Hindawi
Security and Communication Networks
Volume 2020, Article ID 8872617, 12 pages
https://doi.org/10.1155/2020/8872617

Research Article

SPCTR: Sealed Auction-Based Procurement for Closest Pre-Tender with Range Validation

Li Li,[1,2] Jiayong Liu ,[1] and Peng Jia [1]

[1]*College of Cybersecurity, Sichuan University, Chengdu, Sichuan, China*
[2]*School of Mathematics and Information Engineering, Chongqing University of Education, Chongqing, China*

Correspondence should be addressed to Jiayong Liu; ljy@scu.edu.cn and Peng Jia; pengjia@scu.edu.cn

Received 11 May 2020; Revised 26 June 2020; Accepted 13 July 2020; Published 25 August 2020

Academic Editor: Stelvio Cimato

Over the past decades, there have existed extensive research works on the designs of the closest pre-tender procurement bidding. However, most solutions for the closest pre-tender only target at economic benefits while omitting the problem of bid privacy leakage. Moreover, existing works fail to provide approaches with adequate security and high efficiency. In this paper, for the first time, we propose SPCTR, a sealed-price auction-based procurement bidding system for the closest pre-tender with range validation. SPCTR allows a range validation for a supplier's bid without leaking the secret bid. Besides, SPCTR achieves a sealed-price comparison with the pre-tender to find the closest pre-tender bid. Compared with previous works, SPCTR provides strong privacy protection for the bids of suppliers without sacrificing high efficiency. SPCTR is constructed based on carefully designed cryptographic tools with generality and simplicity which enable various operations on the encrypted values, and these tools can be easily applied to other contexts. We not only formally prove that SPCTR is secure against semihonest adversaries but also comprehensively analyze the efficiency. Experimental results validate that SPCTR achieves procurement bidding with light computation time and communication cost in practice.

1. Introduction

1.1. Motivation. Nowadays, Internet communication has made a huge impact on supply chain management and facilitates the ample participation of motivated suppliers. As one aspect of supply chain management, procurement is a necessary process for a company to obtain the manufacturing materials. For leveraging the competitive nature of suppliers to keep the procurement cost reasonable, the auction mechanism is introduced in the procurement. Pre-tender bidding, a popular type of auction-based procurement bidding, is widely used as a standard business requirement for many organizations in recent years [1, 2]. In what follows, the process of pre-tender bidding is briefly described. First, as per the specific project, the procurement manager will generate some critical factors including the lowest bid, the highest bid, and the pre-tender. Then, the procurement manager will issue the lowest bid and the highest bid while keeping the pre-tender private to himself. Later on, suppliers submit their bids required to be validated in the range between the lowest bid and the highest bid. Finally, the supplier with the closest pre-tender bid will be claimed as the winner of the bidding system.

Unfortunately, for the concerns of privacy, the procurement bidding system is vulnerable. For example, the third-party procurement manager is not always trustworthy and suppliers are competitive to obtain the bidding. Thus, they probably interfere with a normal bidding system by soliciting some commercial secret values of suppliers like bids. Then, the procurement manager can adjust the pre-tender for maximum profit rather than the true valuation of a project next time by monitoring the previous bids of suppliers. Moreover, by learning the historical bids of other suppliers, a supplier can submit a bid that may be closest to the pre-tender with high probability. Hence, providing strong privacy protection for suppliers' bids is of great importance in the realistic procurement bidding system.

1.2. Challenges and Solutions. To design such a privacy-preserving closest pre-tender procurement bidding system, we have to face the following two challenges. The first challenge is how to design a privacy-preserving closest pre-tender procurement bidding system which provides privacy protection for the secret bids while the fundamental functions of the system are maintained. Huang et al. presented a secure auction mechanism for secondary spectrum markets based on BGN cryptosystem [3]. However, the involved bilinear pairing operations make the scheme computationally expensive. Blass and Kerschbaum proposed a secure auction for blockchain by leveraging GM encryption [4]. Later on, a solution of fully private auction seeking for the highest bid was presented in [5]. These works provide good protection for bid privacy but also face the challenge of computation efficiency. More recently, Wang et al. presented privacy-preserving truthful double online auction for heterogeneous spectrum [6]. In this work, garbled circuits are used for bid grouping rather than bid comparison [7]. To this end, we employ lightweight cryptographic primitives, such as the Paillier cryptosystem and garbled circuits to the original system. On the one hand, these cryptographic tools allow our scheme to output the correct result. For example, in the original auction, the supplier with the bid b_0 is closest to the pre-tender so that the winner is b_0. In our privacy-preserving system, despite the fact that bids are encrypted, the winner is still b_0 and will not change. On the other hand, they can ensure that the private bid of a supplier is well protected and will not be leaked to other suppliers and the procurement manager.

The second challenge is how to demonstrate whether a secret bid is in a specified range or not, without revealing the secret bid. To cope with the second challenge, the technology of range proof is adopted in this paper. Roughly speaking, the range proof technology is categorized into two types: decomposition-commitment range proof [8] and signature-based range proof [9]. However, the decomposition-commitment range proof is typically computationally expensive due to bit decomposition and commitment generation. In contrast, the signature-based proof only requires a constant number of group elements to be exchanged irrespective of the number of bits of the secret value. Chaabouni et al. presented a more efficient variant of signature-based range proof in [10]. In this work, first, the verifier sends the prover all the signatures of elements in the range, and then the prover proves that its secret value matches one of these signatures. However, the communication complexity depends on the size of the range. Improvement was made in [9], and only a constant number of elements are exchanged in the proof. Hence, for the consideration of a large number of bids, signature-based range proof is taken as our range proof technology.

All in all, our main contributions are listed as follows:

(i) To our knowledge, we firstly propose a privacy-preserving closest pre-tender procurement bidding system. That means a fully secure procurement bidding system is carried out without revealing suppliers' private bids. Based on the delicately designed cryptographic tools, we achieve the phase of bid comparison in a privacy-preserving manner to ensure the privacy of bids.

(ii) We propose a signature-based range proof to achieve the range validation. The commit-challenge-verify process allows the bids submitted by the suppliers to be validated in a specified range (lying between the lowest bid and the highest bid). Also, the validation process will not leak the secret bids of suppliers.

(iii) We present a thorough theoretical analysis of SPCTR for the concerns of both security and efficiency. At last, we construct a prototype to conduct extensive experiments to further validate the practicality in the computation time and communication cost.

The remainder of this paper is organized as follows. The system model, the threat model, and design goals are presented in Section 2. Section 3 introduces the preliminaries and the necessary mathematical notations. Section 4 presents building blocks of SPCTR, including ciphertext multiplication and secure minimum value selection. In Section 5, we elaborately depict the two phases of SPCTR: BidRPrf and CloPCmp. In Section 6, we present the security and efficiency analysis for SPCTR, followed by the performance evaluation in Section 7. At last, Section 9 concludes this paper.

2. Problem Statement

2.1. System Model. We consider the construction bidding project as our application scenario. The system model of SPCTR under this application scenario is depicted in Figure 1. The model is in terms of three entities, a procurement manager, a procurement agent, and m suppliers. The functionalities of three entities are described, respectively, in the following:

Procurement manager (PM): PM sets the lowest bid ($\widehat{b}$), the highest bid ($\widetilde{b}$), and the pre-tender (t) for SPCTR. Therein, the lowest bid $\widehat{b}$ and the highest bid $\widetilde{b}$ are public to each entity. SPCTR requires each bid submitted by each supplier to lie in the range of $[\widehat{b}, t\widetilde{b}]$, whereas the pre-tender (t) is a secret value that only PM knows it. Furthermore, the bid which is nearest to the pre-tender t will be the winner of SPCTR. After a secure interaction between PM and the procurement agent, PM declaims the final winner of the bidding and returns this result to suppliers.

