METHODS AND DEVICES FOR FACILITATING SPLIT INVOICE FINANCING

TECHNICAL FIELD

[0001] The specification relates generally to computer technologies, and more particularly, to methods and devices for facilitating split invoice financing.

BACKGROUND

[0002] Invoice financing is a way for businesses to borrow money against accounts receivable, including, e.g., the amounts due from customers. Invoice financing helps businesses improve cash flow, pay employees and suppliers, and reinvest in operations and growth earlier than they could if they had to wait until their customers paid their balances in full.

[0003] In some situations, a business may choose to, or may be required to, split invoice financing. For example, if a business is only be able obtain invoice financing from a bank for a partial amount (i.e., an amount less than the total amount) stated on an invoice, then the business may attempt to obtain additional invoice financing from one or more additional banks for a part or all of the remaining amount. The banks involved in these situations may want to make sure that the financing they provide can be secured by the invoice. However, because the banks may not be willing to share their finance information with each other, it is difficult for these banks to make such determinations.

SUMMARY

[0004] In one aspect, a computer-implemented method for facilitating split invoice financing includes: recording a public key ring, the public key ring comprising

a plurality of public keys of a plurality of users; recording an invoice token, the invoice token comprising an initial invoice value commitment of an initial invoice value; receiving a first transaction, the first transaction being signed using the public key ring, the first transaction comprising a first loan commitment of a first loan value, a first updated invoice value commitment of a first updated invoice value, and a first range proof for proving that the first loan value is less than or equal to the initial invoice value; and in response to a determination that the first transaction is valid based on the first loan commitment, the first updated invoice value commitment, and the first range proof, replacing the initial invoice value commitment recorded for the invoice token with the first updated invoice value commitment.

[0005] In another aspect, a device for facilitating split invoice 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: record a public key ring, the public key ring comprising a plurality of public keys of a plurality of users; record an invoice token, the invoice token comprising an initial invoice value commitment of an initial invoice value; receive a first transaction, the first transaction being signed using the public key ring, the first transaction comprising a first loan commitment of a first loan value, a first updated invoice value commitment of a first updated invoice value, and a first range proof for proving that the first loan value is less than or equal to the initial invoice value; and in response to a determination that the first transaction is valid based on the first loan commitment, the first updated invoice value commitment, and the first range proof, replace the initial invoice value commitment recorded for the invoice token with the first updated invoice value commitment.

[0006] 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 facilitating split invoice financing. The method includes: recording a public key ring, the public key ring comprising a plurality of public keys of a plurality of users; recording an invoice token, the invoice token comprising an initial invoice value commitment of an initial invoice value; receiving a first transaction, the first transaction being signed using the public key ring, the first transaction comprising a first loan commitment of a first loan value, a first updated invoice value commitment of a first updated invoice value, and a first range proof for proving that the first loan value is less than or equal to the initial invoice value; and in response to a determination that the first transaction is valid based on the first loan commitment, the first updated invoice value commitment, and the first range proof, replacing the initial invoice value commitment recorded for the invoice token with the first updated invoice value commitment.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] 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.

[0008] FIG. 1 is a schematic diagram of a blockchain system, according to an embodiment.

[0009] FIG. 2 is a schematic diagram of a computing device for implementing a node in a blockchain system, according to an embodiment.

[0010] FIG. 3 is a flow chart of a method for facilitating split invoice financing, according an embodiment.

[0011] FIG. 4 is a flow chart of a method for facilitating split invoice financing, according an embodiment.

[0012] FIG. 5 is a block diagram of an apparatus for facilitating split invoice financing, according to an embodiment.

DETAILED DESCRIPTION

[0013] Embodiments of the specification provide methods and devices for facilitating split invoice financing. The methods and devices may utilize a blockchain system to record a public key ring containing public keys of a group of finance providers, including, e.g., banks and other types of financial institutions. The methods and devices may allow users (e.g., businesses or the like) to request for invoice financing from one or more finance providers. If a finance provider approves a user's invoice financing request, that finance provider may utilize the methods and devices to submit a transaction to the blockchain system and record information concerning the amount still available for invoice financing. In some embodiments, the information concerning the amount still available for invoice financing may be represented by a commitment value. The methods and devices may also allow the finance provider to sign the transaction using the public key ring. In this manner, the methods and devices can facilitate data sharing amongst the finance providers without requiring the finance providers to share their actual finance information with each other. Facilitating such data sharing may allow the finance providers to prevent the user from double financing, e.g., by obtaining invoice financing for more than the total amount stated on the invoice.

