Implementation and Analysis of Dutch-style Sealed-bid Auctions
Computational vs Unconditional Security

Sriram Krishnamachari[1], Mehrdad Nojoumian[1] and Kemal Akkaya[2]
[1]*Department of Computer Science, Southern Illinois University Carbondale, Illinois, U.S.A.*
[2]*Department of Electrical and Computer Engineering, Florida International University, Florida, U.S.A.*

Keywords: Privacy-preserving Auction, Sealed-bid Auction, Unconditional Security, Computational Security.

Abstract: Designing a sealed-bid auction protocol is a challenging problem in the field of applied cryptography. In the last couple of decades, numerous protocols have been proposed in the literature where each one has its own property in terms of the security model, communication and computation complexities. To the best of our knowledge, there has been no study to implement and compare a similar class of sealed-bid auction protocols. This paper therefore implements and evaluates five different Dutch-style sealed-bid auction protocols, of which three protocols are computationally secure and two protocols are unconditionally secure. It mainly focuses on the computational cost of the initialization and verification phases of these privacy-preserving protocols.

1 INTRODUCTION

Due to privacy concerns, utilizing sealed-bid auction protocols has been in the center of attention by software engineers, however, the main issue is the efficiency of these protocols in a real-world setting. Generally speaking, an *auction* is a mechanism in which a group of bidders (buyers) compete to buy a product. Then, the auctioneer (seller) sells the item to the bidder who has proposed the highest bid. There exist different types of auction mechanisms in the literature. We briefly review some of them.

An *English auction* is the most common auction. In this type of auction, the bidders continuously bid for a higher price and the auction ends once a single offer is proposed as the final highest price. On the other hand, in a *Dutch-style auction*, the auctioneer continuously reduces the price of the item until a bidder agrees to buy the item for that specific price. Two other popular mechanisms are *first-price* and *second-price* auctions. In the former case, the bidder who proposes the highest price wins and pays the amount that he has offered. In the latter case, the bidder who proposes the highest price wins, however, he pays the amount of the second-highest price.

In a *sealed-bid auction* of any type, the bidders seal their bids by using closed envelopes, that is, a cryptographic technique, and then they submit these envelopes to the auctioneer. At the end of the auction, only the auction outcomes (the winner and the selling price) are revealed and all the losing bids are kept private. The main motivation for constructing sealed-bid auction protocol is to protect the losing bids since they can be used by the auctioneer (seller) to maximize his revenue in the future auctions for similar items.

From a computational perspective, there are two types of schemes: *computationally secure* protocols and *unconditionally secure* protocols. In the former case, the adversary has limited computational capability, i.e., he cannot solve well-known mathematical problems such as factoring two large integers or discrete logarithm. In the latter case, the adversary has unlimited computational power. Furthermore, the adversary might be *passive* or *active*. In the former case, the participants follow the protocol, however, they may attempt to learn the secret, e.g., the losing bids. In the latter case, the participants not only try to learn the secret but also deviate from the protocol.

1.1 Motivation and Contribution

As we stated, various sealed-bid auction protocols have been constructed under different assumptions and with various properties. However, there exists no study to demonstrate the implementation complexity and performance analysis of a similar class of secure auction protocols. Our paper therefore aims at evaluation of five different Dutch-style sealed-bid auction protocols. Note that this type of auction was selected for our experiment since it is widely used by sellers.

Krishnamachari S., Nojoumian M. and Akkaya K..
Implementation and Analysis of Dutch-style Sealed-bid Auctions - Computational vs Unconditional Security.
DOI: 10.5220/0005258501060113
In *Proceedings of the 1st International Conference on Information Systems Security and Privacy* (ICISSP-2015), pages 106-113
ISBN: 978-989-758-081-9

Our major goal is to compare the selected protocols in terms of the computational complexity, i.e., initialization and verification times. We demonstrate how these measurement parameters vary once the assumption, security or adversarial model change. To perform a reasonable and fair comparison, the schemes are chosen from the same class of secure auction protocols. Furthermore, to have comparison across different settings, three computationally secure protocols (Sakurai and Miyazaki, 1999; Sako, 2000; Suzuki et al., 2000) and two unconditionally secure constructions (Nojoumian and Stinson, 2010; Nojoumian, 2012) are selected. Among these candidates, (Sakurai and Miyazaki, 1999) uses an undeniable signature scheme, (Sako, 2000) applies a public key encryption scheme, (Suzuki et al., 2000) employs a collision intractable hash function, and (Nojoumian and Stinson, 2010) utilizes a multicomponent commitment scheme. Note that if multiple protocols have been proposed in a paper, the most efficient method has been chosen for our evaluation.

