논문 17-42-07-21 The Journal of Korean Institute of Communications and Information Sciences '17-07 Vol.42 No.07
https://doi.org/10.7840/kics.2017.42.7.1477

Ethereum Whisper 기반의 안전한 모바일 메신저

이 부 형◆, 이 민 영*, 고 현 서*, 명 소 희*, 김 미 옥*, 이 종 혁°

A Secure Mobile Messenger Based on Ethereum Whisper

Boohyung Lee◆, Min-Young Lee*, Hyeon-Seo Ko*,
So-Hee Myung*, Mi-Ock Kim*, Jong-Hyouk Lee°

요 약

현재 많은 사람들이 주로 사용하는 모바일 메신저는 중앙 집중형 서버를 이용하고 있다. 서버를 거쳐서 전송되는 메시지 내용은 서버에 일정 시간 보관된다. 따라서 데이터를 보관한 서버가 해킹될 경우 저장된 데이터에 대한 안전이 보장되지 않는다. 기존의 모바일 메신저에서는 공격자에 의해, 혹은 내부자에 의해서 서버에 저장된 데이터가 유출될 경우 이에 따르는 데이터 위/변조, 프라이버시 침해 뿐만 아니라 서비스량이 많아질 경우 발생되는 서버 과부하에 따른 문제들이 빈번히 발생할 수 있다. 본 논문에서는 이러한 문제를 해결한 새로운 모바일 메신저를 제안한다. 제안하는 모바일 메신저는 Ethereum의 Whisper를 사용하여 서버를 사용하지 않고 메시지를 전송할 수 있으며 모바일 쿠폰의 관리 및 전송 기능까지 지원한다. 전송하는 모든 데이터는 암호화 전송됨으로써 데이터의 기밀성을 보장하고 디지털 서명을 통해 메시지에 대한 무결성 및 인증을 보장한다.

Key Words : Ethereum, Whisper, Blockchain, Mobile Messenger, End-to-End Security

ABSTRACT

Most of popular mobile messengers nowadays are based on centralized servers. The contents of the message transmitted through the server are stored in the server for a certain period of time. Therefore, if the server that holds the data is hacked, the security of the stored data is not guaranteed. In existing mobile messengers, when data stored in the server is leaked by an attacker or by an insider, there is a problem of data forgery and privacy infringement. In addition, problems due to server overload that occurs when service volume increases are frequently occurring. In this paper, we propose a new mobile messenger that solves the problems. The proposed mobile messenger uses Ethereum's Whisper to transmit data without using a server, and also supports management and transmission of mobile coupons. All transmitted data is encrypted and transmitted to ensure the confidentiality of the data and to ensure integrity and authentication of the message through digital signatures.

Ⅰ. 서 론

서버-클라이언트 구조의 모바일 메신저 어플리케이션은 개인정보 보호에 취약하며 데이터 해킹이나 위조 문제도 발생할 수 있다. 또한 서버의 과부하에 따른 트래픽 문제가 발생할 수 있고 DDoS 공격의 표적이 될 수도 있다. 이미 여러 차례 이러한 문제들은 언론을 통해 보도되었고, 국내에서의 가장 대표적인 사

※ 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2017R1A1A1A05001405).
◆ First Author : Department of Software, Sangmyung University, boohyung@pel.smuc.ac.kr, 학생회원
° Corresponding Author : Department of Software, Sangmyung University, jonghyouk@pel.smuc.ac.kr, 정회원
* Department of Computer Science & Engineering, Sangmyung University, {lmyu0582, bye4567, shmyung, kkkmi1220}@naver.com,
논문번호 : KICS2017-03-089, Received March 30, 2017; Revised June 14, 2017; Accepted June 14, 2017

건이 2014년의 '카카오톡 감청사건'이다. 검찰과 경찰이 수사 과정에서 개인정보가 담긴 두 달치 대화기록 내용을 카카오톡 서버를 통해 통째로 들여다본 사실이 뒤늦게 밝혀져 많은 논란이 있었다. 이를 통해 카카오톡 서버에 메시지가 평문 상태로 일정기간 저장되고 있으며 언제든 실시간 감청이 가능함을 알게 되었다. 2014년의 경우 소수의 범죄자와 그 지인들이 사찰의 대상이었지만, 서버에 메시지가 암호화되지 않고 저장되기 때문에 내부 관계자가 악의를 품고 불특정 다수의 개인 정보와 대화 내용을 유포할 수도 있다는 사실이 그 당시에 심각한 문제점으로 제기되었다. 종단 간 암호화를 지원한다는 텔레그램도 비밀채팅이 아닌 일반채팅 모드에서는 평문 상태로 서버에 메시지가 저장되어 원론적인 문제점 해결에는 도움이 되지 않았다.

