Secure Supply-Chain Collaboration:
A New Technology for Supply-Chain Management[1]

Mikhail Atallah
Vinayak Deshpande
Leroy B. Schwarz*

Purdue University
West Lafayette, IN USA

*Corresponding Author: Lee@mgmt.purdue.edu

Abstract

One of the major sources of inefficiency in managing supply chains is information asymmetry; i.e., information known by one or more links in the chain (e.g., a retailer) is not known by other links (e.g., the manufacturer) . Information asymmetry is known to create inefficiencies in managing supply chains, among them under-investment in capacity, leading to shortages, misallocation of inventory, transportation, increased prices, and reduced customer service. There are several causes of information asymmetry, among them fear that a supply-chain partner will take advantage of private information, that information will leak to a competitor, fear of espionage, hacking, etc.

This paper will introduce and illustrate the use of "secure multi-party protocols" in managing supply chains. These protocols, which we label "secure supply-chain collaboration protocols", enable supply-chain partners to cooperatively achieve desired mutually agreed-upon goals without revealing the private information of *any* of the parties, even though the jointly-computed decisions require the information of *all* the parties.

Although, at first sight, SSCC protocols may seem impossible, there are known general theoretical results in computer science that have established the existence of protocols for such so-called "secure multiparty computation problems". Unfortunately, such general protocols are impractical, and the supply-chain interactions framework is very different from what is found in the multi-party computation literature. Practical secure multiparty techniques do exist in other areas (e.g., information retrieval, electronic voting, scientific computing, approximate matching). We are developing and providing such protocols to manage supply chains.

[1] This work is supported in part by the e-Enterprise Center at Discovery Park, Purdue University and by the National Science Foundation Grant IIS-0325345

1. Introduction

It is well known that information asymmetry — for example, that the manufacturer of some product is ignorant of information known by a retailer, or vice versa — is a major source of inefficiency in managing supply chains. Among other things, information asymmetry lead to the wrong investment in capacity and to the misallocation of capacity. See Cachon and Lariviere, 1999, for example. It can also lead to distorted prices, increased costs, and reduced customer service.

In the early days of supply-chain management, the major obstacle to information-sharing *seemed* to be technological; that is, buyers and their suppliers either didn't have the information available for their *own* decision-making systems, or the appropriate information-sharing systems either weren't available or were expensive or difficult to use.

During the last 10-15 years, the development and wide-spread use of bar-coding, satellite communications, enterprise resource-planning systems, desktop computers, and the internet have removed most of these technical barriers. In addition, companies such as Wal-Mart have demonstrated that information-sharing provides the "Holy Grail" of inventory management: reduced investment *and* improved customer service.

Despite this, information-sharing among partners in a supply chain remains the exception, not the rule. Why? For some buyer-supplier pairs, lack of trained personnel is a significant barrier; for others, the wrong incentive systems are in place. Over time, these obstacles will also diminish or disappear.

Hence, over time the technical and operational obstacles to wide-spread information-sharing will disappear. However, we believe that information asymmetry will remain the rule, not the exception. Until, that is, what, we believe, have been *the* three major obstacles to information-sharing all along have been removed: First, fear that a supply-chain partner will take advantage of "private"[2] information that is shared.

[2] Our working definition of "private" information is any information that its owner(s) do(es) not want revealed.

For example, that a supplier, "seeing" that its buyer has low inventory, will postpone a price promotion, raise prices, or ration supply; or, that a buyer, "seeing" that its supplier has large amounts of inventory, will demand reduced prices and/or more favorable terms. Second, that shared information will unintentionally be leaked to others, including competitors. See Li (2002), for example. Third, that shared information will be subject to hacking and/or industrial espionage.

The Obvious Question

So, then, the obvious question is: "Is it possible to achieve the benefits of information-sharing *without disclosing* private" information? Surprisingly, the answer, in many business scenarios, is "yes".

The answer is "yes" in those business scenarios in which the *value* of information-sharing is neither the *information* nor its *sharing*, but rather in *improved decision-making*; that is, decision-making that is better *because* decisions are based on shared information. This, then, leads to the more basic question: "Is it possible to make decisions based on private buyer-supplier information without actually sharing that information?"

