Journal of Computers Vol. 35 No. 1, February 2024, pp. 65-81
doi: 10.53106/199115992024023501005

Efficient First-Price Sealed E-Auction Protocol Under Secure Multi-Party Computational Malicious Model

Da-Wei Zhou, Su-Zhen Cao*, Xiao Zhao, Dan-Dan Xing, Zheng Wang

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
{1459846311, 576342353, 1563404753, 875419380, 1723863336}@qq.com

Received 11 May 2023; Revised 11 September 2023; Accepted 10 October 2023

Abstract. To solve the problems of existing e-auction protocols such as semi-trustworthiness of outsourced third parties, collusive attacks among participants, unsatisfactory decentralized structure, and inability of public verification, we propose an efficient first-price sealed e-auction protocol under a secure multi-party computational malicious model. First, the protocol combines the additive homomorphism of the ElGamal cryptographic algorithm to achieve a decentralized structure and eliminate the problem of semi-trustworthiness of outsourced third parties; it uses (n, n) threshold encryption and decryption techniques to solve the problem of collusion attacks among participants and uses Hash-based Message Authentication Code (HMAC) technology to achieve public verifiability of auction results. Additionally, the protocol proposes a method to quickly find the maximum value of the data encoding, which can avoid multiple processing of confidential data and thus effectively reduce the number of communication rounds. The combination of zero-knowledge proof and ideal/realistic simulation paradigm proves that the protocol in this paper is resistant to up to n-1 party collusion attacks and satisfies the security of the secure multi-party computational malicious model. Finally, after theoretical analysis and simulation experiments, the protocol not only satisfies higher security performance but also has greater overall operational efficiency.

Keywords: secure multi-party computing, electronic auction protocol, ElGamal cryptographic algorithm, ideal/realistic simulation paradigm

1 Introduction

In real life, traditional auctions have many inconveniences such as time, location and uncertainty of the number of bidders. With the rapid development of the Internet, people are increasingly eager to move their auctions online, making them more flexible, convenient and fast to avoid the disadvantages of traditional auction methods in real life. Based on this, online electronic auctions have been flourishing [1]. At the same time, there are many problems with e-auctions, and the more prominent one in recent years is the problem of collusive attacks by malicious participants in the auction process. Besides, many of the existing e-auction protocols have the security of the auction results in the hands of the auctioneer or a semi-trusted third party, an arrangement that carries some risk. In addition to the above two problems, the correctness of the auction results cannot be publicly verified and needs to be solved. Therefore, in the field of first-price sealed-bid auctions, it is of great academic and practical importance to explore and study new auction protocols that enable each participant to conduct auction transactions securely and efficiently [2-4].

In order to better solve the above problems existing in electronic auction, we thought of secure multi-party computation to solve this problem. Secure Multi-Party Computing (MPC) is a branch of privacy computing. Privacy computing is a collection of technologies that can analyze, compute, and integrate privacy data without revealing the privacy data itself, thus achieving the purpose of “usability and invisibility” of privacy data [5]. Compared with traditional data usage, privacy computing not only maximizes the security of private data, but also facilitates the integration of multiple data resources to maximize data value [6-8]. MPC is a multi-party private computation technique that does not require a trusted third party, and was proposed by Turing Award winner Andrew Chi-Chih Yao in 1982 to answer the millionaire problem [9]. MPC allows multiple participants to jointly compute an objective function while guaranteeing that each party only obtains its own computational results and cannot infer the input data of any other party [10-12].

* Corresponding Author

To implement an efficient sealed electronic auction protocol using secure multiparty computation, the following issues should be considered. The first consideration is the privacy issue, where the bidding information of each participant in the auction process is completely confidential, and once it is leaked, it may cause irreparable economic losses. Secondly, the secure transmission of secret data should be considered, and the security of data transmission between each participant should be guaranteed. Thirdly, we need to consider the fraud problem in the auction process and eliminate the collusion attack between participants. Finally, the public verifiability of auction results should be considered to ensure that the legitimacy and correctness of the final winner can be verified by all participants.

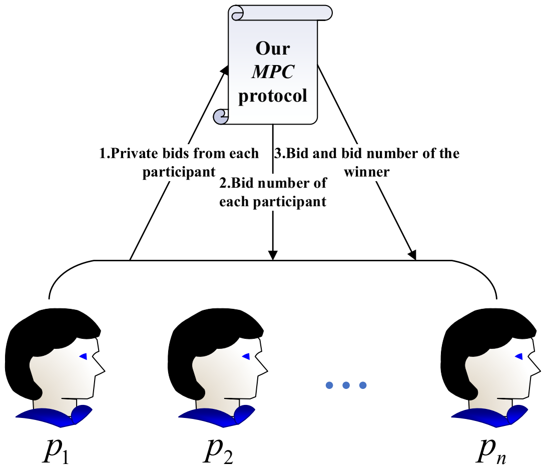
Our
MPC
protocol
1.Private bids from each participant
2.Bid number of each participant
3.Bid and bid number of the winner
p_1
p_2
p_n

Fig. 1. Protocol overview

In summary, based on secure multi-party computation, this paper designs a secure and efficient sealed-bid electronic auction protocol in the malicious model. The protocol not only satisfies the security of first-price sealed auction, but also solves the problems existing in current electronic auction, such as the semi-credibility of outsourcing third party, collusion attack between participants and the auction result cannot be publicly verified. As shown in Fig. 1, the protocol as a whole is divided into three modules. Firstly, each bidder sends its private bid to the protocol. Secondly, the protocol generates a bid number with unique authentication for each bidder based on the private bid. At last, the protocol finally outputs the bid number and bid of the winner of the auction.

This article mainly consists of eight sections. In the first section, we briefly introduce the research status of electronic auction and some existing problems, and lead to the research motivation and research methods of this text. In Section 2, we state the comparison of related works as well as the main contributions of the paper. In Section 3, we introduce some preliminary knowledge. The details of the protocol construction are described in Section 4. In Sections 5 and 6 of the protocol, we introduce the correctness proof and security analysis, respectively. We present the efficiency analysis in Section 7 and the conclusion in Section 8.

2 Related Work and Our Contributions

In recent years, first-price sealed e-auctions have developed rapidly. Brandt [13] proposed the first sealed e-auction protocol without the presence of a third party based on homomorphic cryptography and well safeguards the privacy of bidders, but the computational complexity of the protocol makes it computationally expensive. Bogetoft et al. [14] combined secret sharing with threshold cryptography to propose a new electronic auction protocol that uses linear secret sharing technique instead of trusted third parties, but the protocol requires multiple rounds of interaction to share data and is therefore inefficient. The scheme of Wu et al. [15] removes the participation of auction third parties based on a margin deduction authentication mechanism, which also leads to a protocol that requires a higher number of rounds of communication and is therefore less efficient. Sun et al. [16] used an authentication-enabled group signature technique proposed an improved sealed electronic auction protocol, which is able to resist collusion attacks among auction organizers, and the protocol is not highly applicable due to the overly cumbersome process of proving the security in the auction. Cheng et al. [17] proposed a sealed auction protocol based on digital signature technology, which has a more complex overall structure due to the participation of auxiliary third parties, and the protocol has the hidden danger of collusion between semi-trustworthy third parties and bidders. With the development of blockchain technology, many scholars began to study electronic auction protocols on blockchain, such as the auction protocol proposed by Galal et al. [18] based on blockchain and zero-knowledge proof, and the auction protocol proposed by Xiong et al. [19] based on blockchain and blind signature technology, although both of these protocols are based on blockchain technology making electronic auctions with advantages such as openness, transparency, and tamper-evident , there are still some shortcomings in terms of system privacy leakage protection, public verification of all members, and resistance to collusive attacks by malicious participants. The comparison of related works is shown in Table 1.

