Information Sharing for Supply Chain Management based on Block Chain Technology

Mitsuaki Nakasumi
Faculty of Economics
Komazawa University
Tokyo, Japan
nakasumi@komazawa-u.ac.jp

***Abstract*— Supply Chain Management systems provide information sharing and analysis to companies and support their planning activities. They are not based on the real data because there is asymmetric information between companies, then leading to disturbance of the planning algorithms. On the other hand, sharing data between manufacturers, suppliers and customers becomes very important to ensure reactivity towards markets variability. Especially, double marginalization is a widespread and serious problem in supply chain management. Decentralized systems under wholesale price contracts are investigated, with double marginalization effects shown to lead to supply insufficiencies, in the cases of both deterministic and random demands. This paper proposes a blockchain based solution to address the problems of supply chain such as Double Marginalization and Information Asymmetry etc.**

Keywords— supply chain management; blockchain; homomorphic encryption

I. INTRODUCTION

In the past, most manufacturers produced almost everything in house. But now, high pressure on the prices and the global competition forced them to focus on their core competences like engineering and final assembly as original equipment manufacturers thus outsourcing almost the whole manufacturing operations. These higher levels of complexity are the result of dramatic changes in manufacturing and distribution, including globalization and outsourcing. As a result, independent firms manage different parts of global supply chains. Each firm in the supply chain sets strategic and operational goals to maximize its own profit by using local information such as cost structures, profit margins and forecasts. Even though advances in information technology enable firms to collect, process, and share information, firms may be reluctant to do so because of conflicting incentives. Aligning incentives improves firms' profits and sustains the use of information technology.

In this situation, it is important to build competitive supply chain. To build it, Information in supply chains is one of the most valuable resources for manufacturers. Due to the huge amount of produced and exchanged data needed for the production activities, it is essential to identify the most useful ones and to focus only on the "strategic transaction" leading to potential improvements at the supply chain level. The coordination of information, as well as operations and logistics optimization, has become increasingly more difficult with recent increases in supply chain complexity.

We need to discuss incentive problems to a major risk imbalance such as capacity risk. Because of the imbalance, the impact of capacity risk is more severe for a decentralized supply chain than for a vertically integrated supply chain. To solve this problem, we propose a blockchain based solution to address the double marginalization problem.

The structure of the paper is as follows; Section II describes the problem of supply chain; section III proposes our solution, whereas section IV describes related works, and concluding remarks are found in section V.

II. PROBLEM OF SUPPLY CHAIN

Demand forecasting is becoming difficult because of short product life cycles and long production lead-times. Then, supply chains face the risk of either excess capacity due to low demand realization or lack of product availability. In a decentralized supply chain, lack of proper capacity risk sharing increases the cost of capacity risk. To deliver on time, the contract manufacturer secures capacity in advance of an original equipment manufacturer order. For such a supply chain, if consumer demand turns out to be high, both the contract manufacturer and the original equipment manufacturer face upside capacity risk. However, if consumer demand turns out to be low, only the contract manufacturer faces downside capacity risk.

To reduce capacity risk for each party depends on the contractual agreements. Under a wholesale price contract, the original equipment manufacturer pays a wholesale price w to the contract manufacturer for each unit ordered and sells the product to the market at r per unit. The contract manufacturer secures capacity at a unit cost of c, which could represent an equivalent annual cost of capacity. So, the contract manufacturer's marginal profit w-c is less than the vertically integrated supply chain's marginal profit r-c. This difference is known as double marginalization. The contract manufacturer protects itself by securing less capacity than what would be optimal for a vertically integrated supply chain. The original equipment manufacturer may eliminate this adverse effect of decentralization by sharing the contract manufacturer's upside

2378-1971/17 $31.00 © 2017 IEEE
DOI 10.1109/CBI.2017.56

IEEE computer society

capacity risk. Thus, the contract manufacturer's marginal cost is *c*, whereas the original equipment manufacturer's marginal cost is zero.

To maximize profit of each party, the original equipment manufacturer can agree to pay back *p* per unit of unused capacity. This would reduce the contract manufacturer's marginal cost to *c-p* and induce the contract manufacturer to build a higher capacity, thus aligning incentives. We refer to this as a payback contract.

The severity of the risk depends on demand forecast information asymmetry. For example, under a wholesale price contract, the original equipment manufacturer may influence the contract manufacturer's capacity decision by increasing the demand forecast. Increasing the demand forecast does not change the original equipment manufacturer's behavior by capacity risk but reduces its capacity risk exposure. Then, the contract manufacturer does not consider forecast information by the original equipment manufacturer to be credible. Under a wholesale price contract, lack of credible forecast information sharing makes insufficient capacity when the original equipment manufacturer places actual orders. Since forecast manipulation is widespread in many industries, most manufacturers often submit "phantom orders" to induce their suppliers to secure more capacity.

Through observation of several industries, the unit cost of capacity and the degree of forecast information asymmetry are two primary drivers of capacity risk. There exist two types of contracts that enable credible forecast information sharing.

The first contract type is a capacity reservation contract, which holds the original equipment manufacturer accountable for its forecast information by requiring a fee for reserving capacity. The contract manufacturer provides this contract as a menu of fees for corresponding capacity level that the original equipment manufacturer may reserve. The optimal reservation price has the characteristics of a quantity discount.

The second contract type is an advance purchase agreement, which provides an option to the original equipment manufacturer to place firm orders at an advance purchase price before the contract manufacturer secures capacity. This agreement credibly signals the original equipment manufacturer's forecast and induces the contract manufacturer to secure the necessary capacity. Depending on the per unit cost of capacity and the degree of forecast information asymmetry, original equipment manufacturer and contract manufacturer can choose among structured agreements that enable a mutually beneficial partnership. When degree of forecast information asymmetry is middle level, capacity reservation contract is preferred. And when degree of forecast information asymmetry is high and capacity expansion cost is low, Advance purchase contract is preferred.

When forecast information between the parties is highly imbalanced and per unit cost of component capacity is low, then these agreements allow that the advanced purchase contract generates higher profits for both parties.

