Battling the Bullwhip Effect with Cryptography

Martin Hrušovský(✉) and Alfred Taudes

Institute for Production Management, WU Vienna University of Economics and Business, Vienna, Austria
martin.hrusovsky@wu.ac.at
https://www.wu.ac.at/en/prodmanengl

Abstract. In real-world supply chains it is often observed that orders placed with suppliers tend to fluctuate more than sales to customers and that this deviation builds up in the upstream direction of the supply chain. This bullwhip effect arises because local decision-making based on orders of the immediate customer leads to overreaction. Literature shows that supply chain wide sharing of order or inventory information can help to stabilize the system and reduce inventories and stockouts. However, sharing this information can make a stakeholder vulnerable in other areas like the bargaining over prices. To overcome this dilemma we propose the usage of cryptographic methods like secure multiparty computation or homomorphic encryption to compute and share average order/inventory levels without leaking of sensitive data of individual actors. Integrating this information into the stylized beer game supply chain model, we show that the bullwhip effect is reduced also under this limited information sharing. Besides presenting results regarding the savings in supply chain costs achieved, we describe how blockchain technology can be used to implement such a novel supply chain management system.

Keywords: Supply chain management · Secure multiparty computation · Homomorphic encryption · Bullwhip effect

1 Introduction

A supply chain (SC) is the entire process from the customer's order to the delivery and payment of a product or service. It involves different companies that collectively manage the flows of materials, information and funds. A key challenge of supply chain management (SCM) is the tension between the need for cooperation and the competition for the share of the total surplus. For instance, an end customer facing entity needs to know the inventory and capacities of upstream partners in order to be able to check the feasibility of a demand plan via supply network planning. However, a manufacturer might be reluctant to share this information, as his customer might demand lower prices when learning that the inventory has increased, signaling low sales and a weaker bargaining position.

G. Kotsis et al. (Eds.): DEXA 2022 Workshops, CCIS 1633, pp. 270–281, 2022.
https://doi.org/10.1007/978-3-031-14343-4_25

Thus, often the communication within the SC is reduced to the necessary minimum, i.e. passing of orders between different stages. However, this limited information gives rise to the so-called bullwhip effect: small fluctuations in retail demand cause progressively larger fluctuations in derived demand upstream at the other SC actors. This is similar to the effect of cracking a whip, where a small movement of the wrist causes the whip's wave patterns to amplify [1].

The reason for this phenomenon is that actors have only a partial understanding of demand but through their local actions influence the entire chain. One behavior causing the bullwhip effect is the extrapolation of demand trends and safety stock adjustment. To give an example, suppose a retailer sells 20 units of a product and keeps one period of demand as safety stock. If a permanent increase in demand occurs, now he sells 30 units. Then, instead of ordering 20 units from the wholesaler, he orders 40 units, 30 units for the newly forecasted demand, and 10 units to raise the safety stock to 30 units. Thus, a 50% increase in customer demand has led to a 100% increase of the derived demand. If the other actors employ the same policy, wholesaler orders 60 units, distributor orders 100 and manufacturer 180 units. If the retailer would share his sales figures, the other actors could separate the inventory adjustment effect and learn that they only have to increase their safety stocks to 30. However, the retailer might not be willing to share this data, e.g. fearing that he might not be serviced when capacities are tight and his suppliers infer that he has enough inventory.

Instead of sharing local data, a SC member might be willing to participate in the computation of average values if he can keep his local information private, though. Such methods are available in Secure Multiparty Computation (SMPC) or Homomorphic Encryption (HE). SMPC [2] provides protocols, where, by exchanging messages with each other, participants can learn the value of a function without revealing their inputs and relying on a trustworthy party. In a simplistic setting, the participants are organized in a ring, and a leader starts by adding a private random number to his input. Then he passes the result to his neighbor, who adds his local value to the input and passes the result to his neighbor. The process continues until the leader is contacted, who then subtracts his private random number and announces the average value based on the sum calculated.

HE [3] is an encryption scheme that allows a third party to compute with encrypted inputs while preserving the features of the function and format of the encrypted data. For instance, in the additively homomorphic scheme of [4], each of the stakeholders involved in the information sharing would encrypt their local value using a public key and send the cipher text to a coordinating entity. This entity would multiply these inputs and each entity could decrypt the resulting value using the private key to recover the sum and determine the average value.

