

METHODS AND DEVICES FOR PROVIDING PRIVACY-PRESERVING SUPPLY CHAIN FINANCING

TECHNICAL FIELD

[0001] The specification relates generally to computer technologies, and more particularly, to methods and devices for providing privacy-preserving supply chain financing.

BACKGROUND

[0002] Supply chain financing provides a way for businesses to borrow money to facilitate trades or purchases. For example, a buyer may order a product through a seller, who may in turn order the product from a supplier. If the supplier requires the seller to pay for the product before the seller receives payment from the buyer, the seller may need to borrow money to pay for the product. The seller may, for example, borrow money from a financial institution against an account receivable, including, e.g., the amount due from the buyer. The seller may then use the borrowed money to pay the supplier for the product, collect the amount due from the buyer, and repay the financial institution.

[0003] In some situations, the parties involved (including, e.g., the seller, the buyer, the supplier, and the financial institution) may want to keep records of the purchases and financial transactions for various purposes. For example, the seller may want to keep its invoices issued to the buyer and use the invoices to secure financing. The financial institution may want to have access to such invoices when determining whether to approve loan applications requested by the seller. The parties may also want to keep

other types of records for various reasons.

[0004] Blockchain systems, also known as distributed ledger systems (DLSs) or consensus systems, may enable participating parties to store data securely and immutably. Blockchain systems may include any DLSs, without referencing any particular use case, and may be used for public, private, and consortium blockchain networks. A public blockchain network is open for all entities to use the system and participate in the consensus process. A private blockchain network is provided for a particular entity, which centrally controls read and write permissions. A consortium blockchain network is provided for a select group of entities, which control the consensus process, and includes an access control layer.

[0005] A blockchain system is implemented using a peer-to-peer (P2P) network, in which the nodes communicate directly with each other, e.g., without the need of a fixed, central server. Each node in the P2P network may initiate communication with another node in the P2P network. A blockchain system maintains one or more blockchains. A blockchain is a data structure for storing data, such as transactions, that may prevent tampering and manipulation of the data by malicious parties.

[0006] Users of a blockchain system may utilize the blockchain system to record various types of information. For example, a seller may use the blockchain system to record an invoice issued to a buyer. A financial institution may access the invoice recorded on the blockchain system to determine whether to approve a loan application requested by the seller. However, because data recorded on blockchains are accessible to other users, sensitive financial information may be exposed.

SUMMARY

[0007] In one aspect, a computer-implemented method for providing privacy-preserving supply chain financing includes: receiving a first transaction, the first transaction containing commitment values of a plurality of terms of a loan agreement; determining whether the commitment values contained in the first transaction satisfy a set of requirements; in response to a determination that the commitment values contained in the first transaction satisfy the set of requirements, recording the first transaction on a blockchain; receiving a second transaction, the second transaction indicating a disbursement of a loan made by a lender in accordance with the loan agreement; and in response to receiving the second transaction, recording the second transaction on the blockchain and updating at least one commitment value of an account balance based on at least one of the commitment values contained in the first transaction.

[0008] In another aspect, a device for providing privacy-preserving supply chain financing includes: one or more processors; and one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to: receive a first transaction, the first transaction containing commitment values of a plurality of terms of a loan agreement; determine whether the commitment values contained in the first transaction satisfy a set of requirements; in response to a determination that the commitment values contained in the first transaction satisfy the set of requirements, record the first transaction on a blockchain; receive a second transaction, the second transaction indicating a disbursement of a loan made by a lender in accordance with the loan agreement; and in response to receiving the second transaction, record the second transaction on the

blockchain and updating at least one commitment value of an account balance based on at least one of the commitment values contained in the first transaction.

[0009] In still another aspect, a non-transitory computer-readable medium has stored therein instructions that, when executed by a processor of a device, cause the device to perform a method for providing privacy-preserving supply chain financing. The method includes: receiving a first transaction, the first transaction containing commitment values of a plurality of terms of a loan agreement; determining whether the commitment values contained in the first transaction satisfy a set of requirements; in response to a determination that the commitment values contained in the first transaction satisfy the set of requirements, recording the first transaction on a blockchain; receiving a second transaction, the second transaction indicating a disbursement of a loan made by a lender in accordance with the loan agreement; and in response to receiving the second transaction, recording the second transaction on the blockchain and updating at least one commitment value of an account balance based on at least one of the commitment values contained in the first transaction.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments. In the following description, which refers to the drawings, the same numbers in different drawings represent the same or similar elements unless otherwise represented.

[0011] FIG. 1 is a schematic diagram of a blockchain system, according to an embodiment.

[0012] FIG. 2 is a schematic diagram of a computing device for implementing a node in a blockchain system, according to an embodiment.

[0013] FIG. 3 is a flow chart of a method for providing privacy-preserving supply chain financing, according to an embodiment.

[0014] FIG. 4 is a flow chart of a method for providing privacy-preserving supply chain financing, according to an embodiment.

[0015] FIG. 5 is a block diagram of an apparatus for providing privacy-preserving supply chain financing, according to an embodiment.

DETAILED DESCRIPTION

[0016] Embodiments of the specification provide methods and devices for providing privacy-preserving supply chain financing. The methods and devices may utilize blockchain systems to record information needed to facilitate supply chain financing. The methods and devices may allow the recorded information to be represented by commitment values. The methods and devices may provide operations to be performed by various users (including, e.g., sellers, buyers, suppliers, financial institutions, etc.) to facilitate supply chain financing. The methods and devices may also verify the information submitted by the various users for recordation on the blockchain systems. In this manner, the methods and devices can facilitate supply chain financing without requiring the various users to share their private information on the blockchain systems.

[0017] Embodiments disclosed in the specification have one or more technical effects. In some embodiments, the methods and devices allow users to record

information needed to facilitate supply chain financing on a blockchain system. This allows the methods and devices to record the information in a data structure that can prevent tampering and manipulation by malicious parties. In some embodiments, the methods and devices allow the recorded information to be represented by commitment values. This allows the methods and devices to preserve user privacy because the users do not need to record sensitive financial information in plaintext on the blockchain system. In some embodiments, the methods and devices can support operations and validations based on the commitment values recorded on the blockchain system. This allows the methods and devices to provide a mechanism for the users to verify the legitimacy of the financial information based on the commitment values without exposing the underlying financial information on the blockchain system.

[0018] A blockchain is a data structure that stores data, e.g., transactions, in a way that may prevent tampering and manipulation of the data by malicious parties. The transactions stored in this manner may be immutable and subsequently verified. A blockchain includes one or more blocks. Each block is linked to a previous block immediately before it in the blockchain by including a cryptographic hash of the previous block. Each block also may include a timestamp, its own cryptographic hash, and one or more transactions. The transactions, which generally have already been verified by the nodes of the blockchain system, may be hashed and encoded into a data structure, such as a Merkle tree. In a Merkle tree, data at leaf nodes of the tree is hashed, and all hashes in each branch of the tree may be concatenated at a root of the branch. This process continues up the tree to the root of the entire tree, which stores a hash that is representative of all data in the tree. A hash purporting to be of a transaction stored in

the tree can be quickly verified by determining whether it is consistent with the structure of the tree.