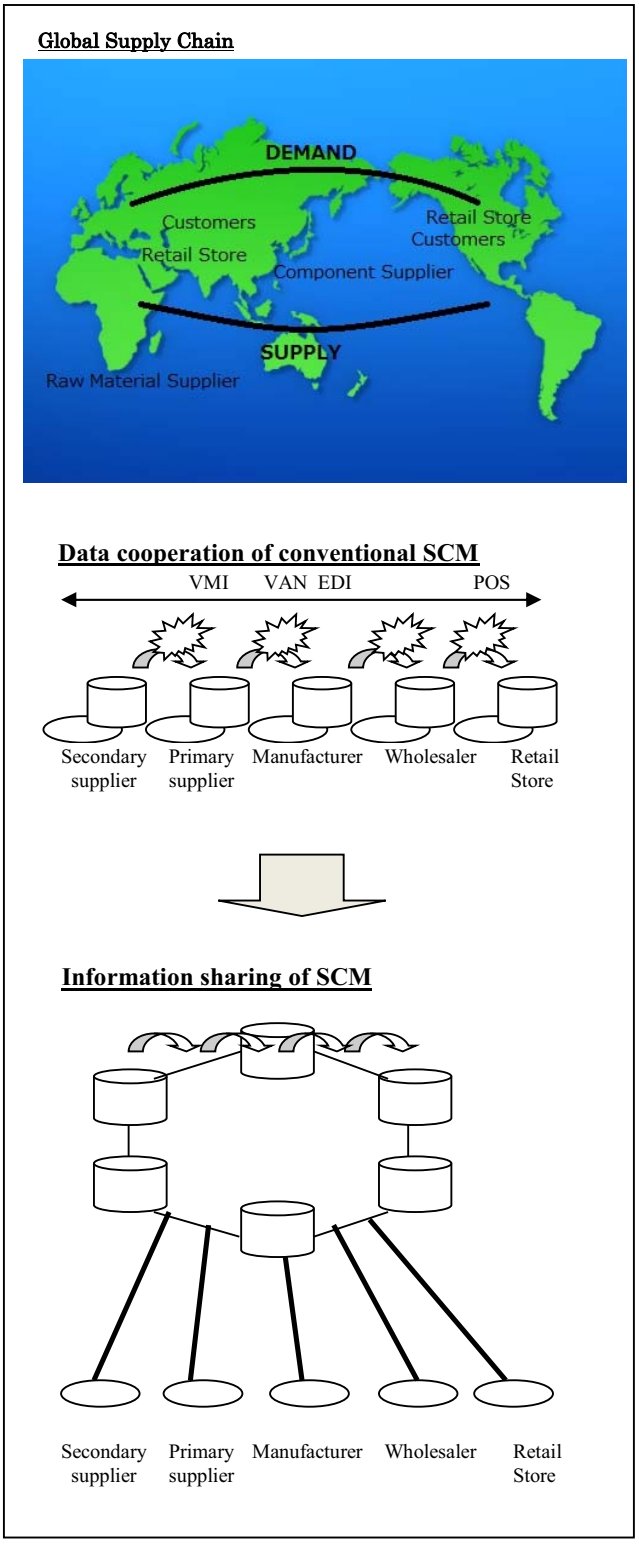

Figure 1: Shared Transaction on Supply Chain Management

However, supply chain is complicated by today's global trading systems as shown at the top in Figure 1. International

shipment cost is becoming expensive and more variable than a domestic shipment. Then we need to improve the contract types to reduce capacity risk for global supply chain.

As shown at the middle in Figure 1, there are many data exchange systems for Supply Chain Management such as POS (Point of Sales), EDI (Electronic Data Exchange), VAN (Value Added Network) and VMI (Vendor Managed inventory). These systems realize only systematization of current applications and the individual optimization within the company.

The information about object (material, product etc.) is succeeded between players like a baton relay on the supply chain. furthermore, peculiar judgment and intention are added by each player. In other words, object is only one from beginning to end of supply chain, but several kind of information will exist on supply chain. Therefore communication takes more time and the freshness of the information is declined.

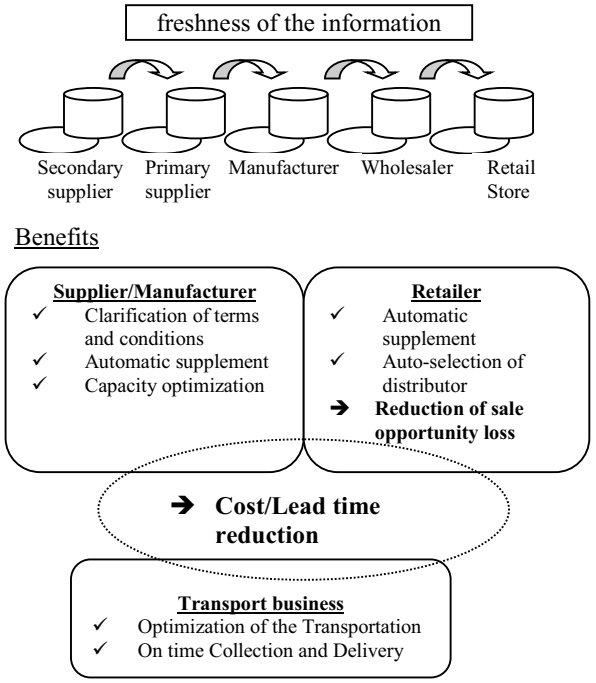

Figure 2: Benefits of Information Sharing

When it can keep the freshness of the information, Supply chain earns the benefits as shown in figure 2.

For supplier and manufacturer, it is possible to adjust production with manufacturers because supplier can acquire downstream information more from makers. Supplier will always take stock risks when manufacturer produce product based on the predictive information from the maker as before, but he can avoid it by information sharing.

For Retailer, because supply chain can share ordering from the situation of POS in real time, it is possible to supply product to a store automatically. It can reduce supplement lead time and loss of sale opportunity. The operation cost can decrease by omitting ordering duties.

