

DIGITAL ASSET CUSTODIAL SYSTEM

[0001] This application claims the benefit of U.S. provisional patent application no. 62/636,106 filed on February 27, 2018, and U.S. provisional patent application no. 62/640,429 filed on March 8, 2018, and U.S. patent application no. 16,011,529 filed on June 18, 2018, each of which is incorporated by reference herein in its entirety.

FIELD

[0002] At least one embodiment of the present disclosure pertains to computer systems, and more particularly, to a computer-implemented system for maintaining custody of digital assets, such as cryptocurrencies.

BACKGROUND

[0003] Cryptocurrencies such as Bitcoin, Ethereum, Ripple and others have gained in popularity and value in recent years and are expected by many to continue to do so. Every day an increasing variety of transactions are conducted based on cryptocurrencies, and it is conceivable that new types of cryptographic assets may be created in the future, i.e., cryptographic assets that are not necessarily currencies.

[0004] With the increasing use of cryptoassets comes the need for a trusted custodial system that can securely store very large quantities of cryptoassets and control access to those cryptoassets. Indeed, U.S. securities regulations require certain entities that hold more than a certain amount of funds (e.g., \$150 million) on behalf of another party to use a custodian to hold those funds. Hardware wallets and other forms of “cold storage” are sometimes used to store cryptocurrency, however, those devices limit access only to the owner of the device and are therefore not suitable for many business uses, where a number of individuals may require access to cryptographic funds or other cryptoassets.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] One or more embodiments of the present disclosure are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

[0006] Figure 1 is a high-level block diagram of a cryptoasset custodial system (CCS).

[0007] Figure 2A is a schematic diagram illustrating an example of a deposit process flow with the CCS.

[0008] Figure 2B is a flow diagram illustrating an example of the deposit process flow.

[0009] Figure 3A is a schematic diagram illustrating an example of a withdrawal process flow with the CCS.

[0010] Figure 3B is a flow diagram illustrating an example of the withdrawal process flow.

[0011] Figure 4 is a flow diagram illustrating an example of a process performed by the hardware security module (HSM) in connection with a requested operation.

[0012] Figure 5 is a flow diagram illustrating an example of a process for using an offline user device to endorse a requested transaction.

[0013] Figure 6 is a high-level block diagram showing an example of a hardware architecture of a processing system that can be used to implement some or all of the CCS or a user device.

DETAILED DESCRIPTION

[0014] In this description, references to “an embodiment”, “one embodiment” or the like, mean that the particular feature, function, structure or characteristic being described is included in at least one embodiment of the technique introduced here. Occurrences of such phrases in this specification do not necessarily all refer to the same embodiment. On the other hand, the embodiments referred to also are not necessarily mutually exclusive.

Overview

[0015] Introduced here is a computer-implemented cryptoasset custodial system (CCS), i.e., a computer-implemented system for maintaining custody of, and controlling access to, cryptocurrencies and/or other cryptoassets. The CCS may be owned and/or operated by a business enterprise, referred to herein as the Cryptoasset Custodian. The CCS includes multiple layers of security so as to enable large volumes of cryptoassets to be maintained in a secure manner. In certain embodiments the CCS includes a combination of biometric-based multi-user validation, transaction risk analysis, and use of a hardware security module (HSM) to provide authentication/validation functionality and secure storage of private keys of cryptoassets. Furthermore, two or more different biometric authentication techniques may be applied to any given transaction request. As used herein, the term “hardware security module” or “HSM” refers to a special-purpose physical computing device that safeguards and manages digital keys for authentication and provides cryptoprocessing functionality. The HSM can be embodied as a plug-in card or an external device that attaches directly to a computer.

[0016] In certain embodiments, when a user requests a transaction involving a cryptoasset, such as a withdrawal or transfer of cryptocurrency funds, the CCS causes an endorsement request message to be sent to each of multiple user devices, each of which is associated with a different user who has been defined as potential member of a quorum for transactions involving that cryptoasset (in other embodiments, multiple users may share the same user device). The endorsement request message is configured to cause each receiving user device to prompt its user to provide an endorsement of the requested transaction. An endorsement in this context is an approval or rejection of an operation by a user. When a user receiving such a prompt endorses the transaction on his or her user device (e.g., a smartphone, tablet or notebook computer), the user device signs an endorsement message with a private key of that user and transmits the signed endorsement message to the CCS. The private key is stored within a secure enclave within the user device. A secure enclave in each user device is used to store the corresponding user's private key and to generate digital signatures of that user.

