecting data transmission when access permission differs from user to user as shown in Figure 2.

During task execution, the mobile agent roams between various hosts in a network. In the process to carry out message exchanges, it may also be required to connect with other mobile agents, which thus imposes security concerns arising out of insecure connections [26], malicious modifications by unauthorized external users, or even deliberate attacks by internal users. Therefore, mobile agent security is an important issue that needs to be overcome to effect into its successful application.

The mobile agent might face the following threats during task execution:

(1) Unauthorized users access the related information of servers

A. Causing deliberate system paralysis and breakdown in a non-authorization situation

B. Unauthorized access to data or resource form server by forging as authorized agents

(2) Attacks on mobile agent by other agents

A. Forging identity codes of other agents for the authorization to access services and resources, thereby avoiding responsibility and breaching other users' trust on the legal mobile agent.

B. Paralyzing the legal agent by sending repetitious messages.

(3) Attacks on mobile agent by other malicious servers

A. Deceive and threaten the mobile agent through the abuse of trusted third party servers' identification codes.

B. Ignore mobile agent's requests deliberately.

C. Deceive negotiating mobile agents by tampering their data field.

(4) Attacks on server by other malicious servers or mobile agents

A. Deliberate delayed response to mobile agent's request: such common attacks are intended to cut off the requests or lower mobile agent's efficiency by making mobile agent wait for response, resulting in repeated requests and hence, lowering system efficiency. Abnormal or abrupt task termination of mobile agents also effects into deadlock state where other mobile agents continue to wait for response.

B. Deter the mobile agents from task completion deliberately, resulting into a live-locked.

3.3.2 Mobile Auction Agent Model (MoAAM)

MoAAM [4] is designed to enable users to use their mobile devices to participate in online auctions. MoAAM consists of four agents: (1) Personal Agent, (2) Customer Agent, (3) Auctioneer Agent and (4) Broker Agent. How these agents communicate with each other in MoAAM through a web server is shown in Figure 3.

Inside the mobile device, there is an interactive interface, called Personal Agent, which would connect with an Agent House Server via the wireless network. In a few words, a Personal Agent is a preset agent that operates on the mobile device and provides an interface to allow users to communicate with the Agent House Server. The Customer Agent, Auctioneer Agent, and Broker Agent all operates in the fixed network. The Personal Agent connects to the Customer Agent when a mobile network user wants to buy a specific product. Then the Person Agent sends the description of desired products and price information to the Customer Agent. On the other hand, an auctioneer registers the information of products to Broker Agent. After the Broker Agent receives the user's request, an auction list, which meets the user's needs, will be generated and sent back to the user. If the user decides to purchase the auction items from the received list, a Bid Agent will be created by the Customer Agent and be dispatched to an Auction House Server to join the bidding.

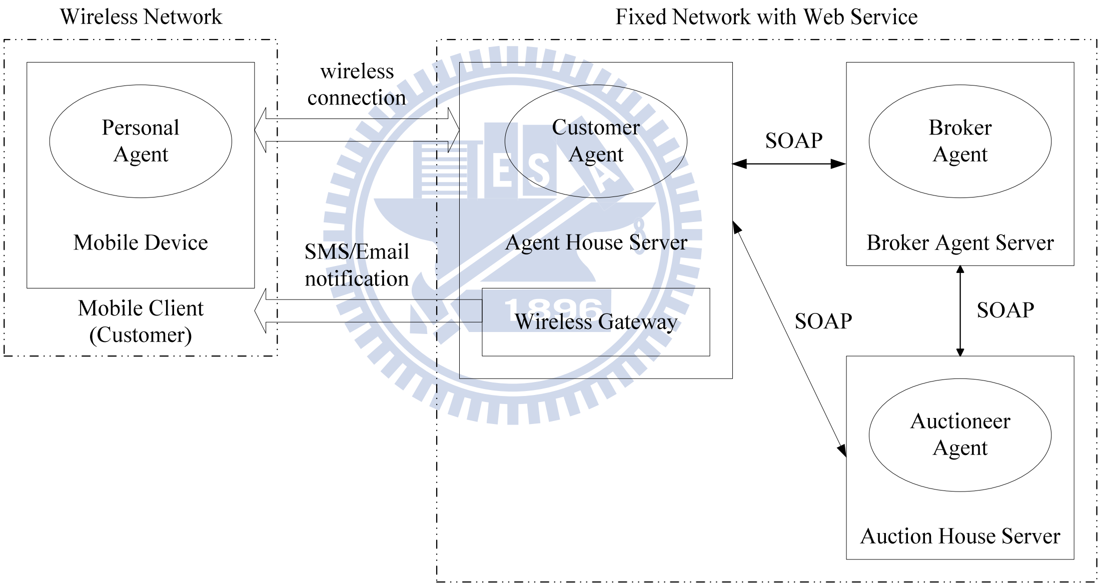

Figure 3: Communication in MoAAM

The architecture of MoAAM [4] is shown in Figure 4, and how it works is described as follows.

(1) Primary participants in MoAAM

(i) Broker Agent: It is responsible to pair up bidders and auctioneers. Moreover, it generates auction item lists and provides bid price information for the users.

(ii) Bid Agent: An individual user would use it to participate in auctions and place the bids.

(iii)Auctioneer Agent: Auctioneers use it as their representative to manage the items they are selling.

(iv)Auction House Server: A platform where online auctions take place.

(2) How Customer Agent operates

The Customer Agent provides an interface with three different functions for the user:

(i) Query the Broker Agent: Know what kind of auction items the Broker Agent so far has registered and bid prices for these items.

(ii) Specify the Bid Agent: A user sends his/her request and bidding information to the Bid Agent generator. The generator will create a Bid Agent from a template.

(iii)Control the Bid Agent: This function allows the user to communicate with the Bid Agent and control the behavior of a Bid Agent.

(3) How Broker Agent operates

First, the auctioneer needs to register his/her agent with the Broker Agent, and then the Broker Agent will store the auctioneer's information in the database. When the Customer Agent sends a request for item information, the Broker Agent would reply a list of recommended items to the Customer Agent.

(4) How Auction House operates

The Auction House Server offers a web interface to allow the auctioneers to execute the following functions:

(i) Specify the Auctioneer Agent: An auctioneer sends his/her request and auction information to the Auctioneer Agent generator. The generator will create an Auctioneer Agent from a template. The newly created agent and auction information would be registered with the Broker Agent.

(ii) Control the Auctioneer Agent: This interface allows the auctioneer to communicate with the Auctioneer Agent and control the Auctioneer Agent's behavior.

(5) Mobile agent platform

The mobile agent platform is where Bid Agent and Auctioneer Agent would be sent to as the auction starts.

