POMS PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT

Vol. 0, No. 0, xxxx–xxxx 2021, pp. 1–21
ISSN 1059-1478 | EISSN 1937-5956 | 21 | 00 | 0001

DOI 10.1111/poms.13356

Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutorial

Zhiyuan Wang
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, 200030, China, bigbigwzy@sjtu.edu.cn

Zhiqiang (Eric) Zheng*
Jindal School of Management, University of Texas at Dallas, Richardson, Texas 75080, USA, ericz@utdallas.edu

Wei Jiang
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, 200030, China, jiangwei@sjtu.edu.cn

Shaojie Tang
Jindal School of Management, University of Texas at Dallas, Richardson, Texas 75080, USA, tangshaojie@gmail.com

Data sharing between upstream and downstream entities is vital for the success of a supply chain. However, distrust, privacy concerns, data misuse, and the asymmetric valuation of shared data between entities often hinder data sharing. This problem calls for a secure, efficient, fair, and trustworthy data-sharing mechanism. The key to such a successful system hinges on how to trace the data usage, determine the value of the seller's data to the buyer and then compensate the seller accordingly. To this end, we design and implement a blockchain-enabled data-sharing marketplace for a stylized supply chain. We demonstrate how a blockchain can be used to overcome these impediments in supply-chain data sharing and provide a detailed tutorial with a step-by-step implementation for how to set up such a data exchange prototype using Hashgraph.

Key words: supply-chain data sharing; blockchain; usage-based data valuation
History: Received: July 2018; Accepted: November 2020 by Subodha Kumar, after 2 revisions.

1. Introduction

In a supply chain, data sharing between upstream and downstream parties is often necessary because of the asymmetry of the private information they possess (Cachon and Fisher 2000). For example, downstream retailers are usually in a better position to detect market trends, anticipate which products are going to break out, and predict when product replenishment needs to be addressed in advance. Meanwhile, upstream suppliers (e.g., manufacturers) have an information advantage in terms of product quality and quantity to supply. Such proprietary information, if properly shared, can reduce inventories and lower operating costs for all supply chain members (Lee et al. 1997). For example, a supplier is able to anticipate market fluctuation and adjust its inventory level when the demand (sales) data are shared by the retailer, mitigating the bullwhip effect (Lee et al. 1997). However, if supply chain members are unwilling to share data, information asymmetry and supply-demand mismatch may lead to lost sales, double marginalization and customer dissatisfaction (Kumar et al. 2018).

In practice, many firms have embraced various information technology systems, such as advanced planning systems (APSs) and collaborative planning, forecasting, and replenishment (CPFR) systems, to facilitate information sharing. For example, Walmart and Sara Lee jointly implemented a CPFR system with the goal of enabling smooth data sharing and integration. The involved parties reported an increase in sales of 32% within 24 weeks of implementation (Kurtuluş 2017). Other leading retailers, such as Target, Costco, and Best Buy, have all reported gains from data sharing with their respective suppliers, such as Procter & Gamble, Kimberly-Clark, and Samsung (Ha et al. 2011). In addition to sharing information directly with partners, acquiring data from third-party vendors is another common choice. Popular data providers, such as IRI and Nielsen, have established their business models around data collection, processing, and consulting services. For example, during the COVID-19 pandemic, IRI launched a special project to share sales and out-of-stock data with manufacturers in the consumer packaged goods

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

industry (https://www.iriworldwide.com/en-au/insights/publications/coronavirus).

Despite extensive benefits of information sharing, a systematic implementation of information sharing across the board is still lacking in practice. In an investigation of 120 firms, approximately 42% of the respondents viewed lack of trust as one of the biggest obstacles impeding information sharing (Seifert 2003). According to the survey of PRG (2013), 84% of 111 companies claimed to embrace a formal strategy for supplier relationship management, but only 1 in 6 of these companies actually implemented an information-sharing program. Similarly, Forrester Research reports that only 27% of retailers shared point-of-sale data (Shang et al. 2015). What has led to such discrepancies in information sharing in reality? In reviewing the large body of data-sharing studies in the operations management (OM) literature, we identify several key challenges that obstruct data sharing in a supply chain.

The first challenge is the concern surrounding information leakage. For example, a retailer sharing its sales data with its supplier fears that the supplier may leak the information to the retailer's competitors. According to a survey conducted by supplychainaccess.com, 64% of supply chain managers ranked information leakage as the number one threat to supply chain operations. For this reason, Walmart stopped sharing its sales information with Inc.com and AC Nielson (Hays 2004). Adewole (2005) also reports that retailers in the UK clothing industry are reluctant to share their sales information with suppliers for fear of these suppliers leaking such private information. Anand and Goyal (2009) even argue that a supplier may benefit from intentionally leaking the sales information obtained from an informed retailer to an uninformed one under a wholesale-price regime. In view of these concerns, some studies have proposed the design of complex contracts to prevent information leakage in information sharing (e.g., Özer and Wei 2006 and Ren et al. 2010). Liu et al. (2020) investigate the use of revenue-sharing contracts to prevent information leakage in a centralized supply chain system. However, their analyses focus on designing contracts and incentives (e.g., through pricing) to effectively disincentivize information leakage rather than designing a leak-proof data-sharing system directly (e.g., using blockchain techniques), as explored in this study.

The second challenge is how to fairly attribute and distribute the value generated from shared data to the data provider. Shih et al. (2015) point out that the value attribution problem is a main barrier preventing supply chain members from collaborating. Surprisingly, very few information-sharing studies have attempted to address the value attribution and distribution problem. Most research in this field has focused on understanding the extent to which data sharing can help improve system efficiency, with the implicit assumption that retailers are willing to share their inventory levels or sales data without demanding any financial compensation. However, this may not be realistic. Data sharing may come at the expense of the data sharer. Mishra et al. (2009) discuss the dilemma in which a retailer sharing its demand forecast benefits its upstream manufacturer but at the cost of its own profit. Ghoshal et al. (2018) also report a case in which the use of shared data may lead to a reduced profit of the sharer. Zhang (2006) suggests that an improperly compensated supplier may refuse to share its information. In practice, this challenge undermines supply chain partners' willingness to share data. Najjar and Kettinger (2013) caution the prevalence of data monetization without proper compensation of data sharers, using the supply chain examples of US-based Fortune 500 drug retailers. Similarly, Munves (2013) raises the concern on packaged goods manufacturers, such as Pepsico, Johnson&Johnson, and P&G, for monetizing the data purchased from their retailers, without sharing the accrued benefits with these retailers. However, to what extent data providers should be compensated has seldom been studied in the literature.

The third challenge is the lack of trust and consensus among data-sharing parties on the quality and value of shared data. Concerns that data providers may even act opportunistically by sharing misleading data have also been raised. For example, in the personal computer industry, semiconductor manufacturers order on average 30% less than the forecasted number they share with their suppliers to secure more component capacity (Cohen et al. 2003). Such biased information sharing has even been observed within the same firm. Scheele et al. (2018) report a case in which the sales division of a global pharmaceutical company intentionally shared a 3-month demand forecast with an average of 16.2% inflation (compared to the actual number) with the operations division of the same company. Acquiring data from third-party data providers leads to the same predicament. Bimpikis et al. (2019) show that it is not optimal for a third-party provider to sell undistorted information to customers when these customers are competitors. Addressing this challenge calls for a new data valuation and pricing mechanism to fairly determine the value of shared data. In current practice, data are often sold at a fixed price (inaccurately) preset by the data owner. Such a pricing procedure does not always reflect the usage value to different data consumers. For example, IRI's basic product, the "Market Advantage Solution," includes a summary of industry sales and a consulting service, is priced at 10,000 US Dollar,

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

regardless of the buyer (Bimpikis et al. 2019). In the few studies that have considered payments between data buyers and sellers (e.g., Ha et al. 2017, Shang et al. 2015, Zhang 2002), the price of the data has conveniently been assumed to be exogenous, bypassing the necessary step of determining the value of the data.

In short, successful information sharing in supply chains requires a systematic solution in which (a) information-sharing parties have a consensus regarding the quality and value of the shared data based on the actual usage of the data, (b) the value of the shared data can be fairly distributed between the data provider and data user, and (c) the data-sharing procedure is leak-proof.

In this study, we propose a blockchain-based approach to address these three challenges to create a transparent, trustworthy, and fair supply chain data-sharing mechanism. Our solution is to build a data exchange using blockchain equipped with a novel usage-based data valuation and pricing mechanism to facilitate data sharing in a supply chain. We demonstrate how to implement such an exchange using Hashgraph, one of the leading blockchain backbone platforms, which represents the next generation of blockchain technologies in terms of speed and security.

2. Related Work

We contribute to the supply chain literature on the importance and inadequacy of data sharing in supply chain management (SCM). Lee et al. (1997) discuss how information sharing can be the key to mitigating the bullwhip effect. They provide ample evidence of how firms in a supply chain benefit from sharing demand and forecast information. Zhang (2006) shows that allowing suppliers to share their inventory statuses helps coordinate a supply chain in a two-echelon assembly system. Ha and Tong (2008), Ha et al. (2011), and Yue and Liu (2006) examine how retailers and suppliers mutually benefit when the former share their demand data with the latter. Jeong and Leon (2012) consider the case in which adjacent members along a supply chain share the selling price, manufacturing cost, and safety stock level. Demirezen et al. (2016, 2018) explore data sharing across partnering firms and propose a model to optimize the value accrued from such coordination. Anand and Goyal (2009), Kong et al. (2013), and Liu et al. (2020) study the implications of information leakage in supply chains under different contract structures. Their equilibrium outcomes highlight the efficiency reduction resulting from information leakage. Özer et al. (2011) and Bimpikis et al. (2019) investigate how information quality plays an important role in information sharing. They show that inflated and imprecise forecast sharing induces a biased operational decision. We enrich this literature by addressing the three aforementioned challenges identified, but not solved, by this body of previous work.

This study is also related to the growing theoretical literature on data valuation. Among this stream of studies, Gallego and Moon (1993) evaluate the expected value of demand information in a newsvendor problem in which the true demand is normally distributed. Yue et al. (2006) compute the upper bound of the expected value when the true demand distribution is known (hypothetically). Zheng and Padmanabhan (2006) examine a data-acquisition scheme with the goal of using the acquired data to minimize the posterior variance in the parameters. Xiang and Sarvary (2013) and Bimpikis et al. (2019) consider a market for information with competition on both the demand and supply sides of the market. Strong et al. (2015) propose a way to quantify the expected value of sample information to enhance model prediction performance. Han et al. (2017) develop an online pricing scheme to incentivize users who arrive sequentially. The pricing scheme dynamically adjusts the posted prices for a heterogeneous dataset by learning from data buyers' behaviors. Mehta et al. (2019) hold that a price quantity schedule, in which the price for a dataset only depends on the number of the data records but not on the content of the data records, is an optimal data-selling strategy. Their results are established on the assumption that it is impractical to specify a set-based pricing schedule due to its exponential complexity. In contrast, we introduce a novel usage-based pricing scheme that can determine the specific value of data based on how the data are used. As a result, we contribute to this literature by determining the data value before the act of data sharing.

We also draw on the nascent literature exploring blockchain applications in supply chains. Casey (2017), Babich and Hilary (2020), and Blossey et al. (2019) provide detailed discussions on how blockchain can improve supply chain efficacy with regard to procurement, production, data aggregation, and automation. Blockchain provides a new means for SCM. Chod et al. (2020) examine how to use blockchain to ensure transaction verifiability for small- and medium-sized enterprises to transmit quality information. They design a protocol called *b_verify* running on the Bitcoin blockchain to provide the transparency needed in supply chains. Zaerens (2018) proposes the use of smart contracts to monitor the information quality and trading balance between participants in a military supply procurement context. Jiang et al. (2018) develop BlocHIE, a blockchain-based healthcare information exchange to integrate off-chain data to help patients better manage their personal

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

healthcare data. In a similar vein, we operationalize our design of supply chain data sharing using Hashgraph. We provide a detailed tutorial in the appendix that elaborates the step-by-step setup of our blockchain platform.

Another related stream of work is cryptography for privacy preservation and accountability. Loshin (2013) and Meijer (2016) provide a comprehensive review of the theory and practice of relevant encryption techniques. Hosseinzadeh et al. (2016) explain how obfuscation and decentralization can help protect data and algorithms from leaking. To achieve secure data trading, Felici et al. (2013) and Jung et al. (2018) propose an accountability-oriented conceptual model to ensure proper data governance. In response to the potential breach of privacy, Sasson et al. (2014) introduce the zero-knowledge proof technique to mask private information, such as user addresses and transaction amounts. Feng et al. (2019) propose a group-signature scheme in which multiple transactions are synthesized into one transaction, which conceals the exact identity of the transactor. We draw on these ideas towards building a secure, accountable, and privacy-preserving data marketplace.

