**SCIS 2014** The 31st Symposium on
Cryptography and Information Security
kagoshima, Japan, Jan. 21 - 24, 2014
The Institute of Electronics,
Information and Communication Engineers

# Non-Interactive First-Price and Second-Price Auction Protocols Using Fully Homomorphic Encryption

Takehiro Mimasu * Masayuki Abe † Tatsuaki Okamoro ‡

**Abstract**— This paper presents new protocols for first-price and second-price auctions. They release no information except auction results, which are the winner who bids the highest value, the highest bidding price and the second highest price. Both the auction protocols are "non-interactive" and "single-pass", while the previous first-price and second-price auction protocols, Mitsunaga et al., and Kurosawa et al. are interactive and two-pass. Our protocol for second-price auction has a single-pass structure where the auction managers can find the second highest price and the winning bidder at the same time. To achieve those desirable properties, we require fully homomorphic encryption that allows to evaluate multiplication up to $O(m^2 \log v)$ for $m$ bidders and maximum price $v$.

**Keywords:** auction, 1st price auction, 2nd price auction, fully homomorphic encryption.

## 1 Introduction

### 1.1 Background

Recently, as the Internet has been expanded, many researchers have become interested in secure auction protocols and various schemes have been proposed to ensure the safe transaction of sealed-bid auctions. About sealed-bid auctions, for example, a first-price auction protocol, in which the highest price bidder is the winner and get products by the highest bidding price, release no information about bidders who lose an auction and losers' bidding prices. A second-price auction protocol, in which the winner takes bids by the second highest bidding price, also leaks no information about bidding prices except the highest one and bidders except a winner. To achieve these protocols, a simple solution is to assume a trusted auctioneer. Bidders encrypt their bids and send to an auctioneer. An auctioneer processes data and broadcasts results. To avoid using the trusted auction manager, some secure multiparty protocols have been proposed. Mitsunaga et al. proposed secure protocols based on 2-DNF formulas which are able to do multiplication only once on ciphertexts. They need to make mix-and-match table before the auction. In regard to this point there is high round complexity. In addition, a previous protocol for second-price auction runs the protocol for first-price auction twice; once to identify and remove the bid containing the highest price, and second time to obtain the second highest price. Such a two-pass structure doubles the workload of the auction managers.

Kurosawa and Ogata[3] suggested the "bit-slice auction", which split bids to bits consisted of 0 or 1 and evaluate them.

### 1.2 Our Result

In this paper, we introduce bit-slice auction protocols based on fully homomorphic encryption. To solve shortcomings of previous protocols, we use threshold fully homomorphic encryption in [2]. In the both of first and second-price auction, our protocols are more efficient in communication complexity than protocols proposed in [1]. Especially, in second-price auction, our protocol consists of single-pass structure; once to output the second highest bidding price and a winner.

## 2 Preliminaries

### 2.1 The Model of Auctions and Outline of Auction Protocols

This model assumes $m$ players, denoted by $P_1$, $P_2$, ... $P_m$ and a public board. Each $P_i$ takes bids by a bidding price, $B_i$. Its outline is as follows.

1. **Input stage:** Each $P_i (1 \leq i \leq m)$ computes ciphertexts of the bits of $B_i$, broadcasts them, and proves that the ciphertext is encrypted by whether 0 or 1 using zero-knowledge proof technique.

2. **Processing stage:** Auction managers, AMs, receive ciphertexts of all players and evaluate them.

3. **Output stage:** After evaluating ciphertexts, AMs output results and decrypt them.

* The author is with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan. (mimasu@ai.soc.i.kyoto-u.ac.jp)

† The author is with NTT Secure Platform Labolatories, NTT Corporation, Musashino -shi, 180-8585 Japan and is an associate guest professor of Kyoto University, Kyoto-shi, 606-8501 Japan.

