PS-TRUST: Provably Secure Solution for Truthful Double Spectrum Auctions

Zhili Chen, Liusheng Huang, Lu Li, Wei Yang, Haibo Miao, Miaomiao Tian, Fei Wang
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou, China
Email: {zlchen3, lshuang}@ustc.edu.cn, liluzq@mail.ustc.edu.cn, qubit@ustc.edu.cn, {mhb, miaotian, scuwf}@mail.ustc.edu.cn

***Abstract*—Truthful spectrum auctions have been extensively studied in recent years. Truthfulness makes bidders bid their true valuations, simplifying greatly the analysis of auctions. However, revealing one's true valuation causes severe privacy disclosure to the auctioneer and other bidders. To make things worse, previous work on secure spectrum auctions does not provide adequate security. In this paper, based on TRUST, we propose PS-TRUST, a provably secure solution for truthful double spectrum auctions. Besides maintaining the properties of truthfulness and special spectrum reuse of TRUST, PS-TRUST achieves provable security against semi-honest adversaries in the sense of cryptography. Specifically, PS-TRUST reveals nothing about the bids to anyone in the auction, except the auction result. To the best of our knowledge, PS-TRUST is the first provably secure solution for spectrum auctions. Furthermore, experimental results show that the computation and communication overhead of PS-TRUST is modest, and its practical applications are feasible.**

I. Introduction

As the rapid development of wireless technologies, the scarcity of radio spectrum attracts more and more attention. Under the traditional static spectrum allocation scheme by government, the utilization of the radio spectrum is severely inefficient. Many spectrum channels are idle most of the time under their current owners, whereas ever-increasing new wireless users are starving for spectrum. Therefore, spectrum redistribution is highly significant for improving the overall spectrum utilization and thus alleviating the problem of spectrum scarcity. Open markets for spectrum redistribution, such as Spectrum Bridge [13], have already appeared to provide services for buying, selling, and leasing idle spectrum.

As a well-known approach to spectrum redistribution, spectrum auctions are preferred by people for its fairness and allocation efficiency. In recent years, there have been extensive studies on spectrum auctions, most of which achieve truthfulness to make bidders reveal their true valuations of spectrum channels. However, revealing one's true valuation causes severe privacy disclosure. Literature [20] illustrated two vulnerabilities of truthful auctions, i.e. frauds of the insincere auctioneer, and bid-rigging between the auctioneer and the bidders, in which the auctioneer takes advantage of the knowledge of bidders' bids. Furthermore, when one bidder knows other bidders' bids after an auction, he will probably not bid his true valuation in repeated auctions. That is, an original truthful auction will probably become untruthful when repeated, due to the revelation of all bidders' bids in the previous auctions [3]. Therefore, protecting the privacy of bidders is of great importance.

978-1-4799-3360-0/14/$31.00

Fig. 1. Auction Framework for PS-TRUST

There have been many researches on privacy preserving auctions, such as [8][11][9][10]. However, spectrum is quite different from traditional goods, for it can be well reused in both spatial and time dimensions. Thus, traditional privacy preserving auctions cannot be directly applied to spectrum auctions. Recently, some works about privacy preserving spectrum auctions have also been proposed [20][21]. These works dealt with only single-sided spectrum auctions. Furthermore, they fell short of providing adequate security. In the sense of cryptography, a protocol is secure implies that no participating party can learn any information beyond the output of the protocol. However, both the two approaches reveal some information that cannot be inferred from the outputs. For example, in [20], the auctioneer can easily get the sums of bids for all the possible allocations for each subnetwork by decrypting $\mathbf{E}_\xi$; in [21], the auctioneer gets to know the bids of all buyer groups and their ranking order in the auctions. The information mentioned above is more than the auction result, which normally includes the winner set and the pricing information.

In this paper, we propose PS-TRUST, a provably secure solution for truthful double spectrum auctions. The auction framework of PS-TRUST is shown as in Fig. 1. This framework introduces an auction agent who cooperates with the auctioneer to securely compute the auctions. Neither the auctioneer nor the auction agent is a trusted party, but they are assumed not to collude with each other. Furthermore, we restrict that bidders can only communicate with the auctioneer, keeping the communication pattern simple and identical to that of an insecure auction. PS-TRUST reveals nothing but the auction result including the selling and buying clearing prices, and the seller and buyer winner sets. The main contributions can be summarized as follows.

(1) We design PS-TRUST based on homomorphic encryp-

tion schemes. By representing the bids in encrypted bit vectors (EBVs), we design secure algorithms for addition, constant multiplication, and maximum/minimum selection for EBV bids. And then, based on these algorithms, we present a secure auction procedure, which reveals nothing about the bids except the auction result.

(2) We apply the definition of security against semi-honest adversaries to formally prove the security of PS-TRUST. To the best of our knowledge, this is the first work to formally prove the security, in the sense of cryptography, of a solution to spectrum auctions.

(3) We analyze the computation and communication complexities of PS-TRUST, implement it in Java to evaluate running times and message volumes, and conclude that its computation and communication overhead is modest.

The remainder of this paper is organized as follows. In Section II, a brief review of related work is given. In Section III, we describe the problem statement. Next, we provide some preliminaries in Section IV. In Section V, we present the detailed design of PS-TRUST, and prove formally its security. Then, in Section VI, we implement PS-TRUST, analyze and evaluate its computation and communication overhead. Finally, we conclude our work in Section VII.

II. Related Work

Spectrum auctions have been studied extensively in recent years. For instance, Zhou et al. proposed VERITAS [1], a single-sided truthful spectrum auction supporting diverse bidding formats. Zhou et al. proposed TRUST [2], the first truthful double spectrum auction framework enabling spectrum reuse. Deek et al. proposed Topaz [14] to tackle time-based cheating in online spectrum auctions. Al-Ayyoub and Gupta [15] designed a polynomial-time truthful spectrum auction mechanism with a performance guarantee on revenue. Xu et al. [16][17] proposed efficient online spectrum allocations in multi-channel wireless networks. TAHES [18] addressed the issue of heterogeneous spectrums in truthful double spectrum auctions. Dong et al. [19] tackled the spectrum allocation problem with time-frequency flexibility in cognitive radio networks via combinatorial auction. However, most of the existing spectrum auction mechanisms do not provide any guarantee of security.

Extensive work has focused on privacy preserving auction design in the past decade. Brandt and Sandholm [12] investigated unconditional full privacy in sealed-bid auctions. In [8] [11][9][10] the authors employed various cryptography techniques to achieve security in diverse auction schemes. Unfortunately, when applied to spectrum auctions, these traditional privacy preserving auctions either require exponential complexity, or lead to significant degradation of spectrum utilization. Recently, papers [20] and [21] provide solutions for privacy preserving spectrum auctions, but they only addressed single-sided spectrum auctions. What is more, as mentioned above, they fell short of providing security in the sense of cryptography.

III. Problem Statement

A. Auction Problem

We consider a double spectrum auction, which is single-rounded with one auctioneer $\mathcal{A}$, a seller set $\mathbb{S} = \{s_1, s_2, ..., s_M\}$, and a buyer set $\mathbb{B} = \{b_1, b_2, ..., b_N\}$. In the auction, each seller s_i contributes exactly one channel and each buyer b_j requests only one channel. The channels are homogenous to buyers so that their requests are not channel specific. Each channel contributed by sellers can potentially be reused by multiple non-conflicting buyers who are separated far enough.

