Journal of Korea Multimedia Society Vol. 25, No. 10, October 2022(pp. 1416-1432)
https://doi.org/10.9717/kmms.2022.25.10.1416

출입문 보안을 위한 블록체인 기반의 출입코드키 검증 서비스 모델

홍기현†, 이병문††

An Access Code Key for Verification Service Model on the Blockchain in a Door Security

Ki Hyeon Hong†, Byung Mun Lee††

ABSTRACT

The access control system is a system that allows users to selectively enter the building by granting an access key to the user for security. Access keys with weak security are easily exposed to attackers and cannot properly perform the role that authenticates users. Access code keys should be protected from forgery or spoofing. For this reason, access key verification service models is important in security. However, most models manage all access keys on one central server. This method not only interrupts all services due to server errors, but also risks forgery and spoofing in the process of transmitting access keys. Therefore, blockchain algorithms are used to reduce this risk. This paper proposes a blockchain-based access key verification service model that used distributed stored blockchain gateways on storing access keys and authenticates the user's identity based on them. To evaluate the performance of this model, an experiment was conducted to confirm the performance of the access key forgery recovery rate and the blockchain network performance. As a result, the proposed method is 100% forgery recovery rate, and the registration and verification process is evaluated at 387.58 TPS and 136.66 TPS.

Key words: Access Code Key, Blockchain Network, Door Security, Verification Service Model

1. 서 론

출입통제 시스템에서는 건물의 보안과 방범, 그리고 시설 내의 정보의 유출 방지를 위해, 출입권한이 부여된 사용자를 출입코드키로 인증하여 선별적으로 출입을 허가한다[1,2]. 그러나 공격자에게 위변조되거나 탈취된다면 출입코드키는 사용자 인증과 같은 정보 보호의 역할을 제대로 수행할 수 없다. 따라서, 사용자를 인증하기 위한 출입코드키는 위변조나 탈취로부터 안전하게 보호되어야 하며, 이를 위해 다양한 출입코드키 검증서비스 모델이 제안되었다.

기존의 출입코드키 검증서비스 모델에서는 중앙 서버에서 출입코드키를 발급하고 관리하며, 사용자가 건물에 출입을 하려면 발급받은 출입코드키를 중앙의 서버로 전송하여 인증을 받는다[3,4]. 이때, 비밀번호 공격자는 리플레이 공격(replay attack), 위장 공격(impersonation attack), 그리고 중간자 공격(man-in-the-middle attack)와 같이 전송 과정에서

※ Corresponding Author : Byung Mun Lee, Address: (13120) 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do, Korea, TEL : +82-31-750-4756, FAX : +82-31-750-4756, E-mail : bmlee@gachon.ac.kr
Receipt date : Jun. 23, 2022, Revision date : Oct. 7, 2022
Approval date : Oct. 20, 2022

† Dept. of IT Convergence Engineering, Graduate School, Gachon University (E-mail : ghdrlgus96@naver.com)
†† Dept. of Computer Engineering, College of IT Convergence, Gachon University
※ This work was supported by the Technology development Program funded by the Ministry of SMEs and Startups(MSS, Korea) (Grants No. S2957039, S3229617)

사용자의 출입코드키를 탈취할 수 있으므로, 기밀성을 보장하기 위해 HTTPS, SSL과 같은 암호화 프로토콜이 사용된다[5,6]. 이 외에도 중앙 서버의 공개키로 출입코드키를 암호화하며, 서버의 개인키를 이용하여 복호화하는 비대칭 암호화 알고리즘을 이용한 사례도 있다[7]. 그러나 이러한 기존의 검증서비스 모델은 전송 과정에서의 보안에만 초점이 맞춰진 한계가 있어 서버에 저장된 출입코드키의 가용성과 무결성을 보장할 수 없다. 예를 들어, 서비스 거부 공격으로 출입코드키를 관리하는 중앙 서버를 마비시켜, 출입통제 시스템의 전체 서비스를 무력화하는 침해를 들 수 있다[8]. 또한, 서버에 저장되어 있는 출입코드키가 위변조 된다면, 정상적인 사용자라도 인증 실패로 출입통제 시스템에 접근할 수 없다[9]. 이러한 점에서 볼 때 보안성이 보장되어도, 출입코드키를 저장하고 검증하는 과정에 문제가 생긴다면, 공격자에 의해 출입통제 시스템이 쉽게 무력화되는 한계가 있어, 중앙의 단일 서버에서 출입코드키를 직접 발급하고 관리하는 것은 보안에 한계가 있다.

이러한 한계점을 해결하기 위해, 블록체인 기반의 정보 보안 기술을 이용한다. 블록체인은 데이터의 무결성을 검증하고, 블록체인 네트워크의 각 노드에 신뢰할 수 있는 정보를 분산 저장하므로, 노드의 장애로 인한 시스템의 작동 중지로부터 안전하게 보호된다[10,11]. 또한, 블록체인 네트워크에 기록된 모든 정보는 전자서명되어 공격자가 기록된 정보를 위변조 할 경우에 블록체인의 무결성을 감지하고 분산 저장된 다른 노드의 블록체인으로 위변조에 대한 복구를 수행한다[12]. 따라서, 공격자가 블록체인에 기록된 출입코드키를 위변조하는 것이 불가능하다. 특히, 개인을 인증하기 위한 서버와 실제 정보가 저장되는 블록체인을 구조적으로 분리하여 공격자의 개인 정보 탈취 시도로부터 안전하게 보호한다[13].

따라서, 본 논문에서는 중앙의 서버에서 사용자를 인증한 다음, 출입통제 게이트웨이에 출입코드키를 저장하고 관리하는 블록체인 기반의 출입통제용 출입코드키 검증서비스 모델을 제안하고자 한다. 이 모델에서는 출입코드키의 등록과 검증 서비스를 제공하는 출입통제 게이트웨이를 블록체인의 노드로 구축한다. 출입통제 게이트웨이로 Raspberry Pi 4 model B를 사용하며, 각각의 게이트웨이는 신뢰할 수 있는 출입코드키를 전자서명하고 배포하기 위해 고유한 개인키와 공개키를 소유한다. 개인키는 출입코드키의 무결성을 검증하기 위해 암호화하는 데에 사용하며, 공개키로 전자서명하여 블록을 등록한 게이트웨이를 증명한다. 만약, 공격자가 게이트웨이로 위장하여 출입코드키의 등록을 시도하여도 거짓된 게이트웨이가 요청한 출입코드키를 복호화할 수 있는 공개키는 블록체인 네트워크에 존재하지 않는다. 따라서, 공격자의 출입코드키 등록 요청이 거부된다. 이렇듯, 사용자의 출입코드키를 중앙통제서버에 등록하는 대신에 블록체인 게이트웨이로 암호화된 출입코드키를 등록하고 각 노드들로 분산 배포를 시켜 안전한 관리가 가능하게 한다.

일단 블록체인 네트워크에 저장되었다면 게이트웨이로 검증을 요청할 수 있으며, 요청받은 게이트웨이는 블록체인에 등록되어 있는 출입코드키를 복호화하여 사용자의 출입 권한을 검증하도록 한다. 본 논문에서는 출입코드키와 도어락 ID를 블록에 생성하고 체인화하여 보안성을 강화시키는 방법을 제안하였으며, 탈취로부터 보호하기 위해 해시화하여 저장한다.

제안한 모델의 효용성을 검증하기 위해 두가지 실험을 하고자 한다. 첫째는 출입코드키의 위변조 복구율 실험이다. 정상적인 출입코드키를 블록체인 네트워크에 분산저장해 놓았을때 무결성 침해공격을 시도하여 의도적으로 위변조한다면 어느정도 복구가 가능한지를 확인하는 실험이다. 복구된 블록의 비율이 높을수록 복구율이 높으며, 복구율은 효용성에 영향을 주므로 중요한 실험이다. 둘째는 블록체인 네트워크의 성능을 확인하는 실험이며, 기존 방식들과의 성능평가를 비교하여 그 우수성을 확인하고자 한다.

본 논문의 2장에서는 기존에 제안된 출입코드키 검증서비스 모델을 고찰하고 블록체인 기반의 정보 보안 기술에 대해 살펴본다. 3장에서는 블록체인 기반의 출입통제 검증모델을 제시하고 출입코드키 등록과 검증 과정을 정의한다. 4장에서는 제안한 출입통제 검증모델의 효용성을 확인하기 위한 실험을 수행하고 결과를 분석한다. 마지막 5장에서는 결론을 맺는다.

2. 관련 연구

2.1 출입통제 시스템 동작 과정

일반적으로 사용자(User)가 출입통제 시스템에

출입코드키를 등록하고 검증하는 과정은 Fig. 1과 같다[14,15]. 사용자는 Fig. 1과 같이 사용자 인증 서버로 출입코드키의 등록을 요청한다. 그 후, 사용자는 출입 권한을 얻기 위해 출입통제 게이트웨이로 출입코드키의 검증을 요청하며, 게이트웨이는 서버를 통하여 검증 결과를 받는다. 사용자는 출입 권한에 따라 도어락을 제어하거나 실패한다.