[0014] Embodiments disclosed in the specification have one or more technical effects. In some embodiments, the methods and devices utilize a blockchain system

to record a public key ring containing public keys of a group of finance providers. This allows the methods and devices to provide each finance provider in the group with the abilities to sign a transaction using the public key ring, effectively allowing the finance providers to conceal their individual identity, which in turn helps the finance providers protect their finance information. In some embodiments, the methods and devices may also allow the finance providers to record commitment values representing the amount still available for invoice financing. This allows the methods and devices to provide a mechanism for the finance providers to update information concerning invoice financing without sharing their actual finance information with each other. In some embodiments, when a finance provider approves a user's invoice financing request, that finance provider may utilize the methods and devices to submit a transaction to the blockchain system and record information concerning the amount still available for invoice financing. This allows the methods and devices to record information and prevent double financing. In some embodiments, the methods and devices may further implement a zero-knowledge range proof. This allows the methods and devices to verify that no finance provider has issued financing for more than the amount still available for invoice financing, further preventing double financing.

[0015] 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.

[0016] 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.

[0017] 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.

[0018] 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.

[0019] 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 crypto-currency networks), proof-of-stake (POS), and proof-of-authority (POA).

[0020] 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).

[0021] 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.

[0022] 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.

[0023] 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.

[0024] 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.

[0025] 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.

[0026] 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.

[0027] 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.

[0028] 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.

[0029] FIG. 3 illustrates a flow chart of a method 300 for facilitating split invoice 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., individuals, businesses, banks, finance providers, financial institutions, hospitals, as well as other types of companies, organizations, and the like.

[0030] For illustrative purposes, multiple users are depicted in FIG. 3, which includes a Borrower, an Administrator, and a group of banks including, e.g., Banks 1 through N. In some embodiments, each Bank i of the group may have a public-private key pair (Pk_i, Sk_i) for signing transactions to be submitted to the Blockchain. At step 302, each Bank i may provide its public key Pk_i to the Administrator, which may generate a public key ring R at step 304 and submit the public key ring R for recordation on the Blockchain at step 306. In some embodiments, each Bank i may provide its public key Pk_i directly to the Blockchain, which may generate and record the public key ring R.

[0031] In some embodiments, the public key ring R may include a list of public keys, e.g., $\{Pk_1, Pk_2, \ldots, Pk_N\}$, received from the group of banks. In some embodiments, the Blockchain may utilize one or more smart contracts executing on the Blockchain to verify whether the public keys contained in the public key ring R are valid. 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 contracts. 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.

[0032] In some embodiments, Banks 1 through N may record their public keys $Pk_1, Pk_2, \ldots, Pk_N$, respectively, on the Blockchain. In such embodiments, the

Blockchain may utilize one or more smart contracts to confirm that every public key contained in the public key ring R is recorded on the Blockchain.

[0033] After the public key ring R is recorded on the Blockchain, any member of the group, e.g., Bank $i \in 1 \ldots N$, may utilize the public key ring R in conjunction with its private key Sk_i to sign a transaction. Bank i may, for example, call a signing method $Sign()$ that takes the transaction, the private key Sk_i, and the public key ring R as input and produce a ring signature σ as output. The signing method $Sign()$ may produce a valid ring signature σ if the signer, e.g., Bank i, owns a private key, e.g., the private key Sk_i, that corresponds to one of the public keys in the public key ring R. The signing method $Sign()$ may then produce the ring signature σ in a manner that allows the validity of the ring signature σ to be verified using the public key ring R (instead of using Bank i's public key Pk_i). In this manner, Bank i can conceal its identify by signing the transaction using the ring signature σ. In other words, the transaction may appear to other users (including other banks of the group) as having been signed by a member of the group. However, the other users may not be able to determine which member of the group signed the transaction.

[0034] At step 308, a Borrower who is interested in obtaining invoice financing may generate an invoice token T for recordation on the Blockchain. The invoice token T may be generated based on information concerning an invoice issued by the Borrower. Such information may include, e.g., the Borrower's identity or identifier, description of product(s) or service(s) rendered by the Borrower to a purchaser, identity or identifier of the purchaser of the product(s) or service(s) rendered, the outstanding balance, payment terms, and the like.

[0035] In some embodiments, the invoice token T may include a token identifier ID and a commitment value $comm(v)$. The token identifier ID may

represent an identifier that can be used to identify the invoice token T. The commitment value $comm(v)$ may represent a commitment value of a total remaining value v of the invoice, and in some embodiments, the total remaining value v of the invoice may represent the outstanding balance remaining in the invoice that can still be used to secure invoice financing. In some embodiments, the initial value of v may be set to the outstanding balance as stated in the invoice, which may be the maximum amount that the Borrower may use to secure invoice financing. For illustrative purposes, the maximum amount that the Borrower may use to secure invoice financing may be denoted as max.