[0017] The HSM determines whether a policy-based quorum of multiple users has endorsed (approved) a requested action, such as a withdrawal or transfer of

cryptocurrency funds. It does this by validating the signature by a public key of a public-private key pair for each of the plurality of users, in endorsement messages received from the users. Only after determining that the policy-based quorum of the multiple users has validly endorsed the requested action, the HSM then allows itself to access the private key of that particular cryptographic asset (e.g., for a specific deposit of cryptocurrency funds), which the HSM previously generated, and uses that private key to sign the transaction as authorization that the transaction may proceed. The private key for that cryptoasset is stored only in the HSM, which does not permit the key to be read by any entity outside the HSM. Approval of the transaction may include, for example, transmitting the transaction onto a known blockchain network. In certain embodiments, approval of the transaction by the HSM occurs only if and after the requested transaction has passed a risk review, which may be partially or fully automated. Other details will become apparent from the description that follows. Note also that it is contemplated the system and techniques introduced here can be used for secure custody of other types of digital assets besides cryptoassets.

[0018] Refer now to Figure 1, which shows a high-level block diagram of the CCS. In the illustrated embodiment, the CCS 1 includes an online server 2, a relay server 3, a risk analysis stage 4, the HSM 5, and a data storage facility 6. The data storage facility 6 may include one or more databases, which can be or include relational databases or any other type of mechanism for storing data in an organized way, where the data may be structured data and/or unstructured data. The HSM 5 also includes its own internal secure storage facility 7. Note that there can be multiple instances of each of the above-mentioned components in the CCS 1, even though only one of each is shown to simplify description. One or more user devices 8, also called clients, can communicate with the CCS 1 via a public computer network 9, such as the Internet. Each of the user devices may be any one of, for example, a smartphone, tablet computer, laptop computer, desktop computer, or the like. Each user device 8 may include a secure enclave 14, such as an iOS-based secure enclave, which is used to store the corresponding user's private key and to generate digital signatures of that user. In at least some embodiments, each user device 8 is associated with a different user, and this description henceforth assumes

such an embodiment to facilitate description. Note, however, that it is possible to have embodiments in which multiple users share the same user device 8.

[0019] The relay server 3 functions as a virtual air gap to isolate the HSM 5 from the public computer network 9. The relay server 3 and HSM 5 operate within a secure zone 10. The HSM 5 may physically reside in a physically secured datacenter with no direct access to any outside network. Messages between the HSM 5 and the online server 2 are routed on a half-duplex (outbound request-responses only) connection to the relay server 3 in the secure zone 10. The relay server 3 disconnects itself from the secure network while communicating with the online server 2, and disconnects itself from all external networks while communicating with the HSM 5, such that no interactive sessions with those devices can be established from the outside. This provides virtual “air gap” security to critical infrastructure.

[0020] In certain embodiments, the CCS 1 also has access to at least one blockchain network 11 corresponding to cryptoassets of which the CCS 1 has custody. Access to the blockchain network 11 may be via the public computer network 9, e.g., the Internet.

[0021] In some embodiments, each transaction submitted by a customer of the CCS 1 will go through the risk analysis stage 4, which may be partially or fully automated. For example, with some embodiments of the CCS 1, a human risk analysis agent may evaluate the output of automated risk analysis software displayed on a risk review dashboard, to make a decision on whether a transaction has been sufficiently authorized to be accepted. The risk analysis agent or the software can follow a policy set on each individual vault and can look at any of various risk signals (e.g., the amount being transacted, how many users have authorized this transaction, the location(s) from which the transaction was requested and approved, the destination address) to compute a final risk score that might lead to the transaction being approved or more information being requested.

Deposits

[0022] Refer now to Figures 2A and 2B, which illustrate an example of the process of depositing a cryptoasset, such as an amount of cryptocurrencies, with the CCS 1. Deposits are initiated by a customer via the Internet through a software

application (hereinafter “the CCS application”)(not shown) executing on a user device 8 of the customer. This can be done by the customer’s selecting an asset type and requesting a deposit for a given amount in the CCS application. Once initiated, the request for a blockchain deposit address is then sent to the online server 2, which receives the request (step 201) and forwards it (step 202) via the relay server 3 to the HSM 5 (which as noted above is isolated from the Internet by the relay server 3). The HSM 5 then generates (step 203) a new public-private key pair 21 to correspond uniquely with the deposit, i.e., to correspond with the requested blockchain address. In certain embodiments, the HSM 5 uses the private key of the relevant Organization and a key derivation function (KDF) to generate the new key pair for the blockchain address. An “Organization” in this context is a data structure that corresponds to a particular customer, as discussed further below. The private key of the newly generated key pair cannot be extracted from the HSM 5, but can be backed up securely in an encrypted file. Key generation inside the HSM 5 ensures that the private keys 21 only exist within the HSM 5, are not available anywhere else in the world and cannot be accessed by any entity that is external to the HSM 5.