또한 일부 어플리케이션은 모바일 상에서 구매한 쿠폰이나 기프티콘을 오프라인 상점에서 사용할 수 있고, 선물 목적으로 상대방에게 전달할 수도 있도록 제공하고 있다. 모바일 쿠폰을 상대방에게 보낼 경우 전송하는 데이터가 메시지에 비해 상대적으로 금전적 의미가 더해져 보안의 중요성은 더욱 강조된다.

이에 본 논문에서는 분산 네트워크 환경에서의 Ethereum Whisper 기반 모바일 메신저를 제안하고 직접 구현하였다. 제안하는 시스템은 사용자 간에 메시지와 모바일 쿠폰을 주고받을 수 있다. 통신 과정 중 종단 간 암호화를 지원하고, 블록체인 기술을 활용하여 서버를 두지 않고 사용자의 정보가 드러나지 않는다. 그리고 전송이 완료된 데이터는 네트워크 상에서 일정 시간이 지나면 사라지기 때문에 재전송 공격에 강인하다.

본 논문의 2장에서는 관련 연구로 국내외에서 자주 사용되는 모바일 메신저 어플리케이션의 보안 기능과 Ethereum, Whisper의 특징을 소개한다. 3장에서는 제안하는 시스템의 전체 구성과 동작 과정을 서술하고, 4장에서 구현한 어플리케이션의 기능을 설명한다. 5장에서 기존 모바일 메신저 어플리케이션과의 기능 비교와 함께 보안적으로 어떻게 안전한 메시지를 전송할 수 있는지 서술하고, 6장에서 본 논문의 결론을 낸다.

Ⅱ. 관련 연구

2.1 모바일 메신저 어플리케이션 보안

SNS가 발전하면서 카카오톡, 텔레그램 등 다양한 모바일 메신저가 개발되었고 그에 따라 전송되는 메시지의 보안 문제가 항상 대두되어 왔다. 중앙 집중형 서버를 이용한 메신저 시스템은 서버에 일정 시간 데이터가 저장된다. 그래서 데이터가 저장되어 있는 서버가 해킹될 경우 저장된 데이터는 위/변조될 위험이 있고, 사용량 증가로 인한 서버의 과부하 문제도 따른다. 본 절에서는 대표적인 국내 외 메신저 어플리케이션이 지원하는 보안 기능을 살펴본다.

- 카카오톡의 비밀 채팅 기능[1]: 비밀 채팅 기능을 사용하면 1:1 또는 그룹 단위로 종단 간 암호화가 적용된다. 암호화된 메시지를 풀 수 있는 비밀 키는 서버에 저장되지 않고 개인이 사용하는 단말기에 저장되기 때문에 키가 유출되지 않는다면 대화 내용은 안전하게 보호된다. 또한 비밀 채팅방의 모든 참여자들이 메시지를 읽으면 서버에 임시 저장되는 암호화된 메시지까지도 자동 삭제된다.
- 네이버 라인의 Letter Sealing/True Delete 기능[2]: Letter Sealing이 활성화된 대화에서 사용자 간 전송된 메시지는 개인이 사용하는 단말기에 저장되어 있는 키에 의해서만 복호화가 가능하기 때문에 종단 간 암호화가 지원된다. 키 교환 방식으로는 ECDH (Elliptic Curve Diffie-Hellman)을 사용하며 HMAC을 통해 메시지의 무결성을 보장한다. True Delete는 삭제한 대화 내역의 복구를 방지하기 위해 데이터를 삭제할 때 임의의 데이터나 0으로 이루어진 데이터를 덮어써서 내용을 은닉하거나 파일을 암호화하여 데이터가 복구, 유출되더라도 내용을 알아볼 수 없게 만든다.
- 텔레그램의 Secure Chat 기능[3]: 텔레그램에서는 사용자 간에 전송되는 메시지를 종단 간 암호화를 이용하여 기밀성을 유지한다. 메시지의 암호화에는 AES 방식을 사용한 256 bits 길이의 키를 사용하고, 2048 bits의 RSA 공개키를 통한 Diffie-Hellman 방식을 사용하여 비밀 키를 상호 교환한다.
- 블립의 메시지 보안 기능[4]: 블립은 비트토렌트의 DHT를 활용한 익명성 인스턴트 메신저 어플리케이션이다. P2P 기반이기 때문에 서버가 존재하지 않으며 종단 간 암호화를 지원한다. 대화 모드 중 비밀 대화 모드 (Whisper)를 이용하면 메시지를 보낸 후 25초가 지나면 자동으로 메시지 내용이 삭제된다.

