OFF-CHAIN INTERACTION FOR ON-CHAIN PROCESSING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of the filing date of India Provisional Patent Application No. 202341043888 filed on June 30, 2023, which is herein incorporated by reference.

BACKGROUND

[0002] An obstacle in the world of crypto (or other blockchain operations) is the complex process of paying for transactions or operations on blockchains. Each operation, whether it is a simple token transfer or a more intricate interaction with a smart contract, incurs a cost or operational value known as a "gas" fee. This represents the computational effort required to execute the operation. In the case of Ethereum, this gas fee must be paid in the blockchain's native token, ETH.

[0003] Despite the availability of stablecoins like USDC for conducting transactions, users are still required to maintain a separate balance of ETH to cover the gas fees on Ethereum. This often leads users to adopt complex and sometimes expensive methods. Some rely on on-ramp services to convert fiat currencies into native tokens like ETH, while others purchase ETH on a centralized crypto exchange and then transfer it to their wallet. However, both of these strategies require additional steps and lack the simplicity and immediacy that users are accustomed to in traditional financial transactions. Furthermore, these methods expose users to the volatility of crypto exchange rates because they need to consistently purchase ETH even when a different cryptocurrency or stablecoin is being used for the payment transaction.

[0004] Other problems with processing additional blockchain transactions, which can result in slow overall processing and high energy consumption. The account funding approaches (onramp services or transfer via centralized crypto exchange) mentioned above und can require blockchain transactions in order to

initially transfer funds to a user's wallet. For example, an Ethereum transaction can take 1-10 minutes to complete depending upon the number of required confirmations, while a Bitcoin transaction can take between 1-1.5 hours to complete. Further, the number of Ethereum transactions is currently over 1.1 million per day and the number of Bitcoin transactions is over 700,000 per day. There are reports that the carbon footprint of 1 Ethereum transaction can equal 59 kgCO2, and an energy consumption of 125 kWh. The amount of energy consumed by blockchain based crypto transactions is significant and ways to reduce this are needed.

[0005] Embodiments address these and other problems, individually and collectively.

BRIEF SUMMARY

[0006] One embodiment of the invention includes a method comprising: transmitting, by a communication device to a master computer, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction, wherein the master computer initiates an authorization request message with a gateway computer in communication with an authorizing entity computer; and transmitting, by the communication device to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein the node requests verification of the off-chain transaction using a first smart contract associated with the master computer, verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0007] Another embodiment of the invention comprises a communication device comprising: a processor; and a non-transitory computer readable medium, the non-transitory computer readable medium comprising code, executable by the processor for performing operations including: transmitting, to a master computer, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction, wherein the master computer initiates an authorization request message with a gateway computer in communication with an authorizing entity computer; and transmitting, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction,

wherein the node requests verification the off-chain transaction using a first smart contract associated with the master computer, verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0008] Another embodiment of the invention includes a method comprising: receiving, by a master computer from a communication device, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction; initiating, by the master computer, an authorization request message with a gateway computer in communication with an authorizing entity computer; receiving, by the master computer, an authorization response message from the authorizing entity computer; and transmitting, by the master computer, an off-chain transaction response to the communication device, wherein the communication device transmits, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein node requests verification the off-chain transaction using a first smart contract and verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0009] Another embodiment of the invention comprises a master computer comprising: a processor; and a non-transitory computer readable medium. The non-transitory computer readable medium comprises code, executable by the processor, to perform operations comprising: receiving, from a communication device, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction; initiating an authorization request message with a gateway computer in communication with an authorizing entity computer; receiving an authorization response message from the authorizing entity computer; and transmitting an off-chain transaction response to the communication device, wherein the communication device transmits, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein node requests verification the off-chain transaction using a first smart contract and verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0010] These and other embodiments are described in further detail below. .

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 shows a block diagram of a system according to an embodiment.

[0012] FIG. 2A illustrates a diagram and process flow illustrating a first interaction method.

[0013] FIG. 2B illustrates a diagram and process flow illustrating a second interaction method.

[0014] FIG. 2C illustrates a diagram and process flow illustrating a third interaction method.

[0015] FIG. 3 shows a block diagram of an examplary communcation device acording to embodiments.

[0016] FIG. 4 shows a block diagram of an examplary node computer according to embodiments.

[0017] FIG. 5 shows a block diagram of an examplary node computer according to embodiments.

[0018] FIG. 6 shows a diagram illustrating a portion of a blockchain according to some embodiments.

TERMS

[0019] A “key” may include a piece of information that is used in a cryptographic algorithm to transform input data into another representation. A cryptographic algorithm can be an encryption algorithm that transforms original data into an alternate representation, or a decryption algorithm that transforms encrypted information back to the original data. Examples of cryptographic algorithms may include triple data encryption standard (TDES), data encryption standard (DES), advanced encryption standard (AES), etc.

[0020] A "public key" may include an encryption key that may be shared openly and publicly. The public key may be designed to be shared and may be

configured such that any information encrypted with the public key may only be decrypted using a private key associated with the public key (i.e., a public/private key pair).

[0021] A "private key" may include any encryption key that may be protected and secure. A private key may be securely stored at an entity and may be used to decrypt any information that has been encrypted with an associated public key of a public/private key pair associated with the private key.

[0022] A “public/private key pair” may refer to a pair of linked cryptographic keys generated by an entity. The public key may be used for public functions such as encrypting a message to send to the entity or for verifying a digital signature which was supposedly made by the entity. The private key, on the other hand may be used for private functions such as decrypting a received message or applying a digital signature. In some embodiments, the public key may be authorized by a body known as a Certification Authority (CA) which stores the public key in a database and distributes it to any other entity which requests it. The private key can typically be kept in a secure storage medium and will usually only be known to the entity. Public and private keys may be in any suitable format, including those based on Rivest-Shamir-Adleman (RSA) or elliptic curve cryptography (ECC).

[0023] An “authorizing entity” may be an entity that authorizes a request. Examples of an authorizing entity may be an issuer, a governmental agency, a document repository, an access administrator, etc. An authorizing entity may operate an authorizing entity computer. An “issuer” may refer to a business entity (e.g., a bank) that issues and optionally maintains an account for a user. An issuer may also issue payment credentials stored on a user device, such as a cellular telephone, smart card, tablet, or laptop to the consumer.

[0024] A “resource provider” may be an entity that can provide a resource such as goods, services, information, and/or access to a location (e.g., a parking space, a transit terminal, etc.). Examples of resource providers include merchants, governmental authorities, secure data providers, etc. A resource provider may operate one or more access devices.

[0025] An "acquirer" may typically be a business entity (e.g., a commercial bank) that has a business relationship with a particular merchant or other entity.

Some entities can perform both issuer and acquirer functions. Some embodiments may encompass such single entity issuer-acquirers. An acquirer may operate an acquirer computer, which can also be generically referred to as a "transport computer."