Table 1. The comparison of related works

	[13]	[14]	[15]	[16]	[17]	[18]	[19]	Ours
Decentralization	√	√	√	×	×	√	√	√
Resisting collusion attacks	×	×	×	√	×	×	×	√
Public verifiability	×	×	×	×	×	×	×	√
Computational cost	High	High	High	High	Low	Low	Low	Low

The main contributions of this paper are fourfold as follows.

1) This paper combines secure multi-party computing with ElGamal homomorphic encryption to achieve a decentralized structure without the help of a semi-trusted third party. And eliminate the security risks of outsourcing third-party semi-trust, so that each participant can participate in the auction process fairly and safely.

2) In this paper, the collusion problem between participants is solved by combining threshold encryption and decryption technology and zero-knowledge proof. And through the ideal/reality simulation paradigm, it is proved that the protocol meets the security of the secure multi-party computational malicious model and can resist the collusive attack of malicious participants.

3) In this paper, HMAC technology is used to realize the public verifiability of auction results, and each participant of the protocol can verify the correctness and legitimacy of auction results locally.

4) This paper presents a method to quickly find the maximum value of data encoding, which can quickly determine the winner of an auction. This avoids multiple processing of confidential data and can effectively reduce the number of communication rounds. After experimental analysis, the protocol has high operating efficiency and constant communication overhead, so it has a high comprehensive efficiency advantage.

3 Preliminary Knowledge

3.1 Homomorphism of the ElGamal Cryptographic Algorithm

The ElGamal public-key cryptographic algorithm [20] is satisfying multiplicative homomorphism, such that multiplication of two ciphertexts $C_1 = (g^{r_1}, m_1h^{r_1})$ and $C_2 = (g^{r_2}, m_2h^{r_2})$ can yield a new ciphertext $C = (g^{r_1+r_2}, m_1m_2h^{r_1+r_2})$, and then decryption of the newly obtained ciphertext can exactly yield the plaintext m_1m_2, i.e., $E(m_1)E(m_2) = E(m_1m_2)$. Where m_1 and m_2 are the plaintexts corresponding to C_1 and C_2 respectively, g is the generating element of Z_p^*, h is the public key, r_1 and r_2 are the respective private keys.

3.2 Full Threshold Public Key Cryptosystem

Threshold encryption and threshold decryption are a combination of encryption and decryption schemes and secret sharing techniques [21], where the (t, n) threshold can be constructed to resist collusion attacks by up to t-1 malicious participants. In order to make the secure multi-party computing protocol have a higher level of resistance to collusion attacks by up to n-1 malicious participants, an (n, n) threshold scheme needs to be constructed [22].

See Section 3.2 for details of the construction.

3.3 Malicious Models for Secure Multiparty Computing

In contrast to the fully trustworthy of the ideal model of secure multi-party computation [23] and the honest but curious of the semi-honest model [23], the malicious model may not run the protocol honestly and may even engage in sabotage. Malicious actions of attackers include, but are not limited to, illegitimate inputs, maliciously tampering with inputs, recording and analyzing private data of honest parties, maliciously aborting protocols, and refusing to execute protocols. The security requirement of a secure multi-party computing protocol under the malicious model is to detect and block malicious operations. Proving the security of a secure multi-party computation protocol under the malicious model is also to prove that the protocol satisfies the security definition of the malicious model.

First introduce the ideal protocol under the malicious model [23]: let participant P_i have confidential data x_i. The participants have to compute the function $f(x_1, x_2, ..., x_n)$ with the help of a Trusted Third Party (TTP). Let the set of malicious participants be $I = \{i_1, \cdots, i_t\} \subseteq \{1, \cdots, n\}$, then $\overline{I} = [n] \setminus I$ is the set of honest participants. Let $\overline{x} = (x_1, x_2, ..., x_n)$, $\overline{x_I} = (x_{i_1}, x_{i_2}, ..., x_{i_t})$, $f(\overline{x}) = f_1(\overline{x}), ..., f_n(\overline{x}))$, $f_I(\overline{x}) = f_{i_1}(\overline{x}), ..., f_{i_t}(\overline{x}))$, where the interaction action is as follows.

1) TTP collects data: if $i \in \overline{I}$, then TTP receives real data x_i from P_i; if $i \in I$, then according to the policy of x_i and I, TTP decides not to perform any operation, i.e., there is no data interaction, or TTP sends data that has no real meaning x_i.

2) TTP sends data to members in I: TTP computes $f(\overline{x})$ independently after receiving $\overline{x}$, and subsequently sends $f_I(\overline{x})$ to all members in I; otherwise, TTP sends $\perp$ to all members in I.

3) TTP sends data to members in $\overline{I}$: If $P_1 \in I$, and $f_I(\overline{x})$ is received by members in I, members in I decide whether TTP sends $f_{\overline{I}}(\overline{x})$ to members in $\overline{I}$. When members in I allow, TTP sends $f_{\overline{I}}(\overline{x})$ to members in $\overline{I}$. When the member in I does not allow, TTP sends $\perp$ to the member in $\overline{I}$.

Definition 1. Suppose that all members in I are controlled by some attacker (I, B), where B denotes the probabilistic polynomial-time algorithm, i.e., the protocol execution policy, owned by that attacker. In the ideal model, (I, B) knows that the auxiliary message z, and chooses a random number r. When the input is $\overline{x} = (x_1, x_2, ..., x_n)$, the joint execution operation of f is noted as $IDEAL_{f, I, B(z)}(\overline{x}) = \Upsilon(\overline{x}, I, z, r)$, where $\Upsilon(\overline{x}, I, z, r)$ is defined as follows.

1) If P_1 is honest, i.e., $P_1 \notin I$, then $\Upsilon(\overline{x}, I, z, r) = (f_{\overline{I}}(\overline{x}), B(\overline{x_I}, I, z, r, f_I(\overline{x})))$ where $\overline{x} = (x_1, ..., x'_n)$. If $i \in I$ then $x_i = B(\overline{x_I}, I, z, r)_i$; otherwise $x_i = x_i$.

2) If P_1 is dishonest and $B(\overline{x_I}, I, z, r, f_I(\overline{x})) = \perp$, then $\Upsilon(\overline{x}, I, z, r) = (\perp^{|\overline{x}|}, B(\overline{x_I}, I, z, r, f_I(\overline{x}), \perp))$.

3) If P_1 is dishonest but $B(\overline{x_I}, I, z, r, f_I(\overline{x})) \neq \perp$ then $\Upsilon(\overline{x}, I, z, r) = (f_{\overline{I}}(\overline{x}), B(\overline{x_I}, I, z, r, f_I(\overline{x})))$.

Definition 2. (Security under the malicious model [23]) Assuming that $f : (\{0, 1\}^*)^n \to (\{0, 1\}^*)^n$ is an n-element function and π is a protocol for computing f. Let I, $\overline{I}$, $\overline{x}$, $(\overline{x})_I$, $f(\overline{x})$ and $f_I(\overline{x})$ have the same definition as the previous equation. (I, A) denotes the attacker in the actual protocol where A denotes the probabilistic polynomial-time algorithm, i.e., the protocol execution policy, possessed by this attacker. In the actual model, (I, A) knows the auxiliary message z. When the input is $\overline{x} = (x_1, x_2, ..., x_n)$, the message output sequence after the n participants in π perform the joint operation is noted as $REAL_{\pi, I, A(z)}(\overline{x})$. $A(\overline{x_I}, I, z)$ determines the message sequence of the participants in I, and the protocol π determines the message sequence of the participants in $\overline{I}$. That is, A determines the messages of the malicious attacker based on the messages of all participants (including the input messages of all malicious participants, auxiliary messages, and messages sent by all honest participants).