Each candidate is evaluated with various price ranges, modulus sizes, and number of bidders and auctioneers. The evaluation shows that the initialization and verification times of unconditionally secure protocols are significantly larger than that of the computationally secure schemes. The analysis demonstrates how our measurement parameters vary across protocols when we modify the price range, the modulus size, and the number of bidders and auctioneers.

2 LITERATURE REVIEW

(K.Franklin and K.Reiter, 2006) proposes the first sealed-bid auction protocol. This paper utilizes *verifiable secret sharing* and *verifiable signature sharing* schemes in a distributed setting to construct a computationally secure protocol. In the opening phase, all the bids are opened to determine the winner, i.e., the bids are only kept private while the auction is running.

(Sakurai and Miyazaki, 1999) provides a computationally secure Dutch-style sealed-bid auction protocol by using an *undeniable signature scheme*. Later, (Sako, 2000) proposes a much faster Dutch-style sealed-bid auction protocol by using a *public-key encryption scheme*. This protocol is computationally secure and it does not require the existence of the bidders during the opening phase. The next computationally secure protocol for Dutch-style sealed-bid auctions is proposed in (Suzuki et al., 2000). This construction utilizes a *hash chain scheme*.

On the other hand, (Nojoumian and Stinson, 2010) and (Nojoumian, 2012) provide three unconditionally secure Dutch-style sealed-bid auction protocols[1]. The authors utilize a *multicomponent commitment scheme* where multiple committers and verifiers are involved in the scheme. These protocols are executed only by the bidders without the existence of any auctioneers.

There exist other types of first-price secure auction protocols. Brandt (Brandt, 2006) presents a computationally secure first-price auction protocol where the bidders use *zero knowledge proofs* to verify the auction outcomes. This protocol preserves privacy as long as one bidder is honest. (Michael Harkavy, 1998) also proposes a computationally secure first-price auction protocol that requires multiple auction servers in a distributed setting. This construction is later improved in (Kun Peng and Viswanathan, 2002) for its shortcomings in terms of the verifiability. The improved version uses a commonly accessible bulletin board such that all the parties can verify the auction outcomes. (Zheng et al., 2007) provides a secure auction protocol based on *public key cryptography* and *one-way collision-free hash functions*. (Kun Peng and Dawson, 2005) constructs a protocol that prevents attacks to the existing secret sharing-based protocols by using a *homomorphic secret sharing* scheme.

For other kinds of auctions, we can refer to the following schemes: (Kikuchi, 2002) proposes a verifiable $(M+1)^{st}$*-price auction* protocol using verifiable secret sharing; (David C. Parkes and Thorpe, 2009) provides a cryptographic *combinatorial clock-proxy auction* protocol. In this scheme, the bidders bid in two phases for a bundle of items such that no one can decrypt any values until all bids are submitted; (Helger Lipmaa and Niemi, 2002) presents a protocol for a sealed-bid *Vickery auction* that is computationally secure. They use a homomorphic auction scheme which requires the seller to participate in the auction along with the auctioneer; finally, (Suzuki and Yokoo, 2002) provides a computationally secure *combinatorial auction* protocol in which multiple items with interdependent values are sold simultaneously. In this scheme, the bidders can bid on any combination of the items.

3 SEALED-BID AUCTIONS

In an open auction, the bids are not kept private. As a result, auctioneers or a group of bidders may collude or use the past losing bids to maximize the auction revenue. Consider the following example as an open auction for a land. The auctioneer sets the starting price of the land as $35,000. Assume that the

[1] For other type of unconditionally secure auction protocols, see (Nojoumian and Stinson, 2014).

actual value of the land is $\$80,000$. While the auction is running, a group of bidders may collude with the auctioneer such that they repeatedly bid a price slightly higher than the current highest price to maximize the auction revenue, e.g., they may stop bidding at $\$90,000$ to sell the land $\$10,000$ higher than its actual value. Furthermore, the auctioneer or a group of bidders may record the losing bids in order to use them in the future auctions for similar items. For instance, in the land example, suppose that the auctioneer observes that the average of the losing bids had been $\$42,000$ in the previous auction. Since the auctioneer learns the average of the evaluations, he can set the starting price to $\$42,000$ or a higher price in the future auctions for a similar land. This way the auctioneer would be able to maximize his revenue.

