

PLATFORM CONTROLLED WALLETS IN BLOCKCHAIN SYSTEMS**CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] This application claims benefit of and priority to U.S. Provisional Patent Application Serial No. 63/516,993, entitled “Platform Controlled Wallets in Blockchain Systems,” filed August 1, 2023, and assigned to the assignee hereof, the entire contents of which are hereby incorporated by reference

INTRODUCTION

[0002] Aspects of the present disclosure relate to security in blockchain systems, and more specifically to wallet security in blockchain systems.

BACKGROUND

[0003] Blockchains can be used in various decentralized systems to provide a ledger of transactions that have occurred within these decentralized systems. Generally, a blockchain may include a chain of blocks, in which latest block includes some information about a transaction that occurred and a reference to an immediate predecessor block, which may be a hashed value of the previous block. Because the reference to the immediate predecessor block may be a value derived from the immediate predecessor block, verification of the transactions in the blockchain may be performed by ensuring that a hash of a block resolves to the same value as that stored as a reference to the immediate predecessor block in a succeeding block in the blockchain. If there is a mismatch between a computed hash of a block and the hashed value of the block in a succeeding block in the blockchain, validation of the blockchain may fail.

[0004] To perform transactions on a blockchain system, wallets are generally established for users in order to hold and transfer assets on the blockchain. Generally, wallets on the blockchain are defined based on a private key and a corresponding public key. The public key allows for other users to transfer assets to the wallet. Meanwhile, the private key allows the owner of the wallet to access the contents of the wallet and initiate transactions on the blockchain (e.g., to transfer assets to another wallet, to withdraw assets from the wallet to another blockchain or an external source, etc.).

[0005] These wallets may be established for different by a centralized platform through which these users perform transactions on the blockchain. For example, these centralized platforms may facilitate the transfer of assets between different users of the platform. However, because possession of a private key for a wallet allows for assets stored in the wallet to be

withdrawn or otherwise transferred, and because sharing private keys associated with a wallet may expose the wallet to malicious activity, the centralized platform may maintain the wallets on behalf of the users of the centralized platform. The users themselves may not have direct access to the wallets (as the user may not have access to the private keys defining these wallets), but instead may access their wallets through user credentials with the centralized platform.

[0006] Generally, preventing malicious actions from being taken in respect of a wallet involves maintaining secrecy around the private key. To do so, some applications link a private key to a series of words, or a recovery phrase, that a user can use to recover a private key associated with a wallet. These recovery phrases generally include a significant number of words. Because of the length of these recovery phrases, users often copy these phrases and store the copy of these phrases in insecure locations which can be easily accessed. In turn, because recovery phrases can often be easily accessed, the security of a wallet and the contents therein may be compromised.

[0007] Accordingly, techniques are needed to allow for secure maintenance of wallets in blockchain systems.

BRIEF SUMMARY

[0008] Certain embodiments provide a computer-implemented method for securely accessing a wallet maintained by a centralized platform on a blockchain. An example method generally includes receiving a request to access one or more wallets on a blockchain. Generally, the request includes an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party. A first portion of a private key is decrypted based on the authorization code and a salt associated with the user credentials associated with the controlling party, and a second portion of the private key is decrypted based on credentials associated with an application through which the wallet is accessed. Access to the one or more wallets is granted based on the decrypted first portion and the decrypted second portion of the private key.

[0009] Other embodiments provide processing systems configured to perform the aforementioned methods as well as those described herein; non-transitory, computer-readable media comprising instructions that, when executed by one or more processors of a processing system, cause the processing system to perform the aforementioned methods as well as those described herein; a computer program product embodied on a computer readable storage medium comprising code for performing the aforementioned methods as well as those further

described herein; and a processing system comprising means for performing the aforementioned methods as well as those further described herein.

[0010] The following description and the related drawings set forth in detail certain illustrative features of one or more embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The appended figures depict certain aspects of the one or more embodiments and are therefore not to be considered limiting of the scope of this disclosure.

[0012] FIGs. 1A and 1B depict example computing environments in which a private key defining user wallets managed by a controlling party on a blockchain is secured using an authorization code, user credentials associated with the controlling party, and platform credentials, according to aspects of the present disclosure.

[0013] FIG. 2 is a message flow diagram illustrating messages exchanged between a key management system and a blockchain to establish a wallet managed by a controlling party based on an authorization code and user credentials associated with the controlling party, according to aspects of the present disclosure.

[0014] FIG. 3 is a message flow diagram illustrating messages exchanged between a key management system and a blockchain to perform transactions on the blockchain using assets in a wallet managed by a controlling party based on an authorization code and user credentials associated with the controlling party, according to aspects of the present disclosure.

[0015] FIG. 4 is a message flow diagram illustrating messages exchanged between a key management system and a blockchain to perform transactions on the blockchain using assets in a wallet managed by a controlling party based on client-side verification of a transaction to be performed on the blockchain, according to aspects of the present disclosure.

[0016] FIG. 5 illustrates example operations for accessing a wallet managed by a controlling party on a blockchain based on an authorization code and user credentials associated with the controlling party, according to aspects of the present disclosure.

[0017] FIG. 6 illustrates example operations for accessing a wallet managed by a controlling party on a blockchain based on an authorization code, user credentials associated with the controlling party, and client-side transaction verification, according to aspects of the present disclosure.

[0018] FIG. 7 illustrates an example system on which embodiments of the present disclosure can be performed.

[0019] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the drawings. It is contemplated that elements and features of one embodiment may be beneficially incorporated in other embodiments without further recitation.

DETAILED DESCRIPTION

[0020] Transactions in cryptocurrency systems may be represented as blocks in a blockchain that track a universe of transactions performed using the cryptocurrency system. In these cryptocurrency systems, processed transactions may not be modified at a later date, thus providing an immutable ledger of the transactions performed using the cryptocurrency system.

[0021] Wallets, defined by a private key and public key pair, generally hold assets, such as native or non-native tokens, that can be transferred on a blockchain. In order to perform a transaction on a blockchain, a user (or a controlling party, such as a centralized platform, which hosts wallets for the benefit of users of the centralized platform) generally signs a transaction using a private key associated with a wallet from which assets are being transferred on the blockchain. That is, a transaction generally identifies the assets to be transferred and the public address (or public key) of the transferee wallet to which these assets are transferred, and the transaction is signed using the private key associated with the transferor wallet. Decrypting the signature of the transaction, using the public key associated with the transferor wallet, allows for the transaction to be verified and thus for assets to be transferred from the transferor wallet to the transferee wallet.

[0022] Without access to a private key, thus, users are generally unable to perform transactions on a blockchain. Because access to a private key associated with a wallet grants access to the assets contained in a wallet and allows transactions to be performed with respect to the assets in the wallet, the privacy and security of the private key should be maintained. Various techniques are generally used to preserve access to the private key associated with a user's wallet. For example, users may maintain a record of the private key associated with their wallet in a separate (digital or physical) repository. In another example, because private key addresses are generally long alphanumeric strings which are difficult to remember, various key derivation techniques can be used to aid in the retrieval of the private key associated with the wallet. For example, a multi-word passphrase can be generated and used as an input into a key

derivation function in order to generate (or retrieve) the private key associated with a wallet. However, while a passphrase may be less complicated to remember than the private key itself, the passphrase generally includes a sufficient number of words such that users store the passphrase in a separate (digital or physical) repository. Thus, users still maintain a record of the private key associated with their wallet in potentially insecure locations, thus exposing the private key of their wallets and potentially allowing malicious actors access to their wallets and the assets contained therein. Maintaining records of private keys associated with a controlling party for multiple accounts may magnify the impact of malicious activity, as a compromise of a private key used by the controlling party to manage multiple accounts may allow a malicious party to exfiltrate assets from many wallets.

