

METHOD AND SYSTEM FOR BLOCKCHAIN-BASED LOAN MANAGEMENT

TECHNICAL FIELD

[1] The disclosure relates generally to methods and systems for managing syndicated loans using blockchain technologies.

BACKGROUND

[2] Fundraising, donation, auction, loan, and other resource gathering activities may involve a process where a party sends out a request seeking for resource contributions and a plurality of resource owners respond with offers to make a certain amount of contributions. For example, syndicated loan is a form of loan business in which two or more lenders jointly provide loans for one or more borrowers on the same loan terms and with different duties. Usually, one bank is appointed as the agency bank to manage the loan business on behalf of the syndicate members. Syndicated loan arises when a project requires too large a loan for a single lender or when a project needs a specialized lender with expertise in a specific asset class. Syndicating the loan allows lenders to spread risk and take part in financial opportunities that may be too large for their individual capital bases.

[3] Blockchain technology offers various advantages for facilitating processes that involve a substantial number of parties. A blockchain may comprise a growing list of records contained in linked blocks and secured by cryptographic techniques. Each block of a blockchain may contain transaction information, account information, information about one or more previous blocks, and other related information. A blockchain may be implemented in a peer-to-peer network comprising a plurality of blockchain nodes adhering to a protocol for inter-node communication, transaction or block validation, and consensus formation.

SUMMARY

[4] Various embodiments of the present specification may include systems, methods, and non-transitory computer readable media for blockchain-based loan management.

[5] According to some embodiments, the method for blockchain-based loan management may comprise: receiving, by a blockchain node associated with a blockchain, a first blockchain

transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units; receiving, by the blockchain node, a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units; receiving, by the blockchain node, a third blockchain transaction specifying a subset of the second blockchain transactions; and executing, by the blockchain node, a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction; wherein the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme.

[6] In some embodiments, the method may further comprise: receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and executing, by the blockchain node, the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit repayment amount and the encrypted number of units in the second blockchain transaction.

[7] In some embodiments, the executing the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises, for each \ second blockchain transaction of the subset, executing, by the blockchain node, the blockchain contract to: determine an encrypted difference between the encrypted unit loan amount and the encrypted unit repayment amount; and update the encrypted loan amount as a product of the encrypted difference and the encrypted number of units in the second blockchain transaction.

[8] In some embodiments, the method may further comprise: executing, by the blockchain node, the blockchain contract to determine and store an encrypted debt amount corresponding to the first blockchain transaction, the encrypted debt amount comprising a sum of the encrypted loan amount corresponding to each second blockchain transaction of the subset.

[9] In some embodiments, the method may further comprise: receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and executing, by the blockchain node, the blockchain contract to update the encrypted debt amount

corresponding to the first blockchain transaction the encrypted unit repayment amount and the encrypted number of units in each second blockchain transaction of the subset.

[10] In some embodiments, the method may further comprise: sending, by a first computing system associated with the first blockchain transaction, the unit loan amount to one or more second computing systems associated with the plurality of second blockchain transactions via an off-blockchain channel.

[11] In some embodiments, the method may further comprise: receiving, by the blockchain node, a range claimed to encompass the unit loan amount and a zero-knowledge range proof that the unit loan amount falls within the claimed range.

[12] In some embodiments, the homomorphic encryption scheme is a Somewhat Homomorphic Encryption (SWHE) scheme based on a public key associated with the first blockchain transaction.

[13] In some embodiments, the first blockchain transaction further comprises a first unique identifier; and each of the plurality of second blockchain transactions further comprises the first unique identifier of the first blockchain transaction and a second unique identifier.

[14] In some embodiments, the third blockchain transaction comprises a plurality of the second unique identifiers respectively corresponding to the subset of the second blockchain transactions.

[15] In some embodiments, the plurality of second blockchain transactions are sorted in a chronological order, and the third blockchain transaction comprises a number K indicating that first K of the plurality of second blockchain transactions are to be included in the subset of the second blockchain transactions.

[16] In some embodiments, the third blockchain transaction further comprises an encrypted residual number of units for one of the subset of the second blockchain transactions, wherein the encrypted residual number of units is smaller than the encrypted number of units corresponding to the one second blockchain transaction.

[17] In some embodiments, the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises a product of the encrypted unit loan amount and the encrypted number of units associated with the second blockchain transaction.

[18] According to other embodiments, a system for blockchain-based loan management comprises 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 the preceding embodiments.

[19] In some embodiments, the system may receive a range claimed to encompass the unit loan amount and a zero-knowledge range proof that the unit loan amount falls within the claimed range.

[20] According to yet other embodiments, a non-transitory computer-readable storage medium is configured with instructions executable by one or more processors to cause the one or more processors to perform the method of any of the preceding embodiments.

[21] According to still other embodiments, an apparatus for blockchain-based loan management comprises a plurality of modules for performing the method of any of the preceding embodiments.

[22] According to some embodiments, a system for blockchain-based loan management may comprise one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system to perform operations comprising: receiving, by a blockchain node associated with a blockchain from a first computing system, a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units; receiving, by the blockchain node from one or more second computing systems, a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units; receiving, by the blockchain node from the first computing system, a third blockchain transaction specifying a subset of the second blockchain transactions; and executing, by the blockchain node, a blockchain contract associated

with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction; wherein the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme.

[23] According to other embodiments, a non-transitory computer-readable storage medium for blockchain-based loan management may be configured with instructions executable by one or more processors to cause the one or more processors to perform operations comprising: receiving, from a first computing system, a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units; receiving a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units; receiving a third blockchain transaction specifying a subset of the second blockchain transactions; and executing a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction; wherein the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme.

[24] According to yet other embodiments, an apparatus for blockchain-based loan management comprises a first receiving module for receiving a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units; a second receiving module for receiving a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units; a third receiving module for receiving a third blockchain transaction specifying a subset of the second blockchain transactions; and an executing module for executing a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction.

[25] Embodiments disclosed herein have one or more technical effects. In some embodiments, a blockchain is used to automatically manage a process for the administration and repayment of a syndicated loan. This improves the reliability and transparency of the syndicated loan process by keeping a permanent record that is available to involved parties. In some embodiments, the data submitted by the borrowers and lenders to the blockchain are homomorphically encrypted. The credibility of the information is verified by zero-knowledge proof. The homomorphically encrypted data may allow the blockchain-based loan management system to keep track of the borrower's outstanding liabilities (e.g., remaining loan amount to repay) and the lenders' outstanding loan amount based on the ciphertexts of the data, without releasing the underlying data to the public. In some embodiments, when a repayment is made by the borrower, the blockchain-based loan management system may correctly determine how to proportionally distribute the repayment to the lenders (e.g., according to the lenders' loan amounts lent to the borrower) without knowing the actual loan amount lent from each lender. This allows protection of the participants' privacy and confidential business information.

[26] These and other features of the systems, methods, and non-transitory computer readable media disclosed herein, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for purposes of illustration and description only and are not intended as a definition of the limits of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[27] **FIG. 1** illustrates a network environment associated with a blockchain in accordance with some embodiments.

[28] **FIG. 2** illustrates a framework for implementing blockchain transactions in accordance with some embodiments.

[29] **FIG. 3** illustrates a network environment associated with a system for blockchain-based loan management in accordance with some embodiments.

[30] **FIG. 4** illustrates a method for blockchain-based loan management in accordance with some embodiments.

[31] **FIG. 5A** illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments.

[32] **FIG. 5B** illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments.

[33] **FIG. 6** illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments.

[34] **FIG. 7** illustrates an example computing device in which any of the embodiments described herein may be implemented.

[35] **FIG. 8** illustrates an example computing device in which any of the embodiments described herein may be implemented.

