I837227

(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 I837227 B**

**(45)公告日： 中華民國 113 (2024) 年 04 月 01 日**

(21)申請案號：108142772 (22)申請日： 中華民國 108 (2019) 年 11 月 25 日

(51)Int. Cl.： ***H04L9/40*** ***(2022.01)*** ***H04L9/32*** ***(2006.01)***

***H04L9/30*** ***(2006.01)***

(30)優先權：2018/11/27 英國 1819290.6

2018/11/27 英國 1819286.4

2018/11/27 英國 1819284.9

2018/11/27 英國 1819297.1

2018/11/27 英國 1819299.7

2018/11/27 英國 1819291.4

2018/11/27 英國 1819293.0

(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)；戴維斯 傑克 O DAVIES, JACK OWEN (GB)；塔爾登 克洛伊 C TARTAN, CHLOE CEREN (GB)；沃恩 歐文 VAUGHAN, OWEN (GB)

(74)代理人：劉法正；尹重君

(56)參考文獻：

CN 107682308A US 2016/0342977A1

網路文獻 S. Morishima and H. Matsutani, "Accelerating Blockchain Search of Full Nodes Using GPUs", 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018/03/21. https://www.semanticscholar.org/paper/Accelerating-Blockchain-Search-of-Full-Nodes-Using-Morishima-Matsutani/0eaeb6afd1dae0f45b601c715b9013170b9c4424

專書 Andreas M. Antonopoulos, "Mastering Bitcoin: Unlocking Digital Cryptocurrencies", second release, O'Reilly, 2015/3/6.

審查人員：黃偉倫

申請專利範圍項數：14 項 圖式數：20 共 70 頁

(54)名稱

用於透過點對點網路儲存、提取及傳遞資料之電腦實施系統及方法

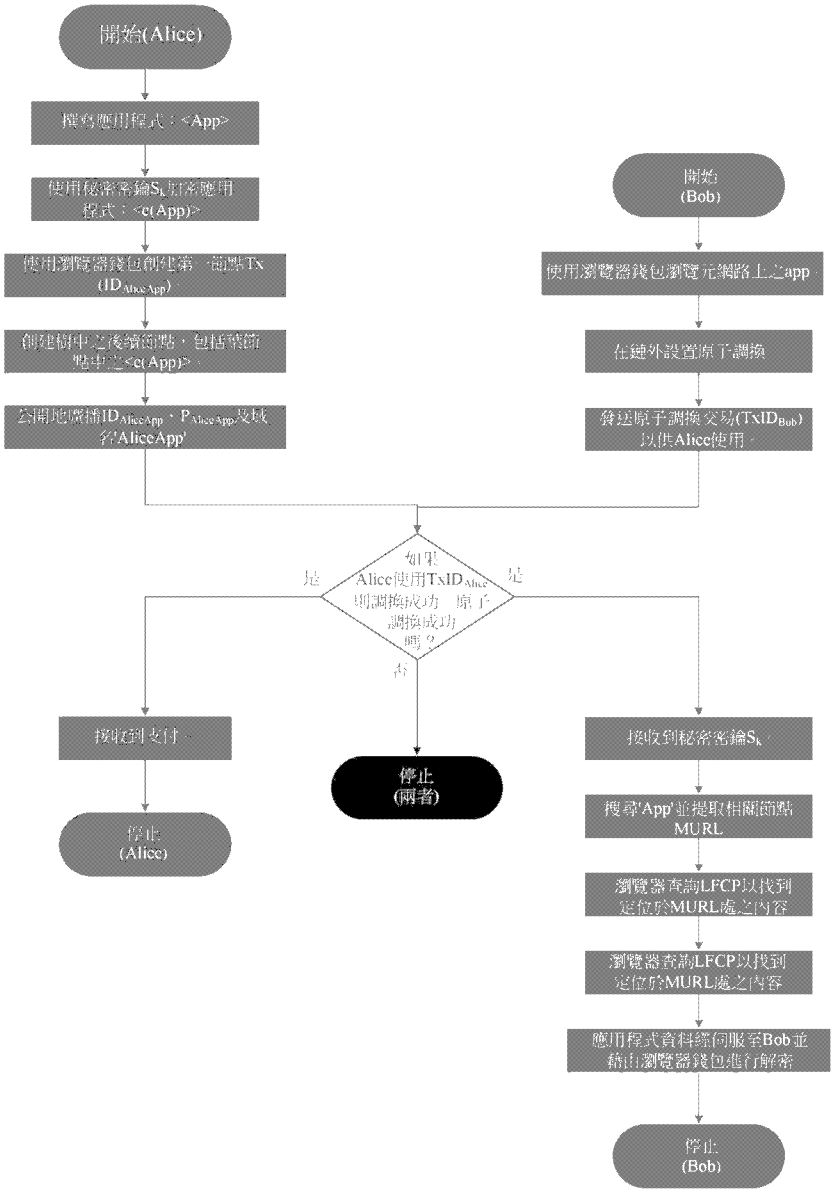
(57)摘要

本發明提供一種用於搜尋一區塊鏈(例如，比特幣)以找到儲存在一或多個區塊鏈交易中之資料/內容，並存取彼資料/內容之系統。該系統可結合用於搜尋該區塊鏈之一協定使用。本發明的一實施例可配置成使得一使用者能夠搜尋、存取、檢視、寫入及/或提取提供在至少一個區塊鏈交易(Tx)中之資料的一部分，且亦配置成基於包含與該交易(Tx)相關聯之一交易 ID 及一公鑰之一交易索引

($TX_{index}$)來識別該至少一個交易(Tx)。該系統可包含一搜尋設施，其提供在區塊鏈搜尋系統內；或配置成與該區塊鏈搜尋系統介接及/或通訊。該系統亦可包含至少一個密碼貨幣錢包。

The invention provides a system for searching a blockchain (e.g. Bitcoin) for data/content stored in one or more blockchain transactions, and accessing that data/content. It may be used in conjunction with a protocol for searching the blockchain. An embodiment of the invention can be arranged to enable a user to search for, access, view, write and/or retrieve a portion of data provided in at least one blockchain transaction (Tx), and also arranged to identify the at least one transaction (Tx) based on a transaction index ($TX_{index}$) comprising a transaction ID and a public key associated with the transaction (Tx). The system may comprise a search facility which is either provided within the blockchain search system; or arranged to interface and/or communicate with the blockchain search system. It may also comprise at least one cryptocurrency wallet.

指定代表圖：




【圖19】

專利案號：108142772

公告本

I837227

# 【發明摘要】

## 【中文發明名稱】

用於透過點對點網路儲存、提取及傳遞資料之電腦實施系統及方法

## 【英文發明名稱】

COMPUTER IMPLEMENTED SYSTEMS AND METHODS FOR STORING, RETRIEVING AND COMMUNICATION DATA VIA A PEER-TO-PEER NETWORK

## 【中文】

本發明提供一種用於搜尋一區塊鏈(例如，比特幣)以找到儲存在一或多個區塊鏈交易中之資料/內容，並存取彼資料/內容之系統。該系統可結合用於搜尋該區塊鏈之一協定使用。本發明的一實施例可配置成使得一使用者能夠搜尋、存取、檢視、寫入及/或提取提供在至少一個區塊鏈交易(Tx)中之資料的一部分，且亦配置成基於包含與該交易(Tx)相關聯之一交易 ID 及一公鑰之一交易索引($TX_{index}$)來識別該至少一個交易(Tx)。該系統可包含一搜尋設施，其提供在區塊鏈搜尋系統內；或配置成與該區塊鏈搜尋系統介接及/或通訊。該系統亦可包含至少一個密碼貨幣錢包。

## 【英文】

The invention provides a system for searching a blockchain (e.g. Bitcoin) for data/content stored in one or more blockchain transactions, and accessing that data/content. It may be used in conjunction with a protocol for searching the blockchain. An embodiment of the invention can be arranged to enable a user to search for, access, view, write and/or retrieve a portion of data provided in at least one blockchain transaction (Tx), and also arranged to identify the at least one transaction (Tx) based on a transaction index ($TX_{index}$) comprising a transaction ID and a public key associated with the transaction (Tx). The system may comprise a search facility which is either provided within the blockchain search system; or arranged to interface and/or communicate with the blockchain search system. It may also comprise at least one cryptocurrency wallet.

【指定代表圖】圖19

【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】

用於透過點對點網路儲存、提取及傳遞資料之電腦實施系統及方法

## 【英文發明名稱】

COMPUTER IMPLEMENTED SYSTEMS AND METHODS FOR STORING, RETRIEVING AND COMMUNICATION DATA VIA A PEER-TO-PEER NETWORK

## 【技術領域】

發明領域

【0001】本發明大體上涉及對跨越電子網路，且特定言之諸如區塊鏈網路之點對點網路的資料通訊及交換之資料改良。其涉及儲存、存取、提取及處理，且更特定言之，涉及區塊鏈上之此等資料相關活動。本發明尤其適於但不限於以類似於由網站及網頁提供之方式來處理資料，但其使用區塊鏈作為基礎機制或平台而非網路伺服器。因此，本發明提供用於資料處理及傳送的安全有效之以密碼方式執行之替代基礎設施。

## 【先前技術】

發明背景

【0002】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佈式總帳。此等總帳包括基於共識之區塊鏈及交易鏈技術、許可及未許可總帳、共用總帳及其變型。區塊鏈技術之最廣泛已知之應用為比特幣總帳，儘管已提出並開發了其他區塊鏈實施。雖然本文中出於方便及說明之目的可提及比特幣，但應注意，本發明不限於與比特幣區塊鏈一起使用，且替代區塊鏈實施及協定屬於本發明的範疇。術語「使用者」在本文中可用以指人類或基於處理器之資源。如本文中所使用之「比特幣」包括自比特幣協定導出的所有版本及變型之協定。

【0003】區塊鏈為點對點電子總帳，其經實施為由區塊構成之基於電腦之去中心化分佈式系統，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之控制的傳送，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，從而使得區塊變為鏈接在一起以產生自一開始便已寫入至區塊鏈之所有交易的永久性不可變更之記錄。交易含有嵌入至其輸入及輸出中的被稱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「驗證」。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藉由執行其鎖定及解除鎖定指令碼而對未用交易(UTXO)執行此驗證工作。若鎖定及解除鎖定指令碼之執行評估為真，則交易係有效的且將交易寫入至區塊鏈。因此，為了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驗證，若交易經驗證，則節點將該交易轉送至網路中之其他節點；且 ii)添加至由挖掘者構建之新區塊；且 iii)經挖掘，亦即添加至過去交易之公用總帳。

【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編譯安全系統及可儲存於區塊鏈上以實施新系統之資料兩者的使用。區塊鏈技術在區塊鏈可用於不限於密碼貨幣範圍之任務及過程時高度有利。此等解決方案將能夠利用區塊鏈的益處(例如，事件的永久性防篡改記錄、分佈式處理等)，同時在其應用中變得更通用。

【0006】一個此所關注領域為使用區塊鏈在使用者當中儲存、共用、存取及控制資料。現今，此係透過網際網路實現的，其中伺服器代管網站及網頁，使用者通常藉由搜尋引擎訪問該等網站及網頁，以便存取所要資料。

【0007】 然而，一些觀察者已開始設想使用區塊鏈來解決網際網路之一些缺點，諸如由中心化各方控制大量資料及內容。參見例如「Life After Google: The Fall of Big Data and the Rise of the Blockchain Economy」(George Gilder，閘道器版本，2018 年 7 月，ISBN-10: 9781621575764 及 ISBN-13: 978-1621575764)。

【0008】 因此，期望提供使得能夠有利地利用區塊鏈之分佈式、不可變更且永久性本質在區塊鏈上儲存、處理、提取、搜尋及/或共用此資料之配置。現在已設計出此改良之解決方案。

【0009】 本發明之實施例至少提供用於實施區塊鏈解決方案，且用於在其上或自其儲存、處理、搜尋及/或提取資料之替代有效安全技術。實施例還至少提供用於在計算節點之間儲存、處理、提取、傳送、搜尋及/或共用資料的替代區塊鏈實施技術之基礎設施。因為本發明實現以新方式使用區塊鏈網路並將其用於提供經改良及技術結果，所以本發明提供經改良區塊鏈實施網路。

【0010】 實施例還提供用於安全控制對包含諸如區塊鏈之點對點網路及區塊鏈協定之技術上不同且經改良計算平台上之數位資源之存取的解決方案。

## 【發明內容】

### 發明概要

【0011】 本發明定義於所附申請專利範圍中。

【0012】 根據本發明，可提供一種電腦實施方法/系統。本發明可配置成用於跨越諸如區塊鏈之點對點網路儲存、提取及/或共用資料。資料可跨越一或多個區塊鏈交易而儲存。傳統地或替代地，本發明可被稱為搜尋系統。該系統可配置成用於與或結合區塊鏈搜尋協定一起操作。區塊鏈搜尋協定可大體上如關於本文中所描述之「元網路」協定所描述。

【0013】 本文中，「共用」可包括向節點或使用者提供對資料部分之發送、傳遞、傳輸或存取。術語「處理」可經解譯為意指與交易或其相關聯資料相關

之任何活動，包括生成、傳輸、驗證、存取、搜尋、至區塊鏈網路之共用提交及/或識別。

【0014】該系統可配置成使得(人類或機器)使用者能夠搜尋、存取、檢視、提取及/或以其他方式處理提供在至少一個區塊鏈交易(Tx)中或由至少一個區塊鏈交易(Tx)參考之資料的一部分。因此，區塊鏈交易可包含資料的一部分，或資料的一部分之參考。資料的部分之參考可為儲存資料之位置的指標、位址或其他指示符。資料的部分可為任何類型之資料或數位內容，例如電腦可執行項、文本、視訊、影像、聲音檔案等。資料之部分可被稱為「內容」。資料之部分或其參考可呈經處理形式。例如，其可為資料之部分的散列摘錄。資料可儲存於區塊鏈上或外(即，「鏈外」)。

【0015】較佳地，該系統配置成基於包含與交易(Tx)相關聯之交易ID及公鑰的交易索引($TX_{index}$)來識別至少一個交易(Tx)。

【0016】如技術人員將容易地瞭解，每一區塊鏈交易(Tx)具有其自身之唯一識別符(ID)，從而使得可在區塊鏈上識別每一區塊鏈交易。有利地，本發明組合此現有ID與相關聯於交易及/或提供在交易內的公鑰，使得該組合形成交易之唯一識別符。此情況促進快速且有效搜尋區塊鏈。

【0017】較佳地，該系統包含搜尋設施，其提供在區塊鏈搜尋系統內(即，本地或內部)；或配置成與區塊鏈搜尋系統介接及/或通訊(即，在搜尋系統外部)。

【0018】較佳地，該系統包含至少一個密碼貨幣錢包。較佳地，至少一個錢包配置成生成、儲存及/或處理階層式確定性密鑰。至少一個密碼貨幣錢包可配置成將至少一個密碼密鑰及/或至少一個符記儲存在可信執行環境(TEE)中。

【0019】較佳地，系統包含解壓縮組件，其配置成在資料部分經壓縮時對其進行解壓縮；重組組件；及/或配置成在資料部分經加密時對其進行解密之解密組件。重組組件可配置成將已自區塊鏈上之多於一個交易提取的資料之部分

重組。

【0020】較佳地，系統包含至少一個呈現組件，其配置成以可聽及/或視覺形式向使用者呈現資料之部分。

【0021】較佳地，搜尋系統包含用於輸入或生成用以識別區塊鏈上之至少一個交易(Tx)的搜尋路徑的構件，該搜尋路徑包含：

i)交易索引(TX$_{index}$)；及

ii)與交易(Tx)相關聯之至少一個屬性。

【0022】較佳地，屬性中之至少一者為與交易相關聯之助憶符；及/或至少一個屬性為空值。

【0023】較佳地，(搜尋)系統配置成與密碼貨幣錢包或其他資源通訊以促進處理、儲存及/或生成密碼密鑰、區塊鏈交易及/或數位簽名。

【0024】較佳地，系統配置成儲存交易索引(TX$_{index}$)。較佳地，系統配置成儲存用於多於一個交易之各別交易索引。

【0025】較佳地，系統進一步配置成在存取資料部分之前將對密碼貨幣的一部分之控制傳送至目的地。有利地，此情況使得系統能夠將區塊鏈用作儲存/提取/搜尋機制以及支付及控制機制兩者，而非必須針對不同活動使用單獨的基礎設施。

【0026】較佳地，系統進一步配置成向區塊鏈上之同級者發送對資料部分之請求；及/或自區塊鏈上之同級者接收資料部分。此情況使得能夠創建涉及使用第三方以與區塊鏈交互，並將所要資料提供至使用者及/或代表使用者寫入資料至區塊鏈之基礎設施。

【0027】較佳地，多個交易提供於區塊鏈上，每一交易具有或提供資料的一部分或資料的一部分之參考。較佳地，多個交易(TX)中之一者、一些或全部包含輸入，該輸入包括：

與由任意交易 ID(DTxID)識別之邏輯父代交易(LPTx)相關聯的父代公鑰(PPK)；及

使用父代公鑰(PPK)生成之簽名。

【0028】 此情況使得能夠在交易與其嵌入資料之間構造邏輯階層。因此，可有效、安全且快速地處理區塊鏈上之多個相關聯或邏輯上鏈接之交易。邏輯上相關聯之交易可並非以連續區塊高度儲存於區塊鏈上，但其可被容易且安全地識別及/或存取。

