

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ TRI THỨC

HỒ THỊ HÀ – NGUYỄN HOÀNG HẢI

LUYỆN BỬU HUY – NGUYỄN THỊ THANH TRANG

**KHẢO SÁT MỘT SỐ
BLOCKCHAIN RIÊNG VÀ ỨNG DỤNG**

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN CNTT

TP. HCM, 2019

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC**

| | |
|-------------------------------|------------------|
| HỒ THỊ HÀ | – 1512135 |
| NGUYỄN HOÀNG HẢI | – 1512140 |
| LUYỆN BỬU HUY | – 1512199 |
| NGUYỄN THỊ THANH TRANG | – 1512587 |

**KHẢO SÁT MỘT SỐ
BLOCKCHAIN RIÊNG VÀ ỨNG DỤNG**

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN CNTT

**GIÁO VIÊN HƯỚNG DẪN
PGS. TS. NGUYỄN ĐÌNH THỨC**

KHÓA 2015 – 2019

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

[Ký tên và ghi rõ họ tên]

[illegible]

TpHCM, ngày tháng năm 2019

Giáo viên phản biện

[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Chúng em chân thành cảm ơn thầy **PGS. TS. Nguyễn Đình Thúc**, người đã hướng dẫn cho chúng em trong suốt thời gian thực hiện đồ án. Mặc dù thầy rất bận nhưng vẫn tận tình chỉ dẫn, định hướng đi cho chúng em để chúng em hoàn thành tốt nhiệm vụ. Một lần nữa, chúng em chân thành cảm ơn thầy và chúc thầy dồi dào sức khỏe.

Chúng em cũng xin cảm ơn thầy cô giáo Khoa Công nghệ Thông tin đã dìu dắt, truyền đạt nhiều kiến thức bổ ích cho chúng em trong suốt thời gian qua.

Cuối cùng, chúng con xin cảm ơn ba mẹ, gia đình và bạn bè đã luôn luôn ủng hộ và động viên chúng con trong quá trình học tập nói chung và quá trình thực hiện đề tài nói riêng.

Mặc dù chúng tôi đã cố gắng rất nhiều trong quá trình thực hiện đề tài, tuy nhiên vì kiến thức chuyên môn còn hạn chế và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu sót, chúng em rất mong nhận được sự góp ý, chỉ bảo của quý thầy cô để báo cáo này được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Tp. HCM, tháng 06 năm 2019

Nhóm thực hiện

Hồ Thị Hà & Nguyễn Hoàng Hải

Luyện Bửu Huy & Nguyễn Thị Thanh Trang

Khoa Công Nghệ Thông Tin
Bộ môn Công Nghệ Tri Thức

ĐỀ CƯƠNG CHI TIẾT

| | |
|--|-----------|
| Tên Đề Tài: Khảo sát một số Blockchain riêng và ứng dụng | |
| Giáo viên hướng dẫn: PGS. TS. Nguyễn Đình Thúc | |
| Thời gian thực hiện: Từ ngày 25/02/2019 đến ngày 30/06/2019 | |
| Sinh viên thực hiện: Hồ Thị Hà | – 1512135 |
| Nguyễn Hoàng Hải | – 1512140 |
| Luyện Bửu Huy | – 1512199 |
| Nguyễn Thị Thanh Trang | – 1512587 |
| Loại đề tài: Phát triển ứng dụng (Hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung) | |

| |
|---|
| Nội Dung Đề Tài: |
| Đề tài bao gồm các phần sau: |
| - Tìm hiểu công nghệ Blockchain |
| - Tìm hiểu Private Blockchain |
| - Tìm hiểu các hướng nghiên cứu dựa trên công nghệ Blockchain cũng như Private Blockchain |
| - Tìm hiểu framework Multichain |

- Tìm hiểu framework Storj
- Phát triển ứng dụng hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung

Kết quả cần đạt được sau khi thực hiện đồ án:

- Ứng dụng đáp ứng việc hỗ trợ xác thực chứng chỉ
- Ứng dụng được triển khai trên các máy vi tính
- Sử dụng tốt framework và các công cụ liên quan.

Kế Hoạch Thực Hiện:

- 25/02/2019 đến 05/03/2019: Tìm hiểu khái niệm tổng quan về Blockchain
- 06/03/2019 đến 15/03/2019: Tìm hiểu khái niệm Private Blockchain
- 16/03/2019 đến 25/03/2019: Tìm hiểu các tính chất đặc trưng của Private Blockchain
- 26/03/2019 đến 10/04/2019: Tìm hiểu Multichain và Storj
- 11/04/2019 đến 25/04/2019: Tiến hành cài đặt Multichain và Storj
- 26/04/2019 đến 01/05/2019: Cấu hình Multichain và Storj
- 02/05/2019 đến 14/6/2019: Tích hợp Multichain và Storj vào ứng dụng hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung
- 15/06/2019 đến 30/06/2019: Tổng hợp tài liệu, phân tích và hoàn thành báo cáo hoàn chỉnh

Phân Công:

- Cả nhóm: Tìm hiểu về Blockchain, Private Blockchain, tính chất của Private Blockchain
- Hà và Trang: Tìm hiểu và cài đặt về Multichain

- Hải và Huy: Tìm hiểu và cài đặt về Storj
- Hải và Trang: Nghiên cứu và phát triển ứng dụng xác thực chứng chỉ phi tập trung
- Hà và Huy: Tập hợp dữ liệu hoàn thành báo cáo

Xác nhận của GVHD

Ngày 28 tháng 05 năm 2019

SV Thực hiện

Hồ Thị Hà

Nguyễn Hoàng Hải

Nguyễn Đình Thúc

Luyện Bửu Huy Nguyễn Thị Thanh Trang



MỤC LỤC

| | |
|---|-----------|
| DANH MỤC HÌNH..... | 4 |
| DANH MỤC BẢNG..... | 6 |
| DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT | 7 |
| TÓM TẮT ĐỒ ÁN | 10 |
| Chương 1 MỞ ĐẦU | 11 |
| 1.1. Giới thiệu chung | 11 |
| 1.2. Lý do chọn đề tài | 12 |
| 1.3. Mục tiêu đề tài | 12 |
| 1.4. Đối tượng và phạm vi nghiên cứu | 13 |
| 1.5. Nội dung đề tài | 13 |
| Chương 2 TỔNG QUAN VỀ BLOCKCHAIN..... | 15 |
| 2.1. Tổng quan về sự ra đời của Blockchain | 15 |
| 2.2. Các hướng nghiên cứu của công nghệ Blockchain trong và ngoài nước | 16 |
| 2.3. Vấn đề mà đề tài tập trung, nghiên cứu giải quyết..... | 19 |
| 2.3.1. Các loại Blockchain..... | 19 |
| 2.3.2. Ưu điểm của Private Blockchain | 20 |
| Chương 3 NGHIÊN CỨU LÝ THUYẾT | 21 |
| 3.1. Cơ sở lý thuyết, lý luận và phương pháp nghiên cứu mà Private Blockchain đã áp dụng..... | 21 |
| 3.2. Giới thiệu hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung..... | 24 |
| 3.2.1. Giới thiệu MultiChain..... | 24 |
| 3.2.1.1. Cơ chế đồng thuận cấp quyền trong MultiChain | 25 |
| 3.2.1.2. Cơ chế phát hành tiền tệ trên MultiChain | 26 |
| 3.2.1.3. Bảo mật trên luồng trong MultiChain | 26 |
| 3.2.1.4. Định dạng address và key | 31 |
| 3.2.2. Giới thiệu storj | 34 |
| 3.2.2.1. Một số tính chất của Storj | 35 |

| | |
|---|-----------|
| 3.2.2.2. Cơ chế hoạt động của Storj | 42 |
| 3.3.3. Giới thiệu Decentralized Certificate Authority (DeCert [19]) | 51 |
| Chương 4 TRÌNH BÀY, ĐÁNH GIÁ, BÀN LUẬN VỀ CÁC KẾT QUẢ | 55 |
| 4.1. Giới thiệu hệ thống cung cấp xác thực chứng chỉ phi tập trung | 55 |
| 4.2. Mô tả hệ thống | 55 |
| 4.3. Mô hình tiến hành (UML: sơ đồ lớp và sơ đồ tuần tự) | 57 |
| 4.4. Quy trình cài đặt hệ thống xác thực chứng chỉ phi tập trung và kết quả | 58 |
| Chương 5 KẾT LUẬN | 70 |
| 5.1. Kết quả đạt được | 70 |
| 5.2. Đóng góp mới và đề xuất mới | 71 |
| Chương 6 HƯỚNG PHÁT TRIỂN | 72 |
| 6.1. Kiến nghị hướng phát triển tiếp theo cho công nghệ Blockchain | 72 |
| 6.2. Hướng phát triển của hệ thống xác thực chứng chỉ phi tập trung | 72 |
| DANH MỤC TÀI LIỆU THAM KHẢO | 73 |
| PHỤ LỤC | 76 |
| A. CHỮ KÝ ĐIỆN TỬ | 77 |
| A.1. Tổng quan | 77 |
| A.2. Chữ ký số | 77 |
| A.3. Chứng chỉ số | 77 |
| A.4. Ứng dụng của chữ ký điện tử | 78 |
| A.5. Lợi ích khi sử dụng chữ ký điện tử | 78 |
| B. AES-GCM | 79 |
| C. SecretBox | 80 |
| D. Cài đặt và cấu hình MultiChain | 81 |
| D.1. Yêu cầu hệ thống | 81 |
| D.2. Cài đặt trên Window | 81 |
| D.3. Cài đặt trên Linux | 81 |
| D.4. Cách tạo chain và kết nối các node với nhau trong Multichain | 82 |
| D.5. Các thông số Blockchain Parameters | 85 |

| | |
|--|-----------|
| E. Cài đặt và cấu hình Storj | 87 |
| E.1. Chuẩn bị môi trường cài đặt storj | 87 |
| E.2. Cấu hình storj..... | 87 |
| E.3. Cấu hình Uplink..... | 90 |

DANH MỤC HÌNH

| | |
|--|----|
| Hình 3. 1: Cách lưu file bằng cơ chế off-chain trên Multichain | 29 |
| Hình 3. 2: Bảng năng suất của Multichain qua từng phiên bản | 31 |
| Hình 3. 3: Cách tạo địa chỉ trên Bitcoin từ public key..... | 32 |
| Hình 3. 4: Mô hình xác thực và định danh nút lưu trữ..... | 39 |
| Hình 3. 5: Các lớp kết nối trong mô hình mạng Storj | 48 |
| Hình 3. 6: Mô hình hoạt động quá trình tải lên và tải xuống trong mạng Storj | 49 |
| Hình 3. 7: Sơ đồ hoạt động của quá trình đăng tải dữ liệu trong mạng Storj | 49 |
| Hình 3. 8: Sơ đồ hoạt động của quá trình tải xuống dữ liệu trong mạng Storj | 50 |
| | |
| Hình 4. 1: Sơ đồ lớp mô hình DeCert | 57 |
| Hình 4. 2: Sơ đồ tuần tự mô hình DeCert | 58 |
| Hình 4. 3: Hình minh họa Bước 1 – Cài đặt server..... | 59 |
| Hình 4. 4: Hình minh họa Bước 1 – Chạy server..... | 60 |
| Hình 4. 5: Hình minh họa Bước 2 – Service Provider 1 kết nối server | 61 |
| Hình 4. 6: Hình minh họa Bước 2 – Service Provider 1 tạo domain | 62 |
| Hình 4. 7: Hình minh họa Bước 2 – Service Provider 1 đăng nhập | 62 |
| Hình 4. 8: Hình minh họa Bước 3 – User kết nối server | 63 |
| Hình 4. 9: Hình minh họa Bước 3 – User yêu cầu lấy chứng chỉ | 64 |
| Hình 4. 10: Hình minh họa Bước 4 – Service Provider 1 đăng nhập | 65 |
| Hình 4. 11: Hình minh họa Bước 4 – Service Provider 1 chấp nhận yêu cầu từ User | 65 |
| Hình 4. 12: Hình minh họa Bước 5 – User kiểm tra phản hồi | 66 |
| Hình 4. 13: Hình minh họa Bước 6 – Service Provider 2 kiểm tra chứng chỉ..... | 67 |
| Hình 4. 14: Hình minh họa Bước 6 – Service Provider 2 tải xuống tập tin chứng chỉ | 67 |
| Hình 4. 15: Hình minh họa Bước 7 – Voter kết nối server và kiểm tra các chứng chỉ | 68 |

| | |
|---|----|
| Hình 4. 16: Hình minh họa Bước 7 – Voter xác thực và tiến hành vote..... | 69 |
| Hình 4. 17: Hình minh họa Bước 7 – Service Provider 2 kiểm tra lại chứng chỉ..... | 69 |
| Hình B. 1: : Sơ đồ hoạt động AES-GCM..... | 79 |
| Hình D. 1: Hình minh họa Bước 1 - Tạo chain..... | 82 |
| Hình D. 2: Hình minh họa Bước 2 - Chỉnh sửa thông số..... | 83 |
| Hình D. 3: Hình minh họa Bước 3 - Khởi tạo block, sẵn sàng kết nối | 83 |
| Hình D. 4: Hình minh họa Bước 4 - Node kết nối vào chain | 84 |
| Hình D. 5: Hình minh họa Bước 5 - Cấp quyền cho node..... | 84 |
| Hình D. 6: Hình minh họa Bước 5 - Node kết nối thành công | 84 |
| Hình E. 1: Hình minh họa - Access key và Secret key | 89 |
| Hình E. 2: Giao diện khởi động của Minio | 89 |
| Hình E. 3: Hình minh họa sử dụng Minio tải lên tập tin..... | 90 |

DANH MỤC BẢNG

| | |
|---|----|
| Bảng D. 1: Chain Parameters cơ bản trong Multichain | 85 |
| Bảng D. 2: Global Permissions cơ bản trong Multichain | 86 |

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

| | |
|---------------|---|
| Address | Địa chỉ IP định danh một máy tính trong mạng máy tính, địa chỉ định danh người sở hữu |
| Admin | Người quản trị |
| AES-GCM | Một thuật toán mã hóa |
| API | Giao diện lập trình ứng dụng, phần mềm trung gian, cho phép các ứng dụng giao tiếp với nhau |
| Assets | Tài sản, thứ có giá trị |
| Bitcoin | Tiền điện tử đầu tiên trên thế giới |
| Block | Khối |
| Bucket | Thùng chứa |
| Byte | Đơn vị lưu trữ dữ liệu trong máy tính |
| Chain | Chuỗi mắt xích |
| Checksum | Kiểm tra dữ liệu với mục đích phát hiện lỗi |
| Chunk | Mảnh, phân đoạn dữ liệu |
| Client | Máy khách |
| CNTT | Công nghệ thông tin |
| Connect | Sự kết nối giữa hai máy tính |
| Data | Dữ liệu |
| Database | Cơ sở dữ liệu |
| Document | Tài liệu |
| ECDSA | Thuật toán chữ ký số áp dụng đường cong Elliptic |
| Ethereum | Một nền tảng điện toán có tính chất phân tán, công cộng dựa trên công nghệ Blockchain |
| File | Tập tin |
| Framework | Khung ứng dụng trong phát triển phần mềm |
| Full data | Toàn bộ dữ liệu |
| Genesis block | Khối được khởi tạo đầu tiên |

| | |
|------------------------|---|
| Get | Lấy, kéo dữ liệu về |
| Hash | Hàm băm chuyển đổi một giá trị sang giá trị khác |
| Header | Các thông số cần thiết cho biết các đặc trưng của dữ liệu |
| ID | Thông số định danh duy nhất |
| IoT | Kết nối vạn vật vào mạng lưới thông tin |
| IP | Dãy số đặc trưng cho địa chỉ của một máy tính trong mạng lưới |
| Key | Khóa |
| Key-value | Cặp giá trị tên, giá trị |
| Multichain | Nền tảng mã nguồn mở cho ứng dụng Blockchain |
| NAT | Cơ chế ánh xạ giữa địa chỉ IP nội miền và ngoại miền |
| Network | Mạng lưới máy tính |
| Node | Một nút trong mạng lưới |
| Off-chain | Lưu một phần dữ liệu trên chuỗi |
| On-chain | Lưu toàn bộ dữ liệu trên chuỗi |
| Path | Đường dẫn đến nơi lưu trữ trên máy tính |
| Ping | Một giao thức để kiểm tra xem có kết nối được không |
| Peer-to-peer handshark | Sự bắt tay, kết nối giữa hai máy tính ngang hàng |
| Permission | Quyền truy cập |
| Private key | Khóa riêng tư |
| Protocol | Giao thức |
| Public key | Khóa công cộng |
| Query | Truy vấn |
| Root | Chỉ phần tử gốc, là nguồn cội đầu tiên |
| SecretBox | Bảng giá trị bí mật dùng để mã hóa |
| Send | Chỉ sự truyền, gửi dữ liệu đi |
| Server | Máy chủ |
| SHA-256 | Một thuật toán băm với 256 bit |
| Size | Kích thước |

| | |
|--------------|---|
| Source | Chỉ nguồn |
| Sqlite DB | Hệ quản trị cơ sở dữ liệu quan hệ |
| SSL | Secure Sockets Layer: Tầng socket bảo mật, giao thức trong mạng |
| Storj | Phần mềm mã nguồn mở, giải pháp lưu trữ tập tin phân cấp |
| Stream | Dòng lưu chuyển |
| Subscribe | Theo dõi, đồng thuận |
| Timestamping | Số giây kể từ một mốc thời gian nhất định |
| TLS | Transport Layer Security: Bảo mật tầng vận chuyển trong mạng |
| Token | Là chữ ký số để phân quyền truy cập tạm thời |
| TTL | Time-to-live: Thời gian tồn tại của tập tin |
| Transaction | Giao dịch |
| Uplink | Tải đường dẫn đến một dữ liệu lên mạng |
| Upload | Tải dữ liệu lên mạng máy tính |
| UML | Ngôn ngữ mô hình hóa thống nhất |
| Version | Phiên bản phát hành |
| XOR | Phép toán cộng hai chuỗi bit không lấy dư |
| Wallet | Ví |

TÓM TẮT ĐỒ ÁN

Đồ án này, khảo sát Blockchain là một cơ sở dữ liệu phân cấp lưu trữ thông tin trong các khối thông tin được liên kết với nhau bằng mã băm và mở rộng theo thời gian.

Blockchain đầu tiên được phát minh và thiết kế bởi Satoshi Nakamoto vào năm 2008. Công nghệ Blockchain tương đồng với cơ sở dữ liệu, chỉ khác ở việc tương tác với cơ sở dữ liệu.

Cụ thể hơn, đồ án khảo sát Private Blockchain, những lợi ích và những điểm mà Private Blockchain mạnh hơn Public Blockchain. Đồng thời sử dụng mã nguồn mở ứng dụng Private Blockchain là Multichain [\[3\]](#) vào xây dựng hệ thống ứng dụng.

Bên cạnh đó, chúng tôi cũng khảo sát thêm khả năng ứng dụng của Blockchain vào việc lưu trữ phi tập trung. Cụ thể, chúng tôi đã sử dụng mã nguồn mở Storj [\[1\]](#) và tiến hành khảo sát các công nghệ cũng như các thuật toán được dùng trong lưu trữ phi tập trung.

Sau khi tìm hiểu và cài đặt Multichain cũng như Storj, chúng tôi đã tiến hành các bước để xây dựng hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung, đảm bảo được tính kết hợp giữa khả năng quản lý người dùng của Multichain và khả năng lưu trữ phi tập trung của Storj.

Hệ thống mang tính thực tiễn có thể triển khai ngoài thực tế, giúp mọi người có thể xác thực chứng chỉ một cách hiệu quả trên mạng mà không cần phải đi xác thực nhiều nơi như hiện nay.

Chương 1

MỞ ĐẦU

1.1. Giới thiệu chung

Hiện nay nhu cầu thực hiện các giao dịch, lưu trữ hồ sơ dữ liệu tại các trung tâm dịch vụ tài chính, ngân hàng,... là diễn ra thường xuyên với số lượng lớn các giao dịch mỗi ngày. Điển hình tại các ngân hàng là trung tâm tiếp nhận những khách hàng tham gia giao dịch mỗi ngày. Để xử lý các giao dịch, trung tâm cần lưu các thông tin khách hàng, thông tin tài khoản khách hàng sở hữu, lịch sử các giao dịch mà khách hàng từng thực hiện... Vì vậy, vai trò của các trung tâm trung gian như ngân hàng đóng một vai trò rất quan trọng trong các giao dịch hiện nay. Song chính vì vai trò to lớn của các trung tâm kiểm soát các giao dịch diễn ra một cách hợp lệ và có hiệu lực, nên hiện nay người sử dụng dịch vụ để giao dịch cần phải chi trả phí giao dịch cho các trung tâm là khá cao và nhất là phải tin tưởng tuyệt đối vào trung tâm. Theo ước tính của tờ Economist, các ngân hàng trên thế giới đã thu tới 1,7 nghìn tỷ USD tiền xử lý giao dịch trong năm 2014 – tương đương với 2% GDP toàn cầu! [\[8\]](#).

Mặt khác, các thông tin lưu trữ tập trung và được kiểm soát tại các server của các ngân hàng cũng gây ra những vấn đề về toàn vẹn và bảo mật dữ liệu. Chúng ta đang đặt niềm tin hoàn toàn vào các trung tâm dịch vụ một cách tuyệt đối. Tuy nhiên, dữ liệu nhạy cảm của cá nhân có thể bị khai thác mà chúng ta không hề hay biết. Hoặc dữ liệu có thể bị chỉnh sửa gây ảnh hưởng đến quyền lợi của người sử dụng dịch vụ. Các server cũng có thể bị hư hoặc bị tấn công gây sai lệch dữ liệu hoặc mất hoàn toàn dữ liệu thì xem như toàn bộ giao dịch và tài sản của khách hàng không cách nào lấy được. Vì vậy việc lưu trữ tập trung tại các trung tâm tiềm ẩn nguy cơ mất an toàn và tính toàn vẹn dữ liệu.

Lượng lớn giao dịch diễn ra mỗi ngày với chi phí cho mỗi giao dịch khá cao và nguy cơ mất an toàn dữ liệu khi lưu trữ tập trung tại các server là vấn đề đáng suy ngẫm.

1.2. Lý do chọn đề tài

Liệu có cách thức hoạt động nào khác thay thế cho các trung tâm dịch vụ mà vẫn đảm bảo các giao dịch được thực hiện đúng quy trình, chính xác và an toàn dữ liệu, tiết kiệm chi phí giao dịch và cơ sở hạ tầng. Chúng tôi nhận thấy hiện nay trên thế giới đã và đang áp dụng hiệu quả công nghệ Blockchain đáp ứng được các tiêu chí nêu trên. Các giao dịch có thể diễn ra một cách tự động, chính xác không cần bên thứ ba như các trung tâm dịch vụ can thiệp. Không những thế Blockchain còn có tính bảo mật cao.

Những vụ tấn công vào hệ thống dữ liệu tập trung sẽ rất khó có thể thực hiện được đối với mạng lưới Blockchain. Blockchain là một cơ sở dữ liệu phân cấp lưu trữ thông tin trong các block thông tin được liên kết với nhau bằng mã hóa và mở rộng theo thời gian. Vì vậy, để tấn công thông tin trong một block cụ thể nào đó, người tấn công sẽ phải tấn công không chỉ block đó mà cả những block nằm trước nó, trên hàng triệu sổ cái trong mạng lưới cùng lúc. Điều này là việc không thể hay rất khó xảy ra trên một mạng lưới Blockchain rộng lớn. Tất cả những lợi ích mà Blockchain đem lại là tiền đề để chúng tôi tìm hiểu, nghiên cứu và phát triển các ứng dụng trên nền tảng Blockchain.

