Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.2, Apr. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.2.377

분산 원장을 이용한 토큰 기반 사물 인터넷 접근 제어 기술*

박 환,† 김 미 선, 서 재 현‡
목포대학교 정보보호학과

Token-Based IoT Access Control Using Distributed Ledger*

Hwan Park,† Mi-sun Kim, Jae-hyun Seo‡
Division of Information Security, Mokpo National University

요 약

최근 사물 인터넷에서 인증, 접근 제어 등을 위해 토큰과 블록체인을 이용한 시스템 연구들이 국내외에서 진행되고 있다. 그러나 기존 토큰을 이용한 방식의 시스템은 중앙 집중적인 특성을 가지고 있으므로 보안성, 신뢰성, 확장성 측면에서 사물 인터넷과는 적합하지 않다. 또한 블록체인을 이용한 방식의 시스템은 블록체인을 유지하기 위해서 해시 등의 계산을 반복적으로 수행하고 모든 블록을 저장해야하므로 IoT 디바이스에 과부하가 따른다. 본 논문에서는 사물 인터넷에 적합한 접근 제어를 위하여 토큰을 기반으로 권한 관리를 한다. 또한 탱글을 적용한 P2P 분산 원장 네트워크 환경을 구성하여 중앙 집중적인 구조의 문제점을 해결하고 토큰을 관리한다. 인증 과정과 접근 권한 부여 과정을 수행하여 토큰을 발급하고 토큰 발급에 대한 트랜잭션을 공유하여 모든 노드들이 토큰에 대한 유효성을 검증할 수 있다. 기존 발급 받은 토큰을 재사용하여 반복적인 인증 과정과 접근 권한 부여 과정을 줄여서 접근 제어 프로세스를 경량화할 수 있다.

ABSTRACT

Recently, system studies using tokens and block chains for authentication, access control, etc in IoT environment have been going on at home and abroad. However, existing token-based systems are not suitable for IoT environments in terms of security, reliability, and scalability because they have centralized characteristics. In addition, the system using the block chain has to overload the IoT device because it has to repeatedly perform the calculation of the hash et to hold the block chain and store all the blocks. In this paper, we intend to manage the access rights through tokens for proper access control in the IoT. In addition, we apply the Tangle to configure the P2P distributed ledger network environment to solve the problem of the centralized structure and to manage the token. The authentication process and the access right grant process are performed to issue a token and share a transaction for issuing the token so that all the nodes can verify the validity of the token. And we intent to reduce the access control process by reducing the repeated authentication process and the access authorization process by reusing the already issued token..

Keywords: IoT, Distributed ledger, Tangle, Access control, Token

Received(02. 12. 2019), Modified(03. 29. 2019), Accepted(04. 03. 2019)
* 본 논문(연구)은 교육부의 재원으로 이공분야기초연구사업의 지원을 받아 수행된 연구임(No.NRF-2018R1D1A1B07051203).
† 주저자,ghks0515@mokpo.ac.kr
‡ 교신저자, jhseo@mokpo.ac.kr(Corresponding author)

I. 서 론

사물 인터넷(IoT, Internet of Things)에서의 접근 제어는 IoT 디바이스가 경량 시스템인 점에 비하여, 짧은 시간동안 기기간의 상호 작용이 일어나고 동일한 요청이 반복적으로 수행된다는 점을 고려해야 한다.

최근에는 사물 인터넷에서 인증, 접근 제어 등을 위해 토큰과 블록체인을 적용한 시스템 연구들이 국내외에서 진행되고 있다. 기존 토큰을 적용한 접근 제어 시스템은 토큰을 통해 접근 제어 프로세스를 경량화하였다. 그러나 중앙 서버가 토큰을 관리하는 중앙 집중적인 시스템이기에 서버의 보안이 취약하다면 전체 시스템 운용에 어려움이 따른다. 블록체인을 적용한 접근 제어 시스템은 중앙 집중적인 구조의 문제점을 해결하였지만 해시 등의 계산을 반복적으로 수행해야하고 모든 블록을 저장해야하므로 IoT 디바이스의 과부하가 따른다[1,2].

본 논문은 분산 원장을 이용하고 토큰을 기반으로 하여 사물 인터넷에서의 인증과 접근 제어를 수행하고자 한다. 탱글(Tangle)을 적용하여 P2P 분산 원장 네트워크를 구성하며 노드들에 대한 인증 과정과 접근 권한 부여 과정을 수행하고 토큰을 발급한다. 토큰을 발급 받은 노드는 네트워크에서 토큰을 이용하여 권한을 소유하고 있음을 증명할 수 있다. 토큰 발급에 대한 트랜잭션을 공유하여 모든 노드들이 토큰에 대한 유효성을 검증 할 수 있도록 한다. 또한 기존 발급 받은 토큰을 재사용하여 반복적인 인증 과정과 접근 권한 부여 과정을 줄여서 접근 제어 프로세스를 경량화하고자 한다.

본 논문의 구성은 다음과 같다. 본론의 2장에서는 IoT 분산 원장 기술과 기존의 사물 인터넷에서의 접근 제어 관련 연구를 설명한다. 3장에서는 본 논문에서 제안하는 분산 원장을 이용한 토큰 기반 사물 인터넷 접근 제어 기술을 설명한다. 3장에서는 제안한 기술을 시뮬레이션하고 타당성 검증을 통하여 제안한 접근 제어 기술과 기존의 접근 제어 기술들을 비교한다.

II. 관련 연구

2.1 분산 원장 기술

분산 원장 기술(Distributed Ledger Technology)은 시스템에서 발생되는 거래 정보를 중앙 서버가 아닌 분산화된 네트워크 노드들이 공유하고 각자의 데이터베이스들을 지속적으로 동기화하는 기술이다. 네트워크에서 발생되는 정보들을 각 노드들이 공유함으로써 P2P 형태의 인증, 인가 등의 과정을 수행하므로 시간과 비용을 감소시킬 수 있다. 또한 투명성을 보장하고 공동으로 저장하므로 보안성과 신뢰성을 확보하여 중앙 집중적인 구조의 문제점을 해결할 수 있다. 분산 원장 기술에는 블록체인과 탱글이 있으며, 지속적인 연구가 진행되고 있다

블록체인은 각 블록들이 연결되어 있는 구조를 가지고 있다. 블록은 헤더 부분과 바디 부분으로 이루어져 있으며, 헤더 부분은 이전 블록의 해시값을 가지고 있어 체인을 구성할 수 있도록 한다. 바디 부분은 거래 정보를 저장하여 노드들이 거래 정보를 공유할 수 있도록 한다. 이전 블록의 해시 값을 통해 체인 형태로 저장되는 블록들은 Fig.1.과 같다.

전 블록의 해시값을 포함하여 저장하므로 블록을 위변조하기 위해서는 이전 블록까지 위변조를 시도해야한다. 체인 구조를 구성하는 블록의 개수가 많을수록 위변조를 시도해야하는 블록의 개수가 늘어나므로 거래에 대한 무결성 보장이 강해진다[3,4,5]. 블록체인은 주로 SHA(Secure Hash Algorithm)와 ECDSA(Elliptic Curve Digital Signature Algorithm) 알고리즘을 사용하여 블록을 생성하는데 해시, 암호화 등의 계산을 수차례 수행하므로 거래 처리 속도가 느려지는 문제점이 있다. 또한 모든 노드들에게 모든 블록을 저장하기 위한 큰 저장소를 필요하므로 IoT 디바이스에 적용하기에는 부적합할 수 있다.

탱글은 IOTA 재단이 기존 블록체인 기반 암호화폐가 거래되는 금액보다 수수료 금액이 더 크게 발생하는 문제점을 해결하기 위해 IOTA 코인을 출시하여 선보인 분산 원장 기술이다[6,7].

