J. A. Montenegro, and J. Lopez, "A practical solution for sealed bid and multi-currency auctions", *Computers & Security*, vol. 45, pp. 186-198, 2014.
http://doi.org/10.1016/j.cose.2014.06.004
NICS Lab. Publications: https://www.nics.uma.es/publications

A practical solution for sealed bid and multi-currency auctions

Jose A. Montenegro, Javier Lopez*

Department of Computer Science. University of Malaga. Campus Teatinos, s/n, 29071, Malaga. Spain

Abstract

This paper introduces a sealed bid and multi-currency auction using secure multiparty computation (SMC). Two boolean functions, a comparison and multiplication function, have been designed as required to apply SMC. These functions are applied without revealing any information, not even to trusted third parties such as the auctioneer. A type of Zero Knowledge proof, discreet proof, has been implemented with three variants, interactive, regular and reduced non interactive proofs. These proofs make it possible to verify the correctness of the functions whilst preserving the privacy of the bid values. Moreover, a system performance evaluation of the proposal has been realized on heterogeneous platforms, including a mobile platform. The evaluation concludes that our proposal is practical even on mobile platforms.

Keywords:
Zero Knowledge Protocols, Secure Multiparty computation, Sealed auctions, Multi-currency auctions.

1. Introduction

Electronic auctions have been an active research field for the last two decades. The interest is not only motivated by their economic influence, but also their security requirements. The work in Franklin and Reiter (1995) lists several examples of improper behavior that could arise from an inappropriate implementation of these requirements. Electronic auctions have been a suitable field of study for applying and deploying cryptography primitives and protocols. Furthermore, their importance has been reinforced by the growing importance of electronic commerce over the same period of time.

An electronic auction is commonly divided into two phases, bidding and winning-selling price determination. During the bidding phase the bidder sends her bid to the auctioneer. In sealed auctions, the system has to provide the appropriate security mechanism to safeguard the privacy of the values. In selecting security techniques it has to be taken into consideration that the bids have to be compared to determinate the winner and the price in the next auction phase. Most of the secure techniques applied to electronic auctions require decrypting the bid to determine the winner and the sell price. Moreover, it would be desirable to keep the privacy of the bids even after the auction has finished as if it is released and the bidder loses the auction, other bidders can use this information to their advantage in future auctions, as pointed out by Nakanishi et al. (2004) and Harkavy et al. (1998).

The aim of this paper is to design a practical solution for a sealed auction using secure multiparty computation (SMC). The adequate application of SMC techniques require the appropriate function definition. Our proposal includes the definition of two functions, the comparison and the multiplicative functions. The comparative function determines the winner and the price of the auction, whereas the definition of a multiplicative function enables bidders to bid on an auction with a set of determinate currencies. The inclusion of Zero knowledge proof techniques makes it possible to safeguard the bid of the losing bidders even after

[*]Corresponding author: Phone: +34 9521-31327
Email addresses: monte@lcc.uma.es (Jose A. Montenegro), jlm@lcc.uma.es (Javier Lopez)

the auction has finished. Our contribution is twofold, on the one hand we propose a practical application of SMC techniques and on the other hand we propose a secure multi-currency and verifiable sealed auction.

The rest of the paper is structured as follows: Section 2 briefly reviews the security requirements of electronic auctions as well as other proposals related to secure auctions; Section 3 outlines the system overview; Section 4 includes the mathematical foundation of the proposal; Section 5 introduces secure multiparty computation and the definition of the comparison and multiplication functions used to perform the multi-currency auction and to determine the winner and sale price; Section 6 explains the concept of the discreet proof, the cornerstone of our proposal, and the three variants that have been developed; Section 7 analyzes the performance of the solution in mobile and non mobile platforms and Section 8 offers some conclusions.

2. Literature review and discussion

Over the last two decades, a large number of papers have been written about security primitives and protocols applied to sealed auctions. The paper by Franklin and Reiter (1995) was the pioneer work on secure auctions using cryptographic techniques.

The security requirements of electronic auctions are well-known and can be found in the plentiful specialized literature. The work in Peng et al. (2003) establishes basic security properties in order, from the framework to a discussion on different auction models. *Basic properties* are those properties that the majority of studies have agreed on: correctness, confidentiality and fairness.

In general terms, the security mechanisms added to the auction system increase the overall computation time, they are orthogonal to the user's usability. The key question is how to meet the security requirements without a decrease in the efficiency of the system. In the specialized literature, several articles Harkavy et al. (1998); Lee et al. (2009); Zhang et al. (2000); Peng (2011) cover the trade-off between security and performance. Detailed information on the performance of our system can be found in Section 7.

The work in Boyd and Mao (2000) points out the importance of *minimization of trust* in one party, particularly the auctioneer. It is possible for the auctioneer to play the role of a bidder or collude with a bidder so as to help that bidder win the auction. In our approach, we do not have to trust the auctioneer because no information about the bids is disclosed, even at the end of the auction.

Non-repudiation is also considered a desirable property in Harkavy et al. (1998) and Zhang et al. (2000). This property can be achieved if the submitted bid is directly associated with the identity of the bidder or indirectly linked to some identifying information for the bidder such as a token. Our proposal is based on the application of asymmetric cryptography and bit commitment protocols; therefore, the objective of non-repudiation can be achieved. The inclusion of the security objective of anonymity in our proposal requires only a slight modification to our design.

The security objective of *verifiability* is discussed in Lee et al. (2009); Omote and Miyaji (2002); Peng et al. (2003). The authors define this as all participating parties being able to check the source and completion of a bid. Zhang et al. (2000) describes a specific aspect of verifiability named *validity of the successful bid*, which occurs when the successful bid is the highest of all the bids. This property implies that the winning bid is compared with all of the bids submitted and that the comparison of bids is made without disclosing information about the bids submitted by the losing bidders. This property is the cornerstone of our proposal.

The works in (Abe and Suzuki, 2002; Chen et al., 2004; Gao et al., 2011; Howlader et al., 2012) establish the security mechanism to prevent bid rigging. The goal of these approaches is to ensure that a coerce cannot obtain a proof of a bidder's price. In our proposal, a coerce would only know that the bidding values are less than or equal to the auction winner, but does not obtain any information about the bid value.

The secure multiparty protocol has been a common technique used to fulfill some requirements, each one in a different way. The work in Damgard et al. (2007) proposes a protocol for secure comparison of two number of 16 bits. A later approach, Damgard et al. (2009), by the same authors, corrects a security flaw in the original protocol. The proposal is applied to online auctions where the bidder automatically bids until a determinate bid. The main goal of the proposal is to preserve the privacy of the maximum bid.The proposal Nakanishi et al. (2004) deals with a common problem of SMC protocols and proposes an SMC protocol where the bidder can submit only a single ciphertext. Our proposal has no limit of bits for secure comparison of two numbers, and as we have mentioned it preserves the security of non winning bids.

Zero knowledge proof is the basis of the proposal in Palmer et al. (2010) to verify the correctness of the auction protocol without revealing the participant's bid values. The authors conclude that the proposal is not efficient with keys greater than 384 bits, which is a very small key length. Our proposal has been positively evaluated with keys length of 1024 bits (see Section 7).

In this section a brief comparative has been made between our proposal and previous proposals, which cover the most usual security requirements in an auction system. Although, the majority of the reviewed proposals do not provide any technical evaluation, an approximation of their evaluations can be performed from a theoretical description. However, some of them are based on strong assumptions that limit their practicality.

Furthermore, our work presents a flexible framework that can be easily adapted to new functionally requirements just by including a suitable Boolean function, without jeopardizing the security of the whole system. Specifically, we provide the definition of two Boolean functions as examples of the application. Whereas the comparison function is vital for the fundamental functionality of the system, the multiplication function adds an additional functionality to the action. The versatility that our proposal offers cannot be found in the reviewed literature.