[0026] A "communication device" may comprise any suitable electronic device capable of remote electronic communication. Examples of remote communication capabilities include using a mobile phone (wireless) network, wireless data network (*e.g.*, 3G, 4G or similar networks), Wi-Fi, Bluetooth, Bluetooth Low Energy (BLE), Wi-Max, or any other communication medium that may provide access to a network such as the Internet or a private network. Examples of communication devices include mobile phones (*e.g.*, cellular phones), PDAs, tablet computers, net books, laptop computers, wearable devices (e.g., watches), vehicles such as automobiles and motorcycles, personal music players, hand-held specialized readers, etc.

[0027] A "user" may include an individual. In some embodiments, a user may be associated with one or more personal accounts and/or mobile devices. The user may also be referred to as a cardholder, account holder, or consumer in some embodiments.

[0028] "Credentials" may comprise any evidence of authority, rights, or entitlement to privileges. For example, access credentials may comprise permissions to access certain tangible or intangible assets, such as a building or a file. Examples of credentials may include passwords, passcodes, or secret messages. In another example, payment credentials may include any suitable information associated with and/or identifying an account (e.g., a payment account and/or payment device associated with the account). Such information may be directly related to the account or may be derived from information related to the account. Examples of account information may include an "account identifier" such as a PAN (primary account number or "account number"), a token, a subtoken, a gift card number or code, a prepaid card number or code, a user name, an expiration date, a CVV (card verification value), a dCVV (dynamic card verification value), a CVV2 (card verification value 2), a CVC3 card verification value, etc. An example of a PAN is a 16-digit number, such as "4147 0900 0000 1234". In some embodiments, credentials may be considered sensitive information.

[0029] A “token” may be a substitute value for a credential. A token may be a string of numbers, letters, or any other suitable characters. Examples of tokens include access tokens such as payment tokens, data that can be used to access secure systems or locations, etc.

[0030] A "payment token” may include an identifier for a payment account that is a substitute for an account identifier, such as a primary account number (PAN) and/or an expiration date. For example, a token may include a series of alphanumeric characters that may be used as a substitute for an original account identifier. For example, a token “4900 0000 0000 0001” may be used in place of a PAN “4147 0900 0000 1234.” In some embodiments, a token may be “format preserving” and may have a numeric format that conforms to the account identifiers used in existing transaction processing networks (e.g., ISO 8583 financial transaction message format). In some embodiments, a token may be used in place of a PAN to initiate, authorize, settle or resolve a payment transaction or represent the original credential in other systems where the original credential would typically be provided. In some embodiments, a token value may be generated such that the recovery of the original PAN or other account identifier from the token value may not be computationally derived. Further, in some embodiments, the token format may be configured to allow the entity receiving the token to identify it as a token and recognize the entity that issued the token.

[0031] “Tokenization” is a process by which sensitive data is replaced with substitute data. For example, a real credential (e.g., a primary account number (PAN)) may be tokenized by replacing the real account identifier with a substitute number that may be associated with the real credential. Further, tokenization can be applied to any other information to substitute the underlying information with a token. “Token exchange” or “de-tokenization” can be a process of restoring the data that was substituted during tokenization. For example, a token exchange may include replacing a payment token with its associated primary account number (PAN). Further, de-tokenization or token exchange may be applied to any other information to retrieve the substituted information from a token. In some embodiments, token exchange can be achieved via a transactional message, such as an ISO message, an application programming interface (API), or another type of web interface (e.g., web request).

[0032] A "token service computer" can include a system that services tokens. In some embodiments, a token service computer can facilitate requesting, determining (e.g., generating) and/or issuing tokens, as well as maintaining an established mapping of tokens to primary account numbers (PANs) in a repository (e.g., token vault). In some embodiments, the token service computer may establish a token assurance level for a given token to indicate the confidence level of the token to PAN binding. The token service computer may include or be in communication with a token vault where the generated tokens are stored. The token service computer may support token processing of payment transactions submitted using tokens by de-tokenizing the token to obtain the actual PAN.

[0033] A "token domain" may indicate an area and/or circumstance in which a token can be used. Examples of the token domain may include, but are not limited to, payment channels (e.g., e-commerce, physical point of sale, etc.), POS entry modes (e.g., contactless, magnetic stripe, etc.), and merchant identifiers to uniquely identify where the token can be used. A set of parameters (i.e., token domain restriction controls) may be established as part of token issuance by the token service computer that may allow for enforcing appropriate usage of the token in payment transactions. For example, the token domain restriction controls may restrict the use of the token with particular presentment modes, such as contactless or e-commerce presentment modes. In some embodiments, the token domain restriction controls may restrict the use of the token at a particular merchant that can be uniquely identified. Some exemplary token domain restriction controls may require the verification of the presence of a token cryptogram that is unique to a given transaction. In some embodiments, a token domain can be associated with a token requestor.

[0034] "Token expiry date" may refer to the expiration date/time of the token. The token expiry date may be passed among the entities of the tokenization ecosystem during transaction processing to ensure interoperability. The token expiration date may be a numeric value (e.g., a 4-digit numeric value). In some embodiments, the token expiry date can be expressed as a time duration as measured from the time of issuance.

[0035] A "contract address" may include a digital address that identifies a smart contract. A contract address can identify a smart contract on a blockchain. A contract address could also be a public key of a smart contract.

[0036] An "interaction application" can be an application that can allow for a user to perform an interaction with another entity. One example of an interaction application is a digital wallet or digital wallet application.

[0037] A "digital wallet" may contain electronic information for conducting transactions. A digital wallet may store user profile information, payment credentials, bank account information, cryptocurrency account information, one or more digital wallet identifiers and/or the like and can be used in a variety of transactions, such as but not limited to eCommerce, social networks, money transfer/personal payments, mobile commerce, proximity payments, cryptocurrency transactions, gaming, and/or the like for retail purchases, digital goods purchases, utility payments, purchasing games or gaming credits from gaming websites, transferring funds between users, and/or the like. A digital wallet may be designed to streamline the purchase and payment process. A digital wallet may allow the user to load one or more payment cards onto the digital wallet so as to make a payment without having to enter an account number or present a physical card.

[0038] A "cryptocurrency network" may include a one or more computers that participate in maintaining a cryptocurrency ledger. In some cryptocurrency networks, the distributed cryptocurrency ledger may comprise a blockchain.