【0029】 較佳地，系統進一步配置成使用時間鎖定機制以控制對資料部分之存取。此情況提供對區塊鏈上之資料存取的較大控制程度。

【0030】 本發明亦提供一種經配置且組配以執行本文中所描述之方法的任何實施例之步驟的對應系統。該系統可包含電腦實施系統，其包含：

處理器；及

記憶體，其包括可執行指令，作為由處理器執行之結果，該等可執行指令致使系統執行如本文中所描述之電腦實施方法之任何實施例。

【0031】 本發明亦提供上面儲存有可執行指令之非暫時性電腦可讀儲存媒體，作為由電腦系統之處理器執行的結果，該等可執行指令致使電腦系統至少執行如本文中所描述之方法的實施例。

【0032】 本發明之方法/系統的一些實施例可包含如下文，且特定言之標題為「瀏覽器/錢包應用程式」之章節中所描述之一或多個特徵。

【0033】 因此，本發明可提供用於透過區塊鏈儲存、搜尋、識別、傳遞及/或存取資料之經改良解決方案。實施例提供對跨越電子網路，具體而言點對點區塊鏈網路之資料通訊及交換的改良。

## 【圖式簡單說明】

【0034】 本發明的此等及其他態樣將自本文中描述的實施例顯而易見且參

考本文中所描述的實施例進行闡明。現將僅借助於實例且參考隨附圖式來描述本發明的實施例，其中：

圖 1 示出體現本發明之區塊鏈交易，其中資料儲存在多個輸出中；

圖 2 示出體現本發明之區塊鏈交易，其中資料儲存在輸入中；

圖 3 示出體現本發明之一系列區塊鏈交易，其中資料儲存在多個區塊鏈交易之輸出上；

圖 4 示出體現本發明之區塊鏈交易，該交易傳送密碼貨幣支付以允許借助於原子調換存取資料；

圖 5 示出體現本發明之區塊鏈交易，其用於兌換圖 4 之交易的支付；

圖 6 示出由體現本發明之區塊鏈交易中的參與者保存之秘密值，該交易發佈符記以允許借助於原子調換存取資料；

圖 7 及圖 8 示出體現本發明之區塊鏈交易，其用於發佈符記以允許借助於原子調換存取資料；

圖 9 及圖 10 示出體現本發明之區塊鏈交易，其用於兌換借助於圖 7 及圖 8 之交易發佈的符記；

圖 11 及圖 12 示出用於存取由圖 9 及圖 10 之交易交換之秘密的區塊鏈交易；

圖 13 提供根據本發明之實施例的元網路(Metanet)圖結構之說明；

圖 14 示出根據本發明之實施例的用於包括 MURL 搜尋路徑之域'bobsblog'的元網路圖樹的說明；

圖 15 示出根據本發明之一個實例的瀏覽器錢包之例示性實施例的示意圖，及可如何跨越應用程式之不同組件分割其核心功能；

圖 16 提供說明可如何在本發明之實施例的基礎設施內執行內容搜尋之圖式；

圖 17 示出根據本發明之實施例的本端全複本同級者與全域全複本同級者之

間的例示性交互；

圖 18 示出用於參考下文所描述之例示性使用案例的元網路樹(或圖)；

圖 19 示出說明由下文所提供之例示性使用案例體現的過程之流程圖；

圖 20 為說明可實施各種實施例之計算環境的示意圖。

【實施方式】

【0035】 較佳實施例之詳細說明

術語「比特幣」在本文中僅為方便起見而使用，且意欲包括所有密碼貨幣/區塊鏈協定，包括但不限於自比特幣協定導出之所有變型以及用於其他區塊鏈之任何替代協定。在此文件之剩餘部分中，本發明之實施例的協定判定操作將被稱作「元網路協定」。

【0036】 根據本發明之實施例，術語「內容」、「數位內容」及「資料」可在本文中互換使用，以指儲存在區塊鏈交易中/由區塊鏈交易參考或以其他方式透過區塊鏈交易存取之資料。與由基礎區塊鏈協定需要的作為交易程式碼自身之部分的資料相反，資料為透過區塊鏈輸送、傳遞或儲存之額外/任意資料。

**概述**

【0037】 如上文所陳述，公認需要用於在計算節點之間及由計算節點儲存、寫入、存取及審查資料之經改良及/或替代基礎設施。使用區塊鏈技術固有之益處(例如，不可變之記錄、以密碼方式執行之控制及存取、內置式支付機制、公開檢查總帳之能力、分佈式架構等)將係有利的。然而，自數個技術視角而言，構造「區塊鏈實施之網際網路」具挑戰性。

【0038】 此等挑戰可包括但不限於：如何在網路內定位特定資料部分；如何保證並控制資料存取，從而使得僅授權方可進行存取；如何以點對點方式將資料自一方傳送至另一方；如何配置資料，從而使得其可在邏輯上相關聯但仍儲存在網路內之不同位置，及如何隨後自不同位置組合資料以提供總體且經擴

增結果；如何以階層式方式提供及/或儲存資料；如何允許使用者及具有不同計算平台之各方存取所要資料；如何跨越(可能全域)計算網路儲存、提供及共用資料，而無需依賴於或需要大型儲存伺服器及中心化資料控制器。

【0039】 本發明以一方式提供此經改良解決方案，該方式在一定程度上類似於網際網路，但使用與先前技術中所已知完全不同之硬體及軟體組件平台以完全不同之方式實現其結果。根據本發明之實施例，儲存網際網路/網路資料並將其提供至終端使用者之伺服器由駐存在區塊鏈網路上之區塊鏈交易替換。為了實現此，必須設計出若干創新。以下章節中描述此等創新。

【0040】 將資料插入至區塊鏈「元網路」中參考圖 1，示出體現本發明之區塊鏈交易，其中待儲存於區塊鏈上之第一資料儲存在交易之一或多個第一輸出中，且表示第一資料之屬性的第二資料儲存在交易之一或多個第二輸出中。第一資料之一或多個第一部分<內容 1>儲存在交易之可用輸出中。表示第一資料之各別屬性的資料<屬性 1>及<屬性 2>，連同指示係根據元網路協定儲存資料之旗標儲存在交易之第二不可用輸出中。術語「不可用」用於指示交易之至少一個第一及/或第二輸出可包括指令碼作業碼(OP RETURN)，其用於將輸出標記為無效的而無法後續用作至後續交易之輸入。

【0041】 將資料之內容及屬性部分單獨地儲存在交易之單獨輸出中係有利的。

【0042】 圖 2 示出體現本發明之區塊鏈交易，其中待儲存於區塊鏈上之第一資料<內容 1>儲存在交易之輸入中。元網路旗標以及屬性資料<屬性 1>及<屬性 2>以類似於圖 1 中所示之配置的方式儲存在交易之不可用輸出中。

**資料插入**

資料插入方法

【0043】 期望能夠將以下資料插入至區塊鏈中

a) 元網路旗標

b) 屬性

c) 內容

【0044】 內容為待儲存於區塊鏈上之資料，元網路旗標為充當關於元網路協定之任何資料的識別符之 4 位元組首碼，而屬性含有關於內容之編索引、許可及編碼資訊。此資訊可包括但不限於資料類型、加密及/或壓縮方案。此等屬性常常亦被稱為元資料。將在本發明文件中避免使用此術語以免混淆交易元資料。

【0045】 以下技術可用於將此資料嵌入比特幣指令碼內：

1. OP_RETURN - 在此方法中，所有資料(屬性及內容)在可證明不可用交易輸出之鎖定指令碼中皆置放在 OP_RETURN 之後。

【0046】 使用此運算符之輸出指令碼的實例為：

UTXO0: OP_RETURN <元網路旗標> <屬性> <內容>

2. OP_RETURN 與 OP_DROP - 在此情況下，OP_RETURN 含有屬性，而內容在可用交易指令碼(鎖定或解除鎖定)中儲存在 OP_DROP 之前。內容可被分割成交易輸入及輸出內之多個資料封包。然而，將資料插入至交易輸出中係有利的，因為其僅為可在比特幣協定中簽署之輸出指令碼。若資料經插入至交易輸入中，則可將 OP_MOD 而非挖掘者驗證用作資料上之校驗和以確保其有效性。例如，吾人可執行 32 位元 OP_MOD 運算並檢查其等於經預計算值。

【0047】 在此情況下，屬性可含有關於如何重組內容資料封包之資訊。另外，將經重組資料封包 H(內容 1 + 內容 2)之散列提供為屬性使得能夠驗證已使用所建議重組方案。

【0048】 圖 1 中示出實施第二資料插入方法之交易。為簡單起見，此交易僅包括由其單個輸入簽署的插入於其輸出中之內容。使用如圖 2 中所示之此方

法來使用 OP_DROP 語句亦將會使插入至額外輸入中之內容成為可能。

【0049】 若內容極大，則在多個交易上分割內容可係有利的。圖 3 中示出此配置。圖 3 示出體現本發明之一對區塊鏈交易，其中待儲存於區塊鏈上之第一資料<內容>分割成兩個組塊<內容組塊 1>及<內容組塊 2>，該等組塊可隨後重組為<內容>=<內容組塊 1>‖<內容組塊 2>，其中運算符'‖'串聯內容資料之兩個組塊。此串聯運算符可由任何所要逐位元或類似逐段二進位運算符替換。兩個組塊<內容組塊 1>及<內容組塊 2>接著儲存在單獨區塊鏈交易之各別可用輸出中，而與內容資料之屬性相關的資料儲存在區塊鏈交易之各別不可用輸出中。再次，屬性可含有關於重組方案之資訊。例如，內容可為原始資料、可執行程式或 HTML 網頁。另外，內容 1 可包括至內容 2 在區塊鏈上之位置的指標，其以相同於網頁內之嵌入 HTML 鏈接的方式起作用。

【0050】 應注意，兩交易皆將同一公鑰*P* (及 ECDSA 簽名)作為輸入，使得儘管<內容組塊 1>及<內容組塊 2>儲存在分別具有 TxID1 及 TxID2 之不同交易中，但其可由同一公鑰*P*相關。

利用挖掘者驗證之作用

【0051】 此處，由挖掘者執行之交易驗證過程用於在儲存此資料時獲得優勢。此係因為交易輸出中之所有資料將由公鑰*P*之所有者在至少一個交易輸入中簽署(若存在 SIGHASH|ALL 旗標)，且此簽名將在所有挖掘者執行之交易驗證過程中經檢查。

【0052】 此確保

- 資料完整性 - 若資料經損毀，則 CHECKSIG 操作將失敗。
- 資料真實性 - *P*之所有者已可證明地見證並簽署資料。

【0053】 此對於在多個交易上經分割之內容尤其有利，因為*P*之輸入簽名在資料之分割分量之間提供了可證明鏈接，如上文參考圖3中所示之配置所描

述。

Rabin 簽名

【0054】 確保資料真實性之另一方式為使用 Rabin 簽名，其可用於簽署資料自身而非整個消息。此可係有利的，因為簽署者無需簽署出現資料之每一個別交易，且簽名可在多個交易中再用。

【0055】 可在指令碼中容易地驗證 Rabin 簽名。可藉由將 Rabin 簽名驗證插入在 OP_DROP 命令之前而將此等併入在上文之情況(2)中，亦即

<內容 1> <Rabin Sig (內容 1)> FUNC_CHECKRABSIG OP_DROP <H($P_1$)> [CheckSig $P_1$]

【0056】 應注意，此方法無法在上文情況(1)中進行，此係由於不論如何，含有OP_RETURN之指令碼皆失敗且因此無法進行驗證。

使用 Rabin 簽名之具體實例

介紹

【0057】 數位簽名為比特幣協定之基本部分。其確保區塊鏈上記錄之任何比特幣交易皆已由被發送比特幣之合法持有者授權。在標準比特幣 P2PKH 交易中，使用橢圓曲線數位簽名演算法(ECDSA)簽署交易消息。然而，ECDSA 簽名大體上應用於整個交易。

【0058】 在比特幣區塊鏈之一些使用情況中，來自網路外部之參與者可能想要為任意資料類型提供簽名，網路參與者接著可使用該資料類型。藉由使用 Rabin 數位簽名，可對任何資料段進行簽署-即使其來源於比特幣區塊鏈外部，且接著將簽名置放在一或多個交易中。

現將示出可如何藉由利用Rabin密碼系統之代數結構直接以比特幣指令碼簽署及驗證資料

Rabin 數位簽名

Rabin 數位簽名演算法

背景數學

定義-整數 mod p

整數模數$p$定義為如下集合

$$\mathbb{Z}_p := \{1,2,\dots,p-1\}$$

Fermat 之小定理

使$p$為質數。接著對於任何整數$a$，如下適用

$$a^{p-1} \equiv 1 \bmod p$$

Euler 之準則

使$p$為質數。當且僅當下式時，$r$為二次餘數 mod $p$

$$r^{\frac{p-1}{2}} \equiv 1 \bmod p$$

模組化平方根 (p = 3 mod 4)

使$p$為質數，使得$p \equiv 3 \bmod 4$。接著對於滿足 Euler 之準則的任何整數$r$，若$a$為整數，則使得

$$a^2 \equiv r \bmod p$$

接著$a$存在具有如下形式之解

$$a \equiv \pm r^{\frac{p+1}{4}} \bmod p$$

中國剩餘定理

給定成對之互質正整數$n_1, n_2, \dots, n_k$與任意整數$a_1, a_2, \dots, a_k$，聯立同餘系統

$$x \equiv a_1 \bmod n_1$$

$$x \equiv a_2 \bmod n_2$$

$$\vdots$$

$$x \equiv a_k \bmod n_k$$

具有唯一的解模數$N = n_1 n_2 \dots n_k$。作為中國剩餘定理之特殊情況，可示出

當且僅當

$$x \equiv r \bmod n_1 \cdot n_2$$

时，

$$x \equiv r \bmod n_1 \text{ 且 } x \equiv r \bmod n_2$$

Rabin 數位簽名演算法

可如下描述Rabin數位簽名演算法：

【0059】 對於任何消息$m$，使$H$為具有$k$個輸出位元之抗衝突散列演算法。

【0060】 為生成密鑰，選擇質數$p$及$q$，其各自具有大約$k/2$之位元長度，使得$p \equiv 3 \bmod 4$、$q \equiv 3 \bmod 4$，並計算乘積$n = p \cdot q$。私鑰為$(p, q)$且公鑰為$n = p \cdot q$。

【0061】 為簽署消息$m$，簽署者選擇填補$U$，使得$H(m||U)$滿足

$$H(m||U)^{\frac{p-1}{2}} \equiv 1 \bmod p$$

$$H(m||U)^{\frac{q-1}{2}} \equiv 1 \bmod q$$

【0062】 使用下式計算簽名S

$$S \equiv \left[\left(p^{q-2} \cdot H(m||U)^{\frac{q+1}{4}} \bmod q\right) \cdot p + \left(q^{p-2} \cdot H(m||U)^{\frac{p+1}{4}} \bmod p\right) \cdot q\right] \bmod n。$$

【0063】 消息$m$之簽名為對$(S, U)$。可藉由針對給定$m$、$U$及$S$檢查下式簡單地進行驗證

$$H(m||U) \equiv S^2 \bmod n \text{ (等式 1)。}$$

【0064】 當且僅當範圍$0, \ldots, n-1$中存在整數$\lambda$，使得

$$H(m||U) + \lambda \cdot n = S^2 \text{ (等式 2)}$$

時此為真。

【0065】 因數$\lambda$可安全地包括於簽名中，以提供組合$(S, \lambda, U)$。

【0066】 如下為 Rabin 簽名方案之有利特徵：

a) 簽名生成在計算上係昂貴的，而簽名驗證在計算上係容易的。

b) 簽名之安全性僅依賴於整數分解之難度。結果，Rabin 簽名在本質上係不可偽造的(不同於 RSA)。

c) 散列函數值$H(m||U)$必須具有公鑰$n$之類似量值。

【0067】 指令碼中之驗證係簡單明瞭的，因為其僅需要對給定簽名進行平方，執行模組化歸約，且接著檢查結果是否等於$H(m||U)$。

Rabin 簽名證明

使$p$、 $q$為互質數且$n = p \cdot q$。藉由中國剩餘定理，可示出當且僅當

$$S^2 \equiv H(m||U) \bmod p$$

$$S^2 \equiv H(m||U) \bmod q$$

時，

$$S^2 \equiv H(m||U) \bmod n$$

可示出

$$S^2 \equiv H(m||U) \bmod q$$

使用

$$S \equiv \left( \left( p^{q-2} H(m||U)^{\frac{q+1}{4}} \bmod q \right) \cdot p \; + \left( q^{p-2} H(m||U)^{\frac{p+1}{4}} \bmod p \right) \cdot q \right) \bmod q$$

$$\equiv \left( \left( p^{q-2} H(m||U)^{\frac{q+1}{4}} \bmod q \right) p \right) \bmod q$$

$$\equiv p^{q-2} H(m||U)^{\frac{q+1}{4}} . p \bmod \mathrm{q}$$

$$\equiv (p^{q-1} \bmod q \,) H(m||U)^{\frac{q+1}{4}} \bmod \mathrm{q}$$

$$\equiv H(m||U)^{\frac{q+1}{4}} \bmod \mathrm{q}$$

因此

$$S^2 \equiv H(m||U)^{\frac{q+1}{2}} \equiv \; H(m||U)^{\frac{q-1}{2}} \cdot \; H(m||U) \equiv \; H(m||U) \bmod q$$

