Giao thức Threshold Signature Scheme (TSS)

1. **Giới thiệu**

Trong các hệ thống ví tiền mã hóa truyền thống, khóa riêng (private key) thường được lưu trữ tập trung và sử dụng trực tiếp để ký giao dịch. Mô hình này tiềm ẩn rủi ro nghiêm trọng về bảo mật do tồn tại điểm lỗi đơn lẻ (single point of failure): nếu khóa riêng bị lộ, bị đánh cắp hoặc mất mát, toàn bộ tài sản của người dùng có thể bị chiếm đoạt vĩnh viễn.

Để giải quyết triệt để vấn đề này, hệ thống ví của chúng tôi áp dụng Threshold Signature Scheme (TSS) – một kỹ thuật thuộc lĩnh vực Multi-Party Computation (MPC). Thay vì tạo và lưu trữ khóa riêng theo cách truyền thống, TSS cho phép tạo khóa phân tán và chia nhỏ khóa riêng thành nhiều phần (key shares). Mỗi phần được phân phối tới một MPC Party – là các thực thể độc lập, không tin cậy lẫn nhau.

Không một party nào có thể đơn phương tái dựng hoặc sử dụng toàn bộ khóa riêng. Việc ký giao dịch chỉ được thực hiện khi có ít nhất t trong số n party phối hợp tính toán – đúng theo ngưỡng bảo mật đã định sẵn. Mô hình này vừa đảm bảo tính bảo mật cao, vừa chống giả mạo, đồng thời giúp hệ thống duy trì hoạt động ngay cả khi một số node gặp sự cố.

1. **Cơ chế khởi tạo khóa bí mật phân tán sử dụng Threshold Signature Scheme (TSS)**
2. **Quá trình tạo khóa**

**A diagram of a company

AI-generated content may be incorrect.**

Khi người dùng khởi tạo yêu cầu tạo ví mới, hệ thống sẽ kích hoạt một phiên khởi tạo khóa phân tán (Distributed Key Generation – DKG), là một thành phần cốt lõi trong kiến trúc chữ ký ngưỡng (Threshold Signature Scheme – TSS). Cơ chế này được triển khai dựa trên thư viện mã nguồn mở [bnb-chain/tss-lib](https://github.com/bnb-chain/tss-lib), tuân thủ thuật toán chữ ký Elliptic Curve Digital Signature Algorithm (ECDSA) – tiêu chuẩn được Ethereum và nhiều nền tảng blockchain công nhận và sử dụng rộng rãi.

Toàn bộ quy trình tạo khóa không diễn ra trên một node duy nhất mà được thực hiện theo cách phi tập trung và an toàn theo ngưỡng (threshold), trong đó các bên tham gia (MPC parties) cùng nhau tạo ra một khóa công khai chung mà không bên nào nắm giữ toàn bộ khóa riêng.

**Bước 1: Phân phối yêu cầu đến các MPC Nodes**

Sau khi nhận tín hiệu tạo ví từ phía client, backend phân phối yêu cầu tới tập n thực thể tham gia tính toán bảo mật đa bên (Multi-Party Computation nodes). Mỗi thực thể (hay còn gọi là MPC party) hoạt động như một node độc lập, không tin cậy lẫn nhau, và có vai trò ngang hàng trong việc đồng thuận và xây dựng khóa bí mật.

**Bước 2. Thực thi Distributed Key Generation (DKG)**

Mỗi node sẽ:

* Tự tạo một thành phần ngẫu nhiên (gọi là "bí mật cục bộ").
* Sử dụng kỹ thuật **chia sẻ bí mật có thể xác minh** (Verifiable Secret Sharing – VSS) để phân phối phần bí mật này cho các bên khác, kèm theo bằng chứng mật mã đảm bảo tính hợp lệ.
* Nhận và xác minh phần chia sẻ từ các bên còn lại.
* Tổng hợp các phần nhận được thành **một đoạn khóa riêng cá nhân** – còn gọi là share – và giữ cục bộ.

Sau khi hoàn tất quá trình, toàn hệ thống đồng thuận trên một **khóa công khai chung** được tính toán từ các bí mật cá nhân đã chia. Khóa công khai này sẽ đại diện cho ví mà người dùng yêu cầu, và có thể được sử dụng để xác minh chữ ký sau này.