Suppliers: in the bidding project, m suppliers wish to compete for undertaking the project. First, these suppliers formulate the budgets as their bids. Afterward, the suppliers should prove to PM that their bids lie in the range of $[\widehat{b}, t\widetilde{b}]$ without disclosing their bids to PM. After accomplishing the range validation, suppliers encrypt their bids by a cryptosystem and send the encrypted bids to PM.

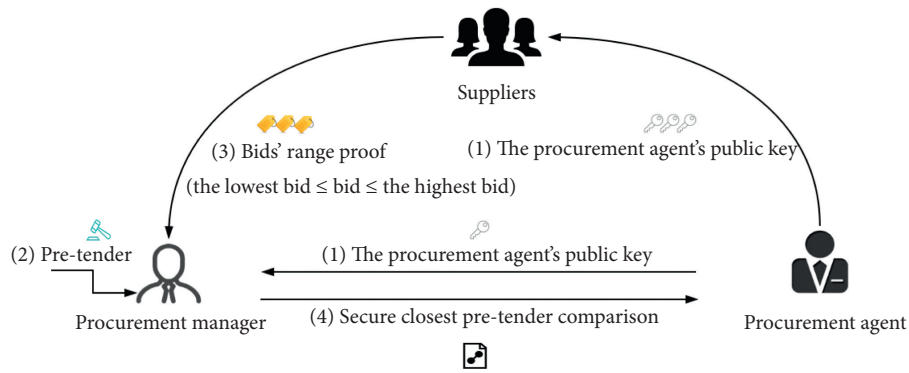

Figure 1: System model for SPCTR.

Procurement agent (PA): in [3, 6], PA is presented to assist PM to promote the secure bidding. To be specific, PA will cooperate with PM to find the bid which is closest to the pre-tender. PA generates a key pair including a private key and a public key. PA holds the private key by itself and publishes the public key to other parties.

Assumptions: suppliers are assumed to submit their true bids faithfully. This means any supplier cannot falsify a bid which is in the range $[\hat{b}, t\tilde{b}]$ to replace the original bid which is not in the range $[\hat{b}, t\tilde{b}]$. Also, suppliers are assumed to submit the same bids in the different phases of SPCTR. Finally, we assume PM and PA do not collude with each other to manipulate the bidding process and determine the final winner.

2.2. Attack Model. In our attack model, we consider PM, PA, and suppliers to be semihonest. This implies that they follow the stipulated rules strictly but they are curious about other entities' sensitive information (e.g., their bids) and will attempt to obtain the information from the view of executing the bidding process. We consider a passive adversary $\mathcal{A}$ under semihonest setting.

Similar to the general security definition in [11, 12], we use the simulation-based proof technique to define security [13]. Let P be a set of involved parties who execute an algorithm Π to compute a function F. In_P is denoted as the input and $O_{\mathcal{J}}$ is denoted as the output. Let $\mathcal{J}$ be a corrupted subset and Φ be auxiliary information during executing Π, e.g., the random numbers selected by P_i, and the view of $\mathcal{J}$ be $V_{\mathcal{J}}^{\Pi}(In_{\mathcal{J}}, In_{\mathcal{J}}, \Phi_{\mathcal{J}})$. $S_{\mathcal{J}}$ is a simulator to generate a transcript of the scheme that takes $In_{\mathcal{J}}$ as the input and $O_{\mathcal{J}}$ as the output.

Definition 1. An algorithm Π is defined to be secure against an adversary $\mathcal{A}$ if there exists a probabilistic polynomial time (PPT) simulator $S_{\mathcal{J}}$ such that two distribution ensembles $S_{\mathcal{J}}$ and $V_{\mathcal{J}}^{\Pi}$ are computationally indistinguishable, that is, $S_{\mathcal{J}}(In_{\mathcal{J}}, F(In_{\mathcal{J}}), O_{\mathcal{J}}, \Phi_{\mathcal{J}}) \approx_c V_{\mathcal{J}}^{\Pi}(In_{\mathcal{J}}, F(In_{\mathcal{J}}), O_{\mathcal{J}}, \Phi_{\mathcal{J}})$.

In combination with the concretization of our scheme, we present a formal security definition against semihonest adversaries based on Definition 1.

Definition 2. (security). An algorithm Π is assumed to have Alice (resp. Bob) and compute $F^A(x, y)$ (resp. $F^B(x, y)$), where (x, y) are inputs of Alice and Bob, respectively. Let $V_{\Pi}^{A}(x, y)$ (resp. $V_{\Pi}^{B}(x, y)$) denote Alice's (resp. Bob's) view during the execution of Π on the input of (x, y). This means $(x, \Phi_{\mathcal{A}})$ (resp. $(y, \Phi_{\mathcal{B}})$) are Alice's (resp. Bob's) input and auxiliary information, respectively, during executing Π, and O_A^{Π} (resp. O_B^{Π}) is the output of Alice (resp. Bob). Then, the algorithm Π is secure against semihonest adversaries if there are probabilistic polynomial time (PPT) simulators S_1 and S_2 that make equation (1) hold.

$$\begin{aligned}\left(S_1\left(x, F^A(x, y)\right), F^B(x, y)\right) &\approx_c \left(V_A^{\Pi}(x, y), O_B^{\Pi}(x, y)\right),\\ \left(S_2\left(y, F^B(x, y)\right), F^A(x, y)\right) &\approx_c \left(V_B^{\Pi}(x, y), O_A^{\Pi}(x, y)\right).\end{aligned} \tag{1}$$

2.3. Design Goals. In what follows, the design goals of our scheme are depicted.

Correctness: in short, after the procurement bidding, the winning supplier returned by PM in our scheme should be the same as the one obtained in the original plaintext domain.

Security: PM, PA, and other suppliers cannot obtain the secret bids of suppliers except for the bidding result.

Lightweight: the cryptographic tools used in our scheme should be lightweight. That is, the computation time and communication cost should be acceptable in practice without sacrificing bid privacy.

3. Preliminaries

In this section, we introduce the necessary preliminaries of our scheme, including Paillier cryptosystem, zero-knowledge proofs of knowledge (ZKPK), and improved garbled circuit. Paillier cryptosystem is used to generate the key pair, encrypt, and decrypt messages. ZKPK is used to achieve range validation, and the improved garbled circuit is used to enable secure closest pre-tender comparison.

3.1. Paillier Cryptosystem. To provide the sealed-bid property, the Paillier homomorphic encryption algorithm is

adopted in our scheme [14]. A Paillier key pair consists of the private key sk and the public key pk. Having said this, the Paillier cryptosystem mainly consists of three parts including key generation, encryption, and decryption. The three parts are listed as follows:

Key generation: first, two large prime integers p and q are selected. Let $n = p \cdot q$, $g \xleftarrow{\$} \mathbb{Z}^*_{n^2}$; then, the public key is $pk = (n, g)$. Let $e = (p-1) \cdot (q-1)$ and $d = (e \bmod n^2)^{-1} \bmod n$; then, the private key is $sk = (e, d)$.

