J. A. Montenegro, M. J. Fischer, J. Lopez, and R. Peralta, "Secure sealed-bid online auctions using discreet cryptographic proofs", *Mathematical and Computer Modelling*, vol. 57, pp. 25832595, 2013.
http://doi.org/http://dx.doi.org/10.1016/j.mcm.2011.07.027
NICS Lab. Publications: https://www.nics.uma.es/publications

Jose A. Montenegro, Michael J. Fischer,Javier Lopez, and Rene Peralta, "Secure Sealed-Bid Online Auctions Using Discreet Cryptographic Proofs", Mathematical and Computer Modelling, Elsevier, pp. 2583-2595, Volume 57, Issues 11-12, June 2013.

Secure Sealed-Bid Online Auctions Using Discreet Cryptographic Proofs

Jose A. Montenegro[a,*], Michael J. Fischer[b], Javier Lopez[a], Rene Peralta[c]

[a] *Dpto. Lenguajes y Ciencias de la Computación. ETSI Informática Málaga. Universidad de Málaga. Spain*
[b] *Department of Computer Science. Yale University. USA*
[c] *Computer Security Division. National Institute of Standards and Technology (NIST). USA*

Abstract

This work describes the design and implementation of an auction system using secure multiparty computation techniques. Our aim is to produce a system that is practical under actual field constraints on computation, memory, and communication. The underlying protocol is privacy-preserving, that is, the winning bid is determined without information about the losing bids leaking to either the auctioneer or other bidders. Practical implementation of the protocol is feasible using circuit-based cryptographic proofs along with additively homomorphic bit commitment. Moreover, we propose the development of a *Proof Certificate* standard. These certificates convey sufficient information to recreate the cryptographic proofs and verify them offline.

Keywords: Discreet Proofs, Multiparty Computation, Online Auctions, Zero-Knowledge Protocols, Proof Certificates.

*Corresponding author

Email addresses: monte@lcc.uma.es (Jose A. Montenegro), michael.fischer@yale.edu (Michael J. Fischer), jlm@lcc.uma.es (Javier Lopez), peralta@nist.gov (Rene Peralta)

Parts of this work were presented by Michael Fischer at the The Institute for Advanced Study and DIMACS *Workshop on Decentralized Mechanism Design, Distributed Computing, and Cryptography*, Princeton, NJ, June 3–4, 2010.

This work was supported in part by the National Science Foundation under Grant CCR-0081823.

1. Introduction

A secure multiparty computation (SMC) is a type of communication protocol that allows a set of participants to agree on the value of a function on private data without disclosing the data itself. A large class of problems in e-commerce and e-government can be solved using SMC. Consider the following:

- Electing a government official: the private data is each voter's choice, the function that needs to be computed is which candidate has the most votes;[1]
- Awarding a government contract: the private data is each bidder's terms, the function to be computed is which bidder offers the best terms;
- Finding common patients in two medical-records databases: the private data are patients' identification information, the function to be computed is the set of social security numbers that appear in both databases.

In these examples, SMC is used to protect voter privacy, protect business secrets, and comply with the regulations of the Health Insurance Portability and Accountability Act of 1996 (HIPAA), respectively. SMC techniques can also enforce some rules related to keeping participants honest: e.g., you are not allowed to change your vote once you cast it, or you are not allowed to lower your bid now that you know you lost. Generic solutions for a large class of SMC problems are feasible as far as Computational Complexity Theory is concerned; these solutions use polynomial rather than exponential resources such as memory and CPU cycles. We are designing practical SMC techniques, i.e., protocols that solve problems such as the above examples under actual field constraints on computation, memory, and communication. The objectives of SMC can be achieved using discreet cryptographic proofs as defined in [7].

We use SMC tools to implement sealed-bid auctions. We developed our system from scratch, including a service that allows users to synchronize computer clocks via the Internet, i.e., the timing of the auction is configured based on NIST's Internet Time Service.

Interactive and non-interactive proofs have been designed and developed. Implementations of both versions are efficient and suitable to real-time applications. Moreover, we develop the concept of "Proof Certificate". These

[1] In real elections, one often additionally desires the vote tally for each candidate.

certificates contain sufficient information to recreate the cryptographic proofs and verify them offline. The proofs can thus be posted in a public repository without any disclosure of private information.

The rest of this paper is structured as follows: Section 2 provides a brief description of auctions, including a discussion of security properties of sealed-bid auctions and related work. Section 3 describes the various components of our auction design. Also this section identifies the different stages of an auction and explains the security mechanisms involved at each stage. Section 4 presents the basic cryptographic concepts that underlie our proposal, including bit-commitment and authentication tools. Section 5 discusses comparison of two committed numbers as a problem in SMC. The method involves a comparison circuit using *AND*, *XOR* and *NOT* gates. The interaction among the participants of the auction is explained in detail.

Section 6 describes discreet proofs and discusses the creation of proof certificates. Several models of discreet proofs are described. Section 7 describes implementation details of the proposal and discusses the computation costs of creation and verification of proofs. Our conclusions are reported in Section 8.

2. Online Auctions

The mathematical theory of auctions has been intensively studied during the last six decades. Many game-theoretic models have been proposed. We include in this paper only what is needed to explain our proposal. Further details about auctions can be found in [27].

Auctions can be categorized according to their particular rules of operation. These rules specify features such as bidding restrictions, event timing, information revelation, and pricing and allocation policies. We mainly focus on how bids are submitted and how the final price is determined. We identify two different categories of auctions based on these properties:

- *Open-cry auctions* are auctions in which bids are publicly announced to all participants. Also this category is divided into two types of auctions depending on whether the price that is bid increases or decreases during the course of the auction. In the *English auction* the auctioneer announces a minimum selling price at the beginning of the auction, and the bidders successively increase their bids until no more offers are announced. On the other hand, the *Dutch auction* uses the strategy of beginning with a high asking price. The price is lowered until a bidder accepts the asking price.

- *Sealed-bid auctions* differ from open-cry auctions in that bidders secretly submit their bids without necessarily having any information about competing bids. This type of auction requires at least two phases i) the bidding phase, when bidders submit their offers, and ii) the resolution phase, when the winner and the selling price are determined. We consider first and second-price auctions. The winner is always the person who submits the highest bid. In first-price auctions, the winner pays the price submitted in the bid. In second-price auctions, the winner pays the amount offered in the second-highest bid.

In the next section, we describe desired properties that a sealed-bid auction must achieve.

2.1. Security Requirements of Sealed-Bid Auctions

Studies of sealed-bid auctions impose various requirements on the process, or increase the relevance of a particular requirement over others. As suggested in [32], it is good practice to establish basic security properties in order to frame the discussion on different auction models. *Basic properties* are those properties that the majority of studies have agreed on: correctness, confidentiality and fairness. Other desired properties are discussed below.

[8] points out the importance of *minimization of trust* in one party, particularly the auctioneer. In general, the auctioneer can play the role of a bidder or can collude with a bidder to attempt to help that bidder win the auction. In our work, we need not trust the auctioneer because no information about the bids is disclosed, even at the end of the auction.