‡ The author is with NTT Secure Platform Labolatories, NTT Corporation, Musashino -shi, 180-8585 Japan and is a guest professor of Kyoto University, Kyoto-shi, 606-8501 Japan.

#### 2.1.1 Security

Our protocol satisfies the following.

- Even if an adversary corrupts up to $n-1$ parties out of $n$ or up to $t$ auctioneers out of $k$ ($t \leq k$), he cannot know about information of a party which are not corrupted.
- Even if an adversary corrupts up to $n-1$ parties or up to $t$ auctioneers, AMs output correct results. Namely, the highest bidder is identified and the highest or second highest bidding price is released in each auction protocol.

### 2.2 Bit-Slice Auction Circuit

We introduce an auction circuit called the bit-slice auction circuit described in [2]. Suppose that $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$ is the highest bidding price and a bid of a player $i$ is $B_i = (b_i^{(k-1)}, ..., b_i^{(0)})_2$, where $()_2$ is a binary expression. In the first round, the most important bit is $b_i^{(k-1)}$. If all of $b_i^{(k-1)} = 0$, $b_{max}^{(k-1)}$ is 1, otherwise 0. A winner flag, $W = (w_1, ..., w_m)$ which is all of $w_i$ is 1 in the beginning is also used and $w_i$ is updated to 0 when a player $i$ is not able to win. In the second round, we evaluates the same process for all of the second most important bit, $b_i^{(k-2)}$, and so on. This process is repeated for k-1 to 0. At the end of all the process, the circuit outputs the highest bidding price and its bidder.

In this circuit, we are able to get only the highest price and its bidder. Therefore, in the second price auction, we have to construct a two-pass structure that after the highest bidding price and its bidder are determined, we evaluate the circuit to decide the second highest bidding price.

Using this circuit, auction managers, AMs, need to have an interaction to bidders. In order to determine $b_{max}^{(k-j)}$, many additional operator '+' is used and so the value of $b_{max}^{(k-j)}$ can be larger than 1. However, $b_{max}^{(k-j)}$ should be only 0 or 1. For this reason, AMs have to map the number larger more than 1 to 1 with a mapping table. Interactions is necessary for preparing a mapping table in advance.

### 2.3 Fully Homomorphic Encryption

#### 2.3.1 Requirements for the Encryption Function

Let $E$ be a public-key probabilistic encryption function. We denote the set of encryptions for a plaintext $v$ by $E(v)$ and a particular encryption of $v$ by $c \in E(v)$.

Function $E$ must satisfy the following properties.

**1.Fully Homomorphic Property** There exist polynomial time computable operations.

1. If $c_1 \in E(v_1)$ and $c_2 \in E(v_2)$, then $c_1 + c_2 \in E(v_1 + v_2)$.
2. If $c \in E(v)$, then $-c \in E(-v)$.
3. If $c_1 \in E(v_1)$ and $c_2 \in E(v_2)$, then $c_1 \times c_2 \in E(v_1 \times v_2)$.

**2.Threshold Decryption** For a given ciphertext $c \in E(v)$, any $t$ out of $m$ players can decrypt $c$ along with a zero-knowledge proof of the correctness. However, any $t-1$ out of $m$ players cannot decrypt it.

#### 2.3.2 Threshold Fully Homomorphic Encryption System

We describe the 2 algorithms and 2 N-party protocols in [2].

- **TFHE.Keygen**(**setup**)-(key generation protocol): initially each party holds setup. At the conclusion of the protocol, each party $P_k$ for $k \in [N]$ outputs a common public-key $pk$, a common public evaluation key $evk$, and a private *share* $sk_k$ of implicitly defined secret key $sk$.
- **TFHE.Enc**$_{pk}(\mu) \rightarrow c$ : Encrypts a bit $\mu \in \{0,1\}$ under public key $pk$. Outputs ciphertext $c$.
- **TFHE.Eval**$_{pk}(f, c_1, ..., c_l) \rightarrow c_f$ : The *homomorphic evaluation algorithm* is a *deterministic* poly-time algorithm that takes the evaluation key $evk$ , a boolean circuit $f : \{0,1\}^l \rightarrow \{0,1\}$, and a set of $l$ ciphertexts $c_1, ..., c_l$.
- **TFHE.Dec**$_{sk_1,...,skin}(c)$-(decryption protocol): Initially, each party $P_k$ holds a common cipher text $c$ and its private share of the secret key $sk_k$. At the end of the protocol each party receives the decrypted plaintext $\mu$.

