ITSEC: An Information-theoretically Secure Framework for Truthful Spectrum Auctions

Zhili Chen[1,2], Liusheng Huang[1,2], Lin Chen[3]
[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
[2]Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou, China
[3]Lab. Recherche Informatique (LRI-CNRS UMR 8623), Univ. Paris-Sud, 91405 Orsay, France
Email: {zlchen3, lshuang}@ustc.edu.cn, chen@lri.fr

***Abstract*—Truthful auctions make bidders reveal their true valuations for goods to maximize their utilities. Currently, almost all spectrum auction designs are required to be truthful. However, disclosure of one's true value causes numerous security vulnerabilities. Secure spectrum auctions are thus called for to address such information leakage. Previous secure auctions either did not achieve enough security, or were very slow due to heavy computation and communication overhead. In this paper, inspired by the idea of secret sharing, we design an *information-theoretically secure framework* (ITSEC) for truthful spectrum auctions. As a distinguished feature, ITSEC not only achieves information-theoretic security for spectrum auction protocols in the sense of cryptography, but also greatly reduces both computation and communication overhead by ensuring security without using any encryption/description algorithm. To our knowledge, ITSEC is the first information-theoretically secure framework for truthful spectrum auctions in the presence of semi-honest adversaries. We also design and implement circuits for both single-sided and double spectrum auctions under the ITSEC framework. Extensive experimental results demonstrate that ITSEC achieves comparable performance in terms of computation with respect to spectrum auction mechanisms without any security measure, and incurs only limited communication overhead.**

I. Introduction

It is nowadays widely acknowledged that radio spectrum has become a precious resource, and the problem of spectrum scarcity is far more pressing than ever. Currently, the main causes of spectrum scarcity are two-fold: the increasing demand and the under-utilization. On one hand, more and more wireless devices and services are starving for new spectrum to use, making the available unlicensed spectrum a scarce resource. On the other hand, large bulks of licensed spectrum are inefficiently utilized by their current owners. To address the spectrum scarcity, a widely deployed solution is the spectrum redistribution which provides incentives to licensed users to trade their unused spectrum with unlicensed users. Recently, open markets for spectrum redistribution, such as Spectrum Bridge [21], have begun to provide services for buying, selling, and leasing idle spectrum channels.

Spectrum auction is a fair and efficient mechanism for spectrum redistribution. As an important research strand, *truthful spectrum auction* design has attracted extensive studies, resulting a number of truthful spectrum auction solutions making bidders reveal their true spectrum valuations. However, revealing one's true value opens the door for many security vulnerabilities. An easily mountable attack is the frauds of the insincere auctioneer [28], in which a dishonest auctioneer falsifies the auction result to get more revenue based on its knowledge on the true valuation of bidders. More sophisticatedly, the dishonest auctioneer may create a bid-rigging with a bidder [28] by colluding with him to falsify the auction result, and shares the extra revenue with him. Thirdly, a malicious auctioneer can further exploit bid information for future auctions, as historical data can be used to evaluate the willingness to pay [15]. In such context, protecting the privacy of bidders' bid information becomes a pressing concern in the design of attack-resilient truthful spectrum auctions.

There exist a number of privacy preserving auction mechanisms for traditional goods (c.f. [15][18][16][17]). However, radio spectrum has its unique feature compared to traditional goods as it can be reused in both spatial and time domains. Consequently, direct application of traditional privacy preserving auction mechanisms to spectrum auctions cannot support spectrum reuse, leading to significant performance degradation. Recently, a handful of solutions for privacy preserving spectrum auctions have also been proposed [28][29][3]. However, they either fell short of providing adequate security, or suffered severe performance overhead. We would like to put the emphasis on *cryptographical security*, where a protocol is said to be secure if no participating party can learn any information beyond the output of the protocol. Using this formal security criteria, the approaches in [28][29] indeed reveal certain information that cannot be inferred from the output [3]. The scheme in [3], on the other hand, is secure, but suffers heavy computation and communication overhead. Furthermore, it only achieves computational security against probabilistic polynomial time (PPT) adversaries.

Motivated by the above arguments, this paper proposes ITSEC, an information-theoretically secure framework for truthful spectrum auctions, which ensures the privacy of bidders' bid information against *any adversary with arbitrary computation power*. Moreover, ITSEC brings almost no extra computation overhead to the underlining spectrum auction and incurs only limited communication overhead. The auction framework of ITSEC is shown in Fig. 1. ITSEC introduces two separate entities, a seller agent and a buyer agent, to cooperate with the auctioneer to securely run the auction. Note that none of the three parties need to be a trusted party, but any two of

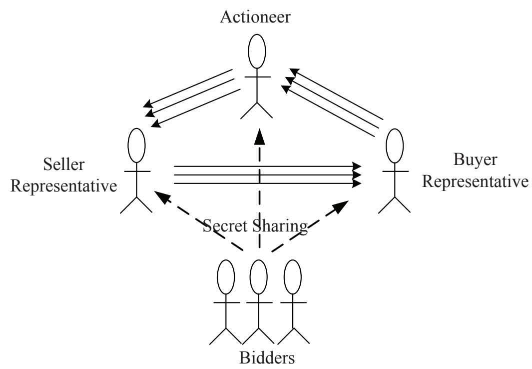


Fig. 1. Spectrum Auction Framework for ITSEC

them are assumed not to collude. As a distinguished feature, *ITSEC reveals nothing about the bids to any adversary with unbounded computation power*, except for the auction result. The main contributions are articulated as follows.

- *Cryptographically secure spectrum auction design:* We design information-theoretically secure framework ITSEC for truthful spectrum auctions based on secret sharing without using any cryptographical components such as encryption and decryption. We formally prove that ITSEC is cryptographically secure.
- *Circuit-level implementation of ITSEC:* We design circuits for both single-sided and double spectrum auction mechanisms implementing ITSEC, and optimize the circuits using "XOR-free" property to further improve performance.
- *Experimental evaluation of ITSEC:* We implement both single-sided and double auction mechanisms using the designed circuits under ITSEC framework, and carry out extensive experiments to evaluate their performance. Our results demonstrate that ITSEC achieves comparable performance in terms of computation with respect to spectrum auction mechanisms without any security measure, and incurs only limited communication overhead.

The remainder of this paper is organized as follows. Section II briefly reviews related work. Section III provides technical preliminaries. We present ITSEC framework, and formally prove its information-theoretic security in Section IV. In Section V, we design circuits for both single-sided and double spectrum auction mechanisms. In Section VI, we implement the designed circuits under ITSEC framework, and evaluate the performance in terms of computation and communication overhead. Finally, the paper is concluded in Section VII.

