SPA: A Secure and Private Auction Framework for Decentralized Online Social Networks

Arun Thapa, *Member, IEEE*, Weixian Liao, *Student Member, IEEE*, Ming Li, *Member, IEEE*, Pan Li, *Member, IEEE*, and Jinyuan Sun, *Member, IEEE*

Abstract—The security and privacy threats on e-commerce have attracted intensive attention recently. The explosive growth of online social networks (OSNs) has made them potential new great marketplaces for e-commerce, which, however, raise serious security and privacyconcerns. This is mainly due to the centralized system architecture where the service provider knows all users' private data and becomes the single point of failure. To this end, we propose a secure and private auction framework, called SPA, for decentralized online social networks (DOSNs). SPA consists of three phases: identity initiation, buyer-seller matching, and private auction. It requires no trust among the participants but can provide security, privacy, authenticity, non-repudiation, and correctness for the auctions. We analyze the computation and communication complexities of the proposed private auction scheme, which are $O(n + K)$ for each node where n is the number of bidders and K is the number of pricing points. In contrast, those of previous auction schemes are $O(nK)$ at best. The storage complexity is significantly lower than before as well. Security and privacy of SPA are also analyzed. Extensive experiments are conducted to validate the efficiency of SPA.

Index Terms—Distributed online social networks, auction, security, privacy

1 Introduction

E-COMMERCE has exploded in the last decade. It enables the buying and selling of goods and services via electronic channels, primarily on the Internet, and has become an indispensable part of our daily lives. On the other hand, the security and privacy threats on e-commerce are also on the rise. Particularly, attackers may compromise the e-commerce service providers' servers and steal consumers' confidential information like their personal data and buying/selling history. Such information can be used for many privacy intrusive purposes like directed marketing, user profiling. Besides, in auction based e-commerce like eBay, users' bidding statistics reveal their valuations for the items being auctioned and the server can utilize the statistical information to increase its financial gain in future auctions of similar items.

The explosive growth of online social networks (OSNs) over the past several years has dramatically changed the way that information is produced and propagated in the world, and has made OSNs potential new great marketplaces for e-commerce. In particular, in OSNs, the traditional unidirectional information flows, where information (e.g., breaking news, events) flows from a source (e.g., news organizations) to the consumers, are being replaced with multidirectional flows, where the ordinary users of OSNs (like Facebook, Twitter, Youtube) are both the sources and the consumers of the information. This shift in information flow paradigm has been proven to be powerful in strengthening the connections among users, and hence can facilitate large-scale e-commerce. However, OSNs also raise serious concerns about users' privacy since the traditional OSNs store users' private personal, historical, and relationship information. For instance, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered[1]. Facebook, Apple, Microsoft were under similar attacks in February 2013 [2]. Thus, auctions in traditional OSNs will inevitably lead to security and privacy problems. Recently, decentralized online social networks (DOSNs) like [3], [4], and Dispora [5], attract users' intensive attention, where users own and store their private data on their own computers or on the servers they trust. In this paper, we exploit the rich connectivity and the distributed system architecture of DOSNs to develop a secure and private auction framework called SPA, which requires no trust among the participants for the privacy and correctness of auction outcomes.

The proposed secure and private auction framework called SPA is based on Vickrey auction [6]. Specifically, auctions are frequently employed for determining resource allocations and selling prices. Many auction schemes have been proposed in the literature (please refer to [7] for a review on auction protocols). Vickrey auction adopted in this study, also known as second price auction, is a sealed-bid auction, in which bidders send their sealed bids to a trusted auctioneer. The winning bidder of such an auction is the highest bidder and is charged the second highest bidding price. Due to this pricing mechanism, it has been

- *A. Thapa is with the Department of Electrical Engineering, Tuskegee University, Tuskegee, AL 36088. E-mail: athapa@mytu.tuskegee.edu.*
- *W. Liao and P. Li are with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106. E-mail: {weixian.liao, lipan}@case.edu.*
- *M. Li is with the Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557. E-mail: mingli@unr.edu.*
- *J. Sun is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996. E-mail: jysun@eecs.utk.edu.*

Manuscript received 23 Feb. 2015; revised 5 Sept. 2015; accepted 18 Oct. 2015. Date of publication 26 Oct. 2015; date of current version 20 July 2016. Recommended for acceptance by S. Yu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2015.2494009

proved [6] that the bidders in Vickrey auction have the highest pay-offs when they bid their true valuations of the goods being auctioned and hence have no incentives to bid strategically, i.e., truthfulness. Besides, as all bidders bid their true valuations, they are guaranteed to have non-negative payoffs in the auction, i.e., individual rationality. Vickrey auction has attracted a lot of research interests because of these interesting properties. However, it has rarely been used in practice mainly due to the following reasons. First, bidders hesitate to reveal their true valuations to the auctioneer who may exploit such information for privacy invasive purposes as mentioned before. Second, if the auctioneer is dishonest, it can cheat the winning bidder by creating an artificial second highest bid with bid price just a little bit lower than the highest bid since it knows all users' bidding prices.

The remedy to the problem of limited usage of Vickrey auction is to ensure each individual bidder that first, his/her bidding privacy (e.g., identities, bidding prices, histories) is preserved regardless of the number of possible other colluding bidders, and second, there is no need to bank on the honesty of the auctioneer for the correctness of auction outcomes. Our auction framework SPA consists of three phases: identity initiation, buyer-seller matching, and private auction. It can guarantee users' privacy and auction correctness, while only revealing minimum information, i.e., the winning price and the winning bidder's public pseudo identity.

Specifically, in the identity initiation phase, each user who would like to participate in an auction obtains a public and a private pseudo identity (ID) from a Trusted Third Party (TTP). In the "buyer-seller matching" phase, we develop an efficient algorithm to enable the users interested in buying/selling item(s) to distribute their advertisements and match each other. In particular, the users utilize both social links and the underlying Distributed Hash Table (DHT) links to route their advertisement messages to a randomly chosen user, called *the bridge node*, who then helps match the buyers and sellers. In the private auction phase, we design an efficient bidder-resolved private auction protocol. Particularly, bidders use their public pseudo IDs to get authenticated through non-interactive zero knowledge (NIZK) proofs. Thus, their ID privacy can be protected. Then, the authenticated bidders collaboratively construct a public encryption key based on a distributed exponential Elgamal cryptosystem. They send their encrypted bidding vectors to the bridge node, which can thus be protected against attacks like colluding. The bidders also sign their encrypted bids with an anonymous signature scheme so that the bids are non-repudiable. After that, the bridge node calculates the winning price under public scrutiny, without revealing any bidder's bidding vector. The winning bidder can finally be determined without revealing his/her bidding vector.

Besides, we analyze the computation and communication complexities of the proposed private auction scheme, which are $O(n + K)$ for each node where n is the number of bidders and K is the number of pricing points, while those of previous auction schemes (without the winning bidder identification process) like [8] are $O(nK)$ at best. The storage complexity is shown to be significantly lower than before as well. Security and privacy of SPA are also investigated.

We summarize our major contributions in this work as follows.

- To the best of our knowledge, the proposed auction framework SPA is the first attempt to address auctions in DOSNs.
- We design a distributed private buyer-seller matching scheme to enable auctions in DOSNs where no central server or auctioneer is available. The communication cost is $O(\log\ n)$ where n is the network size.
- We develop a fully private distributed auction protocol, whose computation and communication complexities are both $O(n + K)$ for each node where K is the dimension of a seller's price vector. In contrast, the most efficient existing distributed private auctions like [8] are not fully private and have higher complexities of $O(nK)$. Besides, the storage complexity of a bidder is $O(n + K)$ and that of a bridge node is $O(n^2 + nK)$ in our scheme, while the storage complexity of a bidder in [8] is $O(n^2K)$.
- SPA can provide security, privacy, authenticity, nonrepudiation, and correctness for the auctions. Both completeness and soundness are proved.
- We evaluate the performance of our auction protocol and show that it outperforms previous auction schemes significantly in terms of computation, communication, and storage costs.

2 Related Work

2.1 Decentralized Online Social Networks (DOSNs)

Currently most OSN service providers like Twitter, Facebook, Google$^+$ use central servers to store users' private data, which, however, raises great concerns about users' privacy. For example, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered[1]. Facebook, Apple, Microsoft were under similar attacks in February 2013[2]. Besides, users may not have connectivity to the server all the time. Thus, recently the research on decentralized online social networks has attracted intense attention. There have been several proposals for DOSNs [3], [4] in the literature, and some (e.g., [5]) have taken off and are increasingly popular. Specifically, PeerSoN [3] has a two-layer architecture where peers (with social relationships) communicate with each other using a distributed harsh table based lookup service. Safebook [4] proposes to build concentric rings of nodes around each node, based on the degree of trust among nodes, to provide trusted data storage, profile data retrieval, and communication obfuscation through indirection. Diaspora [5] is a popular DOSN, where users can host their data on their own computers or in the servers they trust. There have been a few works such as [9], [10] which study private friendship matching protocols in DOSNs.

2.2 Distributed Hash Table

Distributed Hash Tables have been used as an efficient lookup scheme in peer-to-peer systems. In particular, each peer is assigned a unique ID and keeps a record of a small fraction (usually $\log n$, n is the network size) of the nodes

in the network, which are determined by certain specific algorithm to guarantee efficient lookup service. For instance, Chord [11] uses consistent hashing [12] to assign node ID and to map a given key or data to a specific node. Other notable and widely referred DHT schemes include Kamedia [13], CAN [14], Pastry [15], and Tapestry [16]. Recently, DHT systems are designed while addressing security (especially against sybil attack) [17], [18] and anonymity [19], [20] issues. Our proposed private auction protocol utilizes DHTs (based on the Chord [11] protocol) and social links for efficient advertisement distribution and buyer seller matching.

2.3 Cryptographic Auction Protocols

The necessity of providing security and privacy for the participants of an auction has led to intensive research activities on cryptographic auction protocols. Yao's garbled circuit [21] based multi-party computation (MPC) is tailored to design secure auction [22], [23] with multiple (two) auctioneers under a passive adversary model, where the two auctioneers are assumed not to collude with each other. Similar threshold based MPC auction protocols [24], [25], [26] rely on multiple auctioneers and are secure as long as there are no more than a certain fraction of the total number of auctioneers colluding with each other. Lipmaa et al. [27] employ homomorphic encryption to design a secure Vickrey auction scheme, where the semi-honest auctioneer therein knows all users' bidding prices.