[20, 25, 48] include *performance* as a desirable property. Usually, there is a trade-off between security and performance. Several proposed cryptographic primitives decrease the performance of a system and affect its usability in other ways. Therefore a balance between security, functionality, and performance must be achieved. For example, [29] bases the security of the sealed-bid auction system on oblivious transfer. Their technique requires the preprocessing of a large amount of information. To address this inconvenience, [29] proposes sending the information stored on a magnetic medium before each auction starts. Although, the cryptographic designs of the protocol fulfill the security requirements, implementation of the process is considered impractical and it can directly affect the security of the entire system. Detailed information on the performance of our system can be found in section 7.

Non-repudiation is also considered a desirable property in [20, 48]. This property can be achieved if the submitted bid is directly associated with the identity of the bidder or indirectly linked to some identity 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 of our design.

The security objective of *verifiability* is discussed in [25, 30, 32]. The authors define this as all participating parties being able to check the source and completion of a bid. [48] describes a specific aspect of verifiability named *validity of the successful bid*, which occurs when the successful bid is the highest among 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 realized without disclosing information about the bids submitted by the losing bidders. This property is the cornerstone of our proposal. The auction security mechanism has been designed taking into consideration that the verification process can be easily executed in any place, and that it does not necessarily have to be executed by any particular party to the auction. Therefore the proof becomes ubiquitous, and it is possible to post the proof in a public repository without suffering any security exposure; also it is possible to audit the outcome of the auction at a later date. The design of the proof certificate (section 6.1) is driven by ubiquitous verification of the auction with minimal disclosure of information.

We consider an auction to be *fair* if it is symmetric. This means that all the bidders have access to the same information during the bidding phase (basically the setup information); if one bidder has more information about the auction than others, the auction will become asymmetric. Furthermore, the auction organization must continue to preserve the symmetric property of the auction process, so that only minimal information is disclosed after the resolution stage. Exactly what "minimal" means depends on the kind of auction (first-price or second-price auction) and whether the identity, bid, and price paid by the winning bidder is deemed to be public knowledge.

2.2. Related Work

The design of secure auctions has been an actively researched topic since the 1990s. This was stimulated by the 1994 decision of the Federal Communications Commission (FCC) to allocate licenses for electronic spectrum via competitive auctions. As of 2011, there are over a hundred published papers on auctions. It is not possible to do justice to all these works here. We limit ourselves to mentioning a few representative papers, roughly corresponding to an ad-hoc classification of techniques. We also restrict our discussion to single-good auctions. The important work on combinatorial auctions is outside the scope of this paper.

[16] was an early effort to accomplish the basic requirements of online auctions. The work is based on a cryptographic primitive developed by the researchers and named verifiable signature sharing. This primitive enables the holder of a signed message to share the signature among a group of users. Only the members of the group can reconstruct the signature, even if some of the group members are faulty. The main drawback of this approach is that all the bids are opened at the end of the bidding period. A more recent paper, [31], proposes a protocol in which only the auctioneer learns the values of losing bids.

Our work belongs to the class of protocols in which neither the auctioneer nor the bidders learn the value of losing bids.[2] This class can be further divided into sub-classes as follows:

- Protocols that do not use an auctioneer. That is, the bidders decide among themselves who has committed to the highest bid. In this model, an auction is an instance of a generic SMC problem. In principle, generic SMC solutions apply. For an auction-specific proposal, see [9].

- Protocols that use one auctioneer only. The work presented here belongs to this class. See also [4, 38, 43].

- Protocols that split the auctioneer role into two or more agents which are assumed not to collude. This is by far the largest class in this classification. Results in this model have various trust models, privacy goals, and computational complexity assumptions [1, 20, 21, 26, 29, 30, 36].

Since the main operation in our chosen problem is zero-knowledge integer comparison, it is fair to ask why we do not use techniques based on Schnorr's protocol [39]. These techniques have been shown to yield very efficient zero-knowledge proofs for integer comparison. The answer is that real auctions, as well as other e-commerce and e-government applications, are quite complex. They typically involve evaluation of arbitrary predicates on encrypted data. Discreet proofs based on circuit methods are, in general, the best way we know how to do this. Thus, as a proof of concept, this work is aimed at providing evidence of the feasibility of practical privacy-preserving solutions to real problems in the online world.

[2]This is not quite true of second-price auctions. In this case, the second highest bid is the selling price and thus must be revealed at least to the auctioneer. We do not concern ourselves with this issue here.

3. Design of a Sealed-Bid Auction System

A secure auction system requires the design and deployment of a number of online services. Some of these services are described below. A simplified version of the system is depicted in figure 1. We use the object model and data flow diagram for a generic auction application that appears in [23].

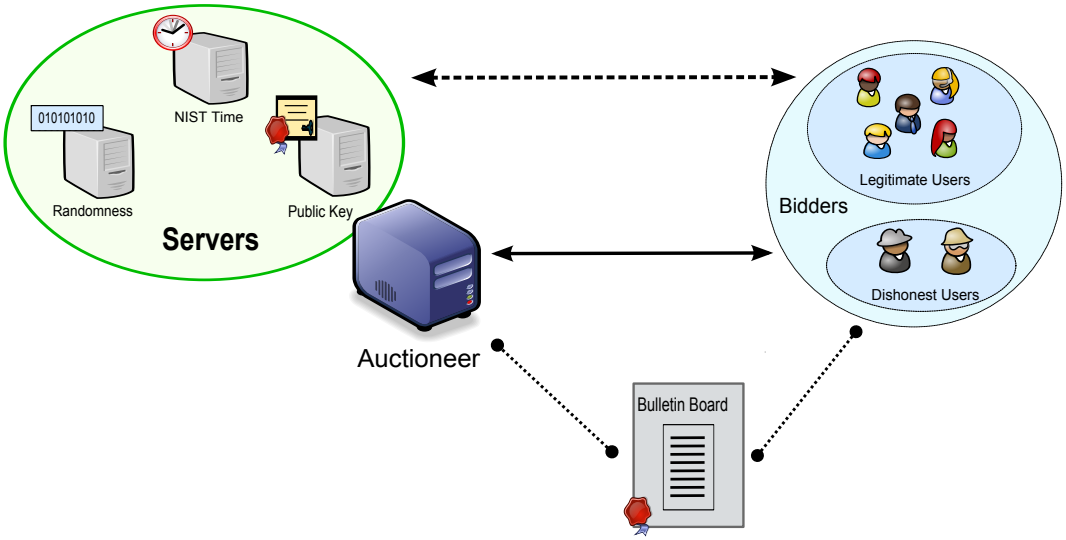

Figure 1: The online environment

Randomness Service Cryptography would be impossible without random numbers. [17] provides a good explanation of the role of randomness in cryptographic functions. Moreover, several cryptographic systems are theoretically designed assuming access to a shared source of random numbers. Cryptographers refer to this model of computation as the "random oracle model". Some security vulnerabilities are caused by incorrect implementation or improper use of a random number source (see [18, 21, 22]).

