논문 21-46-03-19 The Journal of Korean Institute of Communications and Information Sciences '21-03 Vol.46 No.03
https://doi.org/10.7840/kics.2021.46.3.563

블록체인 릴레이 기반 양자키 분배 설계 및 구현

박 세 진◆, 안 재 욱*, 석 우 진**, 이 원 혁**, 주 홍 택°

Design and Implementation of a Blockchain Relay for Quantum Key Distribution

Sejin Park◆, Jaewook Ahn*, Woojin Seok**, Wonhyuk Lee**, Hongtak Ju°

요 약

양자키 분배를 이용하면 키 분배 과정에서 물리적으로 발생한 위변조 행위 자체를 감지할 수 있어 안전한 키 분배를 할 수 있게 된다. 그러나 양자 전달의 물리적인 거리한계로 인해 원거리 노드간 양자키 분배를 하려면 추가적인 중계 장비가 필요한 문제가 있다. 또한 추가 장비 활용에 따른 양자 중계 링크의 충돌, 중계 지연 등의 병목 현상이 발생할 수 있다. 본 논문은 양자키 분배 시 거리의 제약이 없도록 오픈소스 블록체인 상에서 키를 릴레이 하는 방법을 설계하고 실제 구현한다. 구체적으로 블록체인 스마트 컨트랙트 구현에 필요한 통신 채널, 자료구조 및 프로토콜을 제시하였으며, 또한 제안한 시스템을 이용하여 키 분배 실험을 진행한다. 본 실험 결과에 따르면, 릴레이에 대한 추가적인 오버헤드 없이 블록체인의 트랜잭션 처리 오버헤드 만으로 두 노드간 키 전달이 되는 것을 확인할 수 있었다.

Key Words : Blockchain, Quantum Key, Distribution, Relay, Smart Contract

ABSTRACT

A trusted key distribution can be achieved by Quantum Key Distribution (QKD) because malicious behaviors on key distribution process can be detected by the characteristics of quantum physics. However, additional relaying equipments are required for key distribution between remote nodes due to the limitation of physical transmission distance of a single photon. Furthermore, additional bottlenecks can be caused by relaying delay, link conflict because of the additional equipments. This paper describes a detailed design and implementation of key relay based on an open-source blockchain to support no limitation of physical transmission distance upon QKD. This paper includes communication channel, data structure and protocol that are required to create a blockchain smart contract. According to the experimental result, it achieves key distribution between two nodes without additional overhead.

※ 본 연구는 과학기술연구망 센터 KISTI, "양자암호기반 차세대 국가연구망 구축개발" 사업으로 진행되었습니다.
◆ First author : Department of Computer Engineering, Keimyung University, baksejin@kmu.ac.kr, 정회원
° Corresponding author : Department of Computer Engineering, Keimyung University, juht@kmu.ac.kr, 종신회원
* Student author. Department of Computer Engineering, Keimyung University, 5414395@stu.kmu.ac.kr
** KREONET center, KISTI, wjseok@kisti.re.kr, 정회원; livezone@kisti.re.kr, 정회원
논문번호 : 202010-253-D-RN, Received October 9, 2020; Revised December 6, 2020; Accepted December 11, 2020

Ⅰ. 서 론

GPU의 등장으로 시작된 병렬 컴퓨팅 기술의 급속한 발전과 양자 컴퓨팅의 연구의 진전에 따라, 수학적 복잡함에 기반하고 있는 고전적인 암호 알고리즘은 Shor의 알고리즘[1], Grover 알고리즘[2] 등의 연구결과가 나오면서 점차 보안 문제에 직면하고 있다[3]. 양자 컴퓨팅이 보편화되면, 고전적 암호화 알고리즘은 짧은 시간 내에 분석이 되므로, 새로운 키 분배 기술이 필요하게 된다.

이에 따라 최근 그 중요성이 부상하고 있는 양자키 분배(Quantum Key Distribution, QKD) 기술은 양자에 정보를 실어 원격지로 전달하고, 양자물리학의 법칙(중첩의 원리, 얽힘 상태)을 통해 물리적으로 위변조를 여부를 확인할 수 있어 신뢰성 있는 통신을 보장하여 차세대 키 분배 기술로 각광받고 있다[4]. QKD는 실제 장비가 개발되었으며 이를 기반으로 글로벌 양자망의 구축에 많은 진전이 있다[5]. 이러한 양자망이 현실적으로 널리 이용되기 위해서는 현재 QKD 장비가 가진 물리적 한계를 해결해야한다.

현재 양자키 분배 서비스가 직면한 가장 큰 문제는 양자 신호가 전달되는 물리적인 거리에 제약조건이 있다는 것이다[6]. 이를 해결하기 위한 몇 가지 표준안과 방안이 제시되고 있는데, 유럽전기통신표준협회(ETSI)에서 발표한 표준[7]에서는 QKD 시스템에 관련된 주요 통신 자원과 광 네트워크 기반 구조에서 어떻게 QKD를 배치하는가에 대한 내용이 제시되어 있으며, ETSI의 표준[8]에서는 애플리케이션과 QKD 키 관리 계층 간의 객체 기반 원격 기능 호출 방식의 API를 정의하고, QoS를 지원하는 데이터 스트림을 정의하였으며, 또 다른 ETSI 표준[9]에서는 HTTPS 기반 RESTful API 와 JSON 인코딩을 이용한 간단한 키 전달 API를 정의하고 있다.