2.2 Ethereum

Ethereum[5]은 2015년 7월 30일 비탈릭 부테린 (Vitalik Buterin)이 개발한 블록체인 기반의 분산 어플리케이션 개발 플랫폼이다. 블록체인은 데이터를 일

련의 블록 형태로 기록하는 분산 데이터베이스로써 해시 트리[6]와 같은 보안 스킴을 통해 데이터의 무결성을 보장한다. 2008년, Bitcoin[7]은 블록체인의 보안성을 이용하여 사용자 간의 거래기록을 투명하게 관리하는데 사용하기 위해 제안되었다. 그리고 2015년, Ethereum은 블록체인을 기반으로 계약서, SNS, email, 전자투표[8] 등 여러 분야의 어플리케이션을 사용자 스스로 개발, 운영할 수 있도록 고안되었다. 다른 플랫폼들과 비교되는 가장 큰 특징은 스마트 컨트랙트[9]가 가능하다는 점이다. 데이터를 실행 가능한 코드 형태로 블록체인에 업로드하여 일정 조건이 만족될 경우에 자동으로 코드를 실행하고 데이터의 소유권에 대한 임의의 규칙도 사용자가 생성할 수 있다. 또한 C++, JAVA, Python, GO 등 다양한 프로그래밍 언어의 API를 지원하여 확장성을 높이고 있다[10]. Ethereum의 분산 어플리케이션(Dapps: Decentralized Applications)은 P2P 시스템에 의해 중앙 통제가 없는 분산화된 형태로 유지 및 관리되는데 그 분류는 다음과 같다.

- 금융 관련 Dapps: 자산을 블록체인 위에 올리고 스마트 컨트랙트의 대상으로 사용하는 어플리케이션 형태; 채권, 주식, 파생상품, 보험, 복권, 도박 등
- 준/비금융 관련 Dapps: 토큰, 쿠폰, 네임코인, 투표 등
- 탈중앙화 조직/회사: 회사나 조직을 블록체인 상에 올려서 운영하는 형태로 월급 지급, 금전거래, 회계 장부 기록, 지분 표시 등이 가능함
- 탈중앙화 자율조직/회사 (DAO/DAC: Decentralized Autonomous Organization/Company): 블록체인 상의 알고리즘으로 자율적으로 의사를 결정하여 회사나 조직 내에서 영업, 회계, 구매, 판매 및 수익 분배 등을 실현하는 형태

Ethereum은 다양한 Dapp의 개발 및 운영을 지원하기 위해 내부적으로 그림 1과 같은 시스템 구조를 가지며 분산화된 데이터베이스를 제공하여 객관적 검증이 가능한 상태 변화 시스템 (State transition system)을 도입, P2P 네트워크 내 노드 간의 합의 알고리즘을 제공한다[11].

