I834741

(19)中華民國智慧財產局

(12)發明說明書公告本 **(11)證書號數：TW I834741 B**

(45)公告日： 中華民國 113 (2024) 年 03 月 11 日

(21)申請案號：108137096 (22)申請日： 中華民國 108 (2019) 年 10 月 15 日

(51)Int. Cl.： ***G06F21/62 (2013.01)*** ***H04L9/30 (2006.01)***

(30)優先權：2018/10/17 英國 1816936.7

(71)申請人：安地卡及巴布達商區塊鏈控股有限公司 (安地卡及巴布達) NCHAIN HOLDINGS LIMITED (AG)
安地卡及巴布達

(72)發明人：萊特 克瑞格 S WRIGHT, CRAIG STEVEN (AU)；沃恩 歐文 VAUGHAN, OWEN (GB)

(74)代理人：劉法正；尹重君

(56)參考文獻：

TW 200731739A TW 201328279A

CN 108154549A

期刊 GREGORY MAXWELL Simple Schnorr Multi-Signatures with Applications to Bitcoin DESIGNS, CODES AND CRYPTOGRAPHY 87/9 20180115

審查人員：彭智煇

申請專利範圍項數：7 項 圖式數：7 共 30 頁

(54)名稱

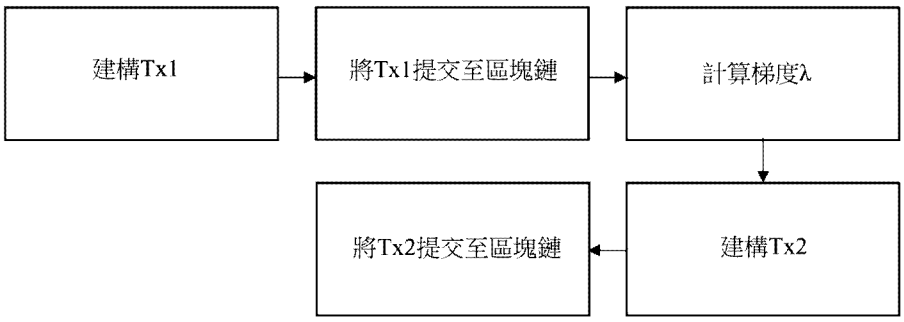
包括公鑰組合驗證之電腦實行系統及方法

(57)摘要

揭露一種電腦實行方法。該方法包括提供包含一公鑰組合驗證函數之一區塊鏈交易。該區塊鏈交易經組配為可兌換的以藉由向該區塊鏈交易提供一輸入來准許對一資源之存取或轉移該資源之控制，該輸入包含：多個公鑰；一梯度值(λ)，其與該多個公鑰中之二者相關；及一群組公鑰，其源自該公鑰與該梯度值(λ)之一組合。該區塊鏈交易經組配以根據該交易之成功兌換而將該公鑰驗證函數應用於該輸入以驗證該群組公鑰係源自該多個公鑰之該組合。

A computer-implemented method is disclosed. The method includes providing a blockchain transaction comprising a public key combination verification function. The blockchain transaction is configured to be redeemable to permit access to, or transfer control of, a resource by providing to the blockchain transaction an input comprising: a plurality of public keys; a gradient value (λ) related to two of the plurality of public keys; and a group public key derived from a combination of the public keys and the gradient value (λ). The blockchain transaction is configured to apply the public key verification function to the input to verify, upon successful redemption of the transaction, that the group public key is derived from the combination of the plurality of public keys.

指定代表圖：



【圖6】

I834741

公告本

【發明摘要】

【中文發明名稱】

包括公鑰組合驗證之電腦實行系統及方法

【英文發明名稱】

COMPUTER-IMPLEMENTED SYSTEM AND METHOD INCLUDING PUBLIC KEY COMBINATION VERIFICATION

【中文】

揭露一種電腦實行方法。該方法包括提供包含一公鑰組合驗證函數之一區塊鏈交易。該區塊鏈交易經組配為可兌換的以藉由向該區塊鏈交易提供一輸入來准許對一資源之存取或轉移該資源之控制，該輸入包含：多個公鑰；一梯度值(λ)，其與該多個公鑰中之二者相關；及一群組公鑰，其源自該公鑰與該梯度值(λ)之一組合。該區塊鏈交易經組配以根據該交易之成功兌換而將該公鑰驗證函數應用於該輸入以驗證該群組公鑰係源自該多個公鑰之該組合。

【英文】

A computer-implemented method is disclosed. The method includes providing a blockchain transaction comprising a public key combination verification function. The blockchain transaction is configured to be redeemable to permit access to, or transfer control of, a resource by providing to the blockchain transaction an input comprising: a plurality of public keys; a gradient value (λ) related to two of the plurality of public keys; and a group public key derived from a combination of the public keys and the gradient value (λ). The blockchain transaction is configured to apply the public key verification function to the input to verify, upon successful redemption of the transaction, that the group public key is derived from the combination of the plurality of public keys.

【指定代表圖】　圖6

【代表圖之符號簡單說明】

(無)

【特徵化學式】

(無)

【發明說明書】

【中文發明名稱】

包括公鑰組合驗證之電腦實行系統及方法

【英文發明名稱】

COMPUTER-IMPLEMENTED SYSTEM AND METHOD INCLUDING PUBLIC KEY COMBINATION VERIFICATION

【技術領域】

【0001】發明領域

本揭露內容大體上係關於資源控制及/或存取之轉移，且更特定言之，係關於在區塊鏈上使用密碼編譯多重簽名法來轉移此控制及/或存取。本揭露內容尤其適合(但不限於)用於比特幣(Bitcoin)區塊鏈或比特幣協定之任何變體

【先前技術】

【0002】發明背景

在此文件中，吾人使用術語「區塊鏈」來包括所有形式的基於電腦之電子分散式分類賬。此等分類賬包括基於共識之區塊鏈及交易鏈技術、許可及未許可分類賬、共用分類賬及其變體。區塊鏈技術之最廣泛已知應用為比特幣分類賬，儘管已提議並開發了其他區塊鏈實施方案。儘管本文中出於便利及說明之目的可提及比特幣，但應注意，本揭露內容不限於與比特幣區塊鏈一起使用，且替代區塊鏈實施方案及協定屬本揭露內容之範疇。術語「使用者」在本文中可指人類或基於處理器之資源。

