Check for updates

Journal of KIIT. Vol. 18, No. 1, pp. 135-141, Jan. 31, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2020.18.1.135

블록체인에서 개인정보보호를 위한 멀티유저 암호 키 적용 방법 연구

강희복*[1], 장행천*[2], 장창수**

A Study on the Application Method of Multi-User Encryption Keys for Personal Information Protection in Blockchain

Hee-Bog Kang*[1], Haeng-Cheon Jang*[2], and Chang-Soo Jang**

요 약

블록체인에서 블록의 원본 여부는 디지털서명에 의해 증명될 수 있는 반면, 블록에 포함된 트랜잭션 내용은 누구나 볼 수 있도록 공개되어 있다. 본 논문은 블록체인을 쇼핑몰에 적용하였을 때, 공개된 트랜잭션에 개인정보가 포함되더라도 거래 당사자 외에는 개인정보를 복호화 할 수 없도록 하는 멀티 유저 암호 키 방법을 제안한다. AES 암호화 알고리즘에 사용된 대칭키는 구매자ID, 판매자ID, 서버 ID로 구성된 멀티 유저 암호 키를 사용한다. 이 키 방법을 적용하면 트랜잭션 마다 다른 대칭키가 사용되기 때문에 암호화된 개인정보는 거래 당사자가 아니면 복호화 할 수 없기 때문에 공개된 트랜잭션에서도 개인정보는 보호된다. 또한 하나의 대칭키만 사용하여 모든 암호문에 대한 복호할 수 있던 기존 방식과는 달리 개인정보가 이전보다 더 안전하게 관리될 수 있도록 매 건별로 암호문에 대응하는 대칭키를 사용해야 한다.

Abstract

In the blockchain, the original status of the block can be proved by digital signature, but the contents of the transactions contained in the block are visible to anyone. This paper proposes a multi-user encryption key method that, when the blockchain is applied to a shopping mall, even if the personal information is included in an open transaction, the private information cannot be decrypted except by the trading party. The symmetric key used in the AES encryption algorithm uses a multi-user encryption key consisting of a buyer ID, a seller ID, and a server ID. When this key method is used, since different symmetric keys are used for each transaction, the encrypted personal information cannot be decrypted unless it is the transaction party. In addition, unlike conventional methods that can decrypt all ciphertexts using only one symmetric key, it is necessary to use a symmetric key corresponding to the ciphertext on a case-by-case basis so that personal information can be managed more securely than before.

Keywords
blockchain, encryption, signature, multi-user encryption key, WebRPC

* 전남대학교 컴퓨터공학과 박사과정
- ORCID[1]: https://orcid.org/0000-0001-8098-6006
- ORCID[2]: https://orcid.org/0000-0002-8249-8919
** 전남대학교 컴퓨터공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0003-3517-3019

· Received: Dec. 04, 2019, Revised: Jan. 08, 2020, Accepted: Jan. 11, 2020
· Corresponding Author: Chang-Soo Jang
Chonnam University 2th gong-hakgwan 4 flor, 50 Daehak-ro, Yeosu-si, Jeollanam-do, 59626, Korea.
Tel.: +82-61-659-7251, Email: csjang@jnu.ac.kr

Ⅰ. 서 론

블록체인은 데이터를 거래할 때 중앙집중형 서버에 기록을 보관하는 기존 방식과 달리 거래 참가자 모두에게 내용을 공개하는 분산디지털 장부를 말한다[1]. 블록체인의 장점인 분산장부와 저비용 P2P 네트워크[2]를 활용하면 암호 화폐 뿐만 아니라 공개마켓 플레이스를 구축하는데도 활용할 수 있다.

그러나 블록체인은 데이터를 모든 노드와 공유하므로 개인정보가 노출되는 문제점[3]과 데이터 불변성 유지를 위해 개인정보를 삭제할 수 없는 문제점[3]을 갖고 있다.

쇼핑몰은 개인정보보호지침[4]에 따라 개인정보는 별도 서버에 관리하고 있다. 이 쇼핑몰에 블록체인을 적용하게 되면 암호 화폐에서는 지갑주소를 개인정보에 대신하여 사용할 수 있었던 것과 달리 배송정보에 개인정보를 포함시켜야 하며 개인정보가 포함된 배송 트랜잭션은 각 노드에 전파되어 모두에게 공개되어 개인정보도 함께 노출된다.