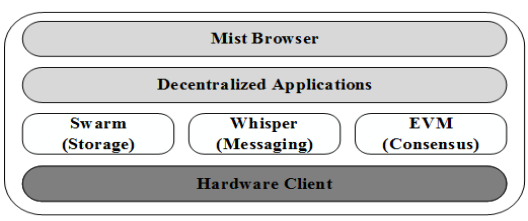

그림 1. Ethereum의 구조
Fig. 1. Structure of Ethereum

- Mist Browser: Ethereum GUI 브라우저
- Decentralized Applications: 분산 어플리케이션
- Swarm: Ethereum P2P 스토리지
- Whisper: Ethereum P2P 메시지 전송 프로토콜
- EVM: Ethereum Virtual Machine; 블록 안의 코드를 컴파일하여 바이트 코드로 변환하고 실행하는 소프트웨어

2.3 Whisper

Whisper는 Ethereum에서의 P2P 네트워크를 사용한 메시지 전송 프로토콜이다. 해시된 *topics*를 기반으로 노드 간에 P2P 통신을 하고, 메시지 암호화와 디지털 서명을 통해 데이터의 무결성 및 기밀성을 제공한다. *Topics*는 Whisper에서 통신 주체를 구분할 때 사용한다. Whisper의 기본 동작은 키 생성, 메시지 전송, 메시지 수신 순으로 이루어진다. 그림 2는 Whisper 프로토콜에서의 메시지 전송 과정을 간략히 나타낸다. 대표 패키지명으로 *shh*를 사용하며 핵심 함수들에 대한 설명은 다음과 같다[12].

- *shh.newIdentity()*: 메시지 교환을 위한 초기화 함수로서, 새로운 통신주체를 생성하고 공개키 쌍을 생성한다. 생성한 공개키 쌍은 세션 키 생성에 사용되는 파라미터 교환과 디지털 서명에서 사용된다. Secp256k1 방식의 공개키를 사용하고, 세션 키는 AES-256을 사용한다.
- *shh.post()*: 메시지를 전송하는 함수이며, 주요 파라미터로 통신 주체 구분값 (*topics*), 데이터 (*payload*), Time-To-Live (*ttl*), 우선 순위 (*work*)를 사용한다. 사용자의 선택에 따라 송신자 (*from*), 수신자 (*to*)를 사용하기도 한다. 만약, 익명의 브로드캐스트 메시지 전송인 경우에는 송신자와 수신자는 함수 내에 파라미터로 사용하지 않는다.

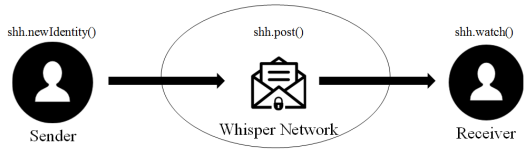

그림 2. Whisper 메시지 전송
Fig. 2. Transmission of a Whisper Message

• *shh.watch()*: 메시지를 수신하는 함수이며, 파라미터로 필터 (*filter*)를 사용한다. 필터는 메시지를 수신받기 위한 일종의 입구 역할을 하는 개체이다.

Ⅲ. 제안하는 기법

본 장에서는 제안하는 모바일 메신저 시스템의 소프트웨어 아키텍처와 네트워크 토폴로지, 데이터 전송 과정을 설명한다.

3.1 Software Architecture

그림 3과 같이 Android OS 위에 XGO (Go CGO cross-compiler)와 Whisper 를 구성하고 geth API를 사용하여 기능을 구현한다. geth API는 Ethereum에서 사용하는 커맨드라인 인터페이스를 지원하며 Go 언어를 이용하여 분산 어플리케이션을 제작할 수 있도록 해준다.

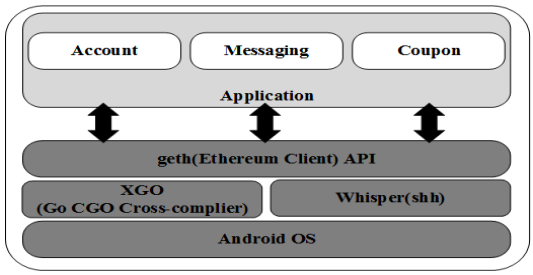

그림 3. 소프트웨어 아키텍처
Fig. 3. Software Architecture

3.2 Network Topology

제안하는 시스템의 네트워크 구성은 그림 4와 같다. 각 노드들은 어플리케이션 사용자이고, 다수의 노드들이 모여분산 P2P 네트워크 (Ethereum 네트워크)를 구성하고 있다. 네트워크를 구성하는 노드들은 어플리케이션을 통해 서로 메시지나 모바일 쿠폰을 전송할 수 있다.