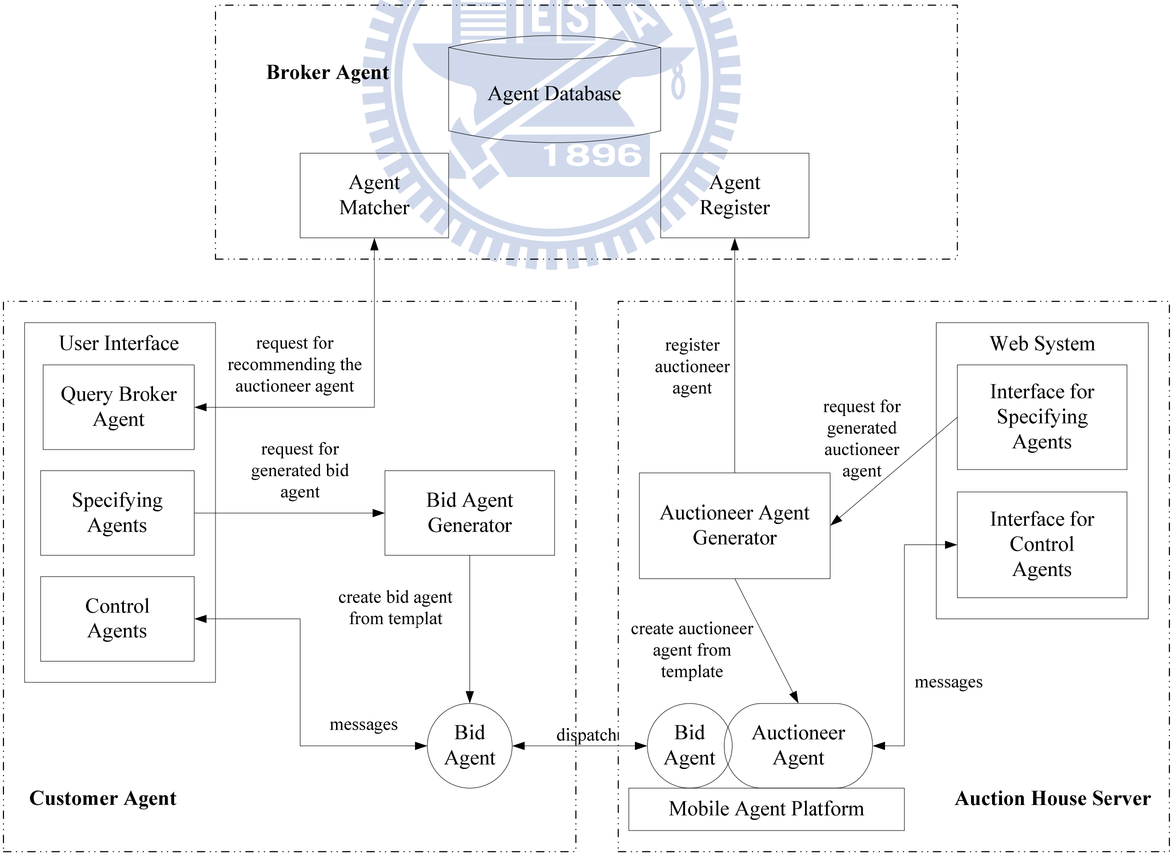
Broker Agent
Agent Database
Agent Matcher
Agent Register
User Interface
Query Broker Agent
Specifying Agents
Control Agents
request for recommending the auctioneer agent
request for generated bid agent
Bid Agent Generator
create bid agent from templat
messages
Bid Agent
dispatch
Customer Agent
register auctioneer agent
request for generated auctioneer agent
Web System
Interface for Specifying Agents
Interface for Control Agents
Auctioneer Agent Generator
create auctioneer agent from template
messages
Bid Agent
Auctioneer Agent
Mobile Agent Platform
Auction House Server

Figure 4: Architecture of MoAAM

4. Proposed scheme

The proposed scheme includes six stages: (1) Initialization, (2) Registration, (3) Generation of Transaction Public Key, (4) Signature, (5) Auction Bidding, and (6) Winner Announcement. The whole process flow is shown in Figure 5. In the process, there are four main participants, which are Registration Manager (RM), Agent House (AH), Auction house (AUH), and Bidder (B).

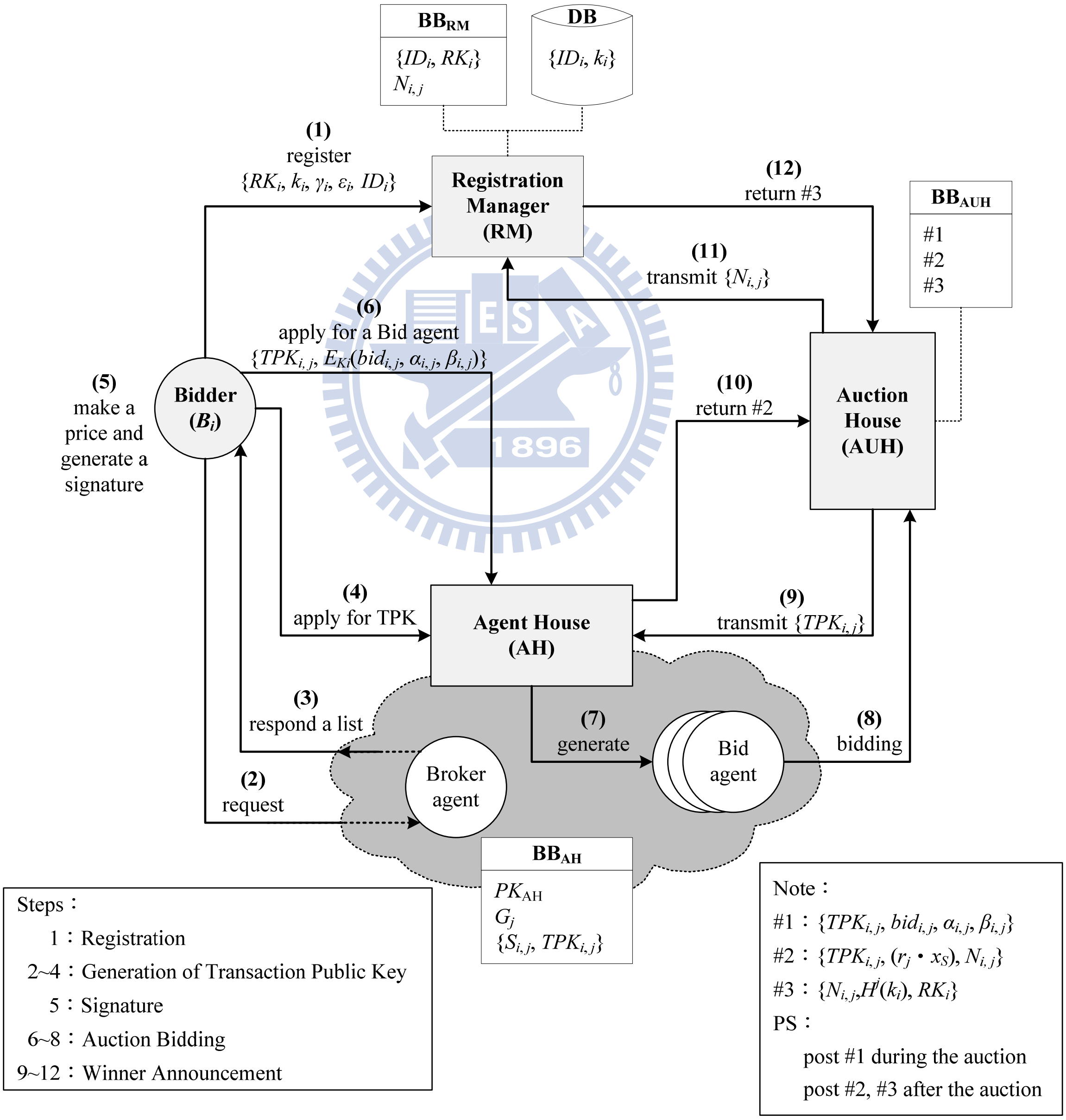

Figure 5: Flow chart of proposed scheme

4.1 The participants

(1) Registration Manager (RM)