Implementing such schemes is costly and should only be considered if they can significantly reduce the bullwhip effect. To study this question we adapt the beer game developed by the MIT Sloan School of Management to introduce students of SCM to the bullwhip effect [5]. This role-play simulation game simulates a beer SC using retailer, wholesaler, distributor and brewer, who manage their

inventories, production and backlogs, and communicate orders with the goal of minimizing operating cost. We will adapt this setting by incorporating information sharing of average orders and inventory levels to infer whether SMPC/HE should be employed to mitigate the bullwhip effect in a data parsimonious way.

2 Literature Review

[6] has first analyzed the bullwhip effect in the context of systems dynamics. A seminal work describing the phenomenon is [1], who identify demand signal processing, rationing game, order batching and price variations as sources of the bullwhip effect. [7] find that smoother forecasts and shorter lead times decrease the bullwhip effect. [8] find that inventory information helps upstream chain members to better anticipate and prepare for fluctuations in inventory needs downstream. [9] show that order-up-to replenishment policies lead to fluctuations and develop a decision rule that avoids amplifications.

Coordination of multiple actors in networks is a widely discussed topic in the literature with different approaches used to solve this problem. As an example, [10] use a mean field approach to coordinate charging of electric vehicles. In cases where independent agents need to be coordinated by sharing only limited information, decentralized techniques, such as the dynamic average consensus, can be used [11,12]. Another option is to use stochastic models to solve inventory network control problems, such as the queue clearing algorithm used by [13].

In SCM context, papers derived from the SecureSCM project [14] apply SMPC to supply network planning via linear programming (LP). They develop algorithms for solving LPs using a variant of the simplex algorithm and secure computation with fixed-point rational numbers based on secret sharing [15]. [16] describe a scheme for coordinating decentralized parties that share central resources but hold private information about their decision problems modeled as LPs, and [17] develop SMPC protocols for the Joint Economic Lot Size Model.

A number of papers describe ways of implementing information sharing via blockchains: [18] develop a blockchain architecture for general SC information sharing based on the codification of contractual relations and receiver-specific encryption, while [19] describe a shared ledger for a particular SC through which all relevant information is broadcasted. [20] enhance this design by adding a reputation system for tracking the truthfulness of the information sharing of a stakeholder and by considering these inputs when determining the validators in a PoA consensus mechanism.

In the case of HE, a permissioned blockchain operated by the SC members can execute a smart contract that collects the encrypted local values, performs the computation of the encrypted sum, checks whether all participants in the computation have taken part in the protocol and stores the decrypted average as common knowledge. Similarly, as described in [21], blockchain can be employed to enhance the efficiency and fairness of SMPC.

3 Modelling Approach

In our approach, we consider a SC consisting of several actors that are participating on a production and delivery of a product to the final customer. Although usually the bullwhip effect is modelled using two (see, e.g., [9,22]) or four actors (see, e.g., [8]), we present a SC containing six actors in order to be able to calculate averages and analyze the effect of proposed policies. Our SC starts with a retailer, who receives orders from the customer and is supplied by the distributor. The distributor receives goods from the manufacturer and the rest of the SC consists of three levels of suppliers: Tier 1, Tier 2 and Tier 3 suppliers, who deliver parts needed for the production. SC actors and the flows between them are shown in Fig. 1. Whereas the retailer receives exact demand from the final customer, all other actors only have the information about the orders from their direct predecessor in the SC which they need to fulfil. The information about orders on other stages or about the customer demand is not shared.

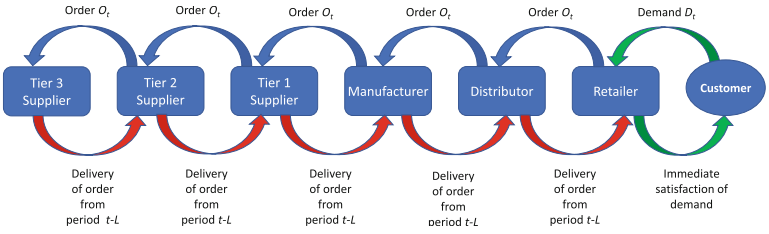

Fig. 1. Supply chain actors and flows between them

3.1 Basic Model

In the basic scenario, we assume that there is no information sharing and therefore only the orders of the predecessor are visible for each actor. Each SC actor uses the order-up-to inventory model, in which the following sequence of actions is observed in every ordering period: (1) at the beginning, a new order is placed, afterwards (2) the goods ordered in previous periods are received, (3) the demand is observed and satisfied, and (4) the end period inventory is counted. Similar to [23], the order quantity O_t in period t for each actor is calculated as:

$$O_t = S_t - S_{t-1} + D_{t-1} \tag{1}$$

for the retailer and

$$O_t = S_t - S_{t-1} + Q_{t-1} \tag{2}$$

for all other stages, where S_t denotes the order-up-to level in period t and D_{t-1}/Q_{t-1} represents the demand of the customer in case of the retailer or the quantity ordered by the predecessor from the last period for all the other stages.

The ordered quantities are delivered after a certain delivery time consisting of L periods. In order to cover the demand between the point when the order is placed and the moment when the goods are delivered, the respective order-up-to level should be chosen so that $L+1$ periods are covered [22], therefore,

$$S_t = (L+1) * \hat{D}_t \tag{3}$$

for the retailer and

$$S_t = (L+1) * \hat{Q}_t \tag{4}$$

for all other stages, where $\hat{D}_t/\hat{Q}_t$ is the expected customer's demand for the retailer or the expected order quantity from the predecessor for all the other stages. We consider two models for forecasting the expected demand/order quantity. Firstly, we use the moving average method [23], where the expected demand/order quantity is calculated as the average of observed values over the last n periods:

$$\hat{D}_t = \frac{1}{n} \sum_{i=1}^{n} D_{t-i} \tag{5}$$

$$\hat{Q}_t = \frac{1}{n} \sum_{i=1}^{n} Q_{t-i} \tag{6}$$

Alternatively, the expected demand/order quantity of the predecessor can be calculated using the simple exponential smoothing model [9] with smoothing parameter α (values between 0 and 1) where

$$\hat{D}_t = \alpha * D_{t-1} + (1-\alpha) * \hat{D}_{t-1} \tag{7}$$

$$\hat{Q}_t = \alpha * Q_{t-1} + (1-\alpha) * \hat{Q}_{t-1} \tag{8}$$

The inventory level is adjusted based on the received goods and the goods shipped to the preceeding stage. Unfulfilled demands/orders are considered as backlogs and will be fulfilled in the next period when sufficient goods are available. We also accept negative orders which represent returns of goods [24].

To summarize the logic of the model, each actor only receives information about the orders of its direct predecessor, which is the basis for calculating the expected demand/order quantity. Based on the expected demand/order, the new order-up-to level is calculated, which is together with the previous order-up-to level and the current order of the predecessor relevant for deciding about the order quantity in the current period that is then given to the next actor in the chain as input for his decision about the order-up-to level and the order quantity.

3.2 Considered Policies

In order to investigate the possibilities of using SMPC/HE in this setting, we extend the demand/order signal received by each actor by additional information about the orders of other preceeding SC actors. For this, we consider two policies

that might help to mitigate the bullwhip effect: Firstly, we adjust the expected demand by the changes in the average SC inventory and, secondly, we combine the expected demand with the average order quantities in the SC. In this way, each SC stage receives additional information about changes in the demand that might be already present in the preceeding stages, but, in the basic case, they would be observed at the given stage only later due to the delays caused by delivery times. In this context, it is necessary to point out that we assume that all parties are acting honestly according to the assumptions of the model so that correct signal about the average order quantity is received by each SC actor.

Policy 1 - Average Inventory Changes: In this case, we extend the average expected demand/order quantity used for the calculation of S_t by the information about the changes in the average inventory level of all predecessors in the SC. Since the actors are usually reluctant to share the exact information, we use SMPC/HE to build an average inventory level at the end of each period. As there are at least two values needed to build an average, this policy is only applied to the manufacturer and suppliers. Therefore, the sum of inventory levels of retailer and distributor divided by two is reported as the average inventory $\bar{I}_t$ in period t to the manufacturer, Tier 1 supplier gets the inventory information based on the inventory levels of retailer, distributor and manufacturer and this continues up to Tier 3 supplier, who receives inventory information as the average of inventory levels of all five predecessors. Then the expected order quantity and the inventory level are combined in the following way:

$$\hat{UQ}_t = \beta * \hat{Q}_t + (1 - \beta) * (\bar{I}_{t-1} - \bar{I}_{t-2}) \tag{9}$$

where $\hat{UQ}_t$ is the updated expected order quantity that replaces $\hat{Q}_t$ in Eq. 4 and β is a smoothing factor with values between 0 and 1. We decided to use the changes in inventory level instead of the inventory level itself since the inventory level might be much higher than the relevant demand or orders.