3. System Overview

The bidding and the resolution phases are the minimum phases required in any auction. During the bidding phase each bidder sends her bid to the Auctioneer. Sealed auctions require that the bid must to be sent encrypted to safeguard its privacy. A wide variety of security mechanisms have been designed to accomplish the privacy of the bid. In our case, we decided to apply a bit commitment scheme to accomplish this security requirement. Among the existing bit commitment schemes, our bit commitment is based on quadratic residue assumption (QRA), from now on referred to in this text as bit commitment. The mathematical foundation of the bit commitment is included in Section 4. The selected bit commitment scheme safeguards the privacy of each bid during the bidding phase and afterward its value is used as the input of the boolean circuits. Finally, it makes the construction of the certification proofs possible. The selection of a different bit commitment scheme would not warrant the aforementioned functionality. Therefore, the selection scheme is driven by a twofold functionality, privacy mechanism and the construction of boolean circuits and proofs. Moreover, this bit commitment scheme enables an advanced authentication method, using the bidder's cryptographic keys (see Definition 1). Examples of authentication protocols based on the QRA can be found in the following publications Chen (1998) and Shparlinski et al. (2000).

Once a bid has been submitted, each bidder builds a comparison circuit or multiplication and comparison circuit, following the instructions detailed in Section 5.3. Both boolean circuits will be created, in the case a currency exchange is required. In this case, the bit commitment of the bid will be the input of the multiplication circuit. Afterward, the output of the multiplication circuit will finally be the input of the comparison function. On the other hand, if the currency exchange is not required, the bit commitment will directly be the input of the comparison function.

After the finalization of the bidding phase, the bidders cannot submit any new bid to the system. At this point, the resolution phase is performed in two steps. Initially, by executing a polling method, the winner and sell price is determined. During this phase only the highest bid is opened to verify the correctness of the resolution process. Since the remaining bids remain unopened, it is necessary for those bids to be verified as indeed being lower than the highest bid. This is accomplished through all participants executing the interactive proofs or publishing appropriate proof certificates. The definition and contents of the proof certificates, as well as their construction and verification, are discussed in detail in Section 6. Basically, the bidders publish just enough data to reconstruct both boolean circuits and perform the verification of the $AND(\wedge)$ gates, inputs and outputs, without revealing their bids.

Briefly, the actions of each bidder are summarized in the following sequence:

1. User creates the cryptography key pair in her own device or delegates the process to a cryptography service. This pair has a twofold function, it will be used to authenticate the bidder inside the system and to perform the required bit commitments.
2. Bidders compute commitment values for their bids and they are published on a bulletin board. The bulletin board ensures that each user has access, during the auction, to the encryption information of every bid in the system.
3. The bidding phase ends. After running a pooling protocol, the winner and sell price are determined and verified. The winner opens the bit commitment, that is to say, send the required information to the system to verify her bid.
4. Bidders compute and post commitment values for the outputs of $AND(\wedge)$ gates in the circuit. The commitments to the inputs and outputs of all $NOT(\neg)$ and $XOR(\oplus)$ gates can be calculated by any agent that has access to the bulletin board, following the steps described in Section 5.3.
5. At this point bidders can execute the Interactive Proofs described in Section 6.1.1 or produce the non interactive proofs. According to the system configuration, regular (Section 6.1.2) or reduced (Section 6.1.3) proofs are created and finally the resulting discreet proof certificates are posted.
6. Any participant, bidder or not, can choose to verify the correctness of the comparison and multiplication function of any bidder using the circuit previously established, the commitments and the proof certificate. The information required to verify the correctness of the auction is published in a public repository without jeopardizing the security of the system.

4. Mathematical foundations of the proposal

The basis of the proposal is commonly found in the cryptography literature, e.g Goldreich (2000), although a brief description is included in this section to offer a better understanding of the mathematical basis of our solution.

Definition 1. *Let $n = pq$ a composite number, where p and q are two primes. We call n a Blum integer if $p \equiv q \equiv 3$ mod 4. n is considered the user's public key, whereas p and q are the user's private key.*

Definition 2. *A quadratic residue modulo n is an integer s such that there exists a $r \in \mathbb{Z}_n^*$ satisfying $s \equiv r^2$ mod n.*

Definition 3. *The Legendre symbol of integer r modulo a prime p, denote $LS_p(r)$ is defined as 0 if p divides r, as +1 in case r is a quadratic residue modulo p ($a \in Q_p$), and as -1 otherwise ($a \in \bar{Q}_p$).*

Definition 4. *The Jacobi symbol of residues modulo a composite N is defined based on the prime factorization of N. Then the Jacobi symbol is calculated as $JS_n(r) = LS_p(r) \times LS_q(r)$.*

Unlike the Legendre symbol, the Jacobi symbol does not reveal whether a is a quadratic residue modulo n, $a \in Q_n$ or $a \in \bar{Q}_n$, respectively. It is indeed true that if $a \in Q_n$ then $JS_n(a) = 1$, however $JS_n(a) = 1$ does not imply that $a \in Q_n$. Note that if $LS_p(a) = -1$ and $LS_q(a) = -1$ then $JS_n(a) = 1$, $a \in \bar{Q}_p$ and $a \in \bar{Q}_q$.

Theorem 1. *If $n = pq$ is a Blum integer, then the function f: $Q_n \rightarrow Q_n$ defined by $f(x) = x^2$ mod n is a permutation. The inverse function of f is:*

$$f^{-1}(x) = x^{((p-1)(q-1)+4)/8} mod\, n \tag{1}$$

The inverse function of f(x), $f^{-1}(x)$, is hereinafter referred to as $Sqrt(x)$.

The bit commitment is based on random numbers applying the aforementioned concepts. The user *Alice* chooses a random number $r \in \mathbb{Z}^*_{n_{Alice}}$ and performs the commitment (c) of a bit ($\widehat{b}$). Alice computes the commitment c, as follows:

$$c = \begin{cases} r^2 \bmod n_A & \widehat{b} = 0 \\ \text{c := -}(r^2) \bmod n_A & \widehat{b} = 1 \end{cases} \tag{2}$$

Therefore the commitment satisfies, $c \in Q_{n_{Alice}}$ if $\widehat{b} = 0$ or otherwise, $\widehat{b} = 1$, $c \in \bar{Q}_{n_{Alice}}$. During the resolution stage Bob can determine the bit information as follows:

$$\widehat{b} = \begin{cases} 0 & \text{if } r^2 = c \bmod n_A \\ 1 & \text{if } r^2 = -c \bmod n_A \end{cases} \tag{3}$$

where r is previously calculated by Alice as $Sqrt(c)$ if $c \in Q_{n_{Alice}}$ or otherwise $Sqrt(N_{Alice} - c)$.

5. Practical Secure Multiparty Computation

The Millionaires Protocol of Yao (1982) can be considered as the starting point of Secure Multiparty Computation (SMC). Research into Multiparty Computation was initiated and principally pursued in the 80s. Today SMC is still a very active field of research, although practical applications have been slow to appear. The application reported in Bogetoft et al. (2009); Damgard and Toft (2008) is of particular importance: SMC was used to calculate the equilibrium price of sugar beet in Denmark, without producers and buyers having to disclose their supply and demand curves.

We use SMC techniques to fulfill the requirements of a sealed bid auction. The aim of SMC is to enable users to perform distributed computing tasks using their private information without disclosing any information. The design of SMC must achieve privacy and correctness requirements even if the system is under attack by an external entity ("the adversary") and/or by a subset of malicious players ("the colluding players").

The basis of SMC techniques is the definition of a function which is able to be evaluated while still preserving the confidentiality of its input values. In our scenario, an auction, two functions have been deployed, a comparison and multiplication function. The comparison function is mandatory for verifying the auction winner whereas the multiplication function will be only used if a currency exchange is required. Both functions are based on well known mathematical properties, our contribution is the design of the boolean functions to accomplish the corresponding mathematical properties, using only the $\wedge, \neg$ and $\oplus$ gates.

