(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0032777**

(51)[2019.01] **H04L 9/00**; G06Q 30/06; H04L 29/06; G06Q 20/38; G06Q 40/04

(13) **B**

(21) 1-2020-00609

(22) 04/02/2020

(30) 10201907333R 08/08/2019 SG

(45) 25/08/2022 413

(43) 25/02/2021 395

(73) Advanced New Technologies Co., Ltd. (KY)
Cayman Corporate Centre, 27 Hospital Road, George Town, Grand Cayman KY1-9008, Cayman Islands

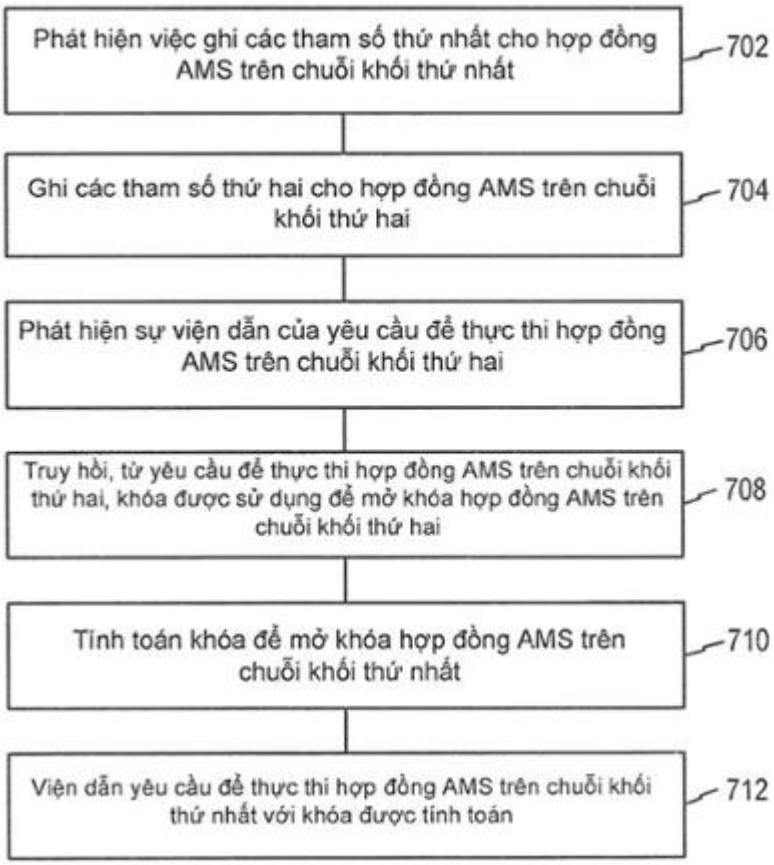
(72) Shengjiao CAO (CN); Yuan YUAN (SG); Hui FANG (SG); Weitao YANG (CN).

(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ THỰC THI CÁC HỢP ĐỒNG ĐA HOÁN ĐỔI ẨN DANH CHUỖI CHÉO

(57) Sáng chế đề cập đến phương pháp và thiết bị thực thi các hợp đồng đa hoán đổi ẩn danh chuỗi chéo. Phương pháp thực thi hợp đồng đa hoán đổi ẩn danh bao gồm các bước: phát hiện việc ghi của các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất; ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai; phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai; truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai; tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

700


Phát hiện việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất
702
Ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai
704
Phát hiện sự viện dẫn của yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai
706
Truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được sử dụng để mở khóa hợp đồng AMS trên chuỗi khối thứ hai
708
Tính toán khóa để mở khóa hợp đồng AMS trên chuỗi khối thứ nhất
710
Viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán
712

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến lĩnh vực công nghệ máy tính và cụ thể hơn là, phương pháp và thiết bị thực thi các hợp đồng đa hoán đổi ẩn danh chuỗi chéo.

Tình trạng kỹ thuật của sáng chế

Các hệ thống chuỗi khối, cũng được biết là các hệ thống sổ cái phân tán (distributed ledger system - DLS) hoặc các hệ thống đồng thuận, có thể cho phép sự tham gia của các bên để lưu trữ dữ liệu một cách an toàn và không thay đổi. Các hệ thống chuỗi khối có thể bao gồm các DLS bất kỳ, không tham chiếu trường hợp sử dụng cụ thể bất kỳ, và có thể được sử dụng cho các mạng chuỗi khối công cộng, cá nhân, và tập đoàn. Mạng chuỗi khối công cộng được mở cho tất cả các thực thể để sử dụng hệ thống và tham gia trong quy trình đồng thuận. Mạng chuỗi khối cá nhân được đề xuất cho thực thể cụ thể, điều khiển quyền đọc và ghi một cách tập trung. Mạng chuỗi khối tập đoàn dược đề xuất cho nhóm các thực thể chọn lọc, điều khiển quy trình đồng thuận, và bao gồm lớp điều khiển truy cập.

Hệ thống chuỗi khối được triển khai sử dụng mạng đồng cấp (peer-to-peer - P2P), trong đó các nút truyền thông trực tiếp với nhau, ví dụ, không cần máy chủ cố định, trung tâm. Mỗi nút trong mạng P2P có thể khởi tạo truyền thông với nút khác trong mạng P2P. Hệ thống chuỗi khối duy trì một hoặc nhiều chuỗi khối. Chuỗi khối là cấu trúc dữ liệu để lưu trữ dữ liệu, như các giao dịch, có thể ngăn ngừa giả mạo và thao túng dữ liệu bởi các bên độc hại.

Người dùng của hệ thống chuỗi khối có thể sử dụng hệ thống chuỗi khối để tiến hành các loại giao dịch khác nhau. Ví dụ, hai bên, Bên A và Bên B, có thể có tài sản sở hữu của chính họ được ghi trên chuỗi khối được duy trì trong hệ thống chuỗi khối. Các bên A và B có thể tham gia vào thỏa thuận trao đổi, hoặc hoán đổi, các tài sản của họ trên chuỗi khối. Các thỏa thuận này thường được tiến hành sử dụng các hợp đồng thông minh, là các giao thức máy tính được thực thi trong dạng của mã máy tính được hợp nhất trong chuỗi khối, để tạo thuận lợi, xác minh, hoặc thực thi đàm phán hoặc thực hiện các hợp đồng.

Trong khi việc sử dụng các hợp đồng thông minh có thể là đủ để tạo thuận lợi hoán đổi các tài sản được ghi trên cùng chuỗi khối, các trường hợp có thể phát sinh trong đó các bên A và B có thể muốn hoán đổi các tài sản được ghi trên các chuỗi khối khác nhau. Ví dụ, Bên A có thể muốn chuyển tài sản thứ nhất, ví dụ, bitcoin, được ghi trên chuỗi khối thứ nhất đến Bên B trong việc trao đổi để Bên B chuyển tài sản thứ hai, ví dụ, ete, được ghi trên chuỗi khối thứ hai đến Bên A. Các hoạt động này có thể được gọi là các hoán đổi chuỗi chéo.

Vì các hoán đổi chuỗi chéo yêu cầu nhiều lần chuyển được thực hiện trên nhiều chuỗi khối, cần lắp đặt giao thức hoán đổi nguyên tử để ngăn ngừa các bên khỏi từ chối chiếm giữ, hoặc rút khỏi, việc thực hiện các công đoạn chuyển một cách đơn phương. Ví dụ, nếu Bên A đã chuyển tài sản của bên này được ghi trên chuỗi khối thứ nhất đến Bên B, việc lắp đặt giao thức hoán đổi nguyên tử có thể trợ giúp ngăn ngừa Bên B khỏi việc từ chối chuyển tài sản của bên này được ghi trên chuỗi khối thứ hai đến Bên A.

Hợp đồng khóa thời gian được xây dựng sẵn, hoặc HTLC, là một kiểu hợp đồng thông minh để sử dụng khóa băm và khóa hẹn giờ để yêu cầu bên nhận của tài sản được chuyển, ví dụ, thanh toán, hoặc để báo nhận việc tiếp nhận của lần chuyển trước thời hạn bằng cách tạo ra bằng chứng mất mã của lần chuyển, hoặc làm mất khả năng yêu cầu tài sản được chuyển, nhờ đó trả lại tài sản này cho bên chuyển. Các HTLC có thể được sử dụng để cài đặt các giao thức hoán đổi nguyên tử. Tuy nhiên, các lắp đặt hiện thời của các giao thức hoán đổi nguyên tử dựa trên HTLC yêu cầu lượng thời gian thiết đặt thực tế.

Bản chất kỹ thuật của sáng chế

Theo một khía cạnh, phương pháp được thực hiện bằng máy tính để thực thi hợp đồng đa hoán đổi ẩn danh (anonymous multi-swap - AMS) giữa người dùng thứ nhất và người dùng thứ hai bao gồm: phát hiện ghi chép các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, các tham số thứ nhất bao gồm: giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, số lần tối đa hợp đồng AMS có thể được thực

thi lớn hơn một; ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, các tham số thứ hai bao gồm: giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất; phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai; truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai; tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

Theo khía cạnh khác, thiết bị thực thi hợp đồng AMS giữa người dùng thứ nhất và người dùng thứ hai bao gồm: một hoặc nhiều bộ xử lý; và một hoặc nhiều bộ nhớ đọc được bằng máy tính được ghép nối với một hoặc nhiều bộ xử lý và có các lệnh được lưu trữ trên đó được chạy bởi một hoặc nhiều bộ xử lý để: phát hiện việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, các tham số thứ nhất bao gồm: giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi; ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, các tham số thứ hai bao gồm: giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất; phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai; truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai; tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

Theo khía cạnh khác nữa, vật ghi đọc được bằng máy tính không tạm thời lưu trữ các lệnh trong đó, khi được thực thi bởi bộ xử lý của thiết bị, khiến thiết bị thực hiện phương pháp để thực thi hợp đồng AMS giữa người dùng thứ nhất và người dùng thứ hai. Phương pháp này bao gồm các bước: phát hiện ghi chép các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, các tham số thứ nhất bao gồm: giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi; ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, các tham số thứ hai bao gồm: giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất; phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai; truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai; tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

Mô tả vắn tắt các hình vẽ

Các hình vẽ kèm theo, được đưa vào trong bản mô tả và cấu thành một phần của bản mô tả này, minh họa các phương án. Trong mô tả sau đây, tham chiếu đến các hình vẽ, các số tham chiếu giống nhau trong các hình vẽ khác nhau biểu diễn các phần tử giống hoặc tương tự trừ khi được thể hiện khác.

Fig.1 là sơ đồ giản lược của hệ thống chuỗi khối, theo một phương án.

Fig.2 là sơ đồ giản lược của thiết bị điện toán để lắp đặt nút trong hệ thống chuỗi khối, theo một phương án.

Fig.3 là lưu đồ của phương pháp để thiết lập hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Fig.4 là lưu đồ của phương pháp để thực thi hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Fig.5 là lưu đồ của phương pháp để thực thi hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Fig.6 là sơ đồ khối của thiết bị thực thi hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Fig.7 là lưu đồ của phương pháp để thực thi hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Fig.8 là sơ đồ khối của thiết bị thực thi hợp đồng đa hoán đổi ẩn danh, theo một phương án.

Mô tả chi tiết sáng chế

Các phương án theo sáng chế đề xuất phương pháp và thiết bị để thiết lập và thực thi các hợp đồng đa hoán đổi ẩn danh (AMS) chuỗi chéo. Phương pháp và thiết bị hỗ trợ quy trình đàm phán ngoài chuỗi một lần trong đó hai bên, hoặc người dùng, có thể đàm phán các điều khoản của hợp đồng AMS để tạo thuận lợi các hoán đổi chuỗi chéo lên đến N lần. Theo một số phương án, phương pháp và thiết bị cũng hỗ trợ quy trình thiết lập trên chuỗi một lần để ghi các điều khoản được đàm phán dưới dạng các tham số của hợp đồng AMS. Theo một số phương án, quy trình thiết lập trên chuỗi có thể ghi các tham số trên các chuỗi khối được viện dẫn trong các hoán đổi chuỗi chéo. Theo phương thức này, các chuỗi khối có thể dựa vào hợp đồng AMS để đảm bảo rằng các hoán đổi có thể diễn ra nếu tất cả người dùng tuân theo hợp đồng AMS và tất cả các hoán đổi là thuộc hoán đổi nguyên tử và hợp lý. Nếu một người dùng không tuân theo hợp đồng AMS, người dùng đó sẽ tự động mất khả năng để yêu cầu tài sản được chuyển, do đó mà trả lại tài sản cho người dùng chuyển. Hơn nữa, theo một số phương án, phương pháp và thiết bị có thể lắp đặt giao thức để giấu mỗi tương quan giữa các tham số được ghi trên các chuỗi khối để ngăn ngừa bên thứ ba khỏi liên kết các tham số được ghi trên các chuỗi khối với nhau. Theo phương thức này, tính bí mật của các bên đối với hợp đồng AMS có thể được bảo vệ.

Các phương án được bộc lộ trong bản mô tả này có một hoặc nhiều hiệu quả kỹ thuật. Theo một số phương án, phương pháp và thiết bị hỗ trợ quy trình đàm phán ngoài chuỗi một lần để thiết lập hợp đồng AMS sao cho có thể được thực thi lên đến N lần. Điều này cho phép người dùng giảm bớt nhu cầu để đàm phán lại hợp đồng đó lên đến N-1 lần. Theo một số phương án, phương pháp và thiết bị cũng hỗ trợ quy trình thiết lập trên chuỗi một lần để ghi các điều khoản được đàm phán dưới dạng các tham số trên các chuỗi khối. Điều này cho phép các chuỗi khối ghi các tham số chính xác và hiệu quả, và giảm bớt nhu cầu lặp lại quy trình thiết lập trên chuỗi lên đến N-1 lần. Theo một số phương án, phương pháp và thiết bị sử dụng các chuỗi hàm một chiều (ví dụ, các chuỗi băm) để triển khai các khóa hàm cho hợp đồng AMS. Điều này cho phép mỗi người dùng chỉ duy trì một khóa, có thể làm việc kết hợp với chuỗi hàm một chiều để mở khóa hợp đồng AMS lên đến N lần. Theo một số phương án, phương pháp và thiết bị lắp đặt giao thức để che dấu mối quan hệ giữa các khóa hàm được sử dụng trên các chuỗi khối khác nhau. Điều này cho phép các chuỗi khối sử dụng các khóa hàm cho các hợp đồng AMS mà không làm lộ các mối quan hệ giữa người dùng thiết lập các hợp đồng AMS, nhờ đó bảo vệ tính bí mật của người dùng. Hơn nữa, theo một số phương án, phương pháp và thiết bị có thể tự động làm mất hiệu lực hợp đồng AMS sau N lần thực thi, hoặc sau khi hợp đồng AMS hết hiệu lực. Điều này cho phép các chuỗi khối ngăn ngừa người dùng khỏi thực thi các hoán đổi trái phép. Theo một số phương án, phương pháp và thiết bị còn hỗ trợ sự sắp thứ tự lên đến N lần hoán đổi trong hợp đồng AMS. Điều này cho phép người dùng xác định không những số lần hoán đổi mà họ đồng ý thực hiện, mà còn xếp thứ tự các hoán đổi này.

