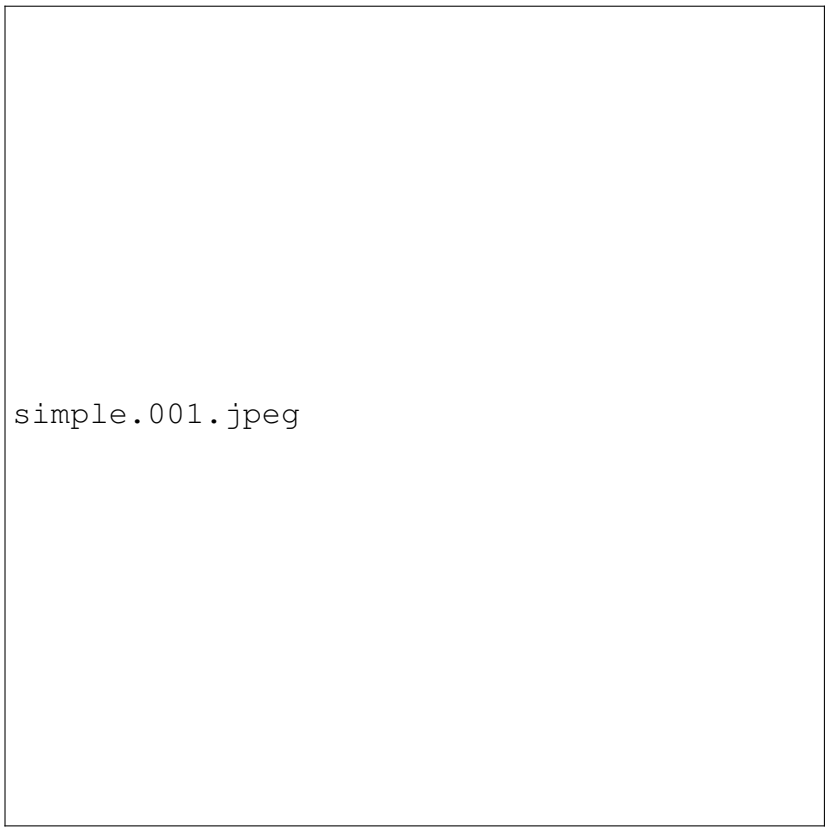
0.1 Ý tưởng cơ bản

Một ý tưởng về một layer off-chain đã xuất hiện từ lâu trong cộng đồng, ý tưởng chính là thay vì thực hiện các giao dịch với các phép tính phức tạp bên trong blockchain network, người ta thực hiện các giao dịch đó bên ngoài blockchain, sau đó bằng một cách nào đó người ta đảm bảo rằng các giao dịch đó là hợp lệ. Người dùng có thể gửi các tài sản số vào các layer này, thực hiện tính toán và rút các tài sản này ra khi cần thiết.

Bằng cách này chi phí thực thi một giao dịch trong Ethereum giảm đi đáng kể, nhưng câu hỏi đặc ra là làm sao ta đảm bảo được cách chúng ta thực hiện các giao dịch là đúng? Thật tuyệt vời zk-SNARK gần như là một công cụ hoàn hảo để làm điều này. Trong mỗi giao dịch ta sinh ra một bằng chứng cho giao dịch đó sau đó gửi bằng chứng đó lên blockchain network, network sẽ có một smart contract có thể kiểm chứng bằng chứng có đúng hay không. Từ đó cập nhật trạng thái mới vào smart contract.



Hình 1: Layer 2 ban đầu

Nhưng ta có thể làm tốt hơn nữa thay vì mỗi lần chỉ xử lý một giao dịch một lúc, tại sao ta không gộp nhiều transaction lại và sinh bằng chứng trong một lần. Ta gọi một tập transaction như vậy là batch. Hơn nửa ta có thể dùng merkle tree để lưu lại trạng thái hiện tại của các account. Bài toán cần xác thực lúc này là kiểm tra việc cập nhật merkle tree. Việc dùng merkle tree ở đây giảm chi phí cho việc kiểm tra và cập nhật trạng thái rất nhiều, điều này tương đồng với cách các blockchain network dùng merkle tree để nét dữ liệu trong block thành merkle root vậy. Trên blockchain network, ta lưu giá trị của merkle root hash trên blockchain. Với một batch ta có thể kiểm chứng việc cập nhật trạng thái, tính toán merkle root mới và cập nhật lại merkle root lưu trên blockchain. Đây là ý tưởng cơ bản của zk-Rollup.