If any probabilistic polynomial-time algorithm A representing the malicious attacker's execution strategy in the actual protocol, and the probabilistic polynomial-time algorithm B representing the malicious attacker's execution strategy in the ideal protocol corresponding to it always exists. Then for any $I \subseteq [n]$, we can calculate the following equation.

$$\{IDEAL_{f,I,B(z)}(\overline{x})\}_{\overline{x},z} \overset{c}{\equiv} \{REAL_{\pi,I,A(z)}(\overline{x})\}_{\overline{x},z} \quad . \tag{1}$$

Where $\overset{c}{\equiv}$ denotes computational indistinguishability, so that the function f can be computed safely, i.e., the protocol π is secure under the malicious model.

3.4 Ideal/Realistic Simulation Paradigm

Security proofs for secure multi-party computation mainly use a constructive proof approach, i.e., abstract simulation through computational complexity. Fully trusted TTP is hard to find in real life, so there is no TTP in actual secure multi-party computing protocols. In the process of proving the security of secure multi-party computation protocols under the realistic model, we can consider the secure multi-party computation protocol in the actual realistic model as the secure multi-party computation protocol in the ideal model with the highest security. Assuming that an attacker does not gain more information by attacking a protocol τ under the realistic model than by attacking a protocol π under the ideal model, it can be said that τ is at least as secure as π [24].

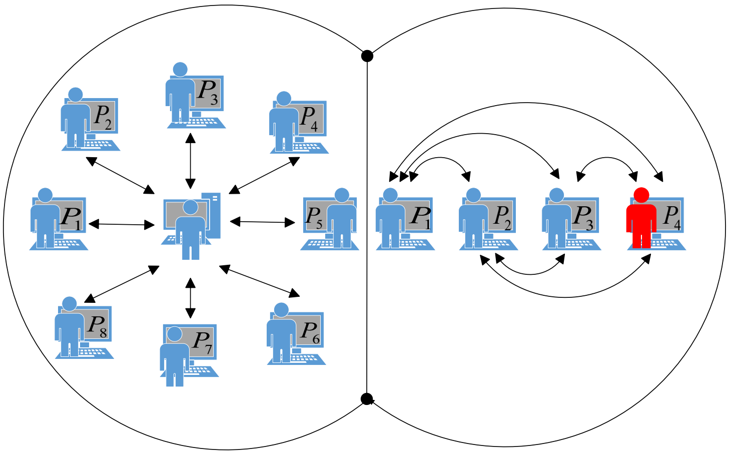

Fig. 2. Ideal/realistic simulation paradigm

Specifically, as shown in Fig. 2, the ideal/realistic simulation paradigm uses simulation to establish a statute relationship between the realistic model and the ideal model, a relationship that attributes security under the realistic model to the security of the ideal model. A general framework for such security proofs is generally to construct a simulator S to simulate the same attack behavior of the adversary in the ideal model and the real model, respectively. Next, the global output of the true model and the ideal model are calculated respectively. Then it is proved that the outputs of the two models are computationally indistinct, so that the secure multi-party computation protocol in the real model has the same security as that in the ideal model.

3.5 A Fast Coding Method for Finding the Maximum Value

In this paper, we propose a fast-coding method to find out the maximum value in a set of data, which can be applied to effectively reduce the number of communication rounds of the protocol and improve the communication efficiency of the protocol. The details are as follows.

Let n participants $P_i\,(i = 1, 2, ..., n)$ hold data $(x_1, x_2, ..., x_n)$ respectively, where $x_i \in \{z_1, z_2, \cdots, z_l\} \subseteq U$, and U denote the full set. Let $z_1 < z_2 < ... z_l$, and $|U| = l$ [25].

1) P_i encode the data x_i into the corresponding array $X_i = \{x_{i1}, x_{i2}, ..., x_{il}\}$, the equation of the encoding is shown below.

$$x_{il} = \begin{cases} r_{ij}, x_i = z_j \\ 1, x_i \neq z_j \end{cases}. \tag{2}$$

Where r_{ij} is a random number not equal to 1, $r_{ij} \in Z_p^*$, $2 \leq r_{ij} \leq p - 1$, and p is the modulus of the algorithm. Each participant encodes the data held by itself according to Eq. (2) and finally obtains the corresponding array X_i.

2) A new array $Y=\{y_1,y_2,\cdots,y_l\}=\{\prod_{i\square 1}^{\square} x_{i1},\prod_{i\ 2} x_{i2},\cdots,\prod_{i\ l} x_{il}\}$ is obtained by multiplying *n* participants by the corresponding positions in *n* arrays X_1, X_2, ..., X_n.

3) In the new array *Y*, search in the direction from right to left, and when $y_j = r_{ij}$ appears for the first time, the position of y_j is the position where the maximum value in the *n* data is located in the full set *U*, and the value of the element in that position in the full set *U* is numerically equal to the maximum value in (x_1, x_2, ..., x_n), i.e., max$\{x_1, x_2, ..., x_n\} = z_j$.

For example, there are 5 participants whose holdings are $x_1 = 11$, $x_2 = 13$, $x_3 = 19$, $x_4 = 15$ and $x_5 = 17$. Let $U_1 = \{z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}, z_{11}\} = \{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$.

According to Eq. (2), it can be derived that $X_1 = \{1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$, $X_2 = \{1, 1, 1, 7, 1, 1, 1, 1, 1, 1, 1\}$, $X_3 = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1\}$, $X_4 = \{1, 1, 1, 1, 1, 9, 1, 1, 1, 1, 1\}$, $X_5 = \{1, 1, 1, 1, 1, 1, 1, 6, 1, 1, 1\}$.

Then multiply the elements of the above 5 arrays in the corresponding positions to find a new array $Y = \{1, 5, 1, 7, 1, 9, 1, 6, 1, 4, 1\}$.

Next, retrieve the position number z_{10} in U_1 corresponding to the position of the first occurrence of the number not 1 in the array *Y* in the direction from right to left, and then the maximum value max$\{x_1, x_2, ..., x_n\} = z_{10} = 19$ can be obtained.

4 First-price Sealed Electronic Auction Protocol Under the Malicious Model

4.1 Protocol Environment and Framework

The agreement involves two types of entities, one is the bidder P_i (i = 1, 2, ..., n) and the other is the bulletin board server.

1) Bidder, a person who participates in the auction using their respective terminals in accordance with the auction rules and procedures.

2) Bulletin Board Server, publish auction information such as auction task, auction rules, and auction time.

Among them, all bidders can use their respective terminals to access the bulletin board server, and can interact with the bulletin board server after registration and qualification verification. The intermediate results of the auction process involving the bulletin board server are open and transparent, and all bidders can monitor and access them. The overall framework of the protocol is shown in Fig. 3.

4.2 The Specific Process of Electronic Auction Protocol

The whole protocol is divided into four phases, which are parameter generation phase, data preprocessing phase, auction execution phase, and result verification phase.

1) Parameter generation phase

Some parameters are generated in this phase, such as each participant generates the corresponding public key according to the private key and random number they choose, and the bulletin board server collects the public key to generate the joint public key and HMAC key, etc. The specific process is detailed in Algorithm 1.

2) Data preprocessing phase

In this phase, each participant's private data is pre-processed, such as encoding the private data and then encrypting it into the corresponding cipher text. The specific process is detailed in Algorithm 2.

3) Auction execution phase

This phase mainly involves the specific execution of the auction, where the bid number of each bidder and the joint decryption key are obtained based on the cryptographic data of each participant and the partial decryption key, and finally the decryption ciphertext is decoded to obtain the bid number and the bid of the winning bidder. The specific process is detailed in Algorithm 3.

4) Result verification phase

This phase focuses on the verification of the auction results and involves the bulletin board server verifying the computed results for each auction participant, followed by the participants' verification of the bulletin board server's output. The specific process is detailed in Algorithm 4.