3. Designing a Supply Chain Data Exchange

A data exchange facilitates data sharing between data consumers and data providers and creates value for both through enhanced market efficiency, resource allocation efficiency, and matching between supply and demand (Soh et al. 2006). For example, Azure Marketplace and Infochimps help small data producers sell data through their platforms. Some data marketplaces, such as Factual and Personal, focus on providing a data marketplace for specific fields; Factual specializes in location-based data sharing, whereas Personal facilitates the exchange of personal data (Li and Miklau 2012). Other state-of-the-art data marketplaces include Intelligent IoT Integrator, Terbine, Streamr, and IOTA. Koutroumpis et al. (2017) provide a detailed review of existing projects on building data marketplace and highlight that a successful marketplace should offer effective trading functionalities, such as pricing, contracting, and settling. We propose such a blockchain-enabled data exchange to address the challenges faced by data sharing in a supply chain, with a focus on data valuation and pricing.

3.1. Data Valuation and Pricing

A key component of a data exchange is its price discovery mechanism. A variety of possible pricing schemes exist (e.g., Mehta et al. 2019). Popular schemes include flat fee, quantity-based, and query-based fee structures, as summarized in Table 1.

However, these conventional pricing schemes may not adequately accommodate the needs of data sharing in a supply chain. The key challenge here is the lack of mutual agreement with respect to the value of the data being shared, especially before knowing how the data are going to be used and what added value the data are going to bring to beneficiaries. We maintain that *the valuation of data cannot be isolated from the usage of the data*. Depending on how data are to be used, by different buyers or by the same buyer at different times, the same data can yield different (usage) values. For example, JPMorgan estimated that each unique user's data are worth different amounts for different companies because these companies use the data for different purposes. The value ranges from approximately $4 for Facebook to $24 for Google (Brustein 2012). In a supply chain, the same sales data may have different meanings when they are used for sales arbitrage as opposed to being used to help a firm optimize its inventory level. This suggests that proper data valuation should (a) specify (or at least anticipate) how the data are to be used before they are shared, (b) track how the data are being used after sharing, and (c) capture and measure the value that the data have created for the buyer.

Accordingly, we propose a new data-valuation and data-pricing mechanism, termed *usage-based valuation*, where the data to be shared are valued and priced based on their value expected to be generated according to the specific usage of the data. To our knowledge, this represents the first attempt in the SCM literature to address data sharing from the perspective of usage-based valuation.

We depict the architecture of our proposed data exchange in Figure 1. The entire trade process is delineated by the time at which the data value is realized. Along the timeline, the data exchange embodies three unique functionalities: screening, usage-based

Table 1 Summary of Data Pricing Schemes

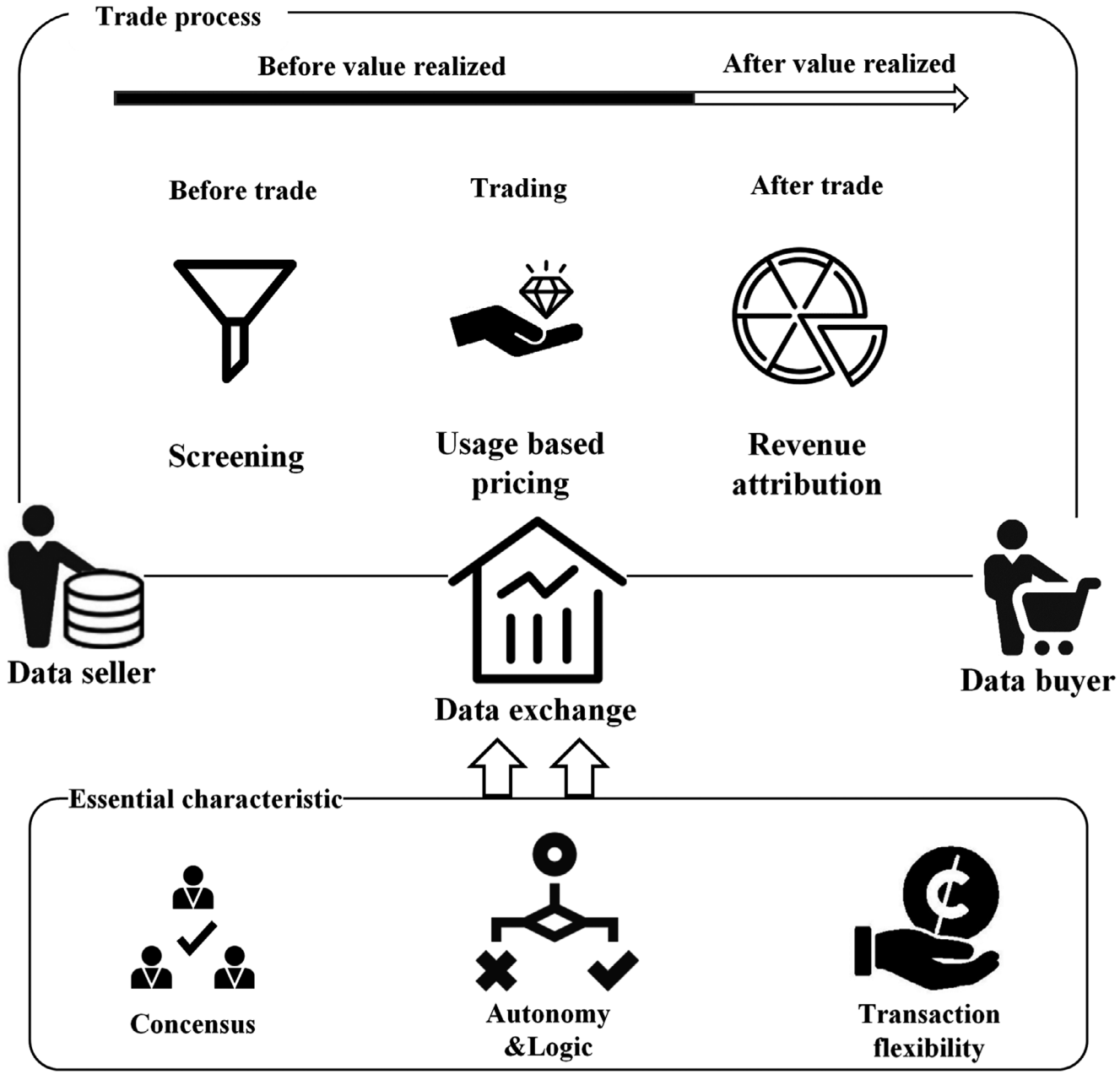
Pricing mode	Descriptions	Example
Flat fee	Dataset-based pricing: data buyer can only buy the entire dataset, pricing is at the dataset level.	SalesLead http://sales-lead.org/
Quantity based	Record-based pricing: the price depends on how many records the data have.	BookYourData https://www.bookyourdata.com/
Query based	Attribute-based pricing: the buyer can issue query for the needed data. The price depends on which attributes of the data are to be used.	DirectMail https://www.directmail.com/

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

pricing, and value (revenue) attribution (and distribution accordingly).

Both the screening and pricing are based on the value expected to be generated from using the seller's data (i.e., usage-based data valuation). The screening module of the data exchange matches the data buyer with the most valuable data seller (i.e., that whose data yield the highest expected benefits for the buyer). On behalf of the buyer, the data exchange then posts the price (according to the expected value) to the data seller through the module of usage-based pricing. For example, suppose that a budget-constrained supplier (buyer) can only afford to buy a demand forecast from one of the five candidate retailers (sellers) for the purpose of optimizing its inventory level. Without being able to analyze the data (before the data are shared), the retailer does not know which seller's data are of the highest value before consuming them. The data exchange in this case acts as a trustable third party that helps determine the value to be created by the data (which we elaborate on in the next section) and then sets the initial price accordingly. The actual realized value may deviate from the initial set price. In the third module, for value attribution and distribution, the data exchange tracks and verifies the revenue (or loss) generated from using the data and then

Figure 1 A New Data Exchange Structure



Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

compensates the buyer and seller according to the terms prespecified by the two parties using the smart contract (we show how to specify such a smart contract in the appendix). This way, the final settlement is only completed after the actual use of the shared data is accomplished and the usage value is realized and observed. In the current design, the process of actual data sharing is securely executed by the data exchange and the plaintext of the shared data is not passed to the buyer, thus mitigating the information leakage concern.

Our data-usage-based valuation builds on but extends the related literature, which typically measures the added value of the data through demand forecast enhancement or demand uncertainty reduction. We summarize the key differences between this study and the literature in Table 2, which shows that we are the first to address the three challenges of data leakage, data valuation, and value attribution (and the corresponding distribution of the attributed value).

To fulfill the modules specified in Figure 1, we embrace the blockchain technologies that accommodate consensus, autonomy, and micropayments. The transactions going through a blockchain-based data exchange are secure, autonomous, and trustless. The buyer does not need to worry about not receiving the designated data from the retailer and the seller does not need to worry about not receiving an agreed-upon amount of compensation. This requires transparency in terms of how data are to be used and how their value is to be determined and distributed, which in turn requires consensus among all of the parties involved. The transactions need to be run autonomously according to preset clauses (logic) in the smart contract. The data exchange may also have to deal with transactions with very granular data that may be worth only a minuscule amount of money. This requires the data exchange to facilitate flexible micropayments.

In view of these requirements, we introduce Hashgraph blockchain as our chosen backbone technology for the data exchange.

3.2. Blockchain

Blockchain's immutable digital ledgers enable verifiability of transactions across a peer-to-peer network, achieving four key objectives: (a) security, (b) transparency (auditability), (c) low transaction cost, and (d) automation (typically via smart contracts). Emerging blockchain technologies provide new means for businesses to validate, store, and access data in a distributed ledger rather than in a centralized legacy system. Blockchain has become one of the strategic focuses of many high-tech firms, such as Facebook (Libra), IBM (Hyperledger), and Microsoft (Azure). Blockchain has also penetrated and, to a large extent, disrupted the supply chain industry. Blockchain enables the restructuring of business ecosystems by facilitating the trustless integration of capital, material, and information flow in supply chains (Babich and Hilary 2020, Kumar et al. 2018). Golden State Foods, one of the largest suppliers providing services to restaurants, has deployed blockchain (using Fabric) to track, trace, and monitor its food supply. In 2018, Nike, Macy's, and Kohl's teamed up to launch the Chain Integration Pilot (CHIP) project. Since then, the CHIP project has successfully uploaded 223,036 serialized data points of goods (through a distributed ledger), which enables partners to share these data (https://cointelegraph.com/news/nike-explores-blockchain-for-supply- chain-data-collection). The World Trade Organization's recent white paper enumerated the success stories of more than 20 blockchain-enabled supply chain projects (https://www.wto.org/english/res_e/booksp_e/blockchainrev19_e.pdf.

Bitcoin and Ethereum are the two most popular blockchain platforms. Bitcoin, which heralded the emergence of blockchain, was invented by Satoshi Nakamoto, using digital signatures and the proof-of-

Table 2 Comparison with the Related Literature

	Anti-leakage		Value attribution and distribution			
	Contract	IT	Side payment	Usage based	Ex post	Data valuation
1. Anand and Goyal (2009), Ren et al. (2010), Chen and Özer (2019), Liu et al. (2020)	√					
2. Gallego and Moon (1993), Cachon and Fisher (2000), Zheng and Padmanabhan (2006), Yue et al. (2006), Strong et al. (2015)						√
3. Zhang (2002), Mishra et al. (2009), Najjar and Kettinger (2013), Shang et al. (2015), Ha et al. (2017), Bimpikis et al. (2019)			√			√
4. Nakasumi (2017), Jiang et al. (2018), Zaerens (2018), Chod et al. (2020)		√				
5. This study		√	√	√	√	√

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

work consensus mechanism to prevent the double-spend problem (Nakamoto 2019). Ethereum is considered to be the second generation of blockchain, the main added feature of which is smart contracts, whereby the terms (conditions) of a contract are specified and guaranteed to be executed once these conditions are met. We show how to use smart contracts to facilitate data sharing in the appendix.