【0003】區塊鏈為同級間電子分類賬，其經實行為由區塊構成之基於電腦之去中心化分散式系統，該等區塊又由交易構成。每一交易為一資料結構，該資料結構編碼區塊鏈系統中之參與者之間的數位資產或資源(例如密碼貨幣或符記化項)之控制之轉移，且包括至少一個輸入及至少一個輸出。每一區塊含有先前區塊之散列，使得該等區塊變為鏈接在一起以產生自一開始就已寫入至區塊鏈之所有交易的永久性不可變更記錄。交易含有嵌入至其輸入及輸出中之已知為指令碼的小型程式，其指定可如何及由誰存取交易之輸出。在比特幣平台上，此等指令碼係使用基於堆疊之指令碼處理語言來撰寫。

【0004】為了將交易寫入至區塊鏈，交易必須經「確證」。網路節點(挖掘者(miner))執行工作以確保每一交易有效，其中無效交易被網路拒絕。安裝於節點上之軟體用戶端根據執行其鎖定及解除鎖定指令碼而對未用交易(unspent transaction，UTXO)執行此確證工作。若鎖定及解除鎖定指令碼之執行評估為 TRUE，則交易為有效的且將交易寫入至區塊鏈。因此，為將交易寫入至區塊鏈，該交易必須：i)由接收交易之第一節點進行確證，若交易經確證，則節點將該交易轉送至網路中之其他節點；且 ii)添加至由挖掘者建構之新區塊中；且 iii)經挖掘，亦即添加至過去交易之公用分類賬。

【0005】雖然區塊鏈技術由於密碼貨幣實施之使用而為最廣泛已知的，但數位企業家已開始探索比特幣所基

於之密碼安全系統及可儲存於區塊鏈上以實行新系統之資料二者的使用。若區塊鏈可用於不限於密碼貨幣範圍之自動任務及程序，則將非常有利。此等解決方案將夠利用區塊鏈的益處(例如，事件之永久性防篡改記錄、分散式處理等)，同時在其應用中變得更通用。

【0006】區塊鏈相關關注之另一領域係使用『符記(token)』(或『彩色幣』)來表示真實世界實體及經由區塊鏈轉移真實世界實體。潛在地敏感或秘密之項可由符記表示，該符記不具有可辨別之含義或值。符記因此充當允許自區塊鏈引用現實世界項之識別符。

【0007】區塊鏈交易可利用內置多重簽名協定來限制交易，使得 *N* 個總簽名中之 *M* 個需要呈現為兌換指令碼兌換交易之輸入。舉例而言，可使用五個多重簽名法中之三個鎖定交易，使得可僅藉由使用對應於該五個簽名中之任何三個的三個私鑰來解除鎖定交易。

【0008】對於 *N* 個方法中之 *M* 個，參考比特幣區塊鏈協定之變化中使用的命令，作業碼 OP_MULTISIG 採用 *N* 個公鑰及 *M* 個簽名作為輸入。在此實例中，*N* 個公鑰儲存於其自身兌換指令碼中。作業碼開始於第一簽名且檢索 *N* 個公鑰測試以查看該簽名是否藉由該公鑰產生。其丟棄與簽名不匹配的每一密鑰。出於此原因，簽名之階數(order)必須匹配提供公鑰之階數。以下給出多重簽名兌換指令碼之實例：

兌換指令碼
<PubKey 1><PubKey 2>…<PubKey *N*> OP_CHECKMULTISIG

對於解除鎖定，將需要呈現以下輸入：

輸入
<sig 1><Sig 2>…<Sig *M*>

【0009】根據上文顯而易見，兌換指令碼大小與所需簽名之數目 *M* 及參與者之數目 *N* 二者成線性比例。因此，隨著 *M* 及 *N* 增大，兌換交易中之兌換交易所需之簽名的數目增大，公鑰之數目亦如此。因此，傳播後續兌換交易所需之網路速度及儲存後續兌換交易所需之空間增加。此外，在合併兌換交易之解除鎖定指令碼與兌換指令碼時需要執行之操作之數目在 *N* 中且在 *M* 中線性地增大。

【0010】因此，需要提供一種用於減少與區塊鏈交易、利用多重簽名協定之此等區塊鏈交易且尤其需要儲存較大數目之公鑰之此等交易的傳輸、操縱及儲存相關聯的操作及儲存要求的解決方案。

【0011】現在已設計出此改良之解決方案。因此，根據本揭露內容，提供一種如隨附申請專利範圍中所定義之方法。

【發明內容】

【0012】發明概要

根據本揭露內容，提供一種電腦實行方法。其可描述為一種安全方法。

【0013】方法可包含以下步驟：提供包含一公鑰組合驗證函數之一區塊鏈交易，該區塊鏈交易經組配為可兌換的以藉由向該區塊鏈交易提供一輸入來准許對一資源之存取或轉移該資源之控制，該輸入包含多個公鑰、與該多個

公鑰中之各別二者相關之至少一個梯度值以及源自該多個公鑰與該至少一個梯度值之一組合的一群組公鑰，其中該區塊鏈交易經組配以根據該交易之成功兌換而將該公鑰驗證函數應用於該輸入以驗證該群組公鑰係源自該多個公鑰之該組合。

【0014】此方法提供一種以一安全、可靠且可公開驗證的方式轉移控制或由多個私鑰表示及/或控制之一資源或對該資源之存取的方法。

【0015】方法可包含以下步驟：自一已知值導出一導出公鑰；以及將該導出公鑰配置為次級公鑰之一序列，其中次級公鑰之該序列為該多個公鑰之一子集。

【0016】此使得一使用者能夠將方法之已知要求轉換為待用於方法中之公鑰，藉此提供改良方法之通用性之優勢。提供之其他優勢為可預先計算該序列之係數，藉此提高方法可藉以執行的速度。

【0017】方法可包含以下步驟：在將該導出公鑰配置為次級公鑰之一序列之前將一限制函數應用於該已知值，藉此限制該子集之大小。

【0018】此提供進一步提高方法之效率的優勢。

【0019】方法可包含以下步驟：至少部分地基於該多個中之一另外公鑰及該導出公鑰計算該多個中之一公鑰。

【0020】此提供以下優勢：鑒於該多個中之一或多者可源自一個儲存的公鑰，因此移除儲存給定多個公鑰之要求。

【0021】該區塊鏈交易可進一步包含：一群組簽名，其對應於該群組公鑰；一默克爾(Merkle)根；及一默克爾路徑驗證函數，其中藉由向該區塊鏈交易提供以下各者而將該區塊鏈交易組配為可兌換的：一默克爾路徑，其與該多個公鑰相關聯；及一群組私鑰，其對應於該群組簽名。

【0022】此方法需要更少處理步驟以實現安全性之給定位準，藉此提供提高方法之效率同時維持安全性之優勢。

【0023】本揭露內容亦提供一種系統，其包含：一處理器；及記憶體，該記憶體包括可執行指令，該等可執行指令因由該處理器執行而使得該系統執行本文中所描述的電腦實行方法之任何實施例。