Chuỗi khối là cấu trúc dữ liệu để lưu trữ dữ liệu, ví dụ, các giao dịch, theo cách mà có thể ngăn ngừa làm giả mạo và thao túng dữ liệu bởi các bên độc hại. Các giao dịch được lưu trữ theo phương thức này có thể là bất biến và sau đó được xác minh. Chuỗi khối bao gồm một hoặc nhiều khối. Mỗi khối được liên kết với khối trước ngay trước khi nó ở trong chuỗi khối này bằng cách đưa vào hàm băm mật mã của khối trước. Mỗi khối cũng có thể bao gồm nhãn thời gian, hàm băm mật mã của chính nó, và một hoặc nhiều giao dịch. Các giao dịch, thường đã được xác minh bởi các nút của hệ thống chuỗi khối, có thể được xây dựng sẵn và được

mã hóa trong cấu trúc dữ liệu, như cây Merkle. Trong cây Merkle, dữ liệu tại các nút lá của cây được băm, và tất cả các hàm băm trong mỗi nhánh của cây có thể được móc nối tại gốc của nhánh. Quy trình này tiếp tục lên đến cây cho tới gốc của toàn bộ cây, lưu trữ hàm băm có thể biểu diễn tất cả dữ liệu trong cây này. Hàm băm có ý nghĩa là một giao dịch được lưu trữ trong cây có thể được xác minh nhanh chóng bằng cách xác định liệu có phù hợp với cấu trúc của cây hay không.

Hệ thống chuỗi khối bao gồm mạng của các nút tính toán để quản lý, cập nhật, và duy trì một hoặc nhiều chuỗi khối. Mạng này có thể là mạng chuỗi khối công cộng, mạng chuỗi khối cá nhân, hoặc mạng chuỗi khối tập đoàn. Ví dụ, các thực thể khác nhau, như hàng trăm, hàng nghìn, hoặc thậm chí hàng triệu thực thể, có thể vận hành trong mạng chuỗi khối công cộng, và mỗi trong số các thực thể vận hành ít nhất một nút trong mạng chuỗi khối công cộng. Theo đó, mạng chuỗi khối công cộng có thể được xem xét mạng công cộng tương ứng với các thực thể tham gia. Đôi khi, phần lớn các thực thể (các nút) phải đánh dấu mỗi khối đối với khối là hợp lệ và được thêm vào chuỗi khối của mạng chuỗi khối. Các ví dụ về các mạng chuỗi khối công cộng bao gồm các mạng thanh toán đồng cấp cụ thể để tác dụng đòn bẩy lên sổ cái phân tán, được gọi là chuỗi khối.

Nói chung, mạng chuỗi khối công cộng có thể hỗ trợ các giao dịch công cộng. Giao dịch công cộng được chia sẻ với tất cả các nút trong mạng chuỗi khối công cộng, và được lưu trữ trong chuỗi khối toàn cầu. Chuỗi khối toàn cầu là chuỗi khối được nhân rộng trên tất cả các nút, và tất cả các nút ở trạng thái đồng thuận hoàn hảo tương thích với chuỗi khối toàn cầu. Để đạt được đồng thuận (ví dụ, thỏa thuận thêm một khối vào chuỗi khối), giao thức đồng thuận được thực hiện trong mạng chuỗi khối công cộng. Các ví dụ của các giao thức đồng thuận bao gồm bằng chứng công việc (proof-of-work - POW) (ví dụ, được thực hiện trong một số mạng tiền mã hóa), bằng chứng cổ phần (proof-of-stake - POS), và bằng chứng ủy quyền (proof-of-authority - POA).

Nói chung, mạng chuỗi khối cá nhân có thể được đề xuất cho thực tể cụ thể, điều khiển quyền đọc và ghi một cách tập trung. Thực thể này điều khiển các nút nào có thể tham gia vào mạng chuỗi khối. Do đó, các mạng chuỗi khối cá nhân

thường được gọi là các mạng được cấp phép để đặt các giới hạn cho ai được phép tham gia trong mạng, và đối với các cấp độ tham gia của họ (ví dụ, chỉ trong các giao dịch nhất định). Các kiểu cơ chế điều khiển truy cập khác nhau có thể được sử dụng (ví dụ, những người tham gia hiện có bình chọn việc thêm vào các thực thể mới, cơ quan điều hành có thể điều chỉnh quyền quản lý).

Nói chung, mạng chuỗi khối tập đoàn có thể là cá nhân trong số các thực thể tham gia. Trong mạng chuỗi khối tập đoàn, quy trình đồng thuận được điều khiển bởi tập hợp các nút được cấp phép, một hoặc nhiều nút được vận hành bởi thực thể tương ứng (ví dụ, tổ chức tài chính, công ty bảo hiểm). Ví dụ, sự liên kết của mười (10) thực thể (ví dụ, các tổ chức tài chính, công ty bảo hiểm) có thể vận hành mạng chuỗi khối tập đoàn, mỗi trong số các thực thể này vận hành ít nhất một nút trong mạng chuỗi khối tập đoàn. Theo đó, mạng chuỗi khối tập đoàn có thể được xem xét mạng cá nhân tương ứng với các thực thể tham gia. Theo một số ví dụ, mỗi thực thể (nút) phải đánh đấu mỗi khối để cho khối này hợp lệ, và được thêm vào chuỗi khối. Theo một số ví dụ, ít nhất tập con của các thực thể (các nút) (ví dụ, ít nhất 7 thực thể) phải đánh đấu mỗi khối để cho khối này là hợp lệ, và được thêm vào chuỗi khối.

Fig.1 minh họa sơ đồ giản lược của hệ thống chuỗi khối 100, theo một phương án. Tham chiếu đến Fig.1, hệ thống chuỗi khối 100 có thể bao gồm nhiều nút, ví dụ, các nút 102-110, được tạo cấu hình để vận hành trên chuỗi khối 120. Các nút 102-110 có thể tạo ra mạng 112, như mạng đồng cấp (P2P). Mỗi trong số các nút 102-110 có thể là thiết bị điện toán, như máy tính hoặc hệ thống máy tính, được tạo cấu hình để lưu trữ bản sao của chuỗi khối 120, hoặc có thể là phần mềm chạy trên thiết bị điện toán, như quy trình hoặc ứng dụng. Mỗi trong số các nút 102-110 có thể có mã nhận dạng duy nhất.

Chuỗi khối 120 có thể bao gồm danh sách bản ghi ngày càng nhiều ở dạng của các khối dữ liệu, như các khối B1-B5 trên Fig.1. Mỗi trong số các khối B1-B5 có thể bao gồm nhãn thời gian, hàm băm mật mã của khối trước, và dữ liệu của khối hiện có, có thể là các giao dịch như các giao dịch tiền tệ. Ví dụ, như được minh họa trên Fig.1, khối B5 có thể bao gồm nhãn thời gian, hàm băm mật mã của

khối B4, và dữ liệu giao dịch của khối B5. Ngoài ra, ví dụ, hoạt động băm có thể được thực hiện trên khối trước để tạo ra hàm băm mật mã của khối trước. Hoạt động băm có thể chuyển đổi các đầu vào có các độ dài khác nhau thành các đầu ra mật mã có độ dài cố định thông qua thuật toán băm, như SHA-256.

Các nút 102-110 có thể được tạo cấu hình để thực hiện hoạt động trên chuỗi khối 120. Ví dụ, khi nút, ví dụ, nút 102, muốn lưu trữ dữ liệu mới trên chuỗi khối 120, nút đó có thể tạo ra khối mới được thêm vào chuỗi khối 120 và quảng bá khối mới cho các nút khác, ví dụ, các nút 104-110, trong mạng 112. Dựa trên tính hợp pháp của khối mới, ví dụ, tính hợp lệ của chữ ký và các giao dịch, các nút khác có thể xác định để chấp nhận khối mới, sao cho nút 102 và các nút khác có thể thêm khối mới vào các bản sao tương ứng của chúng của chuỗi khối 120. Khi quy trình này lặp lại, ngày càng nhiều khối dữ liệu có thể được thêm vào chuỗi khối 120.

Fig.2 minh họa sơ đồ giản lược của thiết bị điện toán 200 để lắp đặt nút, ví dụ, nút 102 (Fig.1), trong hệ thống chuỗi khối, theo một phương án. Tham chiếu đến Fig.2, thiết bị điện toán 200 có thể bao gồm giao diện truyền thông 202, bộ xử lý 204, và bộ nhớ 206.

Giao diện truyền thông 202 có thể tạo thuận lợi cho truyền thông giữa thiết bị điện toán 200 và các thiết bị lắp đặt các nút khác, ví dụ, các nút 104-110 (Fig.1), trong mạng. Theo một số phương án, giao diện truyền thông 202 được tạo cấu hình để hỗ trợ một hoặc nhiều tiêu chuẩn truyền thông, như tiêu chuẩn hoặc giao thức internet, tiêu chuẩn mạng kỹ thuật số dịch vụ tích hợp (Integrated Services Digital Network - ISDN), v.v. Theo một số phương án, giao diện truyền thông 202 có thể bao gồm một hoặc nhiều trong số thẻ mạng cục bộ (Local Area Network - LAN), môđem cáp, môđem vệ tinh, bus dữ liệu, cáp, kênh truyền thông không dây, kênh truyền thông dựa vào vô tuyến, kênh truyền thông chia ô, thiết bị truyền thông dựa vào giao thức internet (Internet Protocol - IP), hoặc các thiết bị truyền thông khác cho truyền thông có dây và/hoặc không dây. Theo một số phương án, giao diện truyền thông 202 có thể dựa trên cơ sở hạ tầng đám mây công cộng, cơ sở hạ tầng đám mây cá nhân, cơ sở hạ tầng đám mây công cộng/cá nhân lai hóa.

Bộ xử lý 204 có thể bao gồm một hoặc nhiều đơn vị xử lý dành riêng, mạch

tích hợp chuyên dụng (application-specific integrated circuit- ASIC), mảng cổng lập trình được dạng trường (field-programmable gate array - FPGA), hoặc các kiểu bộ xử lý khác nhau hoặc các đơn vị xử lý. Bộ xử lý 204 được ghép nối với bộ nhớ 206 và được tạo cấu hình để thực thi các lệnh được lưu trữ trong bộ nhớ 206.

Bộ nhớ 206 có thể lưu trữ các lệnh và dữ liệu thực thi được bằng bộ xử lý, như bản sao của chuỗi khối 120 (Fig.1). Bộ nhớ 206 có thể bao gồm loại thiết bị bộ nhớ khả biến hoặc không khả biến bất kỳ, hoặc kết hợp của chúng, như bộ nhớ truy cập ngẫu nhiên dạng tĩnh (Static Random Access Memory - SRAM), bộ nhớ chỉ đọc lập trình được xóa được bằng điện (Electrically Erasable Programmable Read-only Memory - EEPROM), bộ nhớ chỉ đọc lập trình được xóa được (Erasable Programmable Read-only Memory - EEPROM), bộ nhớ chỉ đọc lập trình được (Programmable Read-only Memory – EEPROM), bộ nhớ chỉ đọc (Read-Only Memory - ROM), bộ nhớ từ tính, bộ nhớ chớp, đĩa từ tính hoặc quang học. Khi các lệnh trong bộ nhớ 206 được chạy bởi bộ xử lý 204, thiết bị điện toán 200 có thể thực hiện hoạt động trên chuỗi khối 120.

Fig.3 minh họa lưu đồ của phương pháp 300 để thiết lập hợp đồng đa hoán đổi nguyên tử chuỗi chéo (AMS) theo một phương án. Theo phương án này, hợp đồng AMS được thiết lập bởi phương pháp 300 có thể được thực thi lên đến N lần.

Tham chiếu đến Fig.3, người dùng thứ nhất và thứ hai, người dùng 1 và người dùng 2, có thể có các tài khoản lần lượt trên các chuỗi khối riêng biệt thứ nhất và thứ hai, chuỗi khối A và chuỗi khối B. Các chuỗi khối A và B có thể được lắp đặt trên một hoặc nhiều hệ thống chuỗi khối, ví dụ, hệ thống chuỗi khối 100 (Fig.1) và tương tự, như được mô tả ở trên. Các chuỗi khối A và B có thể được lắp đặt để hỗ trợ các kiểu người dùng khác nhau, hoặc các bên, bao gồm, ví dụ, các cá nhân, doanh nghiệp, ngân hàng, tổ chức tài chính, bệnh viện cũng như các kiểu công ty, tổ chức khác, và tương tự.

Đối với mục đích minh họa, giả sử rằng người dùng 1 muốn hoán đổi tài sản X được sở hữu bởi người dùng 1 và được ghi trên chuỗi khối A với tài sản Y được sở hữu bởi người dùng 2 và được ghi trên chuỗi khối B. Tài sản X có thể bao gồm một hoặc nhiều đơn vị của kiểu tài sản nhất định, ví dụ, bitcoin, hoặc chùm của các

kiểu tài sản khác nhau, được ghi trên chuỗi khối A. Tài sản Y có thể bao gồm một hoặc nhiều đơn vị của loại tài sản nhất định, ví dụ, ete, hoặc chùm của các kiểu tài sản khác nhau, được ghi trên chuỗi khối B. Cũng giả sử rằng người dùng 1 muốn hoán đổi với người dùng 2 với cùng các điều khoản trong một số lần, như lên đến N lần, trong tương lai. Người dùng 1 có thể yêu cầu sự giúp đỡ của người dùng 2 sử dụng kênh truyền thông và bắt đầu quy trình đàm phán với người dùng 2. Người dùng 1 và 2 có thể đàm phán các điều khoản hoán đổi cụ thể, bao gồm, ví dụ, tỷ giá hối đoái giữa tài sản X và tài sản Y, số lần hoán đổi tối đa (hoặc các lần hợp đồng AMS có thể được thực thi), N, người dùng 1 được cho phép để viện dẫn hoạt động hoán đổi, thời điểm hết hiệu lực T_1 khi thỏa thuận hết hiệu lực trên chuỗi khối A, và thời điểm hết hiệu lực T_2 khi thỏa thuận hết hiệu lực trên chuỗi khối B. Hợp đồng AMS có thể được thiết lập sau khi người dùng 1 và 2 đạt được thỏa thuận.

Tại bước 302, người dùng 1 có thể tạo ra một hoặc nhiều tham số cho hợp đồng AMS dựa trên thỏa thuận đạt được bởi người dùng 1 và 2. Theo một số phương án, các tham số có thể bao gồm các giá trị của số lượng tối đa N, thời điểm hết hiệu lực T_1 khi thỏa thuận hết hiệu lực trên chuỗi khối A, và thời điểm hết hiệu lực T_2 khi thỏa thuận này hết hiệu lực trên chuỗi khối B.

Theo một số phương án, các tham số có thể cũng bao gồm giá trị hàm một chiều thứ nhất H_N và giá trị hàm một chiều thứ hai Δ_N được tính toán bởi người dùng 1. H_N có thể được sử dụng để đóng vai trò làm khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối A và $H_N + \Delta_N$ có thể được sử dụng để đóng vai trò làm khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối B. Theo cách khác, H_N có thể được sử dụng để đóng vai trò làm khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối B và $H_N + \Delta_N$ có thể được sử dụng để đóng vai trò làm khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối A.