B. TRUST

TRUST [2] has provided a truthful framework for this double spectrum auction problem, with spatial spectrum reuse being well exploited. Since TRUST [2] is based on McAfee's double auction design, we briefly review both of them.

1) McAfee's Design: McAfee's design of double auctions is most widely used [6], which achieves economic properties of truthfulness, individual rationality, and ex-post budget balance. This design assumes that there are M sellers and N buyers, and all goods auctioned are homogenous. Each seller s_i bids v_i^s to sell a good, and each buyer b_j bids v_j^b to buy a good. The auction proceeds as follows:

(1) Bid sorting: Sort bids of sellers in non-decreasing order and bids of buyers in non-increasing order:

$$v_1^s \leq v_2^s \leq ... \leq v_M^s$$
$$v_1^b \geq v_2^b \geq ... \geq v_N^b$$

(2) Winner determination: Find $k = \arg\max\{v_k^s \leq v_k^b\}$, the index of the last profitable transaction. Then the first $(k-1)$ sellers and the first $(k-1)$ buyers are the auction winners.

(3) Pricing: Pay each winning seller equally by v_k^s, and charge each winning buyer equally by v_k^b.

2) TRUST Design: TRUST followed the methodology of McAfee's design, and enabled spectrum spatial reuse. It consists of the following three steps:

(1) Buyer group formation: form non-conflicting buyer groups based on buyers' conflict graph (Any two buyers share an edge in the graph if the reuse of the same channel causes interference between them.) but independent of their bids.

(2) Winner determination: Each buyer group bids a value obtained by multiplying its smallest buyer bid with its size, and acts as a single "buyer". Then the auctioneer applies just the same winner determination as that of the McAfee's design, resulting in that the first $(k-1)$ sellers and the buyers in the first $(k-1)$ buyer groups are the auction winners.

(3) Pricing: Pay each winning seller equally by the k^{th} seller bid, and charge each buyer group equally by the k^{th} buyer group bid, which is evenly shared among the buyers in the group.

C. Securing TRUST

As described above, TRUST has provided a good solution to the auction problem mentioned. However, in TRUST, no security issues are considered, and all bids are completely exposed to the auctioneer, and even to all bidders. This

could result in the following two problems: (1) A dishonest auctioneer could temper the auction result to increase his utility [20]; (2) The knowledge of the historical true valuations of other bidders could make one bidder conceal his true valuation in a repetition of a truthful auction [3].

In this work, our aim is to secure TRUST by protecting the privacy of bidders, i.e., their bids. However, how to correctly compute the auction while reveal nothing about the bids beyond the auction result (including selling and buying clearing prices, seller and buyer winner sets) in the auction process, is challenging. Furthermore, how to prove the security in the sense of cryptography is non-trivial, too.

IV. Preliminaries

In this section, we introduce some preliminaries for the design of PS-TRUST.

A. Security Formulation

In cryptography area, the standard security formulation is called ideal/real simulation paradigm [4] [5], as shown in Fig. 2. In this formulation, a real protocol execution in the "real world" is mapped to an ideal functionality calling in the "ideal world". In the ideal world, there is an external trusted (and incorruptible) party willing to help the parties carry out their computation. The ideal functionality calling means that the parties simply send their inputs to the trusted party, which computes the desired functionality and passes each party its prescribed output. While, in the real world, there is no external trusted party, and the real protocol execution means the parties run the protocol amongst themselves without any help. We say that a protocol is *secure* if its real protocol execution emulates its ideal functionality calling. That is, no adversary can do more harm in its real protocol execution than in its ideal functionality calling. However, successful adversarial attacks cannot be performed in the ideal functionality calling. We therefore conclude that all adversarial attacks on the real protocol execution must also fail for a secure protocol.

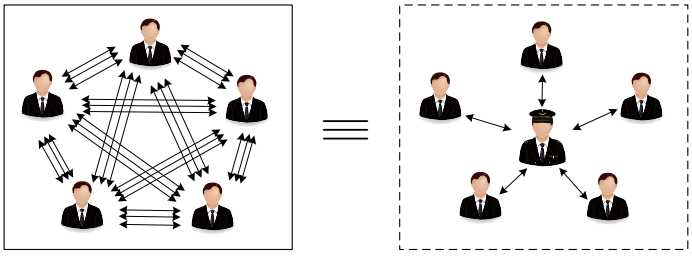

Fig. 2. The Security Formulation of Ideal/Real Simulation Paradigm

B. Adversarial Models

Under the security formulation of "ideal/real simulation paradigm", the adversarial models can be classified as semi-honest adversarial model and malicious adversarial model [4].

In semi-honest adversarial model, even a corrupted party correctly follows the protocol specification. However, the adversary obtains the internal state of all the corrupted parties, and attempts to use this to learn information more than the output. This adversarial model may be used in settings where running the "correct" protocol can be enforced. Semi-honest adversaries are also called "honest-but-curious adversaries" and "passive adversaries".

In malicious adversarial model, the corrupted parties can arbitrarily deviate from the protocol specification, according to the adversary's instructions. Security against malicious adversaries is so strong that it ensures that no adversarial attack can succeed. Malicious adversaries are also called "active adversaries".

Although protocols secure against malicious adversaries exist theoretically, they are far too inefficient to implement. So, in this paper, we apply semi-honest adversarial model for the cause of practical applications. Specifically, in our context, we assume that the auctioneer and the auction agent follow the auction protocol specification, but one of them could act as a semi-honest adversary. The adversary obtains the internal state of the auction, and attempts to learn information about the bids beyond the auction result.

C. Paillier Cryptosystem

In order to achieve the security of spectrum auctions, a semantically secure cryptosystem is needed. In our design, Paillier's homomorphic cryptosystem (G, E, D) is applied, where G, E and D denote the key generation algorithm, encryption algorithm, and decryption algorithm, respectively. The properties of a Paillier cryptosystem include homomorphic addition, indistinguishability, and self-blinding [7]:

(1) Homomorphic addition: The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts, and the k^{th} power of a ciphertext will decrypt to the product of k and its corresponding plaintext.

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n$$

$$D(E(m, r)^k \bmod n^2) = k \cdot m \bmod n$$

where n is the product of two large primes, which is public to users, and r_1, r_2 and r are random numbers.

(2) Indistinguishability: If the same plaintext m is encrypted twice, these two ciphertexts $E(m, r_1)$ and $E(m, r_2)$ are totally different, and no one can succeed in distinguishing them with a significantly higher probability than random guess without decrypting them.

(3) Self-blinding: Any ciphertext can be publicly changed into another one without affecting the plaintext. This means that a randomized chipertext $E(m, r')$ can be computed from the ciphertext $E(m, r)$ without knowing either the decryption key or the original plaintext.

V. PS-TRUST

In this section, we present the design of PS-TRUST. We first describe the secure bid representation and operations, then present the detailed secure auction design, and finally prove formally that PS-TRUST is secure against semi-honest adversaries.