There is currently no standard for an online source of shared randomness. We support the creation of a service such as the one outlined in [15]. This service would generate random numbers following the basic recommendations of [12, 35] as well as NIST's Special Publication SP800-90 ([3]). Based on the fundamentals of Service Oriented Architecture (SOA) [14], the proposed service would provide certified and time-stamped random numbers.

Time Service The timing of the different auction steps has a significant influence on the performance of the system. Usually, the bid events

are asynchronous and controlled by participants, but other steps are synchronous and are controlled by the auctioneer. This means we need a trustworthy time source for synchronization. [45] explores in detail the real-time issues of online auctions.

In our work, when the server and the protocol have not been specifically deployed to provide real-time services, we make use of the standard time service provided by NIST[3]. There are three protocol standards available to access time information. The standards for the Time Protocol [34] and the Daytime Protocol [33] provide very basic and efficient services. However, they are insufficient for our application, as the specifications do not include support to time certification mechanisms. The Network Time Protocol (NTP) [28] improves upon the previous standards and includes security properties. Therefore, NTP was the selected option.

Public-Key Service Our proposal is based on public-key cryptography; this means that each bidder uses a specific pair of keys (a public key and a private key) to submit an encrypted bid. Asymmetric cryptography requires a certification authority infrastructure. These trust authorities link the public key to the identity of the user. Numerous certification solutions have been proposed and mainly differ in the trust model, hierarchical or anarchic, and the data structure of the certificate. In our case, following the standards, a root node of a Public-Key Infrastructure (PKI) has been implemented.

Bulletin Board The previously detailed services are related to security properties. The bulletin board is a service that fulfills the requirements of the auction system. It is a trustworthy repository where all the data concerned with the auction is made public. It provides the bidders with access to all the shared information generated in the auction.

The bulletin board has a close relationship with the time service, because much of the information that is submitted must be time-stamped. For example, the encrypted bids must be included in the bulletin board before the bidding stage of the auction ends. The user who does not submit information before the end of the bidding phase is not considered a bidder and can not participate in the following stages. Moreover the encrypted bid is used as input to the comparison of bids function. The discreet proofs generated are also published in the bulletin board.

[3]The service can be reached at http://www.nist.gov/pml/div688/grp40/its.cfm. Available time servers are listed at http://tf.nist.gov/tf-cgi/servers.cgi.

3.1. Stages of the Auction

Although an auction is usually performed in two phases, the bidding and the resolution phase, we decided to split these phases to emphasize the importance of the security procedures in the auction. The proposed auction system is composed of the following five stages:

Setup The auctioneer establishes the configuration parameters of the auction and sends all the data to the user after the authentication process. There are two parameters relating to the auction and two more regarding to the security algorithms.

The auction parameters are:

- Price interval: The bottom (T1) and top (T2) valid amounts and the bid increment amount (S) are established. This information is used to optimize the input of the comparison function, reducing the necessary bits. Valid bids are in the integer set $\{T1 + iS \mid 0 \leq i \leq (T2 - T1)/S\}$. A bid of $T1 + iS$ is represented by the value i.
- Auction duration: The auction organization determines the end of the bidding process. After this point, no more bids are allowed and the security protocols begin to transmit the related security information.

The security parameters are:

- Alpha (α): This parameter determines the security of the cryptographic proofs. High values of α mean more security (at the cost of more computation and communication). It is necessary to configure the correct security parameter based on the resource constraints and security needs of the system. The security of the system is $1 - 2^{-\alpha}$. Setting $\alpha = 20$ yields approximately 99.9999% security.
- Matrix Method: If this option is selected a random Boolean matrix is used to reduce the length of the cryptographic proofs. A detailed description of the creation and verification process of discreet proofs using the matrix can be found in section 6.5.

Bidding After configuration of the auction parameters, the users can authenticate themselves and become bidders. Two authentication methods have been deployed. The basic method is password-based, although the password is not sent directly; it is preprocessed using a hash function. A man-in-the-middle attack could be successful if a secure channel

were not established. The advanced authentication method we implemented is based on the Quadratic Residuosity Assumption (QRA). It makes use of the bidder's cryptographic keys and the trusted randomness service. Examples of authentication protocols based on the QRA can be found in [10, 41].

The authenticated user becomes a bidder by sending an encrypted bid to the auctioneer. The bid is encrypted bit-by-bit using a bit-commitment protocol. This primitive is explained in detail in section 4.1).

Time up At this point the bidding process is stopped and the determination and verification stages are executed. It is important that the clocks of the server and the bidder are synchronized to avoid any situations that might result in disputes. The NIST standard time server is used for this purpose.

Winner and Selling Price Determination We implemented two sealed-bid models, first-price and second-price (Vickrey) auctions [44]. The main difference between them is that in first-price auctions, the winner and the selling price are determined in only one round; the Vickrey auction requires two rounds. Additionally, our second-price auction implementation reveals the identity and bid of the second highest bidder to the auctioneer. Preventing this release of information is possible in theory, but at a considerable increase in the complexity of the protocol.

Each round of the winner and sell-price determination stage is based on a polling method. The server asks the bidders about a range of prices and the bidders send either an affirmative or a negative response. Starting with the top price T2, and using the step size as the decrement value, the auctioneer polls the bidders. This continues until one or more bidders states that his/her bid matches the current value. At this point, the corresponding bidder opens his/her bid for public inspection. That is, the bidder executes the revealing stage of the bit-commitment protocol (section 4.1, step 4.1). That bidder is declared the winner if the bit-commitment verification process is successful. If so, the process to determine the selling price can start. If not, the bidder is cheating and the process to determine the winner continues without the cheating bidder. In case a tie occurs, the Vickrey auction is converted to a first price auction; the winner is the user who submitted the highest bid first. Immediately after the completion of the winner determination stage, the price determination is executed.

Proof Publication and Verification Only the highest (and, in the Vickrey auction implementation, the second highest) bids are opened in the previous stage. Since the remaining bids are not opened, it is necessary for those bids to be verified as indeed being lower than the highest (or second highest) bid. This is accomplished through all participants 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 integer comparison circuits and perform the verification of the AND gates, inputs and outputs, without revealing their bids.

4. Security Basics

The security of the auction is based on several cryptographic techniques and primitives. These definitions can be commonly found in the cryptographic literature, but we have included brief descriptions here to foster a better understanding of the protocol. For a further discussion on these concepts see [42] .

Definition *Blum Integer*: An integer is a Blum integer if $N = pq$ where p, q are distinct primes congruent to 3 modulo 4.

Definition *Quadratic Residue*: Let N be an integer. The set of integers which are mutually prime with N is denoted by $\mathbb{Z}_N^*$. $a \in \mathbb{Z}_N^*$ is said to be a *quadratic residue* modulo n, if there exists an $x \in \mathbb{Z}_N^*$ such that $x^2 \equiv a \pmod{N}$. If no such x exists, then a is called a *quadratic non-residue modulo* N. The set of all quadratic residues modulo N is denoted by $\mathcal{Q}_N$ and the set of all quadratic non-residues is denoted by $\bar{\mathcal{Q}}_N$.