이때, 출입통제 시스템의 공격자는 시스템에 접근하기 위해 다양한 방법으로 사용자의 출입코드키를 탈취한다. 예를 들어, Fig. 1의 출입코드키를 서버에 등록하는 과정에서 공격자는 사용자 인증 서버로 위장하여 사용자의 출입코드키를 중간에 탈취하는 중간자 공격을 시도한다[16]. 중간자 공격으로 탈취된 출입코드키는 위변조되어 서버에 전송되거나 키 교환 과정에서 암호화에 사용하는 비밀키를 탈취할 수 있다[17]. 또한, 탈취된 비밀키는 위장 공격에 사용되며, 탈취된 로그와 함께 리플레이 공격이 가능하다[18]. 이러한 점에서 전송되는 출입코드키는 공격자에게 쉽게 탈취되며, 공격자는 탈취한 출입코드키를 이용하여 도어락이나 게이트웨이에 접근할 수 있다[19]. 이렇듯 보안성이 취약한 출입코드키는 사용자 인증과 같은 정보 보호의 역할을 제대로 수행할 수 없으므로, 출입코드키는 위변조와 탈취로부터 안전하게 보호되어야 한다. 이러한 한계점을 해결하기 위해 출입코드키의 전송을 안전하게 보호할 수 있는 다양한 출입코드키 검증서비스 모델이 제안되었다.

2.2 기존의 출입코드키 검증서비스 모델과 한계점

기존의 스마트 홈과 같은 시스템에서의 출입코드키의 보안은 시스템 사용자의 치안과 안전에 직접적인 영향을 주므로, 출입코드키를 위변조나 탈취로부터 보호하는 연구가 진행되어 왔다[20]. 예를 들어, 출입코드키를 전송하는 과정에서 안전한 세션을 위해 SSL, HTTPS와 같은 암호화 프로토콜을 이용하거나, 출입코드키를 공개키로 암호화하여 전송할 수 있다[5-7]. 그중에서 서버의 공개키로 출입코드키를 암호화하는 방식은 다음 Fig. 2와 같다. Fig. 2의 (a)에서 출입코드키의 안전한 전송을 위해 서버의 공개키는 신뢰할 수 있는 게이트웨이로 배포된다. 게이트웨이는 서버의 공개키로 출입코드키를 암호화하여 검증을 요청하며, 서버는 개인키로 암호화된 출입코드키를 복호화한다. 이는 서버에 존재하는 사용자의 출입코드키 목록과 비교하여 목록에 일치하는 출입코드키가 존재하는지 판단한 후, 출입 권한을 부여한다.

그러나, 기존의 출입통제 시스템은 단순히 출입코드키의 전송 과정에서의 보안에만 초점이 맞춰져 있다. 이는 출입코드키의 전송 과정에서의 기밀성을 보장하지만, 서버에 관리되는 출입코드키에 대한 위협 모델에 취약하다. 예를 들어, 공격자는 출입코드키의 탈취뿐만이 아니라 저장되어 있는 출입코드키의 위변조를 시도할 수 있다. Fig. 2의 (b)는 공격자가 인증서버의 출입코드키 목록을 위변조한 상황이다. 이러한 경우에 게이트웨이는 사용자의 출입코드키를 올바르게 암호화하여 전송하여도 복호화된 출입코드키는 서버가 가진 목록과 일치하지 않으므로, 출입코드키의 검증에 실패한다[9]. 또한, Fig. 2의 (c)와 같이 공격자가 서비스 거부 공격을 시도하여 서버가

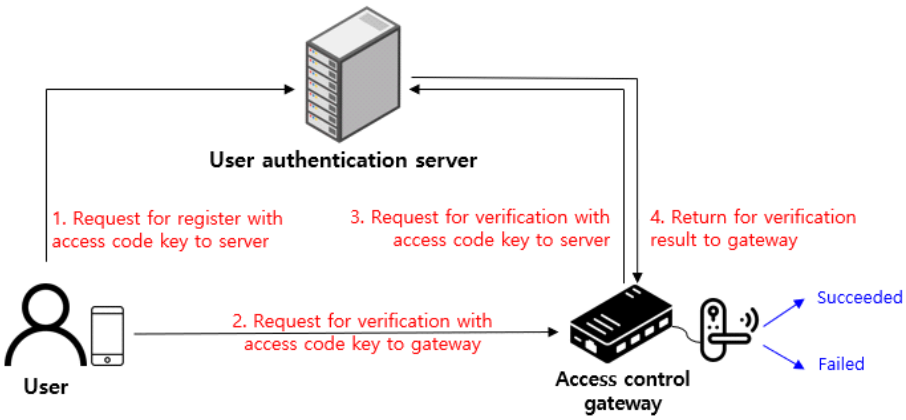

Fig. 1. Operation process of access control system.

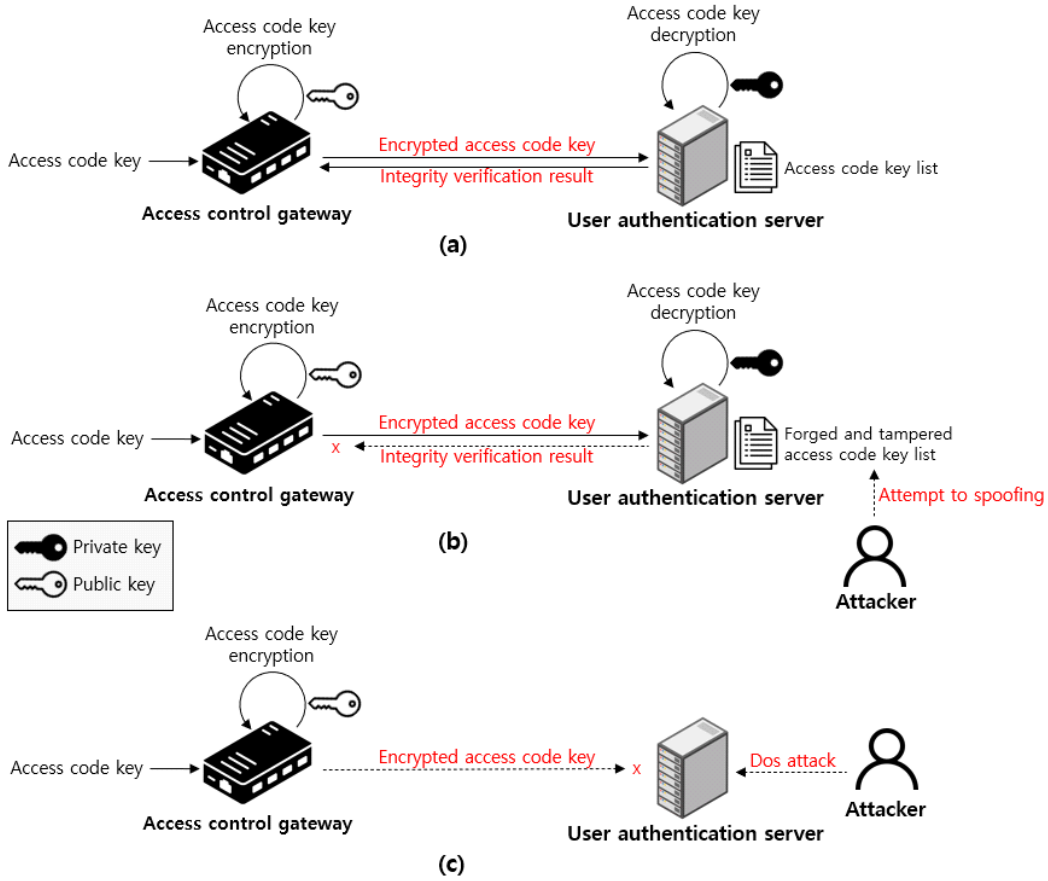

Fig. 2. Existing model for access code key verification service. (a) Access code key verification process, (b) Spoofing attempts by system attacker, and (c) Attempted Dos attack by system attacker.

마비된다면, 출입통제 시스템의 전체 서비스가 무력화되어 출입통제 서비스를 제공할 수 없다[8].

이러한 점에서 출입코드키의 기밀성이 보장되어도, 등록되어 있는 출입코드키나 서버에 장애를 발생시키는 위협 모델로 공격자는 쉽게 출입통제 시스템을 무력화할 수 있는 한계가 있다. 이러한 한계점을 해결하려면 출입코드키의 기밀성 뿐만이 아니라, 저장하고 관리되는 출입코드키의 무결성과 검증 서비스의 가용성에 대한 보안 모델이 필요하다. 이를 해결하기 위한 방법 중 하나로 블록체인을 이용한 기술이 있다.

2.3 블록체인 기반의 정보 보안 기술

블록체인은 블록체인 네트워크에 참여 중인 노드에서 생성한 정보를 체인으로 연결하는 정보 관리 기술이며, 모든 노드에 공유하고 분산 저장하는 데이터베이스이다[21,22]. 생성한 정보는 위변조로부터 안전하게 보호하기 위해 체인으로 연결하는데, 이는 Fig. 3과 같다[23]. Fig. 3에서 각 블록은 정보가 저장되는 트랜잭션(Transactions)과 블록의 무결성을 검

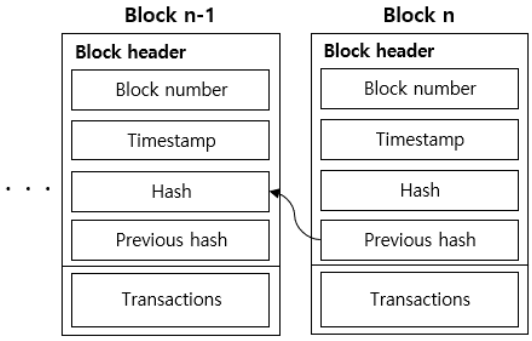

Fig. 3. Structure of blockchain for the prevention of counterfeiting.

증하고 유지하기 위한 블록 헤더(Block header)로 구분한다. 블록 헤더에는 생성한 블록의 순서인 블록 번호(Block number), 블록을 생성한 타임스탬프(Timestamp), 그리고 블록의 무결성을 유지하고 검증하기 위한 해시(Hash)와 이전 해시(Previous hash)로 구성한다. 특히, 해시는 이전 해시와 트랜잭션을 기반으로 생성하여 이전 블록이나 트랜잭션에 위변조가 발생하는 것을 감지한다[24].

