Secret Sharing in Sealed-Bid Auctions

CAT 6427 Secret Sharing Protocols - Spring 2017 Project Report

Maciej Medyk
Computer Science Graduate Student
Florida Atlantic University
Boca Raton, United States

Abstract — **With expansion of mobile devices and online access to e-commerce services, more transactions including all types of auctions are now done online. Security and privacy have become critical issues in design of those auctions and various schemes were introduced to ensure secure implementation of sealed bid auctions that are based on secure multiparty computation or variety of secret sharing protocols. In this paper, my motivation is to explain the auction process and provide comprehensive evaluation most popular secret sharing techniques used in auctions to provide security and privacy to the bidders. In those auctions bids are never revealed to any bidder, regardless if the auction is concluded. In this paper, I will primarily concentrate on First Price and Second Price Sealed-Bid auctions and review their applications and methods used in implementation successful bidding system.**

I. Introduction

Since the early 2000s, more and more businesses have been turning to online services to provide their more features to the public including facilitations of business transactions. This allowed the auction services to flourish as bidders didn't have to travel to the same location as in classical auctions and could display their interest and produce the bid from anywhere in the world. For this reason, we saw explosion of services like eBay, uBid, and GovDeals, but auction systems have also been introduced not only to direct consumers, but also vendors that tend to bid for contracts and services. Auctions are divided to three categories which include absolute auction, reserve auction, and minimum bid auction. An English auction is the best known and most common type of auction, were participants continuously bid for a higher price and the bidding ends once one person makes the highest bid that no other bidder wants to surpass. This is a model on which eBay and many consumer auctions were built with small modifications of creating a closing period for the auction. However; in this model, the bids are exposed and visible to everybody and bidder may submit multiple bids. Another well-known auction is Dutch style auction in which auctioneer starts with a very high price and reduces it with time until bidder places the bid. In this type of auction the there is no up bidding, but it is very time constrained as anytime bid can be lost and for online auctions it is not as feasible as English auction or sealed-bid auctions [1]. Therefore, online we can see a majority of auctions being English or sealed-bid. In sealed-bid auctions the participants submitted their bids through private channels, either sealed envelopes or encrypted, and at the end of the auction the winning bid is revealed [2]. Due to security issues and various business model needed discretion, sealed-bid auctions became very popular and pushing even eBay to adopt private auctions. In this paper, we will discuss the way sealed-bid auctions are performed, how they are adopted online, how security of bids is preserved, and how secret sharing protocols are used in implementation.

II. BACKGROUND

A. Sealed-Bid Auction

Auction has been an excellent method of allowing sellers to trade goods at the highest possible price based on market conditions. Sealed-bid auctions due to the way that is constructed allow a person honest consideration about how much is the item worth to them at that time. There is no up bidding the person as it happens in open auction, but simply bidder establishes his own price equilibrium. Auction do also have their issues with favors and fairness. In classical auctions, auctioneers have often been corrupt and have repudiated or adjusted the bids in favor for certain bidders. On the other hand, online auctions give bidder additional privacy and confidence guaranteeing that the process is secure and fair. The sealed-bid auctions include two main types which are first-place sealed bid auction and second-place sealed bid auction. In the first place sealed -bid auction, all bids are submitted at the same time. All the bids are sealed or encrypted so no participant can know what another person was bidding. Once all the bids are analyzed the bidder with the highest bid is selected. The bidder at that moment pays the amount that he put on the bid. In the second place sealed-bid auction, the bidding procedure happens very much in the same way. Again, all bids are submitted at the same time and all bids are sealed. One big difference is that once the highest bidder has been identified, he no longer pays the amount he actually bid, but pays the next closest amount chosen by the runner-up bidder [3,4].