[0023] The HSM 5 next generates (step 204) the blockchain address for the deposit from the public key of the newly-created key pair. This can be done by using a well-known blockchain-specific transformation of the public key of the blockchain address. The HSM 5 then signs (step 205) the blockchain address with the Organization’s private key and returns the signed blockchain address to the online server 2. The online server 2 then causes (step 206) the signed blockchain address 22 to be sent to the customer’s user device 8, to cause the user device 8 to present the address to the customer in the CCS application on a user device, in an easy-to-consume format (e.g., as a QR code), for use as a destination address in a blockchain transaction. The CCS application on the user device verifies (step 207) the signature of the address before presenting the address to customer.

[0024] The customer’s user device 8 uses the public key of the Organization (which it previously received from the CCS 1 and locally stored) to verify the authenticity of the blockchain address it receives from the CCS 1. The customer then initiates (step 208) a transaction to deposit assets into the CCS 1. The transaction might be initiated from an exchange, from the customer’s personal wallet,

or from another cryptoasset store. No confirmation is required for the assets to show up in the CCS 1.

[0025] The address of the deposit is stored in a collection with other addresses belonging to the customer in the CCS 1, known as the customer's "vault." A vault in this context is a data entity that contains assets and a policy map containing one or more policies governing deposits and withdrawals from those assets. A cryptoasset is represented as a slot inside a vault that can hold an amount of an asset type (e.g., Bitcoin, Ethereum). Once under custody and stored with the CCS 1, an asset is completely under the control of the CCS 1.

[0026] The online server 2 determines whether the customer has confirmed the transaction within the defined time period (steps 209, 210). Once the deposit transaction is confirmed by customer and confirmed on the blockchain, the customer is so notified (step 211) by the online server 2, and the assets are considered to be under custody of the CCS 1. In the event confirmation is not received within the defined time period, the online server notifies (step 212) the customer of an error in the transaction.

Withdrawals

[0027] Figures 3A and 3B illustrate an example of the process of withdrawing an amount of a previously deposited cryptoasset, such as cryptocurrency. Withdrawals can be initiated from the CCS application on a user device 8A by selecting a specific cryptoasset to withdraw and an amount. Once initiated, all authorizing parties are made aware of the withdrawal request and are required to authorize it individually on their mobile devices 8A and 8B.

[0028] During this process users are required to review the transaction and approve it, where each user's approval is subject to biometric authentication (e.g., fingerprint, facial recognition and/or voice recognition). In certain embodiments, before a withdrawal can successfully move on to the next phase, every request is sent to the risk analysis stage to be inspected for suspicious activity and authorized as legitimate. The HSM 5 validates that a defined quorum (e.g., a majority) of users authorized the transaction, and that the transaction was approved by the risk review stage 4. For example, for a given corporate customer that has five distinct employees who require the ability to transfer funds, a suitable quorum configuration

might be to have a group of three of those five employees be necessary to move any funds. The HSM 1 then proceeds with the signature and submission of the asset-moving transaction to the blockchain 11.

[0029] An example of the withdrawal process is further illustrated in Figure 3B. The online server 2 initially receives (step 301) the withdrawal request 31 from the customer. The online server 2 then checks (step 302) the approval policy for the cryptoasset that is the subject of the transaction, as indicated in the vault of the cryptoasset, to determine which individuals' authorizations (endorsements) may be used to satisfy a quorum to approve the withdrawal. The online server 2 then sends (step 303) endorsement requests to the mobile devices 8A, 8B of those individuals (the mobile devices have been previously registered with the CCS 1). In response to these requests, one or more endorsement messages may be received from users' mobile devices 8A, 8B, where the endorsement messages were signed locally by the users' respective private keys stored securely in their respective mobile devices and subjected to one or more biometric authentication techniques, as described further below. Accordingly, the online server 2 determines (step 304) whether, within a timeout period, a quorum of authorizations have been received and the corresponding authorizing parties have been authenticated, as specified in the policy for this cryptoasset. If so, the online server 2 passes (step 305) the transaction request 31 to the risk analysis stage 4. Otherwise, the online server sends (step 310) a transaction denial notification to at least the user who requested the transaction (and possibly to all other users identified in the policy for this cryptoasset).