본 논문은 트랜잭션에 포함된 개인정보를 대칭키로 암호화하고 거래 당사자 외에는 대칭키를 알 수 없도록 하여 제 3자는 배송 트랜잭션에 노출된 개인정보를 열람할 수 없게 하는 방법을 제안한다.

거래 당사자를 인식하는 방법은 IUWT 토큰을 이용하여 로그인 및 자동 로그인한 구매자 ID[5]와 상품마다 등록되어 있는 여러 판매자 ID중에서 해당 배송 트랜잭션을 이행할 판매자를 결합하고 주문거래를 발생시킨 서버ID를 이용하는 것이다. 대칭키 암호화 알고리즘은 암호화 및 복호화 처리속도를 비교하여 AES-128-CBC, AES-256-CBC를 적용하였다.

논문의 제 2장에서는 블록체인의 디지털서명과 암호화에 관하여 고찰하였고 제 3 장은 개인정보를 유추할 수 없게 하는 방법과 블록을 WebRPC 방식의 P2P 네트워크를 사용하여 POST로 전달하는 방법을 기술하였다. 또한 데이터 내부에 개인정보가 포함될 경우 멀티 유저 키로 암호화하여 당사자 이외에는 개인정보를 복호화하지 못하도록 하는 방법을 제안하였다. 제 4장은 개인정보의 암호화 및 복호화 처리 속도를 측정하였고 제 5장 결론에서는 멀티 유저 암호 키 방식이 적용된 개인정보가 블록의 트랜잭션에서 노출되더라도 당사자 이외의 제 3자가 쉽게 개인정보를 복호화 할 수 없기 때문에 SSL 통신보안 상태가 아닌 P2P 네트워크에서 데이터 보안에 적합함을 확인하였다.

Ⅱ. 블록체인의 디지털서명과 암호화

2.1 트랜잭션 디지털 서명

분산처리를 통해 장부가 동시에 사라질 위험은 줄었지만 기록을 위.변조하려는 시도는 암호학의 도움을 받아서 해결하고 있다[1]. 디지털 서명의 대표적인 기술은 해쉬 함수와 비대칭 키 암호화가 있다. 디지털 서명은 개인키로 암호화하고 공개키로 복호화 하여 데이터 변질 여부를 확인할 수 있는 기술이기 때문에 진본을 유지할 수 있다[1][6].

블록체인은 개인키 및 공개키와 지갑 주소의 길이를 줄이기 위해 공개키 암호기술의 한 종류인 ECC(Elliptic Curve Cryptography: 타원곡선암호) 방식에 속하는 ECDSA(Elliptic Curve Digital Signature Algorithm) 알고리즘을 사용한다[7].

ECC 암호방식은 RSA 암호방식에 비해 암호 키의 길이가 짧다[8]. 암호 키의 길이가 길면 보안은 강화되지만 연산 속도가 느려진다. ECC 방식은 암호 길이가 짧지만 높은 암호 성능을 보여주기 때문에 블록체인 기술에 사용되고 있다.

타원곡선 방정식의 범위를 좁혀서 ECDSA에서 사용하는 제한된 방정식으로 설명하면 타원곡선 방정식 y^2=x^3+ax+b를 만족하는 (x,y)의 집합을 곡선 그래프로 표시한 것이다[9]. 블록체인에서 사용하는 secp256k1 curve에서는 a=0, b=7을 사용하여 y^2= x^3+7의 방정식을 이용한다[10].

그림 1은 트랜잭션을 해시 한 값을 다시 개인 키로 암호화하여 트랜잭션과 함께 P2P 네트워크로 노드에게 전파하면 이를 수신한 노드는 함께 제공된 서명을 함께 제공된 공개 키로 복호화한 결과와 트랜잭션을 해시 한 결과를 비교하는 과정을 보여주고 있다.

