Interhead Hydra
Hai Head tốt hơn một

Maxim Jourenko[1,2], Mario Larangeira[1,2] và Keisuke Tanaka[1]

[1]Khoa Khoa học Toán học và Máy tính, trường máy tính,
Viện Công nghệ Tokyo.
Tokyo-to Meguro-ku Oookayama 2-12-1 W8-55, Nhật Bản.
{jourenko.m.ab@m, mario@c, keisuke@is}.titech.ac.jp
[2]Input Output Hồng Kông.
{maxim.jourenko, mario.larangeira} @iohk.io
http://iohk.io

Tóm lược. Sổ cái phân tán được duy trì thông qua các giao thức đồng thuận được thực hiện bởi các bên không cần tin tưởng lẫn nhau. Tuy nhiên, các giao thức đồng thuận này có những hạn chế cố hữu, do đó dẫn đến các vấn đề về khả năng mở rộng của sổ cái. Các giao thức Layer 2 hoạt động trên *các kênh* và cho phép các bên tương tác với bên khác mà không cần thông qua giao thức đồng thuận mặc dù dựa vào tính bảo mật của giao thức này như dự phòng. Các giao thức Layer 2 nổi bật là các kênh thanh toán cho Bitcoin cho phép hai bên trao đổi tiền, kênh trạng thái cho Ethereum cho phép hai bên thực thi một máy trạng thái, và các Hydra *Head* [FC'21] cho Cardano cho phép nhiều bên thực thi Máy phát ra ràng buộc (CEM - Constraint Emitting Machine). Các kênh có thể được nối vào các mạng bằng cách sử dụng các kỹ thuật như Hash hợp đồng khóa thời gian để thực hiện thanh toán hoặc các kênh trạng thái ảo như được giới thiệu bởi Dziembowski và cộng sự [CCS'18] để thực thi các máy trạng thái. Các cấu trúc này cho phép tương tác giữa hai bên trên mạng lưới kênh, tức là hai điểm cuối của đường dẫn kênh. Điều này được thực hiện bằng cách sử dụng *các bên trung gian*, là các bên trên đường dẫn kênh ở giữa cả hai điểm cuối, những người phải trả tài sản thế chấp để đảm bảo an toàn cho công trình. Mặc dù các phương pháp này có thể được sử dụng với Hydra, nhưng chúng không thể được mở rộng nhỏ để cho phép thực thi CEM giữa một số lượng tùy ý các bên trên các Hydra Head khác nhau. Công việc này giải quyết khoảng cách này bằng cách giới thiệu cấu trúc Interhead cho phép tạo lặp đi lặp lại các Hydra Head ảo. Vì lợi ích độc lập, công trình của chúng tôi là công trình đầu tiên (1) hỗ trợ các kênh với số lượng bên tùy ý và (2) cho phép thanh toán tài sản thế chấp bởi nhiều trung gian, cho phép chia sẻ gánh nặng này nên sẽ cải thiện tính thực tế.

Từ khóa: Blockchain, Kênh trạng thái, Mạng lưới kênh

1 Giới thiệu

Sổ cái phi tập trung lần đầu tiên được giới thiệu bởi Nakamoto [20] với công nghệ Blockchain. Kể từ đó, các loại tiền mã hoá dựa trên thiết kế này có mức độ phổ biến ngày càng tăng đều đặn. Tuy nhiên, trong khi sự thích ứng rộng rãi hơn của sổ cái phi tập trung cho thấy mức độ phù hợp của công nghệ này, thì các triển khai hiện tại lại gặp khó khăn trong việc mở rộng quy mô để đáp ứng nhu cầu ngày càng tăng. Các giao dịch, ví dụ như thanh toán, trên sổ cái phi tập trung yêu cầu nó phải được xử lý thông qua cơ chế đồng thuận, được phân loại là giao thức Layer 1. Thông lượng giao dịch tiềm năng của một sổ cái bị giới hạn bởi giao thức đồng thuận của nó và việc tăng nó là rất không cần thiết [5]. Người phát hành giao dịch có thể trả phí để tăng mức độ ưu tiên mà theo đó giao dịch của họ được xử lý, bỏ qua dòng. Điều này dẫn đến việc tạo ra một thị trường xử lý giao dịch yêu cầu thanh toán một khoản phí tương ứng với nhu cầu xử lý giao dịch. Chẳng hạn, vào ngày 20 tháng 4 năm 2021, chi phí trung bình để xử lý một giao dịch bằng Bitcoin đạt đỉnh hơn 60 USD[1].

Các giao thức Layer 2 là một phân loại các kỹ thuật nhằm mục đích giảm số lượng giao dịch phải được phát hành trên sổ cái bằng một lớp gián tiếp. Các giải pháp Layer 2 bao gồm Sidechain [10,24], Plasma của Ethereum [23], kênh thanh toán [22,21,6] và nói chung hơn là các kênh trạng thái [7,8] đáng chú ý là Hydra Head [4]. Công việc này tập trung vào các kênh trạng thái. Các kênh thanh toán được thiết lập bởi hai bên bằng cách sử dụng giao dịch khóa tiền của họ vào ví dùng chung. Kênh lưu trữ một trạng thái, đó là cách các khoản tiền bị khóa bên trong nó được phân phối giữa cả hai bên. Sau đó, các bên có thể thực hiện một giao thức *Off-Chain* thay đổi trạng thái này. Cuối cùng, các bên phát hành một giao dịch để mở khóa tiền từ kênh tương ứng với trạng thái mới nhất của nó. Lưu ý rằng giao dịch sau được tạo trước để tránh việc tiền bị khóa trong kênh vô thời hạn nếu một trong các bên không phản hồi. Bằng cách này, các bên có thể thực hiện $O(n)$, $n \in \mathbb{N}$, thanh toán lẫn nhau trong khi chỉ phát hành giao dịch $O(1)$ trên sổ cái, do đó cải thiện khả năng mở rộng của hệ thống. Kênh trạng thái mở rộng khái niệm này bằng cách cho phép lưu trữ trạng thái tùy ý, với sự kết hợp của một cơ chế biểu đạt tập lệnh đủ tốt, có thể cho phép hai bên thực thi các máy trạng thái Off-Chain. Hydra khái quát hóa khái niệm kênh hơn nữa bằng cách cho phép một số lượng tùy ý các bên tham gia vào một kênh. Các bên này có thể thực thi các máy trạng thái, được mô hình hóa dưới dạng Máy phát ra ràng buộc (CEM) [2], Off-Chain. Các giao thức Off-Chain khác như Hash hợp đồng khóa thời gian (HTLCs -Hash Timelocked Contracts) [22] và các kênh ảo [8,12,13] cho phép các kênh liền kề, tức là nhiều kênh với một bên chung, được nối vào mạng lưới kênh cho phép các bên tương tác với nhau mà không cần nhu cầu mở kênh mới, sử dụng lại hạ tầng kênh hiện có.

[1]https://ycharts.com/indicators/bitcoin phí giao dịch trung bình

Trong khi các giao thức để thực hiện thanh toán qua mạng lưới kênh thanh toán hoặc thực thi các máy trạng thái trên mạng lưới kênh trạng thái tồn tại, không có giao thức tương tự cho Hydra. Mặc dù các HTLC có thể được tái sử dụng để thực hiện thanh toán Off-Chain trên các Hydra Head, nhưng không có cách nào để thực thi các CEM Off-Chain tùy ý trên nhiều Hydra Head chồng chéo một phần. Hơn nữa, cách tiếp cận hiện có để kết nối các kênh trạng thái bằng cách sử dụng cấu trúc kênh ảo của Dziembowski và cộng sự [8] dường như không dễ dàng mở rộng cho các cấu trúc kênh cho nhiều hơn hai bên như trong Hydra, vì nó yêu cầu xác định duy nhất các bên độc hại để sau đó trừng phạt họ. Thứ nhất, không rõ liệu chúng ta có thể xác định duy nhất hành vi sai trái trong trường hợp kênh có nhiều hơn hai bên hay không. Mặt khác, các cơ chế trừng phạt trong các cấu trúc kênh ảo [7,8,12] sử dụng khả năng quy lỗi duy nhất để yêu cầu các bên vi phạm tất cả Coin hoặc tài sản thế chấp của họ trong kênh ảo cho các bên trung thực. Tuy nhiên, cơ chế trừng phạt sẽ như thế nào đối với các kênh có nhiều bên là không rõ ràng. Hãy tưởng tượng một tình huống trong đó tất cả các bên trong một kênh ảo chơi một trò chơi trong đó tất cả các bên trả một lượng Coin vào trò chơi và người chiến thắng sẽ nhận được tất cả số tiền đó, tuy nhiên, người chiến thắng không được quyết định cho đến khi kết thúc trò chơi. Sau đó, một bên có hành vi sai trái để kích hoạt cơ chế trừng phạt trước khi trò chơi kết thúc. Một cơ chế trừng phạt tương tự như cơ chế trong công việc liên quan sẽ dẫn đến việc chấm dứt trò chơi và trả Coin cho tất cả các bên trung thực để đảm bảo không ai trong số họ bị mất Coin do điều này. Tuy nhiên, vì trò chơi vẫn chưa được giải quyết, nên không rõ Coin nên được phân phối như thế nào giữa các bên trung thực một cách công bằng.

Đóng góp của chúng tôi. Công việc này đề xuất một giao thức để tạo các kênh trạng thái *ảo* và cung cấp phân tích bảo mật. Cách tiếp cận của chúng tôi bao gồm hai thành phần, một máy trạng thái Interhead cũng như một giao thức xác định hành vi của tất cả các bên. Việc xây dựng (1) có thể được thực hiện lặp đi lặp lại để tạo thành một kênh ảo trên mạng lưới kênh, (2) hoạt động Off-Chain một cách lạc quan, nghĩa là nếu tất cả các bên hợp tác thì không có giao dịch nào được thêm vào sổ cái, (3) được bảo mật khi có sự hiện diện của một kẻ tấn công độc hại có thể làm hỏng tĩnh tất cả trừ một bên khi bắt đầu giao thức, (4) không khiến bên trung thực có nguy cơ mất Coin của họ, (5) hỗ trợ các kênh với số lượng bên tùy ý (6) cho phép sự hiện diện của nhiều trung gian có thể chia sẻ gánh nặng cam kết tài sản thế chấp bắt buộc, cải thiện đáng kể tính thực tế.

Mặc dù công việc này tập trung vào việc tạo ra một kênh ảo cho các Hydra Head, nhưng chúng tôi lập luận rằng máy trạng thái được đề xuất trong công việc này có thể được triển khai cho các cấu trúc kênh khác trên Blockchain với các hợp đồng thông minh đủ biểu đạt như Ethereum. Như vậy, theo hiểu biết của chúng tôi, chúng tôi trình

bày việc xây dựng kênh ảo đầu tiên với các thuộc tính (5) và (6) từ trên mà chúng tôi nghĩ là có lợi ích độc lập.

Kết cấu. Trong phần còn lại của công việc này, đầu tiên chúng tôi cung cấp tổng quan về công việc liên quan trong Phần 2 và cung cấp thông tin cơ bản liên quan đến công việc này trong Phần 3. Tiếp theo, chúng tôi giới thiệu cách tiếp cận của mình. Chúng tôi trình bày máy trạng thái trong Phần 4, sau đó là cách nó có thể được triển khai dưới dạng CEM trong Phần 5 và một giao thức mô tả hành vi của tất cả các bên trong Phần 6. Tiếp theo là phân tích bảo mật trong Phần 7. Cuối cùng, chúng tôi kết luận trong Phần 8.

2 Công việc liên quan

Các kênh. Về mặt khái niệm, các kênh hoạt động như sau. Hai bên mở một kênh thanh toán giữa họ bằng cách thực hiện một giao dịch *cấp vốn* (Funding) cho sổ cái để khóa tiền của họ vào ví dùng chung. Giao dịch *hoàn tiền* (Refund) thứ hai được thiết lập để chi tiêu số tiền trong ví đó và hoàn lại tiền cho cả hai bên, tương ứng với số tiền ban đầu mỗi bên đã trả vào kênh. Lưu ý rằng giao dịch *hoàn tiền* phải được hoàn thành trước giao dịch *cấp vốn* để tránh việc Coin bị khóa trong kênh vô thời hạn. Các bên giữ lại cam kết giao dịch hoàn trả vào sổ cái, nhưng thay vào đó hãy giữ nó trong bộ nhớ. Giao dịch hoàn tiền thể hiện trạng thái của kênh, là kết quả của việc cam kết giao dịch hoàn tiền vào sổ cái. Quá trình chuyển đổi trạng thái xảy ra bằng cách tạo một giao dịch hoàn tiền mới và vô hiệu hóa hoàn toàn hoặc rõ ràng tất cả các giao dịch hoàn tiền trước đó. Các cách tiếp cận hiện tại khác nhau về cách thực hiện việc vô hiệu hóa. Các kênh trong mạng Lightning [22] có cả hai bên trao đổi khóa vô hiệu cho phép họ yêu cầu tất cả Coin trong một kênh nếu đối tác của họ xuất bản một giao dịch hoàn tiền cũ. Các kênh thanh toán song công [6] sử dụng khóa thời gian, trong khi Eltoo [21] sử dụng các giao dịch treo, tức là các giao dịch mà đầu vào có thể được thay đổi sau khi tạo trước khi chuyển chúng vào sổ cái. Khái niệm về các kênh có thể được mở rộng sang các kênh trạng thái [19,7,8] bằng cách sử dụng các hợp đồng thông minh. Các kênh như vậy có thể lưu trữ trạng thái tùy ý và hơn nữa, cho phép các bên thực thi các máy trạng thái.

Mạng lưới kênh. Một số giao thức tồn tại cho phép thực hiện thanh toán qua đường dẫn kênh thanh toán có độ dài $n \in \mathbb{N}$. Một khoản thanh toán trong mạng được mô phỏng bằng cách thực hiện nó trên mỗi bước nhảy trên đường dẫn thanh toán. Thách thức là thực hiện các khoản thanh toán này một cách nguyên tử, tức là nó được thực hiện trên tất cả các kênh trong đường dẫn thanh toán hoặc không có kênh nào. Ở đây, chúng tôi phân biệt giữa người gửi và người nhận thanh toán và các bên trung gian trong một đường dẫn kênh giữa cả hai bên. Hash hợp đồng khoá thời gian (HTLCs)

[22] biến người nhận mẫu thanh toán thành bí mật $x \in \mathbb{N}$ s.t. $H(x) = y$ trong đó H là hàm Hash mật mã. Mỗi kênh trên đường dẫn thanh toán sẽ thiết lập một khoản thanh toán có điều kiện để thực hiện thanh toán nếu người được thanh toán có thể cho người thanh toán biết hình ảnh trước của *y*. Sau khi thiết lập, người nhận tiết lộ *x* cho người tiền nhiệm trên đường dẫn thanh toán cho phép cả hai bên giải quyết thanh toán trên kênh của họ. Đổi lại, người tiền nhiệm học *x* và có thể chuyển tiếp nó để lấy lại Coin của nó. Một nhược điểm của HTLC là tổng tài sản thế chấp bắt buộc, tức là số lượng Coin bị khóa trong các khoản thanh toán có điều kiện nhân với thời hạn, tổng bằng $O(n^2)$. Các phương pháp tiếp theo là Cập nhật đa kênh nguyên tử [9] giúp giảm tổng tài sản thế chấp xuống $O(n)$ mặc dù người ta thấy rằng các bên trung thực dễ bị mất Coin [13]. Hơn nữa, Jourenko và cộng sự [13] giới thiệu Cây thanh toán, một giao thức làm giảm tổng tài sản thế chấp xuống còn $O(n)$ nhưng yêu cầu bất kỳ bên nào cam kết giao dịch tối đa $O(\log n)$ vào sổ cái trong trường hợp có tranh chấp, trái ngược với giao dịch O(1) như với các cách tiếp cận trước đó. Sprites [19] cho phép thanh toán với tổng tài sản thế chấp bằng $O(n)$ và không tăng số lượng giao dịch trên O(1) cho mỗi bên, nhưng yêu cầu sử dụng hợp đồng thông minh **PreimageManager**.

Các kênh ảo. Các cách tiếp cận khác để cho phép các bên tương tác với nhau qua mạng lưới các kênh ở dạng *kênh ảo*. Giả sử Alice và Ingrid, cũng như Ingrid và Bob chia sẻ một kênh. Các giao thức này cho phép hai kênh liền kề với một bên chung được gọi là *trung gian* để tạo kênh thứ ba. Trong ví dụ trên, Ingrid có thể đóng vai trò trung gian để tạo kênh giữa Alice và Bob. Các kênh này được tạo Off-Chain lạc quan, tức là không thực hiện bất kỳ giao dịch nào trên sổ cái trừ trường hợp có tranh chấp. Dziembowski và cộng sự [7,8] đã tạo các kênh trạng thái ảo dựa trên hợp đồng thông minh, trong khi các kênh thanh toán ảo nhẹ [12] cho phép tạo kênh thanh toán ảo mà không cần hợp đồng thông minh.