[0023] Aspects of the present disclosure provide techniques for improving the security of and ease of access to wallets maintained by a controlling party on behalf of users of a centralized platform used to conduct transactions on a blockchain by using an intermediate layer to access a private key defining a wallet and therefore introducing an abstraction layer to techniques for generating private keys defining the wallet. As used herein, a “platform” or “centralized platform” may include, without limitation, an application or service controlled by a developer of such an application or service and made available for use by one or more third parties. Generally, a private key can be sharded, or split, into multiple portions, with some portions being encrypted using credentials associated with the controlling party and other portions being encrypted using credentials associated with an application or platform on which the wallet is hosted. By doing so, multiple user-facing techniques can be exposed to allow a user to decrypt the portions of the private key encrypted using credentials associated with the controlling party, while not exposing techniques by which the private key itself can be directly recovered (e.g., by not exposing a passphrase or other private key recovery mechanism by which the entirety of the private key associated with a wallet can be recovered). By doing so, aspects of the present disclosure may allow users to easily access wallets while maintaining the security of these wallets, as a private key defining a wallet may not be derived directly from data known to a user of the application or platform hosting the wallet.

Example Generation of and Access to Wallets in Blockchain Systems Using Sharded Private Keys

[0024] FIG. 1A illustrates an example computing environment 100A in which a private key defining a user wallet on a blockchain is secured using an authorization code, user

credentials, and platform credentials. As illustrated, computing environment 100A includes a key management system 110 and a network 120.

[0025] Key management system 110 generally is representative of any system on which a private key defining wallets managed by a controlling party on behalf of users of a centralized platform can be generated and accessed, such as a mobile device (e.g., smartphone, tablet, etc.), server computer hosting a service accessible by one or more remote systems, a cloud computing instance, or the like. As illustrated, key management system 110 includes a wallet generator 112, key entropy pool 114, wallet key decryptor 116, and a transaction signer 118.

[0026] As discussed, a wallet is generally defined in terms of a pairing of a public key and a private key. In order to perform a transaction on a blockchain, the controlling party maintaining a centralized system through which users execute transactions on the blockchain 122 generally signs a transaction using a private key associated with a wallet from which assets are being transferred on the blockchain. That is, a transaction generally identifies the assets to be transferred and the public address (or public key) of the transferee wallet to which these assets are transferred, and the transaction is signed using the private key associated with the transferor wallet (which, in a centralized system, is the private key associated with the controlling party that maintains the centralized system). Decrypting the signature of the transaction, using the public key associated with the transferor wallet, allows for the transaction to be verified and thus for assets to be transferred from the transferor wallet to the transferee wallet.

[0027] Wallet generator 112 generally receives requests to generate a wallet in which a controlling party associated with the wallet can, on behalf of the ultimate owner of the wallet, store digital assets and initiate the transfer of digital assets to other users on blockchain 122. Generally, a request to generate a wallet may include an authorization code selected by the controlling party for use in accessing the wallet at a future point in time and user credentials associated with the controlling party's account on a platform provided by the key management system 110. The authorization code may include multiple alternatively usable mechanisms by which the controlling party can be authenticated. Generally, these mechanisms may be information which the controlling party (or its delegates) knows, which the controlling party is (e.g., data derived from user biometrics), or other information which the controlling party need not commit to a potentially insecure repository for retrieval when the controlling party wishes to recover the private key defining a wallet. For example, the authorization code may include an alphanumeric personal identification number (PIN) code. As an alternative to this

alphanumeric PIN code, the authorization code may also include a series of questions and answers. The questions may be selected from an *a priori* defined list of questions or may be generated by the controlling party when executing a wallet creation process on behalf of a user of a centralized platform, through key management system 110.

[0028] Based on the received request to generate the wallet for the ultimate owner of the wallet, wallet generator 112 requests and receives a private key and public key pairing from the key entropy pool 114. Generally, the key entropy pool 114 includes a plurality of private keys which can be assigned to users of key management system 110 who wish to establish a wallet for use in performing transactions on the blockchain 122. Keys included in the key entropy pool 114 generally are keys in which various elements of random data are stored and used as seeds to generate the private keys included in the key entropy pool (and the associated public keys). Because a key generator used to populate key entropy pool 114 uses random data that differs each time a key is generated, each key in the key entropy pool 114 may be unique and may not be derivable based on other keys in the key entropy pool 114.

[0029] To protect the private key selected from the key entropy pool 114, wallet generator 112 can split, or shard, the private key into multiple portions (or shards). These portions can be independently encrypted such that a compromise of the keys involved in encrypting one portion of the private key does not expose the entire private key, and thus the wallet associated with the private key, to a malicious party. For example, the private key can be sharded into a first portion and a second portion, with the first portion being encrypted based on the authorization code generated by the controlling party of a centralized platform that is creating a wallet for an ultimate owner of that wallet using the key management system 110 and the second portion being encrypted based on credentials associated with an application through which the wallet is accessed. The first portion of the private key may be referred to herein as the “controlling party portion” of the private key, and the second portion of the private key may be referred to herein as the “platform portion” of the private key.

[0030] In some aspects, multiple encrypted versions of the controlling party portion of the private key may be generated to allow for the controlling party portion to be recovered. For example, a first encrypted version of the controlling party portion of the private key may be generated based, at least in part, on an alphanumeric authorization code and a second encrypted version of the controlling party portion of the private key may be generated based on a series of question-answer pairs which the controlling party can use as a “backup” mechanism for accessing the wallet associated with the private key.

[0031] To generate the first encrypted version of the controlling party portion of the private key, the alphanumeric authorization code can be augmented with a cryptographic salt, or random data which is appended to the alphanumeric authorization code in order to generate an encryption key used to protect the controlling party portion of the private key. The cryptographic salt may be bound to a credential associated with the controlling party and may be further protected by a cryptographic key (or set of cryptographic keys) maintained by the key management system 110. By protecting this cryptographic salt using another set of cryptographic keys, the key management system 110 may ensure that malicious parties are unable to recover the cryptographic salt and thus are unable to recover the encryption key used to protect the controlling party portion of the private key, as it is generally computationally impossible to execute a brute force attack against a sufficiently large search space. For example, assuming that the controlling party portion of the private key is encrypted using a 128-bit key, a brute force attack generally involves $2^{128} = 3.4 * 10^{38}$ decryption attempts, which would take longer than the age of the universe to execute. The first encrypted version of the controlling party portion of the private key may be generated using symmetric key cryptography, in which the same key is used to encrypt and decrypt a payload. In this example, the encryption/decryption key used to encrypt the first encrypted version of the controlling party portion of the private key may be generated by concatenating the alphanumeric authorization code and the cryptographic salt into a string representing the encryption/decryption key.

[0032] To generate the second encrypted version of the controlling party portion of the private key, the series of question-answer pairs generated by the controlling party may be converted into an encryption/decryption key using various key derivation functions. The derived key may then be used to encrypt the controlling party portion of the private key, which may be stored at key management system 110, and decrypt the controlling party portion of the private key upon request by the controlling party (e.g., as discussed in further detail below, when the controlling party executes a transaction on the blockchain 122 for and on behalf of the ultimate owner of a wallet).

[0033] In some aspects, to protect the platform portion of the private key, the platform portion of the private key may be further sharded into different sub-portions. Each of these sub-portions may be independently encrypted using different keys so that a compromise of one set of cryptographic keys maintained by the key management system 110, or a platform outage by a portion of the key management system 110, does not compromise the wallet private key in its entirety. In this example, key management system 110 may be maintained across different

systems (e.g., different cloud compute instances), with each system being associated with a unique cryptographic key. In a scenario in which a system associated with a specific cryptographic key fails, a failover process can allow the system to be recovered on a different physical or cloud compute instance so that recovery of private keys, and thus access to the associated wallets, through the key management system 110 is not significantly interrupted.

[0034] In some aspects, to further protect wallet private keys from attacks, such as a man-in-the-middle attack in which a malicious user interposes itself as the key management system 110, the key management system 110 can further encrypt the private key associated with a wallet, prior to sharding and individual encryption, using a platform-specific set of keys. In this example, decrypting the first portion of the private key using the authorization code associated with the wallet and the second portion of the private key using the credentials associated with the key management system 110, and concatenating these decrypted portions of the private key, may recover an intermediate encrypted version of the private key instead of the private key itself. Because the resulting version of the private key is not the actual private key, but instead an intermediate encrypted version of the private key, a man-in-the-middle attack may fail to identify the actual private key associated with the wallet. Further, because brute force attacks of a cryptographic scheme (given a sufficient key size) are computationally impossible to execute (as discussed above), the recovery of the intermediate encryption version of the private key may still not allow for a malicious user to access a wallet.

