Risk-Aware Secure Supply Chain Master Planning

Axel Schröpfer[1], Florian Kerschbaum[1], Dagmar Sadkowiak[2] and Richard Pibernik[2]

[1] SAP Research Karlsruhe, Germany

[2] Supply Chain Management Institute, EBS Wiesbaden, Germany

Abstract. Supply chain master planning strives for optimally aligned production, warehousing and transportation decisions across a multiple number of partners. Its execution in practice is limited by business partners' reluctance to share their vital business data. Secure Multi-Party Computation can be used to make such collaborative computations privacy-preserving by applying cryptographic techniques. Thus, computation becomes acceptable in practice but for additional cost of complexity depending on the protection level chosen. Because not all data to be shared induces the same risk and demand for protection, we assess the risk of data elements individually and then apply an appropriate protection. This speeds up the secure computation and enables significant improvements in the supply chain.

1 Introduction

Supply chain master planning (SCMP) strives for optimally aligned production, warehousing and transportation decisions across multiple partners. In practice, we can commonly observe a decentralized coordination mechanism (referred to as upstream planning) that usually only leads to local optima rather than to global supply chain optima [9]. At least in theory, optimal master plans can be generated for the whole supply chain if some planning unit has at its disposal all relevant information pertinent to the individual partners in the supply chain. It is, however, a well known fact that companies are typically not willing to share sensitive private data (e.g. cost and capacity data) ([17, 18]). They perceive the risk that the central planning unit or other parties misuse data to their disadvantage in order to obtain additional benefits.

The major obstacles to centralized master planning can be removed if a mechanism for securely and privately computing the supply master plan is in place [1]. A central planning unit, e.g. a 4th party logistics provider (4PL), could then determine globally optimal master plans and distribute these to the individual partners involved in the supply chain. To this end, Secure (Multi-Party) Computation (SMC) can be employed such that the relevant data does not need to be disclosed even to central planning unit. This offers the ultimate level of protection, since no data sharing risk remains. In this paper we propose a framework for secure centralized supply chain master planning (SSCMP). We introduce a basic model for centralized supply chain master planning and, from this, derive the relevant data a central planning unit requires to optimally coordinate manufacturing and transportation decisions. We then analyse this data with respect to its

Schröpfer A., Kerschbaum F., Sadkowiak D. and Pibernik R. (2009).
Risk-Aware Secure Supply Chain Master Planning.
In *Proceedings of the 7th International Workshop on Security in Information Systems*, pages 46-56
DOI: 10.5220/0002197300460056

”criticality”. Criticality refers to how sensitive certain pieces of data are and how willing the different partners will be to share this data. The criticality is determined by the perceived risks associated with data sharing and its prior public knowledge. In this context, risk can be characterized by the potential negative impact that occurs if a partner misuses the data to its own benefit and the likelihood for this to happen. We derive an overall criticality assessment for each data element that is relevant for supply chain master planning and use information about on the prior (public) knowledge of the data to determine an overall criticality score. This criticality score constitutes an input to secure computation of centralized supply chain master plans. We map criticality scores to protection levels which consist of certain technologies and parameters for SMC. Lower protection levels lead to faster SMC implementations. We propose a mixed approach to SMC combining the different protection levels in one implementation and formulate a new pivot selection rule in Linear Programming (LP) that optimizes the effort involved by selecting based on the protection level. We experimentally verify the effectiveness of the new algorithm.

