스마트계약 기반 회의용 키 분배 기법

윤성현*
백석대학교 ICT학부 교수

The Smart Contract based Conference Key Distribution Scheme

Sunghyun Yun*
Professor, Division of ICT, Baekseok University

요 약 최근 코로나 19 확산으로 비대면 화상회의에 대한 수요가 급증하고 있다. 기존의 줌, 구글 미트와 같은 서버-클라이언트 기반의 화상회의 시스템에서 서버는 회의키를 이용하여 회의 멤버 및 회의 내용을 제어한다. 만약 서버를 신뢰할 수 없다면 서버에 의한 조작 및 부정이 가능하기 때문에 회의 멤버들의 프라이버시는 보장되지 않는다. 따라서, 서버의 도움 없이 모든 참여자가 신뢰할 수 있는 회의키 생성 및 분배 기법의 개발이 필수적이다. 스마트계약은 블록체인에 저장되는 프로그램으로 블록체인의 특성상 수정될 수 없고 누구나 그 실행 결과를 검증할 수 있다. 본 논문에서는 스마트계약 기반의 회의용 키 분배 기법을 제안한다. 제안한 방법은 키 분배를 위한 스마트계약 발행, 회의키 생성 및 검증 단계로 구성된다. 스마트계약은 기존의 신뢰할 수 있는 서버의 역할을 대체하며, 회의 멤버들은 스마트계약에 구현된 프로토콜에 따라서 회의키를 생성한다. 제안한 방법은 화상회의 시스템에 적용할 수 있으며, 회의 멤버 이외의 다른 사용자들은 회의키에 접근할 수 없다.

주제어 : 블록체인, 스마트계약, 키분배, 화상회의, 회의키

Abstract Recently, epidemic of covid-19 causes rapid increase in demand for untact video conferences. In existing server-client based video conference systems such as Zoom, Google Meet, etc., the server generates the conference key and controls the access rights of meeting members and their contents with it. In this case, the server can fabricate or repudiate the meeting. So, the privacy of the meeting members is not guaranteed. It's necessary to make the conference key distribution scheme where all participants can verify the trustfulness without help of the server. The smart contract is the program stored to the blockchain. Its contents cannot be altered due to the property of the blockchain, and everybody can verify the execution results of it. In this study, we propose the smart contract based conference key distribution scheme. The proposed scheme is consisted of smart contract deployment, conference key generation and verification stages. The smart contract replaces the role of existing trustful server and the meeting members can generate the conference key according to the protocols implemented on it. The proposed scheme can be applied to the video conference systems and only the meeting members can access the conference key.

Key Words : Blockchain, Smart Contract, Key Distribution, Video Conference, Conference Key

이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2018R1D1A3B07050180)
*교신저자 : 윤성현(shcrpt@gmail.com)
접수일 2020년 8월 25일 수정일 2020년 10월 4일 심사완료일 2020년 10월 10일

1. 서론

최근 Covid-19 확산으로 인터넷 기반의 언택트(untact) 화상회의 시스템에 대한 수요가 급증하고 있다[1]. 기업의 업무 회의와 같이 민감한 내용을 다루는 회의는 회의 멤버 및 회의 내용에 대한 인증이 중요하다. 줌, 구글 미트 등 현재 많이 사용되고 있는 인터넷 기반 화상회의 시스템은 대부분 서버-클라이언트 기반으로 서비스되고 있으며 서버 중심으로 회의 인증 및 보안 서비스가 이루어진다[2-4].

기존의 화상회의 시스템에서 서버는 회의키를 생성하여 이 키를 알고 있는 회의 멤버만 화상회의에 참여할 수 있도록 한다. 줌과 같은 시스템의 경우에 회의 주최자는 회의키를 회의 멤버들에게 별도의 채널로 전달해 준다[4]. 이 경우 서버가 회의키를 알고 있기 때문에 서버에 의한 부정이 가능하고 회의 멤버들의 프라이버시가 보장되지 않는다. 회의 멤버들은 서버를 전적으로 신뢰해야 한다.

