Modeling and Verification for Bitcoin GHOST(Greedy Heaviest-Observed Sub-Tree) Protocol by SPIN

노 은 방*, 심 우 진*, 김 기 천*

건 국 대 학 교

shdmsqkd123@naver.com, forestzerg@konkuk.ac.kr, kckim@konkuk.ac.kr

요 약

최근 가상화폐를 통한 거래가 증가하면서 Bitcoin에 대한 관심이 증가하고 있다. Bitcoin은 중앙 집중형 시스템을 탈피하여 분산된 시스템에서의 교환을 통해 무결성을 제공하는 블록체인을 활용한 가상화폐이다. 이러한 Bitcoin 거래를 위해서는 생성된 블록체인 중 가장 긴 블록을 선택하여 거래를 수행해야 하는데 정상적인 블록이 아닌 악의적인 블록이 더 길게 생성되는 경우 이중 지불(Double Spending)에 대한 문제점이 발생할 수 있다. 본 논문에서는 이중 지불 공격을 방지하기 위해 트리 형태로 이루어진 GHOST(Greedy Heaviest-Observed Sub-Tree) 비트코인 프로토콜에 대한 모델링을 수행하고 이에 대해 SPIN Model Checker를 통한 정형 검증을 수행하려고 한다.

Ⅰ. 서 론

Bitcoin은 2009년 나카모토 사토시(가명)가 만든 가상화폐로 통화를 발행하고 관리하는 중앙 장치가 존재하지 않는 P2P(Peer to Peer)기반 분산 네트워크 형태를 통해 거래가 이루어진다. Bitcoin 거래를 위한 분산 네트워크 형태를 블록체인이라고 하는데 블록체인은 이러한 분산 네트워크 형태의 공공 거래 장부 역할을 하며, 가상화폐로 거래할 때 발생할 수 있는 위•변조에 대한 문제를 방지할 수 있는 무결성이 보장된 기술이다[1].

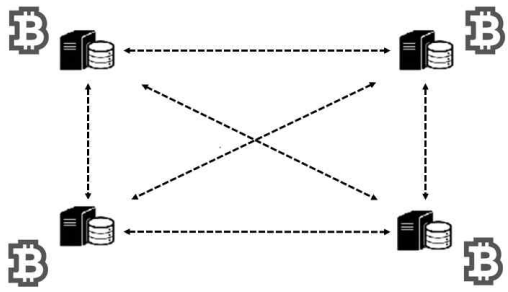
한편 Bitcoin에 대한 지속적인 관심과 함께 다양한 공격 방법 등이 출현하고 있는데 대표적으로 악의적인 블록체인 생성을 통해 Bitcoin에 대한 이중 지불(Double Spending)공격 문제가 있다. 기본적으로 블록체인은 가장 긴 블록 길이를 가지고 있는 체인을 선택하여 거래를 수행하는데 정상적인 블록이 아닌 악의적인 블록이 가장 긴 블록체인의 길이를 가지고 있다면 해당 블록을 블록체인으로 선정하여 이중 지불 문제가 발생하게 된다[2].

따라서 본 논문에서는 이중 지불 문제의 방지를 위해 블록체인을 트리 형태로 구성하여 트리의 가중치를 블록의 길이로 선정해 악의적인 블록체인이 선택되지 않도록 하는 GHOST(Greedy Heaviest-Observed Sub-Tree) 블록체인에 대한 프로토콜 모델링과 이에 대한 정형 검증을 수행하여 이중 지불에 대한 문제점을 방지할 수 있는 방법을 제시하려고 한다.

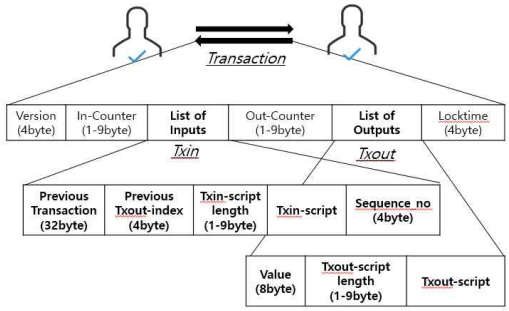
Ⅱ. 관련 정의들

1. 블록체인

블록체인은 각 네트워크 참여자들의 데이터를 수집하여 이를 블록 단위로 구성하고, 해당 블록들을 연결한 후 공유하여 이를 통해 무결성을 보장하고 위•변조가 불가능한 데이터 기술이다. 따라서 블록체인은 분산되고, 독립적이며, 개방된 분산 원장 기술이라고 지칭한다. 기본적으로 블록체인 네트워크는 P2P 네트워크 형태로 이루어져 있으며 블록체인의 네트워크 참여자들은 기본적으로 신뢰할 수 있는 참여자들로 이루어진다. 이를 통해 기존의 중앙 집중형 시스템과 같이 신뢰할 수 있는 기관 또는 서버를 구축할 필요가 없다는 장점을 가지고 있다. 블록체인의 구조는 그림 1과 같은 형태로 이루어져 있다.