[0019] A blockchain system includes a network of computing nodes that manage, update, and maintain one or more blockchains. The network may be a public blockchain network, a private blockchain network, or a consortium blockchain network. For example, numerous entities, such as hundreds, thousands, or even millions of entities, can operate in a public blockchain network, and each of the entities operates at least one node in the public blockchain network. Accordingly, the public blockchain network can be considered a public network with respect to the participating entities. Sometimes, a majority of entities (nodes) must sign every block for the block to be valid and added to the blockchain of the blockchain network. Examples of public blockchain networks include particular peer-to-peer payment networks that leverage a distributed ledger, referred to as blockchain.

[0020] In general, a public blockchain network may support public transactions. A public transaction is shared with all of the nodes in the public blockchain network, and is stored in a global blockchain. A global blockchain is a blockchain replicated across all nodes, and all nodes are in perfect state consensus with respect to the global blockchain. To achieve consensus (e.g., agreement to the addition of a block to a blockchain), a consensus protocol is implemented in the public blockchain network. Examples of consensus protocols include proof-of-work (POW) (e.g., implemented in the some cryptocurrency networks), proof-of-stake (POS), and proof-of-authority (POA).

[0021] In general, a private blockchain network may be provided for a particular entity, which centrally controls read and write permissions. The entity controls which

nodes are able to participate in the blockchain network. Consequently, private blockchain networks are generally referred to as permissioned networks that place restrictions on who is allowed to participate in the network, and on their level of participation (e.g., only in certain transactions). Various types of access control mechanisms can be used (e.g., existing participants vote on adding new entities, a regulatory authority can control admission).

[0022] In general, a consortium blockchain network may be private among the participating entities. In a consortium blockchain network, the consensus process is controlled by an authorized set of nodes, one or more nodes being operated by a respective entity (e.g., a financial institution, insurance company). For example, a consortium of ten (10) entities (e.g., financial institutions, insurance companies) can operate a consortium blockchain network, each of which operates at least one node in the consortium blockchain network. Accordingly, the consortium blockchain network can be considered a private network with respect to the participating entities. In some examples, each entity (node) must sign every block in order for the block to be valid, and added to the blockchain. In some examples, at least a sub-set of entities (nodes) (e.g., at least 7 entities) must sign every block in order for the block to be valid, and added to the blockchain.

[0023] FIG. 1 illustrates a schematic diagram of a blockchain system 100, according to an embodiment. Referring to FIG. 1, the blockchain system 100 may include a plurality of nodes, e.g., nodes 102-110, configured to operate on a blockchain 120. The nodes 102-110 may form a network 112, such as a peer-to-peer (P2P) network. Each of the nodes 102-110 may be a computing device, such as a computer or a computer

system, configured to store a copy of the blockchain 120, or may be software running on the computing device, such as a process or an application. Each of the nodes 102-110 may have a unique identifier.

[0024] The blockchain 120 may include a growing list of records in the form of data blocks, such as blocks B1-B5 in FIG. 1. Each of the blocks B1-B5 may include a timestamp, a cryptographic hash of a previous block, and data of the present block, which may be transactions such as monetary transactions. For example, as illustrated in FIG. 1, block B5 may include a timestamp, a cryptographic hash of block B4, and transaction data of block B5. Also, for example, a hashing operation may be performed on the previous block to generate the cryptographic hash of the previous block. The hashing operation may convert inputs of various lengths into cryptographic outputs of a fixed length through a hash algorithm, such as SHA-256.

[0025] The nodes 102-110 may be configured to perform an operation on the blockchain 120. For example, when a node, e.g., the node 102, wants to store new data onto the blockchain 120, that node may generate a new block to be added to the blockchain 120 and broadcast the new block to other nodes, e.g., the nodes 104-110, in the network 112. Based on legitimacy of the new block, e.g., validity of its signature and transactions, the other nodes may determine to accept the new block, such that the node 102 and the other nodes may add the new block to their respective copies of the blockchain 120. As this process repeats, more and more blocks of data may be added to the blockchain 120.

[0026] FIG. 2 illustrates a schematic diagram of a computing device 200 for implementing a node, e.g., the node 102 (FIG. 1), in a blockchain system, according to

an embodiment. Referring to FIG. 2, the computing device 200 may include a communication interface 202, a processor 204, and a memory 206.

[0027] The communication interface 202 may facilitate communications between the computing device 200 and devices implementing other nodes, e.g., nodes 104-110 (FIG. 1), in the network. In some embodiments, the communication interface 202 is configured to support one or more communication standards, such as an Internet standard or protocol, an Integrated Services Digital Network (ISDN) standard, etc. In some embodiments, the communication interface 202 may include one or more of a Local Area Network (LAN) card, a cable modem, a satellite modem, a data bus, a cable, a wireless communication channel, a radio-based communication channel, a cellular communication channel, an Internet Protocol (IP) based communication device, or other communication devices for wired and/or wireless communications. In some embodiments, the communication interface 202 may be based on public cloud infrastructure, private cloud infrastructure, hybrid public/private cloud infrastructure.

[0028] The processor 204 may include one or more dedicated processing units, application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), or various other types of processors or processing units. The processor 204 is coupled with the memory 206 and is configured to execute instructions stored in the memory 206.

[0029] The memory 206 may store processor-executable instructions and data, such as a copy of the blockchain 120 (FIG. 1). The memory 206 may include any type of volatile or non-volatile memory devices, or a combination thereof, such as a static random-access memory (SRAM), an electrically erasable programmable read-only memory (EEPROM), an erasable programmable read-only memory (EPROM), a

programmable read-only memory (PROM), a read-only memory (ROM), a magnetic memory, a flash memory, or a magnetic or optical disk. When the instructions in the memory 206 are executed by the processor 204, the computing device 200 may perform an operation on the blockchain 120.

[0030] Users of a blockchain system, e.g., the blockchain system 100, may utilize the blockchain system 100 to record various types of information. For example, in some embodiments, one or more users may utilize the blockchain system 100 to record information to facilitate supply chain financing. For example, sellers may utilize the blockchain system 100 to record information regarding their invoices and/or purchase orders, which can be utilized to secure financing. Financial institutions may utilize the blockchain system 100 to record information regarding their agreements with the sellers, who may be obligated to repay the financial institutions according to the recorded agreements. Other users, including, e.g., buyers and/or suppliers, may also utilize the blockchain system 100 to record information to facilitate supply chain financing.

[0031] FIG. 3 illustrates a flow chart of a method 300 for providing privacy-preserving supply chain financing, according to an embodiment. Referring to FIG. 3, multiple users may have accounts on a Blockchain, e.g., the blockchain 120 (FIG. 1). The Blockchain may be implemented to support various types of users, or parties, including, e.g., buyers, sellers, suppliers, banks, financial institutions, businesses, individuals, as well as other types of companies, organizations, and the like.

[0032] For illustrative purposes, multiple users are depicted in FIG. 3, which includes a Buyer, a Seller, a Supplier, a Lender, and an Administrator. At step 302, the Buyer may order one or more products from a Seller. The Seller may issue an invoice to

the Buyer, and in some embodiments, the Seller may place a purchase order at step 304 to purchase the one or more products from the Supplier in order to fulfill the Buyer's order received at step 302.

[0033] In some embodiments, the Seller may obtain a loan to help pay for the purchase order it placed with the Supplier. For example, at step 306, the Seller may submit a loan application to the Blockchain. In some embodiments, the loan application may include information regarding the invoice the Seller issued to the Buyer and/or the purchase order the Seller placed with the Supplier. In some embodiments, information regarding the invoice and/or the purchase order may be utilized as certifying documents to help secure the loan for the Seller. In some embodiments, the Seller may encrypt the information regarding the invoice and/or the purchase order before submitting the information to the Blockchain. It is to be understood that the Seller may encrypt the information in various manners.