Encryption: for a plaintext message $b \in \mathbb{Z}_n$, we denote $b \in \mathbb{Z}_{n^2}$ as its encrypted value. We select $r \xleftarrow{\$} \mathbb{Z}^*_n$; then, the encrypted value b is computed by the following equation:

$$b = g^b \cdot r^n \bmod n^2. \tag{2}$$

Decryption: given the private key sk and b, decryption is computed by equation (3) to obtain the plaintext b.

$$b = \frac{(b^e \bmod n^2 - 1)}{n} \cdot d \bmod n. \tag{3}$$

The Paillier cryptosystem possesses the following interesting properties including homomorphic addition and indistinguishability.

Homomorphic addition: this operation allows a specific computation to be executed on ciphertexts and finally obtains a new ciphertext that can be decrypted to match the result of the computation executed directly on the plaintext. As per the homomorphic addition property of the Paillier cryptosystem, equations (4)–(6) hold, where a and b are plaintexts and u is an integer. In the following context, for easy exposition, we leave out the mod operation without confusion.

$$[a + b] = [a] \cdot [b], \tag{4}$$

$$[a] \ominus [b] = [a - b] = [a] \cdot [b]^{-1}, \tag{5}$$

$$[u \cdot a] = [a]^u. \tag{6}$$

Indistinguishability: for a plaintext a, if a is encrypted twice to obtain a_1 and a_2, respectively, then the probability that an adversary distinguishes a_1 and a_2 will only be negligible better than a random guess.

3.2. Zero-Knowledge Proofs of Knowledge. Zero knowledge (ZK) is termed as an interactive protocol in which a prover (Alice) tries to convince a verifier (Bob) about the validity of a statement without disclosing anything else beyond the fact itself [7]. In the following, the formal description of such a proof is given. For example, there exists a proof that $\pi = PK\{(w, u, v)\colon C = m^r h^u \wedge I = m^v\}$, where $(m, h) \xleftarrow{\$} \mathbb{Z}^*_{n_2}$. This expression explicitly denotes that the prover tries to convince the verifier with the statement of knowing (w, u, v) by $C = m^r h^u$ and $I = m^v$. The variables (w, u, v) remain private to the verifier, whereas the variables (C, I) remain public to the verifier.

3.3. Oblivious Transfer. Oblivious transfer (OT) is one type of two-party computing protocols where a sender (Alice) has an input, and then a receiver (Bob) learns something about the input but Alice does not know what Bob has learned [15]. In the 1-out-of-N OT protocol, the sender has N strings $S_1, S_2, \ldots, S_N$ and the receiver can select one of N strings S_i without learning anything about the other $N-1$ strings. Also, the sender learns nothing about which input has been chosen by the receiver [16].

3.4. Garbled Circuit. In [17], Yao's garbled circuits were firstly proposed for secure two-party computation, and the circuits' practice and security are demonstrated. The basic process for Yao's garbled circuits is briefly described as follows. The circuit constructor (Alice) possesses the value s_1 and the evaluator (Bob) possesses the value s_2, respectively, and they can jointly compute a specified function $f(s_1, s_2)$ without disclosing any secret information beyond the result. Firstly, Alice converts a circuit that computes f into an encrypted form by an algorithm GreateGC. Then, Alice sends the generated circuit and the garbled value to Bob. Bob explicitly computes the output of the circuit without disclosing any other information by an algorithm EvalGC. In the garbled circuit, the oblivious transfer protocol will be executed to transmit the values from Alice to Bob.

In the following, the circuits including subtraction circuit, comparison circuit, multiplexer circuit, and minimum value circuit used in our design will be introduced.

Subtraction circuit: a subtraction circuit is used to subtract two l-bit integers a and b efficiently. The circuit consisting of a chain of 1 bit subtractors (−) is shown in Figure 2. Each 1 bit subtractor has carry-in bits from the output of the last 1 bit subtractor c_i and the bits a_i, b_i. Furthermore, the 1 bit subtractor is composed of a 2-input AND gate and four XOR gates.

Comparison circuit: an integer comparison circuit is constructed for comparing two l-bit integers a and b to get the comparison result z. We use equation (7) to express the integer comparison process.

$$z = [a > b] \coloneqq \begin{cases} 1, & \text{if } a > b, \\ 0, & \text{else.} \end{cases} \tag{7}$$

A comparison circuit can be decomposed to l number of 1 bit comparators (>) in sequence. More specifically, the 1 bit comparator can be constructed by a 2-input AND gate and three XOR gates.

Multiplexer circuit (MUX): a l-bit multiplexer circuit is constructed to choose one of the l-bit integers a and b as output as per the selection bit z. If $z = 0$, then the

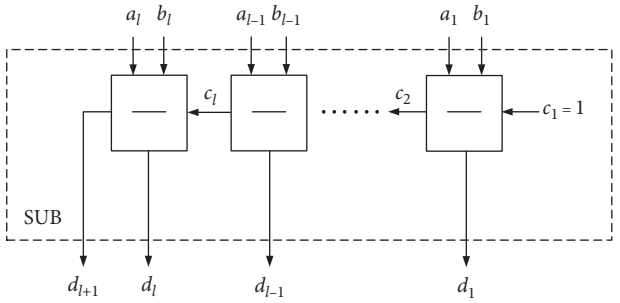

Figure 2: Subtraction circuit.

integer a will be selected. Otherwise, the integer b will be chosen. Usually, this circuit is a composed part to construct a minimum value circuit. The selection bit z can derive from the boolean output of other garbled circuits, e.g., the comparison circuit.

Minimum circuit: a minimum circuit can be used to select the minimum value $\widehat{a}$ and its index i from a list of l-bit values $a_0, a_1, \ldots, a_{m-1}$. For the selected minimum value $\widehat{a}$, it makes equation (8) hold. For example, the minimum value $\widehat{a}$ and its index i of the list $(4, 1, 1, 3)$ are $\widehat{a} = 1$ and $i = 1$, respectively, since the leftmost minimum value 1 lies at the position of 1. Without loss of generality, we assume that the number of elements in the list m is a power of two, and the maximum index can be represented with the value of $\log_2 m$.

$$\forall j \in \{0, 1, \ldots, m-1\}: \left(\widehat{a} < a_j\right) \vee \left(\widehat{a} = a_j \wedge i \leq j\right). \quad (8)$$

The MIN circuit is constructed by a series of minimum blocks (min). The minimum value and its index will be determined by a tournament-like way of using a tree of minimum blocks. It is straightforward to obtain that the depth of the tree is $\log_2 m$. In Figure 3, each minimum block consists of one comparison circuit and two multiplexer circuits. For each minimum block at the depth d, the left part input of the block is $a_{d,L}$ and its index is $i_{d,L}$. Also, the right part input of the block is $a_{d,R}$ and its index is $i_{d,R}$. Through the minimum block, the output of a_{d+1} and i_{d+1} is computed.

We illustrate the function of the minimum block specifically. First, the comparison of $a_{d,L}$ and $a_{d,R}$ is achieved by a comparison circuit. There exists two cases. On the one hand, if $a_{d,L}$ is not bigger than $a_{d,L}$, then the output of the comparison circuit is 0. The minimum value $a_{d,L}$ and its index $i_{d,L}$ are chosen as the output of a_{d+1} and i_{d+1} according to their corresponding multiplexer circuits. On the other hand, if $a_{d,L}$ is bigger than $a_{d,L}$, then the output of the comparison circuit is 1. Hence, the minimum value $a_{d,R}$ and its index $i_{d,R}$ are selected as a_{d+1} and i_{d+1}.

