

**METHODS OF DISTRIBUTING SOFTWARE ACROSS A NETWORK AND
DISTRIBUTION SYSTEMS**

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of Singapore Patent Application number 10201802656T filed 29 March 2018, the entire contents of which are incorporated herein by reference for all purposes.

TECHNICAL FIELD

[0002] Various embodiments relate to methods of distributing software across a network and distribution systems.

BACKGROUND

[0003] Internet of Things (IoT) technology is becoming increasingly ubiquitous. IoT devices may soon become common household items, with most of these IoT devices connected to the Internet. These household IoT devices may be vulnerable, or susceptible to attacks, since these household IoT devices are often designed with no security features in mind. Also, these IoT devices are often unattended to by humans, so manual intervention against security flaws or attacks is unlikely. Also, the manufacturer of an IoT device may not provide technical support over the entire lifetime of the IoT device. While security concerns should be addressed in the design of the IoT device by the manufacturer, it is not always possible and practical to address vulnerabilities that are not yet discovered. Even though new vulnerabilities are discovered frequently and regularly when computers are manually operated by people, patches are often released late and not applied, leading to security risks in these computer systems. IoT devices are even more susceptible to such security risks since they are often designed to operate in the background without manual intervention.

SUMMARY

[0004] According to various embodiments, there may be provided a method of distributing software across a network. The method may include: recording a transaction associated with

the software, in a new block of a blockchain; wherein the new block is linked to a preceding block that records a latest earlier transaction associated with the software; computing a reputation score of the software based on the transaction and a previous reputation score stored in the linked preceding block; and storing the computed reputation score in the new block.

[0005] According to various embodiments, there may be provided a distribution system. The distribution system may include a processor and a non-volatile computer memory storing computer readable instructions configured to generate a blockchain. The blockchain may include: a plurality of blocks, wherein each block of the plurality of blocks stores a record of a transaction associated with software; and a plurality of smart contracts. Each smart contract may correspond to a type of transaction. At least one smart contract of the plurality of smart contracts may be configured to compute a reputation score of any software. Each block may be linked to a preceding block that stores a record of a latest earlier transaction associated with the same software as the transaction of its stored record; and each block may store the reputation score of the software.

[0006] According to various embodiments, there may be provided a distribution system. The distribution system may include: a plurality of nodes, wherein each node may be configured to selectively initiate transactions. At least a subset of the plurality of nodes may be configured to implement a plurality of smart contracts, each smart contract associated with a type of transactions. For each transaction initiated by one of the plurality of nodes, the transaction associated with software: another node of the plurality of nodes may be configured to record the transaction, in a new block of a blockchain. The new block may be linked to a preceding block that records a latest earlier transaction associated with the software. At least one smart contract of the plurality of smart contracts may be configured to compute a reputation score of the software based on the transaction and a previous reputation score stored in the linked preceding block.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] In the drawings, like reference characters generally refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention. In the following description, various embodiments are described with reference to the following drawings, in which:

[0008] FIG. 1 shows a system architecture diagram of a distribution system according to various embodiments.

[0009] FIG. 2 shows an example of a transaction data format according to various embodiments.

[0010] FIG. 3A shows an example of a gateway information field according to various embodiments.

[0011] FIG. 3B shows an example of a SSP information field according to various embodiments.

[0012] FIG. 4 shows an example of a gateway register transaction according to various embodiments.

[0013] FIG. 5 shows an example of a gateway device register transaction according to various embodiments.

[0014] FIG. 6 shows an example of a SSP register transaction according to various embodiments.

[0015] FIG. 7 shows an example of a SSP device register transaction according to various embodiments.

[0016] FIG. 8 shows an example of a release transaction according to various embodiments.

[0017] FIG. 9 shows an example of an interest transaction according to various embodiments.

[0018] FIG. 10 shows an example of a review transaction according to various embodiments.

[0019] FIG. 11 shows an example of a purchase transaction according to various embodiments.

[0020] FIG. 12 shows a conceptual diagram of a blockchain network according to various embodiments.

[0021] FIG. 13 shows an example of a distribution system according to various embodiments.

[0022] FIG. 14 shows a flow diagram for a method of distributing software across a network.

[0023] FIG. 15 is a diagram illustrating an example of a hardware implementation for an apparatus employing a distribution system.

DESCRIPTION

[0024] Embodiments described below in context of the distribution systems are analogously valid for the respective methods, and vice versa. Furthermore, it will be understood that the

embodiments described below may be combined, for example, a part of one embodiment may be combined with a part of another embodiment.

[0025] It will be understood that any property described herein for a specific distribution system or blockchain may also hold for any distribution system or blockchain described herein. It will be understood that any property described herein for a specific method may also hold for any method described herein. Furthermore, it will be understood that for any distribution system, blockchain or method described herein, not necessarily all the components or steps described must be enclosed in the device or method, but only some (but not all) components or steps may be enclosed.

[0026] In this context, the distribution systems as described in this description may include a memory which is for example used in the processing carried out in the distribution systems. A memory used in the embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0027] In order that the invention may be readily understood and put into practical effect, various embodiments will now be described by way of examples and not limitations, and with reference to the figures.

[0028] In the context of various embodiments, a buyer node may be, but not limited to being, a gateway. The properties described herein for a gateway may be applicable to any buyer node.

[0029] In the context of various embodiments, a seller node may be, but not limited to being, a Security Solution Provider (SSP), although a seller node may also be a provider of other software. The properties described herein for a SSP may be applicable to any seller node.

[0030] It should be understood that while the embodiments are described with respect to a security solution, a person skilled in the art would understand that the security solution may be replaced with other types of software.

[0031] In the context of various embodiments, “transaction” may refer to an exchange or interaction between nodes in a blockchain network, or an exchange or interaction between a node and the blockchain network. In the context of various embodiments, “transaction” may not be limited to instances of buying or selling, for example, it may include registration of a

node to the blockchain network, or exchange of information that does not have monetary value.

[0032] According to various embodiments, a method of distributing software across a network may be provided. The method may employ a decentralized distribution system based on blockchain and smart contract technology. The distribution system may include a blockchain network which may serve as a distribution platform for software. The blockchain network may include a plurality of nodes that may include buyer nodes and supplier nodes. The supplier nodes may be computers that provide software. The buyer nodes may be computers that receive the software. The supplier nodes and the buyer nodes may exchange information, for example, to perform a transaction, using a blockchain. Information about the transactions may be stored in blocks of the blockchain. In other words, the blockchain may be a ledger that stores all transactions. The method may include maintaining and updating a reputation score of each software. The buyer nodes may initiate further transactions associated with the software, based on the latest reputation score of the software. The blockchain may include a new set of transaction formats for linking of blocks pertaining to records of the same solutions, and for aggregating reputation scores. The transaction formats may be designed in a way to record necessary information and to trigger corresponding smart contracts.

