Factom Data Structure

Các cấu trúc dữ liệu trong Factom

Building Blocks

Chain Name

- Chain Name là giá trị xác định duy nhất một chuỗi (Chain).
- Chain Name có thể là một số ngẫu nhiên, một chuỗi các kí tự, một public key hoặc hash của một vài đường dẫn thư mục riêng tư.
- Việc lựa chọn Chain Name thuộc về user.
- Chain Name có thể được xác định bởi nhiều chuỗi bytes liên tiếp nhau. Các chuỗi bytes là những phân đoạn dữ liệu khác nhau thay vì ghép nối, để phân biệt các bytes theo sau của một đoạn với các bytes dẫn đầu của đoạn tiếp theo.
- Chiều dài mỗi đoạn (Chain Name Segments) được giới hạn bởi số lượng bytes có thể khít với Entry, xấp xỉ 10 KiB.
- Mỗi phần tử của Chain Name dài ít nhất 1 byte.
- Một Chain Name không có phần tử nào là trường hợp đặt biệt của ChainID có gía trị hash của một chuỗi giá trị null, ví dụ `e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855`.

ChainID

- ChainID là dãy (series) các giá trị hash (băm) hàm SHA256 của các đoạn Chain Name.
- ChainID dài 32 bytes.
- ChainID phải là giá trị hash một tập dữ liệu mờ (opaque data) trong cấu trúc cấp cao của block, tức mỗi ChainID phải lưu trữ thông tin gì đó được che giấu.
- Thuật toán hash mỗi đoạn của Chain Name. Các trị hash này được nối lại và hash thêm lần nữa thành giá trị 32 bytes duy nhất. (SHA256 trả về kết quả là chuỗi giá trị 32 bytes).
- Tạo một ChainID từ một Chain Name chỉ có một đoạn tương đương với việc lấy giá trị hash Chain Name đó 2 lần.
- Công thức tìm ChainID:

```
ChainID = SHA256( SHA256(Name[0]) | SHA256(Name[1]) | ... | SHA256(Name[X]) )
```

trong đó:

- `Name[0..X]` là Chain Name của ChainID cần tìm.
- `X+1` là số lượng đoạn của Chain Name.
- `SHA256(str)` là giá trị hash SHA256 của một chuỗi `str`.

User Elements

Dưới đây là các phần tử thuộc về User, tức do User quyết định.

Entry

- Entry là phần tử mang dữ liệu của user (carries user data).
- Entry Reveal là bản chất (essentially) của dữ liệu này.
- External IDs (ExtIDs) là một phần của Entry, có chiều dài chuỗi bytes được sự kiểm soát của Factom.
- Entry đầu tiên trong một chuỗi Factom (Factom Chain) sử dụng External IDs (ExtID) để định nghĩa Chain Name. Những Entries khác sử dụng External IDs khi ứng dụng của nó chỉ định.
- Để hợp lệ, External IDs phải được phân tích cú pháp và phần cuối của phần tử cuối cùng phải khớp với độ dài đã được xác định cho các External IDs như đã được định nghĩa trong header.
- ExtID hướng đến việc cho ta các thông tin đã được gán nhãn, hay phân loại cho các Entries sao cho các ứng dụng cảm thấy hữu ích. Tức mỗi entry, ta gán các tag liên quan đến entry đó để dặt cho các ExtID. Thông thường, các khoá cơ sở dữ liệu là một cách sử dụng phổ biến.
- ExtID là các dữ liệu đơn giản trong Factom và không được sử dụng trong thuật toán đồng thuận sau entry đầu tiên.
- ExtID và Content không được kiểm tra sự hợp lệ, chỉ có góc nhìn như giá trị nhị phân.

Data	**Field Name**	**Description**
Header		
varInt_F	Version	Bắt đầu bằng 0, những con số cao hơn bị từ chối. Có thể được mã hóa an toàn bằng cách sử dụng 1 byte cho 127 phiên bản đầu tiên (1 bytes có 2^8 = 128 phiên bản).
32 bytes	ChainID	
2 bytes	ExtIDs Size	Số lượng bytes cho tập các ExtIDs. `<= payload size.` Big Endian.
Payload		Dữ liệu giữa phần cuối của header và phần cuối của content.
External IDs		Chỉ được thông dịch và thực thi khi `ExtIDs Size > 0`
2 bytes	ExtID element 0 length	Số lượng bytes theo sau được thông dịch như 1 External ID element. > 0.
variable	ExtID 0	data cho External ID đầu tiên.
2 bytes	ExtID X	kích thước của External ID thứ X.
variable	ExtID X data	Phần tử thứ X. Byte cuối của phần tử cuối phải nằm trên byte cuối của ExtIDs Size trong header.
Content		
variable	Entry Data	Dữ liệu của user. Không được cấu trúc.

