Anyone Can Hold An Auction

Ben Palmer, Kris Bubendorfer and Ian Welch
Victoria University
Wellington, New Zealand.
{ben,kris,ian}@mcs.vuw.ac.nz

***Abstract*—Secure auctions have many potential uses including eVoting, computational resource allocation and FCC spectrum auctions.The SGVA privacy preserving auction scheme is able to conduct combinatorial auctions and keep the losing bid values secret. However, SVGA is a black box so users have no means to assure themselves that the auction has actually taken place and that their bid has been included in the computation of the result. We have designed a verification scheme composed of zero knowledge proofs that extends the basic SGVA secure auction protocol to permit users to verify offline that the protocol has been executed correctly. This is especially useful in environments where participants have no pre-existing trust, such as the Internet.**

I. Introduction

Suppose Alice is running a sealed bid auction of artwork for a charity organisation. She plans to hold the auction on her web site hosted by Sam. Bob and Jim submit bids to the auctioneer as shown in Figure 1.

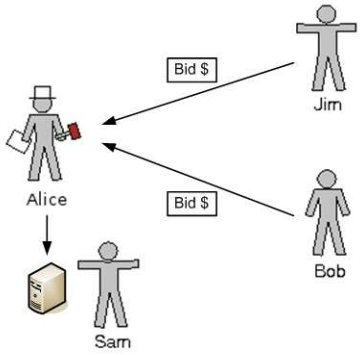


Fig. 1. Art Auction

There are several potential problems with this auction. Either Sam or Alice can peek at the bids and many bidders would prefer their bids to remain private. Alice could also refuse to count certain bidders bids in the auction. Alice could easily defraud the charity organisation by either arbitrarily choosing a winner regardless of the bid values, or by choosing the winner correctly, reporting a reduced winning price, and taking the difference herself. A large amount of trust is placed in Alice with no way of checking whether she has correctly executed the auction.

We can prevent Alice from breaking privacy guarantees by using a privacy preserving auction where the values of bids are hidden by using some form of encryption or obfuscation. Figure 2 shows Alice holding a privacy preserving sealed bid auction on her web site hosted by Sam. Bob and Jim submit encrypted bids to the auctioneer.

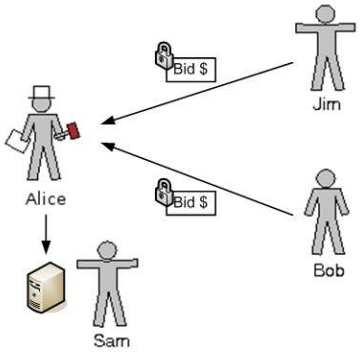


Fig. 2. Privacy Preserving Art Auction

In this auction, Alice and Sam are prevented from being able to peek at bids due to the encryption which solves one of the problems with the original auction. There has been a large amount of recent work in privacy preserving sealed bid auctions [1]–[7]. Auction protocols have been developed that are capable of conducting auctions involving multiple goods and bidders all while using cryptographic techniques to keep losing bid values secret from both auctioneers and other bidders. In a competitive marketplace privacy preserving auctions are a valuable tool as competing parties will not want their bids that contain commercially sensitive information made public.

The security of a privacy preserving auction can be further enhanced with the addition of a verification scheme. A verifiable privacy preserving auction is shown in Figure 3. A verification scheme allows bidders and other third parties to verify that the auction was executed correctly. Anyone can hold an auction as the verification scheme provides confidence in an auction result even when the auctioneer is an untrusted party. The extra protection provided by verification gives the bidders confidence that their bids have been counted and that the auction result is correct. The charity group could also verify the auction to make sure they are getting the correct amount of money from Alice. The combination of verification and privacy preservation significantly reduce the trust placed in Alice. This is an improvement over the current state of online auctions where auctions are usually conducted by some central

trusted party such as eBay or trademe.