其中已假定$H(m||U)$滿足 Euler 之準則。藉由類似計算，吾人亦可示出

$$S^2 \equiv H(m||U) \bmod p$$

比特幣中之 Rabin 簽名

指令碼中之簽名驗證

【0068】需要少量算術及堆疊運算作業碼以驗證 Rabin 簽名。考慮具有如下形式之兌換指令碼

OP_DUP OP_HASH160 <$H_{160}(n)$> OP_EQUALVERIFY OP_MUL OP_SWAP OP_2 OP_ROLL OP_CAT FUNC_HASH3072 OP_ADD OP_SWAP OP_DUP OP_MUL OP_EQUAL

其中$n$為簽署者之公鑰。當且僅當具備如下輸入時，此將評估為真

<$S$> <$U$> <$m$> <$\lambda$> <$n$>

其中$m$為任意消息，且$(S, \lambda, U)$為有效 Rabin 簽名。替代地，若使用上文等式 1 檢查 Rabin 簽名，則兌換指令碼由下式給出

OP_DUP OP_HASH160 <$H_{160}(n)$> OP_DUP OP_TOALTSTACK OP_SWAP <roll index> OP_ROLL OP_CAT FUNC_HASH3072 OP_SWAP OP_MOD OP_SWAP OP_DUP OP_MUL OP_FROMALTSTACK OP_MOD OP_EQUAL

【0069】在此情況下，當且僅當具備如下輸入時，指令碼將評估為真

<$S$> <$U$> <$m$> <$n$>

【0070】在兩兌換指令碼中，皆使用了3072位元散列投影函數'FUNC_HASH3072'。對於給定消息/填補串聯，使用指令碼生成FUNC_HASH3072散列投影

OP_SHA256 {OP_2 OP_SPLIT OP_SWAP OP_SHA256 OP_SWAP} (x11)

OP_SHA256 OP_SWAP OP_SHA256 {OP_CAT}(x11)

**資料壓縮**

【0071】網際網路資料由 JavaScript 及諸如文本檔案(SML、HTML 等)、視

訊檔案(MPEG、M-JPEG 等)、影像檔案(GIF、JPEG 等)及音訊檔案(AU、WAV 等)之常見檔案類型組成，例如如 https://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/mmp/#text 處較詳細描述。使用上文之資料插入技術，此等不同資料類型亦可經嵌入於區塊鏈上。

【0072】 在將較大檔案大小嵌入於區塊鏈上之前，可使用若干現有譯碼方案中之一者進行壓縮。諸如游程長度及 Huffman 編碼之無損資料壓縮演算法可用於若干應用，包括 ZIP 檔案、可執行程式、文本文件及原始程式碼。

【0073】 取決於特定輸入資料，存在許多不同的演算法。蘋果無損及自適應變換聲譯碼可用於壓縮音訊檔案，PNG 及 TIFF 用於壓縮圖形檔案，而電影檔案可使用許多無損視訊編解碼器中之一者進行壓縮。可使用屬性內之旗標指示資料內容之任何壓縮。例如，屬性中用於 LZW 無損譯碼方案之旗標將為<LZW>。

**加密及經支付解密**

資料加密

【0074】 內容之所有者可選擇在將內容嵌入於區塊鏈上之前保護內容。此確保在未獲取必要權限之情況下無法檢視內容。

【0075】 存在許多公認之資料加密技術(明文或其他資料類型)。此等技術可經歸類為非對稱加密或對稱加密。

【0076】 橢圓曲線密碼術(ECC)係非對稱的，因為其依賴於公私鑰對。其為最安全之密碼系統中之一者且通常用於諸如比特幣之密碼貨幣。對於 ECC 密碼術，Koblitz 演算法可用於加密資料。

【0077】 在對稱方案中，單個密鑰用於加密及解密資料兩者。高級加密標準(AES)演算法被視為由此秘密產生之最安全對稱演算法中之一者，例如如 C. Paar 及 J. Pelzl 在「Understanding Cryptography」第 4 章(斯普林格弗拉格出版社，柏林海德堡，第 2 版，2010 年，第 87 至 118 頁)中較詳細描述。

【0078】 當加密儲存於區塊鏈上之資料時，使用與基礎區塊鏈相同之密碼系統存在優勢。在比特幣中，此為非對稱密碼術中之 ECC 密鑰對，及對稱密碼術中之 SHA-256 散列函數的 secp256k1 慣例。此等優勢為：

- 加密之安全位準相同於上面儲存有資料之基礎系統。

- 儲存加密資料所需之軟體架構將具有較小程式碼庫。

- 錢包中之密鑰管理可用於交易及加密/解密兩者。

- 其更有效，因為相同密鑰可用於密碼貨幣中之加密及支付兩者，因此需要較少密鑰。此亦減小儲存空間。

- 可需要較少通訊頻道以交換/購買解密資料之能力。

- 安全性增大，因為用於加密及交易之密鑰為相同資料結構，且因此減輕了對特定密鑰類型之定向攻擊。

- 可使用基礎密碼貨幣購買密鑰。

【0079】 出於例示性目的，描述可如何使用Koblitz演算法以使用ECC來加密資料。

Koblitz 演算法

【0080】 給定 ECC 密鑰對$P_1 = S_1 \cdot G$，Koblitz 演算法允許任何人使用公鑰$P_1$來加密消息，使得僅已知對應私鑰$S_1$之某人可解密消息。

【0081】 假設期望使用 Koblitz 方法加密明文消息$'hello\ world'$。此係逐字符進行的。如下加密及解密第一字符$'h'$。

1.將字符$'h'$映射至secp256k1曲線上之點。此係藉由使用ASCII慣例以將明文字符映射至8位元數字來實現。接著藉由使基點$G$乘以此數字計算出曲線上之點。在本發明實例中，$'h'$經映射至ASCII中之104，且橢圓曲線點由 $P_m = 104 \cdot G$ 給出。

2.接著使用公鑰 $P_1$加密點 $P_m$。此係藉由選擇隨機暫時密鑰$k_0$並計算點對

$C_m = \{k_0 \cdot G, Q\}$(其中$Q \coloneqq P_m + k_0 \cdot P_1$)來實現，接著可廣播該等點對。

3.私鑰$S_1$之所有者可藉由計算$P_m = Q - S_1 \cdot k_0 \cdot G$來解密原始點。其接著可藉由試誤法或借助於查找表恢復原始ASCII數字以確立哪個數字$x$對應於$P_m = x \cdot G$。

使用區塊鏈來購買權限

【0082】 將資料儲存於區塊鏈上具有支付機制構建於系統中之明顯優勢。支付可用於購買

- 解密資料以便檢視/使用

- 在特定位址處插入資料之權限

【0083】 在兩情況下，買方皆使用例如比特幣之密碼貨幣來購買授予其進行某事之權限的秘密。此秘密可為散列原像或私鑰。

【0084】 進行此購買之有效安全方式為使用原子調換。此將安全通訊頻道保持在最低限度，並確保向賣方支付且向買方揭露秘密，或確保此等事件均不發生。

【0085】 除了以密碼貨幣支付外，亦可便利地使用存取符記購買權限。此為買方擁有的其可使用以便進行購買的秘密值(通常為散列原像)。買方可事先大量購買此等符記，且接著在其實際上希望使用權限時進行激活。

【0086】 現將參考圖 4 及圖 5 描述如何執行原子調換。

使用散列謎題或私鑰謎題之原子調換

【0087】 假設 Alice 為秘密之所有者。此秘密可為已知散列摘錄之散列原像，或已知公鑰之私鑰。假設 Bob 希望使用比特幣以自 Alice 購買此秘密。描述實現此交易的稱為原子調換之機制。原子意為 Alice 獲得比特幣支付且向 Bob 揭露秘密，或此等事件均不發生。

方法如下：

【0088】 Alice擁有公/私鑰對$P_A = S_A \cdot G$之私鑰$S_A$，且Bob擁有公/私鑰對$P_B = S_B \cdot G$之私鑰$S_B$。

【0089】 Alice擁有為已知散列摘錄$H(X)$之原像$X$或已知公鑰$P_1 = S_1 \cdot G$之私鑰$S_1$的秘密。

【0090】 他們同意Alice以一比特幣價格將秘密出售給Bob。

【0091】 在此之前，Bob必須設置交易以在區塊外將暫時密鑰$k_0$發送給Alice，從而使得Alice可計算數位簽名之分量$r_0$。

【0092】 現參考圖 4，

1.Bob 將藉由以下兌換指令碼鎖定之比特幣傳送至 Alice

$R$ (示意性地撰寫)：

對於散列原像：

$R$ = [散列謎題 $H(X)$] [CheckSig $P_A$]

【0093】 此迫使原像$X$在兌換指令碼之輸入中曝露。

對於私鑰：

$R$ = [私鑰謎題 $P_1, r_0$] [CheckSig $P_A$]

【0094】 此迫使能夠自至兌換指令碼之輸入計算出私鑰$S_1$。在此情況下，Bob及Alice必須同意用於構造$r_0$之暫時密鑰$k_0$，其中$(r_0, R_y) = k_0 \cdot G$。

【0095】 2.由於Alice已知其秘密($X$或$S_1$)，因此其可借助於圖5中所示之交易花費其在比特幣區塊鏈上之資金。此允許Bob判定Alice之秘密。

【0096】 作為可選安全特徵，Alice及Bob可使用其公鑰$P_A$、$P_B$以確立僅兩方已知之共用秘密$S$。此可以國際專利公開案第WO 2017/145016號中概述之方式實現。在此情況下，$S$可經添加至散列謎題中之原像$X$，以便不在區塊鏈上公開揭露$X$。類似地，在私鑰謎題中，$S$可用作暫時密鑰$k_0$以確保僅Alice或Bob能夠計算私鑰。

【0097】 若Alice決定不花費其資金，可在程序中引入時間鎖定退款以防止Bob之資金被Alice鎖定。

使用符記之購買

【0098】 假設存在如上文所描述之相同情況，但並非在使用時以密碼貨幣支付Alice之秘密，Bob想兌換存取符記——其已事先購買——以交換秘密。

【0099】 Alice及Bob必須遵循之程序類似於先前章節中描述之情況，但實際上使用了一序列類似原子調換。該過程存在兩個階段；符記發佈及符記兌換。

**階段 1：符記發佈**

【0100】 符記發佈階段實際上為Bob單次購買符記。例如，考慮如下情境：Alice具有10個不同的秘密$X_1$，$X_2$，…，$X_{10}$，且Bob希望單次購買各自授予其對各別秘密之存取權的10個符記$T_1$，$T_2$，…，$T_{10}$。

【0101】 首先，Bob 自僅其已知之秘密種子值$Y$生成一組 10 個符記。此等符記係藉由種子之依序散列創建以形成散列鏈，其中每一符記經計算為

$$T_i = H^{10-i}(Y) \quad \text{對於} \quad i \in \{1, 2, \ldots, 10\} \text{。}$$

【0102】 Alice 及 Bob 現在各自具有 10 個秘密值，可在散列謎題中揭露該等秘密值以用於例如兌換符記。然而，為了發佈此等符記，其必須亦分別生成秘密初始化值$I_{Alice}$及$I_{Bob}$。此等值給出為

$$I_{Alice} = k, \qquad k \in \mathbb{Z}_{256} \text{，}$$

$$I_{Bob} = H^{10}(Y) \text{。}$$

【0103】 應注意，Alice 之初始化值簡單地為無特定含義之隨機整數，但 Bob 之初始化值應為其第一符記$T_1 = H^9(Y)$之散列。以此方式將符記鏈擴展至初始化值允許符記發佈亦定義待稍後用於連續兌換之符記。圖 6 中示出由每一參與者保存之全部秘密值。

【0104】 現在 Alice 及 Bob 可同意以 10 個密碼貨幣單位之價格購買 10 個

符記。可以數種方式購買此等符記，此處使用原子調換說明此方式。原子調換藉由 Alice 及 Bob 分別廣播圖 7 及圖 8 中所示之交易起始，在兩交易中輸出皆需要兩個散列謎題之解及有效簽名。

【0105】 一旦區塊鏈中出現兩交易，Alice 及 Bob 可共用其共用初始化值$I_{Alice}$及$I_{Bob}$，並完成符記發佈之原子調換。

【0106】 由於此原子調換，Alice 接收購買 10 個符記之支付且兩初始化值秘密皆經揭露。應注意，此處僅 Bob 之秘密$I_{Bob} = H^{10}(Y)$有意義，因為其將定義待求解之第一散列謎題[散列謎題$(T_1)$]，其解為初始化值 $H^{10}(Y)$之原像 $H^9(Y)$。

**階段 2：符記兌換**

【0107】 在未來的某一時刻，Bob 想要兌換其第一符記$T_1 = H^9(Y)$並接收其第一秘密$X_1$，但前已述及其已藉由購買有效符記支付了此秘密。兌換符記之過程將呈另一原子調換之形式，其中鎖定散列謎題之解為符記$T_i$及對應秘密$X_i$。

【0108】 為兌換其符記，Bob 應廣播圖 9 中所示之交易，該交易之輸出藉由兩個散列謎題鎖定。當 Alice 看到此交易時，她廣播如圖 10 中所示的其自身之類似交易，該交易之輸出藉由相同兩個散列謎題鎖定。兩個參與者現在可交換其秘密$T_1$及$X_1$並解鎖此等交易之輸出。兩方現在皆可藉由提供亦曝露兩秘密之正確解除鎖定指令碼來兌換標稱費用$x$。圖 11 及圖 12 中示出具有此等解除鎖定指令碼之交易。

【0109】 用於兌換符記之此原子調換的完成向 Bob 揭露 Alice 之第一秘密$X_1$，向 Alice 揭露 Bob 之第一符記$T_1$，且鑒於金額$x$適當大以鼓勵兩方花費鎖定之輸出，具有密碼貨幣資金之淨零交換。至關重要地，此亦確立 Bob 可使用之下一符記必須為散列謎題[散列謎題$H(T_2)$]之解$T_2$，其中剛已向 Alice 揭露目標散列$H(T_2) = T_1$。可以遞歸方式重複此過程直至 Bob 已使用其最終符記$T_{10} = Y$為

止。

**命名及定址**

節點及邊緣結構

【0110】 上文已解釋可如何藉由在交易內提供資料而將資料插入至區塊鏈中。現在呈現用於以邏輯方式結構化此等交易之協定，該方式允許對節點、權限及內容版本控制進行定址。此分佈式同級元網路之結構類似於現有網際網路。

【0111】 應注意，此為並不修改基礎區塊鏈之協定或共識規則之「階層-2」協定。

【0112】 此處所描述之結構的目標為

(i) 相關聯不同交易中之相關內容以實現對資料之搜尋、識別及存取

(ii) 允許使用人類可讀關鍵詞搜尋識別內容，以改良搜尋之速度、準確性及效率

(iii) 在區塊鏈內構建並模擬伺服器狀結構

方法為將與元網路相關聯之資料結構化為定向圖。此圖之節點及邊緣對應於：

【0113】 **節點**-與元網路協定相關聯之交易。節點儲存內容。(術語「內容」及「資料」在此文件內可互換使用)。

【0114】 節點藉由包括由<元網路旗標>緊隨之OP_RETURN而創建。每一節點經指派有公鑰$P_{node}$。公鑰與交易ID之組合唯一地指定節點之索引$ID_{node} := H(P_{node}||TxID_{node})$。

【0115】 所使用之散列函數應符合本發明待使用之基礎區塊鏈協定，例如用於比特幣之SHA-256或RIPEMD-160。

【0116】 **邊緣**-子節點與父節點之相關聯。

【0117】 邊緣在簽名$Sig\ P_{parent}$出現在元網路交易之輸入中時經創建，且

因此僅父代可給予創建邊緣之權限。所有節點可具有至多一個父代，且父節點可具有任意數目個子代。在圖論之語言中，每一節點之入度至多為1，且每一節點之出度係任意的。

【0118】 應注意，邊緣為元網路協定之態樣且其自身並非與基礎區塊鏈相關聯之交易。

【0119】 由具有以下形式之交易給出有效元網路節點(具有父代)：

| $TxID_{node}$ | |
|---|---|
| 輸入 | 輸出 |
| < $Sig\ P_{parent}$> <$P_{parent}$> | OP_RETURN <元網路旗標> <$P_{node}$> <$TxID_{parent}$> |

【0120】 此交易含有指定節點及其父代之索引所需的所有資訊

$ID_{node} = H(P_{node}||TxID_{node})$ ， $ID_{parent} = H(P_{parent}||TxID_{parent})$ 。

【0121】 此外，由於需要父節點之簽名，因此僅父代可創建至子代之邊緣。若<$TxID_{parent}$>欄位並不存在或其並不指向有效元網路交易，則節點係孤立的。其不具有可到達的較高層級節點。

【0122】 額外屬性可經添加至每一節點。此等屬性可包括旗標、名稱及關鍵詞。稍後在此文件中論述此等屬性。

【0123】 如所示，節點(交易)之索引可分解成

a) 公鑰$P_{node}$，其經解譯為節點之位址

b) 交易 ID $TxID_{node}$，其經解譯為節點之版本

【0124】 自此結構化產生兩個有利特徵：

1.**版本控制**-若存在具有相同公鑰之兩個節點，則吾人將具有具最大工作證據之交易 ID 的節點解譯為該節點之最新版本。若節點在不同區塊中，則可藉由區塊高度檢查此情況。對於相同區塊中之交易，藉由拓樸交易排序規則(TTOR)來判定此情況。