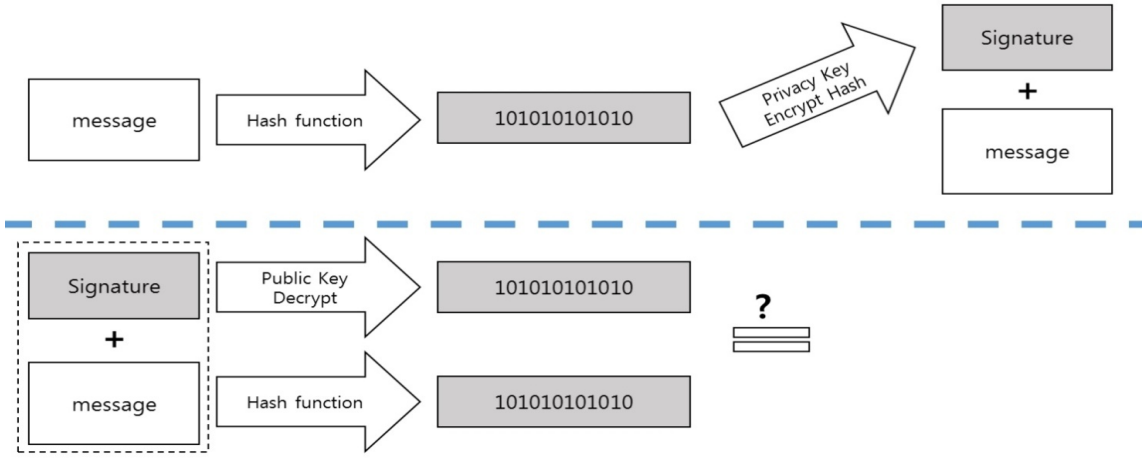


그림 1. 디지털 서명
Fig. 1. Digital signature

$bytes = openssl_random_pseudo_byte(256, $cstring);
$hex = bin2hex($bytes);
$random = $hex . microtime() . fixed String
Private Key = hash('sha256',
hex2bin(hash('sha256,$random)))
Public Key = ECC (Private Key)
Wallet Address = BASE58Check(SHA256(Public Key))

2.2 트랜잭션 메시지에 대한 양방향 암호화

개인정보는 개인정보보호지침에 따라 양방향 암호화 대상인 주민등록번호, 신용카드, 계좌번호, 바이오정보는 AES128, 단방향 암호화 대상인 비밀번호는 AES256 알고리즘을 적용하여 별도 관리한다. 블록체인에서 트랜잭션을 노드에 전파할 때 메시지는 복호화하여 평문으로 전달하기 때문에 제 3자에게 메시지는 공개된다. 비트코인 등 화폐거래는 트랜잭션 메시지에 상대방의 전자지갑 주소와 코인의 가치만 표시되고 개인정보는 없으므로 문제되지 않지만 블록체인을 전자상거래에 적용할 때는 트랜잭션 메시지에 개인정보가 포함될 수 있기 때문에 당사자 이외는 개인정보를 알지 못하도록 양방향 암호화 알고리즘을 적용할 필요가 있다. 트랜잭션의 진위 여부를 검증하는 디지털 서명은 개인 키로 암호화된 해시 값을 공개 키로 복호화하는 반면 트랜잭션 내부에 있는 메시지는 거래 당사자만 개인정보를 열람할 수 있도록 대칭 키를 사용해야 한다.

그러나 단순히 서버 키만 사용하여 암호화에 적용하면 서버 운영자가 모든 트랜잭션의 개인정보를 열람할 수 있게 되므로 구매자 ID, 판매자 ID, 서버 ID를 결합한 멀티 유저 암호 키 방법을 제안한다.