케이블 TV 서비스를 위한 CAS (Conditional Access System)의 경우, 채널 디코딩에 사용되는 마스터 키는 스마트카드에 저장되어 우편 또는 인편으로 가입자에게 전달된다 [5]. 불특정 다수가 접근하는 인터넷과 비교하여 프라이빗 채널을 이용한다는 장점은 있지만, 서비스 제공업자가 세션키를 관리하기 때문에 서버에 의한 부정이 가능하고 화상회의와 같은 양방향 시스템에는 직접 적용할 수 없다.

기존의 인터넷 기반의 키 공유 및 분배 시스템은 공개키 암호 또는 Diffie-Hellman 프로토콜을 주로 사용한다[6-8]. 이 기법들은 점대점 (point-to-point) 보안 연결을 위한 용도로 만들어 졌고, 화상회의와 같이 다자간 키를 공유해야 하는 시스템에 직접 적용하는 것은 적합하지 않다. I. Ingemarsson이 제안한 회의키 분배 기법은 Diffie-Hellman 기법을 확장하여 모든 참여자가 회의키를 공유할 수 있도록 하였지만 중간자 공격(man in the middle attack)에 취약하다[9].

블록체인에 저장된 데이터는 그 사실을 부인할 수 없고 또한 조작할 수 없는 특성을 갖는다. 비트코인, 이더리움으로 대표되는 퍼블릭 블록체인 네트워크는 여기에 참여하는 모든 피어노드가 똑 같은 블록체인 사본을 공유하고 있다. 따라서, 어느 한 노드가 자신의 블록체인에 저장된 데이터를 조작할 경우에 50% 이상의 다른 노드들도 똑같이 조작된 블록체인을 공유해야 하는데, 불특정 다수가 참여하는 블록체인 네트워크에서 현실적으로 불가능하다[10]. 더불어, 해쉬함수의 비가역성 때문에 조작된 블록의 해쉬값에 맞게 기존 블록체인에 저장된 블록의 해쉬값들을 모두 재계산하는 것은 계산상 불가능하다[10].

블록체인 네트워크는 그 사용 목적에 따라서 각기 다른 프로토콜로 운영된다. 이더리움 블록체인은 그 중의 하나로, 모는 노드들은 블록체인에 저장된 프로그램을 EVM에서 실행하여 그 상태를 변화시킬 수 있다[11, 12]. 블록체인은 모든 노드가 공유하는 기억장소이고 EVM은 이를 실행할 수 있는 컴퓨터 역할을 한다. 프로그램 실행 결과는 링크드 리스트로 생성되는 블록에 시간 순으로 저장되고 이전 실행 결과 및 프로그램은 수정 및 삭제할 수 없다[11,12].

스마트계약은 이더리움 블록체인에 저장되는 프로그램으로 다양한 목적으로 블록체인을 이용할 수 있도록 한다. 스마트계약은 그 특성 상 모두에게 공개 되고, 수정할 수 없으며, 입력과 출력이 프로그램에 정해진 순서대로 수행된다[13]. 모든 피어 노드는 이 동작을 확인하고 검증할 수 있다.

본 논문에서는 이더리움 스마트계약에 기반을 둔 회의용 키 생성 및 검증 기법을 제안한다. 2 장에서 기존의 키 분배 기법에 대해서 살펴보고, 3장에서 제안한 회의키 생성 및 검증 프로토콜을 기술한다. 4장에서 제안한 방법의 안전성을 분석하고 5장에서 결론 및 기대효과를 제시한다.

2. 관련연구

줌, 구글 미트와 같은 기존의 화상회의 시스템에서 회의키는 서버 또는 회의 주최자가 생성하여 회의 멤버들에게 분배한다. 회의키를 알고 있는 사용자들만 화상회의에 참여할 수 있다. 단점은 모든 회의 멤버들이 서버 또는 회의 주최자를 전적으로 신뢰해야 한다는 것이다. 더불어 서버의 키 생성 모듈은 서버에서만 관리하기 때문에, 서버에 의한 조작이 가능하다.