II. Related Work

Truthful spectrum auctions. There have been extensive studies on truthful spectrum auctions in recent literature [11][12][23][24][22][25][26][27]. However, most of them are focused on system modeling and mechanism design, leaving the security issue unaddressed. Specifically, Zhou *et al.* [11] proposed a single-sided truthful spectrum auction mechanism VERITAS supporting diverse bidding formats. The same authors [12] also proposed the first truthful double spectrum auction framework TRUST with spatial spectrum reuse. Xu *et al.* [23][24] designed an efficient online spectrum allocation in multi-channel wireless networks. Al-Ayyoub and Gupta [22] brought forward a polynomial-time truthful spectrum auction mechanism with revenue guarantee. Feng *et al.* [25] took into account spectrum heterogeneity in truthful double spectrum auctions. Lin *et al.* [26] proposed TASG, a three-stage auction framework for Spectrum Group-buying to enable group-buying behaviors among secondary users. Yang *et al.* [27] designed a framework for spectrum double auctions called PROMISE by jointly considering spectrum reusability, truthfulness, and profit maximization. All these studies did not consider the security aspect in the auctions and are hence vulnerable to various malicious attacks enumerated in Introduction.

Secure auctions with traditional goods. There is also a large body of work on privacy preserving auction design in the past decade. Specifically, Brandt and Sandholm [19] investigated unconditional full privacy in sealed-bid auctions. The authors of [15][18][16][17] applied a variety of cryptography techniques to achieve security in various auction schemes. Unfortunately, when directly applied to spectrum auctions, these privacy preserving schemes for traditional auctions either require exponential complexity to be implemented, or cannot support spectrum reuse.

Secure spectrum auctions. Recently, the authors of [28] and [29] proposed privacy preserving schemes for truthful spectrum auctions, but they fell short in providing cryptographical security [3]. The author of [3] proposed a provably secure solution PS-TRUST for double spectrum auctions, and achieved security against semi-honest adversaries. However, PS-TRUST only achieved computational security against probabilistic polynomial time (PPT) adversaries, and incurred heavy computation and communication overhead.

III. Technical Preliminaries

In this section, we introduce technical preliminaries for the design of ITSEC framework.

A. Secret Sharing

Secret sharing was proposed in 1979 independently by Adi Shamir [4] and Bob Blakley [5], respectively. It is a fundamental cryptographic primitive providing an elegant way of dispersing a secret s into several pieces of data called shares.

Definition 1 ((t,n)-threshold secret sharing scheme)**.** *In a secret sharing scheme, a secret s is dispersed into n shares in the way that any k shares with $k \leq t$ give no information on s (called t-privacy), whereas any k shares with $k \geq t+1$ uniquely disclose s (called $(t+1)$-reconstruction), where t, n are integers with $0 \leq t < n$.*

In (t,n)-threshold secret sharing scheme, the adversary obtains nothing on the secret if he obtains at most t shares. One of the widely used (t,n)-threshold secret sharing schemes is Shamir's scheme, summarised in the following definition.

Definition 2 (Shamir's (t,n)-threshold scheme)**.** *Let p be a prime larger than n. Let t, n be integers with $0 \leq t < n$. Let*

$\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{F}_p$ be pairwise distinct and non-zero. Note that t, n, q, $\alpha_1, \alpha_2, ..., \alpha_n$ are public data. Denote by $s \in \mathbb{F}_p$ the secret. Shamir's scheme selects a polynomial $f(x) \in \mathbb{F}_p[X]$ uniformly at random, conditioned on $deg(f) \leq t$ and $f(0) = s$. The n shares in the secret s are then given as follows:

$$s_i = f(\alpha_i) \in \mathbb{F}_p, \text{ for } 1 \leq i \leq n.$$

It is proved that Shamir's scheme satisfies t-privacy and $(t+1)$-reconstruction with $t+1 \leq n$.

B. Statistical Indistinguishability

This subsection introduces statistical indistinguishability. To streamline the presentation, we first introduce the family of random variables and negligible function [6].

Definition 3 (Family of random variables)**.** *A family of random variables is a function X mapped from non-negative integers to random variables. That is, for each $\kappa \in \mathbb{N}$, $X(\kappa)$ is a random variable. A family of random variables can be denoted by*

$$X = \{X(\kappa)\}_{\kappa \in \mathbb{N}}.$$

Definition 4 (Negligible function)**.** *We say that a function $\sigma : \mathbb{N} \rightarrow [0, 1]$ is negligible in κ if for all $c \in \mathbb{N}$ there exists $\kappa_c \in \mathbb{N}$ such that $\sigma(\kappa) \leq \kappa^{-c}$ for all $\kappa \geq \kappa_c$.*

We now introduce statistical indistinguishability.

Definition 5 (Statistical indistinguishability)**.** *We say that two families of random variables X_0 and X_1 are statistically indistinguishable if distinguishing function $\sigma(X_0(\kappa), X_1(\kappa))$* [1] *is negligible in κ, denoted as $X_0 \overset{stat}{\equiv} X_1$.Particularly, we say that X_0 and X_1 are perfectly indistinguishable if distinguishing function $\sigma(X_0(\kappa), X_1(\kappa)) = 0$ for all κ, denoted as $X_0 \overset{perf}{\equiv} X_1$.*

An important engineering implication of the above definition is that two families of random variables statistically or perfectly indistinguishable from each other cannot be distinguished with arbitrary computation power because they lead to the same output with essentially the same probability.

C. Information-theoretic Security Formulation

In cryptography, the standard security formulation for protocols is known as ideal/real simulation paradigm [13] [14], as shown in Fig. 2. This security formulation assumes the existence of an "ideal world", in which there is an external trusted (and incorruptible) party willing to help the parties carry out their computation. Functionality calling in the "ideal world" is defined such that the parties simply send their inputs to the trusted party, who computes the desired functionality and passes each party its prescribed output.

However, in practice, there is no external trusted party, and protocol execution requires that the parties have to run the protocol among themselves without any help. This standard security formulation says that a protocol is *secure* if its protocol execution in the real world is indistinguishable from its functionality calling in the "ideal world". In our context, we say that a protocol is *information-theoretically secure* if its protocol execution in the real world is statistically indistinguishable from its functionality calling in the "ideal world" for any adversary. That is, no adversary with arbitrary computation power can do more harm in its protocol execution than in its functionality calling.

[1] $\sigma(X_0(\kappa), X_1(\kappa)) \overset{def}{=} \max_A |Pr[A(X_0(\kappa)) = 0] - Pr[A(X_1(\kappa)) = 0]|$, where A is any algorithm outputing a bit $c \in \{0, 1\}$.

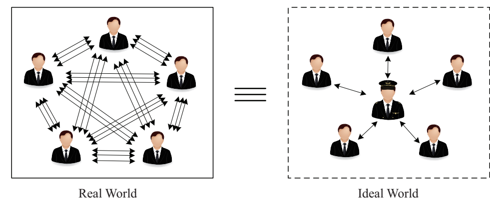


Fig. 2. The Security Formulation of Ideal/Real Simulation Paradigm [3]

IV. ITSEC Framework

In this section, we propose an information-theoretically secure framework ITSEC for truthful spectrum auctions.