On the other hand, in a *sealed-bid auction*, the bidders seal their bids by a cryptographic method and then submit their evaluations to the auctioneer. During the opening phase, the winner is identified and the selling price is determined without revealing the losing bids. This process neither allows the auctioneers and/or bidders to collude nor it reveals any information about the bidding trend. Note that the sealed-bid technique can be utilized for any type of auctions. To construct a sealed-bid auction of any kind, the following essential properties must be achieved:

- *Correctness*: the auction process must provide correct outcomes, i.e., winner and selling price.
- *Privacy*: the losing bids must not be revealed to the auctioneer and the other participating bidders.
- *Verifiability*: all parties who may exchange money must be able to verify the auction outcomes.
- *Fairness*: bidders should not modify/deny the bids that they have submitted, a.k.a, *non-repudiation*.

For instance, if a second-price secure auction is not verifiable, the auctioneer may ask the winner to pay a price that is slightly higher than the second-highest bid. Since the bids are sealed, the winner cannot verify the actual value of the second-highest bid.

In a Dutch-style sealed-bid auction, the auctioneer initiates the auction with some parameters. In the bidding phase, the bidders choose their evaluations from the price set and submit their sealed-bids only once. The bidding phase ends after a predefined time so that the bidders can no longer bid or modify the submitted bids. In the opening phase, the selling price starts from the maximum price and it is decreased step by step until a bidder claims as the winner. At this stage, the winner's bid is opened for verification. Moreover, the losers must prove that their bids have been less than the winning price without opening their bids.

The Dutch-style auction keeps the losing bids secret on its own, however, its sealed-bid version provides additional property. That is, the bidders decide ahead of time and propose their valuations independent of whatever information they may gain during the auction. This property prevents any kind of collusion since the bidders cannot learn any information about the other bids while the auction is running. Next, we briefly review our selected protocols.

3.1 Undeniable Signature Scheme

(Sakurai and Miyazaki, 1999) proposes the first Dutch-style sealed-bid auction protocol. They use a public bulletin board to implement the auction. All the auction results are published on the bulletin board so that each bidder can verify every step of the auction. This protocol utilizes an *undeniable signature scheme*, proposed by (Michels and Stadler, 1997). In an undeniable signature scheme, a prover has to convince the verifier for the equality or inequality of two discrete logarithms. In fact, the undeniable signature scheme is used to comply with the required auction properties, as we discussed earlier. This protocol also employs a registration authority to certify the bidder's public key. Note that in this construction p and q must be two large prime numbers.

3.2 Public-key Encryption Scheme

(Sako, 2000) proposes a Dutch-style sealed-bid auction protocol based on ElGamal public key cryptosystem (ElGamal, 1985). In this protocol, each bidder is required to post his bid in the form of an encrypted message and then the auctioneer processes these messages to define the auction outcomes. The protocol utilizes a probabilistic encryption of a bid in such a way that the bid is not decrypted unless it is the winning bid. The proposed scheme can be run either by a single or multiple auctioneers. For the sake of simplicity, we implemented the single auctioneer model. The scheme considers a set of L possible bid values, i.e., $V = \{v_1, \cdots, v_L\}$. It also employs a set of encryption functions $\{E_v\}$ and a set of decryption functions $\{D_v\}$ for $v \in V$. The cipher text is an encryption $E_v(M_v)$ where M_v is a predefined message.

3.3 Hash Chain Scheme

The protocol proposed in (Sako, 2000) overcomes the computational complexity problem of (Sakurai and Miyazaki, 1999), however, it has a shortcoming. In its multiple-auctioneer model, a malicious auctioneer can reveal the losing bids. (Suzuki et al., 2000) provides a Dutch-style sealed-bid auction protocol that overcomes this issue. In this protocol, a distributed decryption method is used in a multiple-auctioneer setting. The protocol employs hash functions to create hash chains for encryption and decryption.

3.4 Multicomponent Commitment

(Nojoumian and Stinson, 2010; Nojoumian, 2012) proposes a couple of unconditionally secure Dutch-style sealed-bid auction protocols. These constructions do not require any auctioneer to participate in the bidding and opening phases. A trusted initializer first initiates the auction and then he leaves the scheme. During the initialization phase, each bidder $B_1, \ldots, B_n$ receives some information from the initializer. In the opening phase, bidders determine the auction outcomes on their own. In this setting, η and κ denote the minimum and maximum prices respectively, i.e., $\theta = \kappa - \eta + 1$ denotes the number of prices. Bidders are connected through point-to-point secure channels. During the bidding phase, each bidder B_i commits to his bid $\beta_i \in [\eta, \kappa]$. In the opening phase, the winner reveals his bid and the losers prove that their bids had been less than the wining price. All the computations are performed in Z_q. The prime q must be large enough such that n^2/q be very small.