Definition *Jacobi symbol*: Let $N = pq$ be a Blum integer. The Jacobi symbol is defined by

$$J\left(\frac{x}{N}\right) = \begin{cases} 0 & \text{if } x \notin \mathbb{Z}_N^* \\ 1 & \text{if } x \in \mathbb{Z}_N^* \text{ and } (x^{(p-1)/2} \bmod p) = (x^{(q-1)/2} \bmod q) \\ -1 & \text{if } x \in \mathbb{Z}_N^* \text{ and } (x^{(p-1)/2} \bmod p) \neq (x^{(q-1)/2} \bmod q) \end{cases}$$

Using the law of quadratic reciprocity, the Jacobi symbol can be efficiently calculated without knowing the factorization of N (see any book on number theory, e.g., [2]). The QRA states that there exists no efficient algorithm that can guess, with better than negligible advantage, whether a random $x \in \mathbb{Z}_N^*$ with Jacobi symbol 1 is or is not a quadratic residue.

4.1. Bit-Commitment Protocol

A bit-commitment protocol consists of two stages:

- The *commit* stage: Alice (whose identity is defined by her public Blum Integer N_A) has a bit b to which she wishes to commit to Bob. She and Bob exchange messages. At the end of this stage Bob has some information that represents b. At this point, Bob cannot view the actual bit, but Alice can no longer change it.
- The *revealing* stage: at the end of which Bob knows b.

There are many types of bit-commitment protocols. In this work we use the QRA as the underlying mathematics. To commit to the n-tuple $B = (b_0, \ldots, b_{n-1})$, we proceed as follows:

- Commitment Stage:

  For each b_i Alice picks a random number $r_i \in \mathbb{Z}^*_{N_A}$. Alice sends to Bob the resultant commitment tuple, $C_B = (c_0, \ldots, c_{n-1})$, where

$$c_i = \begin{cases} r_i^2 \bmod N_A & \text{if} \quad b_i = 0 \\ -(r_i^2) \bmod N_A & \text{if} \quad b_i = 1 \end{cases}$$

  Therefore all bit-commitments satisfy:

$$b_i = \begin{cases} 0 & \text{if} \quad c_i \in \mathcal{Q}_{N_A} \\ 1 & \text{if} \quad c_i \notin \mathcal{Q}_{N_A} \end{cases}$$

- Reveal Stage:

  If Alice created the commitments herself, then she may store the values r_i. Revealing them opens the commitments. In our protocol, however, Alice will need to "open" commitments for which she does not have the $r_i's$. In this case, she can use the trapdoor information p, q to compute $r_i's$ as follows:

$$r_i = \begin{cases} \sqrt{c_i} & \text{if} \quad c_i = 0 \\ \sqrt{N_A - c_i} & \text{if} \quad c_i = 1 \end{cases}$$

  where $\sqrt{x} = x^{((p-1)(q-1)+4)/8} \bmod N_A$.

  Bob can then obtain the committed values $(b_0, \ldots, b_{n-1})$, where each b_i is calculated as follows:

$$b_i = \begin{cases} 0 & \text{if} \quad r_i^2 = c_i \bmod N_A \\ 1 & \text{if} \quad r_i^2 = -c_i \bmod N_A \\ error & \text{otherwise} \end{cases}$$

5. Practical Secure Multiparty Computation

Yao's Millionaires Problem [47] can be considered the starting point of Secure Multiparty Computation (SMC). Today SMC is still a very active field of research, although practical applications have been slow to appear. The application reported in [5, 11] 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 a set of players to compute target predicates on the private information of all without disclosing information not implied by the target predicates. The design of SMC must maintain 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").

We ensure privacy of losing bids by using several cryptographic tools. Initially, the bit-commitment protocol is used to commit to bids. The encrypted bids are then used as input, along with the selling price of the auction, to an integer comparison circuit. SMC is used to evaluate the output of the circuit without revealing the inputs (see section 5.1).

A homomorphic property of our chosen bit-commitment implementation makes it possible to spread encrypted values along linear components of the circuit without jeopardizing the privacy of the information (non-linear components require special treatment). Once committed values for all wires in the circuit are calculated, the discreet proofs defined in section 6 are used to keep everybody honest.

5.1. Circuit showing that a committed secret number is smaller than a given number

Given two n-bit numbers $X = (x_{n-1}, \ldots, x_1, x_0)$ and $S = (s_{n-1}, \ldots, s_1, s_0)$, we construct a circuit for the predicate $S \geq X$. There are many ways to compute this predicate. Because we use a bit-commitment primitive that is additively homomorphic, we design a circuit that contains as few AND gates as possible.

Let $A = 2^n + S - X$ with binary representation $(a_n, a_{n-1}, \ldots, a_1, a_0)$. Then a_n is 1 if and only if $S \geq X$. Now,

$$(S \geq X) \Leftrightarrow (X - S \leq 0) \Leftrightarrow (X + (2^n - 1) - S < 2^n) \Leftrightarrow (X + \bar{S} < 2^n).$$

I.e., $S \geq X$ if and only if addition of X and the complement of S yields a carry-in of 0 at the n^{th} position. A full-adder typically uses at least two

non-linear gates to compute the carry-in bit $c_k = Majority(\bar{s}_{k-1}, c_{k-1}, x_{k-1})$ at position k. The following recurrence uses only one non-linear gate:

$$\begin{aligned} c_k &= ((\bar{s}_{k-1} \oplus c_{k-1}) \wedge (x_{k-1} \oplus c_{k-1})) \oplus c_{k-1} \quad (0 < k \leq n) \\ c_0 &= 0 \end{aligned}$$

Thus, our integer comparison circuit uses n AND gates. It can be shown that this is optimal for two numbers that are in committed form.

5.2. Timeline of the Interaction with the Circuit

The comparison circuit defined in the previous section can be simplified after the selling price is determined. The timeline in figure 2 shows the main actions related to the comparison circuit. We now describe each event in the timeline.