Các Hydra Head. Cardano/Ouroboros Blockchain cho phép thực thi các Máy phát ra ràng buộc (CEM) [2] là một dạng máy trạng thái có nguồn gốc từ Mealy Automata. Hydra [4] đề xuất một CEM và giao thức cho phép tạo các kênh trạng thái đa bên. Hydra là đẳng cấu, vì nó cho phép các bên không chỉ khóa tiền bên trong kênh mà còn là một tập hợp con thích hợp của trạng thái sổ cái. Làm như vậy, các bên có thể tương tác với nhau trong Hydra Head giống như cách họ có thể làm trên sổ cái. Do đó, các khoản thanh toán đơn giản có thể được thực hiện trên nhiều Hydra Head bằng giao thức HTLC. Tuy nhiên, trong khi các kênh trạng thái ảo [7,8] cho phép các bên thực thi các máy trạng thái trên mạng lưới kênh trạng thái, không có phương pháp nào để thực thi CEM trên mạng các Hydra Head. Công việc này lấp đầy khoảng trống này.

3 Tiểu sử

Ký hiệu. Trong công việc này, chúng tôi thường xuyên sử dụng các Tuple để cấu trúc dữ liệu. Cho α là một thể hiện của Tuple loại A có dạng $(\alpha_0, ..., \alpha_n)$, $n \in \mathbb{N}$ trong đó α_0, ..., α_n là các nhãn của mục nhập. Sau đó, chúng tôi xử lý mục $i \in \mathbb{N}$, $0 \leq i \leq n$ của α sử dụng tên của nó và nhãn của mục nhập, tức là $\alpha.\alpha_i$. Ngoài ra, chúng tôi ký hiệu $\mathbb{N}$ là tập hợp các số tự nhiên và $\mathbb{B}$ là tập hợp Boolean.

Lược đồ chữ ký. Chúng tôi giả định sự tồn tại của hai lược đồ chữ ký số an toàn như sau. Chúng tôi giả định rằng cả hai đều đáp ứng các khái niệm về tính **đầy đủ** và tính **không thể sửa chữa**, tuy nhiên, chúng tôi vẫn còn khá không chính thức trong phần còn lại. Đầu tiên, chúng tôi giả sử lược đồ chữ ký [1] bao gồm các thuật toán (**key_gen**, **verify**, **sign**) s.t. **key_gen** $(1^{\lambda}) = (vk, sk)$ tạo một cặp khóa bí mật sk và khóa xác minh vk dưới tham số bảo mật λ, **sign**$(sk, m) = \sigma$ tạo chữ ký s.t. **verify** (vk, m, σ') đánh giá là **True** khi và chỉ khi $\sigma' = \sigma$. Thứ hai, chúng tôi giả sử tồn tại lược đồ đa chữ ký [11,18] có dạng (**ms_setup**, **ms_key_gen**, **ms_agg_vk**, **ms_sign**, **ms_agg_sign**, **ms_verify**) trong đó **ms_setup**$(1^{\lambda'}) = \Pi$ tạo tham số công khai Π với tham số bảo mật λ', **ms_key_gen**(Π) = (vk', sk') tạo cặp khóa gồm khóa bí mật sk' và khóa xác minh vk', **ms_agg_vk**(Π, V) = **avk** tổng hợp một bộ khóa xác minh V thành khóa xác minh tổng hợp **avk**, **ms_sign**$(\Pi,$ sk', $m')$ = σ'' tạo chữ ký σ'' của thông báo m tương ứng với khóa bí mật sk' trong khi **ms_agg_sign**(Π, V, S, m') = σ_{agg} tổng hợp một tập hợp các chữ ký S thành chữ ký tổng hợp σ_{agg} s.t. **ms_verify**(Π, m', avk, σ'') đánh giá là **True** khi và chỉ khi $\sigma'' = \sigma_{agg}$, nếu không thì sẽ đánh giá thành **False**.

Mô hình EUTxO. Đầu ra giao dịch chưa chi tiêu mở rộng (EUTxO) đã được giới thiệu bởi Chakravarty và cộng sự [2]. EUTxO cải thiện mô hình UTxO như được sử dụng với sổ cái như Bitcoin do Nakamoto giới thiệu [20] bằng cách cho phép thực hiện các hợp đồng thông minh được xác định là Máy phát ra ràng buộc (CEM) trên sổ cái, do đó cải thiện tính biểu đạt của hệ thống. Sổ cái dựa trên EUTxO bao gồm một tập hợp $(\mathbf{out}_{ref}, u)$ trong đó u là EUTxO đại diện cho số Coin đang lưu hành và $\mathbf{out}_{ref}$ là một mã định danh duy nhất có thể được sử dụng để tham chiếu u và thường được lấy từ Context (ngữ cảnh) mà nó được tạo ra. Một giao dịch tx là một Tuple có dạng (I, O, r, S) trong đó I là một tập hợp các đầu vào, tức là các mục có dạng $(\mathbf{out}_{ref}, u)$, O là danh sách các đầu ra, tức là EUTxO mới được xác định, r là khoảng thời gian giá trị, nghĩa là $[r_0, r_1]$ trong đó $r_0, r_1 \in \mathbb{N}$ là các điểm trong thời gian và S là một tập hợp các chữ ký. Nếu một giao dịch được gửi đến sổ cái trong khoảng thời gian r thì số lượng Coin trong O ít nhất phải lớn bằng số lượng Coin được tham chiếu trong I và tất cả các tập lệnh hợp lệ đều đánh giá là **True**, giao dịch sẽ tạo ra chuyển đổi trạng thái trên sổ cái bởi xóa tất cả các mục trong I khỏi trạng thái của nó và thêm EUTxO mới được xác định trong O vào trạng thái của sổ cái. Một giao dịch được sổ cái xử lý trong khoảng

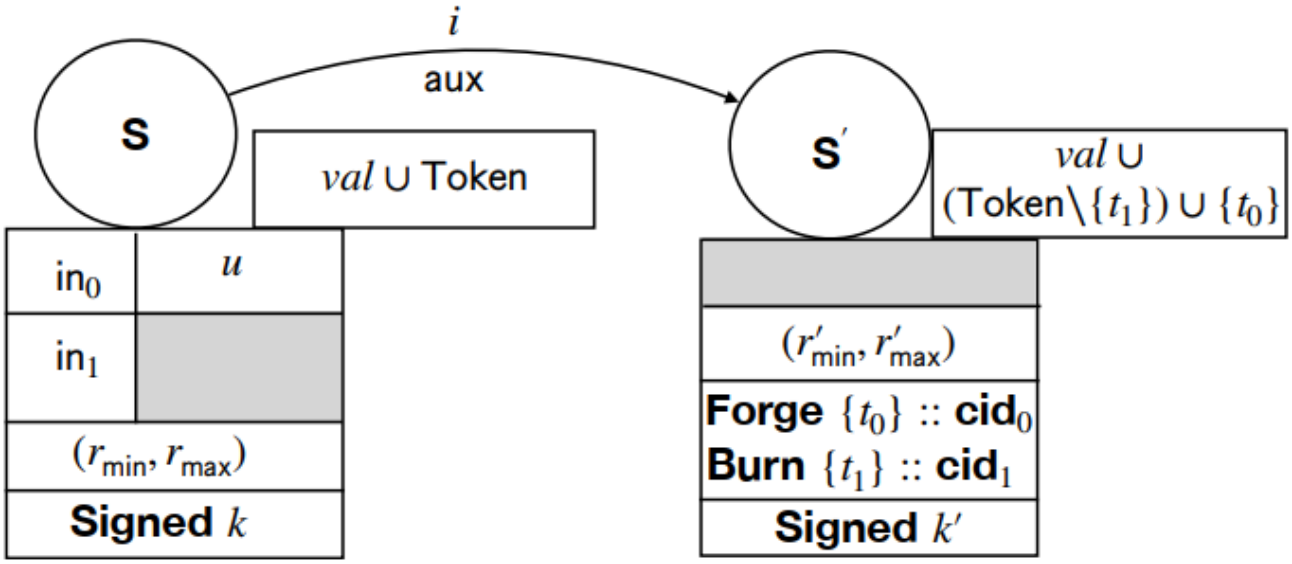
thời gian $\Delta \in \mathbb{N}$. Bản thân EUTxO u là một Tuple có dạng (v, *value*, δ) trong đó $v \in \{0, 1\}^*$ là tập lệnh trình xác thực được viết bằng ngôn ngữ Turing Complete, *value* $\in \mathbb{N}$ là số lượng Coin và $\delta \in \{0, 1\}^*$ là dữ liệu tùy ý. Một EUTxO có thể được chi tiêu để Coin của nó có thể truy cập được, nếu một bên có thể hiển thị giá trị Redeemer $\rho \in \{0, 1\}^*$ s.t. $v(\rho, \delta, \sigma)$ = **True**, trong đó σ là ngữ cảnh xác thực bao gồm thông tin về giao dịch sử dụng u cũng như tất cả EUTxO được tham chiếu trong đầu vào của nó.

Giao thức Off-Chain lạc quan. Một loại giao thức giúp cải thiện khả năng mở rộng của sổ cái bằng cách giảm thiểu số lượng giao dịch được thêm vào sổ cái là các giao thức Off-Chain. Các giao thức này hoạt động trên các cấu trúc được gọi là kênh [22,21,6,8] và cho phép hai bên tương tác với nhau, ví dụ: thực hiện thanh toán mà không cần thêm bất kỳ giao dịch nào trên sổ cái trong trường hợp *lạc quan*, tức là khi tất cả các bên hợp tác và có không tranh chấp.

Các Hydra Head. Hydra CEM [4] cho phép một số lượng người tham gia tùy ý di chuyển EUTxO của họ vào Hydra Head và sử dụng chúng để tương tác Off-Chain với nhau. Hydra đại diện cho một cấu trúc kênh giữa một số bên tùy ý. Trong khi hoạt động, Hydra Head ở trạng thái (**open**, K_{agg}, η, h_{MT}, n, T) trong đó **open** là nhãn của trạng thái, K_{agg} là khóa xác minh tổng hợp giữa n người tham gia, η là bộ EUTxO mà được giữ Off-Chain, h_{MT} là gốc của Cây Merkel đại diện cho tất cả những người tham gia và T là khoảng thời gian *tối thiểu* của giai đoạn tranh chấp có thể xảy ra khi đóng Hydra Head. Bộ EUTxO đã được di chuyển đến một Head được lưu trữ trong ảnh chụp nhanh đại diện cho bộ EUTxO được chuyển đến sổ cái khi đóng Head. Người tham gia có thể sửa đổi bộ EUTxO này bằng cách tạo một ảnh chụp nhanh mới thay thế các ảnh chụp nhanh đã tạo trước đó. Do đó, ảnh chụp nhanh mới nhất đại diện cho trạng thái của EUTxO trong Head có thể được *thực thi* trên sổ cái. Người tham gia có thể tạo ảnh chụp nhanh mới trong khi Head ở trạng thái **open**. Head được đóng lại bằng cách chuyển trạng thái của nó đầu tiên sang trạng thái **closed** và sau đó sang trạng thái **newestSN** và trạng thái **final**. Cuối cùng, một giao dịch **split** (phân tách) làm cho EUTxO có sẵn trên sổ cái. Hơn nữa, EUTxO có thể ngừng hoạt động dần dần, điều này làm cho nó có sẵn trên sổ cái mà không cần đóng Head. Hoạt động này cần có sự đồng ý của tất cả các thành viên Head.

Trạng thái có thể thực thi. Các kênh cho phép duy trì và sửa đổi *trạng thái có thể thực thi*. Chẳng hạn, trong trường hợp các kênh thanh toán, trạng thái có thể thực thi là cách một lượng Coin cố định được phân phối giữa hai bên. Trong trường hợp của Hydra, đó là một bộ EUTxO được xác định bằng ảnh chụp nhanh gần nhất.

Sau đây chúng tôi đưa ra tổng quan về các khái niệm có liên quan.



Hình 1: Minh họa Chuyển trạng thái $S \rightarrow S'$ trên đầu vào (*i*, *aux*). Hộp bên dưới trạng thái hiển thị thông tin về các ràng buộc giao dịch đối với giao dịch đã thực hiện quá trình chuyển đổi trạng thái. Hộp ở bên phải của trạng thái hiển thị tổng quan về trường *value* của EUTxO đại diện cho trạng thái. Các trường giao dịch trống hoặc ẩn khỏi ngữ cảnh sẽ được tô xám hoặc bỏ qua để đơn giản hóa.

EUTxO$_{\text{MA}}$. Mô hình EUTxO được mở rộng bởi EUTxO$_{\text{MA}}$ [3] để thêm hỗ trợ đa tài sản. EUTxO$_{\text{MA}}$ được định nghĩa là EUTxO nhưng cho phép trường *value* mang Token không thể thay thế ngoài các đồng Coin có thể thay thế. Ngoài ra, một giao dịch trong mô hình EUTxO$_{\text{MA}}$ có thêm hai mục nhập, nghĩa là giao dịch đó có dạng (*I*, *O*, **forge**, **fpss**, *S*) trong đó **forge** là gói Token có thể xác định số lượng dương của Token trong trường hợp chúng được đúc, hoặc số lượng âm của Token trong trường hợp chúng bị đốt. Ngoài ra, **fpss** là tập lệnh chính sách được tạo ra lấy ngữ cảnh xác thực σ làm đầu vào và đánh giá xem giao dịch bao gồm trường **forge** của nó có được chấp nhận hay không. Trong phần còn lại, chúng tôi giả định là EUTxO$_{MA}$ nhưng chúng tôi tiếp tục sử dụng thuật ngữ EUTxO cho ngắn gọn.

Máy phát ra ràng buộc. Chakravarty và cộng sự [2] cho thấy mô phỏng sinh học yếu giữa các chương trình chạy trên sổ cái EUTxO và Máy phát ra ràng buộc (CEM) có nguồn gốc từ máy Mealy [17]. Do đó, CEM có thể được sử dụng để xác định các ứng dụng cho sổ cái EUTxO. Một CEM là một Tuple có dạng (**S**, **I**, **step**, **initial**, **final**) trong đó **S** là một tập hợp các trạng thái có thể là vô hạn, **I** là một tập hợp các ký hiệu đầu vào, **initial**, **final** là các hàm **S** → $\mathbb{B}$ tương ứng chỉ trạng thái ban đầu và kết thúc, và hàm **step**: **S** → **I** → **Maybe** (**S**, **TxConstraints**) là một hàm bộ phận ánh xạ tới một trạng thái mới với các ràng buộc **TxConstraints**.

Một CEM được thực hiện trên sổ cái như sau. Một trạng thái **S** được đại diện bởi một EUTxO *u* trong đó trình xác thực $u.nu = v_S$ là duy nhất cho trạng thái và thực thi tính chính xác của các chuyển đổi trạng thái và các ràng buộc giao dịch **TxConstraints**. Một cách trừu tượng hơn, Thuật toán xác minh On-Chain (OVC - Onchain Verification Algorithm) tương ứng với trạng thái được triển khai và thực thi trên sổ cái thông qua η_S. Hình 1 hiển thị cách chúng tôi minh họa các trạng thái và chuyển tiếp trong công việc này. Các hộp bên dưới mỗi trạng thái **S** và **S'** hiển thị thông tin về **TxConstraint** của giao dịch được sử dụng để thực hiện quá trình chuyển đổi. Chẳng hạn, trong Hình 1, *S'* yêu cầu Token có mã định danh tiền tệ $\mathbf{cid}_1$ bị đốt và Token có mã định danh $\mathbf{cid}_0$ được tạo ra. Ngoài ra, nó yêu cầu giao dịch có khoảng thời gian hiệu lực là $[r'_{min}, r'_{max}]$ và nó chứa chữ ký tương ứng với khóa xác minh *k'*. Ngoài ra, người ta cho rằng Redeemer sử dụng EUTxO đại diện cho trạng thái *S* bằng $\rho = i||\mathbf{aux}$. EUTxO đại diện cho hai trạng thái *S* và *S'* trong các đầu vào và đầu ra của các trạng thái là ngầm định và được bỏ qua để đơn giản.

Thread Token. Một mẫu thiết kế cho phép bắt buộc rằng một CEM nhất định (1) đã bắt đầu ở trạng thái ban đầu hợp lệ và (2) là duy nhất so với các phiên bản khác của các CEM tương tự đang sử dụng Thread Token. Trình xác thực trạng thái ban đầu yêu cầu tạo Thread Token với tập lệnh chính sách được tạo ra tương ứng. Token sẽ được lưu giữ trong tất cả các trường giá trị EUTxO thông qua quá trình chạy CEM cho đến khi Token đạt đến trạng thái cuối cùng buộc phải đốt.

