Supplemental Material

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*, Jinyuan Sun, *Member, IEEE*

Abstract—This document provides the supplemental material for our main paper titled "SPA: A Secure and Private Auction Framework for Decentralized Online Social Networks". In particular, Appendix A introduces ElGamal cryptosystem. Appendix B explains three kinds of Zero Knowledge Proofs that we use in this study. Appendix C details the distributed advertisement distribution algorithm. Appendix D presents an example for winning bidder determination. Appendix E and Appendix F demonstrate the proof of Theorem 1 and that of Theorem 2, respectively. Appendix G and Appendix H show the proof of Theorem 3 and that of Theorem 4, respectively. Appendix I gives detailed experiment results for the computatoin, communication, and storage costs.

Index Terms—Distributed online social networks; auction; security; privacy.

APPENDIX A
ELGAMAL CRYPTOSYSTEM

ElGamal cryptosystem [1] is a semantically secure homomorphic cryptosystem based on the intractability of the discrete logarithm problem in finite fields. In particular, 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 $\mathbb{Z}_p^*$ with order q. A user chooses a random $x \in \mathbb{G}_q$ as the private key, and $y = g^x \mod p$ as the public key where g is a common generator of $\mathbb{G}_q$. All the calculations are modulo-p unless mentioned otherwise. A message $m \in \mathbb{G}_q$ for the user is encrypted as $Enc(m) = \langle \alpha, \beta \rangle = \langle g^r, my^r \rangle$, where $r \in \mathbb{G}_q$ is a local random number generated by the encrypting party. The user can then decrypt the message by calculating $Dec(\alpha, \beta) = \frac{\beta}{\alpha^x} = \frac{my^r}{(g^r)^x} = m$. ElGamal cryptosystem is multiplicative homomorphic, i.e., $Dec(Enc(m_1) \cdot Enc(m_2)) = Dec(\langle g^{r_1} \cdot g^{r_2}, m_1 y^{r_1} \cdot m_2 y^{r_2} \rangle) = m_1 \cdot m_2$. Additive homomorphism can be obtained with what is sometimes called "exponential" ElGamal, in which encryption is performed as $Enc(m) = \langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ and decryption can be obtained by $Dec(\alpha, \beta) = \frac{\beta}{\alpha^x} = g^m$. Thus, $Dec(Enc(m_1) \cdot Enc(m_2)) = Dec(\langle g^{r_1} \cdot g^{r_2}, g^{m_1} y^{r_1} \cdot g^{m_2} y^{r_2} \rangle) = g^{m_1+m_2}$. Note that since the decryption results in g^m instead of m, it is computationally intractable to obtain m from g^m due to the intractability of the discrete logarithm problem. The proposed auction scheme employs exponential ElGamal to utilize the additive homomorphic property, and only needs to determine whether m is zero which can be easily done.

- *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.*

- *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 and Computer Engineering, Mississippi State University, Mississippi State, MS 39762. E-mail: {wl373@, li@ece.}msstate.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.*

APPENDIX B
ZERO KNOWLEDGE PROOFS

The Zero Knowledge Proof (ZKP), introduced by Goldwasser, Micali and Rackoff (GMR) [2], 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 for details) using ZKPs in different steps of the protocol. We describe several ZKPs we will use in SPA as follows. All the calculations are modulo-p unless mentioned otherwise.

B.1 Proof of Knowledge of A Discrete Logarithm

Schnorr [3] develops a ZKP that a prover (a bidder) can use to prove the knowledge of x such that $y = g^x$ to a verifier (a bridge node) who knows y and g.

- The bidder chooses a random r and sends $z = g^r$ to the bridge node.

- The bridge node sends a random challenge c to the bidder.
- The bidder computes $a = (r + cx) \bmod q$ and sends to the bridge node.
- The bridge node checks to see if $g^a = zy^c$.

If the equality holds, the bidder is able to prove to the bridge node the knowledge of x such that $y = g^x$ without disclosing x.

B.2 Proof of Equality of Two Discrete Logarithms

When a prover (a bidder) needs to prove that two values (encryptions, say $y_1 = g_1^x$ and $y_2 = g_2^x$) are computed using the same private key (x) to a verifier (a bridge node who knows y_1, y_2, g_1, g_2), the protocol below [4] can be employed to realize the zero-knowledge proof.