생성된 블록은 블록체인 네트워크에 배포되어 관리하는데, 이는 Fig. 4와 같다. Fig. 4은 크게 두 단계로 구분하는데, 블록을 생성하고 블록체인에 등록하는 과정과 생성된 블록의 무결성을 검증하고 등록하는 과정이다. Step 1에서 새로운 정보인 “password123@”를 저장한다면 “iloveyou”가 저장된 블록헤더의 해시값과 저장할 정보인 “password123@”를 이용하여 새로운 블록헤더 해시값을 구한다. 이때, 신뢰할 수 있는 블록을 생성하고 배포하기 위해 합의 알고리즘을 사용하는데, 현재 가장 많이 사용되는 합의 알고리즘은 PoW(Proof of Work)이다[25]. PoW에서는 넌스(nonce)를 추가하여 해시값을 구하는데, 계산된 해시값이 일정 범위 이하가 될 때까지 넌스를 조금씩 변경한다. 합의 알고리즘에 만족한다면 생성된 블록을 블록체인에 저장하며 네트워크에 배포한다.

Step 2에서 블록을 배포받은 블록체인 노드는 동일한 합의 알고리즘을 통해 신뢰할 수 있는 블록인지 판단 후 블록을 블록체인에 저장한다. 이것이 블록을 생성하고 전파하는 과정이며, 블록체인 네트워크에 참여 중인 모든 노드는 동일한 블록체인을 갖는다. 특히, 블록체인의 모든 블록은 서로 체인으로 연결되므로 특정 블록을 위변조하기 위해 전체 블록의 해시를 계산하는 것은 막대한 연산이 발생한다.

또한, 생성한 블록을 블록체인에 참여 중인 모든 노드에 배포하고 동기화하므로, 노드는 동일한 블록체인을 소유한다[26]. 따라서, 공격자가 서비스 거부 공격으로 특정 노드를 무력화하여도, 다른 노드에서 동일한 정보를 제공할 수 있으므로 출입통제 서비스의 가용성을 보장할 수 있다[26].

그러나, 기존의 합의 알고리즘은 매우 많은 연산량이 필요하기 때문에 사용자의 출입코드키를 저장하고 검증하는 데에 적합하지 않다. 따라서, 본 연구에서는 사용자가 출입코드키를 등록하고 검증하는 출입통제용 출입코드키 검증서비스 모델을 제시하며, 출입통제 게이트웨이에 저장된 출입코드키의 위변조와 탈취로부터 무결성과 가용성을 보장하기 위한 블록체인 합의 알고리즘을 제안하고자 한다.

3. 블록체인 기반의 출입통제 검증모델

3.1 출입코드키 검증서비스 모델

제안한 블록체인 기반의 출입통제 검증모델을

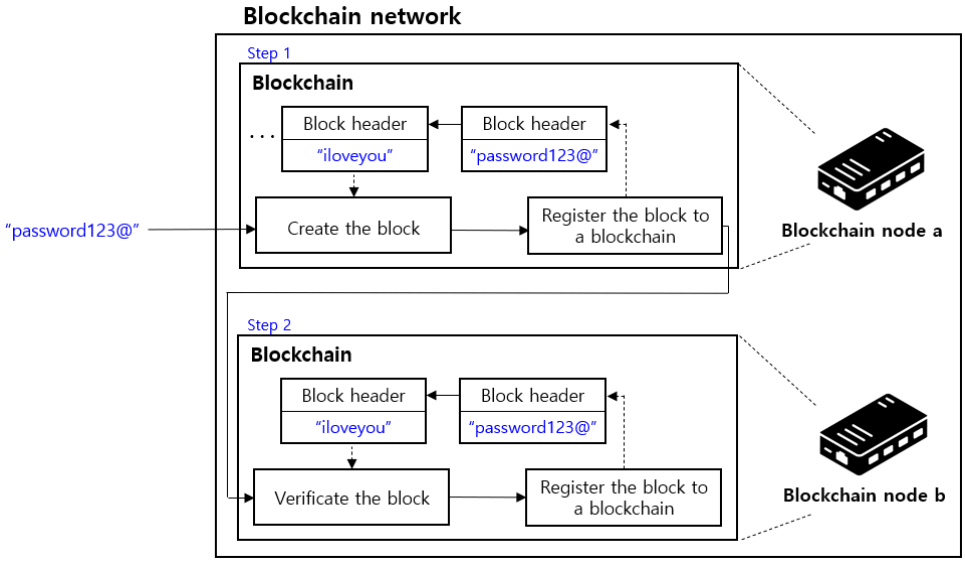

Fig. 4. Process of creating and deploying blocks in a blockchain network.

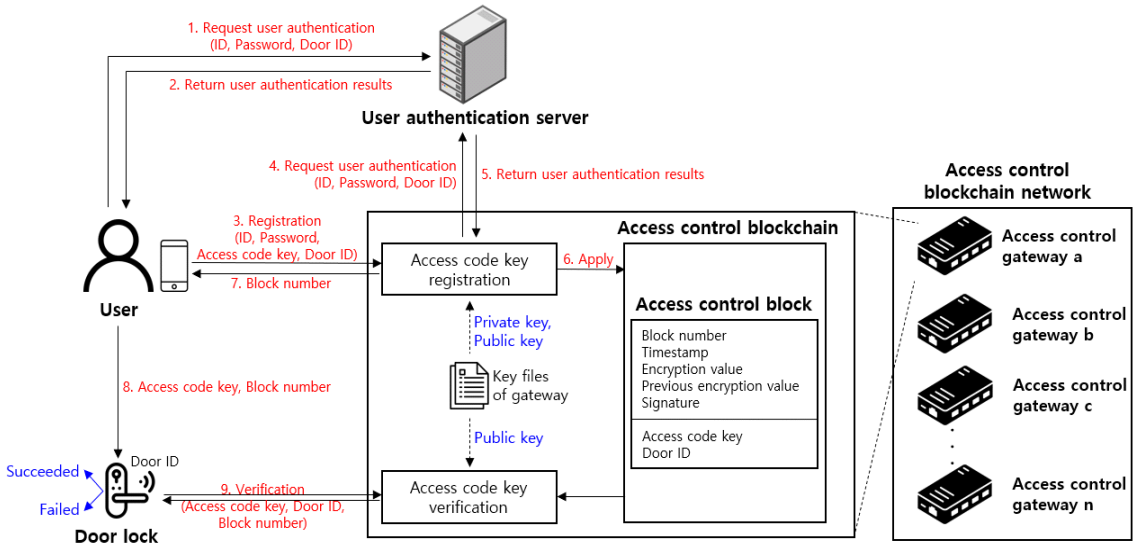

Fig. 5. Verification service model using an access code key.

Fig. 5와 같이 사용자 인증 서버, 출입통제 게이트웨이, 그리고 사용자로 구성한다. 사용자 인증 서버를 두어 인가된 사용자를 등록하고 출입통제 게이트웨이에게 필요한 사용자 인증 서비스를 제공한다. 사용자는 서버에 아이디(ID)와 비밀번호(Password)를 등록하고 게이트웨이에 출입코드키를 제시하여 출입 권한을 등록한다. 출입코드키는 추후 사용자를 인가하는 데에 사용한다. 마지막으로 출입통제 게이트웨이는 사용자의 아이디와 비밀번호를 서버에 제시하여 신뢰할 수 있는 출입증이라면 출입코드키를 반환하고, 블록체인에 저장한다. 그 후, 사용자가 제시한 출입코드키는 블록체인에 저장된 출입코드키와 비교하여 사용자를 인증한다.

이를 위해, 사용자의 출입코드키를 블록체인 네트워크에 등록하는 단계와 사용자의 신원을 등록한 출입코드키로 검증하는 단계로 나눈다. 블록체인 네트워크에 참여하는 노드를 출입통제 게이트웨이로 정의하고, 도어락과 연계하여 사용자의 신원 검증 결과에 따라 출입을 제어한다. 이는 Fig. 5와 같이 제안한다.

Fig. 5의 사용자는 사용자 인증 서버에 신원 등록을 요청한다. 서버는 회원가입등을 통해 신뢰할 수 있는 사용자를 확인한다. 사용자는 도어락을 제어하고 있는 출입통제 게이트웨이 a에 아이디, 비밀번호, 출입코드키, 그리고 도어락 ID를 제공한다. 게이트웨이에서는 사용자의 인증을 위해 출입통제 서버로 신원 검증을 요청하며, 신뢰할 수 있는 사용자인 경우 출입코드키의 무결성을 보장하기 위해 개인키로 출입코드키를 암호화하고 공개키로 서명하여 출입통제 블록을 생성한다. 생성한 블록은 출입통제 블록체인에 등록하고 배포된다. 사용자는 배포된 출입통제 블록의 번호를 반환받으며, 이는 사용자의 출입코드키 저장된 블록에 접근하는 데에 사용한다. 그 후, 사용자가 소지하고 있는 출입코드키로 도어락을 열어야 하는 경우에, 게이트웨이는 출입코드키를 공개키로 블록체인을 복호화하여 검증을 수행한다.

Fig. 5의 사용자 인증 서버에서는 사용자의 인증을 수행한다. 이때 저장되어 있는 사용자의 비밀번호는 해시화되어 공격자가 서버의 정보를 탈취하여도 원본 비밀번호를 획득할 수 없다. Fig. 5의 출입통제 블록체인 네트워크는 출입코드키의 등록과 인증을 수행한다. 블록체인에 등록하는 출입코드키 또한 해시화하여 공격자는 원본의 출입코드키를 획득할 수 없다. 그러나, 공격자는 출입코드키를 임의의 출입코드키로 위변조할 수 있으므로, 무결성을 유지하기 위해 출입코드키를 개인키로 암호화하고, 공개키로 전자서명하여 공격자의 위변조를 방지한다.