[0034] At step 308, the Blockchain may notify the Supplier upon receipt of the loan application. In some embodiments, the Blockchain may utilize one or more smart contracts executing on the Blockchain to provide the notification. Smart contracts are computer protocols implemented in the form of computer code that are incorporated into the Blockchain, to facilitate, verify, or enforce the negotiation or performance of agreed terms or conditions. For example, users of the Blockchain may program agreed terms into a smart contract using a programming language, such as C++, Java, Solidity, Python, etc., and when the terms are met, the smart contract may be automatically executed on the Blockchain, e.g., to perform a transaction. Also for example, the smart contract may include a plurality of subroutines or functions, each of which may be a sequence of

program instructions that performs a specific task. The smart contract may be operational code that is fully or partially executed without human interaction.

[0035] In some embodiments, a smart contract may be incorporated into the Blockchain to determine whether the loan application includes information regarding a purchase order the Seller placed with the Supplier. If so, at step 308, the smart contract may notify the Supplier and request the Supplier to confirm whether the purchase order identified in the loan application is valid. The Supplier may retrieve the loan application from the Blockchain and obtain the purchase order contained therein to perform the confirmation.

[0036] In some embodiments, the Seller may encrypt the information regarding the purchase order using an encryption mechanism that can be decrypted by the Supplier. For example, the Seller may encrypt the information regarding the purchase order using the Supplier's public key, allowing the Supplier to decrypt the information using its private key. The Supplier may confirm whether the purchase order is valid and provide the confirmation to the Blockchain at step 310. In some embodiments, the Supplier may consider the purchase order to be valid if it accurately describes the product(s) ordered by the Seller and/or the amount to be paid by the Seller. In some embodiments, the Supplier may consider the purchase order to be valid if it has not been fulfilled yet. Otherwise, the Supplier may report to the Blockchain that the purchase order is invalid.

[0037] At step 312, the Blockchain may also notify the Buyer upon receipt of the loan application. In some embodiments, a smart contract may be incorporated into the Blockchain to determine whether the loan application includes information regarding an invoice issued to the Buyer. If so, at step 312, the smart contract may notify the Buyer

and request the Buyer to confirm whether the invoice is valid. The Buyer may retrieve the loan application from the Blockchain and obtain the invoice contained therein to perform the confirmation.

[0038] In some embodiments, the Seller may encrypt the information regarding the invoice using an encryption mechanism that can be decrypted by the Buyer. For example, the Seller may encrypt the information regarding the invoice using the Buyer's public key, allowing the Buyer to decrypt the information using its private key. The Buyer may confirm whether the invoice is valid and provide the confirmation to the Blockchain at step 314. In some embodiments, the Buyer may consider the invoice to be valid if it accurately describes the product(s) ordered by the Buyer and/or the amount to be paid by the Buyer. In some embodiments, the Buyer may consider the invoice to be valid if it has not been paid yet. Otherwise, the Buyer may report to the Blockchain that the invoice is invalid.

[0039] It is to be understood that the notification and confirmation steps 308-314 described above are presented as examples and are not meant to be limiting. It is contemplated that the Buyer and/or the Supplier may be notified in other manners, and they may carry out the confirmations in other manners as well. In some embodiments, the Blockchain may require at least one of the Buyer or the Supplier to provide a confirmation before proceeding forward. In other embodiments, confirmations by the Buyer and/or the Supplier may be optional.

[0040] At step 316, the Blockchain may notify the Administrator of the loan application. In some embodiments, the Blockchain may notify the Administrator if the loan application has been confirmed by at least one of the Buyer or the Supplier. In some embodiments, the Administrator may be an independent party responsible for reviewing

the loan application and assessing the Seller's profile to determine whether to accept the recordation of the loan application. In some embodiments, the Administrator may be an owner or an administrator of a system (e.g., the system 100) configured to operate on the Blockchain. In some embodiments, the Administrator may assess the Seller's profile by taking into consideration various factors, including, e.g., the size of the invoice, the Seller's credit score and credit history, the Administrator's prior dealings with the Seller, and the like.

[0041] At step 318, if the Administrator decides to accept the recordation of the loan application, the loan application may be recorded on the Blockchain and become viewable to one or more financial institutions using the Blockchain. For illustrative purposes, suppose that one of the financial institutions, referred to as the Lender, decides to make an offer to finance the loan requested by the Seller, the Lender may, at step 320, submit a loan offer to the Blockchain. The loan offer may include information sufficient to identify the loan application, i.e., a loan identifier, which may enable the Blockchain (or a smart contract executing on the Blockchain) to identify the loan application and notify the Buyer, the Seller, or the Supplier of the loan offer at step 322. Alternatively and/or additionally, the Seller may monitor recordings of loan offers on the Blockchain and identify the loan offers submitted in response to the loan application submitted by the Seller. Once the Seller becomes aware of the loan offer, the Seller may communicate with the Lender to finalize the terms of the loan. In some embodiments, the users involved, including, e.g., the Buyer, the Seller, the Supplier, and the Lender, may communicate privately through one or more off-chain communication channels to finalize the terms of the loan.

[0042] If the users reach an agreement, the users may indicate their agreement to the Blockchain. For example, in some embodiments, the Supplier may indicate at step 324 that it agrees to receive a payment from the Lender instead of the Seller. The Seller may indicate at step 326 that it agrees to the terms of the loan and the applicable fees. The Buyer may indicate at step 328 that it agrees to pay the Lender. In this manner, the purchase can be financed by the Lender, who may pay the Supplier to produce the product(s) ordered by the Seller. The Seller may in turn deliver the product(s) to the Buyer, who may pay the Lender as agreed. The Lender may then deduct an appropriate amount (determined based on the terms of the loan and the applicable fees) from the payment received from the Buyer. The Lender may provide the remainder to the Seller to complete the supply chain financing process.

[0043] In some embodiments, the Blockchain may require the Buyer, the Seller, and the Supplier to indicate their agreement before the Blockchain proceeds to step 330. In other embodiments, indicating their agreement to the Blockchain may be an optional step. In some embodiments, the users may agree to record the loan agreement, or certain terms of the agreement, on the Blockchain. The users may agree to do so for various purposes, including, e.g., to establish a secure and immutable record of their financial dealings. In some embodiments, the users may agree to record the loan agreement, or certain terms of the agreement, in a protected format so that it is not viewable to other users, effectively preserving the privacy of the users involved in supply chain financing.