〔그림 1〕 Blockchain Network



〔그림 2〕 Transaction Structure

1.1. 블록의 구조

블록체인은 각 네트워크 참여자들이 블록 단위로 구성되어 있다. 블록은 크게 헤더(Header)와 바디(Body)로 구성되어 있으며 헤더는 표 1에 나타낸 항목으로 구성되어 있고, 바디는 데이터를 전송하는 트랜잭션(Transaction) 및 기타 정보로 구성되어 있다 [1].

〔표 1〕 The Structure of Block Header

Name	Value
Previous Block Hash	이전 블록의 해시 값
MerkleRoot Hash	악의적인 데이터 변조가 있는지 검증
Time	블록이 생성된 시간
Bits	난이도 조절 수치
Nonce	최초 0에서 시작하여 조건을 만족하는 해시를 찾아낼 때 1씩 증가하는 계산 횟수

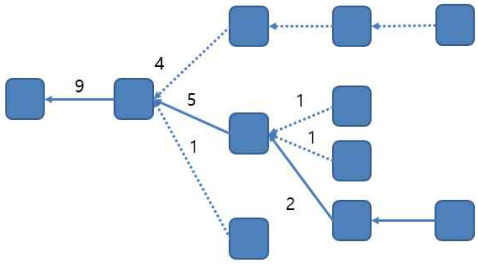
1.2 트랜잭션

트랜잭션은 현재 블록에서 다른 블록으로 데이터를 전송할 때 발생하는 동작 과정이다 [3]. 비트코인 전송 과정에서의 트랜잭션은 블록 간 거래 정보가 공유되었는지를 보여준다. 트랜잭션의 구성은 그림 2와 같다.

트랜잭션은 코드 내의 악의적인 무한 루프 또는 계산 낭비를 방지하기 위해 사용할 수 있는 코드 실행 단계 수를 제한해야 하며 이를 Counter로 정의한다. 또한 모든 트랜잭션에는 입력과 출력 두 가지로 구분하는데 입력 목록에는 이전 트랜잭션에 대한 정보, 이전 출력 값에 대한 index, 다음 블록이 어떻게 접근할 수 있는지 설명하는 정보를 가진 script와 길이 그리고 블록의 시퀀스 번호 등으로 구성되어 있다. 출력 목록에는 출력 값과 script 길이 그리고 출력에 대한 script 등이 포함되어 있다.

1.3 Greedy Heaviest-Observed Sub-Tree

Greedy Heaviest-Observed Sub-Tree는 2013년 12월 Yonatan Sompolinsky와 Aviv Zohar에 의해 등장한 Bitcoin 프로토콜이다 [2]. 블록체인은 블록들이 네트워크를 통해 전파되는데 일정한 시간에 맞게 다음 블록을 생성하며 블록의 길이를 결정한다. 하지만 생성되는 블록들의 블록보다 더 긴 블록이 생성되는 경우 이중 지불 문제가 발생하게 된다. 이를 보완하기 위해 등장한 GHOST 프로토콜은 이러한 블록들을 트리 형태로 생성하여 해당 블록의 부모(parent)블록, 조상(ancestor) 블록 그리고 자손(descendant) 블록까지 블록의 가중치 값을 계산하여 블록의 길이를 정한다 [3]. 이렇게 트리 형태의 블록체인이 형성되면 단일 형태로 생성된 악의적인 블록의 선택을 방지할 수 있고 정보의 무결성을 보장할 수 있을 뿐만 아니라 네트워크 보안 손실을 방지할 수 있다. Greedy Heaviest-Observed Sub-Tree는 그림 3과 같은 형태로 구성되어 있다.



〔그림 3〕 Greedy Heaviest Observed Sub-Tree

Ⅲ. Modeling and Verification

본 논문에서는 GHOST 프로토콜에 대한 모델링 및 검증을 SPIN Model Checker 프로그램을 통해 수행하였다. Bitcoin의 특성 상 시간의 흐름에 맞게 블록이 생성된다는 특징이 있지만 SPIN Model Checker는 시간을 적용할 수 없기 때문에 블록을 생성할 때 초기 시간을 0으로 설정하고 블록을 생성할 시 1씩 증가하도록 적용하였다. 모델링 항목으로는 Peer, Pool, BC Server, Malicious Peer, GHOST Protocol 총 5가지를 모델링했다.

1. Peer

Peer는 블록체인 네트워크에 참여한 블록들로 정의하였다. 각 Peer는 서로 다른 Peer와 트랜잭션을 일으킬 수 있는데, 트랜잭션 발생 이전에 그림 2에서 정의한 항목들과 같이 이전 Peer의 블록과 비교한 후 비교한 항목이 일치한다면 트랜잭션이 발생하도록 모델링 하였다. 모델링한 트랜잭션의 데이터 구조는 표 2와 같다.