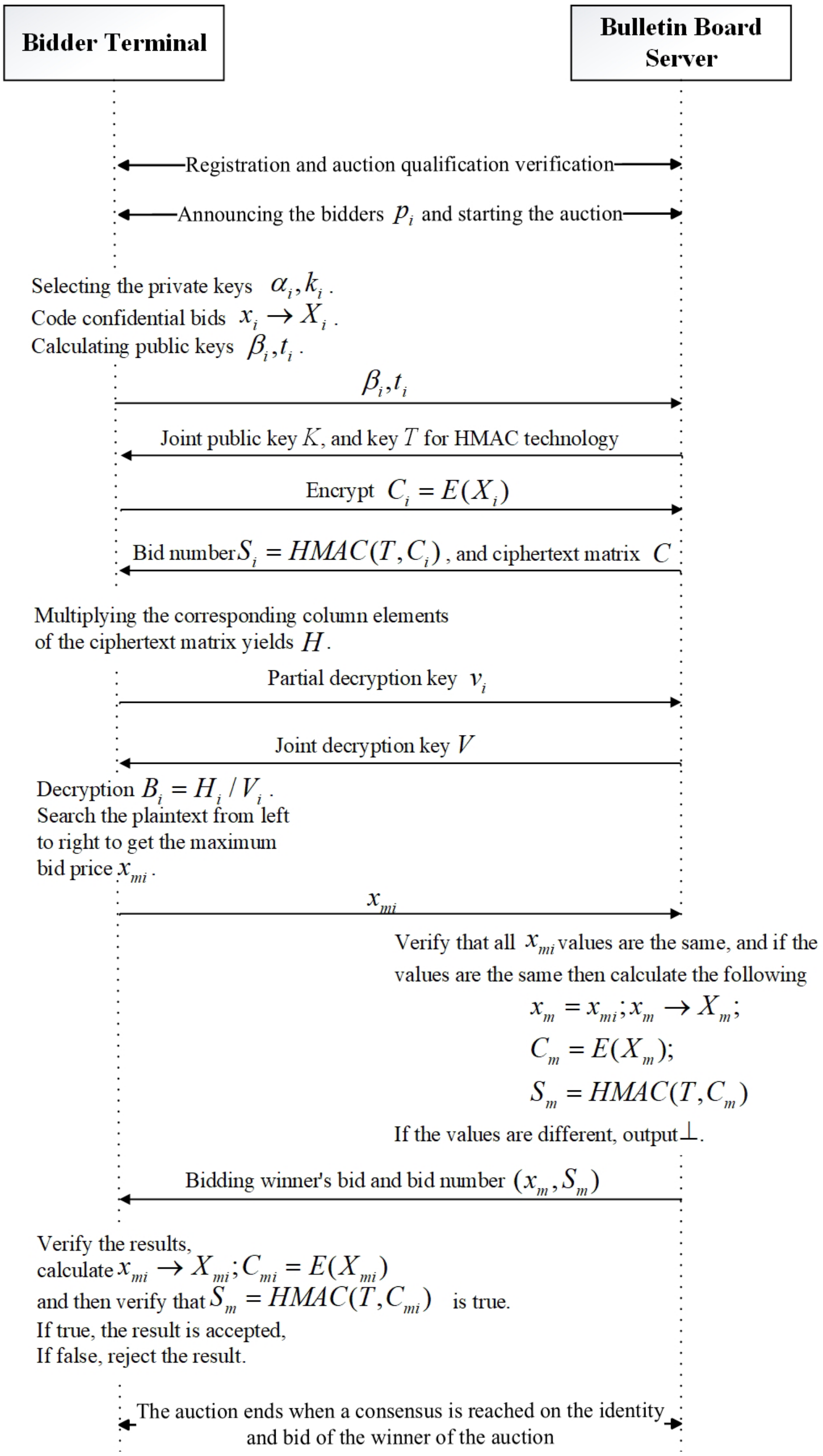
Bidder Terminal
Bulletin Board Server
Registration and auction qualification verification
Announcing the bidders p_i and starting the auction
Selecting the private keys α_i, k_i.
Code confidential bids $x_i \to X_i$.
Calculating public keys β_i, t_i.
β_i, t_i
Joint public key K, and key T for HMAC technology
Encrypt $C_i = E(X_i)$
Bid number $S_i = HMAC(T, C_i)$, and ciphertext matrix C
Multiplying the corresponding column elements of the ciphertext matrix yields H.
Partial decryption key v_i
Joint decryption key V
Decryption $B_i = H_i / V_i$.
Search the plaintext from left to right to get the maximum bid price x_{mi}.
x_{mi}
Verify that all x_{mi} values are the same, and if the values are the same then calculate the following
$x_m = x_{mi}; x_m \to X_m;$
$C_m = E(X_m);$
$S_m = HMAC(T, C_m)$
If the values are different, output $\perp$.
Bidding winner's bid and bid number (x_m, S_m)
Verify the results,
calculate $x_{mi} \to X_{mi}; C_{mi} = E(X_{mi})$
and then verify that $S_m = HMAC(T, C_{mi})$ is true.
If true, the result is accepted,
If false, reject the result.
The auction ends when a consensus is reached on the identity and bid of the winner of the auction

Fig. 3. Overall framework of the protocol

Algorithm 1. Parameter generation process

Input:
The bidder's private key α_i
The random number k_i chosen by the bidder

Output:
The bidder's public key β_i
The bidder's partial HMAC key t_i
The joint public key K of all bidders
The complete HMAC key T

Procedures:

Step 1: After the auction starts, each of the n bidders $P_i(i=1,2,\cdots,n)$ selects a large prime p, a generating element $g \in Z_p^*$, a plaintext set $M = Z_p^*$, and a ciphertext set $C = Z_p^* \times Z_p^*$ based on the ElGamal cryptographic algorithm.

Step 2: P_i randomly selects the integer $\alpha_i(i=1,2,\cdots,n)$, where $\alpha_i \in \mathbb{Z}_{p-1}$, and then computes its own public key $\beta_i = g^{\alpha_i} (\mathrm{mod}\, p)$.

Step 3: P_i randomly selects $k_i \in \mathbb{Z}_{p-1}$, and $\gcd(k_i, p-1)=1$, followed by the calculation of $t_i = g^{k_i} \bmod p$.

Step 4: P_i sends β_i, t_i to the bulletin board server.

Step 5: The bulletin board server calculates the joint public key $K = \prod_{i=1}^{n} \beta_i = g^{\sum_{i=1}^{n} \alpha_i} (\mathrm{mod}\, p)$, the key $T = \prod_{i=1}^{n} t_i = \prod_{i=1}^{n} g^{k_i} (\mathrm{mod}\, p)$ of HMAC, and sends K and T to P_i.

Algorithm 2. Data preprocessing process

Input:
The private bids X_i from bidders

Output:
The array X_i corresponding to private bids after coding

The ciphertext C_i corresponding to the encrypted array X_i

Procedures:

Step 1: P_i encodes the private bids X_i held by himself into the corresponding array X_i according to equation (2), where $X_i = \{x_{i1}, x_{i2}, \cdots, x_{il}\}$.

Step 2: P_i encrypts each element of the array X_i separately using the joint public key K to obtain the ciphertext $C_i = E(X_i)$. For example, encrypting element x_{il} in X_i yields $C_{il} = E(x_{il}) = x_{il} K^{k_i} (\mathrm{mod}\, p)$.

Step 3: P_i then sends C_i to the bulletin board server.

Algorithm 3. Auction execution process

Input:
The complete HMAC key T
The ciphertext C_i corresponding to the encrypted array X_i
The partial decryption key v_i

Output:
The bid number S_i with unique identification

The ciphertext matrix C
The new ciphertext H after the multiplication operation
The complete decryption key V
The maximum bid x_{mi} obtained by the local decryption calculation of bidder P_i

Procedures:

Step 1: The bulletin board server generates bid number $S_i = HMAC\ (T, C_i)$, $(i = 1, 2, ..., n)$ with unique identification for each bidder P_i separately.