4 EVALUATION AND ANALYSIS

The implementation of the selected Dutch-style sealed-bid auction protocols was performed using *Microsoft Visual Studio* and also the *Crypto++ library*. For the sake of simplicity, we used a common graphical user interface for auctioneers and bidders. The bulletin board in (Sakurai and Miyazaki, 1999) was implemented as a table with bidders' identifications in rows and auction parameters in columns.

We selected all the cryptographic modules from *Crypto++* except for the construction of the polynomials in protocols (Nojoumian and Stinson, 2010; Nojoumian, 2012); for each polynomial $g(x) = a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$ of degree $n-1$, we simply generated n random numbers $a_0, a_1, \cdots, a_{n-1}$ in the finite field Z_q and then we submitted the array to the related bidder. Moreover, the Diffie-Hellman library in *Crypto++* was used to generate large prime numbers for the protocols in (Sakurai and Miyazaki, 1999; Sako, 2000). This library also provides generator α of subgroup Z_q (of order q) that is used in (Nojoumian and Stinson, 2010; Nojoumian, 2012).

Note that initialization and verification times not only consider the *computational* cost but also capture the *communication* overhead of the protocols. To evaluate the auction protocols for various rounds, the bidders were allowed to bid from the least element in the price set to 100%, 75%, 50% and 25% of the price set. For instance, in a price set of 100 elements, a bidder could bid from P_1 to P_{100}, P_{75}, P_{50} or P_{25} respectively. We will call this as *price range* hereafter.

We faced some challenges when we implemented (Nojoumian and Stinson, 2010; Nojoumian, 2012). For instance, the point-to-point channels among bidders required $n-1$ ports for every single bidder, i.e., $n(n-1)$ ports for n bidders on a single computer. Furthermore, since the bidders conducted the auction on their own, the opening phase had to be synchronized. Therefore, a notify-and-receive technique was implemented to resolve this issue (i.e., each bidder announces his completion of one round and then he waits until all the bidders accomplish this phase). The same notify-and-receive method was adopted for the opening phase of (Suzuki et al., 2000).

Note that the notify-and-receive technique led to an increased verification time. However, even with this extra delay, we could relatively estimate the verification time. This means that evaluating the test cases in an auction system with *ideal synchronized channels* would certainly give more accurate verification time for these protocols but the real implementation of synchronized channels had been a big challenge and still under debate among software engineers.

Generally speaking, the model of synchronous channels might be (a) the sender sends a message and all the receivers are guaranteed to get the message within a period p, where the length of p is a constant known to everyone from the start of the protocol, or even a stronger model in which (b) the sender sends a message and all the receivers get the message at exactly time t so that each receiver cannot rush and change its behavior in the sending slot after seeing the incoming messages. Neither is, of course, a very realistic channel, which is why it is better to make protocols that work in asynchronous network models.

Our evaluations were conducted on a computer with an Intel i7-2600 CPU @ 3.4GHz processor and 16GB RAM. The protocols were tested with common price set P consisting of 41 elements. They were also evaluated with three combinations of bidders (i.e., 25, 50 and 75) and various modulus sizes (i.e., 128, 256, 512 and 1024). However, unconditionally secure protocols were only tested with 25 and 50 bidders since they were computationally intensive. Finally, the protocol in (Suzuki et al., 2000) was verified with different number of auctioneers (i.e., 5, 10, 15 and 20).

4.1 Computational Protocols

In this section, evaluation and analysis of the computationally secure protocols (Sakurai and Miyazaki, 1999; Sako, 2000; Suzuki et al., 2000) are shown. As we mentioned earlier, the initialization and verification times are our measurement metrics.

The result shows an amplification in initialization time when the modulus size is increased in (Sakurai and Miyazaki, 1999). The initialization time is significantly amplified when we switch to 1024 bits. This was expected as generating large primes for larger modulus is more time-consuming. We executed the initialization procedure a thousand times for each modulus size and then we calculated the average of the timing for each modulus size, shown in Figure 1. Note that the number of bidders as well as the price range have no impact on the initialization time.