Hình 2: Hình dáng cơ bản của một zk-Rollup layer. (Nguồn https://vitalik.ca)

0.2 Kiến trúc zk-Rollup cho bài toán giao dịch token

Ta đã nắm được ý tưởng cơ bản của zk-Rollup, giờ công việc tiếp theo sẽ là xây dựng một layer 2 dùng zk-Rollup. Trong layer của tôi, tôi chỉ cho phép giao dịch một loại token, hiển nhiên ta có thể mở rộng nó trong tương lai.

Cấu trúc account

Trong phần này ta xem xét tới cấu trúc của account leaf của merkle tree. Trong hình 3.2 ta đã thấy các nút lá(leaf) của merkle tree lưu lại sẽ trạng thái của account. Trong trường hợp này tôi quy định trạng thái account sẽ có schema như sau

```
{
  "layer_index": "chỉ số của account trong layer",
    "public_key_x": "public key",
    "public_key_y": "public key",
    "nonce": "số lượng transaction đã thực hiện",
    "balance": "số token trong tài khoản"
}
```

Trong đó public_key_x, publickey_y là khoá công khai của chữ ký số trong thuật toán EdDSA của tài khoản sở hữu layer_index trong merkle tree, layer_index cho biết số thứ tự của account, đồng thời cũng cho biết vị trí của account trong merkle tree của layer, balance

là số tiền còn lại của account, nonce là số lượng giao dịch mà account đó đã thực hiện trên layer 2.

Ta có giá trị hash của nút leaf được tính như sau:

```
leaf_hash = H(public_key_x, public_key_y, layer_index, balance, nonce
```

Trong đó ta có account có index bằng 0 là một index đặt biệt(Từ đây ta gọi nó là index zero). Nó đại diện cho chính smart contract của layer. Khi có token chuyển từ bên ngoài vào smart contract hay nói đúng hơn là quá trình deposit xảy xa. Ta sẽ quy ước có token chuyển từ index zero về index trong layer của account được tạo đó. Tương tự khi có lệnh withdraw, rút tiền từ layer về Ethereum account, bên trong layer ta quy ước sẽ chuyển token từ account đó về index zero cua layer.

Với chứ ký số ta bảo đảm mọi hành động thay đổi trạng thái điều xuất phát từ người có quyền trên account, mà không phải là một tác nhân xấu nào đó. Điều này đảm bảo tính an toàn cho giao dịch.

0.2.1 Chế thay đổi trạng thái của account từ X sang Y

Thông thường để thay đổi trạng thái của một account ta làm như sau:

- Kiểm tra trạng thái X là trạng thái hiện tại của account là hợp lệ trong merkle tree.
- Cập nhật trạng thái mới Y cho account đó.
- Cập nhật merkle root của merkle tree với trạng thái vừa tạo.

Với zk-SNARK ta hoàn toàn có thể sinh ra bằng chứng cho việc này một các dễ dàng. Việc chúng ta public merkle root đảm bảo rằng không ai có thể làm giả việc cập nhật trạng thái của account.

Cấu trúc transaction

Với transaction ta có cấu trúc như sau.

```
{
    "sender": "layer index of sender",
    "receiver": "layer index of receiver",
    "amount": "amount",
    "sig":{
            "A",
            "R",
            "s"
       }
}
```

Trong đó sender, receiver lần lược là index của sender account, amount là số token bạn muốn gửi, sig là chữ ký của người gửi, để chứng thực cho hành động họ muốn thực hiện. Tuỳ theo loại transaction mà ta quy định các cơ chế riêng để xử lý chúng.

Cơ chế deposit

Ta áp dụng cơ chế này khi có một người dùng muốn gửi token amount vào layer của chúng ta. Có 2 trường hợp xảy:

Trường hợp thứ nhất, tài khoản đã có sẵn trong layer. Lúc này ta chỉ cần thực hiện các công việc như sau:

- Kiểm tra chữ ký của người muốn gửi token vào layer
- Thực hiện transaction chuyển amount token từ index zero tới index của account người gửi trong layer.
- Sinh ra bằng chứng cho các sự việc đó.