Step 2: The bulletin board server generates the ciphertext matrix C based on the received C_i, where the procedure for calculating C is shown in Eq. (3).

$$C = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1l} \\ C_{21} & C_{22} & \cdots & C_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nl} \end{pmatrix}. \tag{3}$$

Step 3: P_i in the ciphertext matrix C and multiply each of its column elements to obtain the new ciphertext $H = (H_1, H_2, \cdots, H_l) = (\prod_{i=1}^{n} C_{i1}, \prod_{i=1}^{n} C_{i2}, \cdots, \prod_{i=1}^{n} C_{il})$.

Step 4: P_i sends the partial decryption key $v_i = T^{\alpha_i} (\mathrm{mod}\ p) = (\prod_{i=1}^{n} g^{k_i})^{\alpha_i} (\mathrm{mod}\ p)$ to the bulletin board server. P_i has to prove to the bulletin board server and all other bidders that the v_i it provided is correct by using zero-knowledge proof before sending v_i.

Step 5: The bulletin board server receives v_i and calculates the joint decryption key $V = \prod_{i=1}^{n} v_i$, then sends V to P_i.

Step 6: P_i decrypts the ciphertext $H = (H_1, H_2, ..., H_l)$ locally in order from right to left according to Eq. (4), and terminates the decryption when retrieving the first element y_j equal to the random number r_{ij}. The element at the corresponding position in the full set U is the maximum value, i.e., the maximum bid $x_{mi} = \max\ \{x_1, x_2, ..., x_n\}$ is found. All bidders send their calculated x_{mi} to the bulletin board server.

$$B_i = H_i / V. \tag{4}$$

Algorithm 4. Result verification process

Input:

The maximum bid x_{mi} obtained by the local decryption calculation of bidder P_i

Output:

The bid and bid number (x_m, S_m) of the winner of the auction or $\perp$

Procedures:

Step 1: The bulletin board server verifies that all values x_{mi} received are the same. If they are the same, make $x_m \rightarrow x_{mi}$, encode $x_m \rightarrow X_m$ first, then encrypt to get $C_m = E(X_m)$. Next, calculate the bid number $S_m = HMAC\ (T, C_m)$ of the winner, and finally send the bid of the winner and the bid number (x_m, S_m) to all bidders. If all x_{mi} received are verified to be different then output $\perp$.

Step 2: P_i receives (x_m, S_m) and verifies the result by calculating whether S_m can be derived from the locally derived x_{mi}, i.e., whether Eq. (5) holds. If it holds, accept it, if not, output $\perp$.

$$S_m = HMAC(T, C_{mi}). \tag{5}$$

Step 3: At this point, all n bidders P_i reach a consensus on the bid and bid number (x_m, S_m) of the bid winner, and the entire auction process ends.

The main execution process of the protocol is shown in Fig. 4.

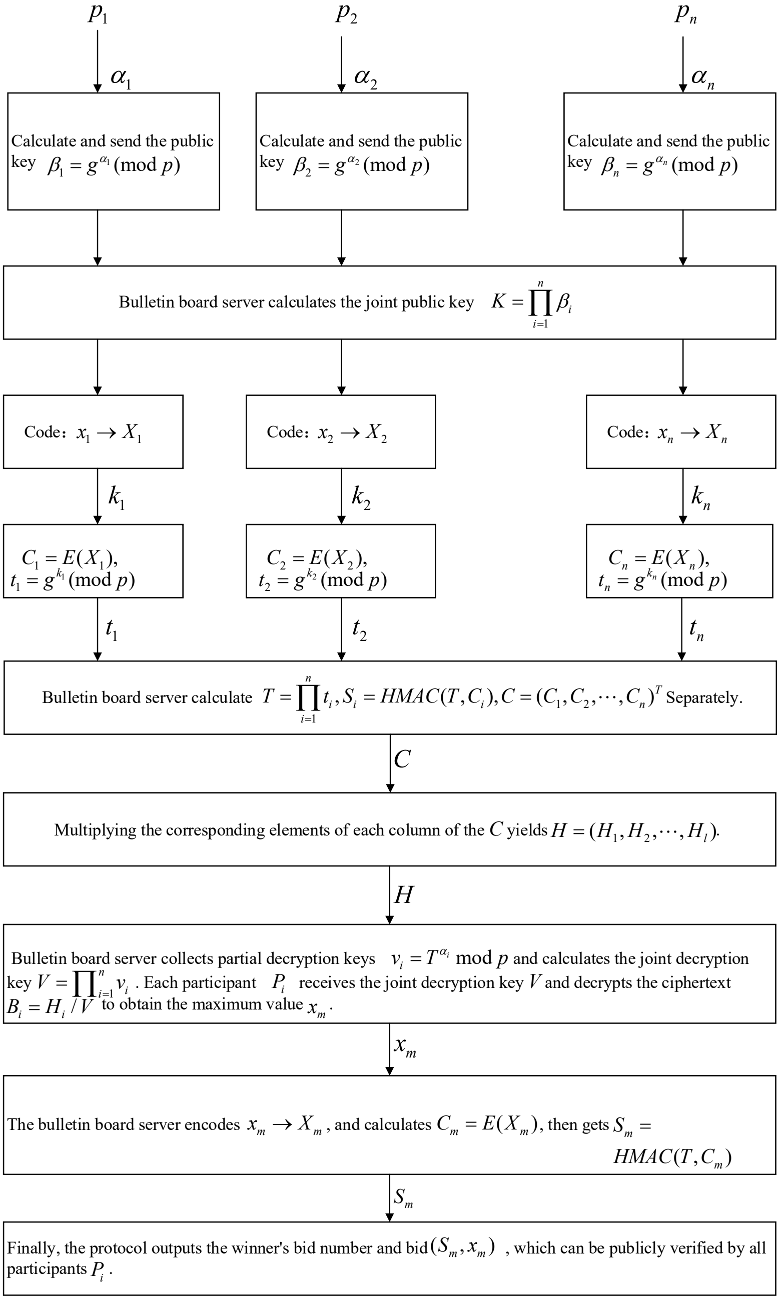
p_1
p_2
p_n
α_1
α_2
α_n
Calculate and send the public key $\beta_1 = g^{\alpha_1} (\bmod p)$
Calculate and send the public key $\beta_2 = g^{\alpha_2} (\bmod p)$
Calculate and send the public key $\beta_n = g^{\alpha_n} (\bmod p)$
Bulletin board server calculates the joint public key $K = \prod_{i=1}^{n} \beta_i$
Code: $x_1 \to X_1$
Code: $x_2 \to X_2$
Code: $x_n \to X_n$
k_1
k_2
k_n
$C_1 = E(X_1)$, $t_1 = g^{k_1} (\bmod p)$
$C_2 = E(X_2)$, $t_2 = g^{k_2} (\bmod p)$
$C_n = E(X_n)$, $t_n = g^{k_n} (\bmod p)$
t_1
t_2
t_n
Bulletin board server calculate $T = \prod_{i=1}^{n} t_i, S_i = HMAC(T, C_i), C = (C_1, C_2, \cdots, C_n)^T$ Separately.
C
Multiplying the corresponding elements of each column of the C yields $H = (H_1, H_2, \cdots, H_l)$.
H
Bulletin board server collects partial decryption keys $v_i = T^{\alpha_i} \bmod p$ and calculates the joint decryption key $V = \prod_{i=1}^{n} v_i$. Each participant P_i receives the joint decryption key V and decrypts the ciphertext $B_i = H_i / V$ to obtain the maximum value x_m.
x_m
The bulletin board server encodes $x_m \to X_m$, and calculates $C_m = E(X_m)$, then gets $S_m = HMAC(T, C_m)$
S_m
Finally, the protocol outputs the winner's bid number and bid (S_m, x_m) , which can be publicly verified by all participants P_i.