최근 발표한 연구[10,11]에서는 블록체인을 기반으로 키 전달을 하는 방안이 제시되었으나, 데이터 저장소를 블록체인 기반으로 한다는 정도로 이론적으로 나타나 있어, 구체적인 구현 방법이 제시되지 않고 있다.

본 논문은 실제 상용 블록체인 Ethereum과 스마트 컨트랙트를 이용하여 QKD 릴레이를 직접 디자인 하여, 구체적인 방법을 제안하고, 수행 성능을 평가하는 데 의의가 있다.

본 논문은 다음의 순서로 작성된다. 2장에서 관련 배경 지식이 설명되며 3장에서 두 원격 노드간 키 교환 프로토콜이 정의된다. 4장에서는 실험결과를 나타내고 마지막 5장에서는 결론을 제시한다.

Ⅱ. 배경지식

2.1 양자키 분배 (Quantum Key Distribution, QKD)

1984년 C. H. Bennett 와 G. Brassard 에 의해 최초로 제안된 양자키 분배[4] 알고리즘(BB84) 는 고전적 암호화 기술들이 사용하는 수학적 복합성에 기반을 두지 않고, 양자의 물리적 특성을 이용하여 두 노드간 키를 교환하는 방법이다. [20]에서 양자키 분배에 대한 프로토콜을 설명하고, 5G 상용망 적용 등의 활용 사례를 발표한 바 있다.

양자암호는 보통 광자(Photon)을 이용하여 키를 교환하게 되는데 광자를 전달하는 안전한 채널과, 노드간 통신을 위한 추가적인 채널이 필요하며, 두 노드가 BB84 알고리즘을 수행하면서 생성되는 키를 공유하는 형태이다. 양자 송신자는 0,1의 랜덤 비트를 생성하여 편광 또는 스핀과 같은 양자 상태를 통해 해당 비트를 매핑시켜 전달한다. 이 때 송신자는 양자를 송신하는 시점에 양자를 임의의 편광 필터를 통과시켜 편광된 상태의 양자를 수신자로 보내게 된다. 편광필터는 X 방향 필터와 ＋ 방향 필터 중 임의로 선택하여 통과시킨다.

수신자 역시 임의로 편광필터를 선택하여 송신자가 보낸 양자를 통과시켜 값을 확인하게 되는데, 송신자와 수신자는 이후 별도의 일반 채널을 통해 서로가 선택한 필터의 종류를 확인한 후, 다른 필터를 통과시킨 값은 폐기하고, 같은 필터를 통과시킨 값을 사용하여 sifted 키를 생성 한다. 키 생성 시점에 임의의 키를 서로 공유하여 오류 율을 측정 하며, 일정 비율 이상의 오류 율이 계산될 경우, 공격으로 탐지되어 키 분배를 중지하며, 오류 율이 거의 없을 경우 나머지 키를 대칭 키로 사용한다.

외부환경에 취약한 광자는 물리적 전송거리가 길지 않으며 케이블을 통한 전송은 최대 140km – 200km 정도 까지 가능하기 때문에 최대 전송 거리보다 먼 거리의 노드들 간의 키 공유를 하려면 이를 중계 해주는 릴레이가 필요하다.

2.2 블록체인(Blockchain)

블록체인은[12] 중앙 서버 없이 데이터를 분산 저장하는 탈중앙형 데이터 저장 기술로, 든 블록체인 네트워크 참여자가 기록을 분산 저장하여 처리하는 기술로 참여자간 합의로 기록의 유효성을 검증하며 다음의 기술적 특징을 가진다.

- 탈 중앙성 (Decentralization) : 거래기록이 담긴 ledger를 중앙은행과 같은 제 3자에게 맡기지 않고 참여자들이 직접 검증, 승인 및 합의 과정을 통해 관리
- 투명성 (Transparency): 새로운 블록은 생성되는 동시에 모든 참여자에게 공유
- 불변성 (Immutability): 생성된 블록은 수정/삭제가 불가능.
- 가용성 (Availability): 블록체인 데이터는 모든 참여자의 노드에 분산 저장되므로, 높은 가용성을 가짐

블록체인은 1세대 블록체인으로 불려지는 Bitcoin을 통해 널리 알려지게 되었으며, Bitcoin 블록체인은 네트워크에 참여하고 있는 모든 유저들의 자산 변동 내역을 기록하고 모든 유저들이 해당 내역을 동기화하여 분산 기록한다. 이때 거래 내역은 Transaction 단위로 생성이 되는데, Transaction 이 모이면 Block 이 생성되고, 이 Block을 네트워크에 참여하고 있는 모든 노드들에게 전파하여 개인들의 거래내역이 투명하게 유지되는 특징을 가진다. 이 Block을 생성하고 전파를 시작하는 노드를 Miner 라고 하며, Miner 가 Block 에 포함시킨 Transaction 들만이 거래가 인정되기 때문에 Miner 의 선택을 받기 위해 Transaction을 생성할 때, 수수료를 같이 포함시켜둔다.