Transport business can optimize both duties in the storage and arrangement of the asset (person and truck) by sharing the plan information of the shipment at an early stage. Collection, delivery and transportation on time make reduction of transportation cost and improve distribution service level. We can explain another benefit of information sharing as well. For example, In case of some components' delay, manufacturer may order substitutes according to the estimated delay time. But it is difficult to estimate the arrival time without finding material flow on real time. As a result, it makes loss of sales opportunity and manufacturing cost. In case of global supply chain, physical transport and logistics activities become widespread. There are the parties concerned and each party grasps material flow within its responsibility. It means shipping agent grasps their ship's location information and land cargo carrier traces their truck's position, but nobody can understand the status of whole supply chain. There needs efforts to find material flow because each agent provides only web service to trace each flow.

To reduce inventory carrying costs and improve supply chain efficiencies, "Visibility" of supply chain is needed. Visibility of supply chain is defined as the openness of specific information related to product orders and physical shipments, including transport and logistics activities. Visibility makes reducing costs and improving operational performance via multi-tiered global supply demand networks.

It is necessary to consider accurate analytics to reduce variability and eliminate dwell time as well. To optimize trade lane performance, you also need accurate visibility into individual order and shipment status. And it must realize flexible inventory adjustment via global supply chain. For optimized inventory management, it is necessary to find the status of inventory as "what, where and how many" on real time. We need to discuss how to realize the situation.

III. PROPOSED SOLUTION

Previous research works have discussed the benefits of information sharing throughout the supply chain. Sharing data such as machine loads, sales previsions and inventory positions has proven to improve the fulfil rate and the product cycle time, and to decrease order fluctuations.

However, it is difficult to share information in global supply chain because there are many code schemes. Using EDI network is an easy solution to integrate code schemes and realize visibility of supply chain, but it is expensive especially for small businesses.

If they try to integrate their code schemes and realize visibility of supply chain by using same ERP package such as SAP, it makes another problem. Most companies don't necessarily want to share information, because they don't want to share their capacity with competitors. It is also necessary to consider about access control on information sharing scheme. It can be difficult to receive and interpret status updates from numerous carriers, brokers, and freight forwarders to gain a

comprehensive perspective and assess performance and bottlenecks. Without this bird-eye's perspective, it becomes nearly impossible to implement cost-saving strategies, such as just-in-time inventory replenishment. When delivery windows are tight, even minor missteps and miscalculations can have major cost and service level consequences.

For satisfying these requirements, we consider low cost and access controllable database system. In recent years, a new distributed database system emerged. The system was Bitcoin [1], which allows users to transfer currency (bitcoins) securely without a centralized regulator, using a publicly verifiable open ledger (or blockchain). Since then, Bitcoin demonstrated how these blockchains can serve other functions requiring trusted computing and auditability.

While companies earn the benefits of information sharing via blockchain, there is growing a company's concern about protection of the order content. So, there is a possibility to satisfy the requirements as mentioned above.

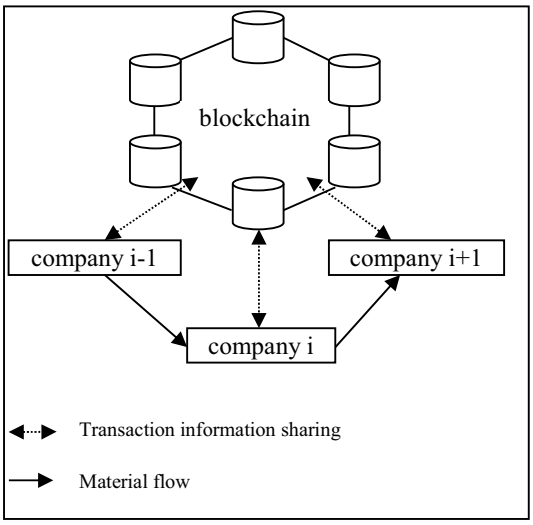

Figure 3: System Overview

We begin to describe an overview of our proposed system. As illustrated in Figure 3, there are two entities comprising our system. One is company which is interested in building supply chain and another is blockchain node. The entities entrusted with maintaining the blockchain and a distributed public/private protected data store in return for incentives.

In general, information sharing scheme is exclusively tied to the major IT companies serving as the trusted third party who process and mediate any electronic transaction.

The role of trusted third party is to validate, safeguard and preserve transactions. A certain percentage of trouble is unavoidable in online transactions and that needs mediation by transactions. This results in high transaction costs.

Bitcoin mechanism in our system uses cryptographic proof instead of the trust in the third party for two willing parties to execute an online transaction over the Internet.

Each transaction is protected through a digital signature. Each transaction is sent to the public key of the receiver digitally signed using the private key of the sender.

In order to ship material, owner of the crypto-invoice needs to prove the ownership of the private key. The entity receiving the digital invoice verifies the digital signature (ownership of corresponding private key) on the transaction using the public key of the sender.

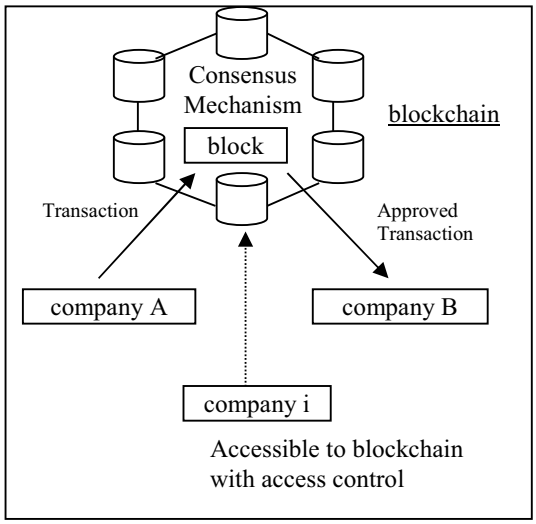

Figure 4: Transaction Process Overview

Figure 4 shows the overview of transaction process. Company A wants to send invoice to company B. The transaction is represented online as a record as mentioned later. The transaction is broadcast to every client in the network. Those clients in the network approve the transaction is valid. Then the transaction can be added to the chain, which provides an indelible and transparent record of transactions. The invoice moves from company A to company B.

For sharing the information about the object (material or product) between parties as "one data" in real time, we need to modify the block.