2.**許可**-僅當公鑰$P_{node}$之所有者在創建子節點時簽署交易輸入時，才可創建節點之子代。因此，$P_{node}$不僅表示節點之位址且還表示創建子節點之權限。此有意地類似於標準比特幣交易——不僅位址中之公鑰，而且與該位址相關聯之許可。

【0125】 應注意，由於父節點之簽名出現在UXTO解除鎖定指令碼中，因此在網路接受交易時通過標準挖掘者驗證過程進行驗證。此意謂創建子節點之權限由比特幣網路自身驗證。

【0126】 值得注意的為，標準網際網路協定(IP)位址僅在某一時刻處在網路內係唯一的。另一方面，元網路中之節點的索引在所有時間皆係唯一的，且不存在單獨網路概念，此允許資料永久性地錨定至單個對象$ID_{node}$。

【0127】 節點及邊緣結構允許吾人將元網路視覺化為圖，如圖 13 中所示。

**元網路內之域、命名及內容定位**

【0128】 元網路圖之階層允許出現豐富的域狀結構。吾人將孤立節點解譯為頂級域(TLD)，將孤立節點之子代解譯為子域，將孫代解譯為子子域等，且將無子節點解譯為端點。參見圖 13。

【0129】 域名經解譯為$ID_{node}$。元網路中之每一頂級域可被視為樹，該樹具有為孤立節點之根及為無子節點之葉。元網路自身為形成圖之樹的全域集合。

【0130】 元網路協定並不規定任何節點皆含有內容資料，但葉(無子)節點表示資料樹上之定向路徑的末端，且因此將大體上用以儲存內容資料。然而，內容可儲存於樹中之任何節點處。節點中作為屬性包括之協定特定旗標可用於指定資料樹中之節點的作用(磁碟空間、文件夾、檔案或許可改變)。

【0131】 前已述及網際網路使用域名系統(DNS)來將人類可讀名稱相關聯至網際網路協定(IP)位址。DNS 在某種意義上係去中心化的，儘管在實踐中其由少量關鍵參與者(諸如政府及大公司)控制。取決於你的 DNS 提供商，同一名稱

可將你帶至不同位址。在將人類可讀的短名稱映射至電腦生成之數字時，此問題係固有的。

【0132】 吾人假定存在將人類可讀之頂級域名映射至根節點之去中心化索引$ID_{root}$的等效分佈式系統。換言之，存在將人類可讀名稱映射至元網路根節點索引之 1-1 函數$\kappa$，例如

$$\kappa('bobsblog') = ID_{bobsblog}\left(= H\left(P_{bobsblog}||TxID_{bobsblog}\right)\right)。$$

【0133】 至左手側之輸入為人類可讀詞，而右手側上之輸出為散列摘錄，其將通常為 256 位元資料結構。應注意，大體而言，$P_{bobsblog}$及$TxID_{bobsblog}$亦為人類不可讀的。在標準 IP 協定中，此將為自$www.bobsblog.com$至網路內之對應域的 IP 位址之映射。

【0134】 映射$\kappa$應經解譯為在複製 DNS 發佈之域名的人類可讀性時，確保元網路與網際網路之向後兼容性的措施，但提供元網路之結構的命名及定址方案並不明確地取決於此映射。

【0135】 映射函數$\kappa$之可能現有形式包括由星際檔案系統(IPFS)或 OpenNIC 服務(https://www.openic.org)採用之 DNSLink 系統。此映射可作為 DNS 之部分儲存在現有 TXT 記錄中。此類似於 IPFS 中之 DNSLink——參見 https://docs.ipfs.io/guides/concepts/dnslink/。然而，大體而言，此等形式犧牲一些去中心化元素以便提供 1-1 之映射——參見 https://hackernoon.com/ten-terrible-attempts-to-make-the-inter-planetary-file-system-human-friendly-e4e95df0c6fa

無用位址

【0136】 用作元網路節點之位址的公鑰並非人類可讀之對象。此可使人類使用者之搜尋、參考及輸入活動易於出錯且很慢。然而，有可能創建人類可辨識之公鑰位址——無用位址$P_{vanity}$——其包括可由使用者直接解譯之明文首

碼。先前技術中已知無用位址。

【0137】 創建此位址之難度取決於所要首碼之字符長度。此意謂人類可辨識位址可用作僅依賴於所有者之創建工作量而非中心發佈之節點位址。對於給定首碼，歸因於尾碼中剩餘之字符，存在許多不同的無用位址，且因此許多節點位址可共用共同首碼，同時仍保持唯一性。

【0138】 具有合乎需要之首碼的無用位址之實例為

$P_{bobsblog}$： bobsblogHtKNngkdXEeobR76b53LETtpyT

首碼： bobsblog

尾碼： HtKNngkdXEeobR76b53LETtpyT

【0139】 上文之無用位址可用於感測檢查自名稱'bobsblog'至節點索引$ID_{bobsblog}$之映射並輔助元網路節點藉由位址之搜尋。應注意，首碼在此處並非唯一的，但整個位址自身係唯一實體。

【0140】 所選擇位址$P_{vanity}$與一起形成$ID_{node}$之$TxID$的組合亦係有益的，因為其意謂不存在域名之中心發佈者(由去中心化之工作量證明生成$TxID$)且名稱可自區塊鏈自身恢復。有利地，網際網路 DNS 內不再存在故障點。

【0141】 由於元網路域已提供權限系統(公鑰)，因此不需要發佈憑證以證明所有權。已例如在域名幣(https://namecoin.org/)中探索將區塊鏈用於此目的。然而，根據本發明，不需要將單獨的區塊鏈用於此功能，因為在一個區塊鏈內能實現所有事情。

【0142】 相比先前技術，此明顯減小本發明所需要之資源(硬體、處理資源及能量)的量。就系統組件之設備及配置而言，其亦提供完全不同之架構。

【0143】 此命名系統之優勢為使用者能夠藉由可記住詞(例如公司名稱)而非散列摘錄識別元網路中之頂級域。此亦使域搜尋較快，因為搜尋關鍵詞而非散列摘錄較快。此亦減小輸入錯誤，從而因此提供區塊鏈儲存資料之經改良搜

尋工具。

【0144】鑒於具有自域名至節點索引之映射，吾人可建立類似於網際網路之統一資源定位符(URL)的資源定位符。吾人將此稱為元網路 URL(MURL)，且呈如下形式

【0145】MURL = 'mnp:' +′//domain name′+ ′/path′ + ′/file′。

【0146】URL之每一分量——協定、域名、路徑及檔案——已映射至MURL之結構，從而使對象更具使用者直觀性且能夠與網際網路之現有結構整合。

【0147】此假定每一節點皆具有與其公鑰(位址)相關聯之名稱，該公鑰在域樹內之層級處係唯一的。此名稱始終為給定節點之 MURL 的最右側分量。若樹中之同一層級處的兩個節點具有相同名稱，則其將具有相同公鑰且因此獲得最新版本。

【0148】下表給出元網路協定與網際網路協定之間的類比：

| 網際網路 | 元網路 |
|---|---|
| **網站/檔案** | **節點** |
| **所有者** | **公鑰**<br>$P_{node}$ |
| **IP 位址(非唯一)** | **節點索引(唯一)**<br>$ID_{node} = H(P_{node}\|\|TxID_{node})$ |
| **域結構** | **節點樹結構** |
| **域名系統** | **自根節點名稱至節點索引之映射** |
| **URL**<br>http://www.bobsblog.com/path/file | **MURL**<br>mnp://bobsblog/path/file |

*表格：網際網路與元網路協定之間的類比彙總。*

**搜尋元網路**

【0149】吾人已定義元網路圖結構之例示性實施例，使得每一節點皆具有唯一索引且可具有歸於其的名稱。此允許使用 MURL 定位內容。為了亦實現快速搜尋功能性，吾人允許將額外關鍵詞歸於節點。

【0150】節點之固定屬性為索引及父節點之索引，且可選屬性為名稱及關鍵詞。

節點屬性

{

索引：　$H(P_{node}||TxID_{node})$；

父代之索引：　$H(P_{parent}||TxID_{parent})$；　(若孤立，則為空值)

名稱：　'bobsblog'；

kwd1：　'travel'；

kwd2：　'barbados'；

⋮

}

【0151】在一個實例中，用於搜尋元網路之切實可行方法可為首先使用區塊探測器在區塊鏈中搜查，並藉由元網路旗標識別所有交易，檢查其是否為有效元網路節點，且若如此，則在資料庫或其他儲存資源中記錄其索引及關鍵詞。此資料庫接著可用於藉由所要關鍵詞有效地搜尋節點。一旦藉由所要關鍵詞找到節點之索引，可自區塊探測器提取其內容並進行檢視。

【0152】借助於實例，考慮圖 14 之分支$P_1$，其中對應於公鑰$P_0$、$P_1$及$P_{1,1}$之節點分別表示首頁、主題頁及子主題頁。此等節點給定有名稱'bobsblog'、'summer'及'caribbean'，且下文示出其屬性

首頁節點$P_0$

MURL：　mnp://bobsblog

{

索引：　$H(P_0||TxID_0)$；

父代之索引：　空值

名稱： 'bobsblog'；

kwd1： 'travel'；

kwd2： 'barbados'；

⋮

}

主題頁節點$P_1$

MURL： mnp://bobsblog/summer

{

索引： $H(P_1||TxID_1)$；

父代之索引： $H(P_0||TxID_0)$；

名稱： 'summer'；

kwd1： 'travel'；

kwd2； 'barbados'；

⋮

}

子主題頁節點$P_{1,1}$

MURL： mnp://bobsblog/summer/caribbean

{

索引： $H(P_{1,1}||TxID_{1,1})$；

父代之索引： $H(P_1||TxID_1)$；

名稱： 'caribbean'；

kwd1： 'travel'；

kwd2： 'barbados'；

⋮

}

【0153】 在此實例中，葉節點 $P_{1,1,1}$、$P_{1,1,2}$ 及 $P_{1,1,3}$ 分別給定有名稱'beaches'、'nightlife'及'food'且用以儲存單獨的部落格文章。次頁圖式上示出全域結構，包括關於樹中之每一節點的 MURL 搜尋路徑。

【0154】 吾人應注意，元網路亦可藉由將由節點交易儲存之內容的散列儲存為額外屬性而併入內容可定址網路(CAN)。此意謂元網路節點亦可為索引式並藉由內容散列搜尋。

【0155】 上文所描述之命名及定址方法提供優於先前技術之眾多技術優勢，包括：

1. **公鑰位址**-系統使用與區塊鏈相同之公私鑰對來指派節點位址。此意謂相同密鑰集合用於管理密碼貨幣資金及內容資料許可兩者。此提供有效安全之解決方案。

2. **去中心化域**-域名之發佈通過包括僅可由工作量證明生成之$TxID_{node}$完全去中心化。域名亦可併入實現所要域公鑰之公平分佈的人類可辨識公鑰 $P_{vanity}$(無用位址)。再次，此解決方案增強效率及安全性。

3. 圖結構-命名及定址架構指定可自包含元網路節點之區塊鏈資料子集構造的圖。此設計使用有序結構將網際網路之複雜性映射至區塊鏈，使得區塊鏈完全複製其功能性及可調性，同時保持安全性。

**瀏覽器錢包應用程式**

【0156】 前已述及在元網路協定中，所有資料皆直接在區塊鏈自身上。在此章節中，吾人呈現可有效地存取、顯示並與儲存於區塊鏈上之元網路資料交互的例示性電腦應用程式之實施例，本文中為方便起見僅稱為「瀏覽器錢包」。

【0157】 吾人將首先論述核心組件及瀏覽器錢包如何與分佈式同級網際網路介接之功能性，之後在此章節之剩餘部分中提供較詳細描述。

**概述**

組件

【0158】 瀏覽器錢包為意欲允許終端使用者與區塊鏈上之元網路基礎設施交互之應用程式。此應用程式應允許探索式搜尋元網路圖以找到嵌入於樹中之特定內容。另外，瀏覽器錢包將處置內容之提取、解密、重組及快取(可選的)。

【0159】 瀏覽器錢包應用程式將藉由支援本地(或外部)錢包而組合此等元件與密碼貨幣支付機制。瀏覽器錢包將包含組合為單個電腦應用程式的以下核心元件。

【0160】 **區塊鏈搜尋引擎** - 支援第三方搜尋引擎藉由包括$ID_{node}$、節點名稱、關鍵詞、區塊高度及$TxID$之多種索引查詢元網路節點。

【0161】 **顯示窗** - 解包封由全複本區塊鏈同級者傳回至瀏覽器之內容的軟體。此涵蓋解密、重組、快取及兌換存取符記。

【0162】 **密碼貨幣錢包** - 用於區塊鏈之貨幣的專用密鑰管理。可在應用程式本地或經授權以與外部錢包(軟體或硬體)通訊及同步。能夠寫入標準區塊鏈交易以及新元網路節點交易。可調解存取密鑰及存取符記之鏈上購買。

【0163】 階層式確定性密鑰管理用於密碼貨幣公鑰及元網路節點位址兩者。

【0164】 **存取密鑰/符記錢包** - 用於購買存取密鑰或符記之專用密鑰管理。可使用密碼貨幣錢包接收購買之密鑰或符記但對密鑰或符記並不具有權限。密鑰或符記可對使用者隱藏以允許稍後到期。此可通過使用可信執行環境來實現。可藉由與區塊鏈同步並查詢當前區塊高度來保證定時存取。

**功能性**

【0165】 元網路瀏覽器錢包之規範確保了應用程式之以下功能性。

【0166】 1. **階層式密鑰管理** - 用於控制資金及管理元網路樹(圖)之密鑰利

用相同的階層式確定性密鑰基礎設施，從而減小使用者維持其元網路內容之密鑰記錄的負擔。

【0167】 2. **指向外部密碼貨幣錢包 -** 授權及與外部(非應用程式本地)錢包同步之能力藉由移除作為故障點之瀏覽器錢包而允許額外安全性。

【0168】 3. 應用程式可寫入區塊鏈交易，且需要容納密鑰之外部錢包的簽名，從而將此職責委託至單獨軟體或硬體。

【0169】 4. **元網路內容之搜尋** - 瀏覽器錢包可支援及查詢第三方搜尋引擎，該引擎之功能可包含耙梳、編索引、服務及評級全域資料庫中之元網路節點交易資料。可構造含有元網路協定旗標之 OP_RETURN 交易的資料庫。參見 BitDB 2.0 - https://bitdb.network/。

【0170】 5. 搜尋引擎可藉由節點索引伺服瀏覽器錢包，此允許找到資料。

【0171】 6. **讀取資料及將資料寫入至區塊鏈 -** 除了使用搜尋引擎及全節點以藉由內容伺服瀏覽器之外，支援密碼貨幣錢包亦允許將內容直接自瀏覽器錢包寫入至元網路中。

【0172】 7. **資料之解壓縮及解密** - 瀏覽器錢包處置解密密鑰且可就地對元網路內容執行解壓縮。

【0173】 8. **快取節點識別碼($ID_{node}$) -** 可在本端快取唯一節點識別碼以用於較有效之查找及查詢。

【0174】 9. **旁路網路伺服器 -** 在給定節點索引情況下，瀏覽器錢包可查詢點對點(P2P)區塊鏈網路之任何全複本成員以獲取位於節點處之內容。因為元網路在鏈上，所以任何全複本同級者必須具有節點及其內容之本端複本。

【0175】 此意謂使用者之瀏覽器錢包僅需要查詢單個同級者，此操作可直接進行且無需中間網路伺服器。

【0176】 圖15示出瀏覽器錢包及如何跨越應用程式之不同組件分割其核心

功能的示意圖。

**區塊鏈搜尋引擎**

**搜尋引擎 - 現有技術**

【0177】 如先前技術中已知之搜尋引擎(SE)依賴於強大網路耙梳程式以根據使用者查詢定位、編索引及評級網路內容。(相同基礎原理可擴展至對元網路進行耙梳之第三方區塊鏈 SE)。

【0178】 SE 通過查詢中之關鍵詞搜尋識別相關 HTML 元標籤及內容。隨後編索引耙梳結果，其中分析及編錄任何嵌入之影像/視訊/媒體檔案。接著在考慮使用者之位置、語言及裝置的情況下以規劃方式評級來自索引之最相關結果。

典型SE應具有以下功能性：

【0179】 1. **耙梳** - 識別網際網路資料並通過諸如域名、鏈接頁面及相關關鍵詞之相關元資料進行耙梳。通過現有內容發現新網際網路內容並亦針對任何相關資訊進行耙梳。

2. **編索引** - 分析及編錄內容資料。此資訊儲存在資料庫中。

3. **服務及評級** - 以與使用者查詢之相關性評級內容索引。

**區塊探測器**

【0180】 最接近網際網路搜尋引擎(SE)之區塊鏈類似物為*區塊鏈探測器*，其有時被稱作‘區塊探測器’或‘區塊鏈瀏覽器’。區塊鏈探測器為可在高層級上對區塊鏈進行使用者友好查詢之網路應用程式，且類似於網路瀏覽器起作用但連接至區塊鏈而非網際網路。參見 https://en.bitcoin.it/wiki/Block_chain_browser。