B. Auction Properties

The following properties and security requirements are often desired in any sealed-bid action:

i. Correctness – the auction outcomes are determined according to all auction rules. If the second price bid auction is administered, then the bidder with the highest bid will pay the second highest bid placed.

ii. Confidentiality – Secrecy of the bid is of outmost importance. Each bid remains confidential to everyone other than the bidder himself and all loosing bids remain confidential even after winning bid is announced, otherwise it is easy to manipulate the auction and divulge information about current status of the bid to a specific bidder.

iii. Fairness – Bidders must not be able to modify or deny any bids of other bidders once they are submitted. Bidders must not be able to interfere in bidding process of others.

iv. Verifiability – All bidders must be able to verify the correctness of the auction outcomes of the auction to make sure the process wasn't compromised by any collusion.

v. Privacy – All personal information about all participants remains secret and depending on the auction the personal information about the winner may also remain a secret. Anonymity is essential in order to make sure there is no favorism in choosing the winner.

vi. Unchangeability – once a bidder submits the bid he should not be able to change it and if bidder winds the bid and does not have ability or desire to pay, then may not deny the bid he submitted. This ensures that no up bidding is present and that bids are not manipulated based on new information.

vii. Robustness – Auction must be able to run properly in any circumstances, even with existence of malicious behaviors and invalid bids. Robustness is necessary to make sure there is no manipulation by a compromised server.

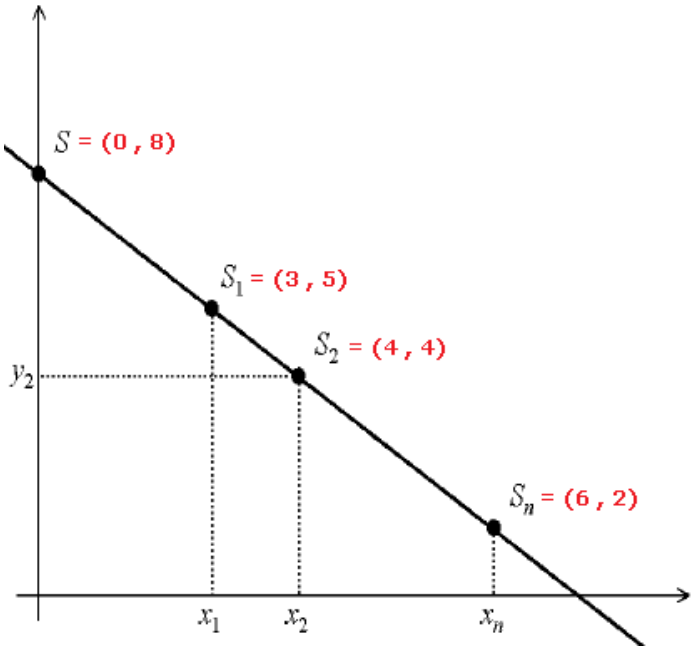
Those properties must be set in order to avoid common problems with sealed bid auction like collusion between auctioneers and a specific bidder, closing time manipulations by the insider, bid diversions, or bid being awarded to a losing bidder [3,4,5]. Security measures and secret sharing protocols have been created to adopt those principles in sealed-bid auctions.

C. Security Methods

Modern online auctions adopt many different measures to provide security. In secret sharing schemes the bid is divided into shares {S_1, S_2, S_3, S_4, …, S_n} using polynomial functions in order to conceal the actual bid amount and those shares are later distributed. The shares then are sent to the auctioneers that collect them until the end of the auction. Afterwards the auctioneers send all the shares collected to the computational party to determine the winner. This party combines all the public shares issued by each bidder and collected all by the auctioneers using LaGrange Interpolation method to retrieve the secret. Once all the bids are reconstructed they are compared to determine the winner. The formula for Lagrange interpolation where secret a_o can be retrieved from summation from 1 to t (threshold) of product c_i and y_i where y_i is the share.