The main impediment to the widespread application of Bitcoin and Ethereum is their low transaction throughput, typically measured by transactions per second (TPS). On average, Bitcoin reaches a capacity of 7 TPS, whereas Ethereum attains 20 TPS. In contrast, PayPal deals with approximately 200 TPS on average, whereas Visa routinely handles 56,000 TPS. In addition, a high transaction fee makes them unsuitable for our data-exchange scenario. The cost of a Bitcoin transaction can be as high as $40 (Chod et al. 2020).

Recently, several new blockchain platforms, such as Hashgraph, Dfinity, and Tangle, have attempted to resolve the low TPS bottleneck by using a directed acyclic graph (DAG) technique. A DAG relinquishes the chain structure on which Bitcoin and Ethereum rely, thus enabling parallel computing (rather than the serial computing imposed by the chain structure). The graph structure makes it possible for the nodes to communicate with other nodes in an asynchronous manner without having to wait for the confirmation of all previous blocks. A DAG is able to balance the trilemma among decentralization, security, and scalability well. Among the DAG-based blockchain techniques, we embrace Hashgraph because it supports the three crucial properties required for a data exchange: a fast consensus mechanism, smart contracts, and micropayments. At the same time, its transaction fee remains at the level of fractions of a cent.

Consensus mechanism: Hashgraph's consensus builds on the asynchronous byzantine-fault tolerance mechanism and the gossip protocol, where each member (node) randomly communicates and syncs with others. In this way, a node may or may not share the same information (in contrast, every full node of Bitcoin contains exactly the same data). However, every node can infer exactly which data are contained in another node in the network, ensuring finality. A brief introduction to the gossip protocol is given in the appendix, and details are available in the Hashgraph white paper (http://hedera-hashgraph.s3.amazonaws.com/hh-whitepaper-v1.0-180313-2.pdf).

Smart contract: A smart contract is a computerized contract that integrates verification and execution of a contract. The conditions (terms) of a smart contract are specified and coded; once the conditions are met and verified, transactions are automatically executed. Any programmable business logic (business rules) can be specified and coded into a smart contract (i.e., Turing completeness according to Buterin 2014). That is, one can create and write codes, commands, and decentralized applications on a blockchain to feature any business rules for ownership, transaction formats, and state transition functions. Once the code has been added to the blockchain, the smart contract cannot be altered, stopped, or forgone. Smart contracts differ from ordinary contracts in the three aspects detailed below (Swan 2015, p.16).

1. Autonomy: Once a smart contract is launched and run, it does not need to be in further contact with its initial agent.
2. Self-sufficiency: A smart contract has the ability to independently marshal any resource designated to it.
3. Decentralization: Smart contracts are registered into the blockchain and are distributed and self-executed across a wide network of nodes.

Smart contracts provide a reliable, secure, and convenient approach to specifying an agreement, which is essential for data sharing. For example, a smart contract can prespecify the penalty if a term is breached (e.g., when a retailer exaggerates its demand estimation) and automatically enforce it. Hashgraph uses Ethereum's Solidity as its smart contract programming language. Currently, smart contracts do not support complex mathematical computation and heavy computing is normally conducted outside a smart contract. The immutability nature of smart contracts can challenge the safety of blockchain, such as in the case of code bugs in the smart contract. To overcome this issue, the data exchange proposes to only undertake tested and legally binding contracts. The contract code will be fully tested according to code regulation, such as Zeus (Kalra et al. 2018) before it is uploaded to the blockchain.

Micropayments: As the value accrued from a piece of shared data may be minuscule, micropayments are necessary. A micropayment is an online financial transaction that involves a very small amount of money. Under a traditional payment system (e.g., credit card payments or money wiring services), dealing with transactions that are only worth pennies or less is unimaginable. However, Hashgraph readily accommodates transactions worth tiny fractions of a penny (see, e.g., Hearo.fm, a global music marketplace allows micropayments between artists and fans using Hashgraph: https://www.hearo.fm/). Micropayments present unprecedented opportunities to innovate SCM. For example, Microsoft's IOTA facilitates machine-to-machine micropayments that can be integrated into a supply chain with a network of

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

electronic devices (e.g., ElaadNL M2M car charging station: https://www.elaad.nl/).

Although we implement our data exchange in the state-of-the-art Hashgraph, our design of the data exchange is generic and is not limited to the Hashgraph platform. The data exchange can be implemented with any other platforms that support these three crucial characteristics.

4. Usage-Based Data Sharing

The data-sharing scheme we propose bears several distinct characteristics, as detailed below.

1. *Trust*: We assume that the data exchange and the nodes in the data exchange can be trusted, enabled by the blockchain. Blockchain techniques (e.g., cryptography and smart contracts) ensure that the data and processes going through the data exchange are immutable and transparent. Having a trustworthy third-party platform is necessary to guarantee the privacy and safety of the shared data, especially when the data buyer and seller need to agree on the usage and value of the data. In Section 5, we develop several mechanisms to guarantee the trustworthiness of the proposed data exchange through decentralization and encryption.
2. *Data and transmission security*: To ensure data security, the data are encrypted before they are transacted. Common encryption methods include SHA256, DES, RSA, and Paillier encryption. The encrypted data are in the form of a hash. The data buyer sees the hash[1] but does not need to have access to the raw data. The data seller and buyer interact with the data exchange via a private transaction channel specified by smart contracts, which is similar to the process on the Ethereum-based platform Quorum. This secure channel ensures that the exact content and status of the transaction are only visible to permitted network participants (whose public keys are generated before the transaction). The other nodes in the blockchain network can only observe the transaction and the updates of all of the contracts but not the actual content of the data. The transaction and contract modification records are hashed and are immutable but easily verifiable. This helps the data exchange to achieve accountability by preventing dishonest behaviors such as forging data or tampering with the data usage.
3. *Pseudonymity of identity*: To protect the privacy of buyers and sellers, we use the public and private key technique common in blockchain. Each user is assigned a pair of keys. The keys represent the user but anonymize her identity. The public key corresponds to the user account (blockchain address) and the user must use the private key to digitally sign each transaction. A digital signature is required to verify the user's ownership of the data. In so doing, the real identity of the user is concealed through these cryptography techniques. Thus, both the seller and the buyer are protected by preventing the leakage of the seller's data and the buyer's data usage.
4. *Active buyer*: Without loss of generality, we assume that a buyer initiates a transaction by seeking data from a seller to optimize its inventory. Our mechanism can be readily adapted to allow the seller to initiate the transaction. The buyer shares its inventory model with the data exchange, which then evaluates the value of the seller's data on the buyer's behalf. The buyer trusts the exchange's valuation and always accepts the price of the data determined by the data exchange.

Next, we demonstrate how to use a blockchain to facilitate data sharing in a supply chain under two scenarios: direct data sharing and indirect data sharing.

In the first scenario, we assume that the retailer is willing to share data (e.g., order quantity or sales forecast) that are critical for the supplier to optimize its inventory level. This mimics the common information-sharing mechanism considered in the SCM literature. In this case, the supplier acts as the data buyer to purchase data from retailers (data sellers).

In the second scenario, we consider a realistic case in which some retailers are not willing to share their data directly. In this case, the data exchange acts as a marketplace to help obtain data from other sources, in a manner similar to "crowdsourcing." As an illustration, suppose that the supplier wants to obtain more accurate predictions on demand distribution. Although the exact demand data are not directly available from the retailer, we assume that there exists a crowd of expert analysts (analogous to the financial analysts of a specific industry in the securities market) who may be able to provide information that can help forecast the retailer's demands more accurately. Note that the first scenario can be a special case of this more general case, in which the analysts themselves are retailers. We show how a blockchain can facilitate data sharing in the second scenario.

In the following sections, we use the terms **data**, **model**, and **decision** to refer to shared data (information) from the data seller, the buyer's inventory optimization model (that consumes the data), and the buyer's decision (on the inventory level), respectively.

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

4.1. Setup of the Data-Sharing Scenarios

We consider the classic data-sharing setting in SCM, where a supplier needs to optimize its inventory level by estimating the demands of multiple retailers (i.e., the total demand is a random variable).

Without loss of generality, we illustrate a case in which a retailer sells its demand forecast to the supplier. The demand forecast can be interpreted as a random draw from the retailer's true underlying demand distribution. The supplier uses this information to update its prior belief on the expected demand quantity to calibrate its inventory decision (see Figure 2).

For simplicity, we assume that the downstream retailers are independent (i.e., their future order quantities are not correlated). We can then simplify the data-sharing scenario by considering a single supplier and a single retailer, one at a time. Finally, we operationalize the model used by the supplier to be the classic newsvendor model as the underlying inventory management model. However, it should be noted that our data-sharing mechanism is generic and is not restricted to the specific model choice. To further simplify the illustration of the data-exchange design, we assume that the supplier focuses on optimizing the current period's inventory level. Zipkin (2000) shows that for many inventory decisions (e.g., perishable goods with a shelf life), a myopic strategy that optimizes the current period's inventory (rather than the long-term inventory level) is optimal.

The supplier's inventory decision can then be modeled as follows:

$$\min_q G(q) \equiv c_u E(x-q)^+ + c_o E(q-x)^+ \tag{1}$$

where q is the order quantity, and x is the demand of the retailer. The notations c_u and c_o represent the underage and overage costs, respectively. In addition, t^+ denotes $max(t,0)$ and $E(\cdot)$ is the expected value operator. At an arbitrary period T, the supplier faces an unknown demand X, the true distribution of which is private to the retailer. The supplier estimates the distribution of X using a normal distribution $N(\mu,\sigma^2)$, the mean parameter of which is unknown and follows $N(\tilde{\mu},\tau^2)$. The common probability density of X given $\mu = m$ is $f(x|\mu = m)$. The variance σ^2 of the common probability is assumed to be known for simplicity[2]. In accordance with a Bayesian perspective, the prior distribution represents the initial beliefs about the mean parameter. The retailer's underlying demand distribution is $X \sim N(\mu_0,\sigma^2)$. In each period, the retailer makes a prediction y regarding its demand quantity which is viewed as a random draw from the true demand distribution.

1. Newsvendor. The newsvendor model is used to characterize the supplier's cost optimization problem.
- q: The order quantity decision made by the supplier.
- c_o: The overage cost, the unit cost of overbooking.
- c_u: The underage cost, the unit cost of lost sales.
- X: A random variable representing a single demand observation.
- x: A single realization of X.
- Y: The samples prior to data collection. It is a random variable.
- y: The realization of the samples.

2. Demand distributions. Demand is described as a normal random variable.
- $N(\mu,\sigma)$: X, i.i.d, follows a normal distribution.
- σ: The variance of the unknown demand, assumed to be known.
- μ: The mean parameter of X, which is a random variable.
- m: A realization of the mean parameter μ.
- μ_0: The true mean parameter of the demand distribution.

3. Hyperdistribution. The mean parameter μ is treated as a sample from a probability distribution. This distribution is referred to as a hyperdistribution.
- $N(\tilde{\mu},\tau^2)$: Prior distribution $\pi\pi$ of the expected value μ. Demand distributions are generated from the hyperdistribution.
- π_Y: A posterior hyperdistribution, evolving by $\pi\pi$ with the information Y.
- τ: The variance of the prior distribution demand assumed to be known.

4.2. Direct Data Sharing

In the scenario of direct data sharing, we assume that the retailer is willing to share its demand information with the supplier directly. Examples of demand information include the planned order quantity and the covariates of the order quantity, such as a sales forecast, planned sales force, or discount plan in the next selling period, which are critical in helping suppliers reduce uncertainty about future demand (Bourland et al. 1996, Cachon and Fisher 2000, Gavirneni et al. 1999). The data exchange facilitates two transaction modes: the prepay mode (Figure 3) and the spot mode (Figure 4).

4.2.1 Prepay Mode. In practice, firms must often commit to ex ante information sharing and make prepayments before the information is shared (e.g., Ha et al. 2017). We consider such a prepay mode, in which the data buyer has to prepay a security deposit for the data *before the seller shares the data*, analogous to earnest money in real estate transactions. The prepayment amount is commensurate with the potential value of the data. In this section, we present a method through which the data exchange can fairly determine the data value given the supplier's prior belief and the current status with regard to the level of demand uncertainty.

We use the newsvendor cost function as the *loss function* for a single demand realization x in the following form:

$$l(x,q) = c_o \cdot [q-x]^+ + c_u \cdot [x-q]^+ \tag{2}$$

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

where q is the order decision made by the supplier and x is the actual demand realization.