[0033] The blockchain may store smart contracts that verify each transaction carried out in the network. A smart contract may be a computer protocol that digitally facilitates, verifies, or enforces a transaction according to a set of predefined rules. The buyer nodes may provide the computing power to execute the smart contracts. When a transaction is requested by either a supplier node or a buyer node, all of the buyer nodes may verify the transaction and may compete to solve a mathematical puzzle. The supplier nodes may be barred from performing the verification process to prevent them from manipulating the verification process. When the transaction is verified by the buyer nodes, the first buyer node to solve the puzzle, also referred herein as the “winner” buyer node, may add a new block to the blockchain and may record the transaction in the new block. The winner buyer node may link the new block to an immediately preceding block in the blockchain, as well as to the most recent related block that records the most recent earlier transaction associated with the same software as the transaction recorded in the new block. To prevent the proliferation of inferior software, or worse, malicious software, the blockchain may store a reputation score for each software that is uploaded to the network. The smart contracts may compute and/or update the reputation score of software based on a current transaction and the last updated reputation

score of the software which may be stored in the most recent related block that the new block is linked to. Transactions that influence the reputation score of the software may include a review, or a purchase of the software. The buyer nodes may determine whether to download the software based on the reputation score of the software. The smart contract associated with purchasing the software may check that the reputation score meets a first threshold, before proceeding with the purchase transaction. The buyer nodes may install the software or apply the software to an electronic device connected to the buyer node, once the purchase transaction is verified. The buyer nodes may install the software onto the electronic devices automatically after the purchase transaction is verified, if the reputation score of the software is above a second threshold. The second threshold may be the same as or higher than, the first threshold. If the reputation score is below a minimum threshold, any one of the buyer nodes may initiate a de-registration transaction for removing the software from the network. The minimum threshold may be lower than both the first threshold and the second threshold.

[0034] The method described above may efficiently and securely distribute, as well as deploy, verifiable software such as IoT security solutions through a decentralized system. The method may ensure the integrity, authenticity and non-repudiation of the transactions using cryptographic primitives, smart contracts and “miners”. The smart contracts may be computer protocols that automatically execute the transactions while ensuring that predefined rules are met. The miners may be computer nodes in the network that verify the transactions, similar to “miners” in a bitcoin platform. As the distribution platform does not have a centralized authority and each participant may act according to predefined rules, the distribution platform may be resistant to manipulation by any single party. The smart contracts may achieve full automation of the entire transaction process. By performing a sequence of logical condition checking, the smart contracts may guarantee the secure execution of the entire solution distribution process, in order to prevent foul play such as purposely providing negative feedback while still purchasing the actually well-working solution.

[0035] According to various embodiments, the buyer nodes may be gateway devices. Each gateway device may be connected to a respective group of user devices. A gateway device may be a machine, or a computer, that connects each user device to a network, such as a Wide Area Network (WAN), a Local Area Network (LAN), or the Internet. The gateway device may controls data traffic received and transmitted by the user devices. The gateway device may act as a controller that analyzes, processes and manipulates data traffic with the aim of protecting the user devices. The gateway device may also serve as a firewall with high

computational and storage capabilities. The gateway device may also be referred herein as “gateway server” or “gateway”. The user devices may be IoT devices.

[0036] According to various embodiments, the software may be security solutions. A security solution may be any software solution that protects a device, for example an IoT device, against a non-empty set of vulnerabilities. Examples of the software include software patch, a firmware update, an independent software solution module that can be installed in the device or a gateway connected to the device. The software may also be a set of rules specific to an IoT or one of its services.

[0037] According to various embodiments, a gateway may be “intelligent” in the sense that it may deploy security solutions to its group of user devices dynamically and as early as possible with respect to the time when a vulnerability is known. The gateway may carry out functions such as patching the user devices, inspecting the traffic of the user devices, checking the traffic for patterns related to new exploits for which patches are not available, and manipulating data packets of the traffic, etc. The gateways may serve as the last line of defense for IoTs in a network, such as a home network. The gateways may have large storage and may be capable of performing heavy computations. The gateways may be connected to the Internet. Each gateway may have its own public-private key pair, which may be used for generating digital signatures. Each device in a home IoT network may have an embedded digital signature public-private key pair, issued by the vendor of the device. This key pair may not be extractable. For example, the key pair may be stored in a tamper-resistant module. Each device in an IoT network may trust its gateway, such that the gateway may act on behalf of the devices it controls.

[0038] FIG. 1 shows a system architecture diagram 100 of a distribution system according to various embodiments. The distribution system may include a blockchain network. The blockchain network may include a plurality of nodes. Each node of the plurality of nodes may be connected to the Internet 102. Alternatively, all of the nodes may be connected to a closed network. The plurality of nodes may consist of only two types of nodes – supplier nodes and buyer nodes. The supplier nodes may be security solution providers (SSP) 106. The buyer nodes may be IoT gateways 104. Each buyer node may be connected to a plurality of user devices 108, which may be electronic devices, for example, IoT appliances. The user devices 108 may include household appliances such as televisions, digital door locks, personal computers, mobile phones, lights, coffeemakers, refrigerators, microwaves and air-conditioners. Each IoT gateway 104 may be connected to appliances belonging to one household such that each household only requires one gateway. The quantity of buyer nodes

may outnumber the quantity of supplier nodes in orders of magnitude. As in a traditional blockchain network, all of the nodes may be connected to form a peer-to-peer (P2P) network. The P2P network of nodes may collectively store a decentralized database of information. Each node may be assigned with a cryptographic key pair (public and private key) that may be used to digitally sign and verify messages.

[0039] An SSP 106 may develop and may release security solutions for any type of user device. When releasing a security solution, the SSP 106 may include a trial version of the security solution. If a gateway 104 decides to verify and apply a particular solution (for example, to fix a security flaw of one of its devices), the gateway 104 may download the trial version of the solution from the SSP 106. The gateway 104 may apply and evaluates the security solution, and may subsequently report the outcome to the network. The reported outcomes, also referred herein as feedback, by the gateways 104 may provide an indication of the effectiveness of the security solutions that are being released into the market, i.e. the blockchain network in this context. Consequently, the reported outcomes by the gateways 104 may influence, or contribute to the reputation of the security solutions. The reputation of the security solutions also contribute to the reputation for the corresponding SSP 106 that developed the security solution(s). Furthermore, the gateway 104, after testing the trial version, may have the right to decide whether to purchase the solution or not. The above described process may be realized through a series of transactions. The series of transactions may be executed by a plurality of smart contracts which may be triggered when the transactions are initiated. The process may include recording these transactions in the blockchain, and miners may verify the authenticity and integrity of these transactions. The miners may be the gateways 104.