1.3. Mục tiêu đề tài

Tìm hiểu về cơ chế hoạt động, nguyên lý làm việc và khả năng ứng dụng của Blockchain, đặc biệt về Private Blockchain. Tìm hiểu các thành phần tạo nên Blockchain, Blockchain hoạt động dựa trên những cơ sở lý luận nào.

Tìm hiểu vai trò của Blockchain trong các ứng dụng công nghệ hiện nay. Giới thiệu ứng dụng hệ thống cung cấp xác thực chứng chỉ. Ứng dụng gồm những chức năng nào? Vận hành ra sao? Tiềm năng và thách thức gì khi triển khai ứng dụng? Đưa ra quy trình triển khai ứng dụng và đề xuất hướng phát triển tiếp theo.

Ngoài ra đồ án cũng cố gắng tạo hướng đi cơ bản để sử dụng và tiếp cận nghiên cứu phát triển ứng dụng với mã nguồn mở. Tạo ra một tài liệu tham khảo và học tập hữu ích đối với các bạn sinh viên CNTT hoặc những người quan tâm.

1.4. Đối tượng và phạm vi nghiên cứu

Đối tượng: Khảo sát một số Private Blockchain và ứng dụng. Triển khai ứng dụng hệ thống cung cấp xác thực chứng chỉ phi tập trung.

Phạm vi:

- Không gian: Có thể triển khai ứng dụng cho các ngân hàng, sở y tế và giáo dục, trung tâm hành chính quốc gia, công ty xí nghiệp...
- Thời gian: dự tính ứng dụng công nghệ Blockchain sẽ phát triển trong hiện tại và cả tương lai.

1.5. Nội dung đề tài

Nội dung đồ án gồm có 6 chương và 5 phụ lục, được cấu trúc gồm:

Chương 1: Mở đầu

Giới thiệu chung về đề tài, đồng thời nêu lên lý do chọn đề tài, mục tiêu, phạm vi và nội dung của khóa luận.

Chương 2: Tổng quan về Blockchain

Giới thiệu về Blockchain, các vấn đề mà đề tài tập trung giải quyết, các loại blockchain và ưu điểm của Private Blockchain.

Chương 3: Nghiên cứu lý thuyết

Giới thiệu cơ sở lý thuyết và phương pháp nghiên cứu Blockchain, Multichain, Storj, giới thiệu ứng dụng hệ thống xác thực chứng chỉ phi tập trung như một minh họa thực tiễn của đề án.

Chương 4: Trình bày, đánh giá bàn luận về các kết quả

Mô tả công việc nghiên cứu, cài đặt Multichain, cài đặt Storj, kết quả nghiên cứu, sơ đồ tiến hành.

Chương 5: Kết luận

Trình bày kết quả đạt được, đóng góp mới và đề xuất.

Chương 6: Hướng phát triển

Tổng kết lại những công việc mà chúng tôi đã thực hiện trong phạm vi đề tài này, những kết quả đạt được, hạn chế còn tồn tại và những hướng phát triển trong tương lai của đề tài.

Phụ lục A: Chữ ký điện tử

Giới thiệu về chữ ký điện tử, lợi ích cũng như ứng dụng của chữ ký điện tử.

Phụ lục B: AES – GCM

Giới thiệu về GCM và cách hoạt động của GCM

Phụ lục C: SecretBox

Giới thiệu chung về SecretBox và Scrypt

Phụ lục D: Cài đặt và cấu hình MultiChain

Hướng dẫn cách cài đặt và sử dụng MultiChain

Phụ lục E: Cài đặt và cấu hình Storj

Hướng dẫn cách cài đặt và sử dụng Storj

Chương 2

TỔNG QUAN VỀ BLOCKCHAIN

2.1. Tổng quan về sự ra đời của Blockchain

Blockchain đầu tiên được phát minh và thiết kế bởi Satoshi Nakamoto vào năm 2008 và được hiện thực hóa vào năm sau đó như là một phần cốt lõi của Bitcoin, công nghệ Blockchain đóng vai trò như là một cuốn sổ cái cho tất cả các giao dịch. Qua việc sử dụng mạng lưới ngang hàng và một hệ thống dữ liệu phân cấp, Blockchain được quản lý tự động. Công nghệ Blockchain đã trở thành nguồn cảm hứng cho một loạt các ứng dụng khác.

Sự xuất hiện của Blockchain cũng như các cột mốc khi máy tính cá nhân hoặc Internet ra đời, hệ thống này sẽ thay đổi cách mà chúng ta hiểu biết và nhìn nhận xã hội.

Công nghệ Blockchain mở ra một xu hướng mới cho các lĩnh vực như tài chính ngân hàng, logistics, điện tử viễn thông, kế toán kiểm toán...

Không chỉ thế Blockchain còn là nòng cốt của Internet vạn vật (IoT). Các thiết bị điện tử có thể giao tiếp một cách an toàn và minh bạch, những nỗ lực bất chính trong thế giới Internet sẽ không thực hiện được, và còn nhiều điều nữa...

Ngoài hợp đồng thông minh, điện toán đám mây, huy động vốn,...Blockchain còn được ứng dụng trong nhiều lĩnh vực như bầu cử trực tuyến, quản lý tài sản và chuỗi cung ứng, thanh toán quốc tế, dự đoán thị trường và sự kiện tương lai, mạng internet phi tập trung (Blockstack, IPFS), gọi vốn cộng đồng (WeiFund), chống hàng giả (BlockVerify), mạng xã hội (Steemit),...

2.2. Các hướng nghiên cứu của công nghệ Blockchain trong và ngoài nước

Trên thế giới có rất nhiều cá nhân, tổ chức quan quan tâm tới công nghệ Blockchain.

Sau đây là rất nhiều ứng dụng đã ra đời phục vụ trên nhiều lĩnh vực khác nhau:

- **Bán lẻ**

Warranteer: Một ứng dụng Blockchain cho phép người tiêu dùng dễ dàng truy cập thông tin về sản phẩm họ đã mua và nhận hỗ trợ dịch vụ trong trường hợp có trục trặc sản phẩm.

Blockpoint: Đơn giản hóa việc tạo ra các hệ thống thanh toán và chấp nhận ví điện tử, chương trình khách hàng thân thiết, thẻ quà tặng và các chức năng khác.

- **Xe sang**

Bitcar: Phân chia quyền sở hữu của các xe sưu tầm bằng token của BitCar.

- **Chuỗi cung ứng và logistics**

IBM Blockchain: Biết rõ tình trạng và điều kiện mỗi sản phẩm trong chuỗi cung ứng, từ vật liệu thô tới phân phối, là điều rất quan trọng. Ứng dụng Blockchain trong chuỗi cung ứng cho phép minh bạch hóa bằng một hồ sơ chia sẻ quyền sở hữu và vị trí của các phần, các sản phẩm trong thời gian thực.

Food industry: Mạng lưới phức tạp của ngành công nghiệp thực phẩm, từ nông dân tới các nhà bán lẻ, khiến cho việc theo dõi các căn bệnh do thực phẩm gây ra khá khó khăn. Blockchain có thể cải thiện tính minh bạch và hiệu quả trong việc tìm ra những loại thực phẩm có thể bị ô nhiễm và ở đâu trong suốt chuỗi cung ứng.

Provenance: Người tiêu dùng ngày càng đòi hỏi sự minh bạch về các sản phẩm họ mua và tiêu dùng để đảm bảo nguồn gốc nguyên liệu và sản xuất sản phẩm tôn trọng các giá trị cá nhân của họ. Pronance sử dụng Blockchain để cung cấp quá trình chăm sóc và chứng nhận của chuỗi cung ứng.

Blockverify: Với tuyên bố "đưa tính minh bạch vào chuỗi cung ứng", Blockverify tập trung vào các giải pháp chống hàng giả bằng cách sử dụng Blockchain để xác định các sản phẩm giả mạo, hàng hóa bị đánh cắp và các giao dịch gian lận.

- **Bảo hiểm**

Accenture: Với mục tiêu thúc đẩy tính hiệu quả và hiệu suất trong ngành công nghiệp bảo hiểm, Accenture xây dựng các giải pháp về Blockchain cho các khách hàng bảo hiểm. Họ chuyển các quy trình chính trong ngành bảo hiểm sang các thủ tục có ứng dụng Blockchain nhằm gia tăng độ tin cậy hơn cho hệ thống.

Proof of insurance: Công ty bảo hiểm này đang thử nghiệm một giải pháp Blockchain để cung cấp thông tin chứng minh về bảo hiểm được gọi là RiskBlock. Cuối cùng, khi công cụ này được triển khai đầy đủ, nó sẽ giúp các cơ quan thực thi pháp luật, người mua bảo hiểm và công ty bảo hiểm xác minh mức độ bảo hiểm trong thời gian thực và tăng tốc quá trình xử lý yêu cầu.

- **Chăm sóc sức khỏe**

MedicalChain: Công ty trong lĩnh vực chăm sóc sức khỏe đầu tiên sử dụng công nghệ Blockchain để tạo thuận lợi trong việc lưu trữ và sử dụng hồ sơ y tế điện tử để cung cấp trải nghiệm y học từ xa (telemedicine) hoàn chỉnh. Họ là các bác sĩ thực tế trong hệ thống chăm sóc sức khỏe của Anh và muốn thay đổi hệ thống này từ bên trong.

MedRec: Để cung cấp cho bất kỳ nhà cung cấp dịch vụ y tế truy cập an toàn vào hồ sơ của bệnh nhân, MedRec sử dụng Blockchain để tiết kiệm thời gian, tiền bạc và các quy trình lặp lại trong việc tiến hành thủ tục giữa các cơ sở và nhà cung cấp khác nhau. Bệnh nhân cũng có thể truy cập vào hồ sơ y tế của họ để nghiên cứu.

- **Bất động sản**

BitProperty: Sử dụng Blockchain và hợp đồng thông minh, BitProperty cho phép bất cứ ai ở bất cứ nơi nào trên thế giới (trừ Mỹ và Nhật Bản, do các vấn đề về pháp lý) đều có thể đầu tư vào bất động sản.

- **Từ thiện**

BitGive: Nền tảng gây quỹ toàn cầu này ứng dụng Bitcoin và công nghệ Blockchain để đem lại sự minh bạch hơn cho các nhà tài trợ bằng cách chia sẻ thông tin tài chính và dự án theo thời gian thực. Save the Children, The Water Project và Medic Mobile là một trong số những tổ chức từ thiện làm việc với BitGive.

- **Dịch vụ tài chính**

Bitcoin Atom: Một nhánh mới của Bitcoin cho phép trao đổi tiền mã hóa dễ dàng mà không tốn phí giao dịch và không bị tấn công giao dịch, khiến Bitcoin thực sự được phân cấp lại. Công nghệ này dựa trên các hoán đổi nguyên tử (atomic swaps) - được xem là một công cụ vô giá để trao đổi các đồng tiền mã hóa và không cần phải có một bên thứ ba đáng tin cậy. Nhưng hiện tại, việc áp dụng rộng rãi các giao dịch hoán đổi nguyên tử đã bị ngăn chặn vì chúng đòi hỏi phải có kỹ năng kỹ thuật cao; Bitcoin Atom có thể giải quyết vấn đề này một phần nào đó.

2.3. Vấn đề mà đề tài tập trung, nghiên cứu giải quyết

2.3.1. Các loại Blockchain

Blockchain là một cơ sở dữ liệu phân cấp lưu trữ thông tin trong các khối thông tin được liên kết với nhau bằng mã hóa và mở rộng theo thời gian. Mỗi khối thông tin đều chứa thông tin về thời gian khởi tạo và được liên kết tới khối trước đó, kèm một mã thời gian và dữ liệu giao dịch. Blockchain được thiết kế để chống lại việc thay đổi của dữ liệu: Một khi dữ liệu đã được mạng lưới chấp nhận thì sẽ không có cách nào thay đổi được nó.

Hệ thống Blockchain chia thành 3 loại chính:

- **Public:** Bất kỳ ai cũng có quyền đọc và ghi dữ liệu trên Blockchain. Quá trình xác thực giao dịch trên Blockchain này đòi hỏi phải có hàng nghìn hay hàng vạn nút tham gia. Do đó để tấn công vào hệ thống Blockchain này là điều bất khả thi vì chi phí khá cao. Ví dụ: Bitcoin, Ethereum... là các nền tảng ứng dụng Public Blockchain.
- **Private:** Người dùng chỉ được quyền đọc dữ liệu, không có quyền ghi vì điều này thuộc về bên tổ chức thứ ba tuyệt đối tin cậy. Tổ chức này có thể hoặc không cho phép người dùng đọc dữ liệu trong một số trường hợp. Bên thứ ba toàn quyền quyết định mọi thay đổi trên Blockchain. Vì đây là một Private Blockchain, cho nên thời gian xác nhận giao dịch khá nhanh vì chỉ cần một lượng nhỏ thiết bị tham gia xác thực giao dịch. Ví dụ: Ripple là một dạng Private Blockchain, hệ thống này cho phép 20% các nút là gian dối và chỉ cần 80% còn lại hoạt động ổn định là được.
- **Permissioned:** Hay còn gọi là Consortium, một dạng của Private nhưng bổ sung thêm một số tính năng nhất định, kết hợp giữa “niềm tin” khi tham gia vào Public và “niềm tin tuyệt đối” khi tham gia vào Private. Ví dụ: Các ngân hàng hay tổ chức tài chính liên doanh sẽ sử dụng Blockchain cho riêng mình.

2.3.2. Ưu điểm của Private Blockchain

Việc hạn chế quyền truy cập vào dữ liệu hoặc các chức năng nhất định như viết hoặc đọc dữ liệu thực sự là một trong những lợi ích chính của các Private Blockchain. Ngoài ra, các Blockchain này có một cơ chế đồng thuận khác so với Public Blockchain và phù hợp hơn với một tập đoàn của các tổ chức, chẳng hạn như ngành ngân hàng.

Sau đây là một vài ưu điểm của Private Blockchain:

- **Nhanh hơn**

Private Blockchain có thể xử lý giao dịch cao hơn nhiều trong một giây (TPS) so với Public Blockchain, vì sự tồn tại của một vài người tham gia được ủy quyền sẽ giảm số lượng đáng kể sự đồng thuận cho mạng. Điều này cho phép nhiều giao dịch được xử lý cho mỗi block, Private Blockchain có thể xử lý hàng ngàn hoặc thậm chí hàng trăm nghìn giao dịch mỗi giây (TPS), so với 7 TPS của Bitcoin.

- **Khả năng mở rộng**

Vì chỉ có một vài nút được ủy quyền và chịu trách nhiệm quản lý dữ liệu, mạng có thể hỗ trợ và xử lý các giao dịch cao hơn nhiều. Không giống như một hệ thống phi tập trung, việc đạt được sự đồng thuận có thể mất thời gian, quá trình ra quyết định trong một mạng riêng được tập trung hơn và do đó nhanh hơn nhiều. Hãy nghĩ đơn giản cần 100 giáo viên để đánh dấu bài kiểm tra của bạn sẽ mất thời gian hơn nhiều so với một giáo viên đánh dấu nó.

Từ những ưu điểm của Private Blockchain mà trong phạm vi nghiên cứu, chúng tôi tập trung tìm hiểu về Private Blockchain, từ đó triển khai ứng dụng trên hệ thống cung cấp dịch vụ xác thực chứng chỉ.

Chương 3

NGHIÊN CỨU LÝ THUYẾT

3.1. Cơ sở lý thuyết, lý luận và phương pháp nghiên cứu mà Private Blockchain đã áp dụng

Thông tin trong Blockchain không thể bị làm giả hoặc rất khó nhưng vẫn sẽ để lại dấu vết, mọi thay đổi cần phải nhận được sự đồng thuận của tất cả các nút tham gia trong hệ thống. Nó là một hệ thống không dễ dàng sụp đổ, vì ngay cả khi một phần mạng lưới tê liệt thì các nút khác vẫn sẽ tiếp tục hoạt động để bảo vệ thông tin.

Công nghệ Blockchain tương đồng với cơ sở dữ liệu, chỉ khác ở việc tương tác với cơ sở dữ liệu. Private Blockchain dựa trên một số cơ sở sau đây:

- **Chuỗi khối và dịch vụ chuỗi khối**

Một chuỗi khối giống như một nơi để lưu trữ dữ liệu bán công cộng trong một không gian chứa hộp block. Bất cứ ai cũng có thể xác nhận việc bạn nhập thông tin vào vì khối chứa có chữ ký của bạn, nhưng chỉ có người sở hữu mới có thể thay đổi được dữ liệu của khối đó vì chỉ có bạn cầm khóa bí mật cho dữ liệu đó.

Vì thế chuỗi khối hoạt động gần giống như một cơ sở dữ liệu, ngoại trừ một phần của thông tin được lưu trữ của nó là công khai.

Dữ liệu lưu trữ có thể là một giá trị hoặc một số dư tiền mã hóa. Một chuỗi khối hoạt động như một hệ thống lưu chuyển giá trị thay thế mà không một quyền lực tập trung hay bên thứ ba nào có thể chen vào nhờ quá trình mã hóa. Nó được dựa trên nền tảng mã hóa khóa công khai, nhìn công khai nhưng lại bị kiểm soát bởi khóa bí mật.

- **Cơ chế đồng thuận phân tán đồng đẳng**

Cơ chế này ngược lại với mô hình cổ điển về cơ chế đồng thuận tập trung, nghĩa là khi một cơ sở dữ liệu tập trung được dùng để quản lý việc xác thực giao dịch. Một sơ đồ phân tán đồng đẳng chuyển giao quyền lực và sự tin tưởng cho một mạng lưới phân tán đồng đẳng và cho phép các nút của mạng lưới đó liên tục lưu trữ các giao dịch trên một block công cộng, tạo nên một chain độc nhất là Blockchain. Mỗi khối kế tiếp chứa một mã hash của mã trước nó; vì thế, mã hóa tạo bởi hàm hash được sử dụng để bảo đảm tính xác thực của nguồn giao dịch và loại bỏ sự cần thiết phải có một trung gian tập trung. Sự kết hợp của mã hóa và công nghệ Blockchain lại đảm bảo rằng sẽ không bao giờ một giao dịch được lưu trữ lại hai lần.

- **Hợp đồng thông minh và tài sản thông minh**

Hợp đồng thông minh là các khối để xây dựng nên các ứng dụng phi tập trung. Một hợp đồng thông minh tương đương với một chương trình nhỏ mà bạn có thể tin tưởng với một đơn vị giá trị và quản lý giá trị đó. Ý tưởng cơ bản đằng sau hợp đồng thông minh là sự quản lý bằng khế ước đối với một giao dịch giữa hai bên liên quan hay nhiều hơn có thể được xác minh theo thứ tự thông qua chuỗi khối, thay vì thông qua một quan tòa tập trung. Không cần phải dựa vào một quyền lực tập trung trong khi hai hay nhiều bên tham gia có thể đồng thuận lẫn nhau, có thể tự đưa ra các điều khoản và thực thi sự đồng thuận bằng chương trình và các điều kiện quy ước sẵn, tiền sẽ được chuyển tự động khi hoàn thành một số yêu cầu cần thiết.

- **Tính toán tin cậy**

Khi bạn kết hợp các nền tảng đằng sau chuỗi khối, cơ chế đồng thuận phi tập trung và hợp đồng thông minh, nhận ra rằng chúng cho phép các máy tính tin tưởng lẫn nhau ở một mức độ sâu.

Vì vai trò của chuỗi khối là người xác nhận giao dịch minh bạch, mỗi khối ngang hàng có thể tiếp tục tin tưởng lẫn nhau đang tuân theo các quy luật đã quy ước.

- **Bằng chứng công việc**

Khái niệm then chốt của hoạt động chuỗi khối là “bằng chứng công việc”. Nó được biểu hiện là một rào cản lớn ngăn cản người dùng thay đổi dữ liệu trên chuỗi khối mà không sửa lại bằng chứng công việc.

Bằng chứng công việc là khối then chốt xây dựng nên Blockchain vì nó không thể “sửa lại” và được bảo vệ thông qua sức mạnh của hàm hash mã hóa.

Mỗi khối lưu trữ thông tin giao dịch đều có liên kết với các khối liền trước nên toàn bộ chuỗi khối sẽ chứa đựng thông tin liên mạch xuyên suốt lịch sử giao dịch của chuỗi đó. Một khi đã được ghi lại trên chuỗi khối, thông tin giao dịch sẽ không thể bị xóa bỏ hay thay đổi. Blockchain đó cũng sẽ liên tục được cập nhật sao cho thông tin trên sổ cái của từng người trong mạng lưới đều giống hệt nhau nên máy tính nào cũng có thể xác nhận ai có số dư tiền mã hóa hay giá trị là bao nhiêu ở thời điểm bất kỳ.

Trên góc độ kinh doanh có thể gọi Blockchain là một sổ cái kế toán, hay một cơ sở dữ liệu chứa đựng tài sản, hay một cấu trúc dữ liệu, mà dùng để ghi chép lại lịch sử tài sản giữa các thành viên trong hệ thống mạng ngang hàng.

Trên góc độ kỹ thuật đó là một phương thức bất biến để lưu trữ lịch sử các giao dịch tài sản.

Trên góc độ xã hội đó là một hiện tượng, mà dùng để thiết lập niềm tin bằng quy tắc đồng thuận giữa các thành viên trong một hệ thống phân cấp.

Công nghệ Blockchain có thể nói là sự kết hợp 3 loại công nghệ bên dưới:

- **Mật mã học**

Sử dụng public key và hàm hash để đảm bảo tính minh bạch, toàn vẹn và riêng tư.

- **Mạng ngang hàng**

Mỗi một nút trong mạng được xem như một client và cũng là server để lưu trữ bản sao ứng dụng.

- **Lý thuyết trò chơi**

Tất cả các nút tham gia vào hệ thống đều phải tuân thủ luật chơi đồng thuận và được thúc đẩy bởi động lực kinh tế.

3.2. Giới thiệu hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung

3.2.1. Giới thiệu MultiChain

MultiChain được phát triển bởi công ty Coin Sciences [\[3\]](#). MultiChain được xây dựng với mục đích giúp các tổ chức có thể xây dựng và triển khai các ứng dụng Blockchain với tốc độ nhanh chóng dựa trên nền tảng Blockchain.

Ưu điểm lớn nhất của MultiChain là khả năng tạo ra mạng lưới Private Blockchain một cách đơn giản và nhanh chóng nhưng vẫn giữ được các tính chất của một Blockchain. Thứ hai là khả năng quản lý các tài sản (assets), trên Blockchain tự tạo ra, người dùng có thể tạo ra vô số loại assets có thể là tiền tệ, và thực hiện hoặc theo dõi các giao dịch nhiều assets, nhiều bên tham gia một cách dễ dàng. Khả năng mới khác với các hệ thống Blockchain khác của MultiChain là có thể tạo dữ liệu streams, thông qua các streams này, các node tham gia đăng kí vào stream có thể chia sẻ và theo dõi dữ liệu, cấp quyền cho các node khác trên stream mình tạo tham gia đọc, ghi dữ liệu và stream. Ưu điểm khác nữa của Multichain là cấp quyền dễ dàng và linh động. Node tạo Blockchain kiểm soát ai có thể kết nối, gửi và nhận giao dịch, tạo assets, stream và block theo mong muốn của node đó.

So với các hệ thống Blockchain trên thị trường hiện nay như Ethereum, Bitcoin, Steem ... tập trung phát triển để quản lý tiền tệ và giao dịch, MultiChain ra đời nhằm tận dụng khả năng đồng bộ hóa của hệ thống Blockchain để chia sẻ thông tin và quản lý các tài nguyên. Mã nguồn mở và giao diện khá đơn giản so với các hệ thống còn lại là điểm lợi cho các nhà phát triển có thể linh hoạt tạo ra hệ thống của riêng mình. Tuy nhiên với việc linh hoạt trong cấp quyền, và khả năng tạo vô hạn loại assets thì yêu cầu Node quản lý hay node có quyền hạn cao nhất tốn khá nhiều công sức để thực hiện phân phát quyền và quản lý các node con sau này. Việc bổ sung tính năng stream để chia sẻ data cũng khiến số lượng block tăng lên và tiêu tốn khá nhiều bộ nhớ.