4 Tổng quan

Phần này trước tiên giới thiệu cài đặt và các khái niệm liên quan đến công việc này. Sau đó, chúng tôi trình bày máy trạng thái Interhead mà chúng tôi triển khai dưới dạng hợp đồng thông minh dưới dạng CEM trong Phần 5. Giao thức mô tả cách các bên tương tác với nhau và máy trạng thái được mô tả trong Phần 6.

Cách tiếp cận và thuật ngữ. Thuật ngữ của chúng tôi được chọn để phù hợp với Hydra [4]. Mục đích của công việc này là tạo ra *cấu trúc Interhead* bao gồm hai phần. Phần đầu tiên, *máy trạng thái Interhead* được vận hành giữa 2 *Hydra Head* riêng biệt. Phần thứ 2, *giao thức Interhead* xác định hành vi của các bên liên quan tùy thuộc vào vai trò của họ trong quá trình xây dựng. Máy trạng thái Interhead hoạt động trên 2 Head nên được chia thành hai phần rời rạc ban đầu. Các *máy trạng thái từng phần* này được vận hành song song bởi các thành viên Head tương ứng giống như thực thi đa *luồng*. Tuy nhiên, chúng được thiết lập sao cho các luồng có thể hợp nhất để mở một Hydra Head thông thường trên sổ cái trong trường hợp có tranh chấp. Tương tự như cách Hydra Head duy trì một bộ EUTxO ở *trạng thái có thể thực thi* giữa những người tham gia, Interhead được sử dụng để tạo một *Head ảo* duy trì một bộ EUTxO ở trạng thái

có thể thực thi giữa một tập hợp con những người tham gia của 2 Hydra Head. Tuy nhiên, Head được tạo mà không cần mở Hydra Head trên sổ cái, nên nó được gọi là *ảo*. Trong khi tính chính xác của máy trạng thái Interhead được xác minh và đảm bảo bởi sổ cái, hành vi của chính các bên được xác định bởi *giao thức Interhead*.

4.1 Cài đặt

Chúng tôi giả sử sự tồn tại của 2 Hydra Head, H_b, $b \in \{0, 1\}$. Trong mỗi Head, $n_b \leq H_b.n$ *người tham gia*, là tập hợp con của các bên, muốn di chuyển một phần EUTxO của họ trong Head tương ứng sang Hydra Head *ảo* mới H^v, do đó cho phép tương tác giữa những người tham gia này. Hơn nữa, có $1 \leq n_i \leq \min(H_0.n, H_1.n)$ có mặt ở cả 2 Head và đóng vai trò *trung gian*. Để đơn giản, trong phần còn lại của công việc này, mỗi bên có chính xác một vai trò, tức là bên trung gian hoặc bên tham gia, nhưng lưu ý rằng các bên có thể có cả hai hoặc không có vai trò nào. Chúng tôi xem xét 4 nhóm các bên. Nhóm thứ nhất có những *người tham gia* G_b từ 1 trong 2 Head. Hợp của 2 nhóm là tập hợp những người tham gia Hydra Head ảo $G^v = G_0 \cup G_1$. Cuối cùng là nhóm *trung gian* G_i.

Mô hình truyền thông và thời gian. Chúng tôi giả định rằng giao tiếp giữa các bên diễn ra thông qua các kênh được xác thực và được thực hiện trong các vòng. Một thông điệp được gửi ở bất kỳ vòng nào sẽ có sẵn cho người nhận ở đầu vòng tiếp theo. Chúng tôi giả định rằng có một mối quan hệ giữa một vòng giao tiếp nhất định và thời gian đồng hồ [14,15,16] tại đó nó đang diễn ra. Chúng tôi sử dụng thời gian và các vòng giao tiếp thay thế cho nhau trong phần còn lại.

Mô hình kẻ tấn công. Mô hình kẻ tấn công của chúng tôi phù hợp với công việc liên quan. Nghĩa là, chúng tôi giả định rằng một kẻ tấn công nguy hiểm khi bắt đầu giao thức có thể làm hỏng tĩnh tất cả trừ một bên, tức là tối đa $n_0 + n_1 + n_i - 1$ bên. Khi bị hỏng, tình trạng nội bộ của một trạng thái bị rò rỉ cho kẻ tấn công và tất cả các thông tin liên lạc đến và từ trạng thái đều thông qua kẻ tấn công. Kẻ tấn công có thể làm cho bất kỳ bên bị hỏng nào đi chệch khỏi giao thức một cách tùy tý. Hơn nữa, kẻ tấn công có thể sắp xếp lại các thông điệp và trì hoãn chúng cho đến vòng giao tiếp sau.

4.2 Thuộc tính mong muốn

Việc xây dựng được thiết kế để đáp ứng các thuộc tính sau đây. Lưu ý rằng ngay khi máy trạng thái Interhead chuyển sang không gian trạng thái Hydra, các thuộc tính bảo mật của Hydra [4] được giữ nguyên. Đầu tiên, chúng tôi xác định các thuộc tính bảo mật và sau đó đưa ra tổng quan về các thách thức.

Định nghĩa 1 (Tính sống động của tài sản thế chấp). *Nếu ít nhất một bên trung gian là trung thực thì tất cả tài sản thế chấp cuối cùng sẽ có sẵn cho họ ở trạng thái có thể thực thi.*

Định nghĩa 2 (Tính sống động của EUTxO). *Cuối cùng, EUTxO của bất kỳ bên trung thực nào trong trạng thái có thể thực thi của Head ảo đều có sẵn cho họ ở trạng thái có thể thực thi (hoặc sổ cái) bên ngoài Head ảo hoặc máy trạng thái Interhead chuyển sang không gian trạng thái Hydra.*

Định nghĩa 3 (Bảo mật số dư). *Tổng số Coin của một bên trung thực chỉ bị giảm khi có sự đồng ý của họ hoặc máy trạng thái Interhead chuyển sang không gian trạng thái Hydra.*

Bảo mật. Việc xây dựng phải an toàn cho tất cả những người tham gia trung thực. Ngay cả khi tất cả những người tham gia xây dựng Interhead khác đang hành xử ác ý, bên trung thực không thể mất bất kỳ Coin nào. Điều này yêu cầu cấu trúc Interhead phải đáp ứng bảo mật số dư, tính sống động của tài sản thế chấp và tính sống động của EUTxO .

Off-Chain lạc quan. Quá trình xây dựng của chúng tôi phải Off-Chain lạc quan, nghĩa là nếu tất cả các bên hợp tác, một Hydra Head ảo có thể được xây dựng, sử dụng và đóng mà không cần cam kết thực hiện bất kỳ giao dịch nào trên sổ cái.

Nhiều trung gian. Công trình của chúng tôi nên cho phép nhiều trung gian. Mặc dù điều này không cung cấp bất kỳ tính năng bổ sung nào cho bản thân công trình, nhưng nó rất phù hợp để biến công trình thành hiện thực. Số lượng tài sản thế chấp được cam kết phải khớp với số lượng Coin và Token trong Head ảo. Việc có nhiều bên trung gian cho phép chúng tôi chia nhỏ gánh nặng cam kết đủ tài sản thế chấp. Tuy nhiên, thực hiện điều này một cách an toàn tự nó là một thách thức không hề nhỏ. Tất cả các phương pháp tiếp cận hiện có cho các kênh ảo [12,7,8] đều chỉ có một trung gian có khả năng cung cấp bảo mật cho việc xây dựng, nhưng đến lượt nó có thể bị đổ lỗi duy nhất nếu chúng đi chệch khỏi giao thức. Thách thức của nhiều bên trung gian là đảm bảo rằng nhóm các bên trung gian có thể cung cấp bảo mật cho công trình giống như với một bên trung gian, tuy nhiên, bên trung gian trung thực không nên mất tài sản thế chấp trong trường hợp tất cả bên trung gian còn lại hành xử ác ý.

Các ràng buộc đối với Hydra Head. Chúng tôi yêu cầu một số ràng buộc bổ sung *nhỏ* đối với Hydra Head để hỗ trợ Hydra Head ảo. Lượng thời gian tối thiểu cần thiết để một EUTxO có sẵn trên sổ cái – giả sử những người tham gia Head không tham gia vào việc ngừng cam kết tăng thêm – phụ thuộc vào khoảng thời gian tranh chấp T của một Head và tổng cộng lên tới $2T$, tức là khoảng thời gian đăng ảnh chụp nhanh tối

thiểu $T_{SN} \geq T + R_C$ và khoảng thời gian đăng giao dịch treo tối thiểu $T_{HT} \geq T + R_{SN}$ [4] trong đó R_C và R_{SN} là khoảng thời gian hiệu lực của các giao dịch chuyển sang trạng thái **closed** và **newestSN** tương ứng. Tuy nhiên, không có giới hạn trên về thời gian cần thiết để thêm EUTxO vào sổ cái vì các bên có thể cố gắng trì hoãn việc đóng Head bằng cách chọn khoảng thời gian lớn cho T_{SN}, T_{HT}, R_C và R_{SN} tương ứng. Chúng tôi yêu cầu giới hạn trên cho các tham số này, tức là chúng tôi yêu cầu T_{SN}^{max}, T_{HT}^{max}, R_C^{max} và R_{SN}^{max} s.t. $T_{SN}^{max} \geq T_{SN} \geq T$, $T_{HT}^{max} \geq T_{HT} \geq T$, $R_C^{max} \geq R_C \geq \Delta$, $R_{SN}^{max} \geq R_{SN} \geq \Delta$. Làm như vậy không ngăn được việc đóng Head, điều này sẽ ảnh hưởng đến thuộc tính *tính sống động* của Hydra vì những người tham gia vẫn có thể tự do trì hoãn quá trình chuyển đổi trạng thái một cách tùy ý, tức là sẽ không bao giờ là quá muộn để đóng Head. Tuy nhiên, một bên cố gắng đóng Head để cung cấp EUTxO trên sổ cái có thể cho rằng họ sẽ sẵn sàng làm như vậy sau thời gian tối đa là $T^{max} \leq T_{SN}^{max} + T_{HT}^{max} + R_C^{max} + R_{SN}^{max} + 2\Delta$. Điều này giữ nguyên nếu nó tích cực tham gia để làm như vậy, tức là miễn là nó cam kết các giao dịch cần thiết vào sổ cái bất cứ khi nào có thể. Lưu ý rằng 2Δ bổ sung đại diện cho lượng thời gian cao hơn để thực hiện chuyển đổi trạng thái sang trạng thái *đóng* và chuyển giao dịch phân tách vào sổ cái.

Token mục đích chung. Hydra Head hạn chế ở chỗ không thể tạo ra và đốt Token tùy ý. Token mục đích đặc biệt, chẳng hạn như Thread Token chỉ có thể được tạo ra trong một ngữ cảnh cụ thể được xác định trong tập lệnh chính sách được tạo ra của nó. Một giao dịch tạo ra Token như vậy không thể được đưa vào trạng thái có thể thực thi vì điều này sẽ dẫn đến việc Token bị tạo ra trong quá trình chuyển đổi trạng thái Hydra CEM, điều này có khả năng vi phạm tập lệnh chính sách được tạo ra của nó. Một giải pháp thay thế cho vấn đề này là điều chỉnh Token cũng như CEM mà nó được sử dụng để nhận biết Hydra sao cho việc di chuyển CEM vào và ra khỏi Hydra Head cũng như việc tạo ra và đốt Token cụ thể trong một trạng thái Hydra. Một cách giải quyết khác là bằng mẫu *Token tổng quát* (Generalized Token). Token tổng quát được tạo ra trong một ngữ cảnh tùy ý nên có thể được tạo ra trong bất kỳ quá trình chuyển đổi trạng thái nào của Hydra Head. Token tổng quát có thể được tạo dưới dạng Token có thể thay thế hoặc không thể thay thế. Một CEM sử dụng Token để thực hiện chức năng không tạo ra hoặc đốt Token, mà thay vào đó lấy số lượng Token cần thiết làm đầu vào khi được yêu cầu và phát hành Token ở trạng thái cuối cùng của CEM muộn nhất.

CEM đa luồng. Chúng tôi mở rộng khái niệm về Thread Token bằng cách cho phép các CEM giữ nhiều Thread Token. Một CEM có thể sinh ra các luồng để được thực thi song song bằng cách có một giao dịch chứa nhiều EUTxO trong đầu ra của nó, mỗi EUTxO đại diện cho một trạng thái CEM riêng biệt và nắm giữ ít nhất một Thread Token. Đổi lại, nhiều luồng có thể được hợp nhất thành một luồng bằng cách giao dịch sử dụng nhiều EUTxO đại diện cho các trạng thái CEM, sử dụng Thread Token của chúng và xác định một EUTxO trong đầu ra chứa tất cả Thread Token trong trường

value của nó. Chúng tôi sử dụng đa luồng trong hai trường hợp. Thứ nhất, chúng tôi sử dụng nhiều luồng – một luồng cho mỗi nhận dạng – để thu thập hiệu quả EUTxO sẽ được chuyển đến Head ảo. Lưu ý rằng điều này tương tự như cách Hydra CEM thu thập EUTxO [4] để chuyển vào Head. Thứ hai, ban đầu chúng tôi tạo ra một luồng trong mỗi Head, tức là mỗi phiên bản của Interhead CEM có chính xác hai trạng thái ban đầu chứa một Thread Token. Nếu Interhead giải quyết một cách lạc quan, CEM vẫn tách biệt và các luồng không bao giờ được hợp nhất. Tuy nhiên, trong trường hợp xảy ra tranh chấp hoặc thiếu hợp tác, khi Interhead được chuyển thành Hydra Head thông thường, các luồng sẽ được hợp nhất.

Khả năng tương thích với Hydra. Chúng tôi cố gắng làm cho công trình xây dựng của mình tương thích với công trình hiện có, tức là các Hydra Head, và chúng tôi yêu cầu những điều chỉnh tối thiểu nhất. Làm như vậy, chúng tôi có thể dựa vào nền tảng được thực hiện cho các Hydra Head và các thuộc tính bảo mật của chúng, cho phép chúng tôi tập trung vào việc xây dựng Interhead.

4.3 Máy trạng thái

Cách tiếp cận của chúng tôi là sử dụng máy trạng thái Hydra [4] và điều chỉnh các khái niệm về kênh ảo dựa trên UTxO [12,13] để tạo máy trạng thái Interhead duy trì Hydra Head ảo. Cách tiếp cận của chúng tôi được lấy cảm hứng từ Eltoo [21], tức là chúng tôi tránh trừng phạt bất kỳ người tham gia nào bằng cách đảm bảo rằng Hydra Head ảo luôn có thể được mở trên sổ cái như một Hydra Head thông thường trong trường hợp có tranh chấp.

Máy trạng thái Interhead được thực thi Off-Chain ở cả 2 Hydra Head H_0 và H_1. Nó duy trì một bộ EUTxO giữa các danh tính G^v ở trạng thái có thể thực thi. Interhead có thể được giải quyết một cách lạc quan trong cả 2 Head, nhưng trong trường hợp có tranh chấp, nó cho phép thực thi trạng thái có thể thực thi của Head ảo bằng cách mở một Hydra Head thông thường có cùng trạng thái có thể thực thi trên sổ cái. Người trung gian là các bên phải tham gia ở cả 2 Head, cho phép họ đảm bảo tính đúng đắn của hệ thống. Các bên trung gian phân bổ tài sản thế chấp vào hệ thống, tài sản này sẽ được trả lại cho họ khi máy trạng thái Interhead giải quyết một cách lạc quan hoặc khi Hydra Head ảo được mở trên sổ cái. Tuy nhiên, trong trường hợp tất cả các trung gian bị hỏng và không thực hiện được nhiệm vụ của mình, họ sẽ mất tài sản thế chấp nên sẽ được sử dụng để đảm bảo có đủ Coin để mở Head ảo trên sổ cái.

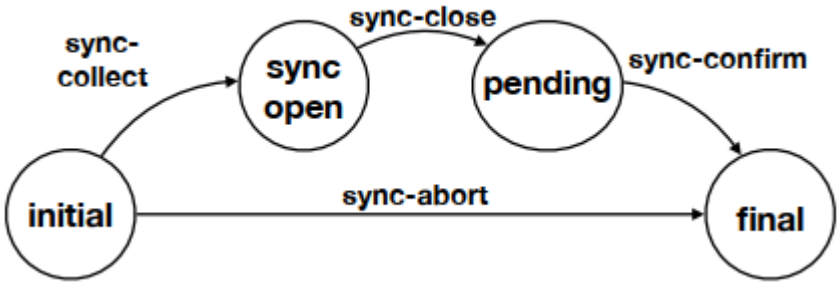
Giai đoạn thời gian và các giả định. Chúng tôi cấu trúc việc thực thi máy trạng thái thành ba giai đoạn, mỗi giai đoạn hoạt động theo các giả định khác nhau và trong các khung thời gian khác nhau. (1) Giai đoạn *sắp xếp* $T_O = [0, t_{C,\text{start}})$ giả định rằng ít nhất một người trung gian là trung thực và tất cả các bên hợp tác. (2) Giai đoạn *chuyển đổi*

$T_C = [t_{C,\text{start}}, t_{C,\text{end}})$ giả định rằng ít nhất một trung gian là trung thực. Chúng tôi yêu cầu thời lượng của giai đoạn này ít nhất là $\max(H_b . T^{\max}, H_{1-b}.T^{\max} + 2\Delta < t_{C,\text{end}} - t_{C,\text{start}}$. (3) Giai đoạn trừng phạt bắt đầu từ $t_{C,\text{end}}$ và tiếp tục vô thời hạn với giả định rằng có ít nhất một bên trung thực, điều này được đảm bảo nhờ mô hình kẻ tấn công. Nếu bất kỳ giả định nào trong một giai đoạn bị vi phạm, CEM sẽ chuyển sang giai đoạn tiếp theo theo thời gian.