[0040] The distribution system may be understood as an interaction between two types of entities, namely the SSP 106 and the gateways 104, through a set of transactions which invoke pre-defined smart contracts. The participants of the distribution system and the blockchain network are described in further details as follows:

[0041] There may be two types of participants in the distribution system, namely the buyer nodes and the seller nodes. The buyer nodes may be computers that represent the recipients of software, i.e. customer or buyer in the distribution system. The buyer nodes may be gateways 104 in a non-limiting embodiment. The gateway 104 may be a machine that connects user devices 108 to the Internet or to a closed network. The user devices 108 that are connected to a single gateway 104 may be closely located or may be located in the same neighborhood, for example, in the same house. The gateway 104 may filter the IoT data traffic sent to the user

devices 108, to prevent unauthorized or even malicious data from reaching the user devices 108. The gateway 104 may include a processor and may also include a storage device such as a memory. The gateways 104 may make up the majority of the computing nodes in the blockchain network.

[0042] The seller nodes may represent the provider of software, i.e. the seller in the distribution system. The seller nodes may be SSPs 106. The SSP 106 may be a machine, such as a computer, that provides the security solutions for the user devices 108. The seller may be entities that develop and/or sell security solution(s) related to a particular (set of) user devices 108, which may be the manufacturer of the user device 108, or any third party service provider.

[0043] There are two roles in the blockchain network, namely the transaction participants and the miners. A transaction participant may initiate transactions by invoking a smart contract that corresponds to the desired type of transaction. Both the buyer nodes and the seller nodes may take on the role of the transaction participant. The miners may verify each transaction and corresponding smart contract executions that take place in the blockchain network, and may subsequently write the verified transaction into the blockchain. Only the buyer nodes may take on the role of the miners, to enhance the trustiness of the distribution system. The seller nodes which may be the SSPs 106, may be banned from being miners. This prevents undesirable software from being released into the blockchain and those transactions as being fraudulently verified or recorded by the seller nodes.

[0044] The buyer nodes and the seller nodes in the distribution system may initiate several types of transactions. The types of transactions may include (1) register, (2) release, (3) interest, (4) review and (5) purchase. These transaction types will be described in further details in subsequent paragraphs. Whenever one of the nodes initiate a transaction, an associated smart contract may be triggered in order to complete a set of tasks associated with the transaction. A smart contract may be a computer program comprising a set of protocols that execute a pre-defined set of actions. The smart contract may receive inputs, executes the rules written in the contract based on the received inputs, and produces an output. The smart contract may include three components, namely an address (for example, a public key), a set of pre-defined rules, and storage. A smart contract may be triggered by any of the network participants, or another smart contract. Triggering a smart contract may include initiating a transaction to the smart contract address, and providing input within the transaction.

[0045] FIG. 2 shows an example of a transaction data format 200 according to various embodiments. The transaction data format 200 may include a plurality of fields. The

transaction (abbreviated as “Txn”) type field 202 may indicate a type or nature of the transaction, for example one of the following five possible values: register, release, interest, report, and purchase. The time stamp field 204 may be used to store the time when the transaction is created. The granularity of the time stamp field 204 may be set according to the application and network requirement. The time stamp field 204 may be a standard field for all transactions, and thus the subsequent paragraphs describing the data format of different types of transactions will omit further description of the time stamp field, unless otherwise required. The transaction data format 200 may include a gateway information field 206 and a SSP information field 208. The gateway information field 206 may be further divided into three sub-fields, as shown in FIG. 3A. The SSP information field 208 may similarly be divided into three sub-fields, as shown in FIG. 3B. The smart contract information field 210 may include an address of the smart contract (for example, a public key of the smart contract) that corresponds to the transaction type. The blockchain network may store a plurality of smart contracts, and the address of the smart contract may point to a storage location in the blockchain network. As mentioned above, each transaction may interact with the associated smart contract to trigger a set of actions to be carried out. Therefore, each transaction may need to include the smart contract address, in order to identify the corresponding smart contract from the pool of stored contracts. The amount field 212 may be used for storing monetary value associated with a particular transaction. The amount field 212 may be left empty for transactions that do not involve any monetary value transfer. The previous transaction hash (abbreviation: “PreTxnHash”) field 214 may store the hash output of the previous transaction that is related to this current transaction. The definition of “related”, or to be exact the “previous transaction”, will be defined in detail in subsequent paragraphs, together with the specific transaction description. The hash output may be generated using a standard, secure hash function such as the secure hash algorithm 256 bits (SHA256). For example, the previous transaction hash may be:

$$\text{PreTxnHash} = \text{SHA256}(\text{previous transaction})$$

The purpose of the previous transaction hash field 214 may be to chain the transactions together, to prevent possible forgery or modification of transactions once they are recorded into the blockchain. The digital signature field 216 may be used by each transaction initiator to digitally sign a transaction, in order to provide authenticity, integrity and non-repudiation for the particular transaction. The digital signature scheme may be a standard scheme such as RSA or ECDSA.

[0046] FIG. 3A shows an example of a gateway information field 206, according to various embodiments. The gateway information field 206 may include a gateway identifier (abbreviation: “ID”) sub-field 302, a device ID sub-field 304 and a data sub-field 306. The gateway ID sub-field 302 may store a unique identifier for each gateway 104. The identifier for the gateway 104 may be a unique public key associated with the gateway 104. The device ID sub-field 304 may store the unique identifier of a user device 108 that the gateway 104 controls. The identifier of the user device 108 may be the user device public key, or the user device serial number, or a combination of both. The data sub-field 306 may be used to store transaction-specific data, such as the digital signature of the user device 108 during registration, or a solution evaluation report for the review transaction. Details of the data sub-field 306 in the context of different transactions, are discussed in the subsequent paragraphs. One or more sub-fields of the gateway information field 206 may be left empty for some types of transactions.

[0047] FIG. 3B shows an example of a SSP information field 208, according to various embodiments. The SSP information field 208 may include a SSP ID sub-field 308, a device ID sub-field 310 and a data sub-field 312. The SSP ID sub-field 308 may store a unique identifier for each SSP 106, which may be a unique public key associated with the SSP 106. The device ID sub-field 310 may store a unique identifier of a user device connected to the SSP 106, that may be of the same product model as the user device 108 controlled by the gateway. The SSP 106 may provide security solutions for the user device 108. The unique identifier stored in the device ID sub-field 310 may serve to verify that the SSP 106 indeed has the user device that it is providing the security solution for. The data sub-field 312 may be used to store transaction-specific data, such as the digital signature of the user device 108 during registration, or a solution evaluation report for the review transaction. One or more sub-fields of the SSP information field 208 may be left empty for some types of transactions.

[0048] In the following paragraphs, details of each transaction type and the associated smart contract will be described.