DETAILED DESCRIPTION

[36] Embodiments disclosed herein provide methods, systems, and apparatus associated with an ecosystem for managing resource gathering activities using blockchain technologies. A resource gathering activity may involve a process where a party sends out a request seeking for resource contributions and a plurality of resource owners respond with offers to make a certain amount of contributions. For example, fundraising is a process of seeking and gathering voluntary financial contributions by engaging individuals, businesses, charitable foundations, or governmental agencies. Some embodiments implement fundraising platforms using blockchain-based technologies to increase transparency, security, and responsiveness. In such a blockchain-based fundraising platform, a fund requester may submit a blockchain transaction as a fundraising request to the blockchain network. The interested parties may monitor or be notified the fundraising request through their corresponding blockchain nodes, and may subsequently make voluntary offers of contributions. In some cases, the parties involved in the fundraising process

may prefer a certain degree of privacy protection while conducting requesting/offering activities in the blockchain network (e.g., a contributor may not want to disclose its amount of contribution to other contributors, the fundraising requester may not want to disclose its target amount). The technologies disclosed herein provide a mechanism for the parties to participate in a consensus process in the blockchain to effectively perform the fundraising activities with heightened privacy protection, such as avoiding disclosing the contribution amounts, and/or the target amount. Other similar use cases may include donations (e.g., charitable foundations or churches may request donations), crowdfunding (e.g., a type of fundraising to collect financial contributions from various individuals), auctions (e.g., some entities collectively buy an auctioned item), and other transactions.

[37] As another example, a syndicated loan management platform may be implemented on a blockchain network, and integrate various components, such as cloud applications, client applications (including borrower-facing application and/or lender-facing application), application programming interfaces, and other suitable components to enable various functionalities related to loan management. In some embodiments, the blockchain-based syndicated loan management platform disclosed herein may allow the loan participants (e.g., borrowers and lenders) to effectively conduct loan borrowing and repaying operations without disclosing an actual amount or even encrypted amount of each loan request or loan repayment. Relevant parties may provide blockchain nodes to blockchain networks and participate in consensus processes of the blockchain networks and may interact with various components of the syndicated loan management platform via one or more interfaces provided by the platform.

[38] For the convenience of description, this specification uses a blockchain-based syndicated loan management platform as an example to describe the embodiments of the claimed technologies. It may be obvious for a person in the art to migrate the disclosed embodiments to other use cases such as fundraising, crowdfunding, donations, auctions, and other transactions.

[39] In some embodiments, functionalities provided by the syndicated loan management platform may include receiving loan request transactions, loan finalizing transactions, and loan repayment transactions from borrowers, loan offer transactions from lenders, recording and updating outstanding liabilities of the borrowers and remaining loan amounts to be collected by

the lenders, and/or transferring digital assets between borrowers and lenders (e.g., if the involved assets are digital assets including crypto currencies). Each loan request transaction or loan repayment transaction may comprise an encrypted unit loan amount (e.g., an encrypted unit loan amount or an encrypted repayment amount) rather than a total amount to improve security and privacy. In this way, the nodes of the blockchain would not know the size of the loan requested by the borrower. Similarly, each loan offer transaction may comprise an encrypted number of units that a corresponding lender offers to lend to the borrower, where each unit corresponds to a unit loan amount requested by the borrower. In this way, the lender is not exposed to the risk of placing the total loan amount into the blockchain.

[40] When a loan request transaction is received from a borrower, the platform may record the encrypted unit loan amount (e.g., unit loan amount) through a smart contract or blockchain contract. The actual value of the unit loan amount may be communicated to the plurality of lenders through various means, such as using an off-blockchain channel, providing a zero-knowledge range proof for the lenders to know the range of the unit loan amount, submitting transactions comprising the unit loan amount encrypted with each lender's public encryption key, or another suitable means. Based on the actual value of the unit loan amount, each lender may determine a number of units to offer to the borrower, wherein the total loan amount to offer may be calculated by a product of the unit loan amount and the number of units. When a plurality of loan offer transactions are received from a plurality of lenders, the platform may record the encrypted number of units through the blockchain contract into the blockchain. By monitoring the received offers, the borrower may select one or more offers to fulfill the requested loan by submitting a loan finalizing transaction to the platform. After receiving the loan finalizing transaction, the platform may determine various information including the outstanding liability of the borrower, each lender's loan amount that is lent to the borrower, and so on. When a repayment transaction is received from the borrower, the platform may proportionally distribute the repayment to each of the lenders who have lent to the borrower.

[41] Herein, "loan share" refers to, with respect to each lender at any time, a fraction (expressed as a percentage), the numerator of which is the outstanding amount of the loans borrowed by the borrower from such lender at such time and the denominator of which is the outstanding amount of all loans borrowed by a borrower from all lenders at such time. "Zero-knowledge range proof"

refers to a mechanism to prove that a value v in a commitment or encryption is within a range without revealing the actual value of v . “Homomorphic encryption” refers to a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. “Fully Homomorphic Encryption” refers to a category of homomorphic encryption that supports an arbitrary number of addition and multiplication. “Somewhat Homomorphic Encryption” (SWHE) refers to another category of homomorphic encryption that supports a limited number of addition and multiplication, and is more efficient than fully homomorphic encryption for use cases involved in some of the embodiments.

[42] FIG. 1 illustrates a network environment associated with a blockchain in accordance with some embodiments. As shown, in the environment 100, a client 111 may couple to a server end 118, and the server end 118 and a Node B may couple to a blockchain network 112 (can also be called a blockchain system) through various communication networks. Similarly, the server end 118 may optionally couple to additional blockchain systems similar to the blockchain system 112 such as blockchain system 113, blockchain system 114, etc. Each blockchain system may maintain one or more blockchains.

[43] In some embodiments, the client 111 may comprise one or more servers (e.g., Node C) and one or more other computing devices (e.g., Node A1, Node A2, Node A3). Node A1, Node A2, and Node A3 may couple to Node C. In some embodiments, Node C may be implemented by an entity (e.g., website, mobile phone Application, organization, company, enterprise), which has various local accounts (e.g., local accounts accessed from Node A1, Node A2, Node A3). For example, a mobile phone application may have millions of end-users accessing the application’s server from respective user accounts. The application’s server may correspondingly store millions of user accounts. The components of the client 111 and their arrangement may have many other configurations.

[44] In some embodiments, Node B may include a lightweight node. A lightweight node may not download the complete blockchain but may instead just download the block headers to validate the authenticity of the blockchain transactions. Lightweight nodes may be served by and effectively dependent on full nodes (e.g., blockchain nodes in the blockchain system 112) to

access more functions of the blockchain. The lightweight nodes may be implemented in electronic devices such as laptops, mobile phones, and the like by installing an appropriate software.

[45] In some embodiments, there may be many more clients coupled to the server end 118 similar to client 111. The server end 118 may provide Blockchain-as-a-Service (BaaS) and be referred to as a BaaS cloud. In one embodiment, BaaS is a cloud service model in which clients or developers outsource behind-the-scenes aspects of a web or mobile application. BaaS may provide pre-written software for activities that take place on blockchains, such as user authentication, database management, and remote updating. The BaaS cloud may be implemented in a server, server cluster, or other devices. In one embodiment, the BaaS cloud provides an enterprise-level platform service based on blockchain technologies. This service may help clients to build a secure and stable blockchain environment as well as manage the deployment, operation, maintenance, and development of blockchain easily. Based on the abundant security strategies and multi-tenant isolation of cloud, the BaaS cloud can provide advanced security protection using chip encryption technologies. Based on highly reliable data storage, this service may provide end-to-end and highly available services that can scale up quickly without interruption. The BaaS cloud can provide native support for standard blockchain applications and data.