Xây dựng lặp đi lặp lại. Một Interhead duy trì trạng thái có thể thực thi giống như Hydra Head, nhưng nó được thiết lập trên hai trạng thái có thể thực thi thay vì một. Lưu ý rằng vì bản thân Interhead duy trì trạng thái có thể thực thi đẳng cấu, nên một Interhead có thể được thiết lập lặp đi lặp lại trên 2 Interhead hoặc 1 Hydra Head và 1 Interhead. Điều này cho phép xây dựng Interhead trên nhiều bước nhảy của cơ sở hạ tầng gồm các Hydra Head. Trong phần sau đây, để đơn giản, chúng tôi giả định rằng Interhead được tạo trên 2 Hydra Head đã được mở trên sổ cái.

Thiết lập. Mỗi nhóm G_0, G_1, G^{ν}, G_i cùng nhau thiết lập lược đồ đa chữ ký như trong Phần 3 dẫn đến việc tạo ra các khóa chữ ký tổng hợp $K_{agg,0}$, $K_{agg,1}$, K_{agg}, $K_{agg,i}$ tương ứng.

Máy trạng thái InterHead. Trong phần sau đây, chúng tôi mô tả các trạng thái của máy trạng thái Interhead cũng như cách thức chuyển đổi trạng thái được thực hiện.



Hình 2: Chuyển đổi trạng thái trong giai đoạn sắp xếp.

Máy trạng thái Interhead được cấu trúc bằng 6 trạng thái, cụ thể là ban đầu (**initial**), mở đồng bộ hóa (**sync_open**), đang chờ xử lý (**pending**), cuối cùng (**final**), được hợp nhất (**merged**) và bị trừng phạt (**punished**). Ở đây **intial** và **final** lần lượt là trạng thái ban đầu và cuối cùng. Ngoài ra, máy trạng thái Interhead chứa các trạng thái, ký hiệu đầu vào, chuyển đổi trạng thái và trạng thái cuối cùng của máy trạng thái Hydra khi nó thực hiện chuyển đổi trạng thái sang trạng thái *mở* của không gian trạng thái Hydra trong trường hợp tranh chấp hoặc không hợp tác. Để đơn giản, chúng tôi loại bỏ điều này và tập trung vào phần duy nhất của máy trạng thái Interhead.

Khi bắt đầu giao thức, các bên trong mỗi Head thiết lập một nửa máy trạng thái Interhead của họ, bắt đầu ở các trạng thái **initial** tương ứng. Máy trạng thái một phần được vận hành trong mỗi Head được hiển thị trong Hình 2. Cả hai máy trạng thái một phần được vận hành song song, nghĩa là mỗi trạng thái **initial** sinh ra một luồng của máy trạng thái Interhead bao trùm. Các trung gian phải đảm bảo rằng các trạng thái ban đầu ở cả 2 Head phù hợp. Trong trường hợp tranh chấp, các luồng có thể được hợp nhất và máy trạng thái Interhead có thể được chuyển sang không gian trạng thái Hydra. Vì lý do này, các trạng thái **initial** chứa dữ liệu (1) đảm bảo các luồng có thể được hợp nhất và (2) các luồng không thể được hợp nhất với một máy trạng thái Interhead khác (3) Hydra Head duy trì trạng thái có thể thực thi giống như Interhead.

Các chuyển đổi trạng thái trong máy trạng thái bị giới hạn trong các giai đoạn thời gian, tức là khoảng thời gian hiệu lực của chúng phải hoàn toàn nằm trong một trong các giai đoạn. Các quá trình chuyển đổi có thể được thực hiện trong một giai đoạn được cấu trúc theo mục đích của giai đoạn đó. Trong phần tiếp theo, chúng tôi trình bày máy trạng thái Interhead được cấu trúc theo ba giai đoạn này, tuy nhiên, tổng quan đầy đủ về toàn bộ máy trạng thái được minh họa trong Phụ lục A.

Giai đoạn sắp xếp. Hình 2 minh họa các trạng thái và quá trình chuyển đổi trong giai đoạn sắp xếp. Trong giai đoạn này, các trung gian có thẩm quyền duy nhất để thực hiện chuyển đổi trạng thái. Trên thực tế, tất cả các quá trình chuyển đổi đều yêu cầu đa chữ ký tương ứng với khóa xác minh $K_{\text{agg},i}$ được ký bởi tất cả các bên trung gian. Một phiên bản của máy trạng thái một phần này được thực thi trong mỗi Head bắt đầu từ trạng thái **initial**. Tất cả các quá trình chuyển đổi yêu cầu ký hiệu đầu vào có đồng bộ hóa tiền tố đều được thực hiện trên cả 2 Head hoặc không. Điều này được đảm bảo thông qua giao thức đồng bộ hóa trong Phần 6 với một cảnh báo: Một bên trung gian bị hỏng có thể cố gắng hủy đồng bộ hóa cả 2 máy trạng thái một phần bằng cách hành động vào thời điểm cuối cùng của giai đoạn sắp xếp và chỉ cung cấp chữ ký của chúng cho một máy trạng thái một phần. Tuy nhiên, điều này có thể được phát hiện bằng chức năng của trạng thái **pending** trong đó các bên trung gian phải xác minh và xác nhận rằng không có nỗ lực hủy đồng bộ hóa nào xảy ra. Việc phát huỷ không đồng bộ hóa dẫn đến máy trạng thái chuyển sang giai đoạn chuyển đổi.

Có nhiều cách để ngăn chặn quá trình chuyển đổi trạng thái nếu một hoặc tất cả các trung gian bị hỏng. Đầu tiên, trình xác thực trong máy trạng thái ngăn chuyển đổi trạng thái không chính xác trong mỗi máy trạng thái một phần, tuy nhiên, điều này là không đủ khi chúng tôi thực thi máy trạng thái song song trong 2 Head. Bất kỳ bên trung gian trung thực nào cũng có thể ngăn quá trình chuyển đổi trạng thái mà nếu không sẽ dẫn đến việc huỷ đồng bộ hóa cả 2 máy trạng thái một phần bằng cách từ chối sự hợp tác của chúng để tạo chữ ký tổng hợp cần thiết. Hãy lưu ý cảnh báo đã đề cập trước đó. Hơn nữa, bất kỳ bên trung thực nào cũng có thể ngăn chặn việc khởi tạo không chính

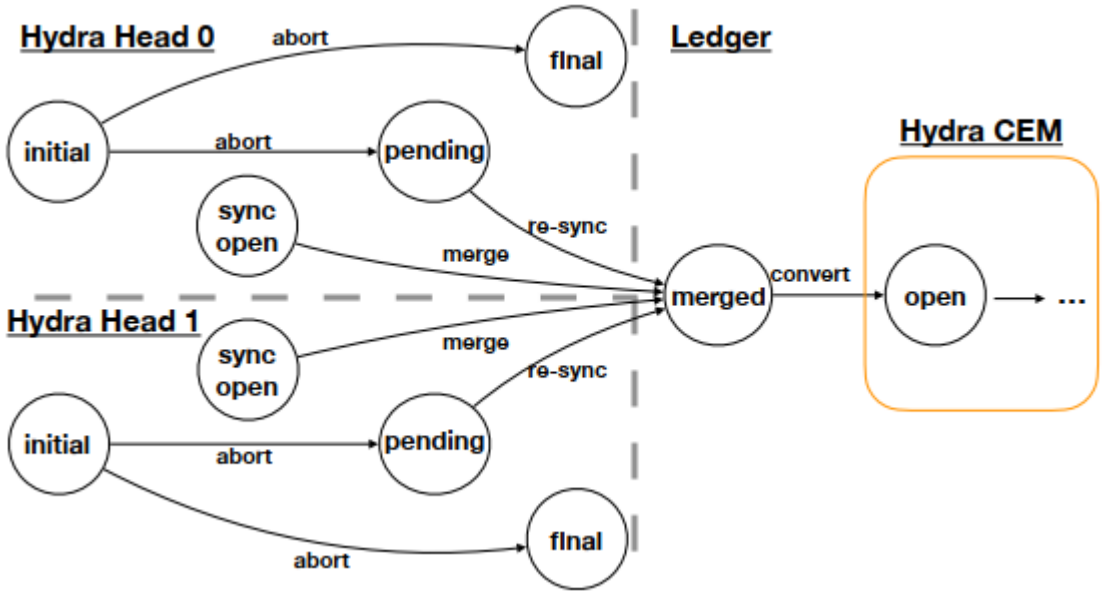
xác và đóng Head lạc quan bằng cách từ chối sự hợp tác của họ đối với các chữ ký tổng hợp được yêu cầu.

Máy trạng thái có thể đạt đến 2 trạng thái từ trạng thái **initial**: (1) Trạng thái **sync_open** đạt được thông qua đầu vào thu thập đồng bộ hóa (**sync-collect**). Bước này thu thập một bộ EUTxO E_b từ tất cả những người tham gia, cũng như một bộ EUTxO C_b từ các bên trung gian đóng vai trò là tài sản thế chấp. Chúng tôi yêu cầu tất cả EUTxO chỉ chứa Coin hoặc Token có thể thay thế được. Mặc dù các bên trung gian được phép cam kết số lượng tài sản thế chấp tùy ý, nhưng chúng tôi yêu cầu ràng buộc rằng tổng số Coin có trong C_b ít nhất phải bằng tổng số Coin có trong E_{1-b}, và số lượng và loại Token có thể thay thế trong E_b và C_b khớp nhau. Ngay khi máy trạng thái Interhead đạt đến trạng thái **sync_open** trên cả 2 Hydra Head, Hydra Head ảo sẽ được mở và các bên có thể sửa đổi trạng thái có thể thực thi của nó. Để đơn giản, chúng tôi bỏ qua việc hiển thị bộ sưu tập EUTxO trong Hình 2, nhưng các chi tiết được trình bày trong Phần 5. (2) Có thể đạt được trạng thái **final** bằng cách sử dụng lệnh hủy bỏ đồng bộ hóa (**sync-absort**) để hủy bỏ việc thực thi và giải phóng tất cả EUTxO đã cam kết trước đó.

Có thể đạt được trạng thái **pending** từ **sync-open** thông qua đầu vào **sync-close**. Quá trình chuyển đổi này yêu cầu gửi ảnh chụp nhanh một phần có chú thích cuối cùng của Head ảo. Ảnh chụp nhanh cuối cùng này phụ thuộc vào Hydra Head, nơi nó được gửi tới và chứa EUTxO của các thành viên cũng như tài sản thế chấp từ các bên trung gian. Ảnh chụp nhanh cuối cùng phải được thương lượng và xác nhận giữa tất cả các bên, nghĩa là bởi G^{ν} và G_i thông qua đa chữ ký tương ứng với các khóa xác minh K_{agg} và $K_{agg,i}$ tương ứng. Lưu ý rằng bước này tương tự như khái niệm đóng Head lạc quan trong Hydra. Mục đích của trạng thái **pending** là phát hiện nỗ lực của các trung gian bị hỏng nhằm hủy đồng bộ hóa các máy trạng thái một phần được thực thi trên mỗi Head. Chúng tôi thảo luận chi tiết trong Phần 6.

Cuối cùng, trạng thái **final** có thể đạt được thông qua đầu vào xác nhận đồng bộ hóa (**sync-confirm**) từ trạng thái **pending**, giải phóng EUTxO theo ảnh chụp nhanh đã thương lượng. Nếu đạt đến trạng thái **final** trong giai đoạn sắp xếp, Interhead vẫn ở trạng thái Off-Chain và kết thúc. Trong quá trình này, tất cả các EUTxO đã được người tham gia cam kết trước đây hoặc cam kết làm tài sản thế chấp của bên trung gian đều được giải phóng.

Mặt khác, nếu bất kỳ bên nào ngừng thực thi, không cộng tác hoặc nỗ lực không đồng bộ hóa được phát hiện, thì sẽ không có quá trình chuyển đổi nào nữa xảy ra trong giai đoạn sắp xếp và máy trạng thái sẽ chuyển sang giai đoạn chuyển đổi theo thời gian.

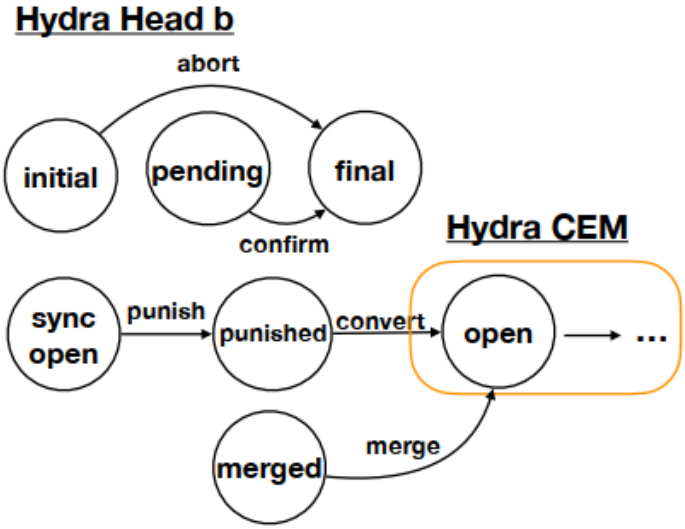


Hình 3: Chuyển đổi trạng thái trong giai đoạn chuyển đổi.

Giai đoạn chuyển đổi. Giai đoạn này được minh họa trong Hình 3. Tương tự như giai đoạn trước, tất cả các bên trung gian đều có thẩm quyền duy nhất để thực hiện chuyển đổi trạng thái, tuy nhiên, giờ đây bất kỳ bên trung gian nào cũng có thể thực hiện chuyển đổi trạng thái thay vì yêu cầu chữ ký tổng hợp. Tại thời điểm này, máy trạng thái có 3 kết quả. (1) Nếu Head chưa được mở và ít nhất một bên không gửi bất kỳ EUTxO nào thì Interhead có thể chuyển trực tiếp từ **initial** sang **final**. (2) Nếu không có bên trung gian (trung thực) nào thực hiện bất kỳ quá trình chuyển đổi trạng thái nào, máy trạng thái sẽ chuyển sang giai đoạn trừng phạt theo thời gian. (3) Trong tất cả các trường hợp khác, cả 2 máy trạng thái một phần sẽ được cam kết trên sổ cái và Hydra Head sẽ được mở để duy trì trạng thái có thể thực thi.

Một Hydra Head được mở như sau. Đầu tiên, tất cả các bên trung gian thực thi máy trạng thái trên sổ cái, điều này có thể được thực hiện thông qua việc ngừng cam kết tăng thêm của máy trạng thái hoặc bằng cách đóng Hydra Head. Điều này đòi hỏi thời gian lên đến $T^{\max}$. Sau đó, nếu tất cả các bên cam kết EUTxO vào sổ cái, máy trạng thái sẽ chuyển từ trạng thái **initial** sang trạng thái **pending** vì có khả năng xảy ra nỗ lực hủy đồng bộ hóa. Nếu cả 2 máy trạng thái một phần đạt đến trạng thái **pending** hoặc **sync-open** thì cả 2 luồng sẽ được hợp nhất thành một luồng thông qua trạng thái **merged** bằng cách sử dụng đầu vào hợp nhất (**merge**). Giao dịch thực hiện quá trình chuyển đổi này trả lại tài sản thế chấp cho tất cả các bên trung gian. Cuối cùng, chúng ta chuyển từ trạng thái **merged** sang trạng thái **open** của máy trạng thái Hydra thông qua đầu vào chuyển đổi (**convert**). Lưu ý rằng chúng tôi loại bỏ bất kỳ giới hạn thời gian nào đối với quá trình chuyển đổi từ trạng thái **merged** sang trạng thái **open**.