[0036] After the initial value of v is determined, its commitment value $comm(v)$ may be calculated based on a commitment scheme. The commitment scheme may include a cryptographic primitive that allows the Borrower to commit to the value of v while keeping the value of v itself hidden from others. In this manner, the Borrower may only need to submit the commitment value $comm(v)$ as a part of the invoice toke T to the Blockchain, which may utilize this commitment value to prevent double financing. As will be described in detail below, the Blockchain may allow the Borrower to borrow up to the initial value of v from the group of banks including, e.g., Banks 1 through N. However, if the Blockchain determines that the Borrower has borrowed up to the initial value of v, the Blockchain may reject the Borrower's additional attempts to borrow money using the same invoice.

[0037] In some embodiments, the commitment scheme used to calculate the commitment value $comm(v)$ 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 v may expressed as $PC(v, r_v) = g^v h^{r_v}$, where $PC()$ denotes Pedersen commitment, v denotes the value for which the Pedersen commitment is calculated, r_v is a random number used to generate the Pedersen commitment of the value v, and g and h are generator values. The Pedersen commitment scheme is additively homomorphic, meaning that $PC(x, r_x)PC(y, r_y) = PC(x + y, r_x + r_y)$.

[0038] It is to be understood that the Pedersen commitment is described above 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 v, expressed as $PC(v, r_v)$ in the example above, is merely one example of the commitment value $comm(v)$. Accordingly, for illustrative purposes, the following description may reference the commitment value of v more generally as $comm(v)$.

[0039] In some embodiments, the Borrower may generate the invoice token T to include an optional data field d, which may include additional information contained in the invoice. The data field d may include information concerning, e.g., the Borrower's identity or identifier, description of product(s) or service(s) rendered by the Borrower to a purchaser, identity or identifier of the purchaser, payment terms, and the like. In some embodiments, the data field d may be encrypted, and in

some embodiments, the data field d may be included to satisfy a regulatory agency's auditing requirement. In such embodiments, the data field d may be encrypted using the regulatory agency's public key.

[0040] In some embodiments, the Borrower may sign the invoice token T using the Borrower's private key. In some embodiments, the Borrower may also request a recipient of the invoice (e.g., the purchaser of the product(s) or service(s) rendered) and/or a trusted entity (e.g., a government agency) to sign the invoice token T using their respective private keys to further prove authenticity of the invoice. The Borrower may submit the invoice token T to the Blockchain for recordation.

[0041] At step 310, the Blockchain may record the invoice token T submitted by the Borrower. In some embodiments, the Blockchain may require the Borrower to provide a range proof π for proving that the commitment value $comm(v)$ included in the invoice token T is calculated based on v that is greater than 0 and less than or equal to the maximum amount the Borrower is allowed to borrow (i.e., $0 < v \leq max$). In some embodiments, the Blockchain may refuse to record the invoice token T submitted by the Borrower if the Borrower fails to provide a satisfactory range proof π. In some embodiments, the Blockchain may accept the range proof π as a zero-knowledge range proof, e.g., as disclosed in J. Camenisch et al.: "Efficient Protocols for Set Membership and Range Proofs," ASIACRYPT 2008, LNCS, vol. 5350, pp. 234-252, or in B. Bunz et al.: "Bulletproofs: Short Proofs for Confidential Transactions and More," 2018 IEEE Symposium on Security and Privacy, IEEE 2018, both 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 Borrower to prove to the Blockchain that $0 < v \leq max$ without revealing the actual value of v to the Blockchain. It is also to be understood that the zero-knowledge range proof π

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 Borrower may only need to disclose the commitment value $comm(v)$ to the Blockchain, allowing the Borrower to keep the underlying value of v private.

[0042] At step 312, the Borrower may proceed to request for invoice financing from a bank, e.g., Bank 1. In some embodiments, the Borrower may send information regarding the invoice and the invoice token T privately to Bank 1 through an off-chain communication channel. Such information may include, e.g., the token identifier ID of the invoice token T, the value of v, and the commitment scheme used by the Borrower to calculate $comm(v)$. If Pedersen commitment is used by the Borrower to calculate $comm(v)$, i.e., $comm(v) = PC(v, r_v)$, then then Borrower may provide the value of r_v to Bank 1. The Borrower may also send additional information to Bank 1 either voluntarily or as requested by Bank 1. Such additional information may include, e.g., the description of product(s) or service(s) rendered, identity or identifier of the purchaser, payment terms, and the like. In some embodiments, the additional information may include information contained in the data field d describe above.

