An Efficient, Privacy-Preserving, and Verifiable Online Auction Mechanism for Ad Exchanges

Minping Zhou, Chaoyue Niu, Zhenzhe Zheng, Fan Wu, and Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems
Shanghai Jiao Tong University, China
{zhouminping1991, rvincency, zhengzhenzhe220}@gmail.com; {fwu, gchen}@cs.sjtu.edu.cn

***Abstract*—Ad exchanges are kind of the most popular online advertising marketplaces for trading ad spaces over the Internet. Ad exchanges run auctions to sell the ad spaces on publishers' web-pages to advertisers, who want to display ads on the ad spaces. However, the parties in the auction cannot check whether the auction is carried out correctly or not. Furthermore, the advertisers are usually not willing to reveal their sensitive information when participating in the auction. In this paper, we jointly consider the auction verifiability and advertisers' privacy preservation, and propose ERA, which is an Efficient, pRivacy-preserving, and verifiAble online auction mechanism for ad exchanges. ERA exploits an Order Preserving Encryption Scheme (OPES) to guarantee privacy-preservation, and achieves verifiability by integrating a Certified Bulletin Board (CBB) and a protocol of Privacy-Preserving Integer Comparison (PPIC), which is based on the Paillier's Homomorphic Encryption Scheme (PHES). We extensively evaluate the performance of ERA, and our evaluation results show that ERA satisfies the properties of verifiability and privacy-preservation with low overhead, so ERA can be easily deployed in today's ad exchanges.**

I. Introduction

An ad exchange is considered as a new type of Internet market, where ad places on web-pages are traded in real-time via an auction mechanism. A number of ad exchanges have emerged on the Internet, such as DoubleClick [1], RightMedia [2] and OpenX [3]. There are billions of ad transactions per day across more than 2 million websites [4], and Internet companies, such as Google and Microsoft, have extracted a large amount of revenue every year from the ad transactions in their ad exchange platforms.

In ad exchanges, auctions are regarded as the most important core technique to efficiently allocate ad spaces. In an ad auction, interested advertisers are allowed to bid for an ad space, and the highest bidding advertiser gets the opportunity to present her advertisement. However, the current ad auction has two undesirable problems: privacy leakage and auction manipulation. On one hand, advertisers are required to submit their bids to participant in the ad auction, which will inevitably disclose their private information. On the other hand, publishers and advertisers have no control over the outcome determination, and are forced to unconditionally accept it, even if the ad exchange manipulates the auction. Under this paradigm, the correctness of ad auctions is totally relied on the reputation of ad exchanges. Thus, it is highly needed to design an ad auction mechanism that achieves both *privacy-preservation* and *verifiability*. In a privacy-preserving and verifiable auction, ad exchanges are able to calculate the auction outcome, and prove its correctness without knowing the private information of advertisers. If privacy-preservation and verifiability are guaranteed, ad exchanges will attract a wider range of advertisers and publishers to engage in.

However, existing auction mechanisms rarely considered these two properties at the same time. The auction mechanisms [5] [6] [7] achieved verifiability with the assumption that the bid information should be revealed to the auctioneer. Although some researchers have proposed solutions for the problem of bid privacy-preservation [8], they ignored the consideration of verifiability. Moreover, the auction in the ad exchange is fundamentally different from these conventional auction mechanisms due to the following two requirements:

- *Low-Latency*: Unlike traditional goods, ad spaces are supremely perishable. If an auction for an ad space does not complete before the web-page is rendered on the user's browser, then the opportunity to place an ad is lost. Therefore, the time for executing the ad auction is usually limited in a short time interval, *e.g.*, typically 100 milliseconds [9].
- *Large-Scale*: There are billions of auctions for ad spaces per day with millions of advertisers participating in [4] [10].

Considering the above two requirements, there exists many challenges designing an efficient ad auction mechanism to achieve both privacy-preservation and verifiability.

The first design challenge is the efficiency requirement of ad auctions, *i.e.*, ad auctions should support a large scale of advertisers with low latency. The process of winner determination and payment calculation should be computed in a short time, which hinders the application of time-consuming encryption schemes even if they have good security properties. The second design challenge comes from the requirement of privacy-preservation. The outcome of the auction should be calculated without knowing private information of advertisers. This seems to be contradictory, because the auctioneer should examine all bids to determine the winner and payment. The last but not least design challenge is the verifiability. Although the auction verification can be performed offline, and has no strict time restriction, it is not a easy job to design a verifiable auction without breaking the property of privacy-preservation.

