Giao thức Bitcoin Backbone với các chuỗi có độ khó thay đổi*

Juan A. Garay†
Đại học Texas A&M
College Station, TX, USA.
garay@cse.tamu.edu

Aggelos Kiayias*†‡
Đại học Edinburgh &
IOHK, Edinburgh, UK
akiayias@inf.ed.ac.uk

Nikos Leonardo
Quốc gia và Kapodistrian
Đại học Athens, Hy Lạp
nikos.leonardos@gmail.com

Ngày 25 tháng 8 năm 2019

Tóm lược

Blockchain được duy trì phân tán và đổi mới của Bitcoin phụ thuộc vào độ khó thích hợp của "bằng chứng công việc" mà các thợ đào phải tạo ra để thực hiện các giao dịch. Điều quan trọng là những bằng chứng công việc này phải đủ khó để các thợ đào có cơ hội thống nhất quan điểm của họ khi có kẻ tấn công can thiệp nhưng có sức mạnh tính toán hạn chế, nhưng đủ dễ để có thể giải quyết thường xuyên và cho phép các thợ đào đạt được tiến bộ. Do đó, khi số lượng thợ đào tăng lên theo thời gian, độ khó của những bằng chứng này cũng tăng theo. Bitcoin cung cấp cơ chế điều chỉnh này với bằng chứng thực nghiệm về tốc độ tạo Block không đổi trước những thay đổi dân số như vậy.

Trong bài nghiên cứu này, chúng tôi cung cấp phân tích chính thức đầu tiên về hàm tính toán (tính toán lại) mục tiêu của Bitcoin trong cài đặt mật mã, tức là chống lại tất cả kẻ tấn công có thể có nhằm mục đích phá hoại các thuộc tính của giao thức. Chúng tôi mở rộng mô hình đồng bộ giới hạn q của *giao thức Bitcoin Backbone* [Eurocrypt 2015], mô hình này đặt ra các thuộc tính cơ bản của cấu trúc dữ liệu Blockchain cơ bản của Bitcoin và cho thấy cách có thể xây dựng một sổ cái giao dịch công khai mạnh mẽ dựa trên chúng, cho các môi trường có thể cho thông qua hoặc làm gián đoạn các bên trong mỗi vòng.

Chúng tôi cung cấp một tập hợp các điều kiện cần thiết liên quan đến cách dân số phát triển, trong đó "Bitcoin Backbone với các chuỗi có độ khó thay đổi" cung cấp sổ cái giao dịch mạnh mẽ khi có kẻ tấn công độc hại tích cực kiểm soát một phần nhỏ thợ đào dưới 50 % tại mỗi thời điểm thực hiện. Nghiên cứu của chúng tôi giới thiệu các kỹ thuật và công cụ phân tích mới cho lĩnh vực hệ thống Blockchain có thể hữu ích trong việc phân tích các giao thức Blockchain khác.

1 Giới thiệu

Hệ thống nền tảng Bitcoin (Bitcoin Backbone) [11] trích xuất và phân tích các thuộc tính cơ bản của cấu trúc dữ liệu *Blockchain* cơ bản của Bitcoin, chẳng hạn như "tiền tố chung" và "chất lượng chuỗi", mà các bên ("thợ đào") duy trì và cố gắng mở rộng bằng cách tạo ra "bằng chứng về công việc" (POW, hay còn gọi là "câu đố mật mã" [8, 22, 1, 13])[1]. Sau đó, nó được trình bày chính thức trong [11] cách ứng dụng cơ bản bao gồm cơ chế đồng thuận [21, 16] và sổ cái giao dịch công khai mạnh mẽ thực hiện Crypto phi tập trung (ví dụ: Bitcoin [19]) có thể được xây dựng dựa trên chúng, giả định sức mạnh Hash của kẻ tấn công kiểm soát một phần nhỏ các bên hoàn toàn nhỏ hơn 1/2.

Tuy nhiên, kết quả trong [11] giữ nguyên cài đặt tĩnh, trong đó giao thức được thực thi bởi một số bên *cố định* (mặc dù những người tham gia không nhất thiết phải biết) và do đó với POW (và do đó là Blockchain) có độ khó cố định. Điều này trái ngược với việc triển khai thực tế giao thức Bitcoin trong đó cơ chế "tính toán (tính toán lại) mục tiêu" điều chỉnh độ khó của POW khi số lượng bên thay đổi trong quá trình thực thi giao thức. Chi tiết hơn, trong [11] mục tiêu T mà đầu ra hàm Hash không được vượt quá, được cài đặt và mã hóa cứng ở đầu giao thức và theo

[1]Trong Bitcoin, việc giải quyết bằng chứng công việc về cơ bản tương đương với việc ép buộc bất đẳng thức Hash dựa trên SHA-256.

*Phiên bản rút gọn của bài viết này đã được xuất bản trên Crypto 2017.

†Một phần của công việc này được thực hiện trong khi các tác giả đang thăm Viện Lý thuyết Máy tính Simons, được hỗ trợ bởi Quỹ Simons và Cộng tác DIMACS/Simons về Mật mã thông qua khoản tài trợ NSF #CNS-1523467.

‡Nghiên cứu được hỗ trợ một phần bởi dự án ERC CODAMODA, số 259152 và dự án Horizon 2020 PANORAMIX, số 2. 653497.

cách thỏa mãn mối quan hệ cụ thể với số lượng bên chạy giao thức, cụ thể là tỷ lệ f gần bằng $qnT/2^{\kappa}$ là nhỏ, trong đó q là số lượng truy vấn tới hàm Hash mà mỗi bên được phép trong mỗi vòng, n là số lượng bên và κ là độ dài của đầu ra hàm Hash. Tính bảo mật chỉ được chứng minh khi số lượng bên là n và việc lựa chọn mục tiêu T không bao giờ được tính toán lại, do đó để lại câu hỏi mở về việc phân tích đầy đủ giao thức trong bối cảnh giống như trong thế giới thực, các bên thay đổi linh hoạt theo thời gian.

Trong nghiên cứu này, lần đầu tiên chúng tôi tóm tắt thuật toán tính toán lại mục tiêu từ hệ thống Bitcoin, trình bày khái quát và phân tích giao thức Hệ thống nền tảng Bitcoin với các chuỗi có độ khó thay đổi, được tạo ra bởi số lượng các bên đang phát triển, do đó trả lời câu hỏi mở đã nói ở trên.

Trong cài đặt này, có một tham số m xác định độ dài của một "Epoch" theo số Block[2]. Khi một bên chuẩn bị tính toán Block thứ j của chuỗi với $j \bmod m = 1$, nó sử dụng thuật toán tính toán mục tiêu xác định giá trị mục tiêu phù hợp để sử dụng, dựa trên quan điểm cục bộ của bên đó về tổng số bên hiện diện trong hệ thống, được phản ánh qua tỷ lệ các Block đã được tạo cho đến nay và là một phần trong chuỗi các bên. (Mỗi Block chứa dấu thời gian về thời điểm nó được tạo; trong cài đặt đồng bộ của chúng tôi, dấu thời gian sẽ tương ứng với số vòng khi Block được tạo, xem Phần 2.) Để phù hợp với số lượng các bên ngày càng tăng, chúng tôi mở rộng mô hình của [11] đến môi trường được tự do cho thông qua và làm gián đoạn các bên trong mỗi vòng. Ở các khía cạnh khác, chúng tôi tuân theo mô hình của [11], trong đó tất cả các bên đều có "sức mạnh Hash" như nhau, mỗi bên được phép đặt q truy vấn cho hàm Hash được mô hình hóa như một "Oracle ngẫu nhiên" [3]. Chúng tôi gọi cài đặt của mình là *cài đặt đồng bộ với giới hạn q động*.

Để đưa ra ý tưởng về các vấn đề liên quan, lưu ý rằng nếu không có cơ chế tính toán mục tiêu, trong cài đặt động, giao thức hệ thống nền tảng sẽ không an toàn *ngay cả khi tất cả các bên đều trung thực* và tuân thủ giao thức một cách trung thực. Thật vậy, có thể dễ dàng nhận thấy rằng sự kết hợp giữa môi trường làm tăng số lượng các bên và điều kiện mạng lưới có những kẻ tấn công có thể dẫn đến sự khác biệt đáng kể (hay còn gọi là "phân nhánh") trong chuỗi các bên trung thực, dẫn đến vi phạm thỏa thuận, loại thuộc tính cần thiết cho các ứng dụng của giao thức, chẳng hạn như duy trì sổ cái giao dịch mạnh mẽ. Cuộc tấn công rất đơn giản: môi trường tăng liên tục số lượng các bên để tốc độ sản xuất Block mỗi vòng tăng lên (gần bằng tham số f đã đề cập ở trên); sau đó, các biến cố mạng có kẻ tấn công có thể chia các bên thành hai tập hợp A và B, và lên lịch gửi thông điệp để các bên trong tập hợp A nhận được các Block do các bên trong A tạo ra trước và tương tự cho tập hợp B. Theo giao thức Bitcoin, các bên sẽ sử dụng Block mà họ nhìn thấy đầu tiên và do đó, hai tập hợp sẽ duy trì hai Blockchain riêng biệt.

Mặc dù về nguyên tắc, cuộc tấn công cụ thể này có thể bị ngăn chặn bằng cách sửa đổi hệ thống nền tảng Bitcoin (ví dụ: bằng cách chọn ngẫu nhiên Block mà một bên áp dụng khi họ nhận được trong cùng một vòng hai Block có cùng chỉ mục trong chuỗi), nhưng chắc chắn nó sẽ không đối phó được với tất cả các khả năng có thể xảy ra khi có một cơ chế tính toán lại mục tiêu và kẻ tấn công toàn diện. Một cuộc tấn công như vậy đã được thể hiện trong [2], trong đó bằng cách khai thác "riêng tư" với dấu thời gian liên tiếp nhanh chóng, những thợ đào không trung thực có thể tạo ra các mục tiêu cao giả tạo trong chuỗi riêng của họ; mặc dù chuỗi như vậy có thể phát triển chậm hơn chuỗi chính, nhưng nó vẫn sẽ đạt được tiến bộ và thông qua lập luận chống tập trung, không thể loại trừ một bước tiến bất ngờ của kẻ tấn công có thể phá vỡ thỏa thuận giữa các bên trung thực.

Với những điều trên, mục tiêu chính của chúng tôi là chỉ ra rằng giao thức hệ thống nền tảng có hàm tính toán lại mục tiêu giống như Bitcoin đáp ứng các thuộc tính chất lượng chuỗi và tiền tố chung, như một bước trung gian để chứng minh giao thức thực hiện một sổ cái giao dịch mạnh mẽ. Dự kiến, lớp giao thức mà chúng tôi sẽ phân tích sẽ không bảo toàn các thuộc tính của nó theo những cách tùy ý mà số lượng bên có thể thay đổi theo thời gian. Để giới hạn lỗi trong việc hiệu chỉnh tốc độ tạo Block mà hàm tính toán lại mục tiêu cố gắng thực hiện, chúng tôi sẽ cần một số giới hạn về cách số lượng các bên có thể thay đổi. Đối với $\gamma \in \mathbb{R}^+$ và $s \in \mathbb{N}$, chúng tôi sẽ gọi một chuỗi $(n_r)_{r\in\mathbb{N}}$ các bên *được vận hành tốt* (γ, s) nếu nó giữ điều đó trong một chuỗi các vòng S với $|S| \leq s$, $\max_{r\in S} n_r \leq \gamma \cdot \min_{r\in S} n_r$, và sẽ xác định giá trị nào của các tham số này mà giao thức hệ thống nền tảng là an toàn.

Sau khi mô tả chính thức các Blockchain có độ khó khác nhau và giao thức hệ thống nền tảng Bitcoin trong cài đặt này, phân tích của chúng tôi ở cấp độ cao như sau. Đầu tiên chúng tôi giới thiệu khái niệm về *bản chất tốt (Goodness)* liên quan đến phép tính gần đúng được thực hiện trên f trong một Epoch. Chi tiết hơn, gọi một vòng

[2]Trong Bitcoin, m được đặt thành 2016 và gần tương ứng với 2 tuần theo thời gian thực, giả sử số lượng bên không thay đổi nhiều.

$r(\eta,\theta)$-*tốt*, đối với một số tham số η, $\theta \in \mathbb{R}^+$, nếu giá trị f_r được tính cho số lượng bên thực tế và mục tiêu được sử dụng trong vòng r bởi một bên trung thực nào đó, nằm trong phạm vi $[\eta f, \theta f]$, trong đó f là tốc độ sản xuất Block ban đầu (lưu ý rằng vòng đầu tiên luôn được coi là tốt). Cùng với "bản chất tốt", chúng tôi đưa ra khái niệm về các phép thực thi *điển hình*, trong đó, một cách không chính thức, đối với bất kỳ tập hợp S các vòng liên tiếp, thành công của kẻ tấn công và các bên trung thực không sai lệch quá nhiều so với mong đợi của họ cũng như không có biến cố "xấu" nào liên quan đến hàm Hash xảy ra (chẳng hạn như xảy ra xung đột). Sử dụng giới hạn Martingale, một khái niệm liên quan đến lý thuyết xác suất và phân phối trong việc xác định độ tin cậy của dữ liệu trong một hệ thống phân tán, chúng tôi chứng minh rằng hầu hết tất cả phép thực thi giới hạn đa thức (trong κ) là điển hình.

Tiếp theo, chúng tôi tiến hành chứng minh trong một phép thực thi điển hình, bất kỳ chuỗi nào mà một bên trung thực chấp nhận (1) đều chứa các dấu thời gian gần như chính xác (nghĩa là không có Block tấn công nào có dấu thời gian khác quá nhiều so với thời gian tạo thực sự của nó) và (2) nó có mục tiêu sao cho xác suất sản xuất Block vẫn gần hằng số cố định f, tức là nó "tốt". Cuối cùng, các thuộc tính này cho phép chúng tôi chứng minh phép thực thi điển hình có các thuộc tính chất lượng chuỗi và tiền tố chung, là bước đệm hướng tới mục tiêu cuối cùng, đó là thiết lập giao thức hệ thống nền tảng với độ khó thay đổi để triển khai một sổ cái giao dịch mạnh mẽ. Cụ thể chúng tôi trình bày như sau:

Kết quả chính (Không chính thức, xem Định lý 5 và 6)**.** Giao thức hệ thống nền tảng Bitcoin với các chuỗi có độ khó thay đổi được tham số hóa phù hợp, đáp ứng xác suất áp đảo về m và κ các thuộc tính của (1) *tính bền vững*, nếu một giao dịch *tx* được xác nhận bởi một bên trung thực thì sẽ không có bên trung thực nào không đồng ý về vị trí của *tx* trong sổ cái và (2) *tính sống động*, nếu một giao dịch *tx* được phát tán thì cuối cùng nó sẽ được xác nhận bởi tất cả các bên trung thực.

Nhận xét. Về việc tham số hóa thực tế của hệ thống Bitcoin (sử dụng các Epoch của Block $m = 2016$), mặc dù nó phù hợp với tất cả các ràng buộc trong các định lý của chúng tôi (xem Nhận xét 3 trong Phần 6.1), nó không thể được chứng minh bằng phân tích Martingale của chúng tôi. Trên thực tế, phân tích xác suất của chúng tôi sẽ cần thời gian dài hơn nhiều để cung cấp xác suất tấn công đủ nhỏ. Thắt chặt việc phân tích hoặc phát hiện các cuộc tấn công để tham số hóa ngoài các định lý bảo mật của chúng tôi là một câu hỏi mở thú vị.

Cuối cùng, hãy lưu ý rằng các phần mở rộng khác nhau cho mô hình của chúng tôi có liên quan đến hệ thống Bitcoin và tạo thành những hướng thú vị để nghiên cứu thêm. Điều quan trọng là phân tích bảo mật trong cài đặt "hợp lý" (xem ví dụ [9, 23, 14]) và trong mô hình mạng "đồng bộ một phần" hoặc "độ trễ có giới hạn" [7, 20][3].

2 Mô hình và định nghĩa

Chúng tôi mô tả các giao thức của mình trong một mô hình mở rộng mô hình mạng truyền thông đồng bộ được trình bày trong [10, 11] để phân tích giao thức Hệ thống nền tảng Bitcoin trong cài đặt tĩnh với một số bên cố định (do đó dựa trên công thức của Canetti về " thế giới thực" về phép thực thi giao thức [4, 5, 6] đối với các giao thức nhiều bên) sang cài đặt động với số lượng bên khác nhau. Trong phần này, chúng tôi cung cấp tổng quan cấp cao về mô hình, nêu bật những khác biệt nội tại trong cài đặt động của chúng tôi.

Cấu trúc vòng và thực hiện giao thức. Như trong [10], phép thực thi giao thức tiến hành theo các vòng với đầu vào được cung cấp bởi chương trình môi trường được ký hiệu là $\mathcal{Z}$ cho các bên thực thi giao thức Π và mô hình tấn công của chúng tôi trong mạng là "thích ứng", nghĩa là kẻ tấn công $\mathcal{A}$ được phép nắm quyền kiểm soát các bên một cách nhanh chóng và "gấp rút", nghĩa là trong bất kỳ vòng nào, kẻ tấn công sẽ xem được tất cả thông điệp của những người chơi trung thực trước khi quyết định chiến lược của mình. Quyền truy cập của các bên vào hàm Hash và cơ chế giao tiếp của họ được nắm bắt bởi hàm khuếch tán/Oracle ngẫu nhiên chung phản ánh cấu trúc ngang hàng của Bitcoin. Hàm khuếch tán [10], cho phép thứ tự của các thông điệp được kiểm soát bởi $\mathcal{A}$, tức là không có đảm bảo nguyên tử nào trong việc phát tán thông điệp [12], và hơn nữa, kẻ tấn công được phép giả mạo thông tin nguồn trên

[3]Trong phiên bản mới nhất của [10], chúng tôi cho thấy rằng trong trường hợp có độ khó cố định, việc phân tích hệ thống nền tảng Bitcoin trong mô hình đồng bộ sẽ mở rộng tương đối dễ dàng đến đồng bộ một phần. Chúng tôi dành phần mở rộng của trường hợp độ khó thay đổi cho nghiên cứu sau này.

mỗi thông điệp (tức là liên lạc không được xác thực). Tuy nhiên, kẻ tấn công không thể thay đổi nội dung của thông điệp cũng như không thể ngăn cản việc gửi chúng. Chúng tôi sẽ sử dụng DIFFUSE làm lệnh truyền thông điệp để thực hiện hàm "gửi đến tất cả" này.

Các bên *có thể* hoạt động trong phép thực thi giao thức được mã hóa như một phần của chương trình điều khiển C và đến từ tập hợp các bên $\mathcal{U}$.

Phép thực thi giao thức được điều khiển bởi chương trình môi trường $\mathcal{Z}$ tương tác với các phiên bản khác của chương trình mà nó sinh ra theo quyết định của chương trình điều khiển C. Cặp ($\mathcal{Z}$, C) tạo thành một *hệ thống máy Turing tương tác* (ITM's) theo nghĩa [5]. Phép thực thi liên quan đến chương trình Π, kẻ tấn công $\mathcal{A}$ (là một ITM khác) và tập hợp các bên $\mathcal{U}$. Giả sử chương trình điều khiển C cho phép môi trường $\mathcal{Z}$ có thể kích hoạt một nhóm bằng cách ghi vào băng đầu vào của nó. Lưu ý rằng môi trường $\mathcal{Z}$ cũng nhận được kết quả đầu ra của các bên khi chúng được tạo ra trong một tương tác giống như chương trình con tiêu chuẩn. Ngoài ra, chương trình điều khiển duy trì một cờ cho mỗi phiên bản của ITM, (viết tắt là ITI trong thuật ngữ của [5]), được gọi là cờ **sẵn sàng** và ban đầu được đặt thành sai cho tất cả các bên.

Môi trường $\mathcal{Z}$ ban đầu bị hạn chế bởi C để tạo ra kẻ tấn công $\mathcal{A}$. Mỗi khi kẻ tấn công được kích hoạt, nó có thể gửi một hoặc nhiều thông điệp có dạng (**Corrupt**, P_i) đến C và C sẽ đánh dấu bên tương ứng là không trung thực.

Các chức năng có sẵn cho giao thức. ITI của giao thức Π sẽ có quyền truy cập vào hàm lý tưởng chung để nắm bắt Oracle ngẫu nhiên và cơ chế khuếch tán được xác định theo cách tương tự như [10] và được giải thích bên dưới.

- Hàm Oracle ngẫu nhiên. Cho một truy vấn có giá trị x được đánh dấu là "phép tính" cho hàm $H(\cdot)$ từ một bên trung thực P_i. Giả sử x chưa được truy vấn trước đó, hàm sẽ trả về một giá trị y được chọn ngẫu nhiên từ $\{0,1\}^{\kappa}$; hơn nữa, nó lưu trữ cặp (x, y) trong bảng $H(\cdot)$, trong trường hợp giá trị x tương tự được truy vấn trong tương lai. Mỗi bên trung thực P_i được phép hỏi q truy vấn trong mỗi vòng được xác định bởi hàm khuếch tán (xem bên dưới). Mặt khác, mỗi bên trung thực được cung cấp các truy vấn không giới hạn để "xác minh" hàm $H(\cdot)$. Mặt khác, kẻ tấn công $\mathcal{A}$ được cung cấp một truy vấn số lượng giới hạn trong mỗi vòng được xác định bởi hàm khuếch tán với giới hạn được khởi tạo là 0 và được xác định như sau: bất cứ khi nào một bên không trung thực được kích hoạt, bên đó có thể yêu cầu tăng giới hạn thêm q; mỗi khi kẻ tấn công yêu cầu một truy vấn, giới hạn sẽ giảm đi 1. Không có truy vấn xác minh nào được cung cấp cho $\mathcal{A}$. Lưu ý rằng giá trị q là hàm đa thức của κ, tham số bảo mật. Hàm này có thể duy trì các bảng cho các hàm khác ngoài $H(\cdot)$ nhưng theo quy ước, hàm này sẽ chỉ áp đặt hạn ngạch truy vấn cho hàm $H(\cdot)$.