[46] In some embodiments, the blockchain system 112 may comprise a plurality of blockchain nodes (e.g., Blockchain Node 1, Blockchain Node 2, Blockchain Node 3, Blockchain Node 4, Blockchain Node i, etc.) that maintain one or more blockchains (e.g., public blockchain, private blockchain, consortium blockchain). Other blockchain systems (e.g., blockchain system 113, blockchain system 114) may comprise similar arrangements of blockchain nodes maintaining other blockchains. Each blockchain node may be found in one or more blockchain systems. The blockchain nodes of each blockchain system may maintain one or more blockchains. The blockchain nodes may include full nodes. Full nodes may download every block and blockchain transaction and check them against the blockchain's consensus rules. The blockchain nodes may form a network (e.g., peer-to-peer network) with one blockchain node communicating with another. The order and the number of the blockchain nodes as shown are merely examples for illustration. The blockchain nodes may be implemented in servers, computers, etc. For example,

each blockchain node may be implemented in a server or a cluster of servers. The cluster of servers may employ load balancing. Each blockchain node may correspond to one or more physical hardware devices or virtual devices coupled together via various types of communication methods such as TCP/IP. Depending on the classifications, the blockchain nodes may also be referred to as full nodes, Geth nodes, consensus nodes, etc.

[47] In the environment 100, each of the nodes and devices may be installed with appropriate software (e.g., application programming interface) and/or hardware (e.g., wires, wireless connections) to access other devices of the environment 100. In general, the nodes and devices may be able to communicate with one another through one or more wired or wireless networks (e.g., the Internet) through which data can be communicated. Each of the nodes and devices may include one or more processors and one or more memories coupled to the one or more processors. The memories may be non-transitory and computer-readable and configured with instructions executable by one or more processors to cause the one or more processors to perform operations described herein. The instructions may be stored in the memories or downloaded over a communications network without necessarily being stored in the memories. Although the nodes and devices are shown as separate components in this figure, it will be appreciated that these nodes and devices can be implemented as single devices or multiple devices coupled together. For example, Node B may be alternatively integrated into Blockchain Node 2.

[48] The devices such as Node A1, Node A2, Node A3, Node B, and Node C may be installed with an appropriate blockchain software to create blockchain accounts, and initiate, forward, or access blockchain transactions. The term “blockchain transaction” may refer to a unit of task executed in a blockchain system and recorded in the blockchain. For example, Node A1 may access the blockchain through communications with Node C, the server end 118, and Blockchain Node 1, and Node B may access the blockchain through communications with Blockchain Node 2. In some embodiments, Node A1 may submit a blockchain account creation request to Node C. Node C may forward the request and other similar requests to the server end 118. The server end 118 may accordingly create blockchain accounts.

[49] In some embodiments, after receiving a blockchain transaction request of an unconfirmed blockchain transaction, a recipient blockchain node may perform some preliminary verification of the blockchain transaction. For example, Blockchain Node 1 may perform the preliminary verification after receiving a blockchain transaction from Node C. Once verified, the blockchain transaction may be stored in a database of the recipient blockchain node (e.g., Blockchain Node 1), which may also forward the blockchain transaction to one or more other blockchain nodes (e.g., Blockchain Node 3, Blockchain Node 4). As each blockchain node may comprise or couple to a memory, the database may be respectively stored in the memories of the blockchain nodes. The database may store a pool of blockchain transactions submitted by the one or more client devices. After receiving the blockchain transaction, the one or more other blockchain nodes may repeat the process done by the recipient blockchain node.

[50] Each blockchain node may select some of the blockchain transactions from the pool according to its preference and form them into a proposed new block for the blockchain. The blockchain node may perform “mining” of the proposed new block by devoting computing power to solve complex mathematical problems. If the blockchain transaction involves a blockchain contract, the blockchain nodes may execute the blockchain contract locally in respective virtual machines (VMs). The blockchain contract may comprise instructions, code, or programs that are automatically executable by a blockchain system when one or more preset triggering conditions are met. To handle the blockchain contracts, each blockchain node of the blockchain network may run a corresponding VM and executes the same instructions in the blockchain contract. A VM is a software emulation of a computer system based on computer architectures and provides functionality of a physical computer. VM in the blockchain context can be understood as a system designed to operate as a runtime environment for blockchain contracts.

[51] A certain blockchain node that successfully mines the proposed new block of blockchain transactions in accordance with consensus rules may pack the new block into its local copy of the blockchain and multicast the results to other blockchain nodes. The certain blockchain node may be a blockchain node that has first successfully completed the verification, that has obtained a verification privilege, or that has been chosen based on another consensus rule, etc. Then, the other blockchain nodes may follow the same order of execution performed by the certain

blockchain node to locally execute the blockchain transactions in the new block, verify the execution results with one another (e.g., by performing hash calculations), and synchronize their copies of the blockchain with that of the certain blockchain node. By updating their local copies of the blockchain, the other blockchain nodes may similarly write such information in the blockchain transaction into respective local memories. As such, the blockchain contract can be deployed on the blockchain. If the verification fails at some point, the blockchain transaction is rejected.

[52] The deployed blockchain contract may have an address, according to which the deployed contract can be accessed. A blockchain node may invoke the deployed blockchain contract by inputting certain parameters to the blockchain contract. In one embodiment, Node C or Node B may request to invoke the deployed blockchain contract to perform various operations. For example, data stored in the deployed blockchain contract may be retrieved. For another example, data may be added to the deployed blockchain contract. For yet another example, a financial transaction specified in the deployed blockchain contract may be executed. Notwithstanding the above, other types of blockchain systems and associated consensus rules may be applied to the disclosed blockchain system.

[53] FIG. 2 illustrates a framework for implementing blockchain transactions in accordance with some embodiments. In some embodiments, the client 111 may transmit information (e.g., a request with relevant information for creating a blockchain account) to the server end 118 for the server end 118 to create a blockchain account. To this end, the server end 118 may generate cryptographic keys, compile the request with other account creation requests, and/or perform other operations. Then, the server end 118 may transmit a blockchain transaction (e.g., blockchain transaction A) including the compiled account creation requests to one or more of blockchain nodes for execution.

[54] In some embodiments, Node B may construct a signed blockchain transaction and transmit it to one or more blockchain nodes for execution. In one embodiment, Node B may construct a blockchain transaction B. The blockchain transaction B may comprise a blockchain contract B for deployment or invoke a deployed blockchain contract. For example, the blockchain transaction B may comprise a blockchain contract that creates a blockchain account or invokes a

deployed blockchain contract A. The blockchain contract B may be programmed in source code at a user-end application 221. For example, a user or machine may program the blockchain contract B. Node B may compile the source code using a corresponding compiler, which converts the source code into bytecode. The blockchain transaction B may comprise information such as nonce (e.g., transaction serial number), from (e.g., a blockchain address of Node B or another blockchain address), to (e.g., empty if deploying a blockchain contract), transaction fee, value (e.g., transaction amount), signature (e.g., signature of Node B), data (e.g., message to a contract account), etc. The Node B may send the blockchain transaction B to one or more blockchain nodes through a remote procedure call (RPC) interface 223 for execution. RPC is a protocol that a first program (e.g., user-end application) can use to request a service from a second program located in another computer on a network (e.g., blockchain node) without having to understand the network's details. When the first program causes a procedure to execute in a different address space, it is as if a normal (local) procedure call, without the programmer explicitly coding the details for the remote interaction.

[55] In some embodiments, on receiving the blockchain transaction (e.g., blockchain transaction A or B), the recipient blockchain may verify if the blockchain transaction is valid. For example, the signature and other formats may be verified. If the verification succeeds, the recipient blockchain node may broadcast the received blockchain transaction (e.g., blockchain transaction A or B) to the blockchain network including various other blockchain nodes. Some blockchain nodes may participate in the mining process of the blockchain transactions. The blockchain transaction may be picked by a certain blockchain node for consensus verification to pack into a new block. If the blockchain transaction involves a blockchain contract, the certain blockchain node may create a contract account for a blockchain contract in association with a contract account address. If the blockchain transaction involves invoking a deployed blockchain contract, the certain blockchain node may trigger its local VM to execute the received blockchain transaction, thereby invoking the deployed blockchain contract from its local copy of the blockchain and updating the account states in the blockchain. If the certain blockchain node succeeds in mining a new block, the certain blockchain node may broadcast the new block to other blockchain nodes. The other blockchain nodes may verify the new block as mined by the certain blockchain node. If consensus is reached, the blockchain transaction B is respectively packed to the local copies of the blockchain maintained by the blockchain nodes. The blockchain

nodes may similarly trigger their local VMs to execute the blockchain transaction B, thus invoking the blockchain contract A deployed on the local copies of the blockchain and making corresponding updates.