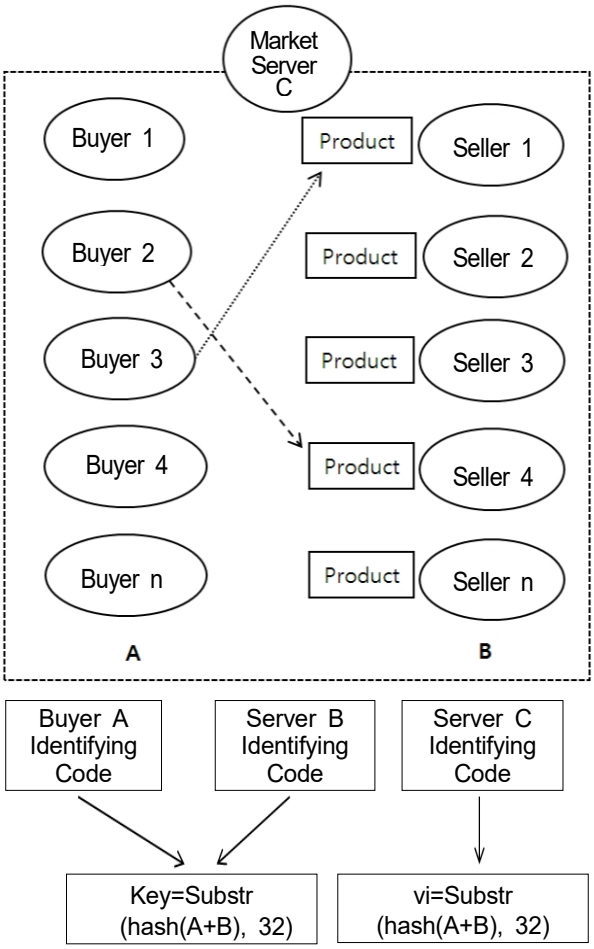


그림 2. 개인정보 암호화를 위한 키 구성
Fig. 2. Key configuration for privacy encryption

그림 2에서 거래 당사자의 ID를 얻고 노드가 보유한 서버 키를 결합하여 새로운 비밀 키를 만든 후 이 키를 이용하여 암호화함으로써 당사자 및 노드 운영자 이외는 거래 정보를 열람할 수 없도록 하고 있다.

이 방법을 사용하면 대응키의 경우에도 별도 암호 키 전달 없이 로그인 정보와 상품내부에 포함된 판매자 정보와 서버 키를 조합하여 해당 트랜잭션 메시지의 당사자 여부가 결정되기 때문에 제 3자가 복호화에 개입할 수 없게 된다.

III. 개인정보 보호를 위한 트랜잭션 구조와 WebRPC를 이용한 전파 방법

3.1 개인정보 보호를 위한 트랜잭션 구조 설계

전자상거래에서 개인정보는 개인정보보호지침에 의거 쇼핑몰 서버에 저장한다. 공개마켓에서 개인정보 열람 당사자는 구매자와 판매자 및 거래를 중개하는 운영자이며 선택적 개인정보 열람 당사자는 배송업체가 있다. 그림 3은 주문에서 배송까지의 트랜잭션 구조를 보여주고 있다. 한 개의 주문트랜잭션에는 여러 개의 상품트랜잭션이 포함되며 상품트랜잭션마다 배송트랜잭션이 포함된다.

본 논문에서는 AES CBC모드 대칭 키 암호화 알고리즘을 사용하며, 개인정보를 저장할 때는 AES-256, P2P 네트워크로 전파할 때는 AES-128을 서로 다르게 적용한다. CBC(Cipher Block Chaining) 모드는 IV(Initialization Vector)를 XOR로 연산하고 이후에는 직전 암호화 결과와 XOR로 연산하므로 매 암호 마다 iv를 바꿔 사용하는 것을 좋다. 그러나 배송정보 테이블에 레코드를 생성할 때 멀티 유저 키 생성 방식을 사용하기 때문에 iv가 동일하더라도 키 값을 유추하기 어려우므로 iv값은 항상 노드마다 다르게 부여한 Node ID를 사용한다.

블록에 포함될 주문트랜잭션에는 상품트랜잭션을 해시 값으로 계산한 후 이들을 머클트리(Merkle Tree)로 만들어 최종 해시 값을 Item Hash Root 값으로 관리한다. 상품트랜잭션들은 주문트랜잭션의 오더 번호(Order number)를 키 값으로 하여 순차번호(Sequence)를 부여하고 상호 연결 키 값으로 사용한다. 상품트랜잭션에는 구매자 ID와 판매자 ID가 포함되어 있고 배송정보용 해시 값을 갖고 있다.