A few approaches have been developed to improve the privacy in auction. Brandt [28] proposes a private auction scheme, in which the bidders engage in cryptographic protocols and jointly compute the outcome of an auction, and later improves the protocol in [8]. In such auctions, collusion between any numbers of bidders but the total number of bidders is insufficient to compromise the auction privacy. The computation complexity and communication complexity of the Vickrey based auction scheme in [8] are both $O(nK)$, where n is the network size and K is the number of possible bidding values. However, Dreier et al. [29] show that the bid privacy in [8] can be breached if interactive Zero Knowledge Proof (ZKPs) are used. More importantly, even if Non-Interactive ZKPs (NIZKPs) are used, due to the lack of authentication, malicious bidders can mount a collaborative attack to breach the privacy of a targeted bidder. In contrast, our proposed auction protocol is a fully private auction protocol with both communication and computation complexities being $O(K)$ ($O(n+K)$ if with the winner identification process), and hence much more efficient.

3 Problem Formulation

3.1 System Model

We consider a Decentralized Online Social Network consisting of three layers as shown in Fig. 1. The OSN layer at the top includes social network users along with the relationships among them. Particularly, an OSN can be defined as a graph $G = (V, E)$, where the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ represent nodes (users) in the network and the set of undirected edges $E = \{e_{ij}\}$ $(1 \leq i, j \leq n, i \neq j)$ represent the friendships or social ties among the nodes. In the absence of a central server, the Distributed Hash Table layer provides the peer-to-peer lookup functionality in the DOSN, which we utilize to distribute the advertisement messages of buyers and sellers. Unlike OSN links, the DHT links are directed. We build the DHT links based on the Chord DHT protocol [11], which will be briefly introduced in Section 4.3. Each node i has a "Chord ID" at the DHT layer denoted by u_i. The actual communications take place at the Internet layer at the bottom. Each user in the DOSN is a potential buyer/seller and has a public page where, if the user is selected as a bridge node (see Section 5.3.2), the information on the item for sale, the encrypted bids from the buyers, and auction related computations are hosted. A fixed time period (e.g., a day or a week) is determined for each auction during which buyers need to submit their bids to the bridge node.

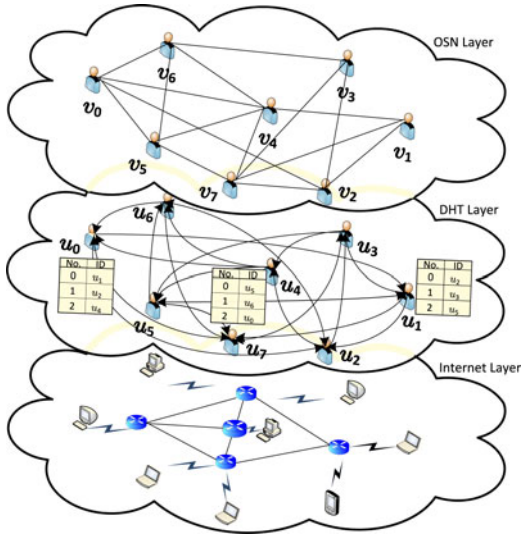

Fig. 1. A Decentralized Online Social Network consisting of three layers.

3.2 Adversary Model

The adversaries taken into consideration in this work are mainly the participants in the auctions, such as bidders, sellers, auctioneers (i.e., bridge nodes). These participants may be interested in bidders' bidding prices in order to enhance their financial gain in the current auction or in the future auctions. For example, if a seller knows the bidding statistics of an item, he/she can exploit that information in future auctions to maximize his/her own financial gain. Similarly, bidders' identities can also be of interest to the adversaries, e.g., for targeted advertisement. Besides, the adversaries include the malicious bidders who may send bogus bidding values just to hinder the outcome of the auction. Note that we do not consider the possible adversaries at the DHT layer and Internet layer who may try to disrupt the auction by replaying or dropping the auction messages. There have been several works addressing such attacks [17], [18] and protecting privacy at the DHT layer [19], [20]. While our scheme can be easily built on these DHT protocols to provide security and privacy at the DHT and Internet layers, in this work we employ a widely referred DHT protocol [11] and mainly focus on the possible adversaries in our auction scheme as mentioned above.

3.3 Design Goals

- *Security*: The proposed system may be under various attacks such as impersonation, colluding. Our goal is to protect malicious attackers from disrupting auction outcomes by indulging in the intermediate computations in the protocol.
- *Privacy*: In the proposed Vickrey based auction scheme, where buyers bid with their true valuations of the items being auctioned, the bid privacy is important to the buyers. All buyers' and sellers' identities should be protected too. Our goal is to provide privacy (e.g., bidding prices, identities) for users during and after the auctions and make sure that users' buying/selling histories cannot be tracked.
- *Authenticity and non-repudiation*: Since bidders' identities are hidden during the auctions to protect their privacy, we need to validate that the bidders are legal users in the system. We also need to verify that the bidding values are legit as they are unknown to the auctioneer. Besides, we aim to achieve non-repudiation in the auctions, i.e., guarantee that bidders cannot deny their bidding. Thus, our goal is to ensure authenticity and non-repudiation in the auctions.
- *Efficiency*: Social networks usually host a large number of users, all of whom can engage in auctions. Therefore, the communication, computation, and storage complexities of auction schemes should not increase rapidly with the number of participating users. Our goal is to obtain high efficiency in the auctions in terms of communication, computation, and storage complexities.

4 Preliminaries

4.1 ElGamal Cryptosystem

ElGamal cryptosystem [30] is a semantically secure homomorphic cryptosystem based on the intractability of the discrete logarithm problem in finite fields. Please refer to the Appendix A, available in the online supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2015.2494009, for details.

We utilize ElGamal cryptosystem in a distributed manner since our system has very strong security and privacy requirements. In particular, users can encrypt and decrypt messages in a distributed fashion, and hence protect their privacy against potential collusion among any number of users. In the following, we describe how distributed encryption and decryption of ElGamal cryptosystem can be carried out.

- *Distributed key generation:* Let p and q be two large strong prime numbers such that $p = 2q + 1$. Let $\mathbb{G}_q$ denote a sufficiently large multiplicative subgroup of $\mathbb{Z}_p^*$ of order q. Each user v_i participating in the distributed key generation selects a random $x_i \in \mathbb{G}_q$ as his/her private key and publishes $y_i = g^{x_i} \bmod p$ as his/her public key. The public key for distributed encryption is then $y = \prod_{i=1}^{n} y_i = g^{\sum_{i=1}^{n} x_i} \bmod p$.
- *Distributed encryption:* A user can use the public key y to encrypt a message m. The resulting ciphertext is $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$.
- *Distributed decryption:* All the users who participated in public key generation need to cooperate to decrypt the encrypted message. Specifically, if $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ is the encrypted message, then each user publishes $\alpha_i = \alpha^{x_i}$. The original message can be recovered by any user by computing $\frac{\beta}{\prod_{i=1}^{n} \alpha_i} = \frac{g^m y^r}{g^{(\sum_{i=1}^{n} x_i) r}} = g^m$.

4.2 Zero Knowledge Proofs

The Zero Knowledge Proof, introduced by Goldwasser, Micali and Rackoff (GMR) [31], is an important tool in cryptography. A prover can use a ZKP protocol to prove the possession of certain information to a verifier without revealing the very information. The absence of a trusted central authority in a DOSN makes the network inherently vulnerable to malicious users who aim to fulfill their malicious intents and do not follow the proposed auction protocol. Besides, the strong privacy requirement in our schemes necessitates preserving bidders' anonymity and their bidding price privacy, which further complicates the authenticity and enforcement of correct protocol execution by all the participants. In order to ensure the bidders follow the proposed auction protocol correctly, we require all bidders (provers) to prove to a bridge node (verifier, see Section 5.3.2 for details) using ZKPs in different steps of the protocol. We describe several ZKPs we will use as follows.

4.2.1 Proof of Knowledge of a Discrete Logarithm

A bidder can use the zero knowledge protocol [32] to prove knowledge of secret x such that $y = g^x$ to a verifier (a bridge node) who knows y and g. The protocol is detailed in Appendix B.1, available in the online supplemental material.

4.2.2 Proof of Equality of Two Discrete Logarithms

When a prover (a bidder) needs to prove that two ciphertexts are computed using the same private key to a verifier (bridge node), who does not know the private key, the prover utilizes zero knowledge proof protocol [33] (detailed in Appendix B.2, available in the online supplemental material) to prove in zero knowledge.

4.2.3 Proof that an Encrypted Value Decrypts to Either 1 Or 0

In our private auction scheme (Section 5.3), a bidder prepares a bidding vector by encrypting each element (either 0 or 1) separately. While the actual bidding price (and bidding vector) remains private to the bidder throughout the auction, it is necessary to make sure the bidding vector is prepared correctly in order to deter any malicious bidder's attempt to disrupt the protocol. A bidder can use the protocol proposed by Cramer et al. [34] to prove to the bridge node that his/her bidding vector is composed of encryptions of $m \in \{0, 1\}$. We detail the protocol in Appendix B.3, available in the online supplemental material.

4.3 Distributed Hash Table Overlay

Chord [11] is a distributed lookup protocol for mapping (and retrieving) a given key (or data) onto a specific node in a distributed and scalable manner. In particular, each node is assigned a k-bit identifier, called "Chord ID", using *consistent hashing* [12]. The nodes' identifiers are arranged in the form of a modulo-2^k one-dimensional identifier circle known as *chord ring*. Each key is assigned to a peer node whose Chord ID is equal to or immediately next to the hash value of the key. In order to provide an efficient and scalable lookup service for the key, each node in the system stores information about a small fraction (i.e., $\log n$ where n is the network size) of all the nodes in what is known as the *finger table*. The ith $(0 \leq i \leq (\log n - 1))$ element in the finger table of a node contains the identifier and the address (IP address, port number, etc.) of the node which is at 2^i distance apart in the chord ring. Fig. 1 shows the finger tables of nodes u_0, u_1, and u_4. The outgoing arrows from a node in Fig. 1 go to the nodes which are included in the current node's finger table. In order to look up a given key (or data) in the network, the source node hashes the key and searches in its finger tables. If it matches the Chord ID of certain node in its table, the source node forwards the request to that node. Otherwise, it forwards the request to the node in its finger table which is closest to the hash value of the key in the chord ring. The same procedure follows at the new node until the request reaches the node which has the key. Both the lookup communication cost and the storage cost scale as $O(\log n)$. In this study, we adapt the Chord DHT to design an efficient advertisement distribution scheme (Section 5.2.1) utilizing both DHT and social links.