【0024】本揭露內容亦提供一種非暫時性電腦可讀儲存媒體，其上儲存有可執行指令，該等可執行指令因由一電腦系統之一處理器執行而使得該電腦系統至少執行本文中所描述之該電腦實行方法之一實施例。

【圖式簡單說明】

【0025】本揭露內容之此等及其他態樣將自本文中所描述之實施例顯而易見且參考本文中所描述之實施例進行闡明。現將僅藉助於實例且參考附圖而描述本揭露內容之實施例，在附圖中：

圖 1 為說明用於未壓縮公鑰之資料編碼的表；

圖 2 為展示執行本揭露內容之一實施例之堆疊的演進的表；

圖 3 為展示執行本揭露內容之一實施例之堆疊之演進的表；

圖 4 為說明用於不同多重簽名法之 M、N 及指令碼大小之間的關係的表；

圖 5 示意性地說明體現本揭露內容之二個交易；

圖 6 為展示體現本揭露內容之步驟之序列的流程圖；且

圖 7 為說明各種實施例可實行於其中之計算環境的示意圖。

【實施方式】

【0026】較佳實施例之詳細說明

揭露驗證比特幣指令碼內二個橢圓曲線點相加之方法。給定二個點P_1, P_2及候選解P_3，所揭露方法在指令碼中驗證$P_3 = P_1 + P_2$。所揭露之另一方法為將前一方法擴展至一系列點相加。

【0027】方法可針對數目個參與者用於簽名法，諸如多重簽名法。方法可使用橢圓曲線數位簽名算法(Elliptic Curve Digital Signature Algorithm，ECDSA)協定。

【0028】質數階p之橢圓曲線由以下等式定義：$y^2 = x^3 + ax + b$。

【0029】根據曲線之組結構，可將曲線$P_1 = (x_1, y_1)$及$P_2 = (x_2, y_2)$上之二個點相加以得到第三點$P_3 = (x_3, y_3)$，亦即：$P_3 = P_1 + P_2$。

【0030】點P_3定義為橢圓曲線上的點，該點穿過連接

P_1及P_2隨後圍繞 x 軸反射的線。用於點相加之公式為：

$$x_3 = \lambda^2 - x_1 - x_2 \mod p$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \mod p，$$

其中，針對$P_1 \neq P_2$情況，梯度λ由以下給定：

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \mod p$$

且針對$P_1 = P_2$情況，梯度λ由以下給定：

$$\lambda = \frac{3x_1^2 + a}{2y_1} \mod p。$$

【0031】比特幣協定等使用 secp256k1 約定，其中橢圓曲線參數由$a = 0$及$b = 7$給定。

【0032】在區塊鏈交易之指令碼中應用擴展的歐幾里得(Euclidean)算法並不實用，此係因為其需要計算模數乘法反元(modular multiplicative inverses)，且因此其無法實行如指令碼中之上述等式所定義的橢圓曲線點相加。

【0033】在下文中，所揭露之方法為驗證二個點P_1及P_2之相加之解。解$P_3 = P_1 + P_2$本身之計算係在指令碼之外執行，同時在指令碼中驗證答案。此減少了原本會施加於區塊鏈節點上之計算要求。

【0034】為實現此驗證，需要解P_3及P_1與P_2之間的線之梯度之值，標示為λ。λ之計算在指令碼之外執行，但λ之驗證可在指令碼中執行。

【0035】下文中更詳細地定義的新公鑰驗證函數<Point Add P_1, P_2, λ, P_3>經組配以在其中所呈現之λ為P_1與

P_2之間的線之梯度的情況下且在$P_3 = P_1 + P_2$的情況下返回TRUE。此二個驗證直接在指令碼中實現，且一旦交易成功地兌換便發佈。

【0036】函數之執行包含以下步驟：

1. 藉由檢查以下來驗證λ之值：

a) 在$P_1 \neq P_2$情況下，吾等具有

$$\lambda(x_2 - x_1) \mod p = y_2 - y_1 \mod p$$

b) 在$P_1 = P_2$情況下，吾等具有

$$2\lambda y_1 \mod p = 3x_1^2 \mod p。$$

2. 藉由檢查$P_3 = (x_3, y_3)$之座標滿足以下等式來驗證P_3之值

$x_3 = \lambda^2 - x_1 - x_2 \mod p$ 且

$y_3 = \lambda(x_1 - x_3) - y_1 \mod p$。

【0037】藉由應用公鑰驗證函數之逐次迭代，可驗證任意數目n個公鑰之總和，其中唯一限制為正使用之區塊鏈協定之要求，諸如比特幣指令碼中之作業碼數目(201)或位元組中之大小(10000)。

【0038】上述公鑰驗證函數作用於三個公鑰以驗證該等公鑰中之二個合併以產生第三個。此可經擴展以驗證三個公鑰之相加$P = P_1 + P_2 + P_3$及超出三個公鑰之相加。標示為 PointAddMulti 之擴展函數在下文中經定義，其中驗證過程驗證 P1、P2 及 P3 合併以產生 P：

<Point Add Multi $P_1, P_2, P_3, \lambda', \lambda, P$>: = <Point Add P_1, P_2, λ', P'> <Point Add P', P_3, λ, P>

其中$P' = P_1 + P_2$，值λ'為P_1與P_2之間的梯度，且值λ為P'與P_3之間的梯度。

【0039】一般而言，亦即，給定合併以產生公鑰P之 n 個公鑰P_i，PointAddMulti 可如下定義：

<PointAddMulti P_i, λ_i, P> :=

<PointAdd $P_1, P_2, \lambda_1, P_1'$ ><PointAdd $P_1', P_3, \lambda_2, P_2'$ > … <PointAdd $P_{n-2}', P_n, \lambda_{n-1}, P$>

【0040】可使用下文中揭露之方法以比特幣指令碼處理語言來建構點相加函數 PointAdd。

【0041】在圖 1 中說明用於未壓縮公鑰之資料編碼。此處，資料之虛擬值獲自 A.安東諾普洛斯(A. Antonopoulos)的著名書《精通比特幣(Mastering Bitcoin)》，第 2 版，O'Reilly 媒體(O'Reilly Media)(2017)。