[0039] A "blockchain" can be a distributed database that maintains a continuously growing list of records secured from tampering and revision. A blockchain can be a non-fungible token blockchain, a cryptocurrency blockchain, or a combination thereof. A blockchain may include a number of blocks of interaction records. Each block in the blockchain can contain also include a timestamp and a link to a previous block. Stated differently, interaction records in a blockchain may be stored as a series of "blocks," or permanent files that include a record of a number of interactions occurring over a given period of time. Blocks may be appended to a blockchain by an appropriate node after it completes the block and the block is validated. Each block can be associated with a block header. In embodiments of the invention, a blockchain may be distributed, and a copy of the blockchain may be

maintained at each full node in a verification network. Any node within the verification network may subsequently use the blockchain to verify interactions. A blockchain can be stored, maintained, and updated in a distributed manner in a peer-to-peer network. A blockchain can be associated with a cryptocurrency application, such as Bitcoin or Ethereum, Ripple, Dash, Litecoin, Dogecoin, zCash, Tether, Bitcoin Cash, Cardano, Stellar, EOS, NEO, NEM, Bitshares, Decred, Augur, Komodo, PIVX, Waves, Steem, Monero, Golem, Stratis, Bytecoin, Ardor, or digital currency exchanges, such as Coinbase, Kraken, CEX.IO, Shapeshift, Poloniex, Bitstamp, Coinmama, Bisq, LocalBitcoins, Gemini and others.

[0040] A "blockchain network" can include a computer network that maintains a blockchain. A blockchain network includes nodes that perform processing to maintain a blockchain.

[0041] A "block header" can be a header including information regarding a block in a blockchain. A block header can be used to identify a particular block on a blockchain. A block header can comprise any suitable information, such as a previous hash, a Merkle root, a timestamp, and a nonce. In some embodiments, a block header can also include a difficulty value.

[0042] A "cryptocurrency blockchain" can include a blockchain that stores cryptocurrency. A cryptocurrency blockchain can be utilized to transfer cryptocurrency from one public address to another public address.

[0043] A "cryptocurrency" can include a digital currency. A cryptocurrency can include a digital currency in which transactions are verified and records maintained by a decentralized system using cryptography. A cryptocurrency may not need to be maintained by a centralized authority.

[0044] An "interaction" may include a reciprocal action or influence. An interaction can include a communication, contact, or exchange between parties, devices, and/or entities. Example interactions include a transaction between two parties and a data exchange between two devices. In some embodiments, an interaction can include a user requesting access to secure data, a secure webpage, a secure location, and the like. In other embodiments, an interaction can include a payment transaction in which two devices can interact to facilitate a payment. An interaction can be a transfer of a resource from a first entity to a second entity.

[0045] A "resource" can include digital items and/or physical items. A resource can be an item that is obtainable. A resource can be owned by an entity. A resource can be a physical item such as goods, etc. A resource can be a digital item such as non-fungible tokens, etc.

[0046] A "processing network computer" may include a server computer used for interaction processing. In some embodiments, the network processing computer may be coupled to a database and may include any hardware, software, other logic, or combination of the preceding for servicing the requests from one or more client computers. The network processing computer may comprise one or more computational apparatuses and may use any of a variety of computing structures, arrangements, and compilations for servicing the requests from one or more client computers. In some embodiments, a network processing computer may include data processing subsystems, networks, and operations used to support and deliver authorization services, exception file services, and clearing and settlement services. An exemplary network processing computer may include VisaNet™. Networks that include VisaNet™ are able to process credit card transactions, debit card transactions, and other types of commercial transactions. VisaNet™, in particular, includes an integrated payments system (Integrated Payments system) which processes authorization requests and a Base II system, which performs clearing and settlement services. The network processing computer may use any suitable wired or wireless network, including the Internet.

[0047] The processing network computer may process interaction-related messages (e.g., authorization request messages and authorization response messages) and determine the appropriate destination computer (e.g., an issuer computer) for the interaction-related messages. In some embodiments, the network processing computer may authorize interactions on behalf of an issuer. The network processing computer may also manage and/or facilitate the clearing and settlement of interactions. In some cases, the network processing computer can include a token service computer, which can perform tokenization and/or de-tokenization services as described above.

[0048] An "authorization request message" may be an electronic message that requests authorization for an interaction. In some embodiments, it is sent to a

transaction processing computer and/or an issuer of a payment card to request authorization for a transaction. An authorization request message according to some embodiments may comply with International Organization for Standardization (ISO) 8583, which is a standard for systems that exchange electronic transaction information associated with a payment made by a user using a payment device or payment account. The authorization request message may include an issuer account identifier that may be associated with a payment device or payment account. An authorization request message may also comprise additional data elements corresponding to "identification information" including, by way of example only: a service code, a CVV (card verification value), a dCVV (dynamic card verification value), a PAN (primary account number or "account number"), a payment token, a username, an expiration date, etc. An authorization request message may also comprise "transaction information," such as any information associated with a current transaction, such as the transaction value, merchant identifier, merchant location, acquirer bank identification number (BIN), card acceptor ID, information identifying items being purchased, etc., as well as any other information that may be utilized in determining whether to identify and/or authorize a transaction.

[0049] An "authorization response message" may be a message that responds to an authorization request. In some cases, it may be an electronic message reply to an authorization request message generated by an issuing financial institution or a transaction processing computer. The authorization response message may include, by way of example only, one or more of the following status indicators: Approval -- transaction was approved; Decline -- transaction was not approved; or Call Center -- response pending more information, merchant must call the toll-free authorization phone number. The authorization response message may also include an authorization code, which may be a code that a credit card issuing bank returns in response to an authorization request message in an electronic message (either directly or through the transaction processing computer) to the merchant's access device (e.g., POS equipment) that indicates approval of the transaction. The code may serve as proof of authorization.

[0050] An "authorizing entity" may be an entity that authorizes a request. Examples of an authorizing entity may be an issuer, a governmental agency, a document repository, an access administrator, etc. An authorizing entity may operate

an authorizing entity computer. An "issuer" may refer to a business entity (e.g., a bank) that issues and optionally maintains an account for a user. An issuer may also issue payment credentials stored on a user device, such as a cellular telephone, smart card, tablet, or laptop to the consumer, or in some embodiments, a portable device.

[0051] The term "verification" and its derivatives may include a process that utilizes information to determine whether an underlying subject is valid under a given set of circumstances. Verification may include any comparison of information to ensure some data or information is correct, valid, accurate, legitimate, and/or in good standing.

[0052] A "processor" may include a device that processes something. In some embodiments, a processor can include any suitable data computation device or devices. A processor may comprise one or more microprocessors working together to accomplish a desired function. The processor may include a CPU comprising at least one high-speed data processor adequate to execute program components for executing user and/or system-generated requests. The CPU may be a microprocessor such as AMD's Athlon, Duron and/or Opteron; IBM and/or Motorola's PowerPC; IBM's and Sony's Cell processor; Intel's Celeron, Itanium, Pentium, Xeon, and/or XScale; and/or the like processor(s).

[0053] A "memory" may be any suitable device or devices that can store electronic data. A suitable memory may comprise a non-transitory computer readable medium that stores instructions that can be executed by a processor to implement a desired method. Examples of memories may comprise one or more memory chips, disk drives, etc. Such memories may operate using any suitable electrical, optical, and/or magnetic mode of operation.

[0054] A "server computer" may include a powerful computer or cluster of computers. For example, the server computer can be a large mainframe, a minicomputer cluster, or a group of servers functioning as a unit. In one example, the server computer may be a database server coupled to a Web server. The server computer may comprise one or more computational apparatuses and may use any of a variety of computing structures, arrangements, and compilations for servicing the requests from one or more client computers.