2 Related Work

Numerous works in the area of supply chain management exists on supply chain master planning as well as information sharing and collaboration in supply chains. In general, it is a well acknowledged fact that sharing relevant information and planning in a collaborative fashion can improve supply chain performance and mitigate the consequences of demand variability, especially with respect to the well-known bullwhip effect (see for example [6, 14, 16, 23]). With respect to supply chain master planning, numerous authors have proposed multi-stage models that can be utilized to coordinate planning activities across multiple locations and firms (e.g. [10, 13, 20]). Various authors have stated that employing a centralized approach to master planning will lead to better results as compared to decentralized approaches that are most commonly employed in industry. [21], for example, analyze the disadvantages of upstream coordination in comparison with centralized coordination. They compute the average gap between centralized and upstream coordination for several test scenarios with varying cost parameters and demand patterns. Similar findings are reported in [18]. However, centralized supply chain planning has not been widely adopted in industry. [12] states: ”it is difficult, or maybe even impossible, to get a large network consisting of independent companies to agree on and implement a centralised planning and control solution.” Reluctance towards information sharing (a prerequisite for centralized master planning) has been identified as the main obstacle that inhibits centralized master planning ([17, 18]). For this reason, alternative approaches have been developed that either build on negotiation based coordination ([9]) or hybrid forms ([17]). So far there has been no research on supply chain master planning based on mechanisms that privacy preserving data sharing and computation. To the best of our knowledge the only approach to secure multi-party computation in the area of supply chain management can be found in [1]. The authors develop secure protocols for a Collaborative Planning, Forecasting, and Replenishment (CPFR) process. Next to the fact that we, in our paper, consider a different problem setting, a major distinction between the research presented in [1] and our research is that they do not consider different protection levels for different risks of data to be shared. They follow the approach to provide the highest protection for all data using a specially developed protocol. Their protocols are two-party protocols, while we consider a multi-party problem. We will now review related work for SMC.

SMC allows a set of n players, $P = P_1, \ldots, P_n$, to jointly compute an arbitrary function of their private inputs, $f(x_1, \ldots, x_n)$. The computation is privacy preserving, i.e. nothing else is revealed to a player than what is inferable by his private input and the outcome of the function. A cryptographic protocol is then run between the players in order to carry out the computation. Even if there are adversarial players, the constraints on correctness and privacy can be proven to hold under well stated settings. These settings consider the type of adversary as well as his computing power which can be bounded or unbounded. An adversary can be passive, i.e. following the protocol correctly but trying to learn more or he can be active, by arbitrarily deviating. For the two-party case it has been proven by Yao in [22], that any arbitrary function is computable in privacy preserving fashion, using garbled binary circuits. This approach has been extended to the multi-party case in [4, 11]. Alternative approaches base on secret sharing schemes. A player's secret s is split into m shares which are then distributed to m players. Players can compute intermediate results on the shares, and in the end a reconstruction is performed in order to receive the final result. Other approaches utilize semantically secure homomorphic encryption (HE) [8], a public encryption scheme, where $E(x) \cdot E(y) = E(x + y)$ and x cannot be deduced by $E(x)$.

Using the general approach leads to solutions that have high complexity and are therefore almost always not practically feasible [15]. Thus, in order to get a practical solution, a dedicated protocol should be constructed. Atallah et al. constructed solutions for a couple of supply chain problems, e.g. planning, forecasting, replenishment, benchmarking, capacity allocation and e-auctions ([1–3]). Their cryptographic protocols base on additive secret sharing, homomorphic encryption and garbled circuits. A contribution of Atallah et al. which is closely related to ours is that of secure linear programming [15]. It uses the simplex method introduced by Dantzig in [7] to solve linear programs which get expressed as a matrix D. The method consists of two steps: selecting the pivot element d_{rs} and pivoting all elements d_{ij} of D over this element. The pivot step sets the new value of d_{ij}, denoted d'_{ij}, by

$$\begin{aligned}
d'_{ij} &= \frac{1}{d_{rs}} && for\ i = r\ and\ j = s\ (pivot\ element)\\
d'_{ij} &= \frac{d_{ij}}{d_{rs}} && for\ i = r\ and\ j \neq\ s\ (pivot\ row)\\
d'_{ij} &= \frac{-d_{ij}}{d_{rs}} && for\ i \neq\ r\ and\ j = s\ (pivot\ column)\\
d'_{ij} &= \frac{d_{ij} - d_{is} d_{rj}}{d_{rs}} && for\ i \neq\ r\ and\ j \neq\ s\ (all\ other\ elements).
\end{aligned}$$

The method is repeated until the optimal solution of the LP is found (resp., it is stated that the problem is unbounded or infeasible). As input to the cryptographic protocol, matrix D gets additively split between both parties (i.e., $D = D^{(a)} + D^{(b)}$). In order to not reveal additional information (e.g. by the pivot column or row index), the matrix gets permuted at the beginning of each iteration. Details are omitted here, but can be found in [15]. The pivot element selection and the pivot step are then carried out using cryptographic tools additive splitting, homomorphic encryption and garbled circuits.