【0042】給定未壓縮公鑰 P，可使用運算符 OP_SPLIT 如下在比特幣指令碼中直接擷取 x 及 y 座標：

<P> OP_1 OP_SPLIT OP_NIP 32 OP_SPLIT = <x> <y>。

【0043】使用此運算，可自輸入P_3, λ, P_1, P_2擷取 x 及 y 座標：

<P_3> <λ> <P_1> <P_2> <x_3> <y_3> <λ> <x_1> <y_1> <x_2> <y_2>

【0044】應注意，可複製且重新配置上述之右側之項，以使用基本運算產生任何所需組合。

【0045】參考圖 2，函數 PointAdd 之結構中之第一步驟為驗證 λ 之輸入值。在其中$P_1 \neq P_2$的情況下，檢查以下等式是否成立：$\lambda(x_2 - x_1) \mod p = y_2 - y_1 \mod p$。此藉由將<$y_2$> <$y_1$> <$\lambda$> <$x_2$> <$x_1$>作為輸入且利用以下運算對此輸入進行操作來實現：

OP_SUB OP_MUL <p> OP_MOD OP_3 OP_ROLL OP_3 OP_SUB <p> OP_MOD OP_EQUAL

【0046】在且僅在滿足等式$\lambda(x_2 - x_1) \mod p = y_2 - y_1 \mod p$的情況下，其將返回 TRUE。圖 2 之表說明執行上述運算時堆疊之演進。

【0047】針對$P_1 = P_2$情況，可建構與上文所描述之運算類似之一組運算，其檢查以下等式是否成立：

$2\lambda y_1 \mod p = 3x_1^2 \mod p$

【0048】參考圖 3，函數 PointAdd 之結構中之第二步驟為驗證P_3之輸入值為$P_1 + P_2$之和。為進行此操作，檢查以下等式中之每一者是否成立：

$x_3 = \lambda^2 - x_1 - x_2 \mod p$，且

$y_3 = \lambda(x_1 - x_3) - y_1 \mod p$，

【0049】由於λ已經驗證，因此其可在以下計算中使用。為檢查上述二個等式中之第一個是否成立，將以下內容呈現為輸入：

<x_3> <λ> <x_1> <x_2>

且利用以下運算進行操作：

OP_3 OP_ROLL OP_DUP OP_MUL OP_SWAP

OP_SUB OP_SWAP <p> OP_MOD OP_EQUAL。

【0050】此處，p為橢圓曲線基礎字段之階數。

【0051】圖 3 之表說明執行上述運算時堆疊之演進。

【0052】可藉由將現有公鑰P_1添加至根據橢圓曲線產生器點G乘以數目$S \in \mathbb{Z}_n^*$而計算之公鑰來產生新公鑰P_3，因此：$P_3 = P_1 + S \cdot G$。

【0053】S可為用以自初始公鑰P_1創建導出公鑰P_3之確定性密鑰。

【0054】式$P_3 = P_1 + S \cdot G$之點相加之驗證可藉由驗證涉及產生器點G之固定倍數之點相加之序列來實現。

【0055】舉例而言，雙加(double-and-add)法可用以將點相加分解為級數(series)

$$P_3 = P_1 + s_0 G + s_1 2G + s_2 4G + s_3 8G + \cdots + s_{256} 2^{256} G$$

其中$s_0, \ldots, s_{256} \in \{0,1\}$為如下$S$之二元展開式中之係數：

$$S = s_0 + s_1 2 + s_2 4 + s_3 8 + \cdots + s_{256} 2^{256}。$$

【0056】藉由使產生器加倍而獲得之點$G, 2G, 4G, \ldots, 2^{256}G$為常識且可預先計算。此意謂存在可能之 255 個個別點相加，其需要驗證以獲得最終結果。

【0057】因此，可利用 255 + 1 = 256 點相加之最大值來驗證$P_3 = P_1 + S \cdot G$。

【0058】除了雙加法之外，可使用其他點相加法算法，諸如滑動窗口(sliding-window)或蒙哥馬利(Montgomery ladder)階梯法。

【0059】為使利用計算密鑰之點相加之計算更易於

管理，S可受限於較小範圍。範圍可為特定於應用程式的。

【0060】S可能需要在$\mathbb{Z}_n^*$內之 32 位範圍中取值，該$\mathbb{Z}_n^*$表示 43 億個值範圍。然而，僅需要 31 個個別點相加來驗證最終結果。舉例而言，在比特幣中，此可藉由以完整 256 位範圍中之S開始且隨後應用模數運算 OP_MOD 來將該範圍限制為 32 位來實現。隨機公鑰P_x可用以藉由獲取和$P_x + S \cdot G$來使範圍模糊。

【0061】舉例而言，二手車拍賣可具有 16,000 USD 之最高出價。呈數目 16000 之形式的t USD 之出價可添加至公鑰P_1中以得到附加密鑰

$$P = P_1 + t \cdot G,$$

其可藉由指令碼中之 14 點相加來驗證。

【0062】為瞭解可如何將其用於出價過程，假設一車輛之當前最高出價為 9,000 USD，該出價由特定 UTXO 追蹤。此 UXTO 具有兌換指令碼：

```
IF t > 9000
  <Point Add Multi P1, t · G, P>
  <CheckSig P>
ELSE
  OP_RETURN
```

【0063】需要以下作為輸入以用於解除鎖定：

```
<Sig P> <P> <Verification data> <P1> <t>
```

【0064】上文所描述之點相加驗證方法可隨後用以驗證$P = P_1 + t \cdot G$。應注意，此涉及 14 個個別點相加之驗證。用於此等驗證之資料在上述輸入中標記為『驗證資料』。

【0065】在提供t >9000 之新出價的情況下，UTXO 可解除鎖定。新出價者必須將其公鑰P_1及解P提供至和$P =$

$P_1 + t \cdot G$。其利用附加密鑰P對交易進行簽名之事實使其始終如一地鏈接至特定t USD 出價。

【0066】作為另一實例，考慮檢索一隨機數(一數目)之比特幣挖掘者，該隨機數在與區塊鏈中之一區塊(諸如當前區塊)進行哈希運算時產生低於某一臨限值之值。挖掘者可創建利用其公鑰加上隨機數之前 9 個數位鎖定之交易$P = P_1 +$(隨機數之前 9 個數位)$\cdot G$。對於一數目之前 9 個數位，存在十億個組合。因此，可使用 30 個點相加來驗證公鑰P。此使密鑰P與挖掘者之公鑰及經挖掘區塊之隨機數始終如一地鏈接。

【0067】上文所描述之記入(inscript)點相加驗證過程可用以實行使用 N 中之 M (M-of-N)多重簽名法對交易進行簽名之有效方式，藉此實現用於使用區塊鏈轉移資源之控制或對該資源之存取的更高效及安全方法。

【0068】對於 N 中之 M 方法，比特幣作業碼 OP_MULTISIG 採用 N 個公鑰及 M 個簽名作為輸入。作業碼開始於第一簽名且檢索 N 個公鑰測試以查看該簽名是否藉由該公鑰產生。其丟棄與簽名不匹配的每一密鑰。出於此原因，簽名之階數必須匹配提供公鑰之階數。