A. Spectrum Auction Framework

In spectrum auctions, there are usually an auctioneer, one or more sellers and a number of buyers. In our framework, we designate three parties, namely the auctioneer, the seller agent, and the buyer agent, to cooperatively run the spectrum auctions. Note that none of the three parties need to be a trusted party, but we do require that any two of them do not collude, and have an authenticated secure communication channel. The three parties are formally defined as follows:

- **Auctioneer (AE)**: AE is an intermediary party trusted by both seller and buyer agents and not collude with any one of them.
- **Seller Agent (SA)**: SA is a party representing all sellers.
- **Buyer Agent (BA)**: BA is a party representing all buyers.

Note that in the spectrum auction framework, the reason why we need three parties to compute the auction is due to the theoretical results from secure multi-party computation (SMC) area [14], which state that at least three parties are needed to achieve information-theoretical security. In practice, similar auction framework has been applied to trade sugar beet quotas in Denmark [20].

In our spectrum auctions, we divide the computations into two categories: bid-independent and bid-dependent computations. Bid-independent computations do not reveal bid information and can be performed by AE alone. Bid-dependent computations, on the other hand, should be performed cooperatively among the three parties to achieve information-theoretic security.

Specifically, to perform bid-dependent computations, all bidders (sellers and buyers for double spectrum auctions, buyers for single-sided spectrum auctions) disperse their bids into three shares, and send one share to each of them. The parties then cooperatively perform bid-dependent computations. At the end, each party holds one share of the output of the spectrum auction. The three parties then cooperate to reconstruct the auction results. The spectrum auction framework in ITSEC is shown in Fig. 1 and presented in detail in the following subsection.

B. Secure Tri-party Computation Scheme

In order to perform bid-dependent computations securely, we need to secure the tri-party computation (S3C). A natural idea is to use a general scheme of secure multi-party computation (SMC) such as Protocol CEPS from [6] by setting the number of parties to be three. However, direct application of Protocol CEPS incurs heavy computation and communication overhead, as analysed as follows:

- In terms of computation, from [6], we can see that Protocol CEPS uses Shamir's secret sharing scheme, and takes as input an arithmetic circuit. To compute a boolean circuit (any feasible function, and thus all bid-dependent computations, can be represented by a polynomial-size boolean circuit), we need to simulate the operations of boolean values, such as XOR and AND gates, with operations in field, such as add and multiplication.
- In terms of communication, in Protocol CEPS, each party needs to send and receive messages from all other parties, so the communication topology is a complete graph, meaning significant protocol overhead.

In our solution ITSEC, we design an easily implementable and more efficient S3C scheme in terms of both communication and computation from scratch. Our S3C scheme is composed of three components, secret sharing, gate computation and bit-sharing circuit.

Secret Sharing

In ITSEC framework, we design a secret sharing scheme called bit-xor (BTX) secret sharing, which serves as a building block of the S3C scheme.

Definition 6 (Bit-xor Secret Sharing). *BTX secret sharing scheme disperses a secret bit σ into n shares as follows:*

- *The first $(n-1)$ bits σ_1, σ_2, $\cdots$, σ_{n-1} are randomly chosen;*
- *The last bit is $\sigma_n = \sigma_1 \oplus \sigma_2 \oplus \cdots \oplus \sigma_{n-1} \oplus \sigma$.*

For notational convenience, we define $[\sigma; n] = (\sigma_1, \sigma_2, \cdots, \sigma_n)$, and denote the share σ_i by $[\sigma; n]_i$.

Obviously, BTX is an $(n-1, n)$-threshold secret sharing scheme. To see this, when $(n-1)$ shares are obtained by adversaries, no information about the secret bit σ is leaked. When n shares are obtained, the secret bit is completely revealed. We further generalise the BTX secret sharing scheme to bitwise-xor (BWX) secret sharing scheme, which is also an $(n-1, n)$-threshold secret sharing scheme, as follows.

Definition 7 (Bitwise-xor Secret Sharing). *BWX secret sharing scheme disperses a secret bit vector $x \in \{0,1\}^K$ into n shares as follows:*

- *The first $(n-1)$ shares x_1, x_2, $\cdots$, x_{n-1} are randomly chosen in $\{0,1\}^K$;*
- *The last share is $x_n \in \{0,1\}^K$ with $\sigma_i(x_n) = \sigma_i(x_1) \oplus \sigma_i(x_2) \oplus ... \oplus \sigma_i(x_{n-1}) \oplus \sigma_i(x)$ and $1 \le i \le K$, where $\sigma_i(v)$ denotes the i^{th} bit of v.*

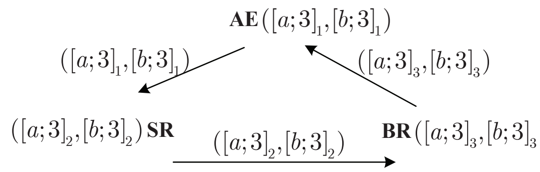


Fig. 3. Communication Pattern for AND Computation

For notational convenience, we define $[x; n] = (x_1, x_2, \cdots, x_n)$, and denote the share x_i by $[x; n]_i$.

Gate Computation

We now focus on how to use the BTX and BWX secret sharing schemes to construct an efficient S3C. Specifically, we demonstrate how to compute the XOR and AND of two bits a and b when each bit is shared among the three parties using the BTX secret sharing scheme. Suppose that $a = a_1 \oplus a_2 \oplus a_3$ and $b = b_1 \oplus b_2 \oplus b_3$, and the auctioneer (AE) holds a_1 and b_1, the seller agent (SA) holds a_2 and b_2, and the buyer agent (BA) holds a_3 and b_3. Our objective is that after XOR or AND computation, each party holds one share of $[a \oplus b; 3]$ or $[ab; 3]$, respectively.

Concretely, the procedures to calculate the outputs of an XOR gate and an AND gate is as follows:

- **XOR Gate:** To compute the output of XOR gate, noticing
$$\begin{aligned} a \oplus b &= (a_1 \oplus a_2 \oplus a_3) \oplus (b_1 \oplus b_2 \oplus b_3) \\ &= (a_1 \oplus b_1) \oplus (a_2 \oplus b_2) \oplus (a_3 \oplus b_3), \end{aligned}$$
we get
$$\begin{cases} [a \oplus b; 3]_1 = a_1 \oplus b_1 = [a;3]_1 \oplus [b;3]_1, \\ [a \oplus b; 3]_2 = a_2 \oplus b_2 = [a;3]_2 \oplus [b;3]_2, \\ [a \oplus b; 3]_3 = a_3 \oplus b_3 = [a;3]_3 \oplus [b;3]_3. \end{cases} \tag{1}$$
Hence, to compute the XOR of two bits, each party only needs to do XOR on its own shares of a and b in order to get his share of the XOR output.
- **AND Gate**: To compute the output of AND gate, noticing
$$\begin{aligned} ab &= (a_1 \oplus a_2 \oplus a_3)(b_1 \oplus b_2 \oplus b_3) \\ &= a_1b_1 \oplus a_1b_2 \oplus a_1b_3 \oplus a_2b_1 \oplus a_2b_2 \oplus a_2b_3 \\ &\quad \oplus a_3b_1 \oplus a_3b_2 \oplus a_3b_3 \\ &= (a_1b_1 \oplus a_1b_3 \oplus a_3b_1) \oplus (a_2b_2 \oplus a_2b_1 \oplus a_1b_2) \\ &\quad \oplus (a_3b_3 \oplus a_3b_2 \oplus a_2b_3), \end{aligned}$$
we get
$$\begin{cases} [ab;3]_1 = a_1b_1 \oplus a_1b_3 \oplus a_3b_1 \\ \qquad\quad = g([a;3]_1, [b;3]_1, [a;3]_3, [b;3]_3), \\ [ab;3]_2 = a_2b_2 \oplus a_2b_1 \oplus a_1b_2 \\ \qquad\quad = g([a;3]_2, [b;3]_2, [a;3]_1, [b;3]_1), \\ [ab;3]_3 = a_3b_3 \oplus a_3b_2 \oplus a_2b_3 \\ \qquad\quad = g([a;3]_3, [b;3]_3, [a;3]_2, [b;3]_2), \end{cases} \tag{2}$$
where $g(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_1x_4 \oplus x_3x_2$. We can see that in order to compute the AND gate, a limited quantity of communication among the parties are needed, as shown in Fig. 3. Specifically, AE sends its shares $[a,3]_1$ and $[b,3]_1$ to SA; SA sends its shares $[a,3]_2$ and $[b,3]_2$ to BA; BA sends its shares $[a,3]_3$ and $[b,3]_3$ to AE. At the end, each party holds a share of $[ab, 3]$.

Bit-sharing Circuit

From the above analysis on the gate computation component, we can see that when each party holds a share of each input bit, the three parties can compute the output bit of an XOR gate using eq (1) without any communication, and that of an AND gate using eq (2) with limited communications shown in Fig. 3. At the end of the computation of an XOR or AND gate, each party holds a share of the output bit of the gate. By repeating the gate computation, we can compute any circuit composed of XOR and AND gates.

It is well known in computation complexity theory that any function feasible to compute can be specified as a polynomial-size boolean circuit using exclusively AND and XOR gates. Therefore, using the BTX secret sharing scheme, we can compute any feasibly computable function through computing the corresponding boolean circuit gate by gate. We call such boolean circuit used in S3C scheme the bit-sharing circuit (BSC). Compared to garbled circuits [1][2] used in secure two-party computation (S2C), our bit-sharing circuits have the following advantages:

- *Reusability.* Bit-sharing circuits are *reusable*. Garbled circuits cannot be reused, since the reuse of random labels representing input or output bits of gates can lead to information leakage. In contrast, our bit-sharing circuits are by nature reusable thanks to the circuit computation procedure. Any bit-sharing circuit can be implemented as a function with input bits being the input arguments and output the return values.
- *Execution speed.* Our bit-sharing circuits are *faster* than garbled circuits. Garbled circuits are based on symmetric encryption algorithms, so computation of circuits often requires a series of encryption or decryption operations. In contrast, our bit-sharing circuits are based on BTX secret sharing scheme, and its computation requires only simple bit-level logic operations.
- *XOR-free.* Similar to garbled circuits, our bit-sharing circuits are also "*XOR-free*". In our context, XOR-free means that the computation of an XOR gate is communication-free, and thus can be done by each party independently.

C. The Global ITSEC Protocol

Armed with the building blocks introduced in the previous subsection, we now specify the global ITSEC framework. We denote the three parties including the auctioneer (AE), the seller agent (SA) and the buyer agent (BA) by P_1, P_2 and P_3, respectively. The protocol is described in Protocol 1. Compared to other secure spectrum auction schemes, IT-SEC not only incurs limited performance overhead because of the advantages of bit-sharing circuits, but also achieves information-theoretic security as proved in the next subsection.

D. Security Analysis of ITSEC

This subsection conducts a formal security analysis on ITSEC. We apply in our analysis the semi-honest adversary model, which is used in almost all existing literature. Specifically, we assume that each party follows the protocol specification, but attempts to learn information beyond the output. According to theoretical results in secure multi-party computation, when at most one party is corrupted, tri-party computation can achieve information-theoretic security for any functionality.

Protocol 1 Information-theoretically secure framework

The protocol proceeds in three phases: input sharing, computation and output reconstruction.

Input Sharing: Each bidder $d_i (1 \leq i \leq n)$ holds as input its bid value $x_i \in \{0,1\}^K$, and distributes $[x_i; 3]$ to the three parties.

Computation: P_1 (the auctioneer) performs bid-independent computations by itself. Then, P_1 generates a circuit describing bid-dependent computations, and distributes the circuit to other parties, who validate the circuit, respectively. Each party repeats the following operations until all gates are processed: Consider the first gate in the computational order that has not been processed yet. According to the type of gate, do one of the following.

- **XOR gate:** The parties hold $[a; 3]$, $[b; 3]$ for the two input bits a, b to the gate. Each party $P_k (1 \leq k \leq 3)$ computes $[a \oplus b; 3]_k = [a; 3]_k \oplus [b; 3]_k$ using eq (1), independently.
- **AND gate:** The parties hold $[a; 3]$, $[b; 3]$ for the two input bits a, b to the gate.
  - *Communication Step.* Party P_1 sends $[a; 3]_1$, $[b; 3]_1$ to party P_2; party P_2 sends $[a; 3]_2$, $[b; 3]_2$ to party P_3; party P_3 sends $[a; 3]_3$, $[b; 3]_3$ to party P_1.
  - *Computation Step.* Each party P_k computes $[ab; 3]_k$ independently from $[a; 3]_k$, $[b; 3]_k$, $[a; 3]_{(k-2)\%3+1}$ and $[b; 3]_{(k-2)\%3+1}$ using eq (2).

Output Reconstruction: At this point all gates, including the output gates have been processed, and each party holds a share of each output bit. For each output bit σ, the parties reconstruct the bit as follows: First, party P_1 generate a random bit r, and sends $b_1 = [\sigma; 3]_1 \oplus r$, to party P_2; secondly, party P_2 sends $b_2 = [\sigma; 3]_2 \oplus b_1$ to party P_3; thirdly, party P_3 sends $b_3 = [\sigma; 3]_3 \oplus b_2$ to party P_1; finally, party P_1 recovers the bit by calculating $b_3 \oplus r = [\sigma; 3]_3 \oplus [\sigma; 3]_2 \oplus [\sigma; 3]_1 = \sigma$. Once the output reconstructed, P_1 publishes the auction output to each bidder.

To proceed, we first define the information-theoretic security (IT security) against semi-honest adversaries for three-party computation in the sense of cryptography [13].