Đặc biệt, không có thực thể nào trong hệ thống có thể đơn phương xây dựng lại toàn bộ khóa riêng. Việc sử dụng các share để tham gia ký số yêu cầu **một số lượng tối thiểu các bên (theo ngưỡng t)** phải phối hợp.

**Bước 3. Lưu trữ share và đảm bảo tính bảo mật ngưỡng**

Mỗi thực thể sau khi hoàn tất DKG sẽ lưu trữ phần khóa cục bộ một cách biệt lập. Những phần này hoàn toàn vô dụng nếu bị khai thác đơn lẻ, và chỉ có thể tham gia ký số khi có ít nhất t thực thể đồng thuận thực hiện giao thức ký tập thể.

Cơ chế này bảo đảm:

* Không có thực thể nào có thể đơn phương khôi phục hoặc lạm dụng khóa bí mật.
* Hệ thống vẫn hoạt động ổn định ngay cả khi một số thực thể ngừng hoạt động hoặc bị tấn công.
* Quy trình ký ECDSA ngưỡng có thể được thực thi an toàn mà không cần phải tái kết hợp khóa bí mật ở bất kỳ thời điểm nào.

Cách tiếp cận này giúp hệ thống kháng lỗi, kháng tấn công và phù hợp với yêu cầu bảo mật cao trong môi trường blockchain.

1. **Bảo mật trong quá trình phân chia và lưu trữ khóa riêng**

Việc bảo mật khóa bí mật trong hệ thống ký ngưỡng không chỉ phụ thuộc vào thuật toán mật mã cơ bản, mà còn được đảm bảo thông qua một loạt cơ chế phân tán và hợp tác giữa các thực thể tham gia. Hệ thống sử dụng Threshold Signature Scheme (TSS) kết hợp với Distributed Key Generation (DKG) để xây dựng một kiến trúc không tập trung, vừa chống được các hình thức tấn công truyền thống, vừa có khả năng phục hồi cao trong môi trường bất định như blockchain hoặc mạng ngang hang

* **Phân phối khóa theo mô hình bất khả tri (Non-leakable, Non-reconstructable)**

Trong kiến trúc TSS, không có thực thể nào nắm giữ toàn bộ khóa bí mật. Thay vào đó, mỗi thực thể (MPC party) chỉ lưu trữ một share – tức là một thành phần của khóa riêng – được sinh ra trong quá trình DKG. Các share này được xây dựng theo cách bất khả tri lẫn nhau, tức là thông tin từ một share không thể tiết lộ hoặc dự đoán bất kỳ thông tin nào về phần còn lại của khóa.

Do đó, việc rò rỉ hoặc chiếm đoạt một share đơn lẻ là vô nghĩa về mặt mật mã, và không thể dùng để tái dựng khóa riêng hay tạo chữ ký giả.

* **Bảo mật hợp tác theo ngưỡng (Threshold-based Collaborative Security)**

Hệ thống yêu cầu tối thiểu t trong tổng số n thực thể tham gia phối hợp để thực thi một hành động mật mã như ký số. Đây là mô hình (t, n)-threshold, trong đó:

* Nếu số lượng thực thể hợp tác < t, mọi thao tác đều vô hiệu và không thể hoàn thành.
* Nếu số lượng ≥ t, các share được kết hợp một cách bảo mật để tính toán đầu ra hợp lệ (chữ ký).

Cơ chế này không chỉ đảm bảo tính phi tập trung trong kiểm soát khóa, mà còn ngăn chặn mọi hành vi thao túng đơn phương từ bất kỳ thực thể nào trong hệ thống.

* **Khả năng chịu lỗi và phục hồi (Fault Tolerance and Resilience)**

Một trong những ưu điểm then chốt của TSS là khả năng tiếp tục hoạt động ngay cả khi một phần hệ thống không khả dụng. Trong môi trường thực tế, việc một số node bị mất kết nối, hỏng hóc hoặc bị tấn công là điều khó tránh khỏi. Tuy nhiên, miễn là số node còn lại đáp ứng ngưỡng t, hệ thống vẫn có thể thực thi các tác vụ như tạo chữ ký, xác thực giao dịch.

Điều này mang lại tính khả dụng cao (high availability) và khả năng chịu lỗi Byzantine, đặc biệt hữu ích trong các hệ thống phân tán không đồng nhất.