[0035] The wallet generated by wallet generator 112 may be made publicly known on blockchain 122 to allow for other parties to transfer assets into the wallet. Generally, in making the wallet publicly known, the public key associated with the wallet may be published to other parties on the network 120. As discussed, knowledge of the public key associated with the wallet may allow other users to specify the wallet as the destination address for assets to be transferred in one or more transactions on the blockchain 122. However, knowledge of the public key generally does not allow for assets to be withdrawn or transferred from the wallet.

[0036] To execute transactions on the blockchain on behalf of the ultimate owner of a wallet, key management system 110 allows a controlling party to request the private key associated with the wallet through wallet key decryptor 116. Generally, wallet key decryptor 116 receives, from the controlling party, controlling party credentials and an authorization code associated with the controlling party and uses these credentials to decrypt encrypted versions of the controlling party portion of the private key(s) associated with wallets maintained by the key management system 110. Meanwhile, the wallet key decryptor 116 also receives or

retrieves platform-specific credentials for use in decrypting encrypted versions of the platform portion of the private key(s) associated with wallets maintained by the key management system 110. To decrypt the controlling party portion of the private key, wallet key decryptor 116 can use the controlling party credentials to decrypt a cryptographic salt associated with the user and concatenate the decrypted version of the cryptographic salt with an alphanumeric authorization code associated with the controlling party to generate a decryption key. Wallet key decryptor 116 subsequently uses the decryption key to recover the controlling party portion of the private key(s) associated with wallets maintained by the key management system 110.

[0037] In some aspects, where the controlling party (or a user associated with the controlling party) has forgotten the alphanumeric authorization code, alternative authorization codes can be used to recover the controlling party portion of the private key. For example, as discussed, a decryption key may be generated by a key derivation function, using a series of question-answer pairs as an input. If wallet key decryptor 116 verifies that the controlling party has answered the question-answer pairs correctly, wallet key decryptor 116 can input the question-answer pairs into the key derivation function to recover a decryption key. Wallet key decryptor 116 subsequently uses the decryption key to recover the controlling party portion of the private key.

[0038] As discussed, wallet key decryptor 116 can use the platform-specific credentials to decrypt the platform portion of the private key associated with the requested wallet. These platform-specific credentials may include, for example, a cryptographic key used in a symmetric encryption algorithm or a private key used to decrypt a payload encrypted using a corresponding public key. In some aspects, multiple platform-specific credentials may be used to encrypt (and thus to decrypt) different sub-portions of the platform portion of the private key. In such an example, individual sub-portions of the platform portion of the private key may be independently decrypted (e.g., by wallet key decryptor 116 at key management system 110 or on a remote system, not illustrated in FIG. 1A). The decrypted sub-portions of the platform portion of the private key may then be concatenated into the whole platform portion of the private key for use in recovering the private key associated with the wallet and granting access to the wallet, as discussed in further detail below.

[0039] In some aspects, as illustrated, wallet key decryptor 116 may output the decrypted first key portion and the decrypted second key portion to transaction signer 118 for use in signing a transaction to be committed to the blockchain 122. Transaction signer 118 can generate the private key(s) associated with wallets maintained by the key management system

110, which grants the controlling party access to the wallet(s) associated with the private key(s), by concatenating the decrypted first key portion and the decrypted second key portion into a single key for each of the private key(s) associated with wallets maintained by the key management system 110. These decrypted keys may then be used to sign transaction records (according to the ultimate owner of the wallets from which assets are to be transferred), which generally includes information defining assets transferred on the blockchain 122 as part of the transaction and the public key of the destination wallet to which the assets are to be transferred. Transaction signer 118 may then output the signed transaction to blockchain 122 for publication, verification, and finalization.

[0040] In some aspects, as discussed, the controlling party portion and the platform portion of the wallet private key may be further protected by a set of keys associated with the key management system 110. To recover the wallet private key(s) associated with wallets maintained by the key management system 110 and grant access to the associated wallets, the controlling party portion and the platform portion of the wallet private key may be combined (e.g., concatenated), and the combination of the controlling party portion and the platform portion of the wallet private key(s) may be decrypted using the set of keys associated with the key management system 110. As discussed, further encrypting the controlling party portion and the platform portion of the wallet private key(s) may provide further protection against malicious attacks, such as a man-in-the-middle attack, as such additional encryption of the wallet private key(s) may require that malicious parties have possession of additional information beyond a user's authorization code and other credentials in order to compromise a specific wallet, and because a brute force attack to recover any specific wallet private key is computationally impossible to perform.

[0041] Network 120 may, in some aspects, be a cryptocurrency network for which key management system 110 processes transactions. By way of example, network 120 may be a network such as ALGORAND™, BITCOIN™, ETHEREUM®, SOLANA™, STELLAR™, TRON™, and other cryptocurrency networks. Transactions on a blockchain 122 hosted by network 120 may include, for example, the execution of one or more smart contracts on the blockchain 122 or by the generation of one or more blocks on the blockchain evidencing the occurrence of a transaction on the blockchain 122.

[0042] FIG. 1B illustrates an example computing environment 100B in which a private key defining a user wallet on a blockchain is secured using an authorization code, user credentials, and platform credentials in a distributed. As illustrated, computing environment

100B includes a key management system 110, a client transaction gateway 130, and a network 120.

[0043] In the computing environment 100B, the wallet generator 112 can, as in the computing environment 100A, generate a wallet key by requesting a private key and public key pairing from the key entropy pool 114 and shard the private key into multiple portions and independently encrypted so that a compromise of a key used to encrypt one portion of the private key does not compromise the entire private key and thus the wallet associated with the private key. For example, as discussed above, the private key can be sharded into a first portion and a second portion, with the first portion (the controlling party portion of the private key) being encrypted based on the authorization code generated by the controlling party of the centralized platform that is creating a wallet for an ultimate owner using the key management system 110 and the second portion (the client portion of the private key) being encrypted based on credentials associated with the client transaction gateway 130. As discussed in further detail herein, the client transaction gateway 130 may be a system through which a client device initiates the execution of transactions on the blockchain 122 and may be controlled by the ultimate owner of the wallet generated by the wallet generator 112. After encrypting the different portions of the wallet private key, the controlling party portion of the private key may be maintained at the key management system 110, and the client portion of the private key may be distributed to the client transaction gateway 130 for use in distributed computation of the wallet key.

[0044] To perform a transaction involving digital assets stored in a wallet generated and maintained through the key management system 110, a transaction request may be provided to the key management system 110 (e.g., via the client transaction gateway 130) to initiate the process of recovering the wallet private key and signing a transaction using the wallet private key. The transaction request may be, in some aspects, accompanied by information identifying the client so that the appropriate portion of the wallet private key (e.g., the controlling party portion of the private key) may be recovered by the wallet key decryptor 116.

[0045] To initiate a transaction, the controlling party portion of the wallet private key may be provided to the transaction verifier 132 at the client transaction gateway 130. Generally, the transaction verifier 132 verifies that the transaction request is a legitimate transaction request, and if so, recovers the client portion of the wallet private key. The transaction verifier 132 can verify that the transaction request is a legitimate transaction request using various techniques. For example, a transaction request may include unique identifying information, such as a

transaction hash, a sequence number, or other additional information that the transaction verifier 132 can examine in order to verify that a transaction request is a legitimate transaction request. If the identifying information in the transaction request is not valid identifying information (e.g., a sequence number is less than the highest sequence number in a transaction database associated with the client transaction gateway 130, a hash of identifying information is invalid, etc.), the transaction verifier 132 can reject the transaction request. In another example, the transaction verifier 132 can request verification of the transaction from one or more users associated with the client transaction gateway 130. Verification of the transaction may be performed, for example, in a hierarchical manner in which different layers of verification with different numbers of approving parties are specified for the transaction (e.g. based on an asset size associated with the transaction, etc.), by a specified approving party (which may differ based on an asset size associated with the transaction), or the like. If the designated approving user(s) reject the transaction request, transaction processing terminates without the transaction being signed and committed to the blockchain 122.