인터넷 기반 점대점 키 분배는 키 생성자가 공개키 암호를 이용하여 키를 분배하는 방법과 당사자가 모두 참여하여 키를 생성하는 방법으로 구분된다.

일반적으로 공개키 암호를 이용한 키 분배에서, 키 생성자는 세션키를 상대방의 공개키로 암호화하고, 이를 자신의 개인키로 서명하여 상대방에게 전송한다. 수신자는 키 생성자의 공개키로 서명을 검증하고, 자신의 개인키로 암호화된 키를 복원한다.

세션키를 함께 생성하여 공유하는 방법으로는 디피-

헬먼 키 공유 기법이 대표적이다. 상대방의 공개키와 자신의 개인키를 이용하여 함께 키를 생성하는 방법이다. 세션키 생성을 위해서 당사자 모두가 참여해야 하기 때문에 키 생성자에 의한 부정을 최소화 할 수 있다.

여러 명의 사용자가 참여하는 인터넷 화상회의에서 회의 멤버들은 똑 같은 회의키를 공유해야 한다. 기존의 공개키 암호 기반의 키 분배 기법을 직접 적용할 경우, 회의 주최자에 의해서 회의키가 생성되고 이 키를 분배할 회의 멤버도 마찬가지로 회의 주최자에 의해서 결정된다. 이 경우, 특정 멤버가 실제 회의에 참여하지 않았지만, 회의 주최자가 이 멤버가 회의에 참여한 것으로 간주할 수 있고 이를 검증할 수 없다. 결국, 모든 회의 멤버들은 회의 주최자를 전적으로 신뢰해야 하는 부담이 따른다.

디피-헬먼 기법과 같이 참여자가 공동으로 키를 생성하는 방법은 회의키가 회의 멤버 모두에게 종속되기 때문에 상기한 회의 주최자에 의한 부정을 최소화 할 수 있다. 하지만 이 방법은 점대점 보안 연결에 적용되는 방법으로 3 명 이상의 사용자가 참여하는 화상회의 시스템을 위한 키 분배 방법에 직접 적용하는 것은 적합하지 않다.

따라서, 회의키 공유를 위해서 서버 의존도를 최소화하고 모든 멤버가 회의키 생성에 참여하는 새로운 회의용 키 분배 기법의 개발이 필수적이다.

3. 제안한 회의용 키 분배 방법

그림 1은 제안한 회의용 키 분배 모델을 보여준다. 회의 주최자는 회의 멤버를 결정하고 회의키 생성을 시작한다. 모든 멤버들은 순차적으로 회의키 생성 프로토콜에 참여한다. 스마트계약은 각 멤버의 공개키로 암호화된 회의키를 서명한다. 검증 단계에서 모든 멤버들은 스마트계약의 서명을 검증하고 회의키를 공유한다.

3.1 스마트계약 생성 및 등록

가정 1. Owner는 키 생성 및 분배를 할 수 있는 스마트계약 프로그램을 블록체인에 저장한다. n 명의 멤버가 스마트계약 프로그램을 이용하여 회의키를 생성한다고 가정한다. 암호화 및 서명에 RSA 알고리즘을 사용한다 [14].