Policy 2 - Average Orders: This policy is very similar to the previous one, but the difference of inventory levels is replaced by average orders of all predecessors. We again start with the manufacturer, who has direct access to the orders of the distributor, but does not know what the real demand of the customer and the order of the retailer were. Therefore these two values are again summed together and divided by two to build the average order quantity $\bar{Q}_t$ in period t. Tier 1 supplier receives the average of the order values from the customer, distributor and manufacturer and this again continues up to Tier 3 supplier who receives the exact order of Tier 2 supplier and the average orders based on all five previous SC stages. The updated expected order quantity is therefore calculated as

$$\hat{UQ}_t = \beta * \hat{Q}_t + (1 - \beta) * (\bar{Q}_{t-1}) \tag{10}$$

3.3 Model Implementation

The SC is built in form of a simulation model which allows to easily compare different demand patterns. Since each actor makes decisions independently based on the inputs coming from its predecessor (order quantity) as well as its successor (received goods), we chose to model each SC stage as an own agent in an agent-based simulation approach [25].

The model was created using Anylogic 7.2.0 University [26]. This simulation software allows to build agent-based models as well as to define different scenarios and experiments and to build graphs and export the results. Each actor is defined as an agent that follows its own logic, which is illustrated in Fig. 2 that shows the behavior of manufacturer in form of statecharts and transitions.

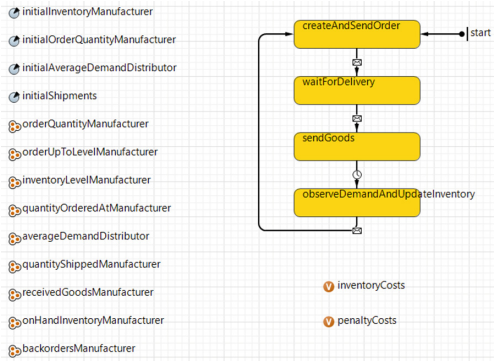

Fig. 2. Behaviour of manufacturer in Anylogic

As it can be observed, the manufacturer starts with creating an order that is then sent to Tier 1 supplier. Afterwards he waits for the delivery of goods ordered L periods before and together with receiving the goods from the supplier he also sends the goods to the distributor. At the end, he observes the demand from the distributor and updates the inventory level. The transitions are triggered by messages sent between the different agents. In addition to that, various parameters showing initial values for, e.g., inventory or average demand, can be defined. Moreover, multiple variables recording the values for, e.g., inventory level, received and shipped goods, demand, backlogs etc. were defined to be able to observe differences between models. The performance of the models is based on total SC costs which consist of inventory costs for each unit left in inventory and penalty costs for each unit of backlogs at the end of each period.

4 Case Study and Results

The case study used for the analysis of the proposed policies is based on the popular "beer game", described by e.g. [5]. In the original version of this game,

each player represents one actor, including retailer, wholesaler, distributor and manufacturer. The retailer observes demand from the customer and orders goods from the wholesaler, who only receives the order information from the retailer and orders from the distributor without knowing the real demand of the customer. The delivery of the goods is delayed due to the time needed to prepare and ship the order. This works similar for the other stages. Due to the delays and missing information sharing, a small change in the demand of the customer creates the bullwhip effect within the whole SC.

In our case we use the initial data from [5]: we start with initial inventory of 12 units and planned deliveries of 4 units at each SC stage. The demand in the first periods is constant and amounts to 4 units. Based on the order-up-to model presented in Sect. 3.1, the order quantity is also stabilized in the first periods at 4 units. Delivery time L is 2 periods, inventory costs are 0.50 EUR/unit per period and penalty costs are 1 EUR/unit per period at each stage. We run the model for 50 periods and observe the total costs at the end of this time horizon.