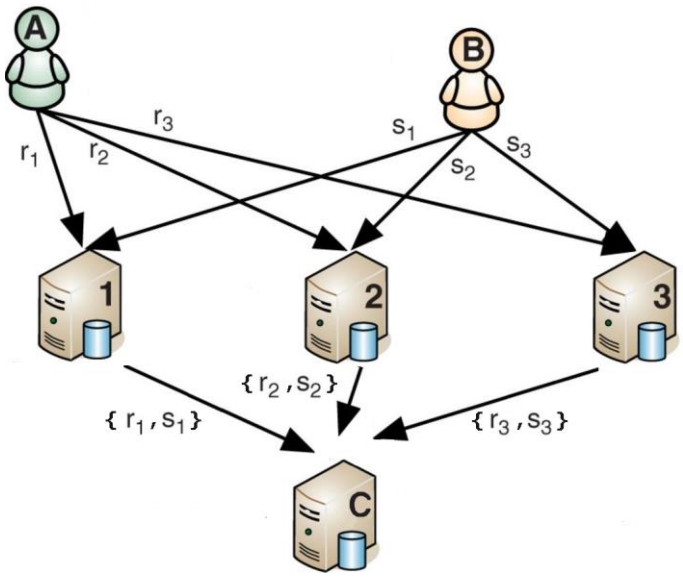
$$f(0) = a_o = S = \sum_{i=1}^{t} c_i y_i, \; where \, c_i = \prod_{1 \le j \le t, j \ne i} \frac{x_j}{x_j - x_i}$$

The Shamir's secret sharing method can be easily explained by graph below. If the bid is 8 and bidder uses function f(x) = 8 – x then he can create shares and give share of 5 to third server and give share 4 to forth server. Any single server by itself cannot determine what their bid is because if server has only one share which translates to one point on this line then he has infinite possibilities of reconstructing that line. To reconstruct the line the server will need at least two points of the line and in the example below we need two points (3,5) and (4,4) in order to determine what secret y value is hidden when x equals to 0. In our example y is equal to 8. As the complexity of the polynomial grows, more points will be needed in order to reconstruct the secret, which is determined by threshold of the polynomial, highest degree encountered.



This scheme is applied to the auction model in order to distribute the share in order to hide the secret. The illustration below shows a very simplistic image of how the secret is hidden within shares and how is distributed to auctioneers and later how is it reconstructed. In this example bidder A would create shares R_1, R_2, R_3 and bidder B would create shares S_1, S_2, S_3. Auctioneers will only receive shares that they were assigned. In example server 1 will only receive share R_1 from bidder 1 and share S_1 from bidder 2. At the end of the auctions all shares will be sent to server C which will do computation on all shares received. If one of the auctioneers will be compromised or disabled, using

LaGrange Interpolation, if threshold was lower than number of auctioneer servers, the secret still can be retrieved and winner determined [7, 20].



Shamir scheme was a basis for other schemes like verifiable secret sharing where bidders can verify the consistency of their shares with other bidders shares by either applying zero knowledge proof or applying bivariate polynomials. There are other security methods other than secret sharing protocols, which are employed to provide security to online auctions and amongst them is homomorphic secret sharing where some arithmetic operations can be done without decrypting the ciphertext. Another security measures employed is secure multiparty computation method which can use sorting algorithm in order to determine first price winner in sealed bid auctions with presence of malicious participants. Lastly, there are other security measures that are sometimes employed using dining cryptographer's problem, hash functions, or knapsack problem. We will now discuss those methods in detail and how they apply to online sealed-bid functions.