(i) It is a unit for bidders to apply for registration. All bidders only require registering once. After that, they can participate in multiple auctions and no more registration is needed.

(ii) It is also responsible to store bidders' identity information and corresponding secret parameters.

(iii)Manages and maintains the bulletin board, which is called $BB_{RM.}$ On the bulletin board, two types of information would be published. One is registration key and identity information of a bidder. Another is pseudonym that a bidder uses in a single auction round. The published information would be supplied to anyone for identification verification. And only the RM has the authority to write and update the bulletin board.

(2) Agent House (AH)

(i) It is responsible to communicates with broker agent and creates bid agents.

(ii) Manages and maintains a bulletin board, which is called BB_{AH}. The bulletin board would provide the bidder's transaction public key for the verification purpose. And only the AH has the authority to write and update the bulletin board.

(3) Auction House (AUH)

(i) Provides the auction place, maintains the operations, and hosts the auctions.

(ii) Manages and maintains a bulletin board, which is called BB_{AUH}. On this bulletin board, the published information would be the bidding information of bidders and the winning bidder’s information. All the published information can be used to verify one’s identity. And only the AUH has the authority to write and update the bulletin board.

(4) Bidder (B)

(i) It is the one who participates and places bids in the auction.

4.2 Using modular exponentiation

In order to know what is the difference in computation amount between using modular exponentiation and using Elliptic Curve Cryptosystem in English Auction protocol, this paper employs the same auction processes on these two methods.

First, the scheme using modular exponentiation in English Auction protocol is presented as follows. The given system parameters are shown in Table 2.

Table 2: System Parameters for modular exponentiation scheme

p, q	Two big prime numbers, satisfying $q \mid p-1$;
g	A generator with order q in Z_p ;
$E_K(m)$	A symmetric encryption method of message m with the key K; (K is the shared key between B$_i$ and AM)
$H(x)$	A one-way hash function, satisfying $H^j(x) = H(x, H^{j\text{-}1}(x))$ and $H^0(x) = x$;
SK_{AH}	AH’s private key;

PK_{AH}	AH's public key;
B_i	The ith bidder;
$bid_{i,j}$	A bid price that is placed by B_i in the jth round of auction;
SK_i	B_i's private key;
RK_i	B_i's registration key;
k_i, $t_{1,i}$, $t_{2,i}$	Three secret parameters that are chosen by B_i;
$N_{i,j}$	A pseudonym, RM creates only for B_i in the jth round of auction;
r_j	A random number chosen by AH in the jth round of auction;
G_j	The public information published by AH in the jth round of auction;
$TPK_{i,j}$	A transaction public key, AH generates only for B_i in the jth round of auction;

The auction processes are described as follows.

4.2.1 Initialization

RM and AH establish system parameters and the steps are as follows:

(1) Registration Manager

Step 1: Sets up a read-only bulletin board ($\mathrm{BB_{RM}}$) and post two kinds of information.

One is registration key and identity information of all bidders. Another is pseudonyms used by the bidders in the *j*th round of auction. RM is the only one can write and update the bulletin board.

Step 2: Declare *p*, *q*, *g*, *H*(*x*) publicly.

(2) Auction House

Step 1: Sets up a read-only bulletin board (BB_{AH}) and publish the transaction public key and related information of all bidders on the bulletin boards. AH is the only one authorized to write and update the bulletin board.

Step 2: Randomly select an integer $SK_{\mathrm{AH}} \in [1, q-1]$ as the private key and use it to calculate the corresponding public key PK_{AH}. The equation is as follows:

$$PK_{\mathrm{AH}} = g^{SK_{\mathrm{AH}}} \bmod p \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Step 3: Post PK_{AH} on $BB_{AH.}$

4.2.2 Registration

As a new bidder (B_i) joins in an auction, he/she must apply for registration with RM at the very beginning. After the registration completes, RM would generates pseudonyms, which can only be used one time, for B_i in the *j*th round of auction.

B_i should first calculate all relevant parameters before registering with RM. The registration process is shown as below:

Step 1: B_i randomly selects an integer $SK_i \in [1, q-1]$ as the private key and computes a corresponding registration key RK_i. The equation is given as follows:

$$RK_i = g^{SK_i} \bmod p \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Step 2: B$_i$ randomly selects an integer $k_i \in [1, q-1]$ as a secret parameter.

Step 3: B$_i$ randomly selects an integer $t_{1,i} \in [1, q-1]$ and computes the verification information $(\gamma_i, \varepsilon_i)$. The computation steps are given as follows:

$$\gamma_i = H(g^{t_{1,i}} \bmod p) \text{ .... } (3)$$

$$\varepsilon_i = (t_{1,i} + \gamma_i \bullet SK_i) \bmod q \text{ .... } (4)$$

Step 4: B$_i$ sends the information $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ and identity information (ID_i) through a secure channel to RM. After RM receives the information transmitted by B$_i$, the registration would proceed.

Step 5: RM authenticates the validity of $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ by the following equations:

$$\gamma_i' = H(g^{\varepsilon_i} \bullet RK_i^{-\gamma_i} \bmod p) \text{ .... } (5)$$

$$\gamma_i' \overset{?}{=} \gamma_i \text{ .... } (6)$$

If (6) holds, $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ is valid. This proves SK_i and RK_i correspond to each other. In contrast, RM would refuse to take the registration application from B$_i$ if the received information is forged.

Step 6: RM keeps B$_i$'s identity information ID_i and the corresponding secret parameter k_i in its own database.

Step 7: RM posts B$_i$'s identity information ID_i and registration key RK_i on BB$_{\text{RM}}$.

Step 8: Before the *j*th round of auction starts, RM would generate a pseudonym ($N_{i,j}$) for every bidder B$_i$. The order of all pseudonyms would be randomly arranged and posted on BB$_{\text{RM}}$. The equation is shown as below:

$$N_{i,j} = RK_i^{H^j(k_i)} \bmod p \text{ .... } (7)$$

Step 9: B$_i$ can use (7) to compute his/her own pseudonym and verify that his/her pseudonym matches with the one that is posted on BB$_{\mathrm{RM}}$. If B$_i$ dose not find his/her pseudonym on BB$_{\mathrm{RM}}$, he/she can appeal to RM.