U_i: 회의 멤버, $i = [1..n]$

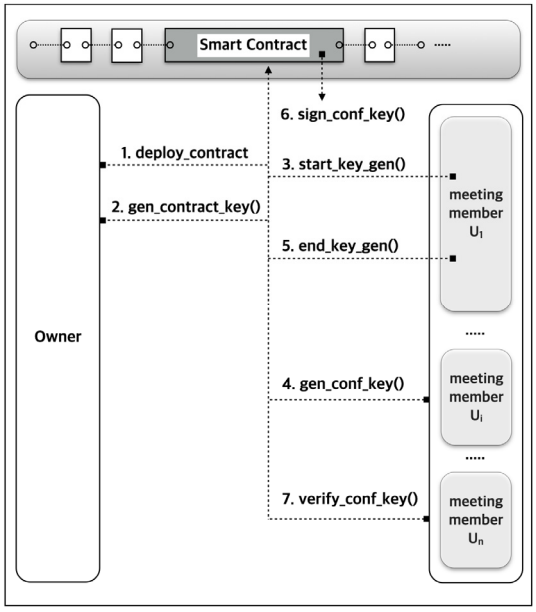
U_1: 회의 주최자

$E_{pkey}(m)$: 공개키 $pkey$로 m을 암호화하는 함수

$D_{skey}(c)$: 개인키 $skey$로 c를 복호화하는 함수

$S_{skey}(m)$: 개인키 $skey$로 m을 서명하는 함수

$V_{pkey}(c)$: 공개키 $pkey$로 서명 c를 검증하는 함수



[Fig. 1] The Smart Contract based Conference Key Distribution Model

가정 2. 스마트계약에서 관리하는 상태 변수 및 테이블은 다음과 같고 블록체인에 저장된다.

UserPubkey는 회의 멤버의 주소와 공개키를 관리하는 구조체이다.

```
struct UserPubkey {
    address addr; // 회의 멤버의 주소
    uint pubKey; // 회의 멤버의 공개키
}
```

ConfKey는 회의키를 저장하기 위한 구조체로 회의 이름, 암호화된 회의키, 스마트계약의 서명 및 공개키로 구성된다.

```
struct ConfKey {
    string confName; // 회의 이름
    UserPubKey users[];
    uint encryptedKey[]; // 암호화된 회의키
    uint signature[]; // 스마트계약의 서명
    uint contractPubKey; // 스마트계약의 공개키
}
```

가정 3. 회의키 생성 및 분배에 사용되는 스마트계약 함수는 다음과 같다.

·gen_contract_key()

Owner에 의해서 실행되고, 스마트계약은 자신의 개인키, 공개키 쌍을 생성한다. 공개키는 블록체인에 저장된다.

·start_key_gen()

회의 주최자 U_1에 의해서 실행되고, 회의 멤버들의 주소와 공개키를 회의키 생성 프로토콜에 참여할 순서대로 스마트계약 테이블에 저장한다.

·gen_conf_key()

모든 회의 멤버는 순차적으로 회의키 생성 프로토콜에 참여한다.

·end_key_gen()

회의 주최자 U_1에 의해서 실행되고, 회의키 생성 프로토콜을 종료한다.

·sign_conf_key()

회의키 생성 프로토콜이 종료되면 스마트계약에 의해서 자동으로 실행된다. 스마트계약은 각 회의 멤버의 암호화된 회의키를 자신의 개인키로 서명하여 테이블에 저장한다.

·verify_conf_key()

회의 멤버에 의해서 실행되고, 스마트계약의 서명을 검증한다. 검증에 성공하면 자신의 개인키로 회의키를 추출한다.

Owner의 스마트계약 등록 절차는 다음과 같다.

단계 1: Owner는 회의키 생성, 검증 및 분배로 구성된 스마트계약 프로그램을 트랜잭션에 저장하고, 이를 블록체인 네트워크에 전송한다.

단계 2: 단계 1의 트랜잭션은 블록체인 네트워크에 있는 마이닝 노드에 의해서 블록에 저장된다.

단계 3: 합의 알고리즘을 통해서 선정된 마이닝 노드의 블록이 블록체인에 등록된다.