〔표 2〕 Data Structure of Transaction

Transaction Datatype in SPIN

```
typedef TXOUT {
    byte address;
    byte value;
};
typedef TXIN {
    byte id;
};
typedef TX{
    byte ID;
    byte time;
    TXIN TxInput;
    TXOUT TxOutput;
};
```

2. Pool

Pool는 블록 간 트랜잭션 발생 시 비교한 트랜잭션 값이 일치하면 비트코인을 채굴하는 모델로 정의하였다. 채굴을 수행하기 위해서는 채굴하는 Bitcoin 값보다 난이도가 같거나 적어야 Bitcoin이 채굴된다. 하지만 본 논문에서는 SPIN이 가지고 있는 State Explosion 특성을 고려하여 채굴 난이도와 Bitcoin 값이 같을 때만 채굴을 수행하도록 설정하였다. pool에 대한 데이터 구조는 표 3에서 정의하였다.

〔표 3〕 Data Structure of Pool

Pool Data type in SPIN

```
typedef BC {
    byte cur;
    byte prev;
    byte length;
    byte time'
};
byte mining;
BC recvBlock;
BC sendBlock;
byte curChain = 0;
```

3. BC Server

BC Server는 Bitcoin을 채굴할 때 난이도를 결정해준다. 앞서 정의한 pool에서의 채굴 값이 BC Server의 난이도와 일치할 경우 채굴을 수행한다고 정의하였고 BC Server에서는 이러한 난이도에 대한 값을 가지고 있다. 초기 난이도 값은 중간 값으로 설정하였고 pool와 채널 간 통신을 수행하여 난이도 값이 일치하는지 확인한 후 BC Server에서 난이도 값을 다시 비교하도록 모델링하였다. BC Server에 대한 데이터 구조는 표 4와 같다.

표 4) Data Structure of BC ServerPool

BC Server Data type in SPIN

```
byte SOLUTION;
chan SolutionToServer;
chan SolutionToFind;
```

4. Malicious Peer

Malicious Peer는 앞서 3.1에서 정의한 Peer의 구조와 비슷하다. 단 Peer에서는 트랜잭션에 대한 비교 및 검증을 수행하고 전송을 하지만 Malicious Peer는 비교 및 검증을 하지 않고 바로 트랜잭션을 전송시키는 특징을 가지고 있다. 이를 수행할 시 트랜잭션 상태에 대한 정의가 바뀌어 Malicious Peer에 대한 트랜잭션이 일어났다고 정의하였다. 상태를 정의한 변수를 표 5에서 정의하였고, Malicious Peer가 발생하면 INVALID상태가 되었다고 정의하였다.

〔표 5〕 Data Structure of Transaction Status

Transaction Status Data type in SPIN
mtype = {UNCONFIRMED, CONFIRMED, INVALID}; //트랜잭션의 상태를 정의, Malicious Peer가 전송되면 INVALID로 표시

5. GHOST Protocol

1.3에서 정의한 대로 GHOST Protocol은 처음 시작하는 블록부터 트리로 구성된 최종 블록까지의 가중치를 합산한 후 블록의 길이를 계산한다. 본 논문에서는 현재 GHOST 체인의 블록과 트리로 구성된 블록들의 최종 가중치를 변수로 설정하여 합산하였고 최종 합산 값 또한 변수로 설정하여 GHOST Protocol을 모델링하였다. GHOST Protocol을 표현하기 위한 데이터 구조는 표 6과 같다.

〔표 5〕 Data Structure of Transaction Status

GHOST Protocol Data type in SPIN
byte temp_GHOST[i]; byte longestChain; byte longestChain_GHOST;

6. Property Verification

GHOST Protocol을 사용하는 목적은 일반적으로 형성되는 블록체인의 길이보다 긴 가중치의 값이 나와야 한다. 본 논문에서는 GHOST Protocol에 대해 정의한 변수인 longestChain_GHOST 값이 longestChain보다 길이가 길면 된다는 것을 SPIN의 LTL Property로 설정하여 모델링 검증을 수행하였다. Property 구문은 다음과 같다. 실제 수행 결과 검증이 수행되었음을 확인하였다.

LTL p1 {[] <> (longestChain < longestChain_GHOST)} -> True

Ⅳ. 결 론

본 논문에서는 악의적으로 생성되는 블록체인으로 인해 Bitcoin에서 발생하는 이중 지불 문제를 해결하기 위해 트리 형태로 가중치를 합산하여 블록의 길이를 결정하는 모델인 GHOST 프로토콜에 대한 모델링 및 검증을 수행하였다. 향후 블록체인에서 필요한 시간에 대한 모델링을 수행하기 위해서는 본 논문에서 사용한 SPIN Model Checker 이외에 Timed Automata 기능을 갖고 있는 Model Checker를 통해 모델링을 수행하여 일정한 시간에 맞게 트랜잭션이 발생하고 블록이 형성되는 것을 확인할 수 있도록 하는 검증이 필요하고 이를 추후 연구과제로 남긴다.

참 고 문 헌

〔1〕 K.Chaudhary, A.Fehnker, J.Van de Pol, "Modeling and Verification of the Bitcoin Protocol", *arXiv preprint arXiv : 1511.04173*, 2015.

〔2〕 Y.Sompolinsky, A.Zohar, "Secure High-Rate Transaction Processing in Bitcoin", *FC*, 2015, 30.

〔3〕 M. Andrychowicz, S. Dziembowski, D. Malinowski and L. Mazurek. "Modeling bitcoin contracts by timed automata" *CoRR*, abs/1405.1861, 2014.