5 A Secure and Private Auction Framework: SPA

The proposed secure and private auction framework for DOSNs, named SPA, consists of three phases. In the first phase called "*Identity Initiation*", all nodes that would like to participate in auctions obtain public/private pseudo IDs [35] from a Trusted Third Party (TTP). In the second phase called "*Buyer-Seller Matching*", the nodes that are interested in buying/selling item(s) distribute their intents through the social network via both DHT and social links during a specific time interval. For a particular item F_i[1] at a specific time interval T_k, a node, called *the bridge node* $v_{B^{i,k}}$, is chosen to match the sellers and buyers. In the third phase called "*Private Auction*", the bidders send sealed (encrypted) bids to the bridge node and the bridge node helps execute the auction. In what follows, we detail these three phases respectively.

5.1 Phase I: Identity Initiation

In order to be able to participate in an auction while preserving the ID privacy, each node requests a pair of public/private pseudo IDs from a TTP. Specifically, the TTP chooses two large primes $\tilde{p}$ and $\tilde{q}$ and publishes $N = \tilde{p}\tilde{q}$ along with a generator $\tilde{g}$ of a sufficiently large subgroup of Z_N^*. When a user v_i needs a public pseudo ID, it sends a signed request together with its certified public key to the TTP. For each such request, the TTP chooses a random prime ρ_i such that $gcd(\rho_i, \lambda(N)) = 1$, where $\lambda(\cdot)$ is the Carmichael function [36]. ρ_i works as a public pseudo ID for user v_i. The TTP also computes a private pseudo ID $s_i = \tilde{g}^{1/\rho_i} = \tilde{g}^{d_i}$, where $\rho_i d_i = 1 \bmod \phi(N)$ and $\phi(N) = (p-1)(q-1)$ is the Eular's totient function. Note that all the calculations are modulo-N unless specified otherwise. The TTP signs the public pseudo ID, i.e., $S_i = K_D^{TTP}(\rho_i)$, and generates a certificate $C_i = (S_i, \rho_i)$, where K_D^{TTP} is the private key of the TTP. The private pseudo ID (s_i) and the certificate (C_i) are then securely delivered (encrypted with user v_i's public key) to user v_i.

TABLE 1
The Format of Advertisement Messages

$MessageID$	$SrcID$	$H(F_i\|\|T_k)$	B/S	$Payload$

5.2 Phase II: Buyer-Seller Matching

5.2.1 Advertisement Distribution

The absence of a central server in a DOSN necessitates the design of a distributed scheme to ensure the advertisement of a seller/buyer of an item reaches the right potential buyers/sellers. One naive solution is to broadcast the advertisements of sellers/buyers throughout the network. However, this will cause a serious message flooding in the network. For example, in a DOSN if each user has 100 friends on average, then the number of broadcast messages from a single user can, in the worst case, lead to 100 million messages in just 4 hops. This kind of broadcast flooding will inevitably congest the network and is not suitable for any practical application in a large-scale DOSN.

In contrast, SPA features a distributed advertisement distribution algorithm. Specifically, the sellers and the potential buyers distribute their intents through unicast advertisement messages utilizing both DHT and social links. All the advertisement messages for an item F_i in a time interval T_k intersect at the same bridge node $v_{B^{i,k}}$. The format of advertisement messages is shown in Table 1. $MessageID$ is a random ID chosen by the source of the advertisement (a buyer or a seller), $SrcID$ changes in every hop and is the Chord ID of the current node, $H(\cdot)$ is a public hash function, B/S denotes whether the source is a buyer or a seller, and the $Payload$ of a seller's advertisement message contains the details about the item F_i, such as the price vector defined by the seller (see Section 5.3 for details) and others like shipping information/estimates. The bridge node is determined based on the hash value $a_{ik} = H(F_i||T_k)$. Particularly, the node $v_{B^{i,k}}$, whose Chord ID is either equal to or immediately next (in clockwise, i.e., increasing order) to a_{ik} in the chord ring, serves as the bridge node for the item F_i during the time interval T_k. The time interval T_k is the time period, e.g., a day or a week, during which the auction for the item takes place. This timestamp T_k serves two important purposes: first, it puts a time limit on each trade, and more importantly, second, it randomly changes the

1. The item F_i is drawn from a universal set of items or goods $\mathbb{F}$, $F_i \in \mathbb{F}$, such that same name is used for an item by all the participants in an auction. More importantly, having a universal set of items prevents the trade of illicit goods that many fear may happen in a private marketplace.

bridge node in each time interval so that not a single node has to bear the computation and storage overhead of being a bridge node all the time.

The advertisement messages can be delivered to the bridge node as follows. Note that in addition to a finger table, we let each node keep the Chord ID and the address (IP address, port number, etc.) of its predecessor and of its successor on the Chord ring, as well as those of its friends and their predecessors on the Chord ring. When receiving an advertisement message, each node first checks to see if it itself is the bridge node, i.e., if a_{ik} is equal to its Chord ID or between its predecessor's Chord ID and its Chord ID. If so, this node sends an acknowledgement message back to the sender of this advertisement message. Otherwise, it stores the $MessageID$ and $SrcID$ (along with the IP address of $SrcID$) of this message. The node then checks if its successor on the Chord ring is the bridge node, i.e., if a_{ik} is equal to its successor's Chord ID or between its own Chord ID and its successor's Chord ID. If so, it forwards the message to its successor. Otherwise, the node checks if any of the nodes in its finger table is the bridge node, i.e., if a_{ik} is equal to any node's Chord ID. If so, it forwards the message to that node. Otherwise, the node checks if any of its friends is the bridge nodes, i.e., if a_{ik} is equal to any friend's Chord ID or between any friend's Chord ID and its predecessor's Chord ID. If so, it forwards the message to that friend. Otherwise, it forwards the message to the node that precedes and is closest to a_{ik}, utilizing both its finger table and its friends list. The procedure continues until the message reaches the bridge node. The above advertisement distribution scheme is detailed in Algorithm 1 in Appendix C, available in the online supplemental material.

For example, as shown in Fig. 1, assume that u_0 is the bridge node for an item F_i in time interval T_k, i.e., $a_{ik} \in (predecessor(u_0), u_0]$ where $predecessor(u_0)$ is the Chord ID of the predecessor of u_0 on the Chord ring. Suppose that node u_1 currently has the advertisement message. We can see that node u_1, its successor u_2, and the nodes in its finger table, and its friends are not the bridge node. In this case, node u_1 forwards the message to the closest preceding node to the bridge node in the Chord ring, considering all the nodes in the finger table (u_2, u_3, u_5) and on its friend list (u_2, u_4, u_7), i.e., node u_7 in this example. Node u_7 then finds that its successor, i.e., node u_0, is the bridge node, and forwards the advertisement message to it. Ultimately, the bridge node receives the advertisement messages from all interested participants (seller and buyers) and starts the matching process.

5.2.2 Acknowledgement (ACK) Message Distribution

Every time a bridge node receives an advertisement message, it sends an ACK message back to the source node of the message reversely along the route that the message was delivered on. The format of ACK messages are shown in Table 2. In particular, each node on the route kept a record of $MessageID$ and $SrcID$ while forwarding the advertisement message to the bridge node. When forwarding the ACK message, the $DestID$ field is set to $SrcID$ of the corresponding advertisement message. The payload of the ACK message contains the address (e.g., IP address and the port number), of the bridge node where the price vectors of all the sellers are accessible and all the auction related computation (see Section 5.3) takes place under the scrutiny of all the bidders in the future. Once a buyer/seller node receives the ACK message, it connects to the bridge node via the Internet layer[2] to access the information provided by the sellers in their advertisement messages, and decides which one of the many sellers' items it wants to bid for.

TABLE 2
The Format of ACK Messages

$MessageID$	$DestID$	$H(F_i \| \| T_k)$	$Payload$

5.3 Phase III: Private Auction

Recall that in a DOSN, there are no trusted central auctioneers. Thus, distributed bidder-resolved auctions are indispensable for security and privacy purposes, in which bidders use cryptographic protocols to jointly determine the auction result. Previously proposed such auction schemes like [8] are not fully private and have high communication, computation, and storage complexities, which limit their usage in practical applications. In the following, we develop a private and more efficient auction protocol.

5.3.1 Outline

We first present the conceptual outline of the proposed private Vickrey based auction protocol. Without loss of generality, let us assume that a seller defines a price vector $\mathbf{p} = (\, p_K \quad p_{K-1} \quad \cdots \quad p_1 \,)^{\mathsf{T}}$ of K possible bidding prices. A bidder, say node v_i, submits a bid $\mathbf{b}^i = \left(\, b_K^i \quad b_{K-1}^i \quad \cdots \quad b_1^i \,\right)^{\mathsf{T}}$, where $b_k^i \in \{0, 1\}$ and $1 \leq k \leq K$. If bidder v_i's bidding price is p_{l_i} $(1 \leq l_i \leq K)$, then b_k^i is equal to 1 when $k = l_i$ and equal to 0 otherwise.