A. Secure Bid Representation and Operations

In PS-TRUST, we use encrypted bit vectors to securely represent bids.

Definition 1 (Encrypted Bit Vector)**.** *The Encrypted Bit Vector (EBV) representation of value v is a vector $\boldsymbol{e}(v)$ of ciphertexts like*

$$\boldsymbol{e}(v) = (e_1, e_2, ..., e_K) = (E(\sigma_1), E(\sigma_2), ..., E(\sigma_K)) \quad (1)$$

where $E(.)$ is Paillier's encryption function, K is the bit length, $(\sigma_1, \sigma_2, ..., \sigma_K)$ denotes the binary representation of v, with σ_1 the most significant bit, and σ_K the least significant bit.

With the definition of EBV, we can develop secure algorithms for EBV bid operations including addition, constant multiplication, and minimum/maximum selection. With these algorithms, the algorithm runner (AR) without the secret key can compute the corresponding bid operations on EBV bids, and get an encrypted result, knowing nothing about the bids. Then this encrypted result can be used as either an intermediate result for further computations or a part of the final output decrypted by the key holder (KH) with the secret key.

But how do we compute on EBV bids? Due to the homomorphic addition, addition of two values in $\mathbb{Z}_n$ can be computed directly by multiplying their ciphertexts, while multiplication can be computed with the help of the KH who can do decryption using Protocol 1. Furthermore, the XOR operation signified by $\oplus$ can be turned into additions and multiplications in $\mathbb{Z}_n$ by the fact that:

$$c \oplus d = c + d - 2cd \quad (2)$$

Thus, to design the secure algorithms for the operations on EBV bids, we only need to turn all operations into additions and multiplications in $\mathbb{Z}_n$, and XOR operations.

Protocol 1 Product of Two Numbers in $\mathbb{Z}_n$

Require:
AR holds $E(x)$ and $E(y)$
Ensure:
AR holds $E(xy)$
Step AR1:
1: $x_1 \in_R \mathbb{Z}_n$; $y_1 \in_R \mathbb{Z}_n$; // Select randomly
2: $E(x_2) = E(x).E(-x_1)$; // $x_2 = x - x_1 \bmod n$;
3: $E(y_2) = E(y).E(-y_1)$; // $y_2 = y - y_1 \bmod n$;
4: Sends $E(x_2)$ and $E(y_2)$ to AA;
Step KH2:
5: $x_2 = D(E(x_2))$; $y_2 = D(E(y_2))$;
6: Sends $E(x_2 y_2)$ to AE;
Step AR3:
7: $E(xy) = E(x_1 y_1) \cdot E(y_2)^{x_1} \cdot E(x_2)^{y_1} \cdot E(x_2 y_2)$;

According to the discussion above, the secure algorithms for EBV bid addition and EBV bid constant multiplication are straightforward, and are shown in Algorithms 2 and 3, respectively.

Now, we design secure algorithms for minimum selection. We first consider the two-bid case. Suppose that the AR holds

Protocol 2 EBVAdd($\mathbf{e}(v^A), \mathbf{e}(v^B)$)

Input:
EBV bids $\mathbf{e}(v^A)$ and $\mathbf{e}(v^B)$
Output:
Sum $\mathbf{e}(v^{AB})$
1: Compute Line 2 to 6 over encrypted bits $E(\sigma_i^A)$ and $E(\sigma_i^B)$, where $1 \le i \le K$, using homomorphic properties and Protocol 1.
// For clarity, we describe these lines by plain bits.
2: $\sigma_K^{AB} = \sigma_K^A \oplus \sigma_K^B$; $c_K^{AB} = \sigma_K^A \cdot \sigma_K^B$;
3: **for** ($i = K-1$; $i >= 1$; $i = i-1$) **do**
4: $\quad \sigma_i^{AB} = \sigma_i^A \oplus \sigma_i^B \oplus c_{i+1}^{AB}$;
5: $\quad c_i^{AB} = \sigma_i^A \cdot \sigma_i^B \oplus \sigma_i^A \cdot c_{i+1}^{AB} \oplus \sigma_i^B \cdot c_{i+1}^{AB}$;
6: **end for**
7: $\mathbf{e}(v^{AB}) = (E(\sigma_1^{AB}), E(\sigma_2^{AB}), ..., E(\sigma_K^{AB}))$;
8: **return** $\mathbf{e}(v^{AB})$;

Protocol 3 EBVMul($\mathbf{e}(v), m$)

Input:
EBV bid $\mathbf{e}(v)$ and integer $m = (\sigma_1^{(m)}, \sigma_2^{(m)}, ..., \sigma_K^{(m)})$
Output:
Product $\mathbf{P} = \mathbf{e}(m \cdot v)$
1: $\mathbf{P} = \mathbf{e}(0)$;
2: **for** ($i = 1$; $i <= K$; $++i$) **do**
3: $\quad$ **if** ($\sigma_i^{(m)} == 1$) **then**
4: $\quad\quad \mathbf{P}^* = \mathbf{e}(v)$ shifted left $(K - i)$ bits;
5: $\quad\quad \mathbf{P} = \text{EBVAdd}(\mathbf{P}, \mathbf{P}^*)$;
6: $\quad$ **end if**
7: **end for**
8: **return** $\mathbf{P}$;

two EBV bids, denoted by

$$\begin{aligned} e(v^A) &= (E(\sigma_1^A), E(\sigma_2^A), ..., E(\sigma_K^A)), \text{ and} \\ e(v^B) &= (E(\sigma_1^B), E(\sigma_2^B), ..., E(\sigma_K^B)) \end{aligned} \quad (3)$$

It can compute the location of the minimum bid as

$$\begin{aligned} R_{AB}^{min} = (\sigma_1^A \oplus \sigma_1^B)\sigma_1^A + (\sigma_1^A \oplus \sigma_1^B \oplus 1)(\sigma_2^A \oplus \sigma_2^B)\sigma_2^A + \\ (\sigma_1^A \oplus \sigma_1^B \oplus 1)(\sigma_2^A \oplus \sigma_2^B \oplus 1)(\sigma_3^A \oplus \sigma_3^B)\sigma_3^A + ... + \\ (\sigma_1^A \oplus \sigma_1^B \oplus 1)...(\sigma_{K-1}^A \oplus \sigma_{K-1}^B \oplus 1)(\sigma_K^A \oplus \sigma_K^B)\sigma_K^A \end{aligned} \quad (4)$$

on the encrypted bits, where R_{AB}^{min} is defined as

$$R_{AB}^{min} = \begin{cases} 0, \ if \ v^A \le v^B \\ 1, \ if \ v^A > v^B \end{cases} \quad (5)$$

Therefore, we can design the secure algorithm for two-bid minimum selection as shown in Algorithm 4. Note that the order of the two bids matters in the result. If the two bids are equal, the first one is picked up.

Based on Algorithm 4, we can develop the algorithm for multi-bid minimum selection as shown in Algorithm 5.