특히, 등록한 출입코드키는 공격자의 위변조로부터 보호하기 위해 블록체인 네트워크에서 주기적으로 동기화 과정을 수행한다. 뿐만아니라, 모든 게이트웨이는 동일한 출입코드키를 가지므로, Fig. 5에서

사용자의 출입코드키를 등록한 게이트웨이 a에 장애가 발생하더라도, 도어락은 게이트웨이 b, 게이트웨이 c, 그리고 게이트웨이 n에게 검증을 요청할 수 있다. 사용자의 출입코드키는 사용자 인증 서버가 아닌 도어락을 실제로 통제하는 게이트웨이에 분산 저장되며, 이를 통해 출입코드키의 무결성과 가용성을 보호한다. 공격자가 출입통제 서비스에 접근하기 위해서는 출입통제 서버에 저장된 사용자의 아이디, 비밀번호가 필요하며, 출입통제 게이트웨이로 위장한 공격자가 출입코드키의 등록을 시도하더라도 신뢰할 수 있는 출입통제 게이트웨이들은 이를 복호화할 공개키를 소유하지 않으므로, 신뢰할 수 없는 블록으로 거부한다.

이때, 출입통제 게이트웨이가 출입코드키를 블록체인에 등록하고 배포하거나, 동기화하는 과정에서 블록체인의 정보 외에 다른 인증수단으로 출입통제 블록의 무결성을 유지하고 검증한다면 공격자에게 위변조와 탈취의 여지를 줄 위험이 있다. 따라서, 게이트웨이가 배포한 블록은 게이트웨이가 소유한 블록체인을 통하여 무결성을 검증할 수 있어야 하며, 이를 고려한 출입통제 블록의 구조를 정의하여야 한다.

3.2 출입통제 블록 구조

공격자는 출입통제 게이트웨이로 위장하여 무결성이 검증되지 않은 출입코드키의 등록을 시도하거나, 이미 등록되어 있는 출입코드키의 위변조를 시도할 수 있으므로, 이를 감지하고 거부할 수 있는 출입통제 블록을 설계한다. 이는 Table 1과 같다.

Table 1의 출입통제 블록은 블록 헤더 정보(Block header data)로 블록체인의 무결성을 검증하며, 트랜잭션 정보(Transaction data)로 사용자의 출입 권한을 검증한다. 그중에서 트랜잭션 정보는 사용자가 등록을 요청한 출입코드키와 도어락 ID이며, 공격자의 탈취로부터 보호하기 위해 SHA256 해시 알고리즘으로 해시화한다. 블록체인의 각 블록은 암호값(Encryption value)과 이전 블록의 암호값(Previous encryption value)을 논리적으로 연결한다. 서명(Signature)은 출입통제 블록을 생성하고 배포한 게이트웨이의 공개키이며, 출입통제 블록을 생성한 게이트웨이를 특정하고 블록을 복호화하는 데에 이용한다. 블록 번호는 생성한 출입통제 블록의 번호로, 순차적으로 1씩 증가하며, 타임스탬프는 블록을 생성한 시각이다.

Table 1. Access control block.

Type	Key
Block header data	Block number
	Timestamp
	Encryption value
	Previous encryption value
	Signature
Transaction data	Access code key
	Door ID

3.3 사용자의 출입코드키 등록 과정

출입통제 블록체인 네트워크의 출입통제 게이트웨이가 사용자의 출입코드키와 도어락 ID를 블록체인에 등록하는 과정은 Fig. 6과 같이 동작한다. Fig. 6의 사용자는 서버에 아이디, 비밀번호, 그리고 도어락 ID를 제시하며, 서버는 신뢰할 수 있는 사용자에게 출입 권한 등록 서비스를 제공한다. 그 후, 사용자는 게이트웨이 중 하나를 선택하여 출입코드키의 등록을 요청한다. 이때, 게이트웨이는 블록체인 네트워크에서 동기화하므로, 모두 동일한 출입코드키 등록 서비스를 제공한다. 게이트웨이는 사용자로부터 출입코드키의 등록을 요청받은 후 신뢰할 수 있는 사용자인지 판단하기 위해 서버에게 사용자의 신원 검증을 요청한다. 신뢰할 수 있는 사용자라면 제시한 출입코드키를 이용하여 출입통제 블록을 생성하고 블록체인에 등록하는데, 이는 Fig. 7과 같이 동작한다.

Fig. 7에서 게이트웨이는 사용자의 출입코드키와 도어락 ID를 해시화하여 출입통제 블록의 기밀성을 유지하며, 해시값은 게이트웨이의 개인키(PriK_a)로 암호화한다. 이것이 Table 1의 트랜잭션 데이터이며, 공격자는 개인키와 쌍을 이루는 공개키(PubK_a)를 소유하지 않으므로 출입통제 블록을 복호화할 수 없다. 블록체인의 무결성을 위해 소유한 출입통제 블록체인의 마지막 블록의 해시값과 트랜잭션 정보를 개인키로 암호화하는데, 이것이 블록의 암호값이다. 이때, 출입코드키와 도어락 ID를 이용해 암호값을 생성하였으므로, 둘 중 하나의 값이 위변조된다면, 생성된 암호값과 일치하지 않아 위변조 여부를 알 수 있다. 또한, 암호값을 생성하기 위해 이전 블록의 암호값을 이용하였으므로 이전 블록의 무결성을 보장한

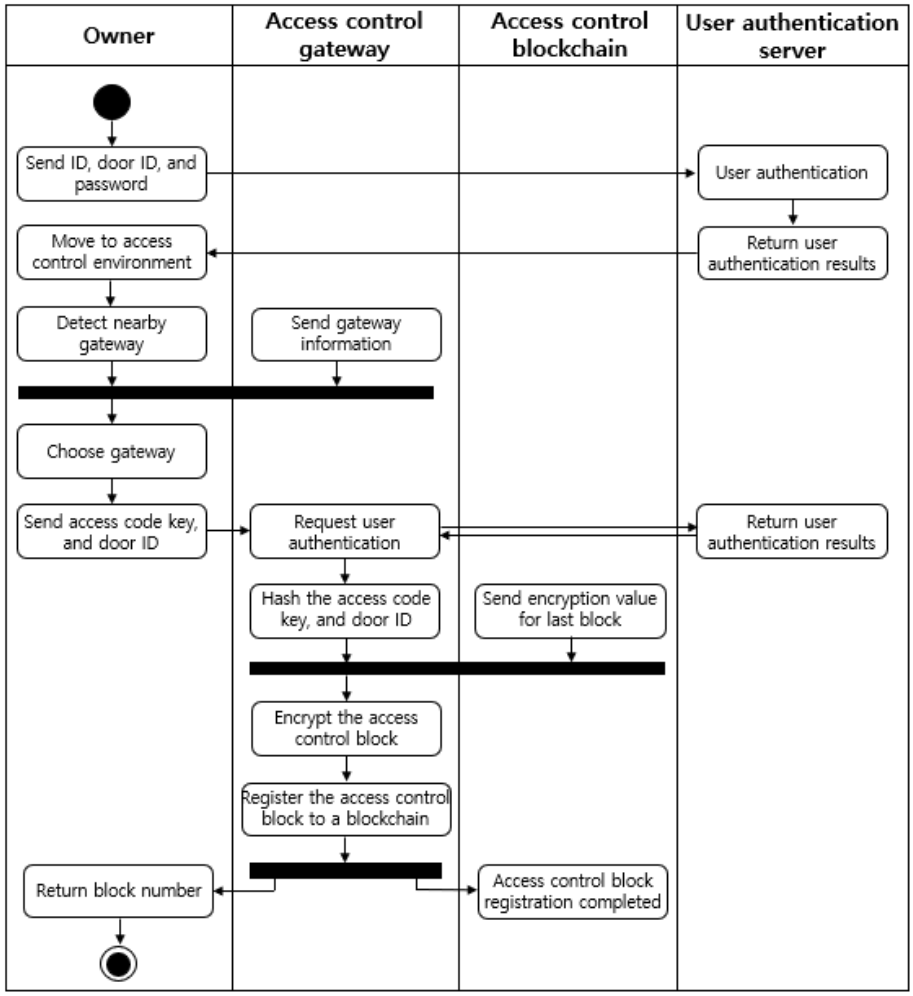

Fig. 6. Registration mechanism for access code key.

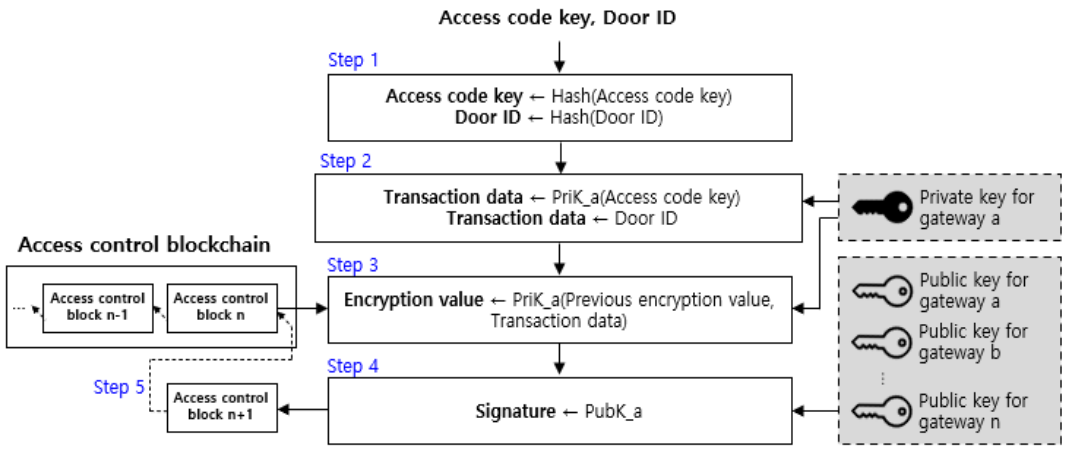

Fig. 7. Encryption flow of access control block.

