경량 블록체인을 위한 경량 해시함수 Spongent의 Javascript 구현

https://youtu.be/PPf5P_hIisk

서론

일반적인 블록체인

제안 시스템

구현

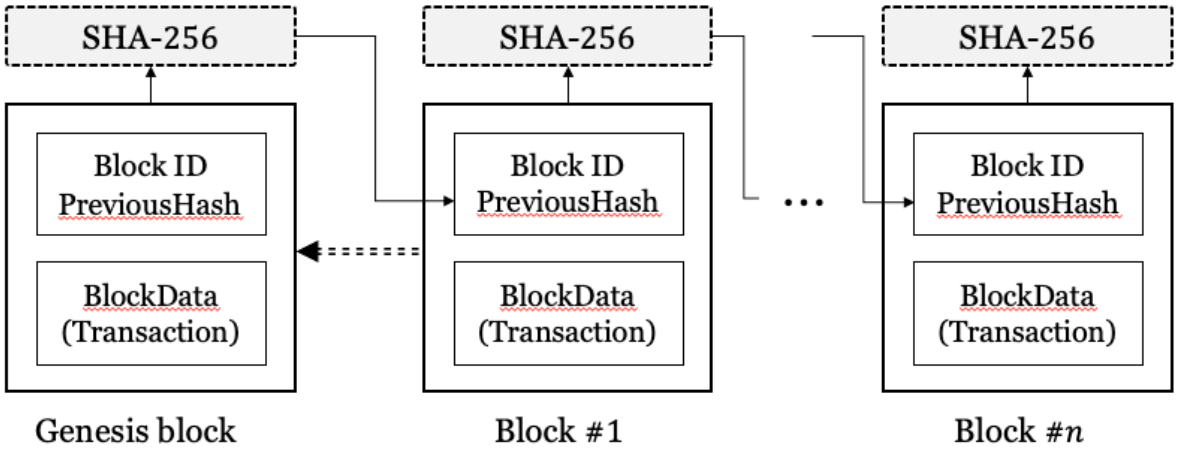
서론

- 강력한 보안 성능을 지니고 있는 블록체인
  → 비교적 보안 성능이 약한 사물인터넷의 문제점을 보완
- 사물인터넷 상에 블록체인을 적용하려는 연구가 다수 이루어지고 있음
- 하지만 사물인터넷에 블록체인을 적용하기 위해서는 블록체인 경량화 필요