- Hàm khuếch tán. Hàm này theo dõi các vòng trong phép thực thi giao thức; vì mục đích này, ban đầu nó đặt một biến *round* thành 1. Nó duy trì một chuỗi RECIEVE() được xác định cho mỗi bên P_i trong $\mathcal{U}$. Một bên được kích hoạt được phép truy vấn hàm và tìm nạp nội dung của chuỗi RECIEVE() cá nhân. Hơn nữa, khi hàm nhận được thông điệp (**Diffuse**, m) từ bên P_i, nó sẽ ghi lại thông điệp m. Bên P_i có thể báo hiệu khi vòng đã hoàn thành bằng cách gửi một thông điệp đặc biệt (**RoundComplete**). Đối với kẻ tấn công $\mathcal{A}$, hàm này cho phép nó nhận nội dung của tất cả nội dung được gửi trong các thông điệp **Diffuse** trong vòng và chỉ định nội dung của chuỗi RECIEVE() cho mỗi bên P_i. Kẻ tấn công phải xác định thời điểm hoàn thành vòng hiện tại. Khi tất cả các bên đã hoàn tất vòng hiện tại, hàm này sẽ kiểm tra nội dung của tất cả các chuỗi RECIEVE() và bao gồm mọi thông điệp m đã được các bên phổ biến trong vòng hiện tại nhưng không được kẻ tấn công đóng góp vào băng RECIEVE() (trong cách này đảm bảo việc gửi thông điệp). Nó cũng xóa bất kỳ thông điệp cũ nào đã được phát tán ở các vòng trước và không được phát tán lại. Biến *round* sau đó được tăng lên.

Cài đặt đồng bộ giới hạn *q* động. Xét hai dãy số tự nhiên $\mathbf{n} = \{n_r\}_{r\in\mathbb{N}}$ và $\mathbf{t} = \{t_r\}_{r\in\mathbb{N}}$. Như đã đề cập, trường hợp đầu tiên $\mathcal{Z}$ tạo ra kẻ tấn công $\mathcal{A}$. Sau đó, môi trường có thể tạo ra (hoặc kích hoạt nếu chúng đã được tạo) các bên $P_i \in \mathcal{U}$. Chương trình điều khiển duy trì một bộ đếm trong mỗi chuỗi kích hoạt và khớp nó với vòng hiện tại được duy trì bởi hàm khuếch tán. Mỗi khi một bên trung thực gửi đi một thông điệp có nhãn "**sẵn sàng**", chương trình điều khiển C sẽ tăng bộ đếm sẵn sàng cho vòng. Trong vòng r, chương trình điều khiển C sẽ cho phép kẻ tấn công $\mathcal{A}$ hoàn thành vòng, chỉ với điều kiện (i) chính xác n_r bên đã truyền thông điệp **sẵn sàng**, (ii) số lượng bên ("không trung thực") do $\mathcal{A}$ kiểm soát phải khớp với t_r.

Khi được kích hoạt, các bên có thể đọc băng đầu vào INPUT() và băng liên lạc RECIVE() của họ từ hàm khuếch tán. Chúng tôi thấy các bên không biết về tập hợp các bên được kích hoạt. Giao thức hệ thống nền tảng Bitcoin yêu cầu các bên (thợ đào) tính toán POW. Điều này được mô hình hóa trong [11] khi các bên có quyền truy cập vào Oracle $H(\cdot)$. Thực tế là các bên (đang hoạt động) có khả năng hạn chế để tạo ra các POW như vậy, được ghi lại như trong [11] bởi hàm Oracle ngẫu nhiên và thực tế là nó thúc đẩy các bên truy vấn một số lượng truy vấn giới hạn *mỗi vòng*. Giới hạn q là hàm của tham số bảo mật κ; theo nghĩa này, các bên có thể được gọi là giới hạn q[4]. Chúng tôi gọi những hạn chế trên đối với môi trường, các bên và kẻ tấn công là *cài đặt đồng bộ giới hạn q động*.

Thuật ngữ $\{\text{VIEW}^{P,\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}(z)\}_{z\in\{0,1\}^*}$ biểu thị tập hợp biến ngẫu nhiên mô tả góc nhìn của bên P sau khi hoàn thành giao thức vận hành phép thực thi Π với môi trường $\mathcal{Z}$ và kẻ tấn công $\mathcal{A}$, trên đầu vào $z \in \{0, 1\}^*$. Chúng tôi sẽ chỉ xem xét phép thực thi "độc lập" mà không có bất kỳ thông tin phụ trợ nào và do đó chúng tôi sẽ hạn chế thực hiện với $z = 1^\kappa$. Vì lý do này, chúng tôi sẽ chỉ đề cập đến tập hợp theo $\text{VIEW}^{P,\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$. Sự kết hợp chế độ xem của tất cả các bên từng được kích hoạt trong phép thực thi được biểu thị bằng chế độ $\text{VIEW}^{\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$.

Thuộc tính của giao thức. Trong các định lý, chúng tôi sẽ quan tâm đến *thuộc tính* của các giao thức Π đang chạy trong cài đặt trên. Các thuộc tính như vậy sẽ được xác định là các vị từ trên biến ngẫu nhiên $\text{VIEW}^{\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$ bằng cách định lượng trên tất cả các kẻ tấn công $\mathcal{A}$ có thể và môi trường $\mathcal{Z}$. Lưu ý rằng tất cả các giao thức của chúng tôi sẽ chỉ đáp ứng các thuộc tính có xác suất sai sót nhỏ trong κ cũng như trong tham số k được chọn từ $\{1,\ldots,\kappa\}$ (với tầm nhìn xa, chúng tôi lưu ý rằng trong thực tế sẽ có thể chọn k nhỏ hơn nhiều so với κ, ví dụ: $k = 6$).

Lớp giao thức mà chúng tôi sẽ phân tích sẽ không thể bảo toàn các thuộc tính của nó cho các chuỗi tùy ý của các bên. Để hạn chế cách dãy **n** dao động, chúng tôi sẽ giới thiệu lớp dãy sau.

Định nghĩa 1. *Đối với $\gamma \in \mathbb{R}^+$, chúng tôi gọi một dãy $(n_r)_{r\in\mathbb{N}}$ được vận hành tốt (γ, s) nếu với bất kỳ tập hợp S nào có nhiều nhất s vòng liên tiếp,* $\max_{r\in S} n_r \leq \gamma \cdot \min_{r\in S} n_r$.

Chúng tôi thấy định nghĩa trên khá chung chung và cũng có thể nắm bắt được sự tăng trưởng theo cấp số nhân; ví dụ: bằng cách đặt $\gamma = 2$ và $s = 10$, cứ sau 10 vòng, số lượng bên sẵn sàng có thể tăng gấp đôi. Lưu ý rằng điều này sẽ không dẫn đến tổng thời gian chạy theo cấp số nhân vì tổng thời gian chạy bị giới hạn bởi đa thức trong κ, (do thực tế là $(\mathcal{Z}, C)$ là một hệ thống của ITM, $\mathcal{Z}$ bị giới hạn đa thức cục bộ, C là một chương trình thời gian đa thức, và do đó [5, Mệnh đề 3] được áp dụng).

Chính thức hơn, một giao thức Π sẽ đáp ứng thuộc tính Q cho một lớp chuỗi **n**, **t** nhất định với điều kiện là với tất cả PPT $\mathcal{A}$ và giới hạn đa thức cục bộ $\mathcal{Z}$, nó cho rằng Q ($\text{VIEW}^{\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$) là đúng với xác suất áp đảo của các đồng Coin của $\mathcal{A}$, $\mathcal{Z}$ và hàm Oracle ngẫu nhiên.

Trong nghiên cứu này, chúng tôi sẽ quan tâm đến các chuỗi **n** được vận hành tốt (γ, s), các chuỗi **t** được giới hạn thích hợp bởi **n**, và các giao thức Π được tham số hóa phù hợp cho **n**, **t**.

3 Blockchain có độ khó thay đổi

Chúng tôi bắt đầu bằng cách giới thiệu ký hiệu Blockchain; chúng tôi sử dụng ký hiệu tương tự như [10] và mở rộng khái niệm về Blockchain để bao gồm rõ ràng *dấu thời gian* (dưới dạng chỉ báo vòng). Cho $G(\cdot)$ và $H(\cdot)$ là các hàm Hash mật mã có đầu ra là $\{0, 1\}^\kappa$. *Block có mục tiêu* $T \in \mathbb{N}$ là bộ tứ có dạng $B = \langle r, st, x, ctr\rangle$ trong đó $st \in \{0, 1\}^\kappa$, $x \in \{0, 1\}^*$, và $r, ctr \in \mathbb{N}$ sao cho chúng thỏa mã vị từ $\mathbf{validblock}^T_q(B)$ được xác định là

$$(H(ctr, G(r, st, x)) < T) \wedge (ctr \leq q) .$$

[4]Trong [11], điều này được gọi là "mô hình phẳng" về mặt sức mạnh tính toán, trong đó tất cả các bên đều được coi là bình đẳng. Trong thực tế, các bên khác nhau có thể có "sức mạnh Hash" khác nhau; lưu ý rằng điều này không làm mất đi tính tổng quát vì người ta có thể tưởng tượng rằng các bên thực sự chỉ đơn giản là một cụm của một số lượng tùy ý các bên mô hình phẳng.

Tham số $q \in \mathbb{N}$ là một giới hạn mà trong quá trình triển khai Bitcoin xác định kích thước của thanh ghi *ctr*; như trong [10], chúng tôi cho phép q tùy ý và sử dụng nó để biểu thị số lượng truy vấn Hash tối đa được phép trong một vòng (xem Phần 2). Chúng tôi làm điều này để thuận tiện và phân tích của chúng tôi áp dụng một cách đơn giản cho trường hợp *ctr* bị giới hạn trong phạm vi $0 \leq ctr < 2^{32}$ và q độc lập với *ctr*.

Blockchain hay đơn giản là *chuỗi* là một chuỗi *các Block*. Block ngoài cùng bên phải là *phần đầu* của chuỗi, ký hiệu là head($\mathcal{C}$). Lưu ý rằng chuỗi rỗng ε cũng là một chuỗi; theo quy ước chúng tôi gọi head(ε) = ε. Chuỗi $\mathcal{C}$ có head($\mathcal{C}$) = $\langle r, st, x, ctr\rangle$ có thể được mở rộng thành chuỗi dài hơn bằng cách nối thêm một Block hợp lệ $B = \langle r', st', x', ctr'\rangle$ thỏa mãn $st' = H(ctr, G(r, st, x))$ và $r' > r$, trong đó r' được gọi là *dấu thời gian* của Block B. Trong trường hợp $\mathcal{C} = \varepsilon$, theo quy ước, bất kỳ Block hợp lệ nào có dạng $\langle r', st', x', ctr'\rangle$ đều có thể mở rộng nó. Trong cả hai trường hợp, chúng tôi có một chuỗi mở rộng $\mathcal{C}_{\mathbf{new}} = \mathcal{C}B$ thỏa mãn head($\mathcal{C}_{\mathbf{new}}$) = B.

Độ dài của chuỗi len($\mathcal{C}$) là số Block của nó. Xét một chuỗi $\mathcal{C}$ có độ dài ℓ và bất kỳ số nguyên không âm k. Chúng tôi biểu thị bằng $\mathcal{C}^{\lceil k}$ chuỗi kết quả từ việc "lược bỏ bớt" k Block ngoài cùng bên phải. Lưu ý rằng với $k \geq$ len($\mathcal{C}$), $\mathcal{C}^{\lceil k} = \varepsilon$. Nếu $\mathcal{C}_1$ là tiền tố của $\mathcal{C}_2$ ta viết $\mathcal{C}_1 \preceq \mathcal{C}_2$.

Cho một chuỗi $\mathcal{C}$ có độ dài len($\mathcal{C}$) = ℓ, chúng tôi gọi $\mathbf{x}_{\mathcal{C}}$ biểu thị Vectơ của các giá trị ℓ được lưu trữ trong $\mathcal{C}$ và bắt đầu bằng giá trị của Block đầu tiên. Tương tự, $\mathbf{r}_{\mathcal{C}}$ là Vectơ chứa dấu thời gian của Blockchain $\mathcal{C}$.

Đối với chuỗi có độ khó thay đổi, mục tiêu T được tính toán lại cho mỗi Block dựa trên dấu thời gian vòng của các Block trước đó. Cụ thể, có một hàm $D : \mathbb{Z}^* \to \mathbb{R}$ nhận một Vectơ tùy ý gồm các dấu thời gian vòng và tạo ra mục tiêu tiếp theo. Giá trị $D(\varepsilon)$ là mục tiêu ban đầu của hệ thống. *Độ khó* của mỗi Block được đo bằng số lần Block đó khó lấy hơn Block mục tiêu T_0. Cụ thể hơn, độ khó của Block có mục tiêu T bằng T_0/T; không mất tính tổng quát, chúng tôi sẽ áp dụng biểu thức đơn giản hơn $1/T$ (vì T_0 sẽ là hằng số trong tất cả các phép thực thi). Chúng tôi sẽ sử dụng **diff**($\mathcal{C}$) để biểu thị độ khó của chuỗi. Nó bằng tổng độ khó của tất cả các Block tạo nên chuỗi.

Hàm tính toán mục tiêu. Hàm tính toán mục tiêu $D(\cdot)$ nhằm mục đích duy trì tốc độ sản xuất Block không đổi. Nó được tham số hóa bởi $m \in \mathbb{N}$ và $f \in (0, 1)$; Mục tiêu của nó là m Block sẽ được tạo ra sau mỗi m/f vòng. Chúng ta sẽ thấy trong Phần 6 xác suất $f(T, n)$ mà n bên tạo ra một Block mới với mục tiêu T là gần đúng bằng

$$f(T, n) \approx \frac{qTn}{2^{\kappa}}.$$

(Lưu ý rằng $T/2^{\kappa}$ là xác suất mà một người chơi tạo ra một Block trong một truy vấn.)

Để đạt được mục tiêu trên Bitcoin cố gắng giữ $qTn/2^{\kappa}$ gần với f. Để đạt được mục tiêu đó, Bitcoin chờ đợi m Block được tạo ra và dựa trên độ khó của chúng cũng như tốc độ tính toán của các Block này, nó sẽ tính toán mục tiêu tiếp theo. Cụ thể hơn, giả sử m *Block* cuối cùng của chuỗi $\mathcal{C}$ dành cho mục tiêu T và được sản xuất tại Δ vòng. Hãy xem xét trường hợp một số người chơi

$$n(T, \Delta) = \frac{2^{\kappa} m}{qT\Delta}$$

cố gắng tạo ra m Block mục tiêu T; lưu ý rằng họ sẽ mất khoảng Δ vòng dự kiến. Số lượng người chơi tại thời điểm m Block được tạo ra được ước tính bằng $n(T, \Delta)$; thì mục tiêu tiếp theo T' được thiết lập sao cho $n(T, \Delta)$ người chơi sẽ cần m/f vòng với kỳ vọng tạo ra m Block mục tiêu T'. Vì vậy, thật hợp lý khi thiết lập

$$T' = \frac{\Delta}{m/f} \cdot T,$$

bởi vì nếu số lượng người chơi là $n(T, \Delta)$ và không thay đổi, họ sẽ phải mất m/f vòng để tạo ra m Block. Nếu ước tính ban đầu của số bên là n_0, chúng tôi sẽ giả sử T_0 được thiết lập thích hợp sao cho $f \approx qT_0 n_0 / 2^{\kappa}$ và sau đó

$$T' = \frac{n_0}{n(T,\Delta)} \cdot T_0.$$

Ghi chú 1. *Hãy nhớ rằng trong cài đặt giới hạn q phẳng, tất cả các bên đều có cùng sức mạnh Hash (truy vấn q mỗi vòng). Theo đó, n_0 đại diện cho công suất Hash ban đầu ước tính, trong khi $n(T, \Delta)$ là công suất Hash ước tính trong m Block cuối cùng của chuỗi $\mathcal{C}$. Kết quả là mục tiêu mới bằng mục tiêu ban đầu T_0 nhân với hệ số $n_0/n(T, \Delta)$, phản ánh sự thay đổi công suất Hash trong m Block cuối cùng.*

Dựa vào những điều trên, chúng tôi định nghĩa chính thức về hàm tính toán (tính toán lại) mục tiêu như sau.

Định nghĩa 2. *Đối với các hằng số cố định κ, τ, m, n_0, T_0, hàm tính toán mục tiêu* $D : \mathbb{Z}^* \to \mathbb{R}$ *được xác định là*

$$D(\varepsilon) = T_0 \qquad \text{và} \qquad D(r_1,\ldots,r_v) = \begin{cases} \frac{1}{\tau}\cdot T & \text{nếu } \frac{n_0}{n(T,\Delta)}\cdot T_0 < \frac{1}{\tau}\cdot T \\ \tau \cdot T & \text{nếu } \frac{n_0}{n(T,\Delta)}\cdot T_0 > \frac{1}{\tau}\cdot T \\ \frac{n_0}{n(T,\Delta)}\cdot T_0 & \text{trong các trường hợp khác} \end{cases}$$

trong đó $n(T, \Delta) = 2^{\kappa} m/qT\Delta$, *với* $\Delta = r_{m'} - r_{m'-m}$, $T = D(r_1,\ldots,r_{m'-1})$, *và* $m' = m \cdot \lfloor v/m \rfloor$.

Trong định nghĩa, $(r_1,\ldots,r_v)$ tương ứng với một chuỗi v Block với dấu thời gian r_i của Block thứ i; m', Δ và T tương ứng với Block, thời lượng và mục tiêu cuối cùng của Epoch hoàn thành cuối cùng .

Ghi chú 2. *Một ghi chú là phù hợp về trường hợp* $\frac{n_0}{n(T,\Delta)}\cdot T_0 \notin \left[\frac{1}{\tau}T, \tau T\right]$, *vì khía cạnh này của định nghĩa không được chứng minh bằng cuộc thảo luận trước Định nghĩa 2. Lúc đầu, có vẻ như không có lý do gì để đưa ra một "bộ lọc giảm thiểu" như vậy trong hàm tính toán lại mục tiêu của Bitcoin và người ta nên để các bên cùng cố gắng đạt được mục tiêu thích hợp. Điều thú vị là trong trường hợp không có sự giảm thiểu như vậy, một cuộc tấn công hiệu quả sẽ được biết đến [2] (chống lại thuộc tính tiền tố chung). Chúng tôi thấy sự suy giảm này đủ để chứng minh tính bảo mật trước tất cả những kẻ tấn công, bao gồm cả những kẻ được xem xét trong [2] (với tầm nhìn xa, chúng tôi có thể nói rằng cuộc tấn công vẫn diễn ra nhưng sẽ mất thời gian theo cấp số nhân để thực hiện).*

4 Giao thức hệ thống nền tảng Bitcoin với độ khó thay đổi

Trong phần này, chúng tôi đưa ra mô tả cấp cao về giao thức hệ thống nền tảng Bitcoin với các chuỗi có độ khó khác nhau; mô tả chi tiết hơn, bao gồm mã giả định của thuật toán được nêu trong Phụ lục A. Việc trình bày dựa trên mô tả trong [11]. Sau đó, chúng tôi xây dựng hai thuộc tính mong muốn của Blockchain *tiền tố chung* và *chất lượng chuỗi* cho cài đặt động.

4.1 Giao thức

Như trong [11], khi mô tả giao thức hệ thống nền tảng, chúng tôi cố tình tránh chỉ định loại giá trị/nội dung mà các bên cố gắng chèn vào chuỗi, loại xác thực chuỗi mà họ thực hiện (ngoài việc kiểm tra các thuộc tính cấu trúc của nó liên quan đến các hàm Hash $G(\cdot)$, $H(\cdot)$) và cách chúng diễn giải chuỗi. Các hoạt động kiểm tra và thao tác này được xử lý bởi các hàm bên ngoài $V(\cdot)$, $I(\cdot)$ và $R(\cdot)$ (lần lượt là *hàm xác thực nội dung*, *hàm đóng góp đầu vào* và *hàm đọc chuỗi*) được chỉ định bởi ứng dụng chạy "trên cùng" của giao thức hệ thống nền tảng. Giao thức hệ thống nền tảng Bitcoin trong cài đặt động bao gồm ba thuật toán.

Xác thực chuỗi. Thuật toán **validate** thực hiện xác thực các thuộc tính cấu trúc của chuỗi $\mathcal{C}$. Nó được đưa ra làm đầu vào giá trị q, cũng như các hàm Hash $H(\cdot)$, $G(\cdot)$. Nó được tham số hóa bởi vị từ xác thực nội dung $V(\cdot)$ cũng như bởi $D(\cdot)$, *hàm tính toán mục tiêu* (Phần 3). Đối với mỗi Block của chuỗi, thuật toán sẽ kiểm tra xem bằng chứng công việc đã được giải quyết đúng cách chưa (với mục tiêu phù hợp được xác định bởi hàm tính toán mục tiêu) và bộ đếm *ctr* không vượt quá q. Hơn nữa, nó thu thập đầu vào từ tất cả các Block $\mathbf{x}_{\mathcal{C}}$ và kiểm tra chúng thông qua vị từ $V(\mathbf{x}_{\mathcal{C}})$. Chuỗi không thực hiện được quy trình xác thực này sẽ bị từ chối.