Although it might seem, at first glance, that the answer to this question is "no", often the answer is "yes". This "yes" answer is provided by a decades-old area in computer science called "secure multi-party computation".

2. Secure Multi-Party Computation

There is an extensive literature in multi-party computation since it was introduced by Yao (1982). See Goldreich, Micali, Wigderson (1987), for example. In addition, secure multi-party protocols are already in use in certain specific applications, such as electronic voting and secure information retrieval.

The Secure Multi-Party Computation Paradigm: An Example

Consider two decision-makers: Alice and Bob. Alice has private information, X_A; Bob has private information, X_B. Bob and Alice agree that they desire to make either a joint decision (or separate decisions) based on a function, f(X_A, X_B), of their joint private information. For simplicity of presentation, we will henceforth assume that f(X_A, X_B) is a jointly agreed-upon decision function.

Given that f(X_A, X_B) can be computed using (X_A, X_B) and other available information, then one general result in secure multi-party computation is that a protocol does exist that will compute f(X_A, X_B) and provide its value to Alice and/or Bob *without* disclosing X_A to Bob or X_B to Alice. Most important, this protocol doesn't require a "trusted third party".

The problem with a so-called "trusted third party", of course, is that knowledge of (X_A, X_B) and/or f(X_A, X_B) has value that might tempt this third party to disclose Alice's private information to Bob, or vice-versa, or, possibly worse, disclose Alice and Bob's joint private information to another party who will take advantage of it. Furthermore, since secure multi-party protocols process only mangled versions of X_A and X_B, these protocols are invulnerable to hacking. Indeed, even the persons who coded the protocol itself is not able to learn either X_A or X_B.

Suppose, for example, that Alice, Bob, and Carl are colleagues in a software-development firm who share an interest in determining their average salary. Hence,

$$f(X_A, X_B, X_C) = X_{AVG} = (X_A + X_B + X_C)/3 \qquad (1)$$

where X_i is the salary of person i = A, B, C. Using a "trusted" colleague, Dwayne, Alice, Bob, and Carl would privately communicate their salaries to Dwayne, who would evaluate (1) and communicate the average back to them. However, now Dwayne not only knows X_{AVG}; Dwayne knows the vector (X_A, X_B, X_C), which *none* of his trusting colleagues know.

One simple secure protocol to evaluate (1) is for Alice, Bob, and Carl to jointly select a random number, R, and each add R to their salaries before communicating them to Dwayne. In other words, Alice, Bob, and Carl individually compute their X_i', where $X_i' = X_i + R$, i = A, B, C. As above, each would then communicate their X_i' to Dwayne, who would compute

$$f'(X_A, X_B, X_C) = X_{AVG}' = (X_A' + X_B' + X_C')/3 = X_{AVG} + R \qquad (2)$$

Once Dwayne provides X_{AVG}' to Alice, Bob, and Carl, each could then learn X_{AVG} simply by subtracting R from X_{AVG}'. Note, under this protocol, Dwayne doesn't learn either X_{AVG} or the vector (X_A, X_B, X_C).

3. Secure Supply-Chain Collaboration (SSCC)

In our research, we are applying secure multi-party computation to supply-chain management. These protocols are much more complicated than the protocol illustrated above, but in their basic function is the same: to compute f(•,•,... •) without disclosing the private information in its arguments. We call this "secure supply-chain collaboration (SSCC)".

We have developed SSCC protocols for simple e-Auction scenarios (Atallah, et al. 2003), one of which we will describe below, and for simple capacity-allocation scenarios. We are developing SSCC protocols for bullwhip scenarios. We have also developed a SSCC protocol to do "price-masking" for a large consumer-electronics company. See section 5.

Although our research has only begun, it is our intention to compare the effectiveness of SSCC protocols with non-cooperative decision-making (using asymmetric information) in a wide variety of supply-chain business scenarios, stylized and real-world, to develop proof-of-concept software, and to examine security versus cost tradeoffs.