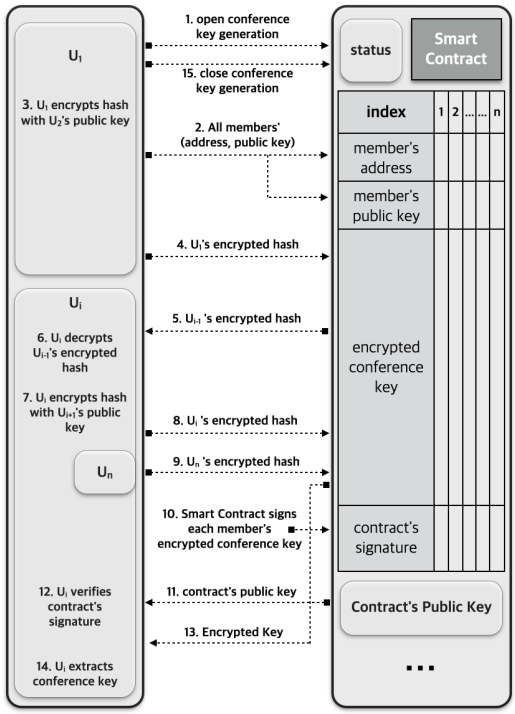
단계 4: Owner는 블록체인에 저장된 단계 1의 스마트계약 주소를 공개한다.

단계 5: 스마트계약은 회의키 서명을 위한 RSA 개인키 및 공개키 쌍을 생성한다.

단계 6: 단계 5에서 생성된 공개키는 스마트계약 테이블에 저장된다.

3.2 회의키 생성

그림 2는 제안한 회의키 생성 단계를 보여준다.



[Fig. 2] Proposed Conference Key Generation & Verification Stage

단계 1: U_1은 스마트계약의 회의키 생성 프로토콜을 시작하고, 스마트계약은 관련 상태 변수 및 테이블을 초기화 한다. 다른 멤버의 접근을 제한하기 위해서 스마트계약의 상태를 회의키 생성 중인 상태로 변경한다.

단계 2: U_1은 모든 멤버의 주소와 공개키를 스마트계약의 테이블에 저장한다. 모든 회의 멤버는 테이블에 저장된 순서대로 회의 키 생성 프로토콜에 순차적으로 참여한다.

단계 3: U_1은 난수 값 $rand_1$을 생성하고 이 값에 대한 해쉬값을 다음과 같이 생성한다 [15].

$$hash_1 = temp_1 = SHA256(rand_1)$$

단계 4: U_1은 U_2의 테이블에 저장된 공개키를 가져온다. U_1은 U_2의 공개키로 $hash_1$을 암호화하고 이 값을 서명 한다. U_1은 암호화된 해쉬값과 서명을 U_2의 테이블에 저장한다.

$(E_{U_{2-PKEY}}(hash_1),\ S_{U_{1-SKEY}}(E_{U_{2-PKEY}}(hash_1)))$

단계 5: U_i는 자신의 테이블에 저장된 암호화된 해쉬값과 서명을 가져온다.

$(E_{U_{i-PKEY}}(hash_{i-1})$
$S_{U_{i-1-SKEY}}(E_{U_{i-PKEY}}(hash_{i-1})))$

단계 6: U_i는 스마트계약 테이블에 저장된 U_{i-1}의 공개키를 가져와서 다음과 같이 단계 5의 서명을 검증한다.

$E_{U_{i-PKEY}}(hash_{i-1}) ==$
$V_{U_{i-1-PKEY}}(S_{U_{i-1-SKEY}}(E_{U_{i-PKEY}}(hash_{i-1})))$

서명 검증에 성공하면, 자신의 개인키로 단계 5의 암호화된 해쉬값으로부터 U_{i-1}의 해쉬값을 추출한다.

$hash_{i-1} = D_{U_{i-SKEY}}(E_{U_{i-PKEY}}(hash_{i-1}))$

단계 7: U_i는 난수 값 $rand_i$를 생성하고 이 값에 대한 해쉬값 $temp_i$를 생성한다. 단계 6에서 추출한 해쉬값과 $temp_i$를 결합하고, 이 값에 대한 해쉬값을 다음과 같이 생성한다.