Fig. 4. The main execution process of the protocol

5 Correctness of Threshold Encryption and Decryption

Theorem 1. In our proposed efficient first-price sealed electronic auction protocol under the secure multi-party computational malicious model, each bidder $P_i (i = 1, 2, ..., n)$ can correctly obtain the plaintext $B_i (i = 1, 2, ..., n)$ by jointly decrypting the ciphertext $H(H_1, H_2, ..., H_l)$ with a joint decryption key V.

Proof. According to Algorithm 3 we know that each bidder P_i at this phase knows the ciphertext H to be decrypted and the joint decryption key V. Therefore, the calculation according to Eq. (4) can be obtained as follows.

$$\begin{aligned}
B_i &= \frac{H_i}{V} (\bmod p) \\
&= \frac{\prod_{i=1}^{n} (x_{ij} \cdot K^{k_i})}{\prod_{i=1}^{n} ((\prod_{i=1}^{n} g^{k_i})^{\alpha_i})} (\bmod p) \\
&= \frac{\prod_{i=1}^{n} (x_{ij} \cdot g^{k_i \sum_{i=1}^{n} \alpha_i})}{(\prod_{i=1}^{n} g^{k_i})^{\sum_{i=1}^{n} \alpha_i}} (\bmod p) \\
&= \frac{(\prod_{i=1}^{n} x_{ij}) \cdot (\prod_{i=1}^{n} g^{k_i \sum_{i=1}^{n} \alpha_i})}{(g^{\sum_{i=1}^{n} k_i})^{\sum_{i=1}^{n} \alpha_i}} (\bmod p) \\
&= \frac{(\prod_{i=1}^{n} x_{ij}) \cdot (g^{(\sum_{i=1}^{n} k_i) \cdot (\sum_{i=1}^{n} \alpha_i)})}{g^{(\sum_{i=1}^{n} k_i) \cdot (\sum_{i=1}^{n} \alpha_i)}} (\bmod p) \\
&= \prod_{i=1}^{n} x_{ij}
\end{aligned}$$

According to the above derivation process, we can obtain the decrypted plaintext $\prod_{i=1}^{n} x_{ij}$. Therefore, decrypting the ciphertext H by Eq. (4) can decrypt the plaintext B correctly. The proof is over.

6 Security Analysis

The general idea of proving that a secure multi-party computation protocol is secure under the malicious model is that from Definition 1 and Definition 2, to prove that a computation protocol π is secure in secure multi-party computation, it is necessary to satisfy that an arbitrary probabilistic polynomial-time algorithm A representing the malicious attacker's execution policy in the actual protocol, and a probabilistic polynomial-time algorithm B representing the malicious attacker's execution policy in the ideal protocol corresponding to it always exists, and can prove that the probabilistic polynomial-time algorithms A and B make the Eq. (1) hold.

Theorem 2. The first-price sealed e-auction protocol based on secure multi-party computation is secure under the malicious model.

Proof. In a plain secure multi-party computing protocol, each participant P_i ($i = 1, 2, ..., n$) is in the same functional position. In the presence of a malicious participant, the most serious threat to an honest participant comes from a collusive attack by a malicious participant [23-25]. Therefore, consider constructing a maximal attacker set with n-1 malicious participants in the set with the most malicious participants involved in collusive attacks. If the protocol can be shown to be secure for the maximal attacker set; then it is also secure for any other non-maximal attacker set. Without loss of generality, it is useful to assume that P_n is honest if the first n-1 participants are malicious. The malicious participants perform collusive attacks, in which case the maximum set of attackers is $I = \{P_1, ..., P_{n-1}\}$ and the set of private data is $X = \{x_1, ..., x_n\}$.

Algorithm A represents an arbitrary probabilistic polynomial-time algorithm under the execution policy of a malicious attacker in a realistic protocol, and Algorithm B represents a probabilistic polynomial-time algorithm under the execution policy of a malicious attacker in an ideal protocol. Suppose $h_j = \max\{x_1, ..., x_n\}$, in decrypting the ciphertext $C_t = E(X_i) = x_{ij} \cdot K^{k_i} \bmod p$, P_n sends $t_i = g^{k_i} \bmod p$ and the zero-knowledge proof message l_i, where l_i is used to prove that t_i is correct. The following is considered in two cases.

1) In executing the actual protocol, it is known that the input to the colluder depends on the probabilistic polynomial-time algorithm A described earlier, then the input to the protocol is denoted as $X = (A(x_1, ..., x_{n-1}), x_n)$. The protocol is aborted if any participant $P_i \in I$ is not able to prove with zero-knowledge proof message l_i that the t_i it provides to the other participants for decryption is correct. Then according to the attacker's probabilistic polynomial-time algorithm A, the attacker gets the output $A(X_I, I, r, z, C_i, t_i, l_i, f(X))$, so the following equation is obtained.

$$\{REAL_{\pi,I,A(z)}(X)\}_{X,z} = \{A(X_I,I,r,z,C_i,t_i,l_i,f(X)),\perp\} \, . \tag{6}$$

And if the protocol is not aborted, then the following equation is obtained.

$$\{REAL_{\pi,I,A(z)}(X)\}_{X,z} = \{A(X_I,I,r,z,C_i,t_i,l_i,f(X)),f(X) \, . \tag{7}$$

2) In the execution of the ideal protocol, all participants P_i send their private data x_i to the TTP, then P_n sends x_n to the TTP. X_I is provided to A by the probabilistic polynomial-time algorithm B, from which $A(X_I)$ can be obtained and then sent to the TTP. the TTP finally obtains $X = (A(X_I), x_n) = (A(x_1, ..., x_{n-1}), x_n)$, then the TTP computes $f(X)$ and sends $f(X)$ to B. B will randomly choose x_n' such that $f = (A(x_1, ..., x_{n-1}), x_n')$ is equal to $f = (A(x_1, ..., x_{n-1}), x_n)$. B will provide I with the ciphertext C_i' of x_n and the t_i' and l_i' needed for the zero-knowledge proof. If the protocol is aborted because any of the conspirators does not have the zero-knowledge proof, then B gets the output $A(X_I, I, r, z, C_i', t_i', l_i', f(X))$, so the following equation is obtained.

$$\{IDEAL_{f,I,B(z)}(X)\}_{X,z} = \{A(X_I,I,r,z,C_i',t_i',l_i',f(X)),\perp\} \, . \tag{8}$$

If the protocol is not aborted, then the following equation is obtained.

$$\{IDEAL_{f,I,B(z)}(X)\}_{X,z} = \{A(X_I,I,r,z,C_i',t_i',l_i',f(X)),f(X)\} \, . \tag{9}$$

Combining the two cases above, it can be found that the output of P_n under $\{IDEAL_{f,I,B(z)}(X)\}_{X,z}$ and $\{REAL_{\pi,I,A(z)}(X)\}_{X,z}$ is the same. Meanwhile, because the semantic security of zero-knowledge proof theory and the ElGamal algorithm can guarantee the computational indistinguishability between t_i and t_i', l_i and l_i', C_i and C_i', the following equation can be obtained.

$$A(X_I,I,r,z,C_i',t_i',l_i',f(X)) \overset{c}{\equiv} A(X_I,I,r,z,C_i,t_i,l_i,f(X)) \cdot \tag{10}$$

Therefore, the following equation can be obtained.

$$\{IDEAL_{f,I,B(z)}(X)\}_{X,z} \overset{c}{\equiv} \{REAL_{\pi,I,A(z)}(X)\}_{X,z} \cdot \tag{11}$$

And from equation (1) and equation (11), we can obtain that the protocol is secure under the malicious model and can resist the collusive attack of $n-1$ participants. The proof is over.

7 Performance and Efficiency Analysis

7.1 Safety Performance Comparison

In this section, some articles in recent years are selected to make some comparisons with this paper in terms of security performance. In addition to the basic security requirements such as fairness, anonymity, non-forgeability, and non-repudiation, the first-price sealed e-auction protocols need to consider deeper security properties such as decentralization, resistance to collusion attacks, and public verifiability.