DETAILED DESCRIPTION

[0055] Embodiments of the invention can use a master computer. In some embodiments, the master computer may be characterized as a paymaster computer. A paymaster is a specialized type of smart contract account that can sponsor gas fees for user Contract Accounts (think of these as user-centric smart contracts). Embodiments of the invention liberate users from the need to hold native blockchain tokens or constantly engage in bridging tokens merely to cover gas fees. From a user's perspective, this solution is appealing in its simplicity and ease of adoption.

[0056] The master computer can abstract away the complexities of the gas fee mechanism while offering alternative ways for fee funding. An implementation does this by accepting the user's gas fee payment off-chain from a device such as a payment card or payment card account, covering the equivalent amount on-chain, on behalf of the user. The gas fee experience on the user's end is as simple as a regular card payment. Users can choose to use such a paymaster when sending User Operations. User Operations are like regular blockchain interactions, in that they specify the action the user wants to take on the blockchain. But unlike transactions, User Operations do not need to be signed by an externally owned account and can be verified and executed directly by the user Contract Accounts.

[0057] The off-chain gas payment capability can involve a first smart contract, which can be master smart contract associated with a master computer. The master smart contract can delegate all necessary checks and sourcing of information to an off-chain component such as an off-chain master computer. The on-chain master smart contract can then use the data and approval provided by the off-chain component to authorize and pay for the gas fee. The way to reliably transfer this information from the off-chain service to the paymaster contract is through public key cryptography. The off-chain web service can a secret key (or private key) to produce a digital signature to send along with the information. The master smart contract can in turn use the corresponding public key to verify the signature and therefore the authenticity of the information.

[0058] A master computer and a master smart contract offers users an alternative way to pay transaction fees. The recent ERC-4337 upgrade to the Ethereum network allows users to delegate gas fee payment to a 3rd party through

the help of a class of smart contracts. Embodiments of the invention allow users to pay gas fees in fiat currency off-chain using a credit card or other payment device. A master service provider that operates a master service provider computer and accepts the off-chain card payment. The corresponding master smart contract covers gas fees on-chain on behalf of the user. Thus, the need for a blockchain user to hold any amount of a native blockchain currency (e.g., Ether for Ethereum) to be able to send transactions is obviated.

[0059] Embodiments of the invention have a number of other technical advantages. As user operational values (e.g., gas fees) are paid using fiat currency, the number of transactions processed by a blockchain network can be reduced, and this improves the overall processing speed of the blockchain network because fewer transactions need to be processed. For example, the account funding approaches (onramp services or transfer via centralized crypto exchange) mentioned above in the Background can require blockchain transactions in order to initially transfer funds to a user's wallet. These transactions do not need to take place in embodiments of the invention. There is no need to transfer funds (e.g., ETH) to a user account for the purposes of paying gas fees. For example, as noted above, an Ethereum transaction can take 1-10 minutes to complete depending upon the number of required confirmations, while a Bitcoin transaction can take between 1-1.5 hours to complete. In contrast, performing an off-chain payment card authorization to pay gas fees can take just a few seconds. Thus, reducing the number of transactions speeds overcall processing of the blockchain network and can save thousands of hours of absolute time and computational time. Further, the number of Ethereum transactions is currently over 1.1 million per day and the number of Bitcoin transactions is over 700,000 per day. There are reports that the carbon footprint of 1 Ethereum transaction can equal 59 kgCO_2, and an energy consumption of 125 kWh, while a single payment card transaction would be equal 0.00045 kgCO_2, and an energy consumption of 0.0015 kWh. When viewed at the scale of the overall network reducing the number of transactions conducted on blockchains such as cryptocurrency blockchains can significantly reduce the amount of power and natural resources consumed by such processing. In addition, as explained above, embodiments of the invention are much more convenient for users than paying for gas fees in the currency of the blockchain.

[0060] FIG. 1 shows a system according to an embodiment. The system 10 shows a user communication device 20 in communication with a blockchain network 40 comprising a plurality of blockchain nodes 40A-40E, and a master computer 50. The master computer 50 can be an off-chain component. The master computer 50 can be in communication with a gateway computer 60, a processing network computer 70, and an authorizing entity computer 80. The communication device 20 can have an interaction application 30, which can manage operation of cryptographic keys associated with an account of a user on a blockchain managed by the blockchain network 40. The master computer 50 can also manage and store a master computer public key and a master computer private key.

[0061] For simplicity of illustration, a certain number of components are shown in FIG. 1. It is understood, however, that embodiments of the invention may include more than one of each component. In addition, some embodiments of the invention may include fewer than or greater than all of the components shown in FIG. 1.

[0062] Messages between the devices included in the system 100 illustrated in FIG. 1 can be transmitted using a secure communications protocol such as, but not limited to, File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure Hypertext Transfer Protocol (HTTPS), SSL, ISO (e.g., ISO 8583) and/or the like. The communications network may include any one and/or the combination of the following: a direct interconnection; the Internet; a Local Area Network (LAN); a Metropolitan Area Network (MAN); an Operating Missions as Nodes on the Internet (OMNI); a secured custom connection; a Wide Area Network (WAN); a wireless network (e.g., employing protocols such as, but not limited to a Wireless Application Protocol (WAP), I-mode, and/or the like); and/or the like. The communications network can use any suitable communications protocol to generate one or more secure communication channels. A communications channel may, in some instances, comprise a secure communication channel, which may be established in any known manner, such as through the use of mutual authentication and a session key, and establishment of a Secure Socket Layer (SSL) session.

[0063] FIG. 2A illustrates an example framework 100 to allow for an off-chain payment of gas fees. Framework 100 can be used in a “pay per transaction” model, where each transaction is paid for off-chain each time a transaction or event is

performed on the blockchain. Stated another way, each transaction initiated by an interaction application 110 such as a cryptowallet may initiate one or more steps described with respect to framework 100. Framework 100 can also be thought of as logically containing a "user domain" component, a "blockchain environment" component facilitated by a blockchain network," and "off-chain environment" component, as illustrated in FIG. 2A. The blockchain network would have a plurality of nodes such as those shown in FIG. 1, where the nodes may have one or more of the specific components shown in the blockchain environment in FIG. 2A.

[0064] Illustrated in FIG. 2A is interaction application 110 such as a cryptowallet that can be present in a user communication device in a user domain. An off-chain environment can include a master computer 120 (e.g., a paymaster computer), a gateway computer 130 such as a payment gateway computer (e.g., CyberSource™), a processing network computer 132, and an authorizing entity computer 134.

[0065] The blockchain component can include features or software that would reside in the blockchain network (e.g., blockchain network 40 in FIG. 1). The blockchain component includes a first smart contract 140 (e.g., a paymaster smart contract), a mempool 150, a bundler/node 160, an entry point contract 170, a second smart contract 180 (e.g., a smart contract account of associated with the user), and representative blocks 191, 192, 193 of a blockchain.