【0069】多重簽名兌換指令碼看起來像：

<PubKey 1 ><PubKey 2 > … <PubKey N > OP_CHECKMULTISIG

【0070】需要以下作為輸入以用於解除鎖定：

<sig 1><Sig 2>…<Sig M>。

【0071】根據上文可見，指令碼大小與所需簽名之數目 M 及參與者之數目 N 二者成線性比例。

【0072】參考圖 4 至 6，創建一默克爾樹，其中葉子對應於公鑰之每一 N 中之 M 組合，該默克爾樹含有總計 N 個選擇 M 個葉子。接著，創建一兌換指令碼，其含有默克爾根，且需要解除鎖定含有默克爾路徑之輸入及簽名驗證方法之呈現。

【0073】下文描述用於轉移資源之控制或對該資源之存取的二種方法。每一種方法具有其自身優勢及劣勢。

1. **個別簽名法。**此使用每一簽名來分別地對交易進行簽名，且具有與多重簽名相同的功能性但指令碼大小較小。

2. **群組簽名法。**此在簽名群組中為 M 個參與者提供單一簽名。此可為參與者之每一集合 M 預先同意之任何公鑰，諸如源自共用秘密之密鑰，或藉由對群組之公鑰進行求和而形成的密鑰。在後一種情況中，用於驗證密鑰為密鑰之個別成員之和的額外選項使用先前所描述的驗證方法。

【0074】每一方法具有不同縮放特性，如圖 4 中佈置之表中所概括。

【0075】考慮 5 中之 4 方法。在此情況下，默克爾樹將具有 5 個選擇 $4 = \binom{5}{4} = \frac{5!}{4!(5-4)!} = 5$個葉子。默克爾樹本身將為三個層級深。用於驗證具有根<R>之默克爾路徑的運算為

<Verify Merkle Path> =

6 OP_PICK OP_SHA256 (OP_SWAP OP_IF OP_SWAP OP_ENDIF OP_CAT OP_SHA256)*3 <R> OP_EQUALVERIFY

其將對於下式之默克爾路徑返回 TRUE：

<Leaf> <Grandparent Sib> <0,1> <Parent Sib> <0,1> <Sib> <0,1>

其中在同胞(sibling)為左側節點的情況下存在 0，且在同胞為右側節點的情況下存在 1。

【0076】默克爾路徑及用以驗證默克爾路徑之操作的大小與簽名組N選擇M之選擇之數目呈對數增長。

【0077】在個別簽名法中，群組之每一成員提供交易之個別簽名。此方法之功能性與標準多重簽名法一致，然而，指令碼之大小減小，此提高指令碼所執行之效率。該指令碼具有以下形式：

<Verify Merkle Path> <CheckSig>*M

【0078】其需要解除鎖定以下輸入之呈現：

<Sig P_1>…<Sig P_M> <P_1>…<P_M> <Merkle Path $P_1,\dots,P_M$>

【0079】兌換者必須提供具有其對應公鑰之M個簽名及P之默克爾路徑作為輸入。

【0080】此方法中之$P_1,\dots,P_M$對默克爾樹葉之映射可為例如使用 OP_CAT 之$P_1,\dots,P_M$之級聯之散列。

【0081】此方法之密鑰特徵為每一成員對一特定交易進行簽名。其並不彼此揭露關於其私鑰的任何資訊，且

因此並不損害其私鑰之安全性。

【0082】參考圖 4，可見此方法對於傳統多重簽名法具有有利的縮放特性。在多重簽名體系位於低M及大N (例如 1000 中之 5)時，此最為明顯。

【0083】在群組簽名法中，對交易進行簽名之 M 個成員之群組僅需要一個簽名。該指令碼具有以下形式：

<Verify Merkle Path> <CheckSig>

【0084】其需要解除鎖定以下輸入之呈現：

<Sig P><P> <Merkle Path>

【0085】此處，P為用於M個參與者之集合之群組公鑰。

【0086】此方法之優勢為解除鎖定指令碼大小極小，此係因為其僅需要一個簽名。記入公鑰相加之數目與參與者之數目 M 成線性比例，因此在處理功率方面，此方法提供比檢查個別簽名更低成本的運算。

【0087】此方法需要參與者彼此共享其私鑰。此意謂其在此方法中使用之密鑰對為單次使用密鑰對以改良方法之安全性為適用的。另一特徵為可對群組私鑰進行存取之參與者中之任何成員可對其所希望之任何交易進行簽名。此提高方法之通用性。存在尤其需要此等特徵之應用。舉例而言，引向器之群組希望激活已在區塊鏈上記錄為 UTXO 之良好確立的最終手段(last-resort)條項。此可藉由使用此方法解除鎖定彼特定 UTXO 來激活。

【0088】在上述方法中，外部觀測器將完全未察覺此

實際上為群組簽名法。為公佈簽名及對應公鑰與使用者之群組相關之事實，方法可經修改以需要驗證群組公鑰為指令碼中之個別參與者之公鑰之和$P = P_1 + \cdots + P_M$。此調適方法可使用上文所描述之密鑰相加驗證方法。

【0089】在此調適方法中，兌換指令碼可具有以下形式：

<Verify Merkle Path> <Point Add Multi $P_1, \ldots, P_M, P$> <CheckSig>

【0090】其需要解除鎖定以下輸入之呈現：

<Sig P><P> <Verification data><P><P_1>$\cdots$<P_M> <Merkle Path>

【0091】其中<Point Add Multi $P_1, \ldots, P_M, P$>函數為先前所介紹之物，且<Verification data>含有每一個別點相加之間的梯度。

【0092】圖 5 說明一對交易 Tx1 及 Tx2。Tx1 定義具有兌換指令碼之 UTXO，該兌換指令碼含有驗證默克爾路徑(Verify Merkle Path)函數、對特定 N 中之 M 要求定製之 PointAddMulti 函數及 CheckSig 函數。Tx2 定義隨後提交至區塊鏈之交易，該交易意欲藉由在輸入中向中呈現群組簽名、與群組簽名相關之群組公鑰、PointAddMulti 函數所需之梯度、使用者群組之公鑰及相關聯默克爾路徑來兌換 Tx1 之 UTXO。