In Algorithm 5, the inputs are EBV bids of a set of bidders indexed from 1 to n. $\mathbf{e}(v^{i*})$ represents the minimum EBV bid of bidders from bidders 1 to i. $R_{i*,i+1}^{min}$ denotes the comparison result of the minimum bid of bidders from 1 to i and the bid of bidder $i+1$, with 0 meaning the former is not greater than the latter, 1 otherwise. $R_{1,i}^{min}$ denotes the index (starting from 0) of the first bidder with the minimum bid among the bidders from 1 to i.

Algorithm 4 TwoBidMin($\mathbf{e}(v^A), \mathbf{e}(v^B)$)

Input:
EBV bids $\mathbf{e}(v^A)$ and $\mathbf{e}(v^B)$
Output:
Comparison result $E(R_{AB}^{min})$ and minimum bid $\mathbf{e}(v^{AB})$
1: Compute Line 2 to 9 over encrypted bits $E(\sigma_i^A)$ and $E(\sigma_i^B)$, where $1 \leq i \leq K$.
// For clarity, we describe these lines by plain bits.
2: **for** ($i = 1$; $i <= K$; $++i$) **do**
3: $x_i^{AB} = \sigma_i^A \oplus \sigma_i^B$; $x_i^{AB*} = x_i^{AB} \oplus 1$;
4: **end for**
5: $R_{AB}^{min} = x_1^{AB} \cdot \sigma_1^A$; $R = 1$;
6: **for** ($i = 2$; $i <= K$; $++i$) **do**
7: $R = R \cdot x_{i-1}^{AB*}$;
8: $R_{AB}^{min} = R_{AB}^{min} + R \cdot x_i^{AB} \cdot \sigma_i^A$;
9: **end for**
10: Compute $\mathbf{e}(v^{AB}) = (E(\sigma_1^{AB}), E(\sigma_2^{AB}), ..., E(\sigma_K^{AB}))$, where $\sigma_j^{AB} = \sigma_j^A \cdot (1 - R_{AB}^{min}) + \sigma_j^B \cdot R_{AB}^{min}$, $1 \leq j \leq K$;
11: **return** $(E(R_{AB}^{min}), \mathbf{e}(v^{AB}))$;

Algorithm 5 MultiBidMin($E(\mathbb{B})$)

Input:
EBV bids $E(\mathbb{B}) = \{\mathbf{e}(v^i) | 1 \leq i \leq m\}$
Output:
Comparison result $E(R_{1,m}^{min})$ and minimum bid $\mathbf{e}(v^{m*})$
1: $(E(R_{1,2}^{min}), \mathbf{e}(v^{2*})) =$ TwoBidMin($\mathbf{e}(v^1), \mathbf{e}(v^2)$);
2: **for** ($i = 2$; $i < m$; $++i$) **do**
3: $(E(R_{i*,i+1}^{min}), \mathbf{e}(v^{(i+1)*})) =$ TwoBidMin($\mathbf{e}(v^{i*}), \mathbf{e}(v^{i+1})$);
4: $E(R_{1,i+1}^{min}) = E(R_{1,i}^{min} \cdot (1 - R_{i*,i+1}^{min}) + (i+1) \cdot R_{i*,i+1}^{min})$;
5: **end for**
6: **return** $(E(R_{1,m}^{min}), \mathbf{e}(v^{m*}))$;

It is trivial to use Algorithms 4 and 5 for maximum selection (by inverting the bits of EBV bids and then calling the minimum selection algorithms). In the following, we directly use algorithms TwoBidMax(.,.) and MultiBidMax(.) for maximum selection.

B. Secure Auction Design

Based on the secure bid representation and operations, we now present the secure auction design. Our main idea is that the auction agent first runs the key generation algorithm of Paillier cryptosystem, and publishes the public key to the auctioneer and the bidders. Next, all bidders convert their bids to EBV bids using the public key, and send these EBV bids to the auctioneer. Then, the auctioneer computes the auction on the EBV bids and gets an encrypted auction result, with the help of the auction agent. Finally, the auctioneer gets the auction result by asking the auction agent to decrypt it, and reports the auction result to the bidders. As long as the auctioneer and the auction agent do not collude with each other, they can get nothing about the bids, except the auction result. PS-TRUST includes three steps as follows.

1) Buyer Group Formation: Buyers submit their location information to the auctioneer, who generates a conflict graph of buyers based on the information. Without knowing the bid values of the buyers, the auctioneer forms buyers into non-conflict buyer groups based on the conflict graph. Specifically, the auctioneer forms buyer groups by finding independent sets in the conflict graph repeatedly. To find an independent set, the auctioneer randomly chooses a node in the current conflict graph to add to the set, eliminates the node and its neighbors, and updates the conflict graph. This is repeated recursively until the conflict graph is empty, then an independent set is found. We denote by $\mathbb{G} = \{\mathbb{G}_1, \mathbb{G}_2, ..., \mathbb{G}_H\}$ the set of non-conflict buyer groups formed.

2) Secure Auction Computation: In this step, all bidders submit their EBV bids to the auctioneer. Then, the auctioneer and the auction agent cooperate to securely compute the auction. This step can be divided into further two steps:

(1) Buyer Group Bidding

For each buyer group $\mathbb{G}_i$ ($1 \leq i \leq H$), the auctioneer finds its minimum EBV bid $\mathbf{e}(v_i^{min})$ by calling $(R_{\mathbb{G}_i}^{min}, \mathbf{e}(v_i^{min})) =$ MultiBidMin $(E(\mathbb{G}_i))$, and compute its EBV group bid $\mathbf{e}(v_i^g) =$ EBVMul($\mathbf{e}(v_i^{min}), n_i$), where $E(\mathbb{G}_i)$ denotes the EBV bid set of group $\mathbb{G}_i$, and $n_i = |\mathbb{G}_i|$. Note that the auctioneer knows nothing about the buyers' bids. At the end, the auctioneer holds the EBV group bid of each buyer group.

(2) Winner Determination

A natural idea for winner determination proceeds as follows. The auctioneer finds the encrypted seller index (starting from 1) $E(\alpha)$ with the minimum bid in the seller set $\mathbb{S}$, the encrypted buyer group index $E(\beta)$ with the maximum group bid in the buyer group set $\mathbb{G}$, and their corresponding EBV bids (using Algorithm 5), then compares the two EBV bids to get an encrypted result (using Algorithm 4). $E(\alpha)$, $E(\beta)$, and the comparison result are sent to the auction agent, who decrypts these encrypted data, and sends the decrypted information (α, β, and the comparison result) back to the auctioneer. Then if the trading condition, namely, the maximum group bid is not less than the minimum seller bid, is satisfied, the auctioneer removes α from $\mathbb{S}$, β from $\mathbb{G}$, and adds α to a winner-candidate seller set $\mathbb{W}^s$, β to a winner-candidate buyer group set $\mathbb{W}^g$. Otherwise, the auction is over. This process can be repeated to find the winner-candidate seller-buyer-group pairs until either the seller set or the buyer group set is empty, or the trading condition is unsatisfied. At last, the auctioneer removes the last added seller α_c from $\mathbb{W}^s$, treating it as the critical seller, and removes the last added buyer group β_c from $\mathbb{W}^g$, treating it as the critical buyer group. Then, the auctioneer reports the sellers in $\mathbb{W}^s$ and the buyers belonging to the buyer groups in $\mathbb{W}^g$ as winners, and the bid of α_c and the group bid of β_c (which are decrypted by auction agent) as the selling and buying clearing prices, respectively.