3 Supply Chain Master Planning

In this section we first provide a basic model for centralized supply chain master planning. This model will be used to derive the relevant data that partners in the supply chain need to share for centralized master planning. We then propose a simple approach to assess the criticality of the individual elements.

3.1 Model for Centralized Supply Chain Master Planning

As a basis for our subsequent analysis we utilize a simple generic supply chain master planning model presented in [18]. Although rather simple, this model is sufficient for the illustration of our concept and can easily be extended in order to account for further practical requirements and restrictions.
We consider a supply chain with I stages on which different operations (e.g. manufacturing, warehousing, etc.) are performed. We use index i $(i = 1, \dots, I)$ to distinguish the different stages. By $I + 1$ we denote the final customer stage. By K_i we denote the set of nodes on stage i. Every node $k \in K_i$ represents one production facility or warehouse on stage $i = 1, \dots, I$. The final customer locations are modelled through nodes $k \in K_{I+1}$ on stage $i = I + 1$. By N_i we denote the set of products produced on stage i and use $m \in N_{i-1}$ and $n \in N_i$ as indices for the input and output products of stage i. For a given supply chain, master planning determines the production and inventory quantities for every node and the material flows between the nodes for a given time period. We introduce the following additional notation to formulate a centralized master planning model:

Master planning parameters (input)

D_l^n	Demand for final finished product $n \in N_I$ at customer location $l \in K_{I+1}$
$\alpha^{m,n}$	Quantity of input product m required for manufacturing one unit of output product n
β_k^n	Unit capacity requirement at location $k \in K_i$ for output of product $n \in N_i$
$cap_{i,k}$	Production capacity at location $k \in K_i$
$cp_{i,k}^n$	Unit production costs of product $n \in N_i$ at location $k \in K_i$
$cs_{i,k,l}^n$	Unit shipping costs of product $n \in N_i$ from location $k \in K_i$ to location $l \in K_{i+1}$
$ch_{i,k}^n$	Unit holding costs of product $n \in N_i$ at location $k \in K_i$

Master planning variables (output)

$x_{i,k}^n$	Production quantity of output product $n \in N_i$ manufactured at location $k \in K_i$
$y_{i,k,l}^n$	Shipping quantity of product $n \in N_i$ shipped from location $k \in K_i$ to $l \in K_{i+1}$

The following deterministic, linear programming model can be used to determine a supply chain master plan.

Objective function

$$\text{Min C} = \sum_{i=1}^{I} \sum_{k \in K_i} \sum_{n \in N_i} cp_{i,k}^n x_{i,k}^n + \sum_{i=1}^{I} \sum_{k \in K_i} \sum_{n \in N_i} ch_{i,k}^n x_{i,k}^n + \sum_{i=1}^{I} \sum_{k \in K_i} \sum_{l \in K_{i+1}} \sum_{n \in N_i} cs_{i,k,l}^n y_{i,k,l}^n \qquad (1)$$

Constraints

$$\sum_{k \in K_I} y^n_{I,k,l} = D^n_l \qquad \forall n \in N_I, l \in K_{I+1} \tag{2}$$

$$x^n_{i,k} = \sum_{l \in K_{i+1}} y^n_{i,k,l} \qquad \forall n \in N_i, i \in \{1, \ldots, I\}, k \in K_i \tag{3}$$

$$\sum_{j \in K_i - 1} y^m_{i,j,k} = \sum_{n \in N_i} \alpha^{m,n} x^n_{i,k} \qquad \forall m \in N_{i-1}, i \in \{1, \ldots, I\}, k \in K_i \tag{4}$$

$$\sum_{n \in N_i} \beta^n_k x^n_{i,k} \leq cap_{i,k} \qquad \forall i \in \{1, \ldots, I\}, k \in K_i \tag{5}$$

$$x^n_{i,k}, y^n_{i,k,l} \geq 0 \qquad \forall n \in N_i, i \in \{1, \ldots, I\}, k \in K_i \tag{6}$$

The objective of the model is to minimize the total relevant costs of the SC for fulfilling final customer demand. The objective function (1) accounts for production costs, holding costs, and shipping costs for finished products. Constraints (2) ensure that the final customer demand at stage $I + 1$ is met. (3) and (4) represent finished product and intermediate product balance constraints. The capacity constraints (5) ensure that the available capacity of any location will not be exceeded. Constraints (6) ensure non-negativity of all decision variables.