The supplier's prior belief (on the demand distribution) is denoted as π. Thus, given the mean of the demand distribution m and the order quantity q, the Bayes risk $R(m,q)$ is defined as the expected loss:

$$R(m,q) = E_{X|\mu=m}[l(x,q)] = \int f_m(x) l(x,q) dx \tag{3}$$

The risk should be evaluated on the mean level over every possible μ. Thus the average risk of choosing q as the order quantity under the prior π is

$$\begin{aligned} \bar{R}(\pi,q) &= E_\mu[R(\mu,q)] \\ &= \int f_{\tilde{\mu}}(m) \int f_m(x) l(x,q) dx dm \end{aligned} \tag{4}$$

with $\mu \sim \pi$. Under the above specified demand distribution, the average risk becomes

$$\begin{aligned} \bar{R}(\pi,q) = & \int [(q-m)(c_o+c_u) Z(\frac{q-m}{\sigma}) - c_u(q-m) \\ & + (c_o+c_u)\frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(q-m)^2}{2\sigma^2}}] f_{\tilde{\mu}}(m) dm \end{aligned} \tag{5}$$

The corresponding Bayesian solution for the current prior π is

$$q^*(\pi) = \arg\min_q \bar{R}(\pi,q) \tag{6}$$

To solve the optimization problem, we examine the first derivative of $\bar{R}(\pi,q)$ with respect to q and obtain:

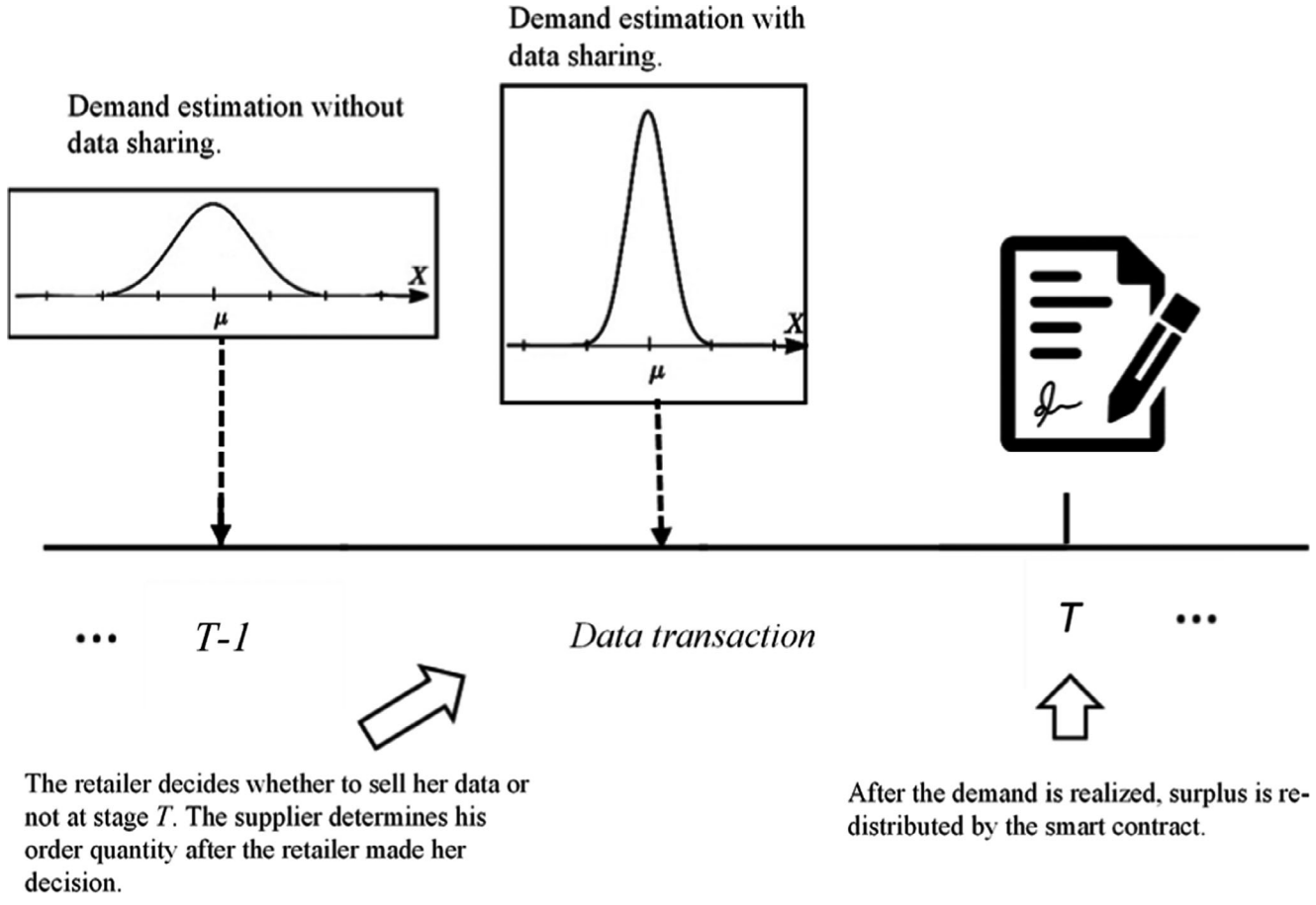
$$\begin{aligned} \frac{\partial \bar{R}(\pi,q)}{\partial q} &= \int [(c_o+c_u) \int_{-\infty}^{q} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-m)^2}{2\sigma^2}} dx - c_u] f_{\tilde{\mu}}(m) dm \\ &= (c_o+c_u) \int_{-\infty}^{q} \int \frac{1}{2\pi\sigma\tau} e^{-\frac{(m-\frac{z\tau^2+\tilde{\mu}\sigma^2}{\sigma^2+\tau^2})^2}{\frac{2\sigma^2\tau^2}{\sigma^2+\tau^2}}} e^{-\frac{(z-\tilde{\mu})^2}{2(\sigma^2+\tau^2)}} dm dz - c_u \end{aligned} \tag{7}$$

Setting $\frac{\partial \bar{R}(\pi,q)}{\partial q} = 0$ yields the following optimal decision for minimizing the Bayes risk:

$$q^*(\pi) = F_\pi^{-1}(\frac{c_u}{c_o+c_u}) \tag{8}$$

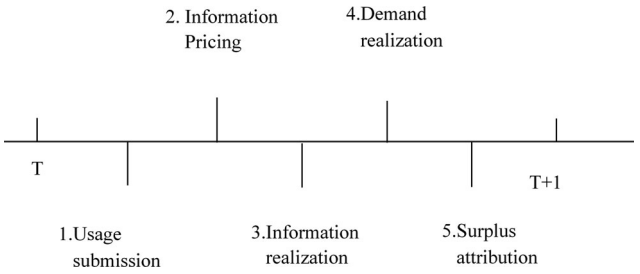
where F_π^{-1} denotes the inverse cumulative distribution function of $N(\tilde{\mu}, \sigma^2+\tau^2)$. Note that F_π^{-1} is equal to the classic newsvendor solution under the

Figure 2 A Graphic Representation of the Transaction Over the Time Horizon



Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Figure 3 A Graphic Representation of the Prepaid Mode Transaction Horizon



predictive demand distribution with posterior π. The predictive distribution summarizes the information value that a new observation brings in, given the likelihood and the prior (Carlin and Louis 2009, p.26). The Bayes risk for π is $\bar{R}(\pi,q^*(\pi))$. The risk reduction attributable to the new data quantifies the value of the data.

It helps to understand how we determine the value of the data by considering two types of suppliers with identical priors. The first type of suppliers, the "uninformed" suppliers, makes their order decisions without the data shared by the retailer. These suppliers serve as the benchmark. The second type of suppliers, the "informed" suppliers, acquires the data y to refine their decisions. Ideally, the value should be determined by comparing against the true demand distribution. Imagining there is an oracle that knows the true demand distribution $N(\mu_0,\sigma^2)$. From the oracle's perspective, the risk for the uninformed supplier with prior π is $R(\mu_0,q^*(\pi))$. For the informed supplier, the risk calculation can be summarized into the following steps:

1. Use the data y from the retailer and apply the Bayes rule to update the supplier's prior π to the posterior π_Y.
2. Under the posterior π_Y, calculate the corresponding optimal order decision $q^*(\pi_Y)$.
3. Calculate the risk under the true demand distribution and the Bayes decision $q^*(\pi_Y)$:$R(\mu_0,q^*(\pi_Y))$.

As Y is unknown to the supplier and the data exchange before the transaction, we need to integrate over every possible value of Y that the retailer may sell to the supplier under the true distribution:

$$E_{Y|\mu=\mu_0}R(\mu_0,q^*(\pi_Y)) = \int f_{\mu_0}(y)R(\mu_0,q^*(\pi_Y))dy \qquad (9)$$

We can then evaluate the value of the data sample Y before data sharing, by comparing the risk reduction between the uninformed supplier and the informed supplier:

$$V(\mu,\pi) = R(\mu_0,q^*(\pi)) - E_{Y|\mu=\mu_0}R(\mu_0,q^*(\pi_Y)) \qquad (10)$$

In the absence of the oracle (i.e., the true value of the mean parameter μ_0 is unknown), the value of data Y before its acquisition relies on the beliefs of what value μ_0 may be. Notice that the prior π is the only information that can be utilized. Therefore, we treat μ_0 as a random variable with the prior distribution and calculate the expected value of sample information (EVSI) as follows:

$$\begin{aligned} V(\pi) &= E_\mu V(\mu_0,\pi) \\ &= \bar{R}(\pi,q^*(\pi)) - E_\mu E_{Y|\mu}R(\mu,q^*(\pi_Y)) \\ &= \bar{R}(\pi,q^*(\pi)) - \\ &\quad \int f_{\tilde{\mu}}(\mu)\int f_\mu(y)\int f_\mu(x)l(x,q^*(\pi_Y))dxdyd\mu \end{aligned} \qquad (11)$$

Due to the challenge of deriving a closed-form solution of $V(\pi)$, we use Monte Carlo simulation to obtain an approximation. The algorithm embeds three levels of loops, as presented below.

Algorithm 1 The Monte Carlo approach to calculate EVSI

Input:loss function l, common distribution $N(\mu,\sigma)$,prior distribution $N(\tilde{\mu},\tau)$
Output: EVSI
for $i = 1$ to n_1 **do**
  $\mu_i \leftarrow$ draw a random sample from the prior distribution$N(\tilde{\mu},\tau^2)$
  for $j = 1$ to n_2 **do**
    $y_j \leftarrow$ draw a random sample from the common distribution $N(\mu_i,\sigma^2)$
    Calculate the optimal Bayesian order decision with $y_j, Q^*_{\pi^{y_j}_Y} = F^{-1}_{\pi^{y_j}_Y}(\frac{c_u}{c_o+c_u})$
    for $k = 1$ to n_3 **do**
      $x_k \leftarrow$ draw a random sample from the common distribution $N(\mu,\sigma^2)$
      Calculate the loss function $l(x_k, Q^*_{\pi^{y_j}_Y})$
    end for
    $E_{X|Y=y_j,\mu=\mu_i}[l(x_k,Q^*_{\pi^{y_j}_Y})] \leftarrow \frac{1}{n_3}\sum_{k=1}^{n_3} l(x_k,Q^*_{\pi^{y_j}_Y})$
  end for
  $E_{Y|\mu=\mu_i}R(\mu_i,Q^*_{\pi^{y_j}_Y}) \leftarrow \frac{1}{n_2}\sum_{j=1}^{n_2} E_{X|Y=y_j,\mu=\mu_i}$
end for
$E_\mu E_{Y|\mu}R(\mu,Q^*_{\pi^{y_j}_Y}) \leftarrow \frac{1}{n_1}\sum_{i=1}^{n_1} E_{Y|\mu=\mu_i}R(\mu_i,Q^*_{\pi^{y_j}_Y})$
EVSI$\leftarrow \bar{R}(\pi,q^*(\pi)) - E_\mu E_{Y|\mu}R(\mu,Q^*_{\pi^{y_j}_Y})$
return EVSI

The sequence of the transaction is as follows:

1. Model submission: The supplier submits its (newsvendor) model to the data exchange.
2. Data pricing: The data exchange sets the bid price for the supplier based on the EVSI.
3. Information realization: Once the retailer's data are realized (shared), the data exchange calculates the updated decision (the supplier's new order decision using the retailer's data) and sends it to the supplier.
4. Demand realization: The demand is realized from $N(\mu_0,\sigma^2)$.
5. Surplus attribution: Surplus is the difference between the EVSI and the true value (i.e., the cost reduction of the supplier by modifying its order decision with the retailer's information).