【0093】圖 6 展示說明經採用以執行根據上文所描述之個別簽名法及群組簽名法之方法之步驟的流程圖。在圖

6 中，Tx1 及 Tx2 可指代根據任一方法使用的一對交易。

【0094】現轉而參看圖 7，提供計算裝置 2600 之說明性簡化方塊圖，該計算裝置可用於實踐本揭露內容之至少一個實施例。在各種實施例中，計算裝置 2600 可用以實行上文所說明及描述之系統中之任一者。舉例而言，計算裝置 2600 可經組配以用作資料伺服器、網頁伺服器、攜帶型計算裝置、個人電腦或任何電子計算裝置。如圖 7 中所展示，計算裝置 2600 可包括具有快取記憶體之一或多個層級的一或多個處理器以及可經組配以與包括主記憶體 2608 及持久性儲存器 2610 之儲存子系統 2606 通訊的記憶體控制器(共同地標記為 2602)。主記憶體 2608 可包括如所展示之動態隨機存取記憶體(dynamic random-access memory，DRAM) 2618 及唯讀記憶體(read-only memory，ROM) 2620。儲存子系統 2606 及快取記憶體 2602 且可用於儲存資訊，諸如與如本揭露內容中所描述之交易及區塊相關聯的細節。處理器 2602 可用以提供如本揭露內容中所描述之任何實施例的步驟或功能性。

【0095】處理器 2602 亦可與一或多個使用者介面輸入裝置 2612、一或多個使用者介面輸出裝置 2614 及網路介面子系統 2616 通訊。

【0096】匯流排子系統 2604 可提供用於使計算裝置 2600 之各個組件及子系統能夠按預期彼此通訊之機制。儘管匯流排子系統 2604 經示意性地展示為單一匯流排，但

匯流排子系統之替代實施例可利用多個匯流排。

【0097】網路介面子系統 2616 可提供至其他計算裝置及網路之介面。網路介面子系統 2616 可充當用於自其他系統接收資料及將資料自計算裝置 2600 傳輸至其他系統之介面。舉例而言，網路介面子系統 2616 可使資料技術員能夠將裝置連接至網路，使得資料技術員可能夠在處於諸如資料中心之遠程位置中時將資料傳輸至裝置及自裝置接收資料。

【0098】使用者介面輸入裝置 2612 可包括：一或多個使用者輸入裝置，諸如鍵盤；指標裝置，諸如整合式滑鼠、軌跡球、觸控板或圖形平板電腦；掃描器；條形碼掃描器；觸控螢幕，其併入至顯示器中；音訊輸入裝置，諸如語音辨識系統、麥克風；及其他類型之輸入裝置。一般而言，使用術語「輸入裝置」意欲包括用於將資訊輸入至計算裝置 2600 之所有可能類型的裝置及機構。

【0099】一或多個使用者介面輸出裝置 2614 可包括顯示子系統、列印機或諸如音訊輸出裝置之非視覺顯示器等。顯示子系統可為陰極射線管(cathode ray tube，CRT)、諸如液晶顯示器(liquid crystal display，LCD)之平板裝置、發光二極體(light emitting diode，LED)顯示器，或投影裝置或其他顯示裝置。一般而言，使用術語「輸出裝置」意欲包括用於輸出來自計算裝置 2600 之資訊的所有可能類型的裝置及機制。舉例而言，一或多個使用者介面輸出裝置 2614 可用以呈現使用者介面以在使用者與

應用程式之交互可為適當的時促進此交互，該等應用程式執行所描述之處理及其中之變化。

【0100】儲存子系統 2606 可提供用於儲存可提供本揭露內容之至少一個實施例之功能性的基本程式設計及資料建構的電腦可讀儲存媒體。應用程式(程式、程式碼模組、指令)在由一或多個處理器執行時可提供本揭露內容之一或多個實施例之功能性，且可儲存於儲存子系統 2606 中。此等應用程式模組或指令可由一或多個處理器 2602 執行。儲存子系統 2606 可另外提供用於儲存根據本揭露內容所使用之資料的儲存庫。舉例而言，主記憶體 2608 及快取記憶體 2602 可提供用於程式及資料之揮發性儲存器。持久性儲存器 2610 可提供用於程式及資料之持久性(非揮發性)儲存器且可包括快閃記憶體、一或多個固態驅動機、一或多個磁性硬碟驅動機、具有相關聯可移媒體之一或多個軟碟驅動機、具有相關聯可移媒體之一或多個光學驅動機(例如 CD-ROM 或 DVD 或藍光光碟(Blue-Ray))驅動機及其他類似儲存媒體。此程式及資料可包括用於進行如本揭露內容中所描述之一或多個實施例之步驟的程式以及與如本揭露內容中所描述之交易及區塊相關聯的資料。

【0101】計算裝置 2600 可屬於各種類型，包括攜帶型電腦裝置、平板電腦、工作站或下文所描述之任何其他裝置。另外，計算裝置 2600 可包括可經由一或多個埠(例如 USB、頭戴式耳機插口、雷電型連接器等)連接至計算

裝置 2600 的另一裝置。可連接至計算裝置 2600 之裝置可包括經組配以接受光纖連接器之多個埠。因此，此裝置可經組配以將光學信號轉換成可經由將裝置連接至計算裝置 2600 之埠傳輸的電信號以供處理。歸因於電腦及網路不斷改變之本質，出於說明裝置之較佳實施例之目的，圖 7 中所描繪之計算裝置 2600 之描述僅意欲作為特定實例。具有比圖 7 中所描繪之系統更多或更少組件的許多其他組配為可能的。

【0102】應注意，上文所提及之實施例說明而非限制本發明，且熟習此項技術者將能夠設計許多替代實施例，而不脫離本揭露內容之如由所附申請專利範圍定義的範疇。在申請專利範圍中，置放於圓括號中之任何參考符號不應被認為限制申請專利範圍。詞「包含(comprising 及 comprises)」及其類似者並不排除除任何申請專利範圍或說明書中整體列出之彼等元件或步驟外的元件或步驟之存在。在本說明書中，「包含(comprises 及 comprising)」意謂「包括(includes 及 including)或由……組成(consists of 及 consisting of)」。元件之單數參考並不排除此等元件之複數參考，且反之亦然。本揭露內容可藉助於包含若干獨特元件之硬體且藉助於經合適程式化之電腦實行。在枚舉若干構件之裝置申請專利範圍中，此等構件中之若干者可由硬體之同一物件體現。在相互不同之附屬申請專利範圍中敍述某些措施之純粹實情並不指示不能有利地使用此等措施之組合。