Bitcoin 은 UTXO (Unspent Transaction Output) 모델을 이용하여 잔액정보를 유지하며 이중 지불문제를 해결하기 위해 작업증명 (Proof of Work, PoW) 기반 합의 알고리즘을 채택하고 있다. 이중 지불 문제란 한 참여자가 동시에 두 군데 이상의 사용자 계정으로 송금을 하는 문제로 예를 들어 잔고 100원을 가진 사용자가 동시에 2명의 사용자에게 100원을 보낼 수 있는 문제를 의미하는데, 은행과 같이 중앙 제어 시스템이 있으면 동시성 제어로 인해 해당 문제가 발생할 수 없지만, 블록체인은 참여자들이 관리하기 때문에 문제 발생 소지가 있다.

이를 해결하는 것이 합의 알고리즘이며, 전체 네트워크에서 오직 거래를 인정하는 주체인 Miner 가 블록을 생성 하는 권한을 받게 되어, Transaction을 검증하여 이중 지불 문제가 발생되지 않도록 한다.

Bitcoin에서 사용하는 PoW 기반 합의 알고리즘은 조건에 맞는 해시 값을 생성하는 입력 값을 찾는 유저가 Miner 가 된다. 구체적으로 SHA256(버전, 이전 블록 해시, Merkle root, 시간, bits, Nonce) 의 결과가 target value 보다 낮은 값을 가질 수 있는 Nonce 값을 먼저 찾는 유저에게 블록 합의 승인 권한을 주는 방법을 취한다. Target value 는 bitcoin client 소스코드의 src/main.cpp 내의 GetNextWorkRequired() 함수에서 계산되며, 평균 10분마다 1개의 블록이 생성되도록 2주마다 난이도가 조정되는 특징이 있다.

1세대 블록체인이 단순히 자산거래에 이용되는 특징을 가졌다면 2세대 블록체인은 스마트 컨트랙트 개념이 추가된다. 스마트 컨트랙트는 1996년 Nick Szabo 가 발표한 “Smart Contract: Building Blocks for Digital Free Markets” 논문에서 처음 제안되었다. 이 개념은 인터넷상에서 전자상거래를 진행 할 때, 서로 모르는 사람들간 계약을 만들고 해당 계약이 진행될 수 있는 방법을 제시하고 있다.

2013년 Vitalik Buterin 이 제안한 Ethereum[13]에 도입된 스마트 컨트랙트는 블록체인을 블록의 저장 공간이라는 개념에서 코드를 구동시킬 수 있는 컴퓨팅 장치로 개념을 확장하여, Ethereum 의 암호화폐 단위인 Ether 의 거래내역 기록용도 뿐만 아니라, 프로그램을 구동시킬 수 있다.

예를 들어, 특정 일자에 돈을 납부하는 계약서를 구매자와 판매자가 서로 작성하는 경우, 전통의 계약서는 계약을 이행하지 않을 경우, 집행을 위한 추가적인 복잡한 법률적, 제도적 절차가 필요하다. 만약 특정 일자에 자동 송금이 될 수 있도록 스마트 컨트랙트 코드를 만들어 구동하면, 전통의 계약서와 달리 실제 송금이 이루어지기 때문에 자동적으로 계약이 이행되는 장점이 있다. 이는 Ethereum 은 내부에 EVM (Ethereum Virtual Machine) 이라는 스마트 컨트랙트 실행 환경을 갖고 있어, 스마트 컨트랙트 코드를 직접 수행시키기 때문에 가능하다. 보통 Solidity 로 작성되는 스마트 컨트랙트는 solc 컴파일러를 통해 EVM에서 수행되는 바이트 코드로 변환되고 블록체인에 기록되며, 일반적인 프로그래밍 언어의 함수 호출과 같이 스마트 컨트랙트의 주소 (Account)를 통해 해당 스마트 컨트랙트를 수행 시킬 수 있다. 또한 Solidity 는 튜링완전성(Turing- Completeness)를 지원하여 다양한 분야에 적용할 수 있다.

1세대 블록체인인 Bitcoin 은 초당 7개의 거래 (7 Transactions per second, TPS) 를 기록할 수 있으며, 2세대 블록체인 Ethereum은 초당 20개의 거래 (20 TPS)를 기록할 수 있어서 실제 금융활동에 적용되기 어렵다는 문제가 지속적으로 논의되어 왔다. 3세대 블록체인은 기존 1,2 세대 블록체인들의 단점으로 부각되는 성능 문제를 개선하고 있으며, 최근 2000 TPS 이상의 성능을 나타내는 Hyperledger[14], HashGraph[15] 등

이 관심을 받고 있다.

블록체인 기술은 블록체인 네트워크 참여 노드간 정보가 투명하게 공유되며, 변조가 불가능한 특성을 가지며, 또한 스마트 컨트랙트를 이용한 프로그래밍이 가능하기 때문에 신뢰성 있는 QKD 릴레이 매체로 적용이 가능하다.