[0030] The risk analysis stage 4 performs a risk analysis (step 306), which as noted above may be fully or partially automated, or in some embodiments may be performed entirely by one or more human beings (based on computer output data). If the transaction passes risk analysis (step 306), then control flow is passed to the HSM 5, which verifies (step 308) that the quorum requirement has been satisfied, by performing the same determination as step 304 or a similar determination, as does the risk analysis stage 4 (step 306) (described further below). If satisfaction of the quorum is verified by the HSM 5, the HSM signs the withdrawal transaction with the private key of the blockchain address and submits the transaction onto the blockchain 11 to execute the withdrawal (step 309). Otherwise, the HSM 5 signals a

failure to the online server 2, which in response sends (step 310) a transaction denial notification to at least the user who requested the transaction (and possibly to all other users identified in the policy for this cryptoasset).

User Authentication

[0031] As mentioned above, when a user endorses a transaction request, they are subjected to one or more forms of authentication by their mobile device and/or the CCS 1, to establish that they are the expected person taking the action. These authentication forms may include one or more biometric authentication techniques, such as fingerprint verification, voiceprint verification, speech recognition, facial recognition and/or gesture recognition. The user's mobile device (e.g., smartphone) may perform one or more of these authentication techniques.

[0032] Additionally, or alternatively, the user may be required to upload to the CCS 1 a video, captured by their mobile device, from which their identity can be proven by, for example: identifying the user's face in the video against images of known faces (e.g., previous videos of the user); identifying the user's voice in the video against their trained voice profile; requiring the user to say certain words or take certain actions in the video based on the transaction (see further discussion below); requiring the user to make a previously specified gesture, or a distress gesture if they are in distress; requiring the user to identify on video the expected room they are in; and/or other performing any other actions that are considered to increase the level of confidence that the user is who he or she purports to be.

[0033] When determined to be necessary, a user may be asked to complete challenges to authenticate that he or she is in fact the person who is authorized to act on the transaction. These challenges may be generated deterministically based on the context of the transaction. For example, based on critical information in a transaction such as the ID, amount, destination, etc., the CCS 1 may generate a random number that can be used to select a few (e.g., three to five) words from a set of known words. The CCS 1 may present those words to the user and have the user speak them in a video captured by the user's mobile device, which the user's mobile device then transmits to the CCS 1. When reviewing the transaction, the reviewing mechanism or a human reviewer can independently generate the expected words based on transaction data and verify that the user spoke those words. The video

can also be subject to facial and/or voice recognition. By performing this type of deterministic challenge generation, an attacker can be prevented from faking a transaction by capturing and reusing previously transmitted authentication videos from the user.

HSM Logic

[0034] The main role of the HSM 5 is to verify the validity of operations. The HSM 5 carries out the will of the signers and authenticates that the signers are the authorized parties of an operation through the HSM's privileged access to keys. Keys needed for signing transactions are stored securely in the HSM 5 and never leave it. In some embodiments, the HSM 5 enforces these policies through a Secure Execution Environment (SEE) that runs code that cannot be changed except through physical access to the HSM 5 and requires a set of smartcards held securely by multiple employees of the Cryptoasset Custodian.

[0035] In certain embodiments, to facilitate the above-mentioned functionality the HSM 5 stores, in its internal storage 7, multiple instances of a data structure called "Organization," i.e., one instance for each customer of the Cryptoasset Custodian. The Organization data structure may contain the following fields: an identifier (ID) of the organization, a name of the organization, a public key of the organization, a list of users who belong to the organization, a policy map, a list of vaults that belong to the organization and their respective policy maps, and a generation number that is incremented each time the organization structure is updated. A "policy map" is a set of policies, including one policy for each possible action that may be carried out (e.g., add user, change vault policy, etc.). An Organization is signed by the HSM, using the Organization's private key (which is stored in the HSM 5 and cannot be read by any external entity), to indicate that it was produced through a valid set of changes authorized by the users and risk reviewers. The HSM keeps track of the most recent version to prevent rollback attacks.

[0036] To onboard a new customer, the HSM 5 creates a new Organization instance. To help ensure adequate security, the HSM 5 may create the Organization with the requested set of users already in it. In some embodiments, the HSM 5 must generate new unique keys for every new Organization created this way. This

prevents an attacker from asking the HSM 5 to generate a “new” organization that has the same ID as an existing one and tricking users into trusting it instead.

[0037] Figure 4 illustrates an example of a process that may be performed by the HSM 5, in at least some embodiments, in response to a request to carry out an operation. The request may be received by the HSM 5 from the relay server 3. Initially, the HSM 5 receives (step 401) from the relay server 3 an operation description, which specifies an Organization. The operation description is a set of data and metadata describing a requested operation, such as a requested deposit, withdrawal or transfer of cryptocurrency. The HSM 5 then verifies (step 402) the integrity of the specified Organization.