* **Đảm bảo tính toàn vẹn và chống gian lận thông qua DKG**

Quá trình DKG không chỉ tạo ra khóa chung một cách phân tán mà còn sử dụng cơ chế Verifiable Secret Sharing (VSS) để đảm bảo rằng mọi dữ liệu chia sẻ giữa các thực thể đều có thể xác minh công khai mà không tiết lộ bí mật. Điều này ngăn chặn các thực thể gửi dữ liệu sai lệch hoặc giả mạo.

Mỗi share đi kèm với cam kết mật mã (commitment) và được xác thực chéo giữa các bên tham gia, giúp đảm bảo tính toàn vẹn và sự tin cậy lẫn nhau trong môi trường không đồng nhất, không tin cậy (trustless).

* **Tăng cường bảo vệ trước các cuộc tấn công chủ động**

Mô hình bảo mật ngưỡng còn mang lại lợi thế khi chống lại các hình thức tấn công chủ động, như:

* *Tấn công chiếm quyền kiểm soát node (node compromise):* Kẻ tấn công cần kiểm soát ít nhất t node để thực hiện ký giả, điều này làm tăng đáng kể chi phí và độ phức tạp khi tấn công.
* *Tấn công từ bên trong (insider threats):* Không một thực thể nào có thể độc lập tạo hoặc lạm dụng khóa riêng, do đó hạn chế nguy cơ thao túng từ nội bộ.

1. **Quy trình ký giao dịch phân tán với chữ ký ngưỡng (Threshold Signing Process)**
   1. **Quy trình ký giao dịch**

**A screenshot of a computer

AI-generated content may be incorrect.**

Khi người dùng thực hiện một giao dịch, hệ thống sẽ tiến hành một quy trình ký phân tán, trong đó nhiều node MPC (Multi-Party Computation) phối hợp để cùng nhau thực hiện việc ký giao dịch theo chuẩn ECDSA – mà không có bất kỳ node nào nắm giữ toàn bộ khóa riêng. Mỗi node chỉ giữ một phần khóa (share), và chỉ khi đủ số lượng node tối thiểu (đạt ngưỡng), hệ thống mới có thể hoàn tất việc ký.

**Bước 1: Giao dịch được khởi tạo bởi người dùng**

Khi người dùng thực hiện một hành động chuyển tiền (ETH hoặc token), ứng dụng sẽ tạo một giao dịch Ethereum chuẩn, bao gồm các tham số như địa chỉ đích, giá trị, phí gas, nonce, chain ID… Giao dịch này được encode ở định dạng RLP, sẵn sàng để ký.

**Bước 2: Backend tiếp nhận yêu cầu và khởi tạo phiên ký**

Giao dịch chưa được ký được gửi từ client lên backend. Tại đây, hệ thống tạo một phiên ký ngưỡng mới (signing session), gắn định danh duy nhất và truyền yêu cầu ký đến các MPC Party – là những node đã giữ sẵn các phần khóa riêng từ quá trình DKG trước đó.

**Bước 3. Các MPC Party tham gia giao thức ký phân tán**

Quá trình ký diễn ra theo một giao thức phân tán nhiều vòng, trong đó các node MPC (MPC Party) phối hợp tính toán chữ ký số mà không để lộ bất kỳ thông tin bí mật nào. Cụ thể, mỗi node thực hiện các bước sau:

* *Khởi tạo nội bộ:* Mỗi node sinh các giá trị nonce ngẫu nhiên và tạo ra các cam kết mật mã (cryptographic commitments) tương ứng. Các giá trị này được bảo vệ bởi cơ chế mã hóa nhằm đảm bảo không tiết lộ thông tin nhạy cảm trong quá trình trao đổi.
* *Xác minh chéo giữa các node:* Các cam kết được gửi qua lại giữa các node để tiến hành xác minh chéo (cross-verification). Mỗi MPC Party kiểm tra tính hợp lệ của các giá trị mà các party khác cung cấp, thông qua các bằng chứng mật mã, điển hình là bằng chứng không tiết lộ (zero-knowledge proof). Cơ chế này đảm bảo rằng tất cả các node đều tuân thủ đúng quy trình, không gian lận và không gửi thông tin giả mạo. Nếu phát hiện có dữ liệu sai lệch hoặc vi phạm giao thức, quá trình ký sẽ bị hủy để đảm bảo an toàn.
* *Tính toán phần chữ ký***:** Khi các xác minh chéo đều thành công và đủ số lượng node tham gia đạt ngưỡng (threshold), mỗi node sử dụng phần khóa của mình để tính toán các thành phần trung gian của chữ ký ECDSA, bao gồm các tham số liên quan đến r và s.
* *Tổng hợp chữ ký:* Các phần dữ liệu trung gian từ các node được backend thu thập và xử lý để tính ra cặp chữ ký số (r, s). Toàn bộ quá trình không yêu cầu bất kỳ node nào tái dựng toàn bộ khóa riêng, giúp đảm bảo tính bảo mật theo nguyên tắc không điểm yếu đơn lẻ (no single point of failure).