After the demand is realized, the surplus or deficit of the data is determined. In our case, the realized value of the retailer's information is expressed as follows:

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

$$V_R = l(x_R, q^*(\pi)) - l(x_R, q^*(\pi_y)) \tag{12}$$

where x_R is the realized demand. Thus, $[V_R - EVSI]^+ and [EVSI - V_R]^+$ are the surplus and deficit, respectively. To distribute the surplus (or deficit) fairly, we suggest setting a weight of r for the supplier and $(1-r)$ for the retailer according to their respective contributions to the usage value from the specific use of the shared data. Notice that the supplier uses the retailer's data y to update its prior mean as follows:

$$\begin{aligned} \tilde{\mu}_{post} &= \frac{y\tau^2 + \tilde{\mu}\sigma^2}{\tau^2 + \sigma^2} \\ &= By + (1-B)\tilde{\mu} \end{aligned} \tag{13}$$

here B is the "shrinkage factor" which specifies the corresponding contributions of the data to the posterior distribution. Thus, we choose $r = B$ as the proportion (to allocate the surplus to the supplier) in our data attribution mechanism. Intuitively, if the retailer sells inaccurate or forged data y, the posterior should lead to a deficit when using the data. The shrinkage factor B would thus punish the retailer for being dishonest.

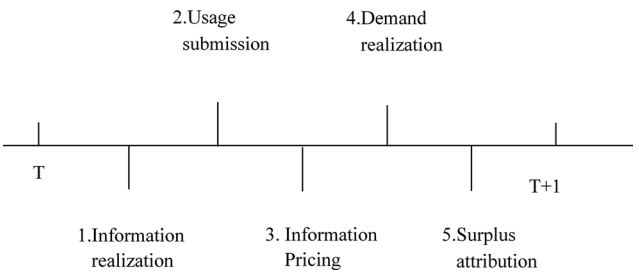
4.2.2 Spot Mode. In this mode, the supplier only pays for the data after the data are shared (realized). As the realized data may contain random noise (aside from the theoretical demand distribution), the price of the data depends on the average benefits they generate in reducing overage and underage costs. This is referred to as the expected value of the information (EVI). The calculation for the EVI is similar to that for the EVSI, but with a directly observed y. It can be computed as follows:

$$V(\pi) = \bar{R}(\pi_Y, q^*(\pi)) - \bar{R}(\pi_Y, q^*(\pi_Y)) \tag{14}$$

The attribution procedure remains the same as that of the prepayment mode. The procedures of the spot mode are depicted in Figure 4.

1. Data realization: y is drawn from the demand distribution $N(\mu_0, \sigma^2)$.
2. Model submission: The supplier submits its model to the data exchange.
3. Data pricing: The data exchange sets the bid price for the supplier according to the EVI and then sends the updated order decision to the supplier.
4. Demand realization: The demand of this selling period is realized.
5. Surplus attribution: The data exchange attributes the surplus or deficit to the supplier and the retailer.

Figure 4 A Graphic Representation of the Spot Mode Transaction Over the Time Horizon



4.3. Indirect Data Sharing

We now consider the case in which some retailers may not be willing to share their data with the supplier directly. To estimate the part of the demand of these non-cooperative retailers, we consider an indirect but realistic approach where the supplier resorts to external analysts in the field to help estimate the demand distribution more accurately (Fu and Zhu 2010). This resembles crowdsourcing (Chen et al. 2014), prediction market (Wolfers and Zitzewitz 2004) or financial analysts, such that these external experts serve as third-party data providers. The data exchange helps screen out analysts who have valuable data. Furthermore, the analysts' data also help the supplier verify whether the data purchased in the previous period are accurate (or manipulated), the results of which are used as the input to specify the terms of the smart contract as shown in Figure 5.

First, we present a mathematical formulation for the above scenario. Assume that one supplier and k retailers are unwilling to share their demand information directly. Additionally, suppose that a group of m expert analysts $u_1, u_2, \cdots, u_m$ provides demand forecasts on these retailers. We assume that the demand forecast provided by the analyst follows a normal distribution. Each analyst u_n holds a private cost $z_{i,n}$ for providing information on retailer i. In the case in which analyst u_n cannot provide the demand information of retailer i, we set its private cost to infinity, $p_{i,n} \to \infty$. The supplier has a limited budget for purchasing the (demand forecast) information from analysts.

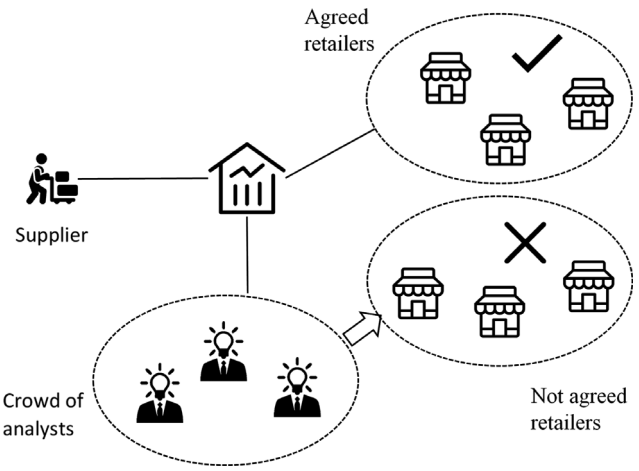
Analyst consulting is an alternative means of data acquisition for the supplier.

u_n	The n-th analysts in the data exchange.
$z_{i,n}$	The cost of analyst n providing information on retailer i.
n^*	The index of the analyst who generates the highest expected revenue.
$l_{i,T}$	The posted price for purchasing the information of retailer i at period T.

To incentivize the analysts to contribute, in each period T, the supplier posts a take-it-or-leave-it price

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Figure 5 Indirect Data Sharing in the Data Exchange



vector (menu) $L_T = (l_{1,T}, \ldots, l_{k,T})$, where $l_{i,T}$ is the monetary payment for the information on i at period T. The supplier dynamically change the posted price L_T according to the dynamics of the data market and retailers' willingness to share their information in different periods. Considering a posted price L_T, the supplier will obtain the information on the retailers in $B_T = \{i \mid \min_{n\in[m]} z_{i,n} \le l_{i,T}\}$ by paying $\sum_{i\in B_T} l_{i,T}$. In the case in which multiple analysts provide information on the same retailer, the data exchange helps the supplier screen out analyst u_{n^*} whose data can generate the most revenue for the supplier. We define the reward of L_T as $\sum_{i\in B_T} RE_{i,T}$, where $RE_{i,T}$, defined in Section 4.2, is the expected supplier's profit squeeze from the demand information on retailer i in period T. The supplier repeats this process period by period until the supplier runs out of budget or until it reaches the last period. The supplier then faces an online decision requiring it to select a price L_T and screen out the best analyst in each period T. The supplier would like to maximize the expected reward subject to the budget constraint. In this study, we implement the classic online learning algorithm, LIME (Han et al. 2017), to compute the pricing policy.

5. Architecture Design

Our proposed blockchain-based architecture aims to prevent data leakage or unauthorized data usage. It uses smart contracts to enable automatic and consensual value attribution. However, several concerns must be addressed first in building such a trustworthy data exchange, including the possibility of the data exchange (a) misreporting the value of the data, (b) leaking the model of the buyer, (c) leaking the data of the seller, and (d) forgoing accountability. To prevent these issues and facilitate a generic data transaction, we devise the procedure outlined in Table 3 below.

Below, we highlight the key functions of the decentralized data exchange.

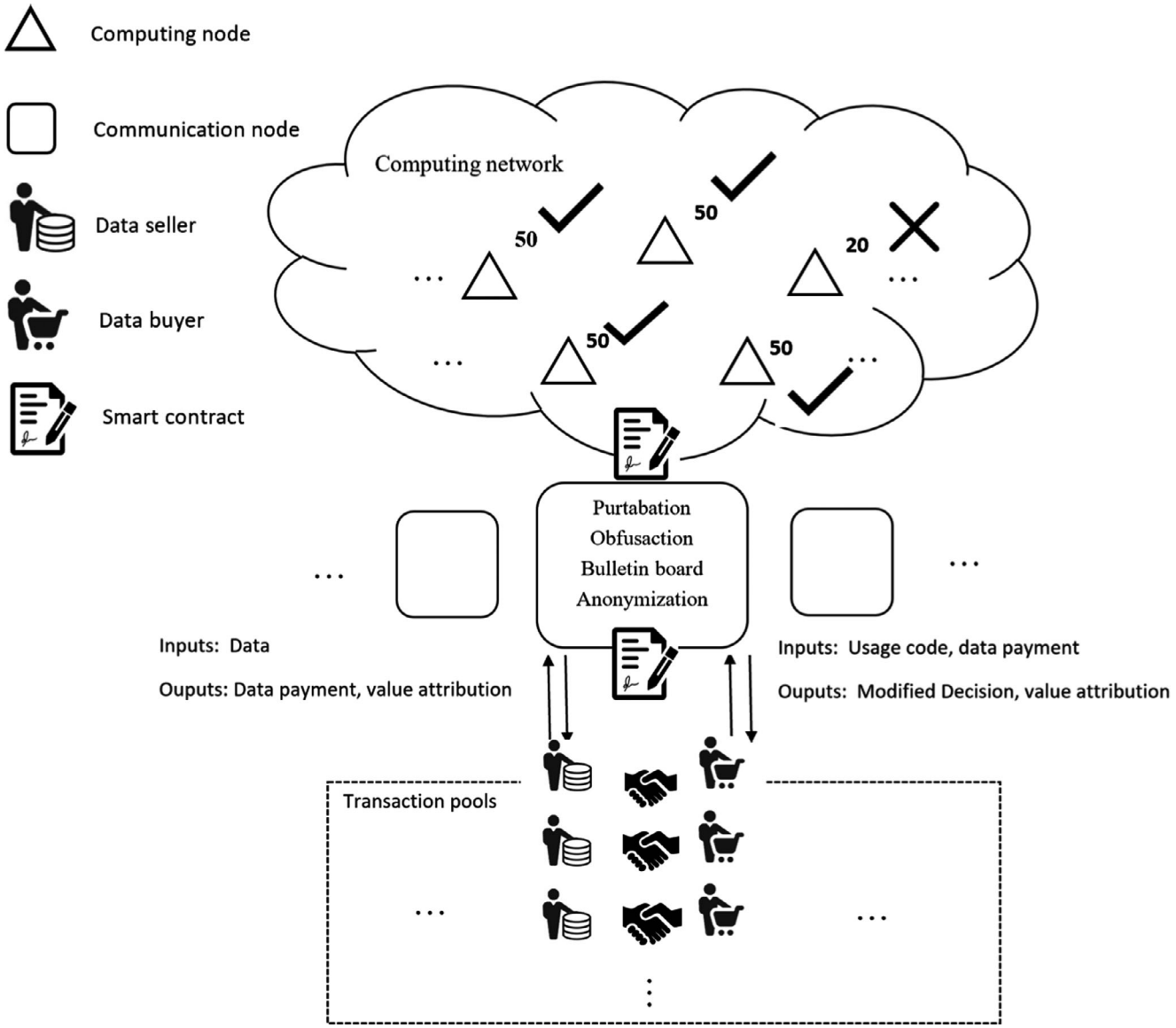
Decentralized governance: The data exchange permits two types of nodes[3]. The first type is the processing node which acts as an (ad-hoc) agent to facilitate communications among the buyer, seller, and the data exchange. This design is necessary to ascertain that none of the key players (i.e., the buyer, the seller, and the exchange) has direct access to each other's private information, thus preventing information leakage[4]. Each transaction is assigned to a randomly selected processing node. This helps (a) prevent the data exchange from manipulating the transaction given that it is the processing node that directly handles data-sharing transactions, and (b) preclude collusion or manipulation among sellers or buyers in advance as the node is randomly chosen before the transaction. The buyer and the seller send the model and data to the processing node respectively. This helps prevent the buyer from leaking the data (of the seller) and the seller from leaking the model (of the buyer). Thus, the processing node executes anonymization, data perturbation, code obfuscation and accountability (by tracing the usage of the data in the model).

The second type is the computing node. Once the data, model (e.g., newsvendor model) and valuation code (e.g., the EVSI and the EVI) are released from the processing node, each computing node independently computes the usage value of the data and announces the updated (order quantity) decision for the buyer. If the majority(e.g., two thirds) of the computing nodes reach the same result, then the value and the decision are determined and parsed to the smart contract. Otherwise, the transaction is nullified. The processing node does not participate in the calculation and evaluation procedure to preclude collusion as shown in Figure 6.

