CERIAS Tech Report 2003-13

SECURE SUPPLY-CHAIN PROTOCOLS

by Mikhail J. Atallah, Hicham G. Elmongui,
Vinayak Deshpande, and Leroy B. Schwarz

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907

Secure Supply-Chain Protocols *

Mikhail J. Atallah
CERIAS and Computer Sciences
Purdue University
mja@cs.purdue.edu

Hicham G. Elmongui
CERIAS and Computer Sciences
Purdue University
elmongui@cs.purdue.edu

Vinayak Deshpande
Krannert School of Management
Purdue University
deshpandev@mgmt.purdue.edu

Leroy B. Schwarz
Krannert School of Management
Purdue University
lee@mgmt.purdue.edu

Abstract

Supply chain interactions have huge economic importance, yet these interactions are managed inefficiently. One of the major sources of inefficiency in supply-chain management is information asymmetry; i.e., information that is available to one or more organizations in the chain (e.g., manufacturer, retailer) is not available to others. There are several causes of information asymmetry, among them fear that a powerful buyer or supplier will take advantage of private information, that information will leak to a competitor, etc. We propose Secure Supply-Chain Collaboration (SSCC) protocols that enable supply-chain partners to cooperatively achieve desired system-wide goals without revealing the private information of any of the parties, even though the jointly-computed decisions require the information of all the parties. Secure supply-chain collaboration has the potential to improve supply-chain management practice, and, by removing one major inefficiency therein, improve productivity. We present specific SSCC protocols for two types of supply-chain interactions: Capacity allocation, and e-auctions. In the course of doing so, we design techniques that are of independent interest, and are likely to be useful in the design of future SSCC protocols.

Keywords: *Supply-chain online interactions, privacy, security, secure multi-party computation, capacity allocation in e-commerce, e-auctions.*

*Portions of this work were supported by Grants EIA-9903545 and ISS-0219560 from the National Science Foundation, Contract N00014-02-1-0364 from the Office of Naval Research, by sponsors of the Center for Education and Research in Information Assurance and Security, and by Purdue Discovery Park's e-enterprise Center.

1. Introduction

Information asymmetry is known to create inefficiencies in managing supply chains, among them under-investment in capacity leading to shortages; misallocation of inventory, transportation, and management resources; increased prices; and reduced customer service. It can also lead to increased use of premium shipping, increased penalties resulting from line shut-downs, and lost future business contracts. Unfortunately, the barriers to information-sharing are significant, among them fear that information voluntarily shared with a partner will be used against the volunteer, fear that sensitive information will leak to a competitor, government regulations about information-sharing, etc. Further, if one of the parties is government, then there are national security reasons to protect secret information.

These barriers can be overcome if traditional methods for information-sharing are replaced by Secure Supply-Chain Collaboration (SSCC) protocols, which would enable supply-chain partners to cooperatively achieve desired system-wide goals without revealing any private information, even though the jointly-computed decisions require this information. The contributions of this paper are (i) to present such protocols for two classes of supply-chain interactions (capacity allocation under various policies, and bidding and auctions under both discriminatory and nondiscriminatory pricing), and (ii) to give techniques and building blocks that are likely to be useful in the design of future SSCC protocols.

2. Related work

Because this paper relates to both protocols and supply chains, we need to review related work from both.

2.1. Related Cryptographic techniques

The closest area of cryptography relevant to our work is secure multiparty computation [28]. Secure multi-party protocols are a form of cooperative distributed computing that preserves the privacy of the participants' data. This general class of computations typically takes the following form between two parties: "Alice" and "Bob" each have a private input (say, x_A for Alice and x_B for Bob), and they want to compute $f(x_A, x_B)$ where the efficiently computable function f is known to both Alice and Bob However, neither side is willing to disclose his/her private data to the other party, or even to any third party. A protocol that involves only Alice and Bob, is said to be secure if, at its end, Alice and Bob know only $f(x_A, x_B)$ (and their respective inputs, of course). Of course, Alice might infer something about x_B from her knowledge of x_A, f, and $f(x_A, x_B)$, but that is unavoidable.

Goldreich states in [10] that although the general secure multi-party computation problem is solvable in theory, using the solutions derived by these general results for special cases can be impractical. In other words, efficiency dictates the development of special solutions for special cases for efficiency reasons. In addition, as noted above, the characteristics of supply-chain settings necessitate the development of such solutions. For example, whereas in most secure multiparty settings all the parties know the function f they are cooperatively computing, in our case each party (e.g., Alice) is computing her own individual function f_A that is just as important to keep hidden from the other participants as her private data x_A. Another complication is that the decision computed by a participant (e.g., Alice) can depend not only on the other participants' private data ($x_B, x_C, x_D, \ldots$), but also on their private functions $f_B, f_C, f_D, \ldots$ as well. To see how this can happen, simply consider the case of a multi-party supply chain negotiation where each party (say, Bob) can "drop out" of the negotiation depending on the value of $f_B(x_A, x_B, x_C, \ldots)$. The theoretical general secure multiparty computation techniques can be modified to handle this, but the resulting methods are even more impractical than the above-mentioned ones for the case when all sides are cooperatively computing the same function.

In Selective Private Function Evaluation (SPFE) [22], a client interacts with one or more servers holding copies of a database $x = x_1, \ldots, x_n$ in order to compute $f(x_{i_1}, \ldots, x_{i_m})$, for some function f and indices $i = i_1, \ldots, i_m$ chosen by the client. Ideally, the client must learn nothing more about the database than $f(x_{i_1}, \ldots, x_{i_m})$, and the servers should learn nothing. The requirement that the server not know f in SPFE is similar to our requirement that f_A not be known to any of the other participants, although we have not used SPFE techniques in our protocols.