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61422208, 61472252, 61272443 and 61133006, in part by CCF-Intel Young Faculty Researcher Program and CCF-Tencent Open Fund, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, and in part by Jiangsu Future Network Research Project No. BY2013095-1-10. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

F. Wu is the corresponding author.

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

In this paper, we jointly consider the three design challenges, and develop ERA, which is an Efficient, pRivacy-preserving, and verifiAble online auction mechanism for ad exchanges. ERA first models the ad space allocation as a three-tier auction model: one auctioneer, intermediary ad networks and advertisers. Rather than imposing all computation on the ad exchange, the intermediary ad networks can help the ad exchange calculate the auction outcome in a parallel way, and thus the auction latency can stay at a low level. By using OPES, ERA then encrypts the bids of advertisers while still maintaining the order of bids. Therefore, the auction outcome can be calculated in the ciphertext domain, which guarantees the property of privacy-preservation. At last, ERA achieves the property of verifiability by adopting a protocol of PPIC, which is based on PHES.

In general, our contributions are summarized below:

- We model the problem of ad space allocation in ad exchanges as a three-tier auction model, in which there are one auctioneer, intermediary ad networks, and advertisers.
- We propose an efficient, privacy-preserving and verifiable online auction mechanism for ad exchanges, namely ERA. To the best of our knowledge, ERA is the first *private-preserving* and *verifiable* auction mechanism that can support a large scale of advertisers with low latency.
- We have implemented ERA, and extensively evaluated its performance. Our evaluation results show that ERA achieves good efficiency and is practical for today's ad exchanges.

The remainder of this paper is organized as follows. The ad exchange model and required properties are proposed in Section II. We describe several relevant cryptographic tools in Section III, and propose a protocol of PPIC in Section IV. The detailed design of ERA is proposed in Section V. In Section VI, we evaluate ERA, and report evaluation results. In Section VII, we briefly review the related work. Finally, we draw conclusions in Section VIII.

II. Preliminaries

In this section, we first propose the system model and auction model for ad exchanges, and then present two required properties for a practical ad exchange.

A. System Model

We consider a real-time online advertising marketplace, where there are web users, publishers, one ad exchange, ad networks, and advertisers. Generally, every view of web users on a publisher's web-page stimulates the conduct of a second-price ad auction, in which the ad spaces on the web-page are efficiently sold to advertisers. We now describe the system model of an ad exchange, which is based on the model proposed by Muthukrishnan [11], and is a generalization of current ad exchange models in the literature.

As shown in Figure 1, an ad auction is initiated when a web-user visits to a publisher's web-page, which contains an HTML iframe of JavaScript snippet that generates an ad request to the ad exchange (Step ①). From the ad request, the ad exchange can extract the relevant information, *e.g.*, the behaviour feature of the web-user, the reserve price set by the publisher and the type of the ad space (Step ②). The relevant information is transmitted to all advertisers through ad networks (Step ③).

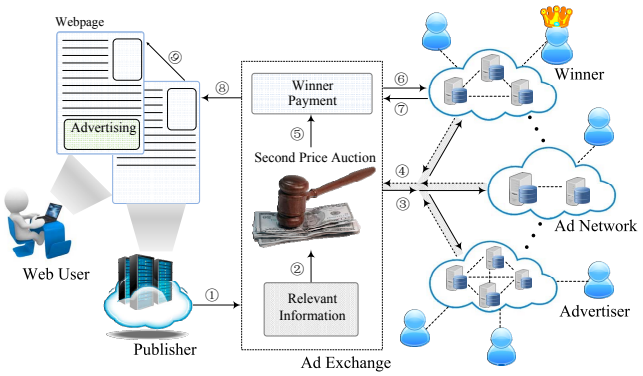

Fig. 1. An ad exchange ecosystem.

Based on these information, advertisers can accurately derive the valuation over the ad space. The advertisers submit their bids, which are calculated according to their valuations, to the ad networks they belong to, and then to the ad exchange (Step ④). According to the reported bids, the ad exchange runs a second-price auction to determine the winner and her payment, which are published to the public (Step ⑤). The ad exchange requires the winner to submit her ad tag and charge, which are further sent to the publisher. (Step ⑥, ⑦ and ⑧). Finally, the publisher presents the winner's advertisement on her web-page (Step ⑨). This process typically completes within 100ms.

B. Auction Model

We model the ad space trading as a sealed bid auction with a single item. The trading items in ad auctions are ad spaces, which are classified into several types, *e.g.*, videos, images, and texts, based on the information of web-pages. Without loss of generality, we consider a specific type of ad spaces in the following discussion. In our ad auction model, there are four major participants: ad bidders, ad networks, an agent, and an auctioneer, which are introduced in details as follows.

Ad bidders: The set of l ad bidders are denoted by $\mathbb{S} = \{s_1, s_2, \ldots, s_l\}$. Each bidder $s_i \in \mathbb{S}$ has an ad tag and an *original* bid b_i for the trading ad space. The original bids of all bidders are denoted by $\mathbf{b} = \{b_1, b_2, \cdots, b_l\}$.