Khóa hàm có thể đề cập đến khóa, L, có thể được mở khóa chỉ khi phím hiệu chỉnh, Key, được đưa vào cho hàm $g()$, có thể sinh ra giá trị $g(Key)$ để tương thích giá trị của L. Nếu hàm $g()$ là hàm băm, thì khóa hàm có thể được gọi là khóa băm. Một trong số các lý do để sử dụng các khóa hàm khác trên các chuỗi khối A và B để ngăn ngừa bên thứ ba liên kết hai khóa hàm với nhau. Nếu bên thứ ba có

thể liên kết hai khóa hàm với nhau, ví dụ, bằng cách nhận dạng hai khóa hàm giống nhau, bên thứ ba có thể được cho phép để nhận dạng các lần chuyển được tiến hành đối với cả hai bên của hợp đồng AMS, có thể làm lộ thông tin là bí mật với người dùng 1 và 2. Ví dụ, giả sử rằng chuỗi khối A ghi nhận sở hữu bất động sản và chuỗi khối B ghi thông tin thanh toán. Nếu người dùng 1 và 2 tiến hành hoán đổi chuỗi chéo để chuyển bất động sản được ghi trên chuỗi khối A từ người dùng 1 đến người dùng 2 và chuyển thanh toán được ghi trên khối B từ người dùng 2 đến người dùng 1, bằng cách liên kết hai lần chuyển với nhau, bên thứ ba có thể có khả năng xác định giá cả mà người dùng 2 được thanh toán cho bất động sản. Theo ví dụ khác, nếu chuỗi khối A sử dụng nhận dạng công khai đối với người dùng của chuỗi này trong khi chuỗi khối B bắt buộc ẩn danh, liên kết hai lần chuyển với nhau có thể cho phép bên thứ ba xác định các nhận dạng của người dùng vận hành trên chuỗi khối B, nhờ đó thỏa hiệp tình trạng ẩn danh được bắt buộc bởi chuỗi khối B. Theo đó, bằng cách yêu cầu các chuỗi khối A và B để sử dụng các khóa hàm khác nhau, ví dụ, H_N trên chuỗi khối A và $H_N + \Delta_N$ trên chuỗi khối B, mỗi quan hệ giữa các lần chuyển được tiến hành đối với cả hai phía của hợp đồng AMS có thể được che dấu, nhờ đó bảo vệ tính bí mật của Người dùng 1 và 2.

Theo một số phương án, cả H_N và Δ_N có thể được tính toán sử dụng hàm một chiều $g()$. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng hàm g: $\mathcal{D} \rightarrow \mathcal{R}$ là một chiều nếu, đã cho phần tử ngẫu nhiên $x \in \mathcal{R}$, sẽ khó tính toán $y \in \mathcal{D}$ sao cho $g(y) = x$. Nói cách khác, sẽ khó để tính toán giá trị của biến độc lập của hàm một chiều từ giá trị của biến phụ thuộc của hàm một chiều, làm cho hàm thực sự không đổi chiều được và, do đó, hàm được goi là "một chiều". Các hàm băm mật mã, như SHA 256 và tương tự, là các ví dụ của các hàm một chiều.

Theo một số phương án, hàm một chiều $g()$ có thể cũng là đồng cấu, có nghĩa là nếu $\mathcal{D}$ và $\mathcal{R}$ xác định hai nhóm abel, thì đối với mỗi cặp $(a, b) \in \mathcal{D}^2$, được cho là $g(a \oplus b) = g(a) \oplus g(b)$, trong đó $\oplus$ chỉ phép toán nhóm, bao gồm, ví dụ, cộng hoặc nhân. Nếu $g()$ là đồng cấu cộng, ví dụ, các tính chất của $g()$ sẽ đề xuất là $g(a + b) = g(a) + g(b)$. Để dễ minh họa, và không mất tính tổng quát, các mô tả sau đây và các hình vẽ kèm theo có thể sử dụng "+" để chỉ ra là "$\oplus$."

Theo một số phương án, H_N có thể được tính toán dựa trên chuỗi hàm một chiều, là ứng dụng kế tiếp của $g()$ đến giá trị ban đầu H_0. Theo một số phương án, giá trị H_0 có thể được chọn ngẫu nhiên. H_1 sau đó có thể được tính toán là $H_1 = g(H_0)$, H_2 có thể được tính toán là $H_2 = g(H_1)$, và v.v. Ứng dụng kế tiếp của $g()$ có thể tiếp tục cho đến khi H_N được tính toán là $H_N = g(H_{N-1})$. Δ_N có thể được tính toán theo cách thức giống hoặc tương tự. Nghĩa là, giá trị Δ_0 có thể được chọn ngẫu nhiên. Δ_1 sau đó có thể được tính toán là $\Delta_1 = g(\Delta_0)$, Δ_2 có thể được tính toán là $\Delta_2 = g(\Delta_1)$, và v.v. Ứng dụng kế tiếp của $g()$ có thể tiếp tục cho đến khi Δ_N được tính toán là $\Delta_N = g(\Delta_{N-1})$.

Tại bước 304, người dùng 1 có thể gửi các tham số, bao gồm, ví dụ, H_N, $H_N + \Delta_N$, Δ_0, số lượng tối đa N, thời điểm hết hiệu lực T_1 khi thỏa thuận hết hiệu lực trên chuỗi khối A, và thời điểm hết hiệu lực T_2 khi thỏa thuận hết hiệu lực trên chuỗi khối B, đến người dùng 2. Tuy nhiên, người dùng 1 có thể không chia sẻ giá trị của H_0 với người dùng 2. Theo một số phương án, người dùng 2 có thể có lựa chọn để xác minh liệu các giá trị của N, T_1, và T_2 có phù hợp với các điều khoản của thỏa thuận đạt được giữa người dùng 1 và 2 hay không. Nếu các giá trị của N, T_1, và T_2 không phù hợp với các điều khoản được thỏa thuận, người dùng 2 có thể gửi thông báo lỗi đến người dùng 1 chỉ ra sự không phù hợp. Mặc khác, nếu các giá trị của N, T_1, và T_2 phù hợp với các điều khoản được thỏa thuận, người dùng 2 có thể gửi thông báo xác nhận đến người dùng 1.

Theo một số phương án, người dùng 2 cũng có thể tiến hành bước 306 để xác minh tính hợp lệ của các giá trị là H_N, $H_N + \Delta_N$, và Δ_0. Ví dụ, người dùng 2 có thể tính toán độc lập $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$. Người dùng 2 sau đó có thể tính toán giá trị của $(H_N)+(\Delta_N)$, trong đó (H_N) biểu diễn giá trị là H_N khi được nhận từ người dùng 1 và (Δ_N) biểu diễn giá trị của Δ_N được tính toán độc lập bởi người dùng 2. Người dùng 2 có thể so sánh giá trị của $(H_N)+(\Delta_N)$ với giá trị của $H_N + \Delta_N$ được nhận từ người dùng 1. Nếu $H_N + \Delta_N \neq (H_N)+(\Delta_N)$, người dùng 2 có thể gửi thông báo lỗi đến người dùng 1 chỉ ra sự không phù hợp. Mặc khác, nếu $H_N + \Delta_N = (H_N)+(\Delta_N)$, người dùng 2 có thể gửi thông báo xác nhận đến người dùng 1. Theo một số phương án, người dùng 2 có thể bỏ qua bước 306 nếu Người

dùng 1 được tin tưởng bởi người dùng 2. Theo một số phương án, người dùng 2 có thể luôn tiến hành bước 306 để xác minh tính hợp lệ của H_N, $H_N + \Delta_N$, và Δ_0.

Theo một số phương án, các bước từ 302 đến 306 có thể được thực hiện ngoài chuỗi. Ví dụ, người dùng 1 và 2 không cần sử dụng các chuỗi khối A và B để tạo thuận lợi các cuộc đàm phán của họ liên quan đến các điều khoản của hợp đồng AMS. Theo phương thức này, các điều khoản của hợp đồng AMS có thể được ghi trên các chuỗi khối A và B, sau khi người dùng 1 và 2 đạt được thỏa thuận. Theo một số phương án, các bước từ 302 đến 306 có thể chỉ cần được thực hiện một lần cho hợp đồng AMS.

Sau khi người dùng 1 và 2 đạt được thỏa thuận và người dùng 1 tạo ra và gửi các tham số đến người dùng 2, hợp đồng AMS có thể được thiết lập. Tại bước 308, người dùng 1 có thể thiết lập hợp đồng AMS bằng cách yêu cầu việc ghi trên chuỗi khối A của các tham số bao gồm, ví dụ, H_N, đóng vai trò là khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối A, số lượng tối đa N, thời điểm hết hiệu lực T_1 khi thỏa thuận hết hiệu lực trên chuỗi khối A, và thao tác người dùng 1 đồng ý thực hiện, tức là, chuyển tài sản X đến người dùng 2. Theo một số phương án, người dùng 1 có thể yêu cầu sự ghi các tham số trên chuỗi khối A bằng cách viện dẫn hàm, ví dụ, setupAMS(Contract_ID_A, Object), trở thành khả dụng trên chuỗi khối A thông qua giao diện hợp đồng thông minh. Tại bước 310, chuỗi khối A có thể tiến hành các lệnh được quy ước ở setupAMS(Contract_ID_A, Object) và ghi các tham số được quy ước bởi người dùng 1 trên chuỗi khối A. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu được đối tượng có thể là biến, cấu trúc dữ liệu, hàm, hoặc phương pháp trong lĩnh vực công nghệ máy tính.

Theo một số phương án, giá trị của Contract_ID_A được quy ước ở setupAMS(Contract_ID_A, Object) có thể biểu diễn mã nhận dạng duy nhất, ví dụ, mã nhận dạng duy nhất phổ biến, được chỉ định hoặc được tạo ra để nhận dạng hợp đồng AMS trên chuỗi khối A. Theo một số phương án, giá trị của khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối A, ví dụ, H_N, có thể được sử dụng làm mã nhận dạng duy nhất để nhận dạng hợp đồng AMS trên chuỗi khối A. Giá trị của đối tượng có thể biểu diễn đối tượng chứa các tham số, bao gồm H_N, N, T_1, và thao tác

mà người dùng 1 đồng ý thực hiện, được ghi trên chuỗi khối A. Theo một phương án, đối tượng có thể được xác định như sau:

```
Object
{
        N; // maximum number of swaps agreed by both parties
        L; // function lock of the contract, with initial value set to H_N
        T; // contract expiration time, which is set to T_1 for Blockchain
A
        Action; // action to be performed on Blockchain A
}
```

Theo phương thức này, người dùng 1 có thể quy ước, sử dụng đối tượng được ghi trên chuỗi khối A, trong đó Người dùng 1 đồng ý thao tác chuyển tài sản X đến người dùng 2 trên chuỗi khối A (như được quy ước trong Object.Action) nếu người dùng 2 có thể mở khóa khóa hàm (như được quy ước trong Object.L) trước khi hợp đồng AMS hết hiệu lực trên chuỗi khối A (như được quy ước trong Object.T). Người dùng 1 cũng có thể quy ước rằng người dùng 1 đồng ý cho phép thao tác này được thực hiện lên đến N lần (như được quy ước trong Object.N).

Người dùng 2 có thể thiết lập hợp đồng AMS trên chuỗi khối B theo cách thức tương tự. Ví dụ, tại bước 312, người dùng 2 có thể thiết lập hợp đồng AMS bằng cách yêu cầu việc ghi trên chuỗi khối B của các tham số bao gồm, ví dụ, $H_N + \Delta_N$, đóng vai trò là khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối B, số lượng tối đa N, thời điểm hết hiệu lực T_2 khi thỏa thuận hết hiệu lực trên chuỗi khối B, và thao tác mà người dùng 2 đồng ý thực hiện, tức là, chuyển tài sản Y đến người dùng 1. Theo một số phương án, người dùng 2 có thể thực hiện bước 312 sau bước 310. Vì người dùng 2 có tài khoản trên chuỗi khối A, người dùng 2 có khả năng nhận dạng các tham số hợp đồng AMS mà người dùng 1 được ghi trên chuỗi khối A, bao gồm người dùng 1 của Contract_ID_A đã sử dụng để nhận dạng hợp đồng AMS. Theo cách khác hoặc ngoài ra, người dùng 1 có thể cung cấp Contract_ID_A cho người dùng 2 thông qua kênh truyền thông khác.

Theo một số phương án, người dùng 2 có thể yêu cầu việc ghi nhận các tham số trên chuỗi khối B bằng cách viện dẫn hàm, ví dụ, setupAMS(Contract_ID_B, Object), trở thành khả dụng trên chuỗi khối B thông qua giao diện hợp đồng thông minh. Tại bước 314, chuỗi khối B có thể tiến hành các lệnh được quy ước trong

setupAMS(Contract_ID_B, Object) và ghi các tham số được quy ước bởi người dùng 2 trên chuỗi khối B.

Theo một số phương án, giá trị của Contract_ID_B được quy ước trong setupAMS(Contract_ID_B, Object) có thể khác với giá trị của Contract_ID_A để ngăn ngừa bên thứ ba khỏi liên kết các mã nhận dạng hợp đồng được ghi trên các chuỗi khối A và B cùng nhau. Ví dụ, giá trị của Contract_ID_B được quy ước trong setupAMS(Contract_ID_B, Object) có thể biểu diễn mã nhận dạng duy nhất khác, ví dụ, mã nhận dạng duy nhất phổ biến, được chỉ định hoặc được tạo ra để nhận dạng hợp đồng AMS trên chuỗi khối B. Theo một số phương án, giá trị của khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối B, ví dụ, $H_N + \Delta_N$, có thể được sử dụng làm mã nhận dạng duy nhất để nhận dạng hợp đồng AMS trên chuỗi khối B. Giá trị của đối tượng có thể biểu diễn đối tượng chứa các tham số, bao gồm $H_N + \Delta_N$, N, T_2, và thao tác mà người dùng 2 đồng ý thực hiện trên chuỗi khối B. Theo một phương án, đối tượng có thể được xác định như sau:

```
Object
{
        N; // maximum number of swaps agreed by both parties
        L; // function lock of the contract, with initial value set to H_N +
Δ_N
        T; // contract expiration time, which is set to T_2 for Blockchain
B
        Action; // action to be performed on Blockchain B
}
```

Theo phương thức này, người dùng 2 có thể quy ước, sử dụng đối tượng được ghi trên chuỗi khối B, người dùng 2 đồng ý thao tác chuyển tài sản Y đến người dùng 1 trên chuỗi khối B (như được quy ước trong Object.Action) nếu người dùng 1 có thể mở khóa khóa hàm (như được quy ước trong Object.L) trước khi hợp đồng AMS hết hiệu lực trên chuỗi khối B (như được quy ước trong Object.T). Người dùng 2 cũng có thể quy ước rằng người dùng 2 đồng ý cho phép thao tác này được thực hiện lên đến N lần (như được quy ước trong Object.N).

Theo một số phương án, người dùng 1 và 2 có thể chỉ cần tiến hành các bước 308 đến 314 một lần để thiết lập hợp đồng AMS. Sau khi hợp đồng AMS được thiết lập, người dùng 1 và 2 có thể thực thi hợp đồng AMS để tạo thuận lợi hoán đổi các

tài sản X và Y lên đến N lần.

Fig.4 minh họa lưu đồ của phương pháp 400 để thực thi hợp đồng AMS, ví dụ, hợp đồng AMS thiết lập bởi người dùng 1 và 2, theo một phương án. Tại bước 402, người dùng 1 có thể khởi tạo hoán đổi thứ nhất bằng cách viện dẫn hàm trên chuỗi khối B, ví dụ, callAMS(Contract_ID_B, Key), để yêu cầu chuyển tài sản Y từ người dùng 2 đến người dùng 1 trên chuỗi khối B. Người dùng 1 có thể sử dụng Contract_ID_B để nhận dạng hợp đồng AMS được thiết lập bởi người dùng 1 và 2, và sử dụng khóa (key) để quy ước khóa có thể được sử dụng để mở khóa khóa hàm được áp đặt bởi hợp đồng AMS trên chuỗi khối B. Theo một số phương án, khóa hàm được áp đặt bởi hợp đồng AMS trên chuỗi khối B được thiết lập cho $H_N + \Delta_N$ ban đầu, có thể được mở khóa sử dụng $H_{N-1} + \Delta_{N-1}$.