Giao thức được triển khai dựa trên thư viện mã nguồn mở bnb-chain/tss-lib, được thiết kế để hoạt động an toàn trong môi trường phân tán và chống lại các hành vi gian lận, với sự hỗ trợ của cơ chế xác minh chéo và các biện pháp mã hóa tiên tiến.

**Bước 4. Tính toán phần v và hoàn thiện chữ ký**

Mặc dù (r, s) là đủ để xác minh chữ ký, Ethereum yêu cầu thêm tham số v (recovery ID) để xác định chính xác public key tương ứng. Do đó, backend sử dụng ecrecover và khóa công khai để tính toán v tương ứng. Quá trình này không liên quan đến bất kỳ phần bí mật nào nên hoàn toàn an toàn khi thực hiện phía backend.

**Bước 5. Gắn chữ ký và gửi giao dịch lên blockchain**

Sau khi có đủ ba thành phần (r, s, v), hệ thống ghép chúng vào cấu trúc giao dịch Ethereum đã encode. Giao dịch được phát tán lên mạng thông qua API như eth\_sendRawTransaction. Tại thời điểm này, hệ thống chỉ cần đợi blockchain xác nhận giao dịch.

**Bước 6. Phản hồi trạng thái cho người dùng**

Khi giao dịch được mining và ghi vào block, backend gửi phản hồi về client bao gồm:

* Mã giao dịch (transaction hash)
* Trạng thái giao dịch (success/failure)
* Thời gian xác nhận, block số, lượng gas sử dụng,…

Thông tin này giúp người dùng kiểm chứng giao dịch trực tiếp trên blockchain.

* 1. **Tính bảo mật của quy trình ký**
* *Không tái dựng khóa riêng:* Tại mọi thời điểm, khóa riêng không bao giờ được tái cấu trúc hay tồn tại dưới dạng đầy đủ. Mỗi node chỉ giữ một phần khóa, và các phần này không thể tự giải mã hoặc kết hợp lại bên ngoài giao thức đã định.
* *Ngăn chặn giả mạo chữ ký:* Việc tạo ra một chữ ký hợp lệ đòi hỏi sự phối hợp của ít nhất t node hợp lệ. Điều này đồng nghĩa: kẻ tấn công buộc phải kiểm soát đồng thời ≥ t node để giả mạo chữ ký — một mục tiêu gần như bất khả thi nếu hệ thống được triển khai phân tán đúng cách.
* *Khả năng chịu lỗi cao:* Hệ thống vẫn có thể tiến hành ký ngay cả khi một số node bị lỗi, mất kết nối hoặc bị tấn công. Chỉ cần đủ số node theo ngưỡng (threshold), quá trình ký vẫn được đảm bảo an toàn và liền mạch.

1. **Triển khai trong hệ thống**

Trong dự án, chúng tôi triển khai các thành phần TSS như sau:

* Ngôn ngữ: Go
* Thư viện chính: bnb-chain/tss-lib
* Giao tiếp: gRPC cho request/reponse giữa client và các node
* Đồng bộ: Redis sử dụng cơ chế Pub/Sub để thông báo và nhận kết quả từ các node
* Lưu trữ key share: PostgreSQL – mỗi node lưu phần key share của mình sau khi sinh khóa, dữ liệu được mã hóa trước khi lưu
* Ngưỡng chữ ký: 2 trên 3 node (t = 2, n = 3) – chỉ cần đạt đủ ngưỡng số lượng node tham gia là có thể tạo ra chữ ký hợp lệ
* Dữ liệu share: mỗi node đồng thời lưu local (dạng file .txt) và gửi về client, có thể được phục hồi khi cần ký
* Kết quả ký: chữ ký số (r, s) được trả về client thông qua Redis và có thể xác minh bằng khóa công khai tương ứng