[0043] At step 314, Bank 1 may proceed to decide whether to provide invoice financing to the Borrower. In some embodiments, Bank 1 may attempt to retrieve the invoice token T from the Blockchain using the token identifier ID provided by the Borrower. If the invoice token T cannot be retrieved, Bank 1 may refuse to process the Borrower's invoice financing request. If the invoice token T can be retrieved, Bank 1 may proceed to calculate $comm(v)$ based on the value of v received at step 312 and compare the calculated $comm(v)$ against the $comm(v)$ recorded in the

invoice token T. If the calculated $comm(v)$ does not match the $comm(v)$ recorded in the invoice token T, Bank 1 may refuse to proceed further because the Borrower may have altered the value of v. Similarly, if Bank 1 has received additional information from the Borrower that includes information contained in the data field d, Bank 1 may encrypt the additional information received at step 312 and compare the encrypted result against the encrypted data field d recorded in the invoice token T. If the encrypted result calculated by Bank 1 does not match the encrypted data field d recorded in the invoice token T, Bank 1 may refuse to proceed further because the Borrower may have altered the information contained in the data field d.

[0044] Bank 1 may also take various other factors into consideration in determining whether to provide invoice financing to the Borrower. Such factors may include, e.g., the value of v, the amount of financing sought by the Borrower, the Borrower's credit score and credit history, Bank 1's prior dealings with the Borrower, and the like. Additional factors may include, e.g., the purchaser's credit score and the Bank 1's prior dealings with the purchaser.

[0045] If Bank 1 decides to deny the Borrower's request for invoice financing, Bank 1 may communicate that decision to the Borrower privately through an off-chain communication channel, in which case the Borrower may choose to repeat step 312 and request for invoice financing from a different bank. On the other hand, if Bank 1 decides to provide invoice financing to the Borrower for a particular loan amount, e.g., a_1, Bank 1 may communicate that decision to the Borrower and also generate a transaction to update the invoice token T recorded on the Blockchain.

[0046] In some embodiments, the transaction generated by Bank 1 may include the token identifier ID of the invoice token T, a commitment value $comm(a_1)$

of the loan amount a_1, and a commitment value $comm(v_{updated})$ of an updated total remaining value $v_{updated}$ of the invoice. In some embodiments, Bank 1 may calculate $v_{updated}$ by subtracting the loan amount a_1 from the total remaining value v before the loan is issued, i.e., $v_{updated} = v - a_1$.

[0047] In some embodiments, the transaction generated by Bank 1 may further include a range proof π for proving that the loan amount a_1 is greater than 0 and less than or equal to v (i.e., $0 < a_1 \leq v$). In some embodiments, Bank 1 may generate the range proof π as a zero-knowledge range proof, which may allow Bank 1 to prove to the Blockchain that $0 < a_1 \leq v$ without revealing the actual value of a_1 to the Blockchain.

[0048] At step 316, Bank 1 may submit the transaction to the Blockchain for recordation. For illustrative purposes, this transaction may be referred to as an "UPDATE" transaction. In some embodiments, Bank 1 may sign the UPDATE transaction using a ring signature σ generated based on Bank 1's private key and the public key ring R as described above. It is noted that signing this UPDATE transaction using the ring signature σ may allow Bank 1 to conceal its identity. In other words, Bank 1 may simply identify itself as a member of the group that includes Banks 1 through N. However, other users (including other banks in the same group) may not be able to determine which member of the group signed this UPDATE transaction.

[0049] At step 318, the Blockchain may utilize a smart contract to verify the validity of the UPDATE transaction. In some embodiments, the smart contract may verify the validity of the UPDATE transaction based on the ring signature σ, the range proof π, as well as the commitment values $comm(a_1)$ and $comm(v_{updated})$

contained in the UPDATE transaction.

[0050] For example, in some embodiments, the smart contract may verify whether the ring signature σ used to sign the UPDATE transaction is valid. In some embodiments, the smart contract may verify the validity of the ring signature σ using the public key ring R. If the ring signature σ cannot be verified using the public key ring R, then the smart contract may conclude that Bank 1 does not have any knowledge of a private key that corresponds to one of the public keys in the public key ring R. Accordingly, the smart contract may conclude that Bank 1 is not a member of the group that includes Banks 1 through N. The smart contract may therefore consider the UPDATE transaction to be invalid and refuse to proceed further.

[0051] Similarly, in some embodiments, the smart contract may determine whether the range proof π contained in the UPDATE transaction is acceptable. If the smart contact determines that the range proof π contained in the UPDATE transaction is unacceptable, e.g., the range proof π fails to prove $0 < a_1 \leq v$, the smart contract may consider the UPDATE transaction to be invalid and refuse to proceed further.