4. Building Blocks

Before presenting our design, we first introduce the composed building blocks for our scheme. These building blocks are constructed based on the secure interaction between two parties, Alice and Bob, in which we assume only Bob owns the private key for decryption in the Paillier cryptosystem.

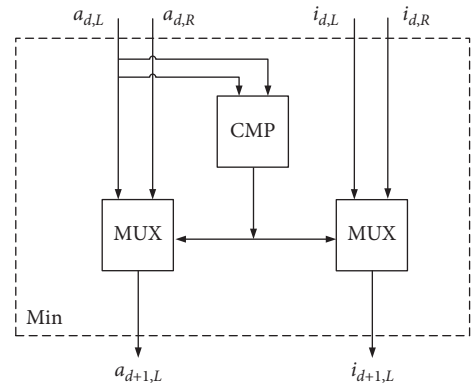

Figure 3: Minimum block (min).

4.1. Ciphertext Multiplication. Given two l-bit values a and b, their encrypted forms are $[\![a]\!]$ and $[\![b]\!]$, respectively. By using the homomorphic property of Paillier encryption, the encrypted form of $[\![a \cdot b]\!]$ can be obtained. We use the technology of random mask to provide a statistical security [18]. First Alice blinds a and b with two k-bit values r_a and r_b, respectively, and sends $[\![a + r_a]\!]$ and $[\![b + r_b]\!]$ to Bob. Bob decrypts $[\![a + r_a]\!]$ and $[\![b + r_b]\!]$ to obtain $a + r_a$ and $b + r_b$. Then, Bob computes $(a + r_a) \cdot (b + r_b)$ and re-encrypts $(a + r_a) \cdot (b + r_b)$. Afterward, Bob sends $(a + r_a) \cdot (b + r_b)$ to Alice. Finally, Alice computes $[\![a \cdot b]\!]$ according to the following equation:

$$[\![a \cdot b]\!] = [\![(a + r_a) \cdot (b + r_b)]\!] \cdot [\![a]\!]^{-r_b} \cdot [\![b]\!]^{-r_a} \cdot [\![r_a r_b]\!]^{-1}. \quad (9)$$

4.2. Secure Minimum Value Selection. As shown in Figure 4, to securely select the minimum value and its index from a list of encrypted values $[\![a_0]\!], [\![a_1]\!], \ldots, [\![a_{m-1}]\!]$, Alice first generates a minimum circuit consisting of m subtraction circuits and a minimum circuit using GreateGC and gets the garbled values of m random numbers $r_0, r_1, \ldots, r_{m-1}$. Then, Alice sends the garbled values $\overline{r_0}, \overline{r_1}, \ldots, \overline{r_{m-1}}$ and the blinded ciphertexts $[\![a_0 + r_0]\!], [\![a_1 + r_1]\!], \ldots, [\![a_{m-1} + r_{m-1}]\!]$ to Bob. Afterward, Bob invokes the oblivious transfer protocol and executes decryption. Then, Bob gets the garbled values $\overline{a_0 + r_0}, \overline{a_1 + r_1}, \ldots, \overline{a_{m-1} + r_{m-1}}$. Finally, EvalGC is leveraged to evaluate the garbled circuit created by GreateGC.

A SUB circuit is used to get the difference between two garbled values, e.g., $\overline{a_0 + r_0}$ and $\overline{r_0}$. Then, the l-bit value of the results will be taken as the inputs of the minimum circuit. And the index $i_{\min}$ of the minimum value is obtained as the output. Finally, the result $i_{\min}$ is sent to Alice. The process of secure minimum value selection is described in Algorithm 1.

In the following context, we specify $i_{\min} = \text{MinValSel}([\![a_0]\!], [\![a_1]\!], \ldots, [\![a_{m-1}]\!])$ for Alice to get the index of the minimum value from a list of encrypted values directly.

5. Our Scheme

Our procurement bidding system consists of two phases: bids' range proof validation (BidRPrf) and secure closest

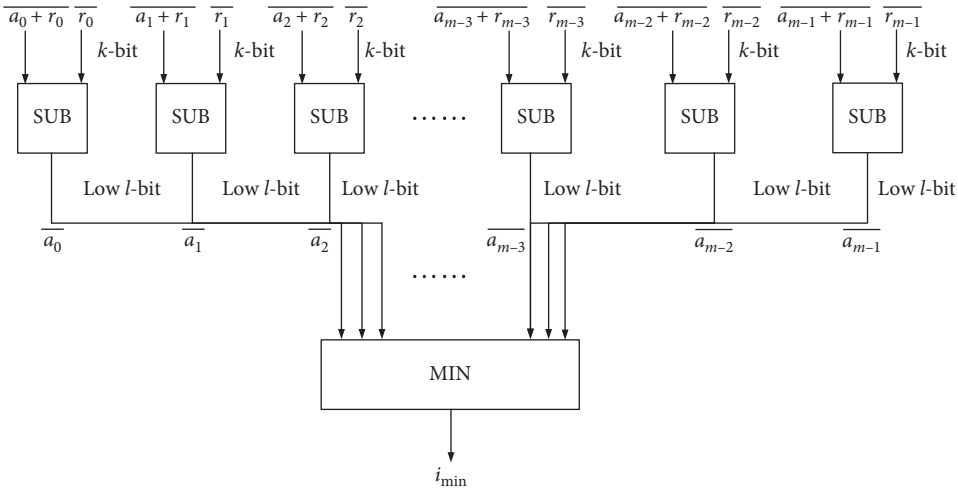

Figure 4: The high-level structure of secure minimum value selection.

Input:
Alice: A list of encrypted bids $[a_0], \ldots, [a_{m-1}]$;
Bob: The private key sk and the public key pk;
Output:
Alice: $\perp$;
Bob: The index $i_{\min}$ of the minimum value;
(1) At Alice:
(2) Choose m number of k-bit random numbers $r_0, \ldots, r_{m-1}$
(3) Compute $[a_0 + r_0], [a_1 + r_1], \ldots, [a_{m-1} + r_{m-1}]$;
(4) Create the garbled circuit, garbled values $\overline{r_0}, \overline{r_1}, \ldots, \overline{r_{m-1}}$
(5) Send $[a_0 + r_0];, [a_1 + r_1], \ldots, [a_{n-1} + r_{m-1}]$, the garbled circuit, $\overline{r_0}, \overline{r_1}, \ldots, \overline{r_{m-1}}$ to Bob;
(6) At Bob:
(7) Decrypt $[a_0 + r_0], [a_1 + r_1], \ldots, [a_{m-1} + r_{m-1}]$ to get $\overline{a_0 + r_0}, \overline{a_1 + r_1}, \ldots, \overline{a_{m-1} + r_{m-1}}$;
(8) Invoke oblivious transfer to get garbled values $\overline{a_0 + r_0}, \overline{a_1 + r_1}, \ldots, \overline{a_{m-1} + r_{m-1}}$;
(9) Evaluate the circuit to get the index $i_{\min}$ of the minimum value;
(10) Send $i_{\min}$ to Alice;
(11) At Alice:
(12) **return** $i_{\min}$;

Algorithm 1: MinValSel.

pre-tender comparison (CloPCmp). BidRPrf is used for PM to validate whether the suppliers' bids are in the specified range or not. Moreover, CloPCmp is used for PM to choose the index of the supplier whose bid is closest to the pre-tender securely.