이러한 관계는 독립적인 테이블(Table)에 의해 관리되며 배송정보에 포함된 개인정보는 AES-256으로 암호화되어 있어서 배송정보 테이블 자체로는 복호화용 멀티 유저 키 값을 알 수 없고 상위의 상품트랜잭션으로부터 정상적으로 연결된 때만 개인정보를 복호화 할 수 있게 된다.

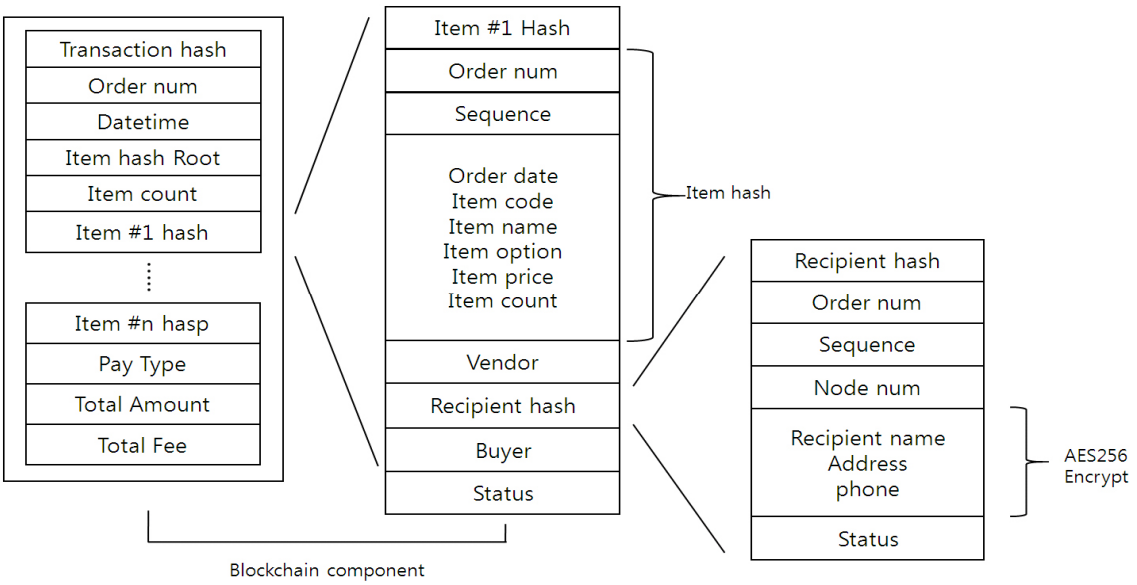


그림 3. 주문거래에 따른 상품 및 배송 거래
Fig. 3. Composition of items and delivery by order transaction

iv_key=server_key
privacy_key=privyA+privy B
key=substr(hash('sha256', privacy_key),0,32)
iv=substr(hash('sha256',iv_key),0,32)
stext=openssl_encrypt(msg,'AES-128-CBC',key,OPENSSL_RAW_DATA,iv);

위의 암호문을 평문으로 복호화 할 때는 다음과 같다.

text=openssl_decrypt(stext,'AES-128-CBC,key,OPENSSL_RAW_DATA,iv);

3.2 WebRPC를 이용한 상품정보 전달 방법

WebRPC(Remote Procedure Call)는 웹 브라우저 간 실시간 영상, 음성, 채팅 등을 구현할 수 있다. WebRPC를 기반으로 P2P 통신을 하면 HTTP 기반 적응형 미디어 스트리밍 시스템의 효율을 높일 수 있는 기술이 연구되었다[11]. WebRPC와 유사한 기술에는 구글이 오픈소스로 개발한 WebRTC(Real-Time Communication)이 있다. Chrome, Opera, Firefox 브라우저에서 동작하며 웹 브라우저 간 플러그인 없이 실시간으로 영상, 음성, 데이터 등을 통신할 수 있는 기술이다. 일반적으로 WebRPC는 HTTP GET 트랜잭션으로써 입력은 GET 인수에 ?para1=val1&para2=val2&…&paran=valn 형식의 문자열을 대입하는 방식으로 사용된다. Val1,valn 등 매개 변수 값은 URL encode를 적용하여 사용한다. 그러나 GET 방식은 Internet Explorer에서 최대 2,083 문자까지 사용할 수 있고 URL 길이를 제외 하면 2,048 문자까지 사용할 수 있다는[12] 제한이 있다.