[0052] The smart contract may also determine whether the commitment values $comm(a_1)$ and $comm(v_{updated})$ contained in the UPDATE transaction are acceptable. For example, the smart contract may use the token identifier ID contained in the UPDATE transaction to retrieve the invoice token T from the Blockchain. The smart contract may then retrieve the most up-to-date value of $comm(v)$ recorded for the invoice token T and subtract $comm(a_1)$ from $comm(v)$ to calculate the value of $comm(v) - comm(a_1)$ (or $comm(v)/comm(a_1)$, depending on

the homomorphic property of the commitment scheme used). If the calculated value of $comm(v) - comm(a_1)$ (or $comm(v)/comm(a_1)$) does not match the value of $comm(v_{updated})$ contained in the UPDATE transaction, the smart contract may consider the UPDATE transaction to be invalid and refuse to proceed further. Otherwise, if the calculated value of $comm(v) - comm(a_1)$ (or $comm(v)/comm(a_1)$) matches the value of $comm(v_{updated})$ contained in the UPDATE transaction, the smart contract may update the invoice token T and replace the most up-to-date value of $comm(v)$ with the received value of $comm(v_{updated})$. In this manner, $comm(v_{updated})$ may become the most up-to-date value of $comm(v)$ in subsequent steps. Also, in this manner, information regarding the total remaining value of the invoice can be verified and recorded on the Blockchain without revealing the actual total remaining value of the invoice to the public.

[0053] At step 320, the Borrower may attempt to request for invoice financing from another bank, e.g., Bank 2. In some embodiments, the Borrower may send information regarding the invoice and the invoice token T privately to Bank 2 through an off-chain communication channel. Such information may include, e.g., the token identifier ID of the invoice token T, the value of the most up-to-date v, and the commitment scheme used by the Borrower. It is noted that the value of the most up-to-date v should now reflect the updated total remaining value of the invoice. In other words, the loan amount a_1 should be subtracted from v now because it has already been used to secure the loan issued by Bank 1. The Borrower may also send additional information to Bank 2 either voluntarily or as requested by Bank 2. Such additional information may include, e.g., the description of product(s) or service(s) rendered by the Borrower to a purchaser, identity or identifier of the

purchaser, payment terms, and the like.

[0054] At step 322, Bank 2 may proceed to decide whether to provide invoice financing to the Borrower. In some embodiments, Bank 2 may attempt to retrieve the invoice token T from the Blockchain using the token identifier ID provided by the Borrower. If the invoice token T cannot be retrieved, Bank 2 may refuse to process the Borrower's invoice financing request. If the invoice token T can be retrieved, Bank 2 may proceed to calculate $comm(v)$ based on the value of v received at step 320 and compare the calculated $comm(v)$ against the $comm(v)$ recorded in the invoice token T. If the calculated $comm(v)$ does not match the $comm(v)$ recorded in the invoice token T, Bank 2 may refuse to proceed further because the Borrower may have altered the value of v. Similarly, if Bank 2 has received additional information from the Borrower that includes information contained in the data field d, Bank 2 may encrypt the additional information received at step 320 and compare the encrypted result against the encrypted data field d recorded in the invoice token T. If the encrypted result calculated by Bank 2 does not match the encrypted data field d recorded in the invoice token T, Bank 2 may refuse to proceed further because the Borrower may have altered the information contained in the data field d.

[0055] Bank 2 may also take various other factors into consideration in determining whether to provide invoice financing to the Borrower. Such factors may include, e.g., the value of v, the amount of financing sought by the Borrower, the Borrower's credit score and credit history, Bank 2's prior dealings with the Borrower, and the like. Additional factors may include, e.g., the purchaser's credit score and the Bank 2's prior dealings with the purchaser. In some embodiments, Bank 2 may deny the Borrower's request for invoice financing if the value of v is already set to zero, effectively preventing the Borrower from double financing.

[0056] If Bank 2 decides to deny the Borrower's request for invoice financing, Bank 2 may communicate that decision to the Borrower privately through an off-chain communication channel, in which case the Borrower may choose to repeat step 320 and request for invoice financing from a different bank. On the other hand, if Bank 2 decides to provide invoice financing to the Borrower for a particular loan amount, e.g., a_2, Bank 2 may communicate that decision to the Borrower and generate another UPDATE transaction to update the invoice token T recorded on the Blockchain.

[0057] In some embodiments, the UPDATE transaction generated by Bank 2 may include the token identifier ID of the invoice token T, a commitment value $comm(a_2)$ of the loan amount a_2, and a commitment value $comm(v_{updated})$ of an updated total remaining value $v_{updated}$ of the invoice. In some embodiments, Bank 2 may calculate $v_{updated}$ by subtracting the loan amount a_2 from v before the loan is issued, i.e., $v_{updated} = v - a_2$.

[0058] In some embodiments, the transaction generated by Bank 2 may further include a range proof π for proving that the loan amount a_2 is greater than 0 and less than or equal to v (i.e., $0 < a_2 \leq v$). In some embodiments, Bank 2 may generate the range proof π as a zero-knowledge range proof, which may allow Bank 2 to prove to the Blockchain that $0 < a_2 \leq v$ without revealing the actual value of a_2 to the Blockchain.