So sánh chuỗi. Mục tiêu của thuật toán thứ hai, **maxvalid**, là tìm ra chuỗi "tốt nhất có thể" khi cho một tập hợp các chuỗi. Thuật toán rất đơn giản và được tham số hóa bằng hàm **max**$(\cdot)$ áp dụng một số sắp xếp cho không gian của Blockchain. Khía cạnh quan trọng nhất là độ khó của chuỗi, trong đó trường hợp **max**$(\mathcal{C}_1, \mathcal{C}_2)$ sẽ trả về *độ khó* cao nhất trong hai trường hợp. Trong trường hợp **diff**$(\mathcal{C}_1)$ = **diff**$(\mathcal{C}_2)$, một số đặc tính khác có thể được sử dụng để phá

vỡ ràng buộc. Trong trường hợp của chúng tôi, **max**$(\cdot,\cdot)$ sẽ luôn trả về toán hạng đầu tiên để phản ánh thực tế là các bên áp dụng chuỗi đầu tiên mà họ nhận được từ mạng lưới.

Bằng chứng công việc. Thuật toán thứ ba, **pow**, là quy trình tìm kiếm bằng chứng công việc. Nó lấy đầu vào là một chuỗi và cố gắng mở rộng nó thông qua việc giải quyết bằng chứng công việc. Thuật toán này được tham số hóa bởi hai hàm Hash $H(\cdot)$, $G(\cdot)$ cũng như tham số q. Hơn nữa, thuật toán gọi hàm tính toán mục tiêu $D(\cdot)$ để xác định giá trị T sẽ được sử dụng cho bằng chứng công việc. Quy trình, cho chuỗi $\mathcal{C}$ và giá trị x được chèn vào chuỗi, Hash các giá trị này để thu được h và khởi tạo bộ đếm ctr. Sau đó, nó tăng ctr và kiểm tra xem liệu $H(ctr, h) < T$; trong trường hợp tìm thấy một ctr phù hợp thì thuật toán sẽ thành công trong việc giải POW và mở rộng chuỗi $\mathcal{C}$ thêm một Block.

Giao thức hệ thống nền tảng Bitcoin. Cốt lõi của giao thức hệ thống nền tảng với độ khó thay đổi tương tự như trong [11], với một số điểm khác biệt quan trọng. Đầu tiên là quy trình phải tuân theo khi các bên bắt đầu hoạt động. Các bên kiểm tra cờ **sẵn sàng** mà họ sở hữu, cờ này sai khi và chỉ khi họ không hoạt động ở vòng trước. Trong trường hợp cờ **sẵn sàng** là sai, họ sẽ phát tán một thông điệp '**Tham gia**' đặc biệt để yêu cầu phiên bản mới nhất của (các) Blockchain. Tương tự, các bên nhận được thông điệp yêu cầu đặc biệt trong băng RECIEVE() sẽ phát tán chuỗi của họ. Như các bên trước đây, chạy "vô thời hạn" (phân tích bảo mật của chúng tôi sẽ áp dụng khi tổng thời gian chạy là đa thức trong κ). Hàm đóng góp đầu vào $I(\cdot)$ và hàm đọc chuỗi $R(\cdot)$ được áp dụng cho các giá trị được lưu trữ trong chuỗi. Các bên kiểm tra băng liên lạc RECIEVE() của họ để xem liệu có bất kỳ cập nhật cần thiết nào đối với chuỗi cục bộ hay không; sau đó họ cố gắng mở rộng nó thông qua thuật toán bằng chứng công việc **pow**. Hàm $I(\cdot)$ xác định đầu vào sẽ được thêm vào chuỗi dựa trên trạng thái st của bên tham gia, chuỗi $\mathcal{C}$ hiện tại, nội dung của băng đầu vào INPUT() và băng liên lạc RECIEVE() của các bên. Băng đầu vào chứa hai loại ký hiệu, READ và (INSERT, *value*); đầu vào khác được bỏ qua. Trong trường hợp chuỗi $\mathcal{C}$ cục bộ được mở rộng, chuỗi mới sẽ được phân bổ cho các bên khác. Cuối cùng, trong trường hợp ký hiệu READ xuất hiện trong băng liên lạc, giao thức sẽ áp dụng hàm $R(\cdot)$ cho chuỗi hiện tại của nó và ghi kết quả vào băng đầu ra OUTPUT().

4.2 Thuộc tính của giao thức hệ thống nền tảng với độ khó thay đổi

Tiếp theo, chúng tôi xác định hai thuộc tính của giao thức hệ thống nền tảng mà giao thức sẽ thiết lập. Chúng là các biến thể gần giống của các thuộc tính trong [11], được sửa đổi phù hợp cho cài đặt đồng bộ giới hạn q động.

Thuộc tính *tiền tố chung* về cơ bản vẫn giữ nguyên. Nó được tham số hóa bởi một giá trị $k \in \mathbb{N}$, xem xét một môi trường và kẻ tấn công tùy ý, và nó giữ nguyên miễn là chuỗi của hai bên bất kỳ chỉ khác nhau trong k Block gần nhất của chúng. Việc xác định thuộc tính giữa chuỗi của một bên trung thực và chuỗi khác có thể là tấn công thực sự rất hữu ích. Định nghĩa như sau.

Định nghĩa 3 (Thuộc tính tiền tố chung)**.** *Thuộc tính* tiền tố chung Q_{cp} *với tham số* $k \in \mathbb{N}$ *cho biết với mọi cặp người chơi trung thực* P_1, P_2 *chấp nhận chuỗi* $\mathcal{C}_1$, $\mathcal{C}_2$ *tại vòng* $r_1 \leq r_2$ *tương ứng theo* $\text{VIEW}^{\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$, *nó cho rằng* $\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$.

Thuộc tính thứ hai là *chất lượng chuỗi*, thể hiện số lượng đóng góp của bên trung thực được chứa trong một phần đủ dài và liên tục trong chuỗi của một bên. Bởi vì chúng tôi xem xét các chuỗi có độ khó thay đổi nên sẽ thuận tiện hơn khi nghĩ về đóng góp của các bên theo tổng độ khó mà họ thêm vào chuỗi thay vì số Block mà họ thêm vào (như được thực hiện trong [11]). Tính chất nêu rõ rằng những kẻ tấn công bị giới hạn ở độ khó mà họ có thể đóng góp vào bất kỳ đoạn chuỗi đủ dài nào.

Định nghĩa 4 (Thuộc tính chất lượng chuỗi)**.** *Thuộc tính* chất lượng chuỗi Q_{cq} *với các tham số* $\mu \in \mathbb{R}$ *và* $\ell \in \mathbb{N}$ *nêu rõ rằng đối với bất kỳ bên P trung thực nào có chuỗi* $\mathcal{C}$ *trong* $\text{VIEW}^{\mathbf{t},\mathbf{n}}_{\Pi,\mathcal{A},\mathcal{Z}}$, *nó giữ điều đó đối với bất kỳ* ℓ *Block liến tiếp của* $\mathcal{C}$ *có tổng độ khó d, các Block trung thực góp phần tạo ra độ khó ít nhất* $\mu \cdot d$.

4.3 Ứng dụng: Sổ cái giao dịch mạnh mẽ

Bây giờ hãy đến với ứng dụng (chính) mà giao thức hệ thống nền tảng Bitcoin được thiết kế để giải quyết. *Sổ cái giao dịch mạnh mẽ* là một giao thức duy trì sổ cái các giao dịch được tổ chức dưới dạng chuỗi $\mathcal{C}$, đáp ứng hai thuộc tính sau.

- *Tính bền vững:* Được tham số hóa bởi $k \in \mathbb{N}$ (tham số "độ sâu"), nếu một bên trung thực P, duy trì chuỗi $\mathcal{C}$, báo hiệu rằng một giao dịch *tx* nằm trong $\mathcal{C}^{\lceil k}$ thì nó giữ cho mọi bên trung thực P' khác duy trì chuỗi $\mathcal{C}'$ thì $\mathcal{C}'$ chứa tx ở cùng một vị trí.
- *Tính sống động:* Được tham số hóa bởi $u, k \in \mathbb{N}$ (tương ứng là các tham số "thời gian chờ" và "độ sâu"), nếu một giao dịch *tx* được cung cấp cho tất cả các bên trung thực trong u vòng liên tiếp thì nó giữ nguyên điều đó cho bất kỳ người chơi P nào, duy trì một chuỗi $\mathcal{C}$, *tx* sẽ ở $\mathcal{C}^{\lceil k}$.

Chúng tôi lưu ý rằng, như trong [11], tính sống động có thể áp dụng cho các giao dịch "trung lập" (nghĩa là những giao dịch mà chúng không bao giờ "xung đột" với các giao dịch khác trong sổ cái) hoặc các giao dịch được tạo bởi Oracle **Txgen** tạo ra giao dịch được tạo ra một cách trung thực.

5 Tổng quan về phân tích

Mục tiêu chính của chúng tôi là chỉ ra rằng giao thức hệ thống nền tảng đáp ứng các thuộc tính tiền tố chung và chất lượng chuỗi (Phần 4.2) trong môi trường được vận hành tốt (γ, s) như một bước trung gian để chứng minh giao thức thực hiện một sổ cái giao dịch mạnh mẽ. Trong phần này, chúng tôi trình bày tổng quan cấp cao về cách tiếp cận; phân tích đầy đủ sau đó sẽ được trình bày trong Phần 6. Để chứng minh các tính chất nói trên, trước tiên chúng tôi mô tả đặc điểm của tập hợp các phép thực thi *điển hình*. Một cách không chính thức, phép thực thi là điển hình nếu đối với bất kỳ tập S các vòng liên tiếp, thành công của kẻ tấn công và các bên trung thực không sai lệch quá nhiều so với mong đợi của họ, và không có sự kiện xấu nào xảy ra đối với hàm Hash (mà chúng tôi mô hình hóa là "Oracle ngẫu nhiên"). Sử dụng giới hạn Martingale của Định lý 7, chúng tôi chứng minh hầu hết tất cả các phép thực thi giới hạn đa thức đều là điển hình. Sau đó, chúng tôi tiến hành chứng minh trong một phép thực thi điển hình, bất kỳ chuỗi nào mà một bên trung thực chấp nhận (1) đều chứa các dấu thời gian gần như chính xác (nghĩa là không có Block tấn công nào có dấu thời gian khác quá nhiều so với thời gian tạo thực sự của nó) và (2) có một mục tiêu sao cho xác suất sản xuất Block vẫn ở gần một hằng số cố định f. Cuối cùng, các thuộc tính này của một phép thực thi điển hình sẽ đưa chúng ta đến mục tiêu cuối cùng: chứng minh một phép thực thi điển hình có thuộc tính tiền tố chung và chất lượng chuỗi, và do đó người ta có thể xây dựng trên Blockchain một sổ cái giao dịch mạnh mẽ (Phần 4.3). Ở đây chúng tôi nêu bật các bước chính và các khái niệm mới được giới thiệu.

Những phép thực thi "tốt". Để có thể nói một cách định lượng về các phép thực thi điển hình, trước tiên chúng tôi đưa ra khái niệm về các phép thực thi (η, θ)-*tốt*, thể hiện mức độ các bên ước tính f. Giả sử ở vòng r có chính xác n bên truy vấn Oracle với mục tiêu T. Xác suất ít nhất một trong số họ thành công là

$$f(T, n) = 1 - \left(1 - \frac{T}{2^\kappa}\right)^{qn}.$$

Đối với mục tiêu ban đầu T_0 và ước tính ban đầu về số lượng bên n_0, chúng tôi biểu thị $f_0 = f(T_0, n_0)$. Nhìn về phía trước, mục tiêu của cơ chế tính toán lại mục tiêu là duy trì mục tiêu T cho mỗi bên sao cho $f(T, n_r) \approx f_0$ cho tất cả các vòng r. (Để ngắn gọn, chúng tôi sẽ bỏ chỉ số dưới và chỉ gọi nó là f.)

Bây giờ, ở vòng r của phép thực thi E, các bên trung thực có thể đang truy vấn Oracle ngẫu nhiên để tìm các mục tiêu khác nhau. Biểu thị $T_r^{min}(E)$ là mức tối thiểu và $T_r^{max}(E)$ là mức tối đa đối với các mục tiêu đó. Chúng ta nói r là điểm tính toán lại mục tiêu của chuỗi hợp lệ $\mathcal{C}$ nếu có một Block có dấu thời gian r và m chia chính xác số Block cho đến (và bao gồm) Block này. Xét các hằng số $\eta \in (0, 1]$ và $\theta \in [1, \infty)$ và một phép thực thi E:

Định nghĩa 5 (Tóm tắt). Một vòng r là (η, θ)-*tốt* trong E nếu $\eta f \leq f(T_r^{min}(E), n_r)$ và $f(T_r^{max}(E), n_r) \leq \theta f$. Phép thực thi E là (η, θ)-*tốt* nếu mỗi vòng của E là (η, θ)-*tốt*.

Chúng tôi sẽ chỉ nghiên cứu tiến trình của các bên trung thực khi mục tiêu của họ nằm trong phạm vi hợp lý. Với khả năng cao, những bên trung thực luôn làm việc với những mục tiêu hợp lý. Giới hạn sau đây sẽ hữu ích vì nó đưa ra ước tính về tiến độ mà các bên trung thực đã đạt được trong một phép thực thi (η, θ)-tốt. Chúng tôi sẽ quan tâm đến tiến trình đến từ *các vòng thành công duy nhất*, trong đó chính xác một bên trung thực đã tính ra POW. Gọi

Q_r là biến ngẫu nhiên bằng độ khó (tối đa) của các vòng đó (hãy nhớ rằng Block có mục tiêu T có độ khó $1/T$); bằng 0 trong các trường hợp khác. Chúng tôi gọi Q_r cũng là độ khó "duy nhất". Chúng tôi có thể hiển thị như sau.

Mệnh đề 2 (Không chính thức). Nếu r là một vòng (η, θ)-tốt trong phép thực thi E, thì $\mathbf{E}[Q_r(E_{r-1})] \geq (1-\theta f)pn_r$, trong đó $Q_r(E_{r-1})$ là độ khó duy nhất liên quan đến phép thực hiện cho đến nay, và $p = \frac{q}{2^\kappa}$.

Đối số "mỗi vòng" liên quan đến các biến ngẫu nhiên có liên quan là không đủ, vì chúng tôi cần phép thực thi với hành vi "tốt" trong một chuỗi các vòng, tức là các biến phải được tập trung xung quanh phương tiện của chúng. Điều này không dễ đạt được, vì xác suất của các thí nghiệm được thực hiện mỗi vòng phụ thuộc vào lịch sử (do tính toán lại mục tiêu). Để giải quyết vấn đề thiếu tập trung/phương sai này, chúng tôi đưa ra biện pháp sau.

Những phép thực thi điển hình. Ý tưởng mà khái niệm này nắm bắt được như sau. Lưu ý rằng ở mỗi vòng của phép thực thi E, các bên thực hiện phép thử Bernoulli với xác suất thành công có thể bị ảnh hưởng bởi kẻ tấn công. Dựa vào phép thực thi, những thử nghiệm này được xác định và chúng tôi có thể tính toán tiến độ dự kiến mà các bên đạt được dựa trên xác suất tương ứng. Sau đó, chúng tôi so sánh giá trị này với tiến độ thực tế. Nếu chênh lệch là "hợp lý" thì chúng tôi tuyên bố E là *điển hình*. Tuy nhiên, lưu ý rằng việc xem xét riêng sự khác biệt này không phải lúc nào cũng đủ vì phương sai của quá trình có thể quá cao. Định nghĩa của chúng tôi, theo Định lý 7 (Phụ lục C), nói rằng hoặc có sự khác biệt cao so với tập hợp các vòng mà chúng tôi đang xem xét hoặc các bên đã đạt được tiến bộ trong các vòng này như mong đợi. Chính thức hơn một chút, đối với một truy vấn Oracle ngẫu nhiên nhất định trong một phép thực thi E, lịch sử của phép thực thi ngay trước khi truy vấn diễn ra, xác định các tham số của phân phối mà kết quả của truy vấn này tuân theo dưới dạng POW (thử nghiệm Bernoulli). Đối với các truy vấn được thực hiện trong một tập hợp các vòng S, gọi $V(S)$ biểu thị tổng phương sai của các phép thử này.

Định nghĩa 8 (Tóm tắt). Một phép thực thi E là (ϵ, η, θ)-*điển hình* nếu đối với bất kỳ tập hợp S các vòng liên tiếp sao cho $V(S)$ được giới hạn thích hợp từ phía trên:

- Độ khó duy nhất trung bình được giới hạn *dưới* bởi $\frac{1}{|S|}\left(\sum_{r\in S}\mathbf{E}\left[Q_r(E_{r-1})\right] - \epsilon(1-\theta f)p\sum_{r\in S}n_r\right)$;
- Độ khó tối đa trung bình được giới hạn *trên* bởi $\frac{1}{|S|}(1+\epsilon)p\sum_{r\in S}n_r$;
- Độ khó trung bình của các Block có mục tiêu "dễ" của kẻ tấn công được giới hạn *trên* bởi $\frac{1}{|S|}(1+\epsilon)p\sum_{r\in S}t_r$, trong khi *số* Block có mục tiêu "khó" được giới hạn dưới m bởi một hằng số phù hợp; và
- Không có "biến cố xấu" nào xảy ra đối với hàm Hash (ví dụ: xung đột).

Sau đây là một trong những bước chính trong phân tích của chúng tôi.

Mệnh đề 4 (Không chính thức). Hầu như tất cả các phép thực thi giới hạn đa thức (trong κ) là điển hình. Xác suất một phép thực thi không điển hình được giới hạn bởi $\exp(-\Omega(\min\{m, \kappa\}) + \ln L)$ trong đó L là tổng thời gian chạy.

Hãy nhớ lại (Ghi chú 2) rằng cài đặt động (cụ thể là việc sử dụng các hàm tính toán lại mục tiêu) mang lại nhiều cơ hội hơn cho các cuộc tấn công của kẻ tấn công [2]. Bổ đề trung gian quan trọng sau đây cho thấy nếu phép thực thi thông thường tốt đến một điểm nhất định thì các chuỗi bị kẻ tấn công khai thác riêng trong thời gian dài sẽ không được các bên trung thực chấp nhận.

Bổ đề 2 (Không chính thức). Giả sử E là một phép thực thi điển hình trong môi trường (γ, s) được vận hành tốt. Nếu E_r là (η, θ)-tốt thì không có bên trung thực nào chấp nhận ở vòng $r+1$ một chuỗi chưa được bên trung thực kéo dài ít nhất trong $O\left(\frac{m}{\tau f}\right)$ vòng liên tiếp.

Một hệ quả dễ hiểu ở trên là trong các phép thực thi điển hình, chuỗi của các bên trung thực không thể chứa các Block có dấu thời gian khác quá nhiều so với thời gian tạo thực tế của Block.

Hệ quả 1 (Không chính thức). Giả sử E là một phép thực thi điển hình trong môi trường (γ, s) được vận hành tốt. Nếu E_{r-1} là (η, θ)-tốt thì dấu thời gian của bất kỳ Block nào trong E_r tối đa là $O\left(\frac{m}{\tau f}\right)$ so với thời gian tạo thực tế của nó (xem khái niệm về *độ chính xác* trong Định nghĩa 6).

Các kết quả quan trọng khác mà chúng tôi thu được liên quan đến phép thực thi (η, θ)-tốt là các Epoch của chúng kéo dài đến mức cần thiết (Bổ đề 3), cũng như thuộc tính "tự sửa", nếu mọi chuỗi được áp dụng bởi một bên trung thực là $\left(\eta\gamma, \frac{\theta}{\gamma}\right)$-tốt trong E_{r-1} (xem khái niệm về một *chuỗi tốt* trong Định nghĩa 5), thì E_r là (η, θ)-tốt (Hệ quả 2). Những điều trên (cùng với một số bước trung gian nhỏ hơn mà chúng tôi bỏ qua trong phần tổng quan cấp cao này) cho phép chúng tôi kết luận:

Định lý 1 (Không chính thức). Một phép thực thi điển hình trong môi trường (γ, s) được vận hành tốt là $O\left(\frac{m}{\tau f}\right)$-chính xác và (η, θ)-tốt.

Tiền tố chung và chất lượng chuỗi. Các phép thực thi điển hình cung cấp cho chúng ta hai thuộc tính bậc thấp mong muốn của Blockchain:

Định lý 2 và 4 (Không chính thức). Giả sử E là một phép thực thi điển hình trong môi trường (γ, s) được vận hành tốt. Theo yêu cầu của Bảng 1 (Phần 6.1), tiền tố chung đúng cho mọi $k \geq \theta\gamma m/8\tau$ và chất lượng chuỗi đúng cho $\ell = m/16\tau f$ và $\mu \leq 1 - \delta/2$, trong đó với mọi r, $t_r < n_r(1-\delta)$.

Sổ cái giao dịch mạnh mẽ. Với những điều trên, chúng tôi chứng minh các thuộc tính của sổ cái giao dịch mạnh mẽ:

Định lý 5 và 6 (Không chính thức). Theo yêu cầu của Bảng 1, giao thức hệ thống nền tảng đáp ứng tính bền vững với tham số $k = \Theta(m)$ và tính sống động với thời gian chờ $u = \Omega(m + k)$ cho độ sâu k. Hãy tham khảo Phần 6 để biết phân tích đầy đủ về giao thức.

6 Phân tích đầy đủ

Trong phần này, chúng tôi trình bày phân tích và bằng chứng đầy đủ về giao thức hệ thống nền tảng và ứng dụng sổ cái giao dịch mạnh mẽ với các chuỗi có độ khó khác nhau. Phân tích ở mức độ cao tuân theo lộ trình được trình bày trong Phần 5.