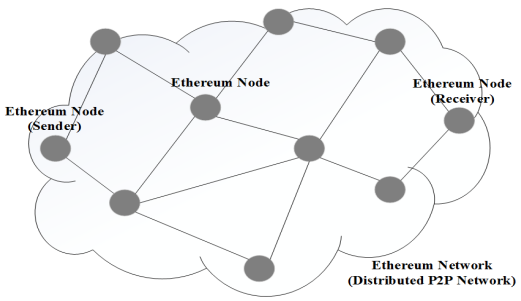

그림 4. 네트워크 토폴로지
Fig. 4. Network Topology

3.3 동작 과정

송신자는 메시지와 쿠폰 데이터를 *post* 메소드를 사용하여 전송한다. 데이터는 P2P 네트워크에 의해 전송되고 *post* 메소드의 매개변수에 입력된 주소를 가진 노드만이 메시지를 복호화할 수 있다. 주소는 사용자의 공개키를 기반으로 만들어지기 때문에 프라이버시가 보호된다. 수신자는 *watch* 메소드를 통하여 수신 대기하고 있는 상태에서 자신의 주소를 따라 온 메시지가 도착하면 암호화된 메시지를 복호화한다. 네트워크에 참여한 데이터는 송신자가 지정한 일정 시간이 지나면 소멸된다. 그림 5는 메시지 전송 과정을 그림으로 나타낸 것이다.

① 송, 수신자는 *web3.shh.newIdentity()*로 공개키 쌍 (개인키, 공개키)을 생성한다.
② 송, 수신자는 키 교환을 위한 필터를 생성하고 *watch* 메소드로 수신 대기 상태에 들어간다. 필터의 매개변수로는 통신 주체를 구분하는 *topics* 만을 필요로 한다.
③ 송, 수신자는 페이로드에 디지털 서명을 붙여서 전송한다. 페이로드에는 세션 키를 만들기 위한 파라미터와 자신들의 공개키가 들어가고, *post* 메소드를 사용한다. <*from*: 자신의 주소, *payload:* 데이터, *topics:* 통신 주체 식별값, *ttl:* Time-To-Live (네트워크 상에서의 메시지의 유효 기간)>
④ *watch* 메소드로 수신 대기하고 있던 양 측은 전송받은 메시지의 서명을 검증한다. 페이로드를 해시

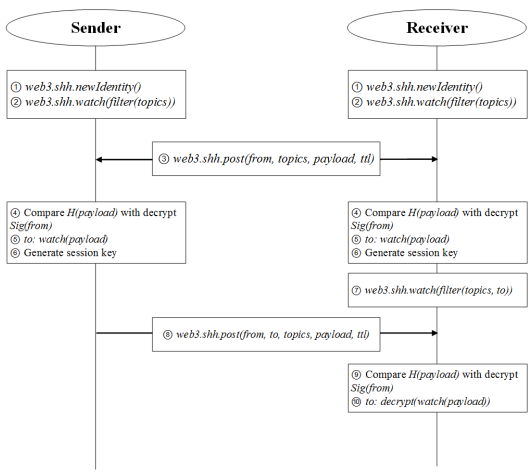

그림 5. 메시지 전송 과정
Fig. 5. Procedure of Message Transmissions

한 값과 서명을 송신자의 공개키로 복호화한 값을 서로 비교하여 일치하는지 확인한다.

⑤ 송, 수신자는 *to*에 수신자의 주소를 저장한다.

⑥ 송, 수신자는 메시지 암호화를 위한 세션 키를 생성한다. 이 과정이 끝나면 메시지 및 쿠폰 전송이 가능하다.

⑦ 수신자는 메시지를 전송받기 위한 필터를 생성하고 수신 대기 상태에 들어간다. 필터의 매개변수로는 *topics*와 *to*를 필요로 한다.

⑧ 송신자는 데이터 원본을 Whisper 내에서 제공하는 ECIES 암호화 방식을 통해 세션 키로 암호화한 뒤 자신의 개인키로 데이터의 해시값을 암호화하여 서명을 만들고, 데이터에 붙여서 수신자에게 전송한다.