We then define "*a doubly-integrated bid vector*", denoted by $\hat{\mathbf{b}}^i$, for bidder v_i as

$$\hat{\mathbf{b}}^{\mathbf{i}} = \begin{pmatrix} \hat{b}_K^i \\ \hat{b}_{K-1}^i \\ \hat{b}_{K-2}^i \\ \vdots \\ \hat{b}_k^i \\ \vdots \\ \hat{b}_2^i \\ \hat{b}_1^i \end{pmatrix} = \begin{pmatrix} b_K^i \\ 2b_K^i + b_{K-1}^i \\ 2b_K^i + 2b_{K-1}^i + b_{K-2}^i \\ \vdots \\ 2b_K^i + 2b_{K-1}^i + \cdots + 2b_{k+1}^i + b_k^i \\ \vdots \\ 2b_K^i + 2b_{K-1}^i + \cdots + 2b_3^i + b_2^i \\ 2b_K^i + 2b_{K-1}^i + \cdots + 2b_2^i + b_1^i \end{pmatrix}.$$

Thus, when the bidding price is p_{l_i} $(1 \leq l_i \leq K)$, the vector $\hat{\mathbf{b}}^{\mathbf{i}}$ is as follows:

$$\hat{b_k^i} = b_K^i \text{ when } k = K, \tag{1}$$

2. One may argue that the bridge node is able to know the IP address and may be able to identify the bidders when they connect to its page. However, the bidders can use services like Tor [37] to hide their true IP addresses.

and

$$\hat{b_k^i} = 2\sum_{m=k+1}^{K} b_m^i + b_k^i = \begin{cases} 0, & \text{when } l_i < k < K, \\ 1, & \text{when } k = l_i \\ 2, & \text{when } k < l_i \end{cases}. \quad (2)$$

Assume that there are totally n bidders bidding for the same item. The sum of all the doubly-integrated bid vectors, denoted by $\hat{\mathbf{B}}$, can be obtained as[3]

$$\hat{\mathbf{B}} = \sum_{i=1}^{n} \hat{\mathbf{b}}^{\mathbf{i}}. \quad (3)$$

The vector $\hat{\mathbf{B}}$'s elements would be 1, 3, 5,..., $(2M-1)$ corresponding to the $1^{st}, 2^{nd}, 3^{rd}, \ldots,$ and Mth highest bidding prices.

Winning price determination. The winning price can be found in the following. We calculate a vector $\mathcal{P}$ as follows:

$$\mathcal{P} = \left(\hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K\right) * \mathbf{R}, \quad (4)$$

where $\mathbf{R}$ is a random K-dimensional vector jointly generated by all the bidders ($\mathbf{R}(k) \neq 0$ for $1 \leq k \leq K$), $\mathbf{U_K}$ is a K-dimensional vector whose elements are all 1's, and '$*$' refers to component-wise multiplication. Thus, all the elements in $\mathcal{P}$ are non-zero random numbers, except the element corresponding to the second highest bidding price which is zero. Thus, if $\mathcal{P}(w) = 0$, then $\mathbf{p}(w)$ is the winning price of the auction.

Winning bidder identification. If a malicious winning bidder does not come forward and claim the bid, the auction would be incomplete and the item remains unsold. Therefore, in order to ensure non-repudiation, it is necessary to identify the winning bidder. Particularly, the winner of the auction is bidder v_i if $\mathcal{W}^i$ is zero, where

$$\mathcal{W}^i = (\hat{\mathbf{b}}^{\mathbf{i}}(w) - 2) \cdot R_i, \quad (5)$$

and R_i is a non-zero random number generated by bidder v_i. An example of four bidders can be found in Appendix D, available in the online supplement material.

5.3.2 Cryptographic Protocol Design

Next we describe the details of the proposed cryptographic private auction protocol. Recall that after each bidder bidding for the same item receives an ACK message from the bridge node containing its address (IP address, port number, etc.), each bidder can access the advertisements from all the sellers available at the bridge node and decide which particular seller's item to bid for. The bidders then send to the bridge node their encrypted bids according to the price vector defined by the seller they choose. The bridge node finally determines the winning price and the winning bidder. The proposed auction protocol consists of five processes as follows.

Public pseudo ID authentication. After receiving a buying advertisement message from a bidder, the bridge node needs to verify if it is an authentic user in the network so as to defend attacks like impersonation. Thus, a bidding node v_i needs to prove that it possesses the private pseudo ID s_i corresponding to the pubic pseudo ID ρ_i. We apply Fiat-Shamir heuristic to convert the interactive proof [35] between a prover (a bidder) and a verifier (the bridge node) to a non-interactive proof. Note that the purpose of having non-interactive zero knowledge (NIZK) proofs is not only to reduce the communication complexity between the bidders and the bridge node, but more importantly, to relax the assumption on trustworthy bridge nodes (i.e., honest verifiers) in ZKPs. This is because the non-interactive proof of authenticity can be verified by all the parties participating in the auction and a dishonest bridge node will get caught. In particular, the Fiat-Shamir heuristic [38] makes use of a hash function $\tilde{H}(\cdot)$, modelled as a Random Oracle (RO), to construct a random challenge from the verifier. The public pseudo ID authentication can be carried out following the steps below:

- Bidder v_i chooses a random $\tilde{r}$, calculates $z = \tilde{r}^{\rho_i} \bmod N$, and sends z, $y = \tilde{r} s_i{}^c \bmod N$, and the certificate C_i to the bridge node, where $c = \tilde{H}(z)$.
- The bridge node checks and accepts the proof if $z = y^{\rho_i} \tilde{g}^{-c} \bmod N$.

Theorem 1 (Completeness). *A legal bidding node can always be successfully authenticated.*

Proof. Please refer to Appendix E, available in the online supplemental material, for the proof. □

Theorem 2 (Soundness). *An illegal bidding node who does not have a valid s_i can only be successfully authenticated with a negligible probability.*

Proof. Please refer to Appendix F, available in the online supplemental material, for detailed proof. □

Note that in order to further reduce the communication cost between the bidders and the bridge node, bidders can include this non-interactive proof in the payload of their advertisement messages as mentioned before.

Distributed encryption key generation. Each bidder then chooses a random key $x_i \in \mathbb{G}_q$ and sends $y_i = g^{x_i} \bmod p$ to the bridge node with a ZKP of the knowledge of x_i, i.e., a discrete logarithm regarding y_i (Section 4.2.1). The bridge node makes all the y_i's and the corresponding ZKPs public. Each bidder can compute the encryption key (public key) as $y = \prod_{i=1}^{n} y_i$. Note that similarly, in order to reduce the communication complexity of interactive ZKPs and relaxing the assumption on a reliable bridge node, we employ Fiat-Shamir heuristic [38] to make the ZKP (and all the following ZKPs as well) non-interactive, i.e., use NIZK proofs.

Bid encryption. Each bidder prepares his/her own bid and sends the encrypted bid to the bridge node as follows.

Bid preparation: Without loss of generality, we denote a seller's price vector by $\mathbf{p} = (\, p_K \quad p_{K-1} \quad \cdots \quad p_1 \,)^{\mathsf{T}}$ and a bidder's (node v_i's) bidding vector by $\mathbf{b}^i = \left(\, b_K^i \quad b_{K-1}^i \quad \cdots \quad b_1^i \,\right)^{\mathsf{T}}$. Suppose node v_i's bidding price is p_{l_i}. Then, we have b_k^i $(1 \leq k \leq K)$ is equal to 1 when $k = l_i$ and equal to 0 otherwise.

Bid encryption: The bidder then encrypts the bidding vector with the encryption (public) key element by element, i.e., for any $1 \leq k \leq K$, the bidder computes $Enc(b_k^i) =$

3. The bridge node can index the bidders by the order of received bids.

$\langle \alpha_k^i, \beta_k^i \rangle = \langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$ where $r_k^i \in \mathbb{G}_q$ is a random number generated by bidder v_i.

ZKP Generation: The bidder v_i needs to prove that the encrypted bidding vector is generated adhering to the protocol. In particular, it needs to prove the following facts in zero knowledge:

- *Each element in its bidding vector is the encryption of either 1 or 0.* The bidder generates a ZKP as described in Section 4.2.3.
- *Only one element in its bid vector corresponds to 1,* i.e., $\sum_{k=1}^{K} b_k^i = 1$. The bridge node uses the protocol described in 4.2.2 to show $\log_y(\frac{\prod_{k=1}^{K} \beta_k^i}{g}) = \log_g(\prod_{k=1}^{K} \alpha_k^i)$ in zero knowledge.

Bid signing and publishing: Note that the encrypted bidding vectors obtained above are repudiable. Before sending the encrypted bids to the bridge node, in order to ensure authentication and non-repudiation, all bidders sign their bids with an anonymous (pseudo ID based) signature scheme [35] shown below. In the following, we detail the process for bidder v_i to sign each of the encrypted elements in the bidding vector $\mathbf{b}^i$, i.e., $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle = \langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$, and for the bridge node to verify it. The calculations in this process take place in modulo-N unless mentioned otherwise.

- Bidder v_i computes $z_{\alpha_k^i} = r_{\alpha_k^i}^{\rho_i}$, $\epsilon_{\alpha_k^i} = h(\alpha_k^i || z_{\alpha_k^i})$, and $y_{\alpha_k^i} = r_{\alpha_k^i} s_i^{\epsilon_{\alpha_k^i}}$, where $r_{\alpha_k^i}$ is a random number generated by v_i, and $h(\cdot)$ is a publicly known hash function. Bidder v_i also generates $r_{\beta_k^i}$ and computes $\epsilon_{\beta_k^i}$ and $y_{\beta_k^i}$ in a similar way. Then, bidder v_i generates the signature for the encrypted bid $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle$, which is $\langle (\epsilon_{\alpha_k^i}, y_{\alpha_k^i}), (\epsilon_{\beta_k^i}, y_{\beta_k^i}) \rangle$, and sends it along with his/her certificate C_i to the bridge node.
- The bridge node obtains the public pseudo ID ρ_i of bidder v_i from the certificate C_i and computes $m_{\alpha_k^i} = y_{\alpha_k^i}^{\rho_i} \tilde{g}^{-\epsilon_{\alpha_k^i}}$ and $m_{\beta_k^i} = y_{\beta_k^i}^{\rho_i} \tilde{g}^{-\epsilon_{\beta_k^i}}$. The bridge node accepts the bid if $h(\alpha_k^i || m_{\alpha_k^i}) = \epsilon_{\alpha_k^i}$ and $h(\beta_k^i || m_{\beta_k^i}) = \epsilon_{\beta_k^i}$.

Theorem 3 (Completeness). *If the bid from bidder v_i is authentic, the following verification equations would hold: $h(\alpha_k^i || m_{\alpha_k^i}) = \epsilon_{\alpha_k^i}$ and $h(\beta_k^i || m_{\beta_k^i}) = \epsilon_{\beta_k^i}$ for any $1 \le k \le K$.*

Proof. Please refer to Appendix G, available in the online supplemental material, for detailed proof. □

Note that any participants in the auction can check the verification equations.