4.2.3 Generation of Transaction Public Key

In the *j*th round of auction, B$_i$ can obtain auction information through AH who could ask broker agent to supply the information about currently open auctions. The broker agent would prepare an auction house list that matches the needs of B$_i$ and send the list back to the AH for B$_i$ to review. When B$_i$ decides which auction he/she wants to participate, B$_i$ would have to apply a transaction public key ($TPK_{i,j}$), which is managed by AH. AH would generate $TPK_{i,j}$ with B$_i$'s pseudonym on BB$_{\mathrm{RM}}$ for the bidder. Steps are given as follows:

Step 1: AH randomly selects an integer $r_j \in [1, q-1]$ and computes public information G_j. And then, AH would post G_j on BB$_{\mathrm{AH}}$. The equation is shown as below:

$$G_j = g^{r_j} \bmod p \text{ .... } (8)$$

Step 2: AH would use $N_{i,j}$ and its own private key SK_{AH} to generate a parameter $S_{i,j}$ and $TPK_{i,j}$ for each B$_i$, and post the generated information on BB$_{\mathrm{AH}}$. The equation is shown as below:

$$S_{i,j} = N_{i,j}^{SK_{\mathrm{AH}}} \bmod p \text{ .... } (9)$$

$$TPK_{i,j} = N_{i,j}^{r_j \cdot S_{i,j}} \bmod p \text{ .... } (10)$$

4.2.4 Signature

Before B$_i$ starts to participate in the auction, B$_i$ must verify $TPK_{i,j}$ given by the AH on BB$_{\text{AH}}$. If the key is valid, B$_i$ would calculate the corresponding signature with his/her bid price and the related information. Subsequently, B$_i$ can start participating in the bidding. The steps are stated as follows:

Step 1: B$_i$ uses AH's public key PK_{AH} to compute a parameter $S'_{i,j}$, as follows:

$$S'_{i,j} = PK_{\text{AH}}{}^{H^j(k_i)\cdot SK_i} \bmod p \,\ldots\ldots\ldots\ldots\ldots\ldots\,(11)$$

Step 2: B$_i$ combines his/her private key SK_i and parameter $S'_{i,j}$ to generate $TPK'_{i,j}$, as follows:

$$TPK'_{i,j} = G_j{}^{H^j(k_i)\cdot S'_{i,j}\cdot SK_i} \bmod p \,\ldots\ldots\ldots\ldots\ldots\ldots\,(12)$$

$S'_{i,j}$ and $TPK'_{i,j}$ must be verified that they are matched with the information posted on the BB$_{\text{AH}}$; if not, B$_i$ can appeal to AH.

Step 3: B$_i$ randomly selects an integer $t_{2,i} \in [1, q-1]$ and decides a bid price $bid_{i,j}$. Afterward, a corresponding signature $\{\alpha_{i,j}, \beta_{i,j}\}$ is created, shown as below:

$$\alpha_{i,j} = G_j{}^{t_{2,i}} \bmod p \,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\,(13)$$

$$\beta_{i,j} = (t_{2,i} + H^j(k_i)\cdot S_{i,j}\cdot SK_i \cdot H(bid_{i,j} \,\|\, \alpha_{i,j})) \bmod q \,\ldots\ldots\,(14)$$

Step 4: B$_i$ uses AH's public key PK_{AH} to compute the shared key K_i, as follows:

$$K_i = PK_{\text{AH}}{}^{H^j(k_i)\cdot S_{i,j}\cdot SK_i} \bmod p \,\ldots\ldots\ldots\ldots\ldots\ldots\,(15)$$

Step 5: B$_i$ used K_i to encrypt the bid $bid_{i,j}$ and the signature $\{\alpha_{i,j}, \beta_{i,j}\}$, obtaining the ciphertext $E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})$.

4.2.5 Auction Bidding

Before the auction starts, B$_i$ needs to obtain a bid agent from the AH. After a bid agent is acquired, B$_i$, then, is allowed to bid. The auction bidding process is stated as follows:

Step 1: B$_i$ should first send out the data tuple $\{TPK_{i,j},\ E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})\}$ to AH and apply for a bid agent.

Step 2: After the AH receives the data from B$_i$, AH must compute the shared key K_i to decrypt the ciphertext $E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})$ and authenticates the validity of $\{TPK_{i,j},\ E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})\}$. The equations for verification are shown as below:

$$K_i = TPK_{i,j}^{SK_{\mathrm{AH}} \cdot r_j^{-1}} \bmod p \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (16)$$

$$G_j^{\beta_{i,j}} \overset{?}{=} TPK_{i,j}^{\mathrm{H}(bid_{i,j} \| \alpha_{i,j})} \bullet \alpha_{i,j} \bmod p \ldots\ldots\ldots\ldots\ldots\ (17)$$

If (17) holds, this proves $\{TPK_{i,j},\ E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})\}$ is valid, and vice versa. AH can reject the bidding request from B$_i$ if the received information is false.

Step 3: AH uses the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ to create a new bid agent for B$_i$, and then AH would send this agent to the selected AUH to represent B$_i$ to participate in the auction.

Step 1, 2, and 3 can be skipped if the bid is placed more than once. Only the bidding information would be verified.

Step 4: When the bid agent arrives at the AUH, it has to be verified that $TPK_{i,j}$ is same as on BB_{AH}. If not, AUH can reject B*i*'s application.

Step 5: AUH verifies the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ using (17). If (17) holds, it means that $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ is valid.

Step 6: AUH posts the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ on BB_{AUH}. Anyone can use (17) to verify the bidding information of B*i*.

4.2.6 Winner Announcement

When the *j*th round of auction ends, the one who places the highest bid price would be announced as the winner. Then AUH would take the winner's $TPK_{i,j}$ to reconfirm the winner's information, $N_{i,j}$ and RK_i with AH and RM. Afterward, the result would be published on BB_{AUH} and can be obtained by anyone to verify. The steps are stated as follows:

Step 1: AUH takes the winner's $TPK_{i,j}$ to AH and ask for the pseudonym $N_{i,j}$ used by the winner.

Step 2: AH returns the information $\{TPK_{i,j}, (r_j \cdot S_{i,j}), N_{i,j}\}$ back to the AUH.

Step 3: AUH can use (10) to confirm the relationship between $TPK_{i,j}$ and $N_{i,j}$.

Step 4: AUH takes the winner's $N_{i,j}$ to RM and ask for the winner's RK_i.

Step 5: RM returns the information $\{N_{i,j}, H^j(k_i), RK_i\}$ back to the AUH.

Step 6: AUH can use (7) to confirm the relationship between $N_{i,j}$ and RK_i.