Ⅲ. 설계 및 구현

본 논문은 기존 연구들[10,11]에서 제안한 블록체인 릴레이 기반 양자키 분배 기법을 실제 블록체인에 구현하여 평가하는데 그 목적이 있으며, 이 장에서는 Ethereum 블록체인에 QKD Relay 구축 설계와 각 모듈들의 세부 구현에 대해 설명하고 있다.

3.1 QKD 노드간 데이터 통신 채널

블록체인을 기반으로 QKD 릴레이를 구축하기 위해서는 릴레이 노드들은 블록체인 네트워크에 참여해야 하며, 또한 별도의 이벤트를 전달하는 채널이 필요하다. 그림 1은 블록체인 기반 QKD Relay를 위한 기본 통신 채널을 나타낸다.

기본적으로 각 인접 QKD 노드들은 기본 QKD Link로 양자키 분배를 진행할 수 있다. 그림 1에서 Node 1 과 Node 2는 서로 양자키 분배를 통해 공유키를 생성할 수 있게 된다. 그러나 Node1 과 Node 3은 직접 연결되지 않았기 때문에 직접 양자키 분배 프로토콜을 실행할 수 가 없다.

또한 각 QKD 노드들은 블록체인 네트워크의 참여 노드로 스마트 컨트랙트를 구동시켜 블록체인에 기록된 QKD 관련 정보를 읽어오거나 쓸 수 있다. 또한 QKD 노드들간 별도의 이벤트 채널이 동적으로 생성되는데 이는 각 노드들 간 키 교환 이벤트 시그널을 위한 용도로 사용된다. 예를 들어, Node 1 이 Node 3 과 비밀키를 공유하려면, Node 1 은 Node 2, Node 3 의 이벤트 채널을 생성하여 키 공유 메시지를 전달하게 된다.

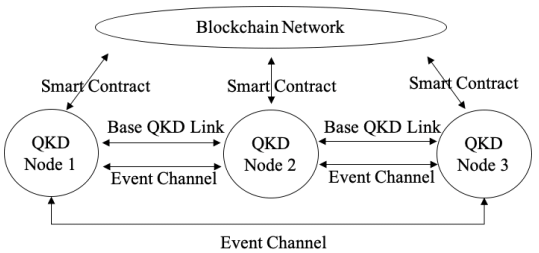

그림 1. 블록체인 기반 QKD Relay를 위한 기본 통신 채널
Fig. 1. Basic communication channels for blockchain QKD relay

본 논문에서는 Ethereum 블록체인을 기반으로 구현을 진행했으며, Solidity 로 스마트 컨트랙트를 작성하였다. 이벤트 채널은 각 노드들의 IP 주소를 기반으로 연결 할 수 있으며, 본 논문에서는 node.js를 이용하여 구현하였다.

3.2 XOR 기반 릴레이 노드간 키 공유 기법

본 논문은 기존 알려져 있는 XOR 기반의 키 공유 기법을 구현 하였다[11]. 그림 2는 4개의 노드 Node 1 ~ Node 4 가 릴레이로 연결되어 있는 상황을 나타내고 있다.

그림 2에서 키 Kn 은 QKD Node n 과 QKD Node n+1 이 QKD 프로토콜에 의해 생성된 공유키를 나타낸다. 이 상황에서 QKD Node 1 과 Node 4 가 서로 공유키를 생성하기 위해서는 XOR 의 특징을 활용한 수식 (1) 을 활용한다.

$$(K1 \oplus K2) \oplus (K2 \oplus K3) = (K1 \oplus K3) \quad (1)$$

즉 Node 2 가 (K1 ⊕ K2) 의 결괏값을 Node 1과 공유하고, Node 3이 (K2 ⊕ K3) 의 결괏값을 Node 1과 공유하면 두 값의 XOR 연산을 통해 Node 1은 (K1 ⊕ K3) 결괏값을 계산할 수 있게 된다.

$$K1 \oplus (K1 \oplus K3) = K3 \quad (2)$$

수식 (2) 에 나타난 것처럼 Node 1은 (K1 ⊕ K3) 결괏값을 자신이 소유하고 있는 K1 과 XOR 연산을 하여 K3 키 값을 계산 할 수 있게 된다.

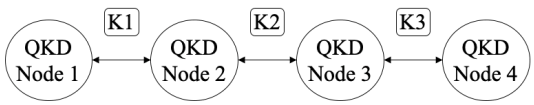

그림 2. 4개의 QKD Node 구성의 예시: K1부터 K3 은 인접한 두 Node간 공유된 비밀 키를 의미함
Fig. 2. An example of 4 QKD nodes: K1~K3 denotes shared secret key between adjacent two nodes

3.3 스마트 컨트랙트 구성 및 자료구조

제안하는 시스템을 구축하기 위해서는 QKD Node 등록 스마트 컨트랙트와 XOR 데이터 접근 스마트 컨트랙트가 필요하다.

QKD Node 등록 스마트 컨트랙트는 QKD Node 가 블록체인 네트워크에 새롭게 참여할 때 수행되는

표 1. QKD Node 등록을 위한 데이터 구조
Table 1. Data structure for QKD Node registration

Field name	Description
QKD Node ID	ID for QKD Node
QKD Node IP	IP address for QKD Node
Adjacent Node ID	ID for adjacent node
Adjacent Node IP	IP address for adjacent node

컨트랙트로써, 참여 대상 노드ID 및 IP 와 인접 노드 ID 및 IP를 기록 한다. QKD Node 는 하나 이상의 다른 QKD Node 와 연결되어 있기 때문에 신규 Node 가 블록체인 네트워크에 참여할 때 마다 QKD Node Network Topology 가 업데이트 된다. 표 1에 QKD Node 등록을 위한 데이터 구조가 나타나 있다.