Agent: We introduce a new party, called agent, to provide bidders with *mapped* bids, which are used to design a privacy-preserving auction. The agent employs OPES to construct the set of mapped bids, denoted by $\hat{\boldsymbol{b}} = \left\{\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_l\right\}$. We assume that the agent is honest-but-curious[1].

Ad networks: The set of m ad networks is denoted by $\mathbb{A} = \{a_1, a_2, \ldots, a_m\}$. Each ad network $a_j \in \mathbb{A}$ contains a few subscribed ad bidders, who pre-store their mapped bids and ad tags on the ad network a_j. In this way, the bidders do not need to encrypt and submit their bids on the fly, and thus the interaction between bidders and ad networks can be reduced.

Auctioneer: The auctioneer acts as the ad exchange: calculate the auction outcome: the winner s_{max} and payment b_{sec}.

We now define two requirements of ad auctions as follows.

Definition 1 (Privacy-Preserving Ad Auction): An ad auction is *privacy-preserving* if and only if *no* more information than the outcome of the auction, *i.e.*, s_{max} and b_{sec}, is revealed to any participant in the auction.

[1]The agent is honest-but-curious if she follows the designed protocol but tries to gather information about other participants.

System parameters (g, h, G_q).
Sender's input: $\mathbf{M} = (m_1, m_2, \ldots, m_n)$; **Receiver's choice**: α.
Protocol:

1. $\mathcal{R}$ sends to $\mathcal{S}$: $y = g^r h^\alpha$, $r \in_R Z_q$;
2. $\mathcal{S}$ replies $\xi_i = (g^{k_i}, m_i(y/h^i)^{k_i}), k_i \in_R Z_q, 1 \leq i \leq n$;
3. By $\xi_\alpha = (a, b)$, $\mathcal{R}$ computes $m_\alpha = b/a^r$.

Fig. 2. **Efficient 1-out-of-n Oblivious Transfer (OT)**

Scheme parameters: (p, q, r).
Public key: $n = pq$; **Private key**: $\phi = (p-1)(q-1)$.
Encryption: $C = E_n(m, r) = (1 + mn) \cdot r^n (mod\ n^2)$;
$C^{-1} = E_n^{-1}(m, r) = (1 - mn) \cdot r^n (mod\ n^2)$.
Decryption: $m = D(C, \phi) = \frac{(C^\phi - 1)/\phi \mod n^2}{n}$;
$m = D(C, r) = \frac{(C \cdot r^{-n} \mod n^2) - 1}{n}$.
Random Value Recovery: $r = C^{n^{-1}(mod\ \phi)} \ (mod\ n)$.
Additive Homomorphism: $C_1 \times C_2 = E_n(m_1 + m_2, r_1 \cdot r_2)$.

Fig. 3. **Paillier's Homomorphic Encryption Scheme (PHES)**

Definition 2 (Verifiable Ad Auction): An ad auction is verifiable if and only if the outcome of the auction can be verified by ad bidders and any external party.

III. Cryptographic Tools

In this section, we briefly describe the cryptographic tools.

Certified Bulletin Board (CBB) is an electronic version of traditional bulletin board, which can be a public and trustworthy website maintained and updated by a certain authority. A CBB can be read by anybody, but can be written only by some authorized parties, such as the auctioneer and ad networks. We note that all posts on the CBB should be signatured by the corresponding data owners for non-repudiation. The CBB is introduced to solve the problem of information asymmetry so as to facilitate the design of verifiable auctions.

Order Preserving Encryption Scheme (OPES), introduced by Agrawal *et al.* in paper [12], is an encryption technique that preserves the ordering of plaintexts in the ciphertext space. The OPES allows the auctioneer to learn the order of plaintexts by applying comparison operations on the mapped data. By exploiting this property, the auction outcome can be calculated among the mapped bids, such that the property of privacy-preservation can be achieved.

Efficient 1-Out-n Oblivious Transfer (OT) was proposed in [13] to secretly exchange a certain message between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. Specifically, the sender $\mathcal{S}$ has n messages: $\mathbf{M} = (m_1, m_2, \ldots, m_n)$, and the receiver $\mathcal{R}$ wants to know one of the messages, *e.g.*, m_α. The OT guarantees that $\mathcal{R}$ just obtains the message m_α without knowing the other $n-1$ messages, and $\mathcal{S}$ does not know the receiver's choice α. Figure 2 shows the pseudocode.

Paillier's Homomorphic Encryption Scheme (PHES) [14] introduced by Paillier, is shown in Figure 3, where p and q are two large primes. The parameter $r \in [1, n]$ is a random value, and should be the common divisor of n.

IV. Privacy-Preserving Integer Comparison

In this section, we propose a PPIC protocol, which is the basic for the design of the verifiable ad auction in next section.