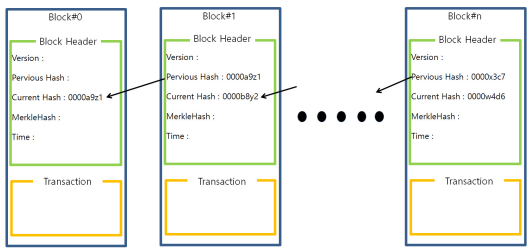

Fig. 1. Storage structure of Blockchain

블록체인과 달리, 블록을 구성하지 않고 각 트랜잭션들로만 분산 원장을 구성한다. 또한 블록체인 프로세스와 같이 저장하기 전 합의 과정을 수행하지 않는다. 탱글 네트워크에서 거래가 발생하면 우선적으로 거래의 당사자는 자신의 데이터베이스에서 합의 횟수가 부족한 트랜잭션 2개를 선택하고 각 트랜잭션들의 해시값들을 검증하는 합의 과정을 수행한다.

탱글에서 트랜잭션을 처리하는 과정은 다음 Fig.2.와 같다.

합의를 완료한 2개의 트랜잭션 해시값을 포함하는 트랜잭션을 저장함으로써 체인 형태의 선형 구조가 아닌 방향성 비순환 그래프(DAG, Directed Acyclic Graph) 형태로 연결하여 저장하는 구조를 구성한다. 트랜잭션들이 저장되어지는 DAG 형태는 Fig.3.와 같다.

IOTA 재단[6]은 기존 컴퓨팅 환경에서의 bit 단위 처리 방식인 2진법이 아닌 trit 단위 처리 방식인 3진법을 사용하며, 자체 개발한 경량 해시 알고리즘인 Curl을 이용하므로 탱글이 블록체인보다 사물 인터넷에 적합하다고 제시하고 있으며 현재도 지속적으로 개발 중이다.

블록체인과 탱글은 많은 차이점을 가지고 있으며

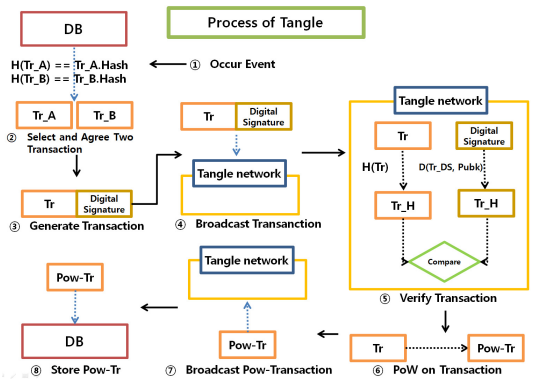

Fig. 2. Process of tangle

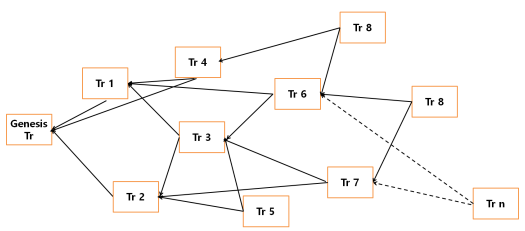

Fig. 3. Storage structure of Tangle

Table 1. Compare blockchain to tangle

	Blokchain	Tangle
Crypto-currency fee	High	Low
Speed of processing transaction	Low	High
Scalability	Low	High
Conformance with IoT	Low	High

비교사항은 Table 1.과 같다.

탱글 기반 암호 화폐는 작업 증명의 난이도가 낮으며, 트랜잭션을 생성하기 위해서는 합의를 우선적으로 수행해야하므로 블록체인 기반 암호 화폐에 비하여 수수료가 낮다. 또한 탱글은 trit 단위로 처리하므로 적은 주소 공간에서도 bit로 처리했을 경우보다 더 많은 데이터를 표현할 수 있으며, IOTA 재단이 자체 개발한 Curl라는 경량 해시 알고리즘을 이용하므로 블록체인보다 빠른 거래 처리 속도를 가진다. 블록들이 순차적으로 저장하는 블록체인과는 달리, 비순차적으로 저장하는 탱글이 더 큰 확장성을 가진다. 실시간성과 확장성을 요구하는 사물 인터넷에서는 거래 처리 속도가 높고 확장성을 가진 탱글이 더욱 적합하다[6,7].

2.2 사물 인터넷에서의 접근 제어

접근 제어란 주체가 정책에 따라 객체의 작업을 수행 할 수 있는지 여부를 나타내는 것으로, 자원에 대한 인가되지 않은 접근을 감시한다. 접근 요청에 대한 이용자를 식별하며, 접근 요청이 정당한 것인지를 확인하여 기록하고 보안 정책에 따라 접근 승인 또는 거부함으로써 비인가자로부터의 불법적인 자원 접근 및 파괴를 예방한다.

사물 인터넷에서의 접근 제어는 IoT 디바이스가 경량 시스템인 점에 비하여, 짧은 시간동안 기기간의 상호 작용이 일어나고 동일한 요청이 반복적으로 수행된다는 점을 고려해야한다. 따라서 기존 컴퓨팅 환경에서 적용하는 접근 제어 방식과는 다른 접근 제어 방식을 필요로 한다[1].

S.Gusmeroli 등[1]은 Capability 토큰을 기반으로 하여 사물 인터넷에서 접근제어를 수행하였다. 서버가 객체에 대한 주체의 권한을 Capability 토

큰으로 정의하고 주체에게 부여한다. 주체는 접근 요청 시 서버에게 부여 받은 Capability 토큰을 전달하여 객체에 대한 접근 여부를 결정한다. 토큰의 유효성 검증만으로 객체에 대한 접근 여부를 결정하여 접근 제어 프로세스의 경량화를 제공하였다. 기발행된 Capability 토큰의 재사용이 가능하고 다른 주체에게 Capability 토큰의 위임이 가능하다. 그러나 인터넷에 연결되는 IoT 디바이스 수가 기하급수적으로 증가하고 있어, 서버가 IoT 노드들을 관리하는 중앙 집중적인 구조는 적합하지 않다. 또한 서버의 보안이 취약하다면 전체 시스템의 운용에 어려움이 따른다.

A.Dorri 등[2]은 스마트 홈에서 블록체인을 적용하여 IoT 디바이스의 접근 제어를 수행하였다. 각 가정에 있는 스마트 홈 게이트웨이와 IoT 디바이스들로 블록체인 네트워크를 구성하였다. 각 가정 내에 있는 IoT 디바이스에 대한 접근 권한 목록을 블록체인을 통해 공유함으로써 접근 제어를 수행하는 연구가 진행되었다. 그러나 IoT 디바이스의 수가 증가할수록 접근 권한 목록의 크기가 증가되고 각 디바이스들은 불필요한 접근 권한 목록을 저장해야하므로 관리에 어려움이 따른다.

A.OUADDAH 등[8,9]은 사물 인터넷에서 블록체인을 적용한 접근 제어 모델을 제안하였다. AMP(Authorization Management Point) 역할을 수행하는 Wallet이 연결된 디바이스의 접근 제어 관련 트랜잭션을 생성하고, 이를 네트워크에 분산 공유하여 트랜잭션을 검증한다. 이 연구에서 블록체인은 네트워크의 모든 접근제어 정책을 트랜잭션의 형태로 저장하는 분산된 데이터베이스를 의미한다. 그러나 블록체인을 유지하기 위해서는 해시, 암호화 등의 계산을 반복적으로 수행해야하므로 IoT 디바이스의 과부하가 따른다.

본 논문에서는 토큰을 기반으로 인증 및 접근 제어를 수행하여 프로세스를 경량화하고자 한다. 또한 중앙 집중적인 구조의 문제점을 해결하고 사물 인터넷과의 적합성을 고려하여 탱글을 적용한 P2P 분산 원장 네트워크를 구성하여 분산화된 인증과 접근 제어를 수행하고자 한다.

III. 토큰 기반 사물 인터넷 접근 제어

사물 인터넷에서의 접근 제어는 IoT 디바이스가 경량 시스템인 점에 비하여, 짧은 시간동안 기기간의 상호 작용이 일어나며 동일한 요청이 반복적으로 수행된다는 점을 고려해야한다.