[56] Upon receiving the new block, the other blockchain nodes may perform verifications. If a consensus is reached that the new block is valid, the new block is respectively packed to the local copies of the blockchain maintained by the blockchain nodes. The blockchain nodes may similarly trigger their local VMs (e.g., local VM 1, local VM i, local VM 2) to execute the blockchain transactions in the new block, thus invoking local copies of the blockchain (e.g., local blockchain copy 1, local blockchain copy i, local blockchain copy 2) and making corresponding updates. The hardware machine of each blockchain node may have access to one or more virtual machines, which may be a part of or couple to the corresponding blockchain node. Each time, a corresponding local VM may be triggered to execute the blockchain transaction. Likewise, all other blockchain transactions in the new block will be executed. Lightweight nodes may also synchronize to the updated blockchain.

[57] FIG. 3 illustrates a network environment 300 associated with a system for blockchain-based loan management in accordance with some embodiments. As shown, the network environment 300 may include a blockchain network 330 providing one or more services to a plurality of users, such as loan borrowers 310 and lenders 320. The blockchain network 330 may include a plurality of blockchain nodes 331, with each of a plurality of the blockchain nodes 331 maintaining a copy of the blockchain (e.g., a ledger comprising various data associated with the blockchain) hosted by the blockchain network 330. The blockchain may comprise one or more blockchain contracts 332. The blockchain network 330 may be configured to operate one or more virtual machines and execute the one or more blockchain contracts 332 to implement one or more of the services. Once the one or more blockchain contracts are validated and deployed, each of a plurality of the blockchain nodes may store a copy of the contracts.

[58] In practical terms, a person or entity may be both a borrower 310 and a lender 320. Each of the users, such as loan borrowers 310 and lenders 320 may interact with the blockchain network 330 via one or more client or server systems. Herein, the loan borrowers 310 and lenders 320 also refer to computing systems respectively corresponding to the users. The computing systems

associated with the users may interact with the blockchain network 330 via one or more service applications or interfaces. The service applications or interfaces are herein individually or collectively referred to as the “service application.” In some embodiments, the service application may be installed on one or more client devices associated with one or more of the users 310 and 320. The service application may provide one or more user interfaces to interact with one or more services provided by blockchain network 330. For example, the service application may allow a borrower 310 to submit loan request blockchain transactions (e.g., a blockchain transaction for requesting a loan), loan finalizing blockchain transactions (e.g., a blockchain transaction for selecting one or more loan offers to fulfill a loan request), or loan repayment blockchain transactions (e.g., a blockchain transaction for recording a repayment of a loan), and allow lenders 320 to submit loan offer blockchain transactions (e.g., a blockchain transaction for making an offer in response to a loan request).

[59] In some embodiments, the blockchain network 330 may include one or more blockchain contracts 332 (e.g., smart contracts) implementing various functionalities to serve the borrowers 310 and the lenders 320. The one or more blockchain contracts 332 may be executed to perform various functionalities associated with loan management. The blockchain contracts 332 may be executed to process blockchain transactions for requesting a loan, for committing to provide a loan, for selecting lenders, for recording a repayment for a loan, other operations, or any combination thereof. The blockchain contracts 332 may also be executed to record and update various information such as an outstanding liability of the borrower, an outstanding loan amount of each lender, a repayment amount, other suitable information, or any combination thereof. The blockchain contracts 332 may be executed to create one or more blockchain transactions comprising return values of the operation of the blockchain contracts 332.

[60] FIG. 4 illustrates a method for blockchain-based loan management in accordance with some embodiments. As shown in FIG. 4, a borrower 310 and a plurality of lenders 320a, 320b may be involved in a syndicated loan transaction. A blockchain network 330 may provide necessary services to the borrower 310 and the plurality of lenders (e.g., lenders 320a, 320b) to implement the syndicated loan management system.

[61] Typically, a syndicated loan may involve two phases: a loan request phase 402 and a loan repayment phase 404. During the loan request phase 402, the borrower 310 may send a loan request at step 412 to the blockchain network 330 requesting a loan. The loan request may specify an amount of the loan in a form of “N multiples of unit loan amount v .” The unit loan amount v may refer to the smallest amount of a loan share. For instance, a loan request for one million dollars may be represented as a thousand multiples of a unit loan amount \$1,000. The unit loan amount may be referred to as a share or a unit in the context of syndicated loan. The borrower 310 may generate a blockchain transaction comprising information associated with the loan request and submit it to the blockchain network 330 for adding to the blockchain.

[62] Subsequently, the blockchain network 330 may store the loan request at step 422. For example, the loan request blockchain transaction may specify a blockchain contract (e.g., a smart contract) associated with the blockchain network 330. The blockchain network 330 may perform consensus validation on the blockchain transaction and add the blockchain transaction to the blockchain. The lenders 320 may monitor the blockchain consistently or periodically and may thereby obtain information associated with the loan request from the blockchain.

[63] Once a lender has learned about the loan request and the corresponding unit loan amount, it may propose a number of units as an offer in response to the loan request. For instance, lender 1 320a may send a loan offer blockchain transaction comprising s_1 shares (e.g., units) to the blockchain network 330 at step 423. The s_1 shares may indicate that lender 1 320A offers a loan amount of $s_1 * v$ to fulfill the loan request. Similarly, lender k 320b may send another loan offer blockchain transaction comprising s_k shares to the blockchain network 330 at step 425, indicating that it offers a loan amount of s_k shares of the unit loan amount. Either the borrower 310 or the blockchain network 330 may determine which offers from the lenders are selected to fulfill the loan request (not shown).

[64] During the loan repayment phase 404, the borrower 310 may send a loan repayment blockchain transaction to the blockchain network 330 at step 413. The loan repayment blockchain transaction may comprise information about a partial repayment in a form of “N multiples of unit repayment amount a ,” where the unit repayment amount “ a ” is less than or equal to the unit loan amount v in the loan request blockchain transaction. Accordingly, the

blockchain network 330 may update information stored in the blockchain based on the repayment at step 427 (e.g., decreasing the borrower's liability and each lender's outstanding loan amount). The blockchain network 330 may then extract the unit repayment amount a , and proportionally distribute the partial repayment to the lenders whose offers are selected to fulfill the loan. For example, if the offer with s_1 shares (or units) from lender 1 320A was part of the loan, the blockchain network 330 may distribute a repayment amount of $s_1 * a$ to lender 1 320A at step 428; similarly, if the offer with s_k shares from lender k 320B was part of the loan, the blockchain network 330 may distribute a repayment amount of $s_k * a$ to lender k 320B at step 429. In some embodiments, the blockchain network 330 may directly distribute the repayment to the lenders. In other embodiments, the blockchain network 330 may record repayment of the loan in response to confirmation of the occurrence of the repayment.

[65] In the method illustrated in FIG. 4, the blockchain network 330 may store the total amount of the loan request (e.g., $N*v$), the total loan amount from each lender (e.g., s_1*v), the amount of the partial repayment (e.g., $N*a$), and the outstanding loan amount from each lender (e.g., after each repayment).

[66] FIG. 5A illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments. The method 500A in FIG. 5A may be applicable during the loan request phase 402 shown in FIG. 4, and may include several steps.

[67] In some embodiments, a borrower 310 may submit a loan request blockchain transaction (e.g., the first blockchain transaction) to the blockchain network at step 512. Each of a plurality of blockchain nodes 331 associated with the blockchain network may receive the blockchain transaction. The loan request blockchain transaction may comprise an encrypted unit loan amount. The unit loan amount may be encrypted by one or more homomorphic encryption methods. The loan request may be associated with a total loan amount that equals the unit loan amount multiplied by a total number of units. For example, if the borrower 310 wants to request for a loan in an amount of N (e.g., $N=1,000$) multiples of a unit loan amount v (e.g., $v=\$1,000$), the corresponding loan request blockchain transaction may comprise the homomorphically encrypted version of v , denoted by ctx_v .