[0044] In some embodiments, the agreement may be defined based on the following terms:

<i>cost</i>	Denotes the amount the Supplier charges for the purchase order placed by the Seller. In some embodiments, <i>cost</i> may be a positive number, i.e., $cost > 0$.
<i>invoice</i>	Denotes the amount the Buyer agrees to pay for ordering the product(s) through the Seller. In some embodiments, <i>invoice</i> may be a positive number, i.e., $invoice > 0$.
<i>loan</i>	Denotes the loan amount, i.e., the amount the Lender agrees to lend to the Seller. In some embodiments, <i>loan</i> may include <i>cost</i> plus a fee. In some embodiments, <i>loan</i> may be a positive number, i.e., $loan > 0$.
<i>repayment</i>	Denotes the total repayment amount, i.e., <i>loan</i> plus interest. If the interest is i and the tenure of the loan is t , then $repayment = (1 + i)^t \times loan$. In some embodiments, <i>repayment</i> may be a positive number, i.e., $repayment > 0$.
<i>remainder</i>	Denotes the remaining balance the Seller receives after the Lender deducts the repayment amount from the payment received. Therefore, <i>remainder</i> may be defined as $remainder = invoice - repayment$. In some embodiments, <i>remainder</i> may be a non-negative number, i.e., $remainder \geq 0$.
<i>f1</i>	Denotes a fee the Seller may be required to pay, e.g., to the Administrator, for facilitating the supply chain financing. In some embodiments, <i>f1</i> may be proportional to the loan

	amount, i.e., $f1 = \alpha \times loan$. Alternatively, $f1$ maybe a fixed amount $\varepsilon1$. In some embodiments, $f1$ may be a non-negative number, i.e., $f1 \geq 0$.
$f2$	Denotes a fee the Lender may be required to pay, e.g., to the Administrator, for facilitating the supply chain financing. In some embodiments, $f2$ may be proportional to the loan amount, i.e., $f2 = \beta \times loan$. Alternatively, $f2$ maybe a fixed amount $\varepsilon2$. In some embodiments, $f2$ may be a non-negative number, i.e., $f2 \geq 0$.

[0045] In some embodiments, one or more of the users involved in the supply chain financing, e.g., the Seller, may submit the values of *cost*, *invoice*, *loan*, *repayment*, *remainder*, $f1$, and $f2$ to the Blockchain at step 326 for recordation. In some embodiments, the Seller may submit these values in protected forms instead of their plaintext values. For example, in some embodiments, the Seller may submit a transaction to the Blockchain that includes commitment values $comm(cost)$, $comm(invoice)$, $comm(loan)$, $comm(repayment)$, $comm(remainder)$, $comm(f1)$, and $comm(f2)$, where the commitment values are calculated based on a commitment scheme. In some embodiments, the commitment values $comm(cost)$, $comm(invoice)$, $comm(loan)$, $comm(repayment)$, $comm(remainder)$, $comm(f1)$, and $comm(f2)$ may jointly represent the loan agreement between the Seller and the Lender, and in some embodiments, the Seller may include a loan identifier in the transaction submitted to the Blockchain so that the loan agreement recorded on the Blockchain can be identified later using the loan

identifier.

[0046] In some embodiments, the commitment scheme may include a cryptographic primitive that allows the Seller to commit to the value of x while keeping the value of x itself hidden from other users. In some embodiments, the commitment scheme used may be additively homomorphic, which means that for a particular value x and a particular value y , the commitment scheme provides that $comm(x) \oplus comm(y)$ equals $comm(x + y)$, where $\oplus$ denotes a type of algebraic operation (e.g., addition or multiplication) that can be performed on $comm(x)$ and $comm(y)$. In some embodiments, the commitment scheme used may be a Pedersen commitment scheme, e.g., as disclosed in Torben Pryds Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," Advances in Cryptology--CRYPTO 91 Procs., Lecture Notes in Computer Science, vol. 576, pp. 129-140, 1991, which is herein incorporated by reference in its entirety. The Pedersen commitment of the value x may be expressed as $PC(x, r_x) = g^x h^{r_x}$, where $PC()$ denotes Pedersen commitment, x denotes the value for which the Pedersen commitment is calculated, r_x is a random number used to generate the Pedersen commitment of the value x , and g and h are generator values. The Pedersen commitment scheme is additively homomorphic, meaning that $PC(x, r_x)PC(y, r_y) = PC(x + y, r_x + r_y)$.

[0047] It is to be understood that the Pedersen commitment described above is presented merely as an example and is not meant to be limiting. It is contemplated that other commitment schemes may be utilized instead of, or in addition to, the Pedersen commitment. It is also to be understood that the Pedersen commitment of x , expressed as $PC(x, r_x)$ in the example above, is merely one example of the commitment value

$comm(x)$. However, for illustrative purposes, and without loss of generality, the examples described herein may express the Pedersen commitment of value x as $comm(x) = PC(x, r_x) = g^x h^{r_x}$. In other words, the Seller may submit the commitment values $comm(cost) = g^{cost} h^{r_{cost}}$, $comm(invoice) = g^{invoice} h^{r_{invoice}}$, $comm(loan) = g^{loan} h^{r_{loan}}$, $comm(repayment) = g^{repayment} h^{r_{repayment}}$, $comm(remainder) = g^{remainder} h^{r_{remainder}}$, $comm(f1) = g^{f1} h^{r_{f1}}$, and $comm(f2) = g^{f2} h^{r_{f2}}$ to the Blockchain for recordation.

[0048] It is also to be understood that while the example above describes the commitment values as being submitted by the Seller, such a description is presented merely as an example and is not meant to be limiting. It is contemplated that other users, including, e.g., the Lender, may also submit the commitment values instead of, or in addition to, the Seller. However, for illustrative purposes, and without loss of generality, the examples described herein may reference the Seller as the user submitting the commitment values.

[0049] In some embodiments, the Blockchain may record the commitment values received at step 330. In some embodiment, the Blockchain may utilize one or more smart contracts executing on the Blockchain to verify the validity of the commitment values before agreeing to record the commitment values. In some embodiments, the Blockchain (or a smart contract executing on the Blockchain) may verify the validity of the commitment values without requiring the users to disclose the underlying plaintext values corresponding to the commitment values.

[0050] For example, in some embodiments, the loan amount *loan* (i.e., the amount the Lender agrees to lend to the Seller) may include *cost* (i.e., the amount the Supplier charges for the purchase order placed by the Seller) plus a fee (i.e., the fee the Seller

agrees to pay to the Administrator), e.g., $f1$. In other words, $loan$ should equal to $cost + f1$. In such embodiments, the user submitting the commitment values (e.g., the Seller) may set r_{cost} and r_{f1} to random numbers when calculating $comm(cost) = g^{cost} h^{r_{cost}}$ and $comm(f1) = g^{f1} h^{r_{f1}}$, and set r_{loan} to $r_{cost} + r_{f1}$ when calculating $comm(loan) = g^{loan} h^{r_{loan}}$. Because r_{cost} and r_{f1} are randomly distributed, the sum of r_{cost} and r_{f1} , i.e., r_{loan} , is also randomly distributed and is therefore suitable for calculating $comm(loan)$. In this manner, $loan = cost + f1; r_{loan} = r_{cost} + r_{f1} \Leftrightarrow g^{loan} h^{r_{loan}} = g^{cost} h^{r_{cost}} * g^{f1} h^{r_{f1}} = g^{cost+f1} h^{r_{cost}+r_{f1}} \Leftrightarrow comm(loan) = comm(cost) * comm(f1)$, where “ $\Leftrightarrow$ ” means “indicate,” and by verifying that $comm(loan) \equiv comm(cost) * comm(f1)$, the Blockchain can verify that $loan$ equals $cost + f1$ without knowing the plaintext values of $loan$, $cost$, and $f1$.