Theorem 4 (Soundness). *An illegal bidder, who generates a signature without a valid s_i, can only pass the verification at the bridge node with a negligible probability.*

Proof. Please refer to Appendix H, available in the online supplemental material, for detailed proof. □

Winning price determination. Once all the bids are received within the time frame of current auction, the bridge node combines the encrypted bidding vectors to obtain the encrypted doubly-integrated bid vector. For each bidder v_i ($1 \le i \le n$), the bridge node computes $Enc(\hat{\mathbf{b}}^i)$ as:

$$Enc(\hat{\mathbf{b}}^i) = \left(\langle \hat{\alpha}_K^i, \hat{\beta}_K^i \rangle \langle \hat{\alpha}_{K-1}^i, \hat{\beta}_{K-1}^i \rangle \dots \langle \hat{\alpha}_1^i, \hat{\beta}_1^i \rangle \right)^{\mathsf{T}},$$

where $\langle \hat{\alpha}_k^i, \hat{\beta}_k^i \rangle$ is

$$\left\langle \prod_{m=k+1}^{K} (\alpha_m^i)^2 \cdot \alpha_k^i, \prod_{m=k+1}^{K} (\beta_m^i)^2 \cdot \beta_k^i \right\rangle = \left\langle g^{\sum_{m=k+1}^{K} 2r_m^i + r_k^i}, y^{\sum_{m=k+1}^{K} 2b_m^i + b_k^i} \right\rangle$$

when $1 \le k < K$, and $\langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$ when $k = K$. Define δ_k^i as $\sum_{m=k+1}^{K} 2r_m^i + r_k^i$ when $1 \le k < K$ and r_k^i when $k = K$. Thus, for any $1 \le k \le K$, we have

$$\langle \hat{\alpha}_k^i, \hat{\beta}_k^i \rangle = \langle g^{\delta_k^i}, g^{\hat{b_k^i}} y^{\delta_k^i} \rangle, \tag{6}$$

where $\hat{b_k^i}$ is defined in (1) and (2).

Similarly, the bridge node can obtain the encryption of the sum of all the doubly-integrated bid vectors as follows

$$\begin{aligned} Enc(\hat{\mathbf{B}}) &= \left(\prod_{i=1}^{n} Enc(\hat{b_K^i}) \dots \prod_{i=1}^{n} Enc(\hat{b_1^i}) \right)^{\mathsf{T}} \\ &= \left(\left\langle \prod_{i=1}^{n} \hat{\alpha_K^i}, \prod_{i=1}^{n} \hat{\beta_K^i} \right\rangle \dots \left\langle \prod_{i=1}^{n} \hat{\alpha_1^i}, \prod_{i=1}^{n} \hat{\beta_1^i} \right\rangle \right)^{\mathsf{T}} \\ &= \left(\langle \hat{\alpha}_{B_K}, \hat{\beta}_{B_K} \rangle \langle \hat{\alpha}_{B_{K-1}}, \hat{\beta}_{B_{K-1}} \rangle \dots \langle \hat{\alpha}_{B_1}, \hat{\beta}_{B_1} \rangle \right)^{\mathsf{T}}, \end{aligned} \tag{7}$$

where for any $1 \le k \le K$,

$$\langle \hat{\alpha}_{B_k}, \hat{\beta}_{B_k} \rangle = \left\langle g^{\sum_{i=1}^{n} \delta_k^i}, y^{\sum_{i=1}^{n} \hat{b}_k^i \sum_{i=1}^{n} \delta_k^i} \right\rangle. \tag{8}$$

Recall that we determine the winning price through (4). Thus, the bridge node first computes the encryption of a vector $\mathbf{P} = \hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K$ as follows:

$$Enc(\mathbf{P}) = Enc(\hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K) = Enc(\hat{\mathbf{B}}) * Enc(-3\mathbf{U}_K),$$

which we denote by

$$\left(\langle \alpha_{P_K}, \beta_{P_K} \rangle \quad \langle \alpha_{P_{K-1}}, \beta_{P_{K-1}} \rangle \quad \dots \quad \langle \alpha_{P_1}, \beta_{P_1} \rangle \right)^{\mathsf{T}}.$$

The bridge node then publishes the above calculations on its public profile, so that all the bidders can verify the correctness of the computations.

In the next step, each bidder participates in the distributed decryption of the clearing price. Specifically, each bidder v_i computes and sends to the bridge node

$$\begin{aligned} \boldsymbol{\alpha}'_{P^i} &= \left((\alpha_{P_K})^{R_K^i} \quad (\alpha_{P_{K-1}})^{R_{K-1}^i} \quad \dots \quad (\alpha_{P_1})^{R_1^i} \right)^{\mathsf{T}}, \boldsymbol{\beta}'_{P^i} \\ &= \left((\beta_{P_K})^{R_K^i} \quad (\beta_{P_{K-1}})^{R_{K-1}^i} \quad \dots \quad (\beta_{P_1})^{R_1^i} \right)^{\mathsf{T}}, \end{aligned}$$

where $\mathbf{R}^i$ is a K-dimensional vector of non-zero random numbers generated by bidder v_i. In addition to $\boldsymbol{\alpha}'_{P^i}$ and $\boldsymbol{\beta}'_{P^i}$,

each bidder also proves in zero knowledge that the corresponding elements, e.g., the kth elements, of the vectors $\boldsymbol{\alpha}'_{P^i}$ and $\boldsymbol{\beta}'_{P^i}$ are obtained using the same random value, e.g., R_k^i (as shown in Section 4.2.2).

The bridge node then combines the received values from the bidders to calculate $\langle \boldsymbol{\alpha}'_{\mathbf{P}}, \boldsymbol{\beta}'_{\mathbf{P}} \rangle$ as follows:

$$\left(\left\langle \prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(K), \prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(K) \right\rangle, \ldots, \left\langle \prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(1), \prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(1) \right\rangle \right)^{\mathsf{T}}.$$

Thus, all the bidders can calculate the winning price of the auction by following the distributed decryption approach, i.e., computing

$$\boldsymbol{\Omega} = \left(\frac{\boldsymbol{\beta}'_{\mathbf{P}}(K)}{\prod_{i=1}^{n} (\boldsymbol{\alpha}'_{\mathbf{P}}(K))^{x_i}} \quad \frac{\boldsymbol{\beta}'_{\mathbf{P}}(K-1)}{\prod_{i=1}^{n} (\boldsymbol{\alpha}'_{\mathbf{P}}(K-1))^{x_i}} \quad \cdots \quad \frac{\boldsymbol{\beta}'_{\mathbf{P}}(1)}{\prod_{i=1}^{n} (\boldsymbol{\alpha}'_{\mathbf{P}}(1))^{x_i}} \right)^{\mathsf{T}},$$

where $(\boldsymbol{\alpha}'_{\mathbf{P}}(k))^{x_i}$'s $(1 \leq k \leq K)$ are transmitted by bidder v_i to the bridge node and made public (along with a proof that the same x_i was used as in the distributed key generation process, as shown in Section 4.2.2), and

$$\begin{aligned} \Omega(k) &= \frac{\prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(k)}{\prod_{i=1}^{n} (\prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(1))^{x_i}} = \frac{(\beta_{P_k})^{\sum_{i=1}^{n} R_k^i}}{(\alpha_{P_k})^{\sum_{i=1}^{n} R_k^i \sum_{i=1}^{n} x^i}} \\ &= \frac{(g^{\mathbf{P}(k)} \cdot y^r)^{\sum_{i=1}^{n} R_k^i}}{(g^r)^{\sum_{i=1}^{n} R_k^i \sum_{i=1}^{n} x^i}} = g^{\mathbf{P}(k) \cdot \sum_{i=1}^{n} R_k^i} = g^{\mathcal{P}(k)}, \end{aligned} \tag{9}$$

where $\mathcal{P}$ is defined in (4). Therefore, the element p_w of the price vector $\mathbf{p}$ is the winning price if $\Omega(w) = g^{\mathcal{P}(w)} = g^0 = 1$.

Winning bidder identification. Recall that the winning bidder can be determined by checking if (5) is equal to 0. Only the winning bidder's public pseudo ID will be known to others. The winning bidder determination process is as follows.

First, for any $1 \leq i \leq n$, the bridge node computes,

$$Enc(W^i) = Enc(\hat{\mathbf{b}}^i(w) - 2) = Enc(\hat{\mathbf{b}}^i(w)) \cdot Enc(-2),$$

which we denote by $\langle \alpha_{W^i}, \beta_{W^i} \rangle$.

Then, each bidder v_i computes $\langle \alpha_{W^i}^{R_i}, \beta_{W^i}^{R_i} \rangle$, and sends it and a ZKP (as shown in Section 4.2.2) that $\alpha_{W^i}^{R_i}$, $\beta_{W^i}^{R_i}$ are computed using the same random number to the bridge node. The bridge node makes such values public and ask all the bidders to jointly decrypt for W^i's. Particularly, for any W^i $(1 \leq i \leq n)$, each bidder v_j transmits $(\alpha_{W^i}^{R_i})^{x_j}$ (along with a proof that these n x_j's are the same as that used as in the distributed key generation process, as shown in Section 4.2.2) to the bridge node, which can then compute

$$\Phi^i = \frac{\beta_{W^i}^{R_i}}{\prod_{j=1}^{n} (\alpha_{W^i}^{R_i})^{x_j}} = \frac{(g^{W^i} \cdot y^r)^{R_i}}{(g^r)^{R_i \sum_{j=1}^{n} x_j}} = g^{W^i R_i} = g^{\mathcal{W}^i}. \tag{10}$$

Finally, bidder v_i is the winning bidder if $\mathcal{W}^i = 0$, or $\Phi^i = g^0 = 1$.

5.3.3 Tie Breaking

We find that the auction scheme presented above fails to produce an outcome if there is a tie in the highest or/and the second highest bidding price. A simple solution would be to decrypt all the elements of vector $\hat{\mathbf{B}}$, which gives the locations of all the ties (including the ties in the highest bid and the second highest bid) and the winning price as well. However, revealing $\hat{\mathbf{B}}$ in public constitutes a breach in privacy of the bidders whose bidding statistics will be available to potential adversaries who can use the information to their advantage in future auctions. The only information needed to be revealed is the winning price and the pseudo public ID of the winning bidder. In the following, we develop a scheme to determine the auction result in presence of tie(s).