[0046] If the transaction request is verified at the transaction verifier 132, the transaction verifier 132 proceeds with recovering the client portion of the private key. Generally, the transaction verifier 132 can recover the client portion of the private key by decrypting an encrypted version of the client portion of the private key using a credential associated with the client transaction gateway 130 and/or a user associated with the transaction request verified by the transaction verifier 132. After the client portion of the private key is decrypted, the private key can be recovered by combining the controlling party portion of the wallet private key with the client portion of the wallet private key. For example, as discussed above, the wallet private key may be recovered by concatenating the different portions of the private key (e.g., the controlling party portion of the wallet private key and the client portion of the wallet private key) together.

[0047] The wallet private key may be provided to a transaction signer 134. In turn, the transaction signer 134 can sign transaction records associated with the transaction request and commit these signed transaction records to the blockchain 122 for publication, verification, and finalization. Generally, as discussed, these transaction records may include information defining the assets to be transferred on the blockchain 122 as part of the transaction and the public key or other identifying information for the wallet into which the assets are to be transferred.

[0048] In the computing environment 100B illustrated in **FIG. 1**, the verification of a transaction at the client transaction gateway 130 may provide an additional level of security protecting assets stored in a wallet from being maliciously or erroneously transferred to a destination wallet (which, due to the nature of transactions executed on the blockchain 122, may result in a permanent loss of those assets if such a transaction were executed and committed to the blockchain 122). A transaction request may be initiated by any party that can provide transaction requests to the client transaction gateway 130 for processing. However, before a transaction request is signed and committed to the blockchain 122, the transaction request may be held for verification, and only verified transactions may be committed to the blockchain 122. By doing so, the risk of malicious or erroneous transfers of assets being performed from an ultimate owner's wallet may be reduced, as multiple parties may need to be compromised in order for such a transfer to be effectuated.

Example Wallet Private Key Generation and Access

[0049] **FIG. 2** illustrates example operations 200 that may be performed by a key management system 202 and a blockchain 204 to establish a wallet on behalf of a user of a centralized platform managed by a controlling party based on an authorization code and user credentials associated with the controlling party, according to aspects of the present disclosure. The key management system 202 may correspond, for example, to key management system 110 illustrated in **FIGs. 1A-1B**, and the blockchain 204 may correspond to the blockchain 122 illustrated in **FIGs. 1A-1B**.

[0050] As illustrated, a wallet generation process is generally prompted by the receipt, at key management system 202, of a wallet creation request 210 from the controlling party of the key management system 202. The wallet creation request 210 generally requests the creation of a wallet for and on behalf of the ultimate owner of the wallet and includes an authorization code and user credentials associated with the controlling party. As discussed, the authorization code may include multiple authentication mechanisms, such as an alphanumeric code, a series of question-answer pairs, or the like. Generally, the authorization code may be data that the controlling party and owner of the key management system 202 knows, is, or has possession of.

[0051] Response to receipt of the wallet creation request 210, the key management system 202 selects wallet keys from an entropy pool at block 212. The wallet keys may include a public key which may be published to allow other parties on the blockchain 204 to specify the wallet

as a destination wallet for digital assets to be transferred on the blockchain 204 via one or more signed transactions and a private key which allows the owner of the wallet (e.g., the controlling party of key management system 202) to access the digital assets contained in the wallet. In some aspects, the wallet keys may be generated using a random generator that uses data in the entropy pool, such as log data or other random data that cannot be easily replicated, in order to generate a set of keys from which subsequently generated keys cannot be derived and which cannot itself be derived from previously generated keys. After the keys are selected from the entropy pool at block 212, a wallet creation message 214 is published on the blockchain to establish the wallet as a location into which digital assets can be received and a location from which digital assets can be transmitted.

[0052] To protect the private key, key management system 202 proceeds to generate a salt at block 216. The salt may be a randomly generated alphanumeric string, a randomly generated number, or some other data which cannot easily be recovered or replicated and which can be used to augment an authentication code received from the controlling party of the key management system 202. At block 218, the key management system 202 proceeds to encrypt the salt using the user credentials received in wallet creation request 210. The encrypted salt may be stored with the user credentials in an access management system used by key management system 202 to authenticate a user and grant the user access to the key management system 202.

[0053] At block 220, the wallet private key is sharded into a plurality of portions, or shards. Each shard may correspond to a unique portion of the wallet private key and may be encrypted separately using one or more encryption techniques and sets of keys.

[0054] At block 222, the key management system 202 encrypts the wallet private key shards. As discussed, the wallet private key shards may correspond to a controlling party portion of the wallet private key and a platform portion of the wallet private key. To protect the controlling party portion of the wallet private key and allow for the controlling party portion to be recovered, multiple encrypted versions of the controlling party portion of the wallet private key may be generated. A first encrypted version of the controlling party portion of the wallet private key may be encrypted using a cryptographic key generated based on a concatenation or other combination of an alphanumeric authorization code and the cryptographic salt generated at block 216. Meanwhile, a second encrypted version of the controlling party portion of the wallet private key may be encrypted using a cryptographic key generated based on a key derivation function using a series of question-answer pairs as an input.

It should be recognized, however, that other data sources can be used to generate the second encrypted version of the controlling party portion of the wallet private key, and the foregoing provides non-limiting examples of the data based on which encryption keys used to encrypt the controlling party portion of the wallet private key can be generated.

[0055] Meanwhile, the key management system 202 uses various platform-specific credentials to encrypt the platform portion of the wallet private key. In some aspects, to protect against security risks that may exist from using a single system to encrypt the platform portion of the wallet private key, multiple encrypted versions of the wallet private key can be generated. In one example, the platform portion of the wallet private key may be encrypted using multiple key pairs, and the platform portion of the wallet private key may be verified by ensuring that the different decrypted versions of the platform portion of the wallet private key match. In another example, the platform portion of the wallet private key may be divided into a plurality of sub-portions, with each sub-portion being independently protected using different sets of encryption/decryption keys (which, in some aspects, may correspond to keys for different authentication systems used within a computing environment). The encrypted platform portion of the wallet private key may be stored with user credentials to allow for future access to the wallet through key management system 202.

[0056] While not illustrated, it should be recognized that, in some aspects, the platform portion of the wallet private key may correspond to a client-specific portion of the wallet private key (e.g., in instances in which key computation is distributed between a key management system maintained by a controlling party and a transaction gateway maintained by another party). In such a case, the platform portion (or client portion) of the wallet private key may be distributed to a client transaction gateway (such as the client transaction gateway 130 illustrated in **FIG. 1B**) for separate encryption, storage, and use in signing transactions committed to the blockchain.

[0057] **FIG. 3** illustrates example operations 300 that may be performed by a key management system 302 and a blockchain 304 to perform transactions on the blockchain 304 using assets in a wallet based on an authorization code and user credentials, according to aspects of the present disclosure. The key management system 302 may correspond, for example, to key management system 110 illustrated in **FIG. 1A**, and the blockchain 304 may correspond to the blockchain 122 illustrated in **FIG. 1A**.

[0058] As illustrated, to execute a transaction on the blockchain 304, a wallet access request 310 may be received at key management system 302. The wallet access request generally includes controlling party credentials, a controlling party authentication code, information identifying digital assets to be transferred as part of a transaction, and the public key of the wallet to which the identified digital assets are to be transferred. In some aspects, the wallet access request 310 may further include information identifying an ultimate owner of a wallet from which the identified digital assets are to be transferred.

[0059] At block 312, key management system 302 decrypts a cryptographic salt using controlling party credentials included in the wallet access request 310. For example, the controlling party credentials may be associated with a decryption key (which may be a key in a symmetric or an asymmetric encryption scheme) used to encrypt the cryptographic salt. The decryption key may be used to decrypt an encrypted payload (e.g., from profile or account data stored at key management system 302 or another user identity authority) and recover the cryptographic salt from the encrypted payload.

[0060] At block 314, key management system 302 decrypts the controlling party private key shard (e.g., the controlling party portion of the private key) using user credentials and the decrypted cryptographic salt. In some aspects, the user credentials may be an alphanumeric PIN code that may be combined with the cryptographic salt to generate the controlling party private key shard. In some aspects, where the user has entered a recovery model, the user credentials may be other data which the user can input into a key derivation function in order to recover the private key shard, alone or in combination with the decrypted cryptographic salt.