- The bidder chooses a random r and sends $z_1 = g_1^r$ and $z_2 = g_2^r$ to the bridge node.
- The bridge node sends a random challege c to the bidder node.
- The bidder then computes $a = (r + cx) \bmod q$ and sends to the bridge node.
- The bridge node checks to see if $g_1^a = z_1 y_1^c$ and $g_2^a = z_2 y_2^c$.

If both the equalities hold, the bridge node is convinced that the same x is used to compute y_1 and y_2.

B.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. [5] to prove to the bridge node that his/her bidding vector is composed of encryptions of $m \in \{0, 1\}$. Specifically, let $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ be the ElGamal encryption of message m.

- If $m = 0$, the bidder chooses r_1, d_1, w at random and sends $\langle \alpha, \beta \rangle, a_1 = g^{r_1} \beta^{d_1}, b_1 = y^{r_1} (\alpha/g)^{d_1}$ and $a_2 = g^w, b_2 = y^w$ to the bridge node.
  If $m = 1$, the bidder chooses r_2, d_2, w at random and sends $\langle \alpha, \beta \rangle, a_1 = g^w, b_1 = y^w, a_2 = g^{r_2} \beta^{d_2}$, and $b_2 = y^{r_2} \alpha^{d_2}$ to the bridge node.
- The bridge node sends a challenge c, chosen at random, to the bidder node.
- If $m = 0$, the bidder sends $d_1, d_2 = c - d_1 \bmod q$, r_1, and $r_2 = w - rd_2 \bmod q$ to the bridge node.
  If $m = 1$, the bidder sends $d_1 = c - d_2 \bmod q$, d_2, $r_1 = w - rd_1 \bmod q$, and r_2 to the bridge node
- The bridge node checks whether $c = d_1 + d_2 \bmod q$ $a_1 = g^{r_1} \beta^{d_1}, b_1 = y^{r_1} (\alpha/g)^{d_1}, a_2 = g^{r_2} \beta^{d_2}$, and $b_2 = y^{r_2} \alpha^{d_2}$.

If all the equalities hold, the bidder is able to prove that the ciphertext decrypts to either 1 or 0.

Appendix C
The Distributed Advertisement Distribution Algorithm

Below (Algorithm 1) please find the detailed descriptions of the distributed advertisement distribution algorithm.

Algorithm 1: Distributed Advertisement Distribution

```
a_ik ← H(F_i||T_k);
v_c ← Current Node ID ;
if a_ik ∈ (predecessor(v_c), v_c] then
    Send an ACK message back to SrcID and quit;
    /* v_c is the bridge node */
end
Store MessageID, SrcID pair;
SrcID ← v_c ;
if (a_ik ∈ (v_c, sucessor(v_c)]) then
    Forward advertisement message to sucessor(v_c)
    and quit;
    /*sucessor(v_c) is the bridge node*/
end
for (∀j|∃e_cj ∈ E) do
    if (a_ik ∈ (predecessor(v_j), v_j]) then
        Forward the advertisement packet to v_j and
        quit ;
        /* Friend v_j of v_c is the bridge
        node */
    end
end
for (j = 2 → m) do
    if (a_ik = j.finger(v_c)) then
        Forward the advertisement packet to
        j.finger(v_s) and quit;
        /* j.finger(v_c) is the bridge node */
    end
end
v_next ← ∅ ;
for (j = 1 → m − 1) do
    if a_ik ∈ (j.finger[v_c], (j + 1).finger[v_c]) then
        v_next ← j.finger[v_c];
    end
end
if (v_next = ∅) then
    v_next ← m.finger[v_c] ;
end
for (∀j|∃e_cj ∈ E) do
    if (0 < (a_ik − v_j) < (a_ik − v_next)) then
        v_next ← v_j;
    end
end
Forward the Advertisement Packet to v_next.
```

Appendix D
An Example for Winning Bidder Determination

Below (Example 1) please find an example for winning bidder determination with 4 bidders, in which X repre-