【0181】 在大多數情況下，此等探測器允許將區塊(由區塊標頭之散列編索引)、交易(由$TxID$編索引)、位址及未用交易輸出(UTXO)作為輸入並搜尋前述各項。許多探測器亦提供其自身之應用程式規劃介面(API)以用於提取原始交易及

區塊資料。參見 https://blockexplorer.com/api-ref。

【0182】 區塊探測器雖然能力不同，但大體上用於以使用者易於摘錄之形式編錄交易並顯示其基本資訊——諸如交易貨幣值、幣之確認及歷史以及位址。諸如 Bitcoin.com https://explorer.bitcoin.com/bch 及 Blockchain.com https://www.blockchain.com/explorer 之許多探測器亦允許檢視交易之個別輸入及鎖定指令碼，但此等及如 Blockchair https://blockchair.com/之較高級站點在如何選擇以提供此資訊之間存在不一致。

【0183】 近來，用於基於區塊鏈資料運行網路應用程式之基本區塊鏈探測器存在許多擴展。諸如 Memo.cash https://memo.cash/protocol 及 Matter https://www.mttr.app/home 之此等應用程式如區塊探測器般對含有特定協定識別符之區塊鏈交易進行編錄及組織，以及顯示在彼等特定交易內編碼之資料。

【0184】 然而，使用區塊鏈探測器存在兩個重要問題，本發明之實施例解決了該等問題：

1. **廣用性** - 當前不存在用於瀏覽儲存在交易中之內容資料的行業標準。內容資料係指並不涉及用於創建及保證基礎區塊鏈之協定的任何資料。

2. **關鍵詞搜尋** - 儲存在交易中之內容資料需要可由人類可讀關鍵詞提取。此大體上並非當前區塊探測器之功能，因為其用於查詢交易之基於協定的性質，諸如區塊高度、$TxID$及位址而非將關鍵詞作為搜尋輸入。(然而，若詞直接包括於交易之指令碼中，則例如 Blockchair 之一些站點可搜尋該等詞)。

【0185】 重要的為，如上文所論述，本發明之強大命名及定址結構促進且實現構造相比此項技術中所已知較複雜之區塊鏈探測器。

**所提出元網路搜尋引擎**

【0186】 瀏覽器錢包應用程式與第三方搜尋引擎通訊以用於發現節點識別碼($ID_{node}$)。應設想到，此第三方可提供複製現有網際網路搜尋引擎之能力的強

大且多功能服務。

【0187】 元網路搜尋引擎第三方維持挖掘至區塊鏈中的可由元網路協定旗標識別之所有元網路交易的資料庫。此資料庫可藉由包括$ID_{node}$、節點名稱、密鑰詞、$TxID$及區塊高度之範圍索引編錄所有元網路節點。

【0188】 已存在諸如 Bit DB https://bitdb.network/之服務，其連續地與區塊鏈同步且以標準資料庫格式維持交易資料。瀏覽器錢包將對元網路交易進行耙梳、編索引、服務及評級之責任分擔至此第三方，並在定位儲存於元網路圖上之內容時連接至其服務。

【0189】 藉由具有僅專用於元網路資料之資料庫，可節約效率。不同於 Bit DB，此資料庫將不儲存與所有交易相關聯之資料，而僅儲存含有元網路旗標之彼等資料。諸如如 MongoDB 之非相關資料庫的某些資料庫在儲存元網路之圖結構時可較有效。此將允許較快查詢、較低儲存空間，及較有效地相關聯元網路域內之相關內容。

【0190】 圖 16 示出當使用者搜尋元網路基礎設施內之內容時，瀏覽器錢包如何與第三方搜尋引擎交互。重要的為，應注意，與網際網路對比，無需路由且因此本發明在效率、速度、處理及所需資源方面提供重要優勢。

【0191】 過程如下

1. 終端使用者將關鍵詞輸入至瀏覽器錢包搜尋列中。

2. 瀏覽器錢包將關鍵詞查詢發送至第三方 SE。

3. SE 針對其資料庫檢查關鍵詞並傳回含有相關內容之任何元網路節點的$ID_{node}$。第三方亦可向使用者傳回每一節點上之其他索引，以及提供對相關內容之建議。

4. 瀏覽器錢包使用節點識別碼及與其相關聯之域名以構造 MURL。

5. 瀏覽器錢包向具有區塊鏈之全複本的任何網路同級者請求屬於指定節點

之內容。

6. 網路同級者藉由所請求內容伺服瀏覽器錢包。因為同級者具有區塊鏈之複本，所以其必定亦具有內容之複本，且因此僅作出一個請求，且從不將請求轉遞至其他網路同級者。

【0192】 要強調的為，第三方 SE 僅負責編索引及維持元網路節點之屬性記錄，而儲存於節點上的原始內容資料實際上由具有區塊鏈之全複本的網路同級者(例如，全複本同級者、挖掘者、存檔)儲存。

**內容顯示器-元網路瀏覽器**

【0193】 瀏覽器錢包應用程式模擬任何典型網路瀏覽器應提供之相同前端能力。此等功能包括但不限於：

1.**搜尋 -** 提供對搜尋引擎(SE)之存取以用於定位內容。

2.**提取 -** 與伺服器通訊以促進使用例如超文本傳送協定(HTTP)之已知協定傳送內容。

3.**解譯 -** 解析原始程式碼(例如，以 JavaScript)並執行。

4.**顯現 -** 有效顯示待由終端使用者檢視之經解析內容。

5.**使用者介面(UI) -** 為使用者提供用以與內容交互之直觀介面，包括動作按鈕及用於使用者輸入之機制。

6.**儲存 -** 用於快取網際網路內容、小型文字檔等之本端臨時儲存容量，以改良對內容之重複存取。

【0194】 在某些實施例中，負責充當網路瀏覽器的瀏覽器錢包應用程式之軟體組件能夠對嵌入於區塊鏈中的可使用其屬性搜尋(使用 SE)且可使用其屬性提取(自同級者)之元網路內容執行上文功能。

**重組、解壓縮及解密**

【0195】 根據本發明之某些實施例，瀏覽器錢包應用程式之網路瀏覽器軟

體組件能夠處置需要對給定元網路內容執行之所有操作。大體而言，存在需要執行之許多此等操作，但吾人假定至少以下操作由應用程式使用元網路協定及基礎設施執行。

【0196】 **重組 -** 在元網路內容需要經分割並插入至多個單獨的節點交易中之情況下，應用程式將向所有相關節點請求內容並復原原始內容。可使用每一節點之屬性中之額外旗標編碼碎片內容之排序及結構。

【0197】 **解壓縮 -** 在內容資料以經壓縮形式儲存於區塊鏈上之情況下，應包括向瀏覽器錢包指示已使用哪一標準壓縮方案之旗標。應用程式將根據此旗標解壓縮內容。

【0198】 **解密 -** 在內容經加密之情況下，應使用表示加密方案之旗標。應用程式將自其解密密鑰錢包(如下文所論述)定位密鑰，並根據使用之加密方案解密內容資料以供使用。

【0199】 在對內容資料執行此等操作時，旗標可用於向瀏覽器錢包表示需要執行給定操作。此適用於任何其他操作，其中可將合適的<operation_flag>包括為該操作所應用的節點之屬性之部分。

**快取**

【0200】 快取本端檔案及小型文字檔為典型網路瀏覽器之共同且重要的功能。瀏覽器錢包應用程式亦以類似方式使用本端儲存器，以便可選地保存涉及所關注內容之$ID_{node}$及其他節點屬性的記錄。此允許自頻繁訪問之元網路節點較有效地查找及提取內容。

【0201】 元網路解決了快取網際網路資料之固有問題，該問題為可變的且可由網路瀏覽軟體取決於提供商進行改變或審查。在快取元網路資料時，使用者可始終容易地驗證資料與最初作為不可變記錄包括在區塊鏈上時處於相同狀態。

**密碼貨幣錢包**

**階層式確定性密鑰管理**

【0202】 確定性密鑰$Dk$為自單個「種子」密鑰初始化之私鑰(參見 Andreas M. Antonopoulos，「Mastering Bitcoin」第 5 章(奧萊利出版社，第 2 版，2017 年，第 93 至 98 頁))。種子為充當主鑰之任意生成的數字。散列函數可用於組合種子與其他資料(諸如索引數字或「鏈式程式碼」(參見 HD 錢包- BIP-32/BIP-44))以導出確定性密鑰。此等密鑰彼此相關且可藉由種子密鑰完全恢復。若使用者希望結合元網路瀏覽器錢包使用外部錢包，則種子亦准許在不同的錢包實施之間輕鬆導入/導出錢包，從而給予額外自由度。

【0203】 階層式確定性(HD)錢包為熟知之確定性密鑰導出方法。在 HD 錢包中，父代密鑰生成一序列子代密鑰，子代密鑰繼而導出一序列孫代密鑰等等。此樹狀結構為用於管理若干密鑰之強大機制。

【0204】 在較佳實施例中，HD 錢包可併入至圖 16 中所說明之元網路架構中。使用 HD 錢包之優勢包括：

1. **結構** 可出於不同目的使用子密鑰之不同分支表示額外組織含義。例如，使用者可將不同分支(及其對應子密鑰)專用於不同類型之資料。

【0205】 2. **安全性** 使用者可無需對應私鑰而創建一序列公鑰，從而使HD錢包具有僅接收能力且適於在不安全伺服器上使用。又，由於需要儲存較少秘密，存在較低曝露風險。

【0206】 3. **恢復** 若密鑰丟失/損毀，則可自種子密鑰恢復密鑰。

**本地(內部)及外部錢包支援**

【0207】 有利地，本發明之實施例可直接合併傳統網路瀏覽器與一或多個密碼貨幣錢包之功能性。根本上而言，此為元網路如何將「網際網路」內容之支付與向終端使用者之交付組合。

【0208】 為實現此情況，瀏覽器錢包之實施例可具有操作為密碼貨幣錢包之專用內置式軟體組件。此錢包係應用程式自身本地的且可用以管理密碼貨幣私鑰，並授權作為瀏覽器錢包自身內的元網路內容支付之交易。

【0209】 此意謂應用程式之瀏覽器組件可提示錢包組件授權所需之支付——藉由購買解密密鑰、存取符記或以其他方式——以檢視元網路內容。應用程式無需調用外部第三方以處理支付，且因此由應用程式就地耗用並支付所關注元網路內容。

**外部錢包**

【0210】 若使用者實際上希望在外部錢包(軟體或硬體)上管理或保存其密碼貨幣私鑰或甚至使用多個錢包，則可藉由應用程式之實施例實現相同優勢及功能性。此實施例可代替或結合應用程式之本地錢包執行。

【0211】 在此等實施例中，應用程式確立與外部錢包之鏈接或配對並與之同步，但並不在瀏覽器錢包自身中儲存私鑰。實際上，當瀏覽器組件提示支付內容時，應用程式向所選外部錢包請求藉由數位簽名進行授權。此授權由使用者作出且瀏覽器錢包可廣播交易並檢視經支付內容。

**讀取及寫入元網路交易**

【0212】 元網路之本質優勢為其使用相同資料結構——區塊鏈——來記錄支付及內容資料兩者。此意謂除了創建僅僅基於密碼貨幣之交換的交易之外，軟體錢包可用於將內容資料寫入至元網路基礎設施。

【0213】 內置至應用程式之本地錢包能夠將相比典型簡化支付驗證(SPV)用戶端較複雜之交易寫入至區塊鏈——參見 https://bitcoin.org/en/glossary/simplified-payment-verification。錢包允許使用者藉由自其電腦選擇待嵌入於區塊鏈中之內容資料而選擇將元網路節點交易直接自應用程式寫入至區塊鏈。

【0214】 由於瀏覽器錢包應用程式具有使用者介面(UI)，因此其允許錢包組件創建並廣播包括已預先在瀏覽器組件中或使用者電腦上構造之內容資料的交易。對於自行處置的專用錢包而言，將較難以實現此能力。

**存取密鑰/符記錢包**

【0215】 前已述及，元網路協定內置有使用 ECC 密鑰對或 AES 對稱密鑰加密內容之能力，及購買對應解密密鑰或符記之能力。吾人將此等密鑰或符記稱為*存取密鑰*或*存取符記*。

【0216】 此等密鑰/符記授予使用者檢視或編輯內容之權限(單次使用或多情況使用)，且與控制使用者密碼貨幣錢包之密鑰起著不同的作用(但在需要時同一密鑰可用於兩目的)。出於此原因，引入與應用程式之本地密碼貨幣錢包分離的用於儲存及管理存取密鑰及符記之新錢包係有利的。

【0217】 吾人亦可藉由允許存取密鑰/符記在某一時間週期之後經燒毀而引入對元網路內容之定時存取的概念。此可藉由需要存取密鑰/符記儲存在可信執行環境(TEE)中且使用者不可直接存取存取密鑰/符記而實現。

【0218】 存取密鑰/符記可經「燒毀」之實情亦為不將其儲存在密碼貨幣錢包中以確保不存在密碼貨幣私鑰被燒毀之風險的動機因素。

【0219】 以類似於密碼貨幣錢包之方式，可確定性地儲存及管理解密密鑰及存取符記以促進有效處置及部署。可藉由至主鑰之後續添加生成及恢復解密密鑰(例如，ECC 私鑰)，而可使用由一些初始符記播種之散列鏈重構存取符記。

【0220】 此處重要的為，應區分密碼貨幣錢包處置用於與其他使用者進行交易並創建新元網路節點的密鑰對之確定性密鑰生成，而密鑰/符記錢包處置已由密碼貨幣錢包購買之密鑰及符記。

**區塊高度許可**

【0221】 時間鎖可包括於比特幣指令碼語言中以實現區塊高度許可。

op_code OP_CHECKLOCKTIMEVERIFY (CLTV)設定准許使用交易輸出(UTXO)的區塊高度。

【0222】 區塊高度許可之優勢係雙重的：

1. **版本控制** - 在元網路協定中，可自最大區塊高度處之節點識別節點之最新版本。瀏覽器錢包可設置成藉由區塊高度僅顯示檔案之最近版本，從而實現工作量證明版本控制。

2. **定時存取** - 瀏覽器錢包應用程式可週期性地燒毀由使用者在原子級上購買之解密密鑰。此確保檢視者僅可在其已支付之時間週期期間存取內容資料。可藉由將解密密鑰儲存在可信執行環境(TEE)中來避免對解密密鑰之複製。此外，原子調換涉及確定性密鑰$Dk$(用於解密內容資料)之購買。儘管此確定性密鑰係公開可見的，但 TEE 可用於簽署$Dk$與安全包圍的私鑰之組合。

【0223】 瀏覽器錢包可配置成與區塊鏈之當前狀態同步，以便將區塊高度用作其自身之時間代理，而非依賴於任何外部時脈或第三方時間預告。

**旁路網路伺服器**

【0224】 本發明允許旁路域名系統(DNS)伺服器及典型網路路由程序的用於瀏覽器(用戶端)與網路伺服器在分佈式同級網際網路上通訊及交換資訊之新機制。參見 http://www.theshulers.com/whitepapers/internet_whitepaper/。本發明提供包含維持區塊鏈之全複本的同級者之新網路架構，瀏覽器錢包應用程式可自該新網路架構經伺服有內容。

本端全複本同級者

【0225】 考慮在例如郵區、城鎮、城市之每一地理區域中的本端同級者之系統。吾人假定在此區域網路內，至少一個同級者維持區塊鏈之全複本，吾人將該同級者稱為本端全複本同級者(LFCP)。出於吾人之目的，LFCP 僅需要儲存包括元網路旗標之區塊鏈交易，但不限於此。

【0226】所有使用者預設向 LFCP 發送‘獲得’請求。由於同級者維持整個區塊鏈之完整且最新複本，因此其可伺服所有請求，因為所查詢之任何節點 ID 將可用於 LFCP。應注意，若 SE 足夠強大且較大以儲存元網路內容及執行典型 SE 之主要功能，則元網路搜尋引擎亦可充當 LFCP。

【0227】在最簡單的情況下，每一 LFCP 將具有相同儲存及磁碟空間額外負荷，因為其將皆需要能夠儲存全區塊鏈(在寫入時約 200 GB)。每一 LFCP 之間的區別為其應縮放其能力以對來自元網路使用者之本端請求要求作出回應。因此，若全球之每一元網路使用者皆藉由預設查詢其最接近 LFCP，則每一 LCFP 皆應努力縮放其操作能力以滿足其本端需求。如城市之人口密集區域將需要包含許多集群伺服器之 LFCP 操作，而如小鎮之稀少區域將需要較少 LCFP 操作。

【0228】值得注意的為，磁碟空間要求係通用的，而每一 LFCP 之 CPU 要求適應於區域網路需求。此為*可調式網路*之實例，諸如 Freenet——參見 https://blockstack.org/papers/。

【0229】此系統之一個優勢為在提取與給定$ID_{node}$相關聯之內容時，使用者僅需要單次(本端)連接至其 LFCP。LFCP 無需將請求轉遞至其他同級者，因為其自身保證能夠伺服所需內容。

【0230】元網路提供優於網際網路之許多優勢——諸如去中心化及去除重複——類似於如 IFPS 之其他點對點(P2P)檔案共用服務。然而，元網路藉由確保不可變性，及至關重要地移除藉由對給定內容之請求充斥網路之需要而改良此等現有 P2P 模型。