In PPIC protocol, there are two parties: a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. The prover $\mathcal{P}$ knows two non-negative integers x_1 and x_2 as well as their comparison relation, *e.g.*, $x_1 \geq x_2$. The main goal of $\mathcal{P}$ is to convince $\mathcal{V}$ that the declared comparison relation, *i.e.*, $x_1 \geq x_2$, is true without disclosing the value of x_1 and x_2. Here, we emphasize that if $x_1 < x_2$, it is computationally infeasible for $\mathcal{P}$ to convince $\mathcal{V}$ that $x_1 \geq x_2$.

For two non-negative integers $x_1, x_2 < n/2$, the inequality $x_1 \geq x_2$ holds if and only if $(x_1 - x_2)\ mod\ n < n/2$. In order to demonstrate that $x_1 \geq x_2$, $\mathcal{P}$ can prove the three inequalities: $x_1 < n/2$, $x_2 < n/2$, and $(x_1 - x_2)\ mod\ n < n/2$. Therefore, the problem of PPIC can be reduced to *Range Proof*: the prover $\mathcal{P}$, who knows the value of a plaintext x, proves to $\mathcal{V}$ that $x < 2^t \leq n/2$ without leaking x.

Before proposing the Range Proof protocol, we first introduce the concept of test set TS, which is a set of Paillier's ciphertexts: $TS = \{C_1, C_2, \ldots, C_t\}$, where $C_i = E_n(m_i, r_i)$ and the plaintext $m_i = 2^i$. We note that all elements in TS should be randomly ordered to ensure the privacy of m_i.

Given $C = E_n(x, r_x)$, $\mathcal{P}$ can prove to $\mathcal{V}$ that $x < 2^t < n/2$ by Range Proof, which consists of the following two steps.

Step 1: Proof Generation An integer x can be uniquely represented by $x = 2^{t_1} + 2^{t_2} + \ldots + 2^{t_k}$. The $\mathcal{P}$ selects the ciphertext set $\mathbb{C}_x = \{C_{t_1}, C_{t_2}, \ldots, C_{t_k}\}$ of the plaintexts $2^{t_1}, 2^{t_2}, \ldots, 2^{t_k}$ from TS, and uses the corresponding random values $r_{t_1}, r_{t_2}, \ldots, r_{t_k}$ and r_x to calculate a new random value:

$$r^* = (r_x^{-1} \times r_{t_1} \times r_{t_2} \ldots \times r_{t_k})\ (mod\ n).$$

The set of ciphertexts $\mathbb{C}_x$ and the random value r^* are packaged as the proof, which is sent to the verifier $\mathcal{V}$.

Step 2: Verification After receiving the proof, $\mathcal{V}$ can verify the relation $x < 2^t < n/2$ by calculating the following equation:

$$E_n^{-1}(x, r_x) \cdot C_{t_1} \cdot C_{t_2} \cdot \ldots \cdot C_{t_k}\ (mod\ n^2) = E_n(0, r^*),$$

Due to the *Additive Homomorphism* of PHES, the above equation holds if and only if $x = 2^{t_1} + 2^{t_2} + \ldots + 2^{t_k}$. Together with the fact that the number of elements in $\mathbb{C}_x$ is less than or equal to t, $\mathcal{V}$ can conclude that $x < 2^t < n/2$.

Given the three crypttexts $E_n(x_1)$, $E_n(x_2)$ and $E_n[(x_1 - x_2)\ mod\ n] = E_n(x_1) \times E_n^{-1}(x_2)\ (mod\ n^2)$, a prover $\mathcal{P}$ can convince a verifier $\mathcal{V}$ that $x_1 \geq x_2$ by applying the Range Proof protocol to check the following three inequations: $x_1 < 2^t < n/2$, $x_2 < 2^t < n/2$ and $(x_1 - x_2)\ mod\ n < 2^t < n/2$.

V. Efficient, Privacy-Preserving, and Verifiable Online Ad Auction

In this section, we design an efficient, privacy-preserving, and verifiable online ad auction mechanism.

A. Design Overview

We illustrate the design challenges and design rationale of ERA. The first design challenge is privacy preservation in terms of bids. We introduce an honest-but-curious agent to encrypt the original bids as mapped bids using OPES. Therefore, the ad networks and the auctioneer can learn the ordering of the original bids by comparing the corresponding mapped bids, to calculate the auction outcome. However, the agent can know the original bids if she can obtain the mapped bids, and we tackle this problem by adopting two cryptographic methods. First, each bidder fetches her mapped bid from the agent via OT, which guarantees that the bidder does not leak any information about her original bid during

the mapped bid selection. However, the agent may still get the mapped bids in some other ways, such as learning the mapped bids from the public information on the CBB. So we require the auctioneer to provide one more encryption on the mapped bids. In this way, as long as there is no collusion between the agent and the auctioneer, the privacy of bid is well protected.