Step 7: The AUH will post the winner's information, $\{TPK_{i,j}, (r_j \bullet S_{i,j}), N_{i,j}\}$ and $\{N_{i,j}, H^j(k_i), RK_i\}$, on BB$_{\text{AUH}}$. The winner's information on BB$_{\text{AUH}}$ can be obtained by anyone to verify again by using (7) and (10).

4.3 Using Elliptic Curve Cryptosystem

The auction processes are the same as above, but the way of computation is different. The scheme using Elliptic Curve Cryptosystem in English Auction protocol is presented as follows. The given system parameters are shown in Table 3.

Table 3: System Parameters for Elliptic Curve Cryptosystem scheme

p	A big prime number;
q	A big prime number; q is the order of a generative point on an elliptic curve and its value is within $p+1 \pm 2\sqrt{p}$;
E	Elliptic curve equation $y^2 = x^3 + ax + b \,(\bmod\, p)$, where a, b are real numbers and satisfy $4a^3 + 27b^2 \bmod p \neq 0$;
G	A generative point on an elliptic curve with order as q;
F	A point on an elliptic curve;
x_F, y_F	The value of the x-coordinate and y-coordinate of point F on the elliptic curve;
$E_K(m)$	A symmetric encryption method of message m with the key K; (K is the shared key between B$_i$ and AM)

$H(x)$	A one-way hash function, satisfying $H^{j}(x)=H(x,H^{j\text{-}1}(x))$ and $H^{0}(x)=x$;
$SK_{\mathrm{AH}}, PK_{\mathrm{AH}}$	AH’s private key and public key;
B_i	The i^{th} bidder;
$bid_{i,j}$	A bid price that is placed by B_i in the *j*th round of auction;
SK_i, RK_i	B_i’s private key and registration key;
k_i, $t_{1,i}$, $t_{2,i}$	Three secret parameters that are chosen by B_i;
$N_{i,j}$	A pseudonym, RM creates only for B_i in the *j*th round of auction;
r_j	A random number chosen by AH in the *j*th round of auction;
G_j	The public information published by AH in the *j*th round of auction;
$TPK_{i,j}$	A transaction public key, AH generates only for B_i in the *j*th round of auction;

The auction processes are described as follows.

4.3.1 Initialization

RM and AH establish system parameters and the steps are as follows:

(1) Registration Manager

Step 1: Sets up a read-only bulletin board ($\mathrm{BB}_{\mathrm{RM}}$) and post two kinds of information. One is registration key and identity information of all bidders. Another is pseudonyms used by the bidders in the *j*th round of auction. RM is the only one can write and update the bulletin board.

Step 2: Select a big prime number for p.

Step 3: Declare an elliptic curve equation, $E_p(a,b): y^2 = x^3 + ax + b \,(\text{mod } p)$, that satisfies $(a,b) \in Z_p$ and $4a^3 + 27b^2 \neq 0 \,(\text{mod } p)$.

Step 4: Select and declare a generative point G with an order as q, which is a big prime number and its value should be within $p + 1 \pm 2\sqrt{p}$.

(2) Auction House

Step 1: Sets up a read-only bulletin board (BB$_{\text{AH}}$) and publish the transaction public key and related information of all bidders on the bulletin boards. AH is the only one authorized to write and update the bulletin board.

Step 2: Randomly select an integer $SK_{\text{AH}} \in [1, q\text{-}1]$ as the private key and use it to calculate the corresponding public key PK_{AH}. The equation is as follows:

$$PK_{\text{AH}} = SK_{\text{AH}} G \ldots\ldots\ldots\ldots\ldots\ldots (18)$$

Step 3: Post PK_{AH} on BB$_{\text{AH.}}$

4.3.2 Registration

As a new bidder (B$_i$) joins in an auction, he/she must apply for registration with RM at the very beginning. After the registration completes, RM would generates pseudonyms, which can only be used one time, for B$_i$ in the jth round of auction.

B$_i$ should first calculate all relevant parameters before registering with RM. The registration process is shown as below:

Step 1: B$_i$ randomly selects an integer $SK_i \in [1, q-1]$ as the private key and computes a corresponding registration key RK_i. The equation is given as follows:

$$RK_i = SK_i G \quad \text{..................} \ (19)$$

Step 2: B$_i$ randomly selects an integer $k_i \in [1, q-1]$ as a secret parameter.

Step 3: B$_i$ randomly selects an integer $t_{1,i} \in [1, q-1]$ and computes the verification information $(\gamma_i, \varepsilon_i)$. The computation steps are given as follows:

$$F_{1,i} = t_{1,i} G = (x_{F_{1,i}}, y_{F_{1,i}}) \quad \text{............................} \ (20)$$
$$\gamma_i = H(x_{F_{1,i}} \| y_{F_{1,i}}) \quad \text{....................................} \ (21)$$
$$\varepsilon_i = (t_{1,i} + \gamma_i \bullet SK_i) \bmod q \quad \text{..........................} \ (22)$$

Step 4: B$_i$ sends the information $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ and identity information (ID_i) through a secure channel to RM. After RM receives the information transmitted by B$_i$, the registration would proceed.

Step 5: RM authenticates the validity of $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ by the following equations:

$$F'_{1,i} = \varepsilon_i G - \gamma_i RK_i = (x_{F'_{1,i}}, y_{F'_{1,i}}) \quad \text{...................} \ (23)$$
$$\gamma'_i = H(x_{F'_{1,i}} \| y_{F'_{1,i}}) \quad \text{.......................................} \ (24)$$
$$\gamma'_i \stackrel{?}{=} \gamma_i \quad \text{...} \ (25)$$

If (25) holds, $\{RK_i, k_i, \gamma_i, \varepsilon_i\}$ is valid. This proves SK_i and RK_i correspond to each other. In contrast, RM would refuse to take the registration application from B$_i$ if the received information is forged.

Step 6: RM keeps B$_i$'s identity information ID_i and the corresponding secret parameter k_i in its own database.

Step 7: RM would post B$_i$'s identity information ID_i and registration key RK_i on BB$_{\text{RM}}$.

Step 8: Before the *j*th round of auction starts, RM would generate a pseudonym ($N_{i,j}$) for every bidder B$_i$. The order of all pseudonyms would be randomly arranged and posted on BB$_{\text{RM}}$. The equation is shown as below:

$$N_{i,j} = H^{j}(k_i)RK_i \text{ ....(26)}$$

Step 9: B$_i$ can use (26) to compute his/her own pseudonym and verify that his/her pseudonym matches with the one that is posted on BB$_{\text{RM}}$. If B$_i$ dose not find his/her pseudonym on BB$_{\text{RM}}$, he/she can appeal to RM.