【0231】元網路基礎設施亦藉由採用此等同級者之網路而綜合平衡任一個 LFCP。此意謂若停用一 LFCP，則終端使用者簡單地預設使用其下一最接近 LFCP。若 LFCP 彼此通訊以在任何給定時間處指示哪些附近同級者就請求而言容量低或高，則此可更有效。此可允許使用者將其請求發送至最適當同級者並

在附近 LFCP 之間確立請求分佈之動態平衡。

**全域全複本同級者**

【0232】 現在考慮當通用磁碟空間要求變得對於較小同級者而言過大時的情境，隨著區塊鏈之元網路部分縮放且隨採用增長會發生此情況。

【0233】 在此情況下，較小 LFCP 應基於風行度系統(存在用於藉由請求量及本質評級內容之現有技術)來使用其磁碟空間容量儲存元網路節點交易。此意謂 LFCP 現在修整其 CPU(用於請求處置能力)及其儲存分配(用於內容伺服能力)兩者，以適應其在內容量及本質兩方面上的本端地理要求。

【0234】 為了解決 LFCP 現在不能儲存所有元網路交易內容之實情，可利用全域全複本同級者(GFCP)之概念。GFCP 為具有以下性質之全複本同級者：

1.GFCP 增長其磁碟空間容量以便始終維持區塊鏈之全複本。

2. GFCP 具有相當大的 CPU 資源，使得其相比 LFCP 可處置明顯較多請求。若許多 LFCP 受損，全域全複本同級者應能夠處置需求之突然增大。

【0235】 GFCP 存在兩個主要功能。首先，在來自 LFCP 之請求溢出時，充當元網路內容之使用者請求的故障保護。其次，GFCP 充當存檔同級者以儲存歷史上挖掘之所有元網路內容，此確保即使許多 LFCP 自其本端儲存佈建省略一些內容，仍可存取任何元網路節點內容。

**全域資料庫**

【0236】 GFCP 之概念係強大的且說明元網路之總架構如何提供現有問題之解決方案；創建涵蓋所有之全域資料庫。

【0237】 在此之前，尚不可能安全地構造通用且可全域存取之資料庫，因為需要由中心機構來維持資料庫。此中心機構會給系統帶來故障點及信任點。至關重要地，若依賴於一個組織來儲存及維持所有網際網路資料，則吾人需要相信該組織正確且合法地如此操作，而不會損毀資訊之實情。

【0238】 在元網路基礎設施情況下，有效地自全域資料中心之概念移除了信任及中心性之此等兩問題。現在，可創建此 GFCP，因為僅依賴於其來提供儲存所需之磁碟空間而不驗證及認證待儲存之資訊。

【0239】 在元網路情況下，驗證儲存內容之過程由挖掘者進行且因此通用全域資料庫可係可信的，因為其無法損毀區塊鏈資訊。GFCP 無需係可信的且僅需要提供儲存。

【0240】 所有 GFCP 可儲存可始終針對區塊鏈自身驗證及證明之相同資訊的實情意指可跨越許多此等 GFCP 複製資訊。

【0241】 此意謂吾人藉由使許多全域資料庫並行存在且可證明地儲存相同資訊亦解決具有單個故障點之問題。

【0242】 圖 17 示出具有兩個 LFCP 及一個 GFCP 之系統，且說明每一同級者可如何在綜合平衡個別同級者之網路中支援另一者。

【0243】 可實施於上文所描述之瀏覽器錢包應用程式之實施例中的本發明態樣提供優於先前技術之眾多區別性特徵及優勢，包括但不限於：

1. **確定性密鑰 -** 在應用程式之同一錢包組件中執行用於密碼貨幣及元網路位址兩者之階層式確定性密鑰管理。此允許藉由減小其儲存要求且實現密鑰恢復之多個功能組織密鑰。

2. **支付機制 -** 應用程式允許消費者直接向商家支付，而無需指向將習知地認證並提供信任之另一應用程式或第三方支付服務。此允許透過同一區塊鏈平台進行數位內容之購買及交付。應用程式繼承比特幣支付之優勢，包括低價值交換或涉及多方之較複雜交易。

3. **旁路網路伺服器 -** 應用程式促進旁路將習知地處理大量訊務、請求及路由之傳統網路伺服器。此係因為應用程式僅需要自單個 LFCP 請求內容，此保證無需將請求轉遞至其他 LFCP 來伺服使用者。此減小總訊務量以及每一請求之完成

時間。

4. 定時存取 - 應用程式藉由與區塊鏈同步並基於其當前狀態使用區塊鏈來執行存取許可而促進對內容之定時存取。此移除對隨時間推移監視使用者特權之第三方服務的需求，同時保護原始所有者之權利。

**使用案例 - 去中心化 app 商店(Swapp 商店)**

【0244】 此處呈現的元網路架構之第一使用案例(僅出於例示性目的)為應用程式(app)之去中心化支付及分佈。

【0245】 考慮如下情境：app 開發者 Alice 與消費者 Bob 希望彼此交易。此交易將呈原子調換之形式，其中以金錢交換授予 Bob 對應用程式資料之存取的秘密密鑰。經加密應用程式資料已作為元網路節點交易之部分公開。

【0246】 原子級上調換之應用程式被稱為 *Swapp*。第三方平台(Swapp 商店)可用於對存在於元網路上之應用程式進行編錄並通告，但存取密鑰之支付及至諸如 Bob 之使用者的存取密鑰傳送並不需要涉及任何第三方且可直接在商家與消費者之間進行。

【0247】 以下章節詳述可用於購買及出售 Swapp 之過程，該過程自 Alice 創建 app 至 Bob 部署該 app。貫穿該過程，Alice 及 Bob 將使用其各別瀏覽器錢包與元網路交互。

發佈

【0248】 1.Alice 撰寫應用程式。構成此應用程式之資料為由<App>表示之內容。她亦使用秘密密鑰 $S_k$ 進行加密<e(App)>。

【0249】 2.Alice 創建節點交易 $ID_{AliceApp}$ 以設置其第一元網路域(樹)。其生成待用作節點位址之 1AliceAppHtKNngkdXEeobR76b53LETtpy ($P_{AliceApp}$)。

【0250】 3.Alice 接著創建第一節點之子代以形成對應於其應用程式之元網路庫的樹。圖 18 中示出 Alice 之樹域。

【0251】 此樹上之一個葉節點為對應於具有索引$ID_{App}$之其應用程式<App>的節點。在此節點中，Alice將經加密應用程式資料<e(App)>插入至節點之輸入指令碼(scriptSig)中。使用秘密密鑰$s_k$來使用Koblitz方法加密app資料。

【0252】 下文示出此節點交易。

| $TxID_{App}$ | |
|---|---|
| 輸入 | 輸出 |
| < $Sig\ P_{Puzzle}$> <$P_{Puzzle}$> <e(App)> OP_DROP | OP_RETURN <元網路旗標> <$P_{App}$> <$TxID_{Puzzle}$> |

【0253】 4.Alice 公開廣播$ID_{AliceApp}$、$P_{AliceApp}$及域名'AliceApp'。此可透過社交媒體、網際網路網站或藉由使用第三方元網路網站進行。

購買

【0254】 1.Bob 想要下載益智遊戲，並在其瀏覽器錢包上檢視的元網路網站(Swapp 商店)上看到列出之 Alice app。

【0255】 2.Bob 接著使用來自網站之資訊與 Alice 通訊，並設置原子調換。調換經設計成使得 Bob 將以比特幣向 Alice 支付商定的價格，且 Alice 將揭露秘密密鑰$s_k$或此等事件均不發生。

【0256】 3.原子調換完成且 Bob 之瀏覽器錢包將秘密密鑰$s_k$儲存在其存取密鑰/符記錢包中。

部署

【0257】 Bob 現在具有將允許其解密 Alice 先前公開之應用程式資料的密鑰$s_k$。為了下載 app 並進行部署，Bob 進行以下操作。

1.Bob使用元網路搜尋引擎(SE)找到與經加密app資料<e(App)>相關聯之MURL。其在瀏覽器錢包中將關鍵詞'AliceApp'及'App'用作至搜尋列之輸入。第三方SE解析查詢並傳回以下MURL：

mnp://aliceapp/games/puzzle/app

此定位符對應於其輸入指令碼中包括經加密 app 資料之唯一元網路節點$ID_{App}$。

2.Bob之瀏覽器錢包接收此MURL並發送請求至最接近適當LFCP。此同級者藉由所請求資料<e(App)>伺服Bob。

3.瀏覽器錢包根據$ID_{App}$之屬性處理資料。此包括使用秘密密鑰$s_k$來解密應用程式資料及處理<App>。

4.Bob將應用程式<App>自其瀏覽器下載至其電腦。Bob現在可在本端部署應用程式而不必重新購買存取。

【0258】 圖 19 說明上文例示性使用案例中概述之整個過程。流程圖示出兩個動作分支：Alice 分支(左手側上開始)及 Bob 分支(右手側上開始)。對應於 Alice 之分支示出初始發佈階段且 Bob 分支示出透過原子調換設置購買之階段。

【0259】 在 Bob 分支上，其將以下交易$TxID_{Bob}$廣播為原子調換設置階段：

| $TxID_{Bob}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ BCH | < $Sig\ P_B$> <$P_B$> | $x$ BCH | [私鑰謎題$P_k, r_0$] [CheckSig $P_A$] |

【0260】 在此交易中，藉由需要待向 Bob 揭露以便 Alice 使用之秘密解密密鑰$s_k$的私鑰謎題鎖定輸出。

此圖中的 Alice 及 Bob 分支在 Alice 成功地完成原子調換交易時彙聚。此係在 Alice 廣播交易$TxID_{Alice}$時實現：

| $TxID_{Alice}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$個比特幣 | < $Sig\ P_A$> <$P_A$> < $Sig\ P_1, r_0$> <$P_1$> | $x$個比特幣 | [CheckSig $P_A$] |

【0261】 一旦此交易經廣播，Alice 及 Bob 之動作分支再次發散。Alice 接

收$x$個比特幣之支付，而 Bob 接收秘密解密密鑰$s_k$且能夠自元網路提取及解密 Alice 之應用程式。

【0262】 現在轉向圖 20，提供計算裝置 2600 之例示性簡化方塊圖，該計算裝置可用於實踐本發明之至少一個實施例。在各種實施例中，計算裝置 2600 可用以實施上文所說明及描述之系統中之任一者。例如，計算裝置 2600 可經組配以用作資料伺服器、網路伺服器、攜帶型計算裝置、個人電腦或任何電子計算裝置。如圖 20 中所示，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 可包括如所示之動態隨機存取記憶體(DRAM) 2618 及唯讀記憶體(ROM) 2620。儲存子系統 2606 及快取記憶體 2602 且可用於儲存資訊，諸如與如本發明中所描述之交易及區塊相關聯的細節。處理器 2602 可用以提供如本發明中所描述之任何實施例的步驟或功能性。

【0263】 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。

【0264】 匯流排子系統 2604 可提供用於使計算裝置 2600 之各種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 經示意性地示出為單個匯流排，但匯流排子系統之替代實施例可利用多個匯流排。

【0265】 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。例如，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端位置時將資料傳輸至裝置及自裝置接收資料。

【0266】 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；併入至顯示器中之觸控螢幕；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。大體而言，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制。

【0267】 一或多個使用者介面輸出裝置 2614 可包括顯示子系統、印表機或諸如音訊輸出裝置之非視覺顯示器等。顯示子系統可係陰極射線管(CRT)、諸如液晶顯示器(LCD)之平板裝置、發光二極體(LED)顯示器，或投影裝置或其他顯示裝置。大體而言，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。例如，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係適當的時促進此交互，該等應用程式執行所描述之過程及其中之變型。

【0268】 儲存子系統 2606 可提供用於儲存可提供本發明之至少一個實施例之功能性的基本規劃及資料構造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發明之一或多個實施例的功能性，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發明所使用之資料的儲存庫。例如，主記憶體 2608 及快取記憶體 2602 可提供用於程式及資料之依電性儲存器。持久性儲存器 2610 可提供用於程式及資料之持久性(非依電性)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、具有相關聯可移除媒體之一或多個軟碟驅動機、具有相關聯可移除媒體之一或多個光學驅動機(例如，CD-ROM 或 DVD 或藍光光碟)驅動機及其他相似儲存媒體。此程式及資料可包括用於進行如本發明中所描述的一或多個實施例之步驟的程式，以及與如本發明中所描述之交易及區塊相關聯的資料。

【0269】 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 可包括可通過一或多個埠(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通過將裝置連接至計算裝置 2600 之埠傳輸的電氣信號以供處理。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 20 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 20 中所描繪之系統更多或更少組件的許多其他組配係可能的。

【0270】 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背離本發明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)」等並不排除除任何技術方案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」意謂「包括或由……組成」。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。本發明可借助於包含若干獨特元件之硬體且借助於經合適規劃之電腦予以實施。在枚舉若干構件之裝置技術方案中，此等構件中之若干者可由硬體之同一物件體現。在相互不同之附屬技術方案中敍述某些措施之純粹實情並不指示無法有利地使用此等措施之組合。

## 【符號說明】

【0271】

2600:計算裝置

2602:快取記憶體/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(DRAM)

2620:唯讀記憶體(ROM)

# 【發明申請專利範圍】

【請求項1】 一種電腦實施系統，其經配置以使得使用者能夠搜尋、存取、檢視、寫入及/或提取在至少一個區塊鏈交易(Tx)中提供的資料之一部分，其中：

該系統經配置以基於包含與該交易(Tx)相關聯之一交易 ID 及一公鑰的一交易索引($TX_{index}$)來識別該至少一個交易(Tx)。

【請求項2】 如請求項 1 之系統，其中該系統包含一搜尋設施，其：

提供在區塊鏈搜尋系統內；或

配置成與該區塊鏈搜尋系統介接及/或通訊。

【請求項3】 如請求項 1 或 2 之系統，且該系統進一步包含：

至少一個密碼貨幣錢包。

【請求項4】 如請求項 3 之系統，其中：

該至少一個密碼貨幣錢包配置成生成、儲存及/或處理階層式確定性密鑰。

【請求項5】 如請求項 3 之系統，其中該至少一個密碼貨幣錢包配置成將至少一個密碼密鑰及/或至少一個符記儲存在一可信執行環境中。

【請求項6】 如請求項 1 或 2 之系統，且該系統進一步包含：

一解壓縮組件，其配置成若資料之該部分係被壓縮，則對其進行解壓縮；

一重組組件；

及/或

一解密組件，其配置成若資料之該部分係被加密，則對其進行解密。

【請求項7】 如請求項 1 或 2 之系統，且該系統進一步包含：

至少一個呈現組件，其配置成以一可聽及/或視覺形式向一使用者呈現資料之該部分。

【請求項8】 如請求項 1 或 2 之系統，且該系統進一步包含：

用於輸入或生成用以識別區塊鏈上之該至少一個交易(Tx)的一搜尋路徑的

構件，該搜尋路徑包含：

i)該交易索引(TX$_{index}$)；以及

ii)與該交易(Tx)相關聯之至少一個屬性。

【請求項9】 如請求項 8 之系統，其中：

該等屬性中的至少一者為與該交易相關聯之一助憶符；及/或

該至少一個屬性為空值。

【請求項10】如請求項 1 或 2 之系統，且該系統經進一步配置成：

與一密碼貨幣錢包或其他資源通訊以促進處理、儲存及/或生成密碼密鑰、區塊鏈交易及/或數位簽章。

【請求項11】如請求項 1 或 2 之系統，且該系統經進一步配置成：

儲存該交易索引(TX$_{index}$)，較佳地其中該系統配置成儲存用於多於一個交易之各別交易索引。

【請求項12】如請求項 1 或 2 之系統，且該系統經進一步配置成：

在存取資料之該部分之前，將對密碼貨幣的一部分之控制傳送至一目的地。

【請求項13】如請求項 1 或 2 之系統，且該系統經進一步配置成：

向該區塊鏈上之一同級者發送對資料之該部分的一請求；及/或

自該區塊鏈上之一同級者接收資料之該部分。

【請求項14】如請求項 1 或 2 之系統，其中該系統進一步配置成：

使用一時間鎖定機制以控制對資料之該部分的存取。

# 【發明圖式】

| *TxID* | |
|---|---|
| 輸入 | 輸出 |
| < *Sig P*> <*P*> | OP_RETURN <元網路旗標> <屬性1> <屬性2> |
| | <內容1> OP_DROP <*H*(*P'*)> [CheckSig *P'*] |

【圖1】

| *TxID* | |
|---|---|
| 輸入 | 輸出 |
| < *sig P*> <*P*> <內容1><br>OP_DROP | OP_RETURN <元網路旗標> <屬性1> <屬性2> |

【圖2】

| $TxID_1$ | |
|---|---|
| 輸入 | 輸出 |
| $<Sig\ P>\ <P>$ | OP_RETURN <元網路旗標> <屬性1,1 = 內容名稱><br><屬性1,2 = 重組方案><br><屬性1,3 = 內容組塊索引1> |
| | <內容組塊1> OP_DROP <$H(P')$> [CheckSig $P'$] |

| $TxID_2$ | |
|---|---|
| 輸入 | 輸出 |
| $<Sig\ P>\ <P>$ | OP_RETURN <元網路旗標> <屬性2,1 = 內容名稱><br><屬性2,2 =重組方案><br><屬性2,3 =內容組塊索引2> |
| | <內容組塊2> OP_DROP <$H(P')$> [CheckSig $P'$] |