The computing node is introduced instead of online smart contract computing for three main reasons. The first reason is that smart contract languages, such as (the most popular) Solidity , are not designed to undertake heavy computation. The current design of smart contracts does not contain a math library in which to build complex mathematical models. Bold attempts have been recently made, such as Algorithmia's Danku, to incorporate some computation ability into their smart contract protocols. However, such protocols remain in their infancy. The second reason pertains to security—the simpler, the securer. The state-of-the-art design principle of smart contracts is to keep business logic in the smart contract but not complex computation (Pop et al. 2020). The third reason is that, the speed of execution already presents to be a main blockchain bottleneck, which will only be exacerbated with complex computations in a smart

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Figure 6 The Architecture of the Data Exchange



contract. For these reasons, keeping computation off-chain in a decentralized manner is a prudent and rational choice, given the current state of blockchain techniques.

Model obfuscation: In the context of supply chain data sharing, directly implementing the usage-based data valuation model (e.g., the EVSI) may pose a challenge: the transaction may contain some confidential content of the model that the buyer is not willing to disclose to others. For example, the supplier may not want to reveal how it makes decisions on order quantity (e.g., the specific parameters with respect to the overstock and understock fees). Given this consideration, we embrace an obfuscation-based method, which is the act of deliberately obscuring the source code of the model to prevent attackers from inferring the actual model. It requires intruders to expend numerous resources (e.g., time, space, and computing power) to reverse engineer if the target model is well obfuscated. The obfuscation procedure includes layout transformation, variable renaming, string encryption, reordering, dummy code insertion, and many other techniques. The output is a new model that is functionally equivalent to the original one, but with the key components obfuscated. The data buyer can freely choose how much obscurity to add to its model. For example, under the Java environment, classes can be merged or split, methods can be changed or created, and data structures can be modified or combined (Collberg et al. 1998). For models involving

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Table 3 Transaction Procedures Under the Prepay and Spot Modes

Prepay mode	Spot mode
1. The data exchange randomly picks a node (a registered member) in the blockchain system to process the transaction (thus preventing collusion in advance).	
2. The data buyer uploads the hash of the original model through the smart contract and sends the obfuscated model to the processing node (thus preventing the content of the model from being leaked).	
3. The chosen node sends the obfuscated model to a group of randomly selected nodes to compute the EVSI. The consensus of the EVSI value (e.g., agreed by at least two thirds of all of the computing nodes) of the data is parsed to the smart contract.	The data seller uploads the data.
4. The data seller uploads the data with a certain level of perturbation (to mitigate data leakage) that is agreed on by the buyer and seller.	The data buyer and seller jointly agree on how to attribute the surplus and the perturbation level. The agreement is then uploaded to the smart contract with their digital signatures.
5. The data buyer and seller make a joint agreement on how to attribute the surplus and the agreement is then uploaded to the smart contract with their digital signatures.	The processing node sends the obfuscated model to the computing nodes. The computing nodes reach consensus on the EVI of the data and post it on the smart contract.
6. The computing nodes reach a consensus on the updated (order quantity) decision and send it to the buyer.	
7. After the true realization of the data usage, the buyer uploads the true value of the data to the processing node.	
8. The computing nodes verify and reach a consensus on the realized data usage value and post it on the smart contract. The smart contract automatically allocates the surplus (or deficits) to the buyer and the seller.	

strict confidentiality, a large pool of code transformation can be applied repeatedly to the model until the required obfuscation standard is achieved. The use of irrelevant or unusable code is also a common way to manage the obscurity level of the model. A sample of the before and after cases of applying obfuscation to calculate the EVSI using the Monte Carlo algorithm is shown in Figure 7.

Data encryption and perturbation: A blockchain is proven to ensure trust and security during a transaction. However, post-transaction, the nature of blockchain transparency leaves the door open for attackers to infer private data after observing the completed transaction recorded in the blockchain. To ensure data confidentiality, encryption is necessary at every step of data transmission. Popular encryption algorithms include SHA-256, MD5 and RSA but identifying the best choice is not a focus of this study. For the purpose of illustration, we leverage an asymmetric cryptographic algorithm called the Paillier cryptosystem (Paillier 1999) to explain how secure computation can be achieved. It is worth noting that the proposed encryption scheme not only protects the privacy of the data seller but also safeguards the model of the data buyer. The details of the encryption and the generation process of the keys are described below.

Figure 7 Monte Carlo Algorithm to Calculate EVSI: Before and After Obfuscation

```
function computeEVSI(mu_post, tau,mu,sig,n1,n2,n3) {
  Emu_loss=0;
  for (var i=0;i<n1;i++){
    rm=normalRandom(mu_post,sig);
    EY_loss=0;
    for (var j=0;j<n2;j++){
      ry=normalRandom(rm,sig);
      qy=updateOrderQuantity(co,cu,ry,rm,tau,sig);
      EX_loss=0;
      for (var k=0;k<n3;k++){
        rx=normalRandom(mu,sig);
        rloss=lossFunction(co,cu,qy,rx);
        EX_loss=EX_loss+rloss;
      }
      EX_loss=EX_loss/n3;
      EY_loss=EY_loss+EX_loss;
    }
    EY_loss=EY_loss/n2;
```

(a) Original code

```
function _0x24d9c3
(_0x41a3d9,_0x2b0f86,_0x82e766,_0x59f385,_0xa150a3,_0x2e4c6c,_
0x380e64){Emu_loss=0x1940+-0x8f+0x3*-0x83b;for(var
_0x567974=0x464+0x7*-
0x2e3+0xfd1;_0x567974<_0xa150a3;_0x567974++)
{rm=normalRandom(_0x41a3d9,_0x59f385),EY_loss=-0x131*-0x1f
+0x1772+-0x3c61;for(var _0x3423dc=0x5+0x19*-
0xb7+0x11da;_0x3423dc<_0x2e4c6c;_0x3423dc++)
{ry=normalRandom(rm,_0x59f385),qy=updateOrderQuantity
(co,cu,ry,rm,_0x2b0f86,_0x59f385),EX_loss=0x6*0x65e+-
0x4f8*0x3+-0x174c;for(var _0x4b7a0c=-0x6a*-0x13+0x1*-0x239e
+0x1bc0;_0x4b7a0c<_0x380e64;_0x4b7a0c++){rx=normalRandom
(_0x82e766,_0x59f385),rloss=lossFunction
(co,cu,qy,rx),EX_loss=EX_loss+rloss;}
EX_loss=EX_loss/_0x380e64,EY_loss=EY_loss+EX_loss;}
EY_loss=EY_loss/_0x2e4c6c,Emu_loss=Emu_loss+EY_loss;}
```

(b) Obfuscated code

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Key generation. Generate an *one-time* public and secret key pair, (pk,sk) for a transaction.

1. Randomly choose two large prime P and Q which satisfy $gcd(PQ, (P-1)(Q-1)) = 1$. gcd means greatest common divisor.
2. Compute $N = PQ$ and $\lambda = lcm(P-1,Q-1)$. lcm means least common multiple.
3. Denote $Z^*_{N^2} \subset Z_{N^2} = \{0,1,\ldots,N^2-1\}$ as the set of non-negative integers that have multiplicative inverse model. Select random integer $g \in Z^*_{N^2}$ which satisfies $gcd(L(g^\lambda mod N^2), N) = 1$, where $L(z) = \frac{z-1}{N}$.
4. Compute $\psi = (L(g^\lambda \bmod N^2))^{-1} \bmod N$
5. The public encryption key pk is (N,g) and the secret decryption key sk is (λ,$\psi\psi$).

Encryption. Encrypt a plaintext message.

1. Let A be a plaintext message to be encrypted. $0<A<N$
2. Randomly choose a number w where $0<w<N$ and $w \in Z^*_{N^2}$.
3. The ciphertext of message A is computed as: $C = E_{pk}(A) = g^A \cdot w^N \bmod N^2$.

Decryption. Decrypt a ciphertext message.

1. Let C be a ciphertext message to be decrypted. $0<C<N$
2. The plaintext of message C is computed as: $A = D_{pk}(C) = L(C^\lambda \bmod N^2) \cdot \psi \bmod N$

Assume that $E_{pk}(A,w)$ is the ciphertext form of A encrypted by the public key pk and that w is the random integer described in Step 2 of **Encryption**. v is a plaintext-form parameter. The Paillier cryptosystem supports the properties below.

1. **Additive homomorphism** The product of two ciphertexts is the same as the encryption of the sum of their corresponding plaintexts:

$$D(E_{pk}(A_1,w_1) \times (A_2,w_2) \ mod \ N^2) = A_1 + A_2 \ mod \ N \tag{15}$$

2. **Plaintext multiplication** An encrypted plaintext raised to a constant k decrypts to the product of the plaintext and the constant:

$$D(E_{pk}(A,w)^v \ mod \ N^2) = vA \ mod \ N \tag{16}$$

With the help of the additive homomorphism and plaintext multiplication properties, we achieve secure computation as follows:

1. The data seller sends the encrypted data and the one-time public key pk to the data buyer.
2. The data buyer encrypts its main model parameters with public key pk and computes the encryption of the product of the data and model parameters. The data buyer sends the encrypted product and its model (after obfuscation) to the data exchange.
3. The data seller sends the one-time secret key sk to the computing node. The computing node decrypts the product of the data and model parameters and then computes the usage value of the target data.

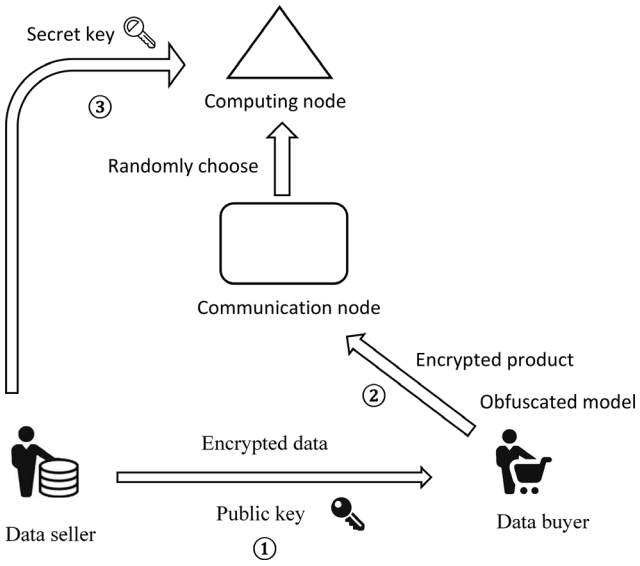
A schematic diagram of the framework is shown in Figure 8. Here, we provide a simple example in which the posterior mean $\tilde{\mu}_{post}$ (Equation (13)) is securely computed to illustrate how our encryption strategy works. To securely calculate $\tilde{\mu}_{post}$, the retailer sends the encrypted data $E_{pk}(y)$ and the public key pk to the supplier. According to the computing procedure, the supplier encrypts its model parameters $\tilde{\mu}\sigma^2, \gamma^2 + \sigma^2$ and homomorphically computes the product of the encrypted data, encrypted parameters, and plaintext parameters.

$$\begin{aligned} E_{pk}(\tilde{\mu}_{post}) &= E_{pk}\left(\frac{y\tau^2 + \tilde{\mu}\sigma^2}{\tau^2+\sigma^2}\right) \\ &= E_{pk}(y\tau^2 + \tilde{\mu}\sigma^2)^{1/(\tau^2+\sigma^2)} \\ &= (E_{pk}(y\tau^2) \times E_{pk}(\tilde{\mu}\sigma^2))^{1/(\tau^2+\sigma^2)} \\ &= (E_{pk}(y)^{\tau^2} \times E_{pk}(\tilde{\mu}\sigma^2))^{1/(\tau^2+\sigma^2)} \end{aligned} \tag{17}$$

The supplier then submits $E_{pk}(\tilde{\mu}_{post})$ to (randomly selected) computing nodes through a processing node. The computing nodes decrypt the ciphertext $E_{pk}(\tilde{\mu}_{post})$ using the secret key sk and start to compute the usage value of the data. Note that even if the computing nodes know the value of $\frac{y\tau^2+\tilde{\mu}\sigma^2}{\tau^2+\sigma^2}$, they are not able to infer the retailer's data y, as well as the supplier's model parameters $\tilde{\mu}, \sigma^2$ and τ^2. Thus, our encryption scheme prevents data and model leakage.