We use the signature-based proof technology to achieve the range proof. For each bid, it is required to be validated in the range between the lowest bid $\widehat{b}$ and the highest bid $\widetilde{b}$. First, PM generates a signature for each element in $[\widehat{b}, t\widetilde{b}]$. Then, PM picks $x \xleftarrow{\$} \mathbb{Z}^*_{n_2}$ and computes $\lambda_b = g^{1/(x+b)}$. Note that g is the public key of the Paillier cryptosystem. These values $\lambda_{\widehat{b}}, \lambda_{\widehat{b}+1}, \ldots, \lambda_{\widetilde{b}}$ are precomputed for the suppliers publicly to download and use in the range proof below. We call the precomputing process as PreCmp.

5.1. Bids' Range Proof. To prove a bid $b \in [\widehat{b}, t\widetilde{b}]$, Algorithm 2 is designed on the basis of ZKPK. First, a commitment C of the bid b and the values U, V, W are generated by the supplier. These values are related to λ_b. Then, C, U, V, W are sent to PM. After receiving these values, PM sends a challenge $e \xleftarrow{\$} \mathbb{Z}^*_{n_2}$ to the supplier. Afterward, the supplier generates the proof composed of $\varphi_r, \varphi_b, \varphi_v$ and sends the proof to PM. Finally, PM verifies the proof and decides whether to accept or reject the proof. Note that the supplier does not need to compute the predetermined value λ_b. The value λ_b has been precomputed for public downloading.

5.2. Secure Closest Pre-Tender Comparison. The high-level structure of closest pre-tender comparison is described in Figure 5. In order to find the closet pre-tender supplier securely, first, we should measure the distance between the bid b_i and the pre-tender t for each supplier, and then we have to find the minimum absolute value of these distances. This means we devote to find the minimum value of

Input:
Supplier: the bid b, the values $\lambda_{\widehat{b}}, \lambda_{\widehat{b+1}}, \ldots, \lambda_{\widetilde{b}}$;
Output:
PM: True or False
Supplier: $\perp$;
(1) At supplier:
(2) Choose $v, h, h_1, r, \beta, \rho, \nu \xleftarrow{\$}$ and compute a commitment $C = h^r h_1^b$;
(3) Compute $V = \lambda_b^v$, $U = h^\rho h_1^\beta$, $W = V^{-\beta} g^\nu$
(4) Send the commitment C and V, U, W to PM
(5) At PM:
(6) Send a challenge e to the supplier;
(7) At supplier:
(8) Set $\varphi_r = \rho - er$, $\varphi_b = \beta - eb$, $\varphi_v = \nu - ev$;
(9) Send $\varphi_r, \varphi_b, \varphi_v$ to PM
(10) At PM:
(11) if $U \stackrel{?}{=} C^e h^{\varphi_r} h_1^{\varphi_b} \wedge W \stackrel{?}{=} V^{ex} V^{-\varphi_b} g^{\varphi_v}$ then
(12) Accept the proof
(13) **return** True
(14) else
(15) Reject the proof;
(16) **return** False;

Algorithm 2: BidRPrf.

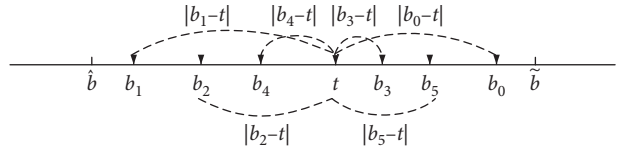

Figure 5: Closest pre-tender comparison.

$|b_0 - t|, |b_1 - t|, \ldots, |b_{m-1} - t|$. For the sake of privacy preservation, the minimum value selection of $|b_0 - t|, |b_1 - t|, \ldots, |b_{m-1} - t|$ is required to be achieved on the ciphertexts. However, there does not exist an efficient method to compute the minimum value from $|b_0 - t|, |b_1 - t|, \ldots, |b_{m-1} - t|$. Therefore, we transform the problem to compute the minimum value from the ciphertexts of $(b_0 - t)^2, ((b_1 - t)^2), \ldots, ((b_{m-1} - t)^2$. Using the homomorphic property of the Paillier cryptosystem, Algorithm 3 is designed to achieve closest pre-tender comparison securely.

The details of CloPCmp are depicted in Algorithm 3. PM has a list of encrypted bids $[\![b_0]\!], \ldots, [\![b_{m-1}]\!]$ and the encrypted pre-tender $[\![t]\!]$ while PA keeps the key pair (pk, sk). PM selects $m-1$ masking random integers $r_0, r_1, \ldots, r_{m-1}$ to get $B_0, B_1, \ldots, B_{m-1}$ and sends $B_0, B_1, \ldots, B_{m-1}$ to PA for decryption. PA decrypts them to get B_i^* and computes $B_0^* B_0^*, B_1^* B_1^*, \ldots, B_{m-1}^* B_{m-1}^*$. Then, PA re-encrypts each $B_i^* B_i^*$ and sends each $B_i^* B_i^*$ to PM. Subsequently, with no need to decrypt them, based on the equation of $(b_i - t)^2 = (b_i - t + r_i)^2 - 2(b_i - t) r_i - r_i^2$, PM can remove the masking numbers to get D_i through the homomorphic property of the Paillier cryptosystem by $D_i = [\![B_i^* B_i^*]\!] \ominus ([\![b_i]\!] \ominus [\![t]\!])^{2r_i} \ominus r_i^2$. Finally, the index of the minimum value can be selected through invoking MinValSel which is presented in Algorithm 1.

6. Theoretical Analysis

6.1. Security Analysis. In this section, the security proof is formally proved. In what follows, Lemma 1 is introduced to prove that SPCTR is secure against semihonest adversaries based on the security definition of Definition 2 that is defined in Section 2.2.

Lemma 1. *Assume Bob generates the key pair (pk, sk) for the homomorphic cryptographic system and issues the public key pk for Alice. Then, Alice and Bob run the algorithm Π. All the ciphertexts transmitted from Alice to Bob are uniformly distributed and independent of Alice's inputs. And all the messages transmitted from Bob to Alice are encrypted by the cryptographic system. Therefore, the algorithm Π is secure against semihonest adversaries.*

Proof. To prove Lemma 1, we should consider two cases, in which the party that is corrupted by the adversary is different. In the first case, Alice is corrupted, and in the second case, Bob is corrupted. In each case, we can finally infer that equation (1) holds. Therefore, we conclude that the algorithm Π is secure against semihonest adversaries. In [6], we can see more details about this proof.