2.2. Supply-chain background

Historically, supply-chain management research has focused on "centralized" policies; i.e., decision-rules for optimizing a single objective function (e.g., system profit) under the assumption that all the information about the system (e.g., costs, capacity, inventory status) is available to a central planner. In mathematical terms, supply-chain research has historically focused on problems of the form optimize $f(x)$, where the input vector x is centrally available and a single decision-maker optimizes this function. See [21] and [8], for examples of this research. Although this literature has contributed decision-rules for managing supply chains that employ centralized information and control, in fact, most real-world supply chains are managed, not by a single decision-maker, but by several decision-makers, each with their own, often incompatible, objective functions, and each using her/his own proprietary information.

Today, research in supply-chain management is largely focused on multiple decision-makers with multiple objective functions, each formulating their own decision-rules on the basis of "asymmetric information" (i.e., information available to any given decision-maker is not necessarily available to any other decision-makers). In mathematical terms, this stream of research splits the traditional objective function $f(x)$ into separate objective functions $f_A(x_A, x_B)$ and $f_B(x_A, x_B)$ for Alice and Bob respectively, based on private inputs x_A and x_B. The intellectual roots of this new focus on decentralized supply-chains is auctions and other information-asymmetry models in economic game theory. In these information-asymmetry models, without loss of generality, Alice typically acts as a leader and provides incentives to Bob to "reveal" his private input x_B, in addition to participation constraints on f_B. Alice then takes action based on Bob's data.

There is also a national program underway, sponsored by the Voluntary Intraindustry Commerce Standards (VICS) association to develop standards and procedures under which independent buyers and sellers can share plans, forecasts, and decision-making involving inventory replenishment. This program, called Collaborative Planning, Forecasting, and Replenishment (CPFR) has attracted the interest of literally hundreds of companies (http://www.cpfr.org/Members.html). Unfortunately, CPFR must overcome at least one major obstacle in order to achieve success for buyer and sellers: the reluctance of either/both to share private, proprietary information.

Several researchers have examined the value of information-sharing in a supply-chain. Iyer and Ye [12], for example, assess the value of information sharing in a retail environment, where retailers share promotion infor-

mation with their suppliers. Song and Zipkin [25] develop an inventory-replenishment policy to take advantage of information about supply conditions. Cachon and Fisher [4] study the value of sharing demand and inventory level information in a supply chain. More recently, Aviv [2, 3] has examined the effect of collaborative forecasting on supply-chain performance. This work, the literature on centralized decison-making, and the agency loss associated with decentralized decision-making, provide the supply-chain motivation and foundation for our work.

3. Capacity Allocation in E-Commerce

Consider a single supplier and N retailers. The supplier has a constant marginal production cost, but limited capacity, K. The retailers operate in non-competing retail markets, each with a linear demand curve: $q = \theta - p$, where q is market demand, p is the retail price, and θ is the market potential (i.e., what demand will be if price equaled zero). Everyone, including the supplier, knows the form of the retailer's demand curve, but each retailer's θ is its private information.

In order to maximize its profits, each retailer wants to maximize its revenue, pq. Hence, retailer i (defined to have $\theta = \theta_i$) would like to order (and sell) $q = \theta_i/2$ units, and at price $\theta_i/2$ per unit, and receive revenue $\theta_i^2/4$. Note that $\theta_i/2$ is the maximum transfer price/unit retailer i is willing to pay the supplier; i.e., a higher transfer price will yield a loss; a lower price provides a profit; and, the lower the transfer price, the higher retailer i's profit.

If the supplier has unlimited capacity and knows each retailer's θ, then, in order to maximize its profits, the supplier will charge retailer i a transfer price $\theta_i/2$ per unit, thereby sucking up all of the profits in the system. The supplier is able to do so because she knows that the retailer is making a profit at a lower price, since she knows $\theta_i/2$. If the supplier has limited capacity, K, and knows every retailer's θ, then she will allocate the capacity, K, such that the marginal revenues are equal across all retailers. For example, if there exist two retailers with parameters θ_1 and θ_2 and capacity is tight, then the retailer will allocate the capacity such that $\theta_1 - 2q_1 = \theta_2 - 2q_2$. And, again, because the supplier knows every retailer's θ, the supplier will be able to price these units so that she captures all the retailers' profits.

Now suppose that all the supplier has is a probability distribution on each retailer's θ. The supplier's goal is the same; i.e., to maximize its profit, but now has to do so in light of uncertainty about each retailer's θ. There are two consequences of the corresponding misallocation: (1) total profit in the supply chain decreases below its maximum possible value; and (2) some retailers get some positive profit.

In [7], Deshpande and Schwarz designed an incentive compatible pricing and allocation scheme to get every retailer to reveal its θ, thereby allowing the retailer to maximize its profit without forcing any retailer to experience a loss. Moreover, they show that the optimal allocation policy, $Q(\theta_1, \theta_2, \ldots, \theta_N)$, for the supplier is to equalize the information rent-adjusted marginal revenues across all retailers. They were able to establish the following optimal allocation rule:

Theorem 1 *If retailers face deterministic downward sloping linear demand, with the intercept of the demand-curve θ private to the retailers, then the linear allocation mechanism (defined below) is optimal for the supplier.*

Definition 3.1 (Linear Allocation) *Index the retailers in decreasing order of their quantities; i.e., $q_1 \geq q_2 \geq \ldots \geq q_N$. Retailer i is allocated $Q(q_i, \bar{n})$ where $Q(q_i, \bar{n}) = q_i - \frac{1}{\bar{n}} \max\left(0, \sum_1^{\bar{n}} q_i - K\right)$ if $i \leq \bar{n}$ and zero otherwise. Here $\bar{n}$ is the number of the retailers who will actually buy, K is the total capacity that the supplier can provide, and $Q(q_i, \bar{n}) \geq 0$ for all $i \leq \bar{n}$. Note that the linear allocation scheme should be jointly implemented with the optimal pricing scheme, as designed in [7], for the scheme to be incentive compatible.*

Intuitively, linear allocation is simply an "equal sharing of the pain" among the buyers, with the understanding that if that pain exceeds the q_i of a buyer then that buyer drops out. This is why $\bar{n}$, the number of buyers who do not drop out, can be less than the number n of initial buyers. If there are $\bar{n}$ actual buyers, then they each get the *same* amount less than their order, i.e., the "pain" inflicted on each buyer is equal to (total shortage)/$\bar{n}$ where the total shortage equals what the $\bar{n}$ buyers would have wanted minus K. (Note that the total shortage is *not* $q_1 + \cdots + q_N - K$.)