Tại bước 404, chuỗi khối B có thể tiến hành các lệnh được quy ước trong callAMS(Contract_ID_B, Key). Các lệnh này có thể được minh họa sử dụng mã giả được xác định sau đây:

```
callAMS(Contract_ID_B, Key)
{
        Retrieve recorded Object from Blockchain B using Contract_ID_B;

        IF g(Key) == Object.L AND time < Object.T AND Object.N > 0
        {
                Perform Object.Action on Blockchain B;
                Set Object.L to Key;
                Reduce Object.N by 1;
        }
}
```

Lưu ý rằng vì khóa hàm được áp đặt bởi hợp đồng AMS trên chuỗi khối B được thiết đặt cho $H_N + \Delta_N$ ban đầu, viện dẫn callAMS(Contract_ID_B, Key) và thiết lập khóa (Key) cho $H_{N-1} + \Delta_{N-1}$ sẽ cho phép người dùng 1 mở khóa khóa hàm ban đầu. Vì người dùng 1 được tính toán H_N và Δ_N dựa trên chuỗi hàm (bước 302, Fig.3), người dùng 1 có các giá trị của toàn bộ chuỗi, bao gồm $H_{N-1} = g(H_{N-2})$, $H_{N-2} = g(H_{N-3})$, ... H_0 và $\Delta_{N-1} = g(\Delta_{N-2})$, $\Delta_{N-2} = g(\Delta_{N-3})$, ... Δ_0. Do đó, miễn là người dùng 1 viện dẫn callAMS(Contract_ID_B, Key) trước khi Object.T (là T_2 vì hàm được viện dẫn trên chuỗi khối B), người dùng 1 có thể sẽ thỏa mãn cả khóa hàm và khóa hẹn giờ được áp đặt bởi hợp đồng AMS. Chuỗi khối

B có thể sau đó tiếp tục thực hiện thao tác được ghi (tức là, Object.Action, để chuyển tài sản Y từ người dùng 2 đến người dùng 1 trên chuỗi khối B), cập nhật khóa hàm bằng cách thiết lập Object.L cho khóa (Key) (tức là, thiết lập Object.L cho $H_{N-1} + \Delta_{N-1}$, có thể được mở khóa sử dụng giá trị trước đó trong chuỗi hàm, $H_{N-2} + \Delta_{N-2}$), và làm giảm giá trị của Object.N đi một.

Tại bước 406, người dùng 2 có thể phản hồi hoạt động hoán đổi được khởi tạo bởi người dùng 1 bằng cách viện dẫn hàm trên chuỗi khối A, ví dụ, callAMS(Contract_ID_A, Key), để yêu cầu chuyển tài sản X từ người dùng 1 đến người dùng 2 trên chuỗi khối A. Người dùng 2 có thể sử dụng Contract_ID_A để nhận dạng hợp đồng AMS được thiết lập bởi người dùng 1 và 2, và sử dụng khóa (Key) để chỉ rõ khóa này có thể được sử dụng để mở khóa khóa hàm được áp đặt bởi hợp đồng AMS. Theo một số phương án, khóa hàm này được áp đặt bởi hợp đồng AMS trên chuỗi khối A được thiết đặt H_N ban đầu, có thể được mở khóa sử dụng H_{N-1}.

Người dùng 2 có khả năng tính toán giá trị của H_{N-1} sau bước 404 vì người dùng 2 có tài khoản trên chuỗi khối B và có thể truy hồi các bản ghi từ chuỗi khối B, chứa bản ghi về sự viện dẫn của người dùng 1 của callAMS(Contract_ID_B, Key) được thực hiện tại bước 404. Hơn nữa, vì người dùng 1 đã viện dẫn callAMS(Contract_ID_B, Key) tại bước 404 bằng cách thiết lập khóa (Key) cho $H_{N-1} + \Delta_{N-1}$, người dùng 2 có thể truy hồi giá trị của $H_{N-1} + \Delta_{N-1}$, có thể được sử dụng để tính toán giá trị của H_{N-1} bằng cách trừ đi Δ_{N-1} từ $H_{N-1} + \Delta_{N-1}$. Giá trị của Δ_{N-1} có thể khả dụng với người dùng 2 dựa trên các tính toán mà người dùng 2 đã thực hiện tại bước 306 (Fig.3). Theo cách khác, người dùng 2 có thể tính toán giá trị của Δ_{N-1} trong thời gian thực sử dụng chuỗi hàm như được xác định tại bước 306 (Fig.3). Trong cả hai trường hợp, người dùng 2 có thể tính toán $H_{N-1} = (H_{N-1} + \Delta_{N-1}) - \Delta_{N-1}$ và sử dụng giá trị của H_{N-1} để mở khóa hợp đồng AMS trên chuỗi khối A.

Tại bước 408, chuỗi khối A có thể tiến hành các lệnh được quy ước trong callAMS(Contract_ID_A, Key). Các lệnh này có thể được minh họa sử dụng mã giả được xác định sau đây:

```
callAMS(Contract_ID_A, Key)
{
        Retrieve recorded Object from Blockchain A using Contract_ID_A;
        If g(Key) == Object.L and time < Object.T and Object.N > 0
        {
                Perform Object.Action on Blockchain A;
                Set Object.L to Key;
                Reduce Object.N by 1;
        }
}
```

Lưu ý rằng vì khóa hàm được áp đặt bởi hợp đồng AMS trên chuỗi khối A được thiết đặt cho H_N ban đầu, viện dẫn callAMS(Contract_ID_A, Key) và thiết lập khóa (Key) cho H_{N-1} sẽ cho phép người dùng 2 mở khóa khóa hàm ban đầu. Do đó, miễn là người dùng 2 viện dẫn callAMS(Contract_ID_A, Key) trước khi Object.T (là T_1 vì hàm được viện dẫn trên chuỗi khối A), người dùng 2 sẽ có thể đáp ứng cả khóa hàm và khóa hẹn giờ được áp đặt bởi hợp đồng AMS. Chuỗi khối A sau đó có thể tiếp tục thực hiện thao tác được ghi (tức là, Object.Action, để chuyển tài sản X từ người dùng 1 đến người dùng 2 trên chuỗi khối A), cập nhật khóa hàm bằng cách thiết lập Object.L cho khóa (Key) (tức là, thiết lập Object.H đến H_{N-1}, có thể được mở khóa sử dụng H_{N-2}), và làm giảm giá trị của Object.N đi một. Tại điểm này, người dùng 1 sở hữu tài sản Y trên chuỗi khối B và người dùng 2 sở hữu tài sản X trên chuỗi khối A. Hoạt động hoán đổi thứ nhất được xem là hoàn thành.

Hợp đồng AMS có thể được thực thi lại để tiến hành các hoạt động hoán đổi bổ sung. Ví dụ, người dùng 1 có thể khởi tạo hoán đổi thứ hai bằng cách viện dẫn hàm callAMS(Contract_ID_B, Key) tại bước 410, có thể làm cho chuỗi khối B tiến hành các lệnh được quy ước trong callAMS(Contract_ID_B, Key) tại bước 412. Lưu ý rằng khi người dùng 1 viện dẫn callAMS(Contract_ID_B, Key) cho lần thứ hai, người dùng 1 có thể thiết đặt khóa (Key) cho $H_{N-2} + \Delta_{N-2}$, được biết với người dùng 1 (được tính toán bởi người dùng 1 tại bước 302, Fig.3) và có thể được sử dụng bởi người dùng 1 để mở khóa khóa hàm được cập nhật, Object.L trên chuỗi khối B, vì Object.L trên chuỗi khối B được thiết đặt cho $H_{N-1} + \Delta_{N-1}$ sau hoán đổi thứ nhất. Người dùng 2 có thể truy hồi bản ghi của việc viện dẫn thứ hai của người dùng 1 của callAMS(Contract_ID_B, Key) được thực hiện tại bước 412 và đạt được giá trị của khóa (Key), được thiết đặt cho $H_{N-2} + \Delta_{N-2}$ bởi người dùng 1. Theo

phương thức này, người dùng 2 có thể tính toán $H_{N-2} = (H_{N-2} + \Delta_{N-2}) - \Delta_{N-2}$ dưới dạng khóa để mở khóa hợp đồng AMS trên chuỗi khối A cho lần thứ hai, cho phép người dùng 2 phản hồi bằng cách viện dẫn hàm callAMS(Contract_ID_A, Key) tại bước 414, có thể làm cho chuỗi khối A tiến hành các lệnh được quy ước trong callAMS(Contract_ID_A, Key) tại bước 416.

Theo một số phương án, người dùng 1 và 2 có thể viện dẫn hàm callAMS() lên đến N lần để thực hiện lên đến N hoán đổi. Người dùng 1 có thể thiết đặt khóa (Key) cho $H_{N-n} + \Delta_{N-n}$ mỗi lần mà người dùng 1 viện dẫn callAMS(Contract_ID_B, Key), trong đó n là số lần callAMS(Contract_ID_B, Key) được viện dẫn. Ví dụ, lần thứ nhất người dùng 1 viện dẫn callAMS(Contract_ID_B, Key), người dùng 1 có thể thiết đặt khóa (Key) cho $H_{N-1} + \Delta_{N-1}$, lần thứ hai người dùng 1 viện dẫn callAMS(Contract_ID_B, Key), người dùng 1 có thể thiết đặt khóa (Key) cho $H_{N-2} + \Delta_{N-2}$, lần thứ ba mà người dùng 1 viện dẫn callAMS(Contract_ID_B, Key), người dùng 1 có thể thiết đặt khóa (Key) cho $H_{N-3} + \Delta_{N-3}$, và v.v. Theo phương thức này, người dùng 1 có thể sử dụng các giá trị trong chuỗi hàm $H_{N-1} = g(H_{N-2})$, $H_{N-2} = g(H_{N-3})$, … H_0 và $\Delta_{N-1} = g(\Delta_{N-2})$, $\Delta_{N-2} = g(\Delta_{N-3})$, … Δ_0 để mở khóa hợp đồng AMS trên chuỗi khối B khi người dùng 1 tiếp tục viện dẫn callAMS(Contract_ID_B, Key) trên chuỗi khối B. Người dùng 2 có thể tính toán các giá trị của $H_{N-1} = (H_{N-1} + \Delta_{N-1}) - \Delta_{N-1}$, $H_{N-2} = (H_{N-2} + \Delta_{N-2}) - \Delta_{N-2}$, … $H_0 = (H_0 + \Delta_0) - \Delta_0$ và sử dụng các giá trị được tính toán để mở khóa hợp đồng AMS trên chuỗi khối A khi người dùng 2 tiếp tục viện dẫn callAMS(Contract_ID_A, Key) trên chuỗi khối A.

Lưu ý rằng vì chuỗi khối A sử dụng các khóa hàm được biểu diễn là H_N, H_{N-1}, H_{N-2},… H_0 và chuỗi khối B sử dụng các khóa hàm được biểu diễn là $H_N + \Delta_N$, $H_{N-1} + \Delta_{N-1}$, $H_{N-2} + \Delta_{N-2}$,… $H_0 + \Delta_0$, bên thứ ba sẽ không có khả năng liên kết chúng với nhau vì các giá trị của Δ_N, Δ_{N-1}, Δ_{N-2},… Δ_0 không được biết đối với bên thứ ba. Theo đó, các giao dịch được tiến hành trên các chuỗi khối A và B phản hồi callAMS() có thể được che dấu, nhờ đó bảo vệ tính bí mật của người dùng 1 và 2.

Cũng được lưu ý rằng việc sử dụng các khóa hàm được biểu diễn là H_N,

$H_{N-1}, H_{N-2},\ldots H_0$ trên chuỗi khối A và sử dụng các khóa hàm được biểu diễn là $H_N + \Delta_N, H_{N-1} + \Delta_{N-1}, H_{N-2} + \Delta_{N-2},\ldots H_0 + \Delta_0$ trên chuỗi khối B chỉ được thể hiện là các ví dụ và không có nghĩa là bị giới hạn. Được dự tính rằng các khóa hàm được biểu diễn là $H_N, H_{N-1}, H_{N-2},\ldots H_0$ có thể được sử dụng trên chuỗi khối B (thay cho chuỗi khối A) và các khóa hàm được biểu diễn là $H_N + \Delta_N, H_{N-1} + \Delta_{N-1}$, $H_{N-2} + \Delta_{N-2},\ldots H_0 + \Delta_0$ có thể được sử dụng trên chuỗi khối A (thay cho chuỗi khối B) mà không làm ảnh hưởng đến khả năng của chúng để bảo vệ bí mật của người dùng 1 và 2. Ví dụ, theo một số phương án, người dùng 1 có thể thiết đặt $H_N + \Delta_N$ là khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối A tại bước 308 (Fig.3) và người dùng 2 có thể thiết đặt H_N là khóa hàm ban đầu của hợp đồng AMS trên chuỗi khối B tại bước 312 (Fig.3). Tiếp theo, tại bước 402 (Fig.2), người dùng 1 có thể sử dụng H_{N-1} là khóa (Key) khi viện dẫn callAMS(Contract_ID_B, Key) để mở khóa khóa hàm H_N trên chuỗi khối B, và tại bước 406, người dùng 2 có thể truy hồi giá trị của H_{N-1} từ viện dẫn của người dùng 1 của callAMS(Contract_ID_A, Key) và tính toán giá trị của $H_{N-1} + \Delta_{N-1}$, có thể được sử dụng bởi người dùng 2 để mở khóa khóa hàm $H_N + \Delta_N$ trên chuỗi khối A. Người dùng 1 và 2 có thể viện dẫn hàm callAMS() lên đến N lần để thực hiện lên đến N lần hoán đổi, như được mô tả ở trên.

Theo một số phương án, các chuỗi khối A và B có thể theo dõi số lần mà người dùng 1 và 2 được cho phép để thực thi hoạt động hoán đổi được quy ước trong hợp đồng AMS bằng cách giảm đi giá trị của N (được ghi trên các chuỗi khối là Object.N) mỗi lần hoạt động hoán đổi được thực hiện. Theo phương thức này, các chuỗi khối A và B có thể ngăn ngừa việc thực thi trái phép của hoạt động hoán đổi ngay khi giá trị của N đạt đến 0. Theo một số phương án, các chuỗi khối A và B có thể xóa đối tượng ngay khi giá trị N đạt đến 0.

Theo một số phương án, chuỗi khối B có thể chỉ cho phép người dùng 1 thực hiện hoạt động hoán đổi trước khi hợp đồng AMS hết hiệu lực trên chuỗi khối B tại thời điểm T_2. Tương tự, chuỗi khối A có thể chỉ cho phép người dùng 2 thực hiện hoạt động hoán đổi trước khi hợp đồng AMS hết hiệu lực trên chuỗi khối A tại thời điểm T_1. Theo phương thức này, các chuỗi khối A và B có thể ngăn ngừa việc thực

thi trái phép của các hoạt động hoán đổi sau khi hợp đồng AMS được thiết đặt là hết hiệu lực. Theo một số phương án trong đó người dùng 2 là người dùng phản hồi hoạt động hoán đổi được khởi tạo bởi người dùng 1, người dùng 2 có thể yêu cầu thiết lập hợp đồng AMS là hết hiệu lực trên chuỗi khối A tại thời điểm T_1 sau T_2, tức là, $T_2 < T_1$. Theo một số phương án, người dùng 2 có thể xác minh liệu T_2 có thật sự được thiết đặt là nhỏ hơn T_1 hay không khi người dùng 2 nhận các tham số từ người dùng 1 (Fig.3 tại bước 304). Nếu T_2 lớn hơn hoặc bằng T_1, người dùng 2 có thể gửi thông báo lỗi đến người dùng 1.