[0061] At block 316, key management system 302 decrypts the application private key shard (e.g., the platform portion of the private key, as discussed above). The application private key shard may be decrypted from an encrypted payload stored as part of a user profile (e.g., stored at key management system 302 or another user identity authority) using decryption keys associated with the key management system 302 and/or other systems involved in authenticating users of key management system 302, processing transactions to be committed to blockchain 304, and the like. In some aspects, the application private key shard may be divided into multiple sub-portions, with different decryption keys being used to decrypt different portions of the application private key shard.

[0062] At block 318, key management system 302 obtains the wallet private key for one or more wallets controlled by the controlling party on behalf of ultimate owners using key

management system 302 to perform transactions on the blockchain 304 based on the decrypted controlling party private key shard and the decrypted application private key shard. In some aspects, the wallet private key may be generated by concatenating the decrypted controlling party private key shard and the decrypted application private key shard into a single key, represented by a string of a defined bit length corresponding to the encryption bit length used by a cryptographic scheme used to sign transactions on blockchain 304.

[0063] At block 320, key management system 302 signs a transaction using the wallet private key. The transaction generally includes information identifying the digital assets to be transferred as part of the transaction and a destination wallet for the identified digital assets. The signed transaction 322 is then output from key management system 302 to the blockchain 304 for processing.

[0064] FIG. 4 illustrates example operations 400 that may be performed by a key management system 402, a client transaction gateway 404, and a blockchain 406 to perform transactions on the blockchain 406 using assets in a wallet based on client-side verification of the transaction, according to aspects of the present disclosure. The key management system 402 may correspond, for example, to key management system 110 illustrated in FIG. 1B, the client transaction gateway 404 may correspond to the client transaction gateway 130 illustrated in FIG. 1B, and the blockchain 406 may correspond to the blockchain 122 illustrated in FIG. 1B.

[0065] As illustrated, to execute a transaction on the blockchain 406, a transaction request 410 may be received at the key management system 402. Generally, the transaction request 410 may include controlling party credentials, a controlling party authentication code, information identifying digital assets to be transferred as part of a transaction, and the public key of the wallet to which the identified digital assets are to be transferred. In some aspects, the transaction request 410 may further include information identifying the ultimate owner of the wallet from which the identified digital assets are to be transferred. The ultimate owner of the wallet from which the identified digital assets are to be transferred may be, for example, the owner or other controlling party associated with the client transaction gateway 404.

[0066] At block 412, the key management system 402 decrypts a cryptographic salt using controlling party credentials included in the transaction request 410. For example, the controlling party credentials may be associated with a decryption key (which may be a key in a symmetric or an asymmetric encryption scheme) used to encrypt the cryptographic salt. The

decryption key may be used to decrypt an encrypted payload (e.g., from profile or account data stored at key management system 402 or another user identity authority) and recover the cryptographic salt from the encrypted payload.

[0067] At block 414, the key management system 402 decrypts the controlling party private key shard (e.g., the controlling party portion of the private key) using user credentials and the decrypted cryptographic salt. In some aspects, the user credentials may be an alphanumeric PIN code that may be combined with the cryptographic salt to generate the controlling party private key shard. In some aspects, where the user has entered a recovery model, the user credentials may be other data which the user can input into a key derivation function in order to recover the private key shard, alone or in combination with the decrypted cryptographic salt.

[0068] After the controlling party private key shard is decrypted, the key management system 402 outputs a transaction verification request 416 to the client transaction gateway 404 for further processing. In some aspects, the transaction verification request 416 may include the transaction information included in the transaction request 410 and the controlling party private key shard. The contents of the transaction verification request 416 may be signed, for example, using one or more encryption keys agreed upon by the key management system 402 and the client transaction gateway 404. For example, the controlling party portion of the private key may be used to sign the payload of the transaction verification request 416. By signing the payload of the transaction verification request 416, the client transaction gateway 404 can determine whether the transaction verification request 416 is a legitimate verification request (e.g., whether the transaction verification request 416 is received from the key management system 402 or from some other, potentially untrusted, party).

[0069] At block 418, the client transaction gateway 404 verifies the transaction request. Verification of the transaction request may be a multi-step process involving verification that the transaction request has been received from the key management system 402 and that the transaction for which verification is requested is a legitimate, authorized transaction initiated by a legitimate user of the client transaction gateway 404. To verify that the transaction request was received from the key management system 402, a hash of the contents of the transaction verification request 416 may be generated and compared to a decrypted version of the signature associated with the transaction verification request 416. If the hash and the decrypted version of the signature associated with the transaction verification request 416 match, the transaction verification request 416 may be determined to have been received from a legitimate key

management system 402 (and not an imposter system), and verification may proceed with requesting, from one or more approving parties, verification of the transaction itself.

[0070] Verification of the transaction identified in the transaction verification request 416 may be performed automatically or based on authentication messages received from one or more approving parties interfacing with the client transaction gateway 404. In an automated verification scheme, unique identifying information associated with the transaction in the transaction verification request may be compared to prior transaction information stored in an external database to determine whether the transaction includes valid parameters indicative of the transaction being a legitimate transaction. For example, mismatches between an expected sequence number for the transaction and a sequence number included in the transaction verification request 416 may indicate that the transaction is not a valid transaction. In another example, information identifying the source of the transaction request 410 may be compared to an identifier associated with the client transaction gateway 404 and/or client devices authorized to interface with the client transaction gateway 404. If the source identifier for the transaction does not match an identifier associated with the client transaction gateway 404 and/or client devices authorized to interface with the client transaction gateway 404, transaction verification may fail, and the operations 400 may terminate.

[0071] In a verification regime in which a transaction is verified based on authentication messages received from one or more approving parties interfacing with the client transaction gateway 404, the client transaction gateway 404 may transmit messages to one or more approving parties requesting verification of the transaction specified in the transaction request 410. The number of approving parties and the identity of the approving parties may be, in some aspects, be based on the amount and/or types of digital assets involved in the transaction and specified in the transaction request 410. If approval messages are received from the requisite number of approving parties, the transaction verification request 416 may be satisfied, and operations 400 may proceed to block 418. Otherwise, the transaction may fail verification, and accordingly, operations 400 may terminate.

[0072] At block 420, the client transaction gateway 404 recovers the client private key shard based on the verification of the transaction request at block 418. In some aspects, the client private key shard may be recovered based on user credentials and/or a user authorization code associated with the client controlling the client transaction gateway 404.

[0073] At block 422, the transaction is signed using a private key recovered from the controlling party and client key shards. Generally, the private key may be the wallet private key for one or more wallets controlled by the controlling party on behalf of ultimate owners using the key management system 402 and the client transaction gateway 404 to perform transactions on the blockchain 406. The wallet private key may be generated, for example, by concatenating the controlling party private key shard (which may be provided to the client transaction gateway 404 in the transaction verification request 416 or in a separate message) and the recovered client private key shard into a single key, represented by a string of a defined bit length corresponding to the encryption bit length used by a cryptographic scheme used to sign transactions on the blockchain 406. The transaction generally includes information identifying the digital assets to be transferred as part of the transaction and a destination wallet for the identified digital assets. The signed transaction 424 is then output from the client transaction gateway 404 to the blockchain 406 for processing.

Example Operations for Accessing Wallets in Blockchain Systems Using Sharded Private Keys

[0074] FIG. 5 illustrates example operations 500 for accessing a wallet on a blockchain based on an authorization code and user credentials, according to aspects of the present disclosure. Operations 500 may be performed, for example, by a key management system 110 illustrated in FIGs. 1A-1B or other processing system that can grant access to a wallet used in performing transactions on a blockchain.

[0075] As illustrated, operations 500 begin at block 510, with receiving a request to access one or more wallets on a blockchain. Generally, the request includes an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party. The controlling party may be, for example, an owner or other superuser associated with a key management system that allows for wallets to be generated and maintained for and on behalf of ultimate owners through the key management system.

[0076] In some aspects, the authorization code associated with the controlling party includes an alphanumeric string.