본 논문에서는 공개마켓을 블록체인으로 구축할 때 상품정보에는 10개의 이미지를 포함할 수 있도록 설계하였는데 이미지가 포함된 상품정보용 블록을 WebRPC 방식에 의한 P2P 네트워크로 브로드캐스팅(Broadcasting)할 수 있도록 GET방식 대신 2Kb 제한이 없는 POST 방식을 그림 4와 같은 방법으로 사용한다.

상품정보에 포함된 이미지를 블록에 저장하는 방법은 바이너리 형식의 이미지파일은 문자 코드에 영향을 받지 않는 8비트 이진데이터로 바꾸는 base64_encode 작업이 필요하다.

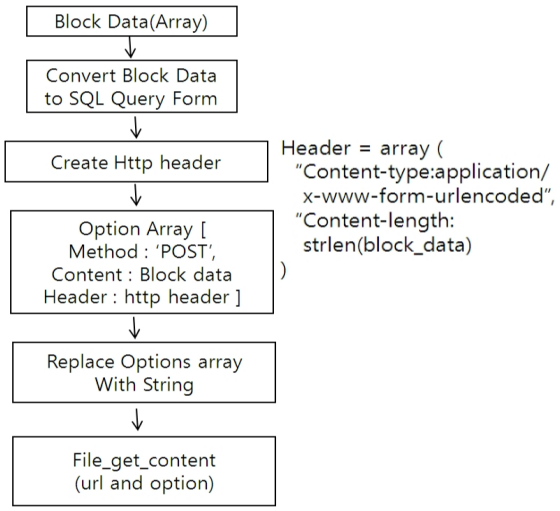


그림 4. 블록 전송을 위한 POST 방식의 RPC 모듈
Fig. 4. Post RPC module for block transmission

인코딩된 문자는 알파벳 대소문자와 숫자 그리고 "+", "/" 등 64개로 이루어져 있고 끝에는 "=" 코드가 붙어있다. 이 결과물에는 예기치 않은 특수문자가 포함될 수 있으므로 chunk_split 명령을 사용하여 base64에서 허용하는 특수문자를 제외한 다른 특수문자가 포함되지 않도록 제거해야 한다.

$img = chung_split(base64_encode(file_get_contents ($tmpfile)));

이미지를 꺼내 보려면 base64 디코딩을 거친 후 img tag를 사용한다.

$img= base64_decode(img)
<imgsrc=$img>

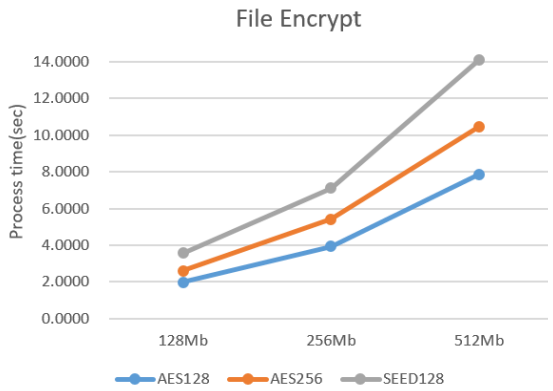
Ⅳ. 개인정보 암호화 및 복호화 성능 측정

양방향 암호화 알고리즘으로 공인된 AES-128-CBC, AES-256-CBC, SEED-CBC를 사용하였을 때의 암호화와 복호화 처리시간을 비교하기 위해 Linux 서버에서 fallocate 명령을 사용하여 128Mb, 256Mb, 512Mb 크기의 더미 파일(Dummy File)을 먼저 생성하였다. 더미 파일은 반복적으로 AES-128-CBC, AES-256-CBC, SEED-CBC 알고리즘으로 암호화하고 그 결과 파일을 생성한 후 다시 동일한 알고리즘으로 복호화하고 그 결과를 생성하였으며 더미 파일과 복호화 파일의 내용을 비교하여 정상 처리되었음을 확인하였다.

그림 5를 보면 암호화를 할 때 모든 크기에서 AES-128 알고리즘이 가장 빠른 것을 알 수 있다.