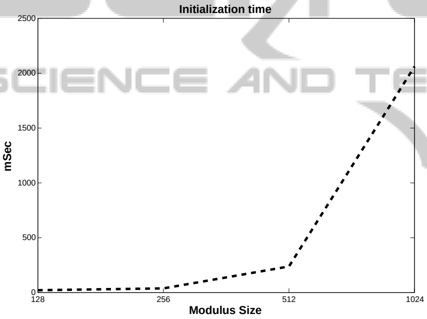

Figure 1: Initialization Time: (Sakurai and Miyazaki, 1999) .

As shown in Figure 2, the verification time is measured for two modulus sizes,128 bits (top) and 512 bits (bottom). This time varies when the price range, modulus size and number of bidders are changed. It is minimized (almost 203 *ms*) when these parameters are 100%, 128 bits and 25 respectively. On the other hand, it is maximized (63 *secs*) when these parameters are 25%, 1024 bits and 75. The longer verification time is mainly attributed to the presence of bidders in the opening phase as the auctioneer and bidders exchange many messages to define the outcomes.

While the evaluation result of (Sakurai and Miyazaki, 1999) shows a significant change in the verification time for various price ranges, modulus sizes and number of bidders, (Sako, 2000) does not exhibit a significant variation in the verification time for similar changes in the auction parameters. This variation can be justified by the presence of bidders during the opening phase. In (Sako, 2000), the computation load is fully on the auctioneer and the bidders do not participate in the opening phase whereas in (Sakurai and Miyazaki, 1999) the bidders participate in the opening phase with multiple rounds of message handshake. The initialization time is very much similar to that of (Sakurai and Miyazaki, 1999) as the parameters are the same, shown in Figure 3.

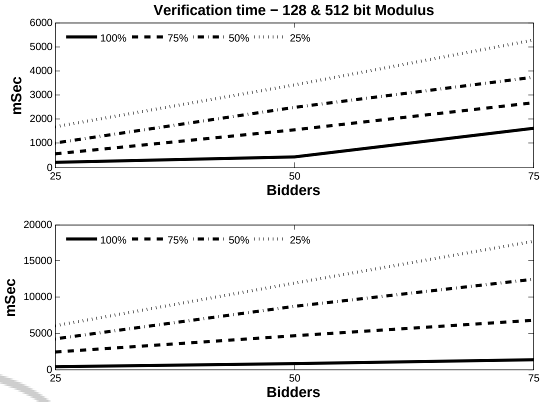

Figure 2: Verification Time: (Sakurai and Miyazaki, 1999).

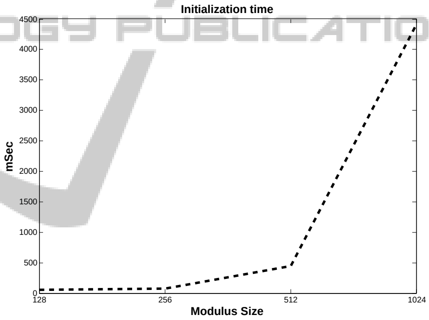

Figure 3: Initialization Time: (Sako, 2000).

The verification time is plotted in Figure 4: 128 bits (top) and 512 bits (bottom). It is minimized (almost 2 *ms*) when the parameters are 100%, 128 bits and 25 respectively and it is maximized (3 *secs*) when these parameters are 25%, 1024 bits and 75.

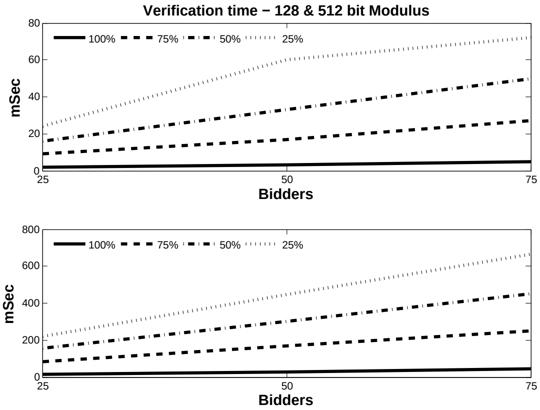

Figure 4: Verification Time: (Sako, 2000).

The proposed protocol in (Suzuki et al., 2000) employs multiple auctioneers to define the outcomes. As a result, the number of auctioneers has an impact on the verification time, however, the initialization time is not affected by this number. Furthermore, since the generation of the encryption and decryption functions as well as predefined messages directly correlate to the price set, the initialization time varies with the size of this set in addition to the modulus size.