## 3 New non-interactive protocols

### 3.1 First Price Auction Using Fully Homomorphic Encryption

We assume m players, $P_1$, ..., $P_m$ and a set of auction managers, AMs. We can assume that AMs are either a subset of players or a different group such as management group for auctions. They decide respectively their bidding price, $B_i$, for i=1 to m. We also show $B_i = (b_i^{(k-1)}, ..., b_i^{(0)})_2$ as a binary expression.

#### 3.1.1 setting

All parties share a common setup consisting of params and a common random string, CRS. By using these parameters as an input, all parties execute **TFHE.Keygen**, which is two-round protocol, together and generate public evaluation key, $evk$, public encryption key $pk$ and share of secret key, $s_D^i$ which each party $P_i$ has.

#### 3.1.2 Bidding Phase

Each player $P_i$ decides their bidding price, $B_i = (b_i^{(k-1)}, ..., b_i^{(0)})_2$, and encrypts it

$$Enc_i = (c_{i,k-1}, ..., c_{i,0})_2$$

where $c_{i,j} \in E(b_i^{(j)})$ and publishes them on bulletin board. $P_i$ has to prove $b_i^{(j)} \in \{0,1\}$. Several ways can

be used. For example, we can use a technique that an output of an evaluation function, $\text{eval}(E(b_i^{(j)})) = E(b_i^{(j)} \times (1 - b_i^{(j)}))$ is whether 0 or not, or a zero-knowledge proof. If you want to avoid interactions completely in this process, you can use non-interactive zero-knowledge proof.

#### 3.1.3 Opening Phase

The AMs generate $W_j = (w_{1,j}, ..., w_{m,j})$. When j=1, each $w_{i,j} = 1$. They encode $W_1$ as $\tilde{W}_1 = (\tilde{w}_{1,1}, ..., \tilde{w}_{m,1})$ by Enc algorithm.

**(Step 1)** For j=k-1 to 0, perform the following.

For $\tilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AMs compute $s_{i,j} = \tilde{w} \times c_{i,j}$ for each player i and

$$S_j = \{(\tilde{w}_{1,j} \times c_{1,j}), (\tilde{w}_{2,j} \times c_{2,j}), ..., (\tilde{w}_{m,j} \times c_{m,j})\}$$
$$h_j = 1 - \prod_{i=1}^{m} (1 - s_{i,j})$$
$$\tilde{w}_{i,j-1} = h_j \times s_{i,j} + (1 - h_j) \times \tilde{w}_{i,j}$$

**(Step2)** AMs decrypt each $\tilde{w}_i$. If and only if $w_i = 1$, $P_i$ is the winner of this auction and AMs decrypt $Enc_i$ as the highest bidding price.

#### 3.1.4 Correctness

General syntax of this protocol is same with previous protocol[2]. So, it satisfies correctness.

#### 3.1.5 Example

We show an example 3-player auction. The information we need to find is the highest bidder and the highest bidding price. Suppose that each bidding price is below.

$B_1 = (101)_2 = 5$
$B_2 = (111)_2 = 7$
$B_3 = (100)_2 = 4$

So, the winner is $P_2$ and the winning price is $B_2 = (111)_2 = 7$.