To understand the solution, we give examples of the Bitcoin's implementation. Bitcoin uses cryptographic proof instead of the trust-in-the-third-party mechanism for two willing parties to execute an online transaction over the Internet. A transaction is a line of code that has 4 components:

1. input: The origin of the amount transacted. (i.e.,vender)
2. output: The destination of the amount transacted (i.e.,customer)
3. amount: Quantity of the unit transacted (in Bitcoin blockchain, the unit is a *bitcoin*)
4. metadata: Additional information that can be stored along with the transaction (in Bitcoin, this is where relevant information can be added. Generally the

metadata consists of a number that refers to a version of digital assets, also called “hash”)

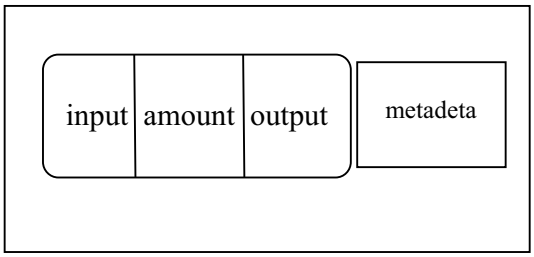

Figure 5: Representation of the Transaction Components

Figure 5 is a simple representation of the transaction. Each transaction is protected through a digital signature, is sent to the public key of the receiver, and is digitally signed using the private key of the sender. In order to spend money, the owner of the cryptocurrency needs to prove his ownership of the private key. The entity receiving the digital currency then verifies the digital signature, which implies ownership of the corresponding private key, by using the public key of the sender on the respective transaction.

Each transaction is broadcasted to every node in the Bitcoin network and is then recorded in a public ledger after verification. Every single transaction needs to be verified for validity before it is recorded in the public ledger. The transactions don't come in generated order. Then it is necessary for a system to make sure that double-spending of the crypto-currency does not occur. Considering that the transactions are passed node by node through the Bitcoin network, there is no guarantee that orders in which they are received at a node are the same order in which these transactions were generated. The above means that there is a need to develop a mechanism so that the entire Bitcoin network can agree regarding the order of transactions in a distributed system.

Our proposed block is as almost same as Bitcoin's block. However, for realizing "one data", we need to customize the block as described later.

The Bitcoin solved this problem by a mechanism that is now popularly known as blockchain technology. The Bitcoin system orders transactions by placing them in groups called blocks and then linking these blocks through what is called blockchain. The transactions in one block are considered to have happened at the same time.

A block is a group of transactions. Each block is composed of the number of transactions broadcasted to the network in 10 minute increments. The 10-minute time span is a parameter of the Bitcoin protocol and is equivalent to the average time required to validate a block. Each block has its own name, or header. This header has information about its own block (transactions + a counter called nonce) and the previous block (Hash *n-1*).

Thus, a blockchain is a group of blocks that are linked by the hash of the block headers. As explained above, each block has a header containing information about its block and some bit of information about the previous block. That piece of information is the hash of the header of the previous block. Figure 6 depicts this relationship. If there is any change to the input of a hash function, the output is completely different.

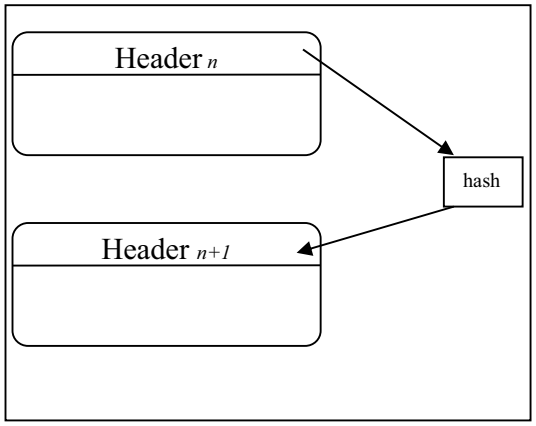

Figure 6: Representation of the Link between 2 blocks

Therefore, if someone wanted to change a past transaction, the system would reject it because the hash of the header of the block containing the most recent transaction would differ from the other versions of the blockchain on the other nodes. The network nodes communicate with one another in the case that different blocks are created at the same time. It is important to know that along with the creation of the blockchain, other measures have been taken to avoid the creation of several different blockchains within the system.

There still remains one discussion: How to decide which block should be next in the blockchain.

Any node in the network can collect unconfirmed transactions and create a block and then broadcast it to the rest of the network as a suggestion as to which block should be the next one in the blockchain. Bitcoin solves this problem by "proof of work": a node generating a block needs to prove that it has put enough computing resources to solve a mathematical puzzle. For instance, a node can be required to find a "nonce" which when hashed with both transactions and hashes of previous blocks produces a hash with certain number of leading zeros. The average effort required is exponential in the number of zero bits required but verification process is very simple and can be done by executing a single hash.

As mentioned above, Blockchain technology ensures the elimination of the double-spend problem, with the help of public-key cryptography, whereby each agent is assigned a private key and a public key shared with all other agents. A transaction is initiated when the future owner of the bitcoins (or digital tokens) sends its public key to the original owner. The coins are transferred by the digital signature of a hash. Public keys are cryptographically generated addresses stored in the blockchain. Every coin is associated with an address, and a transaction is simply a trade of coins from one address to another. The preferable feature of the blockchain is that public

keys are never tied to a real-world identity. Transactions are traceable but enabled without disclosing one's identity; this is a major difference with transactions in fiat currencies. This mathematical puzzle is not trivial to solve and the complexity of the problem can be adjusted so that on average it takes ten minutes for a node in the Bitcoin network to make a right guess and generate a block.

There is very small probability that more than one block will be generated in the system at a given time. To solve the problem, The first node broadcasts the block to the rest of the network. Occasionally, more than one block will be solved at the same time, leading to several possible branches. However, the math process needed to be solved is very complicated then the blockchain quickly stabilizes. At the completion of this, every node is in agreement about the ordering of blocks.

The nodes donating their computing resources to solve the puzzle and generate blocks are called "miner nodes" and are financially awarded for their efforts. The network only accepts the longest blockchain as the valid one. Hence, it is next to impossible for an attacker to introduce a fraudulent transaction since it has not only to generate a block by solving a mathematical puzzle, but it also has to race mathematically against the good nodes to generate all subsequent blocks in order for it to make the other nodes in the network accept its transaction and block as the valid one.