3.2.1.1. Cơ chế đồng thuận cấp quyền trong MultiChain

MultiChain được xây dựng dựa trên kiến trúc Private Blockchain nên khi các Node tham gia vào mạng lưới tức vào một chain nào đó thì chúng sẽ được cấp quyền tùy theo từng hệ thống mà được cấp những quyền khác nhau.

Có 8 loại quyền cơ bản trong MultiChain:

- **Connect:** cho phép kết nối với các nodes khác và thấy được nội dung Blockchain
- **Send:** cho phép gửi tiền dựa trên quy ước tiền tệ trước đó
- **Receive:** cho phép nhận tiền
- **Issue:** cho phép phát hành loại tiền tệ riêng
- **Create:** cho phép tạo streams mới
- **Mine:** cho phép tạo blocks
- **Activate:** cho phép thay đổi quyền hạn, cấp phép quyền của các node khác
- **Admin:** có tất cả các quyền trên

Tất cả các quyền trên được cấp phát dựa trên một địa chỉ address của một node. Trong trường hợp node có nhiều địa chỉ address thì các địa chỉ address đó có

thể có các quyền không giống nhau. Mỗi địa chỉ address có thể là một pubkey hashes hay script hashes.

Một node được cấp quyền connect thông qua cơ chế peer-to-peer handshaking giữa hai node (node đã tham gia vào chain và node mới). Sau khi peer-to-peer handshaking thành công node mới được cấp một private key và địa chỉ address tương ứng. Nếu node bị tước quyền connect bởi node cấp quyền connect thì lập tức node sẽ bị mất liên kết với các nodes khác mà node sử dụng địa chỉ address được cấp để kết nối.

3.2.1.2. Cơ chế phát hành tiền tệ trên MultiChain

MultiChain cho phép người dùng có thể phát hành vô số loại tiền tệ và số tiền cũng như thực hiện các giao dịch trên chain.

Một loại tiền tệ khi được phát hành nó có thể được chọn là tiền mở hoặc đóng, nếu là mở thì các node khác đều có thể phát hành số tiền và giao dịch chúng, các node muốn phát hành một khoản tiền nào đó sẽ phải được cấp quyền trước để thực hiện việc này.

Tuy nhiên, MultiChain không qui định việc chuyển đổi tiền tệ lẫn nhau, nên tùy thuộc vào hệ thống, nhà phát triển có thể tự đặt qui định theo yêu cầu.

3.2.1.3. Bảo mật trên luồng trong MultiChain

Trong MultiChain, các node có thể chia sẻ dữ liệu an toàn và toàn vẹn trên chain đã tham gia với nhau. Streams giống như một căn phòng, khi các node tham gia vào căn phòng ký dựa trên địa chỉ address của mình. Các data được đăng lên streams được lưu như một giao dịch tiền tệ, nên không thể thay đổi và đảm bảo được sự toàn vẹn.

Stream là một chức năng của multichain cho ta dễ hình dung các trường hợp sử dụng Blockchain trong việc truy suất, lưu trữ data hay timestamping thay vì hiểu dưới dạng chuyển đổi tài khoản, tiền ...

Stream có thể triển khai trên 3 loại dữ liệu:

- Cơ sở dữ liệu khóa-giá trị (Key-value database) hay tài liệu lưu trữ theo kiểu NoSQL
- Cơ sở dữ liệu chuỗi thời gian (A time series database)
- Cơ sở dữ liệu dựa trên danh tính (An identity-driven database)

Stream có thể được khởi tạo với con số bất kỳ trong Multichain, như một collection của các item chỉ được bổ sung thêm mà không được xóa mất.

Các đặc điểm của item trong stream:

- Một hoặc nhiều người tạo ra đăng ký lên đó
- Có thể có key thuận tiện cho tìm kiếm item sau này
- Data có thể là đoạn văn bản nhỏ hoặc hoặc tệp nhị phân vài megabytes
- Timestamp được lấy trong header của block mà item được xác thực

Về cơ bản mỗi item trong stream là một Blockchain transaction, nhưng các nhà phát triển có thể đọc hay viết mà không cần sợ các quy tắc ngầm nào.

Stream có thể được sử dụng chung với cơ chế cấp phép của Multichain ví dụ như:

- Có thể được tạo bởi ai có quyền
- Sau khi được tạo nó đóng hay cho mở
 - Stream mở có thể được viết bởi những người có quyền send transaction
 - Stream đóng được hạn chế chỉ trong một danh sách các địa chỉ được cấp phép sẵn
- Stream có thể có một hoặc nhiều admin để thay đổi các quyền này

Mỗi Blockchain có một root stream, được tạo từ đầu khi chain được tạo, có thể được sử dụng ngay lập tức mà không cần chờ stream nào được tạo.

Stream hỗ trợ việc mã hóa dữ liệu vào Blockchain, như:

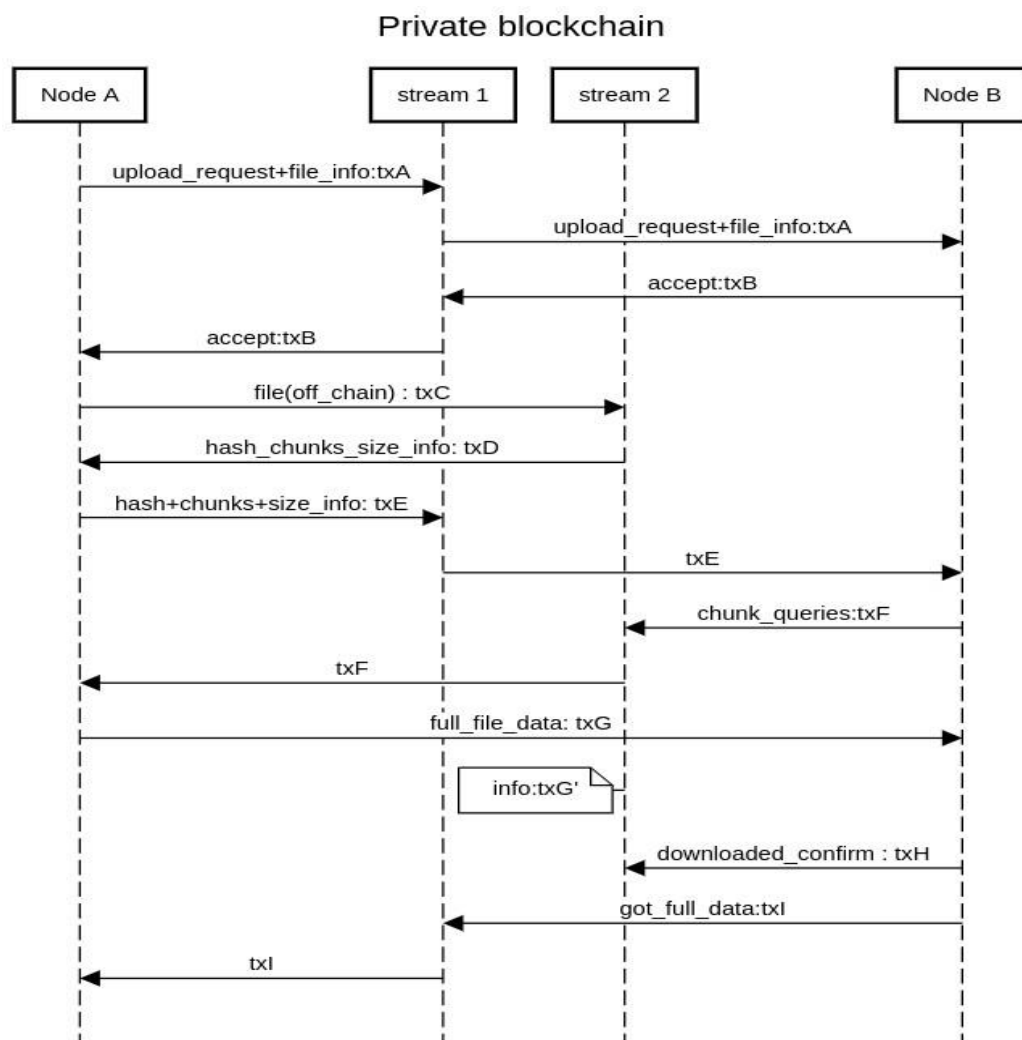
- Stream 1 cho mọi người đăng public key của họ

- Stream 2 cho đăng các data được mã hóa đối xứng với một khóa duy nhất
- Stream 3 để cấp quyền truy cập các data được mã hóa, bằng cách mã hóa khóa duy nhất của stream 2 với khóa public key của người được chọn và đăng lên stream 3.

Cơ chế chuyển, lưu file on-chain và off-chain:

- Cơ chế on-chain thông thường các node của chain đều giữ một bản sao đầy đủ dữ liệu, việc này gây ra vấn đề các node sẽ bị ngăn chặn khi chuyển lưu dữ liệu quá lớn. Cơ chế off-chain giải quyết được vấn đề khi số lượng dữ liệu quá lớn và ngăn node khác có đầy đủ dữ liệu trên Blockchain.
- Quá trình hoạt động của off-chain:
 - Bước 1: Node publisher lưu dữ liệu vào bộ nhớ cục bộ, cắt dữ liệu thành các chunk nhỏ để dễ lưu trữ và phân phối
 - Bước 2: Giao dịch (Transaction) dùng để công khai các phần của off-chain stream được tạo, trong đó bao gồm các mã hash của các chunk và kích thước
 - Bước 3: Transaction được ký và phát lên mạng lưới và lưu vào Blockchain của các node
 - Bước 4: Khi một node đăng ký vào một stream sẽ thấy được một số thông tin cơ bản của dữ liệu này, node sẽ dùng thông tin kết hợp với mã hash của chunk để truy hồi dữ liệu theo hàng đợi truy xuất
 - Bước 5: Sau đó các lệnh truy vấn sẽ được gửi lên mạng lưới để tìm vị trí lưu của các chunk này dựa trên mã hash của chúng
 - Bước 6: Bất kỳ node nào có chunk dữ liệu cũng có thể phản hồi lại theo đường dẫn y hệt như đường dẫn của lệnh truy vấn.
 - Bước 7: Nếu không có phản hồi cho một chunk truy vấn nào đó, thì sẽ được đưa lại vào hàng đợi để thử lại sau
 - Bước 8: Source node gửi dữ liệu được yêu cầu theo đường dẫn đã truy vấn.

- Bước 9: Node đăng ký sẽ kiểm tra lại dữ liệu như site và mã hash có giống thông tin ban đầu, nếu đúng sẽ lưu ngay vào bộ nhớ cục bộ của chúng
- Off-chain có các nhược điểm :
 - Cần có map của các address IP trong Blockchain để liên kết trực tiếp với publisher
 - Khi publisher node tắt khỏi mạng lưới, hoặc tạm thời ngưng dịch vụ thì dữ liệu sẽ không thể được truy hồi
 - Nếu nhiều node muốn có dữ liệu và node publish quá tải sẽ gây tắc nghẽn và khó truy cập được dữ liệu.



Hình 3. 1: Cách lưu file bằng cơ chế off-chain trên Multichain

Trong *Hình 3.1*:

- Node A là node user muốn upload một file lên Node B server cung cấp dịch vụ lưu trữ
- Stream 1: Là stream lưu các transaction yêu cầu dịch vụ
- Stream 2: Lưu các transaction chứa thông tin thực hiện các yêu cầu
- Sau quá trình trên stream 1 sẽ lưu trữ:
 - txA: Yêu cầu upload file của user và thông tin file
 - txB: Bản chấp nhận yêu cầu của server B
 - txE: Thông tin cụ thể hash chunks của file mà user muốn upload lên B
 - txI: Thông tin xác nhận từ B đã có được file
- Sau quá trình trên stream 2 sẽ lưu trữ:
 - txC: mã hash của file theo chunk (do cơ chế off chain) từ user A
 - txG: info của việc user A chuyển full data cho B (time,size...)
 - txH: bản xác nhận việc nhận file của B có thành công hay không.

Multichain protocol:

- Khi các transaction đã được đóng lại trong block và được ký bởi người tạo ra block, transaction sẽ được đưa lên network và tất cả node tham gia đều có khả năng xác thực transaction này.
- Multichain lưu data ta bằng sqlite DB, và hỗ trợ công cụ multichain-explorer như là một ứng dụng để hiển thị và xác thực các transactions giống như etherscan.io

Đây là bản năng suất của Multichain qua các phiên bản và dựa trên tổng số transaction trên giây. Bao gồm các thao tác trên tầng API và cả việc xây dựng, ký, đào, xác nhận các transaction và block.

| Total transactions | 1.0 alpha 3 | 1.0 alpha 21 | 1.0 alpha 22 | 1.0 beta 1 | 1.0 beta 2 |
|--------------------|-------------|--------------|--------------|------------|------------|
| 100 | 6.5 tps | 7.8 | 541.7 | 830.6 | 1465.7 |
| 1,000 | 7.0 | 7.6 | 583.9 | 889.4 | 1199.6 |
| 10,000 | 4.1 | 6.4 | 566.9 | 746.6 | 1071.2 |
| 100,000 | — | 6.6 | 558.0 | 771.9 | 1034.2 |
| 1,000,000 | — | — | 548.6 | 773.6 | 1055.4 |

Average transactions per second, including API overhead and building, signing, mining and verifying transactions and blocks.
 Tests performed using the [ab](#) HTTP server benchmarking tool sending two concurrent requests to the [sendtoaddress](#) API.
 Server specifications: Intel Core i7-4770, 4 cores @ 3.4 MHz, 32 GB RAM, Seagate 2 TB 7200 RPM SATA, CentOS 6.4.

Hình 3. 2: Bảng năng suất của Multichain qua từng phiên bản [\[16\]](#)

Một số thông tin về Multichain phiên bản 2.0:

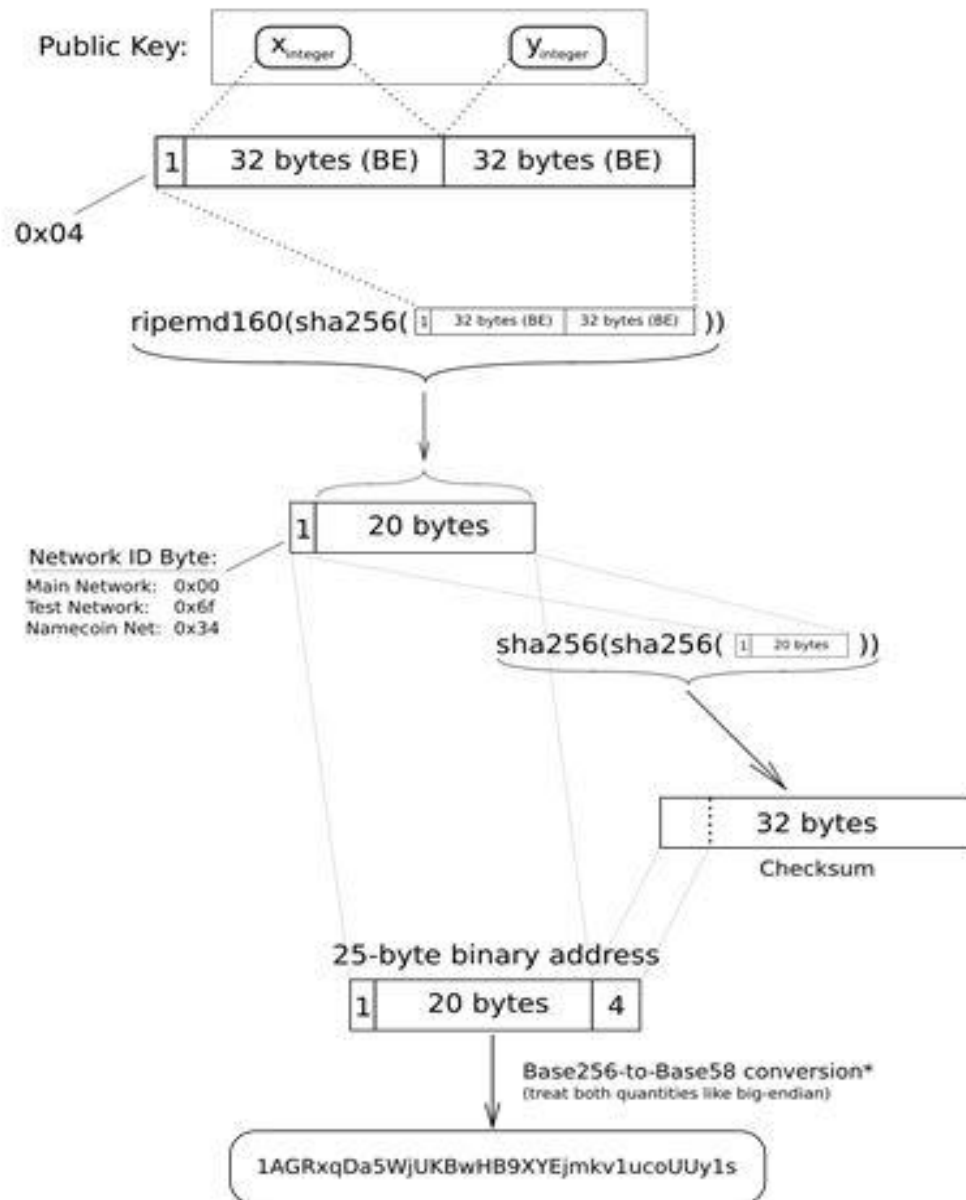
- Số lượng byte tối đa trong mỗi block: 8388608
- Kích thước byte tối đa mỗi chunk: 1048576
- Số lượng chunk tối đa mỗi off-chain item: 1024
- Kích thước byte tối đa một transaction chuẩn: 4194304

3.2.1.4. Định dạng address và key

Định dạng address, private keys của MultiChain cũng giống như Bitcoin. Nhưng có một vài điểm khác trong MultiChain để đảm bảo các địa chỉ khác nhau trên từng chain.

Address trên Bitcoin sử dụng 1 byte tức là định nghĩa được 256 không gian địa chỉ khác nhau.

Elliptic-Curve Public Key to BTC Address conversion



*In a standard base conversion, the 0x00 byte on the left would be irrelevant (like writing '052' instead of just '52'), but in the BTC network the left-most zero chars are carried through the conversion. So for every 0x00 byte on the left end of the binary address, we will attach one '1' character to the Base58 address. This is why main-network addresses all start with '1'.

Hình 3. 3: Cách tạo địa chỉ trên Bitcoin từ public key [\[17\]](#)

Ở MultiChain sử dụng 4 bytes (gọi là version bytes) để lưu địa chỉ và 4 bytes giá trị checksum (gọi là value bytes). Tức có thể lưu khoảng 2^{64} không gian địa chỉ riêng biệt.

Khi một chain mới được tạo 4 bytes giá trị checksum được tạo ngẫu nhiên thông qua các thông số address-pubkeyhash-version và address-checksum-value được định nghĩa trước đó.

Ngoài ra MultiChain còn cho phép sử dụng các giá trị bytes này thông qua các thông số address-scripthash-version và private-key-version. Ví dụ để thay đổi format [\[13\]](#) địa chỉ MultiChain giống hệt như Bitcoin thì định nghĩa các thông số như sau:

- address-pubkeyhash-version=00
- address-scripthash-version=05
- private-key-version=80
- address-checksum-value=00000000

Để tạo một MultiChain address:

- B1. Tạo private ECDSA [\[18\]](#) key
- B2. Sử dụng public key tương ứng, public key này có hai dạng là nén và không nén. Dạng không nén 65 bytes và dạng nén 33 bytes.
- B3. Tính SHA-256 hash của public key.
- B4. Tính RIPEMD-160 hash của SHA-256 hash
- B5. Thêm bytes đầu của version bytes có được từ thông số address-pubkeyhash-version vào đầu RIPEMD-160 hash trước đó.

Nếu version bytes > 1 bytes. Các bytes sau thêm theo dạng cứ mỗi $\text{floor}(20/\text{len}(\text{address-pubkeyhash-version}))$ bytes thì thêm vào trước giá trị đó.

- B6. Tính giá trị SHA-256 hash của giá trị có được trước đó
- B7. Tính giá trị SHA-256 hash của giá trị có được trước đó
- B8. Lấy 4 bytes đầu của SHA-256 hash trước làm checksum

- B9. XOR 4 bytes checksum với address-checksum-value
- B10. Thêm kết quả XOR vào cuối kết quả hash có được ở B5
- B11. Chuyển qua string bằng Bitcoin base58 encoding

Để tạo một MultiChain private key:

- B1. Lấy private ECDSA key
- B2. Thêm giá trị 0x01 vào cuối private key
- B3. Thêm bytes đầu của version bytes có được từ thông số private-key-version vào đầu giá trị B2 trước đó.

Nếu version bytes > 1 bytes. Các bytes sau thêm theo dạng cứ mỗi $\text{floor}(33/\text{len}(\text{private-key-version}))$ bytes thì thêm vào trước giá trị đó.

- B4. Tính SHA-256 hash của B3
- B5. Tính SHA-256 hash của B4
- B6. Lấy 4 bytes đầu của B5 làm checksum
- B7. XOR checksum với thông số address-checksum-value
- B8. Thêm kết quả vào cuối giá trị của B3
- B9. Chuyển sang string bằng Bitcoin base58 encoding

3.2.2. Giới thiệu storj

Storj [\[1\]\[5\]](#) là một nền tảng lưu trữ phi tập trung không thể kiểm duyệt hoặc giám sát và có thời gian chết. Đây là kho lưu trữ mật mã đầu cuối được phân cấp đầu tiên và cuối cùng sử dụng công nghệ Blockchain và mật khẩu để bảo vệ tệp của bạn.

Hệ thống này sẽ lưu trữ thông tin về mức độ nguyên vẹn của tập tin và vị trí lưu trữ. Các tệp của bạn được mã hóa, chia thành các phần nhỏ gọi là ‘mảnh’. Các mảnh này sẽ được lưu trữ trong một mạng lưới phân tán của các máy tính trên toàn cầu. Không ai ngoài bạn có một bản sao đầy đủ của tập tin của bạn, thậm chí không ở dạng được mã hoá.

Storj có thể nhanh hơn, rẻ hơn và an toàn hơn các nền tảng lưu trữ đám mây truyền thống. Nhanh hơn bởi vì nhiều máy đang đồng thời phục vụ tệp của bạn. Rẻ hơn vì bạn đang thuê không gian ổ cứng trống của người dùng thay vì phải trả tiền cho một trung tâm dữ liệu có mục đích. Và an toàn hơn vì tệp của bạn được mã hoá và chia nhỏ.

Mạng Storj sẽ tự động phát hành, thu thập và phân phối những token này để giúp người dùng có thể thanh toán cho không gian lưu trữ cũng như được cung cấp không gian lưu trữ mới. Tất cả dữ liệu trước khi được tải lên mạng đều sẽ được mã hoá và ưu tiên client-side. Storj đặt mình vào vị trí là một cơ chế lưu trữ dữ liệu cho người dùng. Người dùng cuối cùng luôn có quyền kiểm soát đối với dữ liệu và quyền riêng tư của họ.

3.2.2.1. Một số tính chất của Storj

Nodes lưu trữ

Nhiệm vụ chính của nút lưu trữ là lưu trữ và trả lại dữ liệu một cách đáng tin cậy. Người vận hành node là những cá nhân hoặc tổ chức có dung lượng ổ cứng dư thừa và muốn kiếm thu nhập bằng cách thuê không gian của họ cho người khác. Các nhà khai thác này sẽ tải xuống, cài đặt và định cấu hình phần mềm Storj cục bộ, không cần tài khoản ở bất cứ đâu. Sau đó, họ sẽ định cấu hình không gian đĩa và phụ cấp băng thông cho mỗi vệ tinh. Trong quá trình phát hiện nút, các nút lưu trữ sẽ thông báo có bao nhiêu băng thông và dung lượng ổ cứng khả dụng và địa chỉ ví.