Supposed that setting j=1, at the beginning of this round, all players have a possibility to win this auction. In the round, j=1, AMs calculate $s_{1,1} \in E(0)$, $s_{2,1} \in E(1)$, $s_{3,1} \in E(0)$ and $h_1 \in E(1)$. Because $h_2 \in E(1)$, $\tilde{w}_{i,0}$ is updated by $s_{i,2}$, namely $\tilde{w}_{1,3} \in E(0)$, $\tilde{w}_{2,3} \in E(0)$ and $\tilde{w}_{3,3} \in E(0)$. In a same way, in j=0 round, AMs calculate $w_{i,-1}$. AMs decrypt $w_{i,-1}$ for $i = 1$ to $m$ and we can get a winner. AMs decrypt the bidding price of the winner.

### 3.2 Second Price Auction Using Fully Homomorphic Encryption

#### 3.2.1 Opening Phase

Only a winner is decided by using our first-price auction protocol. We also need to find the second highest bidding price.

**(Step 1)** AMs eliminate winner's $\tilde{w}$ flag, $\tilde{w}_{win,j}$, from the history of $\tilde{W}$. We define it as $\tilde{W}' = (\tilde{w}'_1, ..., \tilde{w}'_{m-1})$.

**(Step 2)** $A_j = \tilde{w}'_{1,j} \vee ... \vee \tilde{w}'_{m,j}$ (OR flags) and $x_j = A_{j-1} \otimes A_j$ (XOR flags) are defined for $k-1$ to 0. $\tilde{W}'$ is masked by $X$. Namely, $\tilde{W}' \wedge X = (\tilde{w}_{1,j} \wedge x_j), ..., (\tilde{w}_{m-1,j} \wedge x_j)$ for $k-1$ to 0.

**(Step 3)** By using updated $\tilde{W}'$, Ams calculate $\tilde{w}''_i = \tilde{w}'_{i,k-1} + ... + \tilde{w}'_{i,0}$.

**(Step 4)** $B_i (1 \leq i \leq m-1)$ is masked by each $\tilde{w}''_i (B_i \wedge \tilde{w}'' = (b_i^{(k-1)} \wedge \tilde{w}'', ..., b_i^{(0)} \wedge \tilde{w}''))$, so only the second highest price appear.

#### 3.2.2 Correctness

General syntax of this protocol is same with previous protocol[2]. So, it satisfies correctness.

## 4 Security and Efficiency

### 4.1 First-Price Auction Protocol

Comparing with previous protocol, our protocol is more efficient in communication complexity. In a first-price auction in Mitsunaga et al.[1], AMs refer to maps $mk$ times and checks plaintext equality test (PET), which has communication among players, $k$ times. The total number of checking PET is $mk + k$. However, no PET is needed in our first-price auction protocol. In addition, no map is necessary too.

This protocol satisfies correctness as stated above and no release about losers' information because of using threshold decryption protocol[3].

### 4.2 Second-Price Auction Protocol

In our second-price auction protocol, we also improve communication complexity. In addition, we can calculate result in a single-pass. Bidders just send their encrypted bid to AMs and they can get the second highest bidding price and the winner. It does not bother AMs. In the same way as our first-price auction protocol, this protocol satisfies correctness and an indistinguishability of input.

## 5 Conclusion

We introduce new efficient auction protocols based on fully homomorphic encryption and show that they are more efficient than [1].

## References


[1] T.Mitsunaga, Y.Manabe and T.Okamoto "Efficient Secure Auction Protocols Based on the Boneh-oh-Nissim Encryption," *IEICE TRANS. FUNDAMENTALS, VOL.E96-A, NO.1 JANUARY 2013*, pp.68-75.

[2] G.Asharov, A.Jain, A.Lopez-Alt, E.Tromer, V.Vaikuntanathan and D.Wichs "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE," *EUROCRYPTO 2012 Lecture Notes in Computer Science Volume 7237, 2012*, pp.483-501.

[3] K.Kurosawa and W.Ogata "Bit-Slice Auction Circuit," *ESORICS 2002 Lecture Note in Computer Science 2502, 2002*, pp.24-38.