This job becomes even more difficult since blocks in the blockchain are linked cryptographically together. There can be multiple blocks created by different nodes at the same time. One can't rely on the order since blocks can arrive at different orders at different points in the network.

To illustrate the solution, consider the modified blockchain. Transaction data (i.e. vender, amount, customer/product and conditions) is encrypted using a secret encryption key and sent to the blockchain. Conditions are agreements with vendors about contract types (such as capacity reservation contract and advance purchase agreement), prices, surcharges and discounts, and so on. Conditions can be maintained when entering quotations, info records, outline agreements (contracts, scheduling agreements) and purchase orders. The net and effective prices in a purchasing document are determined on the basis of these conditions.

The modified blockchain accepts a new type of transaction for calculation, used for calculating total of selected amount. Any companies can now query the calculated data using a query transaction with the keyword (i.e. product, contract type etc.) associated to it. It includes homomorphic encryption scheme [3].

The purpose of homomorphic encryption is to allow computation on encrypted data. Thus data can remain confidential while it is processed, enabling useful tasks to be accomplished with data residing in untrusted environments. In a world of distributed computation networking, this is a hugely valuable capability. Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertexts and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintexts. This is a desirable feature in the system architectures. Homomorphic encryption allows for restricted computations on encrypted data. Computing on encrypted data means that if a user has a function f and want to obtain $f(m_1, \ldots, m_n)$ for some inputs $m_1, \ldots, m_n$, it is possible to instead compute on encryptions of these inputs, $c_1, \ldots, c_n$, obtaining a result which decrypts to $f(m_1, \ldots, m_n)$. In some cryptosystems the input messages (plaintexts) lie within some algebraic structure, often a group or a ring. In such cases the ciphertexts will often also lie within some related structure, which could be the same as that of the plaintexts.

We describe in detail the underlying protocol used in the system. We utilize standard cryptographic building blocks in our platform: a homomorphic encryption scheme defined by the generator, encryption and decryption algorithms respectively; a digital signature scheme described by the generator, signature and verification algorithms respectively.

A. Building Blocks

1) Identities: Blockchains utilize a pseudo-identity mechanism. We introduce compound identities, an extension of this model used in our system. A compound identity is a shared identity for two or more parties, where some parties (at least one) own the identity (owners), and the rest have restricted access to it (guests). The identity is comprised of a homomorphic encryption used to encrypt (and decrypt) the data, so that the data is protected from all other players in the system.

2) Blockchain Memory: We let L be the blockchain memory space, and can store sufficiently large documents. We assume this memory to be tamperproof under the same adversarial model used in Bitcoin and other blockchains. To intuitively explain why such a trusted data-store can be implemented on any blockchain (including Bitcoin), consider the following simplified implementation: A blockchain is a sequence of timestamped transactions, where each transaction includes a variable number of output addresses. L could then be implemented as follows – the first two outputs in a transaction encode the memory address pointer, as well as some auxiliary meta-data. The rest of the outputs construct the serialized document. When looking up L, only the most recent transaction is returned, which allows update and delete operations in addition to inserts.

B. Blockchain Process

Here we provide a detailed description of the core process executed on the blockchain. Figure 7 shows a representation of the record in transaction.

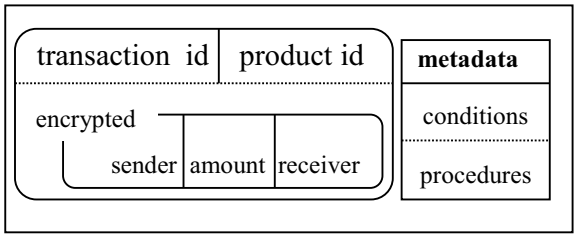

Figure 7: Representation of Transaction

It contains sender, amount, receiver, transaction id, product id and metadata. Sender, amount and receiver are components of invoice and these are encrypted because the disclosure of these data leads to their disadvantage. Sender and receiver are the parties as supplier, manufacturer, transport business and retailer in supply chain.

Metadata contains the address both conditions and procedures which verifies and/or enforces terms and conditions in the contract.

Condition allows procedures to set trigger events according to terms and conditions in the contract, such as delivery delay. Procedure is executed by nodes in the blockchain network when trigger event is received.

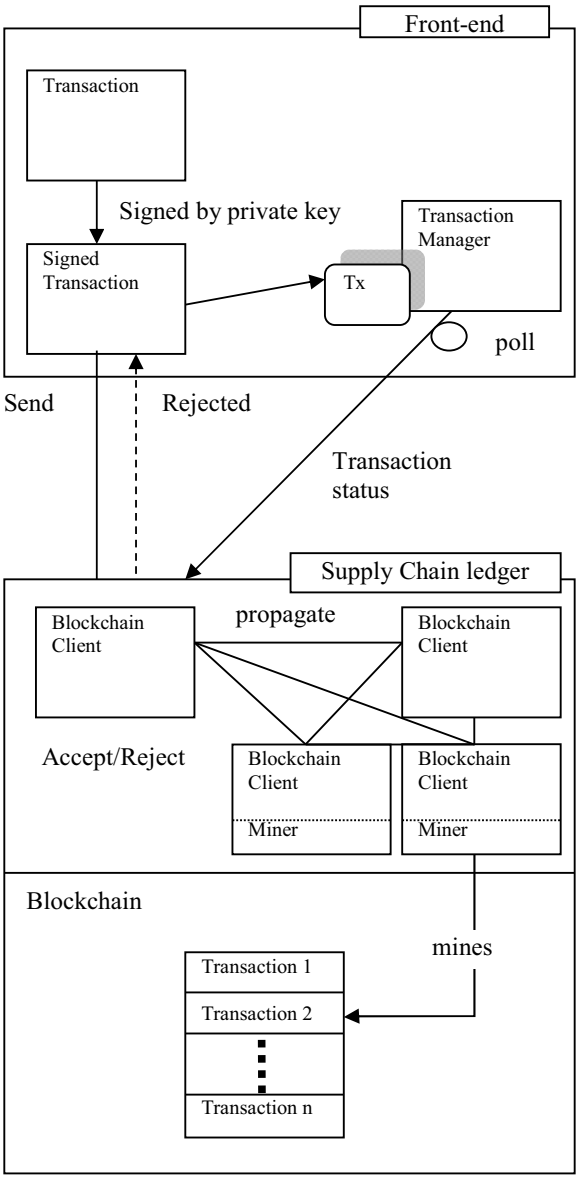

Figure 8: Overview of Transaction Process

Figure 8 shows an overview of transaction process as follows.