5.1. Defining a Comparison function

The objective of the comparison function is to determinate whether a bid is smaller than a given number. This comparison function will be used at the end of the auction to determine the auction winner and the price of the bought good.

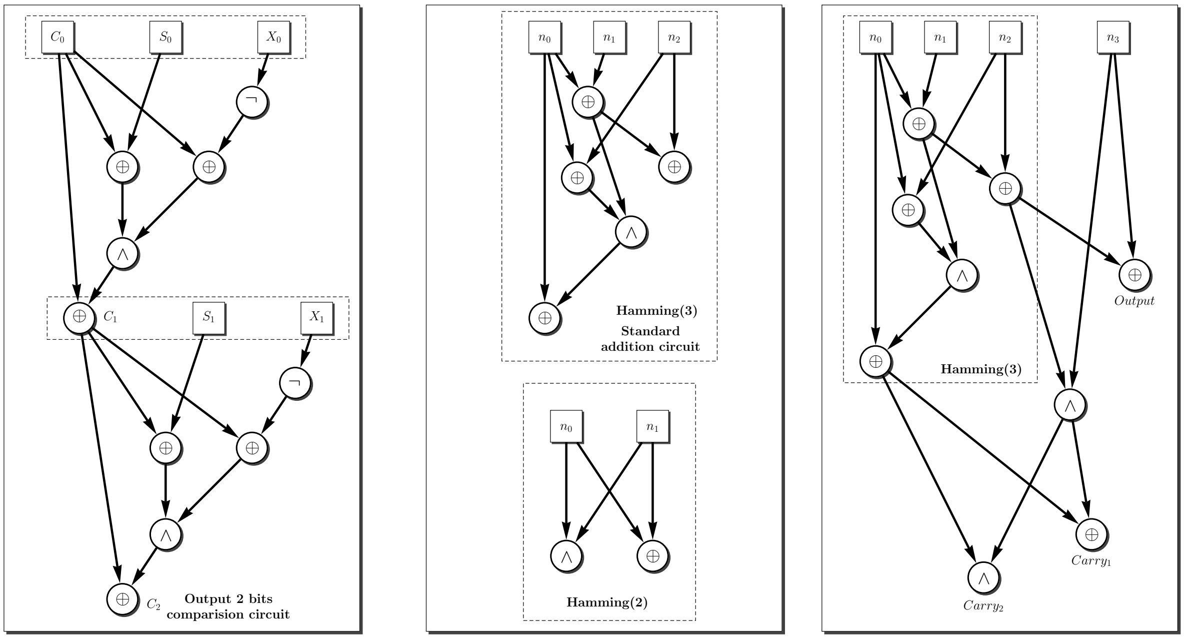
Assume that a bidder has submitted a bid X. This number must be positive. The binary representation of X using n bits is $x_{n-1}2^{n-1} + ... + x_1 2^1 + x_0 2^0$. Therefore the representation of X is a vector $(x_{n-1}, ..., x_1, x_0)$ where $x_i \in \{0, 1\}$.

Given a number S, also in binary representation $s_{n-1}2^{n-1} + ... + s_1 2^1 + s_0 2^0$, the bidder wants to construct a discreet proof that $S \geq X$. To solve this problem, we first define a circuit that outputs 1 if and only if $S \geq X$.

Let $A = 2^n + S - X$ and its binary representation as $(a_n, a_{n-1}, ..., a_1, a_0)$. Then a_n is 1 if and only if $S \geq X$. Moreover A could be $A = S + (2^n - 1 - X) + 1 = S + \bar{X} + 1$ where $\bar{X}$ is the bit-wise complement of X. Therefore a_n is simply the n^{th} carry bit when adding S, $\bar{X}$ and 1. The output of the function A using the boolean gates $\wedge$, $\oplus$ and $\neg$ is C_{k+1} where

$$\begin{aligned} &C_0 = 1, \\ &C_{k+1} = ((S_k \oplus C_k) \wedge (\neg\, X_k \oplus C_k)) \oplus C_k; \end{aligned} \tag{4}$$

In Figure 1(a) is a visual representation of the comparison function.



(a) Comparison (b) Hamming (3) & Hamming (2) (c) Hamming Weight (4)

Figure 1: Graphical representation of comparison and multiplication circuits

5.2. Defining a Multiplication function

Let X be a bid submitted in a currency a, and Y the exchange rate of the currency a to the official auction currency b. It is necessary to multiply the bid X by Y to obtain the bid in the official auction currency. This action must to be transparent to the bidder and be performed without disclosing any information about the original bid.

The multiplication function is based on the standard procedure to multiply two binary numbers. For this reason, we need the binary representation of both numbers X and Y in order to perform the multiplication of binary integers, b_X and b_Y. The multiplicand is shifted m times, where m is the length b_Y. Thereby, the partial products can be represented as m x n matrix, where n is the sum of b_X and b_Y length. If the bit is 1, shifted multiplicand is added to the product. Also we have to take into consideration the carried bit obtained in each operation, since the resulting carries of processing column i are added to following columns.

To calculate the multiplication, we have to add up each column of the product, including the carried bits. At this point, we can design a boolean function with a fixed number of inputs to perform each partial product or evaluate the Hamming weight function of each partial products. The first option is inefficient as in some cases it will have unused inputs or inputs assigned to zero values, whereas the Hamming weight option only uses the precise inputs in each operation.

The boolean circuit for calculating the Hamming weight is carried out, using the algorithm described in the work by Boyar and Peralta (2005). We include here a brief description of the algorithm to improve the readability of the solution.

Let n be an integer number to calculate the Hamming weight, $H(n)$, and b_n its binary representation. We have two cases:

1. n=2^k - 1: For k > 1, a string b_n of length 2^k - 1 can be split into two strings u,v of length $2^{k-1} - 1$ each, plus one string c of length 1.
2. n= $2^k + i$: As in the previous case we split b_n into three strings u,v,c of lengths $2^k - 1$, i and 1 respectively. Note that v may be the empty string.

In both cases, we recursively compute $H(u)$ and $H(v)$.Then, we compute $H(b_n)$ as the sum of $c+H(u)+H(v)$.

The algorithm is based on the standard addition circuit, detailed in Figure 1(b), which computes the sum of three bits (b_0, b_1 and b_2), using only one conjunction:

$$\begin{aligned} o &= b_0 \oplus b_1 \oplus b_2, \\ c &= ((b_0 \oplus b_1) \wedge (b_1 \oplus b_2)) \oplus b_0); \end{aligned} \tag{5}$$

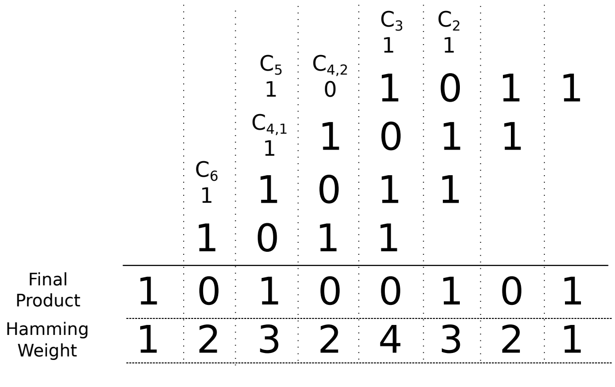
Note that the sum of two bits can be obtained from a standard addition circuit where $b_2 = 0$ (Figure 1(b)).

$$\begin{aligned} o &= b_0 \oplus b_1, \\ c &= b_0 \wedge b_1; \end{aligned} \tag{6}$$

Furthermore, Hamming functions with values higher than three are calculated based on Hamming functions with lower values as described in the aforementioned algorithm. Figure 1(c) details the evaluation of Hamming(4). Its calculation requires two circuits of Hamming(3). The output and carry of the first Hamming(3) circuit are part of the inputs, together with the original input n_3, of the second Hamming(3) circuit.