[68] In some embodiments, the homomorphic encryption scheme may refer to a fully homomorphic encryption or a Somewhat Homomorphic Encryption, which support addition or multiplication operations based on ciphertexts without the need to decrypt the ciphertext. For example, given homomorphic encryptions $C_1, \dots, C_u$ of values $x_1, \dots, x_u$, where $C_i = \text{Encrypt}(Pk, x_i), i = 1, \dots, u$, Pk may refer to a public key (e.g., a borrower's public key), the homomorphic encryption of a function of $x_1, \dots, x_u$, denoted as $F(x_1, \dots, x_u)$, may be calculated using the ciphertexts $C_i, i = 1, \dots, u$ rather than the plaintext of $x_i, i = 1, \dots, u$. For example, $\text{Encrypt}(Pk, F(x_1, \dots, x_u))$ may be determined by $F(C_1, \dots, C_u)$.

[69] In some embodiments, the borrower 310 may submit the loan request blockchain transaction with the following payload to a blockchain contract on the blockchain network:

$$T_{request} = \{\#ID_a, ctx_v = \text{Encrypt}(Pk, v)\}$$

[70] where Pk may refer to the borrower's public key, v may refer to a per-share (e.g., per-unit, or the smallest unit) loan amount, ctx_v may refer to the homomorphically encrypted version of v , $\#ID_a$ may refer to a unique identifier associated with the loan request, and Encrypt may refer to the homomorphic encryption operation.

[71] The plurality of lenders 320 may be notified of the loan request blockchain transaction in various ways. For example, each lender may monitor the blockchain and detect the loan request blockchain transaction. As another example, after receiving the loan request blockchain transaction, the blockchain contract may submit a plurality of blockchain transactions to the blockchain, and the plurality of lenders may monitor the blockchain and detect the blockchain transactions submitted by the blockchain contract.

[72] In order for the lenders to make offers in response to the loan request, the lenders may need to know the actual unit loan amount v , or a rough range of the unit loan amount v . In some embodiments, v may be communicated by the borrower 310 to the lenders through off-blockchain communication channels. In other embodiments, the borrower 310 may choose not to disclose v completely but provide a zero-knowledge range proof that proves v is within a range. Based on the zero-knowledge range proof, the lenders may have a rough idea about the range of the per-share loan amount. The borrower 310 may disclose v to the lenders after the loan is

finalized (e.g., after the borrower 310 selects the loan offers from the lenders). In some embodiments, the zero-knowledge range proof may be submitted to the blockchain contract as part of the loan request transaction. The blockchain contract may verify the zero-knowledge range proof and add the loan request to the blockchain at step 522.

[73] In some embodiments, if a lender (e.g., lender i) decides to offer a loan amount in response to the loan request, it may submit a loan offer blockchain transaction (e.g., the second blockchain transaction) to the blockchain network 330. Each of a plurality of blockchain nodes 331 associated with the blockchain network may receive a plurality of loan offer blockchain transactions. Each loan offer blockchain transaction may include an encrypted number of units. In some embodiments, the loan offer blockchain transaction may comprise following payload:

$$T_{offer_i} = \{\#ID_a, \#ID_{b_i}, ctx_i = Encrypt(Pk, s_i)\}$$

where Pk may refer to the borrower's public key, s_i may refer to a number of shares (e.g., units) that the lender i offers to the borrower 310, ctx_i may refer to the homomorphically encrypted s_i , $\#ID_a$ may refer to the unique identifier associated with the loan request, $\#ID_{b_i}$ may refer to a unique identifier associated with this offer, and $Encrypt$ may refer to the homomorphic encryption operation. For example, as shown in FIG. 5A, lender 1 320a may submit a blockchain transaction (e.g., the third blockchain transaction) comprising a loan offer 1 with ctx_1 to the blockchain network 330 at step 523. The blockchain transaction may invoke a blockchain contract on the blockchain network 330. The blockchain contract may store the loan offer 1 in the blockchain on the blockchain network 330 at step 524. For a syndicated loan, a plurality of lenders may participate to collectively provide the requested loan amount. Therefore, another lender k 320b may similarly submit a loan offer k with ctx_k in a blockchain transaction to the blockchain contract on the blockchain network 330 at step 525, and the blockchain network 330 may record such loan offer k on the blockchain at step 526.

[74] In some embodiments, a subset of the loan offers from the lenders may be selected to fulfill the loan request. Here, a subset may include some or all of the loan offers from the lenders. Each of a plurality of blockchain nodes 310 associated with the blockchain network may receive a loan finalizing blockchain transaction from the borrower 310. The loan finalizing blockchain

transaction may identify a subset of the loan offers specified by the loan offer transactions. In some embodiments, the borrower 310 may make the selection by submitting a loan finalizing blockchain transaction to the blockchain network 330 at step 514. In some embodiments, because the number of unit loan amounts requested may not be submitted to the blockchain network 330 or be submitted in an encrypted form, one or more blockchain contracts 332 associated with the blockchain network 330 may not be configured to select the loan offers and may rely on the loan finalizing blockchain transaction from the borrower 310 for the selection.

[75] Here, the borrower 310 may read the loan offers from the blockchain associated with the blockchain network 330. Since the loan amount in the loan offers are encrypted with the borrower's public key, the borrower 310 may decrypt the number of shares in each of the loan offers using its own private key to make selections. The loan offer selection process may be implemented in various ways. For example, the loan offers may be ranked chronologically. If the loan offers are required to be selected based on a first-come-first-pick manner, the loan finalizing blockchain transaction may comprise a number K that indicates the first K loan offers are selected. The corresponding payload may be represented as following:

$$T_{\text{finalizing}} = \{\#ID_a, K\}$$

where $\#ID_a$ may refer to the unique identifier associated with the loan request.

[76] In some cases, the loan finalizing blockchain transaction may further comprise an encrypted residual number of units for one of the subset of the second blockchain transactions (e.g., the loan offer blockchain transactions) that are selected, wherein the encrypted residual number of units is smaller than the encrypted number of units corresponding to the one second blockchain transaction.

[77] For example, the $T_{\text{finalizing}}$ may require one or more loan offers to be selected partially (e.g., only a portion of a loan offer is needed). For example, if the borrower 310 requests N shares of unit loan amount v, and $\sum_{i=1}^k s_i > N$ and $\sum_{i=1}^{k-1} s_i < N$ (e.g., the first k-1 loan offers are insufficient to fulfill the loan request, but the first k loan offers exceeds the requested loan amount), the borrower 310 may specify a loan offer and the corresponding number of shares to

be selected from the loan offer. For example, the corresponding payload may be represented as following:

$$T_{\text{finalizing}} = \{\#ID_a, K - 1, \text{ctx}'_k\}$$

wherein $\#ID_a$ may refer to the unique identifier associated with the loan request, $K - 1$ may indicate the first $K - 1$ loan offers are selected in their entirety, and the K_{th} offer may be partially selected. Specifically, ctx'_k refers to a homomorphically encrypted version of a number of shares (e.g., a residual number of units) selected from the corresponding K_{th} offer.

[78] In some embodiments, if the loan offers are not required to be selected based on the first-come-first-pick manner, the borrower 310 may select the loan offers using the corresponding identifiers. In that case, the payload in $T_{\text{finalizing}}$ may comprise a list of identifiers of blockchain transactions comprising the selected loan offers. In some embodiments, one or more of the selected loan offers may be partially selected.