Input:
At PM: the encrypted bids $[\![b_0]\!], \ldots, [\![b_{m-1}]\!]$, the encrypted pre-tender $[\![t]\!]$;
At PA: The key pair (sk, pk);
Output:
At PM: The index $i_{\min}$ of the supplier who owns the closest pre-tender bid
At PA: $\perp$;
(1) At PM:
(2) Choose m number of k-bit random values $r_0, \ldots, r_{m-1}$;
(3) **for** $i = 0$ to $m - 1$**do**
(4) Compute $B_i = [\![b_i]\!] \ominus [\![t]\!] \cdot r_i$
(5) Send $B_0, B_1, \ldots, B_{m-1}$ to PA;
(6) At PA:
(7) **for** $i = 0$ to $m - 1$**do**
(8) Decrypt B_i to get B_i^*;
(9) Compute $B_i^* B_i^* = (b_i - t + r_i)^2$;
(10) Re-encrypt $B_i^* B_i^*$;
(11) Send $B_0^* B_0^*, B_1^* B_1^*, \ldots, B_{m-1}^* B_{m-1}^*$ to PM;
(12) At PM:
(13) **for** $i = 0$ to $m - 1$**do**
(14) Compute $D_i = [\![B_i^* B_i^*]\!] \ominus (b_i \ominus t)^{2r_i} \ominus r_i^2$
(15) $i_{\min} = \text{MinValSel}(D_0, D_i, \ldots, D_{m-1})$
(16) **return** $i_{\min}$

Algorithm 3: CloPCmp.

Theorem 1. *BidRPrf (Algorithm 2) is secure against semihonest adversaries.*

Proof. It is straightforward to demonstrate the security of BidRPrf against semihonest adversaries since it is constructed based on ZKPK. In the context of ZKPK, the corrupted party (the verifier) has no private input nor output. Thus, the only task for the simulator is to generate a view that is indistinguishable from the real execution. Having said this, it is easy for us to pick some random values and compute new values $V^*, U^*, W^*, e^*, \varphi_r^*, \varphi_b^*, \varphi_v^*$ which are indistinguishable from $V, U, W, e, \varphi_r, \varphi_b, \varphi_v$. In addition, due to the property of the semihonest model which follows the protocol rules exactly, the values $V^*, U^*, W^*, e^*, \varphi_r^*, \varphi_b^*, \varphi_v^*$ can be validated successfully. Therefore, BidRPrf is secure against semihonest adversaries.

Theorem 2. *MinValSel (Algorithm 1) is secure against semihonest adversaries.*

Proof. Messages which are transmitted between Alice and Bob are encrypted by the semantically secure Paillier cryptosystem and are uniformly distributed in the ciphertext space $\mathbb{Z}_{n^2}$. The result of MinValSel is revealed to determine the index of the minimum value on the ciphertexts. Note that the random masked technique we leverage will not thwart the security guarantees. In addition, the garbled circuits (e.g., the minimum circuit and the comparison circuit) which are adopted in this paper have been proved to be secure against semihonest adversaries in [19]. Thus, based on the foundation of Lemma 1 and sequential composition theory [20], MinValSel is secure against semihonest adversaries. □

Theorem 3. *CloPCmp (Algorithm 3) is secure against semihonest adversaries.*

Proof. In Algorithm 3, messages are exchanged between PM and PA. By masking with some random values, messages that are sent from PM to PA include $B_0, B_1, \ldots, B_{n-1}$, which are uniformly distributed in the ciphertext space $\mathbb{Z}_{n^2}$. Messages that are sent from PA to PM include $B_0^* B_0^*, B_1^* B_1^*, \ldots, B_{n-1}^* B_{n-1}^*$, which are encrypted by the semantically secure Paillier cryptosystem. Moreover, MinValSel has been proved to be secure against semihonest adversaries. Therefore, on the basis of Lemma 1 and sequential composition theory [20], CloPCmp is secure against semihonest adversaries.

6.2. Efficiency Analysis. We individually measure each phase of SPCTR to derive the computation and communication complexities and then we measure the overall computation and communication complexities.

Bids' range proof: the computation and communication complexities of this phase are mainly derived from Algorithm 2. This phase is executed by the suppliers and PM to validate whether the bids are in the required range. In terms of the computation overhead from each supplier, besides the regular modular operations, it requires 8 exponentiation operations, 3 multiplication operations, and 3 subtraction operations. Additionally, for PM, it requires 6 exponentiation operations. For the concerns of communication overhead, each supplier and PM need to exchange 7 values. Hence, the computation and communication complexities in this phase are both $O(m)$.

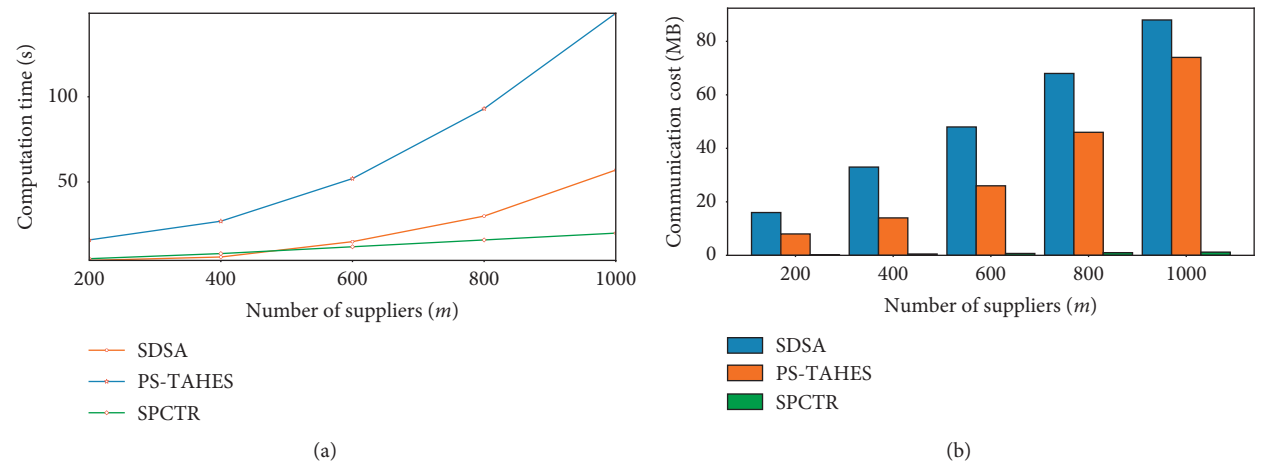

Figure 6: Performance comparison in computation time and communication cost. (a) Computation time. (b) Communication cost.

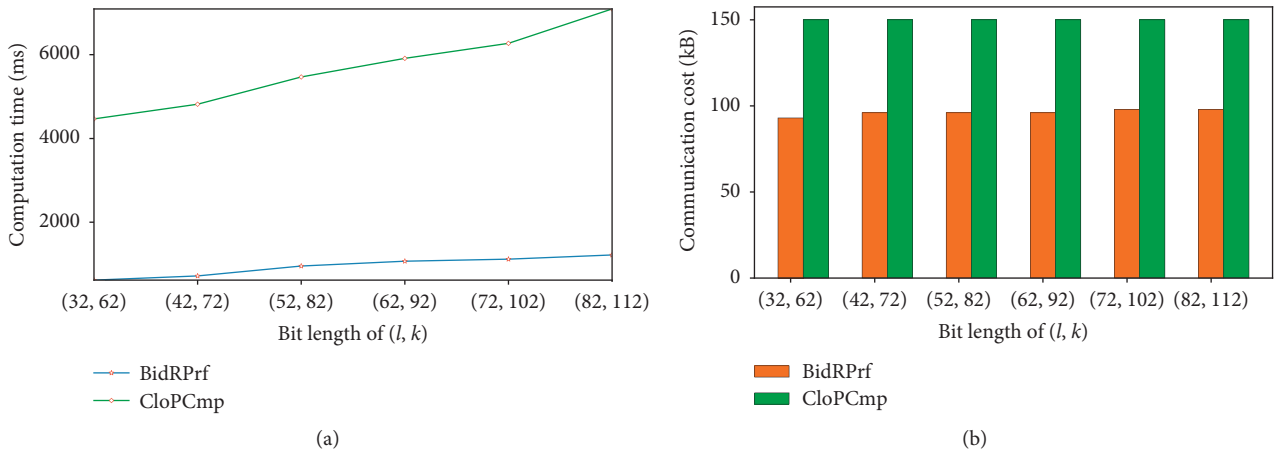

Figure 7: Computation time and communication cost with the bit lengths (l, k). (a) Computation time. (b) Communication cost.