Therefore, the product is calculated using a set of cascade Hamming circuits with different weights. The weight of a circuit is determined by the number of ones in a partial multiplication and the carry bits of previous operations. From right to left, in each partial multiplication, the execution of a Hamming circuit produces two values, as described in its definition, an output value which is part of the multiplication value and a carry element which is part of the following circuits inputs. This operation is repeated until the last partial multiplication.

Figure 2 shows an example of how the designed Hamming circuits are applied to an ordinary multiplication. The example is the multiplication of 11 $(1011)_2$ and 15 $(1111)_2$, its result being 165 $(10100101)_2$. The multiplication requires the execution of several Hamming circuits, three circuits of Hamming(2), two circuits of Hamming(3) and finally one circuit of Hamming(4).



					C_3 1	C_2 1		
			C_5 1	$C_{4,2}$ 0	1	0	1	1
			$C_{4,1}$ 1	1	0	1	1	
		C_6 1	1	0	1	1		
		1	0	1	1			
Final Product	1	0	1	0	0	1	0	1
Hamming Weight	1	2	3	2	4	3	2	1

Figure 2: Multiplication 11 x 15 and Hamming parameters of each column

5.3. Creation procedure of the boolean circuits

A boolean circuit is created based on a previously defined boolean function. The boolean function is recursively built depending of the number of bits. Figures 1(a) and 1(c) are the visual representation examples of a comparison circuit of 2 bits and a Hamming weight of 4 bits, respectively.

All the bidders have to do their own pair of boolean circuits, one for the multiplication and one for the comparison function. Each node of the circuit is either an input or a boolean ($\neg$,$\oplus$ or $\wedge$) gate and will contain at least two values, the bit value or the plaintext, and its corresponding bit commitment, the encryption value.

Two procedures must be executed to complete the circuit. In order to calculate the bit value of the boolean circuit only the standard boolean operation must be calculated. However, the encrypted value of the inputs and the gates are based on the bit commitment protocol, therefore initially the inputs of the boolean circuit must be calculated using Equation 2. The homomorphic property of our chosen cryptographic scheme makes it possible to spread encrypted values along linear components of the boolean circuit, without jeopardizing the privacy of the information, following these rules:

- The bit commitment value of a $\neg$ gate will be $N - Input$ where N is the bidder's public key and Input is the bit commitment of the input gate.
- In the case of $\oplus$ gate will be $Input_1 \times Input_2 \ mod \ N$ where $Input_1$ and $Input_2$ are the bit commitment of the input gate and N is the bidder's public key.

Non-linear components require special treatment ($\wedge$ gates) and are calculated following the same procedure as used in the input, in accordance with Equation 2.

The reader will have noticed that the bit value of the inputs and the gates cannot be published, only the encrypted values. Each bidder of our proposal plays different roles. On the one hand, she will be the prover of her own boolean circuits. The prover's circuit will contain both values, bit and commitment values. On the other hand, she will be the verifier of the other bidders in the system, and therefore will have as many circuits as there are bidders in the auction. We call this circuit the verifier's circuit and it only contains the encrypted value of the gates. The following subsection explains how the verifier can build the verifier's circuits in her own way and check its correctness.

5.4. Verification procedure of the boolean circuits

The security of the system is based indirectly on the processing of the boolean circuits. We would like to remark that the prover's boolean function contains the binary data of each gate as its encrypted value. The encrypted values are calculated following the steps, as described in the previous section, for creating the boolean circuit. However, the verifier's function only contains the bit commitment values of the boolean gates. In this section, we explain how the verifier can build and verify the prover's circuit without knowing the bit value of the boolean gates.

Initially, the verifier only has access to the bit commitment of the prover's bids, which are published by each bidder in the bidding stage of the auction. As we have explained, these values correspond to the input of the boolean circuit. The verifier can determinate the value of the linear gates ($\oplus$, $\neg$), in the same way as the prover does, starting from the commitment of the inputs, whereas non-linear components ($\wedge$ gates) must to be provided by the prover, once the auction has finished. At this point, the verifier knows the bit commitment values of all the elements in the boolean circuit as well as the circuit's owner, the prover.

After the winner and selling price determination stage, each bidder discloses the binary value of the boolean circuit's output and the square root of its corresponding bit commitment. At this point, each verifier can check the result of the boolean functions of all the bidders. The verification is performed by applying Equation 3 to the binary value and the commitment information of the output.

The positive verification of the output does not imply the correctness of the circuit because the computations are based on the data which the prover has previously sent to the verifier, the $\wedge$ gates. Obviously, neither the input values nor the $\wedge$ gates can be revealed, but it is possible to build a challenge mechanism for the verification of the bit commitment values of the $\wedge$ gates.

Figure 3 graphically summaries the verification procedure of the $\wedge$ gates, based on the creation of two challenges. The mechanism is based on the selection of three random numbers $r_1, r_2, r_3 \leftarrow \mathbb{Z}_N^*$ which are used to verify the properties established in both challenges. These three random numbers are the commitment of the two inputs and the output of the $\wedge$ gate without any specific or previously established order. If the binary value of input or output of the $\wedge$ gate is 0, then its bit commitment will be $c_i = r_i^2 \bmod N$, or otherwise if its value is 1, then its bit commitment will be $c_i = -(r_i^2) \bmod N$, where N is her public key. In accordance with the inputs and the output of the selected $\wedge$ gate, the prover calculates c_1, c_2, c_3

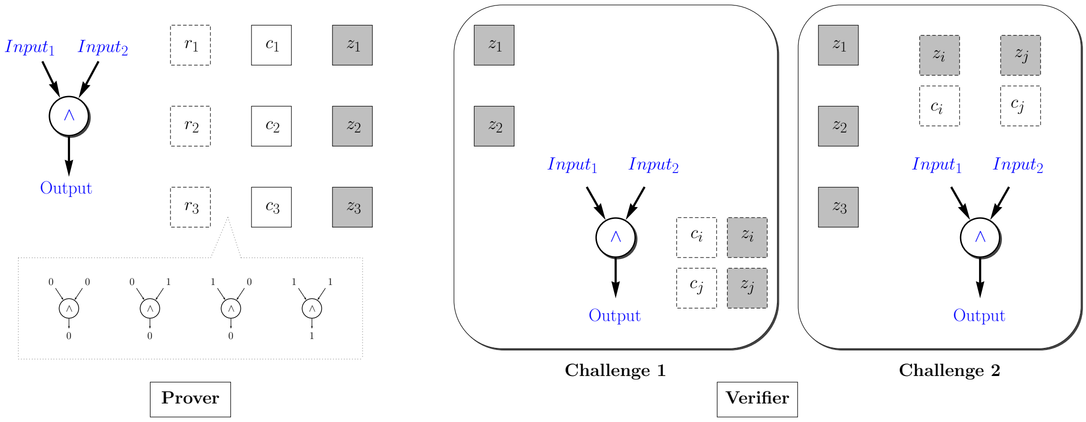


Figure 3: Visual Representation of Challenge One and Two

Challenge one is defined in Construction 1. Its primary goal is to prove that the prover knows the binary value of the output of the ∧ gate without disclosing it but rather by using the bit commitment of the ∧ output ($Output$). Using challenge one the verifier can check whether the bit commitment of the input was correct, but there is also the possibility that the prover cheated in the output of the gate, sending reverse bit commitments. Therefore a new challenge must be defined, challenge two.

The second challenge checks whether or not the prover cheated with the commitments of the output of the ∧ gate, following a similar technique as in the first challenge but now using the bit commitment of the output. To do this, the prover and verifier completes the steps detailed in Construction 2. The steps used to verify challenge two are essentially similar to those in challenge one with slight modifications.

This section has defined two challenges to verify the ∧ values. In Section 6, based on these challenges, three variants of a type of zero knowledge proof are defined to ensure everyone's honesty.