Theo một số phương án, người dùng 1 và 2 cũng có thể chỉ ra sự sắp thứ tự lên đến N lần hoán đổi trong hợp đồng AMS. Nói cách khác, người dùng 1 và 2 không những có thể chỉ rõ số lần hoán đổi mà họ đã đồng ý thực hiện, mà còn xếp thứ tự các hoán đổi này. Ví dụ, người dùng 1 và 2 có thể đồng ý hoán đổi tài sản X1 với tài sản Y1 trước tiên, được theo sau bởi hoán đổi tài sản X2 với tài sản Y2, và v.v. Theo một số phương án, sự sắp thứ tự có thể được chỉ ra dưới dạng danh sách theo thứ tự trong đối tượng được xác định như sau:

```
Object
{
        N; // maximum number of swaps agreed by both parties
        L; // function lock of the contract
        T; // contract expiration time
        ActionArray; // an ordered list of actions to be performed
}
```

Cần hiểu rằng danh sách theo thứ tự của các thao tác có thể được thực thi theo các cách tương tự với nội dung được mô tả ở trên. Tuy nhiên, thay cho việc lặp lại các lần thực thi Object.Action lên đến N lần, chuỗi khối để hỗ trợ việc thực thi danh sách thứ tự của các thao tác có thể sử dụng hợp đồng thông minh để thực thi các lệnh tương tự với mã giả được xác định như sau:

```
callAMS(Contract_ID, Key)
{
        Retrieve recorded Object from Blockchain using Contract_ID;

        IF g(Key) == Object.L AND time < Object.T AND Object.N > 0
        {
                Perform Object.ActionArray[N-Object.N];
                Set Object.L to Key;
                Reduce Object.N by 1;
```

}
}

Như được minh họa trong mã giả nêu trên, chỉ số đối với Object.ActionArray, [N-Object.N], có thể chỉ ra phần tử thứ nhất của mảng khi callAMS() được viện dẫn cho lần thứ nhất, cho phép chuỗi khối thực hiện thao tác được quy ước trong Object.ActionArray[0]. Tiếp theo, chỉ số [N-Object.N] có thể tăng lên khi giá trị của Object.N giảm xuống, cho phép chuỗi khối thực hiện các thao tác liên tiếp được quy ước trong Object.ActionArray theo cách có thứ tự.

Cần hiểu rằng các đối tượng được xác định ở trên được thể hiện là các ví dụ và không có nghĩa giới hạn. Cần hiểu rằng các tham số được chứa trong các đối tượng này có thể được lưu trữ trên các chuỗi khối theo các cách thức khác nhau. Ví dụ, theo một số phương án, các tham số khác nhau được mô tả ở trên có thể được lưu trữ trong các định dạng như các cặp giá trị chính hoặc tương tự. Theo một số phương án, các tiền tố có thể cũng được sử dụng để giúp phân biệt các cặp giá trị chính liên quan đến các hợp đồng AMS khác nhau. Cũng được hiểu rằng các khai báo của các hàm và mã giả được mô tả ở trên được thể hiện là các ví dụ và không có nghĩa là giới hạn.

Fig.5 minh họa lưu đồ của phương pháp 500 để thực thi hợp đồng AMS, theo một phương án. Phương pháp 500 có thể được thực hiện bằng một hoặc nhiều nút trong hệ thống chuỗi khối, ví dụ, các nút 102-110 trong hệ thống chuỗi khối 100 (Fig.1). Các nút 102-110 trong hệ thống chuỗi khối 100 có thể thực hiện các hoạt động trên chuỗi khối, ví dụ, chuỗi khối 120 (Fig.1). Chuỗi khối 120 có thể được triển khai dưới dạng chuỗi khối A hoặc B trong các ví dụ được mô tả ở trên.

Tại bước 502, nút, ví dụ, nút 102, có thể ghi các tham số cho AMS trên chuỗi khối 120. Tiếp tục với các ví dụ được mô tả ở trên, nút 102 có thể nhận các tham số từ người dùng thứ nhất, ví dụ, người dùng 1 (Fig.3). Các tham số có thể bao gồm khóa hàm, thời điểm hết hiệu lực, ít nhất một thao tác được thực hiện trên chuỗi khối 120 khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, cũng là số lượng tối đa của các hoán đổi được thỏa thuận, trước thời điểm hết hiệu lực. Theo phương thức này, người dùng 1 có thể quy ước, sử dụng các tham số được ghi trên chuỗi khối 120, người dùng 1 đồng ý để cho ít

nhất một thao tác được thực hiện trên chuỗi khối 120 nếu người dùng thứ hai có thể mở khóa khóa hàm trước thời điểm hết hiệu lực. Người dùng 1 có thể cũng chỉ rõ rằng người dùng 1 đồng ý cho phép hợp đồng AMS được thực thi lên đến N lần.

Tại bước 504, nút 102 có thể nhận, từ người dùng thứ hai, ví dụ, người dùng 2 (Fig.4), yêu cầu thứ nhất để thực thi hợp đồng AMS và khóa thứ nhất để mở khóa khóa hàm của hợp đồng AMS. Theo một số phương án, người dùng 2 có thể đưa ra yêu cầu thứ nhất thông qua hàm call. Ví dụ, người dùng 2 có thể gọi hàm callAMS(Contract_ID_A, Key) được mô tả ở trên (bước 406 của Fig.4).

Tại bước 506, nút 102 có thể xác định liệu có xử lý yêu cầu thứ nhất hay không dựa trên một hoặc nhiều trong số khóa thứ nhất, thời điểm hết hiệu lực, và số lần tối đa hợp đồng AMS có thể được thực thi. Theo một số phương án, nút 102 có thể áp dụng hàm một chiều cho khóa thứ nhất để thu được giá trị hàm thứ nhất và xác định liệu giá trị hàm thứ nhất có tương thích với khóa hàm hay không. Nếu giá trị hàm thứ nhất không tương thích với khóa hàm, nút 102 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, nút 102 có thể xác định liệu hợp đồng AMS có hết hiệu lực hay không dựa trên thời điểm hết hiệu lực. Nếu hợp đồng AMS hết hiệu lực, nút 102 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, nút 102 có thể xác định liệu hợp đồng AMS có được thực thi với số lần tối đa hay không. Nếu hợp đồng AMS được thực thi với số lần tối đa, nút 102 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, nút 102 có thể xác định để xử lý yêu cầu thứ nhất chỉ khi được xác định rằng giá trị hàm thứ nhất tương thích với khóa hàm, hợp đồng AMS chưa hết hiệu lực, và hợp đồng AMS không được thực thi với số lần tối đa.

Nếu nút 102 xác định để xử lý yêu cầu thứ nhất, nút 102 có thể tiếp tục đến bước 508. Tại bước 508, nút 102 có thể thực thi ít nhất một thao tác được xác định trong hợp đồng AMS trên chuỗi khối 120. Nút 102 cũng có thể cập nhật khóa hàm của hợp đồng AMS với khóa thứ nhất sao cho khóa thứ nhất sẽ dùng làm khóa hàm đối với yêu cầu thứ hai để thực thi hợp đồng AMS. Theo phương thức này, nút 102 có thể lặp lại các bước 504-508 khi nút 102 nhận yêu cầu thứ hai từ người dùng 2. Nút 102 có thể lặp lại các bước 504-508 lên đến số lần tối đa để hợp đồng AMS có

thể được thực thi.

Theo một số phương án, ít nhất một thao tác được thực hiện trên chuỗi khối 120 có thể bao gồm ít nhất một thao tác để chuyển tài sản được ghi trên chuỗi khối 120 từ người dùng thứ nhất, ví dụ, người dùng 1, đến người dùng thứ hai, ví dụ, người dùng 2.

Theo một số phương án, ít nhất một thao tác được thực hiện trên chuỗi khối có thể bao gồm danh sách thứ tự của các thao tác được thực hiện trên chuỗi khối 120. Nút 102 có thể thực thi thao tác thứ nhất được liệt kê trong danh sách thứ tự của các thao tác trên chuỗi khối 120 khi nút 102 xác định để xử lý yêu cầu thứ nhất để thực thi hợp đồng AMS. Nút 102 có thể thực thi thao tác thứ hai được liệt kê trong danh sách thứ tự của các thao tác trên chuỗi khối 120 khi nút 102 xác định để xử lý yêu cầu thứ hai để thực thi hợp đồng AMS. Theo phương thức này, nút 102 có thể cho phép người dùng không những chỉ rõ số lần tối đa hợp đồng AMS được cho phép để thực thi, mà còn việc sắp xếp các thao tác được thực thi.

Fig.6 là sơ đồ khối của thiết bị thực thi hợp đồng AMS 600, theo một phương án. Thiết bị 600 có thể là phương án thực hiện của việc xử lý phần mềm, và có thể tương ứng với phương pháp 500 (Fig.5). Tham chiếu đến Fig.6, thiết bị 600 có thể bao gồm môđun nhận 602, môđun ghi 604, môđun xác định 606, và môđun thực thi 608.

Môđun nhận 602 có thể thu thông tin từ người dùng của chuỗi khối. Theo một số phương án, môđun nhận 602 có thể nhận, từ người dùng thứ nhất, các tham số đối với hợp đồng AMS. Các tham số có thể bao gồm khóa hàm, thời điểm hết hiệu lực, ít nhất một thao tác được thực hiện trên chuỗi khối khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi trước thời điểm hết hiệu lực. Môđun nhận 602 có thể cung cấp các tham số được nhận cho môđun ghi 604.

Môđun ghi 604 có thể ghi các tham số được nhận trên chuỗi khối. Môđun ghi 604 có thể ghi các tham số được nhận ở các định dạng khác nhau. Ví dụ, môđun ghi 604 có thể tạo ra đối tượng mà chứa các tham số được nhận dưới dạng các tính chất của đối tượng và ghi đối tượng này trên chuỗi khối. Môđun ghi 604 có thể

cũng ghi các tham số được nhận trong các định dạng khác, bao gồm các cặp giá trị chính và tương tự.

Theo một số phương án, môđun nhận 602 có thể cũng nhận các yêu cầu từ người dùng của chuỗi khối. Theo một số phương án, môđun nhận 602 có thể nhận, từ người dùng thứ hai, yêu cầu thứ nhất để thực thi hợp đồng AMS và khóa thứ nhất để mở khóa khóa hàm của hợp đồng AMS. Môđun nhận 602 có thể đưa ra yêu cầu được nhận đến môđun xác định 606.

Môđun xác định 606 có thể xác định liệu có xử lý yêu cầu thứ nhất hay không dựa trên khóa thứ nhất, thời điểm hết hiệu lực, và số lần tối đa hợp đồng AMS có thể được thực thi. Theo một số phương án, môđun xác định 606 có thể áp dụng hàm một chiều cho khóa thứ nhất để thu được giá trị hàm thứ nhất và xác định liệu giá trị hàm thứ nhất có tương thích với khóa hàm hay không. Nếu giá trị hàm thứ nhất không tương thích với khóa hàm, môđun xác định 606 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, môđun xác định 606 có thể xác định xem liệu hợp đồng AMS đã hết hiệu lực hay chưa dựa trên thời điểm hết hiệu lực. Nếu hợp đồng AMS đã hết hiệu lực, môđun xác định 606 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, môđun xác định 606 có thể xác định liệu hợp đồng AMS được thực thi với số lần tối đa hay không. Nếu hợp đồng AMS được thực thi với số lần tối đa, môđun xác định 606 có thể xác định là không xử lý yêu cầu thứ nhất. Theo một số phương án, môđun xác định 606 có thể xác định để xử lý yêu cầu thứ nhất chỉ khi được xác định rằng giá trị hàm thứ nhất tương thích với khóa hàm, hợp đồng AMS chưa hết hiệu lực, và hợp đồng AMS không được thực thi với số lần tối đa. Môđun xác định 606 sau đó có thể đưa ra yêu cầu thứ nhất cho môđun thực thi 608.

Môđun thực thi 608 có thể thực thi ít nhất một thao tác được xác định trong hợp đồng AMS trên chuỗi khối. Môđun thực thi 608 có thể cũng cập nhật khóa hàm của hợp đồng AMS với khóa thứ nhất sao cho khóa thứ nhất sẽ dùng làm khóa hàm đối với yêu cầu thứ hai để thực thi hợp đồng AMS. Theo phương thức này, nếu người dùng thứ hai đưa ra yêu cầu thứ hai cho môđun nhận 602 để thực thi lại hợp đồng AMS, môđun nhận 602 có thể đưa ra yêu cầu thứ hai cho môđun xác định

606, có thể xác định liệu có đưa ra yêu cầu thứ hai cho môđun thực thi 608 để thực thi hay không.

Fig.7 minh họa lưu đồ của phương pháp 700 để thực thi hợp đồng AMS, theo một phương án. Phương pháp 700 có thể được thực hiện bởi người dùng sử dụng một hoặc nhiều thiết bị điện toán, ví dụ, thiết bị điện toán 200 (Fig.2). Các thiết bị điện toán này có thể bao gồm các nút trong một hoặc nhiều hệ thống chuỗi khối, nhưng có thể cũng bao gồm các thiết bị ngoài các mạng tạo thành các hệ thống chuỗi khối.

Tại bước 702, người dùng thứ nhất, ví dụ, người dùng 1 (Fig.3), có thể yêu cầu việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, ví dụ, chuỗi khối A (Fig.3). Các tham số thứ nhất có thể bao gồm giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi. Việc ghi các tham số thứ nhất có thể được phát hiện bởi người dùng thứ hai, ví dụ, người dùng 2 (Fig.3).

Tại bước 704, người dùng thứ hai có thể yêu cầu việc ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, ví dụ, chuỗi khối B (Fig.3). Các tham số thứ hai có thể bao gồm giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi.

Theo một số phương án, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất. Theo một số phương án, giá trị ban đầu của khóa hàm thứ nhất có thể được thiết đặt cho H_N, có thể được tính toán bởi người dùng thứ nhất dựa trên ứng dụng kế tiếp thứ nhất của hàm một chiều $g()$ cho giá trị được chọn thứ nhất, H_0. Người dùng thứ nhất có thể, ví dụ, áp dụng hàm một chiều $g()$ N lần để tính toán $H_1 = g(H_0)$, $H_2 = g(H_1)$,... $H_N = g(H_{N-1})$, trong đó N là số lần tối đa hợp đồng AMS có thể được thực thi. Giá trị ban đầu của khóa hàm thứ hai có thể được thiết đặt cho $H_N + \Delta_N$, có thể được tính toán bởi người dùng thứ nhất dựa trên H_N và ứng dụng kế tiếp thứ hai của hàm một chiều $g()$ cho giá trị được chọn thứ hai, Δ_0. Người dùng thứ nhất có thể, ví dụ, áp dụng hàm một chiều

$g()$ N lần để tính toán $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$, và tính toán giá trị của $H_N + \Delta_N$ theo đó. Theo một số phương án, giá trị ban đầu của khóa hàm thứ nhất có thể được thiết đặt cho $H_N + \Delta_N$ thay cho H_N, và giá trị ban đầu của khóa hàm thứ hai có thể được thiết đặt cho H_N thay cho $H_N + \Delta_N$, như được mô tả ở trên.

Tại bước 706, người dùng thứ nhất có thể yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai (Fig.4, bước 402). Người dùng thứ nhất có thể viện dẫn hàm trên chuỗi khối thứ hai, ví dụ, callAMS(Contract_ID_B, Key), như được mô tả ở trên. Người dùng thứ nhất có thể sử dụng Contract_ID_B để nhận dạng hợp đồng AMS và sử dụng khóa (Key) để chỉ rõ khóa có thể được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai. Nếu giá trị ban đầu của khóa hàm thứ hai được thiết đặt cho $H_N + \Delta_N$, ví dụ, người dùng thứ nhất có thể sử dụng $H_{N-1} + \Delta_{N-1}$ dưới dạng khóa để mở khóa khóa hàm thứ hai. Nếu giá trị ban đầu của khóa hàm thứ hai được thiết đặt cho H_N, người dùng thứ nhất có thể sử dụng H_{N-1} dưới dạng khóa để mở khóa khóa hàm thứ hai. Viện dẫn của yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai có thể được phát hiện bởi người dùng thứ hai.