An Example: e-Auction Scenario

Consider a simple e-Auction scenario involving a single supplier and N potential buyers. The supplier has supply curve $q = p + \theta$ and each potential buyer i has a price-quantity pair (p_i, q_i) that it wants to purchase, based on its demand curve $q_i = \theta_i - p$. The auction is designed to determine the fixed (i.e., non-discriminatory) price, p_F, that every buyer will pay and quantity q_F, the total quantity demanded and supplied such that $q_F =$

$p_F + \theta = \Sigma_{\iota \varepsilon P} q_{\mathrm{i}}$ where P is the set of potential buyers whose $p_i \geq p_F$. The potential buyers and the supplier agree that none of the potential buyers θ_i nor the supplier's θ are to be disclosed before, during, or after the auction has taken place.

Note that after the common price p_F is announced, only those buyers i whose price p_i is lower than p_F should be allowed not to buy, while those buyers i whose $p_i > p_F$ should be not allowed to increase their q_i. This is achieved by having each buyer i send the supplier a "commitment" to its p_i and (separately) one for its q_i, without revealing either of them to the supplier. Figuratively, this "ties the hands" of each buyer i and prevents them from modifying either p_i or q_i after the auction is over. For details of how commitment is done, see, for example, Schneier, 1995.

At the end of the auction, any buyer i whose $p_i < p_F$ will "open" her commitment to p_i (i.e., reveal p_i to the supplier) as a justification for not buying at price p_F whereas a buyer whose $p_i \geq p_F$ will open her commitment to q_i (i.e., reveal q_i to the supplier) as proof that she did not change her original q_i after learning p_F. It is a essential property of cryptographic commitment protocols that the supplier can verify that the revealed p_i or q_i match the commitment previously sent by buyer i. Note that no buyer i reveals to the supplier both p_i and q, that potential buyer i doesn't know (p_j, q_j) for $j \neq i$, and that no buyer knows $\Sigma_\iota q_{\mathrm{i}}$..

The corresponding secure multi-party protocol is as follows:

1. Every buyer i gives the supplier separate cryptographic commitments to its p_i and q_i.
2. Every buyer initially marks itself as "active" (some will later mark themselves as "passive" as the protocol proceeds). Let P to denote the set of active buyers and n to denote the cardinality of P; at this stage $n = N$.
3. Repeat (a)–(c) below until n ceases to change from one iteration to the next:
   (a) The buyers and the supplier all engage in a secure summation protocol (twice) to simultaneously determine n and $p_F = \Sigma_{\iota \varepsilon P} q_i - \theta$; recall that $p = q +$

θ is the supplier's supply curve. For the p_F computation, the data used by an active buyer i in this summation protocol is q_i, for a passive buyer. 0; whereas the supplier uses θ. For the n computation, the data is 1 if that buyer is active (i.e., $i \varepsilon P$), and 0 otherwise.

(b) If the computed n is the same as it was in the previous iteration of sub steps (a)–(c) then the protocol moves to Step 4; otherwise it continues with sub step (c).

(c) Buyers whose $p_i < p_F$ mark themselves as "passive" (i.e., no longer in P).

4. Buyers whose $p_i \geq p_F$ reveal their q_i to the supplier, who verifies that it matches the commitment received in Step 1.

We have also developed secure protocols to conduct a similar auction, but with discriminatory prices. See Atallah, et al., 2003, for details.

5. Price-Masking for a Consumer-Electronics Company

We are working with Privacy-Preserving Collaboration Technologies, L.L.C. (PriProTex™), a consulting firm, to help one of its clients, a large consumer-electronics company, to apply SSCC technology to what the client calls "price-masking".

To understand price-masking, and how it is competitively important to the client, consider the following fictitious business scenario, which, nonetheless, represents the important elements of the client's real business situation: There are several consumer-electronics companies (e.g., Panasonic, Sony), each making products that compete with one another in the same marketplace (e.g., MP3 players). Each of these companies designs, but does not assemble its products. Instead, products are assembled by one of several contract manufacturers (e.g., Solectron, Flextronics). The parts for these products are manufactured by one or more parts manufacturers (e.g., Intel, Mallory).

It is important to understand that each consumer-electronics company typically out sources assembly of different products to different contract manufacturers and also buys parts from many of the parts manufacturers under long-term contract. Similarly, each contract manufacturer typically assembles products for several consumer-electronics companies and buys parts from several parts manufacturers, also under long-term contract. Hence, the consumer-electronics industry supply chain is a complex network, with any given company being linked either directly or indirectly to virtually every other company.