Deshpande and Schwarz also prove the structure of the optimal policy for the supplier if the retailers are "newsvendors", i.e., retailers, like real newsvendors, face demand generated from a probability distribution.

Theorem 2 *If retailers are newsvendors with a normal demand distribution with mean θ, and an exponential prior on θ, then the linear allocation mechanism is optimal for the supplier.*

Theorem 3 *If retailers are newsvendors with a uniform demand distribution on $[0, \theta]$, and a Pareto supplier's prior on θ, then the proportional allocation mechanism (defined below) is optimal for the supplier.*

Definition 3.2 (Proportional Allocation) *Retailer i is allocated $Q(q_i, N)$ where $Q(q_i, N) = \frac{K q_i}{\sum_1^N q_i}$. Here N is the number of the retailers, and K is the total capacity that the supplier can provide.*

Note that in proportional allocation every retailer is allocated a positive quantity, and therefore $\bar{n} = N$.

The optimal allocation mechanisms derived above are based on the revelation principle, which states that the supplier can induce the retailers to truthfully reveal their order quantities q_i, thus indirectly revealing their private information parameter θ_i to the supplier.

3.1. Information Required for SSCC

The allocation mechanism described above and its corresponding pricing policy described in [7] might be appropriate if allocation decisions are made once and only once. However, if allocation decisions are repeated, say, weekly over a selling season of several months, then there would be no incentive for the retailers to participate after the first allocation, since, after that, they would make no profits, because their θ's would have been revealed as part of the first allocation process. (Note that the linear allocation mechanism requires each retailer to reveal its true order quantity q_i based on its information parameter θ_i.)

In the next subsection, we sketch secure protocols for the above allocation mechanisms. These protocols for allocation use the retailer order quantities $q_i, i = 1, \ldots, N$ as inputs and compute the allocation, $Q(q_1, q_2, \ldots, q_N)$ defined above, without revealing any retailer's private information parameter θ_i to either to the supplier or to the other retailers. Since these protocols do not reveal the individual retailers' private information parameter, these protocols can be used repeatedly, unlike the auction mechanism described in [7].

Before the protocol every retailer knows his quantity q_i, the supplier knows her capacity K. After the protocol is completed, every retailer knows the actual quantity $Q(q_i, \bar{n})$ she would be allocated under the allocation policy (whether linear or proportional), $\bar{n}$, and nothing else (other than what she can infer from $Q(q_i, \bar{n})$, which is unavoidable). The protocol itself does not reveal the individual q_i or $\sum_i q_i$.

3.2. Secure Information Protocols

We present the protocols corresponding to both capacity allocation models. The details of some of the used building blocks come in later sections. In the versions of the protocols presented in this document, we assume by default that the participating parties are *honest-but-curious*, i.e. they will follow the protocol, but while doing so they could nevertheless try to illegally compute information about the other party's secret data. However, we often give protocols that can handle dishonest behavior that is worse than the honest-but-curious (participants who do not follow the protocol, or who collude with some of the participants against other participants). Finally, we treat prices and quantities as essentially continuous, so the protocols (in their current form) are not appropriate for interactions about small numbers of "widgets" (like large ships or aircraft, where rounding to within 1 unit is significant). We believe our protocols can be modified to handle such cases as well, but we have not yet looked at the details of these modifications.

Linear Allocation Protocol

1. Every retailer initially marks himself as "active" (some will mark themselves as "passive" as the protocol proceeds). We use $\mathcal{P}$ to denote the set of active retailers . We use $\bar{n}$ to denote $|\mathcal{P}|$.

2. Repeat the following substeps (a)–(d) until $\bar{n}$ ceases to change from one iteration to the next:

   (a) Every retailer i generates a random R_i. Let $R = \sum_{i=1}^{N} R_i$; note that no single party knowns R.

   (b) Using a secure simultaneous summation protocol (discussed later), the participants cooperatively compute both $\bar{n}$ and $D = (\sum_{i \in \mathcal{P}} q_i) - K + R$ in such a way that $\bar{n}$ is known to all participants but D is known only to the supplier.

   (c) If the computed $\bar{n}$ is the same as it was in the previous iteration of these substeps (a)–(d) then the protocol moves to Step 3 below, otherwise it continues with the next substep (d).

   (d) The participants run a secure simultaneous summation protocol in which the supplier's item (used in the summation) is $D/\bar{n}$, and every retailer i's item is $R_i/\bar{n}$, such that the answer to the summation is known to the retailers but not to the supplier. All the retailers therefore simultaneously learn the quantity $(D/\bar{n}) - (R/\bar{n}) = (\sum_{i \in \mathcal{P}} q_i - K)/\bar{n}$, which happens to be the current (tentative) pain per active retailer. If that pain exceeds any active retailer's q_i then that retailer i marks itself as "passive" (and is implicitly no longer in $\mathcal{P}$ even though it continues to be a party to the protocol).

3. The "pain per active retailer" that was computed in the last iteration of the above Step 2(d) is taken to be the true one, and every active retailer i computes his allocation $Q(q_i, \bar{n})$ as being equal to q_i minus that "pain per active retailer".