[0049] Registration transactions

[0050] Any first-time entity may need to register in the blockchain network in order to access the services of the blockchain network. The registration process may be similar for both the gateways 104 and the SSPs 106. Each of the gateway 104 and the SSP 106 may require two types of registration – first, registration for themselves; and second, registration for each device that they control for the security solution. SSPs 106 may control similar user devices 108 as gateways 104, or a particular type of user device 108 depending on its business model.

For example, a SSP 106 that provides security solutions only for smart televisions may control many types of smart televisions, but may not control any other type of user devices 108 such as smart phones.

[0051] FIG. 4 shows an example of a gateway register transaction 400 according to various embodiments. A gateway 104 may be required to perform self-registration with the blockchain network before it may be permitted to register the user devices 108 that it controls. The gateway register transaction 400 may be filled out in the transaction data format 200. The transaction type field 202 may be filled out to indicate that this transaction is a registration. As this is a self-registration performed by the gateway 104, the gateway information field 206 may be filled only with the gateway address (GW_add) in the gateway identifier field 302. The gateway address may be, for example, the public key of this particular gateway. The device ID sub-field 304 and the data sub-field 306 may be filled with zero(s) or left empty. As there is no SSP 106 involved in this transaction, the SSP information field 208 may be filled with zero or left empty. The smart contract information field 210 may be filled with an address of a smart contract “registration smart contract”. This smart contract may be pre-written and stored on the blockchain. The smart contract may be distributed across the blockchain network, i.e., each node may store a copy of all available types of smart contract. Alternatively, the smart contracts may be stored in a centralized location, for example a server that is external to the blockchain network. By specifying the address of the registration smart contract, the gateway 104 may trigger the registration smart contract to be executed. When the registration smart contract is executed, it may invoke codes to carry out the registration procedures. In this example, the self-registration of the gateway 104 does not require payment of fees, so the amount field 212 may be left empty. The amount field 212 may be customizable to suit the application, for example, in another embodiment, the distribution system could charge a fee even for registration and in that embodiment, the amount field 212 may be filled with the system-defined monetary value. The PreTxnHash field 214 may store a hash of a string of zeros by default, since there is no transaction earlier than the self-registration for this gateway 104. The gateway 104 may generate a digital signature Sig_GW based on the information in the abovementioned fields, to complete the digital signature field 216. When the gateway 104 initiates this transaction, the registration smart contract stored at the address SC_reg may be triggered, and the registration smart contract may self-execute a set of pre-defined rules. The registration smart contract may perform the following tasks:

1. Check that GW_add, the gateway address in the gateway ID sub-field 302 has not been previously registered; and
2. Verify the validity of Sig_GW in the digital signature field 216.

[0052] If both the above two conditions are met, the registration smart contract may output 1, so that the gateway 104 may be successfully registered. If any one of the above two conditions are not met, the registration smart contract may output 0 so that the registration request is declined.

[0053] For all types of transactions, the digital signature field may be generated in a similar way as described above, using all the previous fields in the transaction data format 200 as the underlying message to be signed. The Time Stamp may be used to prevent replay attack.

[0054] After the gateway 104 has completed self-registration, the gateway 104 may then register the user devices 108 that it controls, in order to access services provided by SSPs, for example, to be able to initiate a successful transaction with the SSPs. If a gateway 104 has not registered any user devices 108, the distribution system may prohibit the gateway 104 from performing any other transactions. This is to prevent a malicious entity from registering many “empty shell” gateways, each one without any user device 108, for the purpose of launching attacks on nodes in the blockchain network. To register a specific user device 108, the gateway 104 may initiate a gateway device register transaction, like shown in FIG. 5.

[0055] FIG. 5 shows an example of a gateway device register transaction 500 according to various embodiments. The gateway device register transaction 500 may be filled in a similar manner as the gateway register transaction 400, except for a few fields such as the gateway information field 206 and the PreTxnHash field 214. The device ID sub-field 304 of the gateway information field 206 may be filled with the serial number, or public key, or a combination of both, of a particular user device 108. The device ID, denoted as D_ID, may uniquely identify the user device 108. The data sub-field 306 of the gateway information field 206 may be a signature generated by the user device 108, denoted as D_sig. Each user device 108 may have an embedded public-private key pair. The public-private key pair may be tamper-resistant. The public key may be provided by the user device manufacturer, and therefore publicly available. The gateway may trigger the user device 108 to generate such a signature. The underlying message used for this signature generation may include the timestamp for this transaction, the gateway ID and the device ID. The user device signature is required to verify that the gateway 104 indeed has control over the user device 108, so as to prevent the gateway 104 from falsely registering user devices that are not controlled by itself. The PreTxnHash field 214 may be filled with the hash result of the previous transaction by

the same gateway 104, which is the gateway register transaction 400. This is to make sure that a gateway 104 has registered itself before registering a user device 108. The gateway 104 may generate a digital signature of this transaction like described above, using the data in the fields 202, 204, 206, 208, 210, 212 and 214. The gateway device register transaction 500 may designate to the same smart contract SC_reg, as the gateway register transaction 400. Upon being activated, the smart contract SC_reg, may execute the following tasks:

1. Check that the gateway 104 is already registered;
2. Check that the user device 108 has not yet been registered;
3. Check the validity of the digital signature in the data sub-field 306;
4. Finds the previous transaction, and checks the validity of the PreTxnHash field 214; and
5. Checks the validity of the digital signature field 216.

[0056] Each gateway 104 may maintain locally, a database of all transactions that it has executed. This allows each gateway 104 to quickly identify the previous transaction, for efficiently computing the PreTxnHash field 214 and other required data. Upon successful verification of the above tasks, the smart contract SC_reg may output a Boolean value 1. If any of the above tasks yield a negative result, SC_reg may output a Boolean value 0. This may complete the user device registration.

[0057] FIG. 6 shows an example of a SSP register transaction 600 according to various embodiments. The format of the SSP register transaction 600 may be similar to the gateway register transaction 400. The SSP may similarly register itself using the SSP register transaction 600, and also register its device as shown in FIG. 7. In the SSP register transaction 600, the gateway information field 206 may be zero since there is no interaction with a gateway. The SSP information field 208 may store the SSP address, such as the SSP public key, in the SSP ID sub-field 308. The SSP register transaction 600 may indicate the address of the corresponding smart contract in the smart contract information field 210. The corresponding smart contract may be SC_reg, the same smart contract as for the gateway register transaction 400. The digital signature field 216 may include an SSP digital signature. The smart contract SC_reg may verify the legitimacy of this transaction by performing the same checking tasks as for the gateway register transaction 400.

[0058] FIG. 7 shows an example of a SSP device register transaction 700 according to various embodiments. The format of the SSP device register transaction 700 may be similar to the gateway device register transaction 500. The SSP information field 208 may be filled with the SSP address in the SSP ID sub-field 308, the device ID in the device ID sub-field 310 and the digital signature of the device in the data sub-field 312. The previous transaction

hash field 214 may be filled with the SSP register transaction 600. The registration smart contract may carry out a sequence of tasks or actions, similar to those for the gateway device register transaction 500, to check the legitimacy of the transaction.