The measurement of the initialization time for two price sets of 41 and 200 elements illustrates only a difference of 30 *ms*. This means that the variation in the number of elements in the price set does not drastically affect the initialization time. For this reason, we plotted the initialization time in the same way as of the previous protocols, as shown in Figure 5.

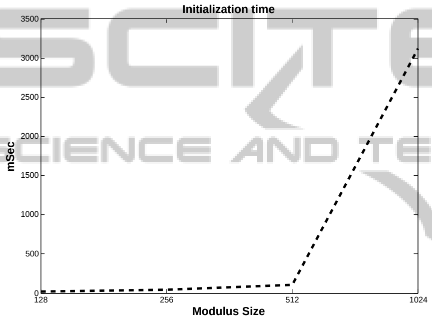

Figure 5: Initialization Time: (Suzuki et al., 2000)

The verification time varies with the number of auctioneers as well as the price range, modulus size and the number of bidders; similar to the other two protocols. For the sake of simplicity, we fixed the modulus size to 128 bits and changed the other parameters, as shown in Figure 6: 5 auctioneers (top) and 20 auctioneers (bottom). The verification time is minimized (almost 80 *ms*) when we consider 100% of the price range, 5 auctioneers and 25 bidders. On the other hand, it is maximized (7 *secs*) when these parameters are 25%, 20 and 75 respectively.

4.2 Unconditional Protocols

In this section, evaluation and analysis of the unconditionally secure protocols (Nojoumian and Stinson, 2010; Nojoumian, 2012) are demonstrated. In these constructions, the initialization cost is measured based on the interactions between the initializer and the bidders. Furthermore, since these protocols do not require the existence of an auctioneer during the opening phase, the verification cost is measured based on the interactions among the bidders. As we stated earlier, part of the complexity cost is due to the implementation of the notify-and-receive technique.

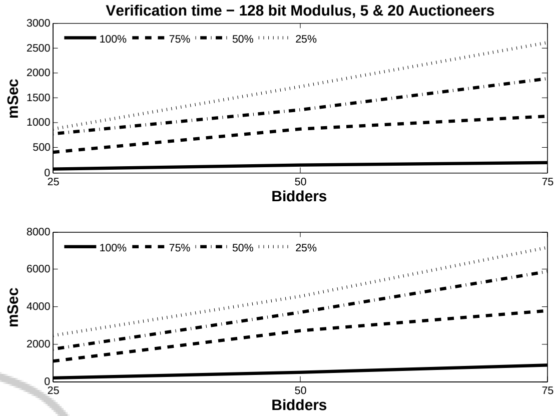

Figure 6: Verification Time: (Suzuki et al., 2000).

In the first protocol, named VNR, the initialization time is directly proportional to the size of the price set, modulus size and the number of bidders. To be consistent with the previous diagrams, we fixed the price set to 41 elements and the number of bidders to 25. Figure 7 compares the initialization time for various modulus sizes. As shown, with the fixed parameters, this time varies from 6 to 36 *secs* for various modulus sizes. Similarly, for each modulus size, we executed the initialization procedure for a thousand times and then calculated the average of the timing. In our experiments with 1024 bits modulus, 100 bidders and a price set of 100 elements, the initialization time was in the order of a few minutes.

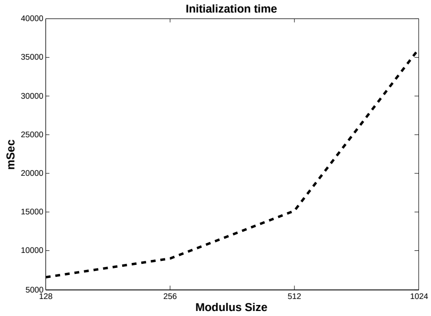

Figure 7: Initialization Time: VNR.

Although the verification time varies with the price range, modulus size and the number of bidders, we restricted our test cases to 128 bit modulus and 25 and 50 bidders due to a high computational cost of these protocols, as shown in Figure 8. In this setting, the verification time is minimized to 6 *secs* with 100% price range, 128 bits modulus and 25 bidders. On the other hand, it is maximized to 3 *mins* when these parameters are 25%, 128 and 50 respectively.