Note that, in the above, retailers who are no longer active continue to participate in the protocol (of course they now contribute 0 rather than 1 to the distributed computation of $\bar{n}$): Excluding them from subsequently participating in the protocol would have the drawback of revealing to the other participants who is no longer in $\mathcal{P}$.

The number of iterations in the above protocol could, in the worst case, be N. We have made an observation that brings the number of iterations down to $\log N$. In a nutshell, after defining two functions f and g on the indices $\{1, \ldots, N\}$, we give a characterization of the "stable value" of $\bar{n}$ in terms of the relationship between $f(\bar{n})$ and $g(\bar{n})$ that is reached at the last iteration of Step 2. The characterization in turn makes possible a binary search for the stable $\bar{n}$ in Step 2. We omit the details.

Proportional Allocation Protocol

1. The N retailers cooperatively choose a random R' that is known to all except the supplier.
2. Each retailer i sends $R' * q_i$ to the supplier.
3. The supplier computes D, the sum of what he received, $D = R' * (\sum_{i=1}^{N} q_i)$, and sends $D' = D/K$ to every retailer.
4. Every retailer i computes its allocation q_i' as $q_i' = R' * q_i/D' = q_i * K/(\sum_{i=1}^{N} q_i)$.
5. Every retailer i sends its q_i' to the supplier. The supplier verifies that the sum of the q_i''s equals K. If so the protocol terminates. Otherwise cheating has taken place by one or more retailers, where "cheating" by retailer i means sending a q_i' that is not consistent with the initial q_i that retailer i had used earlier in Step 2; i.e., it consists of retailer i changing its mind about its quantity after it has learned (in Step 4) what its true q_i' would have been. If cheating has been detected then Step 6 below pinpoints which retailer(s) cheated.
6. For every retailer i, the supplier determines whether i has cheated as follows: The supplier compares, for all other retailers j, the ratio q_i'/q_j' (available from Step 5) with the ratio q_i/q_j (available from Step 2). If the two ratios do not equal each other for a majority of other values j then the supplier decides that retailer i is a cheater.
   Note 1. It is easy to see that the above cheater-detection scheme works as long as a majority of the retailers are honest.
   Note 2. Keith Frikken has pointed out that, instead of verifying for every i, j pair in Step 6, the supplier could simply compute for every retailer i the ratio of q_i' to the $R' * q_i'$ he received from i in Step 2 (that ratio must be the same for all i's).

4. E-Auctions

In economics, information asymmetry has been widely studied in using principal-agent models with adverse selection (see [9]). These models assume that a principal makes decisions and sets contracting parameters for single or multiple agents, without complete information about agent's "actions" [1, 20, 26]. Auction theory has also been used to model information-asymmetry problems, as described in the seminal papers by Vickrey [27], Myerson [16], Riley and Samuelson [19], and Milgrom and Weber [15]. See Klemperer [14] for a more recent review on the theory of auctions. The use of auctions for allocating resources such as securities is described by Harris and Raviv [11]. Optimal auctions typically invoke the revelation principle, which states that it is sufficient for a principal to restrict his/her attention to contracts/auctions that induce the agents to tell the truth. Although useful in theory, the revelation principle does not necessarily yield practical procedures and protocols.

We consider two broad models: One where all buyers (= bidders) get the same unit price from the supplier (non-discriminatory pricing), and another where different buyers can get different prices from the supplier depending on their demand (discriminatory pricing). We begin with the former.

4.1. E-Auctions with non-discriminatory Pricing

In this model of supply-chain interaction, a seller wants to fix the selling price for all the buyers. Each buyer i has a price-quantity pair (p_i, q_i) expressing his preference to buy q_i units at a unit price of p_i, based on an underlying demand curve $q_i = \theta_i - p_i$. The seller has a supply curve $q = p + \theta$ and wants to figure out what price $\hat{p}$ she should ask from all of them according to the total demand: $\hat{p}$ is the price from the supply curve that corresponds to the total demand $\sum_{i=1}^{n} q_i$.

Under the rules of the auction, each buyer's demand parameter θ_i is not to be revealed to any other buyer. Further, the seller is to remain ignorant of any buyer's individual demand parameter before setting her price, thereby facilitating a policy of non-discriminatory pricing. The price charged by the seller is a function of the bids received. After the common price $\hat{p}$ is announced, only those buyers i whose price p_i is lower than $\hat{p}$ are allowed not to buy, and those buyers i whose $p_i \geq \hat{p}$ are not allowed to jack up their q_i.

4.1.1 Information and Decision Criteria for non-discriminatory price auctions

The relevant information from the buyers is their price-quantity pair bids (p_i, q_i). The fixed price charged by the seller is a function of the bids received. However, the seller is not supposed to know the total demand of the bidders before setting her price. After the common price $\hat{p}$ is known to everybody, only those buyers i whose price p_i is lower than $\hat{p}$ are allowed not to buy, and those buyers i whose $p_i \geq \hat{p}$ are not allowed to jack up their q_i. This is achieved by having each buyer i, as a first step in the protocol, send the

seller a "commitment" to its p_i and (separately) one for its q_i, without revealing either of them to the seller; this ties the hands of buyer i and prevents her from modifying p_i or q_i after the negotiation is over (for details of how commitment is done using cryptography we refer the reader to textbooks such as [24]).

At the end of the protocol, a buyer i whose $p_i < \hat{p}$ will "open" her commitment to p_i (i.e., reveal p_i to the seller) as a justification for not buying at price $\hat{p}$, whereas a buyer whose $p_i \geq \hat{p}$ will open her commitment to q_i (i.e., reveal q_i to the seller) as a proof that she did not change her original q_i after learning of the advantageous $\hat{p}$. It is a crucial property of cryptographic commitment protocols that the seller can verify whether the revealed p_i or q_i match the commitment originally sent by buyer i. Note that no buyer i reveals to the seller both p_i and q_i, and that no buyer knows $\sum_{i=1}^{N} q_i$. The protocol is given below.