III. Main Body

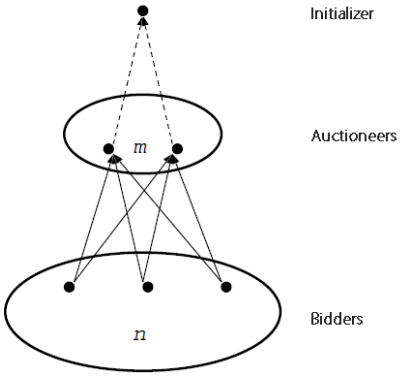
A. Verifiable Secret Sharing

Verifiable secret sharing is a protocol that has built on Shamir foundation. In this protocol, the bidder uses symmetric bivariate polynomials where function f(0,0) equals to the secret. To hide the secret in a symmetric polynomial function has to have same amount of x of same power as y. Example of symmetric polynomial would be $f(x,y) = 30 + 3xy + 2x^2y^2$ where secret is 30 and weight is introduced for value of y. So, player 1 would have generated shares for player 2 and 3 that would be equal to f(2,1) = 44 and f(3,1) = 57. On the other hand, player 2 would create shares for player 1 and 3 that would be equal to f(1,2) = 44 and f(3,2) = 120. Lastly a player 3 would create shares for player 1 and 2 that would be equal to f(1,3) = 57 and f(2,3) = 120. Using private channels the shares would be share for verification so player 2 would send to player three share of 120 to be verified. Verification matrix should show shares this way. Let's assume there was no modular reduction since prime is 127.

	P1	P2	P3
P1	0	44	57
P2	44	0	120
P3	57	120	0

The auction is started by an initiating party that is no longer needed for the remainder of the auction. Many protocols assume thrusted authority but at small, additional computational cost the trusted party

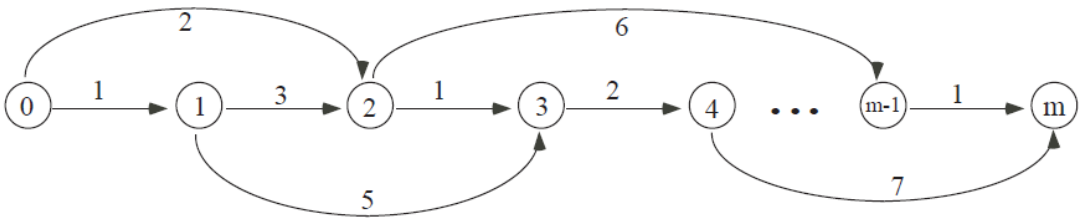
may not be required. Each bidder will act as the dealer and auctioneers will perform the gathering of secrets and computation. In this protocol, the impact of collusion between the auctioneers where they can collude to recover secret bids as well as influence of dishonest bidders with inconsistent shares where they can send incorrect shares to disrupt the auction can be reduced. This is known as active adversary model where participants deviate from protocols and exhibit malicious behavior [2, 7, 9]. To prevent to collusion amongst the auctioneers the threshold has to be equal or higher than expected number of malicious servers. For this model to work assumption is made that dishonest auctioneers are less than fifty percent of all auctioneer servers and less than the threshold of the secret sharing polynomial. To address the dishonest bidders, send their incorrect shares during bid submission or colluding auctioneers send incorrect shares we can use error correction technique through use of Reed-Solomon codes to process interpolation. In this scheme, dishonest bidders alone cannot disrupt the auction process as they are not involved will all parts of the auction. Auctioneer to behave maliciously have two actions that they can undertake. The protocol besides security also focuses on privacy and anonymity of bidders and it often uses addition or multiplication operations for disguise the actual bid value [7, 8]. This also allows to release the masked values of the bids for comparison without releasing actual values of the bids. The model below illustrates the auction process.