1. Frond end application signs and issues transaction to supply chain ledger. Supply chain ledger is the blockchain network and it receives transactions related to the supply chain. He also adds a "transaction id" to the record.
2. Client verifies the transaction and propagates the transaction to other clients.
3. Client also earns the hash for the transaction.
4. Miner can make new block this and other transaction.
5. Transaction Manager on Front-end watch the status of incomplete transactions on its memory space.
6. Transaction Manager creates new transaction when trigger event occurs.

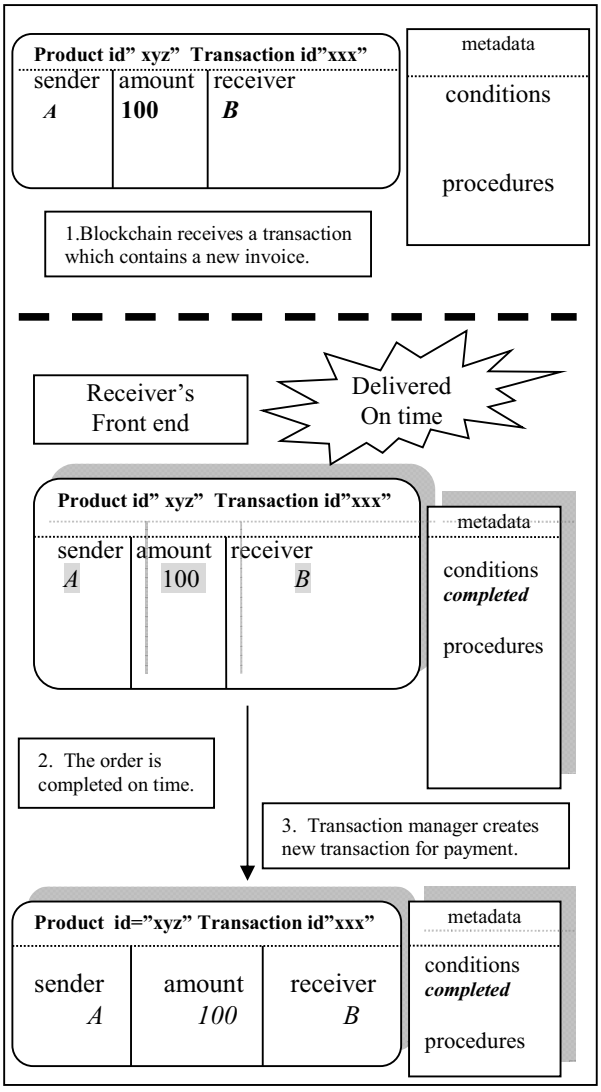

Figure 9: Transaction Completed as normal

”One data” in Supply chain means the set of transactions by supplier, manufacturer and retailer. Transactions for the product are associated with transaction id and product id. Product id field contains both product name and code. Transaction id is to manage the status of transaction. When a transaction is completed, Transaction Manager deletes the transaction from its memory.

Now, we explain some examples to understand well.

The process in Figure 9 shows the normal transaction. The flow of the transaction is as follows;

1. Receiver receives the product.
2. Transaction Manager on Front-end seeks incomplete transactions for the product on its memory space.
3. Transaction Manager copies the transaction and deletes from its memory space.
4. Transaction Manager creates new transaction using the copy.
5. The blockchain client adds this transaction to new block.

Figure 10 shows calculation process with condition “total amount”. When a company wants to find total amount of delayed shipments of “xyz”, the company sends a transaction (product=”xyz”) to the blockchain. The flow of the transaction is as follows;

1. Transaction manager creates a query transaction with pay for mining procedure.
2. Miner in the blockchain network seeks incomplete transactions for the query.
3. Miner extracts and sums up all amounts of ordered product.
4. Miner returns the result of request.
5. Transaction manager confirms the result and releases “pay for mining” procedure to pay the charge. The incentive of the process can be funded with transaction fees.

Figure 11 shows emergency order process. In spite of the worst situation, it is necessary to keep on supplying. The flow of the transaction is as follows;

1. Blockchain network receives a transaction which contains delay condition.
2. Transaction manager extracts delayed order.
3. Transaction manager creates a new transaction for emergency order.
4. Transaction manager finds emergency transaction.
5. Transaction manager sends emergency order transaction to the blockchain.

Transaction manager confirms the result and releases “pay for mining” procedure to pay the charge.

For having provided for the emergency order, the firm considers about stochastic amount of the product. When it remains lower level, it must be made a capacity reservation contract and advance purchase agreement because we must avoid high volatility. It makes purchase cost higher, but it defines the lower bound of supply cost.