[0038] The HSM 5 then looks up the policy in the Organization’s or the vault’s policy map (step 403). It then looks at the policy for internal risk reviewers to determine which and how many internal risk endorsements (i.e., endorsements by personnel of the Cryptoasset Custodian) must be fulfilled (step 404). Next, the HSM 5 determines (step 405) whether any of the received endorsements (from users) indicates to “REJECT” the requested operation. If so, the HSM 5 rejects (step 411) the requested operation, by returning a “REJECT” message to the relay server, which then returns a corresponding “REJECT” message to the online server, to cause notification to the requester. In this case, the HSM 5 does not bother checking the signatures and just rejects the operation.

[0039] The HSM 5 then determines (step 406) whether all of the received endorsements for the transaction are valid. This includes verifying the validity of the endorsements provided by checking that: i) the user is in the Organization, ii) the signature is correct for the specified operation, and iii) each of the signatures has an “APPROVE” decision. If not all of the received endorsements for the transaction are valid, the process proceeds to step 411 as described above.

[0040] If all received endorsements for the transaction are valid, the HSM 5 then determines (step 407) whether the endorsements satisfy the relevant policy for the subject cryptoasset (i.e., satisfy the specified quorum). If the valid endorsements do not satisfy the policy, the process proceeds to step 411 as described above. If the endorsements satisfy the policy, then the HSM 5 determines (step 408) whether the requested operation passed the risk analysis stage. If not, the process proceeds to step 411 as described above. If the requested operation passed the risk analysis

stage, the HSM 5 determines (step 409) whether the requested operation is valid. This step can include verifying that the operation is internally consistent and that the operation can be applied to the Organization, vault or asset that it targets. If the requested operation is not valid, the process proceeds to step 411 as described above. Otherwise, the HSM 5 executes (step 410) the requested operation (or triggers an action to cause it to be executed). An operation to change the Organization, vault or policy results in a new signed Organization data structure with a higher generation value and the change applied to it. An operation to withdraw assets results in the HSM 5 signing a blockchain transaction with the private key corresponding to the subject asset. An operation to deposit assets results in the HSM 5 generating a deposit address.

Offline Device Endorsements

[0041] As a method for reducing the risk for users interacting with the CCS application on their personal devices, the CCS 1 may require authorization from an offline device. This device, such as a consumer phone with secure enclave or similarly capable computing device such as an iPod Touch, will be completely disconnected from the Internet in its normal state, and used in an offline manner to sign transactions required for authorization.

[0042] The process may be carried out as follows. The user has a phone or similar device that is a member of his or her vault policy's quorum and is not connected to any wireless or cellular networks. The device runs software similar to the CCS application software for enabling a user to endorse requested transactions, or the same software operating in a different mode. The user initiates a transaction against his or her vault through a different device in the quorum. An online device, such as another phone or web browser, has access to the transaction. It may be another phone/secure device in the quorum or it may exist solely for the purpose of displaying transactions. The device has the ability to transmit data that is required to be signed by the offline device, to the offline device. This can be done through a channel that cannot be accessed over the Internet, such as displaying a QR code, playing a sound or sequence of sounds that encodes data, or transmitting over Bluetooth. The offline device displays the data that was transmitted for it to sign, for the user's approval or rejection. The offline device signs its endorsement of the

operation based on the user's desired action. The offline device communicates its signed payload back to the online device in a similar manner to how it was received (e.g., displaying a QR code, playing a sound or sequence of sounds that encodes data, or transmitting over Bluetooth). The online device communicates the signed decision payload back to the online server of the CCS 1.

[0043] Figure 5 is a flow diagram that further illustrates this process, according to certain embodiments. An online user device receives (step 501) an operation description from the CCS via the Internet. The online user device then transmits (step 502) the operation description (or a portion thereof) to the offline user device via an offline channel. As noted above, the offline channel is a channel that is not accessible via the Internet, such as a local visual display by the online user device, a sound or sequence of sounds generated by the online user device, or a short range wireless transmission from the online user device (e.g., via Bluetooth). The offline user device receives the operation description (step 503) from the online user device via the offline channel, and based on the information thereby received, displays the operation description (or portion thereof) and prompts the user for endorsement of the operation (step 504). If a valid endorsement is received by the offline device (step 505) as user input within a timeout period, the offline device transmits an "ACCEPT" message (step 506) to the online user device via the same offline channel by which it received the operation description, or via a different offline channel. The online user device then receives the results of the endorsement from the offline device (step 507) and transmits the result payload to the CCS via the Internet (step 509). If a valid endorsement is not received by the offline user device from the user within the timeout period (step 505), the offline user device transmits a "REJECT" message to the online user device via the offline channel, which in turn transmits the "REJECT" payload to the CCS via the Internet (step 508).