Sealed-bid auction is divided into three phases: bid initiation, bid submission, and determination of the winner. In initiation stage the the initializer initiates the secret sharing scheme using symmetric polynomial and distributes the shares to auctioneer. The initializer leaves the scheme once the initiation process is complete. During bit submission, each bidder chooses random symmetric polynomial and sends the shares to auctioneers through a private channel. To verify the distributed shares pairwise checks are performed upon which the shares will either be accepted or disqualified. In winner determination phase auctioneer adds the constant value to all received shares to mask the value. Due to bid integrity, the prime modular has to be selected that is large enough that the bid would not be affected by modular reduction during any arithmetic operations. At that time auctioneers start sharing their shares with each other and all computations performed are visible to auctioneers. Once computation is done the shares are sorted in decreasing order and he winning bid is found at index 0 and depending on the type of auction if its first place sealed-bid auction the winner will pay the bid at index 0 minus the constant value that was added. If it was second place sealed-bid auction the winner at index 0 will pay the bid at index 1 minus the constant value. Auctioneers at that moment sent the winner shares minus the constant value to all bidders which will compute locally the selling price. At the end of the protocol the all that bidders know is auction outcome [2, 7, 8, 9]. In this protocol, all seven properties have been maintained. The process would correctly determine the winner while maintaining verifiability when bidders would locally verify the winning bid. The confidentiality of the bid would be maintained as only winning bid would be share and losing bids would not be shared. Privacy of all bidders would also be maintained. The bidders would submit only one bid maintaining unchangeability property and the bid would be fairly verified. Lastly the security features of verifiable secret sharing provide robustness and adaptability to malicious behavior.

B. Homomorphic Secret Sharing

Homomorphic secret sharing protocol is quite unique as it is designed to perform arithmetic operations on ciphertext while encrypted. There are two types of homomorphic encryptions: fully homomorphic and partially homomorphic encryption. At this moment, most of the protocols use partially homomorphic encryption rather than fully homomorphic due to the better efficiency of operations. Major difference between both encryptions is that on fully homomorphic encrypted cyphertext you may run and rerun many arithmetic operations without decryption, while for partially homomorphic encryption, those operations are usually limited to one operation. Therefore, if you want to conduct an addition operation once on cyphertext then either one of the methods would work. There are many partially homomorphic encryptions available and the most common ones are El Gamal or Paillier, or Goldwasser-Micali encryptions [10, 11, 19]. Encryption uses random number for making the encryption indistinguishable. An example of this is that if a person wants to encrypt a bid of 5 dollars and uses random number 5 and prime modulo number of 23 then encryption calculation would look in the following way (5^5 mod 23, and 2^5 x 5 mod 23) = (20,22); however, if we want to encrypt same bid of 5 dollars using random number of 6 calculation would be as follows (5^6 mod 23, and 2^6 x 5 mod 23) = (8,21). Therefore, encryption give you two different outcomes on same price depending on weight used in encryption [10, 12, 13]. In the sealed-bid auction using homomorphic encryption the bid is randomized and encrypted as a vector using ciphertext. While encrypted the constant is added to the bid price to further hide the bid amount. Finding a maximum bid is done by comparing components of the vector and using dynamic programming. The illustration below shows a basic concept of dynamic programing that is used to determine the most optimal path from 0 to m.



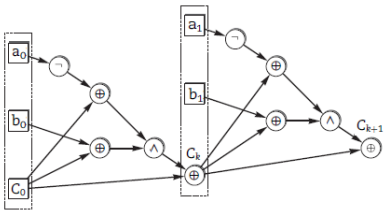
In this case, the best path that results in best value is 0 to 1 then from 1 to 3 then from 3 to 4 and lastly from 4 to m. This path results in value of 15 as you add all the factors on the path 1 + 5 + 2 + 7. In this protocol, all seven properties have been maintained. The process would correctly determine the winner while maintaining verifiability when bidders would locally verify the winning bid. The confidentiality of the bid would be maintained along with privacy of all bidders. The bidders would submit only one bid maintaining unchangeability and due to homomorphic encryption security would be maintained.

C. Secure Multiparty Computation

Typical secure multiparty computation method is built on Shamir's threshold linear secret sharing scheme where parties can evaluate multitude of linear combinations by manipulating the received shares. The aim of secure multiparty computation to perform distributed computing task in total privacy providing correctness even when under attack from external entity. The model mostly supports static adversary model where parties are corrupted before auction begins rather than mobile adversary model. The model also supports active adversary model where there is a malicious behavior of bidders and it employs commitment transfer protocol which allows for transfer of secret from one party to another, commitment sharing protocol which allows to share secret in verifiable way, and commitment multiplication protocol which allows to prove that committed secret shares. Secure multiparty computation is a method that supports many arithmetic functions like addition, multiplication, etc. In auction model, the model must support comparison function which is essential