6.1 Ký hiệu, định nghĩa và mệnh đề bổ sung

Không gian xác suất của chúng tôi bao gồm tất cả các phép thực thi có độ dài tối đa là một số đa thức trong κ. Về mặt hình thức, tập hợp các kết quả cơ bản có thể được định nghĩa là một tập hợp các chuỗi mã hóa mọi biến của mỗi bên trong mỗi vòng của phép thực thi giới hạn đa thức. Chúng tôi sẽ không đi sâu vào hình thức như vậy và không nói rõ các chi tiết. Chúng tôi sẽ biểu thị **Pr** là số đo xác suất của không gian này. Cũng xác định biến ngẫu nhiên $\mathcal{E}$ lấy các giá trị trên không gian này và với sự phân phối các đồng Coin ngẫu nhiên của tất cả các thực thể (kẻ tấn công, môi trường, các bên) và Oracle ngẫu nhiên.

Giả sử ở vòng r có chính xác n bên truy vấn Oracle với mục tiêu T. Xác suất thành công ít nhất là:

$$f(T, n) = 1 - \left(1 - \frac{T}{2^\kappa}\right)^{qn}.$$

Đối với mục tiêu ban đầu T_0 và ước tính ban đầu về số lượng bên n_0, chúng tôi biểu thị $f_0 = f(T_0, n_0)$. Trong tương lai, mục tiêu của cơ chế tính toán lại mục tiêu sẽ là duy trì mục tiêu T cho mỗi bên sao cho $f(T, n_r) \approx f_0$ cho tất cả các vòng r. Vì lý do này, chúng tôi sẽ bỏ chỉ số dưới của f_0 và chỉ gọi nó là f; để tránh nhầm lẫn, bất cứ khi nào chúng ta đề cập đến hàm $f(\cdot,\cdot)$, chúng tôi sẽ chỉ định hai toán hạng của nó.

Lưu ý rằng $f(T, n)$ giảm và tăng theo n và T. Đặc biệt, Sự thật 2 được áp dụng. Mệnh đề sau đây cung cấp các giới hạn hữu ích trên $f(T, n)$. Để thuận tiện, hãy xác định $p = q/2^\kappa$.

Mệnh đề 1. *Với các số nguyên dương κ, q, T, n và $f(T, n)$ được định nghĩa như trên,*

$$\frac{pTn}{1+pTn} \leq f(T,n) \leq pTn \leq \frac{f(T,n)}{1-f(T,n)}, \text{ trong đó } p = \frac{q}{2^\kappa}.$$

Chứng minh. Các giới hạn có thể đạt được bằng cách sử dụng các bất đẳng thức $(1 - x)^{\alpha} \geq 1 - x\alpha$, hợp lệ với $x \leq 1$ và $\alpha \geq 1$, và $e^{-x} \leq \frac{1}{1+x}$, hợp lệ với $x \geq 0$. □

Tại vòng r của phép thực thi E, các bên trung thực có thể truy vấn Oracle ngẫu nhiên để tìm các mục tiêu khác nhau. Chúng tôi biểu thị $T_r^{min}(E)$ và $T_r^{max}(E)$ là mức tối thiểu và tối đa đối với các mục tiêu đó. Chúng tôi nói r là điểm tính toán lại mục tiêu của chuỗi hợp lệ $\mathcal{C}$, nếu có một Block có dấu thời gian r và m chia chính xác số Block cho đến (và bao gồm) Block này.

Bây giờ chúng tôi xác định hai thuộc tính mong muốn của phép thực thi sẽ rất quan trọng trong phân tích. Sau này chúng tôi sẽ chỉ ra rằng hầu hết các phép thực thi đều có những đặc tính này.

Định nghĩa 5. *Xét phép thực thi E và các hằng số* $\eta \in (0, 1]$ *và* $\theta \in [1, \infty)$*. Điểm tính toán lại mục tiêu r trong chuỗi* $\mathcal{C}$ *trong E là* (η, θ)*-tốt nếu mục tiêu T mới thỏa mãn* $\eta f \leq f(T, n_r) \leq \theta f$*. Chuỗi* $\mathcal{C}$ *trong E là* (η, θ)*-tốt nếu tất cả các điểm tính toán lại mục tiêu của nó là* (η, θ)*-tốt. Một vòng r là* (η, θ)*-tốt trong E nếu* $\eta f \leq f(T_r^{min}(E), n_r)$ *và* $f(T_r^{max}(E), n_r) \leq \theta f$*. Chúng ta nói rằng E là* (η, θ)*-tốt nếu mọi vòng của E đều là* (η, θ)*-tốt.*

Đối với vòng r, tập hợp chuỗi dưới đây được quan tâm. Nó chứa đựng những chuỗi có khả năng thuộc về một bên trung thực, bên cạnh những chuỗi mà các bên trung thực có.

$$S_r = \left\{ \mathcal{C} \in E_r \;\middle|\; \begin{array}{l} \text{"C thuộc về một bên trung thực" hoặc} \\ \text{"đối với chuỗi } \mathcal{C}' \text{ của một bên trung thực } \mathbf{diff}(\mathrm{C}) > \mathbf{diff}(\mathcal{C}')\text{" hoặc} \\ \text{"đối với chuỗi } \mathcal{C}' \text{ của một bên trung thực } \mathbf{diff}(\mathrm{C}) = \mathbf{diff}(\mathcal{C}') \text{ và} \\ \qquad \mathrm{head}(\mathcal{C}) \text{ được tính toán không muộn hơn } \mathrm{head}(\mathcal{C}')\text{"} \end{array} \right\}$$

trong đó $\mathcal{C} \in E_r$ có nghĩa là $\mathcal{C}$ tồn tại và hợp lệ ở vòng r.

Định nghĩa 6. *Xét một phép thực thi E. Với* $\epsilon \in [0, \infty)$*, một Block được tạo ở r là* ϵ-chính xác *nếu nó có dấu thời gian* r' *sao cho* $|r' - r| \leq \epsilon m/f$*. Chúng ta nói* E_r *là* ϵ-chính xác *nếu không có chuỗi nào trong* S_r *chứa Block không phải là* ϵ*-chính xác. Chúng ta nói rằng E là* ϵ-chính xác *nếu với mỗi vòng r trong phép thực thi,* E_r *là* ϵ-chính xác.

Bước tiếp theo là xác định tập hợp phép thực thi điển hình. Để đạt được mục đích này, chúng tôi xác định thêm một số đại lượng và biến ngẫu nhiên.

Trong một phép thực thi thực tế E, các bên trung thực có thể được chia thành các chuỗi khác nhau với các mục tiêu có thể khác nhau. Chúng tôi sẽ chỉ nghiên cứu tiến trình của các bên trung thực khi mục tiêu của họ nằm trong phạm vi hợp lý. Với khả năng cao, những bên trung thực luôn làm việc với những mục tiêu hợp lý. Đối với một vòng r, một tập hợp các vòng liên tiếp S, và hằng số $\eta \in (0, 1)$, giả sử

$$T^{(r,\eta)} = \frac{\eta f}{p n_r} \text{ và } T^{(S,\eta)} = \min_{r \in S} T^{(r,\eta)}.$$

Để làm sáng tỏ định nghĩa của $T^{(r,\eta)}$, hãy lưu ý rằng trong một vòng (η, θ)-tốt, tất cả các bên trung thực đều truy vấn mục tiêu ít nhất là $T^{(r,\eta)}$. Bây giờ chúng tôi xác định cho mỗi vòng r một biến ngẫu nhiên thực D_r bằng *độ khó tối đa* trong số tất cả các Block có mục tiêu ít nhất là $T^{(r,\eta)}$ được tính toán bởi các bên trung thực ở vòng r. Đồng thời xác định Q_r bằng D_r khi có chính xác một Block được tính toán bởi một bên trung thực và bằng 0 nếu ngược lại.

Về kẻ tấn công, chúng tôi sẽ quan tâm đến khoảng thời gian mà anh ta đã tập hợp được một số Block theo thứ tự m. Vì mục tiêu của các Block có thể thay đổi, nên cần xem xét độ khó mà kẻ tấn công gặp phải không phải trong một tập hợp các vòng liên tiếp mà là trong một tập hợp các truy vấn tấn công liên tiếp có thể kéo dài một số vòng và không nhất thiết phải là một bội số của q.

Đối với một tập hợp các truy vấn liên tiếp được lập chỉ mục bởi một tập hợp J, chúng tôi xác định giá trị sau đây sẽ đóng vai trò là ngưỡng cho các mục tiêu của các Block mà kẻ tâns công cố gắng thực hiện.

$$T^{(J)} = \frac{\eta(1-\delta)(1-2\epsilon)(1-\theta f)}{32\tau^3\gamma} \cdot \frac{m}{|J|} \cdot 2^{\kappa}$$

Với ngưỡng trên, đối với $j \in J$, nếu kẻ tấn công tính toán ở truy vấn thứ j của anh ta một Block có độ khó tối đa $1/T^{(J)}$, thì cho biến ngẫu nhiên $A_j^{(J)}$ bằng độ khó của Block này; ngược lại, thì $A_j^{(J)} = 0$. Định nghĩa trên gợi ý rằng chúng tôi thu thập trong $A_j^{(J)}$ độ khó mà kẻ tấn công đạt được miễn là nó tương ứng với các Block không quá khó (tức là những Block có mục tiêu nhỏ hơn $T^{(J)}$). Với tầm nhìn xa, chúng tôi lưu ý rằng điều này sẽ cho phép lập luận tập trung cho biến ngẫu nhiên $A_j^{(J)}$. Chúng tôi thường bỏ ký tự trên (J) khỏi A.

Cho $\mathcal{E}_{r-1}$ chứa thông tin của phép thực thi ngay trước vòng r. Cụ thể, giá trị E_{r-1} của $\mathcal{E}_{r-1}$ xác định các mục tiêu mà mỗi bên sẽ truy vấn Oracle ở vòng r, nhưng nó không xác định D_r hoặc Q_r. Nếu E là một phép thực thi cố định (tức là $\mathcal{E} = E$), biểu thị $D_r(E)$ và $Q_r(E)$ là giá trị của D_r và Q_r trong E. Nếu một tập hợp các truy vấn liên tiếp J được xem xét thì với $j \in J$, $A_j^{(J)}(E)$ được xác định tương tự. Trong trường hợp này, chúng tôi cũng sẽ viết $\mathcal{E}_j^{(J)}$ cho phép thực thi ngay trước truy vấn thứ j của kẻ tấn công.

Đối với các biến ngẫu nhiên được xác định ở trên, giới hạn sau đây sẽ hữu ích vì nó đưa ra ước tính về tiến độ mà các bên trung thực đã đạt được trong phép thực thi (η, θ)-tốt. Lưu ý rằng chúng tôi quan tâm đến tiến trình đến từ *các vòng thành công duy nhất*, trong đó chính xác một bên trung thực đã tính toán POW. Độ khó dự kiến sẽ được n_r bên trung thực tính toán ở vòng r là pn_r. Tuy nhiên, việc tính toán POW càng dễ thì $\mathbf{E}[Q_r|\mathrm{E}_{r-1} = E_{r-1}]$ sẽ càng nhỏ hơn đối với giá trị này. Vì phép thực thi là (η, θ)-tốt, nên một POW được tính toán bởi các bên trung thực với xác suất cao nhất là θf. Điều này chứng minh sự xuất hiện của $(1 - \theta f)$ trong giới hạn.

Mệnh đề 2. *Nếu vòng r là (η, θ)-tốt trong E, thì* $\mathbf{E}[Q_r|\mathrm{E}_{r-1} = E_{r-1}] \geq (1 - \theta f)pn_r$.

Chứng minh. Chúng ta hãy bỏ chỉ số r để thuận tiện. Giả sử các bên trung thực truy vấn các mục tiêu $T_1, T_2, \ldots, T_n$ tương ứng. Chúng tôi sẽ cung cấp một giới hạn dưới giả định rằng các bên trung thực thực hiện tất cả các truy vấn q (ngay cả sau khi thành công) và tính tổng xác suất của mỗi truy vấn rằng đó là truy vấn thành công duy nhất. Chúng ta có

$$\mathbf{E}[Q_r|\mathcal{E}_{r-1} = E_{r-1}] \sum_{i\in[n]} q \cdot \frac{1}{T_i} \cdot \frac{T_i}{2^\kappa}\left(1 - \frac{T_i}{2^\kappa}\right)^{q-1} \prod_{i\neq j\in[n]} \left[1 - f(T_j, 1)\right] \geq \sum_{i\in[n]} p \prod_{j\in[n]} \left[1 - f(T_j, 1)\right]$$

$$\geq \sum_{i\in[n]} p \prod_{j\in[n]} [1 - f(T^{\max}, 1)] = \sum_{i\in[n]} p[1 - f(T^{\max}, n)] \geq \sum_{i\in[n]} p(1 - \theta f) = (1 - \theta f)pn,$$

trong đó bất đẳng thức thứ ba đúng vì $f(T, n)$ đang tăng trong T. □

Các thuộc tính mà chúng tôi đã xác định sẽ được hiển thị để giữ trong môi trường (γ, s) được vận hành tốt, đối với γ và s phù hợp. Sự thật đơn giản sau đây là hệ quả của định nghĩa.

Sự thật 1. *Trong môi trường (γ, s) được vận hành tốt, đối với bất kỳ tập hợp S các vòng liên tiếp có $|S| \leq s$, mọi $S' \subseteq S$, và mọi $n \in \{n_r : r \in S\}$,*

$$\frac{1}{\gamma} \cdot n \leq \frac{1}{|S'|} \cdot \sum_{r\in S'} n_r \leq \gamma \cdot n$$

Chứng minh. Trung bình của một số con số được giới hạn bởi giá trị tối thiểu và tối đa của chúng. Hơn nữa, định nghĩa (γ, s) được vận hành tốt nghĩa là $\min_{r\in S} n_r \geq \frac{1}{\gamma}\max_{r\in S} n_r \geq \frac{1}{\gamma}n$ và $\max_{r\in S} n_r \leq \gamma \min_{r\in S} \leq \gamma n$. Như vậy,

$$\frac{1}{\gamma} \cdot n \leq \min_{r\in S} n_r \leq \min_{r\in S'} n_r \leq \frac{1}{|S'|} \cdot \sum_{r\in S'} n_r \leq \max_{r\in S'} n_r \leq \max_{r\in S} n_r \leq \gamma \cdot n.$$

□

Phân tích của chúng tôi bao gồm một số tham số có liên quan phù hợp. Bảng 1 tóm tắt chúng, nhắc lại định nghĩa của chúng và liệt kê tất cả các ràng buộc mà chúng phải đáp ứng.

n_r: số lượng các bên khai thác trung thực trong vòng r. t_r: số lượng bên hoạt động gian lận. δ: lợi thế của các bên trung thực, $\forall r(t_r/n_r < 1-\delta)$ (γ, s): xác định số bên dao động như thế nào qua các vòng, xem Định nghĩa 1. f: xác suất ít nhất một bên trung thực thành công trong một vòng giả sử có n_0 bên và mục tiêu T_0 (tham số khởi tạo của giao thức). τ: bộ lọc giảm thiểu, xem Định nghĩa 2. (η, θ): giới hạn dưới và giới hạn trên xác định mức độ tốt của phép thực thi, xem Định nghĩa 5. ϵ: chất lượng tập trung của các biến ngẫu nhiên trong các phép thực thi điển hình, xem Định nghĩa 8. m: độ dài của một Epoch tính theo số Block. L: tổng thời gian chạy của hệ thống.
[(R0)] $\forall r: t_r \leq (1-\delta)n_r$ [(R1)] $s \geq \frac{\tau m}{f} + \frac{m}{8\tau f}$ [(R2)] $\delta/2 \geq 2\epsilon + \theta f$ [(R3)] $\tau - 1/8\tau > 1/(1-\epsilon)(1-\theta f)\,\eta$ [(R4)] $17(1+\epsilon)\theta \leq 8\tau(\gamma - \theta f)$ [(R5)] $9(1+\epsilon)\eta\gamma^2 \leq 4(1-\eta\gamma f)$ [(R6)] $7\theta(1-\epsilon)(1-\theta f) \geq 8\gamma^2$

Bảng 1: Các tham số hệ thống và yêu cầu đối với chúng. Các tham số như sau: các số nguyên dương s, m, L; các số thực dương f, γ, δ, ϵ, τ, η, θ, trong đó f, ϵ, $\delta \in (0,1)$ và $0 < \eta \leq 1 \leq \theta$.

Ghi chú 3. *Để tham số hóa thực tế các tham số τ, m, f của Bitcoin*[5]*, tức là* $\tau = 4$, $m = 2016$, $f = 0{,}03$, *so với các ràng buộc của Bảng 1, chúng có thể được thỏa mãn với* $\delta = 0{,}99$, $\eta = 0{,}268$, $\theta = 1{,}995$, $\epsilon = 2{,}93 \cdot 10^{-8}$, *với* $\gamma = 1{,}281$ *và* $s = 2{,}71 \cdot 10^5$. *Chúng tôi cho rằng s đo số vòng trong đó sự biến động của γ có thể xảy ra, các ràng buộc có thể thỏa mãn đối với mức dao động lên tới 28% cứ sau khoảng 2 tháng (xem xét một vòng kéo dài 18 giây).*

6.2 Bổ đề tăng trưởng chuỗi

Bây giờ chúng tôi chứng minh bổ đề tăng trưởng chuỗi. Bổ đề này đã xuất hiện trong [11], nhưng nó đề cập đến số Block thay vì độ khó. Trong [15] "tăng trưởng chuỗi" xuất hiện lần đầu và tác giả nêu rõ đặc tính tăng trưởng chuỗi.

Một cách không chính thức, bổ đề này nói rằng các bên trung thực sẽ đạt được nhiều tiến bộ tương ứng với số lượng POW mà họ thu được. Mặc dù chứng minh đơn giản nhưng bổ đề tăng trưởng chuỗi rất quan trọng, bởi vì nó cho thấy rằng dù kẻ tấn công có làm gì thì các bên trung thực sẽ tiến lên (xét về độ khó tích lũy) ít nhất bằng độ khó của POW mà họ có được.

Bổ đề 1. *Cho E là phép thực thi bất kỳ. Giả sử tại vòng u, một bên trung thực gặp phải một chuỗi có độ khó d. Khi đó, đến vòng* $v \geq u$, *mỗi bên trung thực ít nhất sẽ phải nhận một chuỗi có độ khó*

$$d + \sum_{u \leq r < v} D_r(E)$$

[5]Lưu ý rằng để tính f, chúng ta có thể coi một vòng tương tác đầy đủ kéo dài 18 giây; nếu điều này kết hợp với thực tế là mục tiêu được đặt để phát hiện POW khoảng 10 phút một lần, chúng ta có 18/600 = 0,3 là một ước lượng tốt cho f.

Chứng minh. Bằng quy nạp theo $v - u$. Về cơ bản, $v = u$ và $d + \sum_{r=u}^{v-1} D_r(E) = d$. Nếu ở vòng u, một bên trung thực có chuỗi $\mathcal{C}$ có độ khó d, thì bên đó sẽ phát tán $\mathcal{C}$ ở vòng sớm hơn u. Theo đó, mọi bên trung thực sẽ nhận được $\mathcal{C}$ theo vòng u.

Đối với bước quy nạp, lưu ý rằng theo giả thuyết quy nạp, mọi bên trung thực đều nhận được một chuỗi có độ khó ít nhất là $d' = d + \sum_{r=u}^{v-1} D_r$ ở vòng v. Khi $D_v = 0$ mệnh đề được suy ra trực tiếp, vì vậy giả sử $D_v > 0$. Vì mọi bên trung thực đều truy vấn Oracle với một chuỗi có độ khó ít nhất là d' ở vòng v, nếu suy ra thì một bên trung thực thành công ở vòng v sẽ phát tán ít nhất một chuỗi có độ khó $d' + D_v = d + \sum_{r=u}^{v} D_r$. □

6.3 Các phép thực thi điển hình: Định nghĩa và các bằng chứng liên quan

Bây giờ chúng tôi có thể định nghĩa chính thức khái niệm về các phép thực thi *điển hình*. Ý tưởng mà định nghĩa này nắm bắt được như sau. Giả sử chúng tôi kiểm tra một phép thực thi E. Lưu ý rằng tại mỗi vòng E, các bên thực hiện phép thử Bernoulli với xác suất thành công có thể bị ảnh hưởng bởi kẻ tấn công. Với phép thực thi, những thử nghiệm này được xác định và chúng tôi có thể tính toán tiến độ dự kiến mà các bên đạt được dựa trên xác suất tương ứng. Sau đó chúng tôi so sánh giá trị này với tiến độ thực tế, nếu chênh lệch hợp lý thì chúng tôi khai báo E là *điển hình*. Tuy nhiên, lưu ý rằng việc xem xét riêng sự khác biệt này không phải lúc nào cũng đủ vì phương sai của quá trình có thể quá cao. Theo Định lý 7, định nghĩa nói rằng hoặc có sự khác biệt cao đối với tập hợp các vòng mà chúng tôi đang xem xét, hoặc các bên đã đạt được tiến bộ trong các vòng này như mong đợi.

Ngoài hành vi của các biến ngẫu nhiên được mô tả ở trên, phép thực thi điển hình cũng sẽ có đặc điểm là không có một số biến cố xấu về hàm Hash cơ bản $H(\cdot)$ được sử dụng trong bằng chứng công việc và được mô hình hóa như một Oracle ngẫu nhiên. Các biến cố xấu mà chúng tôi quan tâm được xác định như sau (hãy nhớ là thời gian tạo Block là vòng mà nó được tạo thành công bằng một truy vấn tới Oracle ngẫu nhiên bởi kẻ tấn công hoặc một bên trung thực).

Định nghĩa 7. *Việc* chèn thêm *xảy ra khi, cho một chuỗi $\mathcal{C}$ có hai Block liên tiếp B và B', một Block B^* được tạo sau B' sao cho B, B^*, B' tạo thành ba Block liên tiếp của một chuỗi hợp lệ. Một* bản sao *xảy ra nếu cùng một Block tồn tại ở hai vị trí khác nhau. Một* dự đoán *xảy ra khi một Block mở rộng một Block với thời gian tạo muộn hơn.*

Với những điều trên, chúng tôi hiện đã sẵn sàng để xác định thế nào là một phép thực thi điển hình.