Definition 8 (IT Security against Semi-honest Adversaries)**.** *Let $f(x, y, z)$ be a deterministic functionality with three inputs x, y and z, and three outputs $f_A(x, y, z)$, $f_B(x, y, z)$ and $f_C(x, y, z)$. Suppose that protocol Σ computes functionality $f(x, y, z)$ among three parties Alice, Bob and Carl. Let $V_A^\Sigma(x, y, z)$, $V_B^\Sigma(x, y, z)$ and $V_C^\Sigma(x, y, z)$ represent Alice's, Bob's and Carl's views during an execution of Σ on (x, y, z). In other words, if $(x, \boldsymbol{r}_A^\Sigma)$, $(y, \boldsymbol{r}_B^\Sigma)$ and $(z, \boldsymbol{r}_C^\Sigma)$ denotes Alice's, Bob's and Carl's inputs and randomness, then*

$$\begin{cases} V_A^\Sigma(x, y, z) = (x, \boldsymbol{r}_A^\Sigma, \boldsymbol{M}_{Ain}, \boldsymbol{M}_{Aout}), \\ V_B^\Sigma(x, y, z) = (y, \boldsymbol{r}_B^\Sigma, \boldsymbol{M}_{Bin}, \boldsymbol{M}_{Bout}), \\ V_C^\Sigma(x, y, z) = (z, \boldsymbol{r}_C^\Sigma, \boldsymbol{M}_{Cin}, \boldsymbol{M}_{Cout}) \end{cases}$$

where $\boldsymbol{M}_{Xin}$ denotes the set of messages received by party X, $\boldsymbol{M}_{Xout}$ denotes the set of messages sent from party X.

We say that protocol Σ is secure (or protects privacy) against semi-honest adversaries if there exist simulators S_1, S_2 and S_3 with arbitrary computation power such that

$$\begin{cases} \{S_1(x, f_A(x, y, z))\}_{x,y,z} \overset{stat}{\equiv} \{V_A^\Sigma(x, y, z)\}_{x,y,z}, \\ \{S_2(y, f_B(x, y, z))\}_{x,y,z} \overset{stat}{\equiv} \{V_B^\Sigma(x, y, z)\}_{x,y,z}, \\ \{S_3(z, f_C(x, y, z))\}_{x,y,z} \overset{stat}{\equiv} \{V_C^\Sigma(x, y, z)\}_{x,y,z}, \end{cases} \tag{3}$$

where $\overset{stat}{\equiv}$ denotes statistical indistinguishability.

The following theorem establishes the IT security of ITSEC.

Theorem 1. *ITSEC (Protocol 1) is information-theoretically secure against semi-honest adversaries.*

Proof We can prove the IT security of Protocol 1 in three separate cases, depending on which party the adversary has corrupted. Since in the protocol, the three parties are symmetric, we only need to prove the case where one of the parties is corrupted.

Specifically, we show that for all adversaries with unbounded computation power, the adversary's view based on the interaction between the corrupted party and any other party is statistically indistinguishable to the adversary's view when the corrupted party interacts with a simulator instead. Mathematically, we show that there exist simulators S_1, S_2 and S_3 that satisfy conditions in eq (3).

Without loss of generality, assume that party $P_k(1 \leq P_k \leq 3)$ is corrupted. For each part of the protocol, we simulate the protocol execution as follows:

- **Input sharing**: Party P_k receives its input $[x_i; 3]_k(1 \leq i \leq n)$ from the bidders, nothing needs to be simulated.
- **Addition**: Party P_k sends or receives nothing, so there is nothing to show.
- **Multiplication**: As for input sharing, we can assume that the two input bits held by party P_k before the current multiplication are $[a; 3]_k$ and $[b; 3]_k$. This is trivially the case for the first multiplication and can be assumed by induction for the following ones. Then party P_k receives $[a; 3]_{(k-2)\%+1}$ and $[b; 3]_{(k-2)\%+1}$ from honest party $P_{(k-2)\%+1}$. According to the BTX secreting sharing scheme, $[a; 3]_k$ and $[a; 3]_{(k-2)\%+1}$ reveal nothing about bit a, so we can simulate $[a; 3]_{(k-2)\%+1}$ as a random bit; similarly, $[b; 3]_k$ and $[b; 3]_{(k-2)\%+1}$ reveal nothing about bit b, so we can also simulate $[b; 3]_{(k-2)\%+1}$ as a random bit.
- **Output Construction**: We simulate this part in two separated cases, depending on whether the corrupted party is the auctioneer or not:
  - *Case 1: $k = 1$, the auctioneer is corrupted.* We simulate each bit received from party P_3 by P_1 as the XOR result of the corresponding output bit and the random bit r used to send a bit to party P_2 in the same round.
  - *Case 2: $k \neq 1$, the auctioneer is not corrupted.* We simulate each bit received from party P_{k-1} as a random bit.

In all cases, we can see that condition (3) holds. We thus conclude that Protocol 1 is IT secure against semi-honest adversaries. □

V. Circuit Design Implementing ITSEC for Secure Spectrum Auctions

Having presented the IT secure framework for truthful spectrum auctions, the remaining challenge to implement an IT secure spectrum protocol is to design a circuit for the spectrum auction mechanism. By designing a circuit, we mean designing an algorithm in a *data-oblivious* fashion so that the execution path does not depend on the input. In this section, we propose circuit design for both single-sided and double spectrum auction mechanisms implementing the ITSEC framework.

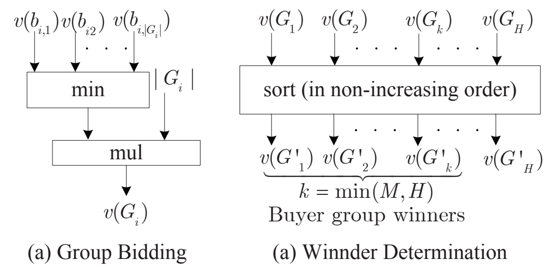


Fig. 4. Main Circuit Components for SPRING

A. Circuit for Single-sided Spectrum Auction Mechanism

Single-sided Spectrum Auction. We start by a sealed-bid single-sided spectrum auction where one auctioneer $\mathcal{A}$ sells M homogenous spectrum channels to a set of buyers $\mathbb{B} = \{b_1, b_2, ..., b_N\}$,each requesting only one channel. Each channel sold can potentially be reused by multiple non-conflict buyers who are separated far enough.

To solve the above single-sided spectrum auction problem, we apply the truthful spectrum auction mechanism proposed in paper [29] termed as SPRING, summarized as follows:

Algorithm 2 Truthful Single-sided Spectrum Auction: SPRING

Input: Bid values, location coordinates of buyers in $\mathbb{B}$, and the number M of channels sold.
Output: Winning buyer groups and the clearing price.

Buyer group formation:

Create a conflict graph of buyers according to their location coordinates, and form non-conflict buyer groups based on the conflict graph independently on bids. Denote the resulted buyer group set by $\mathbb{G} = \{G_1, G_2, ..., G_H\}$.

Winner determination:

for each $i \in [1, H]$ **do**
  $v(G_i) = |G_i| \cdot \min_{b \in G_i} v(b)$; // compute group bid
end for
Sort the buyer group bids in non-increasing order.

$$v(G'_1) \geq v(G'_2) \geq ... \geq v(G'_H)$$

The winning buyer groups are the first $k = \min(M, H)$ buyer groups in the sorted list.