for determining and verifying the winning bid [14, 15]. This function would be based on Boolean functions using AND, NOT, and XOR gates. The main purpose of the comparison function is to determine if the which one of two numbers is larger. Depending on the auction those numbers would be stored in sorted array and winning bid would be announced after computation is finished. The comparison will be done through binary representation of bids in vector format. The method would be constructed that output is 1 if A is greater than or equal to B or output is 0 if A is smaller than B. The illustration of such method is shown below [14, 16].

$$a_{n-1}2^{n-1} + \ldots + a_1 2^1 + a_0 2^0$$
$$b_{n-1}2^{n-1} + \ldots + b_1 2^1 + b_0 2^0$$
$$A = B + (2^n - 1 - A) + 1 = B + \overline{A} + 1$$
$$C_{k+1} = ((B_k \oplus C_k) \wedge (\neg A_k \oplus C_k)) \oplus C_k$$



Auction involving secure multiparty computation would begin using Shamir's based secret sharing algorithm to take the bids and create shares. Once bidder submit all the shares the secure multiparty computation would take place that would produce the result. In this protocol, all seven properties have been maintained. The process would correctly determine the winner while maintaining privacy of all losing bidders and the losing bids; therefore, upholding confidentiality and privacy properties. The bidders would submit only one bid maintaining unchangeability property and the bid would be fairly verified. Lastly the security features provide robustness and adaptability to malicious behavior. The bid can be also verified when secure multiparty computation releases the result and shares to all the bidders for verification.

D. Knapsack

Knapsack or rucksack problem is a well-known and studied combinatorial optimization problem that tries to optimize the most optimal value. This problem has been studied since 1897 and it has been popular problem in computer science, complexity theory, and cryptology. As knapsack problem evaluates various values and chooses the most optimal value it naturally fits the model for financial transactions and maximization of value and it therefore it has been applied to certain models of sealed-bid auctions. The auction in this model is split into initiation, bid creation, creation of shares, and winner determination. In initiation phase prices are being introduced along with the index of prices and modulo prime that would be split into smaller numbers based on how many bidders will participate in the bidding process. Once the bidders choose the price they want to use to bid for the item they use the index and modulo prime share in order to mask their bid [17]. To illustrate we will have nine prices, nine corresponding indexes, and four bidders. So, price of 10 will have index value of 5, while price of 350 will have index value of 120. We will generate modulo prime number equal to 1987 and create 4 shares that in summation would amount to the same value and we will spread it between the bidders. So, if bidder 1 chooses the price of 10 he will get a modulo share of 900 and use index of 5.

prices	10	100	200	250	300	350	400	450	500
indexes	5	9	15	30	60	120	250	500	1000

modulus	1987	>>	900	700	300	87

Once the modulo shares are distributed and prices are chosen amongst the bidders they will compute their bid. Bidder one for example will choose price of 10 so their calculation will be as follows. He will add index 5 to his modulo share of 900 that would equal to 905. Second player would use index of 15

and add it to modulo share of 700 that would equal to 715 by choosing price of 200. The rest of the shares are shown in illustration below.

	actual		index		mod		
bidder 1	10	=	5	+	900	mod 1987	905
bidder 2	200	=	15	+	700	mod 1987	715
bidder 3	450	=	500	+	300	mod 1987	800
bidder 4	350	=	120	+	87	mod 1987	207

Once all the bids are formulated then each bidder splits his bid into using additive shares in order to distribute their shares amongst other bidders. So, first bidder will take his bid of 905 and create 4 shares that in summation will equal 905. Sample shares will be 200, 200, 200, 305. All other bidders will do the same. Once the shares are shared. Bidders will combine all their received shares to one value. So, bidder one will retain his first share of 200 and receive share of 100 from second bidder and received share of 150 from third bidder, and lastly receive share of 50 from last bidder. After combination, he will have collective share of 500. This way no bidder can truly guess what other bidder is bidding as they only get part of their bid. After all this the bids are combined by trusted authority that would do knapsack computation. Collective share of player 1 which is 500 would be added with all other shares 635, 780, 712 and would be added to each other and would equal to 640 after modular reduction.