Định nghĩa 8 (Phép hực thi điển hình)**.** *Một phép thực thi E là* (ϵ, η, θ)-điển hình *nếu có những điều sau:*

(a) Nếu, với bất kỳ tập S các vòng liên tiếp, $pT^{(S,\eta)} \sum_{r \in S} n_r \geq \frac{\eta m}{16\tau\gamma}$, *thì*

$$\sum_{r \in S} Q_r(E) > \sum_{r \in S} \mathbf{E}[Q_r | \mathcal{E}_{r-1} = E_{r-1}] - \epsilon(1 - \theta f)p \sum_{r \in S} n_r$$

và

$$\sum_{r \in S} D_r(E) < (1 + \epsilon)p \sum_{r \in S} n_r.$$

.

(b) Đối với bất kỳ tập J nào lập chỉ mục cho một tập hợp các truy vấn liên tiếp của kẻ tấn công, chúng ta có

$$\sum_{j \in J} A_j(E) < (1 + \epsilon)2^{-\kappa}|J|$$

và các Block có mục tiêu nhỏ hơn $\tau T^{(J)}$ *mà kẻ tấn công thu được nhỏ hơn* $\frac{\eta(1-\epsilon)(1-\theta f)}{32\tau^2\gamma}$.

(c) Không có phần chèn thêm, không có bản sao và không có dự đoán nào xảy ra trong E .

Ghi chú 4. *Lưu ý rằng nếu J lập chỉ mục các truy vấn của kẻ tấn công trong tập S gồm các vòng liên tiếp thì* $|J| = q\sum_{r\in S} t_r$ *và bất đẳng thức trong Định nghĩa 8(b) là* $\sum_{j\in J} A_j(E) < (1+\epsilon)p\sum_{r\in S} t_r$.

Mệnh đề tiếp theo đơn giản hóa việc áp dụng Định nghĩa 8(a).

Mệnh đề 3. *Giả sử E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt. Đối với bất kỳ tập S các vòng liên tiếp với* $|S| \geq \frac{m}{16\tau f}$,

$$\sum_{r\in S} D_r < (1+\epsilon)p\sum_{r\in S} n_r\,.$$

Ngoài ra, nếu E là (*η, θ*)*-tốt thì*

$$\sum_{r\in S} Q_r > (1-\epsilon)(1-\theta f)p\sum_{r\in S} n_r$$

và bất kỳ Block nào được tính toán bởi một bên trung thực ở bất kỳ vòng r nào đều tương ứng với mục tiêu ít nhất là $T^{(r,\eta)}$*, do đó đóng góp vào các biến ngẫu nhiên* D_r *và* Q_r *(nếu r là thành công duy nhất).*

Chứng minh. Đầu tiên chúng tôi chia *S* thành nhiều phần có kích thước ít nhất là $\frac{m}{16\tau f}$ và nhiều nhất là *s*. Theo Mệnh đề 2, đối với cả hai bất đẳng thức, chúng ta chỉ cần kiểm tra phần 'nếu' của Định nghĩa 8(a) cho mỗi phần S' của *S*. Thật vậy, theo các xác định của $T^{(S',\eta)}$ và Sự thật 1, $pT^{(S',\eta)}\sum_{r\in S'} n_r \geq \eta f\,|S'|/\gamma \geq \frac{\eta m}{16\tau\gamma}$. Phần cuối cùng, theo cách xác định của $T^{(r,\eta)}$, tương đương với *r* là (*η*, *θ*)-tốt. □

Hầu hết tất cả các phép thực thi giới hạn đa thức (trong *κ*) đều điển hình:

Mệnh đề 4. *Giả sử hệ thống ITM* $(\mathcal{Z}, C)$ *chạy L bước, biến cố "*$\mathcal{E}$ *không điển hình" được giới hạn bởi* $\exp(-\Omega(\min\{m,\kappa\}) + \ln L)$. *Cụ thể, giới hạn là* $\exp\left\{-\frac{\eta\epsilon^2(1-2\delta)}{64\tau^3\gamma}m - \ln(2)(\kappa-1) + 4\ln L + 2\ln 2\right\}$.

Chứng minh. Vì độ dài *L* của phép thực thi là cố định nên chúng tôi sẽ chứng minh giới hạn đã nêu cho một tập hợp cố định các vòng liên tiếp *S*, sau đó áp dụng giới hạn liên kết trên tất cả các tập hợp đó theo độ dài của phép thực thi. Gọi *k* là kích thước của *S* và xác định nó mà không mất tính tổng quát với $[k] = \{1, 2,..., k\}$. Đối với phần (a), xác định chuỗi các biến ngẫu nhiên bởi

$$X_0 = 0;\ X_r = \sum_{i\in[r]} Q_i - \sum_{i\in[r]} \mathbf{E}[Q_i|\mathcal{E}_{i-1}], r \in [k].$$

Điều này tạo thành một Martingale đối với dãy $\mathcal{E}_0, \mathcal{E}_1,..., \mathcal{E}_k$, bởi vì (nhắc lại các thuộc tính cơ bản của kỳ vọng có điều kiện [17]),

$$\mathbf{E}[X_r|\mathcal{E}_{r-1}] = \mathbf{E}[Q_r - \mathbf{E}[Q_r|\mathcal{E}_{r-1}]|\mathcal{E}_{r-1}] + \mathbf{E}[X_{r-1}|\mathcal{E}_{r-1}] = X_{r-1}$$

Cụ thể, điều trên xuất phát từ tính tuyến tính của kỳ vọng có điều kiện và thực tế X_{r-1} là hàm xác định của $\mathcal{E}_{r-1}$.

Bây giờ giả sử bất đẳng thức đầu tiên của Định nghĩa 8(a) không đúng. Xác suất của biến cố này là

$$\Pr[X_k \leq X_0 - t], \text{với } t = \epsilon(1-\theta f)p\sum_{r\in S} n_r.$$

Đối với *b* và *V* được xác định theo Định lý 7, chúng ta có $b \leq 1/T^{(S,\eta)}$ và $V \leq v$, trong đó $v = p\sum_{r\in S} n_r/T^{(S,\eta)}$. Để chứng minh giới hạn trên *V*, lưu ý rằng

$$\mathrm{var}(X_r - X_{r-1}|\mathcal{E}_{r-1}) = \mathbf{E}[(Q_r - \mathbf{E}[Q_r|\mathcal{E}_{r-1}])^2|\mathcal{E}_{r-1}] = \mathbf{E}[Q_r^2|\mathcal{E}_{r-1}] - (\mathbf{E}[Q_r|\mathcal{E}_{r-1}])^2.$$

Vì vậy, chỉ cần chỉ ra $\mathbf{E}[Q_r^2|\mathcal{E}_{r-1}] \le pn_r/T^{(r,\eta)} \le pn_r/T^{(S,\eta)}$. Cuối cùng, giả sử các bên trung thực ở vòng r được chia thành ℓ chuỗi với các mục tiêu tương ứng $T^{(r,\eta)} \le T_1 \le T_2 \le \cdots \le T_\ell$. Giả sử $\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_\ell$, với $\hat{n}_1 + \cdots + \hat{n}_\ell \le n_r$, là số bên tương ứng của mỗi chuỗi. Khi đó, với mọi E_{r-1},

$$\mathbf{E}[Q_r^2|\mathcal{E}_{r-1} = E_{r-1})] \le \sum_{i\in[\ell]} \frac{1}{T_i^2} \cdot f(T_i, \hat{n}_i) \cdot \prod_{j\ne i}\left[1 - f\left(T_j, \hat{n}_j\right)\right] \le \sum_{i\in[\ell]} \frac{p\hat{n}_i}{T_i} \le \frac{pn_r}{T^{(r,\eta)}}.$$

Áp dụng Định lý 7 cho $-X_0, -X_1, \ldots$, lưu ý $V \le v$ luôn đúng. Nhớ lại Yêu cầu (R2) và $pT^{(S,\eta)} \sum_{r\in S} n_r \ge \frac{\eta m}{16\tau\gamma}$, chúng ta có được

$$\mathbf{Pr}[X_k \le X_0 - t] \le \exp\left\{-\frac{3\epsilon^2(1-\theta f)^2\eta m}{32(3+\epsilon)\tau\gamma}\right\} \le \exp\left\{-\frac{\epsilon^2(1-\delta)\eta m}{32\tau\gamma}\right\}.$$

Đối với các giới hạn $\sum_{r\in S} D_r(E)$ và $\sum_{j\in J} A_j(E)$, chứng minh tuân theo cùng một hướng. Cụ thể, thay Q bằng D và A (trong trường hợp A, Martingale sẽ được lập chỉ mục bởi J) và lưu ý rằng trong những trường hợp này, Martingale không cần phải bị phủ định.

Chi tiết hơn, về giới hạn của D trong phần (a), sử dụng ký hiệu tương tự như trên, chúng ta có

$$\mathbf{E}[D_r|\mathcal{E}_{r-1} = E_{r-1})] = \sum_{i\in[\ell]} \frac{1}{T_i} \cdot f(T_i, \hat{n}_i) \cdot \prod_{j=1}^{i-1}[1 - f(T_i, \hat{n}_i)] \le \sum_{i\in[\ell]} p\hat{n}_i \le pn_r$$

và vì thế

$$\sum_{r\in S} \mathbf{E}[D_r|\mathcal{E}_{r-1}] \le p\sum_{r\in S} n_r.$$

Một lập luận tương tự cung cấp giới hạn $\mathbf{E}[D_r^2|\mathcal{E}_{r-1} = E_{r-1})] \le pn_r/T^{(r,\eta)}$ từ đó chúng tôi có thể thu được giới hạn $V \le v = p\sum_{r\in S} n_r/T^{(S,\eta)}$.

Tiếp theo chúng tôi tập trung vào phần (b). Đầu tiên lưu ý rằng nếu $|J| \le \frac{\eta(1-\delta)(1-2\epsilon)(1-\theta f)}{32\tau^3\gamma} \cdot m$, thì $T^{(J)} \ge 2^\kappa$ và bất đẳng thức là hợp lý. Đầu tiên chúng tôi sẽ chỉ ra rằng đối với Martingale $X_0, X_1, X_2, \ldots$ được $\mathcal{E}_0^{(J)}, \mathcal{E}_1^{(J)}, \mathcal{E}_2^{(J)}$ xác định là

$$X_0 = 0;\ X_j = \sum_{i\in[j]} A_i - \sum_{i\in[j]} \mathbf{E}\left[A_i|\mathcal{E}_{i-1}^{(J)}\right], j \in J,$$

suy ra $b \le 1/T^{(J)}$ và $V \le v$ với $v = 2^{-\kappa}|J|/T^{(J)}$, với các đại lượng b và V được xác định như trong Định lý 7. Hãy xem xét tiền tố của phép thực thi $E_{j-1}^{(J)}$ và mục tiêu được kẻ tấn công chọn trong truy vấn thứ j của nó. Lưu ý rằng chúng tôi có thể liên kết giá trị đó với bất kỳ truy vấn nào có dạng (ctr, g) trong đó $g = G(r, st, x)$ bằng cách khôi phục chuỗi tương ứng với st. Nếu giá trị đó nhỏ hơn $T^{(J)}$ hoặc không được xác định thì $A_j = 0$. Vì vậy, chúng ta có $\mathbf{E}\left[A_j|\mathcal{E}_{j-1}^{(J)} = E_{j-1}^{(J)}\right] \le 2^{-\kappa}$ và $\mathbf{E}\left[A_j^2|\mathcal{E}_{j-1}^{(J)} = E_{j-1}^{(J)}\right] \le 2^{-\kappa}/T^{(J)}$, thu được

$$\sum_{j\in J} \mathbf{E}\left[A_j|\mathcal{E}_{j-1}^{(J)}\right] \le 2^{-\kappa}|J| \text{ và } V \le v.$$

Bây giờ chúng ta có những điều sau đây bằng cách thiết lập $t = \epsilon 2^{-\kappa}|J|$.

$$\mathbf{Pr}\left[\sum_{j\in J} A_j \geq (1+\epsilon)2^{-\kappa}|J|\right] = \mathbf{Pr}\left[\sum_{j\in J} A_j \geq t + 2^{-\kappa}|J|\right] \leq \mathbf{Pr}\left[\sum_{j\in J} A_j \geq t + \sum_{j\in J} \mathbf{E}\left[A_j|\mathcal{E}_{j-1}^{(J)}\right]\right]$$

$$= \mathbf{Pr}\left[\sum_{j\in J} A_j - \mathbf{E}\left[A_j\left|\mathcal{E}_{j-1}^{(J)}\right.\right] \geq t\right] \leq \exp\left\{-\frac{t^2}{2v+2bt/3}\right\} \leq \exp\left\{-\frac{3\epsilon^2 2^{-\kappa}|J|T^{(J)}}{(6+2\epsilon)}\right\}$$

$$\leq \exp\left\{-\frac{3\eta\epsilon^2(1-\delta)(1-\theta f)(1-2\epsilon)m}{64(3+\epsilon)\tau^3\gamma}\right\}.$$

Để chứng minh bất đẳng thức ở dòng cuối cùng, hãy nhớ rằng $T^{(J)} = \frac{\eta(1-\delta)(1-2\epsilon)(1-\theta f)}{32\tau^3\gamma} \cdot \frac{m}{|J|} \cdot 2^{\kappa}$ và Yêu cầu (R2).

Về phần thứ hai của (b), để giới hạn số Block mục tiêu nhỏ hơn $\tau T^{(J)}$ mà kẻ tấn công có thể có được, hãy xác định một biến ngẫu nhiên Boolean Z_j, cho mỗi $j \in J$ như sau. Nếu mục tiêu tương ứng nhỏ hơn $\tau T^{(J)}$ và truy vấn thành công thì $Z_j = 1$, nếu không thì $Z_j = 0$. Sau đó chúng ta có thể định nghĩa một Martingale như trong phần (a), bằng cách cho $k = |J|$ và thay Q bằng Z. Ta có b ≤ 1 và $V \leq 2^{-\kappa}\tau T^{(J)}$.
Từ

$$\sum_{j\in[|J|]} E\left[Z_j|\mathcal{E}_{j-1}^{(J)}\right] \leq 2^{-\kappa}\,\tau T^{(J)} = \frac{\eta(1-2\epsilon)(1-\theta f)}{32\tau^2\gamma} \cdot m$$

và $(1+\epsilon)(1-2\epsilon) < (1-\epsilon)$, với $t = \epsilon \cdot \frac{\eta(1-2\epsilon)(1-\theta f)}{32\tau^2\gamma} \cdot m$ chúng ta có: (sử dụng Yêu cầu (R2) để đơn giản hóa trong bước cuối cùng)

$$\mathbf{Pr}\left[\sum_{j\in[|J|]} Z_j \geq \frac{\eta(1-\epsilon)(1-\theta f)}{32\tau^2\gamma} \cdot m\right] \leq \mathbf{Pr}[X_k \geq X_0 + t] \leq \exp\left\{-\frac{\eta\epsilon^2(1-\delta/2)m}{32\tau^2\gamma}\right\}.$$

Với phần (c) và $i \in \{0, 1, 2, 3\}$, cho $B_i = \langle r_i, st_i, x_i, ctr_i\rangle$ và $g_i = G\,(r_i, st_i, x_i)$. Nếu một Block mở rộng hai Block riêng biệt thì đã xảy ra xung đột. Để thấy điều này, giả sử Block B_3 mở rộng hai Block riêng biệt B_1 và B_2. Khi đó $st_3 = H(ctr_1, g_1) = H(ctr_2, g_2)$; thể hiện một xung đột trong H hoặc trong G, vì B_1 và B_2 là khác nhau.

Sự tồn tại của việc chèn thêm hoặc sao chép cũng thể hiện sự xung đột. Giả sử kẻ tấn công chèn một Block B_2 trong số 2 Block hiện hữu B_1 và B_3. Sau đó, B_3 mở rộng cả B_1 và B_2, vì B_2 kéo dài B_1, $r_1 < r_2$ và các Block là khác nhau. Tương tự, nếu B_3 là bản sao của B_1 (tức là $B_3 = B_1$) thì tồn tại hai Block B_2 và B_0 riêng biệt, cả hai đều được mở rộng bởi cùng một Block. Để thấy điều này, hãy lưu ý rằng B_0 và B_2 là những Block mà B_1 và B_3 mở rộng, hoặc nếu chúng không khác biệt thì B_2 là bản sao của B_0, v.v. Cuối cùng, sẽ đạt được hai Block riêng biệt vì B_1 và B_3 được cho là nằm trên các chuỗi khác nhau. Nếu tổng thời gian chạy của hệ thống ITM là L thì nó cho rằng có nhiều nhất L truy vấn được đặt ra cho G, H. Suy ra xác suất xảy ra va chạm là $\binom{L}{2}2^{-\kappa+1} \leq 2^{-\kappa+1+2\log L}$.

Cuối cùng, lưu ý rằng, đối với nhiều vòng đa thức trong κ, xác suất để một Block được đoán xảy ra là nhỏ theo cấp số nhân trong κ. □

6.4 Các phép thực thi điển hình là tốt và chính xác

Bổ đề 2. *Cho E là một phép thực thi điển hình trong môi trường* (γ, s) *được vận hành tốt. Nếu* E_r *là* (η, θ)*-tốt thì* $\mathcal{S}_{r+1}$ *không chứa chuỗi nào chưa được một bên trung thực mở rộng trong ít nhất* $\frac{m}{16\tau f}$ *vòng liên tiếp.*

Chứng minh. Giả sử theo hướng mâu thuẫn, $\mathcal{C} \in \mathcal{S}_{r+1}$ và chưa được một bên trung thực nào mở rộng trong ít nhất $\frac{m}{16\tau f}$ vòng. Không mất tính tổng quát, chúng ta có thể giả sử rằng $r + 1$ là vòng đầu tiên như vậy.

Cho $r^* \le r$ biểu thị dấu thời gian lớn nhất trong số các Block $\mathcal{C}$ được tính toán bởi các bên trung thực ($r^* = 0$ nếu không tồn tại). Xác định $S = \{r^* + 1, ..., r\}$ và lưu ý rằng, theo giả định của chúng tôi đối với $\mathcal{C}$, $|S| \ge \frac{m}{16\tau f}$. Cho $J = \{1, \dots, q \sum_{r \in S} t_r\}$ là tập hợp chỉ mục của các truy vấn tấn công các vòng trong S. Giả sử các Block của $\mathcal{C}$ có dấu thời gian trong S bao gồm k Epoch với các mục tiêu tương ứng $T_1, ..., T_k$. Với $i \in [k]$ gọi m_i là số Block có mục tiêu T_i và $M = m_1 + \cdots + m_k$.

Kế hoạch của chúng tôi là mâu thuẫn với giả định $\mathcal{C} \in \mathcal{S}_{r+1}$, bằng cách chứng tỏ các bên trung thực đã gặp độ khó nhiều hơn kẻ tấn công. Hãy lưu ý rằng các Block $\mathcal{C}$ đã đạt được trong S tổng độ khó là $\sum_{i \in [k]} \frac{m_i}{T_i}$. Mặt khác, theo Bổ đề Tăng trưởng Chuỗi 1, tất cả các bên trung thực đều đã tiến bộ trong các vòng ở S với $\sum_{r \in S} D_r(E) \ge \sum_{r \in S} Q_r(E)$. Vì $|S| \ge \frac{m}{16\tau f}$, Mệnh đề 3 cho thấy $\sum_{r \in S} Q_r(E)$ lớn hơn $(1 - \epsilon)(1 - \theta f) p \sum_{r \in S} n_r$. Vì vậy, để đạt được mâu thuẫn, chỉ cần chứng minh

$$\sum_{i \in [k]} \frac{m_i}{T_i} \le (1 - \epsilon)(1 - \theta f) p \sum_{r \in S} n_r. \tag{1}$$

Chúng tôi tiến hành bằng cách xem xét các trường hợp trên M.

Đầu tiên, giả sử $M \ge 2M'$, trong đó $M' = \frac{\eta(1-\epsilon)(1-\theta f)}{32\tau^2\gamma} \cdot m$ (xem Định nghĩa 8(b)). Phân chia một phần của $\mathcal{C}$ với M Block này thành ℓ phần, sao cho mỗi phần có các thuộc tính sau: (1) nó chứa tối đa một điểm tính toán mục tiêu và (2) nó chứa ít nhất M' Block có cùng mục tiêu. Lưu ý rằng phân vùng như vậy tồn tại vì $M \ge 2M'$ và $M' < m$. Với $i \in [\ell]$, cho $j_i \in J$ là chỉ mục của truy vấn trong đó Block cuối cùng của phần thứ i được tính toán. Cho $J_i = \{j_{i-1} + 1, ..., j_i\}$, với $j_0 = 0$. Lưu ý rằng Định nghĩa 8(c) thể hiện $j_{i-1} < j_i$ và đây là một phân vùng của J. Nhắc lại Định nghĩa 8(b), tổng độ khó của tất cả các Block trong phần thứ i tối đa là $\sum_{j \in J_i} A_j(E)$. Điều này đúng vì đối với một trong các mục tiêu, hơn M' Block đã được tính toán trong J_i, ít nhất là $\tau T^{(J_i)}$ và các mục tiêu có nhiều nhất một điểm tính toán giữa chúng có thể khác nhau nhiều nhất là τ. Như vậy,

$$\sum_{i \in [k]} \frac{m_i}{T_i} \le \sum_{\substack{i \in [\ell] \\ j \in J_i}} A_j(E) < \sum_{i \in [\ell]} (1 + \epsilon) 2^{-\kappa} |J_i| = (1 + \epsilon) p \sum_{r \in S} t_r \le (1 + \epsilon)(1 - \delta) p \sum_{r \in S} n_r,$$

ở bước cuối cùng chúng tôi đã sử dụng Yêu cầu (R0). Yêu cầu (R1) thể hiện $(1 + \epsilon)(1 - \delta) \le (1 - \epsilon)(1 - \theta f)$; do đó, phương trình (1) suy ra kết luận trường hợp $M \ge 2M'$.