Example 1 Suppose that the price vector given by a seller is $\mathbf{p} = \begin{pmatrix}150 & 140 & 130 & 120 & 110 & 100\end{pmatrix}^{\intercal}$. Assume that there are four bidders: v_1, v_2, v_3, and v_4, and their bidding prices are 140, 130, 120, and 110, respectively. Therefore, $\mathbf{b}^1 = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}^{\intercal}$, $\mathbf{b}^2 = \begin{pmatrix}0 & 0 & 1 & 0 & 0 & 0\end{pmatrix}^{\intercal}$, $\mathbf{b}^3 = \begin{pmatrix}0 & 0 & 0 & 1 & 0 & 0\end{pmatrix}^{\intercal}$, $\mathbf{b}^4 = \begin{pmatrix}0 & 0 & 0 & 0 & 1 & 0\end{pmatrix}^{\intercal}$, Then, we have

$$\hat{\mathbf{B}} = \hat{\mathbf{b}}^1 + \hat{\mathbf{b}}^2 + \hat{\mathbf{b}}^3 + \hat{\mathbf{b}}^4 = \begin{pmatrix}0\\1\\2\\2\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\1\\2\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\0\\0\\1\\2\end{pmatrix} = \begin{pmatrix}0\\1\\3\\5\\7\\8\end{pmatrix}, \text{ and } \boldsymbol{\mathcal{P}} = \left(\begin{pmatrix}0\\1\\3\\5\\7\\8\end{pmatrix} - \begin{pmatrix}3\\3\\3\\3\\3\\3\end{pmatrix}\right) \cdot \begin{pmatrix}\mathbf{R}(1)\\\mathbf{R}(2)\\\mathbf{R}(3)\\\mathbf{R}(4)\\\mathbf{R}(5)\\\mathbf{R}(6)\end{pmatrix} = \begin{pmatrix}X\\X\\0\\X\\X\\X\end{pmatrix}.$$

Since we have $\boldsymbol{\mathcal{P}}(3) = 0$, the winning price is $\mathbf{p}(w) = \mathbf{p}(3) = 130$. According to (5), we get $\mathcal{W}^1 = (\hat{\mathbf{b}}^{\mathbf{1}}(3) - 2) \cdot R_1 = (2-2) \cdot R_1 = 0$, $\mathcal{W}^2 = (\hat{\mathbf{b}}^{\mathbf{2}}(3) - 2) \cdot R_2 = (1-2) \cdot R_2 = X$, $\mathcal{W}^3 = (\hat{\mathbf{b}}^{\mathbf{3}}(3) - 2) \cdot R_3 = (0-2) \cdot R_3 = X$, $\mathcal{W}^4 = (\hat{\mathbf{b}}^{\mathbf{4}}(3) - 2) \cdot R_4 = (0-2) \cdot R_4 = X$. Thus, the winning bidder is v_1.

sents non-zero random values.

Appendix E
Proof of Theorem 1

Theorem 1: [Completeness] A legal node can always be successfully authenticated.

Proof: Note that the bridge node can obtain the public pseudo ID ρ_i from the certificate C_i and that $s_i = \tilde{g}^{1/\rho_i} \mod N$. Thus, we have

$$y^{\rho_i}\tilde{g}^{-c} \equiv (\tilde{r}s_i^c)^{\rho_i}\tilde{g}^{-c} \equiv \tilde{r}^{\rho_i}\tilde{g}^c\tilde{g}^{-c} \equiv \tilde{r}^{\rho_i} \equiv z \mod N.$$

□

Appendix F
Proof of Theorem 2

Theorem 2: [Soundness] An illegal bidder node who does not have a valid s_i can only be successfully authenticated with a negligible probability.

Proof: We observe that an illegal bidder may be able to deceive the bridge node (verifier) if $\tilde{r} + c$ is divisible by ρ_i and it sends $z = \tilde{g}^{\tilde{r}} \mod N$ and $y = \tilde{g}^{(\tilde{r}+c)/\rho_i} \mod N$ to the bridge node. The bridge node will accept the proof because

$$y^{\rho_i}\tilde{g}^{-c} \equiv (\tilde{g}^{(\tilde{r}+c)/\rho_i})^{\rho_i} \equiv \tilde{g}^{\tilde{r}+c}\ \tilde{g}^{-c} \equiv z \mod N.$$

However, the probability of this event is very low ($\sim 1/N$). For a sufficiently large N, e.g., a 1024-bit number, this probability is negligible.

