Secure Computation for Combinatorial Auctions and Market Exchanges

Josiane Nzouonta[†] and Marius-Călin Silaghi[†] and Makoto Yokoo[‡]
[†]Florida Institute of Technology
[‡]NTT Communication Science Laboratories, Japan

Abstract

It was recently shown possible to solve $(M+1)^{st}$ price single item auctions without revealing absolutely any secret except for the solution. Namely, with vMB-share [2], the seller and the buyer only learn each other's identity and learn the selling price for a chosen $(M+1)^{st}$ pricing scheme. No trusted party is necessary. In this paper we show how vMB-share can be extended for the clearing of combinatorial negotiation problems with several items, buyers and sellers. We first show how the more general problem can be reduced to a virtual form, form that is relatively similar to the single item auctions, by having a virtual bidder for each candidate allocation. Then, some modifications in the cryptographic techniques of vMB-share are made such that it can offer a solution to problems in virtual form. As explained in the paper, it is expected that a secure solution hiding details that can be inferred from the running time will have an exponential computation cost. Our preliminary experimental evaluation shows that some small negotiations can nevertheless be solved with acceptable effort.

1. Introduction and Background

Most negotiations are made complex mainly by privacy and security concerns. Multiparty Computation (MPC) protocols are among the best candidates for approaching these problems. In [2], vMB-Share, a novel approach that resists to collusion of any subset of agents and does not require a third-party as auctioneer is presented. Distributing the trust onto the bidders presents a definitive advantage over previously proposed methods because the possibility of a judge divulging secrets can never be completely ruled out.

In this paper, we exploit secure simulations of arithmetic circuit evaluations based on the Shamir's secret sharing scheme [1] to ensure bidders privacy. We describe a technique that allows the clearing of markets with multiple buyers and sellers, and offers an important degree of privacy. It is an extension of vMB-share to combinatorial auctions and market exchanges. The concept of *virtual auction form* proposed here allows for using vMB-share by adding an abstraction: namely, every candidate allocation is represented by a virtual participant shared among real ones. The results of the experiments presented here show that the method can be successfully used for small sized problems.

procedure *market exchange* **do**
- Create table with candidate allocations;
- Select bids (reservation prices) and send shares;
- Sum shares and create differential bid vector;
- Run modified vMB-share with M=0;
- Hide non-null elements by multiplication with random non-null numbers;
- Send shares to other agents (deciding the allocation);

Algorithm 1: Algorithm finding the winning allocation

2. Secure Market Exchanges

Assume agents $A_1, ..., A_n$ negotiate over N items. They agree on a set of candidate allocations of the items among themselves. A candidate allocation is represented by a tuple c_i specifying the agents that own each item after the negotiation. c_i=$(A_{i_1}, A_{i_2}, ..., A_{i_N})$ where $c_i[k]$=A_{i_k} means that agent A_{i_k} remains with product k. The algorithm consists of securely distributing shares of each agent's bid for each candidate allocation, using Shamir's scheme. The bid of an agent for allocation c_i can be made of her bid for the items she gets minus the reservation prices for the items she sells. Each agent can add a constant to all her bids, such that the minimal bid is non-negative. The sum of all the bids for the allocation c_i is denoted s_{c_i}. Each agent obtains a share of s_{c_i} by summing her shares of all the bids for c_i. Each clearing alternative becomes a *virtual* agent when applying vMB-share for determining the highest value from the summed bids over all allocations. We say that the negotiation is represented in its *virtual form*. The steps are shown in Algorithm 1. A differential bid vector for an allocation c_i and a bid with value b, and $\mathcal{K}$ possible bid values, is: $V_{c_i} = \langle \underbrace{0, ..., 0, \overbrace{1}^{b}, 0, ..., 0}_{\mathcal{K}} \rangle$. After the *virtual bid* s_{c_i}

for each candidate allocation c_i is computed as just mentioned, it is transformed into a differential bid vector. The formula used for the transformation is

$$V_{c_i}[j] = \frac{\prod_{k=0,k\neq j}^{\mathcal{K}}(s_{c_i} - k)}{\prod_{k=0,k\neq j}^{\mathcal{K}}(j-k)}$$

Each agent A_i knows only her share $s_{c_i}^{A_i}$. The vMB-share technique is adapted to solve negotiations described in the *virtual form*. Several modifications were required:

- The computations can no longer be done in $\mathbb{Z}$, due to the existence of multiplications. One can use either modular arithmetic or rational numbers. In our experiments we used modular arithmetic.
- The technique used by vMB-share to hide non-null secret numbers by multiplying them with random numbers is not applicable with modular arithmetic. We show how an alternative can be constructed, based on multiplication of nun-null random secrets generated by participants.
- A new technique is needed to reveal results only to involved bidders. Our solution consists in revealing each element i of each vector returned by VMB-share, only to the participants involved in the transaction defined by the allocation of the corresponding *virtual bid*, c_i.