본 논문에서는 분산 원장인 탱글을 적용하여 인증과 권한 관리를 분산화하고, 접근 제어 프로세스를 최소화 할 수 있는 사물 인터넷에 적합한 토큰 기반 접근 제어 기술을 제안한다. 시스템 구성은 Fig.4.와 같다.

시스템 구조에서 풀 노드(Full node)와 라이트 노드(Light node)는 탱글을 적용하여 P2P 분산 원장 네트워크를 구성한다. 풀 노드는 다수의 라이트 노드들을 연결하여 하나의 도메인을 형성할 수 있다. 각 노드들은 인증 및 권한 관련 정보가 저장된 트랜잭션들을 공유한다. 또한 트랜잭션을 저장하기 위한 각자의 Wallet를 가지고 있다.

본 구조에서 각 노드들이 네트워크에 참여하기 위해 토큰을 기반으로 하여 인증과 접근 제어를 수행한다.

인증 단계에서는 노드가 같은 도메인 내의 풀 노드에게 인증을 수행 받는다. 인증이 완료되면 풀 노드는 인증 토큰 발급 트랜잭션을 생성하고 다른 풀 노드들과 공유하여 저장한다. 또한 인증을 요청한 라이트 노드에게 인증 토큰을 전송한다.

접근 제어 단계는 권한 부여를 요청하여 접근 권한 토큰을 부여 받는 단계와 권한 부여를 받은 토큰을 이용하여 노드에 접근하는 두 단계로 나뉘어 진행한다.

먼저, 인증 토큰을 발급 받은 노드는 다른 도메인의 풀 노드에게 인증 토큰을 전송하여 접근 권한 부

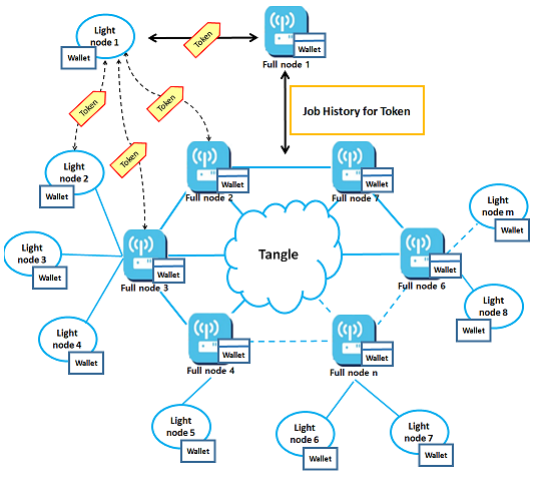

Fig. 4. Proposed system structure

여를 요청한다. 풀 노드는 Wallet에서 해당 인증 토큰의 발급 트랜잭션을 통해 유효성을 검증한다. 검증이 완료되면 접근 대상인 노드에게 요청을 전달한다.

요청을 전달 받은 노드는 소유하고 있는 Policy_Script를 통하여 부여할 권한을 결정하고, 접근 권한 토큰 발급 트랜잭션을 생성하고 풀 노드들과 공유하여 저장한다. 또한 접근 권한 토큰을 요청한 노드에게 전송한다.

다음은, 접근 권한 토큰을 발급 받은 노드는 접근 대상인 노드에게 접근 권한 토큰을 전송하여 인가를 요청한다. 요청을 받은 노드는 Wallet에서 해당 접근 권한 토큰의 발급 트랜잭션을 통해 유효성을 검증하여 접근 여부를 결정한다. 또한 접근 권한 토큰 사용 트랜잭션을 생성하고 풀 노드들과 공유하여 저장한다.

각 노드는 기존 발급 받은 토큰을 재사용함으로써 반복적인 인증 단계와 접근 제어 단계를 최소화하여 접근 제어 프로세스를 경량화할 수 있다.

각 단계의 프로토콜 절차를 기술하기 위한 Table 2.를 따른다.

Table 2. Description

Notation	Description
LN	Light Node.
FN	Full Node.
TN	Tangle Network.
NID	Node ID.
AT	Authentication Token.
TR_ATI	Authentication token issue transaction.
Right	The right that the requestor desires.
ART	Access Right Token.
TR_ARTI	Access right token issue transaction.
TR_ARTU	Access right token use transaction.
TK_DS	Digital signature for token
Pubk	Public key
A{B}->C	A sends B to C.

3.1 IoT 탱글 네트워크 노드

각 노드들은 작업 증명, 합의, 검증 등을 수행한다. IoT 디바이스의 종류에 따라 계산 능력의 차이가 있으므로, 탱글 네트워크 노드를 풀 노드와 라이트 노드로 분류한다.

모든 노드들은 식별을 위하여 노드의 공개키를 해시하여 아이디로 사용하고 트랜잭션을 저장하기 위한 Wallet를 소유하고 있다.

3.1.1 풀 노드(Full node)

풀 노드는 탱글 트랜잭션을 생성, 합의 및 검증하며, 도메인을 구성하여 인증 토큰과 접근 권한 토큰을 발급할 수 있는 노드로 정의한다.

이러한 기능을 수행하기 위하여 풀 노드는 해시, 암호화 및 검증 등의 계산을 반복적으로 수행할 수 있는 고성능의 사양을 가지고 있는 IoT 디바이스가 될 수 있다. 또한 네트워크에서 발생되는 모든 트랜잭션을 Wallet에 저장한다.

3.1.2 라이트 노드(Light node)

라이트 노드는 탱글 트랜잭션 생성 및 합의만을 할 수 있으며, 접근 권한 토큰만을 발급 할 수 있는 노드로 정의한다.

이러한 기능만을 수행하기 위하여 라이트 노드는 해시, 암호화 및 검증 등의 계산을 반복적으로 수행하기 어려운 저성능의 사양을 가지고 있는 IoT 디바이스가 될 수 있다. 풀 노드와는 달리, 라이트 노드의 Wallet은 자신이 발급하거나 발급 받은 토큰에 대한 트랜잭션만을 저장한다.

3.2 트랜잭션(Transaction)

트랜잭션은 인증 및 접근 제어 작업 기록을 의미한다. 노드들은 토큰에 대한 발급, 사용, 위임, 폐기 작업을 수행한 경우, 트랜잭션을 구성하여 탱글 네트워크에 전송하고 각자의 Wallet에 트랜잭션을 저장하여 공유한다. 트랜잭션의 구조는 Fig.5.와 같다.

- TR_ID : 트랜잭션 식별자를 나타낸다.
- TR_Type : 트랜잭션 타입을 나타내며, 인증 토큰 발급(0), 접근 권한 토큰 발급(1), 접근 권한 토큰 사용(2), 접근 권한 토큰 위임(3), 인증 토큰 폐기(4), 접근 권한 토큰 폐기(5)가 있다.
- TK_Sender/TK_Receiver : 각각 토큰의 발급자

```
"Transaction":
{
    "TR_Hash":"",
    "TR_ID":"",
    "TR_Type":"",
    "TK_Sender":"",
    "TK_Receiver":"",
    "TK_ID":"",
    "TK_Hash":"",
    "Trunktransaction":"",
    "Branchtransaction":"",
    "TR_Timestamp":"",
    "TR_Digital Signature":""
}
```

Fig. 5. Transaction structure

와 발급 받은 자의 식별자를 나타낸다.

- TK_ID : 토큰의 식별자를 나타낸다.
- TK_Hash : 토큰의 해시값을 나타낸다.
- Trunktransaction/Branchtransaction : 탱글 구조에서 트랜잭션을 생성한 노드가 합의한 2개의 트랜잭션 해시값을 나타낸다.
- TR_Timestamp : 트랜잭션 생성 시간을 나타낸다.
- TR_Digital Signature : 트랜잭션의 전자서명 값을 나타낸다.
- TR_Hash : 작업 증명 과정을 통해 계산되어진 트랜잭션의 Hash를 나타낸다.