<*from:* 송신자의 주소, *to:* 수신자의 주소, *payload:* 메시지 or 쿠폰, *topics:* 통신 주체 식별 값, *ttl:* Time-To-Live>

⑨ 수신자는 전송받은 메시지의 서명을 검증한다. 검증을 위해 페이로드를 해시한 값과 송신자의 공개키로 복호화한 서명을 비교한다.

⑩ 수신자는 암호화된 메시지를 세션 키로 복호화하여 메시지의 내용을 확인한다.

Ⅳ. 구 현

3장의 설계 내용을 기반으로 P2P 분산 네트워크 환경에서의 모바일 암호화 메신저 어플리케이션을 구현하였다. 어플리케이션의 이름은 “Sock-Dark”으로 정하였다. 개발 환경은 아래와 같다.

- OS: Ubuntu 14.04
- SDK: Android Studio (Android 4.4)
- Language: Java, JS
- API: Geth 1.5.0
- Test Device: 삼성 갤럭시 S3 (Android 4.4.4), 갤럭시 S4 (Android 5.0.1)

사용자는 메시지를 전송받을 수신자의 정보 (이름, 핸드폰번호)를 입력하고, 친구로 등록해야 메시지 전송이 가능하다. 기기에 등록한 친구의 계정은 그룹으로 묶어서 관리할 수 있다. 메시지를 받을 친구의 정보를 클릭하면 채팅방이 개설된다. 채팅방을 통해 메시지나 기기에 저장된 쿠폰을 보낼 수 있다. 전송되는 데이터는 세션 키로 암호화되고 디지털 서명을 사용한다. 수신자는 미리 협상한 세션 키를 통해 복호화를 하고 서명을 검증하여 안전하게 데이터를 전달받는다. 구현한 메신저 시스템은 표 1과 같은 기능을 가진다.

Ⅴ. 분 석

본 장에서는 구현한 모바일 메신저와 기존 모바일 메신저 서비스들과의 기능적 차이를 분석하고, 어플리케이션의 보안성을 분석하기 위해 가능한 사이버 공격들과 그에 따른 대응 방법을 서술한다.

5.1 기능 분석

P2P 방식, 종단 간 암호화, 모바일 쿠폰 기능의 유무를 기준으로 2장 1절에서 언급한 모바일 메신저 어플리케이션 (카카오톡, 네이버 라인, 텔레그램, 블립) 들과 Sock-Dark의 기능을 비교하였다. 기능 비교 표는 표 2와 같다.

기존의 메신저 어플리케이션 중 블립만이 P2P 메시지 전송이 가능하고, 종단 간 암호화는 모두 지원하나 카카오톡과 네이버 라인은 수동으로 암호화 기능을 사용해야 메시지를 보호할 수 있다. 국내의 메신저 어플리케이션들은 메시지 전송 뿐만 아니라 자사의

표 1. 제안된 메신저의 기능
Table 1. Functions of the Proposed Messenger

Function	Functions (Detail)		Description
Account	Account creation		Create user account
	Friend management	Grouping	Manage friends for each group
		Register	Register fiends using name and phone number
		Delete	Delete friend from a group
Message	Message transmission		Transmit message using encryption and digital signature
Mobile coupon	Coupon management	Register	Register mobile coupon in phone gallery
		Delete	Delete mobile coupon
	Coupon transmission		Transmit mobile coupon using encryption and digital signature

표 2. 모바일 메신저 기능 비교
Table 2. Comparison of Mobile Messengers

	KakaoTalk	Naver LINE	Telegram	Bleep	Sock-Dark
P2P communication	X	X	X	O	O
End-to-end encryption	Secure chat (Optional)	Letter Sealing (Optional)	O	O	O
Mobile coupon	Only Kakao service	Only LINE service	X	X	Management and transmission of various mobile coupon

플랫폼에 한정한 모바일 쿠폰의 전송 및 관리를 지원한다. 이와 비교해 제안한 Sock-Dark은 블록체인의 P2P 구조를 기반으로 하고 있어 서버가 존재하지 않고, 모든 통신에서 종단 간 암호화를 지원한다. 또한 쿠폰 발급 플랫폼에 구애받지 않고 관리 및 전송이 가능하다.