【符號說明】

【0103】

2600...計算裝置

2602...處理器

2604...匯流排子系統

2606...儲存子系統

2608...主記憶體

2610...持久性儲存器

2612...使用者介面輸入裝置

2614...使用者介面輸出裝置

2616...網路介面子系統

2618...動態隨機存取記憶體

2620...唯讀記憶體

【發明申請專利範圍】

【請求項1】　一種電腦實行方法，其包含以下步驟：

提供包含一公鑰組合驗證函數之一區塊鏈交易，該區塊鏈交易經組配為可兌換的以藉由向該區塊鏈交易提供一輸入來准許對一資源之存取或轉移該資源之控制，該輸入包含：

多個公鑰，該等公鑰之各者係在一橢圓曲線上之一點；

至少一個梯度值，其係於該等多個公鑰中之各別二個公鑰間之線的梯度；及

一群組公鑰，其係該等各別二個公鑰之該橢圓曲線之點的相加，

其中該區塊鏈交易經組配以根據該交易之成功兌換而將該公鑰驗證函數應用於該輸入以驗證該群組公鑰係該等各別二公鑰之該橢圓曲線之點的相加。

【請求項2】　如請求項 1 之方法，其包含以下步驟：

自一已知值導出一導出公鑰；以及

將該導出公鑰配置為次級公鑰之一序列，

其中該次級公鑰之該序列為該等多個公鑰之一子集。

【請求項3】　如請求項 2 之方法，其包含以下步驟：在將該導出公鑰配置為次級公鑰之一序列之前將一限制函數應用於該已知值，藉此限制該子集之大小。

【請求項4】　如請求項 3 之方法，其包含以下步驟：至少部分地基於該等多個中之一另外公鑰及該導出公鑰計

算該等多個中之一公鑰。

【請求項5】　如請求項 1 至 4 中任一項之方法，該區塊鏈交易進一步包含：

一群組簽名，其對應於該群組公鑰；

一默克爾根；及

一默克爾路徑驗證函數，

其中藉由向該區塊鏈交易提供以下各者而將該區塊鏈交易組配為可兌換的：

一默克爾路徑，其與該等多個公鑰相關聯；及

一群組私鑰，其對應於該群組簽名。

【請求項6】　一種運算系統，其包含：一處理器；及記憶體，該記憶體包含可執行指令，該等可執行指令因由該處理器執行而使得該系統執行如請求項 1 至 5 中任一項之電腦實行方法。

【請求項7】　一種非暫時性電腦可讀儲存媒體，其上儲存有可執行指令，該等可執行指令因由一電腦系統之一處理器執行而使得該電腦系統至少執行如請求項 1 至 5 中任一項之電腦實行方法。

【發明圖式】

表1：公鑰資料編碼

資料結構	位元組長度	資料
未壓縮密鑰旗標	1	04
X	32	F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A
Y	32	07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB

【圖1】

表2：在$P_1 \neq P_2$時檢查λ

堆疊	指令碼	描述
	$\langle y_2\rangle \langle y_1\rangle \langle \lambda\rangle \langle x_2\rangle \langle x_1\rangle$ OP_SUB OP_MUL $\langle p\rangle$ OP_MOD OP_3 OP_ROLL OP_3 OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	添加至堆疊之常數
$\langle y_2\rangle \langle y_1\rangle \langle \lambda\rangle \langle x_2\rangle \langle x_1\rangle$	OP_SUB OP_MUL $\langle p\rangle$ OP_MOD OP_3 OP_ ROLL OP_3 OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	減法
$\langle y_2\rangle \langle y_1\rangle \langle \lambda\rangle \langle x_2 - x_1\rangle$	OP_MUL $\langle p\rangle$ OP_MOD OP_3 OP_ ROLL OP_3 OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	乘法
$\langle y_2\rangle \langle y_1\rangle \langle \lambda(x_2 - x_1)\rangle$	$\langle p\rangle$ OP_MOD OP_3 OP_ ROLL OP_3 OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	添加至堆疊之常數
$\langle y_2\rangle \langle y_1\rangle \langle \lambda(x_2 - x_1)\rangle \langle p\rangle$	OP_MOD OP_3 OP_ ROLL OP_3 OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	
$\langle y_2\rangle \langle y_1\rangle \langle \lambda(x_2 - x_1) \bmod p\rangle$	OP_3 OP_ ROLL OP_3 OP_PICK OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	取模
$\langle y_1\rangle \langle \lambda(x_2 - x_1) \bmod p\rangle \langle y_2\rangle$	OP_3 OP_ ROLL OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	翻轉堆疊之第3項
$\langle \lambda(x_2 - x_1) \bmod p\rangle \langle y_2\rangle \langle y_1\rangle$	OP_SUB $\langle p\rangle$ OP_MOD OP_EQUAL	翻轉堆疊之第3項
$\langle \lambda(x_2 - x_1) \bmod p\rangle \langle y_2 - y_1\rangle$	$\langle p\rangle$ OP_MOD OP_EQUAL	減法
$\langle \lambda(x_2 - x_1) \bmod p\rangle \langle y_2 - y_1\rangle \langle p\rangle$	OP_MOD OP_EQUAL	添加至堆疊之常數
$\langle \lambda(x_2 - x_1) \bmod p\rangle \langle y_2 - y_1 \bmod p\rangle$	OP_EQUAL	取模
TRUE		檢查相等

【圖2】

表3：檢查P_3之χ座標

堆疊	指令碼	描述
	$\langle x_3\rangle \langle \lambda\rangle \langle x_1\rangle \langle x_2\rangle$ OP_3 OP_ ROLL OP_DUP OP_MUL OP_SWAP OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	
$\langle x_3\rangle \langle \lambda\rangle \langle x_1\rangle \langle x_2\rangle$	OP_3 OP_ ROLL OP_DUP OP_MUL OP_SWAP OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	增加至堆疊之常數
$\langle x_3\rangle \langle x_1\rangle \langle x_2\rangle \langle \lambda\rangle$	OP_DUP OP_MUL OP_SWAP OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	翻轉第3項
$\langle x_3\rangle \langle x_1\rangle \langle x_2\rangle \langle \lambda\rangle \langle \lambda\rangle$	OP_MUL OP_SWAP OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	複製
$\langle x_3\rangle \langle x_1\rangle \langle \lambda^2\rangle \langle x_2\rangle$	OP_SWAP OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	乘法
$\langle x_3\rangle \langle x_1\rangle \langle \lambda^2 - x_2\rangle$	OP_SUB OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	交換
$\langle x_3\rangle \langle \lambda^2 - x_2\rangle \langle x_1\rangle$	OP_SWAP $\langle p\rangle$ OP_MOD OP_EQUAL	減法
$\langle x_3\rangle \langle \lambda^2 - x_2 - x_1\rangle$	$\langle p\rangle$ OP_MOD OP_EQUAL	增加至堆疊之常數
$\langle x_3\rangle \langle \lambda^2 - x_2 - x_1\rangle \langle p\rangle$	OP_MOD OP_EQUAL	取模
$\langle x_3\rangle \langle \lambda^2 - x_2 - x_1 \bmod p\rangle$	OP_EQUAL	交換
TRUE		檢查相等