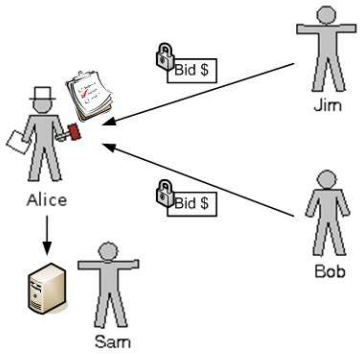


Fig. 3. Verifiable Privacy Preserving Art Auction

Revealing all the bid values to the bidders and allowing them to rerun the auction is the trivial verification solution, however it removes the privacy preserving property of the auction and gives individual bidders an advantage in future auctions as they have additional information about the value other bidders place on the goods. Therefore, a verification mechanism must allow verification but still preserve the secrecy of bids from individual bidders and even from auctioneers. Some privacy preserving auctions support some form of verification [2], [4], [5], [7] ranging from complete verification to verification of certain aspects of the protocol, while others have no verification [1], [3], [6].

This paper introduces the verification protocol we have added to the privacy preserving SGVA auction protocol by Suzuki and Yokoo [1]. The SGVA protocol was chosen as it supports efficient combinatorial auctions, where bidders can bid on combinations of goods, and keeps losing bid values secret. However, it is not verifiable without revealing all the bid values to bidders. Our modified protocol uses zero knowledge proofs, the properties of homomorphic encryption, and a public bulletin board to verify the auction protocol. We break down the auction protocol into different phases and allow verification of each individual phase. Auctioneers can verify that bidders have submitted valid bids and the bidders can verify their bids have been counted and that the correct bid was chosen as the winner without revealing its actual value or the value of any of the other bids.

II. Zero Knowledge Proofs

Zero Knowledge proofs were first introduced by Goldwasser, Micali, and Rackoff [8] and are used to prove some statement, without revealing any other information other than what is known before the proof was executed. A zero knowledge proof takes part between a prover and a verifier and typically consists of a commitment from the prover, a challenge from the verifier, and a response from the prover.

Figure 4 is an example of a zero knowledge proof. The prover Alice can convince the verifier Bob that she knows the secret password to open the door between R and S without revealing it by completing the following steps:

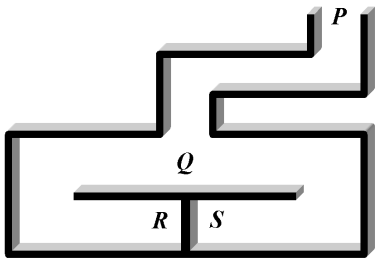


Fig. 4. Ali Baba's Cave [9]

- Bob waits at P while Alice goes to either R or S (commitment).
- Bob goes to Q and calls out for Alice to come out from either the right or left side (challenge).
- If Alice does not know the secret words, there is a 50% chance she will come out from the wrong tunnel.
- Bob can then repeat as many times as he wants to convince himself that Alice knows the secret word, but Bob will never learn the secret word himself.

Traditional zero knowledge proofs involve the interaction of the prover and verifier. In an auction setting, this adds to the communication complexity of the solution. This is especially true if every bidder wanted to verify the auction, in which case every bidder has to communicate with the auctioneer to verify any statements. Additionally, the auctioneer has to generate a new proof for each bidder adding to the computational complexity. Non-interactive zero knowledge proofs, introduced in [10], allow us to avoid this problem by allowing a proof to be published once by the auctioneer, and verified at a later time by any third party without interaction with the auctioneer. We use the Fiat-Shamir heuristic [11] and SHA512 hash function to make the proofs non-interactive zero knowledge proofs of knowledge in the random oracle model.

III. Auction Protocol

The SGVA auction protocol developed by Suzuki and Yokoo [1] uses homomorphic encryption to hide the bid values and dynamic programming to allow distributed calculation of the optimal solution. It involves a seller who initiates the auction, a group of auctioneers that securely compute the output of the auction, and bidders that bid on the available goods. A graph is constructed of the auction with the edges representing the available combinations of goods. A path through the graph represents a possible allocation of goods. For example, Figure 5 represents a graph for a two good auction. The available combinations of goods are $\{1\}, \{2\}, \{1, 2\}$ and the two possible allocations of goods are $\{1\}\{2\}$ and $\{1, 2\}$.