[0051] In another example, the repayment amount $repayment$ may equal to $(1 + i)^t \times loan$, and the values of the interest i and tenure t may be known to the Blockchain. In such embodiments, the Seller may set $r_{repayment}$ to be equal to $\rho \times r_{loan}$, where $\rho = (1 + i)^t$. Because r_{loan} is randomly distributed, $r_{repayment}$ is also randomly distributed and is therefore suitable for calculating $comm(repayment)$. In this manner, $repayment = \rho \times loan, r_{repayment} = \rho \times r_{loan} \Leftrightarrow g^{repayment} h^{r_{repayment}} = g^{\rho \times loan} h^{\rho \times r_{loan}} = (g^{loan} h^{r_{loan}})^\rho \Leftrightarrow comm(repayment) = comm(loan)^\rho$, and by verifying that $comm(repayment) \equiv comm(loan)^\rho$, the Blockchain can verify that $repayment$ equals $(1 + i)^t \times loan$ without knowing the plaintext values of $repayment$ and $loan$.

[0052] In still another example, the invoice amount $invoice$ (i.e., the amount the Buyer agrees to pay for ordering the product(s) through the Seller) may equal $repayment$ plus a $remainder$ (i.e., the remaining balance the Seller receives after the Lender deducts

the repayment amount from the payment received from the Buyer). In other words, $invoice = repayment + remainder$. In such embodiments, the Seller may set $r_{invoice}$ to a random number, and set $r_{remainder}$ to be equal to $r_{invoice} - r_{repayment}$. Because $r_{invoice}$ and $r_{repayment}$ are randomly distributed, $r_{remainder}$ is also randomly distributed and is therefore suitable for calculating $comm(remainder)$. In this manner, $invoice = repayment + remainder$; $r_{invoice} = r_{repayment} + r_{remainder} \Leftrightarrow g^{invoice} h^{r_{invoice}} = g^{remainder} h^{r_{remainder}} * g^{repayment} h^{r_{repayment}} = g^{repayment+remainder} h^{r_{repayment}+r_{remainder}} \Leftrightarrow comm(invoice) = comm(repayment) * comm(remainder)$, and by verifying that $comm(invoice) \equiv comm(repayment) * comm(remainder)$, the Blockchain can verify that $invoice$ equals $repayment + remainder$ without knowing the plaintext values of $invoice$, $repayment$, and $remainder$.

[0053] Furthermore, in some embodiments, the Seller may pay a fee, e.g., $f1$, to the Administrator for facilitating the supply chain financing. In some embodiments, $f1$ may be proportional to the loan amount, i.e., $f1 = \alpha \times loan$. In such embodiments, the Seller may set r_{f1} to $\alpha \times r_{loan}$. In this manner, $f1 = \alpha \times loan, r_{f1} = \alpha \times r_{loan} \Leftrightarrow g^{f1} h^{r_{f1}} = g^{\alpha \times loan} h^{\alpha \times r_{loan}} = (g^{loan} h^{r_{loan}})^{\alpha} \Leftrightarrow comm(f1) = comm(loan)^{\alpha}$, and by verifying that $comm(f1) \equiv comm(loan)^{\alpha}$, the Blockchain can verify that $f1$ equals $\alpha \times loan$ without knowing the plaintext values of $f1$ and $loan$.

[0054] Alternatively, in some embodiments, $f1$ maybe a fixed amount $\varepsilon 1$. In such embodiments, the Blockchain may verify whether $comm(f1)$ represents the commitment value of $f1$, which should be set to $\varepsilon 1$, based on a zero-knowledge proof provided by the Seller. Zero-knowledge proofs refer to techniques that allow a prover to prove to a verifier that a statement is true without revealing any information beyond the validity of the

statement itself. In some embodiments, the Seller and the Blockchain may agree to implement zero-knowledge proof techniques such as the proof of knowledge of a discrete logarithm, also referred to as the Schnorr protocol. In some embodiments, the Seller may also construct the zero knowledge proof using sigma protocols, which can be transformed into non-interactive zero-knowledge proofs of knowledge using Fiat-Shamir transform. Furthermore, in some embodiments, the zero-knowledge proof techniques disclosed in C. Rackoff et al., “Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack,” Advances in Cryptology–CRYPTO 91, Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, pp. 433-444, 1991, which is herein incorporated by reference in its entirety, may be utilized to prove to the Blockchain that $comm(f1)$ represents the commitment value of $f1$, which is set to $\varepsilon1$.

[0055] Additionally, in some embodiments, the Lender may pay a fee, e.g., $f2$, to the Administrator for facilitating the supply chain financing. In some embodiments, $f2$ may be proportional to the loan amount, i.e., $f2 = \beta \times loan$. In such embodiments, the Seller may set r_{f2} to $\beta \times r_{loan}$. In this manner, $f2 = \beta \times loan, r_{f2} = \beta \times r_{loan} \Leftrightarrow g^{f2} h^{r_{f2}} = g^{\beta \times loan} h^{\beta \times r_{loan}} = (g^{loan} h^{r_{loan}})^{\beta} \Leftrightarrow comm(f2) = comm(loan)^{\beta}$, and by verifying that $comm(f2) \equiv comm(loan)^{\beta}$, the Blockchain can verify that $f2$ equals $\beta \times loan$ without knowing the plaintext values of $f2$ and $loan$. Alternatively, in some embodiments, $f2$ maybe a fixed amount $\varepsilon2$. In such embodiments, the Blockchain may verify whether $comm(f2)$ represents the commitment value of $f2$, which should be set to $\varepsilon2$, based on a zero-knowledge proof provided by the Seller using the various protocols described above.

[0056] In some embodiments, the Blockchain (or a smart contract executing on the

Blockchain) may also verify at step 330 whether *cost*, *invoice*, *loan*, and *repayment* are positive and whether *remainder*, *f1*, and *f2* are non-negative. In some embodiments, the Blockchain may verify that *cost*, *invoice*, *loan*, and *repayment* are positive and that *remainder*, *f1*, and *f2* are non-negative without requiring the Seller to disclose the plaintext values of *cost*, *invoice*, *loan*, *repayment*, *remainder*, *f1*, and *f2*. In such embodiments, the Blockchain may require the Seller to provide range proofs to prove that the commitment values $comm(cost)$, $comm(invoice)$, $comm(loan)$, and $comm(repayment)$ are calculated based on positive numbers, and that the commitment values of $comm(remainder)$, $comm(f1)$, and $comm(f2)$ are calculated based on non-negative numbers.

[0057] In some embodiments, the Blockchain may accept the range proofs as disclosed in J. Camenisch et al.: “Efficient Protocols for Set Membership and Range Proofs,” ASIACRYPT 2008, LNCS, vol. 5350, pp. 234-252, A. Poelstra et al.: “Confidential Assets,” Financial Cryptography Workshops, 2018, or B. Bunz et al.: “Bulletproofs: Short Proofs for Confidential Transactions and More,” 2018 IEEE Symposium on Security and Privacy, IEEE 2018, all of which are herein incorporated by reference in their entirety. It is to be understood that accepting a zero-knowledge range proof may allow the Seller to prove to the Blockchain that $cost > 0$, $invoice > 0$, $loan > 0$, $repayment > 0$, $remainder \geq 0$, $f1 \geq 0$, and $f2 \geq 0$ without revealing the actual values of *cost*, *invoice*, *loan*, *repayment*, *remainder*, *f1*, and *f2* to the Blockchain. It is also to be understood that the range proofs may be generated through various schemes, including, but not limited to, Bulletproof scheme, Borromean Ring signatures scheme, Boneh-Boyen signatures scheme, and the like. In this manner, the Seller may only need to disclose the

commitment values to the Blockchain, effectively preserving user privacy.