Các nút lưu trữ được chọn để lưu trữ dữ liệu dựa trên các tiêu chí khác nhau: thời gian ping, độ trễ, thông lượng, giới hạn băng thông, đủ dung lượng đĩa, vị trí địa lý, thời gian hoạt động, lịch sử phản hồi chính xác...

Các nút lưu trữ chỉ được thanh toán phí khi được yêu cầu trả lại dữ liệu dưới băng thông hoặc là việc lưu trữ dữ liệu trên node đó trong một khoảng thời gian nhất định.

Hệ thống giả định rằng các node đang lưu trữ trung thực tất cả dữ liệu, các nút lưu trữ thất bại trong việc kiểm tra ngẫu nhiên về tính trung thực sẽ bị loại khỏi mạng lưới, có thể lấy tiền được giữ trong ký quỹ tức tiền cọc để trả chi phí bổ sung và sẽ bị giới hạn không có chi phí nào trong tương lai

Các nút lưu trữ không được trả phí cho việc chuyển dữ liệu ban đầu để lưu trữ (bằng thông đi vào node), để tránh trường hợp node lưu trữ xóa dữ liệu mà chỉ tính phí cho việc lưu trữ dữ liệu.

Các nút lưu trữ sẽ hỗ trợ ba phương thức: truy xuất, lưu trữ và xóa. Mỗi phương thức sẽ lấy một ID mảnh, ID vệ tinh, chữ ký từ đối tượng. Vệ tinh được liên kết và mức băng thông. ID vệ tinh tạo thành một không gian tên. ID mảnh giống hệt nhau với ID vệ tinh khác nhau để cập đến một mảnh khác.

Các nút lưu trữ sẽ cho phép quản trị viên cấu hình không gian đĩa tối đa được phép và sử dụng băng thông trên mỗi vệ tinh trong 30 ngày qua. Họ sẽ theo dõi số tiền còn lại của cả hai và từ chối các hoạt động không có chữ ký hợp lệ từ vệ tinh phù hợp.

Trong quá trình thiết lập, các nút lưu trữ, vệ tinh và Uplinks đều tạo ra danh tính và chứng chỉ riêng để sử dụng trong mạng. ID nút này được sử dụng để khám phá và định tuyến nút.

Cấu trúc của tập tin lưu trữ

Một tập tin được tách thành các mảnh dữ liệu. Nếu mảnh dữ liệu đó nhỏ hơn siêu dữ liệu cần thiết để lưu trữ trên mạng, dữ liệu sẽ được lưu trữ nội tuyến với siêu dữ liệu. Mảnh dữ liệu này sẽ là mảnh dữ liệu nội tuyến. Đối với các tập tin lớn hơn, dữ liệu sẽ được chia thành một hoặc nhiều mảnh dữ liệu. Phân bố theo cách này có nhiều lợi thế về bảo mật, quyền riêng tư, hiệu suất và tính sẵn sàng. Cuối cùng, giới hạn kích thước của các phân đoạn cho phép lấp đầy nút lưu trữ đồng đều hơn. Do đó, một nút chỉ cần đủ dung lượng để lưu trữ một phân đoạn để tham gia vào mạng và một khách hàng không cần phải tìm các nút có đủ không gian cho một tập tin có dung lượng lớn.

Tiếp theo, các mảnh dữ liệu sẽ được mã hoá và tách thành các đoạn nhỏ hơn. Các đoạn đó sẽ tiếp tục được phân tích thành các sọc dữ liệu. Mỗi đoạn sẽ tạo ra khoảng 40 sọc và chỉ cần 20/40 sọc là đủ để tái cấu trúc lại đoạn dữ liệu đó. Mỗi sọc sẽ có kích thước bằng 1/20 so với kích thước của đoạn dữ liệu.

Chủ sở hữu dữ liệu sẽ cần thông tin về các mảnh dữ liệu từ xa và vị trí của các mảnh dữ liệu trong mạng để khôi phục nó. Điều này được chứa trong cấu trúc dữ liệu con trỏ. Một con trỏ bao gồm: các nút nào đang lưu trữ các mảnh dữ liệu, thông tin mã hóa, chi tiết của các sọc dữ liệu, ngưỡng kích hoạt sửa chữa dữ liệu khi mức độ dư thừa của dữ liệu nằm dưới ngưỡng đó, và các chi tiết khác. Nếu mảnh dữ liệu là một mảnh dữ liệu nội tuyến, con trỏ sẽ chứa toàn bộ dữ liệu nhị phân của mảnh.

Các mảnh tập tin

Khi bạn muốn lưu trữ một tệp trên Storj, trước tiên bạn sẽ phải chia tệp ra nhiều phần nhỏ hơn. Các mảnh tệp tin này có hai ưu điểm. Trước tiên, bạn có thể gửi và rút lại các phần của tệp song song, làm cho việc truyền tệp nhanh hơn. Thứ hai, không một thực thể nào có thể duy trì toàn bộ tệp của bạn. Bạn là người duy nhất biết được vị trí của tất cả các mảnh.

Đây là nơi Blockchain và mật mã đi vào. Storj thực hiện những thứ được biết đến như một bảng băm phân phối để người dùng có thể xác định được vị trí tất cả các mảnh tệp tin ban đầu của họ. Bảng băm này yêu cầu một khoá riêng để phát hiện ra các mảnh vỡ. Nếu không có khoá riêng, gần như không thể đoán chính xác vị trí của một tệp bị mất.

Storj sử dụng một bảng băm phân phối gọi là Kademlia¹[\[7\]](#). Đây là một trong những phần cốt lõi của kiến trúc của Storj.

¹ Kademlia là một bảng băm phân tán cho các mạng máy tính ngang hàng phi tập trung được thiết kế bởi Petar Maymounkov và David Mazières vào năm 2002 [\[12\]](#)

Lưu trữ dữ liệu

Trước khi bắt đầu đưa tập tin lên mạng Storj để lưu trữ dữ liệu, bên máy người dùng sẽ tự động mã hoá dữ liệu bằng các thuật toán AES-GCM hoặc là SecretBox và đồng thời phân tách dữ liệu thành nhiều mảnh nhỏ và đưa lên tất cả các nút lưu trữ trên mạng Storj. Sau đó một siêu dữ liệu sẽ được tạo để chứa các thông tin của tập tin và các mảnh trên các nút lưu trữ, và dựa vào đó ta có thể truy xuất lại được tập tin này.

Truy xuất dữ liệu

Để lấy được tập tin, hệ thống sẽ tham chiếu các thông tin bên trong siêu dữ liệu của tập tin đó để xác định vị trí các mảnh được lưu trữ trước đó trên mạng Storj. Sau đó, dữ liệu gốc sẽ được lấy ra và ghép lại và giải mã thành tập tin hoàn chỉnh trên máy của khách hàng.

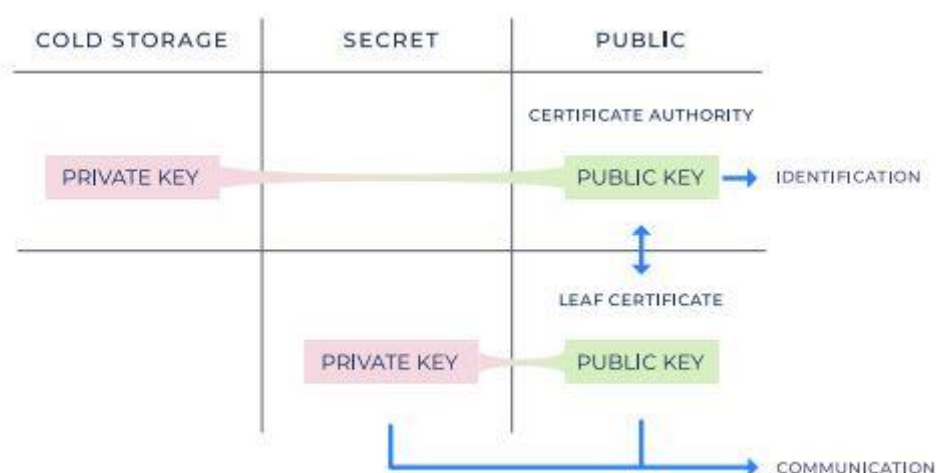
Duy trì dữ liệu

Khi lượng dư thừa của các mảnh dữ liệu giảm xuống dưới một ngưỡng nhất định, dữ liệu cần thiết cho các phần còn thiếu sẽ được tạo lại và thay thế.

Xác thực Nodes

Trong quá trình thiết lập, các nút lưu trữ, vệ tinh và Uplinks đều tạo ra danh tính và chứng chỉ riêng để sử dụng trong mạng. ID nút này được sử dụng để khám phá và định tuyến nút.

Mỗi nút sẽ vận hành cơ quan cấp chứng chỉ riêng, yêu cầu cặp khóa công khai, khóa riêng và chứng chỉ tự ký. Cơ quan cấp chứng chỉ, khóa riêng sẽ được giữ trong kho riêng để ngăn chặn sự thỏa hiệp của khóa. Điều quan trọng là khóa riêng của cơ quan cấp chứng chỉ được quản lý với bảo mật hoạt động tốt vì xoay vòng khóa cho cơ quan cấp chứng chỉ sẽ yêu cầu ID nút hoàn toàn mới.



Hình 3. 4: Mô hình xác thực và định danh nút lưu trữ [11]

Vòng đời

Vòng đời của các tập tin được lưu trữ bao gồm giá trị thời gian sống (TTL) là thời gian hết hạn lưu trữ của các tập tin lưu trên 1 node, nếu không có cấu hình giá trị này thì thời gian lưu trữ xem như là vô hạn. Nhưng thực tế thì có TTL trên mỗi node như là bộ nhớ lưu trữ trên node đó đã đầy và cần xoá đi những dữ liệu cũ, lúc đó thì tập tin có thể bị xoá.

TTL và giới hạn băng thông được lưu trữ trong cơ sở dữ liệu SQLite.

Thao tác đặt sẽ lấy một luồng byte và một TTL tùy chọn và lưu trữ các byte sao cho bất kỳ loại byte nào có thể được lấy lại thông qua thao tác get. Các hoạt động nhận được dự kiến sẽ hoạt động cho đến khi hết hạn (nếu được cung cấp TTL) hoặc cho đến khi nhận được thao tác xoá, tùy theo điều kiện nào đến trước.

Giao thức giao tiếp và tìm kiếm ngang hàng

- Cung cấp khả năng tiếp cận ngang hàng, ngay cả khi đối mặt với tường lửa và NAT.
- Cung cấp xác thực trong đó mỗi người tham gia chứng minh danh tính của người mà họ đang giao tiếp để tránh các cuộc tấn công trung gian.

- Cung cấp sự bảo mật hoàn toàn. Trong các trường hợp như đo bằng thông, nút máy khách và nút lưu trữ phải có khả năng giao tiếp mà không có bất kỳ rủi ro nào của kẻ nghe trộm. Giao thức phải đảm bảo rằng tất cả các thông tin liên lạc là riêng tư theo mặc định.

Siêu dữ liệu

Siêu dữ liệu sẽ chứa các thông tin về tập tin bao gồm vị trí của các nút lưu trữ mảnh dữ liệu, thông tin về các thuật toán sử dụng, các đường dẫn cụ thể.

Mỗi khi một đối tượng được thêm, chỉnh sửa hoặc xóa, một hoặc nhiều mục trong hệ thống lưu trữ siêu dữ liệu này sẽ cần được điều chỉnh. Kết quả là có thể có một sự thay đổi lớn trong hệ thống siêu dữ liệu này và trên toàn bộ cơ sở người dùng, chính siêu dữ liệu đó có thể trở nên khá lớn.

Kích thước đối tượng trung bình càng lớn, chi phí siêu dữ liệu càng ít.

Mã hoá

Vì mỗi tệp phải được mã hóa khác nhau bằng các khóa khác nhau và có khả năng thuật toán khác nhau, siêu dữ liệu về mã hóa đó phải được lưu trữ ở đâu đó theo cách an toàn và đáng tin cậy. Siêu dữ liệu này, cùng với các siêu dữ liệu khác về tệp, bao gồm cả đường dẫn của nó, sẽ được lưu trữ trong hệ thống lưu trữ siêu dữ liệu đã thảo luận trước đó, được mã hóa theo sơ đồ mã hóa phân cấp, xác định. Sơ đồ mã hóa phân cấp dựa trên BIP32 sẽ cho phép các cây con được chia sẻ mà không chia sẻ cha mẹ của chúng và sẽ cho phép một số tệp được chia sẻ mà không chia sẻ các tệp khác

Kiểm toán và danh tiếng

Trong hệ thống lưu trữ của chúng tôi, kiểm toán chi đơn giản là một cơ chế được sử dụng để xác định mức độ ổn định của nút. Kiểm toán thất bại sẽ dẫn đến một nút lưu trữ bị đánh dấu là xấu, điều này sẽ dẫn đến việc phân phối lại dữ liệu cho các nút mới và tránh hoàn toàn nút đó trong tương lai. Thời gian hoạt động của

nút lưu trữ và sức khỏe tổng thể là số liệu chính được sử dụng để xác định tệp nào cần sửa chữa.

Sửa chữa dữ liệu

Mặc dù có nhiều cách khác mà dữ liệu có thể bị mất, chẳng hạn như tham nhũng, hành vi độc hại, phần cứng xấu, lỗi phần mềm hoặc khai hoang không gian do người dùng khởi tạo, những vấn đề này ít nghiêm trọng hơn so với nút hoàn toàn.

Để sửa chữa dữ liệu, chúng tôi sẽ khôi phục dữ liệu gốc thông qua việc tái cấu trúc mã xóa từ các phần còn lại và sau đó tạo lại các phần còn thiếu và lưu trữ lại trong mạng trên các nút lưu trữ mới.

Điều quan trọng trong hệ thống của chúng tôi là khuyến khích những người tham gia làm nút lưu trữ duy trì trực tuyến lâu hơn một vài giờ. Để khuyến khích hành vi này, chiến lược thanh toán của chúng tôi sẽ liên quan đến việc thưởng cho các nhà khai thác nút lưu trữ giữ các nút của họ tham gia trong nhiều tháng và nhiều năm.

Thanh toán

Thanh toán trong các mạng phi tập trung là một phần quan trọng trong việc duy trì một hệ sinh thái lành mạnh của cả cung và cầu. Tất nhiên, các hệ thống thanh toán phi tập trung vẫn còn trong giai đoạn đầu theo một số cách. Để đạt được độ trễ thấp và thông lượng cao, không được phụ thuộc giao dịch vào một Blockchain. Mạng này giả định mã thông báo STORJ (STORJ token) dựa trên Ethereum làm cơ chế mặc định để thanh toán. Mặc dù chúng tôi dự định mã thông báo STORJ là hình thức thanh toán chính, trong tương lai, các loại thanh toán thay thế khác có thể được triển khai, bao gồm Bitcoin, Ether, thẻ tín dụng hoặc thẻ ghi nợ, chuyển khoản hoặc thậm chí chuyển giao trực tiếp.

3.2.2.2. Cơ chế hoạt động của Storj

Tải lên

Uplink sẽ đảm nhận việc thực hiện đăng tải dữ liệu lên mạng Storj

Uplink chọn khóa mã hóa và bắt đầu mã hóa dữ liệu truyền tải.

Uplink gửi yêu cầu tới Vệ tinh để chuẩn bị cho việc lưu trữ mảnh tập tin đầu tiên. Đối tượng yêu cầu chứa thông tin xác thực API, chẳng hạn như macaroons²[\[11\]](#) và chứng chỉ định danh.

Khi nhận được yêu cầu, Vệ tinh sẽ:

- Xác nhận rằng Uplink có đủ quyền và chi phí phù hợp cho yêu cầu. Uplink phải có tài khoản với Vệ tinh cụ thể này.
- Thực hiện lựa chọn các nút có đủ tài nguyên phù hợp với các yêu cầu về độ bền, hiệu suất, vị trí và độ tin cậy.
- Trả về danh sách các nút, cùng với thông tin liên hệ của các nút, bằng thông và ID gốc được chọn. Tiếp theo, Uplink sẽ lấy thông tin này và bắt đầu các kết nối song song với tất cả các nút lưu trữ được chọn trong khi đo băng thông.
- Uplink sẽ bắt đầu chia mảnh thành các đoạn và sau đó mã hóa từng đoạn. Các đoạn được tạo sẽ được ghép thành các mảnh khi chúng chuyển song song đến từng nút lưu trữ.
- Ở cấu hình mặc định, các mảnh sẽ được sao chép ra thành nhiều mảnh giống nhau và lưu ở các nút lưu trữ khác nhau. Tổng dung lượng của một tập tin đưa lên sẽ tăng lên 275% theo cấu hình mặc định. Khi bạn tải xuống tập tin đó, bạn chỉ tải xuống dung lượng gốc của tập tin dữ liệu, việc này sẽ giúp việc tải xuống nhanh hơn và có tính sẵn sàng cao hơn. Nếu bạn muốn tải xuống nhanh, bạn có thể tải xuống nhiều hơn dung lượng gốc của tập tin dữ liệu vì Uplink sẽ yêu cầu các mảnh từ nhiều nút hơn mức cần thiết, để đảm

² Macaroons là mã thông báo ủy quyền linh hoạt hoạt động tốt trong các hệ thống phân tán [\[15\]](#)

bảo bạn có được dữ liệu tại các nút lưu trữ nhanh nhất có thể. Cách tiếp cận này giảm thiểu độ trễ và giảm đáng kể thời gian chờ. Uplink được phép hủy tải lên các nút có tốc độ chậm nhất.

- Dữ liệu sẽ tiếp tục được truyền cho đến khi kích thước mảnh tối đa được xác định hoặc kết thúc luồng dữ liệu, tùy theo thời điểm nào sớm hơn.
- Tất cả các giá trị băm của mỗi đoạn sẽ được ghi vào cuối mỗi luồng. Sau đó, nút lưu trữ sẽ lưu trữ:
 - Bảng thông lớn nhất đã sử dụng.
 - Thời gian sống (TTL) của mảnh dữ liệu (có thể có hoặc không).
 - Dữ liệu của chính nó. Dữ liệu sẽ được xác định bởi ID mảnh dành riêng cho nút lưu trữ và ID vệ tinh ủy nhiệm.
- Nếu tải lên bị hủy vì bất kỳ lý do gì, nút lưu trữ sẽ lưu giá trị bảng thông lớn nhất mà nó nhận được từ Uplink của máy người dùng thay mặt cho Vệ tinh, nhưng sẽ loại bỏ tất cả dữ liệu yêu cầu có liên quan khác. Giả sử thành công:
 - Uplink sử dụng khóa phi tập trung để mã hóa khóa đã dùng cho mã hoá tập tin này.
 - Uplink sẽ gửi một con trỏ tới Vệ tinh, chứa thông tin sau:
 - Nút lưu trữ cuối cùng đã thành công
 - Đường dẫn được mã hóa của tập tin này
 - Thuật toán mã hoá được sử dụng
 - ID mảnh đã chọn
 - Khóa đã được mã hóa và siêu dữ liệu khác
 - Hàm băm sử dụng cho việc băm
 - Chữ ký
- Cuối cùng, Uplink sau đó sẽ tiến hành phân đoạn tiếp theo, tiếp tục xử lý các phân đoạn cho đến khi toàn bộ luồng hoàn thành. Mỗi phân đoạn có một khóa mã hóa mới. Phân đoạn cuối cùng được lưu trữ trong luồng sẽ chứa siêu dữ liệu bổ sung gồm:
 - Luồng có bao nhiêu mảnh dữ liệu

- Kích thước của các mảnh dữ liệu, tính bằng byte
- Vị trí bắt đầu của phân khúc đầu tiên
- Thuộc tính mở rộng và siêu dữ liệu khác
- Theo định kỳ, các nút lưu trữ sau đó sẽ gửi băng thông lớn nhất mà họ nhận được như một phần của việc tải lên Vệ tinh phù hợp để thanh toán.

Tải xuống

Khi người dùng muốn tải xuống một tệp, đầu tiên người dùng sẽ gửi yêu cầu dữ liệu tới Uplink. Sau đó, Uplink cố gắng giảm số lượng việc gửi/nhận dữ liệu tới Vệ tinh bằng cách yêu cầu cụ thể các con trỏ của một vài mảnh dữ liệu đầu tiên và cuối cùng. Uplink cần con trỏ của mảnh dữ liệu cuối cùng để tìm hiểu kích thước của đối tượng, kích thước và số lượng mảnh dữ liệu và cách giải mã dữ liệu.

Đối với mỗi con trỏ mảnh dữ liệu được yêu cầu, Vệ tinh sẽ:

- Xác thực rằng Uplink có quyền truy cập để tải xuống nội dung của con trỏ mảnh dữ liệu và có đủ tiền để trả cho việc tải xuống.
- Tạo băng thông không giới hạn cho việc truyền tải mảnh dữ liệu.
- Tra cứu thông tin liên lạc cho các nút lưu trữ được liệt kê trong con trỏ.
- Trả về con trỏ mảnh dữ liệu được yêu cầu, băng thông và thông tin liên hệ nút lưu trữ cho mỗi mảnh dữ liệu.

Uplink sẽ xác định xem có cần thêm mảnh dữ liệu cho dữ liệu mà nó nhận được hay không và sẽ yêu cầu các con trỏ mảnh dữ liệu còn lại nếu cần.

Khi tất cả các con trỏ mảnh dữ liệu cần thiết đã được trả về, nếu các mảnh dữ liệu được yêu cầu không phải là nội tuyến, Uplink sẽ bắt đầu các yêu cầu song song và đo băng thông cho tất cả các nút lưu trữ thích hợp.

Bởi vì không phải tất cả các đoạn của mảnh dữ liệu đều cần thiết để phục hồi, các đối tượng có thời gian chờ lâu sẽ bị loại bỏ và cải thiện hiệu suất rõ rệt và rõ ràng sẽ đạt được bằng cách cho phép Uplink hủy tải xuống chậm nhất.

Uplink sẽ kết hợp các đoạn của mảnh dữ liệu nhận được và giải mã dữ liệu. Nếu việc tải xuống bị hủy bỏ vì bất kỳ lý do gì, mỗi nút lưu trữ sẽ giữ mức băng thông đã sử dụng lớn nhất mà nó nhận được, nhưng nó sẽ xóa bỏ tất cả dữ liệu yêu cầu có liên quan khác. Dù bằng cách nào, các nút lưu trữ sau đó sẽ gửi băng thông đã được sử dụng như một phần của quá trình tải xuống tới Vệ tinh thích hợp để thanh toán sau.

Xóa

Khi người dùng muốn xóa một tập tin, thao tác xóa đầu tiên được nhận bởi Uplink. Sau đó, Uplink sẽ yêu cầu tất cả các con trỏ mảnh dữ liệu cho tập tin.

Đối với mọi con trỏ mảnh dữ liệu, Vệ tinh sẽ:

- Xác thực rằng Uplink có quyền truy cập để xóa các mảnh tập tin.
- Tạo thỏa thuận và ký xác nhận để xóa phân đoạn, vì vậy nút lưu trữ biết Vệ tinh đang chờ xóa.
- Tra cứu thông tin liên lạc của các nút lưu trữ được liệt kê trong con trỏ.
- Trả lại các mảnh dữ liệu, thỏa thuận và thông tin liên lạc.

Đối với tất cả các phân đoạn từ xa, Uplink sẽ bắt đầu các yêu cầu song song tới tất cả các nút lưu trữ thích hợp để báo hiệu rằng các mảnh dữ liệu đang bị xóa.