Ngược lại, $k \le 2$ và $m_1 + m_2 < 2M'$. Cho S' bao gồm $\frac{m}{16\tau f}$ vòng đầu tiên của S. Trong trường hợp này phương trình (1) đúng với S' thay cho S. Vì chúng tôi đang ở trong một môi trường (γ, s) được vận hành tốt, nên theo Sự thật 1, $\gamma \sum_{r \in S'} n_r \ge n_{r^*} |S'|$. Hơn nữa, vì r^* là (η, θ)-tốt, nên $T_1 \ge T^{(r^*, \eta)} = nf / p n_{r^*}$. Hãy nhớ rằng $T_2 \ge T_1 / \tau$, chúng ta có $\frac{m_1}{T_1} + \frac{m_2}{T_2} \le \frac{m_1 + \tau m_2}{T_1}$, mà lần lượt là nhiều nhất

$$\frac{\tau M}{T^{(r^*, \eta)}} < \frac{2\tau M' p n_{r^*}}{\eta f} \le \frac{2\tau M' p \sum_{r \in S'} n_r}{nf |S'|} \le \frac{32\tau^2 \gamma M' p \sum_{r \in S} n_r}{nm}$$

và sau khi thay M', phương trình (1) suy ra kết luận trường hợp này và chứng minh. □

Hệ quả 1. *Giả sử E là một phép thực thi điển hình trong môi trường* (γ, s) *được vận hành tốt. Nếu* E_{r-1} *là* (η, θ)*-tốt, thì* E_r *là* $\frac{m}{16\tau f}$ *-chính xác.*

Chứng minh. Giả sử theo hướng mâu thuẫn, với một số $r^* \le r$, $\mathcal{C} \in \mathcal{S}_{r^*}$ chứa một Block không phải là $\frac{m}{16\tau f}$ -chính xác, cho $u \le r^* \le r$ là dấu thời gian của Block này và v là thời gian tạo Block. Nếu $u - v > \frac{m}{16\tau f}$, thì mọi bên trung thực sẽ coi $\mathcal{C}$ là không hợp lệ trong các vòng $v, v + 1, \dots, u$. Nếu $v - u > \frac{m}{16\tau f}$, thì để $\mathcal{C}$ hợp lệ, nó không được chứa bất kỳ Block trung thực nào có dấu thời gian trong $u, u + 1, ..., v$. (Lưu ý rằng chúng tôi đang sử dụng Định nghĩa 8(c) ở

đây vì một Block có thể được chèn thêm sau.) Trong cả hai trường hợp, $\mathcal{C} \in \mathcal{S}_{r^*}$, nhưng chưa được một bên trung thực gia hạn trong ít nhất $\frac{m}{16\tau f}$ vòng. Vì E_{r^*-1} là (η, θ)-tốt nên mệnh đề tuân theo Bổ đề 2. □

Bổ đề 3. *Giả sử E là một phép thực thi điển hình trong môi trường* (γ, s) *được vận hành tốt và r^*, một điểm tính toán lại mục tiêu $\left(\eta\gamma, \frac{\delta}{\gamma}\right)$-tốt của chuỗi hợp lệ $\mathcal{C}$. Với $r > r^* + \frac{\tau m}{f}$, giả sử E_{r-1} là (η, θ)-tốt. Khi đó, khoảng thời gian Δ của Epoch $\mathcal{C}$ bắt đầu từ r^* thỏa mãn*

$$\frac{m}{\tau f} \leq \Delta \leq \frac{\tau m}{f},$$

hoặc $\mathcal{C} \notin \mathcal{S}_u$ cho mỗi $u \in \left\{r^* + \frac{\tau m}{f}, \cdots, r\right\}$.

Chứng minh. Cho T là mục tiêu của Epoch được đề cập.

Đối với giới hạn trên, giả sử $\Delta > \frac{\tau m}{f}$. Trước tiên, chúng tôi chứng minh trong các vòng $S = \left\{r^* + \frac{m}{16\tau f}, \cdots, r^* + \frac{\tau m}{f}\right\}$, các bên trung thực đã gặp phải nhiều độ khó nhiều hơn là $\frac{m}{T}$. Lưu ý rằng các vòng của S là (η, θ)-tốt vì chúng đến trước r. Do đó, theo Mệnh đề 3, độ khó mà các bên trung thực gặp phải trong S ít nhất là

$$(1-\epsilon)(1-\theta f)p \sum_{r \in S} n_r \geq (1-\epsilon)(1-\theta f)p \cdot \frac{|S| n_{r^*}}{\gamma} \geq (1-\epsilon)(1-\theta f)|S| \frac{\eta f}{T} > \frac{m}{T}.$$

Đối với bất đẳng thức thứ nhất, chúng tôi sử dụng Sự thật 1. Đối với bất đẳng thức thứ hai, hãy nhớ rằng r^* là $(\eta\gamma, \theta/\gamma)$-tốt và do đó $pTn_{r^*} \geq f(T, n_{r^*}) \geq \eta\gamma f$. Đối với bất đẳng thức cuối cùng, chúng ta thấy $|S| = \frac{m}{f}(\tau - 1/8\tau)$ và do đó tuân theo Yêu cầu (R3).

Tiếp theo, chúng ta thấy chuỗi $\mathcal{C}$ có một Block trong Epoch được đề cập được tính toán bởi một bên trung thực trong một vòng trong khoảng thời gian $[r^*, r^* + \frac{m}{16\tau f})$ hoặc bởi Bổ đề 2, $\mathcal{C} \notin \mathcal{S}_u$ với mỗi $u \in \left\{r^* + \frac{m}{16\tau f}, \cdots, r\right\} \supseteq \left\{r^* + \frac{\tau m}{f}, \ldots, r\right\}$. Giả sử điều đầu tiên xảy ra, thì đến vòng $r^* + \frac{\tau m}{f} - \frac{m}{16\tau f}$, chuỗi của các bên trung thực đã tiến lên với độ khó vượt quá tổng độ khó của Epoch được đề cập. Điều này có nghĩa là không có bên trung thực nào sẽ mở rộng $\mathcal{C}$ trong các vòng $\left\{r^* + \frac{\tau m}{f} - \frac{m}{16\tau f} + 1, \ldots, \Delta\right\}$. Vì nó được giả định $\Delta > r^* + \frac{\tau m}{f}$, nên Bổ đề 2 có thể được áp dụng để thể hiện rằng $\mathcal{C} \notin \mathcal{S}_u$ với $u \in \left\{r^* + \frac{\tau m}{f}, \ldots, r\right\}$.

Đối với giới hạn dưới, chúng tôi giả sử $\Delta < \frac{m}{\tau f}$ và $\mathcal{C} \in \mathcal{S}_u$ với một vài $u \in \{r^* + \Delta + 1, \ldots, r\}$, và tìm kiếm sự mâu thuẫn. Rõ ràng, các bên trung thực chỉ đóng góp trong suốt tập hợp các vòng $S = \{r^*, \ldots, r^* + \Delta\}$. Theo Bổ đề 2, kẻ tấn công có thể chỉ đóng góp trong quá trình $S' = \left\{r^* - \frac{m}{16\tau f}, \ldots, r^* + \Delta + \frac{m}{16\tau f}\right\}$. Cho J là tập hợp các truy vấn có sẵn cho kẻ tấn công trong các vòng trong S'. Chúng tôi chỉ ra rằng trong một phép thực thi điển hình, các bên trung thực cùng với kẻ tấn công không thể gặp độ khó $\frac{m}{T}$ trong các vòng tương ứng ở các tập hợp S và S'. Đối với các bên trung thực, Mệnh đề 3 sẽ được áp dụng. Về phía kẻ tấn công, trước tiên giả sử $T \geq T^{(J)}$ (không khó để xác minh rằng trường hợp $T < T^{(J)}$ dẫn đến một giới hạn thuận lợi hơn). Theo đó, tổng độ khó đóng góp cho Epoch này nhiều nhất là

$$(1+\epsilon)p\left(\sum_{r \in S} n_r + \sum_{r \in S'} t_r\right) \leq (1+\epsilon)p\gamma n_{r^*}(|S| + |S'|) < (1+\epsilon)p\gamma n_{r^*} \cdot \frac{17m}{8\tau f}.$$

Bất đẳng thức đầu tiên suy ra từ Sự thật 1 bằng cách sử dụng $t_r < (1-\delta)n_r$. Đối với bất đẳng thức thứ hai, thay thế giới hạn trên của kích thước S và S'. Tiếp theo, lưu ý rằng r^* là một điểm tính toán lại $(\eta\gamma, \theta/\gamma)$-tốt và do đó $f(T, n_{r^*}) \leq \theta f/\gamma$. Theo Mệnh đề 1, $pTn_{r^*} < f(T, n_{r^*})/\left(1 - f(T, n_{r^*})\right) \leq (\theta f/\gamma)/(1 - \theta f/\gamma)$. Theo đó, số lượng hiển thị cuối cùng là nhiều nhất là $\frac{17(1+\epsilon)\theta}{8\tau(\gamma - \theta f)} \cdot \frac{m}{T}$ và nhớ lại Yêu cầu (R4) ít hơn $\frac{m}{T}$ như mong muốn. □

Mệnh đề 5. *Giả sử E là một phép thực thi điển hình trong môi trường* (*γ*, *s*) *được vận hành tốt . Giả sử r là một vòng sao cho* E_{r-1} *là* (*η, θ*)*-tốt, S là tập hợp các vòng liên tiếp trong* E_r *với* $|S| \geq \frac{m}{32\tau^2 f}$ *, và J là tập hợp các truy vấn tấn công trong các vòng trong S. Khi đó, kẻ tấn công trong các vòng trong S đã đóng góp nhiều nhất* $\sum_{j\in J} A_j$ *độ khó đến* $\bigcup_{v\leq r} \mathcal{S}_r$.

Chứng minh. Không mất tính tổng quát, trong chứng minh này chúng tôi sẽ giả sử rằng $t_r = (1-\delta)n_r$ với mỗi $r \in S$. Hơn nữa, chúng tôi giả sử $|S| \leq \frac{\tau m}{f}$. Nếu không đúng như vậy thì chúng ta có thể phân chia *S* thành các phần có kích thước phù hợp và áp dụng các đối số theo sau cho mỗi tổng. Tuyên bố sẽ được suy ra khi tổng hợp tất cả các phần.

Theo Bổ đề 2, đối với bất kỳ Block *B* nào trong $\mathcal{S}_v$ với $v \leq r$, có một Block B' trong cùng một chuỗi và được tính toán ở hầu hết $\frac{m}{16\tau f}$ vòng trước nó bởi một bên trung thực. Cho *u* là vòng mà bên trung thực tính *B* và *T* mục tiêu của nó. Lưu ý rằng vì *E* là (*η*, *θ*)-tốt, $T \geq T^{(u,\eta)} = \frac{\eta f}{pn_u}$. Theo Bổ đề 3, có nhiều nhất một điểm tính toán lại giữa *B* và B', và do đó mục tiêu của *B* ít nhất là T/τ. Chúng ta cần thể hiện điều đó là $T/\tau \geq T^{(J)}$. Điều này cho thấy tất cả độ khó do kẻ tấn công gây ra đều được tính vào $\sum_{j\in J} A_j$ và theo Định nghĩa 8, chúng tôi đã hoàn thành.

Sử dụng Sự thật 1 và giới hạn dưới trên |*S*|,

$$2^{-\kappa}|J| = (1-\delta)p\sum_{r\in S} n_r \geq (1-\delta)p\cdot\frac{|S|n_u}{\gamma} \geq (1-\delta)p\cdot\frac{mn_u}{32\tau^3 f\gamma}.$$

Nhắc lại cách xác định của $T^{(J)}$ và sử dụng giới hạn này,

$$T^{(J)} = \frac{\eta(1-\delta)(1-2\epsilon)(1-\theta f)}{32\tau^3\gamma}\cdot\frac{m}{|J|}\cdot 2^{\kappa} \leq \frac{\eta f(1-2\epsilon)(1-\theta f)}{\tau p n_u} < \frac{T^{(u,\eta)}}{\tau} \leq \frac{T}{\tau},$$

như mong muốn. □

Bổ đề 4. *Giả sử E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt và* E_{r-1} *là* (*η, θ*)*-tốt. Nếu* $\mathcal{C} \in \mathcal{S}_r$ *thì* $\mathcal{C}$ *là* $(\eta\gamma, \theta/\gamma)$*-tốt trong* E_r.

Chứng minh. Giả định của chúng tôi là mọi chuỗi đều là $(\eta\gamma, \theta/\gamma)$-tốt ở vòng đầu tiên. Do đó, để chứng minh tuyên bố, chỉ cần chỉ ra rằng nếu một chuỗi là $(\eta\gamma, \theta/\gamma)$-tốt tại điểm tính toán lại r^*, thì nó cũng sẽ $(\eta\gamma, \theta/\gamma)$-tốt tại thời điểm tính toán lại tiếp theo $r^* + \Delta$.

Cho r^* và $r^* + \Delta \leq r$ là hai điểm tính toán mục tiêu liên tiếp của chuỗi $\mathcal{C}$ và *T* là mục tiêu của Epoch tương ứng. Theo Bổ đề 3 và Định nghĩa 2 của hàm tính toán lại mục tiêu, mục tiêu mới sẽ là

$$T' = \frac{\Delta}{m/f}\cdot T,$$

trong đó Δ là khoảng thời gian của Epoch.
Chúng tôi muốn cho thấy rằng

$$\eta\gamma f \leq f(T', n_{r^*+\Delta}) \leq \theta f/\gamma$$

Để đạt được mục đích này, cho $S = \{r^*, \dots, r^* + \Delta\}, S' = \left\{\max\left\{0, r^* - \frac{m}{16\tau f}\right\}, \dots, \min\left\{r^* + \Delta + \frac{m}{16\tau f}, r\right\}\right\}$ và *J* lập chỉ mục các truy vấn có sẵn cho kẻ tấn công trong S'. Lưu ý rằng, theo Hệ quả 1, mọi Block trong Epoch được tính toán bởi một bên trung thực trong một vòng ở *S* hoặc bởi kẻ tấn công trong một vòng ở S'.

Giả sử theo hướng mâu thuẫn rằng $f(T', n_{r^*+\Delta}) < \eta\gamma f f$. Sử dụng cách xác định của *f* (*T*, *n*), điều này thể hiện $qn_{r^*+\Delta}\ln\left(1-\frac{T'}{2^\kappa}\right) > \ln(1-\eta\gamma f)$. Áp dụng bất đẳng thức $-\frac{x}{1-x} < \ln(1-x) < x$, đúng cho $x \in (0, 1)$, thay thế biểu thức cho T' ở trên và sắp xếp lại, chúng ta có

$$\frac{m}{T} > \frac{1-\eta\gamma f}{\eta\gamma}\cdot p\Delta n_{r^*+\Delta}.$$

Theo Mệnh đề 3 và 5:

$$\frac{m}{T} \le 2(1+\epsilon)p \sum_{r \in S'} n_r \le 2(1+\epsilon)p \cdot \frac{\Delta + \frac{m}{8\tau f}}{|S'|} \cdot \sum_{r \in S'} n_r .$$

Theo Bổ đề 3, $\Delta \ge \frac{m}{\tau f}$. Như vậy, $\frac{\Delta + \frac{m}{8\tau f}}{\Delta} \le \frac{9}{8}$. Sử dụng điều này, Yêu cầu (R5) và kết hợp các bất đẳng thức trên $\frac{m}{T}$,

$$\gamma n_{r^*+\Delta} < \frac{9(1+\epsilon)\eta\gamma^2}{4(1-\eta\gamma f)} \cdot \frac{1}{|S'|} \sum_{r \in S'} n_r \le \frac{1}{|S'|} \sum_{r \in S'} n_r,$$

mâu thuẫn với sự thật 1.

Đối với giới hạn trên, giả sử $f(T', n_{r^*+\Delta}) > \theta f/\gamma$, (xem Mệnh đề 1) điều này thể hiện

$$\frac{m}{T} < \frac{\gamma}{\theta} \cdot p\Delta n_{r^*+\Delta}.$$

Cho $S = \left\{r^* + \frac{m}{16\tau f}, \dots, r^* + \Delta - \frac{m}{16\tau f}\right\}$. Vì một bên trung thực sở hữu $\mathcal{C}$ ở vòng *r*, nên Bổ đề 2 cho rằng có một Block được tính toán bởi một bên trung thực ở $\mathcal{C}$ trong $\left\{r^*, \dots, r^* + \frac{m}{16\tau f} - 1\right\}$ và một Block trong $\left\{r^* + \Delta - \frac{m}{16\tau f} + 1, \dots, r^* + \Delta\right\}$. Theo Bổ đề Tăng trưởng Chuỗi 1, các bên trung thực đã tính toán ít hơn $\frac{m}{T}$ độ khó trong *S* . Đặc biệt,

$$\frac{m}{T} > (1-\epsilon)(1-\theta f)p \sum_{r \in S} n_r \ge (1-\epsilon)(1-\theta f)p \cdot \frac{\Delta - \frac{m}{8\tau f}}{|S|} \cdot \sum_{r \in S} n_r.$$

Theo Bổ đề 3, $\Delta \ge \frac{m}{\tau f}$. Như vậy, $\frac{\Delta - \frac{m}{8\tau f}}{\Delta} \ge \frac{7}{8}$. Sử dụng điều này, Yêu cầu (R6) và kết hợp các bất đẳng thức trên $\frac{m}{T}$,

$$\frac{n_{r^*+\Delta}}{\gamma} > \frac{7\theta}{8\gamma^2}(1-\epsilon)(1-\theta f) \cdot \frac{1}{|S|} \sum_{r \in S} n_r \ge \frac{1}{|S|} \sum_{r \in S} n_r,$$

mâu thuẫn với Sự thật 1. □

Hệ quả 2. *Giả sử E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt và* E_{r-1} *là* (*η, θ*)*-tốt. Nếu mọi chuỗi trong* $\mathcal{S}_{r-1}$ *là* $\left(\eta\gamma, \frac{\theta}{\gamma}\right)$*-tốt thì* E_r *là* (*η, θ*)*-tốt.*

Chứng minh. Chúng tôi sử dụng các ký hiệu và định nghĩa của Bổ đề 3. Cho $\mathcal{C} \in \mathcal{S}_r$ và *r** là điểm tính toán lại cuối cùng của nó trong E_{r-1}. Cho *T* là mục tiêu sau *r** và T' là mục tiêu tại *r*. Chúng ta cần chứng minh rằng $f(T', n_r) \in [\eta f, \theta f]$. Lưu ý rằng nếu *r* là điểm tính toán lại thì điều này tuân theo Bổ đề 4. Mặt khác, $T' = T$ và $\eta\gamma \le f(T, n_{r^*}) \le \theta f/\gamma$. Sử dụng Bổ đề 3, $r - r^* \le \Delta \le \frac{\tau m}{f}$. Như vậy, $\frac{1}{\gamma} n_{r^*} \le n_r \le \gamma n_{r^*}$. Theo Sự thật 2 ta có $f(T, n_r) \le f(T, \gamma n_{r^*}) \le \gamma f(T, n_{r^*}) \le \theta f$ và $f(T, n_r) \ge f\left(T, \frac{1}{\gamma} n_{r^*}\right) \ge \frac{1}{\gamma} f(T, n_{r^*}) \ge \theta f$. □

Hệ quả 3. *Cho E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt. Khi đó mọi vòng đều là* (*η, θ*)*-tốt trong E.*

Chứng minh. Để mâu thuẫn, hãy gọi *r* là vòng nhỏ nhất của *E* không phải là (*η, θ*)-tốt. Điều này nghĩa là có một chuỗi $\mathcal{C}$ và một bên trung thực sở hữu chuỗi này ở vòng *r* và mục tiêu tương ứng *T* sao cho $f(T, n_r) \notin [\eta f, \theta f]$. E_{r-1} là (*η, θ*)-tốt nên theo Hệ quả 1, E_r là $\frac{m}{16\tau f}$-chính xác. Cho $r^* < r$ là điểm tính toán lại cuối cùng $(\eta\gamma, \theta/\gamma)$-tốt của $\mathcal{C}$ (cho *r** bằng 0 trong trường hợp không có điểm đó).

Đầu tiên giả sử có một điểm tính toán lại khác là $r' \in (r^*, r]$. Theo định nghĩa của *r**, r' không phải là $(\eta\gamma, \theta/\gamma)$-tốt. Tuy nhiên, giả định của Bổ đề 4 vẫn đúng, ngụ ý rằng $\mathcal{C}$ là $(\eta\gamma, \theta/\gamma)$-tốt. Chúng tôi đã đạt đến một mâu thuẫn.