Giai đoạn trừng phạt. Giai đoạn cuối cùng được minh họa trong Hình 4. Nó có kết thúc mở và bất kỳ bên nào cũng có thể thực hiện chuyển đổi trạng thái. Nó có hai kết quả. Nếu máy trạng thái vẫn ở trạng thái **initial** hoặc **pending** thì máy trạng thái có thể được hủy bỏ một cách an toàn và chuyển sang trạng thái **final**. Mặt khác, máy trạng thái chuyển sang trạng thái mở của không gian trạng thái Hydra. Từ trạng thái **sync-open**, máy trạng thái chuyển sang trạng thái bị trừng phạt (**punished**) với đầu vào bị trừng phạt (**punish**). Trạng thái **punished** tương tự như trạng thái **merged** với một ngoại lệ. Các tài sản thế chấp trung gian không được phát hành. Thay vào đó, tài sản thế chấp được sử dụng để cung cấp đủ lượng Coin cho phép chuyển đổi sang trạng thái mở của Hydra. Cuối cùng, máy trạng thái cũng có thể chuyển từ trạng thái **merged** sang trạng thái **open** của Hydra trong giai đoạn này, trong trường hợp điều này chưa được thực hiện trong giai đoạn chuyển đổi.



Hình 4: Chuyển đổi trạng thái trong giai đoạn trừng phạt.

5 Xây dựng CEM

Trong phần tiếp theo, chúng tôi cung cấp thông tin chi tiết về CEM bằng cách mô tả từng trạng thái cũng như các ràng buộc do mỗi người xác minh cung cấp. Chúng tôi minh họa CEM bằng cách sử dụng các hình 6 - 14. Lưu ý rằng chúng tôi không minh họa tất cả các chuyển đổi có thể có vì một số tương tự với các chuyển đổi khác. Ngoài ra, do hạn chế về không gian, chúng tôi không xác định chính thức hành vi của từng người xác minh mà chỉ mô tả hành vi đó ở cấp độ cao.

5.1 Tham số

Các tham số mà theo đó Interhead CEM được thực hiện được thương lượng giữa tất cả những người tham gia và trung gian ngay từ đầu. Chúng tôi cấu trúc dữ liệu được

lưu trữ trong trạng thái CEM trong dữ liệu δ^v cần thiết để mở Head ảo trên sổ cái, dữ liệu δ_c chung cho hai CEM một phần trong cả H_0 và H_1 cũng như dữ liệu δ_b đó chỉ có liên quan trong mỗi Head H_b đơn lẻ.

Đầu tiên, δ^v là một Tuple có dạng (K_{agg}, η, h_{MT}, n, T, $\mathbf{cid}_0$, $\mathbf{cid}_1$) trong đó bốn tham số đầu tiên là trạng thái của Head ảo và $\mathbf{cid}_b$ là ID tiền tệ cho Thread Token $t_{s,b}$ trong Head H_b. Các tham số này không được tạo ra trong quá trình thực thi CEM, nhưng thể hiện cam kết từ những người tham gia Interhead để tạo một Head ảo với các tham số và Thread Token tương ứng. δ^v được sử dụng để đảm bảo rằng luôn tạo ra cùng một Head ảo, ngay cả khi chỉ có sẵn dữ liệu từ CEM một phần, đó là trường hợp khi CEM bước vào giai đoạn trừng phạt. ID tiền tệ được lưu trữ để đảm bảo rằng phiên bản CEM của Interhead là duy nhất và cả hai CEM một phần đều được liên kết với nhau, nghĩa là không thể hợp nhất CEM một phần nào với một phiên bản Interhead tương tự khác. Lưu ý rằng EUTxO chứa trong η không được phép chứa Token không thể thay thế duy nhất.

Thứ hai, δ_c bao gồm Tuple có dạng ($K_{agg,i}$, $h_{MT,i}$, n_i, T_o, T_c) chứa dữ liệu về các trung gian, tức là khóa xác minh tổng hợp của chúng $K_{agg,i}$, phần đầu của cây Merkel chứa tất cả các khóa xác minh riêng lẻ $h_{MT,i}$ và số lượng trung gian n_i. Hơn nữa, nó chứa các điểm trong thời gian T_o, $T_c \in \mathbb{N}$ xác định lần lượt kết thúc giai đoạn sắp xếp và thời gian chuyển đổi. Người xác minh đảm bảo rằng quá trình chuyển đổi diễn ra trong múi giờ có trật tự bằng cách đảm bảo rằng đối với khung thời gian của quá trình chuyển đổi [$r_{\min}$, $r_{\max}$] giữ cho $r_{\max} \leq T_o$. Tương tự, người xác minh đảm bảo rằng giao dịch nằm trong khung thời gian chuyển đổi bằng cách kiểm tra xem $T_0 < r_{\min} < r_{\max} \leq T_c$. Cuối cùng, quá trình chuyển đổi xảy ra trong giai đoạn trừng phạt nếu $r_{\min} > T_c$.

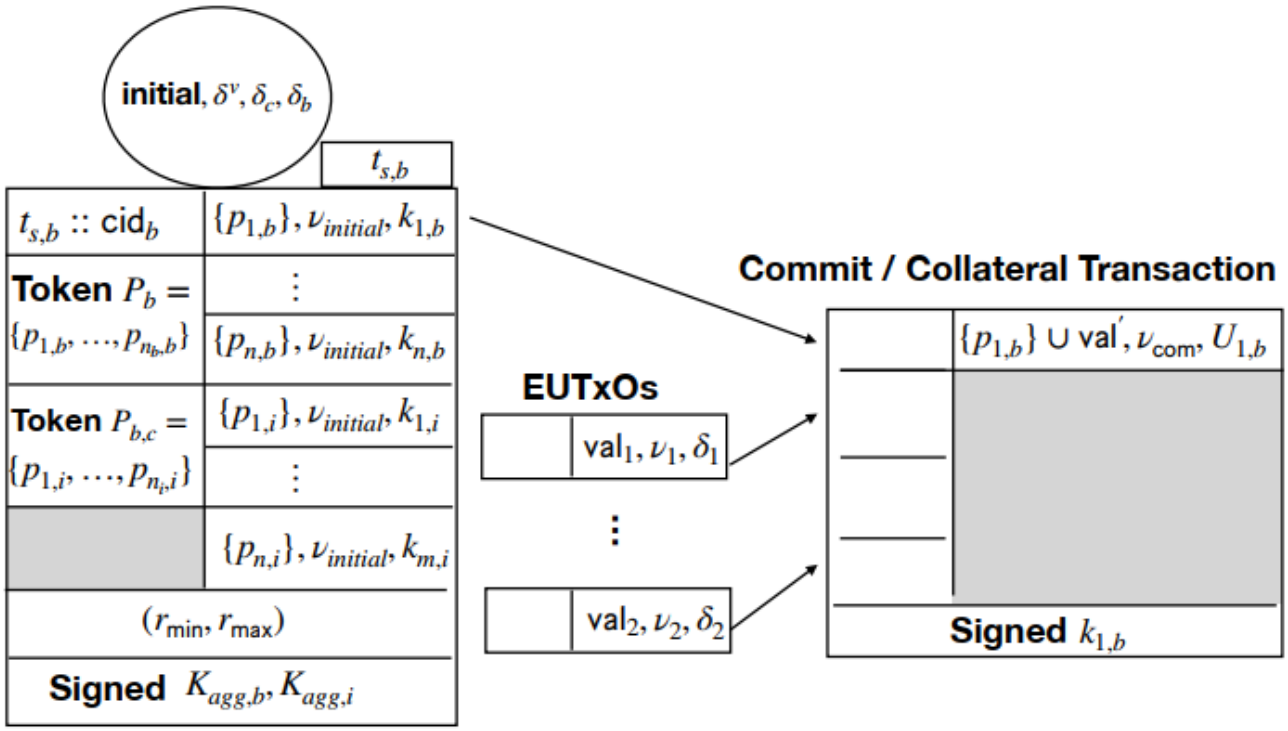
Cuối cùng, δ_b bao gồm Tuple có dạng (b, $K_{agg,b}$, η_b, $h_{MT,b}$, n_b, $\mathbf{col}_b$) trong đó $b \in \{0,1\}$ là một chút xác định thứ tự của cả hai CEM từng phần, $K_{agg,b}$ là khóa xác minh tổng hợp, η_b là cam kết của EUTxO rằng các bên sẽ chuyển sang Head ảo và nó phải giữ cho $\eta = \eta_0 \cup \eta_1$, $h_{MT,b}$ là gốc của cây Merkel bao gồm các khóa xác minh riêng của G_b, $n_b = |G_b|$ là số lượng người tham gia tham gia từ H_b, và $\mathbf{col}_b$ là tài sản thế chấp bắt buộc phải trả bởi các bên trung gian. Lưu ý rằng $\mathbf{col}_b$ ít nhất phải lớn bằng số lượng Coin có trong EUTxO trong η_{1-b}. Số lượng và loại Token có thể thay thế được gửi trong tài sản thế chấp phải khớp với số lượng Token do người tham gia gửi.

5.2 Giai đoạn sắp xếp

Trạng thái ban đầu. Trạng thái ban đầu được tạo ra ở cả 2 Head H_b với tham số δ^v, δ_c, δ_b. Mỗi người tham gia và người trung gian xác minh rằng tham số đã được thương lượng. Hơn nữa, người trung gian đảm bảo rằng cả hai trạng thái ban đầu đều khớp và có thể được chuyển đổi trong giai đoạn chuyển đổi trong trường hợp có tranh chấp. Giao dịch thiết lập trạng thái ban đầu được ký bằng cách sử dụng các khóa xác minh

tổng hợp của những người tham gia trong Head tương ứng cũng như các bên trung gian. Một bên chỉ ký giao dịch sau khi tích cực xác minh tính đúng đắn của nó. Trạng thái ban đầu yêu cầu một trạng thái mà Thread Token, cũng như Token tham gia được cung cấp dưới dạng Token tổng quát làm đầu vào. ID tiền tệ $\mathbf{cid}_b$ phải khớp với ID tiền tệ của Thread Token được cung cấp làm đầu vào. Chúng tôi yêu cầu một Token tham gia cho mỗi người tham gia và mỗi bên trung gian. Quá trình chuyển đổi tạo ra $n_b + n_i$ đầu ra riêng biệt, mỗi đầu ra chứa một Token tham gia và có thể được sử dụng bởi chính xác một người tham gia và người trung gian tương ứng.

Cam kết EUTxO. Tương tự như các Hydra Head, tất cả các bên tham gia và bên trung gian cam kết mỗi bên một bộ EUTxO cho Interhead, việc này được thực hiện song song. Một cam kết được hiển thị trong Hình 5. Mỗi người tham gia và người trung gian tạo một giao dịch *cam kết* sử dụng Token tham gia và một số EUTxO của họ trong đầu vào và lưu trữ thông tin về họ trong trạng thái $U_{i,b}$ của nó.

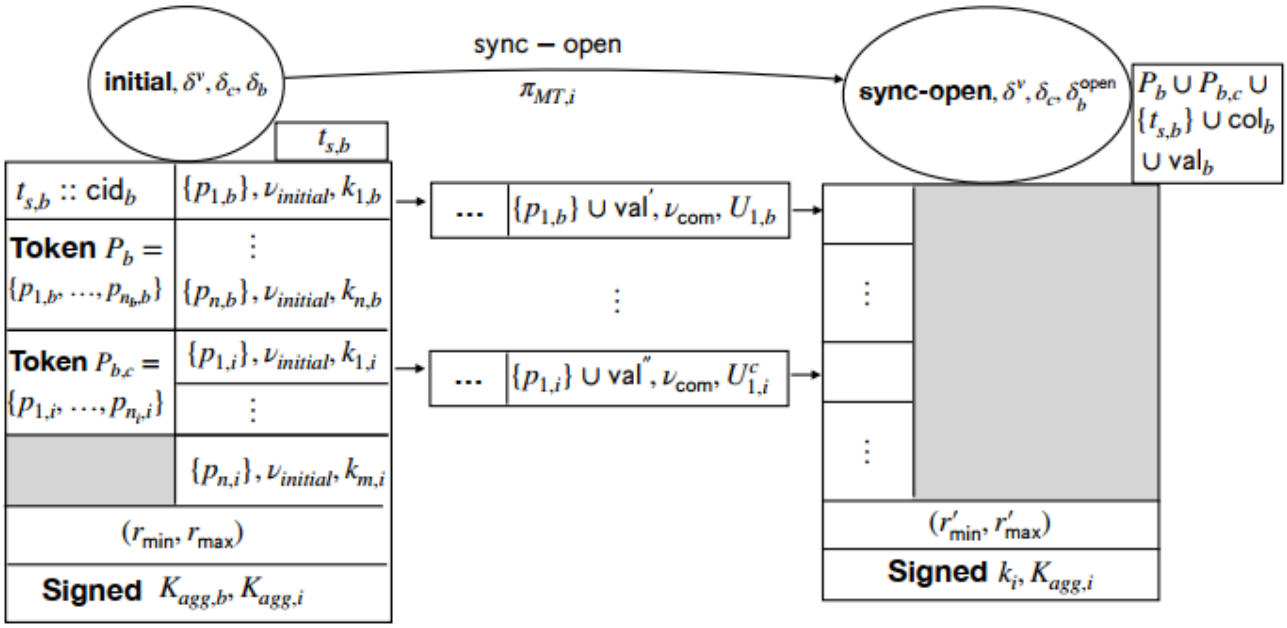


Hình 5: Cam kết của EUTxO và tài sản thế chấp vào Interhead.

Trạng thái sync-open. Nếu tất cả các bên đã tạo một giao dịch cam kết, giao dịch *sync-open* có thể sử dụng chúng như trong Hình 6. Trạng thái *sync-open* xác minh rằng tập hợp EUTxO $\eta = (U_{1,b}, \ldots, U_{nb,b})$ được cam kết bởi những người tham gia, phù hợp với cam kết ban đầu của họ, nghĩa là $\eta = \delta_b.\eta_b$. Ngoài ra, nó xác minh rằng tài sản thế chấp EUTxO $\eta_b^c = (U_{1,b}^c, \ldots, U_{ni,b}^c)$ được cam kết bởi các bên trung gian có chứa đủ số

lượng Coin và Token có thể thay thế được. Chúng tôi lưu trữ δ_b^{open} ở trạng thái CEMs bằng δ_b nhưng chúng tôi thay thế $\delta_b.\mathbf{col}_b$ với η_b^c.

Huỷ bỏ. Việc tạo Interhead có thể bị hủy bỏ bằng cách chuyển đổi từ trạng thái **initial** sang **final** như minh họa trong Hình 7. Trạng thái cuối cùng làm cho EUTxO đã được cam kết khả dụng thông qua các giao dịch phân tách (Split) tương tự như Hydra [4].
Trạng thái Pending. Trạng thái **pending** có thể đạt được từ quá trình **sync-open** và được hiển thị trong Hình 9. Trạng thái **pending** được sử dụng để tạo cơ hội cho các bên trung thực xác nhận việc đóng Head hoặc thay vào đó tiến hành giai đoạn chuyển đổi trong trường hợp nỗ lực huỷ đồng bộ hóa đã được phát hiện. Quá trình chuyển đổi yêu cầu một ảnh chụp nhanh có chú thích cuối cùng để chuyển đổi các bộ EUTxO η_b và η_b^c thành η_b'. Ảnh chụp nhanh cuối cùng η_b' chứa hai thành phần. Thứ nhất, nó chứa một phân vùng của EUTxO trong toàn bộ Interhead, cụ thể là EUTxO sẽ được tạo ra trong Head H_b khi đóng. Hơn nữa, nó chứa EUTxO trả lại tài sản thế chấp của bên trung gian. Lưu ý rằng số Coin được trả lại cho những người trung gian trong Head H_b có thể ít hơn số Coin mà những người trung gian đã trả khi mở Head, chẳng hạn, nếu những người tham gia từ H_{1-b} thực hiện thanh toán cho những người tham gia trong Head H_b. Tuy nhiên, vì số Coin trong CEM một phần là không đổi nên số Coin của người tham gia sẽ được lấy từ Coin được gửi làm tài sản thế chấp. Tuy nhiên, trong trường hợp đó, số Coin chênh lệch này sẽ có sẵn như một tài sản thế chấp bổ sung trong H_{1-b}. Mỗi người trung gian phải đảm bảo rằng tổng tài sản thế chấp được trả lại cho họ ở cả 2 Head bằng với tài sản thế chấp mà họ đã trả ban đầu để tạo ra Interhead.