[79] In some embodiments, after the loan offers are selected based on the loan finalizing blockchain transaction, the blockchain network 330 may execute a blockchain contract to determine and store an encrypted loan amount corresponding to each selected loan offer based on the encrypted unit loan amount and the encrypted number of units in the each selected loan offer, and an encrypted debt amount corresponding to the loan borrowing blockchain transaction. The encrypted debt amount may include a sum of the encrypted loan amount corresponding to each of the selected loan offers. For example, the blockchain contract may read the selected loan offers from the blockchain and records on the blockchain a table of the borrower's homomorphically encrypted liabilities and the lender's homomorphically encrypted outstanding loan amounts at step 527. For simplicity, it is assumed that the first K lenders (e.g., the first K loan offers) are selected, and the table may be created as Table 1. Herein, tables are used as an exemplary data structure for the convenience of description. Information associated with a loan may be stored in the blockchain in any suitable data structure.

Encrypted Liability $\#ID_a$	Encrypted Outstanding loan amount $\#ID_a$				
Borrower	Lender 1, $\#ID_{b_1}$	...	Lender i, $\#ID_{b_i}$	...	Lender k, $\#ID_{b_k}$

$Encrypt\left(Pk, v \sum_{i=1}^k s_i\right)$	$Encrypt(Pk, vs_1)$		$Encrypt(Pk, vs_i)$		$Encrypt(Pk, vs_k)$
--	---------------------	--	---------------------	--	---------------------

Table 1

[80] For lender I, $i = 1, \dots, k$, the encrypted outstanding loan amount $Encrypt(Pk, vs_i)$ may be calculated as a product of the encrypted unit loan amount (e.g., the ctx_v in the loan request blockchain transaction $T_{request}$) and the corresponding encrypted number of units (e.g., the ctx_i in the loan offer blockchain transaction T_{offer_i} from lender i). For the borrower 310, the liability (e.g., the loan amount) $Encrypt(Pk, v \sum_{i=1}^k s_i)$ may be calculated directly based on each lender's encrypted outstanding loan amount.

[81] In some embodiments, after the loan is finalized, the blockchain network 330 may submit a plurality of blockchain transactions to update the borrower's and each lenders' account balance to reflect the asset transfers corresponding to the loan transaction.

[82] FIG. 5B illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments. The method 500B in FIG. 5B may be applicable to the loan repayment phase 404 in FIG. 4, and may include multiple steps.

[83] As shown in FIG. 5B, the borrower 310 may submit a repayment blockchain transaction (e.g., the fourth blockchain transaction) to the blockchain network 330 at step 516. Each of a plurality of blockchain nodes associated with the blockchain network may receive a repayment blockchain transaction that includes an encrypted unit repayment amount. In some embodiments, the repayment blockchain transaction may comprise a homomorphically encrypted version of a repayment unit amount (e.g., a smallest share or unit of a repayment). For example, the payload of the repayment blockchain transaction may be represented as following:

$$T_{payment_1} = \{\#ID_a, \widetilde{ctx}_{a1} = Encrypt(Pk, a_1)\}$$

[84] wherein Pk may refer to the borrower's public key, $T_{payment_1}$ may refer to a first repayment blockchain transaction submitted by the borrower 310, $\#ID_a$ may refer to the unique identifier associated with the loan request, a_1 may refer to the repayment unit amount being paid

by the borrower 310, $\widetilde{ctx}_{a_1}$ may refer to the homomorphically encrypted a_1 , and *Encrypt* may refer to the homomorphic encryption operation. In some embodiments, the repayment blockchain transaction may also include a zero-knowledge range proof attesting the $v - a_1 \geq 0$.

[85] In some embodiments, after the first repayment blockchain transaction is received by a blockchain contract on the blockchain network 330, the blockchain network 330 may execute, through the blockchain contract, to update the encrypted loan amount corresponding to each of selected loan offers (e.g., loan offer blockchain transactions) based on the encrypted unit repayment amount and the encrypted number of units in the each selected loan offer, and to update the encrypted debt amount corresponding to the loan request blockchain transaction the encrypted unit repayment amount and the encrypted number of units in each of selected loan offers. In some embodiments, the updating the encrypted loan amount may include determining an encrypted sum of the encrypted number of units in the each selected loan offer blockchain transaction; determining an encrypted difference between the encrypted unit loan amount and the encrypted unit repayment amount; and updating the encrypted loan amount as a product of the encrypted difference and the encrypted sum.

[86] For example, the blockchain contract may update the table in Table 1 as Table 2, where it is still assumed that the first K lenders (e.g., the first K loan offers) are selected.

Encrypted Liability #ID _a	Encrypted Outstanding loan amount #ID _a				
Borrower	Lender 1, # ID _{b₁}	...	Lender i, # ID _{b_i}	...	Lender k, # ID _{b_k}
$Encrypt\left(Pk, (v - a_1) \sum_{i=1}^k s_i\right)$	$Encrypt(Pk, (v - a_1)s_1)$		$Encrypt(Pk, (v - a_1)s_i)$		$Encrypt(Pk, (v - a_1)s_k)$

Table 2

[87] With the homomorphic properties of the adopted encryption scheme, the blockchain contract may calculate the borrower's liability or loan amount (e.g., the encryption of $(v - a_1) \sum_{i=1}^k s_i$) directly based on the ciphertexts ctx_v , $\widetilde{ctx}_{a_1}$ and $ctx_i, i = 1, \dots, k$. For example, the

borrower's liability in Table 2 is represented as $Encrypt(Pk, (v - a_1) \sum_{i=1}^k s_i)$, where the borrower's public key Pk is known, $Encrypt(v)$ corresponds to ctx_v and refers to the encrypted unit loan amount, $Encrypt(a_1)$ corresponds to $\widetilde{ctx}_{a_1}$ and refers to the encrypted repayment amount, and $Encrypt(\sum_{i=1}^k s_i)$ corresponds to $ctx_i, i = 1, \dots, k$ and refers to the encrypted sum of the encrypted number of units (e.g., number of shares) in the each second blockchain transaction (e.g., the loan offers).

[88] The blockchain contract may also update lender 1's encrypted outstanding loan amount in the table (e.g., $Encrypt(Pk, (v - a_1)s_1)$, the encryption of $(v - a_1) * s_1$) using the ciphertexts ctx_v , $\widetilde{ctx}_{a_1}$, and ctx_1 . Similarly, the blockchain contract may also update lender k's encrypted outstanding loan amount (e.g., the encryption of $(v - a_1) * s_k$) using the ciphertexts ctx_v , $\widetilde{ctx}_1$, and ctx_k . Updating the encrypted outstanding loan amounts in the table may be represented as the blockchain network 330 distributing the repayment to the lenders, such as steps 529a and 529b in FIG. 5B.

[89] After m repayment blockchain transactions have been submitted to the blockchain contract, the table on the blockchain may become Table 3,

Encrypted Liability #ID _a	Encrypted Outstanding loan amount #ID _a				
Borrower	Lender 1, #ID _{b₁}	...	Lender i, #ID _{b_i}	...	Lender k, #ID _{b_k}
$Encrypt\left(Pk, (v - \sum_{j=1}^m a_j) \sum_{i=1}^k s_i\right)$	$Encrypt\left(Pk, (v - \sum_{j=1}^m a_j)s_1\right)$		$Encrypt\left(Pk, (v - \sum_{j=1}^m a_j)s_i\right)$		$Encrypt\left(Pk, (v - \sum_{j=1}^m a_j)s_k\right)$

Table 3

[90] where j represents the index of a repayment blockchain transaction, $Encrypt(Pk, (v - \sum_{j=1}^m a_j) \sum_{i=1}^k s_i)$ represents the borrower's outstanding liability after m repayments have been made, and $Encrypt(Pk, (v - \sum_{j=1}^m a_j)s_i)$, $i = 1, \dots, k$, represents the i_{th} lender's outstanding loan amount after m repayments have been made.

[91] It may be appreciated that the formulas in Table 3 are exemplary, and the encrypted liabilities and encrypted outstanding loan amounts may be determined in various other ways. For example, after a repayment blockchain transaction is received, the blockchain contract may update the table based on the most recently updated table, so that only the current repayment blockchain transaction needs to be taken into account.