Bây giờ chúng ta có thể giả định rằng không có điểm tính toán lại trong $(r^*, r]$ và do đó các điểm *r** và *r* tương ứng với cùng một mục tiêu *T* với $\eta\gamma \le f(T, n_{r^*}) \le \theta f/\gamma$. Lưu ý rằng vì *r** là một điểm tính toán lại $(\eta\gamma, \theta/\gamma)$-tốt và

E_{r-1} là (η, θ)-tốt, chúng ta có $r - r^* \leq \frac{\tau m}{f}$. Điều này suy ra từ Bổ đề 3, vì $\mathcal{C}$ thuộc về bên trung thực ở vòng r. Như vậy $\frac{1}{\gamma} n_{r^*} \leq n_r \leq \gamma n_{r^*}$ vì vậy (do Sự thật 2) $f(T, n_r) \leq f(T, \gamma n_{r^*}) \leq \gamma f(T, n_{r^*}) \leq \theta f$ và $f(T, n_r) \geq f\left(T, \frac{1}{\gamma} n_{r^*}\right) \geq \frac{1}{\gamma} f(T, n_{r^*}) \geq \eta f$. □

Định lý 1. *Một phép thực thi điển hình trong môi trường được vận hành tốt* (γ, s) *là* $\frac{m}{16\tau f}$*-chính xác và* (η, θ)*-tốt.*

Chứng minh. Điều này suy ra từ Hệ quả 3 và 1. □

6.5 Tiền tố chung và chất lượng chuỗi

Mệnh đề 6. *Cho E là một thực thi điển hình trong môi trường được* (γ, s) *vận hành tốt. Bất kỳ* $\frac{\theta\gamma m}{8\tau}$ *Block liên tiếp nào trong một Epoch của chuỗi* $\mathcal{C} \in \mathcal{S}_r$ *đều được tính toán trong ít nhất* $\frac{m}{16\tau f}$ *vòng.*

Chứng minh. Giả sử theo hướng mâu thuẫn rằng các Block của $\mathcal{C}$ được tính trong các vòng trong S^*, với một số S^* sao cho $|S^*| < \frac{m}{16\tau f}$. Xét một S sao cho $S^* \subseteq S$ và $|S| < \frac{m}{16\tau f}$, tính chất là một Block mục tiêu T trong $\mathcal{C}$ được tính toán bởi một bên trung thực trong một vòng $v \in S$. Một S như vậy tồn tại bởi Bổ đề 2 và 3. Theo Mệnh đề 3 và 5, số Block mục tiêu T được tính trong S nhiều nhất là

$$(1+\epsilon)(2-\delta)pT \sum_{u \in S} n_u \leq (1+\epsilon)(2-\delta)pT\gamma n_v |S| \leq \frac{(1+\epsilon)(2-\delta)\gamma|S|\theta f}{1-\theta f} \leq \frac{\theta\gamma m}{8\tau}.$$

Với bất đẳng thức đầu tiên, chúng tôi đã sử dụng Sự thật 1, đối với phần thứ 2 của Sự thật 1 và vòng v là (η, θ)-tốt và đối với phần cuối cùng Yêu cầu (R2). □

Định lý 2 (Tiền tố chung)**.** *Trong bất kỳ phép thực thi điển hình nào của môi trường* (γ, s) *được vận hành tốt, thuộc tính tiền tố chung giữ nguyên cho* $k \geq \frac{\theta\gamma m}{4\tau}$.

Chứng minh. Hãy xem xét theo hướng mâu thuẫn, các bên trung thực P_1 và P_2 với các chuỗi $\mathcal{C}_1$ và $\mathcal{C}_2$ được thông qua lần lượt ở các vòng $r_1 \leq r_2$, sao cho $\mathcal{C}_1^{\lceil k} \not\preceq \mathcal{C}_2$. Xác định $r \geq r_1$ là vòng nhỏ nhất trong đó bên trung thực P chấp nhận chuỗi $\mathcal{C}$ sao cho $\mathcal{C}_1^{\lceil k} \not\preceq \mathcal{C}$. Xem xét Block cuối cùng trên tiền tố chung của $\mathcal{C}_1$ và $\mathcal{C}$ được tính toán bởi một bên trung thực và cho r^* là vòng mà nó được tính toán ($r^* = 0$ nếu không tồn tại Block như vậy). Ký hiệu là $\mathcal{C}^*$ là phần chung của $\mathcal{C}_1$ và $\mathcal{C}$ cho đến (và bao gồm) Block này. Cho $S = \{u : r^* < u < r\}$ và J là tập hợp các truy vấn tấn công trong S. Theo Mệnh đề 6, $|S| \geq \frac{m}{16\tau f}$; không khó để xác minh các giới hạn trên $|J|$ và $|S|$ của Định nghĩa 8 và Mệnh đề 3 giữ nguyên. Chúng ta sẽ tranh luận rằng

$$(1+\epsilon)p \sum_{u \in S} t_u \geq \sum_{j \in J} A_j \geq \sum_{u \in S} Q_u \geq (1-\epsilon)(1-\theta f)p \sum_{u \in S} n_u. \tag{2}$$

Lưu ý rằng điều này gây ra sự mâu thuẫn, vì Yêu cầu (R0) và (R2) cho thấy $t_u < (1-\delta)n_u \leq (1-4\epsilon-2\theta f)n_u < (1-2\epsilon)(1-\theta f)n_u \leq \frac{1-\epsilon}{1+\epsilon} \cdot (1-\theta f)n_u$. Bất đẳng thức thứ nhất và thứ ba đúng với mọi phép thực thi điển hình; chúng tôi tiến hành chứng minh bất đẳng thức ở giữa.

Đầu tiên chúng tôi đưa ra một nhận xét đơn giản nhưng quan trọng. Hãy xem xét bất kỳ Block B nào mở rộng chuỗi $\mathcal{C}'$ được tính toán bởi một bên trung thực trong vòng thành công duy nhất $u \in S$. Với mọi $d \in \mathbb{R}$ sao cho **diff**($\mathcal{C}'$) $\leq d <$ **diff**($\mathcal{C}'B$), chúng tôi viết $d \in B$. Nếu tồn tại ít nhất một chuỗi có độ khó d, thì Block "chứa" điểm độ khó d đã được kẻ tấn công tính toán. Chính thức hơn, giả sử tồn tại một chuỗi $\mathcal{C}''B'$ sao cho $B' \neq B$ và $d \in B'$. Chúng ta thấy B' đã được kẻ tấn công tính toán. Điều này là do không có bên trung thực nào mở rộng $\mathcal{C}''$ ở vòng muộn hơn u vì **diff**($\mathcal{C}''$) $\leq d <$ **diff**($\mathcal{C}'B$); mặt khác, nếu một bên trung thực tính toán B' ở vòng $u' < u$, thì không bên trung thực nào mở rộng $\mathcal{C}'$ ở vòng u vì **diff**($\mathcal{C}'$) $\leq d <$ **diff**($\mathcal{C}''B''$); cuối cùng, lưu ý rằng u cũng bị loại trừ vì là thành công duy nhất.

Bây giờ chúng tôi quay lại chứng minh (2). Theo Bổ đề Tăng trưởng Chuỗi (Bổ đề 1), **diff**($\mathcal{C}$) ≥ **diff**($\mathcal{C}^*$) + $\sum_{u \in S} Q_u$. Quan sát ở đoạn trước cho thấy rằng có sự thêm vào từ tập hợp

$$U = \{d \in B : B \text{ được tính trong vòng thành công duy nhất ở } S\}$$

vào tập hợp

$$W = \{d \in B' : B' \text{ được kẻ tấn công tính toán trong một vòng ở } S\}.$$

Để thấy điều này, trước tiên hãy lưu ý rằng cả hai tập hợp đều là tập con của (**diff**($\mathcal{C}^*$), **diff**($\mathcal{C}$)). Chúng tôi khẳng định rằng với mọi $d \in U$ và B sao cho $d \in B$, luôn tồn tại $B' \neq B$ nằm trên $\mathcal{C}_1$ hoặc $\mathcal{C}$, hoặc trên tiền tố chung của chúng. Thật vậy, B không thể vừa thuộc $\mathcal{C}$ vừa thuộc $\mathcal{C}'$; theo cách xác định của r^*, B không thể nằm trên tiền tố chung của chúng; theo cách xác định của r, B không thể thuộc phần của $\mathcal{C}$ có độ khó lớn hơn **diff**($\mathcal{C}_1$).

Vì kích thước của tập thứ nhất là $\sum_{u \in S} Q_u$ và kích thước của tập thứ hai theo Mệnh đề 5 nhiều nhất là $\sum_{j \in J} A_j$, nên chúng ta suy ra bất đẳng thức trong (2). □

Định lý 3 (Chất lượng chuỗi)**.** *Trong bất kỳ phép thực thi điển hình nào của môi trường* (γ, s) *được vận hành tốt, thuộc tính chất lượng chuỗi giữ nguyên cho* $\ell \geq \frac{\theta \gamma m}{8\tau}$ *và* $\mu = \delta - 2\epsilon - \theta f \geq \delta/2$.

Chứng minh. Cho B_i là Block thứ i của $\mathcal{C}$ sao cho $\mathcal{C} = B_1 \ldots B_{\text{len}(\mathcal{C})}$ và xét L Block liên tiếp $B_u, \ldots, B_v$. Xác định L' là số Block liên tiếp nhỏ nhất $B_{u'}, \ldots, B_{v'}$ bao gồm các Block L đã cho (tức là $u' \leq u$ và $v \leq v'$) và có các thuộc tính là (1) Block $B_{u'}$ được tính toán bởi một bên trung thực hoặc là B_1 trong trường hợp Block đó không tồn tại và (2) tồn tại một vòng mà tại đó một bên trung thực đang cố gắng kéo dài chuỗi kết thúc tại Block $B_{v'}$. Chúng ta thấy số lượng L' được xác định rõ ràng vì $B_{\text{len}(\mathcal{C})}$ đứng đầu chuỗi mà một bên trung thực đang cố gắng mở rộng. Biểu thị d' là tổng độ khó của các Block L' này. Cũng xác định r_1 là vòng mà $B_{u'}$ được tạo ($r_1 = 0$ nếu $B_{u'}$ là Block khởi tạo), r_2 là vòng đầu tiên mà một bên trung thực cố gắng mở rộng $B_{v'}$ và $S = \{r : r_1 \leq r \leq r_2\}$. Lưu ý rằng, bởi Mệnh đề 6, $|S| \geq \frac{m}{16\tau f}$.

Bây giờ gọi x biểu thị tổng độ khó của tất cả các Block từ các bên trung thực được bao gồm trong các Block L và hướng tới sự mâu thuẫn, giả sử

$$x < \mu d \leq \mu d' . \tag{3}$$

Đầu tiên giả sử tất cả các Block L' $\{B_j : u' \leq j \leq v'\}$ đã được tính toán trong các vòng trong tập hợp S. Bây giờ chúng tôi thảo luận về chuỗi bất đẳng thức sau đây.

$$(1-\epsilon)(1-\delta)p \sum_{u \in S} n_u \geq (1+\epsilon)p \sum_{u \in S} t_u > \sum_{j \in J} A_j \geq d' - x \geq (1-\mu)d' \geq (1-\mu) \sum_{u \in S} Q_u \tag{4}$$

Bất đẳng thức thứ nhất là do yêu cầu của chúng tôi là $t_u \leq (1-\delta) n_u$ đúng với bất kỳ vòng u nào. Bất đẳng thức thứ hai đúng trong bất kỳ phép thực thi điển hình và Nhận xét **??**. Bất đẳng thức thứ ba suy ra từ cách xác định của x, d' và Mệnh đề 5. Bất đẳng thức thứ tư xuất phát từ mối quan hệ giữa x và d' được nêu trong (3). Hãy xem bất đẳng thức cuối cùng, giả sử $\sum_{u \in S} Q_u$. Nhưng sau đó, theo Bổ đề 1 về Tăng trưởng Chuỗi, giả định một bên trung thực trong $B_{v'}$ ở vòng r_2 bị mâu thuẫn vì tất cả các bên trung thực phải ở trong chuỗi có độ dài lớn hơn. Bây giờ chúng ta thấy (4) mâu thuẫn với Mệnh đề 3, vì

$$(1-\mu) \sum_{u \in S} Q_u > (1-\mu)(1-\epsilon)(1-\theta f)p \sum_{u \in S} n_u \geq (1+\epsilon)(1-\delta)p \sum_{u \in S} n_u ,$$

trong đó bất đẳng thức cuối cùng suy ra bởi Yêu cầu (R2).

Để hoàn tất chứng minh, chúng ta cần xem xét trường hợp các Block L' này chứa các Block mà kẻ tấn công tính toán theo các vòng bên ngoài S. Không khó để thấy rằng trường hợp này thể hiện một sự dự đoán hoặc một sự chèn thêm và không thể xảy ra trong một phép thực thi thông thường. □

6.6 Tính bền vững và tính sống động

Định lý 4. *Cho E là một phép thực thi điển hình trong môi trường được* (γ, s) *vận hành tốt. Tính bền vững được thỏa mãn với độ sâu* $k \geq \frac{\theta \gamma m}{4\tau}$.

Chứng minh. Nếu chuỗi $\mathcal{C}$ của một bên trung thực ở vòng *r* chứa một giao dịch trong $\mathcal{C}^{\lceil k}$ thì nó sẽ có thuộc tính tiền tố chung đối với chuỗi $\mathcal{C}'$ của một bên trung thực khác ở bất kỳ vòng $r' \geq r$, suy ra $\mathcal{C}^{\lceil k} \preceq \mathcal{C}'$. □

Mệnh đề 7. *Cho E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt. Giả sử một bên trung thực có tiền tố của* $\mathcal{C}$ *tại vòng u và một bên trung thực chấp nhận tiền tố của* $\mathcal{C}$ *tại vòng* $v \geq u + \frac{m}{16\tau f}$. *Trong tập hợp các vòng liên tiếp S =* $\{r : u \leq r < v\}$, $\mathcal{C}$ *thu được ít nhất* $(1-\epsilon)(1-\theta f)nf|S|/\gamma$ *Block.*

Chứng minh. Giả sử $|S| \leq s - \frac{m}{8\tau f}$. Giả sử $\mathcal{C}$ ở vòng *u* có độ khó *d*. Theo Bổ đề Tăng trưởng Chuỗi 1, $\mathcal{C}$ có độ khó ít nhất là $d' = d + \sum_{r \in S} D_r$ tại vòng *v*. Cho *T* là mục tiêu ít nhất trong số các mục tiêu của các Block có được trong các vòng ở *S*. Theo Bổ đề 2 và 3, có một Block trung thực trên $\mathcal{C}$ được tính toán trong $S' = \left\{r: u - \frac{m}{16\tau f} \leq r < v + \frac{m}{16\tau f}\right\}$ nằm trong cùng một Epoch và do đó có mục tiêu *T* . Vì phép thức thực thi là tốt (Định lý 1), $S \subseteq S'$ và $|S'| \leq s$, chúng ta thu được (Sự thật 1) $pTn_r \geq \eta f/\gamma$, với mọi $r \in S$. Theo đó $\mathcal{C}$ đã đạt được ít nhất

$$T(d' - d) = T\sum_{r \in S} D_r \geq T\sum_{r \in S} Q_r > (1-\epsilon)(1-\theta f)\sum_{r \in S} pTn_r \geq (1-\epsilon)(1-\theta f)\eta f\,|S|/\gamma$$

Block trong các vòng ở *S* .

Để tránh giới hạn trên của *S* và đạt được mệnh đề, hãy lưu ý rằng theo Bổ đề 2, chúng ta có thể phân chia *S* sao cho mỗi phần S_i thỏa mãn $\frac{m}{16\tau f} \leq |S| \leq s - \frac{m}{8\tau f}$, một bên trung thực có (tiền tố của) $\mathcal{C}$ ở đầu của S_i, và một bên trung thực đã chấp nhận (tiền tố của) $\mathcal{C}$ ở cuối của S_i. Tổng hợp từng phần chúng ta có được nhận định chung. □

Định lý 5. *Cho E là một phép thực thi điển hình trong môi trường* (*γ, s*) *được vận hành tốt. Tính sống động thoả mãn về độ sâu* $k \geq \frac{\theta\gamma m}{8\tau}$ *và thời gian chờ* $\frac{m}{16\tau f} + \frac{\gamma\kappa}{\eta f(1-\epsilon)(1-\theta f)}$.

Chứng minh. Giả sử một giao dịch tx nằm trong bất kỳ Block nào được tính toán bởi một bên trung thực trong $\frac{m}{16\tau f}$ vòng liên tiếp và xem xét chuỗi $\mathcal{C}$ của một bên trung thực tùy ý sau các vòng này. Theo Bổ đề 2, $\mathcal{C}$ chứa một Block trung thực được tính toán trong $\frac{m}{16\tau f}$ vòng và Block này chứa tx. Ký hiệu *S* là tập hợp ít nhất $\gamma\kappa/[\eta f(1-\epsilon)(1-\theta f)]$ vòng tiếp theo vòng mà Block trung thực này đã được tính toán. Theo Mệnh đề **??**, trên Block này đã được tích lũy ít nhất *k* Block. □

Tài liệu tham khảo

[1] Back, A.: Hashcash. http://www.cypherspace.org/hashcash (1997)

[2] Bahack, L.: Các cuộc tấn công Bitcoin trên lý thuyết với sức mạnh tính toán chưa đến một nửa (bản nháp). IACR Cryptology ePrint Archive 2013, 868 (2013), http://eprint.iacr.org/2013/868

[3] Bellare, M., Rogaway, P.: Những Oracle ngẫu nhiên rất thiết thực: Một mô hình để thiết kế các giao thức hiệu quả. Trong: Denning, DE, Pyle, R., Ganesan, R., Sandhu, RS, Ashby, V. (eds.) CCS '93, Kỷ yếu của Hội nghị ACM lần thứ 1 về An ninh Máy tính và Truyền thông, Fairfax, Virginia, Hoa Kỳ, Ngày 3-5 tháng 11 năm 1993. trang 62–73. ACM (1993), http://doi.acm.org/10.1145/168588.168596

[4] Canetti, R.: Bảo mật và thành phần của các giao thức mật mã đa bên. J. Mật mã học 13(1), 143–202 (2000)

[5] Canetti, R.: Bảo mật tổng hợp toàn cầu: Một mô hình mới cho các giao thức mật mã. Lưu trữ ePrint mật mã, Báo cáo 2000/067 (2000), http://eprint.iacr.org/2000/067

[6] Canetti, R.: Bảo mật tổng hợp toàn cầu: Một mô hình mới cho các giao thức mật mã. Trong: Hội nghị chuyên đề thường niên lần thứ 42 về Cơ sở Khoa học Máy tính, FOCS 2001, 14-17 tháng 10 năm 2001, Las Vegas, Nevada, Hoa Kỳ. trang 136–145. Hiệp hội máy tính IEEE (2001), http://dx.doi.org/10.1109/SFCS.2001.959888

[7] Dwork, C., Lynch, NA, Stockmeyer, LJ: Sự đồng thuận khi có sự đồng bộ một phần. J. ACM 35(2), 288–323 (1988), http://doi.acm.org/10.1145/42282.42283

[8] Dwork, C., Naor, M.: Định giá thông qua việc xử lý hoặc chống lại thư rác. Trong: Brickell, EF (ed.) CRYPTO. Ghi chú các chương trong ngành khoa học máy tính, xuất bản lần 740, trang 139–147. Springer (1992)

[9] Eyal, I., Sirer, EG: Đa số thôi là chưa đủ: Việc khai thác Bitcoin rất dễ bị tổn thương. Trong: Keromytis, AD (ed.) Mật mã tài chính. Ghi chú các chương trong ngành khoa học máy tính, xuất bản lần. 7397. Springer (2014), http://dx.doi.org/10.1007/978-3-642-32946-3

[10] Garay, JA, Kiayias, A., Leonardos, N.: Giao thức hệ thống nền tảng Bitcoin: Phân tích và ứng dụng. Lưu trữ ePrint mật mã IACR 2014, 765 (2014), http://eprint.iacr.org/2014/765

[11] Garay, JA, Kiayias, A., Leonardos, N.: Giao thức hệ thống nền tảng Bitcoin: Phân tích và ứng dụng. Trong: Oswald, E., Fischlin, M. (eds.) Những tiến bộ trong mật mã học - EUROCRYPT 2015 - Hội nghị quốc tế thường niên lần thứ 34 về lý thuyết và ứng dụng kỹ thuật mật mã, Sofia, Bulgaria, ngày 26-30 tháng 4 năm 2015, Kỷ yếu, Phần II . Ghi chú các chương trong ngành khoa học máy tính, xuất bản lần. 9057, trang 281–310. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46803-6_10

[12] Hadzilacos, V., Toueg, S.: Một cách tiếp cận mô-đun đối với các chương trình phát tán có khả năng chịu lỗi và các vấn đề liên quan. Tech. rep. (1994)

[13] Juels, A., Brainard, JG: Câu đố của khách hàng: Một biện pháp đối phó bằng mật mã chống lại các cuộc tấn công làm suy giảm kết nối. Trong: NDSS. Hiệp hội Internet (1999)

[14] Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Trò chơi khai thác Blockchain. Trong: Conitzer, V., Bergemann, D., Chen, Y. (eds.) Kỷ yếu của Hội nghị ACM 2016 về Kinh tế và Tính toán, EC '16, Maastricht, Hà Lan, ngày 24-28 tháng 7 năm 2016. trang 365–382. ACM (2016), http://doi.acm.org/10.1145/2940716.2940773

[15] Kiayias, A., Panagiotakos, G.: Sự cân bằng giữa tốc độ và bảo mật trong các giao thức Blockchain. Lưu trữ ePrint mật mã IACR 2015, 1019 (2015), http://eprint.iacr.org/2015/1019

[16] Lamport, L., Shostak, RE, Pease, MC: Vấn đề về các vị tướng Byzantine. ACM Trans. Program. Lang. Syst. 4(3), 382–401 (1982)

[17] McDiarmid, C.: Phương pháp xác suất cho toán học rời rạc thuật toán, chương. Nồng độ, trang 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg (1998), http://dx.doi.org/10.1007/ 978-3-662-12788-9_6

[18] Mitzenmacher, M., Upfal, E.: Xác suất và tính toán - thuật toán ngẫu nhiên và phân tích xác suất. Nhà xuất bản Đại học Cambridge (2005)

[19] Nakamoto, S.: Triển khai mã nguồn mở Bitcoin của tiền tệ p2p. http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source (tháng 2 năm 2009)

[20] Pass, R., Seeman, L., Shelat, A.: Phân tích giao thức Blockchain trong các mạng không đồng bộ. Trong: Coron, J., Nielsen, JB (eds.) Những tiến bộ trong mật mã học - EUROCRYPT 2017 - Hội nghị quốc tế thường niên lần thứ 36 về lý thuyết và ứng dụng kỹ thuật mật mã, Paris, Pháp, ngày 30 tháng 4, ngày 4 tháng 5 năm 2017, Kỷ yếu, Phần II. Ghi chú các chương trong ngành khoa học máy tính, xuất bản lần. 10211, trang 643–673 (2017), https://doi.org/10.1007/978-3-319-56614-6_22

[21] Pease, MC, Shostak, RE, Lamport, L.: Đạt được thỏa thuận khi có lỗi. J. ACM 27(2), 228–234 (1980)

[22] Rivest, RL, Shamir, A., Wagner, DA: Câu đố khóa thời gian và Crypto giải phóng theo thời gian. Công nghệ. đại diện, Cambridge, MA, Hoa Kỳ (1996)

[23] Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Chiến lược khai thác ích kỷ tối ưu bằng Nitcoin. CoRR abs/1507.06183 (2015), http://arxiv.org/abs/1507.0618 3

A Giao thức hệ thống nền tảng Bitcoin với độ khó thay đổi (tiếp theo)

Trong phần này, chúng tôi cung cấp mô tả chi tiết hơn về giao thức hệ thống nền tảng Bitcoin với các chuỗi có độ khó khác nhau. Việc trình bày dựa trên mô tả trong [11].