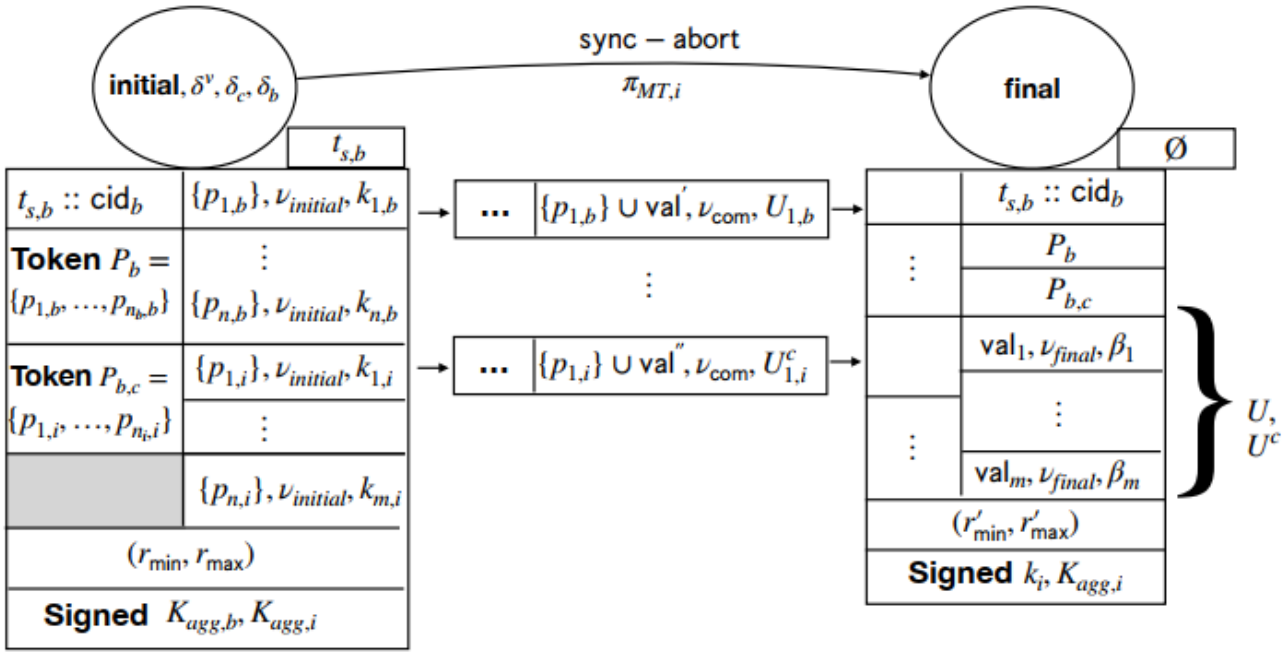
Hình 6: Chuyển đổi trạng thái **initial** sang **open** giới hạn trong giai đoạn sắp xếp.

Trạng thái final. Ngoài việc hủy bỏ việc mở Interhead, trạng thái **final** có thể đạt được từ trạng thái **pending** bằng cách xác nhận của các bên trung gian như trong Hình 10. Tương tự như trường hợp hủy bỏ, các bộ UTxO được cung cấp trong đầu ra của giao dịch. Tuy nhiên, lần này EUTxO khả dụng được lấy từ ảnh chụp nhanh cuối cùng η'_b.

Cung cấp EUTxO. Trạng thái cuối cùng phân vùng tất cả EUTxO trong đó mỗi phân vùng được sử dụng bởi một giao dịch phân tách như trong Hình 11 và được lấy từ Hydra CEM. Giao dịch phân tách chứa EUTxO trong phân vùng tương ứng trong các đầu ra của nó.

5.3 Giai đoạn chuyển đổi

Giai đoạn chuyển đổi có thể kết thúc theo hai cách. Thứ nhất, bất kỳ bên trung gian nào cũng có thể chuyển đổi Interhead CEM thành Hydra CEM thông thường. Điều này yêu cầu cả hai CEM từng phần được chuyển sang trạng thái có thể thực thi chung hoặc sổ cái tương ứng. Điều này có thể xảy ra bằng cách ngừng cam kết tăng thêm hoặc đóng các Hydra Head trong thời gian $T^{\max}$. Lưu ý rằng không có chuyển đổi trạng thái nào khác được phép trong Interhead CEM ngoại trừ chuyển đổi sang Hydra Head thông thường. Tất cả các quá trình chuyển đổi chỉ có thể được thực hiện bởi một bên trung gian, nhưng giờ đây không yêu cầu chữ ký tương ứng với khóa xác minh tổng hợp của bên trung gian $K_{\text{agg},i}$. Mặt khác, nếu việc chuyển đổi chưa được thực hiện, trong trường hợp không tồn tại trung gian trung thực, CEM sẽ chuyển sang giai đoạn trừng phạt.



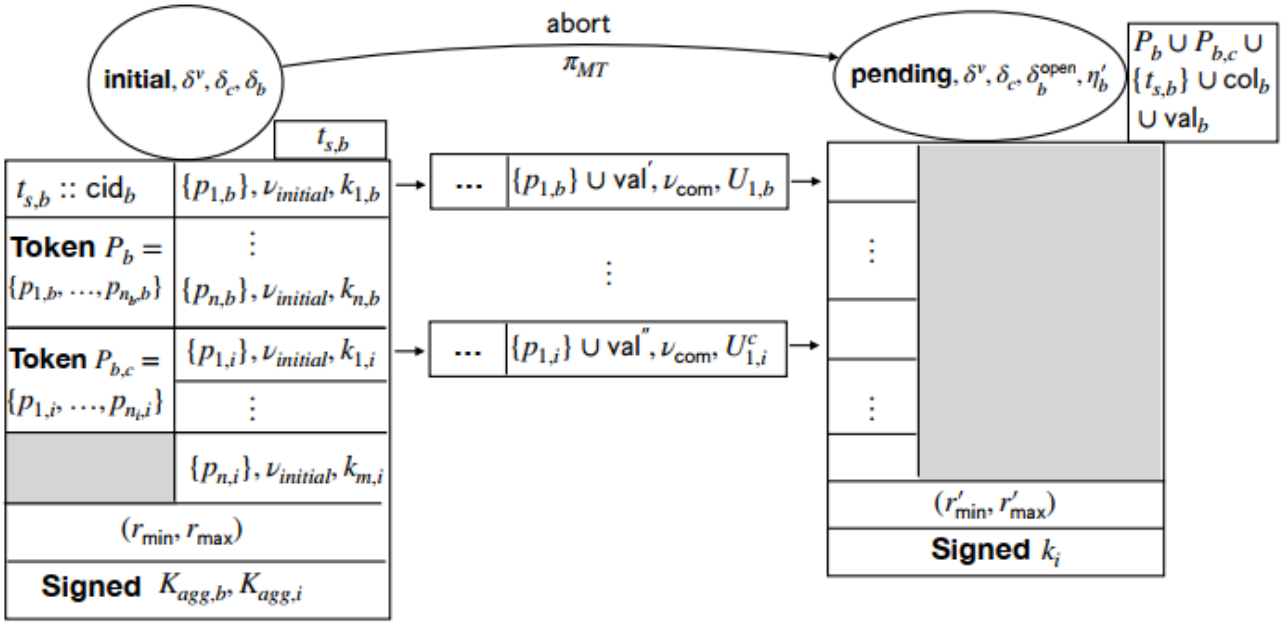
Hình 7: Hủy bỏ công trình trong giai đoạn sắp xếp.

Huỷ bỏ. Nếu có bất kỳ người tham gia hoặc trung gian nào chưa cam kết EUTxO với sổ cái thì việc mở Interhead có thể bị hủy bỏ tương tự như cách nó được thực hiện trong giai đoạn sắp xếp. Tuy nhiên, nếu tất cả người tham gia và người trung gian đã cam kết EUTxO thì việc hủy bỏ yêu cầu phải chuyển sang trạng thái **pending** vì nỗ lực hủy đồng bộ hóa có thể đã xảy ra. Lưu ý rằng nếu việc hủy bỏ được cho phép nhưng CEM đã chuyển sang trạng thái **pending** thì nó sẽ được thực hiện sau giai đoạn trừng phạt.

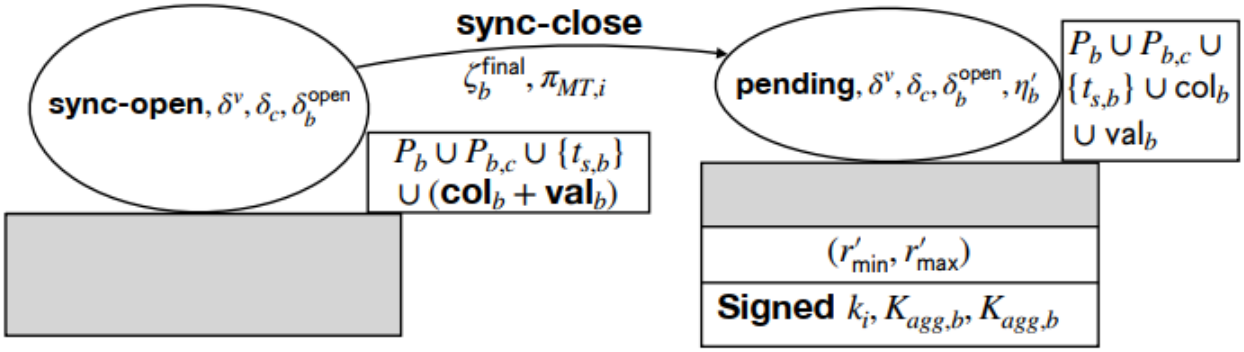
Trạng thái đã hợp nhất (Merged). Chuyển đổi sang CEM thông thường bằng trạng thái *Merged* như trong Hình 12. Có thể hợp nhất cả hai CEM một phần, từ trạng thái **sync-open** hoặc trạng thái **pending** tương tự và bất kỳ sự kết hợp nào của cả hai. Trạng thái hợp nhất yêu cầu phải có 2 Thread Token tổng quát và khớp với $\delta^{\nu}.\mathbf{cid}_0$ và $\delta^{\nu}.\mathbf{cid}_1$. Điều này đảm bảo rằng dự kiến chỉ các CEM môt phần mới có thể được hợp nhất vì cả 2 Thread Token không thể thay thế đều là duy nhất. Ngoài ra, người ta xác minh rằng các bộ EUTxO phù hợp với cam kết ban đầu, nghĩa là $\delta^{\nu}.\eta = \delta_0.\eta_0 \cup \delta_1.\eta_1$. Điều này đảm bảo rằng, trong trường hợp tất cả những người tham gia của một Head cũng như tất cả các bên trung gian bị hỏng, họ không thể thực hiện ít EUTxO hơn so với yêu cầu đối với Hydra Head ảo. Giao dịch giải phóng tất cả Token tổng quát trong đầu ra của nó, nhưng tương tự như trạng thái **initial**, nó lấy một Thread Token cũng như Token tham gia $\delta^{\nu}.n$ cho tất cả người tham gia trên cả 2 Head làm đầu vào. Cuối cùng, CEM có một đầu ra cho mỗi bộ EUTxO được các bên trung gian cam kết giải phóng tài sản thế chấp của họ giống như cách nó được giải phóng ở trạng thái **final**. Lưu ý rằng kể từ thời điểm này, chỉ thông tin trong δ^{ν} là bắt buộc và phần còn lại sẽ bị xóa khỏi trạng thái. Như được hiển thị trong Hình 13, bất kỳ người tham gia nào cũng có thể thực hiện chuyển đổi sang trạng thái *mở* của Hydra CEM thông thường. Chúng tôi không giới hạn quá trình chuyển đổi này đối với giai đoạn chuyển đổi vì nó có thể được thực hiện vô thời hạn sau khi bắt đầu giai đoạn chuyển đổi. Trạng thái được lấy trực tiếp từ dữ liệu trong δ^{ν}, tức là $K_{\text{agg}} = \delta^{\nu}.K_{\text{agg}}$, $\eta = \delta^{\nu}.\eta$, $h_{\text{MT}} = \delta^{\nu}.h_{\text{MT}}$, $n = \delta^{\nu}.n$, $T = \delta^{\nu}.T$.

5.4 Giai đoạn trừng phạt

Mục đích của giai đoạn cuối cùng là cho phép mọi bên trung thực mở Hydra Head ảo giữa tất cả những người tham gia, ngay cả trong trường hợp tất cả các bên khác đều bị hỏng. Interhead CEM chuyển đổi thành Hydra CEM mà không cần hợp nhất cả hai CEM từng phần. Để đảm bảo điều này có thể thực hiện được, Coin và Token có thể thay thế cần thiết để mở Hydra Head được lấy từ tài sản thế chấp của những người trung gian, những người sẽ làm mất nó trong quá trình này. Tất cả chuyển đổi trong giai đoạn này có thể được thực hiện bởi bất kỳ người tham gia hoặc trung gian nào.

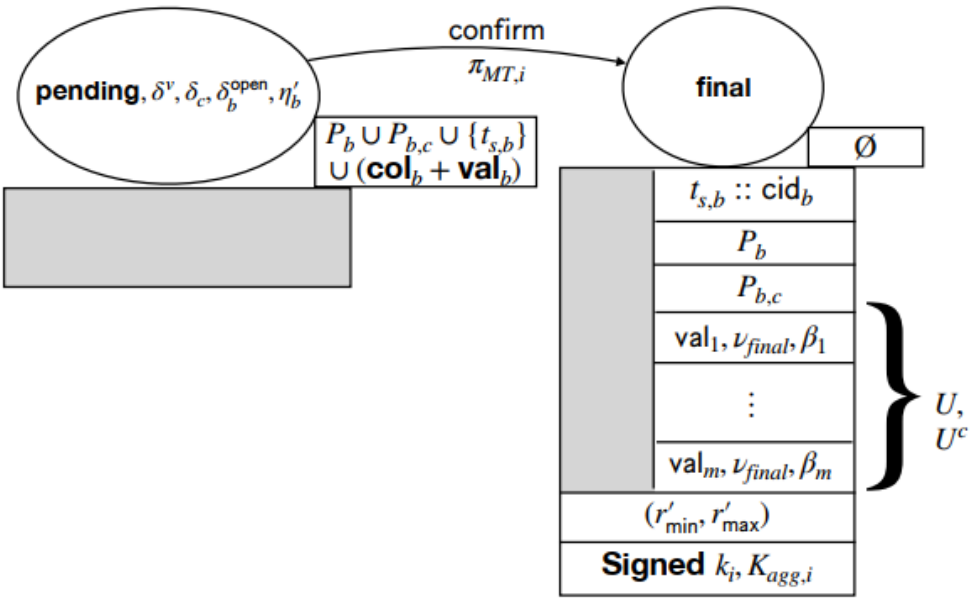


Hình 8: Hủy bỏ trong giai đoạn chuyển đổi nếu tất cả EUTxO và tài sản thế chấp đã được cam kết.



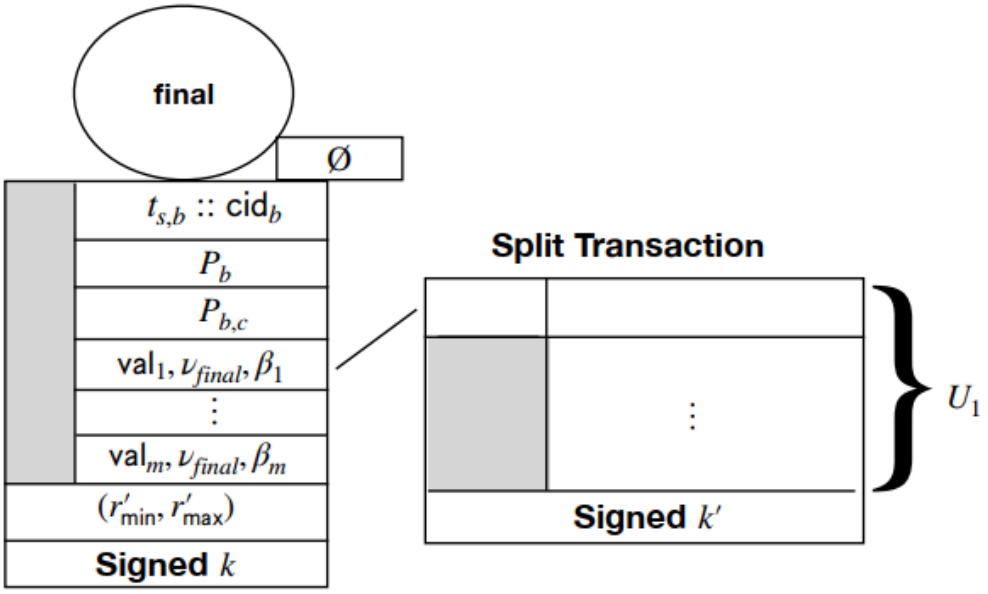
Hình 9: Bước đầu tiên của việc đóng lạc quan trong giai đoạn sắp xếp.

Trạng thái trừng phạt (Punished). Nếu CEM ở trạng thái **sync-open** khi bước vào giai đoạn trừng phạt, CEM sẽ chuyển thành Hydra CEM thông thường thông qua trạng thái trừng phạt (**punish**) như trong Hình 14. Việc chuyển đổi này có thể được thực hiện trong cùng một trạng thái có thể thực thi, trong đó CEM một phần được thực thi nên không yêu cầu ngừng cam kết tăng thêm và không đóng Hydra Head. Quá trình chuyển đổi từ và sang trạng thái **punish** tương tự như chuyển đổi sang và từ trạng thái **merged**, ngoại trừ việc chúng tôi chỉ xác minh tính chính xác của dữ liệu từ Hydra Head cục bộ, tức là $\delta_b.\eta$ và δ^v.**cid**. Chúng tôi sử dụng lại Thread Token hiện tại $t_{s,b}$ cho Hydra CEM. Tài sản thế chấp của các bên trung gian không được cung cấp thông qua đầu ra của giao dịch mà được sử dụng để tài trợ cho việc chuyển đổi sang trạng thái **open**.