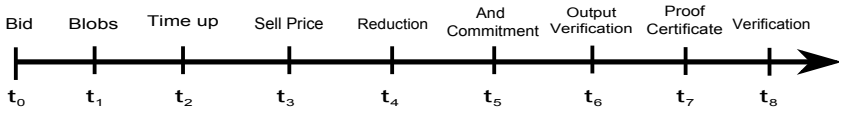

Figure 2: Timeline of the actions related to the comparison circuit

t_0 Bidders compute commitment values for their bids.

t_1 The bit-commitments computed at time t_0 are posted on the bulletin board.

t_2 The bidding phase ends.

t_3 The selling price S is established (the process is explained in section 3). The binary representation of S is assigned to the selling price inputs in the circuit.

t_4 The reduction process is performed (see section 5.3). At this point, the final configuration of each prover's comparison circuit is established. The resulting circuit is the same for all the bidders because the procedures and data used (sell price) to transform the circuit are public and they do not depend on the bidders' data.

t_5 Bidders compute and post commitment values for the outputs of AND gates in the circuit. The cryptosystem chosen has homomorphic properties in the sense that if c_0, c_1 are the encryptions of bits m_0, m_1, then $c_0 \cdot c_1 \bmod N$ will be an encryption of $m_0 \oplus m_1$, and $(-c_0) \bmod N$

will be an encryption of $\neg m_0$. This means that commitments to the inputs and outputs of all NOT and XOR gates can be calculated by any agent that has access to the bulletin board. Thus, all players can now compute commitments to inputs and outputs of all gates for all bidders. In particular, the commitments to the outputs of the circuits are now known by all.

t_6 Let the commitment to the output of prover A's circuit be X_A. If X_A encodes a 0 then A posts a square root of X_A modulo N_A. If it encodes a 1 then A posts a square root of $-X_A$ modulo N_A.

t_7 Bidders produce cryptographic proofs according to the system configuration and post the discreet proof certificates.

t_8 Any participant can choose to verify the correctness of the comparison function of any bidder using the circuit established at t_4, the commitments posted at t_5, and the proof certificate posted at t_7.

5.3. Reducing the comparison circuit

After the sell price has been determined, there are three types of gates in the circuit:

Indeterminate None of the input of the gate has a binary value assigned.

Assigned Both inputs of the gate have binary values assigned.

Hybrid In this case, one input of the circuit has a binary value assigned and the other one has no assigned binary value.

The indeterminate gates can not be modified. Only the assigned and hybrid gates are reduced. This process erases some gates, thereby simplifying the circuit. Elimination of AND gates also cause communication costs to be reduced.

For assigned gates, the binary output is set and the gate is eliminated from the circuit. Hybrid gates are processed as shown in figure 3. The reduction process requires no communication among participants. The resulting circuits are known to all.

6. Discreet Proofs

Discreet proofs were introduced in [6, 7]. They can be thought of as a type of zero-knowledge proof [19]. Denote by $\hat{x}$ the Boolean value of a commitment x. Given QRA commitments (a, b, c), a discreet proof shows

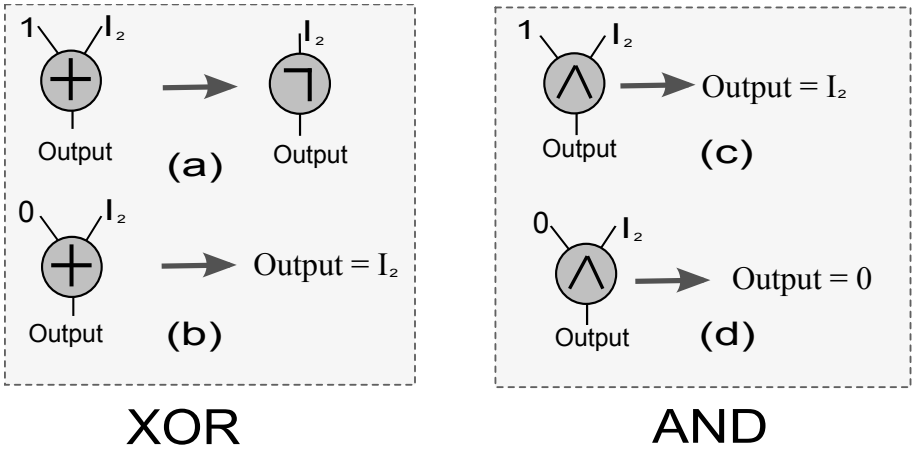

Figure 3: AND and XOR reductions

that $\hat{a}\ AND\ \hat{b} = \hat{c}$. This is done for each AND gate in the circuit. There are several ways to implement a discreet proof. A discussion of the importance of adding a distributed character to the proofs is included in this section. Several proof models are explained, including the advantages and drawbacks of each model in several application domains.

6.1. Discreet Proof Certificates

The gap between theoretical cryptography and software implementations is significant. Part of the reason for this is that the adoption of a new cryptographic primitive usually requires the creation of associated support mechanisms. Occasionally this is understood at the time the primitive is conceived, but the opposite is often the case.

Public-key cryptography is an example where the auxiliary mechanisms played a very important role in the implementation of the technology. Although the necessary mathematics was invented by the end of the 1970s, more than thirty years later there are still important deployment issues being debated. One of these issues relates to the mechanisms by which cryptographic keys are distributed and linked to users' identities. The development of public-key and identity certificates to address this issue gave birth to new research topics, such as trust management, and the proposed use of several models of certificates. Each model of a certificate has its own structure and language to determine how the values are stored.

Representation and transport mechanisms should be designed that enable the verification process to be ubiquitous, atemporal, and platform-independent. This would have an additional advantage: the transition from a simple prover-verifier system to a system involving full or partial delegation

of the verification step. This could be accomplished by the creation of a new type of certificate which we call *Proof Certificate*. A solution based on XML, analogous to identity certificates, would facilitate the integration of discreet proofs in software.

As of 2011, there are three commercial products involving variants of zero-knowledge proofs that are getting close to deployment. These are Microsoft's U-Prove, IBM's Identity Mixer, and Intel's EPID. Additionally, working groups two and five of ISO/IEC JTC1 SC27 are engaged in standardization work on anonymous authentication and group signatures [37]. The structure of our "proof certificates" should be compatible with these technologies.

6.2. Constructing commitments from public random numbers

In an implementation in which the public randomness service simply posts random numbers in a bulletin board (say, every minute). The communication cost of issuing commitments modulo N can be brought down to 1 bit per commitment provided the issuer knows the factorization of N. The method below is loosely based on the techniques appearing in [6, 7].

Denote by β the smallest positive integer such that $J(\beta/N) = -1$, where N is the prover's public key.[4] We denote by n the length of N in bits. The prover requests a time-stamped and certified string R of random bits from the Randomness Service. Next we explain how the prover can use R to construct a sequence of bit commitments simply by concatenating a string of bits to it (one bit per commitment).

Denote by u_i the number in Z_N defined by the i^{th} block of n bits in R. Let v_i be defined as follows: if $J(u_i/N) = 1$ then $v_i = u_i$, otherwise $v_i = u_i \cdot \beta \bmod N$. Thus the $v_i's$ are random numbers modulo N, all with Jacobi symbol 1. The prover uses each v_i to construct a commitment w_i to a chosen Boolean value z_i as follows: if v_i already commits to z_i then the Prover writes '0', in which case w_i is just v_i. Otherwise the prover writes '1', in which case w_i is defined as $(-z_i) \bmod N$.

6.3. Interactive proof for one AND gate

Recall the notation $\hat{x}$ for the Boolean value of a commitment x. Given QRA commitments (a, b, c), an interactive proof that $\hat{a}\ AND\ \hat{b} = \hat{c}$ is as

[4] Half the numbers modulo N have Jacobi symbol -1. If this was a theoretical paper we would be concerned with the fact that the smallest such number need not be polynomial in the size of N. This is not a practical concern here. If it was, then we could either invoke the Extended Riemann Hypothesis or simply make β part of the public key as published by the prover.

follows. Recall that, given two commitments x, y, a zero-knowledge proof that $\hat{x} = \hat{y}$ is simply a modular square root of $x \cdot y \bmod N$.