4.3.3 Generation of Transaction Public Key

In the *j*th round of auction, B$_i$ can obtain auction information through AH who could ask broker agent to supply the information about currently open auctions. The broker agent would prepare an auction house list that matches the needs of B$_i$ and send the list back to the AH for B$_i$ to review. When B$_i$ decides which auction he/she wants to participate, B$_i$ would have to apply a transaction public key ($TPK_{i,j}$), which is managed by AH. AH would generate $TPK_{i,j}$ with B$_i$'s pseudonym on BB$_{\text{RM}}$ for the bidder. Steps are given as follows:

Step 1: AH randomly selects an integer $r_j \in [1, q-1]$ and computes public information G_j. And then, AH would post G_j on BB$_{\text{AH}}$. The equation is shown as below:

$$G_j = r_j G \text{ ....(27)}$$

Step 2: AH would use $N_{i,j}$ and its own private key SK_{AH} to generate a parameter $S_{i,j}$ and $TPK_{i,j}$ for each B$_i$, and post the generated information on BB$_{\text{AH}}$. The equation is shown as below:

$$S_{i,j} = SK_{\text{AH}} N_{i,j} = (x_{S_{i,j}}, y_{S_{i,j}}) \text{..................} (28)$$

$$TPK_{i,j} = (r_j \cdot x_{S_{i,j}}) N_{i,j} \text{..............................} (29)$$

4.3.4 Signature

Before B$_i$ starts to participate in the auction, B$_i$ must verify $TPK_{i,j}$ given by the AH on BB$_{\text{AH}}$. If the key is valid, B$_i$ would calculate the corresponding signature with his/her bid price and the related information. Subsequently, B$_i$ can start participating in the bidding. The steps are stated as follows:

Step 1: B$_i$ uses AH's public key PK_{AH} to compute a parameter $S'_{i,j}$, as follows:

$$S'_{i,j} = (H^j(k_i) \cdot SK_i) PK_{\text{AH}} = (x_{S'_{i,j}}, y_{S'_{i,j}}) \text{..................} (30)$$

Step 2: B$_i$ combines his/her private key SK_i and parameter $S'_{i,j}$ to generate $TPK'_{i,j}$, as follows:

$$TPK'_{i,j} = (H^j(k_i) \cdot x_{S'_{i,j}} \cdot SK_i) G_j \text{..................} (31)$$

$S'_{i,j}$ and $TPK'_{i,j}$ must be verified that they are matched with the information posted on the BB$_{\text{AH}}$; if not, B$_i$ can appeal to AH.

Step 3: B$_i$ randomly selects an integer $t_{2,i} \in [1, q-1]$ and decides a bid price $bid_{i,j}$ Afterward, a corresponding signature $\{\alpha_{i,j}, \beta_{i,j}\}$ is created, shown as below:

$$F_{2,i} = t_{2,i}G_j = (x_{F_{2,i}}, y_{F_{2,i}}) \tag{32}$$

$$\alpha_{i,j} = H(x_{F_{2,i}} \| y_{F_{2,i}} \| bid_{i,j}) \tag{33}$$

$$\beta_{i,j} = (t_{2,i} + \alpha_{i,j} \cdot H^j(k_i) \cdot x_{S_{i,j}} \cdot SK_i) \bmod q \tag{34}$$

Step 4: B$_i$ uses AH’s public key PK_{AH} to compute the shared key K_i, as follows:

$$K_i = H^j(k_i) \cdot x_{S_{i,j}} \cdot SK_i \cdot PK_{\text{AH}} \tag{35}$$

Step 5: B$_i$ used K_i to encrypt the bid $bid_{i,j}$ and the signature $\{\alpha_{i,j}, \beta_{i,j}\}$, obtaining the ciphertext $E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})$.

4.3.5 Auction Bidding

Before the auction starts, B$_i$ needs to obtain a bid agent from the AH. After a bid agent is acquired, B$_i$, then, is allowed to bid. The auction bidding process is stated as follows:

Step 1: B$_i$ should first send out data tuple $\{TPK_{i,j},\ E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})\}$ to AH and apply for a bid agent.

Step 2: After the AH receives the bidding information from B$_i$, AH must compute the shared key K_i to decrypt the ciphertext $E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})$ and authenticates the validity of $\{TPK_{i,j},\ E_{K_i}(bid_{i,j}, \alpha_{i,j}, \beta_{i,j})\}$. The equations for verification are shown as below:

$$Ki = (SK_{\text{AH}} \cdot r_j^{-1})TPK_{i,j} \tag{36}$$

$$F'_{2,i} = \beta_{i,j}G_j - \alpha_{i,j}TPK_{i,j} = (x_{F'_{2,i}}, y_{F'_{2,i}}) \tag{37}$$

$$\alpha'_{i,j} = H(x_{F'_{2,i}} \| y_{F'_{2,i}} \| bid_{i,j}) \tag{38}$$

$$\alpha'_{i,j} \overset{?}{=} \alpha_{i,j} \tag{39}$$

If (39) holds, this proves $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ is valid, and vice versa. AH can reject the bidding request from B$_i$ if the received information is false.

Step 3: AH uses the bidding information to create a new bid agent for B$_i$, and then AH would send this agent to the selected AUH to represent B$_i$ to participate in the auction.

Step 1, 2, and 3 can be skipped if the bid is placed more than once. Only the bidding information would be verified.

Step 4: When the bid agent arrives at the AUH, it has to be verified that $TPK_{i,j}$ is same as on BB$_{\text{AH}}$. If not, AUH can reject B$_i$'s application.

Step 5: AUH verifies the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ using (38) and (39). If (39) holds, it means that $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ is valid.

Step 6: AUH posts the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$ on BB$_{\text{AUH}}$. Anyone can use the equation (38) and (39) to verify the bidding information of B$_i$.

4.3.6 Winner Announcement

When the *j*th round of auction ends, the one who places the highest bid price would be announced as the winner. Then AUH would take the winner's $TPK_{i,j}$ to reconfirm the winner's information, $N_{i,j}$ and RK_i with AH and RM. Afterward, the result would be published on BB$_{\text{AUH}}$ and can be obtained by anyone to verify again. The steps are stated as follows:

Step 1: AUH takes the winner's $TPK_{i,j}$ to AH and ask for the pseudonym $N_{i,j}$ used by the winner.

Step 2: AH returns the information $\{TPK_{i,j}, (r_j \bullet x_{S_{i,j}}), N_{i,j}\}$ back to the AUH.

Step 3: AUH can use (29) to confirm the relationship between $TPK_{i,j}$ and $N_{i,j}$.

Step 4: AUH takes the winner's $N_{i,j}$ to RM and ask for the winner's RK_i.

Step 5: RM returns the information $\{N_{i,j}, H^j(k_i), RK_i\}$ back to the AUH.

Step 6: AUH can use (26) to confirm the relationship between $N_{i,j}$ and RK_i.

Step 7: The AUH will post the winner's information, $\{TPK_{i,j}, (r_j \bullet x_{S_{i,j}}), N_{i,j}\}$ and $\{N_{i,j}, H^j(k_i), RK_i\}$, on BB$_{\text{AUH}}$. The winner's information on BB$_{\text{AUH}}$ can be obtained by anyone to verify again by using (26) and (29).