【圖3】

| $TxID_{Bob}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $<Sig\ P_B>\ <P_B>$ | $x$ | [私鑰謎題$P_1$,$r_0$] [CheckSig $P_A$] |

【圖4】

| $TxID_{Alice}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_A >\ <P_A>\ < Sig\ P_1, r_0 >\ <P_1>$ | $x$ | [CheckSig $P_A$] |

【圖5】

| 使用 | | Alice | Bob |
|---|---|---|---|
| 符記發佈 | 購買**10**個符記 | $I_{Alice} = k$ | $I_{Bob} = H^{10}(Y)$ |
| 符記兌換 | 兌換符記**1** | $X_1$ | $T_1 = H^9(Y)$ |
| | 兌換符記**2** | $X_2$ | $T_2 = H^8(Y)$ |
| | ... | ... | ... |
| | 兌換符記**10** | $X_{10}$ | $T_{10} = Y$ |

【圖6】

| $TxID_{Alice}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_A > < P_A >$ | $x$ | [散列謎題$H(I_{Alice})$] [散列謎題$H(I_{Bob})$] [CheckSig $P_B$] |

階段**1.1 (Alice**至**Bob)**：

【圖7】

| $TxID_{Bob}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $10 + x$ | $< Sig\ P_B > < P_B >$ | $10 + x$ | [散列謎題$H(I_{Alice})$] [散列謎題$H(I_{Bob})$] [CheckSig $P_A$] |

階段**1.2 (Bob**至**Alice)**：

【圖8】

階段**2.1 (Bob至Alice)**：

| $TxID_{Bob}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_B > < P_B >$ | $x$ | [散列謎題$H(X_1)$] [散列謎題$H(T_1)$] [CheckSig $P_A$] |

【圖9】

階段**2.2 (Alice至Bob)**：

| $TxID_{Alice}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_A > < P_A >$ | $x$ | [散列謎題$H(X_1)$] [散列謎題$H(T_1)$] [CheckSig $P_B$] |

【圖10】

階段**2.3 (Alice**至**Alice)**：

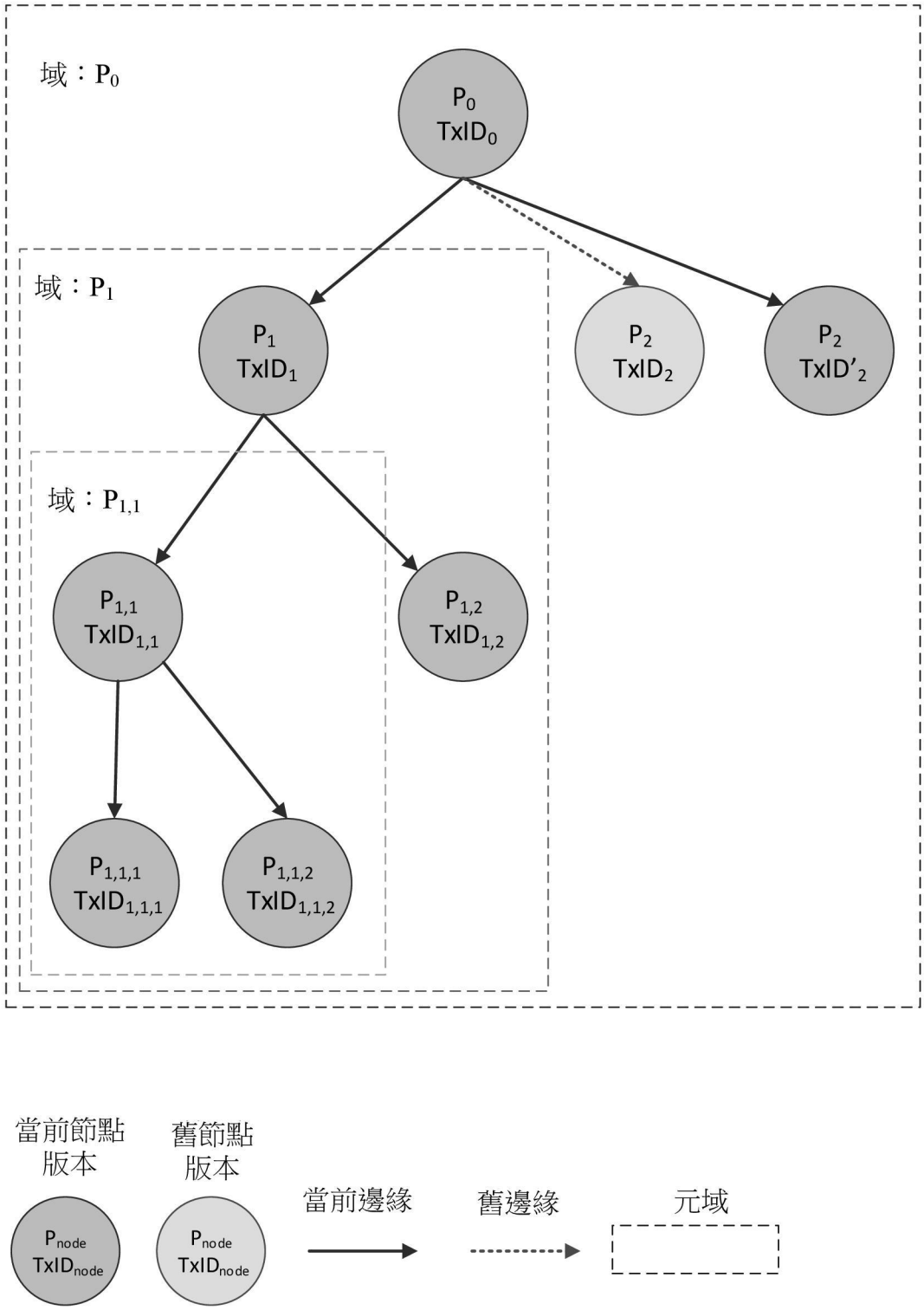
| $TxID_{Alice}$ | | | |
|---|---|---|---|
| 輸入 | | 輸出 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_A >$ $<P_A>$ $<T_1>$ $<X_1>$ | $x$ | [CheckSig $P_A$] |

【圖11】

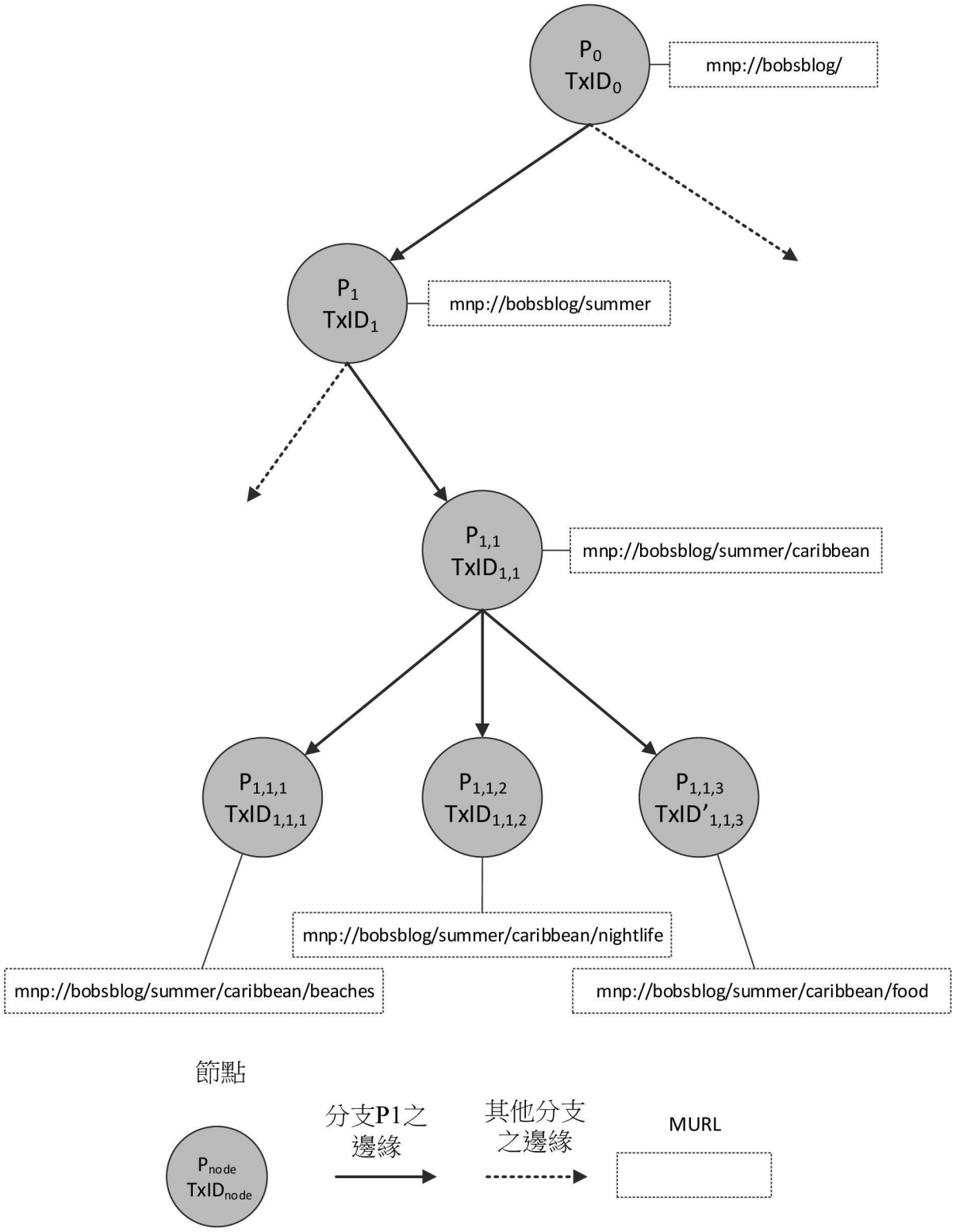
階段**2.4 (Bob**至**Bob)**：

| $TxID_{Bob}$ | | | |
|---|---|---|---|
| 輸入 | | 輸入 | |
| 值 | 指令碼 | 值 | 指令碼 |
| $x$ | $< Sig\ P_B >$ $<P_B>$ $<T_1>$ $<X_1>$ | $x$ | [CheckSig $P_B$] |

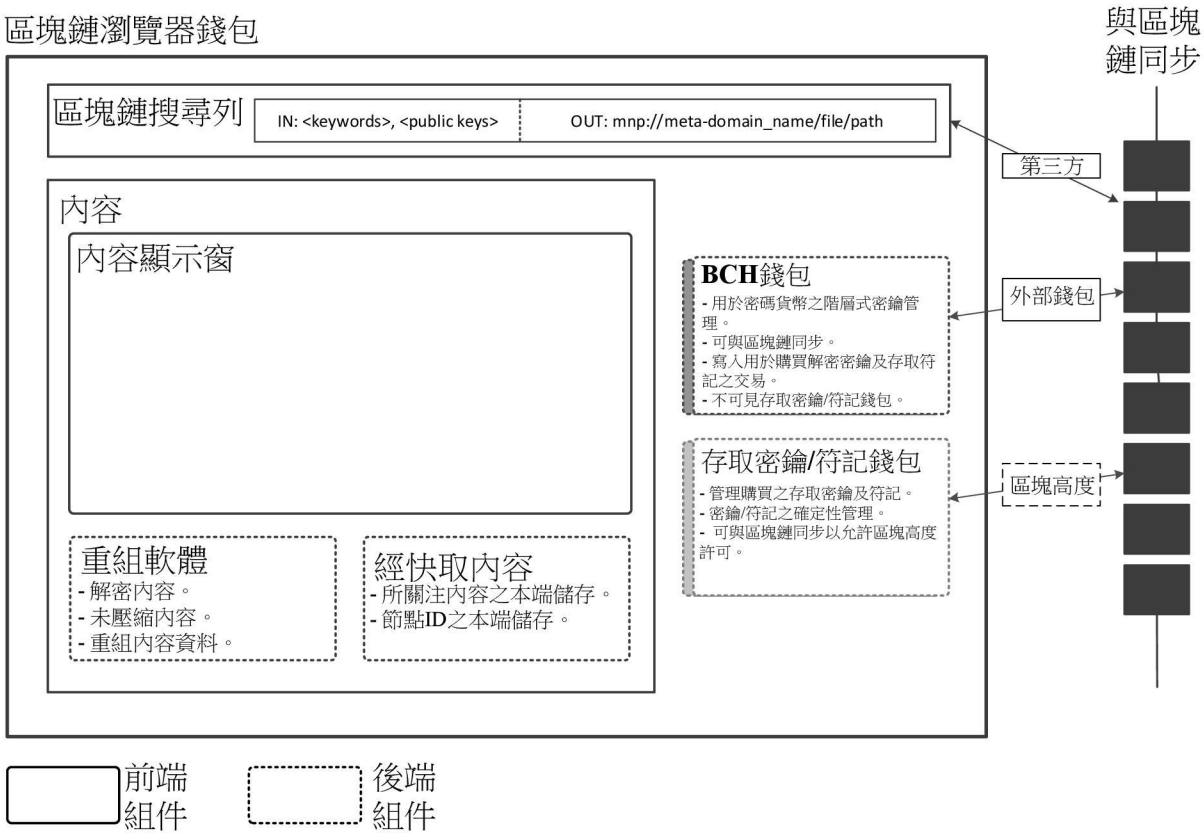
【圖12】


域：$P_0$
$P_0$ $TxID_0$
域：$P_1$
$P_1$ $TxID_1$
$P_2$ $TxID_2$
$P_2$ $TxID'_2$
域：$P_{1,1}$
$P_{1,1}$ $TxID_{1,1}$
$P_{1,2}$ $TxID_{1,2}$
$P_{1,1,1}$ $TxID_{1,1,1}$
$P_{1,1,2}$ $TxID_{1,1,2}$
當前節點版本
舊節點版本
當前邊緣
舊邊緣
元域
$P_{node}$ $TxID_{node}$
$P_{node}$ $TxID_{node}$

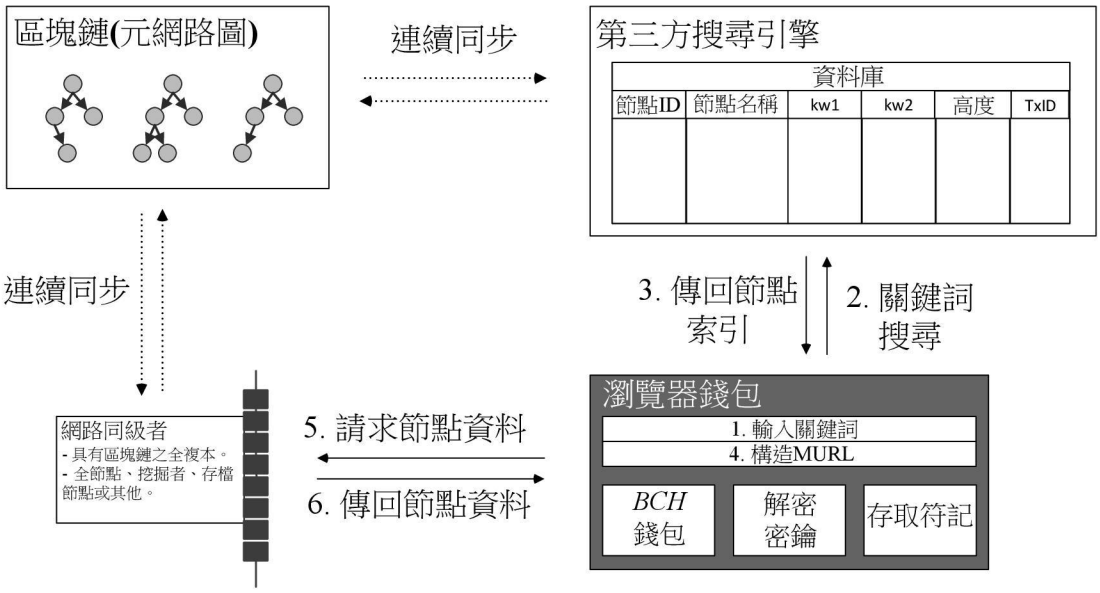
【圖13】


P0
TxID0
mnp://bobsblog/
P1
TxID1
mnp://bobsblog/summer
P1,1
TxID1,1
mnp://bobsblog/summer/caribbean
P1,1,1
TxID1,1,1
P1,1,2
TxID1,1,2
P1,1,3
TxID'1,1,3
mnp://bobsblog/summer/caribbean/nightlife
mnp://bobsblog/summer/caribbean/beaches
mnp://bobsblog/summer/caribbean/food
節點
Pnode
TxIDnode
分支P1之
邊緣
其他分支
之邊緣
MURL