Non-Discriminatory Pricing Protocol

1. Every bidder i gives the seller a cryptographic commitment to its q_i (which, as discussed earlier, does not reveal q_i to the seller yet prevents the bidder from changing its q_i value later on).

2. Every buyer initially marks itself as "active" (some will later mark themselves as "passive" as the protocol proceeds). We use $\mathcal{P}$ to denote the set of active buyers to denote $|\mathcal{P}|$; at this stage $\bar{n} = N$..

3. Repeat the following substeps (a)–(c) until $\bar{n}$ ceases to change from one iteration to the next:

   (a) The buyers and the seller all engage in the secure summation protocol (twice) to simultaneously get (i) $\bar{n}$ and (ii) $\hat{p} = \sum_{i \in \mathcal{P}} q_i - \theta$; recall that $p = q - \theta$ is the seller's supply curve. For the $\hat{p}$ computation, the "data" used by an active buyer i in this summation protocol is q_i, by a passive buyer is 0, whereas the supplier uses $-\theta$. For the $\bar{n}$ computation, the data is 1 if that buyer is active (i.e., in $\mathcal{P}$), and 0 otherwise.

   (b) If the computed $\bar{n}$ is the same as it was in the previous iteration of these substeps (a)–(c) then the protocol moves to Step 4 below, otherwise it continues with the next substep (c).

   (c) Buyers whose $p_i < \hat{p}$ mark themselves as "passive" (i.e., no longer in $\mathcal{P}$).

4. Buyers whose $p_i \geq \hat{p}$ reveal their q_i to the seller, who verifiers that it matches the commitment received in Step 1.

The table below summarizes who knows what after the above protocol completes:

Who knows what	(p_i, q_i)	(p_j, q_j) $i \neq j$	θ	$\sum q_i$	$\hat{p}$
Supplier			√	√	√
Buyer i	√				√

4.2. E-auctions with discriminatory pricing

The main difference from the non-discriminatory case is that, whereas in the former all the buyers get the same price, in this framework the price paid by each buyer is not fixed but rather is a function of its bid.

The relevant information from the buyers is their price-quantity pair bids (p_i, q_i). In this framework, the price paid by each buyer is not fixed, but a function of its bid. The goal for the seller is to set the price paid by the buyers as a function of the bids received so as to maximize its revenue, i.e. $\max \sum_i p_i q_i$, subject to the supply constraint $\sum_i q_i \leq K$.

The "pick-and-choose" protocol (below) reveals to each buyer only which (if any) of that buyer's alternative (p_i, q_i) is accepted by the seller, without revealing to the seller either p_i or q_i; the seller may have to eventually know more for external reasons such as shipping, but that is not inherent to the protocol.

4.2.1 Pick-and-Choose framework

The problem of finding the minimum number of units to be sold to the bidders, with the maximum possible revenue, was investigated by Sandholm and Suri in [23]. They proved that it is $\mathcal{NP}$-Complete, and devised a pseudo-polynomial algorithm to solve it. Our protocol extends that algorithm to make it secure, in the sense that no (price,quantity) pair is to be revealed to other bidders. The details are given in [6]

4.2.2 An Architecture for Discriminatory Pricing in the Single-Seller Case

An immediate problem with discriminatory pricing is that buyer i would apparently have to reveal to the seller both p_i and q_i, which compromises that buyer's demand curve (this was not a problem in non-discriminatory pricing because the seller did not get p_i). This would not be a problem in the case of multiple sellers. For the single-seller case, there is a need for designing architectures to solve this problem. One possibility is to introduce another party to the protocol, i.e., a proxy with the following assumptions and goals: (i) The seller learns the total quantity actually sold (not the individual q_i's), (ii) the proxy who learns the individual q_i (and can therefore direct shipping), (iii) the seller learns the dollar amount due from each buyer (the product $p_i q_i$, not the individual p_i), (iv) no buyer learns the total quantity sold or

any price paid by another buyer, (v) the proxy does not collude with the seller or with any of the buyers, but is otherwise untrusted in the sense that he is not supposed to know any (price,quantity) pair of any other participant in the protocol. We next describe some scenarios for the interactions between the seller and the buyers.

Notes about the results below, using proxy architecture: The results below can be extended to the case of multiple sellers if the allocation of the total quantify to seller i is a fixed fraction $c_i < 1$ of the total quantity. Both the seller proxy and the buyer proxy can be eliminated, and a direct many-buyer and many-seller protocol is possible, if we assume the honest-but-curious model and have no worry about keeping the parties honest (whereas in what follows we do worry about buyers changing their mind about their bid ex-post facto, about the proxy not fulfilling its obligations, etc).

In some of the scenarios described next, the seller is trying to set the price of each buyer. Mainly the following steps are honored by the different parties in order to comply with the proposed architecture. We also consider issues such as how to keep the proxy honest, and how to make the protocols resilient against collusion by a subset of the participants.

1. Each buyer i has his request q_i that he does not want to reveal to the proxy unless his request would be satisfied.

2. The proxy knows the maximum capacity of the seller, which is Q.

3. A protocol is run between the proxy and the buyers in order to settle a $\widetilde{q_i}$ for each buyer i according to some seller's capacity allocation model. This protocol should neither reveal Q to the buyers nor reveal q_i to the proxy or to the other buyers. The result of this protocol is that each buyer knows the available quantity he can receive $\widetilde{q_i}$. The proxy knows each $\widetilde{q_i}$ too, as he will be the distributor of the quantities later.

4. A protocol is run between each buyer separately and the seller himself so that each of the buyers can know what is the total amount that he has to pay, $\widetilde{p_i}\widetilde{q_i}$. The seller also knows that amount as it will be his revenue. One possible protocol is the oblivious polynomial evaluation protocol [17, 5].