The output of this model is a supply chain master plan for a single period that specifies the production quantity for the individual products in each node and the shipping quantities across the whole supply chain. From this basic model we can directly infer the relevant data that needs to be shared in order to realize centralized supply chain master planning. All parties in the supply chain need to make the above listed *input parameters* available to the central planning unit. After generating the supply chain master plan, the central planning unit has to communicate the results (i.e. the values of the *output variables*) to the corresponding partners. In typical industry settings, both the input parameters and the master planning output constitute private data that is only accessible to the planning units (firms, departments) responsible for individual nodes and arcs. The willingness to share this data will depend on the risk the individual data owners perceive. The perceived risk, however, is not identical for all of the relevant data elements. A company may, for example perceive a low risk associated with sharing forecast data, but a high risk when revealing production cost or capacity information. While SMC can overcome the risk of data sharing in theory with the highest protection level, in practice such solutions can become too slow to be useful (e.g. if the computation takes longer than what the continuous planning period is). We therefore use the result of the risk assessment, the criticality scores, to optimize the SMC, such that each data element is handled at its appropriate risk level. We achieve a significant performance improvement in our experiments.

3.2 Data Criticality and Protection Levels

In this section we illustrate a simple approach to determine protection levels for individual data elements in the context of centralized master planning. Although it is rather straightforward to see that the risk will differ across the individual data elements, it is not possible to determine general criticality levels that are valid for any supply chain setting. Whether other partners in the supply chain can use data to their benefit and to the disadvantage of the data owner depends on factors such as the distribution of power among the partners, the type of industry and product, the relative position in the supply chain, trust among partners, etc. Production costs, for example are generally considered as critical data that a data owner will not want to share. However, in many industries (e.g. for commodities) production costs are known by different partners without implying a negative impact. Because a general assessment of the criticality is not likely to be attainable, an individual assessment has to be conducted for any specific supply chain. We propose a simple scheme to support such a criticality assessment. It is based on the following questions that need to be answered for any one of the data elements identified in the previous section:

1. What disadvantage may a data owner potentially incur when sharing private data?
2. What is the probability that a partner in the SC (mis-) uses the shared data to the disadvantage of the data owner?
3. To what extent is the data prior knowledge?

With the first two questions we capture the individual components of the risk induced by sharing a certain data element. When considering the potential negative impacts (question one), we have to consider that these may vary depending on the position of the data source within the supply chain and the potential incentives other partners in the SC may have to (mis-) use the data. We differentiate between partners who are responsible for nodes on the same stage (competitors) and those who are responsible for nodes on previous or subsequent stages (supplier-buyer-relationships). For each of the aforementioned cases it is necessary to assess the likelihood of a disadvantage on the side of the data owner, i.e. the probability that another partner in the supply chain will actually make use of the knowledge of the data element (question 2). The risk cannot be considered independent of the prior knowledge about the data. It is reasonable to assume that the criticality of certain data elements is lower if the data is already accessible for some or all of the partners in the supply chain. Figure 1, a) illustrates our basic scheme for assessing the criticality of individual data elements.

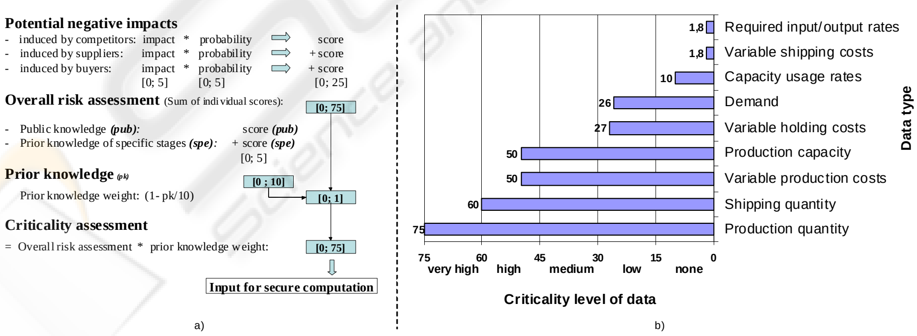
Potential negative impacts
- induced by competitors: impact * probability ⇨ score
- induced by suppliers: impact * probability ⇨ + score
- induced by buyers: impact * probability ⇨ + score
[0; 5] [0; 5] [0; 25]
Overall risk assessment (Sum of individual scores):
[0; 75]
- Public knowledge (pub): score (pub)
- Prior knowledge of specific stages (spe): + score (spe)
[0; 5]
Prior knowledge (pk)
[0 ; 10]
Prior knowledge weight: (1- pk/10)
[0; 1]
Criticality assessment
= Overall risk assessment * prior knowledge weight:
[0; 75]
Input for secure computation
a)
1,8
Required input/output rates
1,8
Variable shipping costs
10
Capacity usage rates
26
Demand
27
Variable holding costs
50
Production capacity
50
Variable production costs
60
Shipping quantity
75
Production quantity
Data type
75
60
45
30
15
0
very high
high
medium
low
none
Criticality level of data
b)