Under certain circumstances, the data buyer may be able to infer what the shared data would be through backward derivation from the output. To prevent this type of attack, we propose a perturbation

Figure 8 Framework of the Proposed Data Encryption Scheme



Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

method to ascertain the confidentiality of the data. Data perturbation is a common and effective privacy preservation technique that by and large falls into two categories: probability distortion and value distortion (Liu et al. 2005). We use the value distortion approach, which adds a random noise from a Gaussian/Laplacian distribution, which has long been used in data perturbation and differential privacy, to the raw data (please note that our architecture is not restricted to the specific choice of a perturbation approach). Based on the amount of noise added to the raw data, the retailer offers the supplier a discount ratio α to compensate for the loss of data quality.

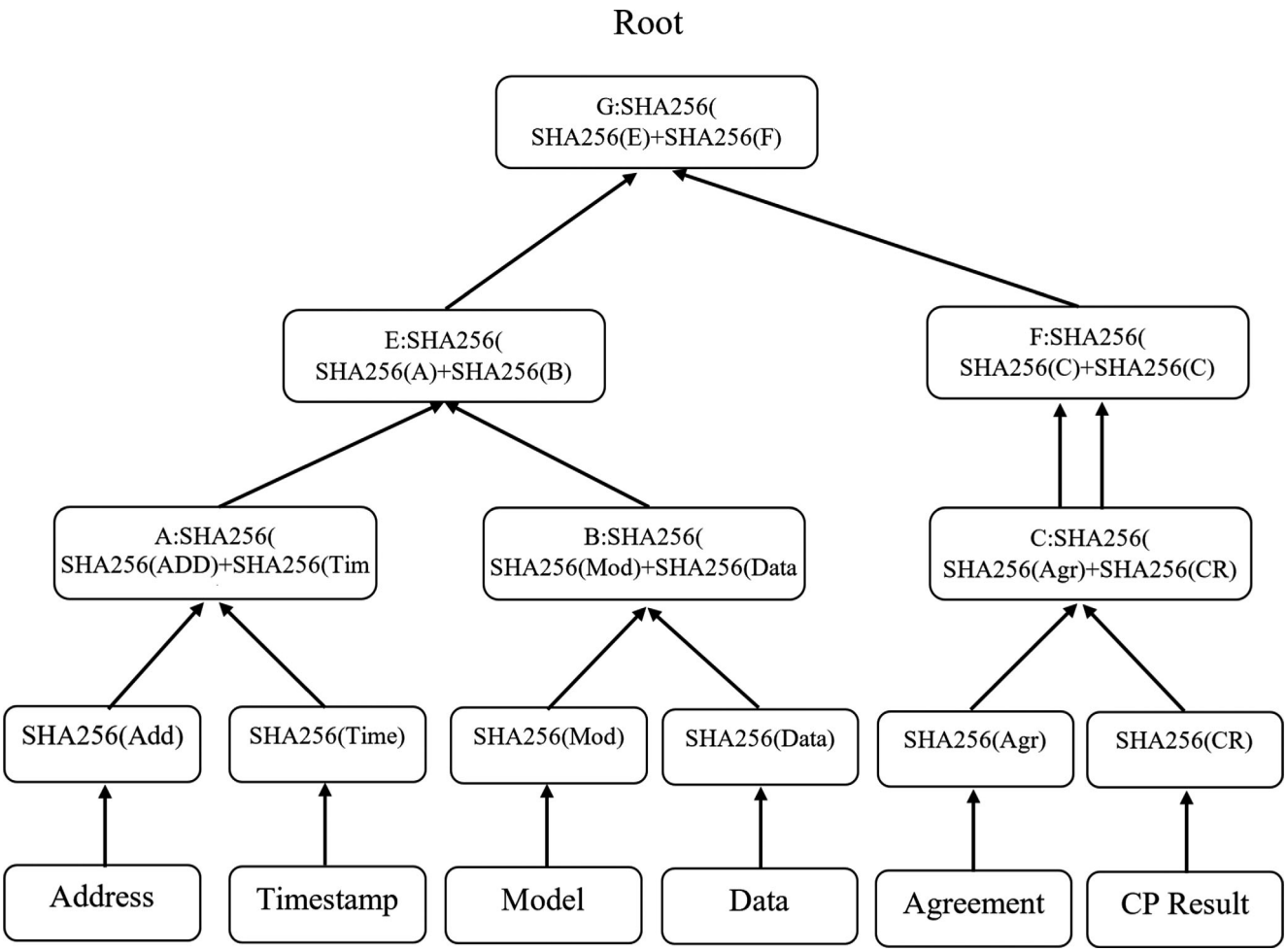
Consensual agreement: Our data exchange requires mutual agreement on several clauses between buyers and sellers to ensure consensual and fair data sharing. Take the scenario in section 4.2 as an example. At first, the supplier and the retailer have to agree on the perturbation level and discount ratio α. Then, both parties need to agree on the evaluation logic/procedure (e.g., the EVSI or the EVI) in determining the usage value of data and on the corresponding logic for the value attribution. In the direct data-sharing scenario, the evaluation logic specifies how to calculate the economic benefits in terms of reducing the demand uncertainty after acquiring the shared data; the value attribution logic determines the proportion B (see Equation (13)). Finally, all of the parties, including the data exchange, need a prespecified agreement on the penalty for misbehavior. Misbehavior, such as failing to upload the data/model on time, submitting false information and intentionally leaking information, is punished differently according to the severity of the violations in terms of consequences. All of the agreements require consensus between the seller and the buyer (e.g., via digital signature) during the transaction.

Accountability: It is often infeasible, if not impossible, to have 360-degree surveillance of the behavior of the seller and buyer. The lessons learned from mobile apps show that any services that require heavy surveillance often fail because of poor user experience and distrust of users concerning such apps (Jung et al. 2018). To ensure accountability, our proposed data exchange embraces the post hoc mechanism that punishes misbehavior after it has happened. The data exchange uses a "bulletin board" module as the building block to regulate users' behavior. All of the actions of users are recorded in the blockchain in an untamperable manner to hold every action accountable. The bulletin board is a public channel in the smart contract that broadcasts the key transaction information which is encrypted using methods such as the SHA256 algorithm and restructured as a Merkle tree to compress the data. A Merkle tree is a special data construct in which each nonleaf node of the tree is composed of the hash values of its child nodes. The leaves of the Merkle tree include the hash of the following information: the account addresses of the seller and buyer, transaction timestamps, the buyer's model, the seller's data, agreements, and computing results. These hash values are used to verify whether the users and the nodes in the data exchange follow the transaction rules. Once the data seller and buyer submit the required information, the data exchange generates a Merkle tree. The Merkle tree structure designed for our proposed data exchange is illustrated in Figure 9. It enables the data exchange to attain accountability against dishonest participants (including the data exchange itself) throughout the data-sharing transaction by tracking everyone's behaviors. Trading-related misbehavior may include misinformation or refusal to upload the necessary information required by the bulletin board. The data exchange has the right to terminate the transaction and enforce punishment if such misbehavior occurs. This combination of blockchain immutability and signature for every involved process provides accountability and thus regulates user behaviors.

Anonymity: In a blockchain transaction system, although every trace of the transaction is recorded and transparent to all, the anonymity of user identity is not compromised. This can be achieved in many ways. For example, the use of the pseudonym method (e.g., hashing user account address) effectively protects the privacy of user's identity. However, in the context of the supply chain, other concerns may arise. For example, as a user is repeatedly involved in multiple transactions, an intruder may be able to infer the user's true identity by mining the transaction patterns from open transaction records, a technique referred to as cryptanalysis. In certain industries (e.g., the food, agriculture, and drug industries) to which our proposed data exchange applies, transaction records with identity are required by law to be public (Chod et al. 2020), and anonymity is not an issue. For general-purpose scenarios, we embrace a mixing service center mechanism, which has been widely adopted by popular platforms such as BitcoinFog and SendShared, to protect privacy and ensure anonymity for cryptocurrencies (Feng et al. 2019). The mixing service center arbitrarily chooses a certain number of transaction requests in its transaction pool, forming a transaction list. After user verification (e.g., via blind signature, group signature, aggregate signature, and ring signature), the transactions in the list are mixed into a single transaction and broadcasted to the blockchain by the mixing service center, thereby hiding the connection between the buyer and seller. This also makes transaction analysis unlikely to be useful. The well-

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Figure 9 The Merkle Tree Structure of the Bulletin Board



known Monero's ring signature encryption technique bears a similar idea. Here we omit the technical details for brevity. The decentralized structure of the proposed data exchange also plays an important role in retaining user anonymity. Recall that each data transaction is assigned to a group of randomly selected processing nodes. The probability of a single processing node de-anonymizing user identity through tracking the transaction record is negligible. Another promising solution to protect transaction privacy and anonymity is the zero-knowledge-proof (ZKP) solution. Sasson et al. (2014) propose a protocol, Zerocash, based on the ZKP solution that enables transactions to hide the payment origin, destination, and amount. When such techniques mature, they can be readily incorporated into our data exchange design.

6. Discussions

We propose and implement a blockchain-based data exchange to facilitate trustable information sharing in a supply chain. Our exchange design addresses several key challenges that often afflict supply chain data sharing, including privacy breach, data leakage, improper valuation of data, and unfair compensation. The exchange achieves transparency, security, fairness, and accountability through our proposed usage-based data valuation mechanism. We demonstrate how to deploy such a data exchange using Hashgraph and provide a detailed tutorial at the code level for the main functionalities in the Online Appendix.

We make several unique contributions to the data-sharing literature in SCM. *First*, we make one of the first attempts to provide a blockchain-based solution to data sharing in SCM. We show how to leverage various blockchain features, such as decentralization, consensus mechanisms, cryptography, and micropayments, to address the challenges of data sharing. *Second*, we propose a novel usage-based valuation approach that determines the value of shared data by tracking how the data are used in the buyer's model and how much improvement it contributes to model performance. We develop the EVSI and EVI methods to accommodate different data-sharing scenarios. *Third*, we provide a closed-loop solution with a

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

detailed tutorial on how to implement the proposed mechanism using Hashgraph.

Our research also bears several limitations that need to be addressed in future research. First, for simplicity, we do not consider price competition. Competition may result in different pricing strategies for both data sellers and buyers. Second, we assume that the supplier always prefers to acquire data from the retailer directly when the retailer is willing to share the data. This is a convenience assumption that is nonessential to the design of the exchange. In practice, acquiring data indirectly from analysts may be more valuable depending on their quality and cost. Third, due to the currently limited computing capacity of smart contract, it is sometimes necessary to compute the data value outside the smart contract when complex computation and optimization are involved. This off-chain computation (edge computing) may come at the expense of the trustworthiness of the blockchain. Fourth, we do not consider cryptocurrency or tokens that can be used to incentivize various parties to share data. Finally, a fully functional data exchange needs to prevent arbitrage by factoring in the possibility that both data sellers and buyers may be strategic. All of these are exciting topics for future research.

Blockchain-enabled data sharing opens up many potential innovations for SCM, such as supply chain finance (SCF). Small- and medium-sized enterprises often lack access to credit and liquidity and their working capital is often occupied by a high level of inventories (Hofmann et al. 2017). The trustable nature of the data on the blockchain and transactions powered by smart contracts provide a dependable ledger that is auditable for the debtor. Financial institutes can use the traceable data in smart contract to build a credit profile of the retailer, whereas suppliers can use such data to seek credit enhancement. The data exchange is able to better estimate true demand by enabling trustworthy data sharing and can thus better help financial institutes determine the risk of an individual supplier.

Acknowledgments

The authors gratefully acknowledge the feedback received from the department editor Subodha Kumar, the senior editor, and two anonymous reviewers for their helpful comments and constructive suggestions. This research is supported in part by the National Natural Science Foundation of China (71531010, 71831006, 71421002, 72010107002).

Notes

[1]The Hash will be adequate for the data buyer to verify that the data are exactly what are supposed to be shared from the seller if dispute occurs.

[2]This simplifying assumption is inconsequential to our design and our model can be extended to factor in the case when the variance is unknown.

[3]To qualify as a node in Hashgraph (and also in other popular platforms such as Hyperledger and Coinbase), an individual is required to go through KYC (know your customer) and AML (anti-money laundering) verification process, through which the individual's IP address, device information (e.g., device name, operating system, location and web log information) are collected.

[4]Here security can be further strengthened using cryptography techniques like the Oblivious Transfer protocol (Li et al. 2021) to prevent data leakage during communications, which we leave for future research to explore.