1. **Prover:** $e_1 := c_i \times Output \bmod N$, $e_2 := c_j \times Output \bmod N$, $i \neq j$,
2. **Prover:** $z_1 := Sqrt(e_1)$ and $z_2 := Sqrt(e_2)$.
3. **Prover** → **Verifier**: c_1, c_2, c_3 and z_1, z_2.
4. **Verifier:** Calculates $y_1 := z_1^2$ and $y_2 := z_2^2$
5. **Verifier:** Selects a number c_i in (c_1, c_2, c_3) and calculates $w_1 := Output \times c_i$. If w_1 is equal to y_1 or y_2 moves on to the following step. If not, the Verifier repeats the process (assign c_i) with the other two numbers in (c_1, c_2, c_3). If at least one case is successful then it moves on to the next step. If the assignation cannot be established then the Prover cheated in the commitment of the output O.
6. **Verifier:** Chooses another different number c_j in (c_1, c_2, c_3) $i \neq j$, and calculates $w_2 := Output \times c_j$. If w_2 is equal to y_1 or y_2 (the value has not been previously selected), we can conclude that the commitment of the inputs is valid. If the assignation cannot be established then the prover cheated in the commitment of the output $Output$.

Construction 1: Challenge One

6. Discreet Proofs

In the previous section two challenges were defined to verify the non-linear components of the boolean circuits. Now, a mechanism has to be defined to apply these challenges to verify the correctness of the boolean circuit. *Discreet proofs* were originally defined in Boyar and Peralta (1996); Boyar et al. (2000).

1. **Prover:** $e_1 := c_i \times Input_1 \bmod N$, $b_2 := c_j \times Input_2 \bmod N$, $i \neq j$, $b_3 := Zero$, where $Zero$ is a bit commitment of a binary value zero.
2. **Prover:** $z_1 := Sqrt(e_1)$ and $z_2 := Sqrt(e_2)$ and $z_3 := Sqrt(e_3)$.
3. **Prover → Verifier**: c_1, c_2, c_3 and z_1, z_2, z_3.
4. **Verifier:** Calculates $y_1 := z_1^2$, $y_2 := z_2^2$ and $y_3 := z_3^2$.
5. **Verifier:** Selects a number c_i in (c_1, c_2, c_3) and calculates $w_1 := Input_1 \times c_i$. If w_1 is equal to y_1 or y_2 moves on to the following step. If not, the Verifier repeats the process (assign c_i) with the other two numbers in (c_1, c_2, c_3). If at least one case is successful then it moves on to the next step. If the assignation cannot be established then the Prover cheated in the commitment of the input $Input_1$.
6. **Verifier:** Chooses another different number c_j in (c_1, c_2, c_3), $i \neq j$ and calculates $w_2 := Input_2 \times c_j$. If w_2 is equal to y_1 or y_2 (the value has not been previously selected), we can conclude that the commitment of the inputs is valid. If the assignation cannot be established then the prover cheated in the commitment of the input $Input_2$.

Construction 2: Challenge Two

This type of proof is considered to be one of the new types of zero-knowledge proofs as it does not fit into any of the previous categories. This method is based on a technique called "set certification", which consists of proving that a vector of bit commitments encodes a vector of bits in a given set without revealing the bit-vector itself. The method was originally described by Naor (1991).

The authors cited in the aforementioned references have explored the concept of certification of proofs from a theoretical point of view. Here, we extend the certification concept to a practical level. Several proof models are explained, and the advantages and disadvantages of each model in several application domains are discussed.

6.1. Models of Discreet Proofs

A consideration that needs to be taken into account when choosing between the three models is the communication and computation complexity, that is, how much knowledge should be communicated in order to prove a theorem and the computational resources used. Our proposal offers three options for implementing the requirements in each system where our proposal can be included or considered.

6.1.1. Interactive Proofs

The first option available is the development of Interactive Proofs. These proofs can be performed only if a graphical representation of the comparison circuit exists. The advantages of this method are that only minimal individual computational and communication complexity are required and the user (verifier) defines the security of the system because the proof can be executed an unlimited number of times. On the other hand, this method has several drawbacks: it requires a visual interface to be implemented and all of the bidders must always be online; or an agent must act on behalf of the bidder.

The verifier triggers the manual proof protocol with the selection of an ∧ gate. The prover, in response to the initial message, assigns a sessionID to the message and links it to the challenge related information. The protocol that is used when the verifier selects the challenge to be executed and the information link to the sessionID is erased; or the information is stored until the protocol is complete. The design of a state protocol makes it possible to avoid the deadlock of the protocol, since each bidder can act in a different instance of the protocol, even using different roles; otherwise a bidder cannot execute a new instance until a previous one has finished. Figure 4 is a visual representation of the process executed by the prover and verifier as well as the messages exchange between them. Construction 3 details the information illustrated in Figure 4 for a better understanding of the interactive proof.

An estimation of how many exchanged messages are necessary to prove the veracity of the circuit can be calculated using the following equation:

$$messages = N\_user \times 4 \times ANDs\_circuit$$

1. **Verifier**: Selects one $\wedge$ gate, which is executed on the discreet proofs to check the validity of the inputs and output.
2. **Verifier** $\rightarrow$ **Prover:** The Identification number of the selected gate is sent to the Prover.
3. **Prover**: Selects three random numbers (r_1,r_2,r_3) which fulfill the properties defined in Section 5.4 (c_1,c_2,c_3).
4. **Prover** $\rightarrow$ **Verifier**: c_1, c_2, c_3.
5. **Verifier**: Selects the type of challenge to be executed one or two.
6. **Verifier** $\rightarrow$ **Prover**: Challenges chosen previous step one or two.
7. **Prover**: Calculates the roots z_1, z_2, z_3 which correspond to the previously selected challenge.
8. **Prover** $\rightarrow$ **Verifier:** z_1, z_2, z_3.
9. **Verifier:** At this point the Verifier has all the information needed to execute the algorithm (Section 5.4) that verifies the challenge.

Construction 3: Interactive Proof message sequence

where *N_user* is the number of bidders in the auction, *ANDs_circuit* is the number of $\wedge$ gates of the comparison circuit and 4 messages are necessary per proof. This formula takes into consideration only the execution of one challenge which is randomly selected, otherwise a double message is transmitted if a verifier executes both challenges.

6.1.2. Non-Interactive Regular Proofs

The Interactive Proofs process allows the verifier to establish the security of the system; therefore the level of trust is managed by the user. Although, this property is desirable in every secure system, its execution could produce a large number of messages in the network and moreover requires all of the bidders to be online for the entire auction process.

The Non-Interactive Proofs process is designed to avoid the dependence on the interaction with the bidders. This approach involves automating the proofs; therefore it is necessary to simulate the random behavior of the user. The proposed solution is to use a trusted randomness source to simulate the user. Moreover, the simulation means that the user cannot select how many times the challenge is applied to each $\wedge$ gate; the auctioneer can determine this value using a parameter α during the setup phase. In this way, the resulting system will be considered secure with a probability equal to $1 - 2^{-\alpha}$.

High values of α mean a more secure system but a disadvantage to this approach is that the process of creating the proofs takes more computation time and the resulting proofs are longer. Therefore it is necessary to estimate the appropriate value of depending on the computational efficiency of the device involved in the system and the security required.

The algorithm that creates the Non-Interactive proofs processes all the $\wedge$ gates of the circuit α times and randomly selects the challenge that is applied each time. The detailed sequence of steps is described in Construction 4. The algorithm used to check the validity of the proofs is similar to the algorithm used for the creation of the proof (see Construction 5).