Tại bước 708, người dùng thứ hai có thể truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được sử dụng bởi người dùng thứ nhất để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai. Tại bước 710, người dùng thứ hai có thể tính toán, dựa trên khóa được truy hồi, khóa mà người dùng thứ hai có thể sử dụng để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất (Fig.4, bước 406). Ví dụ, nếu giá trị ban đầu của khóa hàm thứ nhất được thiết đặt cho H_N và giá trị ban đầu của khóa hàm thứ hai được thiết đặt cho $H_N + \Delta_N$, thì người dùng thứ nhất sử dụng $H_{N-1} + \Delta_{N-1}$ dưới dạng khóa để viện dẫn callAMS(Contract_ID_B, Key) trên chuỗi khối thứ hai, có nghĩa là người dùng thứ hai có thể truy hồi giá trị là $H_{N-1} + \Delta_{N-1}$ và trừ đi Δ_{N-1} từ $H_{N-1} + \Delta_{N-1}$ để tính toán giá trị H_{N-1}. Theo phương thức này, người dùng thứ hai có thể tính toán khóa cần thiết để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất (được thiết đặt cho H_N theo ví dụ này). Tương tự, nếu giá trị ban đầu của khóa hàm thứ nhất được thiết đặt cho $H_N + \Delta_N$ và giá trị ban đầu của khóa hàm thứ hai được thiết đặt cho H_N, thì người dùng thứ nhất sử dụng H_{N-1} dưới dạng khóa để viện dẫn callAMS(Contract_ID_B, Key) trên

chuỗi khối thứ hai, có nghĩa là người dùng thứ hai có thể truy hồi giá trị H_{N-1} và tính toán giá trị $H_{N-1} + \Delta_{N-1}$ theo đó. Theo phương thức này, người dùng thứ hai có thể tính toán khóa cần thiết để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất (được thiết đặt cho $H_N+\Delta_N$ theo ví dụ này).

Tại bước 712, người dùng thứ hai có thể viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất, ví dụ, callAMS(Contract_ID_A, Key), như được mô tả ở trên. Người dùng thứ hai có thể tạo ra khóa được tính toán ở trên cùng với yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất, và dựa trên các mô tả ở trên, khóa này được tính toán bởi người dùng thứ hai sẽ cho phép người dùng thứ hai mở khóa thành công khóa hàm thứ nhất trên chuỗi khối thứ nhất.

Theo một số phương án, các bước 702 đến 712 có thể được lặp lại lên đến N lần (Fig.4, các hoán đổi 2 đến N). Ví dụ, người dùng thứ nhất có thể viện dẫn yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ hai. Người dùng thứ hai có thể phát hiện viện dẫn này và truy hồi, từ yêu cầu thứ hai, khóa thứ hai được sử dụng bởi người dùng thứ nhất để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai. Người dùng thứ hai sau đó có thể tính toán khóa cần thiết cho người dùng thứ hai để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất và viện dẫn yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ nhất.

Theo một số phương án, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất có thể bao gồm ít nhất một thao tác để chuyển tài sản thứ nhất được ghi trên chuỗi khối thứ nhất từ người dùng thứ nhất đến người dùng thứ hai. Ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai có thể bao gồm ít nhất một thao tác để chuyển tài sản thứ hai được ghi trên chuỗi khối thứ hai từ người dùng thứ hai đến người dùng thứ nhất. Theo một số phương án, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất có thể còn bao gồm danh sách các thao tác được xếp thứ tự thứ nhất được thực hiện trên chuỗi khối thứ nhất và ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai có thể còn bao gồm danh sách của các thao tác được xếp thứ tự thứ hai được thực hiện trên chuỗi khối thứ hai.

Fig.8 là sơ đồ khối của thiết bị thực thi hợp đồng AMS 800, theo một phương án. Thiết bị 800 có thể là phương án thực thi của việc xử lý phần mềm, và

có thể tương ứng với phương pháp 700 (Fig.7). Tham chiếu đến Fig.8, thiết bị 800 có thể bao gồm môđun phát hiện 802, môđun ghi 804, môđun truy hồi 806, môđun tính toán 808, và môđun thực thi 810.

Môđun phát hiện 802 có thể phát hiện việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất. Các tham số thứ nhất có thể bao gồm giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi.

Môđun ghi 804 có thể ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai. Các tham số thứ hai có thể bao gồm giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi.

Theo một số phương án, môđun phát hiện 802 cũng có thể phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai. Môđun truy hồi 806 sau đó có thể truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai. Môđun tính toán 808 có thể tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất. Môđun thực thi 810 có thể viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

Mỗi trong số các môđun được mô tả ở trên có thể được lắp đặt dưới dạng phần mềm, hoặc phần cứng, hoặc kết hợp của phần mềm và phần cứng. Ví dụ, mỗi trong số các môđun được mô tả ở trên có thể được lắp đặt sử dụng bộ xử lý thực thi các lệnh được lưu trữ trong bộ nhớ. Ngoài ra, ví dụ, mỗi trong số các môđun được mô tả ở trên có thể được lắp đặt với một hoặc nhiều mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), các bộ xử lý tín hiệu số (digital signal processor - DSP), các thiết bị xử lý tín hiệu số (digital signal processing device - DSPD), các thiết bị logic lập trình được (programmable logic device - PLD), mảng cổng lập trình được dạng trường (field programmable gate array -

FPGA), các bộ điều khiển, các bộ vi điều khiển, các bộ vi xử lý, hoặc các linh kiện điện tử khác, để thực hiện các phương pháp được mô tả. Theo ví dụ khác, mỗi trong số các môđun được mô tả ở trên có thể được lắp đặt bằng cách sử dụng chip máy tính hoặc thực thể, hoặc được lắp đặt bằng cách sử dụng sản phẩm có chức năng nhất định. Theo một phương án, các thiết bị 600 và 800 có thể là máy tính, và máy tính này có thể là máy tính cá nhân, máy tính xách tay, điện thoại di động, điện thoại camera, điện thoại thông minh, thiết bị hỗ trợ kỹ thuật số cá nhân, máy phát phương tiện, thiết bị định vị, thiết bị gửi và nhận email, bàn giao tiếp trò chơi, máy tính bảng, thiết bị đeo vào được, hoặc kết hợp bất kỳ của các thiết bị này.

Đối với quy trình thực hiện của các chức năng và các vai trò của mỗi môđun trong các thiết bị 600 và 800, các tham chiếu có thể được tạo ra với các bước tương ứng trong các phương pháp được mô tả ở trên. Các chi tiết được lược bỏ trong bản mô tả để làm đơn giản.

Theo một số phương án, sản phẩm chương trình máy tính có thể bao gồm vật ghi lưu trữ đọc được bằng máy tính không tạm thời có các lệnh chương trình đọc được bằng máy tính trên đó làm cho bộ xử lý tiến hành các phương pháp được mô tả ở trên.

Vật ghi lưu trữ đọc được bằng máy tính có thể là thiết bị hữu hình có thể lưu trữ các lệnh để sử dụng bởi thiết bị thực thi lệnh. Vật ghi lưu trữ đọc được bằng máy tính này có thể là, ví dụ, nhưng không bị giới hạn với, thiết bị lưu trữ điện tử, thiết bị lưu trữ từ tính, thiết bị lưu trữ quang học, thiết bị lưu trữ điện từ, thiết bị lưu trữ bán dẫn, hoặc bất kỳ kết hợp thích hợp nào của các thiết bị đã nêu. Danh sách không toàn diện của các ví dụ cụ thể hơn của vật ghi lưu trữ đọc được bằng máy tính bao gồm như sau: đĩa máy tính di động, đĩa cứng, bộ nhớ truy cập ngẫu nhiên (random access memory - RAM), bộ nhớ chỉ đọc (read-only memory - ROM), bộ nhớ chỉ đọc lập trình được xóa được (erasable programmable read-only memory - EPROM), bộ nhớ truy cập ngẫu nhiên tĩnh (static random access memory - SRAM), bộ nhớ chỉ đọc bằng đĩa nén di động (portable compact disc read-only memory - CD-ROM), đĩa đa năng số (digital versatile disk - DVD), thẻ nhớ, đĩa mềm, thiết bị được mã hóa cơ học như thẻ đục lỗ hoặc các cấu trúc nâng lên trong rãnh có các

lệnh được ghi trên đó, và bất kỳ kết hợp thích hợp nào của các thiết bị nêu trên.

Các lệnh chương trình đọc được bằng máy tính để tiến hành các phương pháp được mô tả ở trên có thể là các lệnh hợp ngữ, các lệnh kiến trúc tập lệnh (instruction-set-architecture - ISA), các lệnh của máy, các lệnh phụ thuộc vào máy, vi mã, các lệnh phần sụn, dữ liệu trạng thái thiết lập, hoặc mã nguồn hoặc mã đối tượng được viết trong kết hợp bất kỳ của một hoặc nhiều ngôn ngữ lập trình, bao gồm ngôn ngữ lập trình hướng đối tượng, và các ngôn ngữ lập trình thủ tục thông thường. Các lệnh chương trình đọc được bằng máy tính có thể thực thi hoàn toàn trên thiết bị điện toán dưới dạng gói phần mềm độc lập, hoặc một phần trên thiết bị điện toán thứ nhất và một phần trên thiết bị điện toán thứ hai ở cách xa với thiết bị điện toán thứ nhất. Trong trường hợp sau, thiết bị điện toán từ xa thứ hai có thể được kết nối với thiết bị điện toán thứ nhất thông qua loại mạng bất kỳ, bao gồm mạng cục bộ (local area network - LAN) hoặc mạng toàn cục (wide area network - WAN).

Các lệnh chương trình đọc được bằng máy tính có thể được đưa vào bộ xử lý của máy tính đa năng hoặc chuyên dụng, hoặc thiết bị xử lý dữ liệu lập trình được khác để chế tạo máy, sao cho các lệnh, thực thi thông qua bộ xử lý của máy tính hoặc thiết bị xử lý dữ liệu lập trình được khác, tạo ra các phương tiện để thực hiện các phương pháp được mô tả ở trên.

Các lưu đồ và các sơ đồ trên các hình vẽ minh họa cấu trúc, chức năng, và hoạt động của các phương án thực hiện khả dĩ của các thiết bị, các phương pháp, và các sản phẩm chương trình máy tính theo các phương án khác nhau của sáng chế. Về vấn đề này, khối trong các lưu đồ hoặc các sơ đồ có thể biểu diễn chương trình phần mềm, đoạn, hoặc một phần của mã, bao gồm một hoặc nhiều lệnh chạy được để thực thi các chức năng cụ thể. Cũng lưu ý rằng, theo một số phương án thực hiện thay thế, các chức năng được lưu ý trong các khối có thể xảy ra khác với thứ tự được lưu ý trên các hình vẽ. Ví dụ, hai khối được thể hiện theo trình tự có thể, thực tế, được thực thi gần như đồng thời, hoặc đôi khi các khối có thể được thực thi theo thứ tự ngược lại, phụ thuộc vào chức năng được viện dẫn. Cũng được lưu ý rằng mỗi khối của các sơ đồ và/hoặc các lưu đồ, và các kết hợp của các khối trong các sơ

đồ và các lưu đồ, có thể được lắp đặt trong các hệ thống dựa vào phần cứng chuyên dụng để thực hiện các chức năng hoặc thao tác được chỉ định, hoặc các kết hợp của phần cứng chuyên dụng và các lệnh máy tính.

Cần hiểu rằng các dấu hiệu nhất định của bản mô tả, để làm rõ, được mô tả trong ngữ cảnh của các phương án riêng biệt, có thể cũng được đưa vào trong kết hợp của một phương án. Ngược lại, các dấu hiệu khác nhau của bản mô tả, để ngắn gọn, được mô tả trong ngữ cảnh của một phương án, có thể cũng được tạo ra tách biệt hoặc theo kết hợp phụ thích hợp bất kỳ hoặc thích hợp theo phương án được mô tả bất kỳ khác nào của bản mô tả. Các dấu hiệu nhất định được mô tả trong ngữ cảnh của các phương án khác nhau không phải là các dấu hiệu cần thiết của các phương án đó, trừ khi được lưu ý như vậy.

Mặc dù bản mô tả được mô tả kết hợp với các phương án cụ thể, người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu được các thay thế, sửa đổi và cải biến. Theo đó, các điểm yêu cầu bảo hộ sau đây bao quát tất cả các thay thế, sửa đổi và cải biến này nằm trong phạm vi yêu cầu bảo hộ.

YÊU CẦU BẢO HỘ

1. Phương pháp được thực hiện bằng máy tính để thực thi hợp đồng đa hoán đổi ẩn danh (anonymous multi-swap - AMS) giữa người dùng thứ nhất và người dùng thứ hai, phương pháp này bao gồm các bước:

phát hiện việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, các tham số thứ nhất bao gồm: giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, số lần tối đa hợp đồng AMS có thể được thực thi lớn hơn một;

ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, các tham số thứ hai bao gồm: giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất;

phát hiện sự viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai;

truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai;

tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và

viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

2. Phương pháp được thực hiện bằng máy tính để thực thi hợp đồng đa hoán đổi ẩn danh (AMS) giữa người dùng thứ nhất và người dùng thứ hai, phương pháp này bao gồm các bước:

ghi, bởi người dùng thứ nhất, các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất, các tham số thứ nhất bao gồm: giá trị ban đầu của khóa hàm thứ nhất trên chuỗi khối thứ nhất, ít nhất một thao tác được thực hiện trên chuỗi

khối thứ nhất khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, số lần tối đa hợp đồng AMS có thể được thực thi lớn hơn một;

ghi, bởi người dùng thứ hai, các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai, các tham số thứ hai bao gồm: giá trị ban đầu của khóa hàm thứ hai trên chuỗi khối thứ hai, ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai khi hợp đồng AMS được thực thi, và số lần tối đa hợp đồng AMS có thể được thực thi, giá trị ban đầu của khóa hàm thứ hai khác với giá trị ban đầu của khóa hàm thứ nhất;

viện dẫn, bởi người dùng thứ nhất, về yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai;

truy hồi, bởi người dùng thứ hai, khóa được sử dụng bởi người dùng thứ nhất để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai;

tính toán, bởi người dùng thứ hai, khóa được sử dụng bởi người dùng thứ hai để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất; và

viện dẫn, bởi người dùng thứ hai, yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán.

3. Phương pháp theo điểm bất kỳ trong số các điểm nêu trên, trong đó ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất bao gồm ít nhất một thao tác để chuyển tài sản thứ nhất được ghi trên chuỗi khối thứ nhất từ người dùng thứ nhất đến người dùng thứ hai, và ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai bao gồm ít nhất một thao tác để chuyển tài sản thứ hai được ghi trên chuỗi khối thứ hai từ người dùng thứ hai đến người dùng thứ nhất.

4. Phương pháp theo điểm bất kỳ trong số các điểm nêu trên, trong đó ít nhất một thao tác được thực hiện trên chuỗi khối thứ nhất còn bao gồm danh sách các thao tác được xếp thứ tự thứ nhất được thực hiện trên chuỗi khối thứ nhất và ít nhất một thao tác được thực hiện trên chuỗi khối thứ hai còn bao gồm danh sách các thao tác được xếp thứ tự thứ hai được thực hiện trên chuỗi khối thứ hai.

5. Phương pháp theo điểm bất kỳ trong số các điểm nêu trên, trong đó giá trị ban đầu của khóa hàm thứ nhất, H_N, được tính toán dựa trên ứng dụng kế tiếp thứ nhất

của hàm một chiều $g()$ đến giá trị được chọn thứ nhất, H_0, ứng dụng kế tiếp thứ nhất được áp dụng N lần để tính toán $H_1 = g(H_0)$, $H_2 = g(H_1)$,... $H_N = g(H_{N-1})$, trong đó N là số lần tối đa hợp đồng AMS có thể được thực thi.