예를 들어, 그림 2의 Node 구성 상황에서 Node 2

표 2. XOR 값 등록을 위한 데이터 구조
Table 2. Data structure for XOR value registration

Field name	Description
Session ID	A session ID for key sharing
QKD Middle Node ID	A Node ID for QKD Middle Node
Adjacent Node1 ID	A Node ID for QKD Adjacent Node 1
Adjacent Node2 ID	A Node ID for QKD Adjacent Node 2
Encroypted XOR value for Terminal Node 1	An encrypted XOR value for K1 and K2 K1 = a sharing key between Middle Node and adjacent Node 1 K2 = a sharing key between Middle Node and adjacent Node 2 Encrypted by the Public Key of Terminal Node 1
Public Key for Terminal Node 1	A public key of Terminal Node 1
Encrypted XOR value for Terminal Node 2	An encrypted XOR value for K1 and K2 K1 = a sharing key between Middle Node and adjacent Node 1 K2 = a sharing key between Middle Node and adjacent Node 2 Encrypted by the Public Key of Terminal Node 2
Public Key for Terminal Node 2	A public key of Terminal Node 2

가 블록체인에 등록되는 경우, Node 2는 Node 1과 인접되어 있으며 동시에 Node 3과도 인접되어 있기 때문에, 표 1의 데이터 구조 2개가 등록이 된다. 이 정보는 네트워크 토폴로지 확인을 위한 용도로 사용된다.

모든 노드들의 등록이 완료되면, 실제 Node 간 Key 공유를 진행할 수 있는데, 이 때 사용되는 자료구조가 표 2에 나타나 있다.

표 2를 참조하면, 키 교환에 필요한 XOR 데이터를 기록 하며, 이 정보는 스마트 컨트랙트를 통해 기록 및 열람이 된다. 단, 스마트 컨트랙트 내에 있는 데이터는 누구나 열람이 가능하므로, 기록될 XOR 데이터의 암호화가 필요하다. 본 논문에서는 공개키 기반 구조 (Public Key Infrastructure, PKI)를 이용하고 있으며, XOR값을 열람할 양 끝 단말 Node 들의 Public Key 를 이용하여 암호화를 진행하였다.

예를 들어, 그림 2의 Node 구성에서 Node 1과 Node 4가 키를 교환 할 경우, 필요한 정보는 릴레이 Node 2 가 생성하는 (K1 ⊕ K2) 값과 Node 3이 생성하는 (K2 ⊕ K3) 값이다. 따라서 릴레이 Node 인 Node 2와 Node 3은 스마트 컨트랙트를 이용하여 암호화된 XOR 값을 기록한다. 표 3에 실제 Node 2가 기록하는 데이터의 예시를 설명한다. Node 2의 경우 인접 Node 가 Node 1 과Node 3 이므로, Adjacent Node1 ID 와 Adjacent Node2 ID 에 각각 Node 1, Node 3이 기록 된다.

표 3. Node 2의 XOR 값 등록 예시
Table 3. An example of XOR value registration for Node 2

Field name	Value
Session ID	Session #1
QKD Middle Node ID	Node 2
Adjacent Node1 ID	Node 1
Adjacent Node2 ID	Node 3
Encroypted XOR value for Terminal Node 1	(K1 ⊕ K2) encrypted by public key of K1
Public Key for Terminal Node 1	A public key of K1
Encrypted XOR value for Terminal Node 2	(K1 ⊕ K2) encrypted by public key of K4
Public Key for Terminal Node 2	A public key of K4

3.4 시스템 구조도

그림 3은 제안하는 시스템의 전체 구조도를 나타낸다.

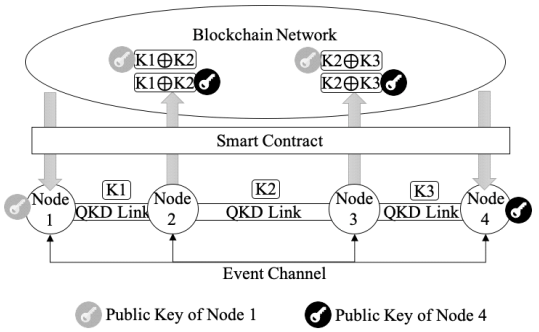

그림 3. 시스템 구조도
Fig. 3. System Architecture

제안 시스템은 스마트 컨트랙트와 이벤트 채널을 통해 Node 간 통신을 처리한다. 키 공유를 할 Node 들은 표1에 기술한 Node 정보를 바탕으로 Network Topology 상 릴레이 Node를 검출 한 후, Event Channel을 통해 해당 릴레이 Node 들에게 키 공유 세션 ID와 XOR 전달 신호를 보내면, 각 릴레이 노드들은 양 단말 노드의 공개키를 이용하여 암호화된 XOR 값을 스마트 컨트랙트로 기록 한다. 이후, 양 단말 Node 인 Node 1 과 Node 4 는 스마트 컨트랙트에서 세션 ID를 기반으로 암호화된 XOR 값을 확인 하고, 각 Node 의 Private Key를 이용하여 복호화 하여 공유 Key를 구한다.