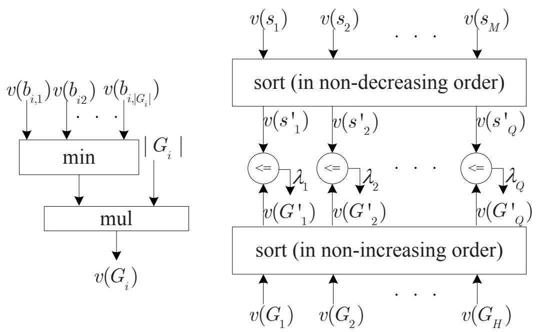
Pricing:

The clearing price is the $(k + 1)^{\text{st}}$ buyer group bid (if existent) or 0 (otherwise), which is evenly shared among the buyers in each winning group.

Circuit Description. From Algorithm 2, we can see that the step "buyer group formation" is bid-independent, and the step "pricing" simply reveals a bid value. So we only need to design a circuit for the step "winner determination" to secure SPRING. Fig. 4 shows the main circuit components. Group bidding can be implemented using circuits of minimum selection and integer multiplication with a constant, and bid sorting using odd-even merge sorting network. We will explain these building-block circuits in detail later in Sec. V-C.

B. Circuit for Double Spectrum Auction Mechanism

Double Spectrum Auction. We now proceed to a sealed-bid double spectrum auction with one auctioneer $\mathcal{A}$, a set of sellers $\mathbb{S} = \{s_1, s_2, \cdots, s_M\}$, and a set of buyers $\mathbb{B} = \{b_1, b_2, \cdots, b_N\}$. Each seller s_i contributes one channel and each buyer b_j requests one. The channels are homogenous



(a) Group Bidding (a) Winnder Determination

Fig. 5. Main Circuit Components for TRUST

to buyers so that the requests are not channel specific. Each channel contributed by sellers can potentially be reused by multiple non-conflict buyers who are separated far enough.

We focus on TRUST [12], a truthful double auction mechanism, as summarized in Algorithm 3.

Algorithm 3 Truthful Double Spectrum Auction: TRUST

Input: Bid values, location coordinates of buyers in $\mathbb{B}$, and bid values of sellers in $\mathbb{S}$.

Output: Winning sellers and buyer groups, and selling and buying clearing prices.

Buyer group formation:

Just as Algorithm 2.

Winner determination:

for each $i \in [1, H]$ **do**

  $v(G_i) = |G_i| \cdot \min_{b \in G_i} v(b)$; // compute group bid

end for

Sort the seller bids and the buyer group bids so that:

$$v(s'_1) \le v(s'_2) \le ... \le v(s'_M)$$
$$v(G'_1) \ge v(G'_2) \ge ... \ge v(G'_H)$$

Find $k = \arg\max\{v(s'_k) \le v(G'_k)\}$, and then the first $(k-1)$ sellers and the first $(k-1)$ buyer groups in the sorted lists are the auction winners.

Pricing:

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

Circuit Description. Performing a similar analysis as in the single-sided case, we can see that we only need to design a circuit for the step "winner determination". The main circuit components for TRUST are shown in Fig. 5.

The circuit design for group bidding and bid sorting is exactly the same as the single-sided spectrum auction case. For the moment, we focus on how the auction winners and the clearing prices are determined in a data-oblivious manner, after both the seller and buyer group bids are sorted, and the Q pairs of seller-buyer-group bids are compared, where $Q = \min(M, H)$, as shown in Fig. 5-(b).

First, we show that direct revelation of the comparison results λ_i $(1 \le i \le Q)$ and the corresponding winner IDs may lead to information leakage. To illustrate this, consider the following example:

$$\begin{pmatrix} \text{Sorted seller bid \& ID:} & (2, s_2) & (4, s_3) & (5, s_1) & (8, s_4) \\ \lambda_i: & 1 & 1 & 1 & 0 \\ \text{Sorted group bid \& ID:} & (10, G_3) & (7, G_4) & (6, G_2) & (4, G_1) \end{pmatrix}.$$

The direct revelation of the values of λ_i and IDs reveals not only auction result (i.e. winners: s_2, s_3, G_3, G_4; clearing prices: 5 for sellers and 6 for buyer groups), but also bid ranking information of both winning sellers and buyer groups (i.e. s_2 bids less than s_3, and G_3 bids more than G_4). Clearly, the auction is not secure.

Motivated by the above analysis, we now design a data-oblivious method to determine auction result. We put the sorted bidder IDs, bids, and the comparison results in an array as follows:

$$\mathbf{V} = \begin{pmatrix} 1: & v(s'_1) & \dots & v(s'_{k-1}) & v(s'_k) & v(s'_{k+1}) & \dots & v(s'_Q) \\ 2: & v(G'_1) & \dots & v(G'_{k-1}) & v(G'_k) & v(G'_{k+1}) & \dots & v(G'_Q) \\ 3: & s'_1 & \dots & s'_{k-1} & s'_k & s'_{k+1} & \dots & s'_Q \\ 4: & G'_1 & \dots & G'_{k-1} & G'_k & G'_{k+1} & \dots & G'_Q \\ 5: & 1 & \dots & 1 & 1 & 0 & \dots & 0 \end{pmatrix}.$$

The fifth row of V is the values of $\lambda_i (1 \le i \le Q)$, i.e. $v_{5,i} = \lambda_i$, where we assume that $\lambda_i = 1$ when $i \le k$, $\lambda_i = 0$, otherwise. We repeatedly compute the following equations from right to left (i.e. i varies from $Q-1$ to 1 by step -1):

$$v_{1,i} = \lambda_{i+1} v_{1,i+1} + (1 - \lambda_{i+1}) v_{1,i}$$
$$v_{2,i} = \lambda_{i+1} v_{2,i+1} + (1 - \lambda_{i+1}) v_{2,i}$$
$$v_{3,Q} = v_{4,Q} = 0$$
$$v_{3,i} = \lambda_i \lambda_{i+1} v_{3,i}$$
$$v_{4,i} = \lambda_i \lambda_{i+1} v_{4,i}$$

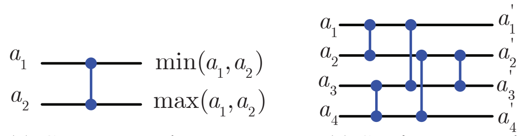
In the end, $v_{1,1}$ and $v_{2,1}$ are revealed to be clearing prices for sellers and buyer groups, respectively. Sort $v_{3,i}$ and $v_{4,i}$ ascendingly, then reveal and indicate winners as follows: if $v_{3,i} > 0$, then $v_{3,i}$ is the ID of a seller winner; if $v_{4,i} > 0$, then $v_{4,i}$ is the ID of a buyer group winner. Note that winners are revealed in ascending order of IDs, which is independent on bid ranking. In this way, we can compute the auction result without leaking bid ranking information.

C. Design and Optimization for Building-block Circuits

In this subsection, we detail the design and optimization of the building-block circuits in our design for both single-sided and double secure spectrum auctions.