$temp_i = SHA256(rand_i),\ i = [2..n]$

$hash_i = SHA256(temp_i \ || \ hash_{i-1})$

U_i는 해쉬값을 U_{i+1}의 공개키로 암호화 하고, 이 값에 대한 서명을 다음과 같이 생성한다.

$(E_{U_{i+1-PKEY}}(hash_i),$
$S_{U_{i-SKEY}}(E_{U_{i+1-PKEY}}(hash_i)))$

단계 8: U_i는 단계 7의 암호화된 해쉬값과 서명을 스마트계약 DB의 U_{i+1} 테이블에 저장한다.

단계 9: U_i가 마지막 참여자이면 U_n은 모든 멤버의 해쉬값이 포함된 회의키를 각 사용자의 공개키로 암호화하여 스마트계약 DB의 각 사용자 테이블에 저장한다. 그렇지 않으면 단계 5, 6, 7, 8을 반복한다.

$(E_{U_{1-PKEY}}(hash_n),\ \cdots,\ E_{U_{n-PKEY}}(hash_n))$

단계 10: 스마트계약은 각 회의 멤버의 테이블에 저장된 암호화된 회의키를 서명하고, 이 값을 해당 테이블에 저장한다.

$(S_{C-SKEY}(E_{U_{1-PKEY}}(hash_n)),\ \cdots,$
$S_{C-SKEY}(E_{U_{n-PKEY}}(hash_n)))$

3.3 회의키 검증 및 분배

단계 11: 각 회의 멤버는 스마트계약의 공개키를 가져온다.

단계 12: 각 회의 멤버는 스마트계약의 공개키로 다음과 같이 서명을 검증한다.

$E_{U_{i-PKEY}}(hash_n)\ ==$
$V_{C-PKEY}(S_{C-SKEY}(E_{U_{i-PKEY}}(hash_n)))$

단계 13: 각 회의 멤버는 자신의 테이블에 저장된 암호화된 회의키를 가져온다.

단계 14. 각 회의 멤버는 자신의 개인키로 다음과 같이 회의키를 추출한다.

$conf_{key} = D_{U_{i-SKEY}}(E_{U_{i-PKEY}}(hash_n))$

단계 15: U_1은 스마트계약의 회의키 생성 프로토콜을 종료한다. 스마트계약은 테이블을 초기화하고 자신의 상태를 회의키 생성 대기 중인 상태로 변경한다.

4. 안전성 분석

제안한 스마트계약 기반의 회의용 키분배 기법에서 회의 멤버만 키를 공유할 수 있고, 회의 멤버는 키 생성에 함께 참여한 것에 대해서 부인할 수 없음을 증명한다.

정리 4.1 회의 멤버를 제외한 제 3자는 회의키를 조작할 수 없고 회의 내용을 감시할 수 없다.

(증명) 3.2절의 회의키 생성 단계에서, 회의주최자는 회의에 참여할 멤버들의 주소와 공개키를 스마트계약 테이블에 저장한다. 회의 멤버들은 테이블에 저장된 순서대로 회의키 생성에 참여한다. 회의 멤버의 해쉬값은 이전 멤버의 해쉬값과 결합되어 생성되고, 이를 다음 멤버의 공개키로 암호화하여 스마트계약 테이블에 저장한다. 마지막 순서의 멤버는 모든 멤버의 해쉬값이 적용된 회의키를 생성하고 이를 각 멤버의 공개키로 암호화하여 스마트계약 테이블에 저장한다. 각 멤버는 자신의 개인키를 이용하여 회의키를 추출한다.

제 3자가 회의키를 알려면 각 회의 멤버의 개인키를 알아야 하는데, 스마트계약 테이블에 저장된 공개키로부터 개인키를 추출하는 것은 계산상 불가능하다 [7]. 회의키 생성 중에 단계별로 생성되는 암호문도 마찬가지로 해당 멤버의 개인키를 알아야만 복원할 수 있다. Owner

는 스마트계약 등록에 참여하지만, 키 생성 및 검증 프로토콜에 참여할 수 없다. 스마트계약 테이블에 저장된 데이터는 블록체인의 특성상 변조할 수 없기 때문에 Owner를 포함한 제 3자는 회의키 조작 및 접근이 불가능하다. Q.E.D.