3.3 인증 단계

인증 단계는 노드에 대한 인증 과정을 수행하여 인증 토큰을 발급하는 단계이다.

라이트 노드가 다른 노드에게 접근하기 위해서는 시스템에서 인증 받은 노드임을 증명해야한다. 이를 위해 라이트 노드는 소속되어 있는 도메인의 풀 노드에게서 인증 받아야한다.

인증을 수행한 풀 노드는 인증 토큰 발급 트랜잭션을 생성하고 다른 풀 노드에게 전송하여 공유한다. 또한 인증 받은 라이트 노드에게 인증 토큰을 전송한다.

라이트 노드는 다른 풀 노드에게 인증 토큰을 사용함으로써 인증 단계를 생략하고 접근 권한 부여 요청을 할 수 있다. 또한 한번 발급 받은 인증 토큰은 재사용할 수 있으므로 반복적인 인증 단계를 수행하지 않아도 된다.

인증 토큰 발급 및 인증 토큰 발급 트랜잭션 생성 과정은 Fig.6.과 같다.

풀 노드가 생성한 인증 토큰은 다른 풀 노드들에게 인증 토큰 발급 트랜잭션이 전송하고 트랜잭션에 대한 검증이 완료된 다음, 요청한 라이트 노드에 전송된다.

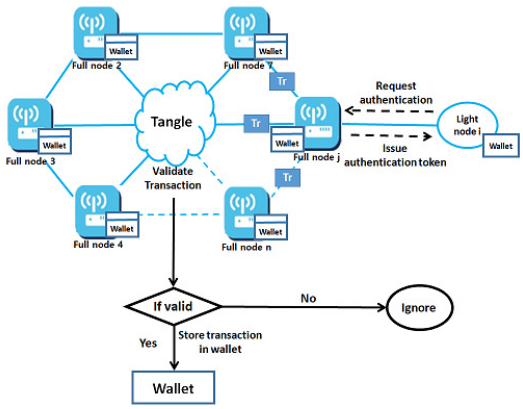

Fig. 6. Process for issuing authentication token

인증 단계를 위한 인증 토큰의 발급 및 인증 토큰 발급 트랜잭션의 프로토콜 절차는 Fig.7.과 같다.

- 인증 요청(①)

다른 노드에게 접근을 원하는 Light node i는 소속되어 있는 도메인의 Full node j에게 인증 요청 메시지를 전송한다.

- 인증 토큰 및 트랜잭션 생성(②-④)

인증 요청 메시지를 전송 받은 Full node j는 의 과정을 거쳐 Light node i에 대한 인증 토큰을 생성한 후, 토큰 발급 트랜잭션을 구성하기 위해 탱글의 합의과정을 수행한다. 이를 위해 Full node j는 자신의 Wallet에서 합의 횟수가 부족한 트랜잭션 2개를 선택하여 합의한다. 합의를 통해 Trunktransaction과 Branchtransaction을 결정하여 토큰 발급 트랜잭션을 생성한다.

- 인증 토큰 발급 트랜잭션 전송(⑤)

Full node j는 생성한 인증 토큰 발급 트랜잭션

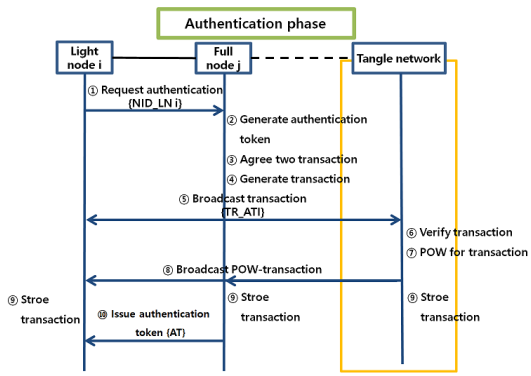

Fig. 7. Authentication phase

을 다른 풀 노드들에게 전송한다.

- 트랜잭션 검증(⑥-⑨)

풀 노드들은 전송 받은 트랜잭션의 TR_Digital Signature를 검증한다. 검증이 완료되면 트랜잭션의 TR_Hash를 생성하기 위해 작업 증명을 수행한다. 제일 먼저 TR_Hash를 생성한 노드가 작업 증명이 완료된 트랜잭션을 다른 풀 노드들에게 전송한다. 풀 노드들은 전송 받은 트랜잭션을 각자의 Wallet에 저장한다.

- 인증 토큰 발급(⑩)

트랜잭션 저장까지 완료한 Full node j는 Light node i에게 인증 토큰을 전송한다.

인증 단계를 통해 풀 노드가 발급하는 인증 토큰의 구조는 Fig.8.과 같다.

- ATID(Authentication Token ID) : 인증 토큰의 식별자를 나타낸다.
- SID(Sender ID)/RID(Receiver ID) : 각각 인증 토큰의 발급자와 발급 받은 자의 식별자를 나타낸다.
- Hash : 토큰의 해시값을 나타낸다.

인증 단계를 거쳐 발급된 토큰은 이후 접근 제어 과정에서 접근 권한 부여 요청을 위해 사용된다. Light node i는 Full node j로부터 인증 토큰을 발급 받았으므로 시스템에서 인증된 노드임을 증명할 수 있다.

Full node j가 인증 토큰 발급 트랜잭션을 생성하고 네트워크 노드들과 공유하여 위변조가 불가능한 분산 원장을 유지함으로써, 모든 노드들은 인증 토큰에 대한 유효성을 검증할 수 있다.

Light node i는 기발급된 인증 토큰을 사용하여 다른 풀 노드와 인증 단계를 수행하지 않아도 되므로 인증 과정을 간소화할 수 있다. 또한 인증 토큰을 재사용할 수 있으므로 인증 단계를 재수행하지 않아도 된다.

```
"Authentication Token":
{
    "ATID":"",
    "SID":"",
    "RID":"",
    "Hash":""
}
```

Fig. 8. Authentication token structure

3.4 접근 제어 단계

본 논문에서는 접근 제어 단계를 두 단계로 구분한다. 접근 권한 부여를 위한 접근 권한 토큰을 발급하는 단계와 접근 권한 토큰 사용을 통한 인가 단계로 나뉜다.

3.4.1 접근 권한 토큰 발급

접근 권한 토큰 발급은 탱글 네트워크에 연결된 노드에 대해 접근 권한을 부여하는 단계이다.

탱글 네트워크에서 다른 노드에게 접근하기 위해서는 해당 노드에 대한 접근 권한을 소유하고 있음을 증명해야한다. 이를 위해 서비스 요청 전에 접근 대상이 되는 노드로부터 접근 권한 토큰을 발급받는 과정이 필요하다.

접근 권한 부여를 요청 받은 노드는 접근 권한 토큰 발급 트랜잭션을 생성하여 풀 노드들과 접근 권한을 요청한 노드에게 전송하여 공유한다. 또한 접근 권한 토큰을 요청한 노드에게 전송한다.

접근 권한 토큰을 사용함으로써 부여 받은 권한에 대한 인가를 수행할 수 있다. 또한 기발급된 접근 권한 토큰은 재사용할 수 있으므로 반복적인 접근 권한

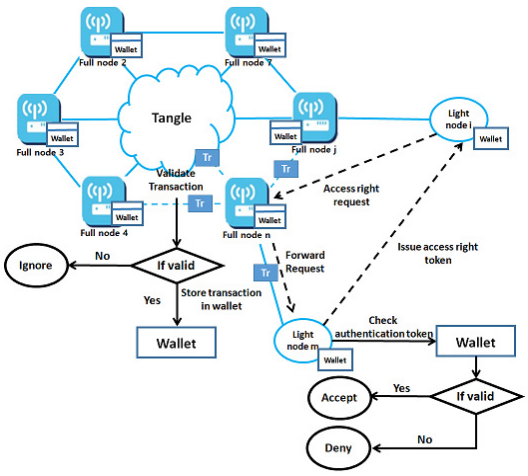

Fig. 9. Process for issuing access right token

토큰 발급을 수행하지 않아도 된다.