[0044] The offline device may be delivered to the user with its secure key pre-enrolled in the Organization, or it may be allowed to be online for the initial enrollment process, or it may send its enrollment through a similar procedure to the authorization process.

[0045] The CCS software on the offline device may need to be updated periodically. To allow such updates, the offline device may be scheduled to connect to the Internet via Wi-Fi and have its software updated at a predefined cadence, or it

may detect that its software needs to be updated as a result of receiving a transaction to sign from the online user device, that indicates that the version of the software on the offline device is no longer compatible. Whenever the device is online, it can record as well as attempt to transmit to the CCS 1 the fact that it can access the Internet so that that information may be used to assess risk by the platform at a later time.

[0046] In addition to being kept offline, the offline user device and one or more online devices may be restricted to act on a transaction only when in range of a predefined beacon. A wireless (e.g., Bluetooth) beacon device can be made available to the user, and the CCS application may refuse to authorize transactions unless it detects that the beacon is available.

Auditability and Proof of Ownership

[0047] Every transaction submitted to the CCS 1 is recorded in an internal ledger that is tamper-resistant and that allows auditors to have cryptographic proof of every historical event on every user's account. The ownership of a blockchain asset is controlled by the possession of the private key corresponding to the public wallet address. The CCS can prove ownership of these assets to auditors by making use of the private key corresponding to a user's vault to sign a string of randomly chosen text chosen by the auditors. Consider the following example:

[0048] An auditor wishes to see proof that the CCS has access to funds in wallet identified by the address, "1BvBMSEYstn5Au4m4GFg7yJaNVN2." The auditor therefore randomly generates a long string, e.g., "xGG8vQFnd8QDwHz6Uj1GX," and submits the following challenge:

```
{
    Address: 1BvBMSEYstn5Au4m4GFg7yJaNVN2 ,
    Token: " AUDIT-CHALLENGE- xGG8vQFnd8QDwHz6Uj1GX",
}
```

[0049] The CCS 1 receives the challenge and forwards it to the HSM 5 as a predefined templated serialized package. The HSM 5 is programmed to accept and sign such audit requests (which are not arbitrary payloads and therefore are not at risk of being later interpreted as a signed blockchain transaction) with the private key associated with the specified address. The CCS 1 then returns the valid signature

for the challenge that can be independently verified by the auditor. This verification proves that the CCS 1 has control over a private key associated with an entry on the blockchain, achieving proof of control of the asset.

Thresholding Service

[0050] In certain embodiments, the CCS 1 includes a Thresholding Service that enables other parts of the system (Risk Analysis stage 4 and HSM 5) to securely determine that user operations and transactions have followed the customer specific business logic and have been approved by a human/automated risk review system. The Thresholding Service can verify multi-signature (multi-user) quorums to achieve this.

[0051] The Thresholding Service validates operations initiated and approved by users to ensure that they've met a threshold quorum before being executed. Such operations may include transactions, adding or removing other users, etc. Different users can have different access control roles (e.g., view-only, initiate-transaction-only, authorizable, necessary). The CCS 1 is able to notify every reportable status of the quorum acceptance lifecycle, but is not able to sign-off on operations that have not been authorized by customers. All actions are logged in an append-only ledger for auditability over all account interactions.

[0052] One function of the Thresholding Service is to verify that a quorum of authorized users have signed-off on a requested operation. Qualifying operations that may require a quorum may include, for example, proposing a transaction (e.g., "withdraw 100 Bitcoin"), adding a user to an account, changing a user's permissions, removing a user from an account, and changing the thresholding logic. A quorum may be defined as an absolute majority of users by default (e.g., 3 out of 5), or it may be set to a custom quorum upon onboarding of the customer. Moreover, an authorized user can configure a quorum to require certain specific users to endorse a transaction to constitute a quorum. The CCS 1 may also allow thresholding across multiple required groups. For example, in a company a majority of the finance team may be required to sign off, as well as the front office.

[0053] In certain embodiments, the Thresholding Service implements a fine-grained access control model in its quorum verification, in which different users can have different access levels, which may include the following levels, for example:

- View-only
 - This is the default access level
 - Users in this level can view all asset positions
 - Users in this level can flag any transaction
 - Users in this level can freeze all assets
- View-authorize
 - Users in this level can act as an authorizing vote for an action toward a quorum
 - Users in this level can view all asset positions
 - Users in this level can flag any transaction
 - Users in this level can freeze all assets
- View-authorize-necessary
 - Users in this level are a *required* vote for an action
 - Users in this level can view all asset positions
 - Users in this level can flag any transaction
 - Users in this level can freeze all assets

[0054] In certain embodiments, the access level for a user can only be changed with an appropriately verified quorum that is verified by the Thresholding Service.