6. Phương pháp theo điểm 5, trong đó giá trị ban đầu của khóa hàm thứ hai, $H_N + \Delta_N$, được tính toán dựa trên H_N và ứng dụng kế tiếp thứ hai của hàm một chiều $g()$ cho giá trị được chọn thứ hai, Δ_0, ứng dụng kế tiếp thứ hai được áp dụng N lần để tính toán $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$.

7. Phương pháp theo điểm 6, phương pháp này còn bao gồm các bước:

nhận giá trị được chọn thứ hai, Δ_0;

áp dụng liên tiếp hàm một chiều $g()$ cho giá trị được chọn thứ hai, Δ_0, để tính toán độc lập $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$.

8. Phương pháp theo điểm 7, trong đó khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai có giá trị là $H_{N-1} + \Delta_{N-1}$, và trong đó bước tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất còn bao gồm:

tính toán khóa được tính toán bằng cách trừ đi giá trị được tính toán độc lập Δ_{N-1} từ giá trị khóa được truy hồi là $H_{N-1} + \Delta_{N-1}$.

9. Phương pháp theo điểm 8, phương pháp này còn bao gồm các bước:

phát hiện viện dẫn của yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ hai;

truy hồi, từ yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa thứ hai được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai, khóa thứ hai được truy hồi có giá trị là $H_{N-2} + \Delta_{N-2}$;

tính toán khóa thứ hai được tính toán bằng cách trừ đi giá trị được tính toán độc lập Δ_{N-2} từ giá trị của khóa thứ hai được truy hồi là $H_{N-2} + \Delta_{N-2}$; và

viện dẫn yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa thứ hai được tính toán.

10. Phương pháp theo điểm bất kỳ trong các điểm từ 1 đến 4, trong đó giá trị ban đầu của khóa hàm thứ hai, H_N, được tính toán dựa trên ứng dụng kế tiếp thứ nhất của hàm một chiều $g()$ cho giá trị được chọn thứ nhất, H_0, ứng dụng kế tiếp thứ nhất được áp dụng N lần để tính toán $H_1 = g(H_0)$, $H_2 = g(H_1)$,... $H_N = g(H_{N-1})$, trong đó N là số lần tối đa hợp đồng AMS có thể được thực thi.

11. Phương pháp theo điểm 10, trong đó giá trị ban đầu của khóa hàm thứ nhất, $H_N + \Delta_N$, được tính toán dựa trên H_N và ứng dụng kế tiếp thứ hai của hàm một chiều $g()$ cho giá trị được chọn thứ hai, Δ_0, ứng dụng kế tiếp thứ hai được áp dụng N lần để tính toán $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$.

12. Phương pháp theo điểm 11, phương pháp này còn bao gồm các bước:

nhận giá trị được chọn thứ hai, Δ_0;

áp dụng liên tiếp hàm một chiều $g()$ cho giá trị được chọn thứ hai, Δ_0, để tính toán độc lập $\Delta_1 = g(\Delta_0)$, $\Delta_2 = g(\Delta_1)$,... $\Delta_N = g(\Delta_{N-1})$.

13. Phương pháp theo điểm 12, trong đó khóa được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai có giá trị là H_{N-1}, và trong đó bước tính toán, dựa trên khóa được truy hồi, khóa được tính toán để mở khóa khóa hàm thứ nhất trên chuỗi khối thứ nhất còn bao gồm:

tính toán khóa được tính toán bằng cách thêm giá trị được tính toán độc lập là Δ_{N-1} cho giá trị khóa được truy hồi là H_{N-1}.

14. Phương pháp theo điểm 13, phương pháp này còn bao gồm các bước:

phát hiện viện dẫn của yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ hai;

truy hồi, từ yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa thứ hai được truy hồi được sử dụng để mở khóa khóa hàm thứ hai trên chuỗi khối thứ hai, khóa thứ hai được truy hồi có giá trị là H_{N-2};

tính toán khóa thứ hai được tính toán bằng cách thêm giá trị được tính toán độc lập là Δ_{N-2} vào giá trị khóa thứ hai được truy hồi là H_{N-2}; và

viện dẫn yêu cầu thứ hai để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa thứ hai được tính toán.

15. Thiết bị thực thi hợp đồng đa hoán đổi ẩn danh (AMS), bao gồm:

một hoặc nhiều bộ xử lý; và

một hoặc nhiều bộ nhớ đọc được bằng máy tính được ghép nối với một hoặc nhiều bộ xử lý và có các lệnh được lưu trữ trên đó được chạy bởi một hoặc nhiều bộ xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 14.

16. Thiết bị thực thi hợp đồng đa hoán đổi ẩn danh (AMS), thiết bị này bao gồm nhiều môđun để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 14.

17. Vật ghi đọc được bằng máy tính không tạm thời có các lệnh được lưu trữ trong đó, khi được thực thi bởi bộ xử lý của thiết bị, làm cho thiết bị này thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 14.

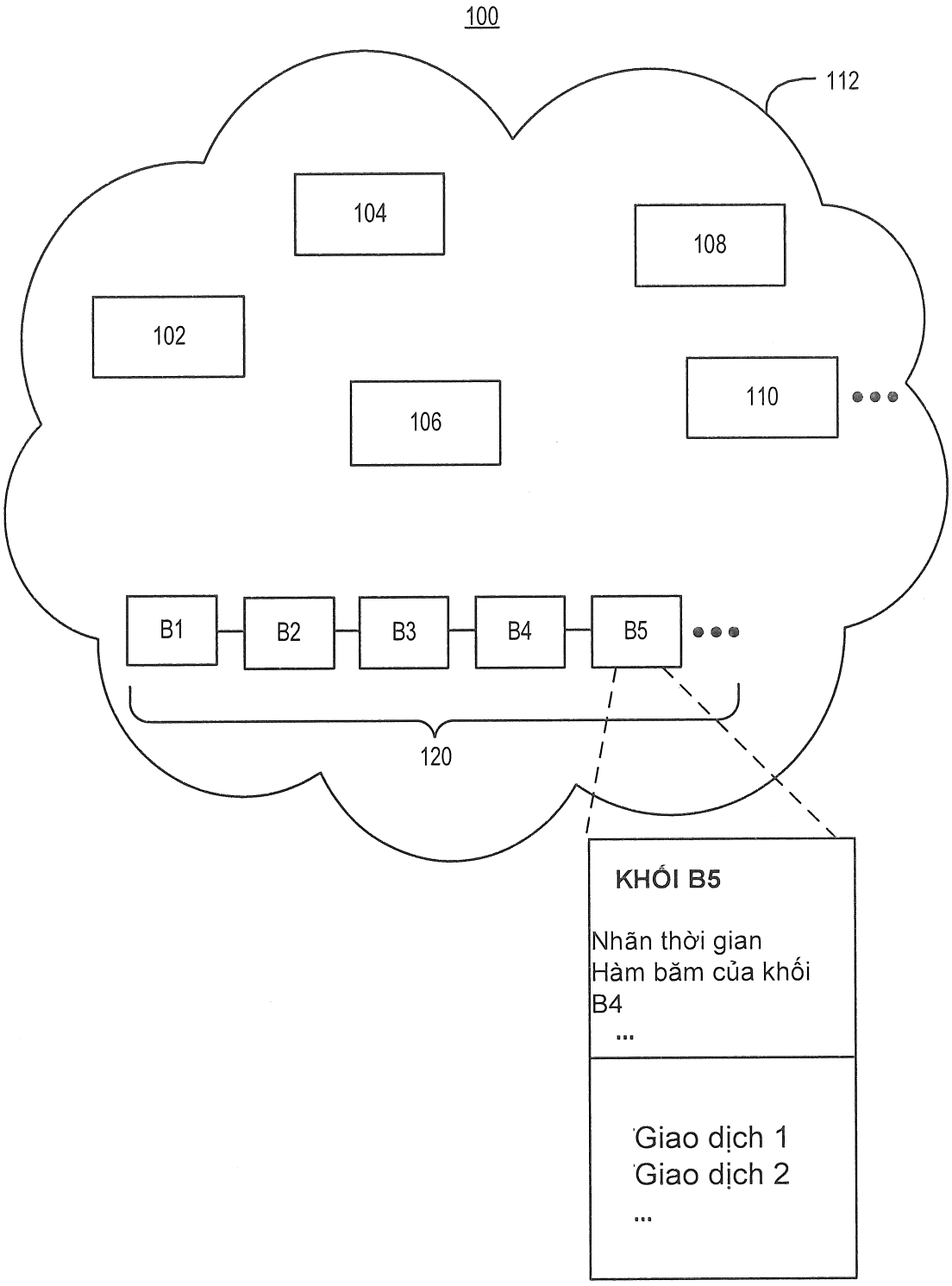

100
112
104
108
102
106
110
B1
B2
B3
B4
B5
120
KHỐI B5
Nhãn thời gian
Hàm băm của khối B4
...
Giao dịch 1
Giao dịch 2
...

FIG. 1

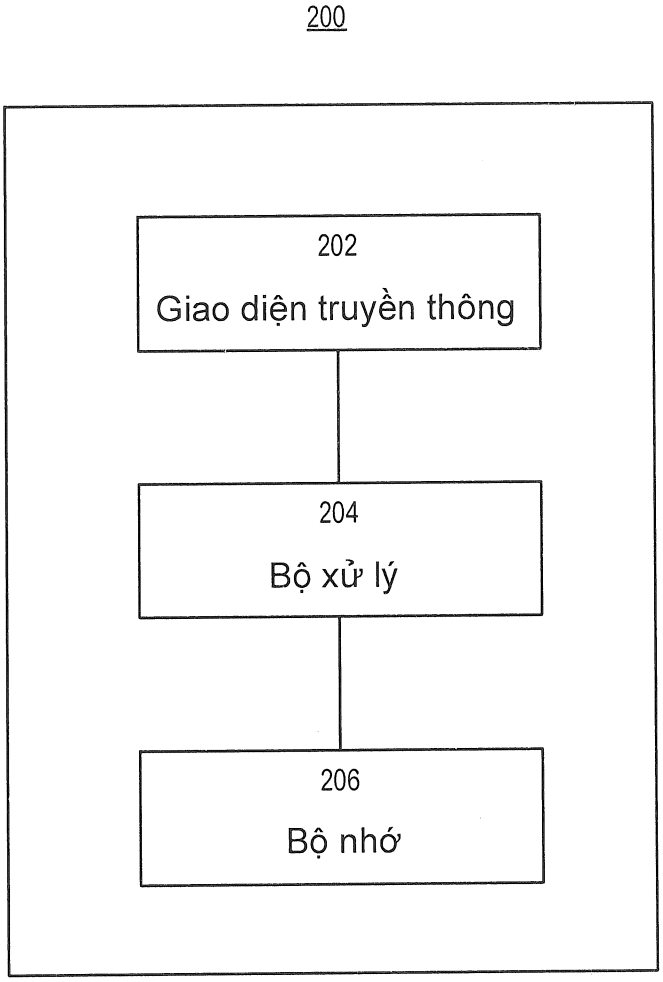


FIG. 2

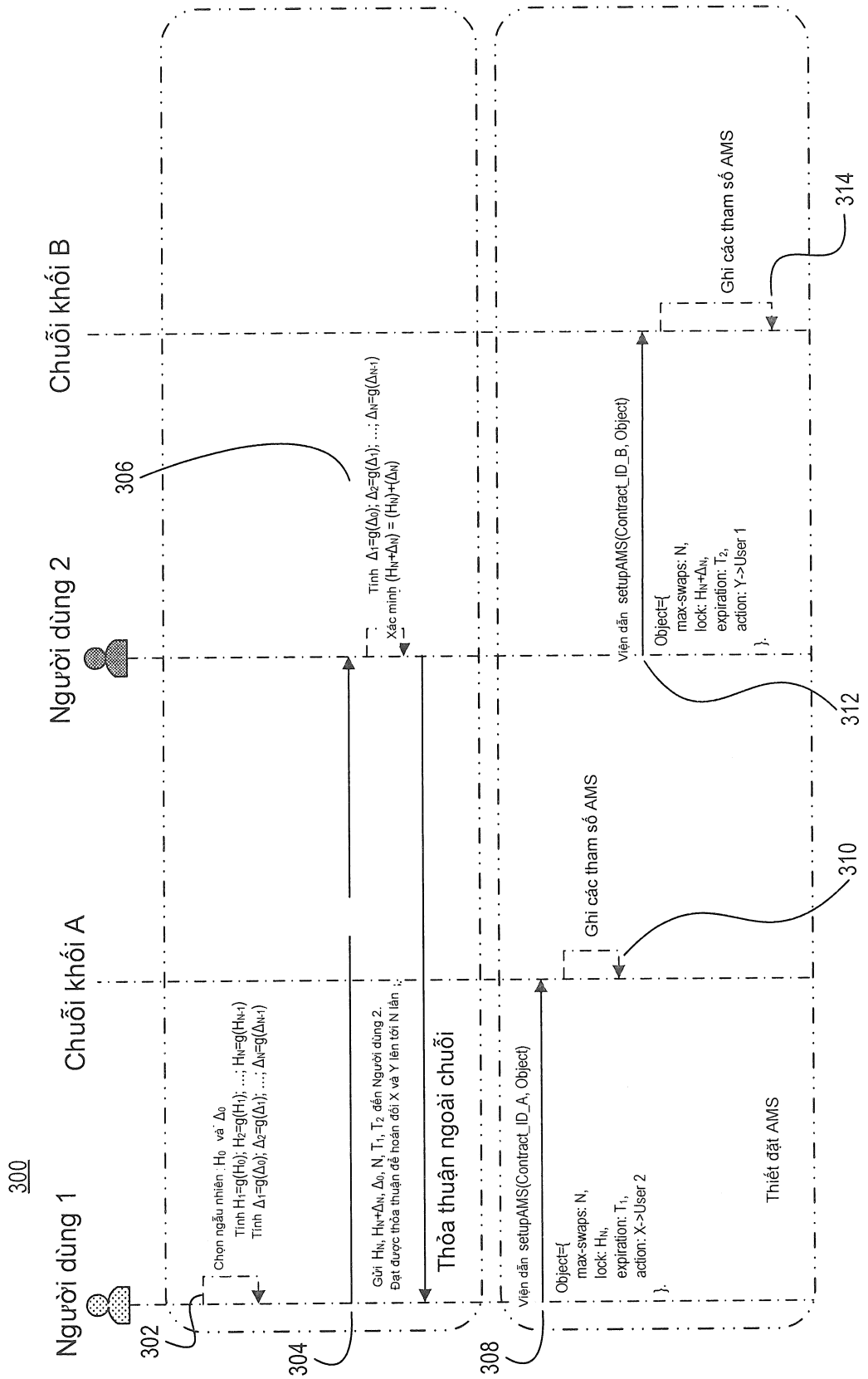

300
Người dùng 1
Chuỗi khối A
Người dùng 2
Chuỗi khối B
302
Chọn ngẫu nhiên : H_0 và Δ_0
Tính $H_1=g(H_0)$; $H_2=g(H_1)$; ...; $H_N=g(H_{N-1})$
Tính $\Delta_1=g(\Delta_0)$; $\Delta_2=g(\Delta_1)$; ...; $\Delta_N=g(\Delta_{N-1})$
304
Gửi H_N, $H_N+\Delta_N$, Δ_0, N, T_1, T_2 đến Người dùng 2.
Đạt được thỏa thuận để hoán đổi X và Y lên tới N lần.
306
Tính $\Delta_1=g(\Delta_0)$; $\Delta_2=g(\Delta_1)$; ...; $\Delta_N=g(\Delta_{N-1})$
Xác minh $(H_N+\Delta_N) = (H_N)+(\Delta_N)$
Thỏa thuận ngoài chuỗi
308
Viện dẫn setupAMS(Contract_ID_A, Object)
Object={
max-swaps: N,
lock: H_N,
expiration: T_1,
action: X->User 2
},
Ghi các tham số AMS
310
Thiết đặt AMS
Viện dẫn setupAMS(Contract_ID_B, Object)
Object={
max-swaps: N,
lock: $H_N+\Delta_N$,
expiration: T_2,
action: Y->User 1
},
312
Ghi các tham số AMS
314