- Các nút lưu trữ sẽ trả về một thông điệp cho biết rằng nút lưu trữ đã nhận được thao tác xóa và sẽ xóa cả tập tin và thông tin liên quan của nó.
- Uplink sẽ tải lên tất cả các tin nhắn mà nó nhận được từ các nút lưu trữ đang hoạt động trở lại Vệ tinh. Vệ tinh sẽ yêu cầu một tỷ lệ phần trăm của tin nhắn xác nhận trên tổng số các nút lưu trữ đã ký để đảm bảo rằng Uplink đã thực hiện việc thông báo các nút lưu trữ là đối tượng đã bị xóa.
- Vệ tinh sẽ xóa các con trỏ mảnh dữ liệu và ngừng tính phí cho người dùng.
- Uplink sẽ trả lại trạng thái thành công.

Theo định kỳ, các nút lưu trữ sẽ yêu cầu Vệ tinh cho phép dọn dẹp dữ liệu rác trên các nút lưu trữ đã được sử dụng trong các thao tác xoá. Vệ tinh sẽ từ chối yêu cầu dọn dẹp dữ liệu rác khi thao tác này xảy ra quá thường xuyên.

Di chuyển

Khi người dùng muốn di chuyển một tập tin, đầu tiên Uplink nhận được yêu cầu di chuyển tập tin sang một đường dẫn mới. Sau đó, Uplink yêu cầu tất cả các con trỏ mảnh dữ liệu của tập tin đó.

Đối với mọi con trỏ mảnh dữ liệu, Vệ tinh sẽ:

- Xác nhận rằng Uplink có quyền truy cập để tải xuống.
- Trả về siêu dữ liệu của mảnh dữ liệu được yêu cầu.

Đối với mọi con trỏ của mảnh dữ liệu, Uplink sẽ:

- Giải mã siêu dữ liệu bằng khóa mã hóa xuất phát từ đường dẫn.
- Lấy thông tin về đường dẫn mới.
- Mã hóa lại siêu dữ liệu bằng khóa mã hóa mới được lấy từ đường dẫn mới.

Uplink yêu cầu Vệ tinh thêm tất cả các con trỏ mảnh dữ liệu đã sửa đổi và xóa tất cả các con trỏ mảnh dữ liệu cũ trong một thao tác so sánh và trao đổi dữ liệu.

Vệ tinh sẽ xác nhận rằng:

- Uplink có ủy quyền thích hợp để xóa đường dẫn cũ và tạo đường dẫn mới.
- Nội dung của đường dẫn cũ đã không thay đổi kể từ khi hoạt động chung bắt đầu.

Nếu xác thực thành công, Vệ tinh sẽ thực hiện thao tác. Các nút lưu trữ sẽ không liên quan và không thực hiện bất cứ yêu cầu nào về việc thực hiện thao tác di chuyển tập tin.

Do sự phức tạp trong việc sửa đổi hàng loạt con trỏ dữ liệu, các hoạt động di chuyển hiệu quả có thể không được thực hiện trong phiên bản đầu tiên của mạng này.

Sao chép

Khi người dùng muốn sao chép một tập tin, đầu tiên Uplink nhận được yêu cầu sao chép một tập tin vào một đường dẫn mới. Sau đó, Uplink yêu cầu tất cả các con trỏ mảnh dữ liệu của tập tin. Đối với mọi con trỏ mảnh dữ liệu, Vệ tinh sẽ:

- Xác nhận rằng Uplink có quyền truy cập để tải xuống.
- Tra cứu thông tin liên lạc cho các nút lưu trữ được liệt kê trong con trỏ.
- Trả về siêu dữ liệu của mảnh dữ liệu được yêu cầu, ID mảnh mới và thông tin liên hệ.

Đối với mọi con trỏ mảnh dữ liệu, Uplink sẽ:

- Giải mã siêu dữ liệu bằng khóa mã hóa xuất phát từ đường dẫn.
- Thay đổi đường dẫn đến đường dẫn mới.
- Gọi một thao tác sao chép trên mỗi nút lưu trữ từ con trỏ để sao chép mảnh bằng ID mảnh mới.
- Chờ các nút lưu trữ phản hồi rằng chúng đã sao chép dữ liệu và nó sẽ loại bỏ các nút không thành công.
- Mã hóa lại siêu dữ liệu với ID mảnh mới và khóa mã hóa được lấy từ đường dẫn mới.

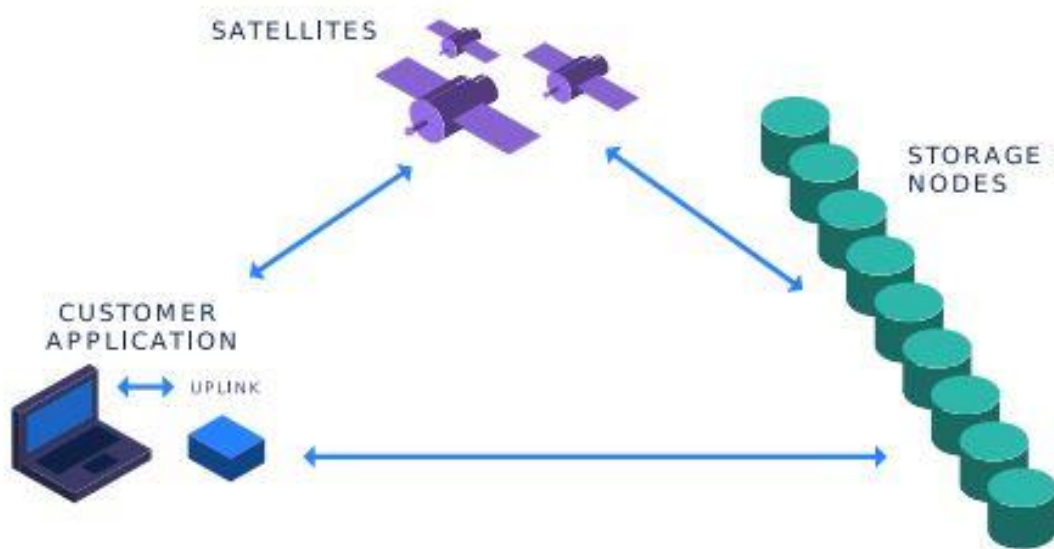
Cuối cùng, Uplink tải lên tất cả các con trỏ phân đoạn đã sửa đổi lên Vệ tinh. Điều quan trọng là nút lưu trữ có thể xác định dữ liệu bằng cả ID mảnh cũ và mới. Nếu một trong các ID mảnh nhận được thao tác xóa, ID mảnh khác sẽ tiếp tục hoạt động. Chỉ sau khi cả hai phần bị xóa, nút lưu trữ mới thực hiện giải phóng không gian.

Liệt kê

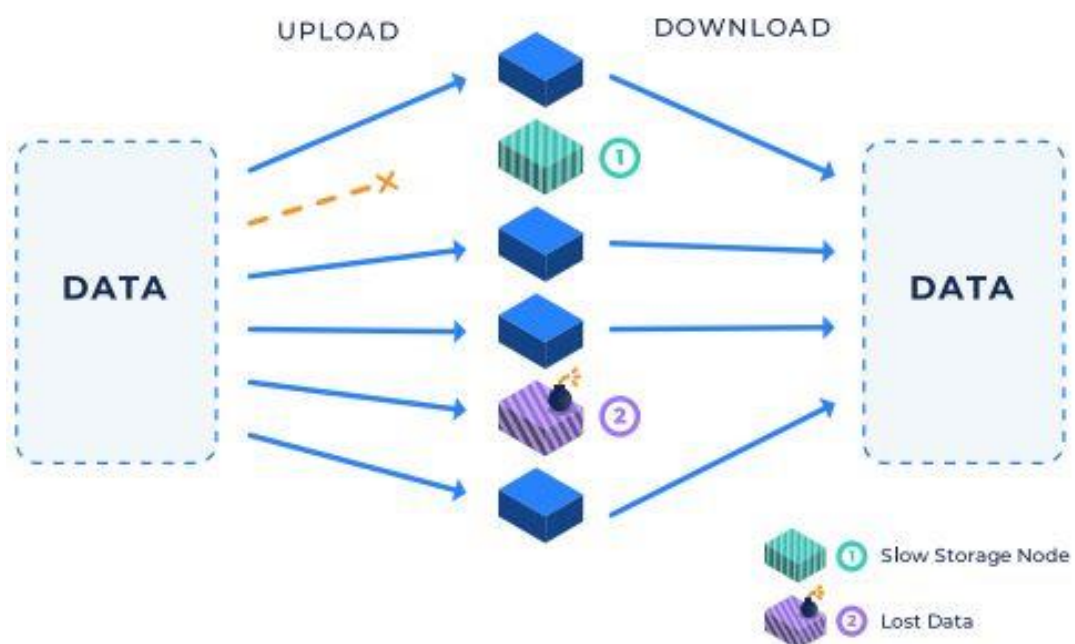
Khi người dùng muốn liệt kê các tập tin:

- Đầu tiên, Uplink sẽ nhận một yêu cầu liệt kê danh sách của đối tượng.
- Sau đó, Uplink sẽ phân tích yêu cầu trên các đường dẫn.
- Tiếp theo, Uplink sẽ yêu cầu tới Vệ tinh thích hợp để lấy danh sách tập tin.
- Sau đó, Vệ tinh sẽ xác thực rằng Uplink có quyền truy cập phù hợp và sau đó trả lại danh sách tập tin được yêu cầu.
- Cuối cùng, Uplink sẽ nhận kết quả từ Uplink và trả về danh sách tập tin cho người dùng.

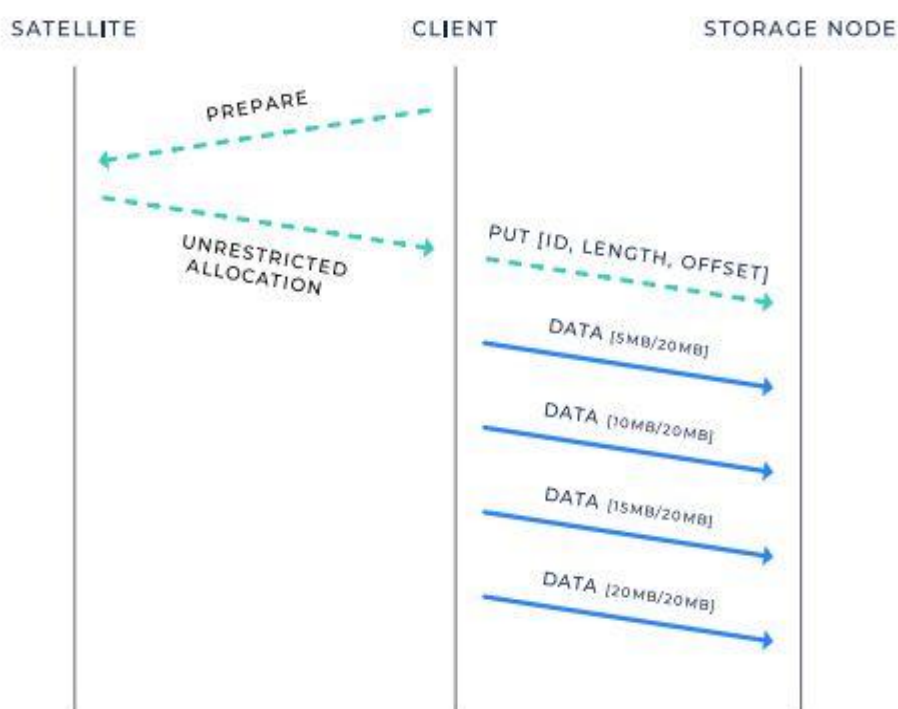
3.2.2.3. Mô hình thực thi của Storj



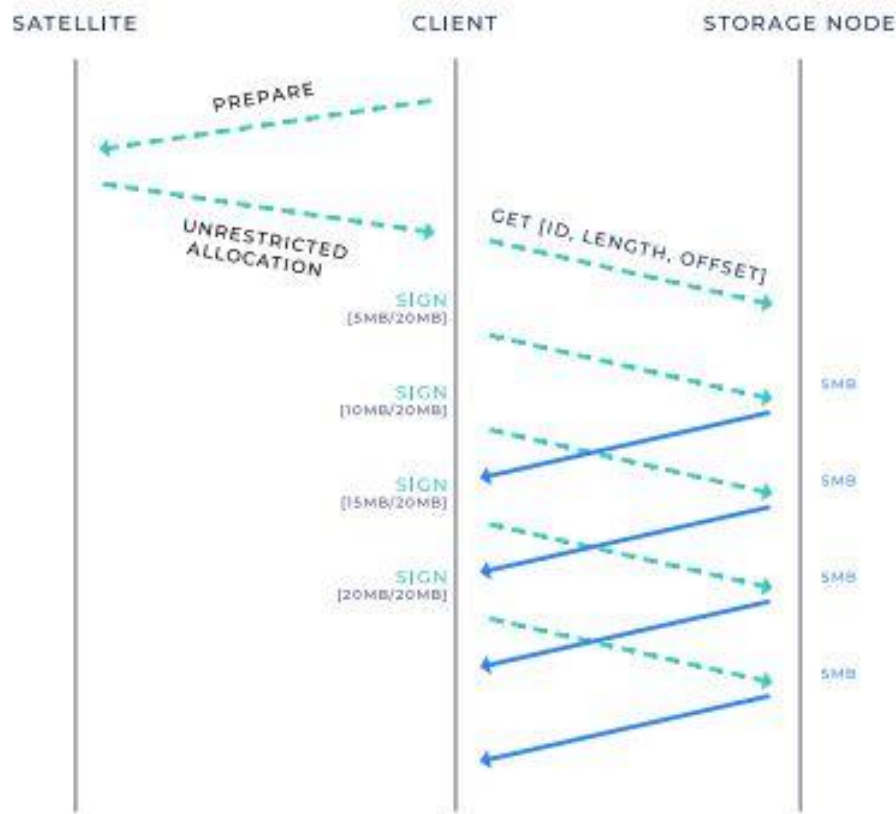
Hình 3. 5: Các lớp kết nối trong mô hình mạng Storj [\[1\]](#)



Hình 3. 6: Mô hình hoạt động quá trình tải lên và tải xuống trong mạng Storj [\[1\]](#)



Hình 3. 7: Sơ đồ hoạt động của quá trình đăng tải dữ liệu trong mạng Storj [\[1\]](#)



Hình 3. 8: Sơ đồ hoạt động của quá trình tải xuống dữ liệu trong mạng Storj [\[1\]](#)

Uplink

Uplink là thuật ngữ được sử dụng để xác định bất kỳ phần mềm hoặc dịch vụ nào gọi libuplink để tương tác với vệ tinh và các nút lưu trữ. Một vài dạng của Uplink:

- Libuplink là một thư viện cung cấp quyền truy cập vào việc lưu trữ và truy xuất dữ liệu trong mạng Storj.
- Gateways hoạt động như các lớp tương thích giữa một dịch vụ hoặc ứng dụng và mạng Storj. Chúng chạy như một dịch vụ cùng vị trí với bất kỳ nơi nào dữ liệu được tạo và sẽ giao tiếp trực tiếp với các nút lưu trữ để tránh chi phí băng thông trung tâm.

- Uplink CLI là một ứng dụng dòng lệnh gọi tới libuplink, cho phép người dùng tải lên và tải xuống các tệp, tạo và xóa các nhóm, quản lý quyền của tệp và các tác vụ liên quan khác. Nó nhằm mục đích cung cấp trải nghiệm quen thuộc với những gì bạn có thể mong đợi khi sử dụng các công cụ Linux / UNIX như scp hoặc rsync.

3.3.3. Giới thiệu Decentralized Certificate Authority (DeCert [\[19\]](#))

Tổng quan về Decert

DeCert là một giao thức xác thực chứng chỉ mới dựa trên sự kết hợp các tiến bộ trong hệ thống phi tập trung và cơ sở hạ tầng khóa công khai. DeCert có thể cấp chứng chỉ TLS và SSL miễn phí một cách nhanh chóng và an toàn. Hơn nữa, trong việc ghi lại tất cả các chứng chỉ được cấp trên Blockchain, DeCert có một cách tiếp cận khác biệt rõ ràng đối với vấn đề tin cậy của người dùng dựa trên cơ sở hạ tầng khóa công khai. Thay vì tin tưởng một cách mù quáng vào các chứng chỉ do DeCert giới thiệu, người dùng mạng tham gia vào việc bỏ phiếu dựa trên mã thông báo để quyết định chứng chỉ nào đáng tin cậy. Vì bất kỳ người tham gia mạng nào cũng có thể thấy tất cả các chứng chỉ và phiếu bầu trên mạng, DeCert là cơ quan chứng nhận đầu tiên cho phép người dùng Internet cá nhân đưa ra quyết định sáng suốt về việc sẽ tin tưởng ai.

Không có thay đổi cơ bản đối với cơ sở hạ tầng khóa công khai (PKI - public key infrastructure) trong nhiều thập kỷ. Các cơ quan cấp chứng chỉ (CA) đáng tin cậy cấp chứng chỉ được mã hóa cho chủ sở hữu trang web cho phép họ chứng minh danh tính. Sau đó, người dùng Internet mã hóa các gói tin bằng các chứng chỉ này, đảm bảo rằng chỉ chủ sở hữu trang web được chỉ định mới có thể đọc chúng.

Công nghệ DeCert đặt tất cả người tham gia vào một mạng phi tập trung, DeCert duy trì bản ghi cập nhật liên tục các chứng chỉ TLS và SSL hợp lệ.

DeCert được tạo thành từ:

- Hợp đồng thông minh - được viết bằng Solidity cho mạng Ethereum

- CA tùy chỉnh - cơ quan chứng nhận tùy chỉnh được viết bằng Golang
- Ứng dụng phân cấp - ứng dụng web để tương tác với DeCert trên Blockchain
- Máy chủ web khác - để xử lý thử nghiệm, triển khai và bảo trì hệ thống

Cơ quan cấp chứng chỉ

Chứng chỉ TLS và SSL là những chứng chỉ cần thiết cho một đường truyền truy cập internet an toàn. Bằng cách mã hóa các gói bằng khóa riêng mà chỉ trang web biết, các chứng chỉ này cho phép chủ sở hữu trang web chứng minh với mọi người rằng họ thực sự sở hữu tên miền của họ.

Các trình duyệt Internet như Google Chrome và FireFox là các tổ chức tin cậy như Comodo, Qualys, Symantec, DigiCert và Let's Encrypt để cấp chứng chỉ TLS và SSL hợp lệ. Do đó, bất cứ khi nào các trình duyệt này gặp phải chứng chỉ được ký bởi một trong những công ty này, họ hoàn toàn tin tưởng vào nó. Cách tiếp cận này có ba vấn đề cơ bản sau:

- CA chỉ có thể ký chứng chỉ với số lượng khóa rất nhỏ.
- Không có cách nào hiệu quả để loại bỏ chứng chỉ cá nhân.
- Không có cách nào để biết khi nào chứng chỉ mới thay cho chứng chỉ cũ sẽ được cấp.

Bởi vì trình duyệt phải lưu trữ tất cả các khóa đáng tin cậy, CA chỉ có thể ký chứng chỉ với một số lượng nhỏ danh tính. Do đó, nếu khóa CA bị lộ, một tỷ lệ lớn các chứng chỉ mà họ đã cấp sẽ trở nên không an toàn. Hơn nữa, bất kỳ sự thỏa hiệp nào đối với khóa định danh của các trình duyệt sẽ có hai lựa chọn không hợp lý:

- Đầu tiên, họ có thể chọn không tải bất kỳ trang web nào có chứng chỉ liên quan đến khóa định danh đó. Như thế giới đã thấy khi CEO của Trustico tiết lộ 23.000 khóa riêng tư trong một email (Z. Whittaker, "Trustico compromises own customers' https private keys in spat with partner," Mar 2018) thì điều này có thể đồng thời gỡ xuống hàng ngàn trang web.

Hoặc, họ có thể chọn tiếp tục duyệt các trang web có khả năng không an toàn và có nguy cơ người dùng Internet có thể bị đánh cắp dữ liệu.

- Thứ hai, không có cơ sở hạ tầng để thu hồi hiệu quả các chứng chỉ cá nhân, chủ sở hữu trang web sẽ bất lực trong việc ngăn chặn tin tặc mạo danh họ sau khi chứng chỉ bị rò rỉ. Điều này có nghĩa là cho đến khi hết hạn chứng nhận, các tác nhân độc hại có thể sử dụng các chứng chỉ bị đánh cắp để giả danh thành chủ sở hữu trang web mà không bị xử phạt. Hậu quả tiềm tàng của việc này đã được nhấn mạnh khi vào năm 2011, người ta đã phát hiện ra rằng Comodo đã cấp chứng nhận gian lận cho Google, Yahoo và Microsoft (P. Roberts, “Phony ssl certificates issued for google, yahoo, skype, others,” Mar 2011). Mặc dù các công ty này có thể làm việc với các trình duyệt để nhanh chóng loại bỏ các chứng chỉ nguy hiểm, nhưng doanh nghiệp Internet nhỏ và vừa không có tài nguyên để nhận ra rằng một trong những chứng chỉ của họ đã bị rò rỉ hoặc bị làm điều gì đó gây nguy hiểm.

Phương pháp tiếp cận của DeCert

Giao thức DeCert thể hiện tính bảo mật thông qua tính minh bạch và các hành động tập thể. Nó thu hút tất cả những người tham gia mạng - người dùng Internet, chủ sở hữu trang web và CA để duy trì hồ sơ công khai về các chứng chỉ hợp lệ. Có ba thao tác quan trọng mà người dùng có thể thực hiện trên mạng:

- Thêm chứng chỉ mới - CA cấp chứng chỉ cho chủ sở hữu trang web và thêm nó vào Blockchain. Hiện tại, các chứng chỉ đang sử dụng một phiên bản của tiêu chuẩn X.509. Sau khi được cấp một chứng chỉ mới, người dùng Internet sẽ có thể tra cứu nó trên Blockchain để xác định xem nó có hợp lệ hay không. Do đó, chứng chỉ trên Blockchain phải chứa tất cả thông tin cần thiết để xác định chứng chỉ đó là duy nhất.

- Bỏ phiếu trên chứng chỉ - chủ sở hữu trang web bỏ phiếu trên Blockchain mà họ nghĩ là chứng chỉ hợp lệ hoặc không hợp lệ. DeCert sử dụng hệ thống bỏ phiếu dựa được xây dựng và tuân thủ theo chuẩn ERC-20.
- Truy vấn chứng chỉ - sau khi thấy chứng chỉ, người dùng Internet sẽ lấy thông tin từ Blockchain để xác định xem họ có tin tưởng hay không. Hệ thống cho phép người dùng linh hoạt lựa chọn chứng chỉ nào để tin tưởng dựa trên ý kiến của các đối tượng tin tưởng được. Đồng thời, hệ thống còn cho phép mạng loại bỏ hiệu quả các chứng chỉ cá nhân. Ví dụ, một chủ sở hữu trang web nhận ra rằng chứng chỉ của họ đã bị xâm phạm và đăng phát thông tin này lên mạng bằng cách bỏ phiếu. Chỉ cần nhìn vào số phiếu hợp lệ và không hợp lệ trên một chứng chỉ nhất định, bất kỳ ai truy cập trang web đều được thông báo ngay lập tức về những nguy hiểm tiềm ẩn xung quanh chứng chỉ đó.

Chương 4

TRÌNH BÀY, ĐÁNH GIÁ BÀN LUẬN VỀ CÁC KẾT QUẢ

4.1. Giới thiệu hệ thống cung cấp xác thực chứng chỉ phi tập trung

Hiện nay, việc xác thực giấy tờ rất cần thiết. Việc mọi người đi đến các quan hành chính nhà nước hay các tổ chức để xin cấp chứng chỉ diễn ra rất thường xuyên. Việc này là cần thiết và quan trọng đối với từng cá nhân hay tổ chức vì việc giả mạo giấy tờ hiện nay diễn ra rất tinh vi. Và những nơi nhận các chứng chỉ cũng sẽ yên tâm hơn khi chúng đã được xác thực.