다. 공격자가 유효한 블록을 생성하고 등록하기 위해서는 게이트웨이의 개인키를 이용하여 암호값을 생성하여야 한다.

게다가 블록체인의 무결성을 위해 블록에 전자서명을 다시 해야 한다. 전자서명이 변경되면 다음 블록의 전자서명의 무결성이 훼손되므로, 이 또한 다시 하여야 한다. 그러나 기존의 모든 노드에서 동일한 합의 알고리즘을 사용한 것과는 달리 각 게이트웨이가 고유하게 소유한 개인키로 암호화하므로, 전자서명을 위해 해당 블록을 생성한 게이트웨이의 개인키가 필요하다. 즉, 위변조한 블록의 다음 블록을 생성한 게이트웨이의 개인키 또한 필요하므로, 블록체인의 위변조는 사실상 불가능하다.

출입통제 블록의 암호값을 계산한 후, 블록 헤더 정보에 암호값, 이전 블록 암호값, 블록 번호, 그리고 타임스탬프를 기록하며, 공개키로 서명한다. Fig. 6의 출입통제 블록 n+1의 이전 블록 암호값은 출입통제 블록 n의 암호값과 일치하며, 사용자가 등록을 요청한 출입코드키와 도어락 ID는 블록체인 네트워크에서 새로운 블록으로 연결된다. 또한, 공격자는 블록을 탈취하더라도 공개키의 원본이 없다면 블록의 복호화가 불가능하므로 출입코드키와 도어락 ID의 정보를 획득할 수 없다. 결과적으로 Fig. 6의 게이트웨이는 사용자의 출입코드키가 등록된 블록체인의 블록 번호를 반환하며, 사용자는 출입코드키와 블록 번호를 이용하여 출입권한을 획득한다.

3.4 사용자의 출입 권한 검증 과정

사용자가 출입코드키를 이용하여 도어락에 접근할 경우에 도어락은 게이트웨이를 통해 사용자의 접근 권한을 검증한다. 이는 Fig. 8과 같이 동작한다. Fig. 8의 사용자가 도어락으로 출입코드키와 블록 번호를 전송하여 도어락에 대한 권한 검증을 요청한다면, 도어락은 사용자의 출입코드키, 자신의 도어락 ID, 그리고 블록 번호를 게이트웨이로 전송한다. 모든 게이트웨이는 동일한 출입통제 블록체인을 소유하므로, 동작 중인 게이트웨이 중 하나를 선택하여 검증시스템의 가용성을 보장한다. 사용자의 신원 검증을 요청받은 게이트웨이는 출입코드키와 도어락 ID를 검증하는 데에 출입통제 블록체인을 이용한다. 이는 Fig. 9와 같이 동작한다.

출입통제 블록체인에 등록되어있는 출입코드키와 도어락 ID는 해시화 된 후 암호화되어 있으므로, 원본의 출입코드키를 유추할 수 없다. 따라서, Fig. 9와 같이 도어락으로부터 검증을 요청받은 출입코드키와 도어락 ID의 해시값끼리 비교한다. 이때, 비교하는 출입코드키는 도어락으로부터 요청받은 블록 번호의 블록이다. Fig. 9에서 출입통제 블록체인의 각각의 블록은 블록을 생성한 게이트웨이의 개인키로 암호화되어 있으며, 공개키로 서명되어 있다. 따라서, 공개키 목록에서 서명과 일치하는 공개키를 찾아 블록을 복호화한다. 복호화한 결과는 이전 블록의 암호값과 트랜잭션 정보이며, 이전 블록의 암호값을 이용하여 출입통제 블록의 무결성을 검증한다. 특히, 트랜잭션 정보는 암호화되어 있는 출입코드키와 도어락 ID이므로, 공개키로 복호화할 수 있다.

그 후, 복호화한 출입코드키, 도어락 ID를 검증이 요청된 출입코드키, 도어락 ID와 비교하여 일치 여부를 확인한다. 만약 일치한다면 출입통제 블록체인에 등록된 신뢰할 수 있는 출입코드키이다. 만약, 일치하지 않은 경우에는 접근 권한을 거부한다.

결과적으로, 접근 권한 검증 결과는 Fig. 8의 도어락에게 반환되며 도어락은 접근 권한이 검증된 사용자의 제어를 받게된다. 이와 같이, 제안한 출입코드키 검증서비스 모델의 효용성을 검증하기 위해 모델을 구현한 후, 출입코드키의 위변조 복구율과 블록체인 네트워크의 성능을 확인하고자 한다. 실험 결과는 기존의 출입코드키 검증서비스 모델의 성능과 비교하고 평가한다.

4. 실험 및 평가

4.1 실험환경

출입통제 검증서비스 모델의 효용성 평가실험은 블록체인의 무결성 검증을 위한 출입코드키의 위변조 복구율 실험과 블록체인의 가용성 평가를 위한 블록체인 네트워크의 성능 확인 실험으로 나누어 진행하였다. Fig. 10에서처럼 실험을 위해 실험환경을 구축하였다. Fig. 10의 (a)는 도어락 프로토타입이며, Raspberry Pi 4 model B의 제어를 받는다. 본 실험에서 사용한 Raspberry Pi 4 model B의 사양은 다음 Table 2와 같다.

도어락 프로토타입은 Fig. 10의 (b)인 출입통제 게이트웨이 프로토타입 디바이스로 출입코드키의 등

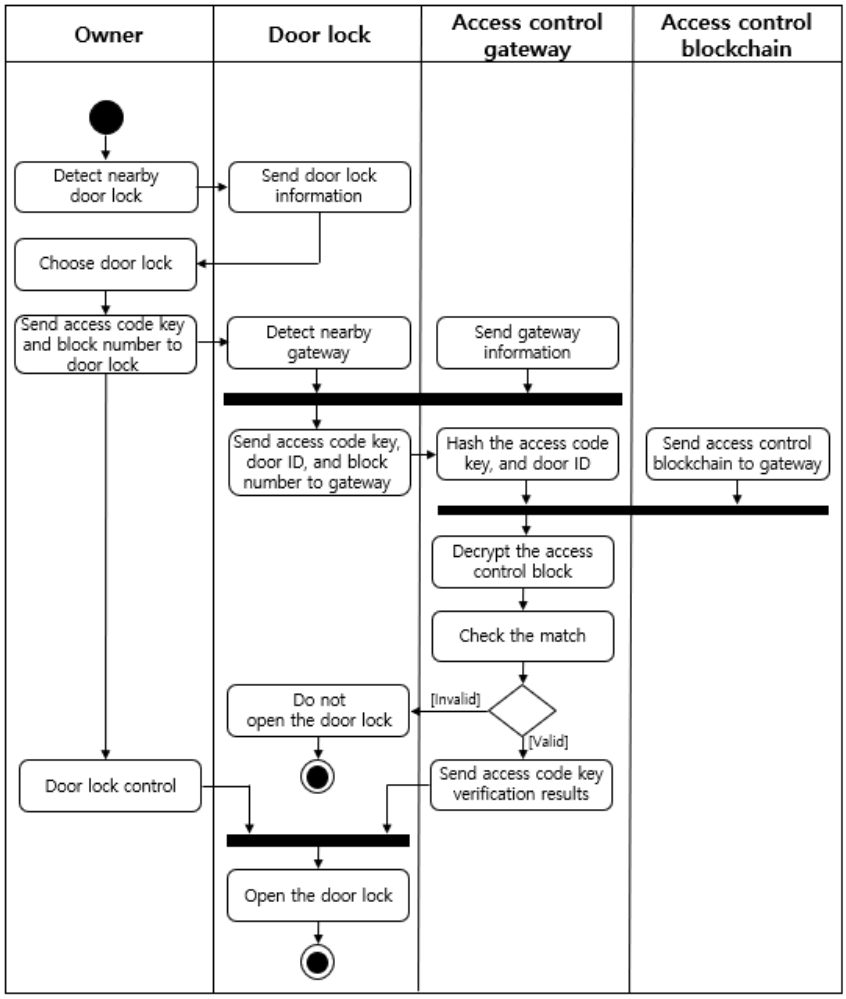

Fig. 8. Verification mechanism for access code key.

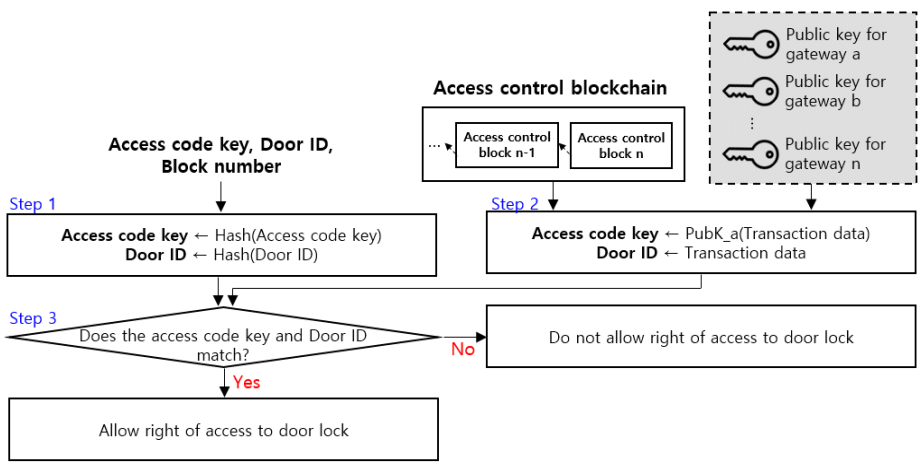

Fig. 9. Decryption flow of access control block.