1. The prover sends, in random order, a set of three commitments $\{u, v, w\}$, two of which encode the same Boolean values as (a, b), and the third encodes the value 0. We call these commitments *auxiliary triples* for the AND gate.
2. The verifier challenges with either 1 or 0.
3. If the challenge is 0, the prover opens one of $\{u, v, w\}$ to show it is a 0, and then shows the correspondence between $\{a, b\}$ and the other two (e.g., if $(\hat{u}, \hat{v}, \hat{w}) = (\hat{b}, 0, \hat{a})$ the prover sends modular square roots of $u \cdot b$, v, and $w \cdot a$).
4. If the challenge is 1, the prover shows that two commitments from $\{u, v, w\}$ encode $\hat{c}$. (If all three commitments encode $\hat{c}$ then the prover selects any two.)

6.4. Non-interactive proof of circuit satisfiability

In section 6.2 we saw how public randomness can be used to construct commitments at a communication cost of one bit per commitment. For auctions, we require that each bidder commits to the bits of his/her bid. These bids are the inputs to the circuit defined in section 5.1. Additionally, provers must

- commit to the output value of each AND gate in the circuit;
- for each AND gate, commit to auxiliary triples, as defined in section 6.3. The number, per AND gate, of such triples is discussed in section 6.6.

Once the sale price has been determined, the losing bidders must show that their bid was below the sale price (i.e., that the output of the corresponding circuit is 1). This is done as follows:

- the circuit is reduced as explained in section 5.3;
- for each remaining AND gate, the proof of section 6.3, is performed as follows:
  - the auxiliary triples are read from the initial commitment phase;
  - the challenges are read (one challenge per auxiliary triple) from a fresh posting of random bits of the randomness server.

Thus the final proof requires no interaction from the verifier. All the information necessary to construct the proof certificates is on the public bulletin board. Verification of the proof certificates involves

- verification of time-stamps and signatures on the random numbers issued by the randomness server;
- verification of modular square roots as defined in section 6.3.

and possibly other entries of the proof certificates such as identity tokens, eligibility certificates, etc.

6.5. Amortization technique

The last step of the proof in section 6.3 is simply to post modular square roots of commitments (thereby proving they all encode 0). Thus, one can think of the techniques in sections 6.3 and 6.4 as a reduction of the assertion "the output of this circuit is 1" to the assertion "this set of commitments all encode 0".

A technique for proving that a vector of k commitments commits to all zeros is simply to take a random subset of the commitments, multiply them together, and disclose a square root of the product. If the vector contains one or more quadratic non-residues (i.e., one or more commitments are to 1) the probability that the product of a random subset is a non-residue is $1/2$. This probability is independent of k, and thus independent of the number of AND gates in the circuit.

We can now describe the complete proof we have implemented. The values k_1 and k_2 are security parameters.

1. The trusted randomness server posts random string R_1.
2. Bidders commit to their bids using R_1.
3. Sell price S is determined (this defines the reduced circuit C).
4. Each losing bidder uses R_1 to commit to
   - inputs and outputs of each AND gate in C;
   - k_1 groups of three auxiliary values for each AND gate in C;
5. Each losing bidder opens the commitment at the output of C (it must open to the value 1).
6. The trusted randomness server posts random string R_2.
7. R_2 is used as the challenge bits in the proof of section 6.3. However, no commitments are opened at this stage. Instead the proof is used to generate between 2ω and 3ω quadratic residues (i.e., commitments to 0), where ω is the number of AND gates in C. Denote by T_A the vector of quadratic residues generated by the losing bidder A in this step. Denote by λ_A the number of elements of T_A.

8. The trusted randomness server posts random string R_3. R_3 is used to construct a $k_2 \times \lambda$ binary matrix M, where $\lambda = \max_A \lambda_A$.
9. Each losing bidder A reveals modular square roots for each of the numbers in $M_A \cdot T_A \bmod N_A$, where M_A is a $k_2 \times \lambda_A$ submatrix of M.

In the above description, R_1, R_2, R_3 and C are common to all losing bidders. The vector T_A and modulus N_A are different for each losing bidder. The matrix M can be considered common to all bidders if we define it to be of dimension $k_2 \times 3k_1\omega$ and each bidder uses a submatrix for his/her proof certificate.

6.6. Security level

If a bid B_A is larger than the selling price S, a valid but incorrect proof certificate (that $B_A < S$) can be constructed only if

1. all elements of T_A commit to 0; or
2. all elements of $M_A \cdot T_A$ commit to 0 even though at least one element of T_A commits to 1.

The probability of the first event is at most $(1/2)^{k_1}$. The probability of the second event, is at most $(1/2)^{k_2}$. Thus the probability that a false proof certificate can be constructed is at most $(1/2)^{k_1} + (1/2)^{k_2}$. Letting $k_1 = k_2 = \alpha + 1$ yields an upper bound of $(1/2)^{\alpha}$.

7. Implementation Details

A prototype of our proposal has been implemented using Java technology. Open source Java libraries were utilized. These included the following: *Derby*[5] for SQL database support; *iText*[6] for storing the cryptographic information in a portable format document such as PDF; *jGraph*[7], which is a graphic library we used for drawing circuits; and *Barcode4J*[8], which we used to visually depict large numbers.

The system makes intense use of the Jacobi symbol calculation. Since the Jacobi symbol algorithm is not included in the standard Java distribution, we implemented different algorithms and compared their performance using the "big number" implementation of Java. The work [13] selects and compares three algorithms: Williams [46], Lesbegue [24] and Modified Binary

[5]http://db.apache.org/derby/
[6]http://www.lowagie.com/iText/
[7]http://www.jgraph.com/
[8]http://barcode4j.sourceforge.net/

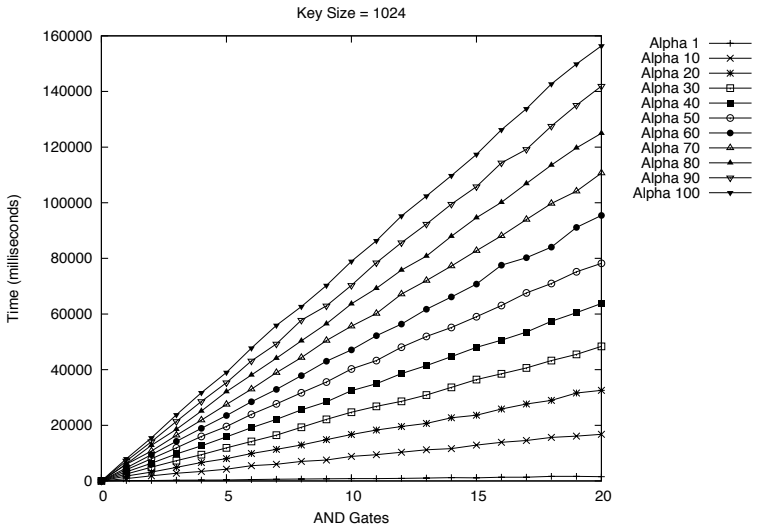

Figure 4: Time processing proofs using 1024 key length

[40]. Asymptotically, the Modified Binary method is the fastest of the three. However, the fastest algorithm in our implementation turned out to be the Williams method.