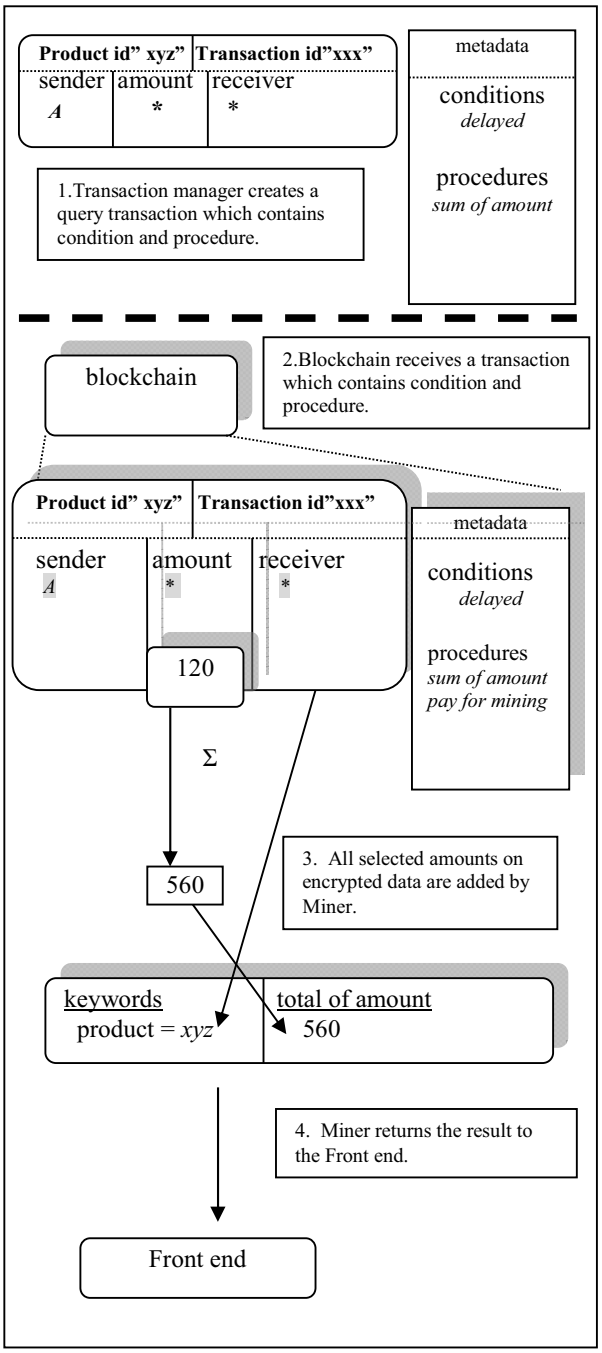

Figure 10: Calculation Transaction

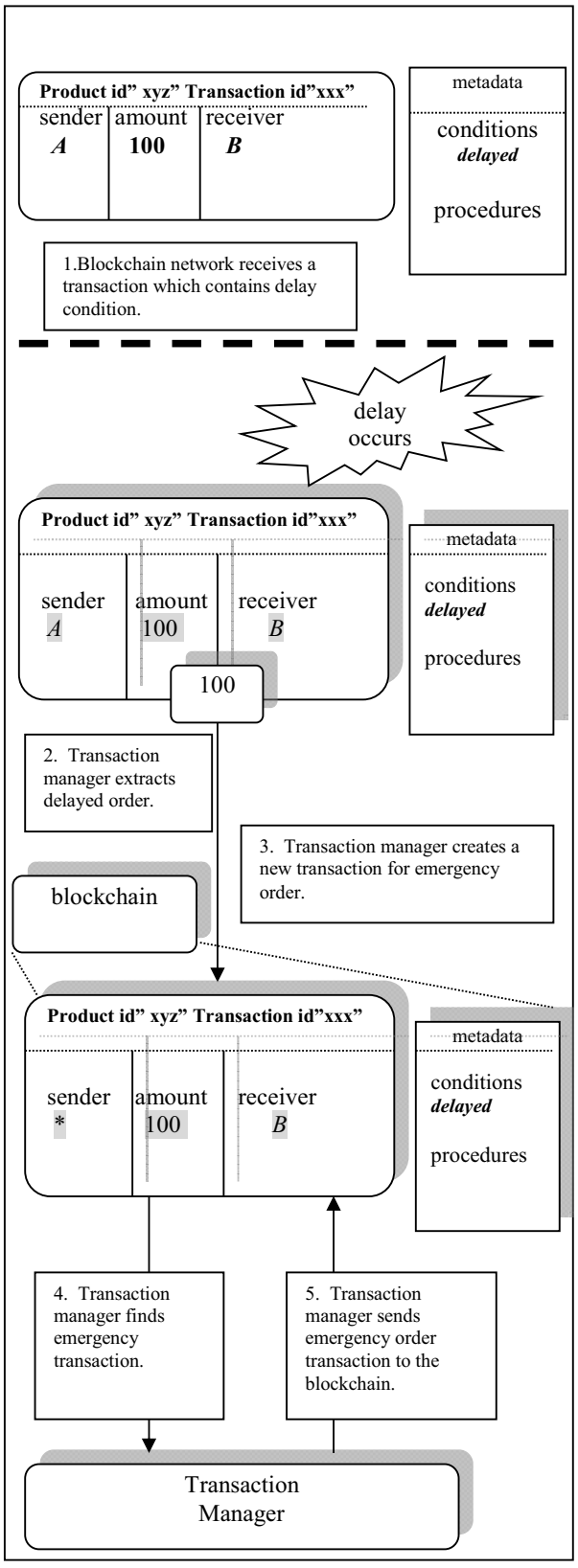

Figure 11: Emergency Order Process

IV. RELATED WORKS

There have been various attempts to address data protection issues. Across the industry, leading companies chose to implement their own proprietary authentication software based on the OAuth protocol [6], in which they serve as centralized trusted authorities. From a security perspective, researchers developed various techniques targeting data protection concerns focused on transaction data. Data anonymization methods attempt to protect personally identifiable information. k-anonymity, a common property of anonymized datasets requires that sensitive information of each record is indistinguishable from at least k−1 other records [7]. Related extensions to k-anonymity include l-diversity, which ensures the sensitive data is represented by a diverse enough set of possible values [11]; and t-closeness, which looks at the distribution of sensitive data [12]. Recent research has demonstrated how anonymized datasets employing these techniques can be de-anonymized [9], [14], given even a small amount of data points or high dimensionality data. Other data-preserving methods include differential protection, a technique that perturbs data or adds noise to the computational process prior to sharing the data [10], and encryption schemes that allow running computations and queries over encrypted data.

As you know, there are similar schemes such as smart contract. NXT [31] is an public blockchain platform which includes a selection of smart contracts that are currently live. Ethereum [30] is a public blockchain platform which is currently the most advanced smart contract enabled blockchain. With a "Turing complete" coding system, theoretically you can put any logic into an Ethereum smart contract, and it will be run by the whole network. There are mechanisms in place to prevent abuse, and you need to pay for compute power, by passing in "ETH" tokens, which act as payment for the miners who run your code. Enigma [32] provides the first solution for protecting data-in-use. Share data with others for processing without actually giving it away. Data are guaranteed to be encrypted at all times, even when complex analytics are required.