정리 4.2 회의 참여자는 회의키 생성에 참여한 사실을 부인할 수 없다.

(증명) 블록체인에 저장되는 데이터는 블록이 저장된 순서, 즉 시간에 종속되며 한 번 저장되면 그 내용을 변조할 수 없다. 회의 멤버들은 스마트계약 테이블에 저장된 순서대로 회의키 생성에 참여하는데, 3.2절에서 단계별로 생성된 값들은 순서대로 블록체인에 저장된다. 따라서, 각 회의 멤버가 생성한 값들은 순서대로 추적 및 검증이 가능하다. 더불어, 각 참여자의 서명이 같이 저장되기 때문에, 회의 멤버는 회의키 생성에 참여한 사실을 부인할 수 없다. Q.E.D.

5. 결론

제안한 스마트계약 기반의 회의용 키 생성 기법은 스마트계약 등록, 회의키 생성 및 검증 단계로 구성된다. 스마트계약 등록 단계에서 회의 멤버 및 회의키 정보를 저장할 수 있는 스마트계약 테이블을 생성한다. 회의키 생성 단계에서 모든 회의 멤버는 순차적으로 회의키 생성에 참여한다. 회의키 검증 단계에서 모든 회의 멤버는 스마트계약의 서명을 검증하고 회의키를 추출한다. 제 3자는 회의키에 접근할 수 없고 회의 멤버들은 회의키 생성에 참여하였음을 부인할 수 없다. 제안한 방법은 줌, 구글 미트 등과 같이 인터넷 기반의 화상회의 시스템에 적용되어 서버에 의한 부정을 최소화할 수 있다.

REFERENCES

[1] https://www.businesswire.com/news/home/20200416005739/en/Impact-of-COVID-19-on-the-Video-Conferencing-Market-2020---ResearchAndMarkets.com

[2] ZOOM Security Guide, https://zoom.us/docs/doc/Zoom-Security-White-Paper.pdf, 2020.

[3] https://meet.google.com

[4] https://zoom.us

[5] Recommendation ITU-R BT.1852-1(10/2016) Conditional-access systems fordigital broadcasting BT Series Broadcasting service(television)

[6] B. Schneier, Applied Cryptography, Wiley, 1996.

[7] R.L.Rivest, A.Shamir, and L.Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, Vol.21, No.2, 1978.

[8] "RFC 4306 Internet Key Exchange (IKEv2) Protocol". Internet Engineeringrg/web/20150107073645/http://www.ietf.org/rfc/rfc4306.txt.

[9] I. Ingemarsson, D.T.Tang and C.K.Wong, "A Conference Key Distribution System," IEEE Transactions on Information Theory, Vol.28, No.5. pp.714-720, 1982.

[10] A.M.Antonopoulos, Mastering Bitcoin 2nd Edition, O'Reilly, 2017.

[11] A. M. Antonopoulos, D. Gavin Wood, Mastering Ethereum, O'Reilly, 2019.

[12] https://ethereum.org/en/

[13] D. Gavin Wood, ETHEREUM: A Secure Decentralised Generalised Transaction Ledger, https://ethereum.github.io/yellowpaper/paper.pdf

[14] https://www.openssl.org/

[15] SHA-2, FIPS PUB 180-4, 2001.

윤 성 현(Yun Sunghyun) [종신회원]

- 1997년 2월 : 고려대학교 일반대학원 컴퓨터학과 (이학박사)
- 1998년 3월 ~ 2002년 2월 : LG전자 선임연구원
- 2002년 3월 ~ 현재 : 백석대학교 ICT학부 교수

〈관심분야〉
블록체인, 사물인터넷, DRM, 정보보호