5. The seller collects $\widetilde{p_i}\widetilde{q_i}$ from each buyer i.

6. The proxy sends the total required quantities, $\sum_i \widetilde{q_i}$, to the seller who will send the items to the distributor (the proxy).

7. the proxy distributes the items on the retailers.

The table below summarizes who knows what after the protocol completes:

Who knows what	(p_i, q_i)	$\widetilde{p_i}$	$\widetilde{q_i}$	$\widetilde{p_i}\widetilde{q_i}$	$\widetilde{p_j}$ $j \neq i$	$\widetilde{q_j}$ $j \neq i$	$\sum \widetilde{q_i}$
Supplier				$\surd$			$\surd$
Proxy			$\surd$			$\surd$	$\surd$
Retailer i	$\surd$	$\surd$	$\surd$	$\surd$			

Keeping the Proxy Honest

In the protocol as given above, the proxy can steal from the seller by sending him a different total quantity, other than the real $\sum_i \widetilde{q_i}$. We need to modify the protocol so that it allows the seller to detect this kind of cheating. The following modification achieves this:

1. The first retailer sends $\widetilde{q_0} + r$ to the next buyer where r is a large random number known only to this buyer.

2. Each other buyer will add his $\widetilde{q_i}$ to the number he received from the previous buyer, then sends the sum to the next buyer.

3. The last buyer sends the sum to the seller, and the first buyer sends r to the seller.

4. The seller adds r to the total quantities received from the proxy. If it is equal to the sum he has received from the last buyer $r + \sum_i \widetilde{q_i}$ then the proxy had sent the correct total quantity, otherwise the proxy was trying to cheat.

Preventing collusion

The previous protocol prevents the proxy from cheating, but what about the seller cheating? She can collude with the second buyer so as to know the q_i of the first one. She can also collude with any buyer so as to get the sum of the q_i's of the buyer before her in that ordering. Now, we modify the previous protocol to get one that keeps the proxy honest (unless a buyer colludes with him), and also prevents the successful collusion of the seller with any buyer (our scheme actually works for collusion by many, but for reasons of space limitations we do not include the general description).

1. The first buyer sends $g^{\widetilde{q_0}+r} \bmod p$ to the next buyer where r is a large random number known only to this buyer, g and p are public (known to all participants), p is a large prime, $g < p$ and it is best if g is a primitive root.

2. Each other buyer will multiply his $g^{\widetilde{q_i}} \bmod p$ to the number he received from the previous buyer, then sends the product to the next buyer.

3. The last buyer sends the product to the seller, and the first buyer sends $g^r \bmod p$ to the seller.

4. The seller multiplies $g^r \bmod p$ wby (g raised to the a power equal to the total quantities received from the proxy, modulo p). If the result is equal to the product he has received from the last buyer $g^{(\sum_i \widetilde{q_i})+r} \bmod p$ then the proxy is honest in sending the total quantities. Otherwise the proxy was stealing from the seller.

Although the seller has $g^r \bmod p$ and can collude with the second buyer to receive $g^{\widetilde{q_0}+r} \bmod p$, knowing q_0 is still as hard as solving the discrete logarithm problem (a problem widely believed intractable). However, this is true only for large values of q_i's – for small q_0, the seller can find it by trying all its possible values. We have a scheme that overcomes this drawback (we cannot include it here due to space limitations — for the same reason we do not include the extension to collusion-resistance against collusion by c entities for any a priori known constant c).

We next describe a possible scenario for the interactions between the seller and the buyers for the purpose of clearing the market.

4.2.3 All-or-None Framework:

In this framework, the bidders make their offers as price-quantity pair bids, and the seller has either to accept or to reject the whole bunch according to her supply curve. The bidders do not want to reveal their offers before the seller's decision is made.

Let (p_i, q_i) be the pair bid of bidder i. Let the supply curve of the seller be $q = p + \theta$. Without knowing the offer, the seller needs to know whether the revenue will be as good as what her supply curve requires. Hence the problem is to compute this predicate without revealing any additional information about the supply curve or about the price-quantity pair bids. The revenue that she will get from this offer is $\sum_i p_i q_i$. The unit price that she expects due the current demand is $\hat{p} = \sum_i q_i - \theta$. Thus she expects a revenue of $\left(\sum_i q_i - \theta\right)\left(\sum_i q_i\right)$. Thus, our problem is defined now as computing the predicate $\sum_i p_i q_i \geq \left(\sum_i q_i - \theta\right)\left(\sum_i q_i\right)$ without revealing any (p_i, q_i) or θ. The following protocol allows the seller to make her decision without revealing her supply curve, and without revealing to her any of the price-quantity pair bids of the buyers.

The table below summarizes who knows what after the protocol completes:

Who knows what	(p_i, q_i)	(p_j, q_j) $i \neq j$	θ	$\sum q_i$	$\sum p_i q_i$
Supplier			$\surd$		
buyer i	$\surd$				

4.2.4 All-or-None Protocol:

Initially, each bidder sends to the seller's proxy a "commitment" to its q_i and to the seller a "commitment" to its $p_i q_i$, without revealing either of them to the seller or to the proxy. Secure summation protocol for additively split data (discussed below) is used to generate x_a and x_b such that $x_a - x_b = \sum_i q_i - \theta$. It is also used to generate x_c and x_d where $x_c - x_d = \sum_i q_i$. These four values should be with four different persons; x_i is with person i. We compute $(x_a - x_b)(x_c - x_d) = (x_a x_c + x_b x_d - x_a x_d - x_c x_b)$ as follows:

- a sends x_a to c who computes $y_c = x_a x_c$.
- b sends x_b to d who computes $y_d = r_d + x_b x_d$, where r_d is a random number chosen by d.
- c sends x_c to b who computes $y_b = r_b - x_c x_b$, where r_b is a random number chosen by b.
- d sends x_d to a who computes $y_a = -x_d x_a$.