In contracting for the manufacture of one of its products, each consumer-electronics manufacturer (e.g., Sony) typically provides a bill-of-material (BOM) to one (or more) of the contract manufacturers (e.g., Solectron). As currently practiced, each contract manufacturer prices each item in the product's BOM, then adds a percentage to the total parts price for the assembly process itself, in order to get its total build price.

Now, imagine Sony is examining the priced BOM provided by Solectron for some proposed Sony product. Imagine that in doing so, Sony observes that Solectron has a lower price on Intel chipset 842 than Sony's contracted price from Intel for the same chipset; but that Solectron's price on Mallory resistor 132 is higher than Sony's contracted price from Mallory. Of course, what Sony would *like* to do is to pay the lowest possible price for every component in the BOM. In the example, Sony would like to pay its price for Mallory resistor 132 but pay Solectron's price for Intel chipset 842.

One way that Sony *could* pay the lowest price for both the resistor and the chipset is for Sony management to tell Solectron that Solectron should buy the Intel chipset from Intel directly, but that Sony would purchase the resistor from Mallory and ship it to Solectron. The "problem" is that Sony, thereby, discloses to Solectron that Sony pays a lower price for the Mallory resistor than Solectron's quoted price. Further, if Solectron has quoted its cost from Mallory, Solectron learns that its price from Mallory is higher than Sony's price from Mallory *for the same resistor*. So, the next time Solectron contracts with Mallory, Solectron will demand a lower price for resistor 132. The long-term consequence of this is that, over time, Sony's parts-price advantage from Mallory shrinks or disappears entirely, since Solectron will eventually provide its lower price from Mallory to Sony's competitors (e.g., Panasonic). Another problem for Sony is that

Sony has to incur the overhead to purchase the resistor from Mallory, receive it, and then ship it to Solectron. Finally, and more fundamentally, from the beginning, Solectron's price (if not its costs) for each item in the BOM has been revealed to Sony. So, in the example, the next time Sony negotiates with Intel over chipsets, Sony will demand the same or lower price from Intel for chipset 842 than Solectron has quoted.

Hence, the goal of "price-masking" is to help Sony pay the lowest possible price for the components in each of its products assembled by Solectron, but without disclosing the private prices of either Sony or Solectron to the other. The technology PriProTex™ uses to facilitate such a transaction is proprietary, but is based, in part, on determining the minimum price for a set of components without disclosing to any party what those components are.

6. We Have Only Just Begun

We have only just begun to apply the principles and methods of secure multi-party computation to develop privacy-preserving protocols for supply-chain management. Based on our experience to date, we believe that SSCC has tremendous potential. However, we are also aware that some complex problems must be solved or compensated for before SSCC becomes a wide-spread reality. We have identified three sets of problems.

Secure Multi-Party Computation Problems

Although secure multi-party computation provide extremely valuable technology for use in managing supply chains, this technology brings with it several associated problems.

Perhaps the most serious of these problems involve collusion. Returning to our average-salary example, suppose that Alice and Bob decide to collude with one another in order to learn Carl's salary. By being willing to share their average salary Alice and Bob can learn Carl's salary. In other words by being willing to share $(X_A + X_B)/2)$, Alice and Bob can determine Carl's salary from X $_{AVG}$.[3] Although the possibility of collusion is not a failing of secure multi-party computation, it would make this technology much more desirable if secure protocols could either prevent collusion during the process and reveal it, if possible, through associated systems.

Another problem in secure multi-party protocols involves simultaneity. That is, although it is almost trivial to develop a protocol to compute f(•,•,...•), as described above, it is considerably more difficult to provide the value of this function *simultaneously* to all of the participants. To the extent that one or more parties might gain an advantage from learning f(•,•,...•) before others, SSCC protocols must solve the technical problem of simultaneity.