Nhưng quá trình xác thực hiện nay tốn khá nhiều thời gian của mọi người và nhiều khi quãng đường đi lại khá xa.

Nắm bắt được khó khăn này, chúng tôi đã thực hiện xây dựng một hệ thống trên mạng ứng dụng Blockchain trong việc xây dựng mạng lưới xác thực chứng chỉ phi tập trung. Khi đó, mọi người chỉ cần tham gia vào hệ thống, yêu cầu cấp chứng chỉ từ một tổ chức nào đó và chứng chỉ đó sẽ được nhiều người xác thực.

Trường hợp cụ thể chúng tôi đang hướng tới đó là khi một sinh viên muốn ứng tuyển vào một công ty nào đó, thì sinh viên đó cần có bảng điểm của trường. Hiện tại, sinh viên muốn có bảng điểm phải đến trường yêu cầu phòng đào tạo cấp bảng điểm. Sau đó sinh viên phải chờ một vài ngày rồi lên lại trường để lấy bảng điểm. Khi có bảng điểm, sinh viên phải nộp cho công ty để công ty xem xét chấp nhận hay không.

4.2. Mô tả hệ thống

Hệ thống bao gồm các thành phần sau:

- Server: Server này tích hợp Multichain để quản lý các Nodes và Storj để lưu trữ các chứng chỉ
- SP1: Là trung tâm cung cấp chứng chỉ uy tín cho các user và user lấy các chứng chỉ đó để nộp cho các trung tâm khác để xác thực.

- SP2: Là trung tâm nhận chứng chỉ và xác thực lại tính toàn vẹn của dữ liệu dựa trên độ tin cậy của tập dữ liệu cần xác thực. Người cung cấp chứng chỉ cần đưa cho SP2 một đường dẫn chứa các dữ liệu cần xác thực.
- Voters: là các trung tâm, cơ quan tham gia vào mạng lưới của hệ thống, có nhiệm vụ bỏ phiếu cho các tập dữ liệu dựa trên hàm xác thực của hệ thống. Dựa vào đó người dùng hoặc các trung tâm khác có thể tin cậy hoặc không tin cậy tập tin đó. Ngoài ra voters vẫn có các quyền hạn của SP1 và user.
- User: là những người dùng tham gia vào mạng lưới và yêu cầu các chứng chỉ để phục vụ cho mục đích của user đó.

Giả sử, ban đầu có các SP1, Voter, SP2 đã trả phí để tham gia hệ thống và đã được hệ thống định danh. Các SP1 đều có các public key để đăng lên chain.

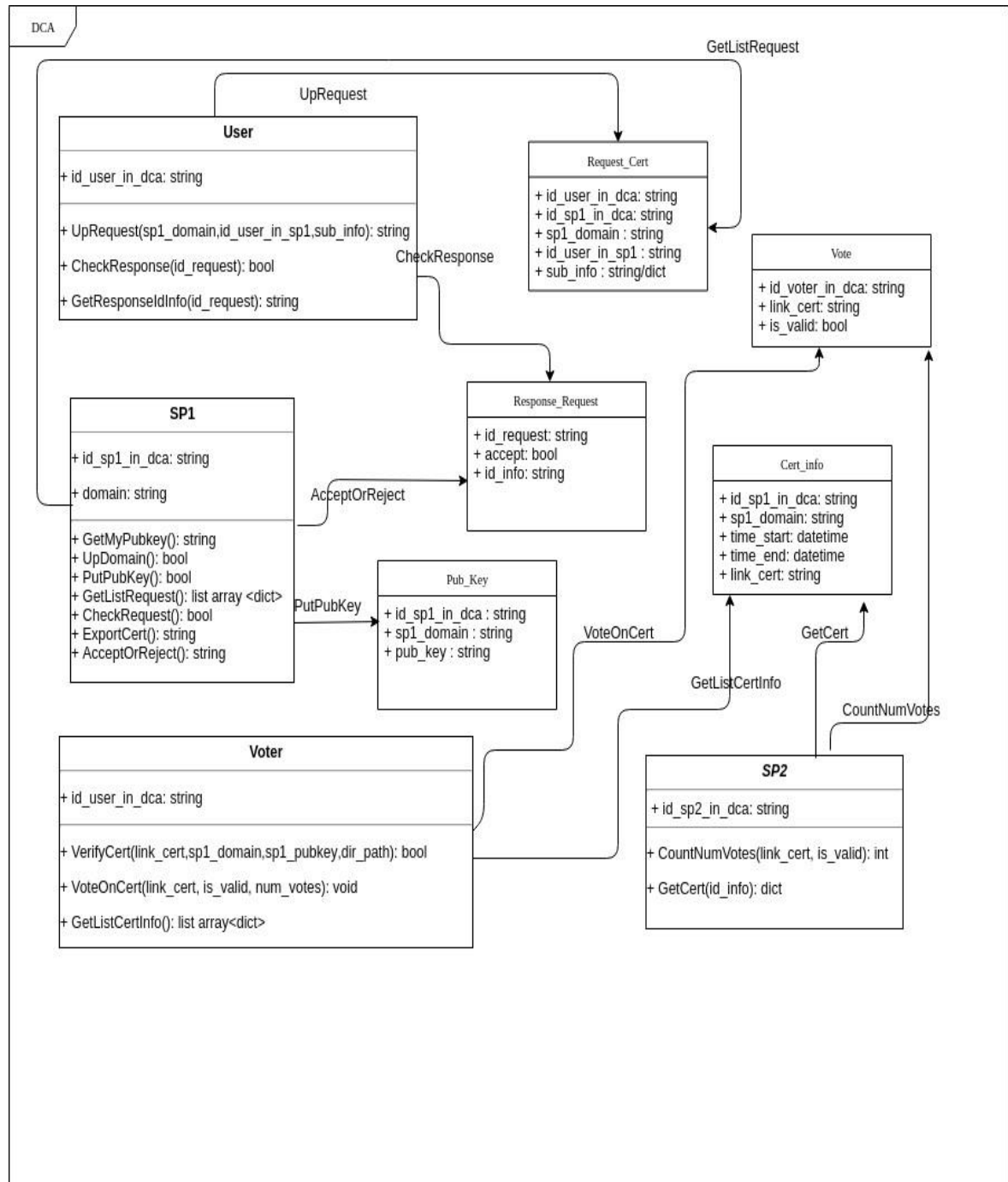
Khi User muốn lấy chứng chỉ từ SP1, thì User sẽ gửi yêu cầu lên SP1 bao gồm các thông tin như: ID của User đó trong hệ thống, ID của User đó trong SP1 (ví dụ như mã số sinh viên trong trường), domain của SP1. Sau đó User sẽ chờ xem liệu SP1 có chấp nhận yêu cầu hay không.

Nếu SP1 không chấp nhận, SP1 sẽ đăng phản hồi lên chain, sau đó User sẽ tìm được phản hồi của SP1. Nếu SP1 đồng ý, SP1 cũng sẽ phản hồi lên chain kèm theo thông tin về chứng chỉ như: ID chứng chỉ trên chain, người cấp chứng chỉ, domain của người cấp chứng chỉ, thời gian hợp lệ, liên kết của tập tin chứng chỉ đã được lưu trên Storj. Cùng lúc đó SP1 sẽ đăng tập tin chứng chỉ đã được ký bởi private key của SP1 lên Storj để lưu trữ. Khi đó, User sẽ có được thông tin về chứng chỉ trên chain, nếu User muốn nộp sang SP2 thì sẽ gửi thông tin này.

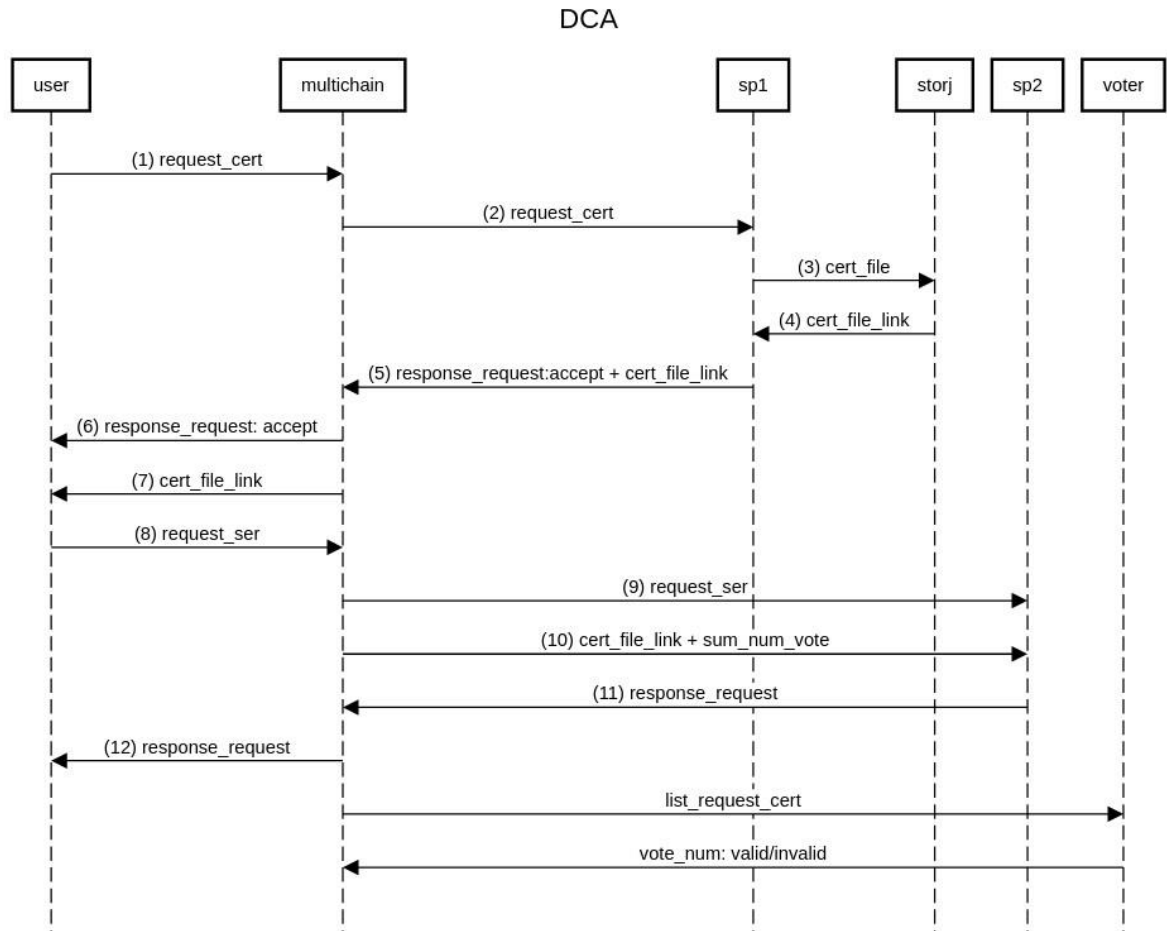
Cộng đồng Voter sẽ luôn cập nhật trên chain để xem tất cả các chứng chỉ cần xác thực. Các Voter sẽ thực hiện xem thông tin chứng chỉ trên chain và từ đó lấy tập tin chứng chỉ về. Sau đó sử dụng public key của SP1 để xác thực xem liệu tập tin này có phải của SP1 ký hay không. Nếu xác thực thành công, các Voter sẽ vote cho chứng chỉ đó một số lượng vote hợp lệ nào đó. Nếu xác thực sai, Voter sẽ vote cho chứng chỉ không hợp lệ.

Khi User nộp thông tin về chứng chỉ lên SP2, SP2 sẽ dựa vào thông tin này để lên chain kiểm tra dựa vào số vote để có nên tin vào chứng chỉ này không.

4.3. Mô hình tiến hành (UML: sơ đồ lớp và sơ đồ tuần tự)



Hình 4. 1: Sơ đồ lớp mô hình DeCert



Hình 4. 2: Sơ đồ tuần tự mô hình DeCert

4.4. Quy trình cài đặt hệ thống xác thực chứng chỉ phi tập trung và kết quả

Chuẩn bị:

- Một máy main server để tạo một private blockchain chung cho mọi node, cho biết mọi node khác biết được địa chỉ IP và port kết nối blockchain của máy.
- Máy service provider, máy user, máy voter và máy service provider khác. Các máy này đều biết trước IP và port blockchain đang mở của máy main server và tên blockchain được tạo với main server.
- Cấu hình tối thiểu yêu cầu:

+ Linux: 64-bit, hỗ trợ Ubuntu 12.04+, CentOS 6.2+, Debian 7+, Fedora 15+, RHEL 6.2+.

+ RAM: 512 MB

+ Không gian ổ đĩa (disk space): 1 GB

- Môi trường và phần mềm:

+ Các máy đã cài đặt thành công multichain, storj và python3

+ Các package python cần thiết:

Mở terminal, cd đến thư mục /DCA (nơi chứa file install.sh cài đặt các package), chạy dòng lệnh: sh install.sh

+ Source code DCA, giải nén và đặt trên thư mục (có dạng : '~/[tên đường dẫn]/DCA/').

Các bước cài đặt hệ thống:

- **Bước 1:** Máy Main server tạo một private blockchain mới, và cấp quyền cho các node truy cập vào chain (Ở hệ thống DCA demo chain mới có tên là chaindemo và tất cả các node đều có quyền xin gia nhập vào chain).

Ở máy Main server chạy file install.sh

Mở terminal, cd đến thư mục DCA/server (nơi chứa file install.sh), chạy dòng lệnh: sh install.sh [ip address của Main server]

```
hainguyen@hai:~/Documents/DCA/server$ ./install.sh 10.10.221.103
MultiChain 2.0.1 Utilities (latest protocol 20009)
Blockchain parameter set was successfully generated.
You can edit it in /home/hainguyen/.multichain/chaindemo/params.dat before running multichaind for the first time.
To generate blockchain please run "multichaind chaindemo -daemon".
MultiChain 2.0.1 Daemon (Community Edition, latest protocol 20009)
Starting up node...
Looking for genesis block...
Genesis block found
Other nodes can connect to this node using:
multichaind chaindemo@10.10.221.103:4771
Listening for API requests on port 4770 (local only - see rpcallowip setting)
Node ready.
```

Hình 4. 3: Hình minh họa Bước 1 – Cài đặt server

Màn hình sẽ in ra IP và port mở để các node khác connect vào chain.

Main server sẽ public IP và port này cho các node khác.

Chạy file main_server.py nhập IP address và port của Main server để lắng nghe và cấp quyền cho các node.

Chạy dòng lệnh: python main_server.py

Nhập IP và port

```
hai Nguyen@hai:~/Documents/DCA/server$ python3 main_server.py
Input server ip address:
10.10.221.103
Input server port:
4771

#####
--          Welcome to DEMO Decentralized CA System          --
--                                                              --
#####

Server is running...
█
```

Hình 4. 4: Hình minh họa Bước 1 – Chạy server

- **Bước 2:** Máy Service provider thứ nhất tạo domain và đăng public key lên chain (tên domain này là duy nhất và sau khi đăng lên multichain không thể xóa đi được nhưng public key có thể thay đổi.).

Ở máy Service provider thứ nhất, cd đến thư mục DCA/client/

Chạy file service_provider_1.py : python service_provider_1.py

Nhập IP và port của Main server

```
thanhtrang@tt-Ins5559:~/Music/DCA/client$ python service_provider_1.py
Input server ip address:
10.10.221.103
Input server port:
4771

MultiChain 2.0 Daemon (Community Edition, latest protocol 20009)
Retrieving blockchain parameters from the seed node 10.10.221.103:4771 ...
Other nodes can connect to this node using:
multichaind chaindemo@10.10.99.90:4771
Listening for API requests on port 4770 (local only - see rpcallowip setting)
Node ready.

MultiChain 2.0 Daemon (Community Edition, latest protocol 20009)
Starting up node...
Other nodes can connect to this node using:
multichaind chaindemo@10.10.99.90:4771
Listening for API requests on port 4770 (local only - see rpcallowip setting)
Node ready.
Your configuration is saved to: /home/thanhtrang/.local/share/storj/uplink/config.yaml
```

Hình 4. 5: Hình minh họa Bước 2 – Service Provider 1 kết nối server

Chọn tạo domain mới (“1: Sign up.” khi máy mới gia nhập blockchain lần đầu). Chọn 1

Nhập tên domain (hệ thống sẽ up tên domain lên chain và tự tạo RSA key rồi up lên chain).

```
#####
--      print(input your choice: )      --
--      Welcome to DEMO Decentralized CA System      --
--      print("#####")      --
#####
if choose == 0 and choose == 1 and choose == 2:
    print(wrong input!\n)

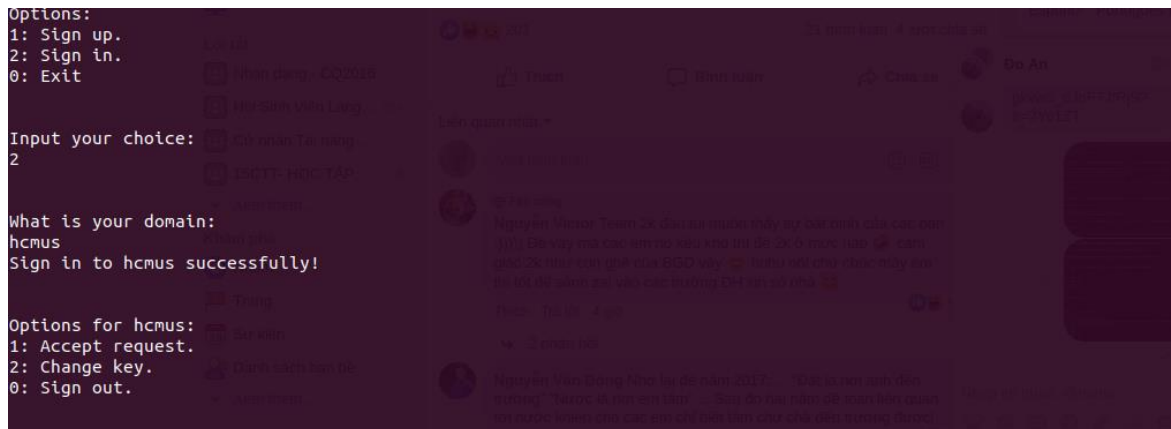
    You are logging in to a service provider account!

    if choose == 0:
        sub.api.stop()
Options:
1: Sign up.
2: Sign in.
0: Exit
    choose == 1:
        print("What is your domain: ")
        spi_domain = input()

Input your choice: 0:Ser_Provider_1(spi_domain)
1
    if spi.UpDomain() is False:
        print("create new domain " + spi_domain)
What is your domain: GenBsakey()
hcmus
pubkey = spi.PutPubKey()
Create new domain hcmus This is your public key: 72863007356d41a6a19c8aa73d82756259991e0061a2c13e38400819e2e95c6f
This is your public key: 72863007356d41a6a19c8aa73d82756259991e0061a2c13e38400819e2e95c6f
Sign up for hcmus successfully! (spi_domain) is False:
Options:
1: Sign up.
2: Sign in.
0: Exit
    print(sign up for " + spi_domain + " successfully!")
    else:
        print(spi_domain + " is already existed!")
```

Hình 4. 6: Hình minh họa Bước 2 – Service Provider 1 tạo domain

Chọn “2: Sign in.” và nhập tên domain.



Hình 4. 7: Hình minh họa Bước 2 – Service Provider 1 đăng nhập

- **Bước 3:** Máy User đăng request

Ở máy user cd đến thư mục DCA/client/

Chạy file user.py

Nhập IP và port của Main server

```
Input server ip address: 10.10.221.103
Input server port: 4771
#####
--
--      Welcome to DEMO Decentralized CA System
--
#####
You are logging in to an user account!

Options:
1: Get a certificate.
2: Check if the request is accepted or not.
0: Exit
```

Hình 4. 8: Hình minh họa Bước 3 – User kết nối server

Chọn “1: Get a certificate.”

Nhập tên domain của tổ chức bạn muốn lấy service

Nhập thông tin của bạn (ở DCA demo là id user trong database của Service provider thứ nhất)

```
--      Welcome to DEMO Decentralized CA System      --
--
#####
You are logging in to an user account!
Options:
1: Get a certificate.
2: Check if the request is accepted or not.
0: Exit

Input your choice:
1

What is the domain of organization:
hcmus

What is your id in hcmus:
1512587

Request sent!
Options:
1: Get a certificate.
2: Check if the request is accepted or not.
0: Exit

Input your choice:
```

Hình 4. 9: Hình minh họa Bước 3 – User yêu cầu lấy chứng chỉ

Sau khi hoàn tất, hệ thống sẽ gửi request đến Service provider thứ nhất.

- **Bước 4:** Máy Service provider thứ nhất duyệt các yêu cầu cấp service và accept hoặc reject từng yêu cầu.

Ở máy Service provider thứ nhất cd đến thư mục DCA/client/

Chạy file service_provider_1.py

Nhập IP và port của Main server

Chọn “2: Sign in.”


```
#####
--
--      Welcome to DEMO Decentralized CA System      --
--
#####

print("#####")
You are logging in to a service provider account!
print("      Welcome to DEMO Decentralized CA System      ")
print("--")
Options:
1: Sign up.
2: Sign in.
0: Exit
Input your choice: 2
What is your domain: hcmus
Sign in to hcmus successfully!
Options for hcmus:
1: Accept request.
2: Change key.
0: Sign out.
Input your choice: 1
1.34 KB / 1.34 KB [=====] 100.00% 0s
Created sj://17692233891453558277/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf
Options for hcmus:
1: Accept request.
2: Change key.
0: Sign out.
```

Hình 4. 10: Hình minh họa Bước 4 – Service Provider 1 đăng nhập

Sau khi login chọn “1: Accept request.”.

```
Options for hcmus:
1: Accept request.
2: Change key.
0: Sign out.

Input your choice:
1

e07c826bf0db1c7d8350fc68692edac173fb4dad
1.34 KB / 1.34 KB [=====] 100.00% 0s
Created sj://17692233891453558277/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf

Options for hcmus:
1: Accept request.
2: Change key.
0: Sign out.
```

Hình 4. 11: Hình minh họa Bước 4 – Service Provider 1 chấp nhận yêu cầu từ User

File certificate sẽ được đăng lên storj.

- **Bước 5:** Máy User kiểm tra request certificate đã được chấp nhận hay chưa.
Ở máy User chọn “2: Check if the request is accepted or not.”

```
Options:
1: Get a certificate.
2: Check if the request is accepted or not.
0: Exit

Input your choice:
2

Your request is accepted
This is your Certificate Infomation ID on chain: ea99a08e5557a77f98ae3666f867fb668
b411439e76de50f47a4a4bd9e0e69b7
You can send it to anyone!
Options:
1: Get a certificate.
2: Check if the request is accepted or not.
0: Exit
```

Hình 4. 12: Hình minh họa Bước 5 – User kiểm tra phản hồi

Nếu request được accept, hệ thống sẽ trả về ID của certificate.

User gửi ID này cho Service provider thứ 2.

- **Bước 6:** Máy Service provider thứ 2 sau khi nhận ID certificate từ User gửi đến, sẽ tiến hành kiểm tra số vote hiện tại của certificate rồi dựa vào đó quyết định có tin cậy hay không.