[0066] The interaction application 110 can be a cryptowallet that can store public and/or private keys in a user communication device, along with cryptocurrency amounts or other cryptocurrency-based information, to perform cryptocurrency transactions or functions. The interaction application 110 can belong to a user of a communication device that wishes to perform functions with respect to a blockchain which requires gas fees.

[0067] Master computer 120 may be a paymaster service computer and can provide a service which can sponsor or provide a transaction gas fee on behalf of users. A "paymaster" may be an entity that can sponsor gas fees for transactions on Ethereum-based networks through the use of account abstraction, such as for example, as specified in the ERC-4337 standard.

[0068] The gateway computer 130 can initiate payment transactions with issuers of fiat currency accounts such as credit or debit card accounts. The gateway 130 can a merchant processor computer, an acquirer computer, etc.

[0069] The first smart contract 140 can be a paymaster smart contract, which facilitates transactions on a blockchain for a paymaster. The paymaster smart contract can be a smart contract of a particular paymaster or a set of smart contracts, or a smart contract account, which can facilitate transaction sponsorship by allowing third-party designed mechanisms to pay for transactions. In some examples, a paymaster smart contract is a smart contract or computer logic which can allow for the payment of a transaction fee on behalf of a user, and executing any logic to decide whether it should forward a transaction in the blockchain.

[0070] Mempool 150 is a memory pool stored in memory in one or more nodes of a blockchain network. In some examples, mempool 150 is a waiting area for transaction that have not yet been executed, are unconfirmed by a blockchain, and have not yet been added to a block. Stated alternatively, mempool 150 can contain transactions which are still being processed by the blockchain.

[0071] Bundler/node 160 can be a node which facilitates transactions in a blockchain. For example, in the context of Ethereum, a “node” can be any instance of Ethereum client software on a computer that is connected to other computers also running Ethereum software, forming a network. A client is an implementation of Ethereum that verifies data against the protocol rules and keeps the network secure. A Bundler can be a node that obtains one or more requested user operations from a Mempool to create a bundle of transactions that is signed and submitted to the blockchain network as a single transaction.

[0072] Entry point contract 170 can be a smart contract which can allow for additional mechanisms and functionalities on a crypto wallet account. In some examples, an entry point or entry point contract is a mechanism which allows cryptowallet accounts to function as smart contracts. Through account abstraction, the functionality of the user cryptowallet is enhanced.

[0073] The entry point contract 170 orchestrates the processing of user operations onchain, enabling smart contracts to operate as user wallet accounts on Ethereum according to the ERC-4337 standard. This may allow for additional actions

to be performed through user wallet accounts, such as setting up automated payments. As another example, an entry point contract allows Ethereum user wallet accounts to operate as smart contracts according to the ERC-4337 standard.

[0074] The second smart contract 180 can be a smart contract account. For instance, a smart contract account can be a smart contract that can be used to interact with other smart contracts on behalf of the owner of the smart contract account. It can enable functionality beyond that which is available through a cryptowallet or other externally owned account alone.

[0075] Blocks 191, 192, 193 may be blocks within an exemplary blockchain. For instance, blocks 191, 192, and 193 may be sequential blocks in an Ethereum blockchain, or any other suitable type of blockchain.

[0076] As explained with respect to FIG. 2A, a series of example steps may be undertaken to allow for an external credit account or payment account to be used to pay for gas fees through fiat currency.

[0077] In step 1, a communication device comprising the interaction application 110 can transmit to the master computer 120, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction. For example, a user may initiate a request at the interaction application 110 (e.g., a cryptowallet) on a user communication device in step 1. In this step, a request may be initiated to pay a gas fee with a payment card. The interaction application 110 can be programmed to determine the gas fee in fiat currency based on the particular operation requested by the user of the user communication device. At step 1, the card access data (e.g., a credential such as a PAN or a token) associated with the payment card, and payment information, such as the fiat equivalent of the gas fee, may be sent to a paymaster service. In some examples, the equivalent may be set based on current fiat to cryptocurrency exchange rates. This information may be sent to the master computer 120 (e.g., a paymaster service computer).

[0078] At step 2, the master computer 120 can initiate an authorization request message with the gateway computer 130 in communication with the authorizing entity computer 134. For example, the master computer 120, after receipt of the information in step 1, may obtain authorization for the gas fee payment by

communicating with the gateway computer 130. The master computer 120 can initiate (e.g., generate) an authorization request message comprising an operational value such as the amount of the gas fee in fiat currency and can send it to a gateway computer 130. The gateway computer 130 can forward the authorization request message to the processing network computer 132. If the authorization request comprises a token, then the processing network computer 132 can de-tokenize the token to obtain the credential (e.g., a PAN or primary account number) associated with the token. The processing network 132 can then transmit the authorization request message with the credential to the authorizing entity computer 134 for authorization. The authorizing entity computer 134 can then send an authorization response message back to the gateway computer 130 via the processing network computer 132. The gateway computer 130 can provide an indication of the authorization to the master computer 120.

[0079] At the end of the day or some other period of time, the authorizing entity computer 134 can clear and settle the transaction with the gateway computer, which may manage a fiat currency account associated with the paymaster service operating the master computer 120.

[0080] At step 3, the master computer 120, may receive confirmation of authorization from the gateway computer 130. This information and/or information about the approval of the gas fee payment may be sent by the master computer 120 to interaction application 110. For example, the information about the gas fee payment including the amount of the gas fee in fiat currency, an address (e.g., a public key) and/or account number of the master computer 120, an authorization code from the authorizing entity computer 134, a date and time, etc. This information and optionally other information about the user's account may then be digitally signed with a master computer private key stored at the master computer 120 to product a digital signature. The digital signature may later be verified on-chain by the first smart contract 140 using a corresponding public key corresponding to the master computer private key. The master computer 120 can transmit an off-chain transaction response message comprising (e.g., in addition to the data mentioned above) the digital signature (e.g., the master computer digital signature) indicating approval of the off-chain transaction, a signature validity time window, and an on-

chain paymaster contract address to the interaction application 110 on the user communication device.

[0081] At step 4, the communication device comprising the interaction application 110 can transmit to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction. For example, in step 4, a blockchain transaction request for a blockchain transaction may be submitted by a user through his or her cryptowallet to the blockchain network in the blockchain environment. The transaction parameters may include information sufficient to process the transaction on the blockchain managed by the blockchain network and the master computer digital signature obtained in step 3. Such information may include the user's address (e.g., the user's public key), the recipient's address (the recipient's public key), the protocol used to conduct the transaction, and the value to transfer from the user to the recipient, and an address (e.g., a public key) of the first smart contract 140. The value may be a blockchain transaction value, which is different than the operational value (e.g., gas fee) that is described above. The transaction information may also include a specific service (e.g., a specific paymaster) which was selected at the time the transaction was submitted. The specific service may be identified with an address (e.g., public key) of the first smart contract and/or the master computer 120. A transaction digital signature of this information (signed by a communication device private key) may also be included in the blockchain request as proof that the user approved of the transaction.