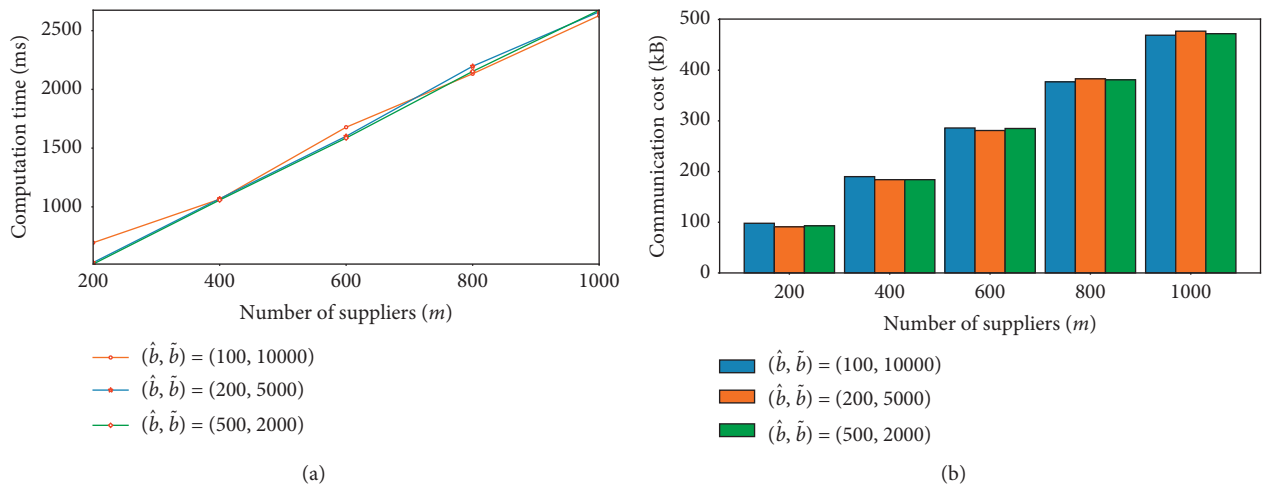

Figure 8: Computation time and communication cost in BidRPrf. (a) Computation time. (b) Communication cost.

Secure closest pre-tender comparison: in this phase, the computation and communication complexities are mainly generated from Algorithm 3, in which PM and PA interact with each other to determine the closest pre-tender supplier. Algorithm 3 includes three iterations which are composed of several specific operations (e.g., encryption, multiplication, exponentiation, decryption, etc,.) and one operation of invoking the function of MinValSel. Moreover, we get the observation that $2m$ number of values $(B_0, \ldots, B_{m-1}, B_0^* B_0^*, \ldots, B_{m-1}^* B_{m-1}^*)$ have been transmitted between PM and PA. Therefore, the computation and communication complexities in this phase are also $O(m)$. All in all, the computation and communication complexities of SPCTR are $O(m)$.

7. Experimental Evaluation

7.1. Parameter Settings. To demonstrate the practicality in the real world, the core cryptographic operations of SPCTR are prototypically implemented by Java. All the experiments were executed in a laptop with Intel i7-6560U CPU, 2.20 GHz clock. The parameters of SPCTR are sized as follows. The Paillier cryptosystem is implemented with a 1024-bit modulus, and 80-bit wire labels are used for garbled circuits.

We are mainly concerned with two metrics: the computation time and communication cost in the performance evaluation. We set the system parameters as follows. The number of suppliers m varies from 200 to 2000. The lowest bid $\widehat{b}$ spans from 50 to 500 and the highest bid $\widetilde{b}$ spans from 1000 to 10000. Moreover, the bit length l of a bid b_i is set from 32 to 82. The bit length k of a masked random value r_i is 30 bit longer than l so that k varies from 62 to 112. We set the default values of suppliers m, the lowest bid $\widehat{b}$, the highest bid $\widetilde{b}$, and the bit length l as 200, 100, 10000, and 32, respectively. All experimental results are based on the average values of 10 runs. Moreover, we compare the experimental results of our scheme with other SOTA (state-of-the-art) works including SDSA [21] and PS-TAHES [6].

7.2. Computation Time and Communication Cost. In Figure 6, when the number of suppliers is 1000, the total computation time and communication cost are 20.91 s and 1236 kB respectively, which are much superior to the costs in SDSA and PS-TAHES. To be specific, the computation time and communication cost of SDSA are nearly 57 s and 88 MB. Besides, the computation time and communication cost of PS-TAHES are about 149 s and 74 MB. Therefore, in comparison with SDSA and PS-TAHES, the computation time and communication costs are proved to be significantly reduced with the same strong privacy protection.

It is shown in Figure 7 that fixing the following factors, including the number of suppliers $m = 200$, the pre-tender $t = 1000$, and the bid range $[\widehat{b}, \widetilde{tb}] = [100, 10000]$, we select (l, k) in pairs. To keep the same statistical security level, the bit length of k is 30 bit longer than l. Therefore, for a l-bit b and a k-bit r, the masked value $b + r$ provides a statistical security of 2^{l-k} for b. In the experiments, the bit lengths (l, k) are varied from (32, 62) to (82, 112). We observe that the computation time increases almost linearly with the bit lengths (l, k) since the bit length increment affects the running time of operating on the bids, e.g., bid range validation in BidRPrf and ciphertext multiplication in CloPCmp. However, the communication cost almost remains the same with the bit lengths (l, k). This result is consistent with the aforementioned analysis that each supplier and PM need to exchange 7 values in BidRPrf. Moreover, in CloPCmp, the size of $2m$ data is transmitted between PM and PA. In the experiments which are implemented using Java, we use a constant number of BigInteger to store these values. Therefore, the communication cost is a constant irrespective of the bit lengths (l, k).

In Figure 8, we observe two main results. First, fixing the factor of the bid range, the computation time and communication cost increase linearly with the number of suppliers m. However, fixing the value of m, the computation time and communication cost are slightly affected by the bid range. Therefore, we validate that it is the number of suppliers m rather than the bid range that determines the cost of validating a bid whether it is in a specified range or not.

8. Related Work

8.1. Range Proof. In the decomposition-commitment-based range proof, the secret is decomposed into individual bits. Then, we demonstrate the commitments of these bits implying the number in the range [22, 23]. Instead of binary decomposition, the works of multibase decomposition are presented. The secret is decomposed in base-u (u is a chosen integer). Then, commitments of these u-ary digits are constructed to prove that each committed digit is indeed a digit in base-u [24, 25]. Although progress has been made so far, decomposition-commitment-based range proof is computationally expensive. Alternately, the idea of signature-based range proof is using the signatures of all the integers in a public interval [26, 27].