References

Adewole, A. 2005. Developing a strategic framework for efficient and effective optimisation of information in the supply chains of the UK clothing manufacture industry. *Supply Chain Manag. Int. J.* **10**(5): 357–366.

Anand, K. S., M. Goyal. 2009. Strategic information management under leakage in a supply chain. *Management Sci.* **55**(3): 438–452.

Babich, V., G. Hilary. 2020. Om forum-distributed ledgers and operations: What operations management researchers should know about blockchain technology. *Manuf. Serv. Oper. Manag.* **22**(2): 223–240.

Bimpikis, K., D. Crapis, A. Tahbaz-Salehi. 2019. Information sale and competition. *Management Sci.* **67**(6): 2646–2664.

Blossey, G., Eisenhardt, J., Hahn, G. 2019. Blockchain technology in supply chain management: An application perspective. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

Bourland, K. E., S. G. Powell, D. F. Pyke. 1996. Exploiting timely demand information to reduce inventories. *Eur. J. Oper. Res.* **92**(2): 239–253.

Brustein, J. 2012. Start-ups seek to help users put a price on their personal data. *The New York Times* **12**(3).

Buterin, V. 2014. A next-generation smart contract and decentralized application platform. Available at http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf (accessed date January 25, 2021).

Cachon, G. P., M. Fisher. 2000. Supply chain inventory management and the value of shared information. *Management Sci.* **46** (8): 1032–1048.

Carlin, B. P., T. A. Louis. 2009. Bayesian methods for data analysis. *J. Roy. Stat. Soc.* **172**(4): 935–936.

Casey, P. W. 2017. Global supply chains are about to get better, thanks to blockchain. Retrieved from https://hbr.org/2017/03/global-supply-chains-are-about-to-get-better-thanks-to-blockchain on November 23, 2020.

Chen, Y., Ö. Özer. 2019. Supply chain contracts that prevent information leakage. *Management Sci.* **65**(12): 5619–5650.

Chen, H., P. De, Y. J. Hu, B.-H. Hwang. 2014. Wisdom of crowds: The value of stock opinions transmitted through social media. *Rev. Financial Stud.* **27**(5): 1367–1403.

Chod, J., N. Trichakis, G. Tsoukalas, H. Aspegren, M. Weber. 2020. On the financing benefits of supply chain transparency and blockchain adoption. *Management Sci.* **66**(10): 4378–4396.

Cohen, M. A., T. H. Ho, Z. J. Ren, C. Terwiesch. 2003. Measuring imputed cost in the semiconductor equipment supply chain. *Management Sci.* **49**(12): 1653–1670.

Collberg, C., C. Thomborson, D. Low. 1998. Manufacturing cheap, resilient, and stealthy opaque constructs. In *Proceedings of the*

25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages.

Demirezen, E. M., S. Kumar, B. Shetty. 2016. Managing co-creation in information technology projects: A differential games approach. *Inform. Syst. Res.* **27**(3): 517–537.

Demirezen, E. M., S. Kumar, B. Shetty. 2018. Two is better than one: A dynamic analysis of value cocreation. *Prod. Oper. Manag.* **29**(9): 2057–2076.

Felici, M., T. Koulouris, S. Pearson 2013. Accountability for data governance in cloud ecosystems. In *Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science—Volume 02*.

Feng, T., X. Chen, C. Liu, X. Feng. 2019. Research on privacy enhancement scheme of blockchain transactions. *Secur. Priv.* **2** (6): 89–102.

Fu, Q., K. Zhu. 2010. Endogenous information acquisition in supply chain management. *Eur. J. Oper. Res.* **201**(2): 454–462.

Gallego, G., I. Moon. 1993. The distribution free newsboy problem: Review and extensions. *J. Oper. Res. Soc.* **44**(8): 825–834.

Gavirneni, S., R. Kapuscinski, S. Tayur. 1999. Value of information in capacitated supply chains. *Management Sci.* **45**(1): 16–24.

Ghoshal, A., S. Kumar, V. Mookerjee. 2018. Dilemma of data sharing alliance: When do competing personalizing and non-personalizing firms share data. *Prod. Oper. Manag.* **29**(8): 1918–1936.

Ha, A. Y., S. Tong. 2008. Contracting and information sharing under supply chain competition. *Management Sci.* **54**(4): 701–715.

Ha, A. Y., S. Tong, H. Zhang. 2011. Sharing demand information in competing supply chains with production diseconomies. *Management Sci.* **57**(3): 566–581.

Ha, A. Y., Q. Tian, S. Tong. 2017. Information sharing in competing supply chains with production cost reduction. *Manuf. Serv. Oper. Manag.* **19**(2): 246–262.

Han, K., Y. He, H. Tan, S. Tang, H. Huang, J. Luo. 2017. Online pricing for mobile crowdsourcing with multiminded users. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 18. ACM.

Hays, C. L. 2004. What wal-mart knows about customers' habits. *The New York Times*, **14**.

Hofmann, E., U. M. Strewe, N. Bosia. 2017. *Supply chain finance and blockchain technology: the case of reverse securitisation*. Springer, Berlin.

Hosseinzadeh, S., S. Hyrynsalmi, V. Leppänen. 2016. Obfuscation and diversification for securing the Internet of things (iot). In *Internet of Things*, pp. 259–274. Elsevier.

Jeong, I.-J., V. J. Leon. 2012. A serial supply chain of newsvendor problem with safety stocks under complete and partial information sharing. *Int. J. Prod. Econ.* **135**(1): 412–419.

Jiang, S., J. Cao, H. wu, Y. Yang, M. Ma, J. He. 2018. Blochie: a blockchain-based platform for healthcare information exchange. In *2018 ieee international conference on smart computing (smartcomp)*, pp. 49–56.

Jung, T., X.-Y. Li, W. Huang, Z. Qiao, J. Qian, L. Chen, J. Han, J. Hou. 2018. Accounttrade: Accountability against dishonest big data buyers and sellers. *IEEE Trans. Inf. Forensics Secur.* **14** (1): 223–234.

Kalra, S., S. Goel, M. Dhawan, S. Sharma. 2018. Zeus: Analyzing safety of smart contracts. In *Network and Distributed System Security Symposium*.

Kong, G., S. Rajagopalan, H. Zhang. 2013. Revenue sharing and information leakage in a supply chain. *Management Sci.* **59**(3): 556–572.

Koutroumpis, P., A. Leiponen, L. D. W. Thomas. 2017. The (unfulfilled) potential of data marketplaces. *Etla Working Papers*.

Kumar, S., V. Mookerjee, A. Shubham. 2018. Research in operations management and information systems interface. *Prod. Oper. Manag.* **27**(11): 1893–1905.

Kurtuluş, M. 2017. *Collaborative Forecasting in Retail Supply Chains*, Springer International Publishing, Cham. pp. 39–61.

Lee, H. L., V. Padmanabhan, S. Whang. 1997. Information distortion in a supply chain: the bullwhip effect. *Management Sci.*, **43**(4): 546–558.

Li, C., G. Miklau. 2012. Pricing aggregate queries in a data marketplace. In *WebDB*, pp. 19–24.

Li, T. L., W. Ren, Y. Xiang, X. Zheng, T. Zhu, G. Kim-Kwang. 2021. Faps: A fair, autonomous and privacy-preserving scheme for big data exchange based on oblivious transfer, ether cheque and smart contracts. *Inf. Sci.* **544**: 469–484.

Liu, K., H. Kargupta, J. Ryan. 2005. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Trans. Knowl. Data Eng.* **18**(1): 92–106.

Liu, H., W. Jiang, G. Feng, K.-S. Chin. 2020. Information leakage and supply chain contracts. *Omega*, **90**: 101994.

Loshin, P. 2013. *Simple Steps to Data Encryption*. Syngress, Boston.

Mehta, S., M. Dawande, G. Janakiraman, V. Mookerjee. 2019. How to sell a dataset? pricing policies for data monetization. *Pricing Policies for Data Monetization*.

Meijer, A. R. 2016. *Algebra for Cryptologists*. Springer International Publishing, Berlin.

Mishra, B. K., S. Raghunathan, X. Yue. 2009. Demand forecast sharing in supply chains. *Prod. Oper. Manag.* **18**(2): 152–166.

Munves, G. 2013. *Wake up, retailers! make money from your big data*. Available at https://chainstoreage.com/news/wake-retailers-make-money-your-big-data (accessed date January 25, 2021).

Najjar, M. S., W. J. Kettinger. 2013. Data monetization: Lessons from a retailer's journey. *MIS Q. Executive* **12**(4): 213–225.

Nakamoto, S. 2019. Bitcoin: A peer-to-peer electronic cash system. Manubot.

Nakasumi, M. 2017. Information sharing for supply chain management based on block chain technology. In In *2017 IEEE 19th conference on business informatics*, volume 1, pp. 140–149.

Özer, Ö., W. Wei. 2006. Strategic commitments for an optimal capacity decision under asymmetric forecast information. *Management Sci.*, **52**(8): 1238–1257.

Özer, Ö., Y. Zheng, K.-Y. Chen. 2011. Trust in forecast information sharing. *Management Sci.* **57**(6): 1111–1137.

Paillier, P. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*.

Pop, C., T. Cioara, I. Anghel, M. Antal, I. Salomie. 2020. Blockchain based decentralized applications: Technology review and development guidelines.

PRG, P. R. G. 2013. The supplier collaboration shortage: uncovering the gaps in supply chain readiness. pp. 1–14.

Ren, Z. J., M. A. Cohen, T. H. Ho, C. Terwiesch. 2010. Information sharing in a long-term supply chain relationship: The role of customer review strategy. *Oper. Res.* **58**(1): 81–93.

Sasson, E. B., A. Chiesa, C. Garman, M. Green, M. Virza. 2014. Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy*.

Scheele, L. M., U. W. Thonemann, M. Slikker. 2018. Designing incentive systems for truthful forecast information sharing within a firm. *Management Sci.* **64**(8): 3690–3713.

Seifert, D. 2003. Collaborative planning, forecasting, and replenishment: How to create a supply chain advantage.

Shang, W., A. Y. Ha, S. Tong. 2015. Information sharing in a supply chain with a common retailer. *Management Sci.* **62**(1): 245–263.

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356

Shih, H.-P., K.-H. Lai, T. E. Cheng. 2015. Examining structural, perceptual, and attitudinal influences on the quality of information sharing in collaborative technology use. *Inform. Syst. Front.* **17**(2): 455–470.

Soh, C., M. L. Markus, K. H. Goh. 2006. Electronic marketplaces and price transparency: Strategy, information technology, and success. *MIS Q* **30**(3): 705–723.

Strong, M., J. E. Oakley, A. Brennan, P. Breeze. 2015. Estimating the expected value of sample information using the probabilistic sensitivity analysis sample: A fast, nonparametric regression-based method. *Med. Decis. Mak.* **35**(5): 570–583.

Swan, M. 2015. *Blockchain: Blueprint for a new economy*. O'Reilly Media, Newton, Massachusetts.

Wolfers, J., E. Zitzewitz. 2004. Prediction markets. *J. Econ. Perspect.* **18**(2): 107–126.

Xiang, Y., M. Sarvary. 2013. Buying and selling information under competition. *Quantit. Mark. Econ.* **11**(3): 321–351.

Yue, X., J. Liu. 2006. Demand forecast sharing in a dual-channel supply chain. *Eur. J. Oper. Res.* **174**(1): 646–667.

Yue, J., B. Chen, M.-C. Wang. 2006. Expected value of distribution information for the newsvendor problem. *Oper. Res.* **54**(6): 1128–1136.

Zaerens, K. 2018. Concept for controlled business critical information sharing using smart contracts. In *2018 2nd Cyber Security in Networking Conference (CSNet)*, pp. 1–8.

Zhang, H. 2002. Vertical information exchange in a supply chain with duopoly retailers. *Prod. Oper. Manag.* **11**: 531–546.

Zhang, F. 2006. Competition, cooperation, and information sharing in a two-echelon assembly system. *Manuf. Serv. Oper. Manag.* **8**(3): 273–291.

Zheng, Z., B. Padmanabhan. 2006. Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Sci.* **52**: 697–712.

Zipkin, P. H. 2000. *Foundations of Inventory Management*. McGraw-Hill, Boston.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Appendix A. Online Appendix: A Tutorial on Building a Blockchain-Enabled Data Exchange

Please Cite this article in press as: Wang, Z., et al. Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutoria. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13356