Figure 5 is a visual comparative between both Non Interactive proofs, the regular and the reduced one, which will be explained in detail in the following subsection. At the end of both methods a proof certificate is created. The basis of this certificate is similar to public key certificates. The proof certificate contains only the required information so any user can check the boolean circuits and therefore verify the auction process and result. All the proof certificates of the boolean circuits can be stored in a public repository identified with the auction identification to facilitate the auction whole process verification. To this end a verification tool has been implemented to verify the correctness of a proof certificate. The tool connects to a public repository to obtain the required certificates. After that, the user is able to manually or automatically check the boolean circuits. The information contained in a proof certificate does not jeopardize the privacy of the bid values of the loser bidders.

Figure 6 is a screenshot of the verification tool. This tool shows graphically the textual information

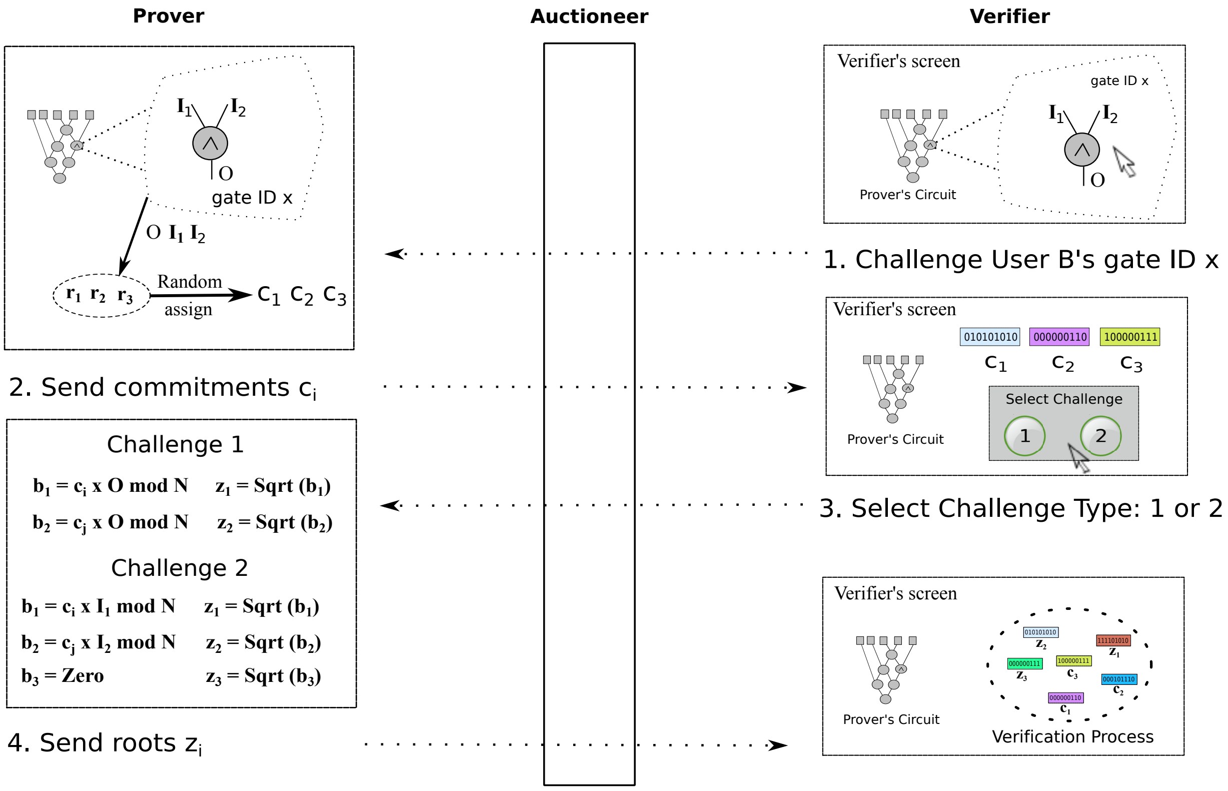


Figure 4: Sequence of messages in Manual Proofs

stored in a proof certificate. Furthermore, this tool enables the user to reproduce and verify the challenges included in the certificate.

6.1.3. Non-Interactive Reduced Proofs

The complexity of the Non-Interactive Proofs depends directly upon the number of $\wedge$ gates of the circuit and the value of the security parameter α.

With a slight modification to the Non-Interactive Proofs, the computational and communication complexity can be reduced and the dependence on the size of the circuit can be circumvented. The solution is to use a random matrix of binary values to reduce the number of commitments generated.

The first five steps of the generation algorithm are the same as for the Non-Interactive Proofs. We describe the following steps:

The Prover only needs to transmit the vector of the square root z and the vector of random selections b. This method was conceived so as to drastically reduce the length of the security proofs. The test also shows that the creation process of this method is quicker than the regular mode in non mobile platforms, but in mobile platforms the matrix multiplication is a costly task.

The verification algorithm is quite similar to the creation algorithm. The Verifier using the vector from the random selections b, follows all the steps of the algorithm until vector y' is obtained. Finally, the Verifier only has to check that all the elements of y' fulfills $y'_i = z_i^2$.

7. System performance evaluation of auction implementation on different platforms

A fully functional system of the Auction proposal has been developed using the Java language. Four servers have been implemented, Auctioneer, Randomness, Public Key Repository and Proof Repository server. In addition to the servers, a client has been implemented as an applet and a mobile application.

1. **Prover:** Requests two certified random numbers, $\lambda \in \{0,1\}^*$ and $\phi \in \{0,1\}^*$, from a Randomness Service.
2. **Prover:** Generates a vector of random boolean numbers $s = (s_1, s_2, \ldots, s_n)$ ($s_i \in \{0,1\}$) using the previous requested number ϕ. The size of vector S, is $\alpha \times nAND$, where $nAND$ is the number of $\wedge$ gates of the comparison circuit. The generated vector of random numbers is used to simulate the election of the challenge, one or two.
3. **Prover:** Generates a vector of random numbers $s' = (s'_1, s'_2, \ldots, s'_n)$ ($s_i \in \{0,1\}^*$) using the previous requested number λ. The size of vector s', is $\alpha \times nAND \times 3$. This vector represents the numbers used for performance the commitment (r_1,r_2,r_3).
4. **Prover:** Computes β, the smallest number with $J(\beta/N) = -1$ where N is the public key of the Prover.
5. **Prover:** Calculates new vector c using the previous generated vector s'.

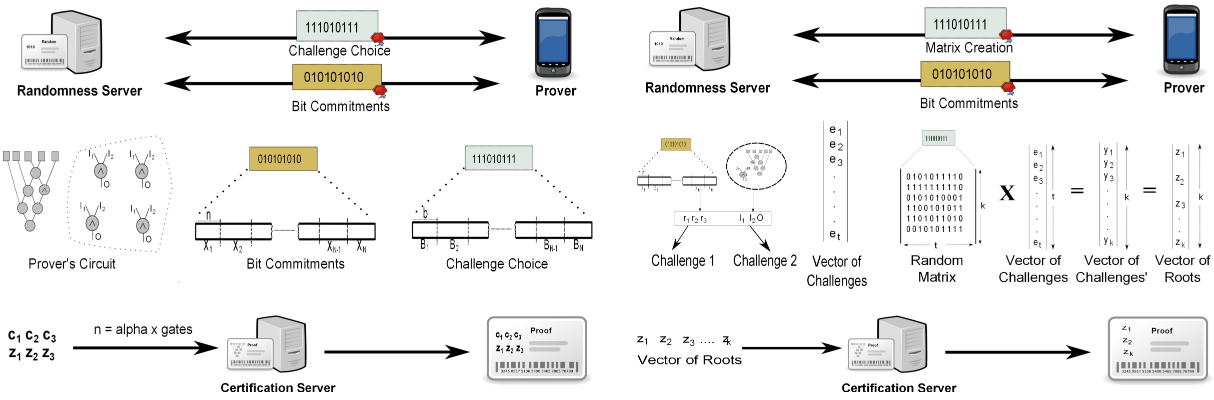
```
For i= 1 to n
    $j_i$ := $J(s_i / N)$
    If $j_i = -1$ then $c_i$ := $s'_i \times \beta$ mod N
    else             $c_i$ := $s'_i$.
```

   When the algorithm ends, all the elements of c have a Jacobi number equal to 1 and the c and s' lengths are the same.
6. **Prover:** The commitment of each $\wedge$ gate is generated α times using the number of vector c. Three numbers are chosen of c (c_i, c_j, c_z) and the Prover checks that each number suits the corresponding commitment. In the case that the number does not fit the commitment, it is inverted using the public key of the Prover, $c_i := -c_i \bmod N$. The Prover uses the boolean vector b ($b_i \in \{true, false\}$) to annotate whether the original numbers need to be inverted. The challenge to be executed is selected from the vector s.
7. **Prover:** The square roots vector is calculated, z, according to the type of challenge selected.
8. **Prover:** Generates a certificate of proofs. The certificate includes a structure of proofs as well as administrative information. Basically, each structure contains the vector of modifications b and the square roots of the commitments Z .