Particularly, notice that the vector $(\mathbf{B} - t \cdot \mathbf{U_K})$, where $\mathbf{B} = \sum_{i=1}^{n} \mathbf{b}^i$, results in 0 at each location corresponding to the element in the price vector where there is a tie of t bidders. Besides, the vector $\hat{\mathbf{B}} - (t + 2h) \cdot \mathbf{U_K}$ leads to 0 at the second highest bid position if t bidders bid the same second highest price and h bidders bid the same highest price. Thus, if $(\mathbf{B} - t \cdot \mathbf{U_K})$ and $\hat{\mathbf{B}} - (t + 2h) \cdot \mathbf{U_K}$ both result in 0 at the same location, then that is corresponding to the second highest price in the price vector. Consequently, the bridge node can first calculate the following vector

$$\mathcal{P}^{t,h} = ((\mathbf{B} - t \cdot \mathbf{U_K}) + (n+1)(\hat{\mathbf{B}} - (t + 2h) \cdot \mathbf{U_K})) * \mathbf{R}$$

for $1 \leq h \leq n - t$ and $1 \leq t \leq n - 1$, where $\mathbf{R}$ is a K-dimensional nonzero random vector jointly generated by the bidders, and the second term is multiplied by $(n + 1)$ to make sure the two terms do not accidently add up to zero. This vector can be re-written as

$$\mathcal{P}^{t,h} = \left((\mathbf{B} + (n+1)\hat{\mathbf{B}}\right) - \left((n+2)t + 2(n+1)h)\mathbf{U_K}\right) * \mathbf{R} \tag{11}$$

and the winning price is p_w if $\mathcal{P}^{t,h}(w) = 0$. In order to ensure security and privacy, we can follow the cryptographic process presented in Section 5.3.2 to verify if $g^{\mathcal{P}^{t,h}(w)} = g^0 = 1$.

Similarly, winning bidders can be identified by checking if$(\hat{\mathbf{b}}^i(w) - 2) \cdot R_i = 0$ along the line in Section 5.3.2, where R_i is a nonzero random number generated by bidder v_i. In the case of a tie at the highest bidding price, some specific rules can be employed to determine the final winner, e.g., the bidder who submitted his/her bid first among all the winners.

5.3.4 (M+1)st Price Auction

The private auction scheme that we have developed so far is for the case where each seller has one unit of an item to sell at a time and a buyer is also interested in buying only one unit of the item at a time. In this part, we investigate private auction for the scenarios where a seller has multiple, say M $(M \geq 1)$, units of the same items to sell, i.e., private $(M + 1)$-st price auction. In particular, each buyer is interested in buying one unit of the item and the top M bidders are winners who pay the $(M + 1)$-st highest bidding price. Note that when $M = 1$, the $(M + 1)$st-price auction reduces to the second price (Vickrey) auction investigated above.

The basic idea for private $(M + 1)$-st price auction is as follows. When there is no tie in the bidding prices, the winning price in an $(M + 1)$-st price auction can be obtained by first calculating the vector below in a similar way to (4), i.e.,

$$\overline{\mathcal{P}} = \left(\hat{\mathbf{B}} - (2M+1)\cdot \mathbf{U}_K\right) * \mathbf{R}. \quad (12)$$

The winning price is p_w if $\overline{\mathcal{P}}(w) = 0$. Similarly, the winning bidders can be identified if

$$\overline{\mathcal{W}}^i = (\hat{\mathbf{b}}^i(w) - 2)\cdot R_i = 0. \quad (13)$$

When there are ties in the bidding prices, the winning price and the winning bidders can be determined by following the same approach in Section 5.3.3. The cryptographic process in Section 5.3.2 can be employed to provide security and privacy.

6 Performance Analysis

6.1 Computation Cost

In what follows, we analyze the computational complexity of each bidder and that of the bridge node, respectively.

6.1.1 A Bidder's Computational Complexity

In the *pseudo ID authentication* process, a bidder conducts two exponentiations, denoted by EXP, in the NIZK proof. In the *distributed encryption key generation* process, a bidder carries out $1 \times EXP$ for public key generation and $1 \times EXP$ for the corresponding ZKP. In the *bid encryption* process, a bidder conducts $3K \times EXP$ for "bid encryption", $(6K+2) \times EXP$ for "ZKP generation", and $4K \times EXP$ for "bid signing and publishing". In the *winning price determination* process, a bidder needs to compute $3K \times EXP$ and another $(3K+1) \times EXP$ for the ZKPs. In the *winning bidder identification* process, a bidder needs to compute $(n+2) \times EXP$ and another $(n+2) \times EXP$ for the ZKPs. Therefore, the total computational complexity of a bidder is $(2n+19K+11) \times EXP$, i.e., on the order of $O(n+K)$ EXP operations. Note that we ignore the multiplication operation, denoted by MUL, as it is insignificant compared to exponentiations.

6.1.2 A Bridge Node's Computational Complexity

In the *pseudo ID authentication* process, a bridge node conducts $2 \times EXP$ and $1 \times MUL$ for each bidder, i.e., $2n \times EXP$ and $n \times MUL$ in total. In the *distributed encryption key generation* process, a bridge node carries out $2 \times EXP$ and $1 \times MUL$ in the ZKP for each bidder, and $(n-1) \times MUL$ for computing the encryption key, i.e., $2n \times EXP$ and $(2n-1) \times MUL$ in total. In the *bid encryption* process, a bridge node computes $8nK \times EXP$ and $4nK \times MUL$ for the first ZKP and $4n \times EXP$ and $2n \times MUL$ for the second ZKP for "ZKP generation", and $4nK \times EXP$ and $2nK \times MUL$ for "bid signing and publishing", i.e., $(12nK+4n) \times EXP$ and $(6nK+2n) \times MUL$ in total. In the *winning price determination* process, a bridge node needs to conduct $6nK \times MUL$ for $Enc(\hat{\mathbf{b}}^i)$, $2nK \times MUL$ for $Ecn(\hat{\mathbf{B}})$, $3K \times EXP$ and $4K \times MUL$ for $Enc(\mathbf{P})$, $3nK \times MUL$ for $\mathbf{\Omega}$, and $(8nK+4n) \times EXP$ and $(4nK+2n) \times MUL$ for the two ZKPs, i.e., $(8nK+4n+3K) \times EXP$ and $(15nK+2n+4K) \times MUL$ in total. In the *winning bidder identification* process, a bridge node computes $6n \times EXP$ and $4n \times MUL$ in the first step, and $n^2 \times MUL$ in the second step, and $(4n^2+8n) \times EXP$ and $(2n^2+4n) \times MUL$ for the two ZKPs, i.e., $(4n^2+14n) \times EXP$ and $(3n^2+8n) \times MUL$ in total. Thus, the total computational complexity of a bridge node is $(4n^2+20nK+24n+3K) \times EXP$ and $(3n^2+21nK+14n+4K-1) \times MUL$, i.e., on the order of $O(n^2+nK)$ EXP and $O(n^2+nK)$ MUL operations.

Note that the above computation cost for a bridge node is largely attributed to the ZKP verification related computations (all the n^2 and nk terms for the EXP operation). Recall that in our protocols, the bidders provide non-interactive ZKPs which can be verified by any participants of the auction process. Thus, the bridge node can distribute the computation load for proof verifications (including those for ZKPs and for signatures) to the bidders without increasing their computational complexities. For example, bidder j can verify the ZKPs of bidder $(j+k)\ mod\ n$, where $k \in [1, n-1]$. In this case, the computational complexity of each bidder and that of the bridge node will be $O(n+K)$ EXP. It is also important to point out that [8] does not include the cost for ZKP verification and still has the computational complexity of $O(nK)$ EXP. We can see that our proposed protocol has much lower computational complexity.

6.2 Communication Cost

Note that the communication cost mainly comes from the bidders since all the ZKPs are non-interactive. We analyze the communication cost of each bidder as follows.

In the *pseudo ID authentication* process, a bidder transmits z, y, and C_i to the bridge node, i.e., $4\lceil \log N \rceil$ bits. In the *distributed key generation* process, a bidder sends $\lceil \log p \rceil$ bits to construct the public key and one $\lceil \log p \rceil + \lceil \log q \rceil$ bits ZKP to the bridge node, i.e., $2\lceil \log p \rceil + \lceil \log q \rceil$ in total. In the *bid encryption* process, each bidder transmits $K((4\lceil \log p \rceil + 4\lceil \log q \rceil) + (2\lceil \log p \rceil + \lceil \log q \rceil))$ bits to prove the bids fulfill the given requirements for "ZKP generation". Each bidder also sends K ElGamal ciphertexts ($2K\lceil \log p \rceil$ bits), K corresponding signatures ($2K(\lceil \log N \rceil + |h|)$ bits with $|h|$ being the size of hash digest), and his/her certificate ($2\lceil \log N \rceil$ bits). So all the cost in this process is $8K\lceil \log p \rceil + 5K\lceil \log q \rceil + (2K+2)\lceil \log N \rceil + 2K|h|$ bits. In the *winning price determination* process, each bidder needs to send $3K\lceil \log p \rceil$ bits for $\boldsymbol{\alpha'}_{P^i}, \boldsymbol{\beta'}_{P^i}$, $(\boldsymbol{\alpha'}_P(k))^{x_i}$, and $(2K+1)(2\lceil \log p \rceil + \lceil \log q \rceil)$ bits for the corresponding ZKPs, i.e., $(7K+2)\lceil \log p \rceil + (2K+1)\lceil \log q \rceil$ bits in total. Lastly, in the *winning bidder identification* process, each bidder sends $(n+2)\lceil \log p \rceil$ bits and $(n+2)(2\lceil \log p \rceil + \lceil \log q \rceil)$ bits, respectively, for distributed decryption and ZKPs, i.e., $(3n+6)\lceil \log p \rceil + (n+2)\lceil \log q \rceil$ bits in total. Therefore, total communication cost per bidder is $(2\lceil \log N \rceil + 15\lceil \log p \rceil + 7\lceil \log q \rceil + 2|h|)K + (3\lceil \log p \rceil + \lceil \log q \rceil)n + (6\lceil \log N \rceil + 10\lceil \log p \rceil + 4\lceil \log q \rceil)$ bits, and hence on the order of $O(n+K)$ bits.

Therefore, the total communication complexity of each node in our scheme can be proved to be $O(n+K)$ bits, while that in [8] is $O(nK)$ bits.

6.3 Storage Cost

In the following we analyze the storage cost of the bidding nodes and the bridge node. Note that the analysis takes into account all the intermediate data as part of storage cost, and hence the results below can be considered as an upper bound.