5.2 기능 분석

이 절에서는 구현한 어플리케이션이 가지는 보안 기능과 함께 어떤 공격에 대응할 수 있는지 분석한다. 구현한 어플리케이션이 가지는 암호 알고리즘[13]및 보안 기능은 아래와 같다.

- 데이터 암호화 (ECIES, AES-256): 기밀성 제공
- 디지털 서명 (Secp256k1 ECDSA[14],[15]): 인증 및 데이터 무결성 제공

메시지나 쿠폰을 주고받기 전에 세션 협상 과정을 통해 사용자가 속한 채팅방에서만 사용할 수 있는 세션 키를 만든다. 세션 키는 채팅 중 메시지나 쿠폰을 전송할 때 암호화에 사용되며, 암호화된 데이터에는 송신자의 개인키를 이용한 디지털 서명이 같이 전송된다. 구현한 어플리케이션은 세션 키를 통한 암호화를 통해 데이터의 기밀성을 보장할 수 있고, 디지털 서명을 통해 인증 및 데이터 무결성이 제공된다. 타원곡선 암호화 방식을 사용하기 때문에 RSA나 DSA보다 짧은 키를 가지고도 높은 수준의 안전성이 보장된다.

공격자는 모바일 메신저 어플리케이션에 대해 중간자 공격을 통해 전송되는 데이터에 대한 악의적인 목적의 도청, 위/변조를 시도할 수 있다. 또한 DDoS 공격을 통해 서버를 무력화시키고, 평문으로 저장되는 개인 정보나 대화 내용을 유출하여 사용자의 프라이버시를 침해할 수 있다. 제안하는 모바일 메신저 시스템은 서버를 두지 않고 종단 간 암호화를 지원하기 때문에 다음과 같은 공격들에 대해 대응할 수 있다. 이 절에서는 암호 키 (세션 키, 공개키 쌍)를 직접 탈취하거나 모바일 기기를 루팅하는 경우는 고려하지 않았다.

5.2.1 데이터 스니핑 및 위/변조

공격자는 중간자 공격을 통해 송, 수신자가 서로 메시지나 모바일 쿠폰을 주고 받을 때, 전송되는 데이터 스니핑 및 위/변조를 시도할 수 있다. 제안하는 시스템을 이용하면 메시지 전송 전, 협상 과정에서 만든 세션 키로 전송 중 데이터가 모두 암호화되기 때문에 데이터 스니핑 및 위/변조 공격을 방어할 수 있다.

5.2.2 프라이버시 침해

서버-클라이언트 구조의 메신저는 사용자의 대화 내용을 평균 3~7일 간 서버에 저장한다. 서버에는 평문 상태로 저장되기 때문에 서버가 공격자로부터 공격을 당해 데이터가 유출된다면 사용자들의 프라이버시 침해는 당연한 결과로 이어질 것이다. 제안하는 시스템에서는 P2P 구조를 사용하여 전송되는 데이터를 어느 한 곳에 저장하지 않으며, 개인정보가 드러나지 않는 주소를 사용함으로써 프라이버시 침해를 방어할 수 있다.

5.2.3 DDoS 공격

DDoS 공격은 여러 대의 공격자나 봇넷을 이용하여 동시에 서비스 거부 공격을 하는 방법이다. DDoS 공격을 통해 충분히 채팅 서버가 무력화될 수 있음을 과거의 사례에서 알 수 있다. 제안하는 시스템에서는 서버를 사용하지 않기 때문에 DDoS 공격의 대상이 특정되지 않는다.

Ⅵ. 결 론

본 논문에서는 Ethereum Whisper를 기반으로 데이터를 안전하게 송, 수신할 수 있는 모바일 메신저 시스템를 제안하고 구현하였다. 제안하는 시스템에서는 메시지 전송 뿐만 아니라 발급 플랫폼의 종류에 상관없이 발행된 모바일 쿠폰을 간편하게 관리 및 전송이 가능하다. 블록체인에서 사용하는 보안 스킴 및 네트워크를 사용하기 때문에 P2P 구조로 송, 수신자 간에 데이터를 주고받는다. 서버를 사용하지 않기 때문에 공격의 대상이 특정되지 않고 관리비용이나 과부하 문제도 고려할 필요가 없고 종단 간 암호화와 디지털 서명을 사용하여 데이터의 무결성과 기밀성이 보장되는 안전한 메시지 전송이 가능하다.