The second design challenge is auction verification. The ad networks and the auctioneer exclusively possess the bidding information, and any other participant cannot access it. This information asymmetry causes significantly difficulties to design a verifiable auction. To solve this problem, we introduce a CBB to publish the encrypted information, *i.e.*, doubly encrypted bids. We then employ the proposed PPIC protocol to enable any party to verify the order of the mapped bids, *i.e.*, the order of the original bids, and thus verify the correctness of auction execution.

B. Design Details

We now introduce ERA, which consists of three stages: *Initialization*, *Auction Execution*, and *Verification Operation*.

1) Initialization: The initialization stage contains two parts: bid encryption and verification preparation.

Bid Encryption: The bid space Θ is defined as the set of n possible bids: $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, where $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$. Based on the bid space, the agent runs OPES to generate a set of mapped bids: $\widehat{\Theta} = \left\{\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n\right\}$, where $\hat{\theta}_i = OPES(\theta_i)$ and $\hat{\theta}_1 \geq \hat{\theta}_2 \geq \cdots \geq \hat{\theta}_n$. Without loss of generality, we assume that the maximum mapped bid is 2^t for some t, *e.g.*, $t = 32$.

Each bidder $s_i \in \mathbb{S}$ with original bid $b_i = \theta_{i'}$ contacts the agent to fetch her mapped bid $\hat{b}_i = \hat{\theta}_{i'}$ from the mapped bid space $\widehat{\Theta}$ via OT. This guarantees that bidder s_i only knows $\hat{\theta}_{i'}$, and has no idea of the other $n-1$ mapped bids in $\widehat{\Theta}$, while the agent does not know which mapped bid is chosen by the bidder s_i. However, the agent may still know the original bid of the bidder s_i if she can access the mapped bid $\hat{b}_i$. Therefore, the ad network a_j, who is responsible for the bidder s_i, further encrypts the bid $\hat{b}_i$ using PHES with the public key n and a random value r_i. We note that the public key n is provided by the auctioneer, and the random value r_i is generated by the ad network a_j of the bidder s_i. The doubly encrypted bid of bidder s_i is denoted by $c_i = E_n(\hat{b}_i, r_i)$.

Verification Preparation: To facilitate the auction verification, the following information is posted on the CBB.

- $l-1$ test sets $\{TS_1, TS_2, \cdots, TS_{l-1}\}$: these test sets are posted by the auctioneer with her signature, and will be used to verify the comparison relation of the l bids.
- l commitments $\{COM_1, COM_2, \cdots, COM_l\}$: the commitment of the bidder $s_i \in \mathbb{S}$ is defined as $COM_i = (c_i, s_i)$. These commitments are calculated and posted by all the ad networks, and will be used to verify the auction outcome.

2) Auction Execution: The auction execution is divided into two stages: the internal auction stage and the global auction stage. Each ad network $a_j \in \mathbb{A}$ conducts an internal auction among her bidder members. The ad network a_j selects the highest and second highest mapped bids, and sends them to the auctioneer with the signature. In the global auction stage, the auctioneer chooses the global highest and second highest mapped bids, *i.e.*, $\hat{b}_{max}$ and $\hat{b}_{sec}$, from the internal outcomes provided by ad networks. Finally, the auctioneer obtains the identity of winner s_{max} and determines the payment. The auctioneer sends the mapped bid $\hat{b}_{sec}$ to the agent, and the agent sends back the winner's payment, which is the original bid b_{sec} of $\hat{b}_{sec}$. The agent can obtain this payment by using the inverse function of $OPES(\cdot)$: *i.e.*, $b_{sec} = OPES^{-1}(\hat{b}_{sec})$.

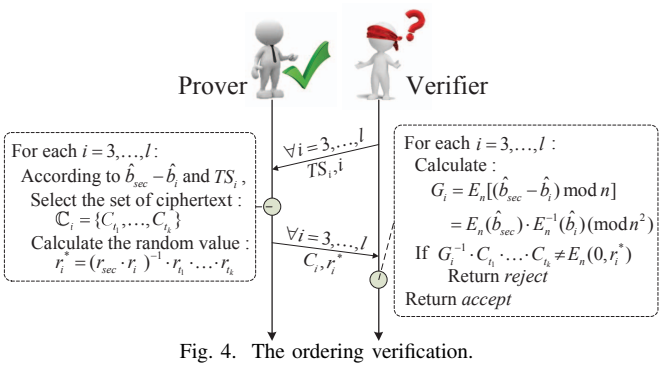

Fig. 4. The ordering verification.

3) Verification Operation: At the end of auction execution stage, the ad network in charge of the bidder with $\hat{b}_{sec}$ is required to mark her commitment, on the CBB for verification. Intuitively, the auction outcome is correct if the ad space is sold to the bidder with the global highest bid, and the payment is equal to the global second highest bid. Formally, we claim that the auction outcome is correct if the two conditions: $b_{max} \geq b_{sec}$ and $b_{sec} \geq b_i, \forall i \neq max$ are satisfied.