After stabilizing the system in the first five periods, customer's demand increases to 8 units starting with period 6 and stays the same until the end of the game. This change in demand causes high fluctuations in orders and inventory levels on the different SC stages. This is illustrated in Fig. 3, where the inventory levels of all stages are shown when the basic model with moving average method from Eqs. 5 and 6 with $n = 5$ is used. As Fig. 3 shows, the inventories react with a slow decrease followed by a sharp increase and another sharp decrease. At the end there are some minor fluctuations until the inventory stabilizes after ca. 30 periods. The total costs after 50 periods are 3,984.5 EUR.

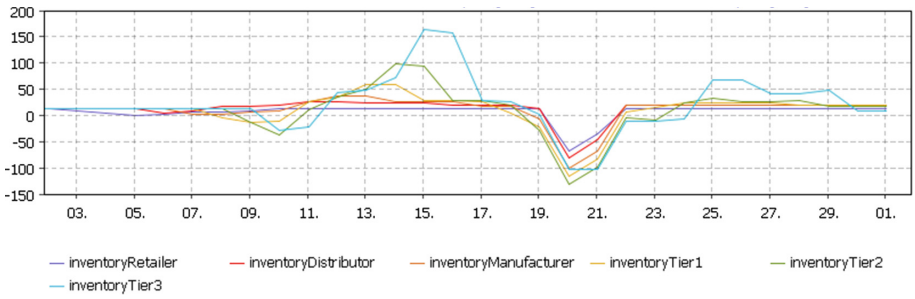

Fig. 3. Inventory levels for basic model with moving average ($n = 5$)

In comparison to that, if the basic model is combined with, e.g., Policy 2 (Average orders) with $\beta = 0.5$, the picture substantially changes, as it is shown in Fig. 4. Due to the additional information used, the fluctuations are flattened and the resulting total costs decrease by 33% to 2,661.5 EUR.

However, the flattening of the curves is not always the case, as our initial results for the two applied policies show. Especially in case of Policy 1 (see Table 1) the inclusion of the inventory differences does not help, but has a rather

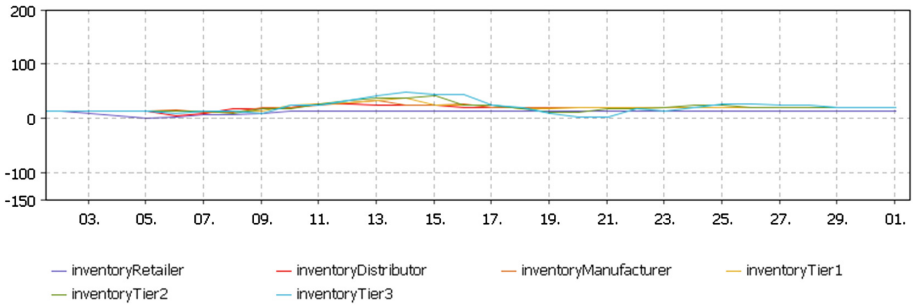

Fig. 4. Inventory levels for basic model with moving average ($n = 5$) combined with Policy 2 where $\beta = 0.5$

negative effect - due to the fact that the fluctuations of the inventories are very high, they even amplify their effect on the average expected demand/orders and therefore the resulting total costs are soaring. Only in cases where β is high and, thus, the inventories have a very low weight in the calculation, it might have positive impact on the total costs for some of the instances.

Table 1. Comparison of total costs for Policy 1

Basic model with	Basic total costs	Policy 1 - total costs and change with					
		$\beta = 0.1$	Change	$\beta = 0.5$	Change	$\beta = 0.9$	Change
Moving average $n = 3$	14,166.5	3.4e7	2.3e5%	2.6e5	1,700%	341,391	2,310%
Moving average $n = 5$	3,984.5	2.3e7	5.9e5%	10,211	156%	3,915	−2%
Exp. smoothing $\alpha = 0.1$	2,417.5	3.3e7	1.4e6%	33,485.5	1,285%	1,684.5	−30%
Exp. smoothing $\alpha = 0.5$	6,796	6.9e7	1.0e6%	3.6e8	5.3e6%	955,144.5	13,954%
Exp. smoothing $\alpha = 0.9$	192,425	4.1e8	2.1e5%	1.7e12	8.8e8%	3.5e11	1.8e8%

A completely different picture can be seen in case of Policy 2 in Table 2. Here the total costs are improved for 4 of the 5 considered models, with improvements ranging between 17% and 99%. These high savings can be achieved by considering the average orders in SC which offer an additional signal about the changes that might be happening at the preceeding stages. The reason why the costs are slightly higher in case of the exponential smoothing model with $\alpha = 0.1$ might be that due to the low α-value a lot of demand smoothing already happens in the basic model and the average orders do not bring much additional value.