- `Empty Entry Size >= 35 bytes`

```
00 954d5a49fd70d9b8bcdb35d252267829957f7ef7fa6c74f88419bdc5e82209f4 0006
0004 74657374 5061796c6f616448657265
00 954d5a49fd70d9b8bcdb35d252267829957f7ef7fa6c74f88419bdc5e82209f4 0000
5061796c6f616448657265
00 954d5a49fd70d9b8bcdb35d252267829957f7ef7fa6c74f88419bdc5e82209f4 0007
0005 48656c6c6f 5061796c6f616448657265
```

Entry Hash

- Entry Hash là giá trị hash (băm) định danh duy nhất cho một Entry data, dài 32 bytes. Nó được tham chiếu trong Entry Block body cũng như trong Entry Commit.
- Trong quá trình sinh Entry Hash có sử dụng thuật toán SHA512 để tránh các nguy cơ bị tấn công và tăng tính bảo mật.
- SHA256 được sử dụng trực tiếp cho merkel root key nhằm gia tăng tốc độ phần cứng CPU, cũng như được kiểm tra kép bằng chuỗi hash.
- Entries (không như blocks), không có 2 cách độc lập để bảo vệ dữ liệu hash, nên đây là lý do mà việc hashing phức tạp hơn.
- Để tính được giá trị entry hash, đầu tiên Entry được serialized (tuần tự hóa) và được đưa vào hàm SHA512. 64 bytes được tạo ra từ hàm hash SHA512 được prepended (nối vào đầu chuỗi) vào Entry đã được tuần tự hóa. Kết quả sau khi Entry đã được tuần tự hóa và nối dài giá trị hàm hash trên sau đó được đưa qua hàm SHA256, kết quả cho ta giá trị Entry Hash.

```
Entry Hash = SHA256( SHA512(Entry) | Entry )
```

trong đó:

- `SHA512(str)` là giá trị hàm hash SHA512, trả về 64 bytes.
- `SHA256(str)` là giá trị hàm hash SHA256, trả về 32 bytes.
- Example:
  - Entry (Entry without ExtID): `00954d5a49fd70d9b8bcdb35d252267829957f7ef7fa6c74f88419bdc5e82209f4000050061796c6f616448657265`
  - Entry is passed into `SHA512`: `0ba3c58955c69b02aa675d8ff15b505a48335fdc9a06354ba55e4149f77b69835c8c2b7002ca3b09202846d03626bada6b408fa1374f22dc396c64d9a3980ed3`
  - The Entry is appended to the `SHA512` result: `0ba3c58955c69b02aa675d8ff15b505a48335fdc9a06354ba55e4149f77b69835c8c2b7002ca3b09202846d03626bada6b408fa1374f22dc396c64d9a3980ed300954d5a49fd70d9b8bcdb35d252267829957f7ef7fa6c74f88419bdc5e82209f400005061796c6f6164486572 65`
  - Entry Hash which hash above with `SHA256`: `72177d733dcd0492066b79c5f3e417aef7f22909674f7dc351ca13b04742bb91`

Entry Commit

- Entry Commit là một payment (thanh toán) cho một entry cụ thể. Nó khấu trừ số dư được giữ bởi khóa công khai cụ thể trong số tiền được chỉ định. Chúng được thu nhập vào Entry Credit (EC) như là bằng chứng cho thấy sự cân bằng (balance) nên được giảm đi.

Data	Field Name	Description
Header		
varInt_F	Version	Bắt đầu bằng 0, những con số cao hơn bị từ chối. Có thể được mã hóa an toàn bằng cách sử dụng 1 byte cho 127 phiên bản đầu tiên (1 bytes có 2^8 = 128 phiên bản).
6 bytes	milliTimestamp	timestamp mà user đã định nghĩa. Giá trị duy nhất cho payment.
32 bytes	Entry Hash	tính toán qua SHA512 + SHA256
1 byte	Number of Entry Credits	Số lượng ECs được trừ từ balance của pubkey. Bất kì giá trị nào > 10 là invalid.
32 bytes	Pubkey	Entry Credit Public Key đã trừ trong balance.
64 bytes	Signature	Chữ kí của entry commit bằng pubkey. Cover từ `Version` tới `Number of Entry Credits`