We assume that the auctioneer serves as the prover $\mathcal{P}$, and any party can decide to be a verifier $\mathcal{V}$. We now describe *Verifying Algorithm*, which consists of three components: payment verification, ordering verification, and patching verification.

Step 1: Payment Verification

In this step, the auctioneer sends the public key n and random value r_{sec}[2] to the $\mathcal{V}$. After that, $\mathcal{V}$ re-encrypts the mapped payment $\hat{b}_{sec}$ provided by the agent: $c_{sec} = E_n(\hat{b}_{sec}, r_{sec})$. The $\mathcal{V}$ then checks whether c_{sec} is equal to the marked doubly encrypted payment on the CBB.

Step 2: Ordering Verification

As the guarantee of OPES, we can just verify the order of mapped bids to see whether the order of original bids is correct. We assume that the mapped bids are sorted in a non-increasing order:

$$\Gamma : \hat{b}_1 \geq \hat{b}_2 \geq \cdots \geq \hat{b}_l,$$

where $\hat{b}_1 = \hat{b}_{max}$ and $\hat{b}_2 = \hat{b}_{sec}$. For ordering verification, the $\mathcal{P}$ should prove that the mapped payment $\hat{b}_{sec}$ is equal to or less than the winner's mapped bid $\hat{b}_{max}$, and $\hat{b}_{sec}$ is equal to or larger than the mapped bids except $\hat{b}_{max}$. Since the l mapped bids are all in the range $[1, 2^t]$, $2^t \leq n/2$, the correctness of the order Γ can be verified by applying PPIC protocol over the $l-1$ pairwise comparisons, *i.e.*, $< \hat{b}_{sec}, \hat{b}_i >, \forall\ i \neq sec$.

We describe the verification for the $l-2$ comparisons $< \hat{b}_{sec}, \hat{b}_i >, \forall\ 3 \leq i \leq l$ in Figure 4. The relation of $\hat{b}_{max}$ and $\hat{b}_{sec}$ can be verified in a similar way. In order to verify the relation $< \hat{b}_{sec}, \hat{b}_i >$, the verifier $\mathcal{V}$ chooses a certain test set TS_i, and sends it with the index i to the prover $\mathcal{P}$. The prover $\mathcal{P}$ then constructs the set of ciphertext $\mathbb{C}_i = \{C_{t_1}, C_{t_2} \cdots, C_{t_k}\}$

[2]Using her private key ϕ, the auctioneer can recover the random value r_{sec}, which is generated by the ad network.

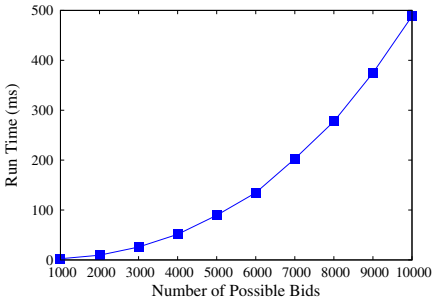

Fig. 5. Computation overhead of mapped bid generation for each bidder.

such that $\hat{b}_{sec} - \hat{b}_i = 2^{t_1} + 2^{t_2} + \cdots + 2^{t_k}$, and calculates a new random value r_i^*. Both the set $\mathbb{C}_i$ and the random value r_i^* are transmitted back to the verifier $\mathcal{V}$, who then calculates the value of $G_i^{-1} \times C_{t_1} \times \cdots \times C_{t_k}$ and $E_n(0.r_i^*)$ to decide whether to accept the ordering verification or not.

Step 3: Patching Verification

If the first two verification steps does not pass, the auctioneer is accused of cheating unless she can provide the evidence that the fault of the outcome is caused by some ad networks. The auctioneer uses her private key to decrypt the doubly encrypted bids on the CBB to obtain all mapped bids, and then re-sorts these mapped bids to check the correctness of the internal auction outcome in each ad network. By doing this, the auctioneer can catch the cheating ad networks.

VI. Evaluation Results

In this section, we show the evaluation results of ERA in terms of computation, storage and communication overhead.

Simulation Setting: We have implemented ERA using network simulation. The range of possible bids is from \$0.01 to \$100 with \$0.01 increment, which is typically the smallest billable unit in today's ad exchanges [15]. The maximum *mapped* bid is set as 2^{32} in OPES scheme and PPIC protocol. In OT, the length of the prime q is set as 1024-bit, and the size of ξ_i is bounded in 32-bit. The PHES is implemented using an open library [16], in which the length of key is set as 1024-bit. The running environment is a standard 64-bit Ubuntu 14.04 Linux operating system with Intel(R) Core(TM) $i5$ $3.10GHz$.

Computation Overhead: We now show the computation overhead of three important components in ERA, *i.e.*, Mapped Bid Generation, Auction Execution and Verification.