Construction 4: Construction of Non Interactive Proofs

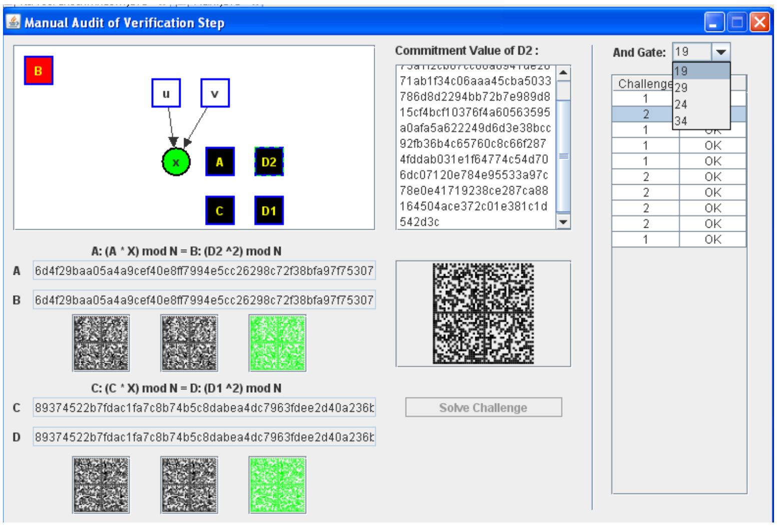
1. **Verifier:** Requests the certified random numbers, λ and ϕ, from the Randomness Server.
2. **Verifier:** Computes β and using the boolean vector (b) obtains the vector c.
3. **Verifier:** Checks the circuit's commitments using the vector c and the square roots z following the algorithm described in Section 5.4.

Construction 5: Steps to validate the Proofs



(a) Regular procedure (b) Reduced procedure

Figure 5: Non-interactive proofs creation process



Commitment Value of D2 :

```
71ab1f34c06aaa45cba5033
786d8d2294bb72b7e989d8
15cf4bcf10376f4a60563595
a0afa5a622249d6d3e38bcc
92fb36b4c65760c8c66f287
4fddab031e1f64774c54d70
6dc07120e784e95533a97c
78e0e41719238ce287ca88
164504ace372c01e381c1d
542d3c
```

And Gate: 19

19
29
24
34

A: (A * X) mod N = B: (D2 ^2) mod N

A 6d4f29baa05a4a9cef40e8ff7994e5cc26298c72f38bfa97f75307

B 6d4f29baa05a4a9cef40e8ff7994e5cc26298c72f38bfa97f75307

C: (C * X) mod N = D: (D1 ^2) mod N

C 89374522b7fdac1fa7c8b74b5c8dabea4dc7963fdee2d40a236b

D 89374522b7fdac1fa7c8b74b5c8dabea4dc7963fdee2d40a236b

Challenge	
1	
2	
1	OK
1	OK
1	OK
2	OK
2	OK
2	OK
2	OK
1	OK

Figure 6: Proof certificates verification tool

Android has been selected as the mobile platform to run the auction client. Android offers several benefits for developers, among other concerns, our choice was driven by the fact that the same java code can be executed in a smartphone and a non mobile platform.

In a practical evaluation, the client has been executed and evaluated on three different platforms, a regular laptop with a mac os x operating system, a windows pc and two android devices, a telephone and a tablet, which differ in their hardware configuration (Table 1).

Four procedures have been chosen to evaluate the system's performance. These elements have been chosen because they are essential in accomplishing the security requirements of these auctions. Figure 7 details the cpu consumption, in milliseconds, of the chosen elements in the aforementioned platforms.

The key generation does not have to be executed in every auction, even the bidder can create the key in a platform and later export to another one. In the case it is necessary to create a key, this is done before the auction's stage. Figure 7 (a) shows that the key generation process is not a costly process in the evaluated platforms, although it is ten times slower on a mobile than on a conventional platform.

The bit commitment is only done once during the bidding stage. As Figure 7 (b), shows its execution does not compromise the performance of the system. Note that each bid is translated into its binary representation and then the bit commitment procedure is apply on each bit. Although, the execution does

6. **Prover:** Generates the vector of challenges e. The elements of the vector of challenges are calculated following the definition in Construction 1 or 2 according the challenge chosen.
   The resulting length of the vector of challenges depends on which challenge has been selected during the process, but it can be estimated using the formula $t := \alpha \times numberAND \times MChallenge$, where $MChallenge$ is 2 or 3 depending whether challenge one or two was chosen.
7. **Prover:** Obtains another certified random number, $\theta \in \{0,1\}^*$ from the Randomness Server, and creates a binary random $t \times k$ matrix m. The length of the rows is a configurable parameter k. The lowest value between k and α establishes the security of the system. Therefore an immediate configuration can set $\alpha = k$.
8. **Prover:** Computes a new vector y using the following formulae $y = e \times m$.
9. **Prover:** Calculates the square root to all the elements of y and obtains the new vector z.

Construction 6: Algorithm to generate Non-Interactive Reduced Proofs

Platform	**CPU**	**Operating System**
Mac mini	2.5 Ghz Intel Core i5	MacOs X 10.9
PC Gateway	2.8 Ghz Pentium Dual core E5500	Windows 7 Professional
Samsung Galaxy S3	1.4 GHz Quad-core Cortex-A9	Android 4.1.2
Samsung Galaxy Tab 2	1 GHz Dual-core Cortex-A9	Cyanogenmod 10.2

Table 1: Description of platforms used in the performance evaluation

not delay the client, the bits necessary to represent the bid can be reduced, in the case of goods with extremely high prices. In this case the price can be determined as a value in an interval $(max - min)/step$, where max and min are the maximum and minimum bid, respectively and $step$ is the step bid.