[0059] At step 324, Bank 2 may submit the UPDATE transaction to the Blockchain for recordation. In some embodiments, Bank 2 may sign the UPDATE transaction using a ring signature σ generated based on Bank 2s private key and the public key ring R as described above. It is noted that signing this UPDATE

transaction using the ring signature σ may allow Bank 2 to conceal its identity. In other words, Bank 2 may simply identify itself as a member of the group that includes Banks 1 through N. However, other users (including other banks in the same group) may not be able to determine which member of the group signed this UPDATE transaction.

[0060] At step 326, the Blockchain may utilize a smart contract to verify the validity of the UPDATE transaction. In some embodiments, the smart contract may verify the validity of the UPDATE transaction based on the ring signature σ, the range proof π, as well as the commitment values $comm(a_2)$ and $comm(v_{updated})$ contained in the UPDATE transaction.

[0061] In some embodiments, the smart contract may verify whether the ring signature σ used to sign the UPDATE transaction is valid. In some embodiments, the smart contract may verify the validity of the ring signature σ using the public key ring R, and if the ring signature σ cannot be verified using the public key ring R, the smart contract may consider the UPDATE transaction to be invalid and refuse to proceed further.

[0062] Similarly, in some embodiments, the smart contract may determine whether the range proof π contained in the UPDATE transaction is acceptable. If the smart contact determines that the range proof π contained in the UPDATE transaction is unacceptable, e.g., the range proof π fails to prove $0 < a_2 \leq v$, the smart contract may consider the UPDATE transaction to be invalid and refuse to proceed further.

[0063] The smart contract may also determine whether the commitment values $comm(a_2)$ and $comm(v_{updated})$ contained in the UPDATE transaction are

acceptable. For example, the smart contract may use the token identifier ID contained in the UPDATE transaction to retrieve the invoice token T from the Blockchain. The smart contract may then retrieve the most up-to-date value of $comm(v)$ from the invoice token T and subtract $comm(a_2)$ from $comm(v)$ to calculate the value of $comm(v) - comm(a_2)$. If the calculated value of $comm(v) - comm(a_2)$ does not match the value of $comm(v_{updated})$ contained in the UPDATE transaction, the smart contract may consider the UPDATE transaction to be invalid and refuse to proceed further. Otherwise, if the calculated value of $comm(v) - comm(a_2)$ matches the value of $comm(v_{updated})$ contained in the UPDATE transaction, the smart contract may update the invoice token T and replace the most up-to-date value of $comm(v)$ with the received value of $comm(v_{updated})$. In this manner, $comm(v_{updated})$ may become the most up-to-date value of $comm(v)$ in subsequent steps. Also, in this manner, information regarding the total remaining value of the invoice can be verified and recorded on the Blockchain without revealing the actual total remaining value of the invoice to the public.

[0064] It is to be understood that steps 320 through 326 may be repeated, and every time steps 320 through 326 are repeated, a new commitment value $comm(v_{updated})$ may be calculated to replace the recorded value of $comm(v)$ for the invoice token T. In this manner, the method 300 may allow the Borrower to borrow up to the initial value of v set by the Borrower when the Borrower initially generated the invoice token T, where $0 < v \leq max$. After the Borrower has borrowed up to the up to the initial value of v, the method 300 may reject the Borrower's additional attempts to borrow money using the invoice token T.

[0065] FIG. 4 illustrates a flow chart of a method 400 for facilitating split

invoice 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.

[0066] At step 402, a node, e.g., the node 102, may record a public key ring, e.g., the public key ring R, on the blockchain 120. The public key ring may include a plurality of public keys of a plurality of users. The users may include, e.g., Banks 1 through N as described above.

[0067] At step 404, the node 102 may record an invoice token, e.g., the invoice token T, on the blockchain 120. The invoice token may be generated by a borrower, e.g., the Borrower (FIG. 3), based on an invoice issued by the borrower. In some embodiments, the invoice token may include an initial invoice value commitment $comm(v)$ of an initial invoice value v. In some embodiments, the initial invoice value v may be set to an outstanding balance as stated in the invoice, which may be the maximum amount that the borrower may use to secure invoice financing. In some embodiments, the invoice token may also include an identifier and a data field. In some embodiments, the invoice token may be signed by the borrower using the borrower's private key. In some embodiments, the borrower may also request a recipient of the invoice and/or a trusted entity to sign the invoice token using their respective private keys to further prove the authenticity of the invoice. In some embodiments, the borrower may provide a range proof for proving that the initial invoice value commitment $comm(v)$ included in the invoice token is calculated based on an initial invoice value v that is greater than 0 and less than or equal to the

maximum amount the borrower is allowed to borrow. In some embodiments, the node 102 may record the invoice token on the blockchain 120 if the invoice token is properly signed and the range proof provided by the borrower is acceptable.