The algorithm for computing the auction is detailed in Figure 6. The seller first advertises the auction details including the details of the goods and the number of possible prices. The auctioneers involved in calculating the auction result then

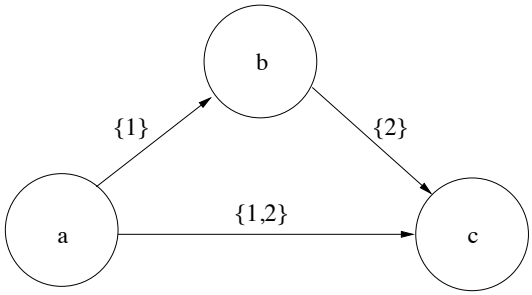
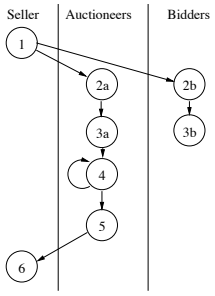


Fig. 5. 2 Good Auction Graph

publish their public El-Gamal keys, and any interested bidders register to bid on the auction. Every auctioneer is associated with a node in the graph and the bidders encrypt and publish bids with the public keys of the auctioneers. Then for every node in the graph starting at the end node, the auctioneer responsible for the node calculates the maximum bid for that node and, if there is a previous link in the graph, adds this value on to the value of the previous link using an operation called shift and randomise. When the first node in the graph is reached the result is the optimal value for this auction. The optimal value is then traced back through the auction graph to find the optimal allocation of goods to bidders.



Round	Action
1	Advertise auction, arrange auctioneers
2a	Publish public keys
2b	Register to bid
3a	Wait for bids
3b	Publish encrypted bid vectors
4	Start from the end node and for every node: – Find the optimal value for this node – Perform a shift and randomise to add on the value from the previous node
5	Trace the optimal value through the graph to find the optimal path
6	Publish winning bidders and prices

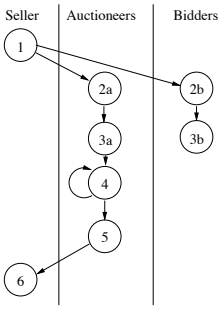
Fig. 6. Auction Algorithm

IV. Verification Protocol

To verify the auctioneers we use a read only bulletin board, where auctioneers and bidders publish information that is signed by them so others can verify who the data was published by, and a combination of non-interactive zero knowledge proofs. The zero knowledge proofs are used to confirm that every step of the auction process has been carried out correctly while not leaking any other information.

The verification protocol uses some well known proofs including the proof of knowledge of a discrete logarithm [12], proof of equality of discrete logarithms [13], proof that an encrypted item decrypts to one of two items [14], a plaintext equality test, and a verifiable shuffle of encrypted items [15]–[17]. We have also constructed zero knowledge proofs of knowledge out of the proofs above for verification of a valid bid, verification that the shift and randomise operation was carried out correctly, and verification of the maximum bid from a group of bids. The use of zero knowledge proofs of knowledge keeps the values of the bids secret while allowing verification of the auction protocol.

The algorithm for computing a verifiable auction is detailed in Figure 7. In the verifiable protocol, the bidders and auctioneers have to publish a proof that the bid vectors are valid in Round 3. The auctioneers also have to publish proofs that the shift and randomise operations and the maximum bids have been correctly computed in Rounds 4 and 5. These proofs can then be verified at a later time by any third party.



Round	Action
1	Advertise auction, arrange auctioneers
2a	Publish public keys
2b	Register to bid
3a	Publish proofs that the bid vectors are valid
3b	Publish encrypted bid vectors and a proof that every item in the bid vector decrypts to 1 or Z
4	Start from the end node and for every node: – Find the optimal value for this node and publish a proof it is the maximum bid – Perform a shift and randomise to add on the value from the previous node and prove the shift and randomise is correct
5	Trace the optimal value through the graph to find the optimal path and publish proofs it is the maximum path
6	Publish winning bidders and prices

Fig. 7. Verifiable Auction Algorithm

V. Maximum Bid Proof

In the SGVA auction scheme by Suzuki and Yokoo [1] bids are represented by bid vectors. These bid vectors contain an encrypted Z or 1 depending on the value of the bid. For example, with a bid vector of length 4 an encrypted bid vector with a value of 2 is represented as $(E(Z), E(Z), E(1), E(1))$.