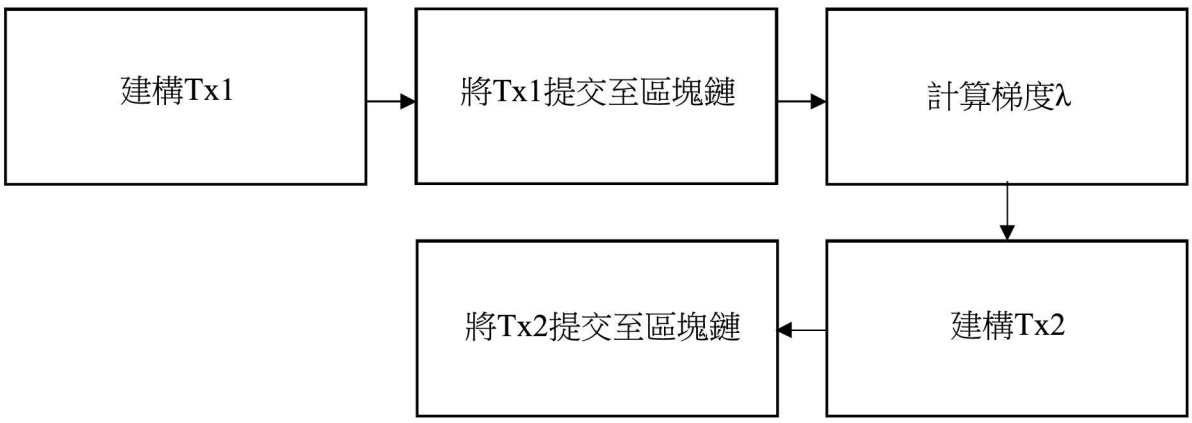
【圖3】

縮放特性	標準多重簽名	個別簽名法	群組簽名法	利用密鑰相加之群組簽名法
簽名	線性M	線性M	零	零
公鑰	線性N	線性M	零	線性M
默克爾路徑	N/A	$\text{Log}\binom{N}{M}$	$\text{Log}\binom{N}{M}$	$\text{Log}\binom{N}{M}$

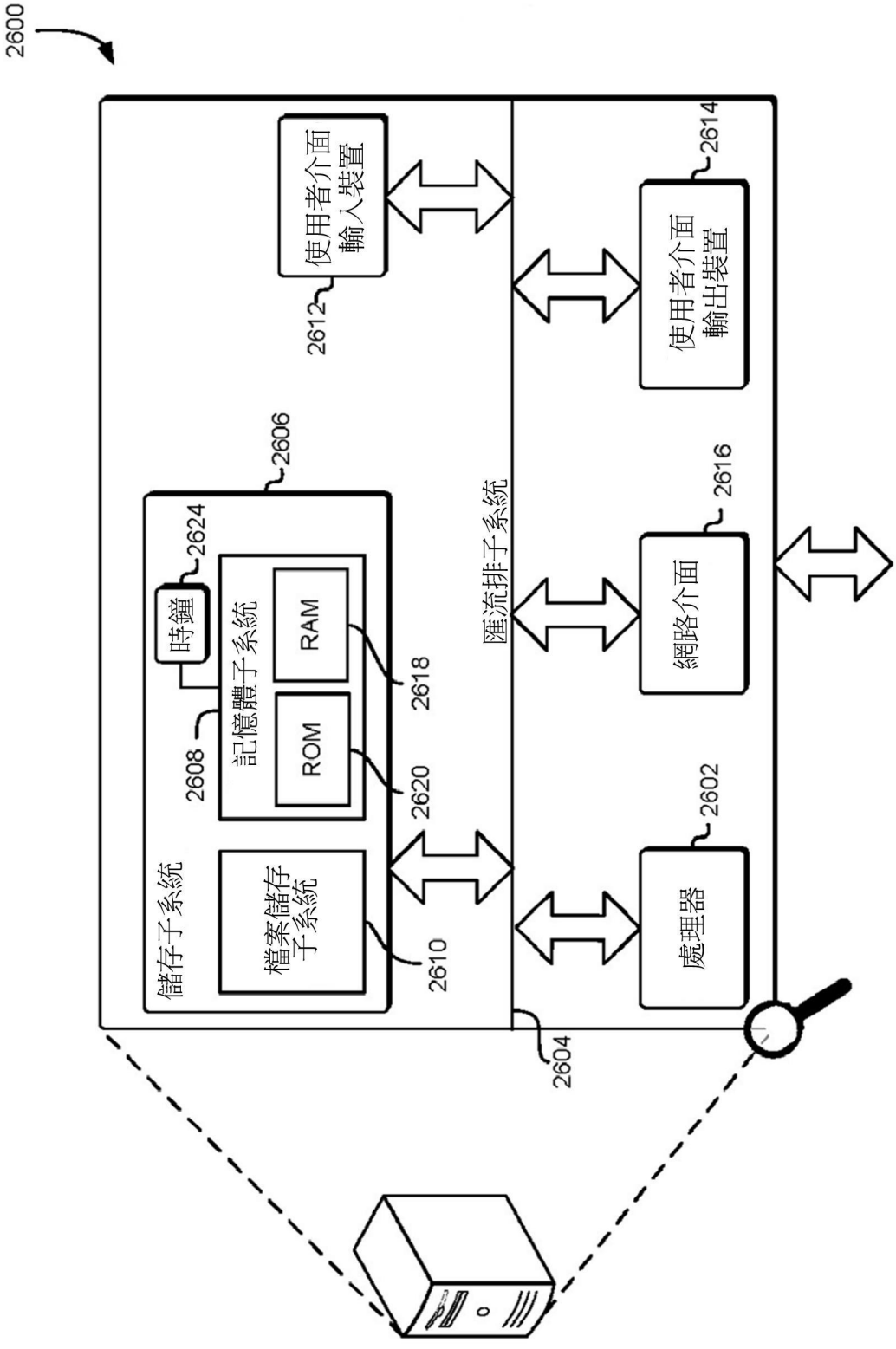
【圖4】

Tx1	Tx2
兌換指令碼： <Verify Merkle path> <PointAddMulti P1,P2,…,PM,P> <CheckSig>	輸入(解除鎖定指令碼)： <Sig P><P> <Gradients><P><P1><P2>…<PM> <Merkle path>

【圖5】



【圖6】


2600
使用者介面輸入裝置
2612
2614
使用者介面輸出裝置
2606
2624
時鐘
記憶體子系統
RAM
2618
2608
ROM
2620
匯流排子系統
2616
網路介面
儲存子系統
檔案儲存子系統
2610
2602
處理器
2604

【圖7】