8.2. Sealed-Price Auction. Sealed-price auction is one common form of secure computations [28–30]. Since the seminal Yao's work in 1982 [17], there is a surge of extensive research endeavors in the sealed-price auction design [3–5]. In [6], garbled circuits are proposed to resolve the problem of secure sealed-bid comparison. In [7], Kolesnikov et al. constructed a secure comparison garbled circuit. To extend more secure implementations of garbled circuits, a minimum value selection circuit is proposed in [11].

9. Conclusion

In this paper, we have presented SPCTR, the first sealed-bid auction-based procurement bidding system with range validation. Different from previous works, SPCTR provides full privacy protection for the bid comparison while enabling the bid range validation without leaking the secret bids. SPCTR is constructed by leveraging carefully designed secure cryptographic tools. Then, security analysis and

performance analysis are presented. Later on, extensive experiments are conducted to verify the practicality of SPCTR. In comparison with the previous works, SPCTR can achieve sealed-bid comparison and range validation with much less computation time and communication cost.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Sichuan Science and Technology Program (no. 2020YFG0076), Science and Technology Research Project of Chongqing Municipal Education Commission of China (KJ1601401), Science and Technology Research Project of Chongqing University of Education (KY201725C), Chongqing Electronics Engineering Technology Research Center for Interactive Learning, Chongqing Big Data Engineering Laboratory for Children, and Innovative Research Group of Higher Institutions in Chongqing (Research on key technology and application of big data analysis in children's education).

References

[1] B. L. Oo, "Release of construction clients' pre-tender cost estimates: an experimental study," *Construction Economics and Building*, vol. 17, no. 4, pp. 37–47, 2017.

[2] S. Hassim, R. Muniandy, A. H. Alias, and P. Abdullah, "Construction tender price estimation standardization (Tpes) in Malaysia," *Engineering, Construction and Architectural Management*, vol. 25, no. 3, pp. 443–457, 2018.

[3] Q. Huang, Y. Gui, F. Wu, G. Chen, and Q. Zhang, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1881–1893, 2016.

[4] E.-O. Blass and F. Kerschbaum, "Strain: a secure auction for blockchains," in *Computer Security*, J. Lopez, J. Zhou, and M. Soriano, Eds., pp. 87–110, Springer International Publishing, Cham, Switzerland, 2018.

[5] J. Ma, B. Qi, and K. Lv, "Fully private auctions for the highest bid," in *Proceedings of the ACM Turing Celebration Conference—China, ACM TURC'19*, Chengdu, China, May 2019.

[6] Q. Wang, J. Huang, Y. Chen, X. Tian, and Q. Zhang, "Privacy-preserving and truthful double auction for heterogeneous spectrum," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 848–861, 2019.

[7] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, Cambridge University Press, Cambridge, UK, 2007.

[8] J. Camenisch, R. Chaabouni, and A. Shelat, "Efficient protocols for set membership and range proofs," in *Advances in Cryptology—ASIACRYPT 2008*, J. Pieprzyk, Ed., pp. 234–252, Springer Berlin Heidelberg, Berlin, Germany, 2008.

[9] T. Dimitriou, T. Giannetsos, and L. Chen, "Rewards: privacy-preserving rewarding and incentive schemes for the smart electricity grid and other loyalty systems," *Computer Communications*, vol. 137, pp. 1–14, 2019.

[10] R. Chaabouni, H. Lipmaa, and A. Shelat, "Additive combinatorics and discrete logarithm based range protocols," in *Information Security and Privacy*, R. Steinfeld and P. Hawkes, Eds., pp. 336–351, Springer Berlin Heidelberg, Berlin, Germany, 2010.

[11] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, Cambridge, UK, 2009.

[12] P. Bunn and R. Ostrovsky, *Secure Two-Party k-Means Clustering*, Association for Computing Machinery, New York, NY, USA, 2007.

[13] Y. Lindell, *How to Simulate It—A Tutorial on the Simulation Proof Technique*, Springer International Publishing, Cham, Switzerland, 2017.

[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT'99*, J. Stern, Ed., pp. 223–238, Springer Berlin Heidelberg, Berlin, Germany, 1999.

[15] M. Naor and B. Pinkas, "Computationally secure oblivious transfer," *Journal of Cryptology*, vol. 18, no. 1, pp. 1–35, 2005.

[16] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Advances in Cryptology—CRYPTO'86*, pp. 234–238, Springer, Berlin, Germany, 1986.

[17] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, IEEE, Chicago, IL, USA, November 1982.

[18] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *Proceedings of the 2013 IEEE INFOCOM*, pp. 2652–2660, Turin, Italy, April 2013.

[19] V. Kolesnikov, A.-R. Sadeghi, and T. Schneider, "Improved garbled circuit building blocks and applications to auctions and computing minima," in *Cryptology and Network Security*, J. A. Garay, A. Miyaji, and A. Otsuka, Eds., pp. 1–20, Springer Berlin Heidelberg, Berlin, Germany, 2009.

[20] C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols: Techniques and Constructions*, Springer Science & Business Media, Berlin, Germany, 2010.

[21] Z. Chen, X. Wei, H. Zhong, J. Cui, Y. Xu, and S. Zhang, "Secure, efficient and practical double spectrum auction," in *Proceedings of the 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, pp. 1–6, Vilanova i la Geltru, Spain, June 2017.

[22] K. Li, R. Yang, M. H. Au, and Q. Xu, "Practical range proof for cryptocurrency monero with provable security," in *Information and Communications Security*, S. Qing, C. Mitchell, L. Chen, and D. Liu, Eds., pp. 255–262, Springer International Publishing, Cham, Switzerland, 2018.

[23] M. Fischlin, "A cost-effective pay-per-multiplication comparison method for millionaires," in *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographers Track at RSA, Ser. CT-RSA 2001*, Springer-Verlag, San Francisco, CA, USA, April 2001.

[24] S. Canard, I. Coisel, A. Jambert, and J. Traoré, "New results for the practical use of range proofs," in *Public Key Infrastructures, Services and Applications*, S. Katsikas and I. Agudo, Eds., pp. 47–64, Springer Berlin Heidelberg, Berlin, Germany, 2014.

[25] S. Meiklejohn, C. C. Erway, A. Küpçü, T. Hinkle, and A. Lysyanskaya, "Zkpdl: a language-based system for efficient zero-knowledge proofs and electronic cash," in *Proceedings of the 19th USENIX Conference on Security, USENIX Security 10,*

USA: USENIX Association, p. 13, Berkeley, CA, USA, August 2010.

[26] I. Teranishi and K. Sako, "k-times anonymous authentication with a constant proving cost," in *Public Key Cryptography—KC 2006*, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds., pp. 525–542, Springer Berlin Heidelberg, Berlin, Germany, 2006.

[27] Y. Lindell, "Fast secure two-party ecdsa signing," in *Advances in Cryptology—CRYPTO 2017*, J. Katz and H. Shacham, Eds., pp. 613–644, Springer International Publishing, Cham, Switzerland, 2017.

[28] A. Peter, E. Tews, and S. Katzenbeisser, "Efficiently outsourcing multiparty computation under multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2046–2058, 2013.

[29] X. Liu, R. H. Deng, K.-K. Raymond Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[30] X. Liu, K.-K. Raymond Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.