The idea above seems to work well: the auctioneer and the auction agent cooperate to determine the winners and no exact bids are leaked to them. However, there is some information about the bids leaking. Specifically, the ranking orders of the winning sellers' bids and the winning buyer groups' group bids are leaked to both the auctioneer and the auction agent. The leaked information is obviously more than what we can infer from the auction result including the winner sets and the clearing prices. Thus, in the sense of cryptography, the above procedure is not really secure.

In order to make this natural procedure of winner determination secure, something has to be done to hide the bid ranking orders of winners. Our idea is that, the auctioneer uses the randomized seller set $\mathbb{S}'$ and buyer group set $\mathbb{G}'$, instead of the original ones, so that each time when the auction agent decrypts the comparison result of a seller-buyer-group pair, he does not know which the pairs are. The auction agent then indicates the selected winner-candidate pairs by encrypted bit vectors, which are sufficient for computing the next winner-candidate pair while reveals nothing about the selection orders to the auctioneer. Finally, when the auction is over, the auction result is decrypted by the auction agent to the auctioneer. The improved winner determination procedure is depicted in Protocol 6. Some details are explained as follows.

In Step AE1, the auctioneer applies random permutations π_s and π_g to seller set $\mathbb{S}$ and buyer group set $\mathbb{G}$, respectively, getting the randomized sets $\mathbb{S}'$ and $\mathbb{G}'$. Note that only the auctioneer knows the permutations.

In Step AA2, two bit vectors $\mathbf{w}^s$ and $\mathbf{w}^g$ are defined to indicate the winner locations in the randomized sets $\mathbb{S}'$ and $\mathbb{G}'$, respectively. $w_k^s = 1$ if seller s_{i_k} is a candidate winner, $w_k^s = 0$ otherwise, and $w_k^g = 1$ if buyer group $\mathbb{G}_{j_k}$ is a candidate winner, $w_k^g = 0$ otherwise. α_c and β_c index the critical seller and buyer group, respectively.

In Step AE3, similarly to the natural idea, the encrypted seller index $E(\alpha)$ with the minimum bid and the encrypted buyer group index $E(\beta)$ with the maximum bid, together with their EBV bids are computed using Algorithm 5. The resulted two EBV bids are then compared using Algorithm 4. These computation results remain in the encrypted form, unknown to the auctioneer. Note that, different from the natural idea, the randomized sets $\mathbb{S}'$ and $\mathbb{G}'$ are used instead.

In Step AA4, the auction agent decrypts the computation results in Step AE3, knowing the locations of the candidate-winner pair in the randomized sets $\mathbb{S}'$ and $\mathbb{G}'$. However, he does not know the random permutations, so he cannot know the true candidate winners. Line 12 tests if the buyer group's bid is not less than the seller's bid. If so, the auction agent sets the corresponding bits of $\mathbf{w}^s$ and $\mathbf{w}^g$ to 1, saves indexes of the last candidate-winner pair, and sends $E(\mathbf{w}^s)$, $E(\mathbf{w}^g)$ and $R_{\beta\alpha}^{max}$ to the auctioneer. Otherwise, the auction is over, and auction agent removes the last candidate-winner pair (i.e. the critical seller α_c and buyer group β_c) from candidate winner sets by setting the corresponding bits of $\mathbf{w}^s$ and $\mathbf{w}^g$ to 0. The auction agent then sends the plain values including $\mathbf{w}^s$, $\mathbf{w}^g$ and $R_{\alpha\beta}^{max}$ to the auctioneer.

In Step AE5, Line 22 tests if seller α and buyer group β can be included to the winner-candidate sets. If so, the auctioneer first saves the EBV bids of the last winner-candidate pair in Line 23, and then updates the EBV bids of all sellers and all buyer groups in Lines 24 and 25, respectively. This updating results that the bid of seller s_{i_k} is set to $(2^K - 1)$ if $w_k^s == 1$, while remains unchanged otherwise, and the bid of buyer group $\mathbb{G}_{j_k}$ is set to 0 if $w_k^g == 1$, while remains unchanged otherwise. That is, all selected winner-candidate sellers are mapped to a maximum value $(2^K - 1)$, and all selected winner-candidate buyer groups are mapped to a minimum bid value 0. As long as the normal bid satisfies $0 < v < 2^K - 1$, the selected winner candidates will not be selected in Step AE3, and the updating is equivalent to removing the winner candidates from the seller set and buyer group set. After doing this updating, the execution goes to Step AE3. If the test of Line 22 fails, the auction repeating is over and the execution goes to Step AE6.

In step AE6, the auctioneer gets v_c^s and v_c^g by asking the auction agent to decrypt $\mathbf{e}(v_c^s)$ and $\mathbf{e}(v_c^g)$, and computes

Protocol 6 Winner Determination

Input:

Auctioneer (AE) holds:
EBV bids $\mathbf{e}(v_i^s)$ of seller s_i, for $1 \le i \le M$;
EBV group bids $\mathbf{e}(v_j^g)$ of buyer group $\mathbb{G}_j$, for $1 \le j \le H$;
Seller set $\mathbb{S} = \{s_1, s_2, ..., s_M\}$;
Buyer group set $\mathbb{G} = \{\mathbb{G}_1, \mathbb{G}_2, ..., \mathbb{G}_H\}$.

Output:

Auctioneer and **Auction Agent (AA)** hold:
The selling and buying clearing prices v_c^s and v_c^g;
Winning seller set $\mathbb{W}^s$;
Winning buyer group set $\mathbb{W}^g$.

Step AE1: AE Initialization:

1: $\mathbb{W}^s = \emptyset$; $\mathbb{W}^g = \emptyset$;
2: $\mathbb{S}' = \pi_s(\mathbb{S}) = \{s_{i_1}, s_{i_2}, ..., s_{i_M}\}$;
3: $\mathbb{G}' = \pi_g(\mathbb{G}) = \{\mathbb{G}_{j_1}, \mathbb{G}_{j_2}, ..., \mathbb{G}_{j_H}\}$;

Step AA2: AA Initialization:

4: $\mathbf{w}^s = (w_1^s, w_2^s, ..., w_M^s) = (0, 0, ..., 0)$;
5: $\mathbf{w}^g = (w_1^g, w_2^g, ..., w_H^g) = (0, 0, ..., 0)$;
6: $\alpha_c = -1$; $\beta_c = -1$;

Step AE3: Finding a Seller-Buyer-Group Pair:

7: $(E(\alpha), \mathbf{e}(v_\alpha^s)) =$ MultiBidMin$(E(\mathbb{S}'))$;
8: $(E(\beta), \mathbf{e}(v_\beta^g)) =$ MultiBidMax$(E(\mathbb{G}'))$;
9: $(E(R_{\beta\alpha}^{max}), \mathbf{e}(v_{\beta\alpha}^{max})) =$ TwoBidMax$(\mathbf{e}(v_\beta^g), \mathbf{e}(v_\alpha^s))$;
10: AE sends $E(\alpha)$, $E(\beta)$, and $E(R_{\beta\alpha}^{max})$ to AA;