[0082] At step 5, the transaction may be received by mempool 150.

[0083] At step 6, the transaction may be provided from the mempool 150 to bundler/node 160 for processing. Upon processing, the transaction be submitted to entry point contract 170 with other transaction in a bundle.

[0084] The node in the blockchain network then requests verification of the off-chain transaction using the first smart contract 140 associated with the master computer, verifies the transaction parameters with a second smart contract 180, and then executes the blockchain transaction.

[0085] For example, at step 7, the entry point contract can interact with second smart contract 180 to validate the smart contract account and ensure that the

requested operation is correct with respect to the selected service (e.g., selected paymaster) and the user's account. In some examples, second smart contract 180 can be a program corresponding to an account associated with the same user as the interaction application 110. This information can be returned to the entry point contract 170. For example, the second smart contract 180 can determine if the user of the interaction application 110 has sufficient funds to conduct a requested cryptocurrency transaction. The second smart contract 130 and/or the entry point contract 170 can also view the blockchain with the blocks 191, 192, 193 to determine a current state of the user's account. The entry point contract 170 may also include logic which acknowledges that the gas fee was already paid in fiat currency, and need to account for the gas fee when checking on the state of the user's account with the second smart contract 180. The logic may recognize that a digital signature from the master computer 120 was included in the transaction information.

[0086] At step 8, entry point contract 170 can ensure that the service associated with the first smart contract 140 and the master computer 120 is valid, that the service's (e.g., paymaster's) balance of cryptocurrency is sufficient to pay for the gas fee, and that the stake of the service is acceptable.

[0087] At step 9, entry point contract 170 request that the first smart contract 140 validate the digital signature of the off-chain transaction details, which was previously generated by the master computer 120 using the master computer private key. The first smart contract 140 can validate the digital signature produced by the master computer 120 using a public key of the master computer 120. The first smart contract 140 can also check an expiration window for verifying the digital signature.

[0088] At step 10, after receiving confirmation from the first smart contract 140 that the gas fee has been paid by the user, the entry point contract 170 can execute the user function or user transaction requested by interaction application 110 and write the results of the user operation to second smart contract 180. Additionally, at a final step, the user operation requested can be written by the appropriate nodes in the blockchain network to a next block on the blockchain such as for example, block 193. This can ensure that the operation is provided to the blockchain and recorded.

[0089] FIG. 2B illustrates an additional example framework 101 to allow for an off-chain payment of gas fees. Framework 101 can be used to enable a "prepaid"

scenario wherein fiat currency is prepaid to a paymaster in exchange of receipt and processing of gas fees. In general overview, in framework 101, the first smart contract 140 (on-chain) can keep track of the "balance" of each user. To increase their balance, users make a card payment to the off-chain master computer 120. After receiving the payment, the off-chain web service calls the on-chain first smart contract 140 and updates the user's balance, increasing it by the payment amount. When user sends a transaction, such as from interaction application 110, as part of the transaction processing flow on chain, the first smart contract 140 is called. The first smart contract 140 then enables the coverage of gas fees and deducts the fee amount from the user's balance contained on the first smart contract.

[0090] In general overview, framework 101 may be similar to framework 100 described above. However, in some examples, the process may differ than framework 100 in step 3'. For example, rather than providing a response to interaction application 110, at done in step 3 of framework 100, in step 3' of framework 101, master computer 120, after confirmation from the payment gateway information, may update a user's balance based on the confirmed payment from the gateway computer 130. This information may be sent by first smart contract 140. It may be noted that the master computer 120 is sending information from an off-chain component to an on-chain component at step 3'.

[0091] Additionally, in framework 101, at step 11, following execution of the user operation in step 10, the balance of cryptocurrency following the spend of gas fees or transaction fees can be recorded by the first smart contract 140. Additionally, at a final step, the user operation requested can be written to a next block on the blockchain such as for example, block 193. This can ensure that the operation is provided to the blockchain and recorded.

[0092] FIG. 2C illustrates an additional example framework 103 to allow for an off-chain payment of gas fees. Framework 103 can be thought of as an on-credit or post-spend payment framework. In framework 103, a user provides card information to off-chain master computer 120, which stores the information securely thereon. Responsive to a user sending a transaction through interaction application 110, as part of the transaction processing flow on chain, first smart contract 140 is called, in which turn the contract covers the gas fees. At a periodic basis, (e.g., weekly,

monthly), the off-chain master computer 120 charges the user based on the amount covered by the paymaster smart contract over the specified period.

[0093] In general overview, framework 103 may be similar to framework 101 described above. In some examples, rather than sending a payment in step 3 or 3', master computer 120 may additionally send information to a paymaster smart contract "whitelisting" the user, enabling additional spend by the user for a period basis. Similarly, step 11" may differ from step 11 in that a periodic user charge may occur based on information provided by first smart contract 140.

[0094] FIG. 3 illustrates a diagram of a communication device 300 according to an embodiment. communication device 300 may include device hardware 304 coupled to a system memory 302.

[0095] Device hardware 304 may include a processor 306, a short range antenna 314, a long range antenna 316, input elements 310, a user interface 308, and output elements 312 (which may be part of the user interface 308). Examples of input elements may include microphones, keypads, touchscreens, sensors, etc. Examples of output elements may include speakers, display screens, and tactile devices. The processor 306 can be implemented as one or more integrated circuits (e.g., one or more single core or multicore microprocessors and/or microcontrollers), and is used to control the operation of communication device 300. The processor 306 can execute a variety of programs in response to program code or computer-readable code stored in the system memory 302, and can maintain multiple concurrently executing programs or processes.

[0096] The long range antenna 316 may include one or more RF transceivers and/or connectors that can be used by the communication device 300 to communicate with other devices and/or to connect with external networks. The long range antenna 316 may be configured to communicate with a remote base station and a remote cellular or data network, over the air. The short range antenna 309 may be configured to communicate with external entities through a short range communication medium. The short range antenna 309 may comprise a contactless interface that can interact with a contactless interface of another device (e.g., a portable device). Examples of a contactless interface may include one or more radio frequency (RF) transceivers that can send and receive communications using near-

field communications (NFC), or other radio frequency or wireless communication protocols. The user interface 308 can include any combination of input and output elements to allow a user to interact with and invoke the functionalities of the communication device 300.

[0097] The system memory 302 can be implemented using any combination of any number of non-volatile memories (e.g., flash memory) and volatile memories (e.g., DRAM, SRAM), or any other non-transitory storage medium, or a combination thereof media. The system memory 302 may store computer code, executable by the processor 306, for performing any of the functions described herein. For example, the system memory 302 may comprise a computer readable medium comprising code, executable by the processor 306, for implementing a method comprising: transmitting, by a communication device to a master computer, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction, wherein the master computer initiates an authorization request message with a gateway computer in communication with an authorizing entity computer; and transmitting, by the communication device to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein node requests verification of the off-chain transaction using a first smart contract associated with the master computer, verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0098] The system memory 302 may also store an interaction application 302A, an authentication module 302B, cryptography module 302C, and an operating system 302D. The interaction application 302A may include instructions or code executable by the processor 306 for initiating and conducting an interaction such as those described above. . The authentication module 302B may comprise code, executable by the processor 306, to authenticate a user. This can be performed using user secrets (e.g., passwords) or user biometrics. The cryptography module may comprise cryptographic processing logic such as encryption, decryption, and signing algorithms. It may also store public and private keys associated with the user of the communication device.