6.3.1 Storage Cost of a Bidder

The storage cost of a bidder comprises of all the data it has computed, transmitted to, and received from a bridge node. In particular, as detailed in Section 6.2, the total amount of data computed by a bidder and transmitted to the bridge node is $(2\lceil \log N\rceil + 15\lceil \log p\rceil + 7\lceil \log q\rceil + 2|h|)K + (3\lceil \log p\rceil + \lceil \log q\rceil)n + (6\lceil \log N\rceil + 10\lceil \log p\rceil + 4\lceil \log q\rceil)$, i.e., $O(n+K)$, bits. In addition, a bidder keeps its own keys, and creates an advertisement message and receives an acknowledgement message, which are nevertheless negligible. Thus, the storage cost of a bidder is on the order $O(n+K)$ bits.

6.3.2 Storage Cost of a Bridge Node

The bridge node accepts the advertisement messages and encrypted bids from all the bidders, and performs the computations for conducting fully private auctions as detailed in Section 5.3.2. Therefore, the storage cost of a bridge node is n times the storage cost of a bidding node, i.e., $n \times \big((2\lceil \log N\rceil + 17\lceil \log p\rceil + 7\lceil \log q\rceil + 2|h|)K + (5\lceil \log p\rceil + \lceil \log q\rceil)n + (6\lceil \log N\rceil + 10\lceil \log p\rceil + 4\lceil \log q\rceil)\big)$ bits. In addition, in the *winning price determination* process, the bridge node computes the encrypted doubly-integrated bid vector $Enc(\hat{\mathbf{B}})$ of $2K\lceil \log p\rceil$ bits, the vector $Enc(\mathbf{P})$ of $2K\lceil \log p\rceil$ bits, and the vector $\langle \boldsymbol{\alpha}'_{\mathbf{P}}, \boldsymbol{\beta}'_{\mathbf{P}}\rangle$ of $2K\lceil \log p\rceil$ bits. Similarly, the bridge node also computes $Enc(W^i)$ for any $1 \le i \le n$ of $2n\lceil \log p\rceil$ bits in the *wining bidder identification* process. The total storage cost for a bridge node is thus on the order $O(n^2 + nK)$ bits.

Note that our proposed protocol has significantly lower storage cost than before, while the storage cost of a bidder in [8] is $O(n^2K)$ bits.

6.4 Security and Privacy Analysis

This section investigates the security and privacy of the proposed auction framework SPA. We show, in the following, that SPA is secure and privacy-preserving not only under the honest-but-curious model, but also with regard to the malicious bidders who may want to deviate from the protocols to disrupt and/or learn more about the other bidders.

Theorem 5 (Privacy). *A bidder's privacy is preserved regardless of the number of other colluding bidders. A seller's privacy is preserved too.*

Proof. A bidder obtains a pair of public/private pseudo IDs in the identity initiation phase whenever he/she wants to participate in an auction. The bidder can use different such pseudo IDs for different auctions. Thus, the identity privacy can be preserved and the bidder cannot be traced. Besides, our proposed auction scheme employs a distributed ElGamal cryptosystem. Unlike (n,k) threshold cryptosystems, where a ciphertext can be decrypted if k-out-of-n participants collude, our encryption scheme constructs a public key in an n-out-of-n secret sharing fashion. Therefore, a bidder's bidding vector encrypted with the public key can only be decrypted if the bidder participates in the distributed decryption. In our auction protocol, bidders do not collaboratively decrypt their own encrypted bidding vectors. They only jointly decrypt for $g^{\mathcal{P}(w)}$ as shown in (9), which is equal to 1 if p_w is the winning price and some random number otherwise. Thus, a bidder's bidding price privacy can also be preserved. Note that as mentioned before, we do not consider the possible adversaries at the DHT and Internet layers, since there have been several works addressing the privacy issues there [19], [20] and our DHT protocols can be easily adapted. □

Theorem 6 (Authenticity and non-repudiation). *A bidder with legal pseudo IDs and legitimate bidding vectors can always be authenticated. Besides, a bid can be traced back to the bidder.*

Proof. Due to the public pseudo ID authentication process, a malicious bidder cannot use a fake public pseudo ID or impersonate some other bidder to pass the authentication process according to Theorem 1 and 2. Legitimate bidding vectors can also be verified in the "ZKP generation" step of the bid encryption process. Besides, since each bidder signs his/her bids using an anonymous signature scheme based on their public and private pseudo IDs as shown in the bid encryption process, a bid can be traced back to the bidder according to Theorem 3 and 4. □

Theorem 7 (Auction correctness). *The proposed auction protocol results in correct outcomes with high probability (w.h.p.).*

Proof. The winning price obtained from (9) will result in unintended outcomes if for some $1 \le k \le K$, we have $(\hat{\mathbf{B}}(k) - 3) \neq 0$ but $g^{(\hat{\mathbf{B}}(k)-3)\cdot R_k} \equiv 1 \bmod p$. Similarly, the winning bidder obtained from (10) will be incorrect if for some $1 \le i \le n$, we have $(\hat{\mathbf{b}}^i(w) - 2) \neq 0$ but $g^{(\hat{\mathbf{b}}^i(w)-2)\cdot R_i} \equiv 1 \bmod p$, where p_w is the winning price. However, since p is usually a very large number (e.g., 1024 bits [39]), the probability of the occurrence of the above events is very low ($\approx 1/2^{1024}$). Therefore, the proposed auction protocol results in correct outcomes with high probability. □

Theorem 8 (Auction security). *A malicious bidder deviating from the auction protocol cannot disrupt the auction outcome without being detected.*

Proof. In our auction protocol, a malicious bidder may try to disrupt the auction outcome by indulging in the intermediate computations in the protocol. Notice that in the winning price calculation process, each bidder needs to submit ZKPs to prove that for each $1 \le k \le K$, $(\alpha_{P_k})^{R^i_k}$ and $(\beta_{P_k})^{R^i_k}$ are obtained using the same random value, and another ZKP to prove that $(\alpha'_{\mathbf{P}}(k))^{x_i}$'s $(1 \le k \le K)$ are computed using the same x_i as in the key generation phase. In the winner bidder determination process, each bidder v_i submits a ZKP that $\alpha^{R_i}_{W^i}$, $\beta^{R_i}_{W^i}$ are computed using the same random number and another ZKP that the x_j's used in computing $(\alpha^{R_i}_{W^i})^{x_j}$ $(1 \le i \le n)$ are the same as that used as in the distributed key generation process. Thus, any attempt to disrupt the auction outcome by deviating the protocol steps can be detected. □

TABLE 3
Performance of Our Advertisement Distribution Scheme

Social Network Data Set	Normalized Hop Count
LiveJournal Social Network [41]	0.78
Astro Physics Collaboration Network [42]	0.75
Orkut Online Social Network [41]	0.71
Synthetic Data Set Using Nearest Neighbor (modified) Model [43]	0.64

7 Performance Evaluation

In this section, we evaluate the performance of the proposed secure and private auction framework SPA. In particular, we analyze the performance of the advertisement distribution algorithm and study the computation, communication, and storage costs of the proposed private auction scheme. Some of the experiment settings are as follows. In the experiments, the primes p and q in ElGamal cryptosystem are 1,024 and 768 bits, respectively. The modulus N in the anonymous signature scheme is 1,024 bits. Any hash function used in SPA results in digests of 128 bits. The size of an advertisement message and that of an acknowledgement message is set by considering 4 bytes each for $MessageID$, $SrcID$, and $DestID$, and 16 bytes for $H(F_i||T_k)$. We consider that the $Payload$ of an advertisement message is 100 bytes, and the $Payload$ of an acknowledgement message is 4 bytes.

7.1 Advertisement Distribution Algorithm

We first evaluate the performance of our advertisement distribution algorithm. We implement it on several real social network graphs obtained from the SNAP project [40] as well as on a synthetic data set simulating OSNs. For each of these networks, we choose 100 random source and bridge node pairs and measure the average number of hops required to reach the bridge nodes. Table 3 shows the hop counts (normalized by $\log_2 n$ where n is the network size) based on different network data sets. The network data sets are listed in increasing order of average node degree. Intuitively, fewer hops are needed to deliver messages in networks with higher average node degrees. We can see that all advertisement messages can be delivered in $O(\log n)$ hops.

7.2 Computation and Communication Cost

The computation cost is primarily the cost of computing exponentiations and multiplications, which is estimated as follows. We first obtain the average time for an exponentiation and a multiplication (implemented in Java) in a group (e.g., $\mathbb{Z}_N^*$) on a PC with an Intel Core i7 processor and 4 GB RAM. Then the total computation time is obtained by counting the number of exponentiations and multiplications as shown in Section 6.1. For the communication cost analysis, we count the number of bits a bidder needs to transmit and/or receive. The results are shown in Fig. 1 and Fig. 2 in Appendix I, available in the online supplemental material.

Fig. 1a compares the computation time of our protocol with that of [8]. The size of the price vector is set to 500. Lipmaa et al. [27] claim that $k \leq 500$ suffices in most auctions in practice. We can see that the computation time of the bridge node and that of each bidder is well within the practical limits ($\sim$100 seconds) even when the number of bidders is large. In contrast, the computation time of a bidder in [8] is much higher (two orders of magnitude higher), and hence the protocol is impractical when the number of bidders is large. Likewise, as shown in Fig. 1b, the communication cost of a bidder in our protocol is much lower than that of a bidder in [8]. For example, in the case where there are 10,000 bidders, the computation time of a bidder in our protocol is about 32.5 seconds and that of a bridge node is about 56.8 seconds, while the communication cost of a bidder is about 47 Mb. In the same case, the computation time of a bidder in [8] is more than 9.5 hours, and the communication cost of a bidder is about 35 Gb.

We also present the experiment results in Fig. 2 with different K's and n's. We can easily find that our auction protocol is much more efficient in terms of both the computation cost and communication cost and is suitable for practical application even when both the size of the price vector and the number of bidders are large.

7.3 Storage Cost

In accordance with the analysis in Section 6.3, we evaluate the storage cost of a bidding node and of a bridge node and compare the costs with that of a bidding node in [8]. The results are shown in Fig. 3 in Appendix I, available in the online supplemental material.

As shown in Fig. 3a, when $K = 500$ and $n = 10{,}000$, the storage cost of a bidder in our scheme is on the order of $O(n + K)$ and is below 10 megabytes; and the storage cost of a bridge node is on the order of $O(n^2 + nK)$ and is about 100 gigabytes. Fig. 3b shows the storage cost when both the number of bidders and number of price points in the price vector vary. We can easily observe that the storage cost of a bidding node in [8] is significantly higher (on the order of $O(n^2K)$), and may not be practical when K or/and n is large.