Step AA4: Determining a Winner-Candidate Pair:

11: $\alpha = D(E(\alpha))$; $\beta = D(E(\beta))$; $R_{\beta\alpha}^{max} = D(E(R_{\beta\alpha}^{max}))$;
12: **if** $(R_{\beta\alpha}^{max} == 0)$ **then**
13: $\quad w_{\alpha+1}^s = 1$; $w_{\beta+1}^g = 1$;
14: $\quad \alpha_c = \alpha$; $\beta_c = \beta$;
15: $\quad E(\mathbf{w}^s) = (E(w_1^s), E(w_2^s), ..., E(w_M^s))$;
16: $\quad E(\mathbf{w}^g) = (E(w_1^g), E(w_2^g), ..., E(w_H^g))$;
17: $\quad$ AA sends $E(\mathbf{w}^s)$, $E(\mathbf{w}^g)$, and $R_{\beta\alpha}^{max}$ to AE;
18: **else**
19: $\quad w_{\alpha_c+1}^s = 0$; $w_{\beta_c+1}^g = 0$;
20: $\quad$ AA sends $\mathbf{w}^s$, $\mathbf{w}^g$ and $R_{\beta\alpha}^{max}$ to AE;
21: **end if**

Step AE5: Auction Repeating:

22: **if** $(R_{\beta\alpha}^{max} == 0)$ **then**
23: $\quad \mathbf{e}(v_c^s) = \mathbf{e}(v_\alpha^s)$; $\mathbf{e}(v_c^g) = \mathbf{e}(v_\beta^g)$;
24: $\quad$ Computes $\mathbf{e}(v_{i_k}^s) = (E(\sigma_{i_k,1}^{s*}), E(\sigma_{i_k,2}^{s*}), ..., E(\sigma_{i_k,K}^{s*}))$, where $\sigma_{i_k,p}^{s*} = \sigma_{i_k,p}^s + w_k^s \cdot (1 - \sigma_{i_k,p}^s)$, $1 \le p \le K$, for all $1 \le k \le M$;
25: $\quad$ Computes $\mathbf{e}(v_{j_k}^g) = (E(\sigma_{j_k,1}^{g*}), E(\sigma_{j_k,2}^{g*}), ..., E(\sigma_{j_k,K}^{g*}))$, where $\sigma_{j_k,p}^{g*} = \sigma_{j_k,p}^g - w_k^g \cdot \sigma_{j_k,p}^g$, $1 \le p \le K$, for all $1 \le k \le H$;
26: $\quad$ Goto **Step AE3**;
27: **end if**

Step AE6: Auction Opening:

28: AE gets v_c^s and v_c^g by asking AA to decrypt $\mathbf{e}(v_c^s)$ and $\mathbf{e}(v_c^g)$;
29: $\mathbb{W}^s = \{s_{i_k} | w_k^s = 1, \forall 1 \le k \le M\}$;
30: $\mathbb{W}^g = \{\mathbb{G}_{i_k} | w_k^g = 1, \forall 1 \le k \le H\}$;

the winner sets $\mathbb{W}^s$ and $\mathbb{W}^g$ from $\mathbf{w}^s$ and $\mathbf{w}^g$ using the randomization permutations in Step AE1.

Note that in Line 13 and 19 in Protocol 6, the need of "adding one" is caused by different ways of indexing, i.e., α and β returned by Algorithms 4 or 5 are starting from 0, while the indexes of sellers and buyer groups are from 1.

3) Pricing: Each spectrum channel is sold from the winning sellers at the selling clearing price v_c^s, and bought by the winning buyer groups at the buying clearing price v_c^g. Each winner buyer in winning buyer group $\mathbb{G}_k$ pays the equal share of the buying clearing price, that is v_c^g/n_k, where $n_k = |\mathbb{G}_k|$.

From the description above, we can see that PS-TRUST exactly follows the auction procedure of TRUST. Therefore, PS-TRUST maintains the properties of economic-robustness and spectrum reuse of TRUST, in the presence of semi-honest adversaries.

C. Security Analysis

In the sense of cryptography, the standard definition of security against semi-honest adversaries can be described as follows [4].

Definition 2 (Security against Semi-honest Adversaries)**.** *Let $f(x, y)$ be a functionality with two inputs x and y, and two outputs $f^A(x, y)$ and $f^B(x, y)$. Suppose that protocol Π computes functionality $f(x, y)$ between two parties Alice and Bob. Let $V_A^\Pi(x, y)$ (resp. $V_B^\Pi(x, y)$) represent Alice's (resp. Bob's) view during an execution of Π on (x, y). In other words, if $(x, \boldsymbol{r}_A^\Pi)$ (resp. $(y, \boldsymbol{r}_B^\Pi)$) denotes Alice's (resp. Bob's) input and randomness, then*

$$V_A^\Pi(x, y) = (x, \boldsymbol{r}_A^\Pi, m_1, m_2, ..., m_t), \text{ and}$$
$$V_B^\Pi(x, y) = (y, \boldsymbol{r}_B^\Pi, m_1, m_2, ..., m_t)$$

where $\{m_i\}$ denote the messages passed between the parties. Let O_A^Π (resp. O_B^Π) denote Alice's (resp. Bob's) output after an execution of Π on (x, y), and $O^\Pi(x, y) = (O_A^\Pi(x, y), O_B^\Pi(x, y))$. Then we say that protocol Π is secure (or protects privacy) against semi-honest adversaries if there exist probabilistic polynomial time (PPT) simulators S_1 and S_2 such that

$$\{(S_1(x, f_A(x, y)), f(x, y))\} \stackrel{c}{\equiv} \{(V_A^\Pi(x, y), O^\Pi(x, y))\} \quad (6)$$

$$\{(S_2(x, f_B(x, y)), f(x, y))\} \stackrel{c}{\equiv} \{(V_B^\Pi(x, y), O^\Pi(x, y))\} \quad (7)$$

where $\stackrel{c}{\equiv}$ denotes computational indistinguishability.

With the above security definition, we now prove the basic lemma that will allow us to argue that our auction solution is secure against semi-honest adversaries. Lemma 1 is similar to Lemma 1 in [22], with slight difference and some extension.

Lemma 1. *Suppose that Alice has run the key generation algorithm for semantically secure homomorphic public-key encryption scheme, and has given her public key to Bob. Suppose also that Alice and Bob run Protocol X, for which all messages passed from Alice to Bob are encrypted using this scheme, or only carry information that can be completely inferred from the output of Bob, and all messages passed from Bob to Alice are uniformly distributed in their value ranges and independent of Bob's inputs, or only carry information that can be completely inferred from the output of Alice. Then Protocol X is secure against semi-honest adversaries.*

Proof: We prove the security of Protocol X in two separate cases, depending on which party the adversary has corrupted. To prove security, we show that for all PPT adversaries, the adversary's view based on Alice and Bob's interaction is indistinguishable to the adversary's view when the corrupted party interacts with a simulator instead. In other words, we show that there exist simulators S_1 and S_2 that satisfy conditions (6) and (7).