Hình 10: Xác nhận đóng hoặc huỷ bỏ Head. Chỉ yêu cầu chữ ký tổng hợp trong giai đoạn sắp xếp.

Kết thúc mở. Nếu CEM một phần ở trạng thái **initial**, nó có thể bị hủy bỏ khi chuyển sang trạng thái **final**. Hơn nữa, nếu CEM ở trạng thái **pending** thì giờ đây nó có thể chuyển sang trạng thái **final** một cách an toàn vì không xảy ra hiện tượng huỷ đồng bộ hóa. Cuối cùng, quá trình chuyển đổi từ trạng thái **merged** sang trạng thái **open** của Hydra CEM đã kết thúc mở nên cũng có thể được thực hiện trong giai đoạn trừng phạt.



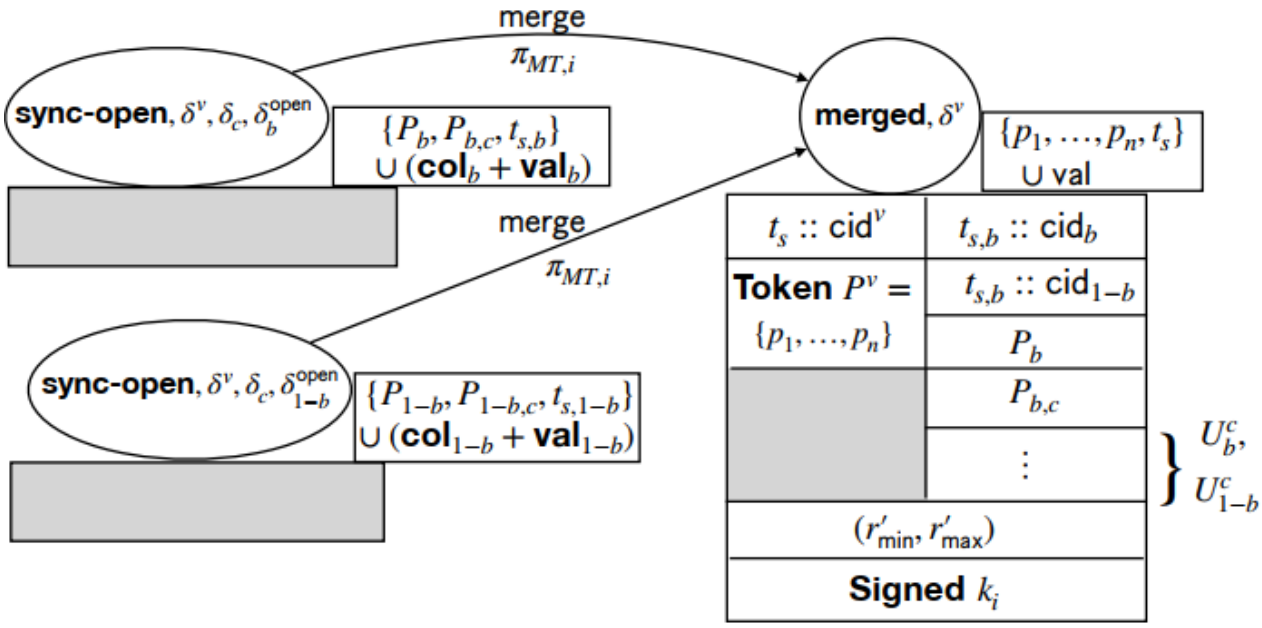
Hình 11: Các giao dịch phân tách làm cho tất cả EUTxO và tài sản thế chấp có sẵn trong đầu ra của chúng.

6 Các giao thức

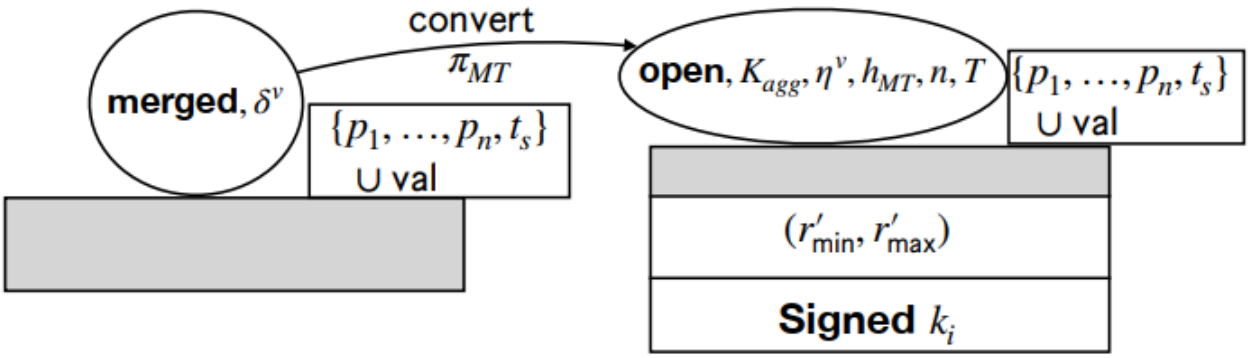
Chúng tôi xác định hành vi của các bên trung thực dưới dạng một giao thức, được chia thành ba giao thức giao thức phụ. Giao thức khởi tạo (**initialization**) được thực hiện ngay từ đầu, giao thức *đồng bộ hóa* (Synchronization) được thực hiện bởi các trung gian và giao thức *đóng lạc quan* (Optimistic Closure) được thực hiện để giải quyết máy trạng thái trong giai đoạn sắp xếp.

Giao thức khởi tạo. Lúc đầu, tham số của trạng thái **initial** được thương lượng giữa tất cả những người tham gia và bên trung gian. Bên trung gian cần xác minh rằng cả hai trạng thái **initial** đều khớp và có thể được hợp nhất trong giai đoạn chuyển đổi. Vì lý do này, chúng tôi yêu cầu trạng thái **initial** chứa các chữ ký tổng hợp tương ứng với cả hai khóa xác minh K_{agg} và $K_{agg,i}$. Bất kỳ bên nào cung cấp chữ ký của họ cho điều này chỉ trong trường hợp xác minh tích cực về trạng thái **initial** có liên quan.

Giao thức đồng bộ hóa. Mặc dù máy trạng thái đã bắt buộc rằng tất cả các chuyển đổi trong giai đoạn sắp xếp đều cần có sự đồng ý của tất cả các trung gian bằng cách yêu cầu chữ ký tổng hợp tương ứng với khóa xác minh $K_{agg,i}$, giao thức đồng bộ hóa mô tả tình huống mà một trung gian cung cấp chữ ký của họ để phê duyệt một chuyển đổi trạng thái nhất định. Việc chuyển trạng thái được một bên chấp thuận theo 2 điều kiện.



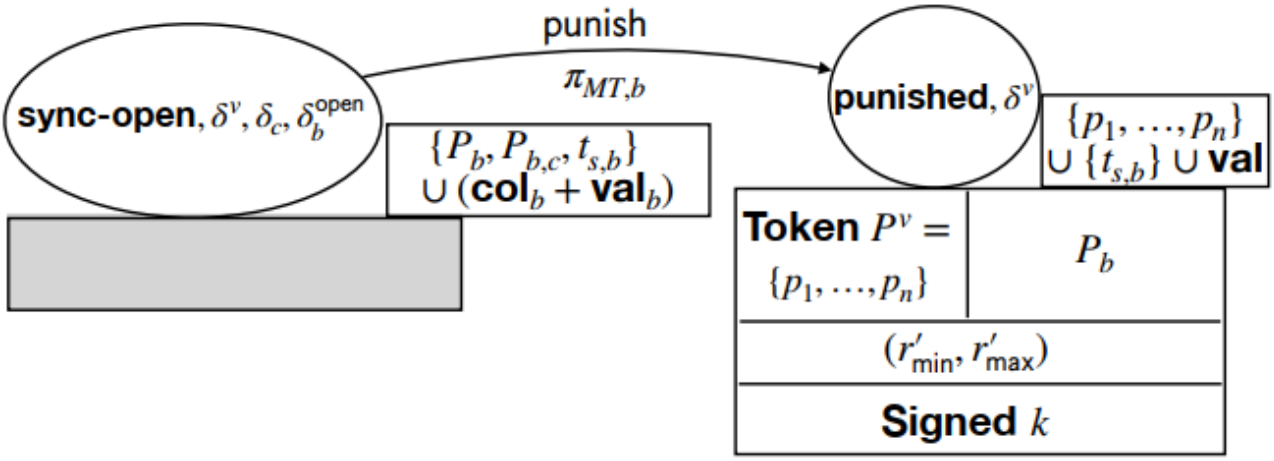
Hình 12: Hợp nhất cả hai CEM một phần trong cùng một trạng thái có thể thực thi hoặc sổ cái. Quá trình chuyển đổi có thể được thực hiện từ bất kỳ sự kết hợp nào giữa các trạng thái **pending** và **sync-open**. Tài sản thế chấp của bên trung gian được mở khóa.



Hình 13: Chuyển đổi thành Hydra Head thông thường. Trạng thái của Head có thể được suy ra trực tiếp từ δ^v. Quá trình chuyển đổi bắt nguồn từ trạng thái **punished** là tương tự.

(1) Tất cả các đầu vào còn lại ngoại trừ chữ ký tổng hợp để thực hiện quá trình chuyển đổi trên cả 2 Head đều được bên đó biết đến. (2) Nếu quá trình thu thập đồng bộ hoá (**sync-collect**) chuyển đổi trạng thái đã được phê duyệt, thì quá trình chuyển đổi trạng thái huỷ bỏ đồng bộ hoá (**sync-abort**) không thể được phê duyệt và ngược lại. Lưu ý rằng điều này thể hiện rằng tất cả các bên đã gửi EUTxO tương ứng với cam kết $\delta^v.\eta$ và tất cả các bên trung gian đã cam kết đủ tài sản thế chấp. Điều này là bắt buộc để cho phép Interhead chuyển sang trạng thái **merged** nếu cần. Lưu ý rằng các trung gian tạo hai chữ ký tổng hợp, mỗi máy trạng thái một phần một chữ ký.

Giao thức này cho phép chuyển đổi trạng thái được thực hiện trên cả hai máy trạng thái Interhead một phần hoặc không có máy nào. Tuy nhiên, có một lưu ý. Một trung gian bị hỏng có thể đợi cho đến khi chữ ký của nó là tất cả những gì còn lại cho hai chữ ký tổng hợp. Sau đó, họ chỉ hoàn thành một trong các chữ ký tổng hợp. Nếu điều này được thực hiện đối với chuyển đổi **sync-absort** hoặc xác nhận đồng bộ hóa (**sync-confirm**), điều này dẫn đến việc một Head đóng sớm hơn Head còn lại, điều này không quan trọng vì nó không ảnh hưởng đến tính bảo mật của hệ thống. Tuy nhiên, nếu nó được thực hiện cho quá trình chuyển đổi **sync-collect** hoặc đóng đồng bộ hóa (**sync-close**), điều này có thể khiến các bên trung gian trung thực mất tài sản thế chấp của họ. Do đó, nếu điều này *có thể* xảy ra, máy trạng thái yêu cầu chuyển sang trạng thái **pending**. Trạng thái **pending** có thể được duy trì thông qua sự đồng ý của tất cả các bên trung gian thông qua quá trình chuyển đổi **sync-confirm**, tuy nhiên, nếu bất kỳ bên trung gian nào phát hiện ra nỗ lực huỷ đồng bộ hóa này, họ sẽ không hợp tác để CEM chuyển sang giai đoạn chuyển đổi. Cuối cùng, nếu máy trạng thái Interhead tiến hành giai đoạn chuyển đổi, mỗi bên trung gian trung thực sẽ tiến hành tích cực cam kết máy trạng thái Interhead trên sổ cái, có khả năng bằng cách đóng các Hydra Head và thực hiện chuyển đổi trạng thái cho đến khi đạt được trạng thái **merged**.



Hình 14: Chuyển sang trạng thái **punished**. Thread Token được sử dụng lại do đó không được cung cấp làm đầu vào. Quá trình chuyển đổi sang trạng thái **open** của Hydra CEM tương tự như khi bắt nguồn từ trạng thái **merged**.

Giao thức đóng lạc quan. Để đóng máy trạng thái Interhead trong giai đoạn sắp xếp, chúng tôi tiến hành tương tự như việc đóng Head lạc quan của Hydra. Tất cả các bên thương lượng hai ảnh chụp nhanh một phần η_0, η_1 cho cả 2 Head H_0 và H_1. Các ảnh chụp nhanh này đại diện cho hai phân vùng của tất cả EUTxO trong trạng thái có thể thực thi của Interhead. (1) η_b chứa các EUTxO sẽ được giải phóng trong Head H_b cũng như (2) tài sản thế chấp sẽ được thanh toán cho các bên trung gian trong H_b. Sau khi thương lượng, tất cả các bên cung cấp chữ ký tổng hợp của (**final**, η_b) tương ứng với các khóa xác minh $K_{\text{agg},b}$ và $K_{\text{agg,i}}$.

Việc EUTxO được phân vùng như thế nào là hoàn toàn miễn phí đối với mỗi bên, tuy nhiên, có một số điều phải được xem xét bởi mỗi bên. Người tham gia là thành viên của Head H_b nhưng không phải của H_{1-b} nên thương lượng rằng tất cả EUTxO đại diện cho các khoản thanh toán cho chính họ phải ở trong η_b. Tương tự, trạng thái EUTxO đại diện cho các máy trạng thái mà nhóm tham gia cũng phải ở trong η_b. Điều này là do nếu tất cả các bên trong H_{1-b} bị hỏng thì tất cả EUTxO trong η_{1-b} đều có khả năng bị mất. Hơn nữa, mỗi bên trung gian phải đảm bảo tổng số tài sản thế chấp mà họ nhận được trong cả hai phân vùng bằng với tổng số tài sản thế chấp mà họ đã trả vào CEM. Lưu ý rằng số lượng tài sản thế chấp được trả cho người trung gian sẽ thay đổi trong mỗi Head tùy thuộc vào cách EUTxO được phân chia do tổng số Coin trong ảnh chụp nhanh trong η_b không thể thay đổi so với số Coin mà người tham gia và người trung gian cam kết trong H_b. Tuy nhiên, tổng số Coin được trả lại cho các trung gian dưới dạng tài sản thế chấp được giải phóng sẽ không đổi trên cả 2 Head H_0 và H_1.

Bất kỳ bên trung thực nào cũng phải tuân thủ hành vi này trong quá trình đàm phán vì nếu không thì các thuộc tính bảo mật trong Mục 4.2 sẽ không được đảm bảo.

7 Bằng chứng bảo mật

Trong phần tiếp theo, chúng tôi cung cấp các tuyên bố bảo mật cho cấu trúc Interhead, bao gồm máy trạng thái Interhead và các giao thức Interhead. Điều này tương ứng với các thuộc tính mong muốn trong Phần 4.2.

Bổ đề 1 (Kết cuộc). *Nếu ít nhất một bên là trung thực thì Interhead CEM cuối cùng sẽ đạt đến trạng thái cuối cùng hoặc chuyển sang không gian trạng thái Hydra CEM.*

Chứng minh. Có 2 trường hợp. Đầu tiên, nếu tất cả các bên hợp tác và không hành xử ác ý thì Interhead CEM có thể đạt đến trạng thái cuối cùng qua 2 đường dẫn trong giai đoạn sắp xếp, tức là thông qua hủy bỏ hoặc thông qua đóng Head lạc quan. Nếu điều này không được thực hiện, bên trung thực sẽ đợi cho đến khi đủ thời gian để CEM bước vào giai đoạn trừng phạt, điều này cuối cùng sẽ xảy ra. Việc chuyển đổi trạng thái trong giai đoạn trừng phạt có yêu cầu duy nhất là chứng minh rằng bên đó là trung gian hoặc người tham gia của Head ảo, điều này có thể được thực hiện bởi bên trung thực. Có một đường dẫn của tất cả các trạng thái trong không gian trạng thái của CEM dẫn đến trạng thái cuối cùng hoặc vào không gian trạng thái Hydra CEM.

Định lý 1 (Tính sống động của tài sản thế chấp). *Nếu ít nhất một bên trung gian là trung thực thì cấu trúc Interhead có thuộc tính tính sống động của tài sản thế chấp.*

Chứng minh. Hãy lưu ý có 2 trạng thái bắt buộc rằng mọi tài sản thế chấp đã được cam kết trước đó sẽ được cung cấp trong bất kỳ trạng thái có thể thực thi nào. Hai trạng thái này là **final** và **merged**. Như trong Bổ đề 1, cuối cùng thì Interhead CEM hoặc đạt đến trạng thái cuối cùng hoặc không gian trạng thái Hydra CEM. Chỉ có một đường chạy qua Interhead CEM có thể đi mà không kết thúc ở trạng thái **final** hoặc chứa trạng thái **merged**. Đó là khi Hydra CEM có một lần chạy chứa trạng thái **punished**.