[0059] Release transaction

[0060] FIG. 8 shows an example of a release transaction 800 according to various embodiments. An SSP 106 may create a release transaction 800 to make a new security solution available on the blockchain network. A release transaction may be created only by SSPs 106. The smart contract associated with this transaction may be a release smart contract, denoted as SC_rel. The data sub-field 312 under the SSP information 208 may store a link or a pointer, for example, a URL, or an address that directs to a location where the security solution being released by the SSP 106 is stored. The security solution may be stored in a memory of the SSP 106. The SSP 106 may limit access to the link, by using for example, a password, or an authentication request. The release transaction 800 may specify a deposit amount in the amount field 212. The deposit amount may be a monetary value deducted from the SSP's account. The purpose for this deposit is to motivate SSPs 106 to provide reliable and useful solutions to the gateways 104, and prevent them from offering low-quality solutions. A portion of the deposit may be deducted when a gateway gives a negative feedback for the security solution using a review transaction, which will be described in subsequent paragraphs. When the deposit amount of the security solution has fallen below a threshold value, the release smart contract may remove the security solution from the blockchain or block off access to the security solution, for example using a de-register transaction which will be described subsequently. As an alternative to the monetary deposit, the amount field 212 may indicate a feedback score of the SSP 106 or the security solution. The feedback score may also be referred herein as a reputation score. The feedback score may be deducted when a negative feedback is issued, similar to the deposit amount. The release smart contract, upon receiving the release transaction 800, may check the following:

1. The SSP 106 has registered the device (i.e., the previous transaction);
2. The PreTxnHash field 214 is correctly formed; and
3. The digital signature of the transaction is legitimate.

[0061] Upon the successful verification of the above three conditions, the release smart contract may store the *rel_deposit* value into its own storage, for future usage.

[0062] Interest transaction

[0063] FIG. 9 shows an example of an interest transaction 900 according to various embodiments. An interest transaction 900 may be created by a gateway 104, to make a

request to a SSP 106 for trying out a security solution offered by the SSP 106. With this transaction, the gateway 104 and the SSP 106 may enter into a contract, wherein the SSP 106 may transfer a trial version of the security solution to the gateway 104. The trial version of the security solution may have an expiry time, beyond which the security solution may cease to function. The transaction type field 202 may be indicated as “interest”. The gateway information field 206 may be filled with information relating to the gateway 104 that is making the request and its user device 108 that may be receiving the security solution. The data sub-field 306 may be filled with zero, as the user device signature may not be required in this transaction. The SSP information field 208 may be filled with information relating to the SSP 106 that provides the security solution that the gateway 104 is requesting to try. The data sub-field 312 of SSP info may be filled with the link of the security solution. The transaction may be targeted at the specific interest smart contract, with SC_int as its address. The amount field 212 may record an interest deposit, denoted herein as “int_deposit”. The gateway 104 may leave an interest deposit with the interest smart contract. The interest deposit may be transferred back to the gateway 104 after it submits a feedback, i.e. review of the security solution via a review transaction. The review may contain information on the merits and demerits of the security solution. The review may explicitly inform all the nodes in the blockchain network whether the solution worked for the user device 108. By having a gateway 104 deposit an amount for the trial, the distribution system may incentivize the gateway 104 to review the security solution that it tested. The interest smart contract may distribute the interest deposit equally between the SSP 106 and the gateway 104 if the gateway 104 does not review the security solution after a predefined time duration, i.e. before a review deadline. Alternatively, the interest deposit may be split according to other ratios and in accordance with other predefined rules. The review deadline may be later than the expiry time of the trial solution, so that the gateway 104 has sufficient time to review the trial solution. Upon receiving the interest transaction 900, the associated *interest smart contract* may check the following:

1. The gateway 104 has registered the device (i.e., the previous transaction);
2. The PreTxnHash field 214 is correctly formed;
3. The digital signature of this transaction is legitimate;
4. Verify that the same gateway 104 has not submitted an interest transaction 900 for the same security solution previously;
5. There is sufficient rel_deposit balance;

6. If the review deadline has expired, the smart contract may also check whether there exists a Review transaction initiated by the gateway 104 for this particular security solution. If the review transaction does not exist, the *interest deposit may be* forfeited and divided according to the pre-defined rules. If the gateway 104 has already submitted a review transaction, the interest deposit may be returned to the gateway 104.

[0064] The interest smart contract may verify that the gateway 104 had not submitted an earlier interest transaction 900 for the same security solution, so as to prevent the gateway 104 from behaving maliciously. It is not necessary for a gateway 104 to test a security solution multiple times. If the gateway 104 has already tested the security solution, the distribution system prevents a malicious gateway 104 in attempting to influence the reputation of a solution or SSP 106. By checking if there is sufficient *release deposit*, the smart contract may ensure that the security solution is still valid for gateways 104 to try out. If the release deposit falls below a threshold, the security solution may have received many negative reviews and therefore, should not be released to another gateway 104.

[0065] Review transaction

[0066] FIG. 10 shows an example of a review transaction 1000 according to various embodiments. After a gateway 104 has tested a trial security solution obtained from an SSP 106, it may review the security solution, via a review transaction 1000. As mentioned above, the gateway 104 may have an incentive to do so, in the form of receiving a refund of the interest deposit. In a review transaction 1000, the gateway 104 may indicate the outcome of the evaluation. For security solutions, there may be only two possible outcomes – success or failure; i.e., the security solution either succeeded in closing the vulnerability, or it failed. Therefore, the gateway 104 may indicate in the review transaction, either *success* or *failure*, corresponding to the outcome of the evaluation. The review transaction 1000 may be similar to the interest transaction 900, except that the data sub-field 306 of the gateway information 206 field may store the outcome of the testing for the security solution, denoted as “out”. The possible values of “out” may be either *success*, or *failure*. The amount field 212 may be zero, as there may be no money transfer in this transaction. When a gateway 104 wants to report that the security solution has failed, it may indicate “*failure*” in the data sub-field 306. However, there is a possibility that a gateway 104 may report a false review for a security solution, for example, to give advantage to another SSP 106. To reduce the occurrence of abuse of the review transaction, the distribution system may bar a gateway 104 from purchasing a security solution after it reports a security solution as *failure in a review transaction 1000*. This scheme may discourage the gateway 104 from lying about a failure as

the gateway may not be allowed to purchase a successful solution. If the solution works successfully, the gateway may specify *success* in the data sub-field 306. The previous transaction hash field 214 may store the hash value of the *interest* transaction 900 for this particular security solution. While it may be possible for an SSP 106 to incentivize a set of gateways 104 to report positive reviews of its security solution which does not work well, such a method of cheating may be impractical in a large network with many gateways 104. The SSP 106 may have to provide incentives to a large percentage of the gateways 104 to cheat the system successfully. This may be too expensive for an SSP to make eventual gain from the market, besides even being financially improbable. Upon receiving the review transaction, the associated review smart contract (SC_rev) may perform the following actions:

1. Checks that the gateway 104 has initiated an *interest* transaction earlier (the previous transaction);
2. Checks that the gateway 104 has not submitted a review report yet;
3. The digital signature field is correctly formed;
4. Based on the *out*, re-calculate the reputation of the security solution, and deduct money from the *rel_deposit* if *out* is set to *failure*; and
5. Trigger the *interest* smart contract to refund the *int_deposit* to the gateway.