bidder 1 splits	905	=	200 + 200 + 200 + 305
bidder 2 splits	715	=	100 + 115 + 400 + 100
bidder 3 splits	800	=	150 + 250 + 100 + 300
bidder 4 splits	207	=	50 + 70 + 80 + 7

bidder 1 adds	500	=	200 + 100 + 150 + 50
bidder 2 adds	635	=	200 + 115 + 250 + 70
bidder 3 adds	780	=	200 + 400 + 100 + 80
bidder 4 adds	712	=	305 + 100 + 300 + 7

$$q8 = 500 + 635 + 780 + 712 = 2627 \bmod 1987 = 640$$

At that moment, the knapsack optimization computation would begin that would look if the 640 is bigger than 1000. If no then 0 would be selected. Then 640 would be compared to 500 to check if it is bigger and since it is 1 would be selected. At that moment, we would deduct 500 from 640 leaving 140 for next comparison. Then we would check if 140 is bigger than 250 and 0 would be selected. Afterwards we would check if 140 is bigger than 120 and 1 would be selected and after subtraction next comparison would be only 20. The details of this operation are shown in illustration below.

is (640 >= 1000)	>>	NO	>> 0	>> 640
is (640 >= 500)	>>	YES	>> 1	>> 640 - 500 = 140
is (140 >= 250)	>>	NO	>> 0	>> 140
is (140 >= 120)	>>	YES	>> 1	>> 140 - 120 = 20
is (20 >= 60)	>>	NO	>> 0	>> 20
is (20 >= 30)	>>	NO	>> 0	>> 20
is (20 >= 15)	>>	YES	>> 1	>> 20 - 15 = 5
is (5 >= 9)	>>	NO	>> 0	>> 5 - 5 = 0
is (5 >= 5)	>>	YES	>> 1	>> 5 - 5 = 0

final set of indexes = {1, 0, 1, 0, 0, 1, 0, 1, 0}
prices = {10, null, 200, null, null, 350, null, 450, null}

The resulting array of (1,0,1,0,0,1,0,1,0) shows all the bids prices that were made (10,200, 350, 450) and it is easy to verify that the highest bid is 450. This method of conduction auction can work easily with first price sealed bid auction and with second price sealed bid auction as only the winning bid would be shared by the trusted party conducting the knapsack optimization [17, 18]. In this protocol, six properties have been maintained. The process would correctly determine the winner while maintaining privacy of all losing bidders and the losing bids; therefore, upholding confidentiality and privacy properties. The bidders would submit only one bid maintaining unchangeability property and the bid would be fairly verified. The bid can be also verified since all shares have been held by the bidders and they can interchange them for verification. This process however is not as secure for malicious participants, because it relies on all shares from all bidders to be not corrupted in order to provide correct calculations unlike threshold secret sharing protocol that can derive at secret value with number of shares that is lower than number of bidders as the threshold depends on the polynomial function rather than number of participants.

IV. Conclusion

In this paper, I demonstrated the use of the secret sharing schemes in sealed-bid auctions. The threshold schemes introduced initially by Shamir and followed by variations seem to be a great way to ensure privacy and security of the bid. In many auctions, we see combinations of secret sharing protocols like when evaluating secure multiparty computation scheme the threshold secret sharing protocol like verifiable secret sharing was used. On the other hand, variable secret sharing protocol or Shamir protocol could implement all phases of the auction without need to use any other protocols. With verifiable secret sharing the security was enhanced to be able to deal with malicious behavior when Shamir protocol lacked some of the enhanced features. Great feature of threshold secret sharing protocols is that you only need enough shares to be greater than threshold of the polynomial encryption function in order to compute correct outcome, allowing room for malicious servers to exist and be eliminated in the process. Knapsack optimization solution also lacked the enhanced security and was less feasible to implement as it required all shares from all bidders to be submitted and be uncorrupted in order to arrive at correct outcome. Homomorphic process that uses dynamic programming also needs all the shares to come to the correct conclusion in order to find the best optimal path.