If Clarke tax mechanism is desired, the actual amount to be paid/received by an agent has to be computed for each subset of $n-1$ agents (as done in [5]).

A more exemplified description appears in [4, 3]. [5] gives a version for Generalized Vickrey Auctions, based on homomorphic encryption and highlights the opportunity of allocative externality, i.e., a bidder might care about what other bidders get.

3. Analysis

Messages send/receive: The number of computation rounds (in a round several messages can be exchanged simultaneously) is a measure of the efficiency of the algorithm. The number of rounds is a factor of the number of agents n, the number of items and the total number of prices $\mathcal{K}$. More precisely, without the extensive parallelization in [1],

$Number\ rounds = n^N * \mathcal{K} * (\mathcal{K}-1) + 3$

n represents the number of agents, N represents the number of items negotiated in the market and $\mathcal{K}$ is the total number of possible bids for each allocation.

The general combinatorial exchanges winner allocation determination problem with privacy requirements is not in NP (its decision equivalent without privacy is NP-complete). Given an optimum allocation, it is impossible to verify its correctness and optimality otherwise then constructively with a protocol like ours, due to the privacy requirements. The combinatorial exchanges with privacy requirements are NP-hard, as it follows by reduction from weighted CSPs.

A secure computation planning to avoid that the computation time reveals anything, must have a computation time that is independent of the problem. A non-exponential cost secure solution would be a proof that NP=P.

As explained in [1], the multiplication of secrets requires a ($\lceil n/2 \rceil$, n)-threshold scheme. The method offers $\lfloor n/2 \rfloor$-privacy.

Experimentations: We ran a simulation of our algorithm on a SunOS 5.8, 2 Gb RAM machine with different number of participants and different number of items, keeping a constant number of prices $K = 5$. The results are summarized in Table 1.

3 participants, 3 items; combinations = 27.	
Time Run:	The negotiation was completed in an average of 1min33s clock time. *Experiment 1: 1min31s; Experiment 2: 1min34s; Experiment 3: 1min33s*
Number of rounds:	We had a total number of 545 rounds of message exchange among participants. Identical in all experiments runs.
3 participants 5 items combinations = 243.	
Time Run:	The negotiation was completed in around 2min20s. *Experiment 1: 2min24s; Experiment 2: 2min16s; Experiment 3: 2min24s*
Number of rounds:	We had a total number of 4865 rounds of message exchange among participants. Identical in all runs of experiments.
5 participants; 5 items; combinations = 3125.	
Time Run:	The negotiation was completed in around 1hour00 min 09s.
Number of rounds:	We had a total number of 62507 rounds of message exchange among participants.

Table 1. Experimental Results

References

[1] M. Ben-Or, S. Goldwasser, and A. Widgerson. Completeness theorems for non-cryptographic fault-tolerant distributed computating. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 1–10, 1988.

[2] F. Brandt. A verifiable, bidder-resolved auction protocol. In L. R.Falcone, editor, *Deception, Fraud and Trust in Agent Societies (AAMAS-W5)*, pages 18,25, Bologna, July 15 2002.

[3] J. Nzouonta. An algorithm for clearing combinatorial markets. Master Thesis CS-2003-23, Florida Institute of Technology, 2003. http://www.cs.fit.edu/ tr/tr2003.html.

[4] M. C. Silaghi. An algorithm applicable to clearing combinatorial exchanges. Technical Report CS-2002-14, Florida Institute of Technology, 2002. http://www.cs.fit.edu/ tr/tr2002.html.

[5] M. Yokoo and K. Suzuki. Secure generalized vickrey auction using homomorphic encryption. In *Financial Cryptography*, 2003.