일반적인 블록체인 구조

- 해시함수 SHA-256을 통해 해시값 계산

  → SHA-256의 경우 경량 해시함수보다 비용이 많이 듦



General blockchain structure

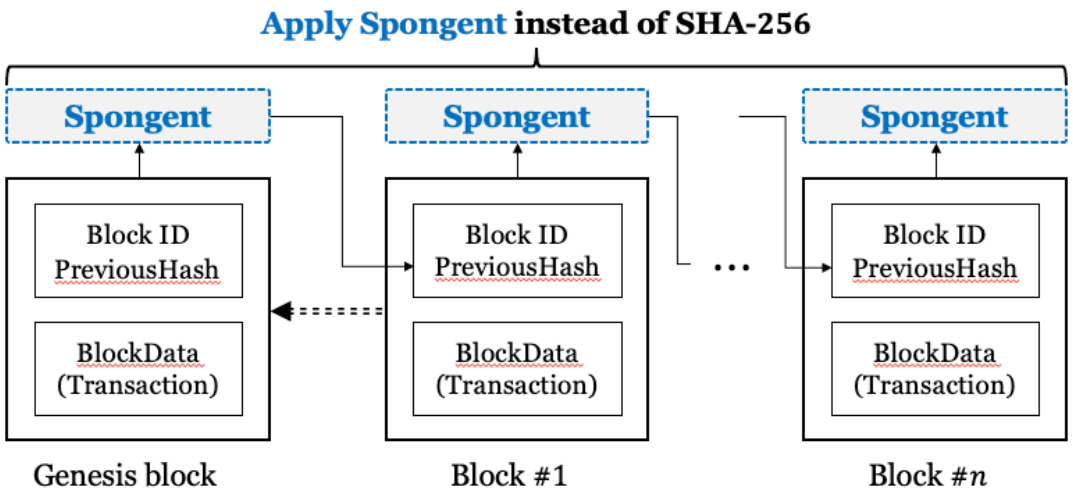
제안 시스템

- SHA-256 해시함수 대신 경량 해시함수인 Spongent 해시함수 사용

  → 경량 해시함수를 사용함으로써 계산 비용 절약

- Spongent가 적용된 블록체인을 프로토타입으로 구현

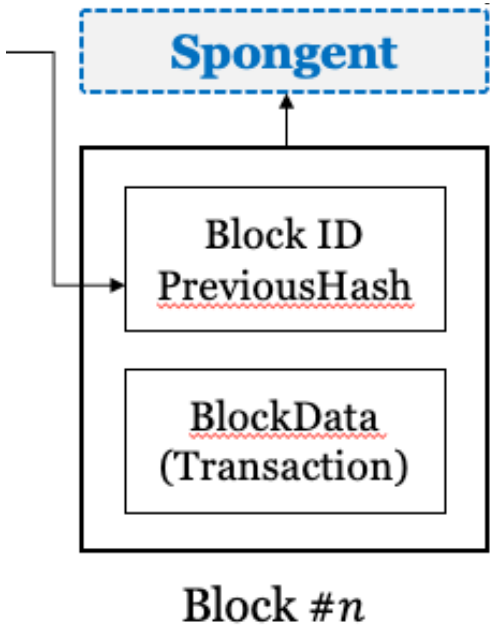
  → Javascript 언어 사용



Proposed blockchain structure

제안 시스템

- 블록체인은 BlockID, PreviousHash, BlockData로 구성
- Genesis 블록의 경우 PreviousHash값이 존재 X
  → 임의로 0으로 초기화
- Genesis 블록의 ID값는 0x0으로 설정
  → 그 후 연결되는 블록의 ID 값은 0x1, 0x2와 같이 1씩 증가
- 해시함수의 입력 값 : BlockId, PreviousHash, BlockData를 연접한 값
- 블록 데이터는 8개의 트랜잭션으로 구성
  → 각 트랜잭션은 랜덤으로 생성



- Javascript에서 사용되는 데이터 타입 Number의 경우 안정적으로 표현할 수 있는 정수의 최대치가 $2^{53} - 1$
  → 해시값 계산시 최대치를 초과하여 오버플로우 발생
  → 이를 방지하고자 BigInt 사용
  → BigInt는 큰 정수를 안전하게 저장하고 연산할 수 있게 해주는 내장 객체

블록체인 출력

- 임의로 블록 3개를 생성하여 블록체인 출력

```
Blocks :  [
  Block {
    BlockID: '0x0',
    previousBlockHash: '0000000000000000000000000000000000000000',
    blockData: [
      [tx], [tx], [tx],
      [tx], [tx], [tx],
      [tx], [tx]
    ]
  },
  Block {
    BlockID: '0x1',
    previousBlockHash: '8bc02f5604c5e2640505589fa894b293acdf651262d3f64f0a8b49da2f057579',
    blockData: [
      [tx], [tx], [tx],
      [tx], [tx], [tx],
      [tx], [tx]
    ]
  },
  Block {
    BlockID: '0x2',
    previousBlockHash: '3ef02972df90de7282d5edc525911a1d94f3984f7cd1d75d8ecd285694d036a5',
    blockData: [
      [tx], [tx], [tx],
      [tx], [tx], [tx],
      [tx], [tx]
    ]
  }
]
```

블록체인 출력

```
=========================
<Genesis Block's Hash>  8bc02f5604c5e2640505589fa894b293acdf651262d3f64f0a8b49da2f057579

<Block1's Hash>         3ef02972df90de7282d5edc525911a1d94f3984f7cd1d75d8ecd285694d036a5

<Block2's Hash>         5b7c916949ad5a0f25c742b5d4901d21bd1a039d19f140270ece3c9dbc3fdfbb
=========================
```

각 블록의 해시값

```
blockData: [
  tx {
    Id: 'TxID #0',
    Data:
    '713a89df71899ebe36b15710cea39fdd8f78cf0960ddf8c2a
    1c1dd4655c37'
  },
  tx {
    Id: 'TxID #1',
    Data:
    '3a23a3eee7a7dc8a3c81fe4206ada1590517657e4f34a71b7
    3731b32aa5d0'
  },
```

트랜잭션 출력 (일부)

실습

Q & A