The main objective of our deployment phase was to show we could do this with low bandwidth and low computation time. Our measurements show that, for a security parameter of α around 10, proof creation was about 14 times more costly than proof verification (for a single AND gate). Several tests were developed to determine how computationally heavy the proof creation process is. Figure 4 shows computation time (as a function of the number of AND gates in the circuit and of the security parameter α) for a key size of 1024 bits. The depicted values are for an implementation that issued individual proofs for each AND gate (i.e., did not use the amortization method of section 6.5).

On the other hand, if we opt for using *Matrix Method* the computation times are reduced considerably. Figure 5 shows the obtained value in a test using a circuit with 20 gates, 1024 key-length and different values of α between an interval of 1 and 100 by steps of 10. If we focus on the more costly situation ($\alpha = 100$, and gates $= 20$), the regular method consumes 153,675 milliseconds whereas the matrix method only needs 38,954 milliseconds. Therefore the matrix method reduces the computation load by a factor of four.

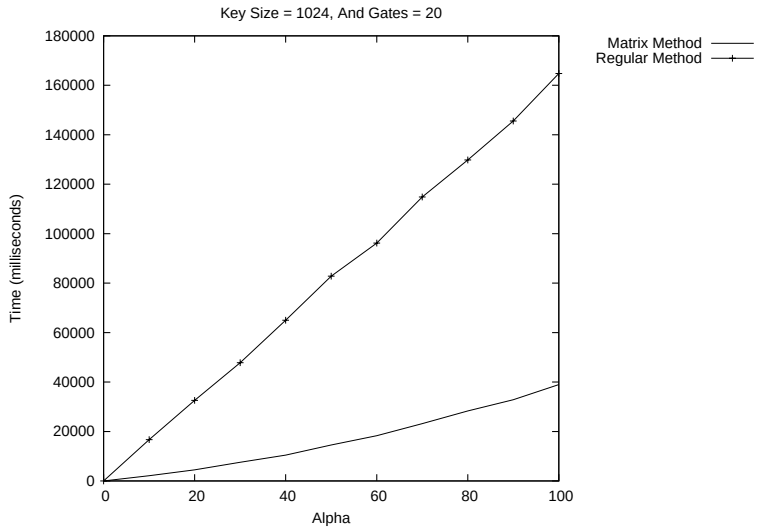

Figure 5: Time processing proofs using 1024 key length and Matrix Method

8. Conclusions

A secure sealed-bid auction was implemented using Java and a source of public randomness. Using techniques from secure multiparty computation, the winning bid can be found without opening the losing bids. At the end of the auction, each bidder produces a proof certificate that can be published and non-interactively verified using the bidder's public key. The proof certificates are non-forgeable and allow all the participants to verify the correctness of the auction results.

Figures 6 and 7 illustrate the user interface of the auction software. The client was developed using Applet technology and it can be executed on any enabled browser. Two server prototypes were deployed in addition to the auctioneer server: a Key Repository for managing identities and the Randomness Server.

9. Acknowledgments

We are grateful to the anonymous referees for helpful comments and to Gina Gallegos García for catching some errors during a careful reading of the final manuscript.

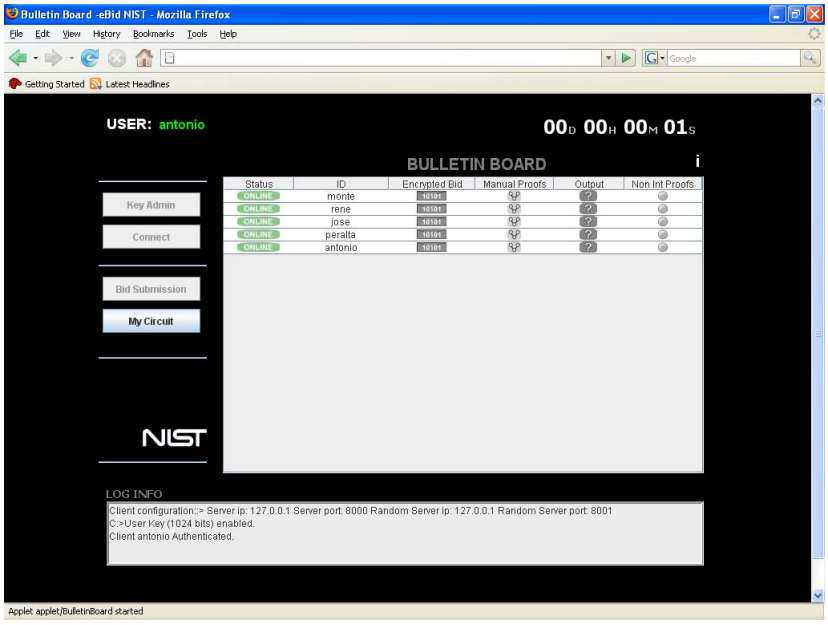

Figure 6: Screenshot of the Bidder Applet

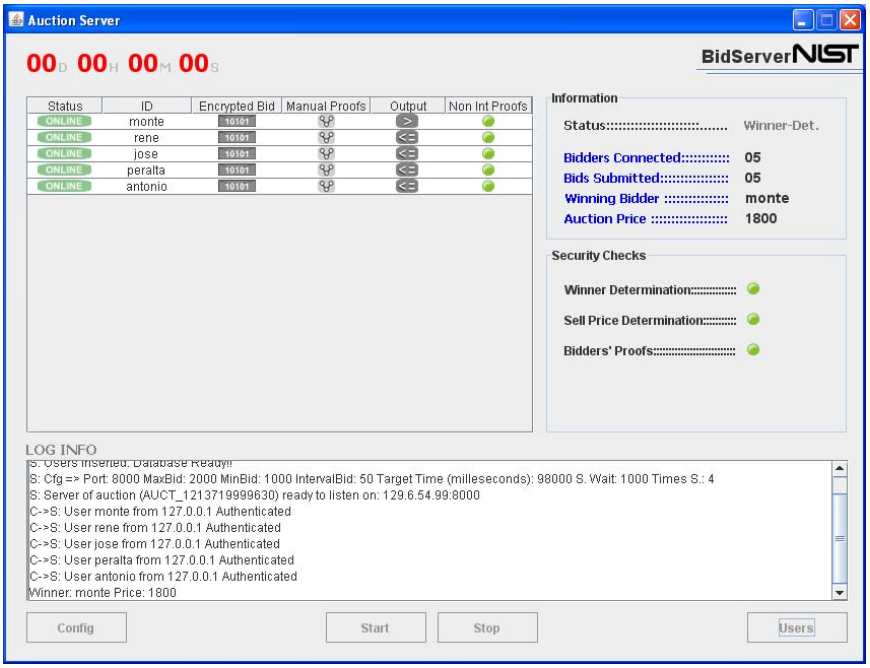

Figure 7: Screenshot of the Auctioneer Server

[1] M. Abe and K. Suzuki. M+1-st price auction using homomorphic encryption. In *Proceedings of the 5th International Workshop on Practice*

and Theory in Public Key Cryptosystems: Public Key Cryptography, PKC '02, pages 115–124, London, UK, 2002. Springer-Verlag.