Table 2. Specifications of access control gateway.

Item	Specification
Model	Raspberry Pi 4 Model B
CPU	quad-core 64-bit 1.5GHz ARM Cortex-A72
RAM	2 GB
SD	32 GB
GPU	Broadcom VideoCore IV
OS	Linux raspberrypi 5.10.63-v7
Network	10/100/1000 Mbps
WiFi	802.11b/g/n/ac Dual-Band

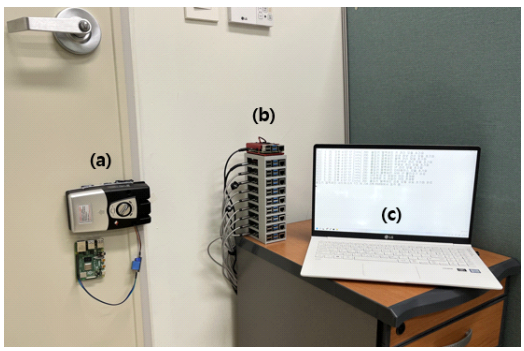

Fig. 10. Experiment environment for access control. (a) door lock, (b) prototype device access control gateway, and (c) log files on the notebook.

록과 검증을 요청한다. 도어락 프로토타입과 출입통제 게이트웨이 프로토타입에 사용한 운영체제는 5.10.63 커널 버전과 Node.js 16.13.1을 사용하였으며, 출입통제 게이트웨이의 구조는 다음 Fig. 11과 같다.

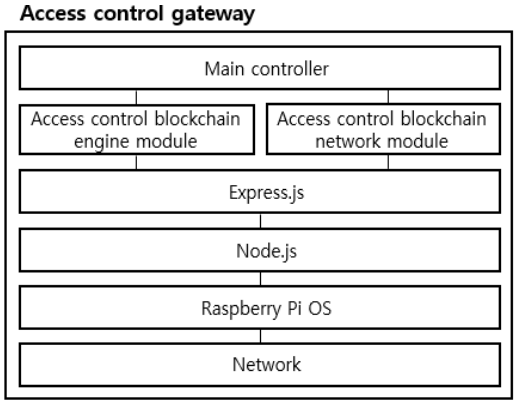

Fig. 11. Structure of access control gateway.

Fig. 11의 출입통제 게이트웨이 프로토타입 디바이스는 Express.js 4.17.3에서 블록체인 엔진 모듈과 블록체인 네트워크 동기화를 위한 통신 모듈로 구분하여 구현하였다[27].

블록체인 엔진은 본 논문에서 제안한 자체 구성한 블록체인을 구현하였다. 출입통제 블록체인 엔진 모듈은 3장에서 제시한 합의 알고리즘과 전자서명을 기반으로 구현하였다. 또한, 게이트웨이가 소유한 블록체인의 일시적 불일치를 해결해야 한다. 이는 프랙티컬 비잔틴 장애 허용(PBFT, Practical Byzantine Fault Tolerance) 합의 매커니즘을 이용한다. 공격자 게이트웨이는 출입코드키를 배포할 수 있는 키 쌍을 보유하지 않으므로 출입코드키의 등록은 불가능하지만, 출입통제 게이트웨이로 위장하여 신뢰할 수 없는 서비스를 제공할 수 있다. 제안한 모델에서는 출입코드키의 검증을 위해 참여 중인 모든 게이트웨이에 검증 요청을 배포하며, 다수결의 원칙으로 검증 결과를 반환한다. 따라서, 공격자 게이트웨이가 네트워크의 절반 이상을 선점하지 않는다면 신뢰할 수 있는 검증 요청 결과를 제공할 수 있다. 블록체인 네트워크에 참여하는 게이트웨이는 10개로 하였다. Fig. 10의 (c)는 출입통제 블록체인 네트워크의 실제 개발과 실험 결과 로그를 확인하기 위한 노트북(15Z980-GA50K)으로, Windows 10 운영체제에서 Putty 0.76을 이용하여 SSH(secure Shell) 접속하였다.

4.2 출입통제 검증서비스 모델의 무결성 평가실험 및 성능분석

블록체인의 무결성은 Fig. 2의 (b)와 같이 공격자가 출입통제키의 위변조를 시도할 경우에 위변조를 감지하고 위변조된 출입코드키를 복구하는 능력을 의미한다. 제안하는 모델에서는 특정 게이트웨이의 출입코드키가 위변조되었을 때, 이를 감지하고 블록체인 네트워크에 참여 중인 다른 게이트웨이에서 무결성이 검증된 출입코드키를 요청하여 복구를 시도한다. 따라서, 블록체인의 무결성을 평가하기 위해 출입코드키 위변조 복구율 실험을 진행하였다.

제안한 검증서비스 모델의 출입코드키 위변조 복구율을 측정하기 위해 정상적으로 등록된 출입코드키를 블록체인 네트워크에 분산 저장해 놓았을 때, 출입코드키를 의도적으로 위변조하고 정상적으로 복구하는지를 확인하였다. 출입코드키의 위변조는

Table 3. Confusion matrix for access control key forgery recovery rate.

Number of experiment	TP	FP	FN	TN	Accuracy (%)
1	100	0	0	100	100
2	100	0	0	100	100
3	100	0	0	100	100
4	100	0	0	100	100
5	100	0	0	100	100
6	100	0	0	100	100
7	100	0	0	100	100
8	100	0	0	100	100
9	100	0	0	100	100
10	100	0	0	100	100

Fig. 10의 (c)에서 SSH 접속한 게이트웨이의 출입코드키의 일부를 수정하였다. 블록체인 네트워크의 위변조 확인은 10,000ms 주기로 수행하였으며, 위변조된 출입코드키 100회, 위변조되지 않은 출입코드키 100회에 대한 위변조 확인과 복구 결과를 혼동 행렬(confusion matrix)로 기록하였다. 혼동 행렬은 식 (1)과 같이 시행 수에 대하여 정답을 맞춘 시행 수로 정확도를 구한다. 예를 들어, 출입코드키 검증서비스 모델은 출입코드키가 위변조되었다고 판단될 경우 출입코드키의 복구를 시도하며, 위변조되지 않은 출입코드키일 경우 복구를 시도하지 않는다. 따라서 위변조 복구율은 위변조된 출입코드키의 복구를 시도하는 경우와 위변조되지 않은 출입코드키의 복구를 시도하지 않은 경우를 정답(TP, TN)으로 하며, 위변조된 출입코드키의 복구를 시도하지 않거나 위변조되지 않은 출입코드키의 복구를 시도하면 오답(FP, FN)으로 한다.

Table 3을 보면 10회의 실험에서 100회의 위변조된 출입코드키의 경우와 100회의 위변조되지 않은 출입코드키의 경우에 대해 전부 올바른 판단을 하였으며, 결과적으로 출입코드키 위변조 복구율은 100%임을 확인하였다. 따라서, 제안한 모델은 공격자가 특정 게이트웨이의 출입코드키를 위변조할 경우에 블록체인의 무결성을 효과적으로 검증할 수 있을 것이다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

4.3 출입통제 검증서비스 모델의 가용성 평가실험 및 성능분석

블록체인의 가용성은 Fig. 2의 (c)와 같이 공격자가 서비스 거부 공격과 같이 게이트웨이를 마비시켰을 때, 다른 게이트웨이에서 출입통제 서비스를 제공할 수 있는 능력을 의미한다. 제안하는 모델에서는 출입코드키의 등록과 출입코드키의 검증을 10개의 게이트웨이에서 분산 처리하였으며, 동일한 서비스를 제공하는지 확인하였다.

블록체인 네트워크의 성능을 확인하기 위해 출입코드키의 등록 시간과 검증 시간을 측정하였다. 그중에서 출입코드키의 등록 시간은 출입통제 게이트웨이가 사용자로부터 출입코드키의 등록요청 트랜잭션을 수신한 후, 블록을 생성하고 블록체인에 배포하는 시간이다. 출입통제 블록을 생성하는 데에 있어서 이전 블록의 암호값을 이용하므로, 동시에 발생한 등록요청 트랜잭션은 순차적으로 처리되어야 한다. 따라서 동시에 발생한 등록요청 트랜잭션의 수를 다르게 하여 등록 시간을 Table 4와 같이 기록하였다. 트랜잭션 수는 국내 아파트 단지 수를 고려하여 최대 10,000개의 출입코드키를 등록하였다[28]. 출입코드키의 등록요청 트랜잭션은 Fig. 10의 (c)에서 10,000개의 쓰레드로 구현하였으며, 각각의 쓰레드에서 요청 결과를 반환받을 때까지의 소요 시간을 기록하였다.

Table 4를 보면 블록체인에 출입코드키 1개를 등록하는데 17 ms가 소요되었으며, 10개를 등록하는데 25 ms가 소요된 것을 볼 수 있었다. 출입통제 게이트웨이는 블록체인을 이용하여 동일한 출입코드키 등록 서비스를 제공하므로 동시에 발생한 10개의 등록

Table 4. Registration time based on the number of registration request transactions.