1) Mapped Bid Generation: By averaging the evaluation of 1000 simulation instances, in Figure 5, we plot the computation overhead of the agent for generating mapped bid for one bidder, when the number of possible bids increases from 1000 to 10000. We can see that the computation overhead increases linearly with the number of possible bids, and achieves around 500ms for 10000 possible bids. This is because the computation overhead mainly comes from running the OT, in which the agent should calculate n messages ξ_i, $1 \leq i \leq n$.

2) Auction Execution: We measure the metrics of auction latency and auction scale to understand the computation overhead of the auction execution in ERA.

Figure 6 shows the auction latency of ERA when the number of bidders ranges from 2×10^5 to 10×10^5 with an increment of 2×10^5, and the number of ad networks can be chosen as 60, 80 and 100. We can see that the auction

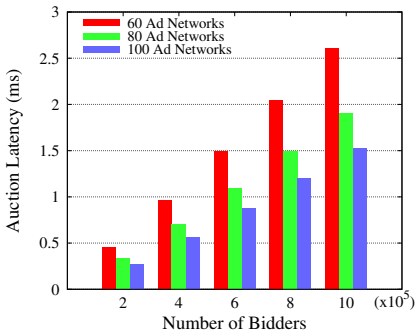

Fig. 6. Auction latency with varying number of bidders and ad networks.

Verification Overhead (s)

# bidders ($\times 10^4$)	Preparation		Operation	
	Test Set Generation	Commitment Generation	Ordering	Patching
2	1.00×10^4	316.85	132.26	117.08
4	2.00×10^4	631.52	265.69	233.58
6	3.00×10^4	946.46	393.86	349.78
8	4.00×10^4	1263.91	527.03	466.04
10	5.00×10^4	1577.34	656.84	582.61

Fig. 7. Computation overhead of verification.

latency of ERA increases with the growing number of bidders when the number of ad networks is fixed, while it decreases when the number of bidders is fixed and the number of ad networks increases. The results demonstrate that ERA can indeed reduce the auction latency by introducing a proper number of intermediary ad networks, especially in large scale auction: we set the upper bound of the auction latency as 10ms and find that the ERA can support more than 5 million bidders.

3) Verification: We now investigate the computation overhead of the verification, which consists of *Preparation* phase and *Operation* phase. In this set of simulation, the maximum number of bidders and the number of ad networks are set as 10^5 and 100, respectively. Figure 7 plots the evaluation results.

The Preparation phase is divided into two parts: the generation of test sets by the auctioneer and the generation of commitments by ad networks. In Figure 7, we can see that the auctioneer has higher computation overhead (about $31.70\times$) than that of one ad network. This is because the auctioneer should calculate $l - 1$ test sets for the verification with l bidders, around 0.5s for each test set generation, while one ad network only generate commitments for her bidder members.

The computation overhead of Operation phase is mainly from the ordering verification and patching verification[3]. From Figure 7, we can see that the computation overhead of ordering verification and patching verification increase when there are more bidders in the auction. When the number of bidders is 10×10^4, the computation overhead of ordering verification and patching verification is 656.84s and 582.61s, respectively.

We can also see from Figure 6 and Figure 7 that the computation overhead of verification is higher that that of auction execution. As the verification can be conducted offline, the running time constrain on verification is not so strict. Therefore, the time consumption of verification is tolerant when ERA is integrated into the practical ad exchanges.

[3]The computation overhead of the payment verification is omitted here because it is extremely lower than the other two steps.

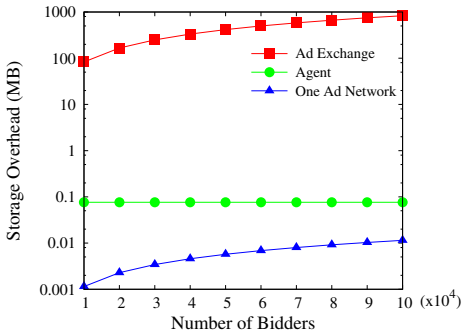

Fig. 8. Storage Overhead of ERA.

Storage and Communication Overhead: Figure 8 shows the storage overhead of the ad exchange, the agent and one ad network. The number of possible bids is fixed at 10000, the maximum number of bidders is 10^5 and the number of ad networks is 100. We can see that the storage overhead of the ad exchange and the ad network grow linearly with the number of bidders, while the storage overhead of the agent remains unchanged. The reason is that the storage overhead of the agent is mainly from storing the original and mapped bids, which are independent of the number of bidders. We also find that the ad exchange costs much more space than the ad network, because the ad exchange has to maintain a Certificate Bulletin Board, on which all test sets and commitments are posted, while each ad network only stores the mapped bids, identities and ad tags of her bidder members.