[0099] FIG. 4 shows a block diagram of a node computer 400. The node computer 400 may include a processor 402 and a computer readable medium 404, a data storage 406 which can store a blockchain 406A as well as a mempool of transactions, and a network interface 408 coupled to the processor 402.

[0100] The computer readable medium 404 may comprise a communication module 404A, a blockchain processing module 404B, and a smart contract interaction module 404C.

[0101] The communication module 404A can comprise code, executable by the processor 402 to cause the processor 402 to communicate with external entities such as the previously described master computer and communication device.

[0102] The blockchain processing module 404B may comprise code that causes the processor 402 to interaction and process transactions for including on a blockchain as described above.

[0103] The smart contract interaction module 404C may comprise code that causes the processor 306 to perform smart contract interactions such as those described above.

[0104] FIG. 5 shows a block diagram of a master computer 500 according to an embodiment. The master computer 500 comprises a processor 502 and a computer readable medium 504, a data storage 506, and a network interface 308 coupled to the processor 306.

[0105] The computer readable medium 504 may comprise a communication module 504A, which may have similar functions as communication module 404A above, a cryptography module 504B which may have similar functions as cryptography module 302C above, and an authorization processing module 504C. The authorization processing module 504C may comprise code, executable by the processor 502 to initiate (e.g., generate) authorization request messages, transmit authorization request messages, receive authorization response messages, etc.

[0106] The computer readable medium 504 may comprise code executable by the processor 502 for performing operations comprising: receiving, by the master computer from a communication device, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain

transaction; initiating, by the master computer, an authorization request message with a gateway computer in communication with an authorizing entity computer; receiving, by the master computer, an authorization response message from the authorizing entity computer; and transmitting, by the master computer, an off-chain transaction response to the communication device, wherein the communication device transmits, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein node requests verification the off-chain transaction using a first smart contract and verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

[0107] FIG. 6 shows a diagram of a blockchain according to embodiments of the present invention. FIG. 6 includes a blockchain 600 comprising a first block 610 and a second block 640. The blockchain 600 can include any suitable number of blocks (e.g., 10, 500, 2000, 500000, etc.).

[0108] Current blockchain technologies, such as Bitcoin and Ethereum, maintain an append-only ledger in a network. The ledger includes a list of blocks of transaction data, the blocks are cryptographically chained together. A block is created by a computationally intensive process called proof-of-work in which valid blocks need to demonstrate a sufficient “difficulty" (i.e., sufficient computation power to create on average). If there are more than one available chains of blocks, then network participants (i.e., nodes) can download all blocks in all chains and follow the chain which has the highest total difficulty. This mechanism guarantees that, eventually, the network will agree on a single and valid chain, see [Garay et al, The Bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310, 2015], [Bitcoin Website. bitcoin.org], and [Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673, Cham, 2017. Springer International Publishing.], which are all incorporated herein by reference for all purposes.

[0109] The blockchain 600 can create a history of data deposits, messages, or entries in a series of blocks where each block contains a mathematical summary,

called a hash, of the previous block. This creates a chain where any changes made to a block will change that block's hash, which must be recomputed and stored in the next block. This changes the hash of the next block, which must also be recomputed and so on until the end of the chain.

[0110] Although the hash can be simple to compute, rules may be imposed, which require the value of the hash to be below a certain threshold value (i.e., a difficulty value). In addition, the hash is based on a type of mathematical function that is not reversible. One cannot predict what input can be used to produce the desired output. A valid hash is found by repeatedly adjusting a changeable value in the block, and recalculating the hash until it meets the validity requirements. The freely changeable value can be a nonce. The unpredictable nature of the hash considerably increases the difficulty of finding a nonce that produces a valid hash of the block.

[0111] As an example, the first block 610 can include a block header 620 and block entries 630. The block header 620 of the first block 610 can comprise a previous hash 612, a timestamp 614, a Merkle root 616, and a nonce 618.

[0112] The previous hash 612 can be a hash of the previous block's header. The previous hash 612 can be the result of a non-reversible mathematical computation using data from the previous block as the input. According to some embodiments, the computation used can include a SHA256 hash function. One of ordinary skill in the art would recognize that any suitable hash function could be used without departing from the spirit and scope of the present invention. The hash function can be designed so that any change to the data in the previous block results in an unpredictable change in the hash of that block. The previous hash 612 can be a link between blocks, chaining them together to form the blockchain 600.

[0113] When calculating the previous hash 612 for the previous block, a node can determine if the previous hash 612 can meet certain criteria defined by a difficulty value. In some embodiments, the difficulty value may include a number that the calculated hash must be less than. However, because the output of the hashing function is unpredictable, the output cannot be determined what input will result in an output that is less than the difficulty value before the hash is calculated. The nonce 618 can be used to vary the data content of the block, allowing for a large

number of different outputs to be produced by the hash function in pursuit of an output that meets the difficulty value. This makes can make it computationally expensive to produce a valid block with a nonce 618 that produces a hash value meeting the criteria of the difficulty value.

[0114] The hash algorithms used for the previous hash 612 can include MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA-3 or any suitable hash function. There is also no requirement that a hash be computed only once. The results of a hash function may be reused as inputs into another or the same hash function again multiple times in order to produce a final result. One of ordinary skill in the art would recognize that any hash function could be used to compute the required hashing without departing from the spirit and scope of the present invention.

[0115] The Merkle root 616 can be a root of a Merkle tree, which can include a tree in which every leaf node is labelled with the hash of a data block, for example an entry. Each leaf of the Merkle tree can represent one of the entries. Each entry can be hashed together with a sibling node (i.e., entry) in the Merkle tree. Successively hashing sibling nodes in the Merkle tree can result in the Merkle root 616.

[0116] The block entries 630 can include interaction data and/or smart contracts. The block entries 630 can include any suitable number of entries. Example entries can include the data described above, such as witnesses, signed witnesses, digital signatures, interaction messages, unconditional payment data, proof of inclusions, other data related to the processing of an offline interaction between two user devices, etc. In some embodiments, an entry may be a null value, for example, in the case of the Merkle tree being a sparse Merkle tree.

[0117] In some embodiments, the number of entries in the block entries 630 may be limited by the overall size of the block (e.g., the first block 610). For example, the blocks on the blockchain may be limited by a certain amount of data (e.g., 1/2 MB, 1 MB, 2 MB, 5 MB, 10 MB, etc.). In other embodiments, the number of entries in the block entries 630 may be a predetermined number of entries. For example, the nodes in the verification network can determine that 1, 10, 150, 500, 1000, etc. entries can be included in the block entries 630. The entries can be transactions, user operations, smart contracts, etc.