[0077] In some aspects, the authorization code associated with the controlling party includes one or more question-answer pairs. The questions may be selected from a set of *a priori* defined questions, and the answers may be generated by the controlling party (or a user

acting on behalf of the controlling party). As discussed, the one or more question-answer pairs may be used as a recovery mechanism to allow a user to recover a private key associated with the wallet, or at least a controlling party portion of the private key, in the event that the user has forgotten a primary authorization code associated with the private key.

[0078] At block 520, operations 500 proceed with decrypting a first portion of a private key based on the authorization code and a salt associated with the user credentials. In some aspects, the salt may be a randomly generated number encrypted using a key associated with the user credentials. The key associated with the user credentials may be, for example, stored as part of a user profile at a key management system or other user identity authority, or may be derived from the user credentials using a key derivation function.

[0079] At block 530, operations 500 proceed with decrypting a second portion of the private key based on credentials associated with an application through which the wallet is accessed.

[0080] In some aspects, the second portion of the private key may include at least a first sub-portion and a second sub-portion. To decrypt the second portion of the private key, the first sub-portion may be decrypted based on a first set of credentials associated with the application, and the second sub-portion may be decrypted based on a second set of credentials associated with the application.

[0081] At block 540, operations 500 proceed with granting access to the one or more wallets based on the decrypted first portion and the decrypted second portion of the private key.

[0082] In some aspects, granting access to the one or more wallets may include granting access to withdraw items stored in the wallet to an external resource based on signing a transaction using the decrypted first portion and the decrypted second portion of the private key.

[0083] In some aspects, granting access to the one or more wallets includes generating an intermediate encrypted version of the private key based on the decrypted first portion and the decrypted second portion of the private key. The private key may be decrypted based on the intermediate encrypted version of the private key and a platform-specific decryption key, and the decrypted private key may grant access to the wallet.

[0084] In some aspects, operations 500 further include receiving a request to generate the one or more wallets, the request including at least the authorization code associated with the

controlling party. A private key associated with the one or more wallets may be selected from an entropy pool of private keys, the one or more wallets may be generated based on the selected private key.

[0085] FIG 6 illustrates example operations 600 for accessing a wallet managed by a controlling party on a blockchain based on client-side transaction verification, according to aspects of the present disclosure.

[0086] As illustrated, the operations 600 begin at block 610, with receiving, from a key management system, a request to execute a transaction on a blockchain. Generally, the request includes at least a first portion of a private key associated with a wallet from which the transaction is to be performed.

[0087] In some aspects, the request to execute the transaction comprises a request to withdraw digital assets stored in the wallet to an external resource based on signing a transaction record based on the first portion of the private key and the second portion of the private key.

[0088] In some aspects, the request to execute the transaction on the blockchain is signed based on the first portion of the private key. The operations 600 may further include validating a signature associated with the request based on a public key counterpart to the first portion of the private key. The recovery of the second portion of the private key, as discussed below with respect to block 630, may further be based on validating the signature associated with the request.

[0089] At block 620, the operations 600 proceed with verifying that the transaction is a legitimate transaction to be executed on the blockchain.

[0090] In some aspects, verifying that the transaction is a legitimate transaction to be executed on the blockchain comprises requesting verification of the transaction from one or more approving parties. The one or more approving parties may be based on digital assets to be transferred from the wallet via the transaction.

[0091] At block 630, the operations 600 proceed with recovering a second portion of the private key based on verifying that the transaction is a legitimate transaction to be executed on the blockchain.

[0092] In some aspects, recovering the second portion of the private key comprises decrypting the second portion of the private key based on credentials associated with a transaction gateway through which the transaction is verified.

[0093] At block 640, the operations 600 proceed with accessing the wallet to execute the transaction on the blockchain based on the first portion of the private key and the second portion of the private key.

[0094] In some aspects, accessing the wallet to execute the transaction includes concatenating the first portion of the private key and the second portion of the private key into a concatenated key. A transaction record identifying assets to be transferred from the wallet to a destination wallet is generated and signed using the concatenated key. The signed transaction record may subsequently be committed to the blockchain for processing (e.g., publication to nodes participating in processing transactions on the blockchain, verification by such nodes, and finalization).

[0095] In some aspects, the operations 600 further include forwarding, to the key management system from a client device, information specifying parameters of the transaction. The request to execute the transaction is received based on forwarding the information specifying the parameters of the transaction.

[0096] In some aspects, the key management system comprises a system associated with a centralized platform through which owners of the wallet perform transactions on the blockchain.

Example System for Generating and Granting Access to Wallets in Blockchain Systems Using Sharded Private Keys

[0097] FIG. 7 illustrates an example system 700 configured to perform the methods described herein, including, for example, operations 200 illustrated in FIG. 2, operations 300 illustrated in FIG. 3, operations 400 illustrated in FIG. 4, operations 500 illustrated in FIG. 5, and/or operations 600 illustrated in FIG. 6. In some embodiments, system 700 may act as a key management system through which wallets are maintained and/or a client transaction gateway through which requests to perform transactions involving assets stored in such wallets are processed, such as the key management system 110 and/or the client transaction gateway 130 illustrated in FIGs. 1A-1B.

[0098] As shown, system 700 includes a central processing unit (CPU) 702, network interface 706 through which system 700 is connected to network 790 (which may be a local network, an intranet, the internet, or any other group of computing devices communicatively connected to each other), a memory 708, and an interconnect 712. The network interface 706 may be used to receive requests to upgrade bridged tokens on a blockchain to native tokens on the blockchain (e.g., as depicted and described with respect to **FIGs. 1** through **6**).

[0099] CPU 702 may retrieve and execute programming instructions stored in the memory 708. Similarly, the CPU 702 may retrieve and store application data residing in the memory 708. The interconnect 712 transmits programming instructions and application data, among the CPU 702, network interface 706, and memory 708.

[0100] CPU 702 is included to be representative of a single CPU, multiple CPUs, a single CPU having multiple processing cores, and the like.

[0101] Memory 708 is representative of a volatile memory, such as a random access memory, or a nonvolatile memory, such as nonvolatile random access memory, phase change random access memory, or the like. As shown, memory 708 includes a wallet generator 720, a key entropy pool 730, a wallet key decryptor 740, a transaction signer 750, and a transaction verifier 760.

[0102] Wallet generator 720 generally corresponds to wallet generator 112 illustrated in **FIGs. 1A-1B**. Generally, wallet generator 720 receives a request to generate a wallet for a user. The user is generally the ultimate owner (also known as a beneficial owner) of the wallet for which a controlling party maintains a wallet. To generate a wallet, wallet generator retrieves a public key and private key pair from key entropy pool 730 (which may correspond to key entropy pool 114 illustrated in **FIGs. 1A-1B**). The private key may be sharded into a controlling party portion and a platform portion, with the controlling party portion being encrypted using an authentication code and user credentials included in the received request and the platform portion being encrypted using credentials associated with the system 700. In some aspects, the controlling party portion may be encrypted using multiple techniques. A first encrypted version of the controlling party portion may be encrypted based on a combination of an authentication code associated with the controlling party and a cryptographic salt, which may be encrypted using the user credentials; meanwhile, a second encrypted version of the controlling party portion may be encrypted based on a key derived from a set of question-answer pairs established by the controlling party.

[0103] Key entropy pool 730 generally includes a plurality of private keys generated using random seeds or other random data input into a random key address generator. A key selected from the key entropy pool 730 may not be derived from keys previously generated by or within key entropy pool 730, and subsequent keys in the key entropy pool 730 may not be derived from the selected key.

[0104] Wallet key decryptor 740 generally corresponds to wallet key decryptor 116 illustrated in **FIGs. 1A-1B**. Generally, wallet key decryptor 740 receives a request to access one or more wallets from a user of system 700 (which may be a controlling party of system 700 acting for and on behalf of the ultimate owner(s) of the one or more wallets). The request generally includes an authorization code and user credentials associated with the controlling party. To grant access to the one or more wallets, wallet key decryptor 740 decrypts the controlling party portion of the wallet private key based on the authorization code and user credentials included in the request and decrypts the platform portion of the wallet private key based on platform credentials associated with system 700 (and/or other authentication or user identity authorities which may be used to authenticate a user and protect the secrecy of wallet private keys).