Supply-Chain Management Problems

Although in many business scenarios it is possible to reduce the fear that private information will be disclosed, thereby improving the profitability of the chain, the potential problem of asymmetric payoffs remains. In other words, although Alice and Bob's combined non-cooperative profits would be increased though SSCC collaboration, there is no guarantee that Alice and Bob will both benefit. Indeed, increased channel profits might require a decrease in profit for one or more of the partners. Of course, this problem isn't due to SSCC, but to the structure of the supply chain, the costs and revenues in each link, and the environment in which the supply chain .

In such business scenarios it is essential that enforceable contracting mechanisms be designed so that partners are motivated to tell the truth and share the benefits of doing so.

[3] Of course, in this example, with only 3 parties, Alice and Bob disclose their salaries to one another, too. However, in general (N-1) parties can collude against the Nth party without necessarily revealing their own private information to one another

Problems in Applying Secure Multi-Party Protocols to Managing Supply-Chains

Finally, there are problems in applying multi-party secure protocols to supply-chain management. One such problem is that in managing supply chains, *sometimes* Alice and Bob share an interest in the same decision function, $f(X_A,X_B)$. This is the case, for example, in the price-masking application described above. However, in other supply-chain scenarios Alice is interested, say, in $f_A(X_A, X_B)$ whereas Bob is interested in $f_B(X_A, X_B)$. If, for example, each is interested in maximizing their respective function with respect to profit, then interactions between the two functions must be resolved either before or during the protocols, and resolved in such a manner that private information is not disclosed.

Another complexity involves the general problem of "inverse optimization" (see Ahuja and Orlin, 2000, for example); that is, the extent to which partners involved in SSCC can "figure out" their partner's private information even though that private information isn't disclosed by the protocols themselves. In the average-salary protocol, for example, to what extent can each participant determine the other participant's salaries based on their own private information and X_{AVG}? To illustrate: if there are only two parties, Alice and Bob, participating in the protocol, then it is trivial for either of them to determine the other's salary (e.g. $X_B = 2 \bullet X_{AVG} - X_A$).

Therefore, in order for SSCC protocols to "guarantee" that private information cannot be determined based on information that is already available plus information provided by the protocols themselves, it may become necessary to mask some of $f(\bullet,\bullet,\ldots\bullet)$ to one or more of the partners.

7. Summary

In this paper we have introduced the notion of applying general principles and techniques developed in secure multi-party computation to practical problems and theoretical concepts in supply-chain management.

We believe that secure-supply chain collaboration SSCC technologies have the potential to revolutionize the practice of supply-chain management, by removing the major barriers to information-sharing: (1) fear that a supply-chain partner will take advantage of shared information; (2) fear that shared information will be unintentionally be leaked; and (3) fear of hacking and/or industrial espionage.

Given its advantages of information-sharing, we believe that in the future *every* new supply-chain management system will incorporate SSCC, even in those scenarios when partners have no fear about sharing any information with their partners. After all, why risk the disclosure of any information, private or otherwise, if the associated decisions can be made without sharing the information?

We further believe that SSCC has the potential to re-invigorate the theory of supply-chain management, which has lately been focused on the fact that supply-chain partners often make decisions based on asymmetric information. Indeed, we believe that much of the early theory in supply-chain management, theory that focused on omni-present information, can be recycled and extended using SSCC tools and techniques.

References

Ahuja, R. K. and J. B. Orlin. 2000. “Inverse Optimization”, *Operations Research* 49:5, pp. 771-783

Atallah, M. J., H. G. Elmongui, V. Deshpande, and L. B. Schwarz. 2003. “Secure Supply-Chain Collaborations”, working paper, Purdue University, West Lafayette, IN 47907

Cachon, G. and M. Lariviere. 1999. “Capacity Choice and Allocation: Strategic Behavior and Supply-Chain Performance”. *Management Science*, 45:8, 1091-1108.

Goldreich, O., S. Micali, and A. Wigderson. 1987. “How to Play Any Mental Game”, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing,* pp. 218-229.

Li, Lode. 2002. “Information Sharing in a Supply Chain with Horizontal Competition”. Lode Li, *Management Science* 48:9, pp. 1196-1212.

Schneier, B. 1995. *Applied Cryptography,* John Wiley & Sons, New York, New York

Yao, A. 1982“Protocols for Secure Computations”, Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science.