[2] E. Bach and J. Shallit. *Algorithmic number theory*, volume 1. MIT Press, 1996.

[3] E. Barker and J. Kelsey. Recommendation for random number generation using deterministic random bit generators. *National Institute of Standards and Technology, Special Publications*, (NIST SP 800-90), 2007.

[4] O. Baudron and J. Stern. Non-interactive private auctions. In *Proceedings of the 5th International Conference on Financial Cryptography*, FC '01, pages 364–378, London, UK, UK, 2002. Springer-Verlag.

[5] P. Bogetoft, D. L. Christensen, I. Damgard, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Multiparty computation goes live. Cryptology ePrint Archive, Report 2008/068, 2008. `http://eprint.iacr.org/`.

[6] J. Boyar, I. Damgård, and R. Peralta. Short non-interactive cryptographic proofs. *Journal of Cryptology*, 13:449–472, 2000.

[7] J. Boyar and R. Peralta. Short discreet proofs. In *Advances in Cryptology - EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 131–142. Springer-Verlag, 1996.

[8] C. Boyd and W. Mao. Security issues for electronic auctions. Technical Report HPL-2000-90, HP labs, http://www.hpl.hp.com/techreports/2000/HPL-2000-90.pdf, May 2000.

[9] F. Brandt. How to obtain full privacy in auctions. *Int. J. Inf. Secur.*, 5:201–216, September 2006.

[10] K. Chen. Authenticated encryption scheme based on quadratic residue. *IEEE Electronics Letters*, 34(22):2115–2116, 1998.

[11] I. Damgard and T. Toft. Trading sugar beet quotas - secure multiparty computation in practice. *Ercim News*, 73:32–33, 2008.

[12] D. Eastlake, S. Crocker, and J. Schiller. *Randomness Recommendations for Security*. IETF Network Working Group, RFC 4086, June 2005.

[13] S. M. Eikenberry and J. P. Sorenson. Efficient algorithms for computing the Jacobi symbol. *Journal of Symbolic Computation*, 26(1):509–523, 1998.

[14] T. Erl. *Service-Oriented Architecture (SOA): Concepts, Technology and Design*. Prentice Hall, 2005.

[15] M. J. Fischer, M. Iorga, and R. Peralta. A public randomness service. In *SECRYPT'11*, 2011. to appear.

[16] M. Franklin and M. Reiter. The design and implementation of a secure auction service. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 302–312, May 1995.

[17] R. Gennaro. Randomness in cryptography. *IEEE Security & Privacy*, 4(2):64 – 67, March 2006.

[18] I. Goldberg and D. Wagner. Randomness and the netscape browser. *Dr. Dobb's Journal*, pages 66–70, January 1996.

[19] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186 – 208, 1989.

[20] M. Harkavy, J. D. Tygar, and H. Kikuchi. Electronic auctions with private bids. In *Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3*, WOEC'98, pages 61–74. USENIX Association, 1998.

[21] A. Juels and M. Szydlo. A two-server sealed-bid auction protocol. In *Financial Cryptography*, volume 2357, pages 72–86, March 2002.

[22] J. Kelsey, B. Schneier, D. Wagner, and C. Hall. Cryptanalytic attacks on pseudorandom number generators. In *Fifth International Fast Software Encryption*, pages 168–188, March 1998.

[23] M. Kumar and S. Feldman. Internet auctions. In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, page 4960, September 1998.

[24] V.A. Lebesgue. Sur le symbole (a/b) et quelques unes de ses applications. *J. Math. Pures Appl*, 12(1):497–517, 1847.

[25] C. Lee, P. Ho, and M. Hwang. A secure e-auction scheme based on group signatures. *Information Systems Frontiers*, 2008.

[26] Hi. Lipmaa, N. Asokan, and V. Niemi. Secure Vickrey auctions without threshold trust. In Matt Blaze, editor, *Financial Cryptography*, volume 2357 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2002.

[27] P. Milgrom. Auctions and bidding: A primer. *The Journal of Economic Perspectives*, 3(3):3–22, 1989.

[28] D. L. Mills. *Network Time Protocol (Version 3) Specification, Implementation and Analysis*. IETF Network Working Group, RFC 1305, May 1992.

[29] M. Naor, B. Pinkas, and R. Summer. Privacy preserving auctions and mechanism design. In *1st ACM Conference on Electronic Commerce*, pages 129–139, 1999.

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

[31] David C. Parkes, Michael O. Rabin, Stuart M. Shieber, and Christopher Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electron. Commer. Rec. Appl.*, 7:294–312, November 2008.

[32] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan. Five sealed bid auction models. In *Australian Information Security WorkShop, AISW*, pages 77 – 86, 2003.

[33] J. Postel. *Daytime Protocol*. IETF Network Working Group, RFC 867, May 1983.

[34] J. Postel and K. Harrenstien. *Time Protocol*. IETF Network Working Group, RFC 868, May 1983.

[35] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2001. Revised 2010 by L. Bassham.

[36] K. Sako. An auction protocol which hides bids of losers. In *Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, pages 422–432, London, UK, 2000. Springer-Verlag.

[37] Kazue Sako. personal communication, 2011.

[38] K. Sakurai and S. Miyazaki. An anonymous electronic bidding protocol based on a new convertible group signature scheme. In *Proceedings of the 5th Australasian Conference on Information Security and Privacy*, pages 385–399, London, UK, 2000. Springer-Verlag.

[39] C. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[40] J. Shallit and J. Sorenson. A binary algorithm for the Jacobi symbol. *SIGSAM Bulletin*, 27(1):4 – 11, 1993.

[41] I. Shparlinski, W. D. Banks, and D. Lieman. An extremely small and efficient identification scheme. In *Proc. 5th Aust. Conf. on Information Security and Privacy*, volume 1841, pages 378–384, August 2000.

[42] D. Stinson. *Cryptography Theory and Practice*. CRC, November 2005.

[43] K. Suzuki, K. Kobayashi, and H. Morita. Efficient sealed-bid auction using hash chain. In *Proceedings of the Third International Conference on Information Security and Cryptology*, ICISC '00, pages 183–191, London, UK, UK, 2001. Springer-Verlag.

[44] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8 – 37, 1961.

[45] M. Wellman and P. Wurman. Real time issues for internet auctions. In *1st IEEE Workshop on Dependable and Real Time E-commerce System (DARE)*, pages 54–56, June 1998.

[46] H. Williams. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory*, 26(6):726–729, 1980.

[47] A. Yao. Protocols for secure computations. In *Symposium on Foundations of Computer Science*, pages 160–164, November 1982.

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