3.5 키 공유 프로토콜

그림 4는 본 논문에서 제안한 키 공유 프로토콜을 나타낸 시퀀스 다이어그램이다. Node 1 이 Node 4 와 키 공유를 하고 싶은 경우, 우선 Node 1 이 Node 4 에게 키 공유를 하는 신호를 보내고 스마트 컨트랙트 로부터 Node 1부터 Node 4 까지의 네트워크 토폴로지와 Session ID를 받아온다.

네트워크 토폴로지를 구한 결과 Node 2와 Node 3 이 Relay node 로 확인이 되면, 해당 Relay Node 들에게 Session ID 와 XOR 키 전달 신호를 보내고, 해당 Relay Node들은 Node 1, Node 4 의 Public Key 를 이용하여 두 키의 XOR 값을 암호화해서 스마트 컨트랙트에 기록 한다. 각 노드들이 스마트 컨트랙트 기록이 끝이 나고 응답을 Node 1에게 전달하면 Node 1 은 암호화된 XOR 값을 읽어오고, Node 4 에게도 릴레이 전달이 끝났음을 알려준다. 이후 Node 4 역시 암호화된 XOR 값을 읽어온 후 자신의 Private Key를 이용하여 복호화 하여 키를 받아온다.

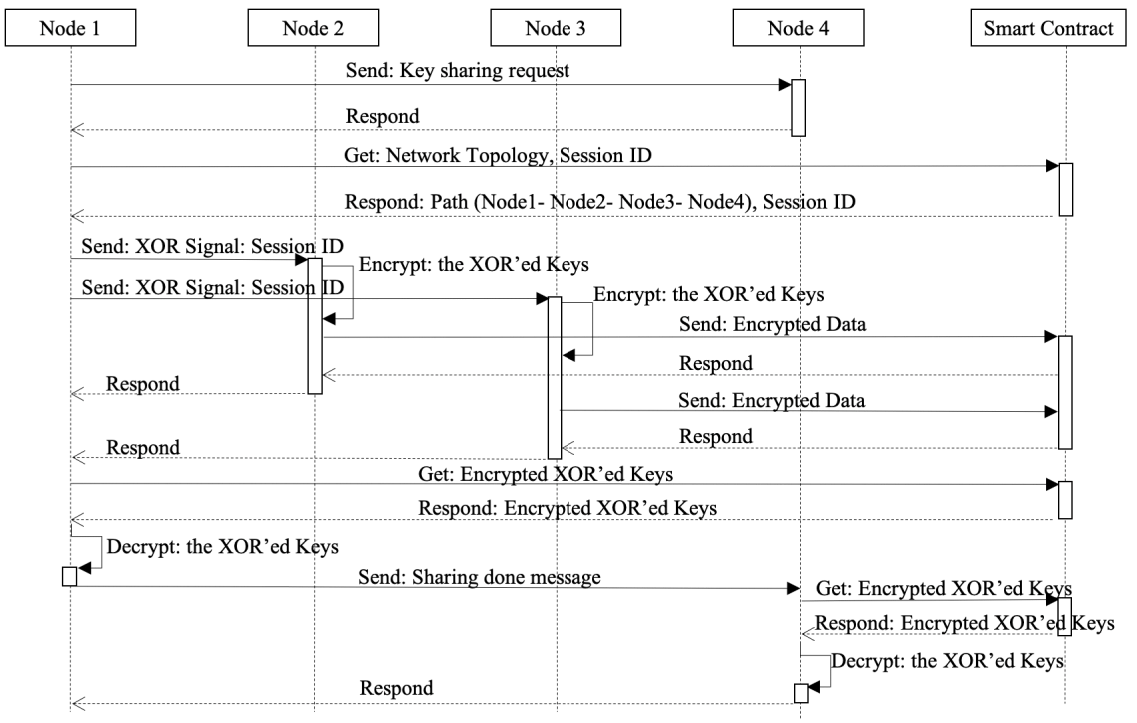

그림 4. 키 공유 프로토콜 시퀀스 다이어그램: Node 1과 Node 4 가 비밀키를 공유하는 시퀀스
Fig. 4. A sequence diagram for the key sharing protocol: a sequence of sharing a secret key between Node 1 and Node 4

Ⅳ. 실 험

제안한 시스템은 Ethereum blockchain 과 Solidity 를 이용하여 구현 하였으며, 이벤트 채널은 Node.js[19] 로 구현 하였다. 표 4에 자세한 실험환경이 나타나 있다. 네트워크 영향을 최소화하고 순수 스마트 컨트랙트 로직 및 블록체인의 오버헤드만 측정하기 위해 Docker[16]를 이용 하였으며, 스마트 컨트랙트 개발환경으로 Truffle[17], 블록체인 구축환경으로 Ganache[18] 를 이용하였다. Truffle 개발환경은 Solidity 코드를 로컬 환경에서 쉽게 컴파일하고 배포할 수 있어서 빠른 개발이 가능하며, Ganache 는 Ethereum network를 구성하여 스마트 컨트랙트를 실행할 수 있도록 해주는 프로그램으로 TestRPC 라는 가상환경을 생성해 준다. 본 실험은 QKD 장비가 서로 인접한 두 노드간 비밀키를 생성하고 공유하고 있는 상황에서 진행한다.