Fig. 1. a) Determination of criticality. b) Criticality levels of different types of data (example).

We propose a scoring range between zero and five to adequately assess by discrete values the potential negative impact and the expected probability of data misuse. Through multiplication of both scores, we obtain a particular risk measure for negative impacts induced by competitors, suppliers, or buyers. Their addition provides a measure for the overall risk. The overall risk for each data element is then weighted with a value that expresses the prior knowledge of data. Similarly, a scoring range from zero to five is used to measure the degree of public knowledge in general as well as specific knowledge of individual SC partners. The sum of both scores measures the level of prior knowledge. A score of zero indicates that the data is pertinent to the data owner, while higher scores indicate that the data may anyways be known prior to centralized master planning. We determine an aggregate weight for the prior knowledge as in order to derive the overall criticality level. In Figure 1,b) we provide an example of a possible outcome of our criticality assessment. With our assessment scheme, a criticality score between zero and 75 is assigned to each data element.

4 Secure Computation

4.1 Protection Levels

The data criticality analysis of section 3.2 shows that different variables in the SCMP problem have different protection demand. The data criticality scores of the variables range from zero to 75. We map a data criticality score to *protection levels*. A protection level specifies a concrete set of SMC technologies and their parameters for protecting a variable. These technologies and parameters are: the computational *setting* (information-theoretic, cryptographic or best-effort), the cryptographic *tools* and the *tool parameters*. Dependencies among the different parameters of a protection level are possible, e.g. there cannot be a SMC computation that is information-theoretically secure, but uses homomorphic encryption as a tool. The protection levels are arranged in order of the effort for an attacker to infer the protected value. Higher protection levels require more effort and add more additional complexity. Higher criticality scores map to higher protection levels. See Table 1 as an exemplary specification for protection levels which limits the available cryptographic tools to additive splitting and homomorphic encryption. Table 1 gives concrete examples for five possible protections levels which will be used in later experiments.

Table 1. Protection Levels for the 4PL Scenario.

Protection	Setting	Tools	Tool Parameters
very high	inf.-theo.	additive split w/ modulus	modulus N, number of parties
high	cryptographic	HE	Darmgard-Jurik, key: 3072 bit
medium	cryptographic	HE	Paillier, key: 512 bit
low	best effort	additive split w/o modulus	
none		-	-

4.2 Mapping

A monotone function maps the criticality score c to a protection level $p = f(c)$. We propose a linear mapping. Other mappings are possible and may depend on the conrecte application context. We assume the ordered protection levels to differ in their effort by an almost constant factor. For a first mapping we define a linear mapping function $f(c)$ which maps data criticality score c to m protection levels by $f(c) = 1 + \lfloor c \cdot m/(c_{max} + 1)\rfloor$, where c_{max} in our case is 75 as received from section 3.2. Applying this mapping to the criticality scores of section 3.2, we receive the values of Table 2. Considering Table 2, nine variables get a protection level assigned below the maximum. Thus, for nine of eleven variables computational effort can be reduced compared to the former approach of applying maximum protection.

Table 2. Linear Mapping.