접근 권한 토큰 발급 과정은 Fig.9.와 같다.

접근 권한이 필요한 노드는 인증된 노드임을 증명하는 인증 토큰을 제시함으로써 접근 권한 부여를 요청한다.

접근 대상인 노드가 생성한 접근 권한 토큰은 다른 풀 노드들과 접근 권한을 요청 노드들에게 접근 권한 토큰 발급 트랜잭션이 전송하고 트랜잭션에 대한 검증이 완료된 다음, 요청한 노드에게 전송된다.

접근 권한 토큰 발급 및 접근 권한 토큰 발급 트랜잭션의 프로토콜 절차는 Fig.10.과 같다.

- 접근 권한 부여 요청(①)

Light node i는 Light node m으로부터 접근 권한을 부여받기 위해 먼저 Full node n에게 인증 토큰, Light node i가 Light node m으로부터 원하는 접근 권한, 인증 토큰에 대한 전자서명과 라이트 노드 i의 공개키를 전송하여 접근 권한 부여를 요청한다.

- 인증 토큰 유효성 검증(②)

접근 권한 부여 요청을 받은 Full node n은 인증 토큰의 전자서명을 검증하여 송신자가 Light node i임을 확인한다. 인증 토큰의 유효성을 검증하는 과정은 다음과 같다.

인증 토큰의 유효성 검증을 위해 Full node n은 Wallet에서 인증 토큰의 식별자(AT.ATID)와 같은 토큰 식별자(TR.TK_ID)가 있는 트랜잭션을 검색한다. 찾은 인증 토큰 발급 트랜잭션의 토큰 해시값(TR_TK_Hash)과 인증 토큰의 해시값(AT.TK_Hash)이 같은 경우, 해당 인증 토큰이 유

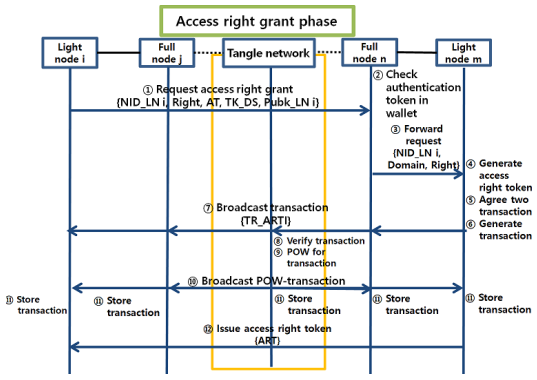

Fig. 10. Access right Token Generation Process

```
If Token_Digital_Signature_Verification(TK_DS) then
  If AT.ATID == TR.TK_ID then
    If AT.TK_Hash == TR.TK_Hash then
        Token_Validation_Complete()
    Else
        Token_Validation_Failure()
  Else
    Token_Validation_Failure()
Else
  Token_Validation_Failure()
```

효함을 검증할 수 있다.

유효성 검증이 완료되면 인증 토큰의 발급자를 확인하여 도메인의 내외부를 판별한다. 인증 토큰의 발급자가 Full node n이 아니면 외부 도메인의 노드라는 사실을 판별 할 수 있다.

- 접근 권한 부여 요청 전달(③)

Full node n은 Light node m에게 Light node i의 ID, Light node i의 Domain과 요청된 Right를 전달한다.

- 접근 권한 토큰 및 트랜잭션 생성(④-⑥)

요청 전달 받은 Light node m은 Policy_Script에 따라 접근 권한을 결정하고 접근 권한 토큰을 생성한다.

Policy_Script는 부여할 접근 권한을 결정하기 위해 도메인의 동일성을 기준으로 정의하여 각 노드들이 소유하고 있는 스크립트이다. 노드는 접근 권한을 요청한 노드의 도메인이 내부인 경우와 외부인 경우를 구분하여 부여할 권한을 정의할 수 있다.

Light node m은 접근 권한 토큰을 생성한 다음 앞서 설명한 탱글 합의 과정을 통해 접근 권한 토큰 발급 트랜잭션을 생성한다.

- 접근 권한 토큰 발급 트랜잭션 전송(⑦)

Light node m은 생성한 접근 권한 토큰 발급 트랜잭션을 네트워크 내의 풀 노드들과 Light node i에게 전송한다.

- 트랜잭션 검증(⑧-⑪)

트랜잭션을 공유한 노드들은 앞서 설명한 트랜잭션 검증 과정을 통해 접근 권한 토큰 발급 트랜잭션을 검증하고 저장한다.

- 접근 권한 토큰 발급(⑫)

트랜잭션 저장까지 완료한 Light node m은 Light node i에게 접근 권한 토큰을 전송하고 라이

트 노드 i는 접근 권한 토큰을 저장한다.

Light node m이 발급하는 접근 권한 토큰의 구조는 Fig.11.과 같다.

- ARTID(Access Right Token ID) : 접근 권한 토큰의 식별자를 나타낸다.
- SID(Sender ID)/RID(Receiver ID) : 각각 접근 권한 토큰의 발급자와 발급 받은 자의 식별자를 나타낸다.
- AR(Access Right): 부여하는 접근 권한을 읽기(r), 쓰기(w), 읽기와 쓰기(rw)로 나타낸다.
- DELEGATION : 토큰의 위임 가능 여부를 True와 False로 나타낸다.
- Hash : 토큰의 해시값을 나타낸다.

접근 권한 토큰은 대상 노드에 대한 서비스 요청을 위해 사용된다. Light node i는 Light node m으로부터 접근 권한 토큰을 발급 받았으므로 Light node m에 대한 접근 권한을 소유하고 있는 노드임을 증명할 수 있다.

Light node m이 접근 권한 토큰 발급 트랜잭션을 생성하고 네트워크 노드들과 공유하여 위변조가 불가능한 분산 원장을 유지함으로서 접근 권한 토큰에 대한 유효성을 검증할 수 있다.

Light node i는 기발급된 접근 권한 토큰을 재사용할 수 있으므로 접근 권한 토큰 발급 과정을 재수행하지 않아도 된다.

```
"Access Right Token":
{
    "ARTID":"",
    "SID":"",
    "RID":"",
    "AR":"",
    "DELEGATION":"",
    "ISSUED TIME":"",
    "EXPIRED TIME":"",
    "Hash":""
}
```

Fig. 11. Access right token structure

3.4.2 접근 권한 토큰을 통한 인가

탱글 네트워크에 연결된 노드에 대한 서비스 요청 시 접근 권한 토큰을 사용함으로써 부여 받은 권한에 대해 인가한다.

서비스를 요청하는 노드는 서비스 대상이 되는 노드에게 접근 권한 토큰을 제시하고, 토큰에 대한 유효성 검증을 통해 인가가 결정된다.

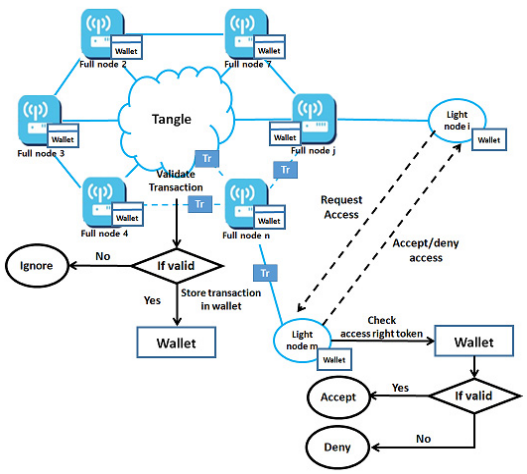

Fig. 12. Using access right token phase

인가가 허용되었을 경우, 서비스 대상이 되는 노드는 접근 권한 토큰 사용 트랜잭션을 생성하고 풀 노드들과 서비스를 요청한 노드에게 전송하여 공유한다.