1) Sorting Network: In our implementations for both single-sided and double spectrum auctions, bid sorting is a dominant operation. We implement the sorting circuit using an odd-even merge sorting network [8], [9], a circuit that sorts an input sequence $(a_1, a_2, ..., a_n)$ into a monotonically increasing sequence $(a'_1, a'_2, ..., a'_n)$. The main building block of sorting network is compare-and-swap circuits, a binary operator taking as input a pair (a_1, a_2), and returning the sorted pair a'_1, a'_2, with $a'_1 = \min(a_1, a_2)$ and $a'_2 = \max(a_1, a_2)$. Fig. 6 gives an illustration of a compare-and-swap circuit and an odd-even merge sorting network for $n = 4$. As desirable properties, the odd-even merge sorting network is data-oblivious by nature and achieves a computation complexity of $O(n \log^2 n)$. In practice, it outperforms most widely used sorting network algorithms [10].

2) Atomic Building-block Circuits: In order to implement buyer group bidding and bid sorting circuits, we need to design the following basic building-block circuits: integer comparison, swap, minimum selection, integer multiplication with a constant. We call them atomic building-block circuits, or atomic circuits more concisely. In our circuit design, similar to garbled circuits, we can make use of the "XOR-free"



(a) Compare-and-swap (a) Sorting Network

Fig. 6. Odd-even Merge Sorting Network: $n = 4$

property to optimize these circuits. Because XOR gate is communication-free in bit-sharing circuits, we aim to use as few as possible AND gates for each atomic circuit to minimize the communication overhead. In the following, we denote two K-bit non-negative integers by $x = (x_K x_{K-1} ... x_2 x_1)$ and $y = (y_K y_{K-1} ... y_2 y_1)$. The four atomic circuits can be designed and optimized as follows.

- **Integer comparison.** We directly apply the circuit proposed in [7] for integer comparison, which is optimized by "XOR-free" property. To compare integers x and y, the circuit can be described as follows:
  $$c_{i+1} = x_i \oplus (x_i \oplus c_i) \wedge (y_i \oplus c_i) \quad \text{subject to } 1 \leq i \leq K$$
  where for $c_1 = 0$, the comparison result $c_{K+1} = [x > y]$; for $c_1 = 1$, $c_{K+1} = [x \geq y]$. Comparison circuits for $[x < y]$ and $[x \leq y]$ can be obtained by interchanging x and y. As we can see, the K-bit integer comparison circuit only contains K AND gates.
- **Swap.** Suppose that x and y are the two integers to swap, and the swap indicator is denoted by b (If $b = 1$, swap x and y; else, remain untouched). A swap can be optimized with only K AND gates as follows:
  $$x'_i = [b \wedge (x_i \oplus y_i)] \oplus x_i, \text{ and } y'_i = x'_i \oplus (x_i \oplus y_i) \quad \text{subject to } 1 \leq i \leq K$$
- **Minimum selection.** Minimum selection circuit with two integers x and y can be optimized with only $2K$ AND gates as
  $$b = [x > y], \quad m_i = [b \wedge (x_i \oplus y_i)] \oplus x_i \quad \text{subject to } 1 \leq i \leq K$$
  The above circuit can be easily extended to minimum selection circuit with n integers using $2K(n-1)$ AND gates.
- **Integer multiplication with a constant.** First, we can apply the circuit proposed in [7] for integer addition with integers x and y, using K AND gates, as follows.
  $$c_{i+1} = c_i \oplus (x_i \oplus c_i) \wedge (y_i \oplus c_i), \quad s_i = c_i \oplus x_i \oplus y_i \quad \text{subject to } 1 \leq i \leq K$$
  where $c_1 = 0$. The final sum is $s = (c_{K+1} s_K s_{K-1} ... s_2 s_1)$. Then based on the optimized integer addition, we can design the optimized integer multiplication with a constant by Algorithm 4 using $K^2 \sim 2K^2$ AND gates.

Algorithm 4 Integer Multiplication with a Constant

Input: x and a constant $c = (c_K c_{K-1} ... c_2 c_1)$
Output: Product $p = c \cdot x$

```
xx = x; p = 0;
for all 1 ≤ i ≤ K do
    p = (c_i == 1) ? (p + xx) : p;
    xx = xx + xx;
end for
return p;
```

VI. Performance Analysis and Evaluation

As our implementations for both single-sided and double spectrum auctions under framework ITSEC exactly follow the procedure of original auction mechanisms, the auction related performance is the same as that of the original ones and is extensively evaluated in existing literatures. So, in this section, we focus on the analysis and evaluation of computation and communication overhead caused by the security measures.

A. Analysis of Computation and Communication Complexities

For both of our single-sided and double spectrum auction implementations, the dominant computation component is bid sorting, whose complexity is $O(n \log^2 n)$ where n is the number of items sorted. Assuming that $M \leq N$, and the average size of buyer groups is roughly constant, we can derive that the computation complexities of both single-sided and double spectrum auction implementations are $O(N \log^2 N)$ with fixed bit length of bids, where N is the number of buyers. This computation complexity is slightly higher than the complexity $O(N \log N)$ of the corresponding unsecured schemes determined by sorting algorithms. Similarly, we can derive that the communication complexities of both implementations are also $O(N \log^2 N)$.

B. Evaluation of Computation and Communication Overhead

We implement both single-sided and double spectrum auction mechanisms under ITSEC using Java, and simulate the three parties with three processes on a computer. We also implement the corresponding unsecured schemes of both single-sided and double spectrum auctions for comparison. Experimental settings are as follows: All buyers are randomly distributed in an area of 1000m × 1000m with protection distance being 500m; Both seller and buyer bids take values randomly in the interval $[0, 2^K - 1]$ with $K = 10$; The number M of channels for single-sided spectrum auctions is 200. All experimental results are averaged on 10 random runs. In our simulation, we focus on the following performance metrics: running time (sum of CPU time spent by the three parties) and message volume (data size of all messages sent between the parties).

Figs. 7 and 8 trace the experimental results in both single-sided and double auctions. The running time curves of ITSEC schemes and unsecured schemes are compared, and message volume curves are illustrated, as N varies from 1000 to 10000 by step 1000 (and for double spectrum auctions, M varies from 300 to 3000 by step 300). From the figures, we can make the following observations:

- *Curve Trend.* Both the running time and message volume curves of ITSEC grow faster than linearly in N in both auction cases, which is in accordance with the theoretical results we obtain. Particularly, the message volume curves scale only slightly faster than linearly in both auction cases, which probably results from the "XOR-free" property and our circuit optimization.
- *Running Time.* Even for large N, the running times of ITSEC scheme are very close to those of the unsecured schemes in both auction cases. For example, when

$N = 10000$, for single-sided auctions, the running time of ITSEC is 65.31 min, and that of the unsecured scheme is 63.63 min; for double auctions, the running time of ITSEC is 67.59 min, and that of the unsecured scheme is 59.48 min. The results demonstrate that ITSEC brings only limited computation overhead.