[92] **FIG. 6** illustrates a method for blockchain-based loan management with privacy preservation in accordance with some embodiments. The method 600 may be performed by a device, apparatus, or system for personalizing offers for users. The method 600 may be performed by one or more components of the setup illustrated by FIG. 1, such as the computing system 102, and the computing device 220. Depending on the implementation, the method 600 may include additional, fewer, or alternative steps performed in various orders or in parallel.

[93] Block 610 includes receiving, by a blockchain node associated with a blockchain, a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units.

[94] Block 620 includes receiving, by the blockchain node, a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units. In some embodiments, the first blockchain transaction further comprises a first unique identifier; and each of the second blockchain transactions further comprises the first unique identifier of the first blockchain transaction and a second unique identifier.

[95] Block 630 includes receiving, by the blockchain node, a third blockchain transaction specifying a subset of the second blockchain transactions. In some embodiments, the third blockchain transaction further comprises an encrypted residual number of units for one of the subset of the second blockchain transactions, wherein the encrypted residual number of units is smaller than the encrypted number of units corresponding to the one second blockchain transaction. In some embodiments, the third blockchain transaction comprises a plurality of the second unique identifiers respectively corresponding to the subset of the second blockchain transactions. In some embodiments, the second blockchain transactions are sorted in a chronological order, and the third blockchain transaction comprises a number K indicating that

first K second blockchain transactions are included in the subset of the second blockchain transactions.

[96] Block 640 includes executing, by the blockchain node, a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction. In some embodiments, the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises a product of the encrypted unit loan amount and the encrypted number of units associated with the second blockchain transaction.

[97] In some embodiments, the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme. In some embodiments, the homomorphic encryption scheme is a Somewhat Homomorphic Encryption scheme based on a public key associated with the first computing system.

[98] In some embodiments, the method 600 may further include receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and executing, by the blockchain node, the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit repayment amount and the encrypted number of units in the second blockchain transaction. In some embodiments, the executing the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises, for each of the subset of second blockchain transactions, executing, by the blockchain node, the blockchain contract to: determining an encrypted difference between the encrypted unit loan amount and the encrypted unit repayment amount; and updating the encrypted loan amount as a product of the encrypted difference and the encrypted number of units in the second blockchain transaction.

[99] In some embodiments, the method 600 may further include executing, by the blockchain node, the blockchain contract to determine and store an encrypted debt amount corresponding to the first blockchain transaction, the encrypted debt amount comprising a sum of the encrypted loan amount corresponding to each second blockchain transaction of the subset.

[100] In some embodiments, the method 600 may further include receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and executing, by the blockchain node, the blockchain contract to update the encrypted debt amount corresponding to the first blockchain transaction based on the encrypted unit loan amount, the encrypted unit repayment amount, and the encrypted number of units in each second blockchain transaction of the subset.

[101] In some embodiments, the method 600 may further include sending, by a first computing system associated with the first blockchain transaction, the unit loan amount to one or more second computing systems associated with the plurality of second blockchain transactions via an off-blockchain channel.

[102] In some embodiments, the method 600 may further include receiving, by the blockchain node, a range claimed to encompass the unit loan amount and a zero-knowledge range proof that the unit loan amount falls within the claimed range.

[103] **FIG. 7** illustrates a block diagram of a computer system for blockchain-based loan management in accordance with some embodiments. The computer system 700 may be an example of an implementation of one or more modules in the blockchain network 330 in FIG. 3, or one or more other components illustrated in FIGs. 1-2. The method 600 may be implemented by the computer system 700. The computer system 700 may comprise one or more processors and one or more non-transitory computer-readable storage media (e.g., one or more memories) coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system or device (e.g., the processor) to perform the above-described method, e.g., the method 600. The computer system 700 may comprise various units/modules corresponding to the instructions (e.g., software instructions).

[104] In some embodiments, the computer system 700 may be referred to as an apparatus for managing loan transactions in a blockchain environment. The apparatus may comprise a first receiving module 710 for receiving, by a blockchain node associated with a blockchain, a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units; a second receiving module 720 for receiving, by the

blockchain node, a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units; a third receiving module 730 for receiving, by the blockchain node, a third blockchain transaction specifying a subset of the second blockchain transactions; and an executing module 740 for executing, by the blockchain node, a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction.

[105] The techniques described herein may be implemented by one or more special-purpose computing devices. The special-purpose computing devices may be desktop computer systems, server computer systems, portable computer systems, handheld devices, networking devices or any other device or combination of devices that incorporate hard-wired and/or program logic to implement the techniques. The special-purpose computing devices may be implemented as personal computers, laptops, cellular phones, camera phones, smart phones, personal digital assistants, media players, navigation devices, email devices, game consoles, tablet computers, wearable devices, or a combination thereof. Computing device(s) may be generally controlled and coordinated by operating system software. Conventional operating systems control and schedule computer processes for execution, perform memory management, provide file system, networking, I/O services, and provide a user interface functionality, such as a graphical user interface (“GUI”), among other things. The various systems, apparatuses, storage media, modules, and units described herein may be implemented in the special-purpose computing devices, or one or more computing chips of the one or more special-purpose computing devices. In some embodiments, the instructions described herein may be implemented in a virtual machine on the special-purpose computing device. When executed, the instructions may cause the special-purpose computing device to perform various methods described herein. The virtual machine may include a software, hardware, or a combination thereof.

[106] FIG. 8 illustrates an example computing device in which any of the embodiments described herein may be implemented. The computing device 800 may be used to implement one or more components of the systems and the methods shown in FIGs. 1-7 The computing device 800 may comprise a bus 802 or other communication mechanism for communicating information

and one or more hardware processors 804 coupled with bus 802 for processing information. Hardware processor(s) 804 may be, for example, one or more general purpose microprocessors.

[107] The computing device 800 may also include a main memory 808, such as a random-access memory (RAM), cache and/or other dynamic storage devices, coupled to bus 802 for storing information and instructions to be executed by processor(s) 804. Main memory 808 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor(s) 804. Such instructions, when stored in storage media accessible to processor(s) 804, may render computing device 800 into a special-purpose machine that is customized to perform the operations specified in the instructions. Main memory 808 may include non-volatile media and/or volatile media. Non-volatile media may include, for example, optical or magnetic disks. Volatile media may include dynamic memory. Common forms of media may include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a DRAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge, or networked versions of the same.

[108] The computing device 800 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computing device may cause or program computing device 800 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computing device 800 in response to processor(s) 804 executing one or more sequences of one or more instructions contained in main memory 808. Such instructions may be read into main memory 808 from another storage medium, such as storage device 809. Execution of the sequences of instructions contained in main memory 808 may cause processor(s) 804 to perform the process steps described herein. For example, the processes/methods disclosed herein may be implemented by computer program instructions stored in main memory 808. When these instructions are executed by processor(s) 804, they may perform the steps as shown in corresponding figures and described above. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[109] The computing device 800 also includes a communication interface 810 coupled to bus 802. Communication interface 810 may provide a two-way data communication coupling to one or more network links that are connected to one or more networks. As another example, communication interface 810 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN (or WAN component to communicate with a WAN). Wireless links may also be implemented.

[110] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented engines may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented engines may be distributed across a number of geographic locations.

[111] Each of the processes, methods, and algorithms described in the preceding sections may be embodied in, and fully or partially automated by, code modules executed by one or more computer systems or computer processors comprising computer hardware. The processes and algorithms may be implemented partially or wholly in application-specific circuitry.

[112] When the functions disclosed herein are implemented in the form of software functional units and sold or used as independent products, they can be stored in a processor executable non-volatile computer readable storage medium. Particular technical solutions disclosed herein (in whole or in part) or aspects that contribute to current technologies may be embodied in the form of a software product. The software product may be stored in a storage medium, comprising a number of instructions to cause a computing device (which may be a personal computer, a server, a network device, and the like) to execute all or some steps of the methods of the embodiments of the present application. The storage medium may comprise a flash drive, a portable hard drive, ROM, RAM, a magnetic disk, an optical disc, another medium operable to store program code, or any combination thereof.