We also measure the communication overhead of ERA, which is mainly caused by the interactions in OT and ordering verification. In OT, each bidder receives message ξ with 32-bits for each of the 10000 possible bids, while the prover and the verifier need to transfer about 800MB data to perform the ordering verification when the number of bidder is 10^5.

VII. Related Work

In this section, we briefly review the related work. Inspired by early works [17], [18], various privacy-preserving and verifiable auction mechanisms have been extensively studied. The existing works mainly fall into the following three categories with different auction models.

No Auctioneer: Bidders themselves jointly determine the auction outcomes by using *secure multiparty computation* [19], [20]. These mechanisms guaranteed the privacy and correctness of the auction, but induced high computation and communication complexity. Therefore, these mechanisms are inefficient and impractical in the scenario of ad exchanges.

One Auctioneer: One auctioneer is responsible for conducting and calculating the auction. In [5], a method based on the Paillier encryption scheme was proposed to achieve verification. In [6], Rabin *et al.* proposed a novel secure and efficient method for validating the correctness of the auction outcome. However, these mechanisms required that the bid information should be revealed to the auctioneer.

Additional Third Party: An additional third party is introduced to cooperate with the auctioneer to run the auction. The scheme proposed by Naor *et al.* in [21] constructed a Boolean Circuit that calculated the auction outcomes for any given set of bid. Based on RSA, Juels and Szydlo proposed a privacy-preserving auction mechanism with a reasonable computational complexity [22]. Unfortunately, these works only guaranteed the privacy of bids, but did not consider the problem of verification. ERA belongs to this category, and moves forwards to solve the problem of verifiability and privacy-preservation at the same time.

The most relevant work is paper [7], in which an online verifiable auction mechanism for ad exchanges was proposed. However, the interaction between the ad exchange and bidders was too much, and the bid privacy was not considered. ERA is the first efficient, privacy-preserving and verifiable online auction mechanism for ad exchanges.

VIII. Conclusion

In this paper, we have proposed the first efficient, privacy-preserving, and verifiable auction mechanism for ad exchanges, namely ERA. In ERA, the outcomes of ad auctions can be calculated and verified to be correct, while not disclosing private information of advertisers. We have implemented ERA and extensively evaluated its performance. Evaluation results have demonstrated that ERA satisfies the properties of low-latency and large-scale for ad exchanges.

References

[1] DoubleClick, http://www.google.com/doubleclick/.
[2] Right Media, https://advertising.yahoo.com/Publishers/index.htm.
[3] OpenX, http://openx.com/product/ad-exchange/.
[4] DoubleClick, "Google white paper: The arrival of real-time bidding and what it means for media buyers," http://static.googleusercontent.com/media/www.google.com/zh-CN/us/doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf.
[5] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. Thorpe, "Practical secrecy-preserving, verifiably correct and trustworthy auctions," *Electronic Commerce Research and Applications*, vol. 7, no. 3, pp. 294–312, 2008.
[6] M. Rabin, Y. Mansour, S. Muthukrishnan, and M. Yung, "Strictly-black-box zero-knowledge and efficient validation of financial transactions," in *ICALP*, 2012.
[7] S. Angel and M. Walfish, "Verifiable auctions for online ad exchanges," in *SIGCOMM*, 2013.
[8] Q. Huang, Y. Tao, and F. Wu, "SPRING: A strategy-proof and privacy preserving spectrum auction mechanism," in *INFOCOM*, 2013.
[9] DoubleClick Ad Exchange Real-Time Bidding Protocol, https://developers.google.com/ad-exchange/rtb/peer-guide.
[10] Y. Mansour, S. Muthukrishnan, and N. Nisan, "Doubleclick ad exchange auction," *arXiv preprint arXiv:1204.0535*, 2012.
[11] S. Muthukrishnan, "Ad exchanges: Research issues," in *WINE*, 2009.
[12] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *SIGMOD*, 2004.
[13] W.-G. Tzeng, "Efficient 1-out-n oblivious transfer schemes," in *PKC*, 2002.
[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999.
[15] Real-Time Bidding Protocol Buffer v.59, https://developers.google.com/ad-exchange/rtb/downloads/realtime-bidding-proto.
[16] Advanced Crypto Software Collection, http://hms.isi.jhu.edu/acsc/libpaillier/.
[17] H. Nurmi and A. Salomaa, "Cryptographic protocols for vickrey auctions," *Group Decision and Negotiation*, vol. 2, no. 4, pp. 363–373, 1993.
[18] M. Franklin and M. Reiter, "The design and implementation of a secure auction service," in *S&P*, 1995.
[19] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *STOC*, 1988.
[20] F. Brandt, "How to obtain full privacy in auctions," *International Journal of Information Security*, vol. 5, no. 4, pp. 201–216, 2006.
[21] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *EC*, 1999.
[22] A. Juels and M. Szydlo, "A two-server, sealed-bid auction protocol," in *FC*, 2003.