The results of the security performance comparison are shown in Table 2.

Table 2. Safety performance comparison

Literature	[26]	[27]	[28]	[29]	Ours
Fairness	√	√	√	√	√
Anonymity	√	√	√	√	√
Decentralization	×	√	×	×	√
Unforgeability	√	√	√	√	√
Non-repudiation	√	√	√	√	√
Resisting collusion attacks	×	×	×	×	√
Public verifiability	√	×	√	√	√

As can be seen from Table 2, the selected similar e-auction schemes can achieve the security guarantees of basic fairness, anonymity, non-repudiation and non-forgery, but still lack in the security performance in resisting collusion attacks, decentralization and full public verifiability.

In this paper, the idea and method of secure multi-party computation are applied to the first-price sealed electronic auction protocol, which realizes the decentralization of the electronic auction structure and eliminates the security risks of the third party. In addition, this paper combines zero-knowledge proof with threshold encryption and decryption to solve the collusion attack problem between malicious participants. Finally, the HMAC technology was used to realize the public verifiability of the auction results. Therefore, compared with the selected literature, the proposed electronic auction protocol has better security performance.

7.2 Computational Cost Analysis and Communication Cost Comparison

In this section, the computational cost of this paper is first analyzed. Then the communication cost of this paper is compared.

When analyzing protocols for secure multi-party computation, the number of the most time-consuming modulo exponential operations is generally used to measure the computational overhead of the protocol. In the protocol proposed in this paper, $2n$ modulo exponential operations are required for n participants to compute the joint public key, followed by $2nl$ modulo exponential operations for each participant to encode the confidential data and encrypt l elements of the encoded array, $4n(l-j)$ modulo exponential operations are required to prove that the participant has provided the correct decryption key, and then $n(l-j)$ modulo exponential operations are required to decrypt the joint decryption in the right-to-left direction, where j is the position of the maximum value in the full set. The final verification that the successful bidder is not cheating requires $2l$ modal exponential operations, so the protocol requires a total of $n\,[2(1+l)+5(l-j)]+2l$ modal exponential operations.

In secure multi-party computation protocols, the number of communication rounds is generally chosen to measure the communication overhead of the protocol. In the protocol proposed in this paper, one round of communication is required for each participant to calculate and obtain the joint public key K. After that, one round of communication is required to encrypt the array X_i and publish the ciphertext C_i. Then one round of communication is required to calculate the bid number, then j rounds of communication are required for all participants to jointly decrypt from right to left, and finally one round of communication is required for all participants to verify the transaction price, so the protocol requires a total of $j+4$ rounds of communication.

Dou et al. [30] and Yang et al. [31] also propose similar optimal secrecy calculation schemes using ElGamal homomorphic encryption and GM homomorphic encryption, respectively, and these two papers are analyzed and compared with the protocol proposed in this paper in terms of communication overhead. The number of communication rounds is $n+y-1$ in [30], n in [31], and $j+4$ in this paper, where n is the number of participants, y is the maximum value, and j is the location of the maximum value in the full set U. In general, y and n are larger while j is smaller, so for an intuitive comparison, it is useful to assume that $y=10$, $j=13$, the comparison of the number of communication rounds for each scheme is shown in Fig. 5.

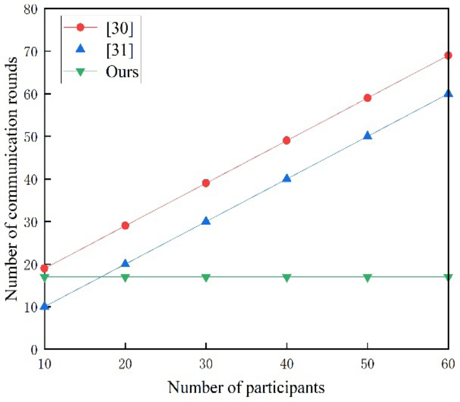

Fig. 5. Comparison of communication rounds

Fig. 5 shows that the number of communication rounds of protocols in [30] and [31] increases linearly with the number of participants, while the number of communication rounds in this paper remains constant. This is because the number of communication rounds of the proposed protocol is only related to the position of the maximum value in the full set *U*, and will not incur more communication overhead with the increase of the number of parties. Compared with the other proposed papers, the communication overhead in this paper does not increase with the increase of participants. This paper has a constant communication overhead, so it has a better communication overhead advantage.

7.3 Efficiency Analysis of This Paper

In this section, the running efficiency of this paper is tested, and the running efficiency of the protocol with different parties and different modulus is shown in Fig. 6. Firstly, the modulus of ElGamal cryptosystem in the protocol are selected as 512 bits, 1024 bits, 1536 bits, 2048 bits and 2560 bits respectively, and then the random number in encryption is selected as 64 bits uniformly, and then their running time under different number of participants is compared and tested, and the running time of the protocol is obtained as shown in Fig. 6. The test environment is shown in Table 3.

In Fig. 6, *m* denotes the number of participants in the protocol, and *m* is selected as 30, 50 and 100 people to analyze the running efficiency of the protocol with different numbers of participants. From the experimental results, it can be seen that the overall efficiency of the protocol is relatively high as the running time of the protocol is about 2.3 seconds when the modulus of the ElGamal cryptosystem is 2560 bits, the random number is 64 bits and the number of participants is 100.

Table 3. Experimental environment parameters

Parameters	Computer configuration
Central processing unit	11th Gen Intel(R) Core (TM) i5-11300H @ 3.10 GHz 3.11 GHz
Random access memory	16.0 GB
Operating system	64-bit Windows 11 Chinese Version
Programming environment	Pycharm+Python3.9

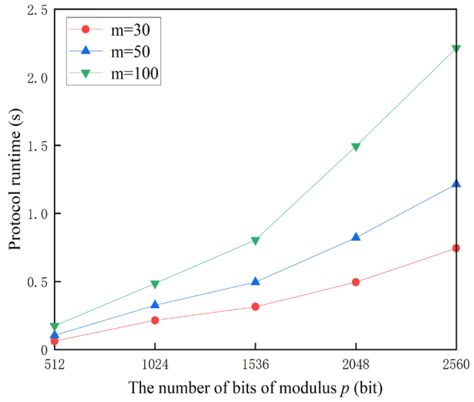

Fig. 6. Overall protocol operation efficiency under different number of participants

7.4 Operation Efficiency Comparison

In this section, we select recent protocols in the field of electronic auctions [32] and [33] and compare them with the protocols in this article. The comparison results of protocol operation efficiency under different number of participants are shown in Fig. 7. In the experiment, the modulus of ElGamal algorithm adopted in this paper is 1024 bits, and the random number is 64 bits.

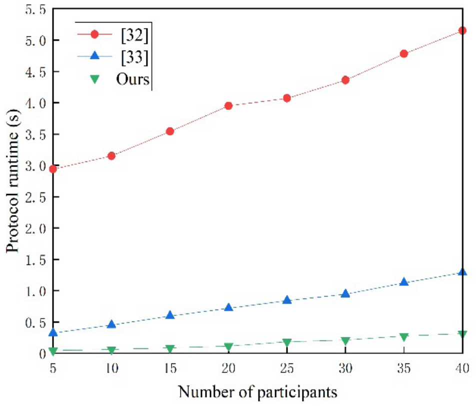

Fig. 7. Comparison of protocol operation efficiency

As can be seen from Fig. 7, compared with the anonymous communication based electronic auction protocol [32] and blockchain-based electronic auction protocol [33], the first price sealed electronic auction protocol proposed in this paper is significantly better than the two. When 40 participants are selected, the running time of our protocol does not exceed 0.5 seconds, while the running time of the other two protocols is about 1.4 seconds and 5.1 seconds respectively, so our protocol has high efficiency.