Do đó, những gì còn lại để chỉ ra là bất kỳ người trung gian trung thực P có thể thực thi một lần chạy không chứa trạng thái **punished**. Chúng tôi quan sát thấy rằng trạng thái **punished** chỉ có thể đạt được từ trạng thái **sync-open** và chỉ trong giai đoạn trừng phạt. Sau đây, chúng tôi giả sử rằng có hai CEM một phần, I_b, $b \in \mathbb{N}$ và CEM I_b ở trạng thái **sync-open**. Trong phần tiếp theo, chúng tôi lập luận về các trạng thái tiềm năng của CEM một phần I_{1-b}.

Đầu tiên, I_{1-b} có thể ở trạng thái **initial** hoặc trạng thái **pending**. Đạt đến trạng thái **sync-open** trạng thái **intial** yêu cầu chữ ký tổng hợp của nhóm trung gian cần có sự cộng tác của P. Vì P trung thực, họ chỉ hợp tác với việc tạo chữ ký tổng hợp nếu tất cả các bên cam kết EUTxO nhất quán với cam kết $\delta^{v}.\eta$ và tất cả các bên trung gian cam kết đủ tài sản thế chấp. Hơn nữa, khi bắt đầu giao thức P đã xác nhận rằng cả hai trạng thái ban đầu của các Interhead một phần đều nhất quán, đặc biệt chứa các giá trị giống nhau cho δ^{v}. Vì tất cả các bên đã cam kết giao dịch và tài sản thế chấp, Interhead ngăn

quá trình chuyển đổi từ trạng thái **initial** sang trạng thái **final** trước giai đoạn trừng phạt. Tuy nhiên, P có thể thực thi chuyển đổi từ trạng thái **initial** sang trạng thái **pending** bằng cách cung cấp bằng chứng họ là trung gian trong giai đoạn chuyển đổi.

Thứ hai, I_{1-b} có thể ở trạng thái **sync-open** và vẫn ở đó hoặc chuyển sang trạng thái **pending** trong giai đoạn sắp xếp. Tuy nhiên, I_{1-b} không thể đạt đến trạng thái **final** từ trạng thái **pending** trong giai đoạn sắp xếp, vì điều này yêu cầu chữ ký tổng hợp từ các trung gian yêu cầu sự cộng tác từ P, người chỉ cộng tác nếu cả hai CEM một phần đều ở trạng thái **pending**. Ngoài ra, không có quá trình chuyển đổi từ trạng thái **pending** sang trạng thái **final** trong giai đoạn chuyển đổi. Do đó, I_b không thể chuyển sang trạng thái **final** từ trạng thái **pending** trước giai đoạn trừng phạt.

Tóm lại, CEM một phần I_{1-b} ở trạng thái **pending** hoặc **sync-open** hoặc có thể được P đưa vào trạng thái **pending** trong giai đoạn chuyển đổi.

Do đó, ngay khi Interhead CEM đạt đến giai đoạn chuyển đổi, P sẽ đóng cả 2 Hydra Head và thực hiện chuyển đổi I_{1-b} sang trạng thái **pending** nếu cần. Vì cả 2 Interhead một phần đều ở trong cùng một trạng thái có thể thực thi, P có thể thực hiện quá trình chuyển đổi sang trạng thái **merged** để ngăn bất kỳ CEM một phần nào chuyển sang trạng thái **punished**. Điều này yêu cầu nhiều nhất thời gian $t = \max(H_b.T^{\max}, H_{1-b}.T^{\max} + 2\Delta)$. Vì nó giữ cho trong khoảng thời gian của giai đoạn sắp xếp $t_{C,\text{end}} - t_{C,\text{start}} > \max(Hb.T^{\max}, H_{1-b}.T^{\max} + 2\Delta) = t$, P trung gian trung thực có thể thực hiện quá trình chuyển đổi sang trạng thái **merged** trước khi giai đoạn trừng phạt bắt đầu.

Định lý 2 (Tính sống động của EUTxO). *Nếu ít nhất một bên là trung thực thì cấu trúc Interhead có thuộc tính tính sống động của EUTxO.*

Chứng minh. Do Bổ đề 1, chúng tôi có 2 trường hợp cần xem xét đối với bất kỳ bên trung thực P nào. CEM đạt đến trạng thái cuối cùng hoặc đạt đến không gian trạng thái Hydra CEM. Trong trường hợp sau, chúng tôi đã hoàn thành. Trong trường hợp trước chúng tôi có 2 trường hợp. Đầu tiên, CEM có thể hủy bỏ việc mở khóa tất cả EUTxO đã cam kết trước đó trong trường hợp chúng tôi kết thúc. Mặt khác, trạng thái cuối cùng đạt được thông qua trạng thái **sync-open** và trạng thái **pending** thông qua đóng lạc quan. Trong trường hợp đó, EUTxO được mở khóa phụ thuộc vào sự thương lượng trong giao thức đóng lạc quan. Cần có sự cộng tác của một đa chữ ký tổng hợp của P để thực hiện đóng lạc quan. Nếu P là thành viên trong Head H_b, nó xác minh tất cả EUTxO mà nó liên quan đều có mặt trong phân vùng ảnh chụp nhanh η_b. Nếu P là thành viên trong cả 2 Hydra Head, nó đảm bảo rằng tất cả EUTxO mà nó liên quan đều nằm trong một trong hai phân vùng ảnh chụp nhanh. Trong cả hai trường hợp, tất cả EUTxO trong các phân vùng đều có sẵn trong các Hydra Head mà P là thành viên.

Định lý 3 (Bảo mật số dư). *Nếu ít nhất một bên là trung thực thì cấu trúc Interhead có thuộc tính bảo mật số dư.*
Chứng minh. Điều này được suy ra trực tiếp từ Định lý 1 và Định lý 2.

8 Kết luận

Trong công việc này, chúng tôi trình bày cấu trúc Interhead, một cách tiếp cận để tạo các Hydra Head ảo cho phép giao tiếp vượt ra ngoài các khoản thanh toán đơn giản mà thay vào đó cho phép thực thi các máy trạng thái tùy ý giữa những người tham gia trên mạng các Hydra Head. Chúng tôi xác định các thuộc tính bảo mật tính sống động của tài sản thế chấp, tính sống động của EUTxO và bảo mật số dư và chứng minh chúng khi có kẻ tấn công độc hại. Chúng tôi trình bày cấu trúc kênh ảo đầu tiên hỗ trợ các kênh có số lượng bên tùy ý và tài sản thế chấp đó được đóng góp bởi nhiều bên trung gian. Do đó, cấu trúc của chúng tôi thu hẹp khoảng cách giữa các giao thức Layer 2 dựa trên Hydra Head và các kênh thanh toán hoặc kênh trạng thái.

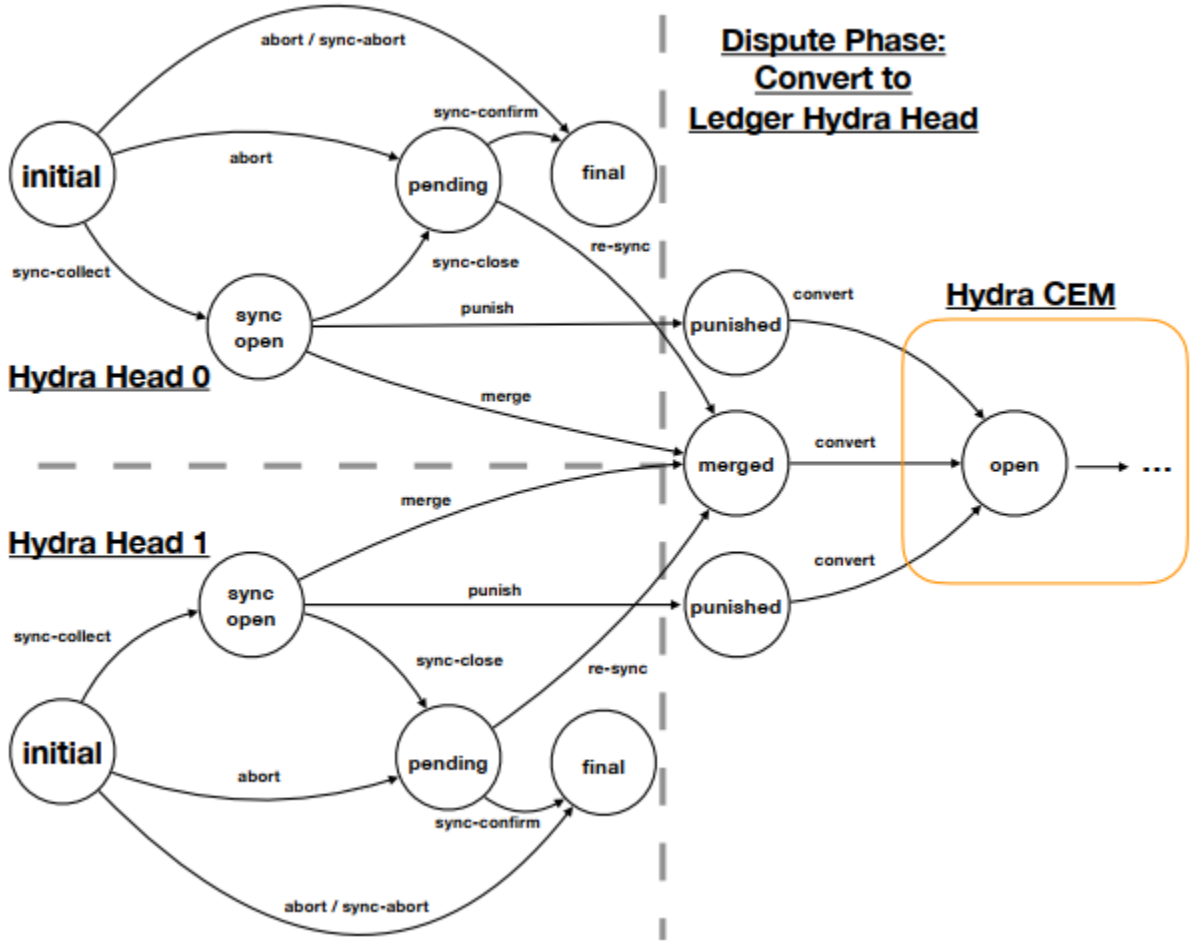
Tài liệu tham khảo

1. Canetti, R.: Chữ ký, chứng nhận và xác thực có thể kết hợp được trên toàn cầu. Trong: Kỷ yếu. Hội thảo về Cơ sở Bảo mật Máy tính của IEEE lần thứ 17, 2004. Trang 219–233. IEEE (2004)
2. Chakravarty, MM, Chapman, J., MacKenzie, K., Melkonian, O., Jones, MP, Wadler, P.: Mô hình UTxO mở rộng. Trong: Hội thảo lần thứ 4 về Hợp đồng thông minh đáng tin cậy (2020)
3. Chakravarty, MM, Chapman, J., MacKenzie, K., Melkonian, O., Muller, J., Jones, MP, Vinogradova, P., Wadler, P.: Token tùy chỉnh gốc trong mô hình UTxO mở rộng. Trong: Hội nghị chuyên đề quốc tế về tận dụng các ứng dụng của các phương pháp hình thức. Trang 89–111. Springer (2020)
4. Chakravarty, MM, Coretti, S., Fitzi, M., Gazi, P., Kant, P., Kiayias, A., Russell, A.: Hydra: Các kênh trạng thái đẳng cấu nhanh. Trong: Hội nghị quốc tế về mật mã tài chính và bảo mật dữ liệu. Springer (2021)
5. Croman, K., Decker, C., Eyal, I., Gencer, AE, Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, EG và cộng sự: Mở rộng quy mô Blockchain phi tập trung. Trong: Hội nghị quốc tế về mật mã tài chính và bảo mật dữ liệu. Trang 106–125. Springer (2016)
6. Decker, C., Wattenhofer, R.: Mạng lưới thanh toán nhanh và có thể mở rộng với các kênh thanh toán vi mô song công Bitcoin. Trong: Hội nghị chuyên đề về các hệ thống tự ổn định. Trang 3–18. Springer (2015)
7. Dziembowski, S., Eckey, L., Faust, S., Malinowski, D.: Perun: Trung tâm thanh toán ảo trên Crypto. Trong: Perun: Trung tâm thanh toán ảo qua Crypto. IEEE (2017)

8. Dziembowski, S., Faust, S., Hostáková, K.: Các mạng lưới kênh trạng thái chung. Trong: Kỷ yếu Hội nghị ACM SIGSAC 2018 về Bảo mật Máy tính và Truyền thông. Trang 949–966. ACM (2018)
9. Egger, C., Moreno-Sanchez, P., Maffei, M.: Cập nhật đa kênh nguyên tử với tài sản thế chấp liên tục trong các mạng lưới kênh thanh toán tương thích Bitcoin. Trong: Cavallaro, L., Kinder, J.,Wang, X., Katz, J.(eds.) ACM CCS 2019. Trang 801–815. Báo chí ACM (tháng 11 năm 2019). https://doi.org/10.1145/3319535.3345666
10. EthHub: Sidechains (2021), https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/sidechains/
11. Itakura, K., Nakamura, K.: Hệ thống mật mã khóa công khai phù hợp với đa chữ ký kỹ thuật số. Nghiên cứu & Phát triển NEC (71), 1–8 (1983)
12. Jourenko, M., Larangeira, M., Tanaka, K.: Các kênh thanh toán ảo nhẹ. Cryptology ePrint Archive, Report 2020/998 (2020), https://eprint.iacr.org/2020/998
13. Jourenko, M., Larangeira, M., Tanaka, K.: Cây thanh toán: Khoản thanh toán thế chấp thấp cho các mạng kênh thanh toán. Trong: Hội nghị quốc tế về mật mã tài chính và bảo mật dữ liệu. Springer (2021)
14. Katz, J., Maurer, U., Tackmann, B., Zikas, V.: Tính toán đồng bộ có thể tổng hợp toàn cầu. Trong: Hội nghị lý thuyết mật mã. trang 477–498. Springer (2013)
15. Kiayias, A., Litos, OST: Một phương pháp xử lý bảo mật có thể kết hợp được cho mạng Lightning. IACR Cryptology ePrint Archive **2019**, 778 (2019)
16. Kiayias, A., Zhou, HS, Zikas, V.: Tính toán đa bên công bằng và hiệu quả bằng cách sử dụng sổ cái giao dịch toàn cầu. Trong: Fischlin, M., Coron, JS (eds.) Những tiến bộ trong Mật mã học – EUROCRYPT 2016. Trang 705–734. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
17. Mealy, GH: Một phương pháp tổng hợp các mạch tuần tự. Tạp chí Kỹ thuật Hệ thống Chuông **34** (5), 1045–1079 (1955)
18. Micali, S., Ohta, K., Reyzin, L.: Đa chữ ký nhóm con chịu trách nhiệm. Trong: Kỷ yếu Hội nghị ACM lần thứ 8 về Bảo mật Máy tính và Truyền thông. Trang 245–254 (2001)
19. Miller, A., Bentov, I., Bakshi, S., Kumaresan, R., McCorry, P.: Sprites và các kênh trạng thái: Mạng thanh toán nhanh hơn Lightning. Trong: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, tập. 11598, trang 508–526. Springer, Heidelberg (tháng 2 năm 2019). https://doi.org/10.1007/978-3-030-32101-7_30
20. Nakamoto, S.: Bitcoin: Hệ thống tiền mặt điện tử ngang hàng (2008)
21. PDecker, C., Russel, R., Osuntokun, O.: eltoo: Một giao thức Layer 2 đơn giản cho Bitcoin. Xem https://blockstream.com/eltoo.pdf (2017)

22. Poon, J., Dryja, T.: Mạng Bitcoin Lightning: Thanh toán tức thời Off-Chain có thể mở rộng. Xem https://lightning.network/lightning-network-paper.pdf (2016)
23. Richards, S.,Wackerow,P.:Plasma(2021), https://ethereum.org/en/developers/docs/scaling/plasma/
24. Richards, S., Wackerow, P.: Sidechains (2021), https://ethereum.org/en/developers/docs/scaling/sidechains/

A. Phụ lục



Hình 15: Tổng quan về tất cả các chuyển đổi trạng thái của Interhead CEM. Lưu ý rằng hầu hết các quá trình chuyển đổi được giới hạn ở một trong ba giai đoạn của quá trình xây dựng.

Người dịch: Nguyễn Văn Tú

Telegram: https://t.me/Tulibra

Nguồn tài liệu: https://iohk.io/en/research/library/papers/interhead-hydra-two-heads-are-better-than-one/