Case 1: Bob is corrupted. We simulate Alice's messages sent to Bob. For each encrypted message that Alice is supposed to send to Bob, we let the simulator S_2 pick a random element from $\mathbb{Z}_n$, and send an encryption of this. Any adversary who can distinguish between interaction with Alice versus interaction with S_2 can be used to break the security assumptions of the used encryption scheme. Thus, no such PPT adversary exists. For each (plain) message that only carries information that can be completely inferred from the output of Bob, the simulator S_2 can of course simulate it using Bob's output of the functionality ($f_B(x, y)$ in equation (7)).Thus, condition (7) holds.

Case 2: Alice is corrupted. We simulate Bob's messages sent to Alice. For each message that is uniformly distributed in its value range and independent of Bob's inputs, simulator S_1 picks a random element from its range and sends to Alice. Again, equation (6) holds due to the fact that Alice cannot distinguish the simulator's random element from the correct element that has been randomized by Bob over its value range. For each message that only carries information that can be completely inferred from the output of Alice, the simulator S_1 can simulate it using Alice's output of the functionality ($f_A(x, y)$ in equation (6)). Thus, condition (6) holds.

Thus, we can conclude that Protocol X is secure against semi-honest adversaries. □

Theorem 1. *Protocol 1 is secure against semi-honest adversaries.*

Proof: It is obvious that all messages passed from AR to KH are uniformly distributed in the ciphertext space $\mathbb{Z}_{N^2}$ (or the values obtained by decrypting the messages are uniformly distributed in the plaintext space $\mathbb{Z}_N$), and the messages passed from KH to AR are encrypted. According to Lemma 1, Protocol 1 is secure against semi-honest adversaries. □

Theorem 2. *Suppose that the auction agent has run the key generation algorithm for semantically secure homomorphic public-key encryption scheme, and has given its public key to the auctioneer. Further suppose that the auctioneer runs Algorithm X (where X is one of 2, 3, 4, 5), and holds the computation result. Then the resulting protocol is secure against semi-honest adversaries.*

Proof: The resulting protocol has no messages exchanged,

except sequentially calling Protocol 1 which is secure against semi-honest adversaries, so due to Lemma 1 and sequential composition theory [5], it is secure against semi-honest adversaries.□

Theorem 3. *Protocol 6 is secure against semi-honest adversaries.*

Proof: We show that all the messages exchanged between the parties satisfy the conditions of Lemma 1. Then, applying Lemma 1 and the sequential composition theory [5], Protocol 6 is secure against semi-honest adversaries.

Specifically, suppose that there are Q winner-candidate pairs (including the critical seller and buyer group), we can list all the messages exchanged between the parties as follows.

Messages sent from AE to AA include:

$$\{E(\alpha_i)\}_1^{Q+1}, \{E(\beta_i)\}_1^{Q+1}, \{E(R^{max}_{\beta_i\alpha_i})\}_1^{Q+1}, \mathbf{e}(v_c^s), \mathbf{e}(v_c^g)$$

Message sent from AA to AE include:

$$\{E(\mathbf{w}_i^s)\}_1^{Q}, \{E(\mathbf{w}_i^g)\}_1^{Q}, \mathbf{w}^s_{Q+1}, \mathbf{w}^g_{Q+1}, \{R^{max}_{\beta_i\alpha_i}\}_1^{Q+1}, v_c^s, v_c^g$$

Now we show that all these messages satisfy the conditions of Lemma 1. First, among the messages sent from AE to AA, α_i and β_i (obtained by decrypting $E(\alpha_i)$ and $E(\beta_i)$) are uniformly distributed over their value ranges (i.e. $[1..M]$ and $[1..H]$) due to the random permutations unknown to AA, and messages $R^{max}_{\beta_i\alpha_i}$, v_c^s and v_c^g can be completely inferred from the output of AA, which is also the auction result including selling and buying clearing prices v_c^s, v_c^g, and the winner sets $\mathbb{W}^s$ and $\mathbb{G}^g$. Second, among the messages sent from AA to AE, messages $E(\mathbf{w}_i^s)$ and $E(\mathbf{w}_i^g)$ are encrypted, and messages $\mathbf{w}^s_{Q+1}$, $\mathbf{w}^g_{Q+1}$, $R^{max}_{\beta_i\alpha_i}$, v_c^s and v_c^g can be completely determined by the output of the auctioneer, which is also the auction result. As a result, all the messages in Protocol 6 satisfy the conditions of Lemma 1.

Furthermore, according to Theorem 2, subprotocols resulted from running Algorithms 4 and 5 (i.e. calling MultiBidMin(.), MultiBidMax(.) and TwoBidMax(.,.)) are secure against semi-honest adversaries. Then, applying Lemma 1 and sequential composition theory, we can conclude that Protocol 6 is secure against semi-honest adversaries. □

Now, we can conclude PS-TRUST is secure against semi-honest adversaries.

Theorem 4. *PS-TRUST is a two-party protocol secure against semi-honest adversaries, between the auctioneer and the auction agent. Additionally, anyone (i.e. the auctioneer, auction agent, and each bidder) cannot know anything about the bids beyond the auction result through the auction.*

The proof is obvious based on the previous theorems, and we only sketch it here. Note that in the auction, we implicitly assume that the bidders' bids are the only privacy needed to protect. So, steps of Buyer Group Formation and Pricing of the auctions are unrelated to the security. That is, we only need to prove that the step of Secure Auction Computation is secure. By Theorem 3, the winner determination procedure is secure, and we can similarly prove the security of buyer group bidding procedure. Thus, PS-TRUST is secure against semi-honest adversaries. What is more, because bidders' bids are encrypted in EBV form, and are input to the auctioneer, according to the definition of security, neither the auctioneer nor the auction agent knows anything about the bids, and no bidder knows anything about other bidders' bids, except the auction result.

TABLE I
COMPUTATION AND COMMUNICATION COMPLEXITIES

Protocol/Algorithm	1	2	3	4	5
Complexity	$O(1)$	$O(K)$	$O(K^2)$	$O(K)$	$O(nK)$

VI. PERFORMANCE ANALYSIS AND EVALUATION

As PS-TRUST exactly follows the procedure of TRUST, the auction efficiency is the same as that of TRUST. So, we only focus on the analysis and evaluation of computation and communication overhead caused by the security measures.

A. Performance Analysis

The analysis of computation and communication complexities for Protocols/Algorithms from 1 to 5 is straightforward and the results are listed in Tab. I. We thus can find the computation complexity of Protocol 6 (which is also the computation complexity PS-TRUST) is $O((M+N)\cdot K\cdot W)$ operations (e.g. addition or multiplication) of big integers, where W represents the number of seller-buyer-group winner pairs. Similarly, we can find the communication complexity of PS-TRUST is $O((M+N)\cdot K\cdot W)$ times of bit length of big integers. Note that, practical running time and message volume will be impacted by the bit length used in the homomorphic encryption scheme.

B. Performance Evaluation

We implement PS-TRUST using Java in Windows XP with Intel's Core 2 Duo CPU 2.93GHz. We let the buyers be randomly distributed in an area of $100\text{m}\times 100\text{m}$, let the protection distance be 50m, and let default experimental setting be as follows: the bit length of homomorphic encryption scheme is 512, i.e., n's bit length is 512; the bit length K of EBV is 8; the numbers (M, N) of sellers and buyers are $(10, 30)$. All experimental results are averaged on 10 random repetitions.