[0068] At step 406, the node 102 may receive a first transaction, e.g., the UPDATE transaction received from Bank 1 (FIG. 3). The first transaction may include a first loan commitment $comm(a_1)$ of a first loan value a_1, where the first loan value a_1 may represent an amount of invoice financing Bank 1 decides to provide to the borrower. The first transaction may also include a first updated invoice value commitment $comm(v_{updated})$ of a first updated invoice value $v_{updated}$, where the first updated invoice value $v_{updated}$ may represent an updated value of v after the first loan is issued. The first transaction may further include a first range proof for proving that the first loan value a_1 is less than or equal to the initial invoice value v before the first loan is issued.

[0069] At step 408, the node 102 may determine the validity of the first transaction, and in response to a determination that the first transaction is valid, replace the initial invoice value commitment $comm(v)$ recorded for the invoice token with the first updated invoice value commitment $comm(v_{updated})$.

[0070] In some embodiments, the node 102 may determine the validity of the first transaction based on whether the first transaction is signed by one of the plurality of users using the public key ring. If the first transaction is not signed by one of the plurality of users using the public key ring, then the node 102 may consider the first transaction to be invalid and refuse to proceed further. In some embodiments, the node 102 may report an error if it considers the first transaction to be invalid.

[0071] In some embodiments, the node 102 may also determine the validity of

the first transaction based on whether the proof contained in the first transaction is acceptable. If the first range proof contained in the first transaction is unacceptable, e.g., the first range proof fails to prove that the first loan value a_1 is less than or equal to the initial invoice value v before the loan is issued, the node 102 may consider the first transaction to be invalid and refuse to proceed further.

[0072] In some embodiments, the node 102 may also determine the validity of the first transaction based on whether a predetermined algebraic relationship between the initial invoice value commitment $comm(v)$, the first loan commitment $comm(a_1)$, and the first updated invoice value commitment $comm(v_{updated})$ holds true. In some embodiments, the commitments may be calculated based on an additively homomorphic commitment scheme. In such embodiments, the predetermined algebraic relationship may establish that the initial invoice value commitment $comm(v)$ subtracted or divide by the first loan commitment $comm(a_1)$ should equal the first updated invoice value commitment $comm(v_{updated})$. In this manner, if the node 102 determines that the predetermined algebraic relationship between the initial invoice value commitment, the first loan commitment, and the first updated invoice value commitment does not hold true, the node 102 may consider the first transaction to be invalid and refuse to proceed further. Otherwise, if the predetermined algebraic relationship holds true, the node 102 may update the invoice token and replace the initial invoice value commitment $comm(v)$ recorded for the invoice token with the first updated invoice value commitment $comm(v_{updated})$. In this manner, $comm(v_{updated})$ may become the most up-to-date value of $comm(v)$ in subsequent steps.

[0073] At step 410, the node 102 may receive a second transaction, e.g., the

UPDATE transaction received from Bank 2 (FIG. 3). The second transaction may include a second loan commitment $comm(a_2)$ of a second loan value a_2, where the second loan value a_2 may represent the amount of invoice financing Bank 2 decides to provide to the borrower. The second transaction may also include a second updated invoice value commitment of a second updated invoice value, where the second updated invoice value may represent the updated value of v after the second loan is issued. The second transaction may further include a second range proof for proving that the second loan value a_2 is less than or equal to the first updated invoice value before the second loan is issued.

[0074] At step 412, the node 102 may determine the validity of the second transaction, and in response to a determination that the second transaction is valid, replace the first updated invoice value commitment recorded for the invoice token (i.e., the most up-to-date value of $comm(v)$) with the second updated invoice value commitment. The node 102 may determine the validity of the second transaction based on whether the second transaction is signed by one of the plurality of users using the public key ring, whether the range proof contained in the second transaction is acceptable, or whether a predetermined algebraic relationship between the first updated invoice value commitment (which is now the most up-to-date value of $comm(v)$), the second loan commitment $comm(a_2)$, and the second updated invoice value commitment holds true. If the predetermined algebraic relationship holds true, the node 102 may update the invoice token and replace the value of $comm(v)$ recorded for the invoice token with the second updated invoice value commitment. In this manner, the second updated invoice value commitment may become the most up-to-date value of $comm(v)$ and the steps 410 and 412 may be repeated for another transaction, e.g., another UPDATE transaction received

from another bank.

[0075] FIG. 5 is a block diagram of an apparatus 500 for facilitating split invoice 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 reporting module 508.