Protection Level	1	2	3	4	5
Criticality Score	0-15	16-30	31-45	46-60	61-75
Number of Data	3	4	0	2	2

4.3 Pivot Rule/Intergration

We adapt the solution by Atallah et al. for secure linear programming. We introduce an additional matrix denoted P. Every element of P, p_{ij}, represents the protection level of the corresponding data element in D, d_{ij}. P is available to both parties, as well as the table with the protection level specifications. Each party may define its own mapping function. Whenever a pivot step is performed in order to receive a new value d'_{ij}, the new protection level value p'_{ij} is set to the highest assigned protection level value of all elements of D involved. According to the pivot step computation rules for processing the current element d_{ij} the involved elements are: d_{rs}, d_{rj}, d_{is} and d_{ij}. The new protection level for d'_{ij} then is received by $max(p_{rs}, p_{rj}, p_{is}, p_{ij})$. Over time, this leads to convergence of matrix P to the highest protection values contained. We construct a pivot selection rule which not only bases on entries of D but also on these of P and moreover prevents P from fast convergence. Recall that the LP is rewritten as a matrix D. Let

$$D = \begin{pmatrix} c^T & -z_0 \\ A & b \end{pmatrix}$$

where c^T denotes the vector of the objective function's coefficients, z_0 the outcome, A the coefficients of the constraints and b the vector of the constraint values. The secure linear programming solution originally uses a slight adaption of the Bland's Rule [5] as pricing scheme. The rule computes the pivot column s by $min(s : c_s < 0)$ and the pivot row r by $min(r : b_r/a_{rs})$ for all $a_{rs} > 0$. Our approach keeps the part of the Bland's Rule for selecting the pivot column. We then replace the part for selecting the row. We define r for $0 \leq r \leq m$ by

$$r = min(\frac{b_r}{a_{rs}}) : min\big(\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} max(p_{rs}, p_{is}, p_{rj}, p_{ij}) - p_{ij}\big).$$

Thus, every element in the selected pivot column s fulfilling the minimum ratio test best (i.e., no row in column s with a smaller ratio exists) is checked for having the lowest impact on the convergence of matrix P. Although every element of matrix P is involved in the computation for every element fulfilling the minimum ratio criteria, selecting the pivot row can still be considered very fast, since P gets updated by each party locally in the exact same manner and the single operations are less complex. Thus, even for big m's and n's, the added computational overhead can be considered very small. In order to have the indexes matching, P gets blinded and permuted in the same way as this is done for D within the original protocol. P may leak little information, e.g. if there is a unique occurrence of a protection level.

4.4 Experimental Results (Mapping, Pivot Rule)

For further examination, we set up an experiment based on the results of section 4.3 and the secure linear programming protocol of Atallah et al. For executing the experiment it is not necessary to actually run the cryptographic protocol, since the main part of the protocol remains unchanged. We rather focus on the local part, i.e. computing the elements of the protection level matrix, and run the simplex algorithm locally in order to simulate pivoting to have correct pricing data available. For preparation of the experiment, we implemented the simplex algorithm in Java, first. We then implemented an instance of a realistic 4PL scenario for medical equipment (details omitted for brevity) and derived a LP matrix from that which has 191 rows and 480 columns. Table 4 shows the results of our experiment using the linear mapping introduced in section 4.2. We receive the total effort by adding the number of assigned protection levels in order to simulate an execution of the pivot part of the cryptographic protocol. The measured values are the number of *pivot steps*, the numbers of steps until *convergence* and the *total effort*.

Table 3. Experimental Results.

	Bland's Rule	Modifier Bland's Rule
Pivot Steps	140	112
Convergence Step	2	99
Total Effort	51249448	32102112

Figure 2 shows the convergence ratio during the run. The convergence ratio is defined as the ratio of the sum of all entries of P divided by the number of elements of P multiplied with the maximum protection level (i.e., five).

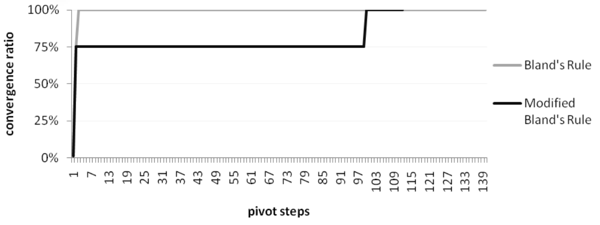

Fig. 2. Convergence Ratio.

The modification of the Bland's Rule led to a decrease of 20% on overall pivot steps. The protection level matrix was kept from convergence up to step 99 while the convergence ratio quickly reached a value of 0.75. The total effort added by the cryptographic protocol was reduced to 63%.