본 논문에서는 속도 측정 결과를 토대로 AES-128은 P2P 네트워크로 블록을 전파할 때 적용하고 개인정보를 저장할 때는 AES-256 알고리즘을 적용하였다.

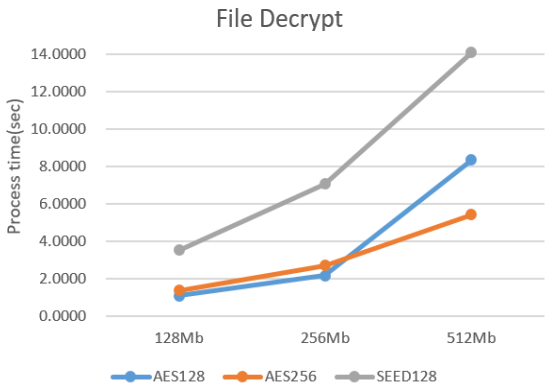
그림 6을 보면 AES-128을 사용하였을 때 256Mb까지는 AES-256에 비해 복호화 속도가 빠르지만 더 큰 파일인 512Mb 부터는 복호화 속도가 느렸다.



	128Mb	256Mb	512Mb
AES128	1.9859	3.9459	7.8760
AES256	2.5988	5.4304	10.4769
SEED128	3.5620	7.1034	14.1268

그림 5. 대칭 키 암호화 알고리즘의 파일 크기별 암호화 처리시간 비교

Fig. 5. Comparison of encrypt processing time by file size of symmetric key encryption algorithm



	128Mb	256Mb	512Mb
AES128	1.0746	2.1504	8.3379
AES256	1.3550	2.7054	5.4173
SEED128	3.5265	7.0565	14.0835

그림 6. 대칭 키 암호화 알고리즘의 파일 크기별 복호화 처리시간 비교

Fig. 6. Comparison of decrypt processing time by file size of symmetric key encryption algorithm

파일 단위 속도 측정 이외에 작은 크기의 문장 단위로 평문(Plaintexts)를 암호화 및 복호화 처리했을 때는 대칭키 암호화 알고리즘인 Seed, AES-128, AES- 256, DES3 모두에서 처리 속도 차이가 경미하였다. 일반적인 블록체인에 비해 비즈니스 모델에 적용된 블록체인에서는 개인정보 사용 빈도가 많기 때문에 평문에서 암호화와 복호화 차이가 경미하더라도 처리 속도와 보안을 고려하여 본 논문에서는 개인정보 보호를 위한 암호화 알고리즘은 AES-128, AES- 256 알고리즘을 적용하였다.

Ⅴ. 결론 및 향후 과제

대표적인 대칭키 암호화 알고리즘인 AES-128, AES-256, SEED-128을 이용하여 128Mb, 256Mb, 512Mb 크기의 파일에 대한 암호화 및 복호화 처리 속도를 측정한 결과 AES-128, AES-256, SEED-128 순서로 속도가 측정되었다. 따라서 본 논문에서는 개인정보를 저장할 때는 AES-256을 적용하고 트랜잭션 메시지에 개인정보를 포함 시킬 때는 AES-128을 이원화하여 적용하였다.

기존 AES-256 암호화 알고리즘이 적용된 개인정보의 경우 대칭키만 알면 일괄적으로 모든 개인정보를 복호화 할 수 있다. 그러나 본 논문에서 제안한 구매자ID와 판매자ID 및 서버 키를 결합하여 생성하는 멀티유저 암호 키 방법을 적용하면 개인정보마다 대칭키가 모두 다르기 때문에 대칭키 하나를 획득했더라도 모든 개인정보를 일괄적으로 복호화 할 수 없다. 디지털서명에 의한 트랜잭션 진위여부만 검증하는 블록체인에 멀티유저 암호키 방법을 적용하여 개인정보를 암호화하고 메시지에 포함시키면 거래 당사자가 아닌 경우, 트랜잭션의 공개된 정보 이외에 메시지에 포함된 암호화된 개인정보는 복호화 할 수 없으므로 메시지 활용도를 높일 수 있게 된다.