- Entry commit chỉ valid trong 12 tiếng trước và sau `milliTimestamp`. Bởi vì ECs được dựa trên tính balance, các cuộc tấn công bằng các re-commit có thể làm giảm balances.
- Ngoài ra, một user có thể chi trả cho Entry 2 lần, và có 2 bản copy trong Factom.
- Bởi vì sử dụng mạng P2P, các payments sẽ cần được phân biệt. Các payments được phân biệt via public key và timestamp. Điều này đặt giới hạn 1000 mỗi giây cho 1 EC Pubkey mỗi một Entry trùng lặp. `milliTimestamp` giúp bảo vệ điều này. Thêm time element cho phép các peers tự động từ chối các payments vượt quá 12 giờ. Dẫn đến việc họ phải kiểm tra sự trùng lặp trong khoản thời gian 1 ngày.
- Số lượng Entry Credits dựa trên kích thước khối hàng (Payload size). Chi phí: `1 EC/KiB`. Empty Entry tốn `1 EC`.
- Các entry commit có 2 giá trị hashes liên quan đến nhau. Mỗi giá trị hash là giá trị hàm SHA256 của data:
    - 1 là giá trị hash của entry commit từ Version tới Signature.
    - 2 là giá trị hash của ledger field từ Version tới Number of Entry Credits (không chứa Signature), cũng gọi là `TXID`. XIT không bao gồm chữ kí.

Chain Commit

Data	Field Name	Description
varInt_F	Version	starts at 0. Higher numbers are currently rejected. Can safely be coded using 1 byte for the first 127 versions.
6 bytes	milliTimestamp	This is a timestamp that is user defined. It is a unique value per payment. This is the number of milliseconds since 1970 epoch.

Data	Field Name	Description
32 bytes	ChainID Hash	This is a double hash (SHA256d) of the ChainID which the Entry is in.
32 bytes	Commit Weld	SHA256(SHA256(Entry Hash | ChainID)) This is the double hash (SHA256d) of the Entry Hash concatenated with the ChainID.
32 bytes	Entry Hash	This is the SHA512+256 descriptor of the Entry to be the first in the Chain.
1 byte	Number of Entry Credits	This is the number of Entry Credits which will be deducted from the balance of the public key. Any values above 20 or below 11 are invalid.
32 bytes	Pubkey	This is the Entry Credit public key which will have the balance reduced.
64 bytes	Signature	This is a signature of this Chain Commit by the pubkey. Parts ordered R then S. Signature covers from Version through 'Number of Entry Credits'

Factoid Transaction

Data	Field Name	Description
Header		
varInt_F	Version	Version of the transaction type. Versions other than 2 are not relayed. Can safely be coded using 1 byte for the first 127 versions.
6 bytes	milliTimestamp	Same rules as the Entry Commits. This is a unique value per transaction. This field is the number of milliseconds since 1970 epoch. The Factoid transaction is valid for 24 hours before and after this time.
1 byte	Input Count	This is how many Factoid addresses are being spent from in this transaction.
1 byte	Factoid Output Count	This is how many Factoid addresses are being spent to in this transaction.
1 byte	Entry Credit Purchase Count	This is how many Entry Credit addresses are being spent to in this transaction.
Inputs		
varInt_F	Value	(Input 0) This is how much the Factoshi balance of Input 0 will be decreased by.
32 bytes	Factoid Address	(Input 0) This is an RCD hash which previously had value assigned to it.

Data	Field Name	Description
varInt_F	Value	(Input X) This is how much the Factoshi balance of Input X will be decreased by.
32 bytes	Factoid Address	(Input X) This is an RCD hash which previously had value assigned to it.
Factoid Outputs		
varInt_F	Value	(Output 0) This is how much the Output 0 Factoshi balance will be increased by.
32 bytes	Factoid Address	(Output 0) This is an RCD hash which will have its balance increased.
varInt_F	Value	(Output X) This is how much the Output X Factoshi balance will be increased by.
32 bytes	Factoid Address	(Output X) This is an RCD hash which will have its balance increased.
Entry Credit Purchase		
varInt_F	Value	(Purchase 0) This many Factoshis worth of ECs will be credited to the Entry Credit public key 0.
32 bytes	EC Pubkey	(Purchase 0) This is Entry Credit public key that will have its balance increased.
varInt_F	Value	(Purchase X) This many Factoshis worth of ECs will be credited to the Entry Credit public key X.
32 bytes	EC Pubkey	(Purchase X) This is Entry Credit public key that will have its balance increased.
Redeem Condition Datastructure (RCD) Reveal / Signature Section		
variable	RCD 0	First RCD. It hashes to input 0. The length is dependent on the RCD type, which is in the first byte. There are as many RCDs as there are inputs.
variable	Signature 0	This is the data needed to satisfy RCD 0. It is a signature for type 1, but might be other types of data for later RCD types. Its length is dependent on the RCD type.
variable	RCD X	Xth RCD. It hashes to input X.
variable	Signature X	This is the data needed to satisfy RCD X.