- *Message Volume.* The message volumes of ITSEC are also limited with respect to the number of sellers and buyers. For example, when $N = 10000$, the message volume is 4.91MB for single-sided auctions, and is 8.73MB for double auctions. The results demonstrate that ITSEC incurs only limited communication overhead.

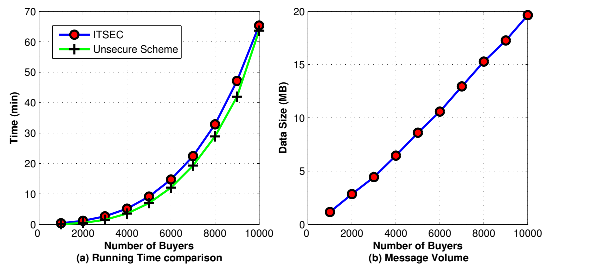


Fig. 7. Computation and communication overhead for single-sided auctions

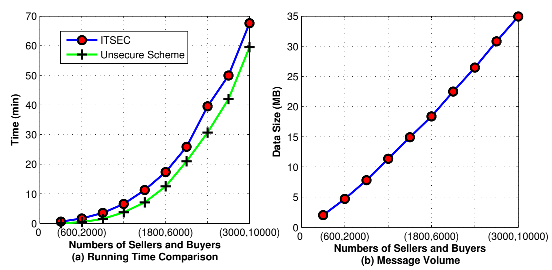


Fig. 8. Computation and communication overhead for double auctions

From the experimental results, we conclude that ITSEC achieves comparable performance in terms of computation with respect to unsecured spectrum auction schemes, and incurs only limited communication overhead.

VII. Conclusion

In this paper, we have proposed ITSEC, the first information-theoretically secure framework for truthful spectrum auctions. Previous studies on secure spectrum auctions either did not provide adequate security, or suffered great performance overhead for security. In contrast, we have achieved information-theoretic security in the sense of cryptography, and formally proved the security in the presence of semi-honest adversaries in this work. Specifically, ITSEC reveals nothing about the bids to any participant with arbitrary computation power, except the auction result. Furthermore, we have implemented both single-sided and double auction mechanisms under ITSEC framework in Java, and have theoretically and experimentally shown that ITSEC achieves comparable performance in terms of computation with respect to spectrum auction mechanisms without any security measure, and incurs only limited communication overhead.

Acknowledgment

The work of Zhili Chen and Liusheng Huang is supported by the National Science Foundation of China under Grant No. 61202407, U1301256, 61170058, the National Science and Technology Major Project under Grant No. 2012ZX03005009, Special Project on IoT of China NDRC (2012-2766), and Fundamental Research Funds for the Central Universities (No. WK0110000033).

References

[1] A. C. Yao. Protocols for secure computations. Proc. FOCS 1982.
[2] M. Bellare, V.T. Hoang, P. Rogaway. Foundations of Garbled Circuits. Proc. of ACM CCS 2012.
[3] Z. Chen, L. Huang, L. Li, W. Yang, H. Miao, M. Tian, F. Wang. PS-TRUST: Provably Secure Solution for Truthful Double Spectrum Auctions. Proc. of IEEE INFOCOM 2014, pp.1249-1257, 2014.
[4] A Shamir. How to share a secret. Communications of the ACM, 22(11): 612-613, 1979.
[5] G. R. Blakley. Safeguarding cryptographic keys. Proc. NCC 48, 1979.
[6] R. Cramer, I. Damgard, J. B. Nielsen. Secure Multiparty Computation and Secret Sharing: An Information Theoretic Approach. Book draft, 2014.
[7] V. Kolesnikov, A. R. Sadeghi, and T. Schneider. Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima. Proc. CANS 2009, LNCS 5888.
[8] K. E. Batcher. Sorting networks and their applications. In Proc. AFIPS Spring Joint Computer Conference 32, 1968.
[9] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, D. Boneh. Privacy-Preserving Matrix Factorization. Proc. ACM CCS 2013.
[10] G. Wang, T. Luo, M. T. Goodrich, W. Du, and Z. Zhu. Bureaucratic protocols for secure two-party sorting, selection, and permuting. Proc. AsiaCCS 2010.
[11] X. Zhou, S. Gandhi, S. Suri, and H. Zheng. ebay in the sky: Strategyproof wireless spectrum auctions. Proc. MobiCom 2008.
[12] X. Zhou and H. Zheng. Trust: A general framework for truthful double spectrum auctions. Proc. INFOCOM 2009.
[13] O. Goldreich. Foundations of Cryptography: Volume 2-Basic Applications. Cambridge University Press, 2004.
[14] C. Hazay, Y. Lindell. Efficient secure two-party protocols: Techniques and constructions. Springer, 2010.
[15] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. Proc. EC 1999.
[16] K. Suzuki, M. Yokoo. Secure generalized vickrey auction using homomorphic encryption. Proc. FC 2003.
[17] M. Yokoo, K. Suzuki. Secure generalized vickrey auction without third-party servers. Proc. FC 2004.
[18] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan. Robust, privacy protecting and publicly verifiable sealed-bid auction. Proc. ICICS 2002.
[19] F. Brandt and T. Sandholm. On the existence of unconditionally privacy-preserving auction protocols. ACM TISSEC, 11(2): 1-21, May 2008.
[20] I. Damgard, T. Toft. Trading sugar beet quotas: secure multiparty computation in practice. Ercim News, 2008.
[21] Spectrum Bridge, Inc., http://www.spectrumbridge.com.
[22] M. Al-Ayyoub and H. Gupta. Truthful spectrum auctions with approximate revenue. Proc. INFOCOM 2011.
[23] P. Xu, X.Y. Li, S. Tang, and J. Zhao. Efficient and strategyproof spectrum allocations in multichannel wireless networks. IEEE Trans. Computers, 60(4): 580-593, Apr. 2011.
[24] P. Xu, X. Xu, S. Tang, and X.Y. Li. Truthful online spectrum allocation and auction in multi-channel wireless networks. Proc. INFOCOM 2011.
[25] X. Feng, Y. Chen, J. Zhang, Q. Zhang, and B. Li. TAHES: Truthful double auction for heterogeneous spectrums. Proc. INFOCOM 2012, Mar. 2012.
[26] P. Lin, X. Feng, Q. Zhang and M. Hamdi. Groupon in the Air: A Three-stage Auction Framework for Spectrum Group-buying. Proc. INFOCOM 2013.
[27] D. Yang, X. Zhang, G. Xue. PROMISE: A Framework for Truthful and Profit Maximizing Spectrum Double Auctions. Proc. INFOCOM 2014.
[28] M. Pan, J. Sun, Y. Fang. Purging the Back-Room Dealing: Secure Spectrum Auction Leveraging Paillier Cryptosystem. IEEE JSAC, 29(4): 866-876, 2011.
[29] Q. Huang, Y. Tao, and F. Wu. SPRING: A Strategy-Proof and Privacy Preserving Spectrum Auction Mechanism. Proc. INFOCOM 2013.