향후 연구과제는 멀티 유저 암호 키를 사용한 암호화를 통해 당사자와 운영자를 제외한 제 3자의 개인정보 열람이 불가능하게 되었지만, 운영자가 당사자 매칭 프로그램으로 개인정보에 접근할 경우에 대한 대책을 세우는 것이다.

References

[1] S. Nakamoto, "Bitcoin:A peer-to-peer electronic cash system", https://bitcoin.org/bitcoin.pdf, 2008.

[2] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network", Proceedings First International Conference on Peer-to-Peer Computing, Linkoping, Sweden, pp. 99-100, Aug. 2002.

[3] Mauro Conti, E. Sandeep Kumar, Chhagan Lal, and Sushmita Ruj, "A Survey on Security and Privacy Issues of Bitcoin", IEEE Communications Surveys & Tutorials, Vol. 20, No. 4, pp. 3416-3452, May 2018.

[4] Ministry of the Interior and Safety, "Standard Privacy Guidelines", Ministry of the Interior and Safety Notice 2017-1, 2017.

[5] Hee-Bog Kang, Haeng-Cheon Jang, and Chang-Soo Jang, "IUWT Based Token Authentication Technology", Journal of KIIT, Vol. 17, No. 2, pp. 143-150, Feb. 2019.

[6] M. Crosby, P. Pattanayak, and S. Verma, "Block Chain Technology: Beyond Bitcoin", Applied Innovation Review, No. 2, pp. 6-19, Jun. 2016.

[7] R. Gennaro and S. Goldfeder, "Threshold-Optimal DSA/ECDSA Signatures and an Application to Bitcoin Wallet Security", International Conference on Applied Cryptography and Network Security, London, United Kingdom, pp. 156-174, Jun. 2016.

[8] Harald Aigner, Holger Bock, M. Hutter, and J. Wolkerstorfer, "A Low-Cost ECC Coprocessor for Smartcards", International workshop on Cryptographic Hardware and Embedded System, Cambridge, MA, USA, pp. 107-118, Aug. 2004.

[9] V. B. Kute, P. R. Paradhi, and G. R. Bamnote, "A Software comparison of RSA and ECC", International Journal of Computer Science and Applications Vol. 2, No. 1, pp. 61-65, Apr./May 2009.

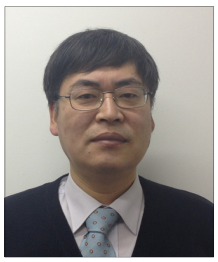
[10] J. W. Bos and J. A. Halderman, "Elliptic Curve Cryptography in Practice", International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, pp. 157-175, Mar. 2014.

[11] R. Roverso and M. H¨ogqvist, "Hive.js: Browser-Based Distributed Caching for Adaptive Video Streaming", in 2014 IEEE International Symposium on Multimedia, Taichung, Taiwan, pp. 143-146, Dec. 2014.

[12] Microsoft, "Maximum URL length is 2,083 characters in Internet Explorer", support. microsoft. com/en-nz/help/208427/maximum-url-length-is-2-083-characters-in-internet-explorer, accept: 2019-08-31.

저자소개

강 희 복 (Hee-Bog Kang)



2015년 2월 : 전남대학교 컴퓨터공학과(공학석사)
2015년 3월 ~ 현재 : 전남대학교 컴퓨터공학과(박사과정)
관심분야 : 컴퓨터네트워크, 클러스터링, 웹서비스, 블록체인, 챗봇

장 행 천 (Haeng-Cheon Jang)

1993년 2월 : 서울과학기술대학교 시각디자인학과(미술학사)
1999년 8월 : 서울과학기술대학교 산업디자인학과(미술학석사)
2015년 3월 ~ 현재 : 전남대학교 컴퓨터공학과(박사과정)
관심분야 : 상거래, 웹서비스

장 창 수 (Chang-Soo Jang)

1980년 2월 : 조선대학교 전자공학과(공학사)
1982년 8월 : 건국대학교 전자공학과(공학석사)
1997년 2월 : 서강대학교 컴퓨터공학과(공학박사)
1984년 ~ 현재 : 전남대학교 컴퓨터공학과 교수
관심분야 : 병렬처리구조, 컴퓨터네트워크, DSP