Ở máy Service provider thứ 2 cd đến đường dẫn DCA/client

Chạy terminal, gõ lệnh: python service_provider_2.py

Nhập IP và port của Main server

Nhập ID của certificate và xem số vote

```

thanhtrang@ctt-lms5559:~/Music/DCA/clients$ python service_provider_2.py
Input server ip address:
10.10.221.103
Input server port:
4771
#####
--
--      Welcome to DEMO Decentralized CA System      --
--
--
#####

You are logging in to a service provider account!

Do you want to check a certificate (Y/N):
y

Input a Certificate Information ID:
ea99a08e5557a77f98ae3666f867fb668b411439e76de50f47a4a4bd9e0e69b7

Certificate Information:
Organization ID: 1L6vkhTJwe7mjkYh62UxSer9acU4KTBHhS3zn
Certificate ID: e07c826bf0db1c7d8350fc68692edac173fb4dad
Domain name: hcmus
Created time: 2019-06-25 19:46:17
Expiration time: 2019-06-25 22:32:57
Valid votes: 0
Invalid votes: 0

Do you want to download this certificate (Y/N):

```

Hình 4. 13: Hình minh họa Bước 6 – Service Provider 2 kiểm tra chứng chỉ

Chọn y hoặc n nếu đồng ý hoặc không đồng ý tải tập tin chứng chỉ xuống

```

Do you want to download this certificate (Y/N):
y
1.34 KB / 1.34 KB [=====] 100.00% 0s
Downloaded sj://17692233891453558277/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf to /home/thanhtrang/Downloads/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf

```

Hình 4. 14: Hình minh họa Bước 6 – Service Provider 2 tải xuống tập tin chứng chỉ

- **Bước 7:** Máy Voter tiến hành xác thực các certificate và thực hiện vote dựa trên hàm xác thực tham khảo của hệ thống.

Chạy terminal, gõ lệnh: `python voter.py`

Nhập IP và port của Main server

```
#####
lib          rsakey      temp          _init_.py
top          You are logging in to a voter account!
ments
List of Certificate:
CERT 0
Organization ID: 1L6vkhTJwe7mjkcYh62UxSer9acU4KTBHhS3zn
Domain name: hcmus
Created time: 2019-06-25 19:46:17
Expiration time 2019-06-25 22:32:57
Certificate ID: e07c826bf0db1c7d8350fc68692edac173fb4dad
CERT 1
Organization ID: 1L6vkhTJwe7mjkcYh62UxSer9acU4KTBHhS3zn
Domain name: hcmus
Created time: 2019-06-25 19:42:22
Expiration time 2019-06-25 22:29:02
Certificate ID: 8455c7f433ea31fc66755c5ab03a754043070293
CERT 2
Organization ID: 1L6vkhTJwe7mjkcYh62UxSer9acU4KTBHhS3zn
Domain name: hcmus
Created time: 2019-06-25 19:33:26
Expiration time 2019-06-25 22:20:06
Certificate ID: 5ff43aca09e17a5b6ea639be180a1e8519e9b23b
Choose one Certificate to vote:
Input a Certificate ID:
```

Hình 4. 15: Hình minh họa Bước 7 – Voter kết nối server và kiểm tra các chứng chỉ

Nhập ID của certificate muốn vote và tên domain của tổ chức tương ứng.

```

Choose one Certificate to vote:
Input a Certificate ID:
e07c826bf0db1c7d8350fc68692edac173fb4dad
Input a Domain name:
hcmus

Verifying...

 1.34 KB / 1.34 KB [=====] 100.00% 0s
Downloaded sj://17692233891453558277/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf to /home/thanhtrang/Music/DCA/client/temp/e07c826bf0db1c7d8350fc68692edac173fb4dad.pdf

This certificate is valid? True
What kind of vote do you want to have? (Valid[V]/Invalid[I]):
v
Number of votes:
10

Voting done!

Do you want to continue to vote? (Y/N):

```

Hình 4. 16: Hình minh họa Bước 7 – Voter xác thực và tiến hành vote

Hệ thống sẽ xuất ra kết quả hàm xác thực tham khảo

Chọn valid hoặc invalid, và chọn số vote muốn vote

```

Input a Certificate Information ID:
ea99a08e5557a77f98ae3666f867fb668b411439e76de50f47a4a4bd9e0e69b7

Certificate Information:
=====
Organization ID: 1L6vkhTJwe7mjkYh62UxSer9acU4KTBHhS3zn
Certificate ID: e07c826bf0db1c7d8350fc68692edac173fb4dad
Domain name: hcmus
Created time: 2019-06-25 19:46:17
Expiration time: 2019-06-25 22:32:57
Valid votes: 10
Invalid votes: 0
=====

Do you want to download this certificate (Y/N):

```

Hình 4. 17: Hình minh họa Bước 7 – Service Provider 2 kiểm tra lại chứng chỉ

Sau khi vote, máy Service provider thứ hai kiểm tra lại lần nữa

Chương 5

KẾT LUẬN

5.1. Kết quả đạt được

Qua quá trình tìm hiểu và nghiên cứu, chúng tôi đã hiểu thêm về Blockchain, framework Multichain và Storj cũng như cơ chế hoạt động của chúng. Từ đó chúng tôi thực hiện và phát triển hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung. Hệ thống gồm những chức năng sau:

Hỗ trợ lưu trữ dữ liệu trên mạng lưới phi tập trung Storj.

Các lệnh được dùng để thao tác với dữ liệu dựa trên Uplink:

- Cho phép người dùng tạo link lưu trữ file Certificate qua hệ thống.
- Cho phép người dùng xác thực được file Certificate có được là đáng tin cậy hay không.

Phân quyền và xác thực người dùng trên mạng private blockchain và đảm bảo tính toàn vẹn của dữ liệu.

Người dùng có thể quyết định mình có thể tin tưởng một tập dữ liệu nào đó hay không thông qua độ tin cậy trên mỗi tập dữ liệu mà không phụ thuộc và tin cậy vào bên thứ ba.

Thông qua hệ thống, quá trình xác thực và tin cậy dữ liệu sẽ diễn ra nhanh chóng hơn và đáng tin cậy hơn.

Những hạn chế của hệ thống:

- Hệ thống chỉ giải quyết các bài toán đơn giản nên chưa thể hoàn toàn áp dụng được vào thực tế.
- Thuật toán vote cho các tập dữ liệu chưa được hiệu quả và nhiều khả năng còn mang tính chất ngẫu nhiên.
- Hệ thống cần một số lượng lớn người tham gia để có thể đạt được hiệu quả nhất định.

5.2. Đóng góp mới và đề xuất mới

Báo cáo như một tài liệu đúc kết từ nhiều nguồn kiến thức khác nhau về Blockchain, sẽ là tài liệu tham khảo cho những ai muốn tìm hiểu về Blockchain, cũng như để có cái nhìn tổng thể về Blockchain. Tạo nên một ứng dụng hệ thống cung cấp dịch vụ xác thực chứng chỉ phi tập trung dựa trên nền tảng Blockchain.

Chương 6

HƯỚNG PHÁT TRIỂN

6.1. Kiến nghị hướng phát triển tiếp theo cho công nghệ Blockchain

Bên cạnh những thành quả đạt được, chúng tôi dự định sẽ cập nhật các tính năng mới và phát triển hoàn thiện hơn trong tương lai như:

- Tạo giao diện người dùng một cách rõ ràng, dễ sử dụng, đầy đủ tính năng cần thiết
- Tăng tốc độ xử lý và tối ưu hóa các hàm chức năng
- Viết tài liệu hoàn chỉnh về hướng dẫn sử dụng ứng dụng xác thực chứng chỉ phi tập trung để người dùng dễ dàng tiếp cận
- Đưa ứng dụng ra sử dụng thực tế và thu thập ý kiến người dùng, từ đó hiệu chỉnh lại ứng dụng hoàn thiện hơn

6.2. Hướng phát triển của hệ thống xác thực chứng chỉ phi tập trung

Hệ thống nếu muốn được tin tưởng thì sẽ phụ thuộc vào cộng đồng voter. Vì vậy để hoàn thiện hệ thống này, chúng tôi có một số hướng phát triển như sau:

- Sẽ thiết kế cho hệ thống tính phí người dùng, nếu muốn sử dụng được nhiều tính năng của hệ thống, các node phải trả phí tương ứng.
- Để có thể tin tưởng vào những người voter, chúng ta sẽ có một hệ thống vote xem ai đạt được sự tin cậy cao để có thể vote.
- Những người voter cũng sẽ trả phí để mua thẻ cho mỗi lần vote
- Tạo ra một động lực để người voter tham gia một cách trung thực

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] I. Storj Labs, “Storj: A Decentralized Cloud Storage Network,” 30 October 2018. [Trực tuyến]. Available: <https://storj.io/storjv3.pdf>.
- [2] C. S. Company, “MultiChain,” 2019. [Trực tuyến]. Available: <https://www.multichain.com/developers/blockchain-parameters/>.
- [3] MultiChain - Coin Sciences Company, “MultiChain | Open source blockchain platform,” 2019. [Trực tuyến]. Available: <https://www.multichain.com/>.
- [4] StevenPalley, “Storj là gì? Tìm hiểu về tiền mã hóa Storj,” bigcoinvietnam, 2018. [Trực tuyến]. Available: <https://bigcoinvietnam.com/storj-la-gi-tim-hieu-ve-tien-ma-hoa-storj>.
- [5] Aoi, “Storj là gì? Tìm hiểu về đồng tiền điện tử STORJ,” cafebitcoin, 27 03 2018. [Trực tuyến]. Available: <https://cafebitcoin.info/tien-dien-tu/storj-la-gi-tim-tien-dien-tu-storj/>.
- [6] K. Leffew, “Getting Started with the Storj V3 Test Network,” Storj, 10 01 2019. [Trực tuyến]. Available: <https://storj.io/blog/2019/01/getting-started-with-the-storj-v3-test-network-storj-sdk/>.
- [7] K. L. A. D. Lott, “A brief overview of Kademlia and its use in various decentralized platforms,” Storj, 14 02 2019. [Trực tuyến]. Available: <https://storj.io/blog/2019/02/a-brief-overview-of-kademlia-and-its-use-in-various-decentralized-platforms/>.
- [8] T. D. ST, “[XU HUỐNG] Blockchain – Bong bóng hay cách mạng công nghệ?,” Cafef.vn, 23 04 2018. [Trực tuyến]. Available: <http://dgroupholdings.com/xu-huong-blockchain-bong-bong-hay-cach-mang-cong-nghe-d144254/>.

- [9] Wikipedia, “GCM - Galois/Counter Mode,” Wikipedia, 2019. [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Galois/Counter_Mode.
- [10] C. Sciences, “MultiChain JSON-RPC API commands,” MultiChain, 2019. [Trực tuyến]. Available: <https://www.multichain.com/developers/json-rpc-api/>.
- [11] P. Cannon, “Flexible File Sharing With Macaroons,” Storj, 03 03 2019. [Trực tuyến]. Available: <https://storj.io/blog/2019/05/flexible-file-sharing-with-macaroons/>.
- [12] Wikipedia, “Kademlia,” Wikipedia, 2019. [Trực tuyến]. Available: <https://en.wikipedia.org/wiki/Kademlia>.
- [13] C. Sciences, “MultiChain - Address and key format,” Coin Sciences, 2019. [Trực tuyến]. Available: <https://www.multichain.com/developers/address-key-format/>.
- [14] G. a. Danagle, “multichain-web-demo,” MultiChain, 2019. [Trực tuyến]. Available: <https://github.com/MultiChain/multichain-web-demo>.
- [15] KZEMEK, “Macaroons are Better Than Cookies!,” KZEMEK, 2019. [Trực tuyến]. Available: <https://github.com/kzemek/libmacaroons-cpp>.
- [16] G. Greenspan, “MultiChain 1.0 beta 2 and 2.0 roadmap,” MultiChain, 15 01 2017. [Trực tuyến]. Available: <https://www.multichain.com/blog/2017/06/multichain-1-beta-2-roadmap/>.
- [17] T. Mỹ, “Xây dựng blockchain đơn giản với golang. P4 - Wallet (Address),” Personal, 14 02 2018. [Trực tuyến]. Available: <https://kipalog.com/posts/Xay-dung-blockchain-don-gian-voi-golang--P4--Wallet--Address>.
- [18] Wikipedia, “Elliptic Curve Digital Signature Algorithm,” Wikipedia, 2019. [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm.

- [19] L. Hentschker, “DeCERT: A Decentralized Certificate Authority,” DeCERT, 29 05 2018. [Trực tuyến]. Available: <https://www.boazbarak.org/cs127/Projects/decert-FINAL.pdf>.
- [20] amasucci, “SecretBOX,” GitHub, 2018. [Trực tuyến]. Available: <https://github.com/amasucci/secret-box>.
- [21] wikipedia, “Chữ ký số,” wikipedia, 2018. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Chữ_ký_số.
- [22] Wikipedia, “Chữ ký điện tử,” Wikipedia, 2018. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Chữ_ký_điện_tử.

PHỤ LỤC

PHỤ LỤC A

CHỮ KÝ ĐIỆN TỬ

A.1. Tổng quan

Chữ ký điện tử [\[22\]](#) (tiếng Anh: Electronic signature) là thông tin đi kèm theo dữ liệu (văn bản, hình ảnh, video...) nhằm mục đích xác định người chủ của dữ liệu đó.

Chữ ký điện tử được sử dụng trong các giao dịch điện tử. Xuất phát từ thực tế, chữ ký điện tử cũng cần đảm bảo các chức năng: xác định được người chủ của một dữ liệu nào đó: văn bản, ảnh, video,... dữ liệu đó có bị thay đổi hay không.

A.2. Chữ ký số

Hai khái niệm chữ ký số [\[21\]](#) (digital signature) và chữ ký điện tử (electronic signature) thường được dùng thay thế cho nhau mặc dù chúng không hoàn toàn có cùng nghĩa. Chữ ký số chỉ là một tập con của chữ ký điện tử (chữ ký điện tử bao hàm chữ ký số)

Chữ ký số là một dạng chữ ký điện tử dựa trên công nghệ mã khóa công khai. Mỗi người dùng chữ ký số phải có một cặp khóa (keypair), gồm khóa công khai (public key) và khóa bí mật (private key). Khóa bí mật dùng để tạo chữ ký số. Khóa công khai dùng để thẩm định chữ ký số hay xác thực người tạo ra chữ ký số đó.

A.3. Chứng chỉ số

Theo cơ chế chữ ký điện tử như đã đề cập ở trên thì trong chứng chỉ, một tham số quan trọng phải có đó là khóa công khai. Ngoài ra chứng chỉ số còn chứa các thông tin về danh tính của đối tượng được cấp chứng chỉ, bao gồm thông tin về chủ sở hữu chứng chỉ như email, số điện thoại... các thông tin này là tùy chọn theo qui định của nhà cung cấp chứng chỉ số.

Vậy còn một tham số quan trọng trong sử dụng chứng chỉ số, đó là khóa bí mật? Khóa bí mật sẽ không được lưu trong chứng chỉ số. Nó được lưu tại máy tính của chủ sở hữu, chủ sở hữu cần chịu trách nhiệm giữ an toàn khóa bí mật này.

A.4. Ứng dụng của chữ ký điện tử

Chữ ký số có thể sử dụng trong các giao dịch thư điện tử, các e-mail, để mua bán hàng trực tuyến, đầu tư chứng khoán trực tuyến, chuyển tiền ngân hàng, thanh toán trực tuyến mà không sợ bị đánh cắp tiền như với các tài khoản Visa, Master. Ngoài ra, Chữ ký số cũng có thể dùng để kê khai, nộp thuế trực tuyến, khai báo hải quan và thông quan trực tuyến mà không phải mất thời gian đi in các tờ khai, đóng dấu đỏ của công ty rồi đến cơ quan thuế xếp hàng để nộp tờ khai này. Chữ ký số giúp cho các đối tác có thể ký hợp đồng làm ăn hoàn toàn trực tuyến không cần ngồi trực tiếp với nhau, chỉ cần ký vào file hợp đồng và gửi qua e-mail.

Một số ứng dụng chữ ký điện tử điển hình:

- Ứng dụng trong chính phủ điện tử: Ứng dụng của Bộ Tài chính, Ứng dụng của Bộ Công thương, Ứng dụng của Bộ KH-CN, Ứng dụng trong thương mại điện tử,...
- Mua bán, đặt hàng trực tuyến, thanh toán trực tuyến,...
- Ứng dụng trong giao dịch trực tuyến, giao dịch qua email,...

A.5. Lợi ích khi sử dụng chữ ký điện tử

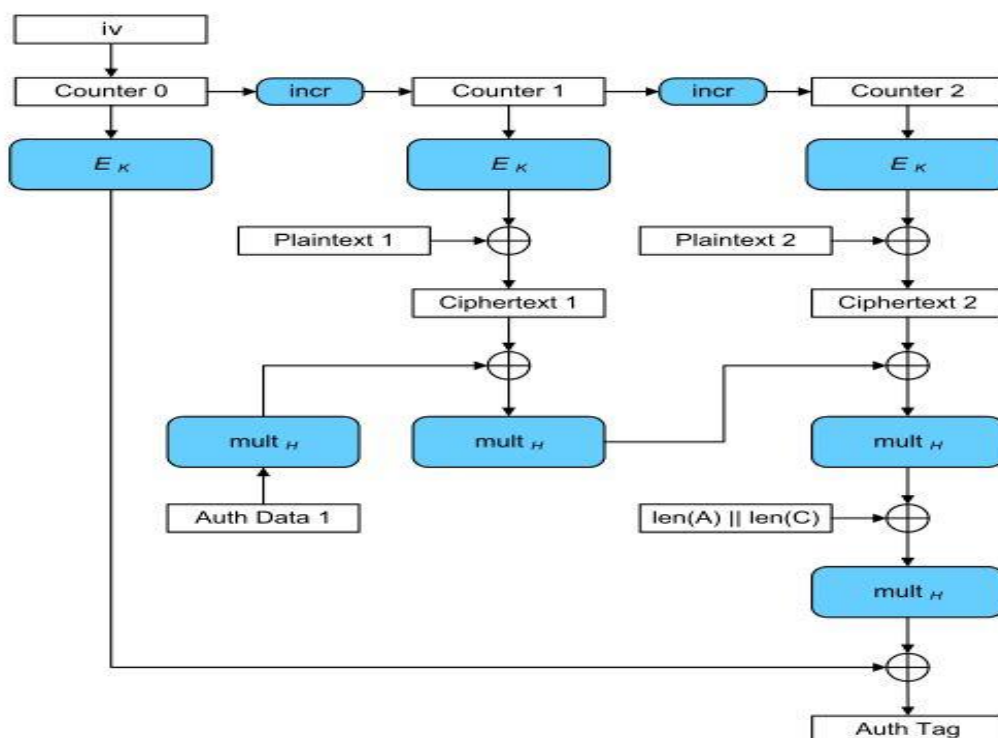
Việc ứng dụng chữ ký số giúp doanh nghiệp tiết kiệm thời gian, chi phí hành chính. Hoạt động giao dịch điện tử cũng được nâng tầm đầy mạnh. Không mất thời gian đi lại, chờ đợi, không phải in ấn các hồ sơ.

Việc ký kết các văn bản ký điện tử có thể diễn ra ở bất kỳ đâu, bất kỳ thời gian nào. Việc chuyển tài liệu, hồ sơ đã ký cho đối tác, khách hàng, cơ quan quản lý... diễn ra tiện lợi và nhanh chóng.

PHỤ LỤC B

AES-GCM

Galois / Counter Mode (GCM) là một chế độ hoạt động cho các khối mật mã mã hóa khóa đối xứng đã được áp dụng rộng rãi vì tính hiệu quả và hiệu suất của nó. Tốc độ thông lượng GCM cho các kênh truyền luôn có tốc độ cao. GCM được định nghĩa cho các thuật toán mã khối với kích thước khối là 128 bit. Mã xác thực tin nhắn Galois (GMAC) là một biến thể chỉ dành cho xác thực tin nhắn của GCM, có thể được sử dụng làm mã xác thực tin nhắn gia tăng. Cả GCM và GMAC đều có thể chấp nhận các vectơ khởi tạo có độ dài tùy ý. Các chế độ hoạt động của mật mã khối khác nhau có thể có các đặc tính và hiệu suất khác nhau đáng kể, ngay cả khi được sử dụng với cùng một mật mã khối. GCM có thể tận dụng tối đa khả năng xử lý song song và triển khai GCM có thể sử dụng hiệu quả kênh truyền dữ liệu hoặc tối ưu với phần cứng. [9]



Hình B. 1: : Sơ đồ hoạt động AES-GCM [9]

PHỤ LỤC C

SecretBox

SecretBox [\[20\]](#) là thuật toán mã hóa và giải mã bí mật. Được xây dựng trên nền tảng AES-256-GCM và Scrypt.

Scrypt là một hàm dẫn xuất khóa trong bộ nhớ cứng. Chức năng bộ nhớ cứng đòi hỏi một số lượng lớn RAM để hoạt động. Scrypt tạo ra rất nhiều số giả ngẫu nhiên cần được lưu trữ ở vị trí RAM. Sau đó thuật toán truy cập các số này một vài lần trước khi trả về một kết quả. Việc tạo ra các con số đòi hỏi nhiều tính toán kỹ và khi chúng được truy cập vài lần, nó hoàn toàn có thể sử dụng bộ nhớ RAM kết hợp với sức mạnh hash chứ không cần phải tạo ra chúng ngay lập tức – một khoảng thời gian và bộ nhớ cân xứng trong điều kiện tối ưu hóa tốc độ.

PHỤ LỤC D

Cài đặt và cấu hình Multichain

D.1. Yêu cầu hệ thống

- Linux: 64-bit, hỗ trợ Ubuntu 12.04+, CentOS 6.2+, Debian 7+, Fedora 15+, RHEL 6.2+.
- Windows: 64-bit, hỗ trợ Windows 7, 8, 10, Server 2008 hoặc mới nhất.
- RAM: 512 MB
- Không gian ổ đĩa (disk space): 1 GB

D.2. Cài đặt trên Window

B1. Truy cập trang web <https://www.multichain.com/download-install/> đến phần *Installing MultiChain Community on Windows* lấy đường dẫn tải về tại mục này (đường dẫn tải về sẽ được cập nhật theo từng phiên bản, hiện tại là bản 2.0.2), với các bản cũ hơn đường dẫn tập tin cài đặt được tại mục *Release history*)

B2. Tải file .zip và giải nén được các file có phần mở rộng là .exe, sử dụng command prompt thay đổi đường dẫn đến thư mục chứa các file đã giải nén.

B3. Các thao tác sau này sẽ thực hiện dưới dạng command line

D.3. Cài đặt trên Linux

Truy cập trang web <https://www.multichain.com/download-install/> đến mục *Installing MultiChain Community on Linux* sẽ có hướng dẫn cài trên linux. Cụ thể như sau:

B1. Mở terminal nhập : *su (nhập mật khẩu của root-user)*

B2. Nhập *cd /tmp* để thay đổi thư mục chứa file cài đặt multichain tạm thời, có thể thay tmp bằng bất kì thư mục nào

B3. Tải về file cài đặt :

Nhập *wget* <https://www.multichain.com/download/multichain-2.0.2.tar.gz>

B4. Giải nén: `tar -xvzf multichain-2.0.2.tar.gz`

B5. Duy chuyển file đã giải nén đến thư mục `/usr/local/bin` để dễ gọi command line sau này bằng cách nhập `cd multichain-2.0.2`, sau đó nhập `mv multichaind multichain-cli multichain-util /usr/local/bin`

B6. Trở về account user thường `exit`

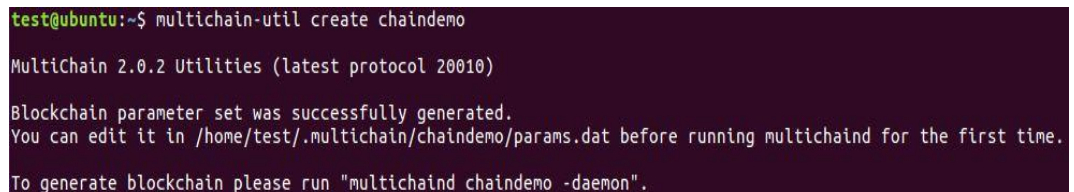
D.4. Cách tạo chain và kết nối các node với nhau trong Multichain

Dữ liệu multichain sẽ được lưu trữ trên linux ở thư mục: `~/.multichain/[tên chain]/`, ở window sẽ lưu tại `%APPDATA%\MultiChain\[tên chain]\`

File cài đặt mặc định (blockchain's default settings) là `params.dat`, file này quy định các thông số blockchain parameters, các permissions và các parameters khác. Node nào tạo chain sẽ có quyền chỉnh sửa file để cho phép các node sau kết nối và cấp quyền cho node đó khi tham gia mạng lưới Multichain.