Table 2. Comparison of total costs for Policy 2

Basic model with	Basic total costs	Policy 2 - total costs and change with					
		$\beta = 0.1$	Change	$\beta = 0.5$	Change	$\beta = 0.9$	Change
Moving average $n = 3$	14,166.5	2,634.5	−81%	3,291	−77%	10,500.5	−26%
Moving average $n = 5$	3,984.5	2,618	−34%	2,661.5	−33%	3,326.5	−17%
Exp. smoothing $\alpha = 0.1$	2,417.5	2,526.5	5%	2,529.5	5%	2,484.5	3%
Exp. smoothing $\alpha = 0.5$	6,796	2,625.5	−61%	2,832	−58%	5,017.5	−26%
Exp. smoothing $\alpha = 0.9$	192,425	2,691.5	−99%	25,533	−87%	140,919.5	−27%

When looking at the changes in costs between the basic case and Policy 2, it can be seen that including the additional information significantly reduces the penalty costs at all stages. For Policy 2 with β-factors 0.1 and 0.5, penalty costs are reduced by 90–100%, whereas in case of $\beta = 0.9$ the reductions are lower, but still significant, ranging between 27% and 52%. This results from the fact that the upstream SC actors can observe the increasing average demand in the downstream stages and prepare earlier for the increasing order quantities. Once the higher order is placed by the predecessor, there is already sufficient inventory available to satisfy that order and, thus, backlogs are eliminated. As a consequence, inventory costs at the downstream stages, especially for the retailer, distributor, manufacturer and Tier 1 supplier, are slightly increasing, on average by 4–18%. However, this increase is overcompensated by the savings in penalty costs and therefore every SC actor can reduce its overall costs, as illustrated in Table 3. Here the average savings in total costs over all demand forecasting models for each actor are shown if Policy 2 is employed. The highest savings are achieved by the Tier 2 and Tier 3 suppliers, who also had the highest total costs, and then by the retailer, who now has sufficient inventory and can therefore better serve customers. Relative savings of the actors in the middle of the SC are slightly lower, but still significant. However, the presented results only represent first investigations of the model and policies and therefore more detailed analysis about the influence of the different factors will be needed in the future.

Table 3. Average costs savings for each SC actor for Policy 2

Policy 2 with	Average total cost change over all demand forecasting models for					
	Retailer	Distributor	Manufacturer	Tier 1 sup.	Tier 2 sup.	Tier 3 sup.
$\beta = 0.1$	−52.35%	−46.73%	−46.89%	−51.29%	−56.81%	−61.16%
$\beta = 0.5$	−49.81%	−44.28%	−43.87%	−47.12%	−51.84%	−55.54%
$\beta = 0.9$	−41.61%	−36.67%	−37.09%	−39.87%	−43.98%	−46.41%

5 Conclusion

Sharing information between the different SC actors is important to avoid fluctuations in inventory levels defined as bullwhip effect. However, practice shows that the actors are reluctant to share any additional information due to various reasons and, therefore, only the minimum necessary information is shared. In order to contribute to the solution of this problem, we proposed to avoid direct information sharing and only calculate average values for inventory or orders using SMPC/HE techniques. As our first results show, this method might have some potential in mitigating the bullwhip effect, but further investigations regarding the best policies are necessary. Moreover, extensions of the model to more complex supply chain networks as well as investigations of additional demand models and scenarios are necessary to increase the robustness of the results. These points should be tackled in further research.