Number of transaction	1	10	100	500	1,000	2,000	3,000
Registration time (ms)	17	25	114	318	574	1,519	2,812
Number of transaction	4,000	5,000	6,000	7,000	8,000	9,000	10,000
Registration time (ms)	3,778	5,816	8,075	11,493	16,126	20,941	25,801

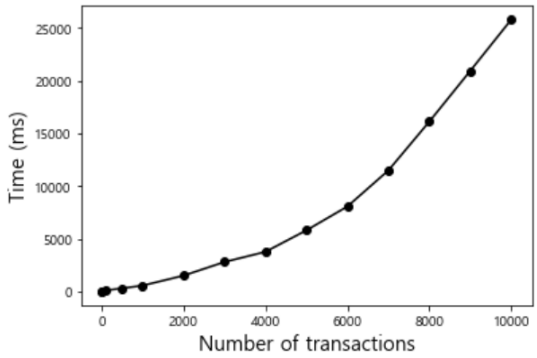

Fig. 12. Registration time by number of transactions.

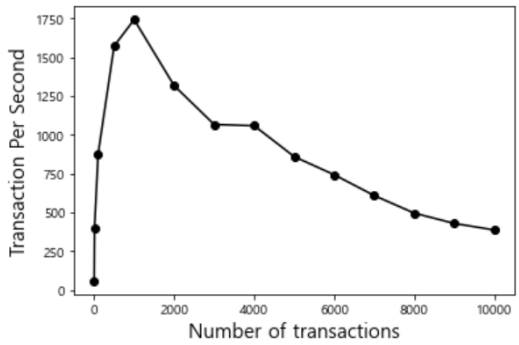

Fig. 13. Transaction per second by number of transactions.

요청 트랜잭션은 10개의 게이트웨이에서 분산 처리된 것으로 보인다. 결과적으로 10,000개의 출입코드키의 등록을 동시에 요청할 경우, 25,801 ms가 소요되었음을 알 수 있었다.

Fig. 12는 출입코드키 등록 요청의 수에 따른 등록 시간이며 등록 요청의 수가 증가할수록 등록 시간의 증가 폭이 커지는 것을 볼 수 있었다. 이는 블록체인의 무결성을 유지하기 위해 이전 블록의 암호값을 이용하므로 출입코드키의 등록을 위해 대기 시간이 발생한 것으로 추측하였다. 이를 고려하여 출입코드키 등록 요청에 대한 TPS(Transaction Per Second)를 계산하였다. TPS는 식 (2)와 같이 트랜잭션의 수를 처리 완료 시간으로 나눈 값으로, 1초 동안 처리할 수 있는 평균 트랜잭션의 수를 의미한다. 이는 Fig. 13과 같이 나타난다. Fig. 13을 보면 트랜잭션의 수가 증가할수록 TPS는 감소하는 경향을 보이며, 최악의 경우인 10,000개의 등록 트랜잭션이 동시에 발생한 경우에, 387.58 TPS를 확인하였다.

$$TPS = \frac{Number\ of\ transactions}{Time(s)} \quad (2)$$

출입코드키의 검증 시간은 출입통제 게이트웨이가 사용자로부터 출입코드키의 검증 요청 트랜잭션을 수신한 후, 블록체인을 이용해 출입코드키의 무결성을 검증하는 시간이다. 출입코드키의 검증을 위해 블록체인의 첫 번째 블록부터 순차적으로 접근해야 하므로, 검증 시간은 블록체인에 등록된 출입코드키의 수와 관련이 있다. 1개의 블록에는 1개의 출입코드키가 저장되어 있으며, 블록체인 네트워크의 블록의 수는 검증을 위해 탐색해야 하는 블록의 개수와 비례한다. 따라서, 블록의 수를 출입코드키 등록 시간 측정 실험의 최대 트랜잭션 수인 10,000개로 정의하였다. 출입코드키의 검증요청 트랜잭션은 Fig. 10의 (c)에서 쓰레드로 구현하였으며, 각각의 쓰레드에서 검증요청 결과를 반환받는 소요 시간을 기록하였다. 또한, 동시에 발생한 검증 요청 트랜잭션의 수는 1명이 출입통제 서비스를 시도하는 1회부터, 출입통제 시스템 사용자의 10%가 서비스를 동시에 시도하는 1,000회까지 측정하였다. 이는 Table 5와 같이 기록하였다.

Table 5를 보면 출입통제 게이트웨이에 사용자의 출입코드키의 검증을 1회 요청할 경우, 127 ms가 소요되었으며, 10회에는 141 ms를 확인할 수 있었다. 출입통제 게이트웨이는 블록체인을 이용하여 동일한 출입코드키 검증 서비스를 제공하므로 동시에 발생한 10개의 검증 요청 트랜잭션은 10개의 게이트웨이에서 분산 처리된 것으로 보인다. 결과적으로 1,000개의 출입코드키의 검증을 동시에 요청할 경우에 7,317 ms가 소요되었음을 알 수 있었다.

Table 5. Verification time based on the number of verification request transactions when the number of blocks is 10,000.

Number of transaction	1	5	10	50	100	200	300
Verification time (ms)	127	141	153	274	522	1,166	1,492
Number of transaction	400	500	600	700	800	900	1,000
Verification time (ms)	2,079	2,596	3,279	3,995	4,818	5,905	7,317

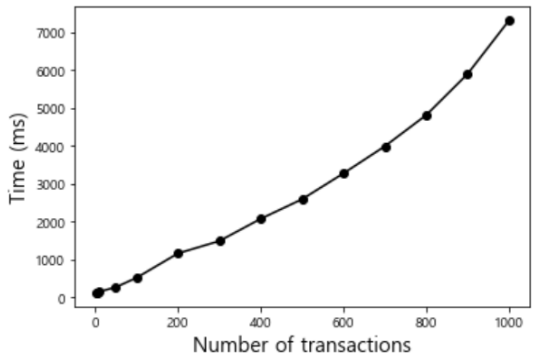

Fig. 14. Verification time by number of transactions and number of blocks.

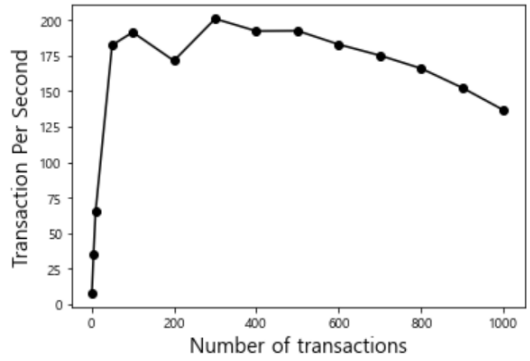

Fig. 15. Transaction per second by number of transactions and number of blocks.

Fig. 14는 검증 요청의 수에 따른 검증 시간이며 검증 요청의 수가 증가할수록 검증 시간이 증가하는 경향을 보였다. 그러나, 검증 요청 시간의 증가는 등록 요청 시간의 증가 폭보다 작은 것을 볼 수 있다. 출입코드키의 등록의 경우, 이전 블록의 정보를 이용하여 새로운 블록을 등록하므로 순차적으로 등록 요청이 처리되어야 하지만, 검증 요청의 경우에는 모든 트랜잭션이 비동기로 처리된 것으로 추측하였다. 이를 고려하여 검증 요청에 대한 TPS를 계산하였으며 Fig. 15와 같다. 결과적으로 10,000개의 출입코드키가 등록되어 있고 1,000회의 출입코드키에 대한 검증 요청 트랜잭션이 동시에 발생한 경우에. 136.66 TPS를 확인하였다. 또한, 10개의 게이트웨이에서 출입코드키를 등록하고 검증하는 데에 동일한 서비스를 제공하며, 공격자가 특정 게이트웨이를 무력화하여도 출입통제 서비스는 가용성을 보장할 수 있음을 확인하였다.

Table 6은 본 논문에서 자체 구성한 블록체인의 성능을 확인하기 위해 기존의 블록체인 기반의 서비스에 관련한 연구에서 실험한 블록체인 등록 시간과 검증 시간 비교이다. 예를 들어, N. Vatcharatiansakul의 연구에서는 출입통제 게이트웨이와 같은 IoT 디바이스에서의 블록체인 성능을 확인하기 위해 이더리움 기반의 네트워크를 구성하고, 등록과 검증 시간을 측정하였다[29]. 결과적으로 출입통제 게이트웨이에서 이더리움 네트워크를 이용할 경우, 출입코드키 등록에 0.21 TPS, 출입코드키 검증에 9.49 TPS로 동시 처리율이 매우 떨어지는 것을 알 수 있었다.

또한, Mihui Kim의 연구에서는 IoT 디바이스에서 블록체인 네트워크를 구성하기 위해 DPoS 합의 알고리즘을 이용하였으며, 기존의 이더리움 네트워크에 비해 등록과 검증 시간을 크게 개선하였다[30]. Ali Dorri의 연구에서는 스마트 홈에서의 IoT 디바이스 관리를 위해 PoW 합의 알고리즘을 사용한 블록체인 네트워크를 제안하였다[31]. IoT 디바이스에서 수집한 데이터를 고성능의 PC로 전송하며, PC는 블록을 생성하고 블록체인에 배포하였다. 그러나, IoT 디바이스에서 PC로, PC에서 블록체인 네트워크로 전파하는 불필요한 과정이 추가되어 등록에 31.25 TPS, 트랜잭션 검증에 52.63 TPS를 확인하였다. 그에 비해 제안한 모델은 미리 배포된 공개키를 이용하여 출입코드키의 등록과 검증에 이용하므로, 사용자의 출입코드키를 효과적으로 관리할 수 있음을 확인할 수 있었다.

결과적으로 본 논문에서 제안한 출입통제용 출입코드키 검증서비스 모델은 출입코드키 위변조 복구

Table 6. Compared to previous blockchain model.