그림 5는 두 단말 노드간 키 공유 지연 시간을 측정 한 실험 결과 이다. 실험은 기본 두 단말 노드가 키를 공유할 때 두 단말 노드 사이에 릴레이 노드의 개수가 1개부터 최대 5개 까지 있는 경우의 키 공유 지연 시간을 측정 하였다. 실험결과 릴레이 노드가 하나씩 늘어날 때 마다 거의 비슷한 지연 시간이 늘어나며 되며, 이것은 블록체인 트랜잭션이 처리되는 시간과 동일하다. 일반적으로 Ethereum 는 20 TPS 정도의 성능을 보여주며 이것은 개별 트랜잭션당 처리 시간이 약 50ms 내외의 시간이 필요한 것을 의미한다. 따라서, 고성능 블록체인을 기반으로 해당 키 릴레이 환경을 구축할 경우 더욱 빠른 성능을 나타낼 수 있음을 알 수 있다.

표 4. 실험환경
Table 4. Experimental Environment

Environment	Specification
CPU	Intel i5 2.4Ghz
RAM	8GB DDR3
Host OS	Ubuntu 20.04 LTS
Docker	v19.03.12
Docker Image	Ubuntu 20.04 LTS
Truffle	v5.0.2
Ganache-cli	v6.9.1
Ganache-core	v2.10.2
Node.js	v12.11.1

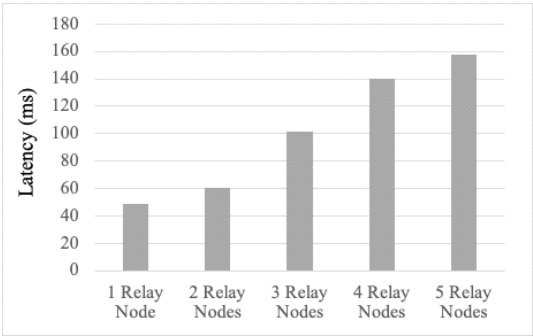

그림 5. 키 공유 지연 시간 측정
Fig. 5. Measurement of key sharing latency

Ⅴ. 보안성 분석

본 논문은 [11]에서 제시한 XOR 기반 키 공유 기법을 실제 Ethereum 에 구현하였다. Ethereum 은 누구나 접근 가능한 블록체인으로 릴레이 과정에서 양자키 유출에 대한 보안문제를 고려해야한다, 제안한 연구는 [11]에서 제시된 XOR 기반 키 교환 방법의 보안성 분석에 제시된 것과 마찬가지로, 릴레이 과정에서 양자키가 노출될 위험은 없다.

만약 양자키의 플레인 키값이 변형 없이 누구나 접근 가능한 블록체인에 기록된다면 해당키가 유출되어 버리는 심각한 보안 문제가 발생한다. 그러나 본 연구는 서로 연관성이 없는 임의의 값으로 생성된 두 양자키 값이 XOR 된 상태로 블록체인에 기록되기 때문에 C. Shannnon 의 Perfect secrecy 의 원리에 따라[19] 해당 XOR 값은 완벽하게 안전한 암호문이 되어, 원래의 양자키 값이 유출 될 수 없다. 따라서, 본 연구에서 구현한 릴레이는 양자키 유출의 보안문제로부터 안전하다.

Ⅵ. 결 론

본 논문은 블록체인을 이용하여 QKD 릴레이에 사용할 수 있는 키 공유 방법에 대한 구체적인 디자인을 제시하고, Ethereum을 기반으로 실제 구현 및 실험을 진행하였다. 릴레이 노드 개수가 늘어남에 따라 추가적인 오버헤드 없이 블록체인 자체의 트랜잭션 처리 시간만으로 릴레이가 완료되는 것을 확인할 수 있었으며, 이것은 Ethereum 이 아닌 다른 고성능 블록체인 플랫폼에서 구현할 경우 훨씬 높은 성능을 얻을 수 있다는 것을 알 수 있다.

본 논문은 Ethereum을 기반으로 구현 하였지만, 구

체적인 데이터 구조 및 통신 구조를 기술하여 다른 블록체인 플랫폼에서도 쉽게 구현이 가능할 것이라 기대한다.

제안하는 방법은 전통적인 암호화 방식인 PKI를 기반하고 있는 블록체인을 활용하고 있으나, 포스트 양자 시대에 구현될 Lattice 등의 암호체계를 사용하는 블록체인플랫폼에도 쉽게 구현이 될 수 있는 설계를 제안하였다. 이를 바탕으로 포스트 양자 시대에도 오버헤드 없는 안전한 키 공유를 할 수 있기 때문에 그 중요성이 더욱 강조 되는 연구이다.