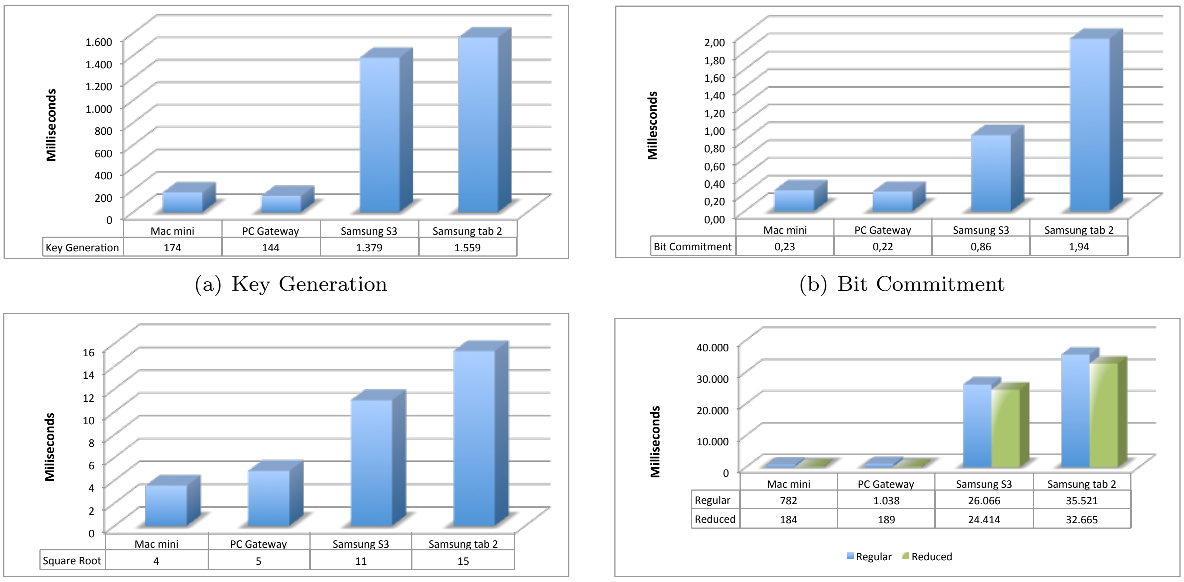
The square roots are calculated each time a challenge has to be executed or a bit commitment is going to be verified. During the manual proof each verification of a $\wedge$ requires the calculation of a square root in the prover client. Moreover, the creation of both Non Interactive proofs involves the calculation of several square roots, depending on the number of $\wedge$ gates and the α parameter. Again the problematic differences between mobile and non mobile platforms are notorious although it could be still considered as a feasible solution.

Lastly, the performance of Non Interactive reduced proofs do not shown the expected result in the mobile application. Although, the execution is less than normal, the difference is reduced, whereas in the non mobile platform it is considerable. In conclusion, the matrix multiplication in mobile platform execution has a similar CPU time consumption as the square roots calculation.

Generally, the evaluation of the performance demonstrates that the implementation of our proposal is practical and viable. Although the system's performance obtains better results on windows and mac platforms, the client's execution in the mobile system is completed within a reasonable time. The differences between android devices, with different hardware configuration, can be seen in Figure 7. The evolution of mobile devices ensures that our proposal will be of a performance close to non mobile platforms relatively soon.

8. Conclusions

This paper has presented a sealed bid and multi-currency auction using secure multiparty computation (SMC). SMC is not a new concept however it is not common in the specialized literature to find practical implementations of a secure system using this technique. Two boolean functions have been designed as the standard requirement to apply SMC and its common goal is to operate with the bids without disclosing any information. On the one hand, the comparison function makes it possible to compare two bids, on the other hand, the objective of multiplication function is to calculate the bid in the auction's currency.



(a) Key Generation

(b) Bit Commitment

(c) Square Root

(d) Non Interactive Proofs

Figure 7: Performance evaluation of client application on heterogeneous platforms

Moreover three models of discrete proofs have been defined: interactive, regular and reduced Non Interactive proofs. The bids of the non winner bidders can be verified without revealing the bids, not even the auctioneer has information related to these bids. These proofs are certificable therefore all the information related to the auction can be publicly stored without jeopardizing the privacy of the system.

A system performance of the implementation has been evaluated to verify how practical our proposal is. The code has been executed on mobile and non mobile platforms. Although the result of the evaluation is significantly better on the non mobile platforms, the execution on mobile platforms has a more than acceptable CPU time consumption, showing beyond doubt the concept of our proposal as a practical SMC solution.

References

Abe, M., Suzuki, K., 2002. Receipt-free sealed-bid auction. In: Proceedings of the 5th International Conference on Information Security. ISC '02. Springer-Verlag, London, UK, pp. 191–199.

Bogetoft, P., Christensen, D., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J., Nielsen, J., Nielsen, K., Pagter, J., Schwartzbach, M., Toft, T., 2009. Secure multiparty computation goes live. In: Financial Cryptography and Data Security. Springer-Verlag, pp. 325–343.

Boyar, J., Damgård, I., Peralta, R., 2000. Short non-interactive cryptographic proofs. J. Cryptology 13, 449–472.

Boyar, J., Peralta, R., 1996. Short discreet proofs. In: Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques. EUROCRYPT'96. Springer-Verlag, pp. 131–142.

Boyar, J., Peralta, R., 2005. The exact multiplicative complexity of the hamming weight function. Tech. rep., Electronic Colloquium on Computational Complexity (ECCC).

Boyd, C., Mao, W., 2000. Security issues for electronic auctions. Tech. Rep. HPL-2000-90, HP labs.

Chen, X., Lee, B., Kim, K., 2004. Receipt-free electronic auction schemes using homomorphic encryption. In: Information Security and Cryptology - ICISC 2003. Vol. 2971 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 259–273.

Chen, K., 1998. Authenticated encryption scheme based on quadratic residue. IEEE Electronics Letters 34(22) pp.2115–2116.

Damgard, I., Geisler, M., Kroigaard, M., 2007. Efficient and secure comparison for on-line auctions. In: Information Security and Privacy. Vol. 4586 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 416–430.

Damgard, I., Geisler, M., Kroigaard, M., 2009. A correction to 'efficient and secure comparison for on-line auctions'. IJACT 1 (4), 323–324.

Damgard, I., Toft, T., 2008. Trading sugar beet quotas - secure multiparty computation in practice. Ercim News 73, 32–33.

Franklin, M., Reiter, M., 1995. The design and implementation of a secure auction service. In: Proceedings of the IEEE Symposium on Security and Privacy. pp. 302–312.

Gao, C., Yao, Z., Xie, D., Wei, B., 2011. Electronic sealed-bid auctions with incoercibility. In: Electrical Power Systems and Computers. Vol. 99 of Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, pp. 47–54.

Goldreich, O., 2000. Foundations of Cryptography: Basic Tools. Cambridge University Press, New York, NY, USA.

Harkavy, M., Tygar, J., Kikuchi, H., 1998. Electronic auctions with private bids. In: Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3. WOEC'98. USENIX Association, pp. 61–74.

Howlader, J., Kar, J., Mal, A., 2012. Coercion resistant mix for electronic auction. In: ICISS. Vol. 6633. Springer-Verlag, pp. 238–248.

Lee, C., Ho, P., Hwang, M., 2009. A secure e-auction scheme based on group signatures. Information Systems Frontiers 11 (3), 335–343.

Nakanishi, T., Yamamoto, D., Sugiyama, Y., 2004. Sealed-bid auctions with efficient bids. In: Information Security and Cryptology - ICISC 2003. Vol. 2971 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 230–244.

Naor, M., 1991. Bit commitment using pseudorandomness. Journal of Cryptology: the journal of the International Association for Cryptologic Research 4 (2), 151 – 158.

Omote, K., Miyaji, A., 2002. A second-price sealed-bid auction with public verifiability. Transactions of Information Processing Society of Japan 43 (8), 2405–2413.

Palmer, B., Bubendorfer, K., Welch, I., 2010. A protocol for verification of an auction without revealing bid values. Procedia Computer Science 1 (1), 2649 – 2658.

Peng, K., 2011. Secure e-auction for mobile users with low-capability devices in wireless network. In: Proceedings of the 5th IFIP WG 11.2 International Conference on Information Security Theory and Practice: Security and Privacy of Mobile Devices in Wireless Communication. WISTP'11. Springer-Verlag, pp. 351–360.

Peng, K., Boyd, C., Dawson, E., Viswanathan, K., 2003. Five sealed bid auction models. In: Australian Information Security WorkShop, AISW. pp. 77 – 86.

Shparlinski I., Banks, W., Lieman, D., 2000. An extremely small and efficient identification scheme. In: Proc. 5th Aust. Conf. on Information Security and Privacy. volume 1841, pp 378–384.

Yao, A., 1982. Protocols for secure computations. In: Symposium on Foundations of Computer Science. pp. 160–164.

Zhang, F., Li, Q., Wang, Y., 2000. A new secure electronic auction scheme. In: EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security. pp. 54–56.