[0058] In some embodiments, the Blockchain may refuse to record the commitment values if the Seller fails to provide satisfactory range proofs to prove $cost > 0$, $invoice > 0$, $loan > 0$, $repayment > 0$, $remainder \geq 0$, $f1 \geq 0$, and $f2 \geq 0$. The Blockchain may also refuse to record the commitment values if the Blockchain determines that one or more of the following conditions are not satisfied:

$$comm(loan) \equiv comm(cost) * comm(f1);$$

$$comm(repayment) \equiv comm(loan)^{\rho}; \text{ and}$$

$$comm(invoice) \equiv comm(repayment) * comm(remainder).$$

[0059] Furthermore, if $f1$ is set to be proportional to the loan amount, the Blockchain may refuse to record the commitment values if the Blockchain determines that $comm(f1) \neq comm(loan)^{\alpha}$, and if $f2$ is set to be proportional to the loan amount, the Blockchain may refuse to record the commitment values if the Blockchain determines that $comm(f2) \neq comm(loan)^{\beta}$. Alternatively, if $f1$ is set to be a fixed amount $\varepsilon1$, the Blockchain may refuse to record the commitment values if the Seller fails to prove to the Blockchain that $comm(f1)$ represents the commitment value of $f1$, which is set to $\varepsilon1$. Likewise, if $f2$ is set to be a fixed amount $\varepsilon2$, the Blockchain may refuse to record the commitment values if the Seller fails to prove to the Blockchain that $comm(f2)$ represents the commitment value of $f2$, which is set to $\varepsilon2$.

[0060] In some embodiments, the Blockchain may record the commitment values at step 330 if the Blockchain accepts all required proofs and determines that the equations express above all hold true. Furthermore, in some embodiments, the Blockchain may record information regarding account balances of various users, including, e.g., the

account balances of the Buyer, Seller, Supplier, Lender, and Administrator. In some embodiments, the Blockchain may allow the users to record commitment values of their account balances to preserve privacy. For example, the Buyer may record $comm(bal_{buyer})$ on the Blockchain instead of the plaintext value of the Buyer's account balance bal_{buyer} . Likewise, the Seller, Supplier, Lender, and Administrator may record $comm(bal_{seller})$, $comm(bal_{supplier})$, $comm(bal_{lender})$, and $comm(bal_{admin})$, respectively. In some embodiments, the commitment values may be calculated using the Pedersen commitment scheme described above.

[0061] In some embodiments, the Blockchain may further record the disbursement of the loan. For example, in some embodiments, the Lender may submit a transaction to the Blockchain at step 332 to record a loan disbursement. The loan disbursement may indicate that the Lender has performed as agreed, meaning that the Lender has paid the Supplier an amount equal to $cost$ (i.e., the amount the Supplier charges for the purchase order placed by the Seller) and paid the Administrator an amount equal to $f1$. In some embodiments, the Lender may utilize the Blockchain to facilitate the disbursement. Alternatively, the Lender may make the disbursement off-chain and use the Blockchain to record that it has made the disbursement.

[0062] In some embodiments, the transaction submitted by the Lender may include a loan identifier that can be used to identify the loan agreement recorded on the Blockchain. In some embodiments, at step 334, the Blockchain (or a smart contract executing on the Blockchain) may use the loan identifier to identify the corresponding loan agreement and obtain the terms of the loan. The Blockchain may then update the information regarding the account balances of various users, including, e.g., the account

balance of the Buyer, Seller, Supplier, Lender, and Administrator, to reflect the disbursement of the loan.

[0063] For example, suppose that the Buyer, Seller, Supplier, Lender, and Administrator have recorded commitment values of their account balances as $comm(bal_{buyer})$, $comm(bal_{seller})$, $comm(bal_{supplier})$, $comm(bal_{lender})$, and $comm(bal_{admin})$ on the Blockchain before the disbursement of the loan. At step 334, the Blockchain may update the commitment value $comm(bal_{lender})$ recorded by the Lender to reflect a decrease of bal_{lender} in the amount of $loan = cost + f1$ (i.e., new $bal_{lender} = bal_{lender} - loan$). In some embodiments, because the commitment scheme used to calculate the commitment values is additively homomorphic, the Blockchain may update the commitment value $comm(bal_{lender})$ by setting it to $comm(bal_{lender})/comm(loan)$. The Blockchain may also update the commitment value $comm(bal_{supplier})$ recorded by the Supplier to reflect an increase of $bal_{supplier}$ in the amount of $cost$ (i.e., new $bal_{supplier} = bal_{supplier} + cost$). In some embodiments, because the commitment scheme used to calculate the commitment values is additively homomorphic, the Blockchain may update the commitment value $comm(bal_{supplier})$ by setting it to $comm(bal_{supplier}) * comm(cost)$. Likewise, the Blockchain may update the commitment value $comm(bal_{admin})$ recorded by the Administrator to reflect an increase of bal_{admin} in the amount of $f1$. In some embodiments, the Blockchain may update the commitment value $comm(bal_{admin})$ by setting it to $comm(bal_{admin}) * comm(f1)$.

[0064] Once the Supplier receives the payment from the Lender, the Supplier may manufacture and deliver the product(s) to the Seller at step 336. And at step 338, the Seller may deliver the product(s) to the Buyer, who may then pay the Lender as agreed

and submit a transaction to the Blockchain at step 340 to record the payment. In some embodiments, the Buyer may agree to pay the Lender the amount equal to *invoice*. In some embodiments, the Buyer may utilize the Blockchain to facilitate the payment. Alternatively, the Buyer may make the payment off-chain and use the Blockchain to record that it has made the payment.

[0065] In some embodiments, the transaction submitted by the Buyer may include a loan identifier that can be used to identify the loan agreement recorded on the Blockchain. In some embodiments, at step 342, the Blockchain (or a smart contract executing on the Blockchain) may use the loan identifier to identify the corresponding loan agreement and obtain the terms of the loan. The Blockchain may then update the information regarding the account balances of various users, including, e.g., the account balance of the Buyer, Seller, Supplier, Lender, and Administrator, to reflect the payment made by the Buyer.

[0066] Continuing with the example above, at step 342, the Blockchain may update the commitment value recorded by the Lender to reflect the fact that the account balance of the Lender should increase by $repayment - f2$, i.e., the total repayment amount, including loan plus interest, subtracted by the fee the Lender may be required to pay to the Administrator. In other words, upon receiving the payment made by the Buyer, the Lender's new account balance should be $bal_{lender} - loan + repayment - f2$. In some embodiments, because the commitment scheme used to calculate the commitment values is additively homomorphic, the Blockchain may update the commitment value recorded by the Lender by setting it to $comm(bal_{lender}) * comm(repayment) / (comm(loan) * comm(f2))$.

[0067] The Blockchain may also update the commitment value recorded by the Administrator to reflect the fact that the account balance of the Administrator should increase by $f2$ to a total of $bal_{admin} + f1 + f2$. In some embodiments, the Blockchain may update the commitment value recorded by the Administrator by setting it to $comm(bal_{admin}) * comm(f1) * comm(f2)$. The Blockchain may further update the commitment value recorded by the Buyer to reflect the fact that the account balance of the Buyer should decrease by an amount equal to $invoice$. In some embodiments, the Blockchain may update the commitment value recorded by the Buyer by setting it to $comm(bal_{buyer})/comm(invoice)$. Likewise, the Blockchain may update the commitment value recorded by the Seller to reflect the fact that the account balance of the Seller should increase by an amount equal to $remainder$, which equals to $invoice - repayment$. Therefore, the Blockchain may update the commitment value recorded by the Seller by setting it to $comm(bal_{seller}) * comm(invoice)/comm(repayment)$. In this manner, the Blockchain can keep the recorded commitment values for the Buyer, Seller, Supplier, Lender, and Administrator updated to reflect their current account balance without exposing their underlying plaintext values to other users.