Next, we prove by contradiction that an illegal bidder, without a valid s_i, cannot increase this probability. Specifically, to increase the probability of passing the authentication, an illegal bidder needs to be able to know $y = (z\tilde{g}^c)^{1/\rho_i}$ so as to let the verification condition hold. Suppose that the bidder is able to compute ρ_i-th roots y' and y'' of $z\tilde{g}^c$ for two challenges c' and c'' ($c', c'' \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$). Note that ρ_i is a prime, we have $\gcd(\rho_i, c' - c'') = 1$. Therefore, there always exist Bezout coefficients $\tilde{m}$ and $\tilde{k}$ such that

$$\rho_i\tilde{m} + (c' - c'')\tilde{k} = \pm 1 \mod N$$

Thus, by conducting the following computation,

$$\begin{aligned}\left(\tilde{g}^m\left(\frac{y'}{y''}\right)^{\tilde{k}}\right)^{\pm 1} &\equiv \left(s_i^{\rho_i\tilde{m}}\left(\frac{y'}{y''}\right)^{\tilde{k}}\right)^{\pm 1} \\ &\equiv (s_i^{\rho_i\tilde{m}}s_i^{(c'-c'')\tilde{k}})^{\pm 1} \equiv s_i \mod n\end{aligned}$$

the bidder can obtain s_i. This, however, contradicts with the assumption that the bidder does not know s_i corresponding to ρ_i. □

Appendix G
Proof of Theorem 3

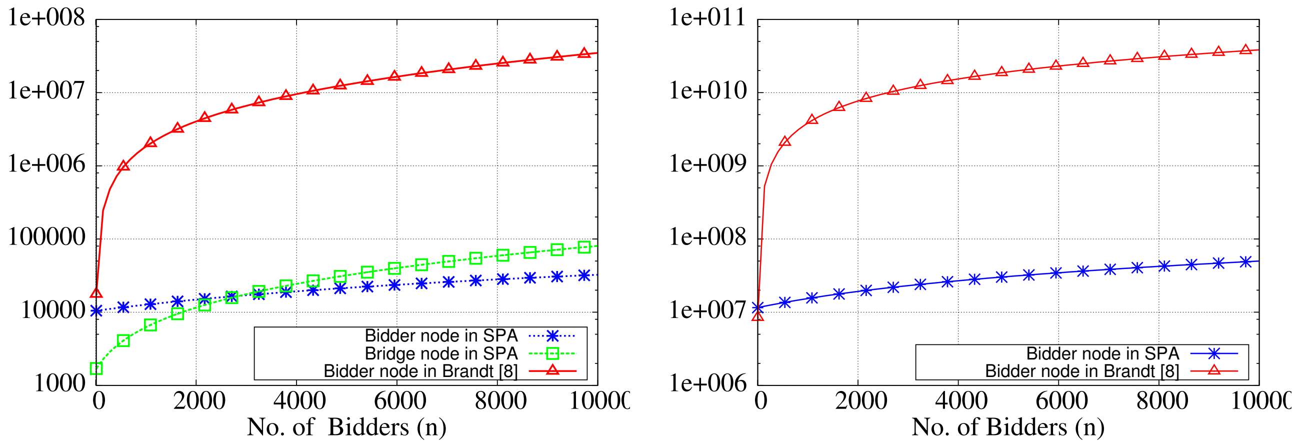
Theorem 3: If the bid from bidder v_i is authentic, the following verification equations would hold: $h(\alpha_k^i||m_{\alpha_k^i}) = \epsilon_{\alpha_k^i}$ and $h(\beta_k^i||m_{\beta_k^i}) = \epsilon_{\beta_k^i}$ for any $1 \le k \le K$.