[0105] Transaction signer 750 generally corresponds to transaction signer 118 illustrated in **FIG. 1A** and/or transaction signer 134 illustrated in **FIG. 1B**. Generally, transaction signer 750 uses the decrypted controlling party portion and the decrypted platform portion of the wallet private key to recover the wallet private key(s) and grant the user access to the wallet(s) maintained for and on behalf of the ultimate owner(s) by a controlling party associated with the system 700. Generally, the decrypted controlling party portion of the private key may be combined with the decrypted platform portion of the private key in order to recover the wallet private key. The recovered private key may then be used to grant access to digital assets stored in the wallet associated with the wallet private key, sign transaction records evidencing the transfer of digital assets from the wallet associated with the wallet private key to a recipient wallet, and the like. These transaction records may then be output (e.g., via network interface 706) to a blockchain (e.g., on network 790) for further processing.

[0106] Transaction verifier 760 generally corresponds to transaction verifier 132 illustrated in **FIG. 1B**. Generally, the transaction verifier 760 receives a request to execute a transaction on the blockchain from a key management server through which wallets are maintained. The request generally includes information about the transaction and a first portion (e.g., a controlling party portion) of a private key associated with a wallet from which assets are to be

withdrawn as part of the transaction. The transaction verifier 760 can identify one or more approving parties from which approval and verification of the transaction is requested. If a sufficient number of parties approve and verify the legitimacy of the transaction, the transaction verifier 760 can indicate, to the transaction signer 750, that the generation, signing, and commitment of a transaction record to the blockchain may commence.

Example Clauses

[0107] Implementation details for various aspects of the present disclosure are described in the following numbered clauses.

[0108] Clause 1: A computer-implemented method, comprising: receiving a request to access one or more wallets on a blockchain, the request including an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party; decrypting a first portion of a private key based on the authorization code and a salt associated with the user credentials; decrypting a second portion of the private key based on credentials associated with an application through which the wallet is accessed; and granting access to the one or more wallets based on the decrypted first portion and the decrypted second portion of the private key.

[0109] Clause 2: The method of Clause 1, wherein the authorization code associated with the controlling party comprises an alphanumeric string.

[0110] Clause 3: The method of any of Clauses 1 or 2, wherein the authorization code associated with the controlling party comprises one or more question-answer pairs.

[0111] Clause 4: The method of any of Clauses 1 through 3, wherein decryption the second portion of the private key comprises decrypting a first sub-portion of the second portion based on a first set of credentials associated with the application and a second sub-portion of the second portion based on a second set of credentials associated with the application.

[0112] Clause 5: The method of any of Clauses 1 through 4, wherein the salt comprises a randomly generated number encrypted using a key associated with the user credentials.

[0113] Clause 6: The method of any of Clauses 1 through 5, further comprising: receiving a request to generate the one or more wallets, the request including at least the authorization code associated with the controlling party associated with the wallet; selecting the private keys associated with the one or more wallets from an entropy pool of private keys; and generating the one or more wallets based on the selected private key.

[0114] Clause 7: The method of any of Clauses 1 through 6, wherein granting access to the one or more wallets comprises granting access to withdraw items stored in the one or more wallets to an external resource based on signing a transaction using the decrypted first portion and the decrypted second portion of the private key.

[0115] Clause 8: The method of any of Clauses 1 through 7, wherein granting access to the one or more wallets comprises: generating an intermediate encrypted version of the private key based on the decrypted first portion and the decrypted second portion of the private key; and decrypting the private key based on the intermediate encrypted version of the private key and a platform-specific decryption key, wherein the decrypted private key grants access to the one or more wallets.

[0116] Clause 9: The method of any of Clauses 1 through 8, wherein the controlling party associated with the one or more wallets comprises a party associated with a centralized platform through which owners of the one or more wallets perform transactions on the blockchain.

[0117] Clause 10: A computer-implemented method, comprising: receiving, from a key management system, a request to execute a transaction on a blockchain, the request including a first portion of a private key associated with a wallet from which the transaction is to be performed; verifying that the transaction is a legitimate transaction to be executed on the blockchain; based on verifying that the transaction is a legitimate transaction to be executed on the blockchain, recovering a second portion of the private key; and accessing the wallet to execute the transaction on the blockchain based on the first portion of the private key and the second portion of the private key.

[0118] Clause 11: The method of Clause 10, further comprising forwarding, to the key management system from a client device, information specifying parameters of the transaction, wherein the request to execute the transaction is received based on forwarding the information specifying the parameters of the transaction.

[0119] Clause 12: The method of any of Clauses 10 or 11, wherein the request to execute the transaction comprises a request to withdraw digital assets stored in the wallet to an external resource based on signing a transaction record based on the first portion of the private key and the second portion of the private key.

[0120] Clause 13: The method of any of Clauses 10 through 12, wherein the request to execute the transaction on the blockchain is signed based on the first portion of the private key.

[0121] Clause 14: The method of Clause 13, further comprising validating a signature associated with the request based on a public key counterpart to the first portion of the private key, wherein the recovering the second portion of the private key is further based on validating the signature associated with the request.

[0122] Clause 15: The method of any of Clauses 10 through 14, wherein verifying that the transaction is a legitimate transaction to be executed on the blockchain comprises requesting verification of the transaction from one or more approving parties.

[0123] Clause 16: The method of Clause 15, further comprising identifying the one or more approving parties based on digital assets to be transferred from the wallet via the transaction.

[0124] Clause 17: The method of any of Clauses 10 through 16, wherein recovering the second portion of the private key comprises decrypting the second portion of the private key based on credentials associated with a transaction gateway through which the transaction is verified.

[0125] Clause 18: The method of any of Clauses 10 through 17, wherein accessing the wallet to execute the transaction comprises: concatenating the first portion of the private key and the second portion of the private key into a concatenated key; generating a transaction record identifying assets to be transferred from the wallet to a destination wallet; signing the transaction record using the concatenated key; and committing the signed transaction record to the blockchain.

[0126] Clause 19: The method of any of Clauses 1 through 18, wherein the key management system comprises a system associated with a centralized platform through which owners of the wallet perform transactions on the blockchain.

[0127] Clause 20: A system, comprising: a memory having executable instructions stored thereon; and a processor configured to execute the executable instructions to perform the operations of any one of Clauses 1 through 19.

[0128] Clause 21: A system, comprising: means for performing the operations of any one of Clauses 1 through 19.

[0129] Clause 22: A computer-readable medium having instructions stored thereon which, when executed by a processor, performs the operations of any one of Clauses 1 through 19.

Additional Considerations

[0130] The preceding description is provided to enable any person skilled in the art to practice the various embodiments described herein. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments. For example, changes may be made in the function and arrangement of elements discussed without departing from the scope of the disclosure. Various examples may omit, substitute, or add various procedures or components as appropriate. Also, features described with respect to some examples may be combined in some other examples. For example, an apparatus may be implemented or a method may be practiced using any number of the aspects set forth herein. In addition, the scope of the disclosure is intended to cover such an apparatus or method that is practiced using other structure, functionality, or structure and functionality in addition to, or other than, the various aspects of the disclosure set forth herein. It should be understood that any aspect of the disclosure disclosed herein may be embodied by one or more elements of a claim.

[0131] As used herein, a phrase referring to “at least one of” a list of items refers to any combination of those items, including single members. As an example, “at least one of: a, b, or c” is intended to cover a, b, c, a-b, a-c, b-c, and a-b-c, as well as any combination with multiples of the same element (e.g., a-a, a-a-a, a-a-b, a-a-c, a-b-b, a-c-c, b-b, b-b-b, b-b-c, c-c, and c-c-c or any other ordering of a, b, and c).

[0132] As used herein, the term “determining” encompasses a wide variety of actions. For example, “determining” may include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, “determining” may include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, “determining” may include resolving, selecting, choosing, establishing and the like.