5 Conclusions

We introduced a solution for Secure Supply Chain Master Planning (SSCMP) using secure computation. Traditional SCMP computes the optimal production and transportation plan across a number of parties using Linear Programming. We showed that by risk assessment and risk handling a significant performance increase in SSCMP is possible. We derived a methodology for risk assessment, the criticality score, in supply chains and then modify the pricing scheme of Linear Programming handling each data item at the appropriate risk level. In an experimental study based on a realistic scenario using this methodology we obtained a performance gain of 37%. Future work is to extend the applicability of the method to other algorithms for linear optimization, e.g. inner point methods, and to extend it to other supply chain optimization problems adapting the risk assessment step.

References

1. M. Atallah, M. Blanton, V. Deshpande, K. Frikken, J. Li, and L. Schwarz. Secure Collaborative Planning, Forecasting, and Replenishment. Working Paper, Purdue University, 2005.
2. M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara. Private Collaborative Forecasting and Benchmarking. Workshop on Privacy in the Electronic Society (WPES), 2004.
3. M. Atallah, H. Elmongui, V. Deshpande, and L. Schwarz. Secure Supply-Chain Protocols. Proceedings of the IEEE International Conference on E-Commerce (CEC'03), 2003.
4. D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In Proceedings of 22nd STOC, 1990.
5. R. Bland. New finite pivoting rules for the simplex method. Math. of Op. Res. 2, 1977.
6. F. Chen, Z. Drezner, J. Ryan, and D. Simchi-Levi. The bullwhip-effect :managerial insights on the impact of forecasting and information on variability in a supply chain, in: Taylor, S., Ganeshan, R., and Magazine, M. (Eds.), Quantitative Models for Supply Chain Management, Boston 1999.
7. G. Dantzig, and M. Thapa. Linear Programming 1: Introduction. Springer-Verlag, 1997.
8. I. Damgård, and M. Jurik. A generalisation, a simplification and some applications of pailler's probabilistic public-key system. In International Workshop on Practice and Theory in Public Key Cryptography (PKC) 2001, 2001.
9. G. Dudek, and H. Stadtler. Negotiation-based collaborative planning between supply chain partners, in: European Journal of Operational Research 163, 2005.
10. B. Fleischmann, and H. Meyr. Planning Hierarchy, Modeling and Advanced Planning Systems, in: De Kok, A. G., Graves, S. C. (Eds.): Supply Chain Management: Design, Coordination and Operation, Handbooks in Operations Research and Management Science, Vol. 11, Amsterdam 2003.
11. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In Proceedings of the 19th annual ACM symposium on Theory of computing, 1987.

12. J. Holström, K. Främling, J. Tuomi, M. Krkkinen, and T. Ala-Risku. Implementing collaboration process networks, in: The International Journal of Logistics Management 13(2), 2002.
13. V. Jayaraman, and H. Pirkul. Planning and coordination of production and distribution facilities for multiple commodities, in: European Journal of Operational research, Vol. 133.
14. H. Lee, V. Padmanabhan, and S. Whang. Information distortion in a supply chain: the bullwhip effect, in: Management Science, Vol. 43, No. 4, 1997.
15. J. Li, and M. Atallah - Secure and Private Collabortive Linear Programming. 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2006.
16. S. Min, A. Roath, P. Daugherty, S. Genchev, H. Chen, A. Arndt, and R. Glenn Richey. Supply chain collaboration: what's happening?, in: The International Journal of Logistics Management, Vol. 16, No. 2, 2005.
17. R. Pibernik, and E. Sucky. Centralised and decentralised supply chain planning, in: International Journal of Integrated Supply Management 2(1/2), 2006.
18. R. Pibernik, and E. Sucky. An approach to inter-domain master planning in supply chains in: International Journal of Production Economics V. 108, 2007.
19. A. Shamir. How to share a secret. Communications of the ACM, 1979.
20. J. Shapiro. Modeling the Supply Chain, Pacific Grove 2001.
21. N. Simpson, and S. Erengüc. Modelling the order picking function in supply chain systems: formulation, experimentation, and insights, in: IIE Transaction 33(2), 2001.
22. A. Yao. Protocols for secure computations. In Proc. 23rd IEEE Symposium on the Foundations of Computer Science (FOCS), IEEE, 1982.
23. Z. Yu, H. Yan, and T. Cheng. Benefits of information sharing with supply chain management, in: Industrial Management and Data Systems, 2001.