Proof: We present the proof by dropping the superscripts/subscripts of the subscripts in the notations above for simplicity. Particularly, since $m_\alpha = y_\alpha^{\rho_i}\tilde{g}^{-\epsilon_\alpha} = (r_\alpha s_i^{\epsilon_\alpha})^{\rho_i}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i}(s_i^{\rho_i})^{\epsilon_\alpha}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i}\tilde{g}^{\epsilon_\alpha}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i} = z_\alpha$ (note that $s_i^{\rho_i} = \tilde{g}^{d_i\rho_i} = \tilde{g}$), we have $h(\alpha||m_\alpha) = h(\alpha||z_\alpha) = \epsilon_\alpha$. Similarly, we can prove that $m_\beta = z_\beta$ and hence $h(\beta||m_\beta) = h(\beta||z_\beta) = \epsilon_\beta$. □

Appendix H
Proof of Theorem 4

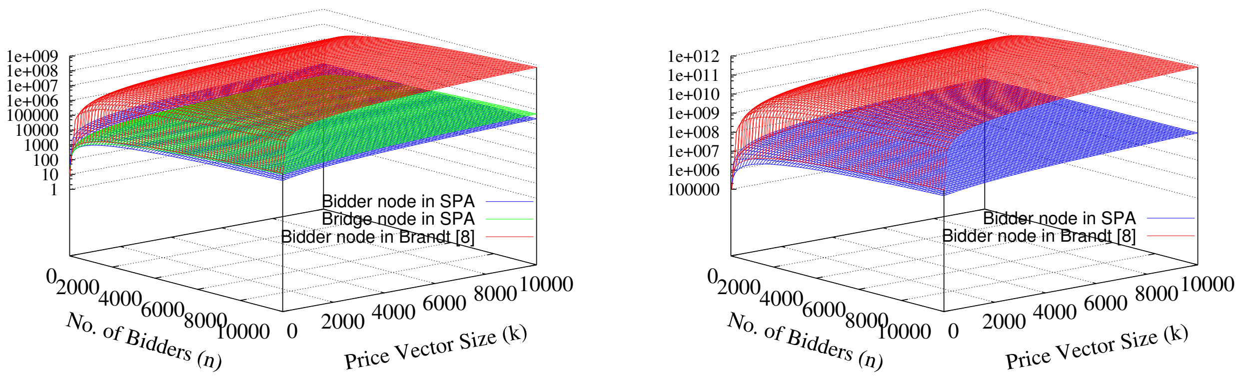
Theorem 4: [Soundness] An illegal bidder, who generates a signature without a valid s_i, can only pass the verification at the bridge node with a negligible probability.

Proof: Consider an illegal bidder v_i who signs his/her bid vector $Enc(b_k^i) = \langle\alpha_k^i, \beta_k^i\rangle$ $(1 \le k \le K)$ by following the above anonymous signature scheme. We can see from Theorem 2 that the illegal bidder can deceive the bridge node, i.e., $r_{\alpha_k^i}^{\rho_i} = y_{\alpha_k^i}^{\rho_i}\tilde{g}^{-\epsilon_{\alpha_k^i}}$ and $r_{\beta_k^i}^{\rho_i} = y_{\beta_k^i}^{\rho_i}\tilde{g}^{-\epsilon_{\beta_k^i}}$, if for each element $Enc(b_k^i) = \langle\alpha_k^i, \beta_k^i\rangle$, the illegal bidder sends $z_{\alpha_k^i} = \tilde{g}^{r_{\alpha_k^i}}$, $y_{\alpha_k^i} = \tilde{g}^{(r_{\alpha_k^i} + \epsilon_{\alpha_k^i})/\rho_i}$, and $z_{\beta_k^i} = \tilde{g}^{r_{\beta_k^i}}$, $y_{\beta_k^i} = \tilde{g}^{(r_{\beta_k^i} + \epsilon_{\beta_k^i})/\rho_i}$ where $(r_{\alpha_k^i} + \epsilon_{\alpha_k^i})$ and $(r_{\beta_k^i} + \epsilon_{\beta_k^i})$ are divisible by ρ_i. However, the probability of this event is very low $\approx 1/N$, and the probability of such events for the whole bid vector is $\ll 1/N$ and



(a) Computation Cost (ms) (b) Communication Cost (bits)

Fig. 1. Computation and communication costs when K=500



(a) Computation Cost (ms) (b) Communication Cost (bits)

Fig. 2. Computation and communication costs when $n = 10000$.

is negligible. Similarly, following the proof in Theorem 2, we can show that a malicious bidder is unable to increase this probability. Thus, a signature generated by an illegal bidder without a valid s_i has only a negligible probability of being successfully verified by the bridge node. $\square$

Appendix I
Performance Evaluation

This section details the experiment results for the computation, communication, storage costs, and auction utility of our proposed protocol, which are shown in Fig. 1, Fig. 2, Fig. 3, Fig. 4, respectively.