Cách tạo một chain mới:

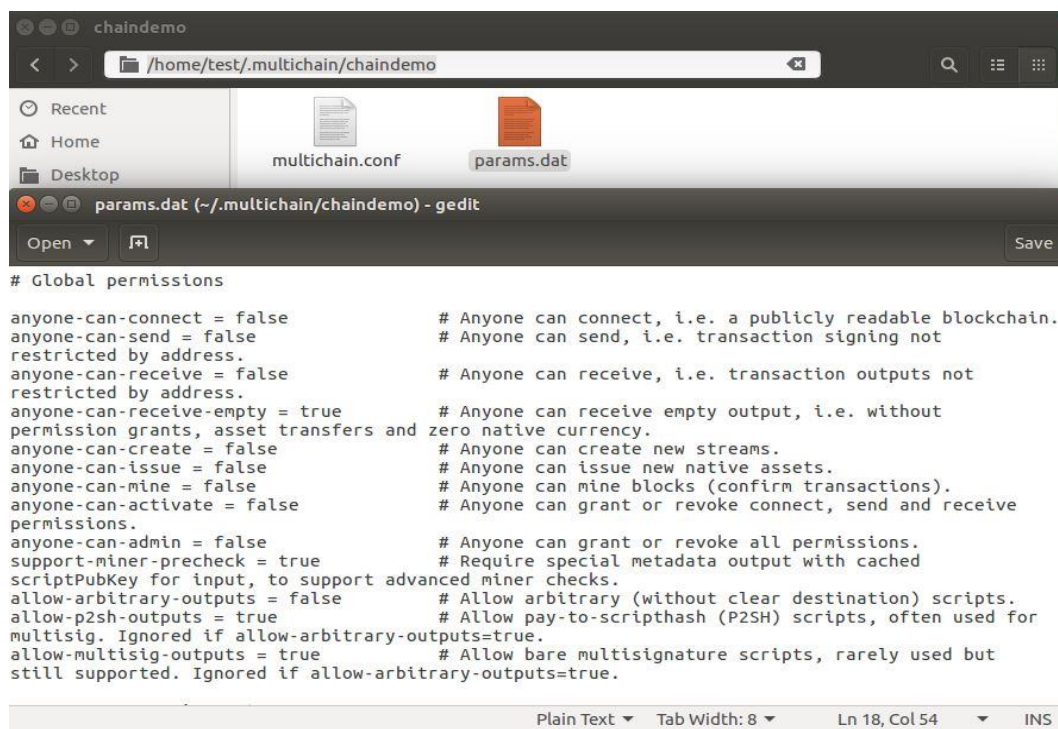
- Bước 1: Chạy command `multichain-util create [tên chain mới]`



```
test@ubuntu:~$ multichain-util create chaindemo
MultiChain 2.0.2 Utilities (latest protocol 20010)
Blockchain parameter set was successfully generated.
You can edit it in /home/test/.multichain/chaindemo/params.dat before running multichaind for the first time.
To generate blockchain please run "multichaind chaindemo -daemon".
```

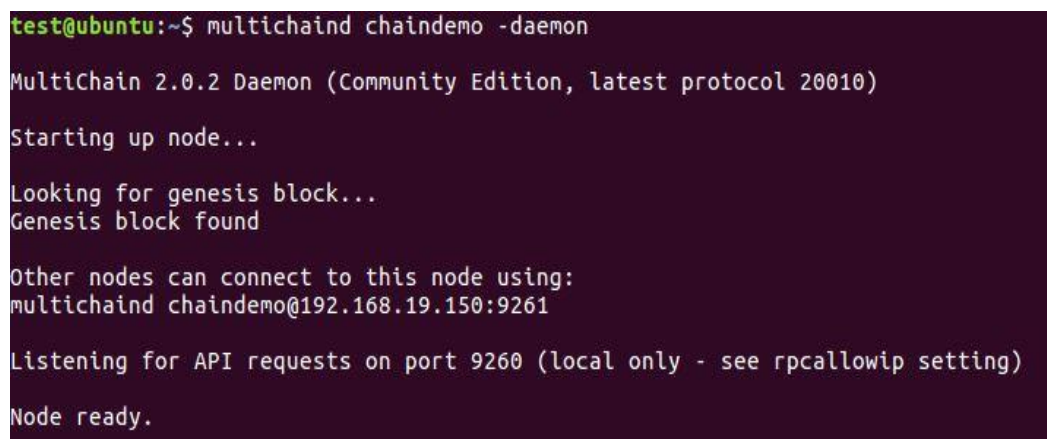
Hình D. 1: Hình minh họa Bước 1 - Tạo chain

- Bước 2: Chỉnh sửa các thông số trong file params.dat cho phù hợp với yêu cầu hệ thống muốn lập.



Hình D. 2: Hình minh họa Bước 2 - Chỉnh sửa thông số

- Bước 3: Chạy command sau: `multichaind [tên chain] -daemon`. Sau khi chạy dòng này Multichain sẽ tạo các genesis block - block khởi nguyên để tạo tiền đề sinh các block sau này, sau đó sẽ mở 2 port: peer-to-peer connect và incoming JSON-RPC API requests.



Hình D. 3: Hình minh họa Bước 3 - Khởi tạo block, sẵn sàng kết nối

- Bước 4: Node thứ hai muốn gia nhập vào chain thì gọi lệnh:

multichaind [tên chain]@[ip-address của node tạo chain]:[port connect peer-to-peer của node tạo chain]. Sau đó nếu thành công ta sẽ được cấp một địa chỉ wallet riêng.

```
node@ubuntu:~/multichain-2.0.2$ multichaind chaindemo@192.168.19.150:2663
MultiChain 2.0.2 Daemon (Community Edition, latest protocol 20010)
Retrieving blockchain parameters from the seed node 192.168.19.150:2663 ...
Blockchain successfully initialized.

Please ask blockchain admin or user having activate permission to let you connect and/or transact:
multichain-cli chaindemo grant 15UKENYLfcSfhQCiPiScLYRzbnaL1vSKiMtJPE connect
multichain-cli chaindemo grant 15UKENYLfcSfhQCiPiScLYRzbnaL1vSKiMtJPE connect,send,receive
```

Hình D. 4: Hình minh họa Bước 4 - Node kết nối vào chain

- Bước 5: Tuy nhiên node thứ hai vẫn chưa có permission để connect đồng bộ với chain, nên node thứ nhất cần cấp quyền cho node thứ hai thông qua lệnh: *multichain-cli [tên chain] grant [địa chỉ wallet của node thứ hai] [quyền muốn được cấp cho]*. [quyền muốn được cấp cho] có thể là một quyền *connect* hoặc list các quyền mong muốn (cách nhau bằng dấu phẩy) như sau: *connect,send,receive*

```
test@ubuntu:~$ multichain-cli chaindemo grant 15UKENYLfcSfhQCiPiScLYRzbnaL1vSKiMtJPE connect,send,receive
{"method": "grant", "params": ["15UKENYLfcSfhQCiPiScLYRzbnaL1vSKiMtJPE", "connect,send,receive"], "id": "45070797-1560838989", "chain_name": "chaindemo"}
63327cb6f76ce6cc0de6d8e96217644e640eb876384adbe3203596c79c41b8d0
```

Hình D. 5: Hình minh họa Bước 5 - Cấp quyền cho node

```
node@ubuntu:~/multichain-2.0.2$ multichaind chaindemo@192.168.19.150:9261
MultiChain 2.0.2 Daemon (Community Edition, latest protocol 20010)
Retrieving blockchain parameters from the seed node 192.168.19.150:9261 ...
Other nodes can connect to this node using:
multichaind chaindemo@192.168.19.149:9261

Listening for API requests on port 9260 (local only - see rpcallowip setting)
Node ready.
```

Hình D. 6: Hình minh họa Bước 5 - Node kết nối thành công

Sau khi kết nối và được cấp quyền, node thực hiện các JSON-RPC API commands thông qua lệnh *multichain-cli*. List các lệnh được Multichain hỗ trợ chi tiết ở [11]. Các lệnh này sẽ cập nhật và thay đổi theo từng phiên bản Multichain.

Multichain hỗ trợ phiên bản web interface đơn giản, sử dụng PHP làm front-end [15]. Khi sử dụng phiên bản này có hạn chế một số lệnh nên phiên bản chỉ có tác dụng cho người sử dụng khái quát về các thao tác cơ bản trên Multichain.

D.5. Các thông số Blockchain Parameters

Các thông số này giúp cấu hình hệ thống, phân quyền người dùng, tùy chỉnh cơ chế đồng thuận cho phù hợp với mạng lưới Private Blockchain mình muốn tạo.

File lưu các thông số là file *params.dat*.

Các thông số chain parameters cơ bản:

| Tên thông số | Mô tả | Giá trị mặc định |
|---------------------|---|------------------|
| chain-protocol | multichain hoặc bitcoin | multichain |
| chain-description | Mô tả về hệ thống chain | |
| root-stream-name | tên root stream khởi nguồn | root |
| root-stream-open | cho phép mọi người ghi lên steam root | true |
| chain-is-testnet | false/true để đặt testnet cho giá trị output của JSON-RPC API calls | false |
| target-block-time | (giây) thời gian xác nhận giao dịch cho các khối | 15 |
| maximum-block-size | Giá trị theo byte lớn nhất của block | 8388608 (8MB) |
| maximum-chunk-size | Giá trị theo byte lớn nhất của chunk | 1048576 |
| maximum-chunk-count | Số chunk có thể tạo cho một single off-chain item. | 1024 |

Bảng D. 1: Chain Parameters cơ bản trong Multichain [2]

Các thông số Global Permissions cơ bản:

| Tên quyền hạn | Mô tả | Giá trị mặc định |
|--------------------------|---|------------------|
| anyone-can-connect | ai cũng có quyền connect | false |
| anyone-can-send | ai cũng có quyền send | false |
| anyone-can-receive | ai cũng có quyền receive | false |
| anyone-can-receive-empty | ai cũng có quyền receive giao dịch không chứa tiền (dùng để truyền địa chỉ) | true |
| anyone-can-create | ai cũng có quyền create | false |
| anyone-can-issue | ai cũng có quyền issue | false |
| anyone-can-mine | ai cũng có quyền mine | false |
| anyone-can-activate | ai cũng có quyền activate | false |
| anyone-can-admin | ai cũng có quyền admin | false |

Bảng D. 2: Global Permissions cơ bản trong Multichain [\[2\]](#)

Ngoài ra còn có các thông số khác, tham khảo thêm tại [\[2\]](#)

PHỤ LỤC E

Cài đặt và cấu hình Storj

E.1. Chuẩn bị môi trường cài đặt storj

Các bước chuẩn bị:

Cài đặt GO phiên bản từ 1.11 trở lên (trong phần hướng dẫn cài đặt go 1.12)

- Link download: <https://golang.org/doc/install?download=go1.12.5.linux-amd64.tar.gz> (Nếu đã cài đặt phiên bản Go thấp hơn 1.11 thì cần phải gỡ và cài đặt lại phiên bản mới hơn.)
- Sau khi tải file cài đặt ta mở terminal và thực hiện các bước sau:
 - `cd <đường dẫn của file cài đặt>`
 - `sudo tar -C /usr/local -xzf go1.12.5.linux-amd64.tar.gz`
 - `export PATH=$PATH:/usr/local/go/bin`

Cài đặt đặt GitHub phiên bản mới nhất theo đường dẫn: <https://git-scm.com/downloads>

E.2. Cấu hình storj

Mở terminal và thực hiện lệnh sau:

- `git clone https://github.com/storj/storj.git storj`
- `cd storj`
- `make install-sim`

Note: các câu lệnh trên được dùng để cài đặt các chương trình storj-sim, satellite, storage node, gateway và uplink trong thư mục được cài đặt Go. Đường dẫn mặc định là ~/go/bin.

Thiết lập môi trường storj-sim (test net), mở terminal và thực hiện lệnh sau:

- `export PATH=~/go/bin:$PATH`
- `cd ~/go/bin`
- `storj-sim network setup <tham số kèm theo>`

- Các tham số kèm theo khi cấu hình storj-sim (có thể có hoặc không)
- `--config-dir <đường dẫn>`: dùng để cài đặt đường dẫn thư mục chứa các config của storj-sim (mặc định là `~/local/share/storj/local-network`)
- `--dev`: sử dụng cấu hình đã được điều chỉnh dành riêng cho nhà phát triển.
- `--host <địa chỉ>`: cấu hình địa chỉ máy chủ của hệ thống (mặc định là “127.0.0.1”, nếu muốn kết nối giữa các máy trong mạng nội bộ thì đổi thành địa chỉ ip của máy chủ).
- `--identities <số nguyên>`: số lượng định danh được tạo bởi hệ thống.
- `-x` hoặc `--print-commands`: hiện các dòng lệnh khi chạy hệ thống storj.
- `--satellites <số nguyên>`: số lượng vệ tinh trong hệ thống (mặc định là 1).
- `--storage-nodes <số nguyên>`: số lượng node dùng để lưu trữ trong mạng storj (mặc định là 10).

Để xem cấu hình hiện tại của mạng storj ta gõ lệnh sau:

- `./storj-sim network env`

Khởi chạy storj-sim

- **storj-sim network run** (satellite sẽ được khởi chạy và kết nối với 10 storage nodes được cấu hình mặc định).
- Ngoài ra, bạn có thể thực hiện câu lệnh **storj-sim -x network run** để xem cụ thể các quá trình thay đổi và các hành vi của hệ thống trên cửa sổ dòng lệnh terminal.
- Gateways khởi động từ port 9000 (port 9000 cho gateway thứ nhất, port 9001 cho gateway thứ hai,...)
- Satellites khởi động từ port 10000 (port 10000 cho satellite thứ nhất, port 10001 cho satellite thứ hai, ...)
- Storage nodes khởi động từ port 11000 (port 11000 cho storage node thứ nhất, port 11001 cho storage node thứ hai,...).

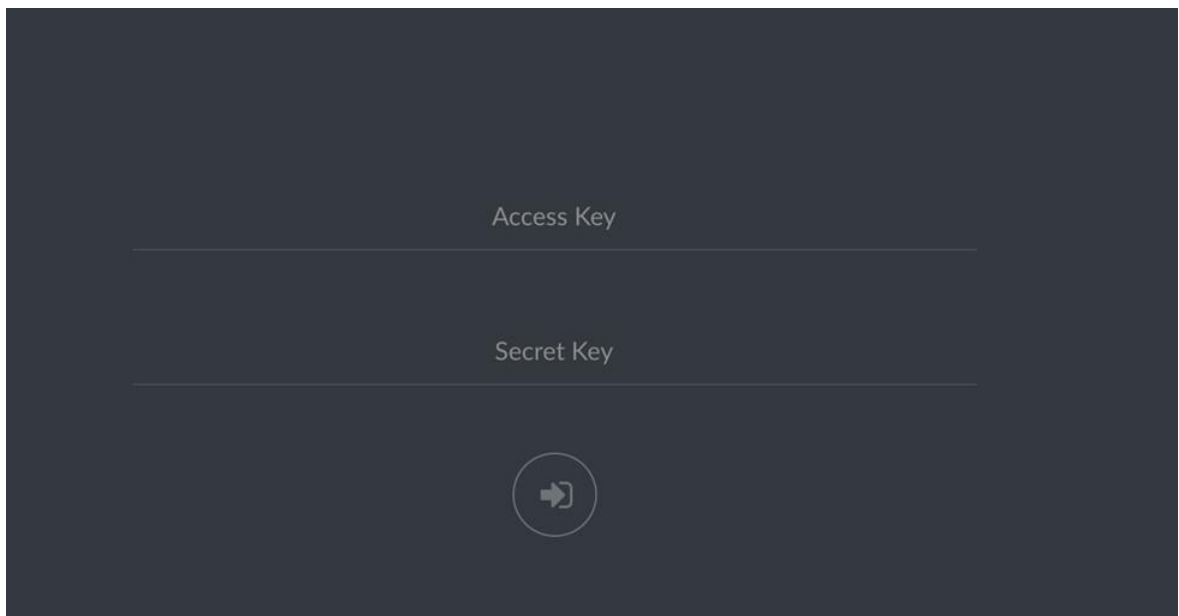
Sử dụng giao diện Minio:

- Sao chép và dán địa chỉ của gateway với nhãn là Endpoint trên terminal vào trình duyệt để sử dụng Minio (mặc định 127.0.0.1:9000)

- Access và secret key được thể hiện ngay bên dưới địa chỉ Endpoint khi chạy `./storj-sim network run` và bạn sử dụng hai key đó để truy cập vào Minio

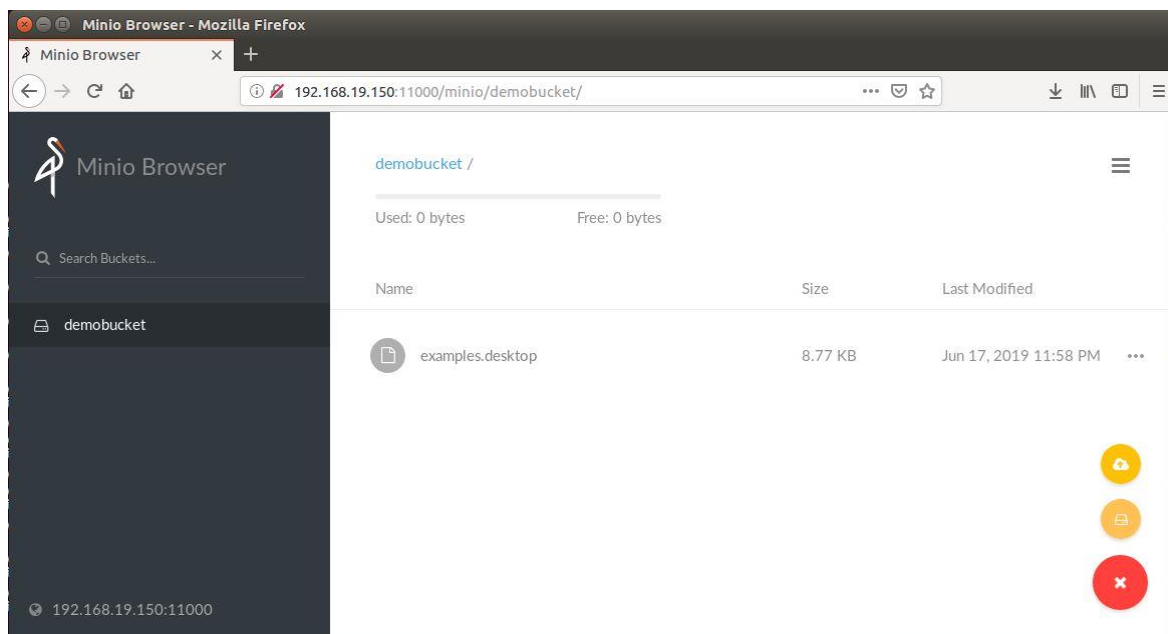
```
| Endpoint: 192.168.19.150:11000  
| Access key: 4QQqckEAU6dGQtLvtY3Cq4Hkf4Vh  
| Secret key: 2PDs7SZrxnAXCw1sTTe6p86tb4B
```

Hình E. 1: Hình minh họa - Access key và Secret key



Hình E. 2: Giao diện khởi động của Minio

- Thông qua Minio, bạn có thể tạo buckets, đăng tải các đối tượng (tập tin, video,...) lên chính buckets đó. Đồng thời còn có thể tạo một liên kết chia sẻ với cấu hình thời gian tồn tại cho các đối tượng đã được đăng tải lên.



Hình E. 3: Hình minh họa sử dụng Minio tải lên tập tin

E.3. Cấu hình Uplink

Uplink được dùng để thực hiện các quá trình lưu trữ và truy xuất dữ liệu trên mạng lưới storj. Tất cả các dữ liệu được lưu trữ hoặc truy xuất đều được mã hoá và giải mã để đảm bảo sự bảo mật cho dữ liệu.

Uplink cần giao tiếp với Storj Satellite bootstrapped trong mạng lưới storj-sim để thực hiện các thao tác truy xuất dữ liệu.

Yêu cầu: đã cài đặt thành công Storj như trên hướng dẫn.

Để thiết lập Uplink ta mở một terminal mới và thực hiện các bước sau:

- `cd ~/go/bin`
- `uplink setup --satellite-addr [địa chỉ server] --enc-key [securekey]`
- Thay thế [securekey] bằng mã số bảo mật của bạn.
- Các thông số kèm theo khi cấu hình Uplink
 - `--api-key <key>`: xác thực api key được sử dụng cho việc kết nối với vệ tinh (satellite).
 - `--client.max-inline-size <kích thước>`:

- `--client.segment-size <kích thước>`: Kích thước tối đa của một segment (mặc định là 64 MiB).
- `--enc.block-size <kích thước>`: kích thước (bytes) cho mỗi khối mã hoá (mặc định là 1.0 KiB)
- `enc.data-type <1 hoặc 2>`: thuật toán được sử dụng để mã hoá nội dung của dữ liệu (1 là thuật toán AES-GCM, 2 là SecretBox, mặc định sẽ là 1).
- `--enc.key <chuỗi mã hoá>`: chuỗi mã hoá được dùng để mã hoá dữ liệu.
- `--enc.path-type <0,1,2>`: loại thuật toán được dùng để mã hoá đường dẫn (0: không mã hoá, 1: AES-GCM, 2: SecretBox, mặc định là 1)
- `--identity.cert-path <đường dẫn>`: đường dẫn của file `identity.cert` (đường dẫn mặc định: `“~/.local/share/storj/identity/uplink/identity.cert”`)
- `--identity.key-path <đường dẫn>`: đường dẫn của file `identity.key` (đường dẫn mặc định: `“~/.local/share/storj/identity/uplink/identity.key”`)
- `--satellite-addr <địa chỉ>`: địa chỉ của vệ tinh (satellite). Mặc định `“127.0.0.1:7777”`.
- `--tls.peer-id-versions <phiên bản>`: phiên bản xác thực có thể hỗ trợ được để kết nối với máy chủ (mặc định là `“lasted”`)
- `--config-dir <đường dẫn>`: Đường dẫn của thư mục chứa cấu hình Uplink. Đường dẫn mặc định là `“~/.local/share/storj/uplink”`
- `--dev`: sử dụng cấu hình thử nghiệm và đang phát triển của hệ thống.
- `identity-dir <đường dẫn>`: xác định đường dẫn của thư mục chứa các tập tin định danh (mặc định là `“~/.local/share/storj/identity/uplink”`)

Các lệnh được dùng để thao tác với dữ liệu dựa trên Uplink

- Kiểm tra và liệt kê các bucket đã tồn tại thông qua câu lệnh: `$ uplink ls`

- Tạo mới một bucket và đăng tải dữ liệu lên bucket đó thông qua uplink ta thực hiện các câu lệnh sau:
 - `uplink mb sj://[bucket-name]`
 - `uplink cp ~/Desktop/[file-name.extension] sj://[bucket-name]`
 - `uplink ls sj://[bucket-name]/`
 - `uplink cp sj://[bucket-name]/[file-name.extension] ~/Desktop/[file-name.extension]`
 - `uplink rm sj://bucket-name/file-name.extension`
 - `uplink rb sj://bucket-name`