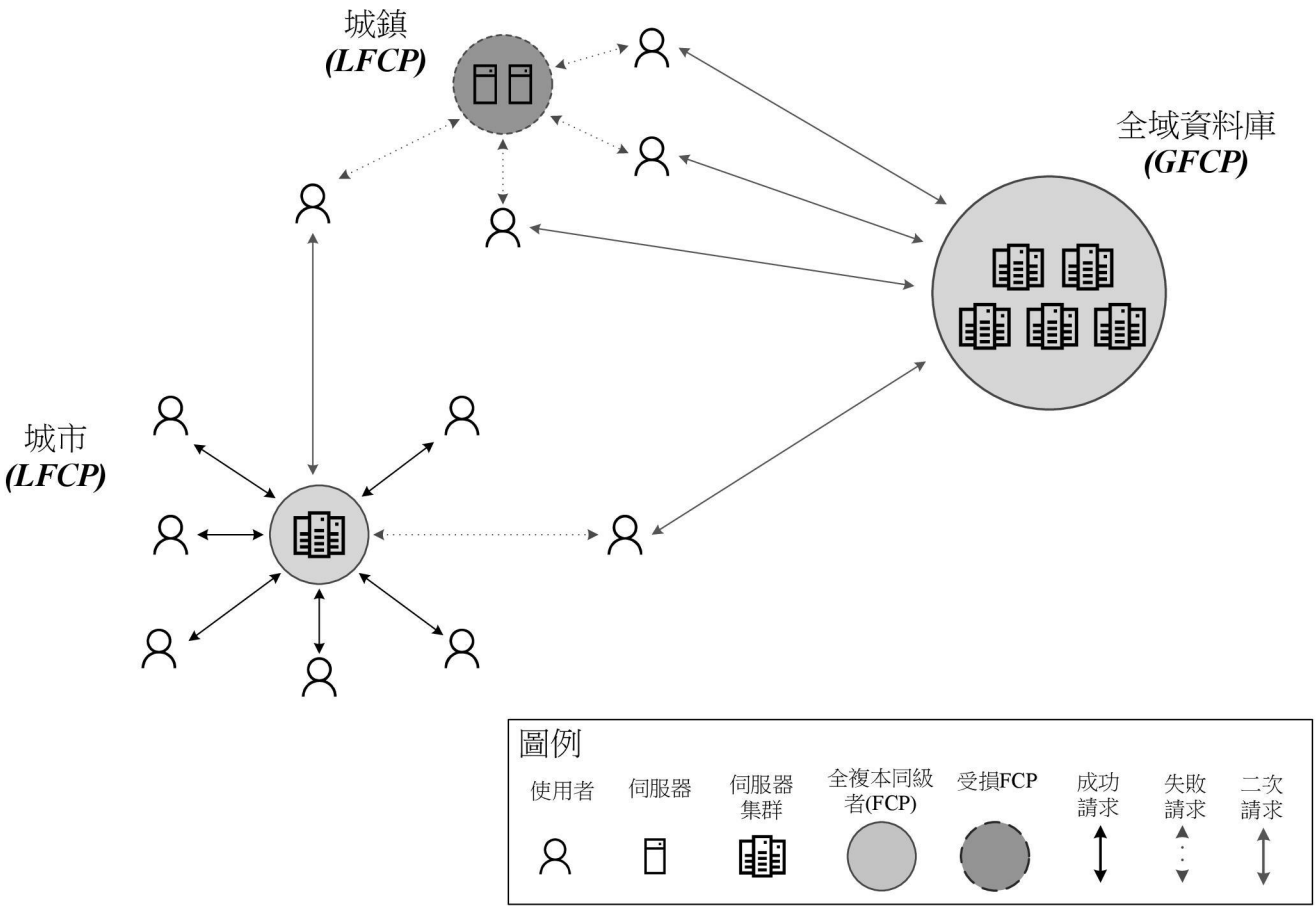
【圖14】


區塊鏈瀏覽器錢包
與區塊鏈同步
區塊鏈搜尋列
IN: <keywords>, <public keys>
OUT: mnp://meta-domain_name/file/path
第三方
內容
內容顯示窗
BCH錢包
- 用於密碼貨幣之階層式密鑰管理。
- 可與區塊鏈同步。
- 寫入用於購買解密密鑰及存取符記之交易。
- 不可見存取密鑰/符記錢包。
外部錢包
存取密鑰/符記錢包
- 管理購買之存取密鑰及符記。
- 密鑰/符記之確定性管理。
- 可與區塊鏈同步以允許區塊高度許可。
區塊高度
重組軟體
- 解密內容。
- 未壓縮內容。
- 重組內容資料。
經快取內容
- 所關注內容之本端儲存。
- 節點ID之本端儲存。
前端組件
後端組件

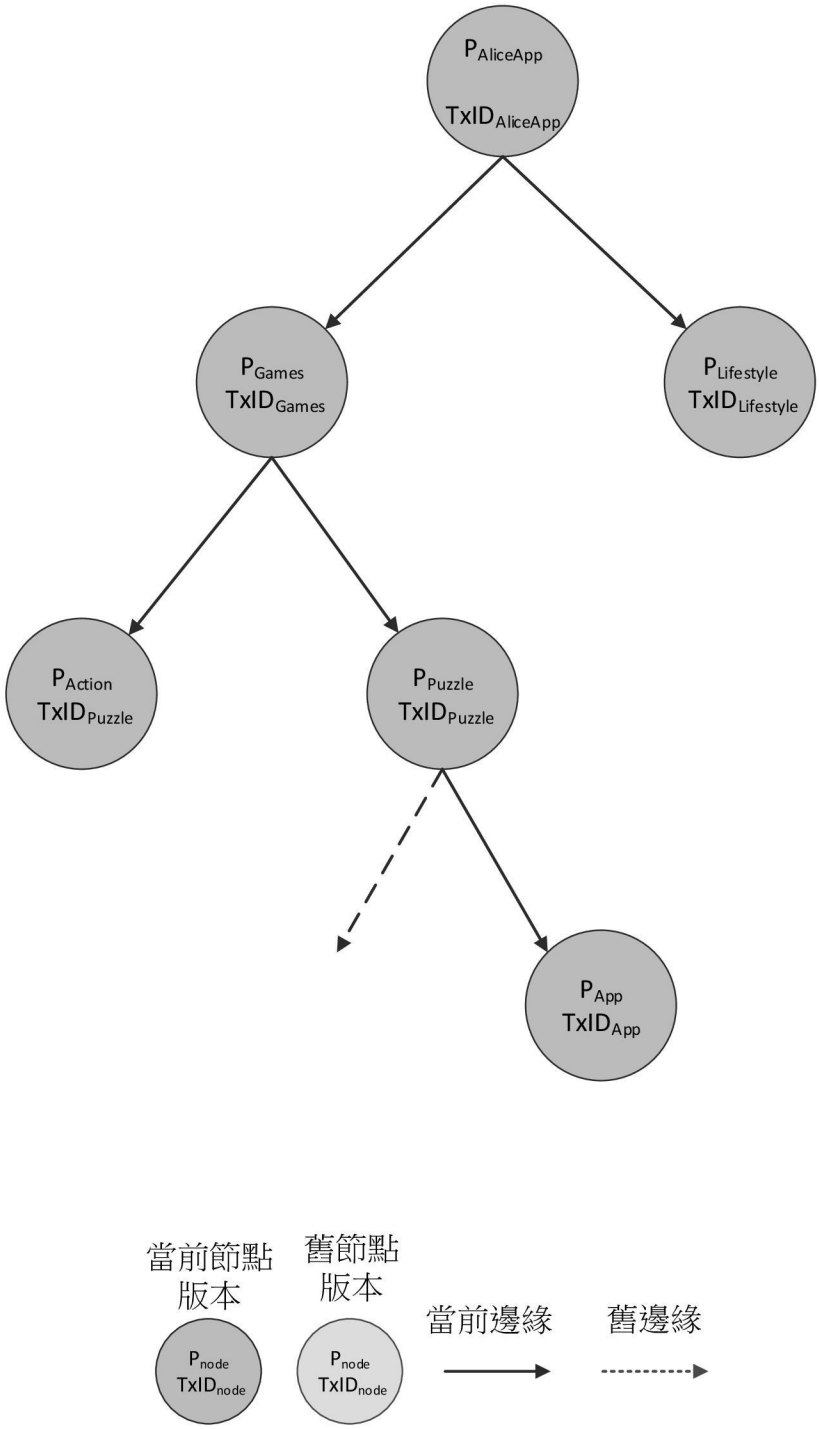
【圖15】


區塊鏈(元網路圖)
連續同步
第三方搜尋引擎
資料庫
節點ID
節點名稱
kw1
kw2
高度
TxID
連續同步
3. 傳回節點索引
2. 關鍵詞搜尋
網路同級者
- 具有區塊鏈之全複本。
- 全節點、挖掘者、存檔節點或其他。
5. 請求節點資料
6. 傳回節點資料
瀏覽器錢包
1. 輸入關鍵詞
4. 構造MURL
BCH 錢包
解密密鑰
存取符記

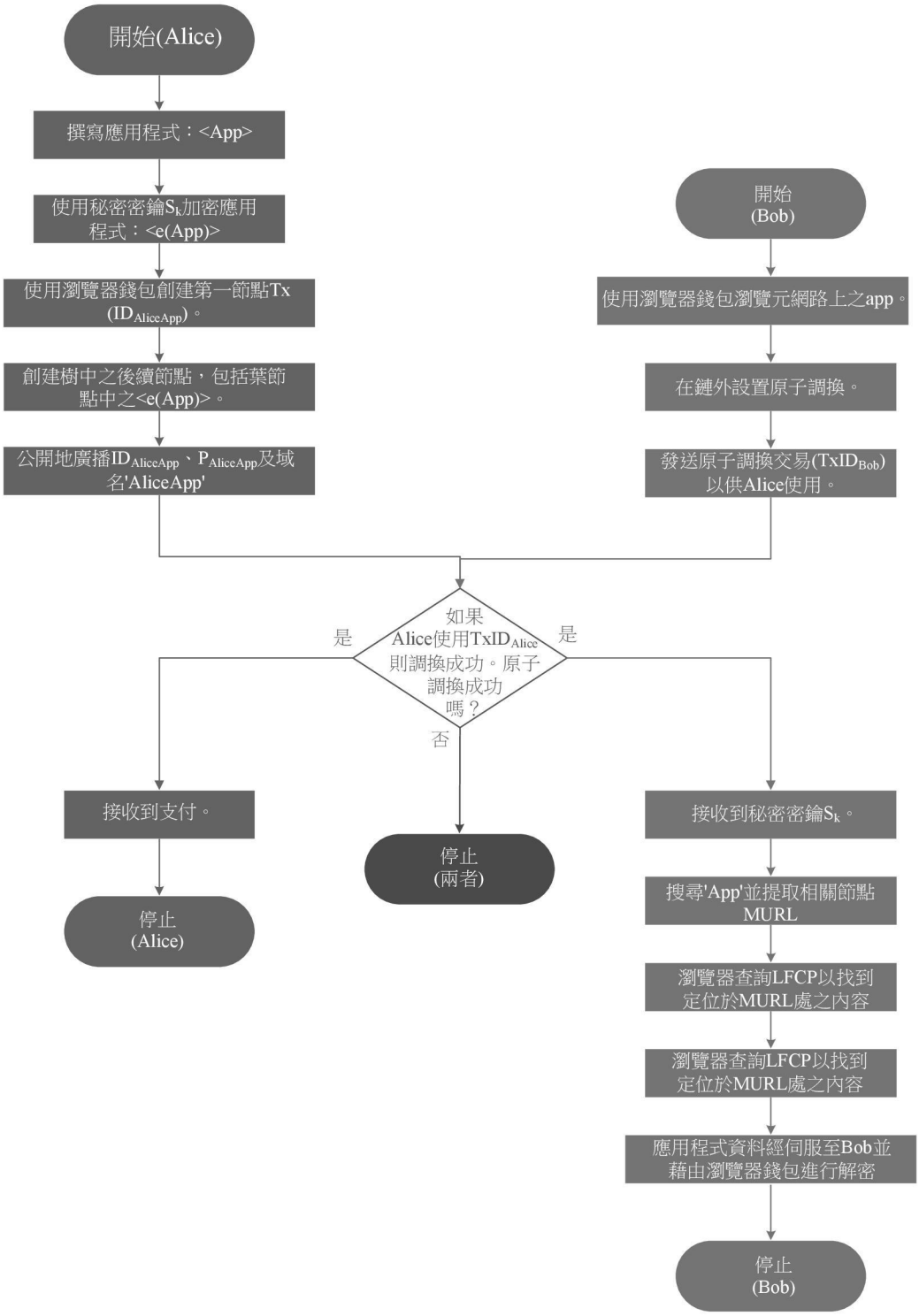
【圖16】


城鎮
(LFCP)
全域資料庫
(GFCP)
城市
(LFCP)
圖例
使用者
伺服器
伺服器集群
全複本同級者(FCP)
受損FCP
成功請求
失敗請求
二次請求

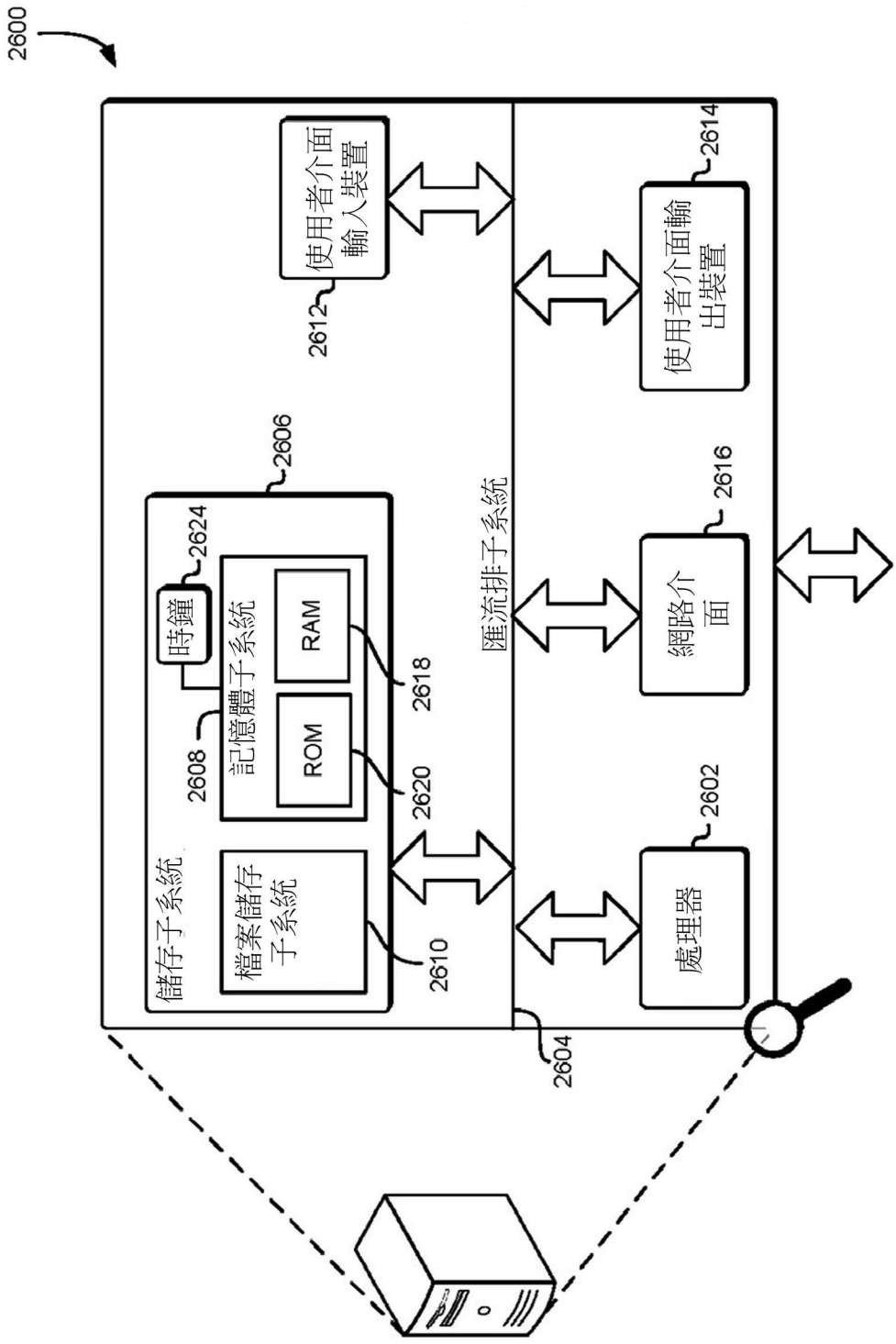
【圖17】


P AliceApp
TxID AliceApp
P Games
TxID Games
P Lifestyle
TxID Lifestyle
P Action
TxID Puzzle
P Puzzle
TxID Puzzle
P App
TxID App
當前節點版本
舊節點版本
當前邊緣
舊邊緣
P node
TxID node
P node
TxID node

【圖18】


開始(Alice)
撰寫應用程式：<App>
使用秘密密鑰$S_k$加密應用程式：<e(App)>
使用瀏覽器錢包創建第一節點Tx ($ID_{AliceApp}$)。
創建樹中之後續節點，包括葉節點中之<e(App)>。
公開地廣播$ID_{AliceApp}$、$P_{AliceApp}$及域名'AliceApp'
開始 (Bob)
使用瀏覽器錢包瀏覽元網路上之app。
在鏈外設置原子調換。
發送原子調換交易($TxID_{Bob}$)以供Alice使用。
如果Alice使用$TxID_{Alice}$則調換成功。原子調換成功嗎？
是
是
否
接收到支付。
停止 (Alice)
停止 (兩者)
接收到秘密密鑰$S_k$。
搜尋'App'並提取相關節點MURL
瀏覽器查詢LFCP以找到定位於MURL處之內容
瀏覽器查詢LFCP以找到定位於MURL處之內容
應用程式資料經伺服至Bob並藉由瀏覽器錢包進行解密
停止 (Bob)

【圖19】


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
ROM
2608
2618
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20】