References

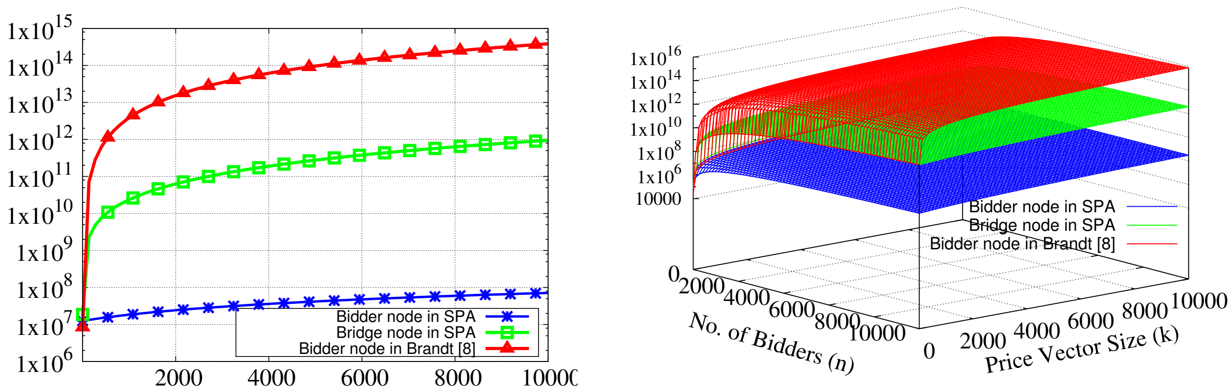
[1] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[2] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, New York, NY, USA, 1987.

[3] C. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[4] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, 1993.

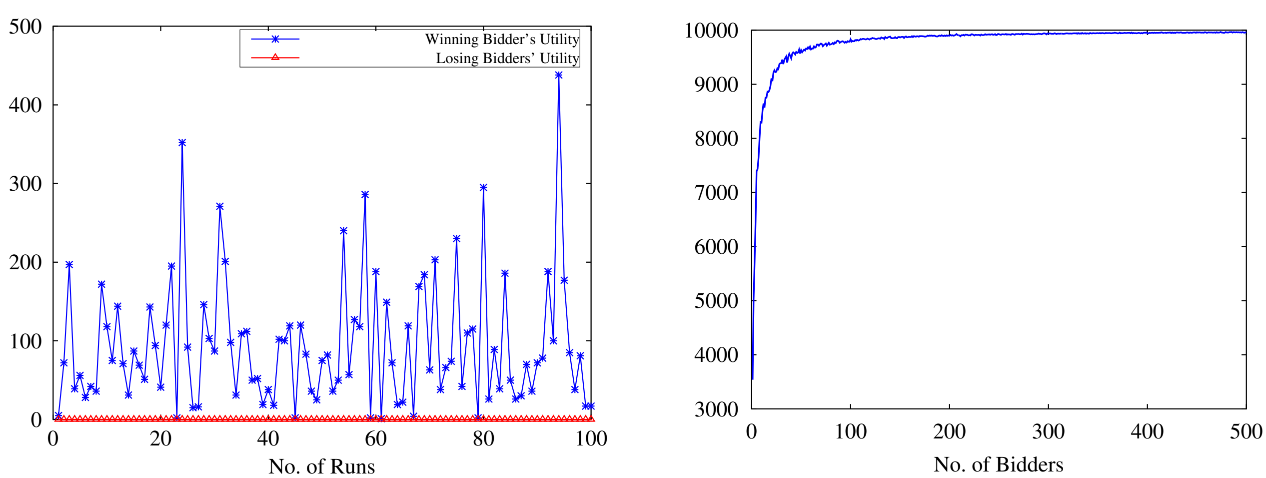
[5] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Proceedings of the EUROCRYPT*, 1997.



(a) Storage cost (bits) when $K = 500$

(b) Storage cost (bits) when both n and K vary

Fig. 3. Storage cost



(a) Bidders' Utility

(b) Seller's Average Revenue

Fig. 4. Auction Utility