Our blockchain scheme has no valuable things such as virtual currency to avoid hacking. Miner can earn the transaction fee and it uses only computational power in the network.

V. CONCLUSION

In this paper, we proposed a new blockchain scheme for information sharing. It brings many benefits for supply chain management.

In general, Transaction data should not be trusted in the hands of third-parties, where they are susceptible to steals and misuse. Instead, users should own and control their data without compromising security or limiting companies' and authorities' ability to provide encrypted transactions.

Our platform enables this by combining a blockchain with a homomorphic encryption solution. Users are not required to trust any third-party and are always aware of the data that is being collected about them and how it is used. In addition, the

blockchain recognizes the users as the owners of their encrypted data. Companies, in turn, can focus on utilizing data without being overly concerned about properly securing and compartmentalizing them.

Furthermore, with a decentralized platform, making legal and regulatory decisions about collecting, storing and sharing sensitive data should be simpler. Moreover, laws and regulations could be programmed into the blockchain itself, so that they are enforced automatically. In other situations, the ledger can act as legal evidence for accessing (or storing) data.

We recognize some problems to be solved. For example, Search operation for emergency order brings heavy load to Miner. We need to consider about efficient incentive mechanism.

REFERENCES

[1] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf [Accessed March 2017]

[2] R. L. Rivest, L. Adleman, and M. L. Dertouzos, On data banks and privacy homomorphisms, Foundations of Secure Computation, Academia Press, pages 169–179, 1978.

[3] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inf. Theory 31(4), 469–472 (1985)

[4] R. Cramer, V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, in Proceedings of Advances in Cryptology, CRYPTO'98, 1998, pp. 13–25

[5] M. Abdalla, M. Bellare, P. Rogaway, DHAES: an encryption scheme based on the Diffie–Hellman problem. Submission to IEEE P1363a, 1998. http://www.di.ens.fr/~mabdalla/papers/dhes.pdf

[6] Juan Perez, Facebook, google launch data portability programs to all,2008.

[7] Latanya Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557–570, 2002.

[8] D. Boneh, E. Goh, K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in Proceedings of Theory of Cryptography, TCC'05, 2005, pp. 325–341

[9] Arvind Narayanan and Vitaly Shmatikov, How to break anonymity of the netflix prize dataset. arXiv preprint cs/0610105, 2006.

[10] Cynthia Dwork, Differential privacy, in Automata, languages and programming, pages 1–12. Springer, 2006.

[11] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam, l-diversity: Privacy beyond kanonymity, ACM Transactions on Knowledge Discovery from Data(TKDD), 1(1):3, 2007.

[12] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian, t-closeness:Privacy beyond k-anonymity and l-diversity, in ICDE, volume 7, pages 106–115, 2007.

[13] D.M. Freeman, Converting pairing-based cryptosystems from composite-order groups to prime-order groups, in Proceedings of Advances in Cryptology, EUROCRYPT'10, 2010, pp. 44–61

[14] Yves-Alexandre de Montjoye, C´esar A Hidalgo, Michel Verleysen, and Vincent D Blondel, Unique in the crowd: The privacy bounds of human mobility, Scientific reports, 3, 2013.

[15] Barratt M and Oke A, Antecedents of supply chain visibility in retail supply chains: a resource-based theory perspective. Journal of Operations Management 25, 1217–1233, 2007

[16] Cagliano R, Caniato F and Spina G, E-business strategy: how companies are shaping their supply chain through the Internet. International Journal of Operations & Production Management 23 (10), 1142–1162, 2003

[17] Craighead CW, Patterson JW, Roth PL and Segars AH, Enabling the benefits of supply chain management systems: an empirical study of electronic data interchange (EDI) in manufacturing. International Journal of Production Research 44 (1), 135–157, 2006

[18] Decarolis DM and Deeds DL, The impact of stocks and flows of organizational knowledge on firm performance: an empirical investigation of the biotechnology industry. Strategic Management Journal 20 (10), 953–968,1999

[19] Dyer JH, Collaborative Advantage: Winning through Extended Enterprise Supplier Networks. Oxford University Press, New York, 2000

[20] Elgarah W, Falaleeva N, Saunders CC, Ilie V, Shim JT and Courtney JF , Data exchange in interorganizational relationships: review through multiple conceptual lenses. The DATA BASE for Advances in Information Systems 36 (1), 8–29,2005

[21] Gosain S, Malhotra A and El Sawy OA, Coordinating for flexibility in e-business supply chains. Journal of Management Information Systems 21 (3), 7–45, 2004

[22] Gunasekaran A, Patel C and Tirtiroglu E, Performance measures and metrics in a supply chain environment. International Journal of Operations & Production Management 21 (1/2), 71, 2001

[23] Handfield RB and Bechtel C, The role of trust and relationship structure in improving supply chain responsiveness. Industrial Marketing Management 31 (4), 367–382, 2002

[24] Handfield RB and Pannesi RT, Antecedents of lead time competitiveness in make-to-order manufacturing firms. International of Journal of Production Research 33 (2), 511–537, 1995

[25] Ho DC, Au KF and Newton E, Empirical research on supply chain management: a critical review and recommendations. International Journal of Production Research 40, 4415–4430, 2002

[26] Huang GQ, Lau JSK and Mak KL, The impacts of sharing production information on supply chain dynamics: a review of the literature. International Journal of Production Research 41 (7), 1483–1517,2003

[27] Hult GTM, Ketchen DJ and Nichols EL, Organizational learning as a strategic resource in supply chain management. Journal of Operations Management 21 (5), 541–556,2003

[28] Rai A, Patnayakuni R and Patnayakuni N, Firm performance impacts of digitally enabled supply chain integration capabilities. MIS Quarterly 30 (2), 225–246, 2006

[29] Sahin F and Robinson EP (2002) Flow coordination and information sharing in supply chains, review, implications, and direction for future research. Decision Sciences 33 (4), 505–536, 2006

[30] https://www.ethereum.org/ [Accessed May 2017]

[31] https://nxt.org/ [Accessed May 2017]

[32] http://www.enigma.co/ [Accessed May 2017]