A.1 Giao thức

Như trong [11] trong phần mô tả về giao thức hệ thống nền tảng, chúng tôi cố tình tránh chỉ định loại giá trị/nội dung mà các bên cố gắng chèn vào chuỗi, loại xác thực chuỗi mà họ thực hiện (ngoài việc kiểm tra các thuộc tính cấu trúc của nó đối với hàm Hash $G(\cdot)$, $H(\cdot)$) và cách chúng diễn giải chuỗi. Các hoạt động kiểm tra và thao tác này được xử lý bởi các hàm bên ngoài $V(\cdot)$, $I(\cdot)$ và $R(\cdot)$ (tương ứng là *hàm xác thực nội dung*, *hàm đóng góp đầu vào* và *hàm đọc chuỗi*) được chỉ định bởi ứng dụng chạy "trên cùng" của giao thức hệ thống nền tảng.

Giao thức hệ thống nền tảng Bitcoin trong cài đặt động được chỉ định là Thuật toán 4 và phụ thuộc vào ba quy trình phụ.

Xác thực chuỗi. Thuật toán **validate** thực hiện xác thực các thuộc tính cấu trúc của chuỗi $\mathcal{C}$. Nó được đưa ra làm đầu vào giá trị q, cũng như các hàm Hash $H(\cdot)$, $G(\cdot)$. Nó được tham số hóa bởi vị từ xác thực nội dung $V(\cdot)$ và *hàm tính toán mục tiêu* $D(\cdot)$ (xem Phần 3). Đối với mỗi Block của chuỗi, thuật toán sẽ kiểm tra xem bằng chứng công việc đã được giải quyết đúng cách chưa (với mục tiêu phù hợp được xác định bởi hàm tính toán mục tiêu) và bộ đếm *ctr* không vượt quá q. Hơn nữa, nó thu thập đầu vào từ tất cả các Block $\mathbf{x}_{\mathcal{C}}$ và kiểm tra chúng thông qua vị từ $V(\mathbf{x}_{\mathcal{C}})$; lưu ý rằng $V(\varepsilon)$ = **true**. Chuỗi không thực hiện được quy trình xác thực này sẽ bị từ chối. (Thuật toán 1.)

Thuật toán 1. *Vị từ xác thực chuỗi* được tham số hóa bởi q, D, các hàm Hash $G(\cdot)$, $H(\cdot)$, và *vị từ xác thực đầu vào* $V(\cdot)$. Đầu vào là chuỗi $\mathcal{C}$.

```
1:  function validate(r_now, C)
2:      valid ← V(x_C)
3:      if valid = Đúng ∧ (C ≠ ε) then                ▷ C là không trống và có nghĩa wrt V(·)
4:          r' ← r_now
5:          ⟨r, st, x, ctr⟩ ← head(C)
6:          st' ← H(ctr, G(r, st, x))
7:          repeat
8:              ⟨r, st, x, ctr⟩ ← head(C)
9:              T ← D(r_{C⌈1})                     ▷ Tính toán mục tiêu dựa trên C⌈1
10:             if validblock_q^T(⟨st, x, ctr⟩)∧(H(ctr, G(r, st, x)) = st')∧(r < r') then
11:                 r' ← r                         ▷ Giữ lại dấu thời gian vòng
12:                 st' ← st                       ▷. Giữ lại giá trị Hash
13:                 C ← C⌈1                        ▷ Bỏ head khỏi C
14:             else
15:                 hợp lệ ← Sai
16:             end if
17:         until (C = ε) ∨ (hợp lệ = Sai)
18:     end if
19:     return valid
20: end function
```

So sánh chuỗi. Mục tiêu của thuật toán thứ hai, **maxvalid**, là tìm ra chuỗi "tốt nhất có thể" khi cho một tập hợp các chuỗi. Thuật toán rất đơn giản và được tham số hóa bằng hàm **max**(·) áp dụng một số thứ tự trong không gian của chuỗi. Khía cạnh quan trọng nhất là độ khó của chuỗi, trong đó trường hợp **max**($\mathcal{C}_1$, $\mathcal{C}_2$) sẽ trả về *độ khó* cao nhất trong hai trường hợp. Trong trường hợp **diff**($\mathcal{C}_1$) = **diff**($\mathcal{C}_2$), một số đặc tính khác có thể được sử dụng để phá vỡ ràng buộc. Trong trường hợp của chúng tôi, **max**(·,·) sẽ luôn trả về toán hạng đầu tiên để phản ánh thực tế là các bên áp dụng chuỗi đầu tiên mà họ nhận được từ mạng. (Thuật toán 2.)

Thuật toán 2. Hàm tìm chuỗi "tốt nhất", được tham số hóa bởi hàm **max**(·). Đầu vào là{$\mathcal{C}_1$,..., $\mathcal{C}_k$}.

```
function maxvalid(r, C_1,..., C_k)
    temp ← ε
        for i = 1 to k do
        if validate(r, C_i) then
          temp ← max(C, temp)
        end if
    end for
    return temp
end function
```

Bằng chứng công việc. Thuật toán thứ ba, **pow**, là quy trình tìm bằng chứng công việc. Nó lấy đầu vào là một chuỗi và cố gắng mở rộng nó thông qua việc giải quyết bằng chứng công việc. Thuật toán này được tham số hóa bởi hai hàm Hash $H(\cdot)$, $G(\cdot)$ và tham số q. Thuật toán gọi hàm tính toán mục tiêu $D(\cdot)$ để xác định giá trị T sẽ được sử dụng cho bằng chứng công việc. Quy trình, cho chuỗi $\mathcal{C}$ và giá trị x được chèn vào chuỗi, Hash các giá trị này để thu được h và khởi tạo bộ đếm ctr. Sau đó, nó tăng ctr và kiểm tra xem liệu $H(ctr, h) < T$; trong trường hợp tìm thấy một ctr phù hợp thì thuật toán sẽ thành công trong việc giải POW và mở rộng chuỗi $\mathcal{C}$ thêm một Block. (Thuật toán 3.)

Thuật toán 3. Hàm *chứng minh công việc* được tham số hóa bởi q và hàm Hash $H(\cdot)$, $G(\cdot)$. Đầu vào là $(x, \mathcal{C})$.

```
function pow(r, x, C)
    if C = ε then                          ▷ Xác định tình huống bằng chứng công việc.
        st ← 0
    else
        ⟨r′, st′, x′, ctr′⟩ ← head(C)
        st ← H(ctr′, G(r′, st′, x′))
    end if
    ctr ← 1
    B ← ε
    T ← D(r_C)                             ▷ Tính toán mục tiêu cho Block tiếp theo dựa trên dấu thời gian.
    h ← G(r, st, x)
    while (ctr ≤ q) do
        if (H(ctr, h) <) then              ▷ Lệnh gọi H(·) này tuân theo giới hạn q.
            B ← ⟨r, st, c, ctr⟩
            break
        end if
        ctr ← ctr + 1
    end while
    C ← CB                                 ▷ Chuỗi được mở rộng.
    return C
end function
```

Giao thức hệ thống nền tảng. Cốt lõi của giao thức tương tự như [11], với một số điểm khác biệt quan trọng. Đầu tiên là quy trình phải tuân theo khi chúng hoạt động. Các bên kiểm tra cờ **sẵn sàng (ready)** mà họ sở hữu là sai khi và chỉ khi họ không hoạt động ở vòng trước. Trong trường hợp cờ **sẵn sàng** là sai, họ sẽ phát một thông điệp đặc biệt là '**Join**' để yêu cầu phiên bản mới nhất của (các) Blockchain. Tương tự, các bên nhận được thông điệp yêu cầu đặc biệt trong RECIEVE() của họ sẽ phát tán chuỗi của họ. Như trước đây, các bên chạy "không xác định" (phân tích bảo mật của chúng tôi sẽ áp dụng khi tổng thời gian chạy là đa thức tính bằng κ). Hàm đóng góp đầu vào $I(\cdot)$ và hàm đọc chuỗi $R(\cdot)$ được áp dụng cho các giá trị được lưu trữ trong chuỗi. Các bên kiểm tra băng RECIEVE() để xem liệu có bất kỳ cập nhật cần thiết nào đối với chuỗi cục bộ của họ hay không; sau đó họ cố gắng mở rộng nó thông qua thuật toán bằng chứng công việc **pow**. Hàm $I(\cdot)$ xác định đầu vào sẽ được thêm vào chuỗi dựa trên trạng thái *st* của bên tham gia, chuỗi $\mathcal{C}$ hiện tại, nội dung băng đầu vào INPUT() của các bên và băng liên lạc RECIEVE(). Băng đầu vào chứa hai loại ký hiệu, READ và (INSERT, *value*); đầu vào khác được bỏ qua. Trong trường hợp chuỗi cục bộ $\mathcal{C}$ được mở rộng, chuỗi mới sẽ được phát tán cho các bên khác. Cuối cùng, trong trường hợp ký hiệu READ xuất hiện trong băng liên lạc, giao thức sẽ áp dụng hàm $R(\cdot)$ cho chuỗi hiện tại của nó và ghi kết quả vào băng đầu ra OUTPUT(). Mã giả định của giao thức hệ thống nền tảng được trình bày trong Thuật toán 4.

Thuật toán 4. Giao thức hệ thống nền tảng Bitcoin trong cài đặt động ở vòng "**round**" ở trạng thái cục bộ (st, $\mathcal{C}$) được tham số hóa bởi *hàm đóng góp đầu vào* $I(\cdot)$ và *hàm đọc chuỗi* $R(\cdot)$. Cờ "**sẵn sàng**" là Sai khi và chỉ khi bên đó không hoạt động ở vòng trước.

if ready = True **then**
  DIFFUSE(**Ready**)
  $\tilde{\mathcal{C}} \leftarrow$ **maxvalid**($\mathcal{C}$, tất cả các chuỗi $\mathcal{C}'$ được tìm thấy trong RECEIVE())
  if INPUT() chứa READ **then**
    write $R(\boldsymbol{x}_{\mathcal{C}})$ to OUTPUT()
  end if
  $\langle st, x\rangle \leftarrow I\big(st, \tilde{\mathcal{C}}, \mathbf{round}, \text{INPUT}(), \text{RECIEVE}()\big)$
  $\mathcal{C}_{\text{new}} \leftarrow \mathbf{pow}\big(\mathbf{round}, x, \tilde{\mathcal{C}}\big)$
  if $(\mathcal{C} \neq \mathcal{C}_{\text{new}}) \vee$ (**Join** $\in$ RECEIVE()) **then**
    $\mathcal{C} \leftarrow \mathcal{C}_{\text{new}}$
    DIFFUSE($\mathcal{C}$) ▷chuỗi được khuếch tán khi cập nhật hoặc khi ai đó tham gia.
  end if
  DIFFUSE(**RoundComplete**)
else
  ready ← True
  DIFFUSE(**Join**, **RoundComplete**)
end if

B Sổ cái giao dịch công khai mạnh mẽ

Trong phần này chúng tôi tái hiện cách trình bày sổ cái giao dịch công khai được đưa ra trong [10, 11]. Một *sổ cái giao dịch công khai* được xác định đối với một tập hợp các sổ cái hợp lệ $\mathcal{L}$ và một tập hợp các giao dịch hợp lệ $\mathcal{T}$, mỗi giao dịch có một bài kiểm tra tư cách thành viên hiệu quả. Sổ cái $\mathbf{x} \in \mathcal{L}$ là một Vectơ gồm các chuỗi giao dịch $\text{tx} \in \mathcal{T}$. Mỗi giao dịch tx có thể được liên kết với một hoặc nhiều *tài khoản*, ký hiệu là a_1 , a_2 ,... v.v.

Các bên tham gia giao thức hệ thống nền tảng, được gọi là *thợ đào* trong ngữ cảnh của phần này, xử lý chuỗi giao dịch có dạng $x = \text{tx}_1 \ldots \text{tx}_e$ được cho là sẽ được tích hợp vào chuỗi cục bộ $\mathcal{C}$ của họ. Đầu vào được chèn vào mỗi Block của chuỗi $\mathcal{C}$ là x giao dịch liên tiếp. Như vậy, sổ cái là một Vectơ của các giao dịch liên tiếp $\langle x_1, \ldots, x_m\rangle$ và chuỗi $\mathcal{C}$ có độ dài m chứa sổ cái $\mathbf{x}_{\mathcal{C}} = \langle x_1, \ldots, x_m\rangle$ nếu đầu vào của Block thứ j trong $\mathcal{C}$ là x_j.

Mô tả và các thuộc tính của giao thức sổ cái sẽ được thể hiện tương ứng với một Oracle **Txgen** sẽ kiểm soát một tập hợp các tài khoản bằng cách tạo chúng và phát hành các giao dịch thay mặt chúng. Trong một phép thực thi của

giao thức hệ thống nền tảng, môi trường $\mathcal{Z}$ cũng như các thợ đào sẽ có quyền truy cập vào **Txgen**. Cụ thể, **Txgen** là một Oracle có trạng thái đáp ứng hai loại truy vấn (mà chúng tôi cố tình chỉ mô tả ở mức cao):

− **GenAccount** (1^κ): Nó tạo ra một tài khoản a.

− **IssueTrans** ($1^\kappa, \tilde{\text{tx}}$): Nó trả về một giao dịch tx với điều kiện $\tilde{\text{tx}}$ là một chuỗi được định dạng phù hợp hoặc $\perp$.

Chúng tôi cũng xem xét một mối quan hệ đối xứng trên $\mathcal{T}$, ký hiệu là $C(\cdot,\cdot)$, biểu thị khi hai giao dịch tx_1, tx_2 xung đột nhau. Sổ cái hợp lệ $\mathbf{x} \in \mathcal{L}$ không thể chứa hai giao dịch xung đột. Chúng tôi gọi Oracle **Txgen** là *rõ ràng* nếu nó cho rằng đối với tất cả PPT $\mathcal{A}$, xác suất $\mathcal{A}^{\mathbf{Txgen}}$ tạo ra giao dịch tx' sao cho $C(\text{tx}', \text{tx}) = 1$ với tx do **Txgen** phát hành là không đáng kể trong κ.

Cuối cùng, một giao dịch tx được gọi là *trung lập* nếu $C(\text{tx}, \text{tx}') = 0$ đối với bất kỳ giao dịch tx' nào. Sự hiện diện của các giao dịch trung lập trong sổ cái có thể hữu ích cho nhiều mục đích khác nhau tiếp theo và trong giao thức BA mà chúng tôi xây dựng trên sổ cái. Để thuận tiện, chúng tôi sẽ giả sử một Nonce ngẫu nhiên duy nhất $\rho \in \{0, 1\}^\kappa$ cũng là một giao dịch hợp lệ. Nonce sẽ là các giao dịch trung lập và có thể được đưa vào sổ cái với mục đích duy nhất là đảm bảo tính độc lập giữa các trường hợp POW được giải quyết bởi các bên trung thực.

Tiếp theo, chúng tôi xác định ba hàm $V(\cdot)$, $I(\cdot)$, $R(\cdot)$ sẽ biến giao thức hệ thống nền tảng thành Π_{PL}, một giao thức thực hiện sổ cái giao dịch công khai. Xem Hình 1.

Vị từ xác thực nội dung $V(\cdot)$	$V(\langle x_1, \ldots, x_m \rangle)$ là đúng khi và chỉ khi Vecto $\langle x_1, \ldots, x_m \rangle$ là một sổ cái hợp lệ, tức là $\langle x_1, \ldots, x_m \rangle \in \mathcal{L}$.
Hàm đọc chuỗi $R(\cdot)$	Nếu $V(\langle x_1, \ldots, x_m \rangle)$ = True thì giá trị $R(\mathcal{C})$ bằng $\langle x_1, \ldots, x_m \rangle$; nếu không thì không được xác định.
Hàm đóng góp đầu vào $I(\cdot)$	$I(st, \mathcal{C}, round,$ INPUT()) hoạt động như sau: nếu băng đầu vào chứa (INSERT, v), nó phân tích v dưới dạng một chuỗi các giao dịch và giữ lại chuỗi con lớn nhất $x' \preceq v$ hợp lệ đối với $\mathbf{x}_{\mathcal{C}}$ (và có giao dịch chưa được bao gồm trong $\mathbf{x}_{\mathcal{C}}$). Cuối cùng, $x = \text{tx}_0 x'$ trong đó tx_0 là giao dịch ngẫu nhiên trung tính.

Hình 1: *Giao thức sổ cái giao dịch công khai* Π_{PL} *được xây dựng trên hệ thống nền tảng Bitcoin.*

Trong Phần 4.3, chúng tôi đã giới thiệu hai thuộc tính thiết yếu cho một giao thức duy trì sổ cái giao dịch công khai: (i) *Tính bền vững* và (ii) *Tính sống động*. Tóm lại, tính bền vững tuyên bố rằng khi một người chơi trung thực báo cáo một giao dịch "đủ sâu" trong sổ cái thì tất cả những người chơi trung thực khác sẽ báo cáo giao dịch đó vô thời hạn bất cứ khi nào họ được yêu cầu và ở cùng một vị trí trong sổ cái (về cơ bản, điều này có nghĩa là tất cả người chơi trung thực đều đồng ý về tất cả các giao dịch đã diễn ra và theo một thứ tự). Trong bối cảnh cụ thể hơn giống như Bitcoin, Tính bền vững là điều cần thiết để đảm bảo các khoản tín dụng là cuối cùng và chúng xảy ra tại một "thời điểm" nhất định trong dòng thời gian của hệ thống (được xác định ngầm bởi chính sổ cái).

Tính bền vững là hữu ích nhưng không đủ để đảm bảo sổ cái tiến triển, tức là các giao dịch cuối cùng sẽ được đưa vào một chuỗi. Điều này được thuộc tính tính sống động ghi lại, miễn là giao dịch đến từ một chủ tài khoản trung thực và được môi trường cung cấp cho tất cả người chơi trung thực thì giao dịch đó sẽ được đưa vào sổ cái của những người chơi trung thực, giả sử môi trường tiếp tục cung cấp nó làm đầu vào cho đủ số vòng.

Để biết thêm chi tiết về đặc điểm kỹ thuật của sổ cái giao dịch mạnh mẽ, đặc biệt là các giao dịch và sổ cái giống Bitcoin, hãy tham khảo [10, 11].

C Chuỗi Martingale và các sự kiện toán học khác

Định nghĩa 9. *[18, Chương 12] Dãy số ngẫu nhiên $X_0, X_1, \ldots$ là một Martingale đối với dãy $Y_0, Y_1, \ldots$, nếu với mọi $n \geq 0$, (1) X_n là hàm số của $Y_0, \ldots, Y_n$, (2) $\mathbf{E}[|X_n|] < \infty$, và (3) $\mathbf{E}[X_{n+1} | Y_0, \ldots, Y_n] = X_n$.*

Định lý 6. *[17, Định lý 3.15] Cho $X_0, X_1, \ldots$ là một Martingale đối với dãy $Y_0, Y_1, \ldots$. Với $n \geq 0$, cho*

$$V = \sum_{i=1}^{n} \text{var}(X_i - X_{i-1}|Y_0, \dots, Y_{i-1}) \text{ và } b = \max_{1 \le i \le n} \sup(X_i - X_{i-1}|Y_0, \dots, Y_{i-1}),$$

trong đó sup được đảm nhận tất cả các phép gán có thể có cho $Y_0, \dots, Y_{i-1}$. *Khi đó, với mọi t,* $v \geq 0$,

$$\mathbf{Pr}[(X_n \geq X_0 + t) \wedge (V \leq v)] \leq \exp\left\{-\frac{t^2}{2v+2bt/3}\right\}.$$

Sự thật 2. *Giả sử* $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *là lõm và* $f(0) \geq 0$. *Khi đó, với mọi* $x, y \in [0, \infty)$ *và* $\lambda \in [1, \infty)$, $f(x/\lambda) \geq f(x)/\lambda$, $f(x\lambda) \leq \lambda f(x)$, $f(x + y) \leq f(x) + f(y)$.

Các bất đẳng thức nổi tiếng sau đây có thể được sử dụng mà không cần tham chiếu.

Sự thật 3. *(1)* $1 + x < e^x$, *với mọi x. (2)* $-\frac{x}{1-x} < \ln(1 - x)$, *với* $x \in (0,1)$. *(3)* $\frac{x}{1+x/2} < \ln(1 + x) < x$, *với* $x > 0$.

Người dịch: Nguyễn Văn Tú
Telegram: http://t.me/Tulibra
Link gốc: https://iohk.io/en/research/library/papers/the-bitcoin-backbone-protocol-with-chains-of-variable-difficulty/