References

[1] D. Schmidt, “A security and privacy audit of KakaoTalk’s end-to-end encryption,” Sept. 2016.

[2] LINE: Whitepaper, LINE Encryption Overview, Sept. 2016.

[3] J. Lee, et al., “Security analysis of end-to-end encryption in telegram,” *SCIS 2017*, Jan. 2017.

[4] F. Fadaie, *How Does Bleep Work?*(2014), Retrieved June 13, 2017, from http://engineering.bittorrent.com/2014/09/17/howdoes-bleep-work/

[5] Ethereum: White paper, A next-generation smart contract and decentralized application platform, Sept. 2014.

[6] B.-H. Lee, S.-B. Lee, J.-Y. Moon, and J.-H. Lee, “Lightweight DTLS message authentication based on a hash tree,” *J. KICS*, vol. 40, no. 10, pp. 1969-1975, Oct. 2015.

[7] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Oct. 2008.

[8] K. Lee, et al., “Electronic voting service using block-chain,” *JDFSL,* vol. 11, no. 2, pp. 123-135, Feb. 2016.

[9] C. D. Clack, Vikram A. Bakshi, and L. Braine, “Smart contract templates: foundations, design landscape and research directions,” *arXiv preprint arXiv:1608.00771*, Aug. 2016.

[10] K. Delmolino, et al., “A programmer’s guide to ethereum and serpent,” University of Maryland, May 2015.

[11] *Ethereum Github Wiki*(2016), Retrieved Mar., 27, 2017, from https://github.com/ethereum/wiki/wiki.

[12] *Whisper PoC 2 Protocol Spec*(2015), Retrieved June, 13. 2017, from https://github.com/ethereum/wiki/wiki/Whisper-PoC-2-Protocol-Spec

[13] K.-J. Ha, C.-H. Seo, and D.-Y. Kim, “Design of validation system for a crypto-algorithm implementation,” *J. KICS*, vol. 39, no. 4, pp. 242-250, Apr. 2014.

[14] *Secp256k1 Wiki*(2016), Retrieved Mar., 27, 2017, from https://en.bitcoin.it/wiki/Secp256k1.

[15] S. Kim, M. Jun, and D. Choi, “Chameleon hash-based mutual authentication protocol for secure communications in OneM2M environments,” *J. KICS*, vol. 40, no. 10, pp. 1958-1968, Oct. 2015.

이 부 형 (Boohyung Lee)

2016년 2월 : 상명대학교 컴퓨터소프트웨어공학과 (공학사)
2016년 3월~현재 : 상명대학교 소프트웨어학과 석사과정
<관심분야> 사물인터넷, 블록체인, 정보보안

이 민 영 (Min-Young Lee)

2017년 2월 : 상명대학교 컴퓨터소프트웨어공학과 졸업
<관심분야> 소프트웨어공학

고 현 서 (Hyeon-Seo Ko)

2013년 3월~현재 : 상명대학교 컴퓨터소프트웨어공학과
<관심분야> 소프트웨어공학

김 미 옥 (Mi-Ock Kim)

2017년 2월 : 상명대학교 컴퓨터소프트웨어공학과 졸업
<관심분야> 소프트웨어공학

명 소 희 (So-Hee Myung)

2017년 2월 : 상명대학교 컴퓨터소프트웨어공학과 졸업
<관심분야> 소프트웨어공학

이 종 혁 (Jong-Hyouk Lee)

2010년 2월 : 성균관대학교 공학박사
2009년 6월~2012년 2월 : 프랑스 INRIA 연구원
2012년 3월~2013년 8월 : 프랑스 그랑제꼴 TELECOM Bretagne 조교수
2013년 9월~현재 : 상명대학교 소프트웨어학과 조교수
<관심분야> IP 이동성, 보안, 프라이버시 보호