Suppose we have three bids for a good:

- $Bid_1 = (E(Z), E(Z), E(Z), E(1))$
- $Bid_2 = (E(Z), E(Z), E(1), E(1))$
- $Bid_3 = (E(1), E(1), E(1), E(1))$

In this example, Bid_1 has value 3, Bid_2 has value 2, and Bid_3 has value 0. We want to provide a zero knowledge proof that Bid_1 is the maximum bid of the group without revealing what the maximum bid is or any other properties of the losing bids.

The auctioneer will first perform a shuffle of the bids, $\pi = (3, 4, 1, 2)$. Once this shuffle has been applied we will have:

- $ShuffledBid_1 = (E(Z), E(1), E(Z), E(Z))$
- $ShuffledBid_2 = (E(1), E(1), E(Z), E(Z))$
- $ShuffledBid_3 = (E(1), E(1), E(1), E(1))$

Now the auctioneer uses a proof of a shuffle of El Gamal encrypted values to prove that the shuffle was correctly carried out, while not revealing the actual shuffle used [17].

The final step is for the auctioneer to prove that the first item in $ShuffledBid_1$ decrypts to Z while the other bids decrypt to 1 using a proof of knowledge of a discrete logarithm [12]. This proves that Bid_1 is the highest bid while revealing nothing about the values of the other bids because the verifier does not know the shuffle that was used.

VI. Conclusion

This paper introduces our verification scheme for the SGVA secure combinatorial auction protocol. Our verification scheme uses a combination of zero knowledge proofs to enable any party to confirm the auction has been conducted correctly and the correct bidders named the winners. Anyone can hold an auction by using the SGVA auction protocol with our verification scheme even in domains where there is no pre-existing trust as the bidders have confidence that the auction was correctly executed without having to trust the auctioneer. Although the verification protocol adds additional complexity to the standard protocol we believe it is practical to implement and use. Our future plans include the implementation of the full verification scheme for our existing SGVA implementation so that we can quantify the performance of the verification.

References

[1] M. Yokoo and K. Suzuki, "Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions," in *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, 2002.

[2] H. Kikuchi, "(m+1)st-price auction protocol," in *FC '01: Proceedings of the 5th International Conference on Financial Cryptography*. Springer-Verlag, 2002, pp. 351–363.

[3] K. Suzuki and M. Yokoo, "Secure combinatorial auctions by dynamic programming with polynomial secret sharing," in *Sixth International Financial Cryptography Conference (FC-02)*, 2002.

[4] F. Brandt, "How to obtain full privacy in auctions," *International Journal of Information Security*, vol. 5, no. 4, pp. 201–216, 2006.

[5] H. Lipmaa, N. Asokan, and V. Niemi, "Secure vickrey auctions without threshold trust," in *FC'02: Proceedings of the 6th Annual Conference on Financial Cryptography*. Springer-Verlag, 2002, pp. 85–101.

[6] M. Yokoo and K. Suzuki, "Secure generalized vickrey auction without thirdparty servers," in *Eighth International Financial Cryptography Conference (FC-2004)*, 2004.

[7] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 1999, pp. 129–139.

[8] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM J. Comput.*, vol. 18, no. 1, pp. 186–208, 1989.

[9] R. Laboratories., "Rsa laboratories crypto faq." [Online]. Available: http://www.rsa.com/rsalabs/node.asp?id=2178

[10] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, 1988, pp. 103–112.

[11] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology — Crypto '86*. New York: Springer-Verlag, 1987, pp. 186–194. [Online]. Available: citeseer.ist.psu.edu/fiat87how.html

[12] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *EUROCRYPT '89: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1990, pp. 688–689.

[13] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1993, pp. 89–105.

[14] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *EUROCRYPT '97: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*, vol. 1233. London, UK: Springer-Verlag, 1997, pp. 103–118.

[15] J. Groth, "A verifiable secret shuffle of homomorphic encryptions," in *PKC '03: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*. London, UK: Springer-Verlag, 2003, pp. 145–160.

[16] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*. New York, NY, USA: ACM Press, 2001, pp. 116–125.

[17] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle," in *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 2001, pp. 368–387.