b sends y_b to a who adds it to his y_a. Meanwhile d sends y_d to c who adds it to his y_c. Then d sends his new y_d to a who adds it to his current y_a. Thus a now has the value of $x_a = \left(\sum_i q_i - \theta\right)\left(\sum_i q_i\right) + r_b + r_d$.
Similarly, the summation protocol for additively split data is run to find $\sum_i p_i q_i$ and a receives z_a and d receives z_d, such that $z_a - z_d = \sum_i p_i q_i$. d sends $r_d - z_d$ to a who adds it to his z_a. Now a has the value of $z_a = \sum_i p_i q_i + r_d$.
a computes $z_a - x_a$ and sends it to the seller. The seller runs Yao's millionaire protocol [28] with b to see whether the value in her hand is larger than r_b. If so, then she accepts all the offers. Otherwise, she reject them all.
In case she accepts the offers, the bidders reveal their q_i's to the proxy who will sum them up and sends $\sum q_i$ to the seller. The bidders also reveal their $p_i q_i$ to the seller. Now the seller can verify that the revealed data are the ones they have committed to and can also check for the verified predicate. In case the seller rejects the whole deal, the bidders do not have to reveal their price-quantity pairs.

5. Some Building Blocks

We now present details of some building blocks that were used in our protocols. by increasing order of complexity.

5.1. Secure Simultaneous Multi-Party Summation Protocol

The purpose of this protocol is to make n parties, each with a number A_i, cooperate to simultaneously find out $S = \sum_{i=0}^{n-1} A_i$ without revealing to each other anything

other than that answer S. In the protocol that follows, when we say that a person i having an item x and a person j having an item y *simultaneously* exchange their respective x and y, we assume that this exchange happens in a single step – the details of how to achieve such a simultaneous exchange of secrets between two parties are in many textbooks and are omitted (see, e.g., [24]). This will be typically necessary only in a protocol's last step (the one that reveals the answer) rather than in the protocol's intermediate steps (in which it is fine if i gives his x to j and then right after that j gives his own y to i). As a practical matter, and because of the considerable overhead and complexity of the known protocols for the simultaneous exchange of secrets, one could avoid them by settling for the less-than-ideal (but perfectly fine for our purpose) exchange of x and y bit by bit: i sends a bit of x, then j sends a bit of y, and they alternate until done – anyone who lies will have to do so before he completely learns the other's secret, but he could have done that anyway by lying about his own A_i in the first place. We will henceforth just use the notion of simultaneous exchange of secrets without specifying which actual technique is used for achieving it.
Essentially the same protocol can be used when the data is additively split and the answer is to come up similarly split (here an x is split is in the sense that two parties have random-looking x' and, respectively, x'' that add up tp x). In preparation, the following is done:

- Every party i gets a random number R_i.
- Every party $2i$ gives to $2i+1$ his $A_{2i} + R_{2i}$, then every $2i+1$ gives to $2i$ his R_{2i+1}.

Now the odd-numbered parties have the $A_j + R_j$ of everybody spread amongst them, and the even-numbered parties will have the R_j of everybody spread amongst them.

Now the odd (resp., even) -numbered parties compute $A+R$ (resp., R), where $A = \sum_{i=0}^{n-1} A_i$ and $R = \sum_{i=0}^{n-1} R_i$. Finally, the odd (resp., even) simultaneously exchange their quantities to obtain A. The computation of $A + R$ (resp., R) is done using a straight forward "tree based" approach whose details are omitted.

5.2. Minimum Finding Protocol for Already-Split Data

The second building block is how to find the minimum of a set of data where each datum is additively split between two parties. Here by "x is additively split" we mean that $x = x' + x''$ and one party has x' while the other has x'' (and x', x'' could be quite large and negative, so that x is effectively unknown to either one of the two parties).

In [13], Atallah et al proposed a secure protocol to compute the minimum element of a vector $\vec{c}$ that is shared additively between two parties: Alice has $\vec{a} = (a_1, \ldots, a_l)$, Bob has $\vec{b} = (b_1, \ldots, b_l)$, and $\vec{c} = \vec{a} + \vec{b}$. After running the protocol, Alice ends up with a γ_A and Bob with a γ_B such that $\gamma_A + \gamma_B = min_{i=1}^{l} c_i$. The protocol is nontrivial and we omit its details from here (they can be found in [13]).

5.3. Secure Filtered Maximization Protocol

Alice and Bob are sharing a vector $\vec{c} = \vec{a} + \vec{b}$ additively, such that Alice has $\vec{a}$ whereas Bob has $\vec{b}$. They want to find $\hat{i} = \max\{i : c_i \leq Q, 1 \leq i \leq n\}$. As usual, neither Alice nor Bob wants to give his vector to the other – in fact the protocol results in the answer itself being additively split between them: Alice gets a random-looking $\hat{i_1}$ and Bob a random-looking $\hat{i_2}$ such that where $\hat{i_1} + \hat{i_2} = \hat{i}$.

The aim of the protocol is to get $\hat{i} = \max\{i : c_i \leq Q, 1 \leq i \leq n\}$ which can be represented also as $\hat{i} = \max\{i * sign(Q - c_i), 1 \leq i \leq n\}$ where $sign(m)$ is equal to 1 if $m \geq 0$ and is equal -1 otherwise. The protocol is executed by repeating the following steps for each $1 \leq i \leq n$ to create two vectors $\vec{K}$ and $\vec{L}$ such that $K_i + L_i = i$ if $c_i \leq Q$ and $K_i + L_i = -i$ otherwise. Alice will hold $\vec{K}$ while Bob will hold $\vec{L}$.