Fig. 3 shows the curves of running times and message volumes of PS-TRUST as (M, N) vary from $(10, 30)$ to $(30, 70)$. Both performance measures grow slightly faster than linear growth according to $(M+N)$. This is because according to the theoretical results, these measures also depend on W, which increases as well with $(M+N)$ on average.

Fig. 4 show the curves of running times and message volumes of PS-TRUST as K vary from 8 to 24. We can see that all the curves are roughly linear to K. This is consistent with the theoretical results fairly well.

From the analytical and experimental results above, we can see that both running times and message volumes are feasible for practical applications. Furthermore, the running time of the

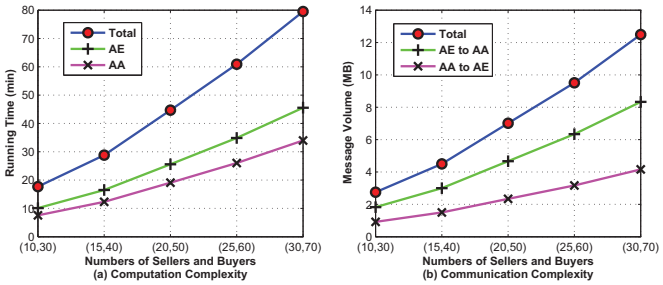

Fig. 3. Overhead Evaluation as the Numbers of Sellers and Buyers Vary

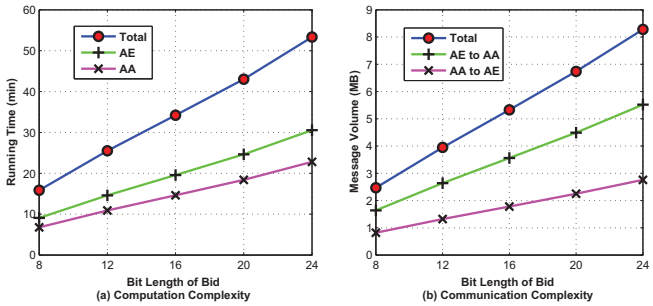

Fig. 4. Overhead Evaluation as the Bit Length of Bids Varies

auctioneer (AE) is about a third more than that of the auction agent (AA), and the message volume of AE is about twice of that of the AA. Finally, the running times can be reduced by parallel computing if needed.

VII. Conclusion

In this paper, we have proposed PS-TRUST, the first provably secure solution to truthful double spectrum auctions. Previous studies on secure spectrum auctions did not provide adequate security, as they revealed information about the bids beyond the auction result. Different from those studies, we have achieved security in the sense of cryptography in this work. Specifically, PS-TRUST reveals nothing about the bids to any participant, except the auction result including clearing prices and winner sets. We have also proved formally the security of PS-TRUST in the presence of semi-honest adversaries. Finally, we have implemented PS-TRUST in Java, and have theoretically and experimentally shown that the computation and communication overhead of PS-TRUST is modest, and its practical applications are feasible.

Acknowledgment

The work is supported in part by the National Science Foundation of China (NSFC) (No. 61202407 & U1301256 & 61202028 & 61303206 & 61170058 & 61228207), Fundamental Research Funds for the Central Universities (No. WK0110000033), National Science and Technology Major Project (No. 2011ZX03005-004-04 & 2012ZX03005009), Special Project on IoT of China NDRC (2012-2766), and the Guangdong Province and CAS Comprehensive Strategic Cooperation Projects (No. 2012B090400013).

References

[1] X. Zhou, S. Gandhi, S. Suri, and H. Zheng. ebay in the sky: Strategyproof wireless spectrum auctions. Proc. of MobiCom08, pp. 2-13, 2008.
[2] X. Zhou and H. Zheng. Trust: A general framework for truthful double spectrum auctions. Proc. of INFOCOM09, pp.999-1007, 2009.
[3] F McSherry, K Talwar. Mechanism Design via Differential Privacy. Proc. of FOCS 2007, pp. 94-103, 2007.
[4] O. Goldreich. Foundations of Cryptography: Volume 2 - Basic Applications. Cambridge University Press, 2004.
[5] C. Hazay, Y. Lindell. Efficient secure two-party protocols: Techniques and constructions. Springer, 2010.
[6] RP MCAFEE. A dominant strategy double auction. Journal of Economic Theory 56, 2 (April 1992), 434-450.
[7] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. Proc. EUROCRYPT 99, LNCS 1592: 223-238, 1999.
[8] M. Naor, B. Pinkas, and R. Sumner, Privacy preserving auctions and mechanism design, in EC99, Oct. 1999.
[9] K. Suzuki, M. Yokoo. Secure generalized vickrey auction using homomorphic encryption. In Proceedings of the financial cryptography conference, FC 2003. Guadeloupe, French West Indies.
[10] M. Yokoo, K. Suzuki. Secure generalized vickrey auction without third-party servers. Proc. of FC'04, 2004.
[11] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan. Robust, privacy protecting and publicly verifiable sealed-bid auction. Proc. of ICICS'02, Dec. 2002.
[12] F. Brandt and T. Sandholm. On the existence of unconditionally privacy-preserving auction protocols. ACM Transactions on Information and System Security, 11(2): 1-21, May 2008.
[13] Spectrum Bridge, Inc., http://www.spectrumbridge.com.
[14] L.B. Deek, X. Zhou, K.C. Almeroth, and H. Zheng. To preempt or not: Tackling bid and time-based cheating in online spectrum auctions. Proc. of INFOCOM'11, Apr. 2011.
[15] M. Al-Ayyoub and H. Gupta. Truthful spectrum auctions with approximate revenue. Proc. of INFOCOM'11, Apr. 2011.
[16] P. Xu, X.Y. Li, S. Tang, and J. Zhao. Efficient and strategyproof spectrum allocations in multichannel wireless networks. IEEE Transactions on Computers, 60(4): 580-593, Apr. 2011.
[17] P. Xu, X. Xu, S. Tang, and X.Y. Li. Truthful online spectrum allocation and auction in multi-channel wireless networks. Proc. of INFOCOM'11, Apr. 2011.
[18] X. Feng, Y. Chen, J. Zhang, Q. Zhang, and B. Li. TAHES: Truthful double auction for heterogeneous spectrums. Proc. of INFOCOM'12, Mar. 2012.
[19] M. Dong, G. Sun, X. Wang, and Q. Zhang. Combinatorial auction with time-frequency flexibility in cognitive radio networks. Proc. of INFOCOM'12, Mar. 2012.
[20] M. Pan, J. Sun, Y. Fang. Purging the Back-Room Dealing: Secure Spectrum Auction Leveraging Paillier Cryptosystem. IEEE Journal on Selected Areas in Communications, 29(4): 866-876, 2011.
[21] Q. Huang, Y. Tao, and F. Wu. SPRING: A Strategy-Proof and Privacy Preserving Spectrum Auction Mechanism. Proc. of INFOCOM'13, 2013.
[22] P. Bunn, R. Ostrovsky. Secure Two-Party k-Means Clustering. Proc. of CCS'07.