[0055] As noted above, user approvals for an action can be expressed by a cryptographic digital signature, to benefit from non-repudiation guarantees. The Cryptoasset Custodian can be certain that the associated user who holds the private key was indeed the user who approved the action, since digital signatures cannot be forged. In certain embodiments, a user's signature is generated from an iOS secure enclave in the user's mobile device, and forwarded to the CCS 1 by the iOS application programming interface (API) component in the user device 8. Signatures can be performed over the cryptographic hash of the transaction contents to ensure that the transaction cannot be tampered with. All users may be required to sign the same hash for the same transaction identifier (ID) in order for the signatures to count toward the quorum. The Thresholding Service can provide templates for the clients to sign, and can verify all completed signatures completed by the iOS client. In at least some embodiments, the Thresholding Service verifies signatures with the

public components of the users' signing keys, but does not hold the private components of those user signing keys.

[0056] Once a threshold has been satisfied, the Thresholding Service will publish the corresponding signature data to the Risk Analysis stage to be further analyzed before sign-off by the Risk Analysis stage, and will serialize the signature data into a payload to be consumed by the HSM signing service. Each additional signature provided to the Thresholding Service and verification can be recorded in the append-only log service. This will provide additional auditing and status updates in addition to the metadata captured in the Thresholding Service's storage, which will be essential for providing consumable updates to user clients.

Maintaining Quorum Liveness

[0057] It is assumed that authorized members of a quorum are available to cryptographically sign transactions. Therefore, the quorum should be kept "live" -- that is, at any given time, the CCS 1 has reasonable confidence that all potential members of the quorum maintain possession of their secure device keys and can actively participate in a transaction. In certain embodiments, the CCS 1 can do the following to achieve this level of confidence:

1. Have access to the set of user public keys required to fulfill a policy's quorums.
2. Set a liveness threshold for a policy, i.e., the amount of time after which one considers a key to be at risk of unavailability. This can be fixed or related to normal transaction cadence.
3. Require users to periodically sign a proof transaction with their private keys. This can be explicit as a liveness check or hidden/implicit by requiring their key for routine operations such as login.
4. Record the latest live time of any one or more users' keys.
5. Continuously monitor whether any user's live time has exceeded the liveness threshold.
6. Use the above information to prompt the user to prove they still have access to their signing key and/or inform other users that the quorum may be at risk.

Risk Analysis Stage

[0058] The Risk Analysis stage 4 can implement an API, called the Risk API, and can further include human review of all transactions and administrative user operations. In some embodiments the Risk API drives the human review system. The Risk API can provide integration with an internal risk dashboard, for human employees of the Cryptoasset Custodian to manually review each transaction.

[0059] In certain embodiments, all transactions are manually approved by designated employee(s); all administrative user operations (adding, removing, permission changes) are manually approved by designated Cryptoasset Custodian employee(s); reviewable entities must have passed an automated verification process before requiring risk analysis; reviewable entities must provide robust context about the user approvals, for both human and further automated inspection; and risk approvals and denials are logged in an append-only ledger for auditability.

[0060] The Risk API re verifies the appropriate threshold as determined by the Thresholding Service. The Risk API may also handle additional business logic, such as in embodiments where the Thresholding Service is simplified: for example, the Risk API could check for necessary signers if the Thresholding Service only checks for quorums. Other functions described herein can also be moved between modules.

[0061] The Risk API can receive contextual data about each user involved in a transaction, to present to a human and/or classification system. This information may include, for example, user(s) who approved the transaction, time of approval(s), location of approval(s), and device/key ID(s) that approved the transaction. This data can be fed into an internal Risk Analysis dashboard, and possibly other automated review systems.

[0062] In some embodiments, the Risk API requires human approval from one or more employees of the Cryptoasset Custodian if a transaction passes the manual and automated risk review. To approve, an employee may be required to sign with a cryptographic key if he or she approves the transaction/operation and present the signature to the Risk API for validation. Moreover, there are preferably multiple keys, one per risk reviewer, such that it is logged who performed the review. Preferably it is made easy to rotate a risk-approval key in case of compromise.