[0076] The receiving module 502 may receive a public key ring submitted by one or more users, e.g., the Administrator or Banks 1 through N (FIG. 3) described above. The public key ring may include a plurality of public keys of a plurality of users. The users may include, e.g., Banks 1 through N. The receiving module 502 may provide the received public key ring to the recording module 506, which may record the received public key ring. The receiving module 502 may also receive an invoice token submitted by a borrower, e.g., the Borrower (FIG. 3), based on an invoice issued by the borrower. The receiving module 502 may provide the received invoice token to the recording module 506, which may record the received invoice token. The receiving module 502 may further receive transactions submitted by one or more of the plurality of users, including, e.g., Banks 1 through N (FIG. 3). The receiving module 502 may provide the received transaction to the determining module 504.

[0077] The determining module 504 may determine whether the transactions are valid. In some embodiments, the determining module 504 may determine whether the transactions are valid based on whether the transactions are signed by one of the plurality of users using the public key ring. In some embodiments, the determining module 504 may determine whether the transactions are valid based on whether range proofs contained in such transactions are acceptable. In some

embodiments, the determining module 504 may determine whether the transactions are valid based on whether a predetermined algebraic relationship between certain commitment values contained in the transactions holds true. In response to a determination that a transaction is invalid, the determining module 504 may request the reporting module 508 to report an error. Otherwise, the determining module 504 may provide the transaction to the recording module 506, which may record the transaction and update the recorded invoice token based on the transaction, as described above.

[0078] 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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).

[0083] 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.

[0084] 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.

[0085] 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.

[0086] 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 facilitating split invoice financing, the method comprising:

recording an invoice token, the invoice token comprising an initial invoice value commitment of an initial invoice value;

receiving a first transaction, the first transaction being signed using a public key ring comprising a plurality of public keys of a plurality of users, the first transaction comprising a first loan commitment of a first loan value, a first updated invoice value commitment of a first updated invoice value, and a first range proof for proving that the first loan value is less than or equal to the initial invoice value; and

in response to a determination that the first transaction is valid based on the first loan commitment, the first updated invoice value commitment, and the first range proof, replacing an initial invoice value commitment recorded for the invoice token with the first updated invoice value commitment.

2. The method of claim 1, wherein the method is performed by one or more nodes in a blockchain system.

3. The method of any preceding claim, further comprising:

determining that the first transaction is valid when:

the first transaction is signed by one of the plurality of users using the public key ring,

the first range proof is acceptable, and

a predetermined algebraic relationship between the initial invoice value commitment, the first loan commitment, and the first updated invoice value commitment holds true.

4. The method of claim 3, further comprising one of:

reporting an error in response to a determination that the first transaction is not signed by one of the plurality of users using the public key ring;

reporting the error in response to a determination that first range proof fails to prove that the first loan value is less than or equal to the initial invoice value; or

reporting the error in response to a determination that the predetermined algebraic relationship between the initial invoice value commitment, the first loan commitment, and the first updated invoice value commitment does not hold true.

5. The method of any of claims 3-4, wherein the predetermined algebraic relationship establishes that the initial invoice value commitment subtracted by the first loan commitment equals the first updated invoice value commitment.

6. The method of any of claims 3-4, wherein the predetermined algebraic relationship establishes that the initial invoice value commitment divided by the first loan commitment equals the first updated invoice value commitment.

7. The method of any preceding claim, wherein the first range proof is a zero-knowledge range proof for proving that the first loan value is less than or equal to the initial invoice value without disclosing the first loan value.

8. The method of any preceding claim, wherein the commitments are Pederson commitments.

9. The method of any preceding claim, further comprising:

receiving a second transaction, the second transaction being signed using the public key ring, the second transaction comprising a second loan commitment of a second loan value, a second updated invoice value commitment of a second updated invoice value, and a second range proof for proving that the second loan value is less than or equal to the first updated invoice value; and

in response to a determination that the second transaction is valid based on the second loan commitment, the second updated invoice value commitment, and the second range proof, replacing the first updated invoice value commitment recorded for the invoice token with the second updated invoice value commitment.

10. The method of claim 9, further comprising:

determining that the second transaction is valid when:

the second transaction is signed by one of the plurality of users using the public key ring,

the second range proof is acceptable, and

a predetermined algebraic relationship between first updated invoice value commitment, the second loan commitment, and the second updated invoice value commitment holds true.

11. The method of any of claims 9-10, wherein the second range proof is a zero-knowledge range proof for proving that the second loan value is less than or equal to the first updated invoice value without disclosing the second loan value.

12. The method of any preceding claim, wherein the plurality of users comprise a plurality of invoice financing providers.

13. A device for facilitating split invoice 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 12.

14. An apparatus for facilitating split invoice financing, the apparatus comprising a plurality of modules for performing the method of any of claims 1 to 12.

15. 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 12.