Paper	Registration(TPS)	Verification(TPS)	Device
N. Vatcharatiansakul [29]	0.21	9.49	Raspberry Pi
Mihui Kim [30]	3.32	3.87	Raspberry Pi
Ali Dorri [31]	31.25	52.63	PC
Proposed model	387.58	136.66	Raspberry Pi

정확도 100%로 공격자의 위변조나 탈취로부터 출입코드키를 안전하게 보호하며, 출입코드키 등록요청 트랜잭션에 대해 387.58 TPS, 출입코드키 검증 요청 트랜잭션에 대해 136.66 TPS로 기존의 방식에 비해 효율적인 서비스를 제공할 수 있는 것을 확인하였다.

5. 결 론

출입통제 시스템은 사용자를 출입코드키로 인증하여 출입 권한에 따라 선별적으로 출입을 허가한다. 그러나 출입코드키가 공격자에 의해 탈취된다면 사용자 인증의 역할을 수행할 수 없으므로, 출입코드키는 안전하게 보호되어야 한다. 기존의 출입코드키 검증서비스 모델은 출입코드키의 전송 과정에만 초점이 맞춰져 있어, 저장된 출입코드키의 위변조나 서비스 거부 공격에 취약하다는 한계가 있다.

따라서 본 연구에서는 블록체인 기술을 이용하여 출입코드키의 탈취뿐만이 아니라 위변조와 시스템 거부 공격에도 출입코드키를 안전하게 보호할 수 있는 출입통제 시스템을 제안하였다. 제안한 시스템은 출입통제 게이트웨이를 블록체인 노드로 하는 블록체인 네트워크를 구성하며, 사용자가 출입코드키의 등록을 요청할 경우 출입통제 블록으로 암호화하여 블록체인에 등록한다. 그 후, 사용자가 출입코드키의 검증을 요청한다면 게이트웨이에서 주기적으로 동기화되고 있는 블록체인을 이용하여 사용자의 출입코드키를 검증하고, 결과를 도어락으로 반환한다.

제안한 블록체인 모델이 기존 서비스의 한계점을 해결하는지 확인하기 위해 출입코드키 무결성 평가실험과 출입코드키 가용성 평가실험을 하였다. 출입코드키 무결성을 평가하기 위해 출입코드키를 위변조하며, 복구하는 위변조 복구율의 경우 100%를 확인하였다. 또한, 가용성을 평가하기 위해 출입코드키 등록과 검증을 10개의 게이트웨이에서 분산 처리하였으며, 387.58 TPS의 등록 요청 처리 능력, 136.66 TPS의 검증 요청 처리 능력, 그리고 출입통제 서비스가 분산처리되어 동작하는 것을 확인하였다. 또한, 기존의 블록체인 네트워크를 이용한 서비스의 등록과 검증 시간보다 우수한 성능이다. 따라서, 제안한 블록체인 기반의 출입통제 검증모델을 이용한다면 기존의 중앙 서버 기반의 출입통제 검증서비스 모델의 한계점을 개선하여 좀 더 안전하고 효율적인 검증서비스를 제공할 수 있을 것으로 기대한다.

REFERENCE

[1] S. Anwar and D. Kishore, “IOT Based Smart Home Security System with Alert and Door Access Control Using Smart Phone,” *International Journal of Engineering Research & Technology (IJERT)*, Vol. 5, No. 12, pp. 504-509, 2016.

[2] N.A. Hussein and I.A. Mansoori, “Smart Door System for Home Security Using Raspberry pi3,” *International Conference on Computer and Applications (ICCA)*, pp. 395-399, 2017.

[3] K. Artem, V. Teslyuk, I. Tsmots, and T. Myroslav, “Implementation of the Face Recognition Module for the “Smart” Home Using Remote Server,” *Conference on Computer Science and Information Technologies*, pp. 17-27, 2018.

[4] D. Pavithra and R. Balakrishnan, “IoT Based Monitoring and Control System for Home Automation,” G*lobal Conference on Communication Technologies (GCCT)*, pp. 169-173, 2015.

[5] C.Y. Xu, X. Zheng, and X.M. Xiong, “The Design and Implementation of a Low Cost and High Security Smart Home System Based on Wi-Fi and SSL Technologies,” *Journal of Physics: Conference Series*, Vol. 806, No. 1, 2017.

[6] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A Survey on the Security of Blockchain Systems,” *Future Generation Computer Systems*, Vol. 107, pp. 841-853, 2020.

[7] S.A. Salami, J. Baek, K. Salah, and E. Damiani, “Lightweight Encryption for Smart Home,” *11th International Conference on Availability, Reliability and Security (ARES)*, pp. 382-388, 2016.

[8] K.N. Mallikarjunan, K. Muthupriya, and S.M. Shalinie, “A Survey of Distributed Denial of Service Attack,” *10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1-6, 2016.

[9] Y.C. Huang, "Secure Access Control Scheme of RFID System Application," *Fifth International Conference on Information Assurance and Security*, pp. 525-528, 2009.

[10] Q. Feng, D. He, S. Zeadally, M.K. Khan, and N. Kumar, "A Survey on Privacy Protection in Blockchain System," *Journal of Network and Computer Applications*, Vol. 126, pp. 45-58, 2019.

[11] W. Liang, Y. Fan, K.C. Li, D. Zhang, and J.L. Gaudiot, "Secure Data Storage and Recovery in Industrial Blockchain Network Environments," *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 10, pp. 6543-6552, 2020.

[12] D. Han, H. Kim, and J. Jang, "Blockchain Based Smart Door Lock System," *International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1165-1167, 2017.

[13] Z. Meng, T. Morizumi, S. Miyata, and H. Kinoshita, "Design Scheme of Copyright Management System Based on Digital Watermarking and Blockchain," *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 359-364, 2018.

[14] S.K. Kwon, H.S. Han, and J.H. Eom, "A Study on the Operation Scheme of Layered Access Control for Effective Visitors's Access Control," *Journal of Security Engineering*, Vol. 9, No. 3, pp. 231-240, 2012.

[15] S. Pawar, V. Kithani, S. Ahuja, and S. Sahu, "Smart Home Security Using IoT and Face Recognition," *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1-6, 2018.

[16] B. Bhushan, G. Sahoo, and A.K. Rai, "Man-in-the-Middle Attack in Wireless and Computer Networking－A Review," *3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)*, pp. 1-6, 2017.

[17] J. Liang, I.J. Jang, and H.S. Yoo, "A Secure Token-Updated Authentication Scheme Using Security Key," *The Journal of Society for e-Business Studies*, Vol. 12, No. 1, pp. 89-97, 2007.

[18] Y. Feng, W. Wang, Y. Weng, and H. Zhang, "A Replay-Attack Resistant Authentication Scheme for the Internet of Things," *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 541-547, 2017.

[19] P. Sirohi, A. Agarwal, and S. Tyagi, "A Comprehensive Study on Security Attacks on SSL/TLS Protocol," *2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 893-898, 2016.

[20] M. Nawir, A. Amir, N. Yaakob, and O.B. Lynn, "Internet of Things (IoT): Taxonomy of Security Attacks," *3rd International Conference on Electronic Design (ICED)*, pp. 321-326, 2016.

[21] D. Kim and K. Seo, "PGP Certification System in Blockchain Environments," *Journal of Korea Multimedia Society*, Vol. 23, No. 5, pp. 658-666, 2020.

[22] S.H. Jung, J.H. Kim, and C.B. Sim, "Implementation of University Point Distributed System based on Public Blockchain," *Journal of Korea Multimedia Society*, Vol. 24, No. 2, pp. 255-266, 2021.

[23] K.H. Lee and Y.H. Jung, "Blockchain-Based Access Control and Material Management System," *Journal of the Korea Academia-Industrial cooperation Society*, Vol. 22, No. 11, pp. 442-448, 2021.

[24] A. Gervais, G.O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Com-*

munications Security, pp. 3-16, 2016.

[25] L.M. Bach, B. Mihaljevic, and M. Zagar, "Comparative Analysis of Blockchain Consensus Algorithms," *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1545-1550, 2018.

[26] H. Kim, "A Study on the Blockchain Based Knowledge Sharing Platform," *The Journal of Society for e-Business Studies*, Vol. 27, No. 1, pp. 95-109, 2022.

[27] Blockchain-based access control access code key verification service model code (2022), https://github.com/KiHyeon-Hong/Access_control_blockchain_network_paper.git (accessed May 31, 2022).

[28] Status of mandatory management fee disclosure complex (2022), http://www.k-apt.go.kr/ (accessed May 29, 2022).

[29] N. Vatcharatiansakul and P. Tuwanut, "A Performance Evaluation for Internet of Things Based on Blockchain Technology," *5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pp. 1-4, 2019.

[30] M. Kim and Y. Kim, "Implementing Blockchain Based Secure IoT Device Management System," *Journal of IKEEE*, Vol. 23, No. 4, pp. 1343-1352, 2019.

[31] A. Dorri, S.S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home," *IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 618-623, 2017.

홍 기 현

2021년 2월 가천대학교 IT융합대학 컴퓨터공학과 학사
2021년 3월~현재 가천대학교 대학원 IT융합공학과 컴퓨터공학 전공

관심분야 : 사물인터넷(IoT), AIoT, 블록체인 네트워크, 시스템 보안

이 병 문

1988년 2월 동국대학교 전자계산학과 학사
1990년 2월 서강대학교 전자계산학과 석사
2008년 2월 인천대학교 컴퓨터공학과 박사
1990년~1997년 ㈜LG전자 중앙연구소 네트워크 연구실 선임연구원
1998년 3월~현재 가천대학교 IT융합대학 컴퓨터공학과 교수
관심분야 : 사물인터넷(IoT), AIoT, 블록체인 네트워크, 시스템 보안