[0067] The review smart contract may check that the gateway 104 has not submitted earlier review reports for the same security solution, to prevent the gateway 104 from flood the system with repetitive reviews. The review smart contract may re-compute the reputation score of the security solution whenever there is a review transaction 1000. After the gateway 104 has submitted a review, the review smart contract may return the *interest deposit* to the gateway 104. Actions 4 and 5 may trigger transactions that involve monetary values. As these may be basic money transfers which may be completed with a basic transaction that indicates (1) the address of an associated smart contract for remitting money, (2) the address of the recipient (such as the gateway 104 for *int_deposit* refund, or a central pool of funds for failure report deduction), and (3) the monetary value. These transactions may be performed using the transaction format 200. These transactions may be available for verification by the miners which may be the gateways 104, may be added into the blockchain.

[0068] Purchase transaction

[0069] FIG. 11 shows an example of a purchase transaction 1100 according to various embodiments. Any gateway 104 may create a purchase transaction 1100, to purchase a security solution from a SSP 106. The amount field 212 of the purchase transaction 1100 may

be filled with the purchase price, i.e. the payment amount to be transferred to the SSP 106 in exchange for the security solution. The distribution system may allow the gateway 104 to purchase a security solution even it has not tested it before. This may be reasonable, as the gateway 104 may have purchased other solutions from the same SSP 106 before and therefore trusts it, or may simply decide on the purchase based on the reputation score. The gateway 104 may initiate the purchase transaction of the security solution when the reputation score of the security solution is above a threshold. The gateway 104 may also compare the reputation scores of a plurality of security solutions available for its user device 108, and then initiate a purchase transaction 1100 for the security solution that has the highest reputation score. The gateway 104 may determine whether to initiate the purchase transaction 1100 or determine which security solution to purchase, using an associated smart contract. The gateway 104 need not have carried out an interest transaction 900 and a review transaction 1000 in order to initiate a purchase transaction 1100. As such, the previous transaction indicated in the PreTxnHash field 214 may not be limited to one type of transaction, for example, it may indicate a device registration transaction, or a review transaction. The smart contract associated with the purchase transaction 1100 may be, for example, a purchase smart contract having the address SC_pur.

[0070] Upon receiving the transaction, the associated purchase smart contract may perform the following actions:

1. Check whether the gateway 104 has a previous *review* transaction 1000 for the same security solution of the same SSP 106. If yes, and the review “*out*” (sub-field 306) is set to *success*, then compare the PreTxnHash field 214 using this *review* transaction 1000. If the “*out*” is *failure*, the purchase smart contract may discard the present purchase transaction. If no such *review* transaction exists, the purchase smart contract may search and find the gateway *device registration* transaction, and use it as the previous transaction, and verify the PreTxnHash field 214 is correctly formed.
2. Check the digital signature field 216 is properly formed;
3. Re-computes the reputation score for the solution; and
4. Transfer the payment to the SSP 106.

[0071] In the third action of re-computing the reputation score, the purchase smart contract may improve the reputation score based on the purchase transaction 1100. The rationale may be that every successful purchase may be an affirmation of the security solution. The last action of payment transfer may include triggering an associated transaction for money

transfer, for example, by activating another smart contract associated with money transfer. For effective and successful security solutions, the major contribution to the reputation score may come from purchase transactions 1100.

[0072] In addition to the above described transactions, the distribution system may also include an optional de-register transaction. The de-register transaction may serve to nullify any *register* transactions. Once a gateway 104 is no longer in control of a particular user device 108, or a user device 108 is no longer present in the network, or similarly if an SSP 106 has quit the market, the gateway 104 or the SSP 106 may initiate a de-register transaction to leave a record in the blockchain. The de-register transaction need not be implemented in the distribution system, as these inactive entities may still exist in the blockchain network without causing any storage or computational overhead to the blockchain network.

[0073] FIG. 12 shows a conceptual diagram 1200 of a blockchain network according to various embodiments. The blockchain 1210 may be a linked (chained) plurality of blocks 1204. Each block 1204 may be created by a miner. The blocks 1204 may be linked using their hash addresses. Each time a miner verifies a set of transactions, it may create a new block 1204 and add the new block 1204 to the blockchain 1210. The blockchain 1210 may function as a ledger of all transactions that take place on the blockchain network. In the blockchain 1210, each block may consist of transactions related to only one particular security solution, for example a solution denoted as *S*. This may be useful in computing the reputation score of the security solution efficiently. Each block 1204 may store the latest computed reputation score for one and only one security solution. To compute the reputation score efficiently, in addition to the hash addresses that link a current block to a previous block (referred herein as a first link 1206), a newly mined block may also contain the hash address that links it to the previous block that has transactions for the same security solution (referred herein as a second link 1208). With this linking of related transactions, a miner may compute the reputation score of a solution, by simply accessing the last block related to the solution *S*, and adding the reputation scores due to transactions in the current block to the reputation score stored in the last block related to the solution *S*. For register transactions for the gateway, the SSP or the devices, as these transactions are not related to any particular security solution, they may be mined and added to the latest block together with other grouped, security solution-related transactions, according to the transaction timestamp.

[0074] The blockchain network may include a plurality of unconfirmed and unordered transactions 1202. The unconfirmed and unordered transactions 1202 may include new transactions that have not yet been verified by the miners which may be the buyer nodes.

After the unconfirmed and unordered transactions 1202 have been verified by the miners, they may be added into the blockchain 1210. For each successfully mined block 1204, a new block ID may be created. The block ID may be the hash value of the block 1204. Further, the miner may chain the newly mined block 1204 to the previous block by listing the “Prev Blk ID”, i.e. indicated by the first link 1206. In addition, as mentioned above, each block 1204 may also be chained to the latest previous block (Prev Soln X Blk) that contains the transactions for the same security solution, indicated by the second link 1208. Each block 1204 may contain the transactions related to the same solution.