[113] Particular embodiments further provide a system comprising a processor and a non-transitory computer-readable storage medium storing instructions executable by the processor to cause the system to perform operations corresponding to steps in any method of the

embodiments disclosed above. Particular embodiments further provide a non-transitory computer-readable storage medium configured with instructions executable by one or more processors to cause the one or more processors to perform operations corresponding to steps in any method of the embodiments disclosed above.

[114] Embodiments disclosed herein may be implemented through a cloud platform, a server or a server group (hereinafter collectively the “service system”) that interacts with a client. The client may be a terminal device, or a client registered by a user at a platform, wherein the terminal device may be a mobile terminal, a personal computer (PC), and any device that may be installed with a platform application program.

[115] The various features and processes described above may be used independently of one another or may be combined in various ways. All possible combinations and sub-combinations are intended to fall within the scope of this disclosure. In addition, certain method or process blocks may be omitted in some implementations. The methods and processes described herein are also not limited to any particular sequence, and the blocks or states relating thereto can be performed in other sequences that are appropriate. For example, described blocks or states may be performed in an order other than that specifically disclosed, or multiple blocks or states may be combined in a single block or state. The example blocks or states may be performed in serial, in parallel, or in some other manner. Blocks or states may be added to or removed from the disclosed example embodiments. The exemplary systems and components described herein may be configured differently than described. For example, elements may be added to, removed from, or rearranged compared to the disclosed example embodiments.

[116] The various operations of exemplary methods described herein may be performed, at least partially, by an algorithm. The algorithm may be comprised in program codes or instructions stored in a memory (e.g., a non-transitory computer-readable storage medium described above). Such algorithm may comprise a machine learning algorithm. In some embodiments, a machine learning algorithm may not explicitly program computers to perform a function but can learn from training data to make a prediction model that performs the function.

[117] The various operations of exemplary methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or

permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented engines that operate to perform one or more operations or functions described herein.

[118] Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented engines. Moreover, the one or more processors may also operate to support performance of the relevant operations in a “cloud computing” environment or as a “software as a service” (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an Application Program Interface (API)).

[119] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented engines may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented engines may be distributed across a number of geographic locations.

[120] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[121] Although an overview of the subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments

without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the subject matter may be referred to herein, individually or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or concept if more than one is, in fact, disclosed.

[122] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[123] Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

[124] As used herein, “or” is inclusive and not exclusive, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, “A, B, or C” means “A, B, A and B, A and C, B and C, or A, B, and C,” unless expressly indicated otherwise or indicated otherwise by context. Moreover, “and” is both joint and several, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, “A and B” means “A and B, jointly or severally,” unless expressly indicated otherwise or indicated otherwise by context. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented

as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[125] The term “include” or “comprise” is used to indicate the existence of the subsequently declared features, but it does not exclude the addition of other features. Conditional language, such as, among others, “can,” “could,” “might,” or “may,” unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

CLAIMS

What is claimed is:

1. A computer-implemented method for blockchain-based loan management, comprising:

receiving, by a blockchain node associated with a blockchain, a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units;

receiving, by the blockchain node, a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units;

receiving, by the blockchain node, a third blockchain transaction specifying a subset of the second blockchain transactions; and

executing, by the blockchain node, a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction;

wherein the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme.

2. The method of claim 1, further comprising:

receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and

executing, by the blockchain node, the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit repayment amount and the encrypted number of units in the second blockchain transaction

3. The method of claim 2, wherein the executing the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises, for each second blockchain transaction of the subset, executing, by the blockchain node, the blockchain contract to:

determine an encrypted difference between the encrypted unit loan amount and the encrypted unit repayment amount; and

update the encrypted loan amount as a product of the encrypted difference and the encrypted number of units in the second blockchain transaction.

4. The method of claim 1, further comprising:

executing, by the blockchain node, the blockchain contract to determine and store an encrypted debt amount corresponding to the first blockchain transaction, the encrypted debt amount comprising a sum of the encrypted loan amount corresponding to each second blockchain transaction of the subset.

5. The method of claim 4, further comprising:

receiving, by the blockchain node, a fourth blockchain transaction comprising an encrypted unit repayment amount; and

executing, by the blockchain node, the blockchain contract to update the encrypted debt amount corresponding to the first blockchain transaction based on the encrypted unit repayment amount and the encrypted number of units in each second blockchain transaction of the subset.

6. The method of claim 1, further comprising:

sending, by a first computing system associated with the first blockchain transaction, the unit loan amount to one or more second computing systems associated with the plurality of second blockchain transactions via an off-blockchain channel.

7. The method of claim 1, further comprising:

receiving, by the blockchain node, a range claimed to encompass the unit loan amount and a zero-knowledge range proof that the unit loan amount falls within the claimed range.

8. The method of claim 1, wherein:

the homomorphic encryption scheme is a Somewhat Homomorphic Encryption (SWHE) scheme based on a public key associated with the first blockchain transaction.

9. The method of claim 1, wherein:

the first blockchain transaction further comprises a first unique identifier; and

each of the plurality of second blockchain transactions further comprises the first unique identifier of the first blockchain transaction and a second unique identifier.

10. The method of claim 9, wherein:

the third blockchain transaction comprises a plurality of the second unique identifiers respectively corresponding to the subset of the second blockchain transactions.

11. The method of claim 9, wherein:

the plurality of second blockchain transactions are sorted in a chronological order, and

the third blockchain transaction comprises a number K indicating that first K of the plurality of second blockchain transactions are to be included in the subset of the second blockchain transactions.

12. The method of claim 1, wherein:

the third blockchain transaction further comprises an encrypted residual number of units for one of the subset of the second blockchain transactions, wherein the encrypted residual number of units is smaller than the encrypted number of units corresponding to the one second blockchain transaction.

13. The method of claim 1, wherein the encrypted loan amount corresponding to each second blockchain transaction of the subset comprises a product of the encrypted unit loan amount and the encrypted number of units associated with the second blockchain transaction.

14. A non-transitory computer-readable storage medium configured with instructions executable by one or more processors to cause the one or more processors to perform the method of any one of the claims 1-13.

15. An apparatus for blockchain-based loan management comprising a plurality of modules for performing the method of any one of the claims 1-13.

16. A system for blockchain-based loan management comprising one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system to perform operations comprising:

receiving a first blockchain transaction comprising an encrypted unit loan amount associated with a loan request, wherein the loan request is associated with a total loan amount that equals the unit loan amount multiplied by a total number of units;

receiving a plurality of second blockchain transactions, each of the second blockchain transactions comprising an encrypted number of units;

receiving a third blockchain transaction specifying a subset of the second blockchain transactions; and

executing a blockchain contract associated with the blockchain to determine and store an encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit loan amount and the encrypted number of units in the second blockchain transaction;

wherein the encrypted unit loan amount, the encrypted number of units, and the encrypted loan amount are generated based on a homomorphic encryption scheme.

17. The system of claim 16, wherein the operations further comprise:

receiving a fourth blockchain transaction comprising an encrypted unit repayment amount; and

executing the blockchain contract to update the encrypted loan amount corresponding to each second blockchain transaction of the subset based on the encrypted unit repayment amount and the encrypted number of units in the second blockchain transaction.

18. The system of claim 16, wherein the operations further comprise:

executing the blockchain contract to determine and store an encrypted debt amount corresponding to the first blockchain transaction, the encrypted debt amount comprising a sum of the encrypted loan amount corresponding to each second blockchain transaction of the subset.

19. The system of claim 18, wherein the operations further comprise:

receiving a fourth blockchain transaction comprising an encrypted unit repayment amount; and

executing the blockchain contract to update the encrypted debt amount corresponding to the first blockchain transaction based on the encrypted unit repayment amount and the encrypted number of units in each second blockchain transaction of the subset.

20. The system of claim 16, further comprising:

receiving a range claimed to encompass the unit loan amount and a zero-knowledge range proof that the unit loan amount falls within the claimed range.