[0068] It is to be understood that the various values and their relationships described above are presented merely as examples and are not meant to be limiting. For example, while the loan amount $loan$ described above includes $cost$ the Supplier charges plus a fee $f1$ to be paid to the Administrator, i.e., $loan = cost + f1$, such an equation is merely an example and is not meant to be limiting. It is contemplated that the Lender may agree to lend the Seller an amount more (or less) than $cost + f1$. In other words, the users may define their loan agreements using different terms than those described

above, and however the loan agreements are defined, they may utilize the method 300 to record their loan agreements as well as their financial and payment information in similar manners as described above.

[0069] FIG. 4 illustrates a flow chart of a method 400 for providing a privacy-preserving supply chain financing, according to an embodiment. The method 400 may be performed by one or more nodes in a blockchain system, e.g., the nodes 102-110 in the blockchain system 100 (FIG. 1). The nodes 102-110 in the blockchain system 100 may perform operations on a blockchain, e.g., the blockchain 120 (FIG. 1). The blockchain 120 may be implemented as the Blockchain in the examples described above.

[0070] At step 402, a node, e.g., the node 102, may receive a loan application submitted by a first user. The first user may include, e.g., the Seller (FIG. 3), and the loan application may include information regarding an invoice and/or a purchase order that can be utilized as certifying documents to help secure the loan. In some embodiments, the Seller may encrypt the information regarding the invoice and/or the purchase order. It is to be understood that the Seller may encrypt the information in various manners.

[0071] At step 404, the node 102 may notify at least a second user the receipt of the loan application submitted by the first user. The second user may include, e.g., the Buyer or the Supplier (FIG. 3). At step 406, the node 102 may receive, from the Buyer or the Supplier, a confirmation confirming the validity of certain information contained in the loan application. For example, as described above, if the second user is the Buyer, the second user may consider the invoice information contained in the loan application to be valid if it accurately describes the product(s) ordered by the Buyer and/or the outstanding amount to be paid by the Buyer. If the second user is the Supplier, the second user may

consider the purchase order to be valid if it accurately describes the product(s) ordered by the Seller and/or the amount to be paid by the Seller.

[0072] At step 408, in response to receiving the confirmation, the node 102 may notify an administrator of the loan application. In some embodiments, the Blockchain may notify the administrator if the loan application has been confirmed by at least one second user. In some embodiments, the administrator may be an independent party responsible for reviewing the loan application and assessing the Seller's profile to determine whether to accept the recordation of the loan application. At step 410, the node 102 may receive, from the administrator, an acceptance of the loan application, and at step 412, in response to receiving the acceptance of the loan application from the administrator, the node 102 may record the loan application on the blockchain.

[0073] At step 414, the node 102 may receive a loan offer submitted by a lender in response to the loan application. At step 416, the node 102 may notify the first user (e.g., the Seller) and/or the second user (e.g., the Buyer or the Supplier) the receipt of the loan offer. Once the users become aware of the loan offer, the users may communicate with the lender to finalize the terms of the loan.

[0074] At step 418, the node 102 may receive a first transaction containing commitment values of the terms of the loan agreement from one of the first user, the second user, or the lender. In some embodiments, the terms of the loan agreement may include an amount the Supplier charges for the purchase order placed by the Seller (denoted as *cost* in the examples above), an amount the Buyer agrees to pay for ordering the product(s) through the Seller (denoted as *invoice* in the examples above), a loan amount (denoted as *loan* in the examples above), a repayment amount (denoted as

repayment in the examples above), and a remaining balance (denoted as *remainder* in the examples above). In such embodiments, the first transaction may contain commitment values $comm(cost)$, $comm(invoice)$, $comm(loan)$, $comm(repayment)$, and $comm(remainder)$. In some embodiments, the terms of the loan agreement may further include a fee payable by the first user (denoted as $f1$ in the examples above) and/or a fee payable by the lender (denoted as $f2$ in the examples above). In such embodiments, the first transaction may further contain commitment values $comm(f1)$ and $comm(f2)$.

[0075] At step 420, the node 102 may determine whether the commitment values contained in the first transaction satisfy a set of requirements. In some embodiments, the node 102 may make this determination by determining whether at least one of the commitment values contained in the first transaction is calculated based on a positive or non-negative number. For example, as described above, the node 102 may determine whether the commitment values $comm(cost)$, $comm(invoice)$, $comm(loan)$, and $comm(repayment)$ are calculated based on positive numbers, and whether the commitment values of $comm(remainder)$, $comm(f1)$, and $comm(f2)$ are calculated based on non-negative numbers.

[0076] In some embodiments, the node 102 may determine whether the commitment values contained in the first transaction satisfy a set of equations established for the commitment values. For example, as described above, the set of equations established for the commitment values may include:

$$comm(loan) \equiv comm(cost) * comm(f1);$$

$$comm(repayment) \equiv comm(loan)^p; \text{ and}$$

$$comm(invoice) \equiv comm(repayment) * comm(remainder), \text{ which should be}$$

satisfied by the commitment values contained in the first transaction.

[0077] Furthermore, if $f1$ is set to be proportional to the loan amount, the set of equations established for the commitment values may further include $comm(f1) = comm(loan)^\alpha$, and if $f2$ is set to be proportional to the loan amount, the set of equations established for the commitment values may further include $comm(f2) = comm(loan)^\beta$. Alternatively, in some embodiments, the node 102 may determine whether at least one of the commitment values represents a commitment value of a fixed number. For example, if $f1$ is set to be a fixed amount $\varepsilon1$, the node 102 may determine whether $comm(f1)$ represents the commitment value of $f1$, which should be set to $\varepsilon1$. Likewise, if $f2$ is set to be a fixed amount $\varepsilon2$, the node 102 may determine whether $comm(f2)$ represents the commitment value of $f2$, which should be set to $\varepsilon2$.

[0078] At step 422, in response to a determination that the commitment values contained in the first transaction satisfy the set of requirements, the node 102 may record the first transaction on the blockchain. Subsequently, at step 424, the node 102 may receive a second transaction, which may indicate a disbursement of a loan made by the lender in accordance with the loan agreement. In response to receiving the second transaction, the node 102 may, at step 426, record the second transaction on the blockchain and update commitment values of one or more account balances based on one or more commitment values contained in the first transaction. For example, in some embodiments, the node 102 may calculate a new commitment value to reflect an increase or a decrease of an account balance as described above. In some embodiments, the node 102 may perform the calculation directly on commitment values.

[0079] Furthermore, at step 428, the node 102 may receive a third transaction,

which may indicate a payment made to the lender in accordance with the loan agreement. In response to receiving the third transaction, the node 102 may, at step 430, record the third transaction on the blockchain and update commitment values of one or more account balances based on one or more commitment values contained in the first transaction. In some embodiments, the node 102 may calculate a new commitment value to reflect an increase or a decrease of an account balance as described above. In some embodiments, the node 102 may perform the calculation directly on commitment values.