References

1. Lee, H.L., Padmanabhan, V., Whang, S.: Information distortion in a supply chain: the bullwhip effect. Manage. Sci. **43**(4), 546–558 (1997)
2. Cramer, R., Damgard, I.B., Nielsen, J.B.: Secure Multiparty Computation and Secret Sharing. Cambridge University Press, New York (2015)
3. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. ACM Comput. Surv. **51**(4), 1–35 (2018)
4. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
5. Sterman, J.D.: Teaching takes off - flight simulators for management education - "The Beer Game". OR/MS Today, pp. 40–44 (October 1992)
6. Forrester, J.W.: Industrial dynamics - a major breakthrough for decision makers. Harv. Bus. Rev. **36**(4), 37–66 (1958)
7. Chen, F., Drezner, Z., Ryan, J.K., Simchi-Levi, D.: Quantifying the bullwhip effect in a simple supply chain: the impact of forecasting, lead times, and information. Manage. Sci. **46**(3), 436–443 (2000)
8. Croson, R., Donohue, K.: Upstream versus downstream information and its impact on the bullwhip effect. Syst. Dyn. Rev. **21**(3), 249–260 (2005)
9. Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R.: Measuring and avoiding the bullwhip effect: a control theoretic approach. Eur. J. Oper. Res. **147**, 567–590 (2003)
10. Zhu, Z., Lambotharan, S., Chin, W.H., Fan, Z.: A mean field game theoretic approach to electric vehicles charging. IEEE Access **4**, 3501–3510 (2016)
11. Yang, P., Freeman, R.A., Lynch, K.M.: Multi-agent coordination by decentralized estimation and control. IEEE Trans. Autom. Control **53**(11), 2480–2496 (2008)
12. Kia, S.S., Van Scoy, B., Cortes, J., Freeman, R.A., Lynch, K.M., Martinez, S.: Tutorial on dynamic average consensus: the problem, its applications, and the algorithms. IEEE Control Syst. Mag. **39**(3), 40–72 (2019)
13. Zargham, M., Ribeiro, A., Jadbabaie, A.: Discounted integral priority routing for data networks. In: 2014 IEEE Global Communications Conference (2014)

14. Kerschbaum, F., et al.: Secure collaborative supply-chain management. Computer **44**(9), 38–43 (2011)
15. Catrina, O., de Hoogh, S.: Secure multiparty linear programming using fixed-point arithmetic. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 134–150. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15497-3_9
16. Albrecht, M., Stadtler, H.: Coordinating decentralized linear programs by exchange of primal information. Eur. J. Oper. Res. **247**(3), 788–796 (2015)
17. Pibernik, R., Zhang, Y., Kerschbaum, F., Schroepfer, A.: Secure collaborative supply chain planning and inverse optimization - the JELS model. Eur. J. Oper. Res. **208**(1), 75–85 (2011)
18. van Engelenburg, S., Janssen, M., Klievink, B.: A blockchain architecture for reducing the bullwhip effect. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 69–82. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_5
19. Ghode, D.J., Yadav, V., Jain, R., Soni, G.: Lassoing the bullwhip effect by applying blockchain to supply chains. J. Glob. Oper. Strateg. Sourcing **15**(1), 96–114 (2022)
20. Sarfaraz, A., Chakrabortty, R.K., Essam, D.L.: A blockchain-coordinated supply chain to minimize bullwhip effect with an enhanced trust consensus algorithm. Preprints (2021)
21. Zhong, H., Sang, Y., Zhang, Y., Xi, Z.: Secure multi-party computation on blockchain: an overview. In: Shen, H., Sang, Y. (eds.) PAAP 2019. CCIS, vol. 1163, pp. 452–460. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2767-8_40
22. Lee, H.L., Kut, C.S., Tang, C.S.: The value of information sharing in a two-level supply chain. Manage. Sci. **46**(5), 626–643 (2000)
23. Damiani, E., Frati, F., Tchokpon, R.: The role of information sharing in supply chain management: the secure SCM approach. Int. J. Innov. Technol. Manag. **8**(3), 455–467 (2013)
24. Chatfield, D.C., Kim, J.G., Harrison, T.P., Hayy, J.C.: The bullwhip effect - impact of stochastic lead time, information quality, and information sharing: a simulation study. Prod. Oper. Manage. **13**(4), 340–353 (2004)
25. Borshchev, A.: The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6. AnyLogic North America, Chicago (2013)
26. AnyLogic: AnyLogic Timeline - Anylogic 7.2. https://www.anylogic.com/features/timeline/. Accessed 13 Apr 2022