In the second protocol, named EVNR, the initialization time is also proportional to the size of the price

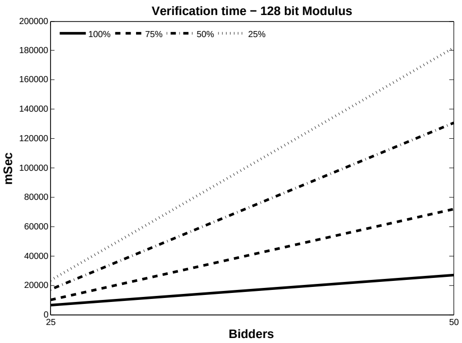

Figure 8: Verification Time: VNR.

set, modulus size and the number of bidders, however, the number of commitments is optimized by a logarithmic factor, i.e., $\lambda = \lceil \log_2 \theta \rceil$ where λ denotes the required number of commitments and θ denotes the number of prices. As such, the number of polynomials (to be generated for each bidder) is reduced drastically. This improves the initialization time as well as the verification time. With a similar approach as of the previous protocol, Figure 9 compares the initialization time for various modulus sizes with 41 prices and 25 bidders. As shown, this time varies from 1 to 6 *secs* for various modulus sizes, which is a significant improvement compared to the VNR protocol.

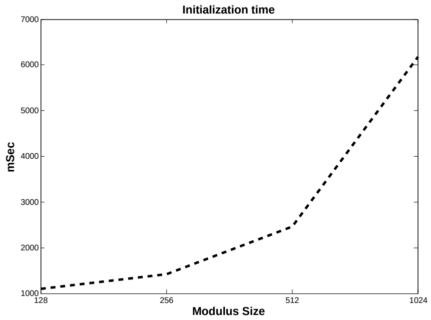

Figure 9: Initialization Time: EVNR.

Similarly, we restricted our test cases to 128 bit modulus and 25 and 50 bidders due to a high computational cost, shown in Figure 10. In this setting, the verification time is minimized to 3 *secs* with 100% price range, 128 bits modulus and 25 bidders. On the other hand, it is maximized to 48 *secs* when these parameters are 25%, 128 and 50 respectively. This is a major improvement compared to the VNR protocol.

4.3 Computational vs Unconditional

Figures 11 and 12 show the initialization and verification times of all protocols. To make the result visible on a single plot, we use a logarithmic scale of *ms*.

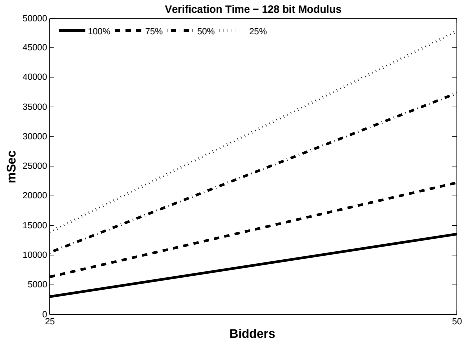

Figure 10: Verification Time: EVNR.

Among the computationally secure protocols, the initialization time of PES is higher than the other two protocols, however, its verification time is much lower. Within the unconditionally secure protocols, EVNR is executed much faster. Overall, the unconditionally secure protocols take more time in both phases, however, they can be run without any auctioneers and they provide a higher level of security.

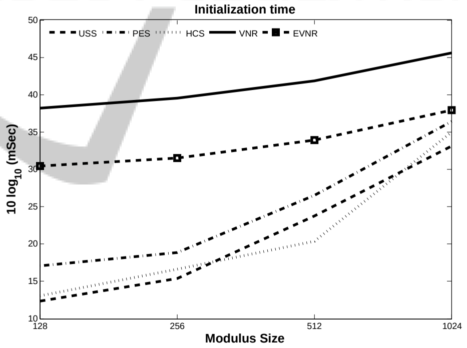

Figure 11: Initialization Times of the Five Protocols.

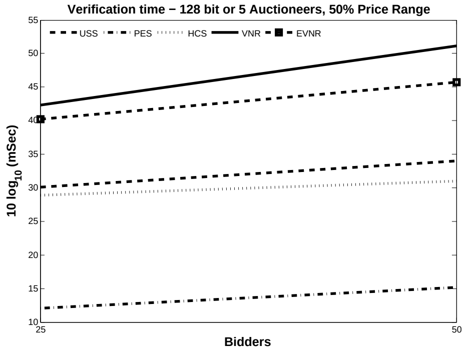

Figure 12: Verification Times of the Five Protocols.

Table 1 summarizes a sample of our test cases. It also demonstrates the initialization time for 1024 bits modulus and the verification time for 50% price range, 128 bits modulus and 50 bidders.

Table 1: Summary of the Test Cases (times are presented in milliseconds).