References

[1] P. W. Shor, "Algorithms for quantum computation, discrete log and factoring," *35th Annu. Symp. Foundations of Comput. Sci. Proc.*, pp. 124-134, Santa fe, NM, USA, Nov. 1994.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," *28th Annu. ACM Symp. Theory of Comput. Proc.*, pp. 212-219, Philadelphia, Pennsylvania, Jul. 1996.

[3] X. Y. Wang, D. G. Feng, et al., "Collisions for hash functions MD4, MD5, HAVAL-128 and RIPRMD," in *Proc. Crypto'04*, Santa Barbara, Aug. 2004.

[4] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput., Syst. and Sign. Process.*, pp. 175-179, Bangalore, India, 1984.

[5] C. Simon, "Towards a global quantum network," *Nature Photonics*, vol. 11, pp. 678-680, 2017.

[6] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum Cryptography," *Rev. Mod. Phys.*, vol. 74, no. 1, p. 145, Mar. 2002.

[7] Romain Alleaume, "*Quantum Key Distribution (QKD): Device and Communication Channel Parameters for QKD Deployment*," ETSI, GS QKD 012 V1.1.1, Feb. 2019.

[8] J. Davila, et al., "*Quantum Key Distribution (QKD): Application Interface*," ETSI, GS QKD 004 V1.1.1 Dec. 2010.

[9] Y. Tanizawa, "*Quantum Key Distribution (QKD): Protocol and data format of REST-based keydelivery API*," ETSI, GS QKD 014 V1.1.1, Feb. 2019.

[10] W. Dur, H. J. Briegel, et al., "Quantum repeaters based onentanglement purification," *Phys. Rev. A*, vol. 59, pp. 169-181, Jan. 1999.

[11] H. Chen, "Quantum relay blockchain and lis applications in keyservice," *ICCSP 2020*, pp. 95-99, Nanjing, China, Jan. 2020.

[12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Manubot*, 2019.

[13] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1-32, 2014.

[14] E. Androulaki, et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, no. 30, pp. 1-15, Apr. 2018.

[15] L. Baird, "*The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance*," Swirlds, Inc. Technical Report SWIRLDS-TR-2016-01, 2016.

[16] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux J.*, vol. 239, no. 2, 2014.

[17] *truffle*, https://www.trufflesuite.com

[18] *ganache*, https://www.trufflesuite.com/ganache

[19] S. Tilkov and S. Vinoski, "Node. js: Using JavaScript to build high-performance network programs," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 80-83, 2010.

[20] H. Choe, M. Na, M. Yoon, D. Cha, D.-H. Sim, and J. Lee, "Overview of quantum key distribution(QKD) over mobile network," in *Proc. Symp. KICS*, pp. 950-953, Jun. 2019.

[21] C. Shannon, "Communication theory of secrecy of systems," *Bell System Technical J.*, vol. 28, pp. 656-715, 1949.

박 세 진 (Sejin Park)

2007년 2월 : 금오공과대학교 컴퓨터소프트웨어 학사
2016년 2월 : 포항공과대학교 컴퓨터공학과 석사
2016년 11월 : SK Telecom 종합기술원 매니저
2018년 3월~현재 : 계명대학교 컴퓨터공학부 조교수
<관심분야> 운영체제, 블록체인, 시스템소프트웨어, IoT, 양자암호통신

안 재 욱 (Jaewook Ahn)

2016년~현재 : 계명대학교 컴퓨터공학과 학사과정
2020년~현재 : 계명대학교 System Software Laboratory 연구원
<관심분야> 양자암호통신, 블록체인, 운영체제, 네트워크 관리

석 우 진 (Woojin Seok)

1998년 2월 : 경북대학교 컴퓨터공학과 학사
2003년 1월 : UNC Chapel Hill Computer Science M.S.
2008년 8월 : 충남대학교 컴퓨터공학과 박사
2012년~2020년 : UST 겸임교원
2018년~현재 : KISTI 과학기술연구망센터장
2019년~현재 : KNOM 연구회 부위원장
2020년~현재 : SDN/NFV 포럼 PoC 분과장
<관심분야> TCP 프로토콜, 양자암호통신, 비면허대역 무선통신

이 원 혁 (Wonhyuk Lee)

2001년 2월 : 성균관대학교 공과대학 전기전자및컴퓨터공학부 학사
2003년 2월 : 성균관대학교 공과대학 컴퓨터공학과 석사
2010년 8월 : 성균관대학교 공과대학 전자전기컴퓨터공학과 박사
2003년 3월~현재 : 한국과학기술정보연구원
2020년~현재 : SDN/NFV 포럼 PoC 분과 Q-KaaS WG 의장
<관심분야> 네트워크 관리, 망 성능측정, 양자암호기반 통신망 구축 관리기술

주 홍 택 (Hongtaek Ju)

1989년 8월 : 한국과학기술원 전자계산학과 학사
1991년 8월 : 포항공과대학교 컴퓨터공학과 석사
1997년 8월 : 대우통신종합연구소 선임연구원
2002년 2월 : 포항공과대학교 컴퓨터공학과 박사
2002년 9월~현재 : 계명대학교 컴퓨터공학부 교수
<관심분야> 네트워크 및 시스템 관리, IoT 관리, SDN 네트워크 관리, 블록체인 모니터링 및 분석