FIG. 3

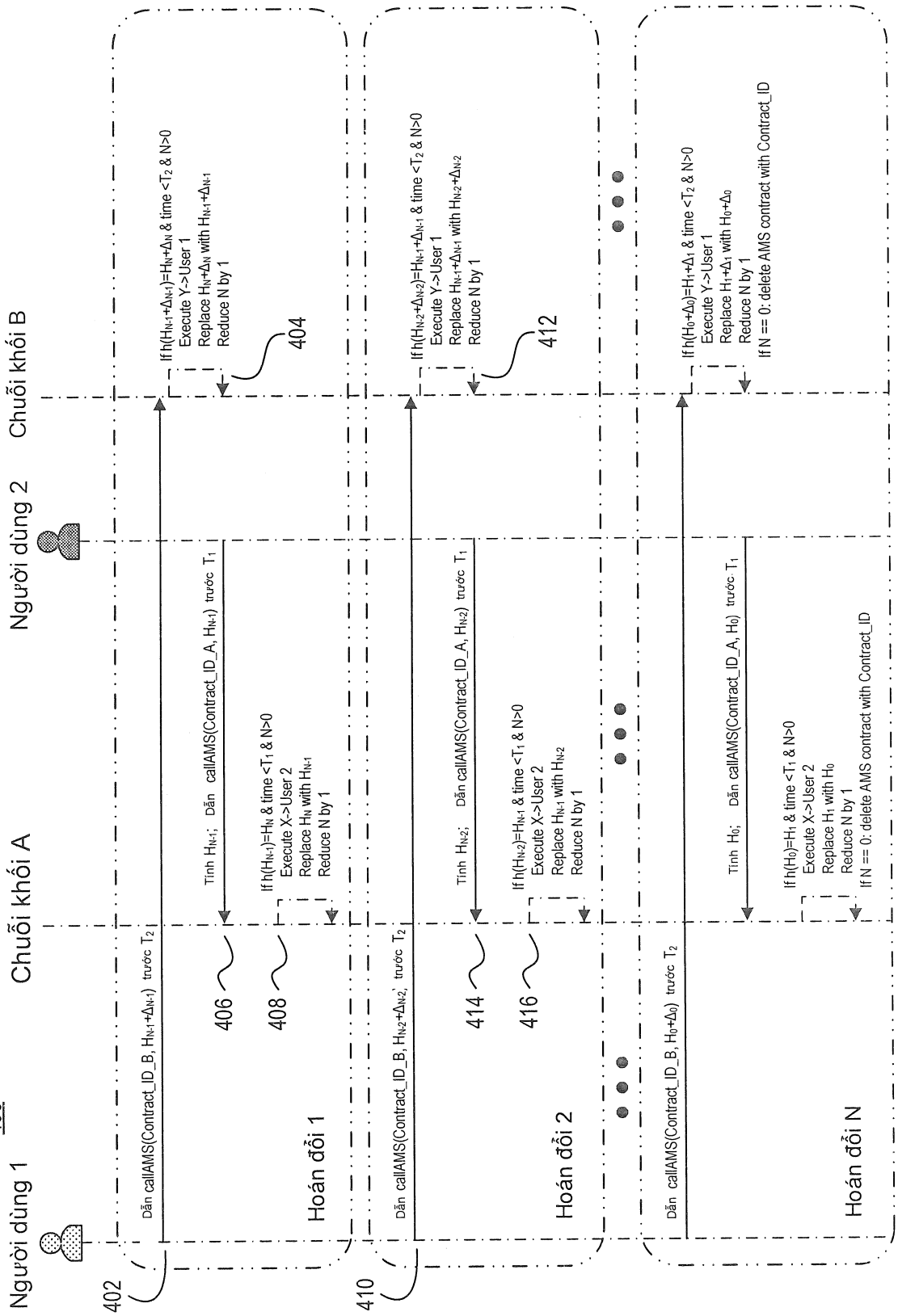

400
Người dùng 1
Chuỗi khối A
Người dùng 2
Chuỗi khối B
402
Dẫn callAMS(Contract_ID_B, $H_{N-1}+\Delta_{N-1}$) trước T_2
Tính H_{N-1}; Dẫn callAMS(Contract_ID_A, H_{N-1}) trước T_1
If $h(H_{N-1}+\Delta_{N-1})=H_N+\Delta_N$ & time <T_2 & N>0
Execute Y->User 1
Replace $H_N+\Delta_N$ with $H_{N-1}+\Delta_{N-1}$
Reduce N by 1
404
406
If $h(H_{N-1})=H_N$ & time <T_1 & N>0
Execute X->User 2
Replace H_N with H_{N-1}
Reduce N by 1
408
Hoán đổi 1
410
Dẫn callAMS(Contract_ID_B, $H_{N-2}+\Delta_{N-2}$, trước T_2
Tính H_{N-2}; Dẫn callAMS(Contract_ID_A, H_{N-2}) trước T_1
If $h(H_{N-2}+\Delta_{N-2})=H_{N-1}+\Delta_{N-1}$ & time <T_2 & N>0
Execute Y->User 1
Replace $H_{N-1}+\Delta_{N-1}$ with $H_{N-2}+\Delta_{N-2}$
Reduce N by 1
412
414
If $h(H_{N-2})=H_{N-1}$ & time <T_1 & N>0
Execute X->User 2
Replace H_{N-1} with H_{N-2}
Reduce N by 1
416
Hoán đổi 2
Dẫn callAMS(Contract_ID_B, $H_0+\Delta_0$) trước T_2
Tính H_0; Dẫn callAMS(Contract_ID_A, H_0) trước T_1
If $h(H_0+\Delta_0)=H_1+\Delta_1$ & time <T_2 & N>0
Execute Y->User 1
Replace $H_1+\Delta_1$ with $H_0+\Delta_0$
Reduce N by 1
If N == 0: delete AMS contract with Contract_ID
If $h(H_0)=H_1$ & time <T_1 & N>0
Execute X->User 2
Replace H_1 with H_0
Reduce N by 1
If N == 0: delete AMS contract with Contract_ID
Hoán đổi N

FIG. 4

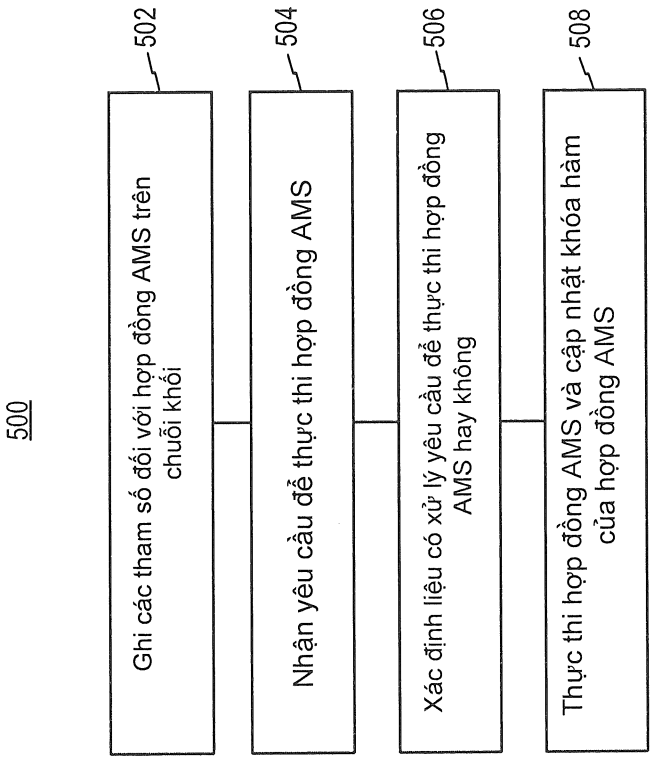

500
502
Ghi các tham số đối với hợp đồng AMS trên chuỗi khối
504
Nhận yêu cầu để thực thi hợp đồng AMS
506
Xác định liệu có xử lý yêu cầu để thực thi hợp đồng AMS hay không
508
Thực thi hợp đồng AMS và cập nhật khóa hàm của hợp đồng AMS

FIG. 5

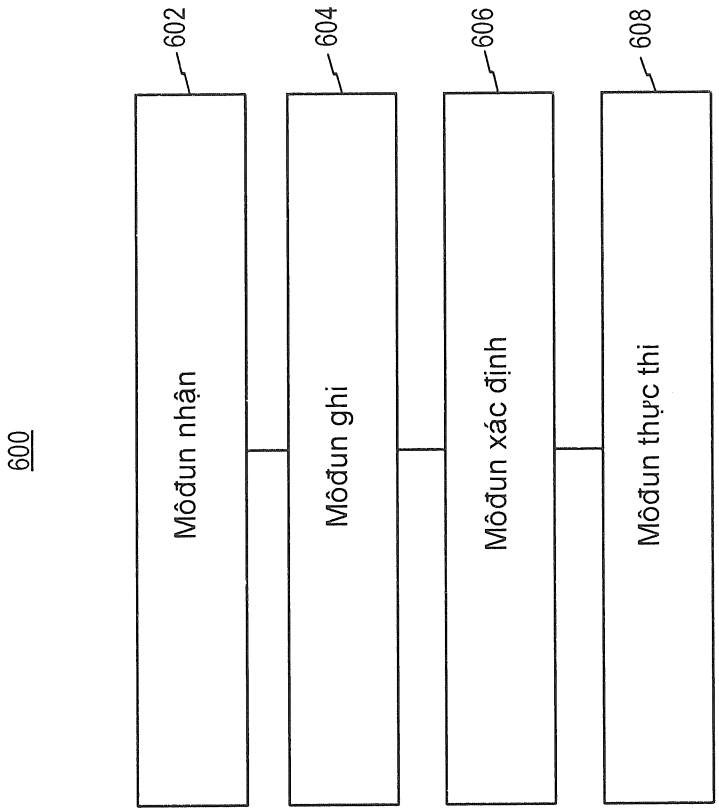

600
602
Môđun nhận
604
Môđun ghi
606
Môđun xác định
608
Môđun thực thi

FIG. 6

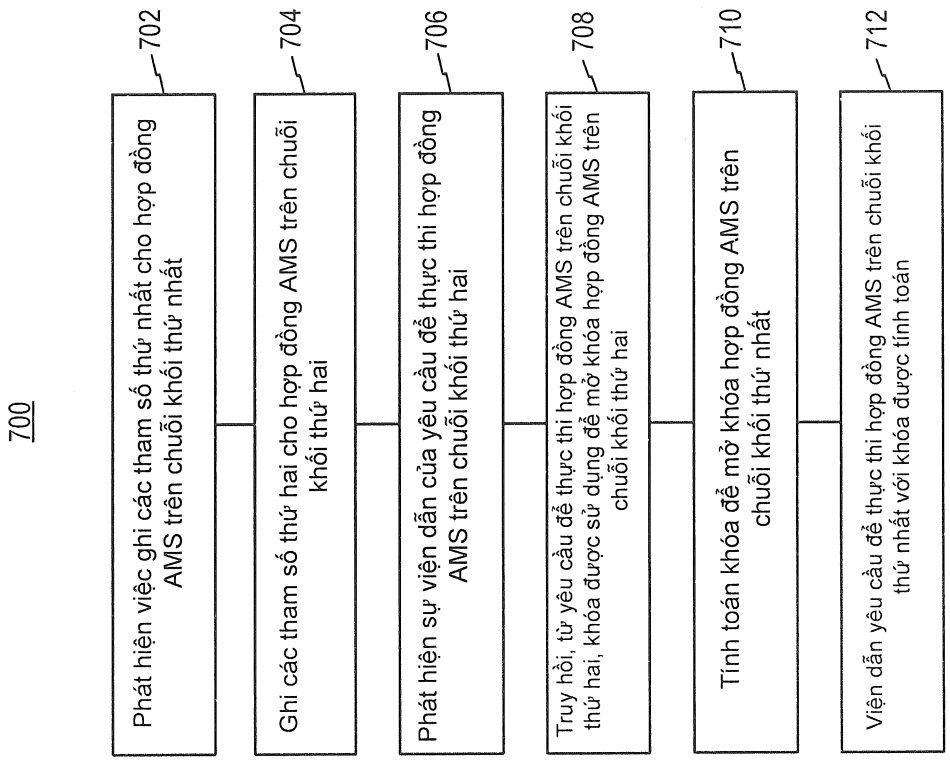

700
Phát hiện việc ghi các tham số thứ nhất cho hợp đồng AMS trên chuỗi khối thứ nhất
702
Ghi các tham số thứ hai cho hợp đồng AMS trên chuỗi khối thứ hai
704
Phát hiện sự viện dẫn của yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai
706
Truy hồi, từ yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ hai, khóa được sử dụng để mở khóa hợp đồng AMS trên chuỗi khối thứ hai
708
Tính toán khóa để mở khóa hợp đồng AMS trên chuỗi khối thứ nhất
710
Viện dẫn yêu cầu để thực thi hợp đồng AMS trên chuỗi khối thứ nhất với khóa được tính toán
712

FIG. 7

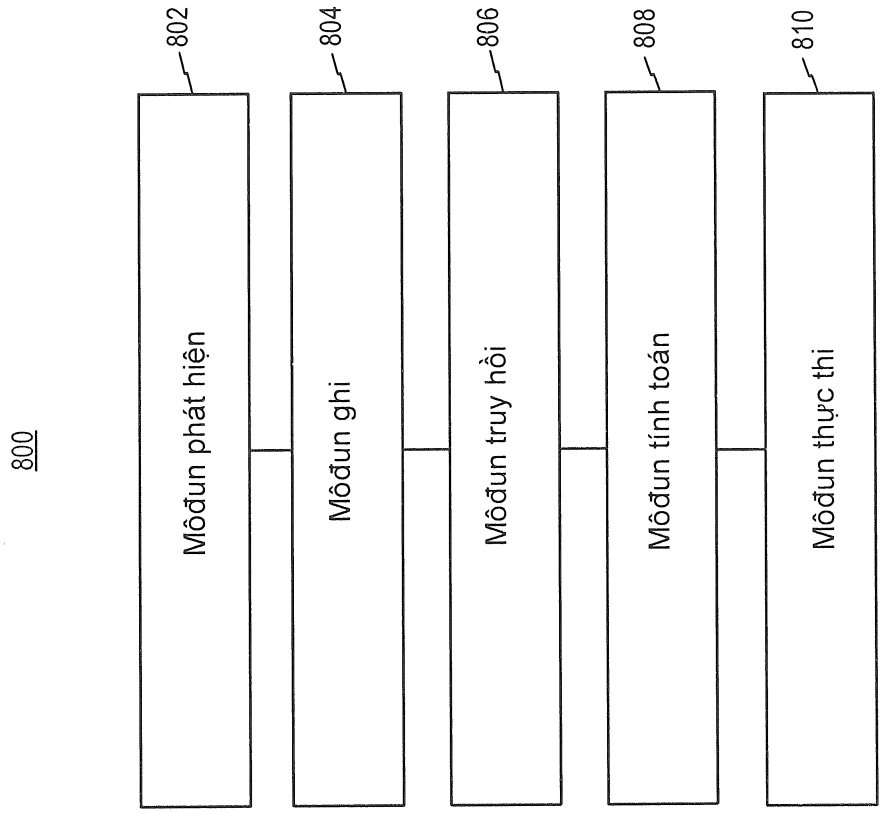

800
802
Môđun phát hiện
804
Môđun ghi
806
Môđun truy hồi
808
Môđun tính toán
810
Môđun thực thi

FIG. 8