탱글 네트워크에서의 접근 권한 토큰을 통한 인가 과정은 Fig.12.와 같다.

접근 권한 토큰을 사용한 인가 및 접근 권한 토큰 사용 트랜잭션의 프로토콜 절차는 Fig.13.과 같다.

- 접근 권한 사용 요청(①)

Light node i는 Light node m에게 접근 권한 토큰, 접근 권한 토큰에 대한 전자서명과 Light node i의 공개키를 전송하여 접근 권한 사용을 요청

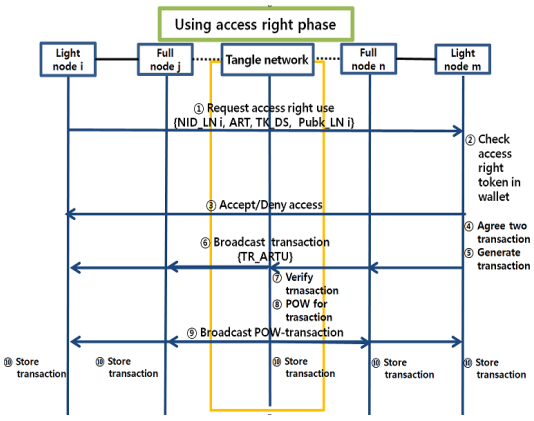

Fig. 13. Using access right phase

```
If Token_Digital_Signature_Verification(TK_DS) then
  If ART.ARTID == TR.TK_ID then
    If ART.TK_Hash == TR.TK_Hash then
        Token_Validation_Complete()
    Else
        Token_Validation_Failure()
  Else
    Token_Validation_Failure()
Else
  Token_Validation_Failure()
```

한다.

- 접근 권한 토큰 유효성 검증(②-③)

접근 권한 사용 요청을 받은 Light node m은 접근 권한 토큰의 전자서명을 검증하여 송신자가 Light node i임을 확인한다. 접근 권한 토큰의 유효성을 검증하는 과정은 다음과 같다.

접근 권한 토큰의 유효성 검증을 위해 Light node n은 Wallet에서 접근 권한 토큰의 식별자(ART.ATID)와 같은 토큰 식별자(TR.TK_ID)가 있는 트랜잭션을 검색한다. 찾은 접근 권한 토큰 발급 트랜잭션의 토큰 해시값(TR_TK_Hash)과 접근 권한 토큰의 해시값(ART.TK_Hash)이 같은 경우, 해당 접근 권한 토큰이 유효함을 검증할 수 있다.

접근 권한 토큰에 대한 유효성이 확인되면, 접근을 허용한다. 만약 유효한 토큰이 아닐 경우, 접근을 거부한다.

- 접근 권한 토큰 사용 트랜잭션 생성(④-⑤)

Light node m은 앞서 설명한 탱글 합의 과정을 통해 접근 권한 토큰 사용 트랜잭션을 생성한다.

- 접근 권한 토큰 사용 트랜잭션 전송(⑥)

Light node m은 생성한 접근 권한 토큰 사용 트랜잭션을 네트워크 내의 풀 노드들과 Light node i에게 전송한다.

- 트랜잭션 검증(⑦-⑩)

트랜잭션을 공유한 노드들은 앞서 설명한 트랜잭션 검증 과정을 통해 접근 권한 토큰을 검증하고 저장한다.

본 논문에서는 토큰 기반 사물 인터넷 접근 제어를 위해 IoT 디바이스에 적합한 탱글을 적용한 P2P 분산 원장 네트워크를 구성하였으며 토큰에 대한 작업 기록을 공유할 수 있는 트랜잭션을 정의하였다.

인증을 수행하여 인증 토큰을 발급하는 인증 단계와 접근 권한 토큰 발급 및 사용을 통해 인가하는 접근 제어 단계를 수행하였다.

토큰 발급에 대한 트랜잭션을 공유하여 모든 노드들이 토큰에 대한 유효성을 검증 할 수 있도록 하였다. 기존 발급 받은 토큰을 재사용하여 반복적인 인증 과정과 접근 권한 부여 과정을 줄여서 접근 제어 프로세스를 경량화하였다.

IV. 시뮬레이션

제안한 기술을 시뮬레이션하여 풀 노드와 라이트 노드를 구성하고 토큰 생성과 트랜잭션 생성 및 저장을 확인하여 시스템 구현 가능성을 확인하고자 한다. 또한 타당성 검증을 통하여 제안한 기술과 기존의 접근 제어 기술들을 비교하고자 한다.

4.1 시나리오

본 시나리오의 시스템에는 Full node 1과 Full node 2가 있으며 각각의 도메인들을 형성하고 있다. Full node 1의 도메인에는 Light node 1이 소속하고 있으며, Full node 2의 도메인에는 Light node 2가 소속하고 있다. 시스템 구성은 Fig.14.와 같다.

Light node 1은 Full node 1에게 인증 단계를 수행하여 인증 토큰을 발급 받는다. 인증 토큰을 발급 받은 Light node 1은 Full node 2에게 접근 권한 부여를 요청하고 Light node 2로부터 접근 권한 토큰을 발급 받는다. 접근 권한 토큰을 발급 받은 Light node 1은 Light node 2에게 접근 권한 토큰을 사용하여 인가를 수행한다.

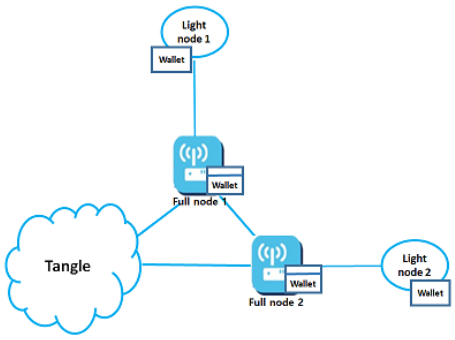

Fig. 14. The Tangle network node in the scenario

4.2 노드 구성 환경

풀 노드는 고성능 사양인 점을 고려하여 미니 PC로 구성하였다. 라이트 노드는 저성능인 점을 고려하여 라즈베리파이로 구성하였다. 노드 구성 환경은 Table 3.과 같다.

Table 3. Node configuration environment

	Full node	Light node
Hardware	Mini PC brix	Raspberry Pi 3 model B+
Operating system	Window 10	Raspbian (November, 2018)
Development language	C#	
Hash algorithm	Curl algorithm of the IOTA Foundation	
Digital signature algorithm	ECDSA algorithm for C# libraries	

4.3 인증 단계 시뮬레이션

본 절에서는 Full node 1과 Light node 1 간의 인증 단계를 시뮬레이션한다. Full node 1과 Light node 1 간의 인증 단계는 Fig.15.와 같다.

Light node 1은 Full node 1에게 인증을 수행받는다. 인증을 수행한 Full node 1은 제이슨(Json)형태의 인증 토큰과 인증 토큰 발급에 대한 트랜잭션을 생성한다. 생성한 인증 토큰과 인증 토큰 발급 트랜잭션은 Fig.16.과 같다.