[0080] FIG. 5 is a block diagram of an apparatus 500 for providing a privacy-preserving supply chain financing, according to an embodiment. The apparatus 500 may be an implementation of a software process, and may correspond to the method 400 (FIG. 4). Referring to FIG. 5, the apparatus 500 may include a receiving module 502, a determination module 504, a recording module 506, and a notification module 508.

[0081] The receiving module 502 may receive a loan application submitted by a first user and report the receipt of the loan application to the notification module 508. The notification module 508 may notify a second user, who may send a confirmation to the receiving module 502. The receiving module 502 may then report the confirmation to the notification module 508, which may notify an administrator. The Administrator may send an acceptance to the receiving module 502 to accept the loan application. The receiving module 502 may report the acceptance to the recording module 506, which may record the loan application on a blockchain.

[0082] The receiving module 502 may also receive a loan offer submitted by a lender in response to the loan application recorded on the blockchain. The receiving module 502 may report the receipt of the loan offer to the notification module 508, which

may notify the first user (e.g., the Seller) and/or the second user (e.g., the Buyer or the Supplier) the receipt of the loan offer. The receiving module 502 may then receive a first transaction containing commitment values of the terms of the loan agreement. The receiving module 502 may report the receipt of the first transaction to the determination module 504, which may determine whether the commitment values contained in the first transaction satisfy a set of requirements as described above. If so, the determination module 504 may report the determination to the recording module 506, which may record the first transaction on the blockchain.

[0083] The receiving module 502 may further receive a second transaction, which may indicate a disbursement of a loan made by the lender in accordance with the loan agreement. The receiving module 502 may report the receipt of the second transaction to the recording module 506, which may record the second transaction on the blockchain and update commitment values of one or more account balances based on one or more commitment values contained in the first transaction, as described above. In some embodiments, the receiving module 502 may also receive a third transaction, which may indicate a payment made to the lender in accordance with the loan agreement. The receiving module 502 may report the receipt of the third transaction to the recording module 506, which may record the third transaction on the blockchain and update commitment values of one or more account balances based on one or more commitment values contained in the first transaction, as described above.

[0084] Each of the above described modules may be implemented as software, or hardware, or a combination of software and hardware. For example, each of the above described modules may be implemented using a processor executing instructions stored

in a memory. Also, for example, each the above described modules may be implemented with one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), controllers, micro-controllers, microprocessors, or other electronic components, for performing the described methods. Further for example, each of the above described modules may be implemented by using a computer chip or an entity, or implemented by using a product having a certain function. In one embodiment, the apparatus 500 may be a computer, and the computer may be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0085] For an implementation process of functions and roles of each module in the apparatus 500, references can be made to corresponding steps in the above-described methods. Details are omitted here for simplicity.

[0086] In some embodiments, a computer program product may include a non-transitory computer-readable storage medium having computer-readable program instructions thereon for causing a processor to carry out the above-described methods.

[0087] The computer-readable storage medium may be a tangible device that can store instructions for use by an instruction execution device. The computer-readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-

exhaustive list of more specific examples of the computer-readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing.

[0088] The computer-readable program instructions for carrying out the above-described methods may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or source code or object code written in any combination of one or more programming languages, including an object oriented programming language, and conventional procedural programming languages. The computer-readable program instructions may execute entirely on a computing device as a stand-alone software package, or partly on a first computing device and partly on a second computing device remote from the first computing device. In the latter scenario, the second, remote computing device may be connected to the first computing device through any type of network, including a local area network (LAN) or a wide area network (WAN).

[0089] The computer-readable program instructions may be provided to a processor of a general-purpose or special-purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the above-described methods.

[0090] The flow charts and diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of devices, methods, and computer program products according to various embodiments of the specification. In this regard, a block in the flow charts or diagrams may represent a software program, segment, or portion of code, which comprises one or more executable instructions for implementing specific functions. It should also be noted that, in some alternative implementations, the functions noted in the blocks may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the diagrams and/or flow charts, and combinations of blocks in the diagrams and flow charts, may be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0091] It is appreciated that certain features of the specification, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the specification, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable subcombination or as suitable in any other described embodiment of the specification. Certain features described in the context of various embodiments are not essential features of those embodiments, unless noted as such.

[0092] Although the specification has been described in conjunction with specific

embodiments, many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the following claims embrace all such alternatives, modifications and variations that fall within the terms of the claims.

What Is Claimed Is:

1. A computer-implemented method for providing privacy-preserving supply chain financing, the method comprising:

receiving a first transaction, the first transaction containing commitment values of a plurality of terms of a loan agreement;

determining whether the commitment values contained in the first transaction satisfy a set of requirements;

in response to a determination that the commitment values contained in the first transaction satisfy the set of requirements, recording the first transaction on a blockchain;

receiving a second transaction, the second transaction indicating a disbursement of a loan made by a lender in accordance with the loan agreement; and

in response to receiving the second transaction, recording the second transaction on the blockchain and updating at least one commitment value of an account balance based on at least one of the commitment values contained in the first transaction.

2. The method of claim 1, further comprising:

receiving a third transaction, the third transaction indicating a payment made to the lender in accordance with the loan agreement; and

in response to receiving the third transaction, recording the third transaction on the blockchain and updating the at least one commitment value of the account balance based on at least one of the commitment values contained in the first transaction.

3. The method of any preceding claim, prior to receiving the first transaction, further comprising:

receiving a loan application submitted by a first user;

notifying a second user the receipt of the loan application submitted by the first user;

receiving, from the second user, a confirmation confirming a validity of information contained in the loan application;

in response to receiving the confirmation, notifying an administrator of the loan application;

receiving, from the administrator, an acceptance of the loan application; and

in response to receiving the acceptance of the loan application from the administrator, recording the loan application on the blockchain.

4. The method of claim 3, further comprising:

receiving a loan offer submitted by the lender in response to the loan application submitted by the first user;

notifying at least one of the first user and the second user the receipt of the loan offer; and

receiving the first transaction containing the commitment values of the plurality of terms of the loan agreement from at least one of the first user, the second user, and the lender.

5. The method of any preceding claim, wherein the method is performed by one or more nodes in a blockchain system implemented to operate the blockchain.

6. The method of any preceding claim, wherein the determining whether the commitment values contained in the first transaction satisfy the set of requirements further comprises:

determining whether at least one of the commitment values contained in the first transaction is calculated based on a positive or non-negative number.

7. The method of any preceding claim, wherein the determining whether the commitment values contained in the first transaction satisfy the set of requirements further comprises:

determining whether the commitment values contained in the first transaction satisfy a set of equations established for the commitment values.

8. The method of any preceding claim, wherein the determining whether the commitment values contained in the first transaction satisfy the set of requirements further comprises:

determining whether at least one of the commitment values represents a commitment value of a fixed number.

9. The method of any preceding claim, wherein the updating the at least one commitment value of the account balance further comprises:

calculating a new commitment value to reflect an increase or a decrease of the account balance.

10. The method of claim 9, wherein the calculation is performed directly on commitment values.

11. A device for providing privacy-preserving supply chain financing, comprising:
one or more processors; and
one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform the method of any of claims 1 to 10.

12. An apparatus for providing privacy-preserving supply chain financing, the apparatus comprising a plurality of modules for performing the method of any of claims 1 to 10.

13. A non-transitory computer-readable medium having stored therein instructions that, when executed by a processor of a device, cause the device to perform the method of any of claims 1 to 10.