[0133] The methods disclosed herein comprise one or more steps or actions for achieving the methods. The method steps and/or actions may be interchanged with one another without departing from the scope of the claims. In other words, unless a specific order of steps or actions is specified, the order and/or use of specific steps and/or actions may be modified without departing from the scope of the claims. Further, the various operations of methods described above may be performed by any suitable means capable of performing the corresponding functions. The means may include various hardware and/or software component(s) and/or module(s), including, but not limited to a circuit, an application specific integrated circuit (ASIC), or processor. Generally, where there are operations illustrated in figures, those

operations may have corresponding counterpart means-plus-function components with similar numbering.

[0134] The various illustrative logical blocks, modules and circuits described in connection with the present disclosure may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device (PLD), discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any commercially available processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0135] A processing system may be implemented with a bus architecture. The bus may include any number of interconnecting buses and bridges depending on the specific application of the processing system and the overall design constraints. The bus may link together various circuits including a processor, machine-readable media, and input/output devices, among others. A user interface (e.g., keypad, display, mouse, joystick, etc.) may also be connected to the bus. The bus may also link various other circuits such as timing sources, peripherals, voltage regulators, power management circuits, and the like, which are well known in the art, and therefore, will not be described any further. The processor may be implemented with one or more general-purpose and/or special-purpose processors. Examples include microprocessors, microcontrollers, DSP processors, and other circuitry that can execute software. Those skilled in the art will recognize how best to implement the described functionality for the processing system depending on the particular application and the overall design constraints imposed on the overall system.

[0136] If implemented in software, the functions may be stored or transmitted over as one or more instructions or code on a computer-readable medium. Software shall be construed broadly to mean instructions, data, or any combination thereof, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Computer-readable media include both computer storage media and communication media, such as any medium that facilitates transfer of a computer program from one place to another. The processor may be responsible for managing the bus and general processing, including the

execution of software modules stored on the computer-readable storage media. A computer-readable storage medium may be coupled to a processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. By way of example, the computer-readable media may include a transmission line, a carrier wave modulated by data, and/or a computer readable storage medium with instructions stored thereon separate from the wireless node, all of which may be accessed by the processor through the bus interface. Alternatively, or in addition, the computer-readable media, or any portion thereof, may be integrated into the processor, such as the case may be with cache and/or general register files. Examples of machine-readable storage media may include, by way of example, RAM (Random Access Memory), flash memory, ROM (Read Only Memory), PROM (Programmable Read-Only Memory), EPROM (Erasable Programmable Read-Only Memory), EEPROM (Electrically Erasable Programmable Read-Only Memory), registers, magnetic disks, optical disks, hard drives, or any other suitable storage medium, or any combination thereof. The machine-readable media may be embodied in a computer-program product.

[0137] A software module may comprise a single instruction, or many instructions, and may be distributed over several different code segments, among different programs, and across multiple storage media. The computer-readable media may comprise a number of software modules. The software modules include instructions that, when executed by an apparatus such as a processor, cause the processing system to perform various functions. The software modules may include a transmission module and a receiving module. Each software module may reside in a single storage device or be distributed across multiple storage devices. By way of example, a software module may be loaded into RAM from a hard drive when a triggering event occurs. During execution of the software module, the processor may load some of the instructions into cache to increase access speed. One or more cache lines may then be loaded into a general register file for execution by the processor. When referring to the functionality of a software module, it will be understood that such functionality is implemented by the processor when executing instructions from that software module.

[0138] The following claims are not intended to be limited to the embodiments shown herein, but are to be accorded the full scope consistent with the language of the claims. Within a claim, 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.” Unless specifically stated otherwise, the term “some” refers to one or more. No claim element is to be construed under the provisions

of 35 U.S.C. §112(f) unless the element is expressly recited using the phrase “means for” or, in the case of a method claim, the element is recited using the phrase “step for.” 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.

WHAT IS CLAIMED IS:

1. A computer-implemented method, comprising:
 - receiving a request to access one or more wallets on a blockchain, the request including an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party;
 - decrypting a first portion of a private key based on the authorization code and a salt associated with the user credentials;
 - decrypting a second portion of the private key based on credentials associated with an application through which the wallet is accessed; and
 - granting access to the one or more wallets based on the decrypted first portion and the decrypted second portion of the private key.
2. The method of Claim 1, wherein the authorization code associated with the controlling party comprises an alphanumeric string.
3. The method of Claim 1, wherein the authorization code associated with the controlling party comprises one or more question-answer pairs.
4. The method of Claim 1, wherein decrypting the second portion of the private key comprises decrypting a first sub-portion of the second portion based a first set of credentials associated with the application and a second sub-portion of the second portion based on a second set of credentials associated with the application.
5. The method of Claim 1, wherein the salt comprises a randomly generated number encrypted using a key associated with the user credentials.
6. The method of Claim 1, further comprising:
 - receiving a request to generate the one or more wallets, the request including at least the authorization code associated with the controlling party associated with the wallet;
 - selecting the private keys associated with the one or more wallets from an entropy pool of private keys; and
 - generating the one or more wallets based on the selected private key.

7. The method of Claim 1, wherein granting access to the one or more wallets comprises granting access to withdraw items stored in the one or more wallets to an external resource based on signing a transaction using the decrypted first portion and the decrypted second portion of the private key.

8. The method of Claim 1, wherein granting access to the one or more wallets comprises:

generating an intermediate encrypted version of the private key based on the decrypted first portion and the decrypted second portion of the private key; and

decrypting the private key based on the intermediate encrypted version of the private key and a platform-specific decryption key, wherein the decrypted private key grants access to the one or more wallets.

9. The method of Claim 1, wherein the controlling party associated with the one or more wallets comprises a party associated with a centralized platform through which owners of the one or more wallets perform transactions on the blockchain.

10. A processing system, comprising:

at least one memory having executable instructions stored thereon; and

one or more processors configured to execute the executable instructions to cause the processing system to:

receive a request to access one or more wallets on a blockchain, the request including an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party;

decrypt a first portion of a private key based on the authorization code and a salt associated with the user credentials;

decrypt a second portion of the private key based on credentials associated with an application through which the wallet is accessed; and

grant access to the one or more wallets based on the decrypted first portion and the decrypted second portion of the private key.

11. The processing system of Claim 10, wherein the authorization code associated with the controlling party comprises an alphanumeric string.

12. The processing system of Claim 10, wherein the authorization code associated with the controlling party comprises one or more question-answer pairs.

13. The processing system of Claim 10, wherein to decrypt the second portion of the private key, the one or more processors are configured to cause the processing system to decrypt a first sub-portion of the second portion based a first set of credentials associated with the application and a second sub-portion of the second portion based on a second set of credentials associated with the application.

14. The processing system of Claim 10, wherein the salt comprises a randomly generated number encrypted using a key associated with the user credentials.

15. The processing system of Claim 10, wherein the one or more processors are further configured to cause the processing system to:

- receive a request to generate the one or more wallets, the request including at least the authorization code associated with the controlling party associated with the wallet;

- select the private keys associated with the one or more wallets from an entropy pool of private keys; and

- generate the one or more wallets based on the selected private key.

16. The processing system of Claim 10, wherein to grant access to the one or more wallets, the one or more processors are configured to cause the processing system to grant access to withdraw items stored in the one or more wallets to an external resource based on signing a transaction using the decrypted first portion and the decrypted second portion of the private key.

17. The processing system of Claim 10, wherein to grant access to the one or more wallets, the one or more processors are configured to cause the processing system to:

- generate an intermediate encrypted version of the private key based on the decrypted first portion and the decrypted second portion of the private key; and

- decrypt the private key based on the intermediate encrypted version of the private key and a platform-specific decryption key, wherein the decrypted private key grants access to the one or more wallets.

18. The processing system of Claim 10, wherein the controlling party associated with the one or more wallets comprises a party associated with a centralized platform through which owners of the one or more wallets perform transactions on the blockchain.

19. A computer-readable medium having executable instructions stored thereon which, when executed by one or more processors, executes an operation comprising:

- receiving a request to access one or more wallets on a blockchain, the request including an authorization code associated with a controlling party associated with the one or more wallets and user credentials associated with the controlling party;

- decrypting a first portion of a private key based on the authorization code and a salt associated with the user credentials;

- decrypting a second portion of the private key based on credentials associated with an application through which the wallet is accessed; and

- granting access to the one or more wallets based on the decrypted first portion and the decrypted second portion of the private key.