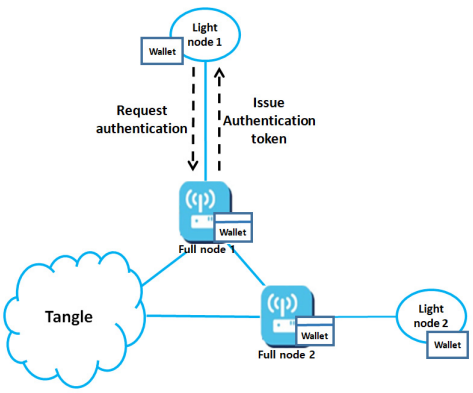

Fig. 15. Authentication phase between full node 1 and light node 1

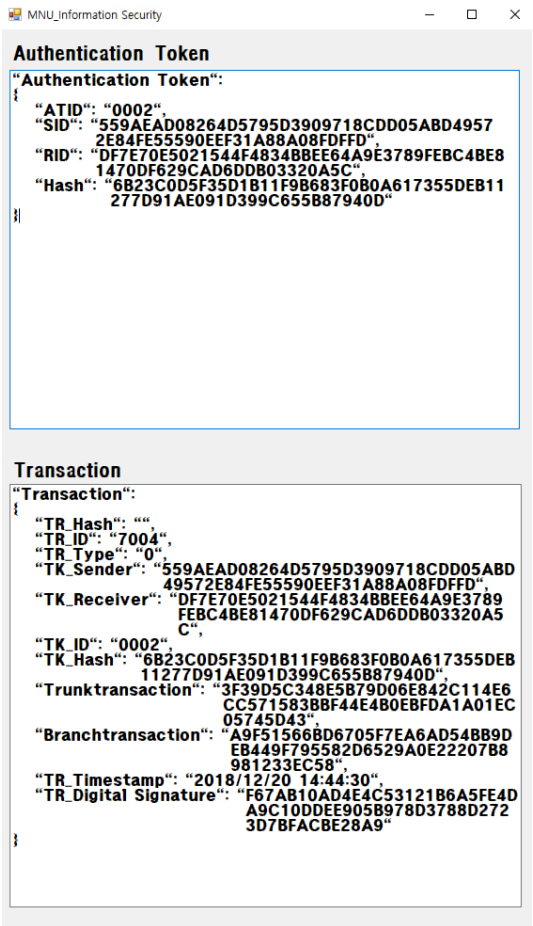

Fig. 16. Generating authentication tokens and authentication token issue transactions

Full node 1은 생성한 트랜잭션을 다른 풀 노드에게 전송한다. 풀 노드는 공유한 트랜잭션의 TR_Digital Signature를 검증한다.

검증이 완료되면 트랜잭션의 TR_Hash를 생성하기 위해 작업 증명을 수행한다. 제일 먼저 TR_Hash를 생성한 노드가 작업 증명이 완료된 트랜잭션을 다른 풀 노드에게 전송한다.

풀 노드들은 전송 받은 트랜잭션을 각자의 Wallet에 저장한다. 인증 토큰 발급 트랜잭션이 저장되어지는 DAG 형태는 Fig.17.과 같다.

각 트랜잭션은 2개의 트랜잭션들을 참조하여 DAG 형태를 구성한다. 트랜잭션의 저장이 완료되면 Full node 1은 Light node 1에게 생성한 인증 토큰을 전송한다. Light node 1은 접근 권한 부여 요청에 사용하기 위해 전송 받은 인증 토큰을 저장한다.

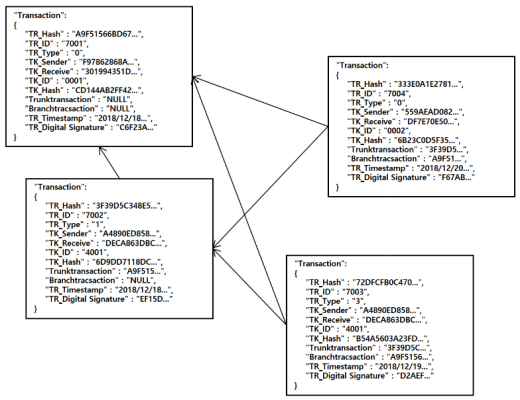

Fig. 17. Storing authentication token issue transaction

4.4 접근 권한 토큰 발급 시뮬레이션

본 절에서는 Light node 2가 Light node 1에게 접근 권한 토큰을 발급하는 과정을 시뮬레이션한다. 접근 권한 토큰을 발급하는 과정은 Fig.18.과 같다.

Light node 1은 Light node 2에 대한 접근 권한 토큰을 발급 받기 위해 Full node 2에게 접근 권한 부여 요청 메시지를 전송한다.

Full node 2는 Light node 1의 인증 토큰에 대한 유효성을 검증한다. 또한 토큰의 발급자를 통해 Light node 1이 외부 도메인임을 구분한다. 검증이 완료되면 Full node 2는 Light node 2에게 접근 권한 부여 요청 메시지를 전달한다.

Light node 2는 소유하고 있는 Policy_Script

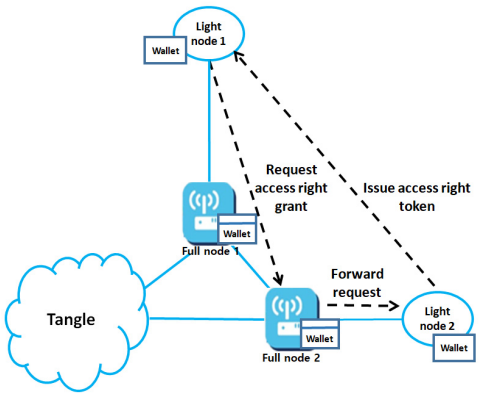

Fig. 18. Access right grant phase between light node 1 and light node 2

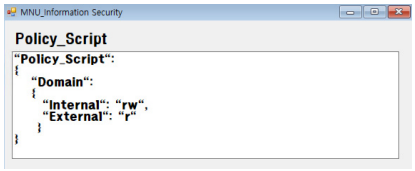

Fig. 19. Light node 2's policy_script

를 확인하여 부여할 권한을 결정한다. Light node 2의 Policy_Script는 Fig.19.와 같다.

Light node 2의 Policy_Script에서는 외부 도메인의 노드에게 읽기 권한만 부여하고 내부 도메인의 노드에게 읽기와 쓰기 권한만을 부여하는 것으로 정의하고 있다. 따라서 Light node 1에게 읽기 권한만을 부여한다.

Light node 2는 제이슨 형태의 접근 권한 토큰과 접근 권한 토큰 발급 트랜잭션을 생성한다. Light node 2가 생성한 접근 권한 토큰과 접근 권한 토큰 발급 트랜잭션은 Fig.20.과 같다.

Light node 2는 생성한 트랜잭션을 풀 노드들과

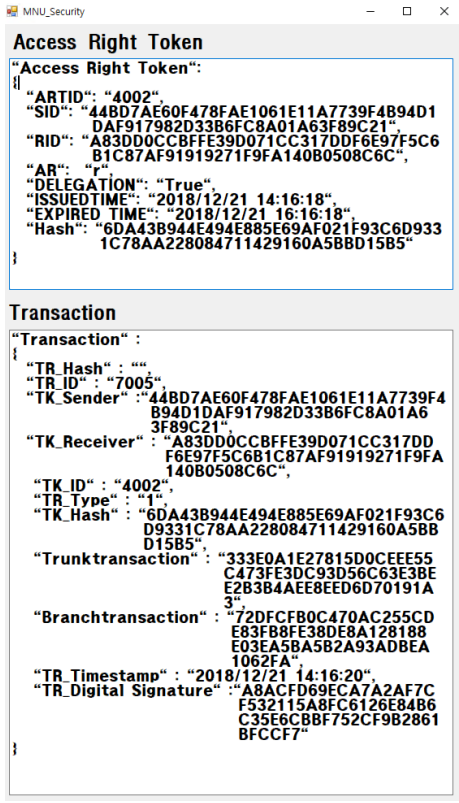

Fig. 20. Generating access right tokens and access right token issue transactions

Light node 1에게 전송한다. 트랜잭션을 전송 받은 노드들은 트랜잭션의 TR_Digital Signature를 검증한다.

검증이 완료되면 트랜잭션의 TR_Hash를 생성하기 위해 작업 증명을 수행한다. 제일 먼저 TR_Hash를 생성한 노드가 작업 증명이 완료된 트랜잭션을 풀 노드들과 Light node 1에게 전송한다.

트랜잭션을 전송 받은 노드들은 각자의 Wallet에 저장한다. 접근 권한 토큰 발급 트랜잭션이 저장되어지는 DAG 형태는 Fig.21.과 같다.