[0075] The miners may responsible for creating blocks of transactions to be added to the ledger. Only the buyer nodes such as the gateways 104 may be the miners in the distribution system, so as to prevent the seller nodes such as the SSPs 106 from gaining any advantage in the mining process. Besides the task of gathering the transactions and creating a block 1204, the miners may also be required to verify each transaction. For such transaction verification, a miner may repeat the same actions as the smart contracts do for each transaction. The miner may verify the conditions listed as being satisfied, as well as ascertain that the smart contracts have faithfully executed the stipulated actions. Once the verifications of the transactions gathered have been completed, the miner may create the new block 1204. Each block 1204 may store records of at least one transaction associated with the same software, for example two or more transactions. A miner which has verified a transaction and subsequently successfully created a block 1204 may be rewarded. This reward may come from the transaction processing fees obtained from executing the smart contracts, and may be provided by the SSPs 106.

[0076] The reputation score of security solutions may be computed by the miners, during the process of verification of transactions. The reputation score for a new untested solution may be initialized to zero. The reputation score of a solution S , may be denoted as $r(S)$. The reputation score may be stored in each block 1204. Since each block 1204 has only transactions related to one particular security solution, there may be only one reputation score stored in each block 1204. For every positive (*success*) review report submitted by a gateway 104, the associated smart contract may execute an *increment_rep()* function to increase the reputation score. For every negative (*failure*) review report submitted, the associated smart contract may execute a *decrement_rep()* function to decrease the reputation score. For all successful purchases, the associated smart contract may execute a *purchase_inc_rep()* function to increase the reputation score. The exact definition of these functions may be dependent on the reputation model chosen for deployment.

[0077] FIG. 13 shows an example of a distribution system 1300 according to various embodiments. A plurality of SSPs 106 and gateways 104 may be communicatively coupled via a network such as the Internet 102. For simplicity of illustration, only one gateway 104 and one SSP 106 is depicted in FIG. 13. The gateways 104 and the SSPs 106 may communicate via a blockchain platform 1310. The blockchain platform 1310 may store smart contracts 1302, un-consolidated transactions 1202, and mined blocks (blockchain 1210). In Action A1, a gateway 104 and an SSP 106 may register themselves and their devices by invoking the “register” smart contract 1312, and may create corresponding transactions, for example denoted as txn 1, txn2, txn 10. The transactions may be not recorded in sequence because there may be other transactions resulting from other gateway/SSP interactions. In Action A2, the SSP 106 may release its security solution by triggering the “release” smart contract 1314. A corresponding transaction, txn 11, may be created as a result. Actions A3.1, A3.2 and A3.3 may be a sequence of actions. In Action A3.1, the gateway 104 may invoke the “interest” smart contract 1316. By doing so, the gateway 104 may express its intention to try a particular solution “XX”. By checking a sequence of logic, if fulfilled, the “interest” smart contract 1316 may trigger Action A3.2. In Action A3.2, the “interest” smart contract 1316 may inform the SSP 106 that the gateway 104 wishes to try the solution. As a result, in Action A3.3, the SSP 106 may provide the solution to the gateway 104. After the gateway 104 has tried out the solution, it may review the solution in Action A4.1. When the gateway 104 initiates a review transaction, the “review” smart contract 1318 may be activated. The “review” smart contract 1318 may update the reputation score of the SSP 106 and this particular solution, through Action A4.2. Actions A5.1, A5.2 and A5.3 may be a sequence of actions. When the gateway 104 decides to purchase the particular solution, it may trigger Action A5.1 with a purchase transaction. The “purchase” smart contract 1320 may be activated to check, and then trigger Action A5.2 to inform the SSP 106 to release the solution. In Action 5.3, the SSP 106 may release the solution to the gateway 104. Action 5.1 may include updating the reputation score of the SSP 106. Each action may create associated transactions which the gateways 104 may mine to form blocks to be added to the blockchain 1210. Each block may record more than one transaction associated with the same software.

[0078] The distribution system 1300 may differ from traditional peer-to-peer (P2P) content distribution networks in that all the actions in the distribution system 1300 as represented through the format of transactions, may be recorded in the ledger. The blockchain 1210 may be the ledger. These actions may be integrity-checked, authenticity-verified and non-repudiation-achieved through cryptographic primitives, and carried out by smart contracts

and miners in the distribution system 1300. The distribution system 1300 may assume minimum trust among the participants, and any modification on the transactions may be easily detected. The distribution system 1300 may utilize smart contract technology to achieve rich functionalities, for example, enforcing time-dependent transactions like providing a deadline to submit reviews, storing of deposits etc. The distribution system 1300 may leverage on the usage of smart contract, to incentivize the participants to behave faithfully, and may penalize them otherwise. The distribution system 1300 may prevent malicious attacks, such as falsely providing negative reviews of successful solutions or falsely removing negative reviews. The distribution system 1300 may use the transactions and the enforcement of the smart contracts to compute the reputation score of the security solutions. The reputation score may help gateways 104 to quickly identify a trustworthy solution, as well as motivate the SSPs 106 to provide high-quality solutions.

[0079] FIG. 14 shows a flow diagram 1400 for a method of distributing software across a network. In some embodiments, the method may be performed by an autonomous system. In some embodiments, the method may be performed by an apparatus. The method may include elements 1402, 1404 and 1406. Element 1402 may include recording a transaction associated with software. A buyer node of the network may record the transaction. The transaction may be recorded in a new block of a blockchain. The new block may be linked to a preceding block that records a latest earlier transaction associated with the software. Element 1404 may include computing a reputation score of the software based on the transaction and a previous reputation score stored in the linked preceding block. A smart contract associated with the transaction may compute the reputation score. Element 1406 may include storing the computed reputation score in the new block. A further transaction associated with the software may be initiated based on the reputation score. A buyer node may initiate the further transaction. The buyer node that initiates the further transaction may be a different buyer node from the buyer node that recorded the earlier transaction. When the reputation score is above a threshold, the further transaction may be a purchase transaction for installing the software onto an electronic device. A smart contract may trigger an installation of the software, by the buyer node, onto the electronic device upon checking that the reputation score is above the threshold. The software may be applied to protect the electronic device from security vulnerabilities. When the reputation score is below another threshold, the further transaction may be a de-registration transaction for removing the software from the network. The reputation score of the software may indicate the trustworthiness or success rate of the software at resolving security vulnerabilities.