[0063] Figure 6 shows a high-level example of a hardware architecture of a processing system that can be used to implement some or all of the CCS, or (separately) any user device, or both. The CCS can include one or more instances of an architecture such as shown in Figure 6, where multiple such instances can be coupled to each other via one or more private networks.

[0064] The illustrated processing system 600 includes one or more processors, including a CPU 610, one or more memories 611 (at least a portion of which may be used as working memory, e.g., random access memory (RAM)), one or more data communication device(s) 612, one or more input/output (I/O) devices 613, and one or more mass storage devices 614, all coupled to each other through an interconnect 615. The interconnect 615 may be or include one or more conductive traces, buses, point-to-point connections, controllers, adapters and/or other conventional connection devices. Each processor 610 controls part of the operation of the processing device 600 and can be or include, for example, one or more general-purpose programmable microprocessors, digital signal processors (DSPs), mobile application processors, microcontrollers, application specific integrated circuits (ASICs), programmable gate arrays (PGAs), or the like, or a combination of such devices.

[0065] Each memory 611 can be or include one or more physical storage devices, which may be in the form of RAM, read-only memory (ROM) (which may be erasable and programmable), flash memory, miniature hard disk drive, or other suitable type of storage device, or a combination of such devices. Each mass storage device 614 can be or include one or more hard drives, digital versatile disks (DVDs), flash memories, or the like. Each memory 611 and/or mass storage 614 can store (individually or collectively) data and instructions that configure the processor(s) 610 to execute operations to implement the techniques described above. Each communication device 612 may be or include, for example, an Ethernet adapter, cable modem, Wi-Fi adapter, cellular transceiver, baseband processor, Bluetooth or Bluetooth Low Energy (BLE) transceiver, or the like, or a combination thereof. Depending on the specific nature and purpose of the processing system 600, each I/O device 613 can be or include a device such as a display (which may include a transparent AR display surface), audio speaker, keyboard, mouse or other pointing device, microphone, camera, etc. Note, however,

that such I/O devices may be unnecessary if the processing device 600 is embodied solely as a server computer.

[0066] In the case of a user device, a communication device 612 can be or include, for example, a cellular telecommunications transceiver (e.g., 3G, LTE/4G, 5G), Wi-Fi transceiver, baseband processor, Bluetooth or BLE transceiver, or the like, or a combination thereof. In the case of a server, a communication device 612 can be or include, for example, any of the aforementioned types of communication devices, a wired Ethernet adapter, cable modem, DSL modem, or the like, or a combination of such devices.

[0067] Unless contrary to physical possibility, it is envisioned that (i) the methods/steps described herein may be performed in any sequence and/or in any combination, and that (ii) the components of respective embodiments may be combined in any manner.

[0068] The machine-implemented operations described above can be implemented by programmable circuitry programmed/configured by software and/or firmware, or entirely by special-purpose ("hardwired") circuitry, or by a combination of such forms. Such special-purpose circuitry (if any) can be in the form of, for example, one or more application-specific integrated circuits (ASICs), programmable logic devices (PLDs), field-programmable gate arrays (FPGAs), system-on-a-chip systems (SOCs), etc.

[0069] Software or firmware to implement the techniques introduced here may be stored on a machine-readable storage medium and may be executed by one or more general-purpose or special-purpose programmable microprocessors. A "machine-readable medium", as the term is used herein, includes any mechanism that can store information in a form accessible by a machine (a machine may be, for example, a computer, network device, cellular phone, personal digital assistant (PDA), manufacturing tool, any device with one or more processors, etc.). For example, a machine-accessible medium includes recordable/non-recordable media (e.g., RAM or ROM; magnetic disk storage media; optical storage media; flash memory devices; etc.), etc.

[0070] The term "logic", as used herein, means: i) special-purpose hardwired circuitry, such as one or more application-specific integrated circuits (ASICs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), or

other similar device(s); ii) programmable circuitry programmed with software and/or firmware, such as one or more programmed general-purpose microprocessors, digital signal processors (DSPs) and/or microcontrollers, system-on-a-chip systems (SOCs), or other similar device(s); or iii) a combination of the forms mentioned in i) and ii).

[0071] Any or all of the features and functions described above can be combined with each other, except to the extent it may be otherwise stated above or to the extent that any such embodiments may be incompatible by virtue of their function or structure, as will be apparent to persons of ordinary skill in the art. Unless contrary to physical possibility, it is envisioned that (i) the methods/steps described herein may be performed in any sequence and/or in any combination, and that (ii) the components of respective embodiments may be combined in any manner.

[0072] Although the subject matter has been described in language specific to structural features and/or acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as examples of implementing the claims and other equivalent features and acts are intended to be within the scope of the claims.