7.4 Auction Utility

Finally, we evaluate the bidders' utility and the seller's revenue in the proposed auction scheme. We assume that the seller's price vector is $p = (10{,}000 \;\; 9{,}999 \;\; 9{,}998 \;\; \cdots \;\; 3 \;\; 2 \;\; 1)^{\mathsf{T}}$, and the bidders' bids are uniformly distributed over the price range. In the first experiment, there are a total of 100 bidders in an auction and we study the utility of the bidders. If V_i is the winning bidder v_i's true valuation of the item being auctioned and p_i is the price paid, then the utility of the winning bidder v_i is $u(i) = V_i - p_i$, whereas that of losing bidders is 0. Fig 4a shows the bidders' utility for 100 runs of the experiment which is always nonnegative. In the second experiment, we study the seller's revenue when the number of bidders goes up to 500. For any number of bidders, we obtain the average revenue of the seller over 100 runs. As shown in Fig. 4b, with the assumed uniform distribution of bids, the seller's revenue approaches the maximum value as the number of bidders increases. Fig. 4 is included in Appendix I, available in the online supplemental material.

8 Conclusion

In this work, we have presented a secure and private auction framework, SPA, for DOSNs. System security can be protected against malicious attackers who try to disrupt auction outcomes by indulging in the intermediate computations in the protocol. Users' privacy, including their IDs and bidding prices, can also be guaranteed. In addition, SPA provides authenticity and non-repudiation, which are not made possible in previous auction schemes. The computation and communication complexities of our auction scheme are both $O(n+K)$. In contrast, the most efficient existing private auction schemes like [8] have complexities of $O(nK)$. The storage cost of a bidder and that of a bridge node are $O(n+K)$ and $O(nK)$, respectively, in our scheme, while the storage cost of a bidder in previous works like [8] is $O(n^2K)$. Extensive experiment results have demonstrated the efficiency of the proposed framework.

Acknowledgments

This work was partially supported by the U.S. National Science Foundation under grants CNS-1343220, CNS-1149786, ECCS-1128768, and the Pacific Northwest National Laboratory under U. S. Department of Energy Contract DE-AC05-76RL01830.

References

[1] CNN. (2013, Feb). Report: Eastern european gang hacked apple, facebook, twitter [Online]. Available: http://www.cnn.com/2013/02/20/tech/web/hacked-apple-facebook-twitter/index.html

[2] IGN. (2013, Feb). Microsoft hacked by same method as apple and facebook [Online]. Available: http://www.ign.com/articles/2013/02/23/microsoft-hacked-by-same-method-as-apple-and-facebook

[3] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "PeerSoN: P2p social networking: Early experiences and insights," in *Proc. 2nd ACM EuroSys Workshop Social Netw. Syst.*, 2009, pp. 46–52.

[4] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 94–101, Dec. 2009.

[5] (2013, Jul.) [Online]. Available: http://www.diasporaproject.org/

[6] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders," *J. Finance*, vol. 16, pp. 8–37, 1961.

[7] V. Conitzer, "Auction protocols," in *Algorithms and Theory of Computation Handbook*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2009.

[8] F. Brandt, "How to obtain full privacy in auctions," *Int. J. Inf. Secur.*, vol. 5, no. 4, pp. 201–216, 2006.

[9] A. Thapa, M. Li, S. Salinas, and P. Li, "Asymmetric social proximity based private matching protocols for online social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1547–1559, Jun. 2015.

[10] R. Zhang, Y. Zhang, J. S. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1969–1977.

[11] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, 2001.

[12] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web," in *Proc. 29th Annu. ACM Symp. Theory Comput.*, 1997, pp. 654–663.

[13] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Revised Papers 1st Int. Workshop Peer-to-Peer Syst.*, 2001, pp. 53–65.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2001, pp. 161–172.

[15] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Int. Conf. Distrib. Syst. Platforms*, 2001, pp. 329–350.

[16] B. Zhao, H. Ling, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," vol. 22, no. 1, pp. 41–53, 2004.

[17] C. Lesniewski-Laas and M. F. Kaashoek, "Whanau: A sybil-proof distributed hash table," in *Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation*, 2010, p. 8.

[18] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A near-optimal social network defense against sybil attacks," in *Proc. IEEE Symp. Security Privacy*, 2008, pp. 3–17.

[19] Q. Wang and N. Borisov, "Octopus: A secure and anonymous DHT lookup," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2012, pp. 325–334.

[20] Q. Wang, P. Mittal, and N. Borisov, "In search of an anonymous and secure lookup: Attacks on structured Peer-to-peer anonymous communication systems," in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 308–318.

[21] A. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci.*, Oct. 1986, pp. 162–167.

[22] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proc. 1st ACM Conf. Electron. Commerce*, 1999, pp. 129–139.

[23] A. Juels and M. Szydlo, "A two-server, sealed-bid auction protocol," in *Proc. 6th Int. Conf. Financial Cryptography*, s2003, pp. 72–86.

[24] M. Harkavy, J. D. Tygar, and H. Kikuchi, "Electronic auctions with private bids," in *Proc. 3rd Conf. USENIX Workshop Electron. Commerce*, Berkeley, CA, USA, 1998, p. 6.

[25] H. Kikuchi, "(m+1)st-price auction protocol," in *Proc. 5th Int. Conf. Financial Cryptography*, 2001, pp. 351–363.

[26] K. Sako, "An auction protocol which hides bids of losers," in *Proc. 3rd Int. Workshop Practice Theory Public Key Cryptosyst.*, 2000, pp. 422–432.

[27] H. Lipmaa, N. Asokan, and V. Niemi, "Secure Vickrey auctions without threshold trust," in *Proc. 6th Int. Conf. Financial Cryptography*, 2003, pp. 87–101.

[28] F. Brandt, "A verifiable, bidder-resolved auction protocol," in *Proc. 5th Int. Workshop Deception, Fraud Trust Agent Soc.*, 2002, pp. 18–25.

[29] J. Dreier, J.-G. Dumas, and P. Lafourcade, "Brandts fully private auction protocol revisited," in *Proc. 6th Int. Conf. Progress Cryptol.*, 2013, pp. 88–106.

[30] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.

[31] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proc. 19th Annu. ACM Symp. Theory Comput.*, New York, NY, USA, 1987, pp. 218–229.

[32] C. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, no. 3, pp. 161–174, 1991.

[33] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Proc. 12th Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 1993, pp. 89–105.

[34] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Proc. 16th Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 1997, pp. 103–118.

[35] D. H. Nyang and J. S. Song, "Knowledge-proof based versatile smart card verification protocol," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 3, pp. 39–44, Jul. 2000.

[36] G. L. Miller, "Riemann's hypothesis and tests for primality," in *Proc. 7th Annu. ACM Symp. Theory Comput.*, 1975, pp. 234–239.

[37] (2013, Jul.). [Online]. Available: http://www.torproject.org/

[38] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. Adv. Cryptol.*, 1987, pp. 186–194.

[39] B. Allen, "Implementing several attacks on plain elgamal encryption," Masters Thesis, Iowa State Univ., Ames, IA, USA, 2008.

[40] (2013, Jun.) [Online]. Available: http://snap.stanford.edu/index.html

[41] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, 2015.

[42] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," ACM Trans. *Knowl. Discov. Data*, vol. 1, no. 1, 2007.

[43] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao, "Measurement-calibrated graph models for social network experiments," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 861–870.

Arun Thapa received the BE degree in electronics and communication engineering from National Institute of Technology Durgapur, West Bengal, India, and the PhD degree in electrical and computer engineering from Mississippi State University, Mississippi State, in 2005 and 2014 respectively. He is currently an assistant professor in the Department of Electrical Engineering, Tuskegee University. Prior to working toward the PhD degree at Mississippi State University, he was a telecom engineer in Nepal Telecom and an Assistant Lecturer in Kantipur City College, Kathmandu, Nepal. His research interests include security and privacy, wireless and mobile networks, complex networks, big data, and cyber-physical systems. He is a member of the IEEE.

Weixian Liao received the BE degree in information engineering from Xidian University, Xi'an, China, in 2012. He is currently working toward the PhD degree in the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His current research interests include network optimization, cybersecurity in wireless networks, cyber-physical systems, and big data. He is a student member of the IEEE.

Ming Li received the BE degree in electrical engineering from Sun Yat-sen University, China, in 2007, the ME degree in electrical engineering from Beijing University of Posts and Communications, China, in 2010, and the PhD degree in electrical and computer engineering from Mississippi State University, Starkville, in 2014, respectively. She is currently an assistant professor in the Department of Computer Science and Engineering, University of Nevada, Reno. Her research interests include cybersecurity, privacy-preserving data analysis, resource management and network optimization in cyber-physical systems, cloud computing, mobile computing, wireless networks, smart grid, and big data. She is a member of the IEEE.

Pan Li received the BE degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, in 2009, respectively. Since Fall 2015, he has been with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. He was an assistant professor in the Department of Electrical and Computer Engineering, Mississippi State University between August 2009 and August 2015. His research interests include network science and economics, energy systems, security and privacy, and big data. He has been serving as an editor for *IEEE Journal on Selected Areas in Communications—Cognitive Radio Series* and *IEEE Communications Surveys and Tutorials*, a feature editor for *IEEE Wireless Communications*, and a Technical Program Committee (TPC) co-chair for Ad-hoc, Mesh, Machine-to-Machine and Sensor Networks Track, IEE VTC 2014, Physical Layer Track, Wireless Communications Symposium, WTS 2014, and Wireless Networking Symposium, IEEE ICC 2013. He received the US National Science Foundation (NSF) CAREER Award in 2012 and is a member of the IEEE and the ACM.

Jinyuan Sun received the BSc degree in computer information systems from Beijing Information Technology Institute, China, in 2003, the MASc degree in computer networks from Ryerson University, Canada, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, in 2010. She was a network test developer at RuggedCom Inc., ON, Canada, 2005-2006. She has been an assistant professor in the Department of Electrical Engineering and Computer Science, University of Tennessee Knoxville since August 2010. Her research interests include the security protocol and architecture design of wireless networks. She is a member of the IEEE and ACM.