[0080] FIG. 15 is a diagram 1500 illustrating an example of a hardware implementation for an apparatus 1502' employing a distribution system 1514. In some embodiments, the apparatus 1502' may be the apparatus described above with reference to FIG. 14. The apparatus 1502' may include one or more computing devices. The distribution system 1514 may be implemented with a bus architecture, represented generally by the bus 1524. The bus 1524 may include any number of interconnecting buses and bridges depending on the specific application of the processing system 1514 and the overall design constraints. The bus 1524 links together various circuits including one or more processors and/or hardware components, represented by the processor 1504 and the computer-readable medium / memory 1506. The bus 1524 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0081] The processing system 1514 includes a processor 1504 coupled to a computer-readable medium / memory 1506. The processor 1504 is responsible for general processing, including the execution of software stored on the computer-readable medium / memory 1506. The software, when executed by the processor 1504, causes the processing system 1514 to perform the various functions described *supra* for any particular apparatus. The computer-readable medium / memory 1506 may also be used for storing data that is manipulated by the processor 1504 when executing software. The computer readable medium/memory 1506 may store computer readable instructions configured to generate a blockchain. The blockchain may be the blockchain 1210 described with reference to FIG. 12. The processor 1504 may be configured to generate the blockchain, for example, based on the computer readable instructions stored in the computer readable medium/memory 1506.

[0082] In the following, various aspects of this disclosure will be described:

[0083] Example 1 is a method of distributing software across a network. The method may include: recording a transaction associated with the software, in a new block of a blockchain; wherein the new block is linked to a preceding block that records a latest earlier transaction associated with the software; computing a reputation score of the software based on the transaction and a previous reputation score stored in the linked preceding block; and storing the computed reputation score in the new block.

[0084] In example 2, the subject matter of Example 1 may optionally include: initiating a further transaction associated with the software based on the reputation score.

[0085] In example 3, the subject matter of Example 2 may optionally include that the further transaction is one of a purchase transaction for installing the software onto an electronic

device when the reputation score is above a threshold and a de-register transaction for removing the software from the network when the reputation score is below a threshold.

[0086] In example 4, the subject matter of any one of Examples 1 to 3 may optionally include recording transactions associated with only one software in any one block of the blockchain.

[0087] In example 5, the subject matter of any one of Examples 1 to 4 may optionally include: broadcasting data about the transaction to a network; wherein the data comprises an address of a smart contract of a plurality of smart contracts stored in the blockchain, the smart contract corresponding to a type of the transaction.

[0088] In example 6, the subject matter of Example 5 may optionally include: retrieving the corresponding smart contract based on the address; and executing the corresponding smart contract; wherein the smart contract comprises a set of instructions which when executed, performs the transaction and computes the reputation score when a set of predefined rules are satisfied.

[0089] In example 7, the subject matter of any one of Examples 5 to 6 may optionally include that the data further comprises an address of the software.

[0090] In example 8, the subject matter of any one of Examples 1 to 7 may optionally include that a further block of the blockchain stores records of transactions associated with another software.

[0091] In example 9, the subject matter of any one of Examples 1 to 8 may optionally include that the software is a security software for protecting an electronic device.

[0092] In example 10, the subject matter of any one of Examples 1 to 9 may optionally include that the transaction is a review of the software, wherein the transaction increases or decreases the reputation score of the software depending on an outcome of the review.

[0093] In example 11, the subject matter of any one of Examples 1 to 10 may optionally include that the transaction is a purchase of the software, wherein the transaction increases the reputation score of the software.

[0094] Example 12 is a distribution system including: a memory; and at least one processor communicatively coupled to the memory. The at least one processor may be configured to: generate a blockchain that includes: a plurality of blocks, wherein each block of the plurality of blocks stores a record of a transaction associated with software; and a plurality of smart contracts, each smart contract corresponding to a type of transaction, wherein at least one smart contract of the plurality of smart contracts is configured to compute a reputation score of any software; wherein each block is linked to a preceding block that stores a record of a

latest earlier transaction associated with the same software as the transaction of its stored record; and wherein each block stores the reputation score of the software.

[0095] In example 13, the subject matter of Example 12 may optionally include that each record of a transaction comprises an address of the corresponding smart contract.

[0096] In example 14, the subject matter of any one of Examples 12 to 13 may optionally include that the at least one smart contract computes the reputation score stored in each block based on the transaction recorded in the block and further based on a previous reputation score stored in the linked preceding block.

[0097] In example 15, the subject matter of any one of Examples 12 to 14 may optionally include that each block of the blockchain stores a plurality of records of transactions associated with only one software.

[0098] In example 16, the subject matter of any one of Examples 12 to 15 may optionally include that each smart contract comprises a set of instructions which when executed, performs the corresponding transaction and computes the reputation score when a set of predefined rules are satisfied.

[0099] In example 17, the subject matter of any one of Examples 12 to 16 may optionally include that the record of the transaction comprises an address of the software.

[00100] Example 18 is a distribution system including: a plurality of nodes, wherein each node is configured to selectively initiate transactions; wherein at least a subset of the plurality of nodes is configured to implement a plurality of smart contracts, each smart contract associated with a type of transactions; wherein for each transaction initiated by one of the plurality of nodes, the transaction associated with software: another node of the plurality of nodes is configured to record the transaction, in a new block of a blockchain, wherein the new block is linked to a preceding block that records a latest earlier transaction associated with the software; and wherein at least one smart contract of the plurality of smart contracts is configured to compute a reputation score of the software based on the transaction and a previous reputation score stored in the linked preceding block.

[00101] In example 19, the subject matter of Example 18 may optionally include that the plurality of nodes comprises: a plurality of supplier nodes that provides software; and a plurality of buyer nodes, each buyer node configured to dispatch software to a respective set of electronic devices.

[00102] In example 20, the subject matter of Example 19 may optionally include that the plurality of buyer nodes is configured to verify the transaction before one buyer node of the plurality of buyer nodes records the transaction in the new block.

[00103] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced. It will be appreciated that common numerals, used in the relevant drawings, refer to components that serve a similar or the same purpose.

[00104] It will be appreciated to a person skilled in the art that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[00105] It is understood that the specific order or hierarchy of blocks in the processes / flowcharts disclosed is an illustration of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of blocks in the processes / flowcharts may be rearranged. Further, some blocks may be combined or omitted. The accompanying method claims present elements of the various blocks in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[00106] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean "one and only one" unless specifically so stated, but rather "one or more." The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any aspect described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects. Unless specifically stated otherwise, the term "some" refers to one or more. Combinations such as "at least one of A, B, or C," "one or more of A, B, or C," "at least one of A, B, and C," "one or more of A, B, and C," and "A, B, C, or any combination

thereof" include any combination of A, B, and/or C, and may include multiples of A, multiples of B, or multiples of C. Specifically, combinations such as "at least one of A, B, or C," "one or more of A, B, or C," "at least one of A, B, and C," "one or more of A, B, and C," and "A, B, C, or any combination thereof" may be A only, B only, C only, A and B, A and C, B and C, or A and B and C, where any such combinations may contain one or more member or members of A, B, or C. All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the claims. The words "module," "mechanism," "element," "device," and the like may not be a substitute for the word "means." As such, no claim element is to be construed as a means plus function unless the element is expressly recited using the phrase "means for."