1. Bob generates a random vector $\vec{X} = (x, x')$, and computes the vector $\vec{Y} = (y, y') = (i - x, -i - x')$.
2. Bob generates a random permutation Π such that $\Pi \circ \Pi = I$.
3. Bob sends $\widetilde{X} = \Pi X$ to Alice.
4. Bob generates two random numbers α and β. He also generates a random split q_A and q_B for Q such that $Q = q_A + q_B$.
5. Bob creates a vector $\vec{S} = (b_i - \alpha - m(\Pi), q_B - \beta)$, where $m(\Pi)$ equals 1 if $\Pi \neq I$ and is 0 otherwise.
6. Alice creates a vector $(a_i, 0)$ and uses it to run a one-sided Blind and Permute protocol [13] with Bob who uses the same Π in it. The outcome of this protocol is that Alice gets a vector $\vec{T} = \Pi(a_i + \alpha, q_A + \beta)$. During this protocol neither Alice nor Bob can deduce a private value of the other party.
7. An asymmetric Yao's millionaire protocol is run between Alice and Bob. In this protocol, Alice uses $T_1 - T_2$ as her input, whereas Bob uses $S_2 - S_1$ if $\Pi = I$ or $S_1 - S_2$ if $\Pi \neq I$. Only Alice knows the result of this protocol. If she figures out that her input to the protocol is larger than Bob's input, then she sets $K_i = \widetilde{X}_2 - \gamma$ (case 1), otherwise she sets $K_i = \widetilde{X}_1 - \gamma$ (case 2), where γ is a random number selected by Alice.

8. A one-sided Blind and Permute protocol is run between Alice and Bob, in which Bob's input is $\vec{Y}$. The output of that protocol is that Bob will receive a value of $\widetilde{Y}_2 + \gamma$ in case 1 or $\widetilde{Y}_1 + \gamma$ in case 2; where $\widetilde{Y} = \Pi Y$. He should sets L_i to his output.

The maximum finding protocol for already-split date is run on $\vec{K}$ and $\vec{L}$ so that to give $\hat{i_1}$ to Alice and $\hat{i_2}$ to Bob, where $\hat{i_1} + \hat{i_2} = \hat{i}$.

6. Conclusion and Future Work

We gave protocols for some supply-chain interactions. In future work, we will examine the impact of SSCC protocols on the well-known "bullwhip" effect [18].

References

[1] G. Akerlof. The market for lemons: Quality uncertainty and the market mechanism. *Quarterly Journal of Economics*, 89:448–500, 1970.

[2] Y. Aviv. The effect of collaborative forecasting on supply chain performance. *Management Science*, 47(10):1–18, 2001.

[3] Y. Aviv. Gaining benefits from joint forecasting and replenishment processes: The case of auto-correlated demand. *Manufacturing & Service Operations Management*, 4(1):55–74, 2002.

[4] G. P. Cachon and M. Fisher. Supply chain inventory management and the value of shared information. *Management Science*, 46(8):1032–1050, August 2000.

[5] Y.-C. Chang and V.-J. Lu. Oblivious polynomial evaluation and oblivious neural learning. In *Proceedings of ASIACRYPT*, Gold Coast, Australia, 2001. a comprehensive tutorial on WinWord Macro Virus.

[6] M. A. H. E. V. Deshpande and L. Schwarz. Secure supply chain protocols. Technical report, Center for Education and Research in Information Assurance and Security (CERIAS), 2003.

[7] V. Deshpande and L. Schwarz. Optimal capacity allocation in decentralized supply chains. Technical Report Working paper, Purdue University, Krannert School of Management, Dec 2002.

[8] A. Federgruen. *Centralized Planning Models for Multi-Echelon Inventory Systems Under Uncertainty*, volume 4 of *Handbooks in OR and MS, S. C. Graves et al. (editors)*, chapter 3. North Holland, 1993.

[9] D. Fudenberg. and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 2000.

[10] O. Goldreich. Secure multi-party computation (working draft). Available from http://www.wisdom.weizmann.ac.il/home/oded/public_html/foc.html, 1998.

[11] M. Harris and A. Raviv. Allocation mechanisms and the design of auctions. *Econometrica*, 49(6):1477–1499, 1981.

[12] A. V. Iyer and J. Ye. Assessing the value of information sharing in a promotional retail environment. *Manufacturing & Service Operations Management*, 2:128–143, Spring 2000.

[13] M. J. A. F. Kerschbaum and W. Du. Secure and private edit distance computaion. Technical report, Computer Science Department, Syracuse University, 2002.

[14] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Reviews*, 3:227–260, 1999.

[15] P. R. Milgrom and R. J. Weber. A theory of auctions and competitive bidding. *Econometrica*, 50(5):1089–1122, 1982.

[16] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[17] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation (extended abstract). In *Proceedings of the 31th ACM Symposium on Theory of Computing*, pages 245–254, Atanta, GA, USA, May 1-4 1999.

[18] H. L. V. Padmanabhan and S. Whang. Information distortion in a supply chain. *Management Science*, 43(4):546–558, 1997.

[19] J. G. Riley and W. F. Samuelson. Optimal auctions. *The American Economic Review*, 71(3):381–392, 1981.

[20] M. Rothschild and J. Stiglitz. Equilibrium in competitive insurance markets: An essay on the economics of imperfect information. *Quarterly Journal of Economics*, 80:629–649, 1976.

[21] J. M. R. Roundy. *Analysis of Multistage Production Systems*, volume 4 of *Handbooks in OR and MS, S.C. Graves et al.(editors)*, chapter 2. North Holland, 1993.

[22] R. C. Y. I. R. K. M. R. R. Rubinfeld, and R.N.Wright. Selective private function evaluation with applications to private statistics (extended abstract). In *Proceedings of the Twentieth ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.

[23] T. Sandholm and S. Suri. Market clearability. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, 2001.

[24] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1995.

[25] J. S. Song and P. H. Zipkin. Inventory control with information about supply conditions. *Management Science*, 42(10):1409–1419, 1996.

[26] A. M. Spence. *Market Signaling*. Harvard University Press, Cambridge, MA, 1974.

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

[28] A. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982.