각 트랜잭션은 2개의 트랜잭션들을 참조하여 DAG 형태를 구성한다. 트랜잭션의 저장이 완료되면 Light node 2는 Light node 1에게 생성한 접근 권한 토큰을 전송한다. Light node 1은 서비스 요청 시, 인가 받기 위해 전송 받은 접근 권한 토큰을 저장한다.

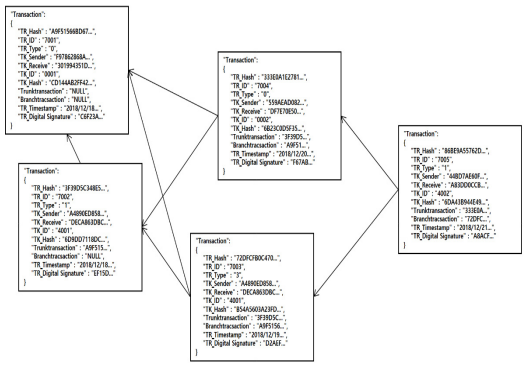

Fig. 21. Storing access right token issue transaction

4.5 타당성 분석

본 논문에서는 제안한 기술과 기존의 접근 제어 기술들을 비교하여 타당성을 분석해보고자 한다. 제안한 기술과 기존 접근 제어 기술 간의 비교는 Table 4. 과 같다.

S.Gusmroli 등[1]은 토큰을 위임함으로서 권한 위임 또한 가능하며 서버가 토큰을 관리하므로 트랜잭션 처리 속도가 빠르다. 그러나 시스템 내 한 곳의 문제가 전체 시스템 작동을 멈추게 하는 단일 지점 장애(Single Point of Failure, SOF) 문제점을 가지고 있다. 단일 지점 장애는 기계 오작동, 고의

Table 4. Comparison with Existing Studies

	S.Gusmeroli et al[1]	A.Dorri et al[2] and A.Ouaddah et al [8,9]	Proposed
Solving SOF	X	O	O
Transaction processing speed	High	Low	Medium
Permission to delegate authority	O	O	O
Scalability	X	X	O
Compatibility with IoT devices	O	X	O

또는 실수에 의한 사람의 행동, 자연 재해, 보안 사고 등 많은 요소가 원인이 될 수 있다.

A.Dorri 등[2]과 A.Ouaddah 등[8,9]은 블록체인을 적용하여 모든 노드들이 원장을 공유하고 있으므로 권한 위임에 대한 트랜잭션을 생성함으로써 권한 위임이 가능하다. 또한 분산화를 통해 단일 지점 장애를 해결하였다. 그러나 블록체인 프로세스는 해시 등의 계산을 반복적으로 수행해야하고 생성되는 모든 블록들을 순차적으로 저장해야하므로 처리 속도가 느리다.

본 논문에서 제안한 기술은 탱글을 적용하여 P2P 분산 환경 네트워크를 구성함으로써 단일 지점 장애를 해결하였다. 서버를 기반으로 하는 시스템보다는 처리 속도가 느리지만 IoT 디바이스를 고려한 해시 알고리즘과 DAG 형태로 트랜잭션을 저장하므로 블록체인 기반 기술보다는 빠른 처리 속도와 확장성을 가질 수 있다.

V. 결 론

최근 사물 인터넷에서 인증, 접근 제어 등을 위해 토큰과 블록체인을 이용한 시스템 연구들이 국내외에서 진행되고 있다. 그러나 기존 토큰을 이용한 방식의 시스템은 중앙 집중적인 특성을 가지고 있으므로 보안성, 신뢰성, 확장성 측면에서 사물 인터넷에 적용은 적합하지 않다. 또한 블록체인을 이용한 방식의 시스템은 블록체인을 유지하기 위해서 해시 등의 계

산을 반복적으로 수행하고 모든 블록을 저장해야하므로 IoT 디바이스에 과부하가 따른다.

본 논문은 사물 인터넷에 적합한 접근 제어를 위해 토큰을 이용하여 인증 및 권한 관리를 하고 모든 작업을 분산 원장 형태로 공유할 수 있는 기술을 제안하였다.

인증 토큰 발급, 접근 권한 토큰 발급, 접근 권한 토큰 사용 작업에 대한 트랜잭션을 생성하고 네트워크에 연결된 풀 노드들이 이를 공유함으로써 모든 노드들이 토큰 및 트랜잭션에 대한 유효성을 검증할 수 있다.

기존 발급 받은 토큰을 재사용할 수 있으므로 반복적인 인증 과정과 접근 권한 부여 과정을 줄여서 접근 제어 프로세스를 경량화하였다. 시뮬레이션을 통해 시스템 구현의 가능성을 확인하였으며 기존의 연구들과 비교하여 타당성을 검증하였다.

본 논문에서는 트랜잭션을 분산 원장으로 저장하기 위해 탱글의 DAG 구조를 적용하고, 작업 증명을 수행하고 있다. 향후에는 탱글과 블록체인을 적용한 테스트베드를 구축하여 사물인터넷 환경에 적합한 토큰 기반 접근 제어 기술에 대한 타당성 검증 및 평가 연구를 진행하고자 한다.

References

〔1〕 S.Gusmeroli, S.Piccione and D.Rotondi, "IoT access control issues: a capability based approach." IMIS-2012, pp787-792, July. 2012.

〔2〕 A.Dorri, S.Kanhere, R.jurdak and P.Gauravaram, "Blockchain for IoT Security and Privacy:The Case Study of a Smart Home", IEEE Percom workshop on security privacy and trust in the internet of things, March. 2017.

〔3〕 Hyung Wook Kim, "A Design of mutual authentication protocol between heterogeneous services in the internet of things Environment", Ph.D. Dissertation, Soongsil University, Korea, June. 2017.

〔4〕 Myung Hwan Lim, "The Effect, Problems and Implications of Block Chain Technology", Weekly ICT Trends, vol. 1776, pp 2-13, Dec. 2017.

〔5〕 S.Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", www.bitcoin.org, Oct. 2008.

〔6〕 S.Popov "The Tangle", www.Iota.org, April. 2018.

〔7〕 B. Breier, "Technical Analysis of the Tangle in th IOTA-Environment", Bacheler's Thesis, Technical University of Munich, Nov. 2017.

〔8〕 A. Ouaddah, A. Abou Elkalam and A. Ait Ouahman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things", Security and Communication Networks, pp. 5943-5964, Feb. 2017.

〔9〕 A. Ouaddah, A. Abou Elkalam and A. Ait Ouahman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things", Security and Communication Networks, pp. 5943-5964, Feb. 2017.

〔10〕 Hwan Park, Mi-sun Kim and Jae-hyun Seo, "Token-based Righs Management Using IoT Blockchain", CISC-W'18, pp. 162-165, Dec. 2018.

〈저자소개〉

박 환 (Hwan Park) 학생회원
2018년 2월: 목포대학교 정보보호학과 졸업
2018년 3월~현재: 목포대학교 정보보호기술학협동과정 석사과정
〈관심분야〉 블록체인, 정보보호, 프로그래밍 언어

김 미 선 (Mi-sun Kim) 정회원
1996년 2월: 목포대학교 컴퓨터공학과 졸업
2000년 2월: 목포대학교 컴퓨터공학과 석사
2007년 2월: 목포대학교 컴퓨터공학과 박사
2012년 12월~현재: 목포대학교 정보보호학과 초빙 교수
〈관심분야〉 정보보호, 프로그래밍 언어, 컴퓨터 네트워크, 모바일 시스템 보안

서 재 현 (Jae-hyun Seo) 종신회원
1985년 9월: 전남대학교 계산통계학과 졸업
1988년 2월: 중앙대학교 전자계산학과 석사
1996년 8월: 전남대학교 전산통계학과 박사
1996년 9월~현재: 목포대학교 정보보호학과 교수
〈관심분야〉 정보보호, 시스템 및 네트워크 보안, 컴퓨터 네트워크, 모바일 네트워크 보안