5. Security and efficiency analysis

In response to the security requirements of the English auction [6], the schemes introduced in this paper are explained as follows:

(1) Anonymity

Except RM and AH work together to reveal the identity during the auction, nobody can find out who the bidder is. We can analyze the anonymity of bidders from the perspectives of RM, AH, and AUH.

(i) For AUH, it is only authorized to obtain the bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$. $\{\alpha_{i,j}, \beta_{i,j}\}$ is the signature for $bid_{i,j}$ and $TPK_{i,j}$ is as a key for verification. Thus, AUH is just allowed using the $TPK_{i,j}$ to verify the signature and comparing $TPK_{i,j}$ to the one that is posted on BB$_{\text{AH}}$. Still it will never know who the bidder is.

(ii) For AH, it merely knows the relationship between $N_{i,j}$ and $TPK_{i,j}$; thus, it does not have enough information to recognize who the bidder is.

(iii) For RM, it can't derive from $TPK_{i,j}$ to obtain the corresponding $N_{i,j}$, even if RM has the bidder's identity information.

(2) Traceability

In modular exponentiation scheme, anyone can get $\{TPK_{i,j}, (r_j \cdot x_{S_{i,j}}), N_{i,j}\}$ and $\{N_{i,j}, H^j(k_i), RK_i\}$ from the BB$_{\text{AUH}}$ and use (7) and (10) to verify the winning bidder's identity.

In Elliptic Curve Cryptosystem scheme, anyone can get $\{TPK_{i,j}, (r_j \cdot x_{S_{i,j}}), N_{i,j}\}$ and $\{N_{i,j}, H^j(k_i), RK_i\}$ from the BB$_{\text{AUH}}$ and use (26) and (29) to verify the winning bidder’s identity.

(3) No framing

Unless attackers get the B$_i$’s SK_i, B_i’s signature cannot be forged. Even if attackers get the RK_i and intend to derive the SK_i from the RK_i, it will be difficult for him/her to obtain SK_i, because of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

(4) Unforgeability

Attackers will be unable to calculate the transaction public key by using the equation $TPK_{i,j} = G_j^{H^j(k_i) \cdot S_{i,j} \cdot SK_i}$ (in modular exponentiation scheme) or $TPK_{i,j} = (H^j(k_i) \cdot x_{S'_{i,j}} \cdot SK_i)G_j$ (in Elliptic Curve Cryptosystem scheme) or forge any valid bidding information $\{TPK_{i,j}, bid_{i,j}, \alpha_{i,j}, \beta_{i,j}\}$. The reason can be explained in three aspects.

(i) Attackers cannot obtain B$_i$’s SK_i, k_i and $S_{i,j}$.

(ii) Attackers have to spend a great deal of time on resolving ECDLP, even if $H^j(k_i) \cdot S_{i,j} \cdot SK_i$ or $(H^j(k_i) \cdot x_{S'_{i,j}} \cdot SK_i)$ is captured.

(iii) Because $H^j(k_i)$ is different in each around of auction; thus, the bidder’s pseudonym $N_{i,j}$ and transaction public key $TPK_{i,j}$ would be different in each around of auction.

(5) Non-repudiation

Signature is hidden inside the bidding information and it has the characteristics of no framing. Therefore, the winning bidder of the auction around cannot deny his/her signature.

(6) Fairness

All bidders use pseudonyms to join the auction. AUH will post the valid bidding information the BB_{AUH}. If B_i does not find his/her bidding information, he/she could appeal to AUH. Like this, AUH can fairly handle all bidders' information.

(7) Public verifiability

Anyone can confirm the validity of the bidder, the validity of a bid and winning bidder's real identity.

(8) Unlinkability among various auction rounds

In each auction, the pseudonym generated by RM and the transaction public key generated by AH are different. Except RM and AH share these keys with each other, no one will know the relationship about B_i among various auction rounds.

(9) Linkability within a single auction round

Within a single around of the auction, B_i holds the same $TPK_{i,j}$ to place a bid in the auction. It is traceable to know how many times the bidder places the bid and who places the bid.

(10) Efficiency of Bidding

The Elliptic Curve Cryptosystem can reduce the computation loads that are generated by online bidding operations.

(11) One-time registration

Bidder uses a pseudonym $N_{i,j}$ to participate in the auction. Hence, B$_i$ only needs to register once with RM.

(12) Easy revocation

It is easy for RM to delete the bidder’s identification and secret parameters from the database. Once the information is removed from the database, the bidder loses the right to participate in the auction.

The following is the efficiency comparison of modular exponentiation scheme and Elliptic Curve Cryptosystem scheme. The symbols used are defined below:

T_{MUL}	the time cost of modulus multiplication operation;
T_{EXP}	the time cost of modulus exponentiation operation;
T_{H}	the time cost of one-way hash function operation;
$T_{\text{EC\_MUL}}$	the time cost of elliptic curve multiplication operation;
$T_{\text{EC\_ADD}}$	the time cost of elliptic curve addition operation;

Before the comparison, the complex time cost of the multiplication operation of the two methods is equated to enable comparison of the two methods. According to the research proposed by Koblitz et al. [27], in the operation of g^t mod p, t is a random integer of 160 bit, p is a prime number of 1024 bits, and the elliptic curve multiplication operation is used to calculate bG, where $G \in E(Z_p)$, $p \approx 2^{160}$, and b is a random number of 160 bits. Using above information, we can know the operation relations are as follows:

$$T_{EXP} \approx 240T_{MUL}$$

$$T_{EC\_MUL} \approx 29T_{MUL}$$

$$T_{EC\_ADD} \approx 0.12T_{MUL}$$

Due to modulus addition and subtraction operation amount is low, they can be omitted.

The table of the time complexity of the two methods is based on the above information, as described below:

Table 4: Time complexity comparison

Stage	Modular Exponentiation		Elliptic Curve Cryptosystem	
	Time complexity	Summary estimation	Time complexity	Summary estimation
Registration	$5n\ T_{\mathrm{EXP}}$ $+ 2n\ T_{\mathrm{MUL}}$ $+ 3n\ T_{\mathrm{H}}$	$1202n\ T_{\mathrm{MUL}}$ $+ 2n\ T_{\mathrm{H}}$	$5n\ T_{EC\_MUL}$ $+ n\ T_{\mathrm{EC\_ADD}}$ $+ n\ T_{\mathrm{MUL}}$ $+ 3n\ T_{\mathrm{H}}$	$146.12n\ T_{\mathrm{MUL}}$ $+ 3n\ T_{\mathrm{H}}$
Generation of Transaction Public Key	$(2n + 1)\ T_{\mathrm{EXP}}$ $+ n\ T_{\mathrm{MUL}}$	$481n\ T_{\mathrm{MUL}}$ $+ 240\ T_{\mathrm{MUL}}$	$(2n + 1)\ T_{EC\_MUL}$	$58n\ T_{\mathrm{MUL}}$ $+ 29\ T_{\mathrm{MUL}}$
Signature	$4\ T_{\mathrm{EXP}}$ $+ 8\ T_{\mathrm{MUL}}$ $+ 2\ T_{\mathrm{H}}$	$968\ T_{\mathrm{MUL}}$ $+ 2\ T_{\mathrm{H}}$	$4\ T_{EC\_MUL}$ $+ 3\ T_{\mathrm{MUL}}$ $+ 2\ T_{\mathrm{H}}$	$119n\ T_{\mathrm{MUL}}$ $+ 2\ T_{\mathrm{H}}$
Auction Bidding	$3\ T_{\mathrm{EXP}}$ $+ 2\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$	$722\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$	$3\ T_{EC\_MUL}$ $+ 1\ T_{\mathrm{EC\_ADD}}$ $+ 1\ T_{\mathrm{H}}$	$87.12n\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$
Winner Announcement	$2\ T_{\mathrm{EXP}}$ $+ 1\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$	$481\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$	$2\ T_{EC\_MUL}$ $+ 1\ T_{\mathrm{H}}$	$58\ T_{\mathrm{MUL}}$ $+ 1\ T_{\mathrm{H}}$