[0118] The timestamp 614 can include a time that the block was created within a certain range of error. According to some embodiments of the present invention, the full nodes of the verification network can check the timestamp 614 against their own known time and can reject any block that seems to have an erroneous timestamp 614.

[0119] The nonce 618 can be a value adjusted by a full node while performing a proof-of-work process, as described herein. A nonce can be input into a hash function along with block data to determine the output hash value. A correct nonce (also referred to as a golden nonce) yields an output hash value that satisfies a predetermined criteria, such as being less than a difficulty value.

[0120] The second block 640 can be similar to the first block 610. For example, the second block 640 can include a block header 650 and block entries 660. The block header 650 of the second block 640 can comprise a previous hash 652, a timestamp 654, a Merkle root 656, and a nonce 658. The block entries 660 can include image data, interaction data, smart contracts, etc., and can be similar to the block entries 630.

[0121] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C, C++, C#, Objective-C, Swift, or scripting language such as Perl or Python using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer readable medium for storage and/or transmission, suitable media include random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a compact disk (CD) or DVD (digital versatile disk), flash memory, and the like. The computer readable medium may be any combination of such storage or transmission devices.

[0122] Such programs may also be encoded and transmitted using carrier signals adapted for transmission via wired, optical, and/or wireless networks conforming to a variety of protocols, including the Internet. As such, a computer readable medium according to an embodiment of the present invention may be created using a data signal encoded with such programs. Computer readable media

encoded with the program code may be packaged with a compatible device or provided separately from other devices (e.g., via Internet download). Any such computer readable medium may reside on or within a single computer product (e.g., a hard drive, a CD, or an entire computer system), and may be present on or within different computer products within a system or network. A computer system may include a monitor, printer, or other suitable display for providing any of the results mentioned herein to a user.

[0123] The above description is illustrative and is not restrictive. Many variations of the invention will become apparent to those skilled in the art upon review of the disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the pending claims along with their full scope or equivalents.

[0124] One or more features from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the invention.

[0125] As used herein, the use of "a," "an," or "the" is intended to mean "at least one," unless specifically indicated to the contrary.

WHAT IS CLAIMED IS:

1. A method comprising:

transmitting, by a communication device to a master computer, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction, wherein the master computer initiates an authorization request message with a gateway computer in communication with an authorizing entity computer; and

transmitting, by the communication device to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein the node requests verification of the off-chain transaction using a first smart contract associated with the master computer, verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

2. The method of claim 1, further comprising:

receiving, by the communication device, an off-chain transaction response from the master computer, the off-chain transaction response comprising a master computer digital signature, the master computer digital signature indicating approval of the authorization request message, and

wherein the blockchain transaction request comprises the master computer digital signature.

3. The method of claim 2, wherein the authorization request message comprises an operational value associated with the blockchain transaction request, and wherein the transaction parameters comprise a blockchain transaction value that is different than the operational value.

4. The method of claim 3, wherein the master computer stores a master computer private key of a master computer public-private key pair, and produces the master computer digital signature with the master computer private key.

5. The method of claim 4, wherein the credential or the token is the token.

6. The method of claim 5, wherein the transaction parameters comprise a public key associated with the communication device, an address of a sender associated with the communication device, an address of a receiver of the blockchain transaction value, the blockchain transaction value, the master computer digital signature, and a first smart contract address associated with the first smart contract.

7. The method of claim 6, wherein the node provides a signature validation request comprising the master computer digital signature to the first smart contract using the first smart contract address, wherein the first contract validates the master computer digital signature using a master computer public key, receives a signature validation response from the first smart contract, and executes the blockchain transaction by initiating a write operation including the transaction parameters of the blockchain transaction to the blockchain.

8. A communication device comprising:

a processor; and

a non-transitory computer readable medium, the non-transitory computer readable medium comprising code, executable by the processor for performing operations including:

transmitting, to a master computer, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction, wherein the master computer initiates an authorization request message with a gateway computer in communication with an authorizing entity computer; and

transmitting, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein the node requests verification the off-chain transaction using a first smart contract associated with the master computer, verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

9. The communication device of claim 8, wherein the operations further comprise:

receiving an off-chain transaction response from the master computer, the off-chain transaction response comprising a master computer digital signature, the master computer digital signature indicating approval of the authorization request message, and

wherein the blockchain transaction request comprises the master computer digital signature.

10. The communication device of claim 8, wherein the authorization request message comprises an operational value associated with the blockchain transaction request, and wherein the transaction parameters comprise a blockchain transaction value that is different than the operational value.

11. The communication device of claim 8, wherein the communication device stores a communication device private key for digitally signing the blockchain transaction request.

12. A method comprising:

receiving, by a master computer from a communication device, an off-chain transaction request comprising a credential or a token for an off-chain transaction in association with a blockchain transaction;

initiating, by the master computer, an authorization request message with a gateway computer in communication with an authorizing entity computer;

receiving, by the master computer, an authorization response message from the authorizing entity computer; and

transmitting, by the master computer, an off-chain transaction response to the communication device, wherein the communication device transmits, to a node in a blockchain network managing a blockchain, a blockchain transaction request comprising transaction parameters associated with the blockchain transaction, wherein node requests verification the off-chain transaction using a first smart contract and verifies the transaction parameters with a second smart contract, and then executes the blockchain transaction.

13. The method of claim 12, wherein the off-chain transaction response comprises a first smart contract address associated with the first smart contract and

a master computer digital signature, the master computer digital signature indicating approval of the authorization request message.

14. The method of claim 13, wherein the master computer digital signature is in the blockchain transaction request.

15. The method of claim 13, wherein the authorization request message comprises an operational value associated with the blockchain transaction request, and wherein the transaction parameters comprise a blockchain transaction value that is different than the operational value.

16. The method of claim 13, wherein the transaction parameters comprise a public key associated with the communication device, an address of a receiver of a blockchain transaction value, the blockchain transaction value, the master computer digital signature, and the first smart contract address associated with the first smart contract.

17. The method of claim 13, wherein the node provides a signature validation request comprising the master computer digital signature to the first smart contract using the first smart contract address, wherein the first contract validates the master computer digital signature using a master computer public key, receives a signature validation response from the first smart contract, and executes the blockchain transaction by initiating a write operation including the transaction parameters of the blockchain transaction to the blockchain.

18. The method of claim 12, wherein the off-chain transaction response does not include a master computer digital signature.

19. The method of claim 12, wherein the master computer receives the credential.

20. The method of claim 12, wherein the off-chain transaction response comprises a first smart contract address associated with the first smart contract and a master computer digital signature, the master computer digital signature indicating approval of the authorization request message, and wherein the master computer

stores a master computer private key of a master computer public-private key pair, and produces the master computer digital signature with the master computer private key.