8 Conclusion

In this paper, we propose a first-price sealed e-auction protocol based on secure multi-party computation malicious model. First, under the guidance of secure multi-party computing idea, the problems of semi-trusted third parties, collusion attacks among participants, unsatisfied decentralized structure and inability of public verification in existing e-auction schemes are solved by applying ElGamal cryptographic algorithm, threshold technology and HMAC technology. Meanwhile, a new method of finding the maximum value by data encoding is also introduced to effectively reduce the communication overhead of the protocol. Not only the security of the protocol is analyzed by the ideal/realistic simulation paradigm theory, but also the efficiency of the protocol operation is tested by simulation experiments.

Finally, the efficiency of the protocol needs to be improved. Because of the use of zero-knowledge proof technology in the process of protocol interaction, it has a certain impact on the efficiency of the protocol. In future work, we will investigate achieving equivalent security strength without using zero-knowledge proofs. If it can be implemented, the operation efficiency of the protocol will be effectively improved.

9 Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 62262060, 62362059); the Industrial Support Plan Project of Gansu Provincial Department of Education (No. 2022CYZC-17, 2023CYZC-09); Key Research and Development Program of Gansu Province (No. 23YFGA0081).

References

[1] T. Wongsamerchue, A. Leelasantitham, An Electronic Double Auction of Prepaid Electricity Trading Using Blockchain Technology, Journal of Mobile Multimedia 18(6)(2022) 1829-1850.

[2] N. Xie, J. Zhang, X. Zhang, W. Li, Double auction mechanisms in edge computing resource allocation for blockchain networks, Cluster Computer (2023) 1-15. https://doi.org/10.1007/s10586-023-04129-0

[3] B.-W. Chen, X. Li, T. Xiang, P. Wang, SBRAC: Blockchain-based sealed-bid auction with bidding price privacy and public verifiability, Journal of Information Security and Applications 65(2022) 103082.

[4] Y.-F. Zheng, L.-S. Zou, W.-J. Zhang, J.-M. Yang, L.-W. Yang, Z.-Q. Lin, Contract-based Cooperative Computation and Communication Resources Sharing in Mobile Edge Computing, Journal of Grid Computing 21(1)(2023) 14.

[5] L.-F. Wei, Q. Wang, L. Zhang, C.-C. Chen, Y.-J Chen, J.-T. Ning, Efficient Private Set Intersection Protocols with Semi-trusted Cloud Server Aided, Journal of Software 34(2)(2023) 932-944.

[6] D. Liu, X.-K. Pei, J.-S. Lai, R.-J. Wang, L.-F. Zhang, Privacy Protection Scheme Combining Edge Intelligent Computing and Federated Learning, Journal of University of Electronic Science and Technology of China 52(1)(2023) 95-101.

[7] J. Furukawa, Y. Lindell, A. Nof, O. Weinstein, High-Throughput Secure Three-Party Computation with an Honest Majority, Journal of Cryptology 36(3)(2023) 21.

[8] O.G. Bautista, M.H. Manshaei, R. Hernandez, K. Akkaya, S. Homsi, S. Uluagac, MPC-ABC: Blockchain-Based Network Communication for Efficiently Secure Multiparty Computation, Journal of Network and Systems Management 31(4)(2023) 68.

[9] A.C. Yao, Protocols for secure computations, in: Proc. 1982 Annual Symposium on Foundations of Computer Science, 1982.

[10] J.-W. Dou, Y.-L. Wang, Secure Sorting Protocols and Their Applications, Journal of Software 33(11)(2022) 4316-4333.

[11] Y. Zhan, Z.-Q. Zhang, Q. Liu, B.-C. Wang, Hiding the input-size in multi-party private set intersection, Designs, Codes and Cryptography 91(9)(2023) 2893-2915.

[12] Y.-B. Yang, X.-L. Dong, Z.-F. Cao, J.-C. Shen, R.-F. Li, Y.-H. Yang, S. Dou, EMPSI: Efficient multiparty private set intersection, Frontiers of Computer Science 18(1)(2024) 181804.

[13] F. Brandt, How to obtain full privacy in auctions, International Journal of Information Security 5(4)(2006) 201-216.

[14] P. Bogetoft, I. Damgård, T.P. Jakobsen, K. Nielsen, J. Pagter, T. Toft, A practical implementation of secure auctions based on multiparty integer computation, in: Proc. 2006 10th International Conference on Financial Cryptography and Data Security, 2006.

[15] C. Wu, C. Chang, I. Lin, New sealed-bid electronic auction with fairness, security and efficiency, Journal of Computer Science and Technology 23(2)(2008) 253-264.

[16] Y. Sun, Y. Sun, M. Luo, L. Gu, S. Zheng, Y. Yang, Comment on Lee et al. ’s group signature and e-auction scheme, Information Systems Frontiers 15(1)(2013) 133-139.

[17] W.-J. Cheng, Y.-Y. Dong, J.-G. Han, A simple and efficient sealed-bid electronic auction scheme, Computer Engineering 40(3)(2014) 171-174.

[18] H.S. Galal, A.M. Youssef, Succinctly verifiable sealed-bid auction smart contract, in: Proc. 2018 Data Privacy Management, Cryptocurrencies and Blockchain Technology, 2018.

[19] J. Xiong, Q. Wang, Anonymous auction protocol based on timed-release encryption atop consortium blockchain, International Journal of Advanced Information Technology 9(1)(2019) 1-16.

[20] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31(4)(1985) 469-472.

[21] C. Trivedi, U.P. Rao, Secrecy aware key management scheme for Internet of Healthcare Things, The Journal of Supercomputing 79(11)(2023) 12492-12522.

[22] S. Devidas, N.R. Rekha, Y.V. Rao, Identity verifiable ring signature scheme for privacy protection in blockchain, International Journal of Information Technology 15(5)(2023) 2559-2568.

[23] O. Goldreich, Foundations of cryptography, Cambridge university press, London, 2004.

[24] O. Goldreich, S. Micali, A. Wigderson, How to play ANY mental game, in: Proc. 1987 19th annual ACM symposium on Theory of computing, 1987.

[25] S.-D. Li, W.-T. Xu, W.-L. Wang, M.-Y. Zhang, Secure maximum (minimum) computation in Malicious Model, Chinese Journal of Computers 44(10)(2021) 2076-2089.

[26] H.S. Galal, A.M Youssef, Verifiable sealed-bid auction on the Ethereum blockchain, in: Proc. 2018 Financial Cryptography, 2018.

[27] Y. Peng, Y. Gao, J.-X. Wu, A privacy preserving sealed-bid auction scheme based on block chains, Cyberspace Security 9(8)(2018) 1-7.

[28] J. Xiong, Research on Anonymous Electronic Auction Protocol Based on Blockchain, [dissertation] Guangzhou: Jinan University, 2019.

[29] R.-C. Yu, Research on the Sealed-Bid Auction Scheme for Blockchain Based on Secure Comparison Protocols, [dissertation] Xianyang: Northwest Agriculture and Forestry University, 2019.

[30] J.-W. Dou, L. Ma, S.-D. Li, Secure multi-party computation for minimum and its applications, ACTA Electonica Sinica 45(7)(2017) 1715-1721.

[31] X.-Y. Yang, S.-D. Li, J. Kang, Private substitution and its applications in private scientific computation, Chinese Journal of Computers 41(5)(2018) 1132-1142.

[32] X.-L. Wang, X.-Y. Li, Anonymous Electronic Auction Protocol Based on Anonymous Communication, Journal of Chinese Computer Systems 41(1)(2020) 85-91.

[33] H.-L. Li, W.-L. Xue, A Blockchain-Based Sealed-Bid e-Auction Scheme with Smart Contract and Zero-Knowledge Proof, Security and Communication Networks 2021(2021) 1-10.