As seen in the comparison table, the computation amount of ECC scheme is obviously lower than modular exponentiation scheme.

6. Conclusion

This paper puts forward an agent-based English Auction protocol to allow bidders to obtain information and participate in auctions through the assistance of an agent. Based upon our proposal, it clearly satisfies all of the security requirements of online auction protocol, such as anonymity, traceability, fairness, and so on. Because we identify the mobile devices inherently have weaker computation capability, Elliptic Curve Cryptosystem is employed on the mobile agent to achieve lower computation amount and small key size in order to reduce the time cost consumed by verification and computation. This is a means to make the online auction on mobile devices to become more efficient and convenient. As wireless networks continue to become extensively used, our proposed scheme for wireless data exchange has just met the minimum for security purpose. In the future, the key point would be focus toward enhancing the auction-related data protection.

Reference

1. Z. X. Huang, “Applying Data Mining to Analyze Online Auction Market,” Master's Thesis, Chaoyang University of Technology, Taichung, 2003.

2. F. M. Lee, J. P. Chen and J. W. Hung, “Applying Software Agent on Internet Auction and Bargaining System,” Institute of Information & Computing Machinery, Vol. 3, No. 2, pp. 67-80, 2000.

3. F. C. Peng, C. O. Chang and M. C. Chen, “A Study of Influence of Different Auction Mechanism to No-performing Assets,” Sun Yat-Sen Management Review, Vol. 16, No. 3, 2008.

4. K. H. Huang, “A Study on Mobile Computing Applications to Secure Transaction Models,” Doctoral Dissertation, National Taiwan University, Taipei, 2008.

5. B. Lee, K. Kim and J. Ma, “Efficient Public Auction with One-time Registration and Public Verifiability,” Second International Conference on Cryptology in India－INDOCRYPT 2001, pp. 162-174, Chennai, India, December 16-20, 2001.

6. K. Omote and A. Miyaji, “A Practical English Auction with One-time Registration,” 6th Australasian Conference on Information Security and Privacy, pp. 221-234, Sydney, Australia, July 11-13, 2001.

7. K. Q. Nguyen and J. Traore, “An online public auction protocol protecting bidder privacy,” 5th Australasian Conference on Information Security and Privacy, pp. 427-442, Brisbane, Australia, July 10-12, 2000.

8. T. C. Wu, K. Y. Chen and Z. Y. Lin, “An English Auction Mechanism for Internet Environment,” *ISC 2002*, pp. 331-337, 2002.

9. C. C. Chang and Y. F. Chang, “Efficient Anonymous Auction Protocols with Freewheeling Bids,” Computers and Security, Vol. 22, No. 8, pp. 728-734, 2003.

10. R. Jiang, L. Pan and J. H. Li, “An Improvement on Efficient Anonymous Auction Protocols,” Computers and Security, Vol. 24, No. 2, pp. 169-174, 2005.

11. C. C. Chang and Y. F. Chang, “Enhanced Anonymous Auction Protocols with Freewheeling Bids,” 20th International Conference on Advanced Information Networking and Applications, pp. 353-358, Vienna, Austria, April 18-20, 2006.

12. 賴松溪，韓亮與張真誠，近代密碼學及其應用，松崗出版社，2001 年 10 月。

13. W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transaction on Information Theory, Vol. 22, No.6, pp. 644-654, Nov. 1976.

14. T. ElGamal, “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” IEEE Transactions on Information Theory, Vol. 31, No. 4, pp. 469-472, 1985.

15. C. P. Schnorr, “Efficient Identification and Signature for smart Cards,” 9th Annual International Cryptology Conference on Advances in Cryptology, pp. 239-252, 1990.

16. N. Koblitz, “Elliptic Curve Cryptosystems,” Mathematics of Computation, Vol. 48, No. 177, pp. 203-209, 1987.

17. V. S. Miller, “Use of Elliptic Curves in Cryptography,” *Advances in Cryptology: Proceedings of Crypto '85*, Vol. 218, pp. 417-426, 1986.

18. S. T. Wu, “Authentication and Group Secure Communications Using Elliptic Curve Cryptography,” Doctoral Dissertation, National Taiwan University of Science and Technology, Taipei, 2005.

19. Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, “Access control in user hierarchy based on elliptic curve cryptosystem,” Information Sciences, Vol. 178, No. 1, pp. 230-243, 2008.

20. C. W. Shieh, “An Efficient Design of Elliptic Curve Cryptography Processor,” Master's Thesis, Tatung University, Taipei, 2006.

21. K. H. Huang, Y. F. Chung, C. H. Liu, F. Lai and T. S. Chen, “Efficient migration for mobile computing in distributed networks,” Computer Standards & Interfaces, Vol. 31, No. 1, pp. 40-47, 2009.

22. D. Johnson, A. Menezes and S. Vanstone, “The Elliptic curve Digital Signature Algorithm (ECDSA),” Information Security, Vol. 1, pp. 36-63, 2001.

23. D. J. Guan and L. H. Jen, “Study and Implementation of Elliptic Curve Cryptosystem,” Master's Thesis, National Sun Yat-Sen University of Technology, Kaohsiung, 2005.

24. S. G. Akl and P. D. Taylor, “Cryptographic solution to a problem of access control in a hierarchy,” ACM Transactions on Computer Systems, Vol. 1, No. 3, pp. 239-248, 1983.

25. R. Volker and J. S. Mehrdad, “Access Control and Key Management for Mobile Agents,” Computer & Graphics, Vol. 22, No. 4, pp. 457-461, 1998.

26. I. C. Lin, H. H. Ou, and M. S. Hwang, “Efficient access control and key management schemes for mobile agents,” Computer Standards & Interfaces, Vol. 26, No. 5, pp. 423-433, 2004.

27. N. Koblitz, A. Menezes, and S. Vanstone, “The State of Elliptic Curve Cryptography,” Designs, Codes and Cryptography, Vol. 19, pp. 173-193, 2000.