Scheme	Security	Initializer	Auctioneers	Bidders	Prices	Initialization	Verification
USS	computational	1	1	25, 50, 75	41	2066	2489
PES	computational	-	1	25, 50, 75	41	4416	33
HCS	computational	-	5, 20	25, 50, 75	41	3124	1260
VNR	unconditional	1	-	25, 50	41	36034	130734
EVNR	unconditional	1	-	25, 50	41	6174	37411

5 CONCLUSION

Our motivation to analyze the complexity of sealed-bid auction protocols led us to the implementation of various computationally and unconditionally secure constructions. As we stated earlier, the unconditionally secure protocols are more expensive, however, they provide a higher level of security. On the other hand, the computationally secure protocols are faster but they rely on computational assumptions. Therefore, selecting an appropriate sealed-bid auction protocol is a trade off between the level of security and the time complexity.

It is also worth mentioning that we selected the Dutch-style auction for our experiments since it is widely used in real-world settings. As our future work, we intend to perform a similar analysis for second-price and combinatorial auctions.

REFERENCES

Brandt, F. (2006). How to obtain full privacy in auctions. In *International Journal of Information Security*, pages 201–216. Springer.

David C. Parkes, M. O. R. and Thorpe, C. (2009). Cryptographic combinatorial clock-proxy auctions. In *13th International Conference on Financial Cryptography FC*, volume 5628 of *LNCS*, pages 305–324. Springer.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472.

Helger Lipmaa, N. A. and Niemi, V. (2002). Secure vickrey auctions without threshold trust. In *6th International Conference on Financial Cryptography FC*, volume 2357 of *LNCS*, pages 87–101. Springer.

K.Franklin, M. and K.Reiter, M. (2006). The design and implementation of a secure auction server. *IEEE Transactions on Software Engineering*, 22(5):302–312.

Kikuchi, H. (2002). (m+1)st-price auction protocol. In *6th International Conference on Financial Cryptography FC*, pages 351–363. Springer.

Kun Peng, C. B. and Dawson, E. (2005). Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing. In *1st International Conference on Cryptology in Malaysia*, volume 3715 of *LNCS*, pages 84–98. Springer.

Kun Peng, Colin Boyd, E. D. and Viswanathan, K. (2002). Robust, privacy protecting and publicly verifiable sealed-bid auction. In *4th Int. Conf. on Information and Communications Security ICICS*, volume 2513 of *LNCS*, pages 147–159. Springer.

Michael Harkavy, J. D. Tygar, H. K. (1998). Electronic auctions with private bids. In *3rd Workshop on Electronic Commerce*, pages 61–74. Springer.

Michels, M. and Stadler, M. (1997). Effcient convertible undeniable signature. In *4th Int. Workshop on Selected Areas in Cryptography SAC*, pages 231–244.

Nojoumian, M. (2012). *Novel Secret Sharing and Commitment Schemes for Cryptographic Applications*. PhD thesis, Department of Computer Science, University of Waterloo, Canada.

Nojoumian, M. and Stinson, D. R. (2010). Unconditionally secure first-price auction protocols using a multicomponent commitment scheme. In *12th Int. Conf. on Information and Communications Security ICICS*, volume 6476 of *LNCS*, pages 266–280. Springer.

Nojoumian, M. and Stinson, D. R. (2014). Efficient sealed-bid auction protocols using verifiable secret sharing. In *10th International Conference on Information Security Practice and Experience, ISPEC'14*, volume 8434 of *LNCS*, pages 302–317. Springer.

Sako, K. (2000). An auction protocol which hides bids of losers. In *3rd Int Workshop on Practice and Theory in Public Key Cryptography PKC*, volume 1751 of *LNCS*, pages 422–432. Springer.

Sakurai, K. and Miyazaki, S. (1999). A bulletin-board based digital auction scheme with bidding down strategy. In *Int Workshop on Cryptographic Techniques and E-commerce CrypTEC*, pages 180–187.

Suzuki, K., Kobayashi, K., and Morita, H. (2000). Efficient sealed-bid auction using hash chain. In *3rd Annual Int Conference on Information Security and Cryptology ICISC*, volume 2015 of *LNCS*, pages 183–191. Springer.

Suzuki, K. and Yokoo, M. (2002). Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In *6th Int. Conf. on Financial Cryptography FC*, volume 2357 of *LNCS*, pages 44–56. Springer.

Zheng, S., McAven, L., and Mu, Y. (2007). First price sealed bid auction without auctioneers. In *International Conference on Wireless communications and mobile computing*, pages 127–131. ACM.