References

[1] Harkavy, M., Tygar, J., Kikuchi, H. (1998) Electronic Auctions with Private Bids.

[2] Nojoumian, M., Krishnamachari, S., Akkaya, K. (2015) Implementation and Analysis of Dutch-style Sealed-bid Auctions : Computational vs Unconditional Security.

[3] Cao, G. (2014). Secure and efficient electronic auction scheme with strong anonymity. Journal of Networks, 9(8), 2189. doi:10.4304/jnw.9.8.2189-2194

[4] Zhu, Y., Liu, L., & Chen, X. (2015). Efficient first-price sealed-bid auction protocols from modified comparable encryption. Paper presented at the 417-421. doi:10.1109/BWCCA.2015.36

[5] Boyd, C., Dawson, E., & Peng, K. (2005). Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing.

[6] Franklin, M. K., & Reiter, M. K. (1996). The design and implementation of a secure auction service. IEEE Transactions on Software Engineering, 22(5), 302-312. doi:10.1109/32.502223

[7] Brandt, F. (2002). A Verifiable, Bidder-Resolved Auction Protocol. In Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies, pages 18–25.

[8] Nojoumian M., Stinson D.R. (2014) Efficient Sealed-Bid Auction Protocols Using Verifiable Secret Sharing. In: Huang X., Zhou J. (eds) Information Security Practice and Experience. ISPEC 2014. Lecture Notes in Computer Science, vol 8434. Springer, Cham

[9] Larson, M., Hu, C., Li, R., Li, W., & Cheng, X. (2015). Secure auctions without an auctioneer via verifiable secret sharing.

[10] Peng K., Boyd C., Dawson E. (2005) Optimization of Electronic First-Bid Sealed-Bid Auction Based on Homomorphic Secret Sharing. In: Dawson E., Vaudenay S. (eds) Progress in Cryptology – Mycrypt 2005. Mycrypt 2005. Lecture Notes in Computer Science, vol 3715. Springer, Berlin, Heidelberg
[11] Suzuki K., Yokoo M. (2003) Secure Generalized Vickrey Auction Using Homomorphic Encryption. In: Wright R.N. (eds) Financial Cryptography. FC 2003. Lecture Notes in Computer Science, vol 2742. Springer, Berlin, Heidelberg
[12] Boyd, C., Dawson, E., & Peng, K. (2005). Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing.
[13] Yokoo, M., & Suzuki, K. (2002). Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. Paper presented at the 112-119. doi:10.1145/544741.544770
[14] Montenegro, J. A., Fischer, M. J., Lopez, J., & Peralta, R. (2013). Secure sealed-bid online auctions using discreet cryptographic proofs. Mathematical and Computer Modelling, 57(11-12), 2583. doi:10.1016/j.mcm.2011.07.027
[15] Zhang, B. (2011). Generic constant-round oblivious sorting algorithm for MPC. In Provable Security, volume 6980 of Lecture Notes in Computer Science, pages 240–256.
[16] Montenegro, J., Lopez, J. (2014). A practical solution for sealed bid and multi-currency auctions.
[17] Chodisetti, N. (2014). Analysis of digital knapsack based sealed bid auction.
[18] Chodisetti, N. (2012). Sealed Bid Auction Using Digital Knapsacks.
[19] Peng, K., & Dawson, E. P. (2007). Efficient bid validity check in ElGamal-based sealed-bid E-auction.
[20] Bogetoft, P. (2006). A practical implementation of secure auctions based on multiparty integer computation. In International Conference on Financial Cryptography and Data Security.