The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 23, No. 6, pp.27-32, Dec. 31, 2023. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2023.23.6.27
JIIBC 2023-6-5

암호화를 적용한 위치 공유 앱 개발

Development of Location Sharing App with Encryption

김도은**, 이재문*, 황기태**, 정인환**

Do Eun Kim**, Jae-Moon Lee*, Kitae Hwang**, Inhwan Jung**

요 약 인터넷상에서 친구 위치 찾기 또는 배달 상황 공유 등 앱에서 위치 공유하는 경우가 많아지고 있다. 그러나 위치 정보는 개인의 중요한 정보이고, 경우에 따라 범죄 등 악용될 수도 있기 때문에 이러한 앱을 개발할 때는 위치 정보에 대한 암호화는 반드시 필요하다. 본 논문은 친구 찾기, 모임 장소 정하기 등의 목적으로 친구들 사이의 위치 정보를 암호화하여 공유하는 앱을 개발하는 것이다. 암호화의 성능을 높이기 위하여 비대칭 키를 이용하여 대칭 키를 암호화하여 전송하였고, 위치 공유를 위해서는 오직 대칭 키만 이용하여 암호화하였다. 제안한 앱은 iOS 상에서 개발되었으며, 성능 측정결과 위치 정보 암호화에 대하여 비대칭 키를 사용하는 것보다 대칭 키를 사용하는 경우 최소 5000배 이상 빨랐다는 것을 알 수 있었다.

Abstract Location sharing through apps is increasing, such as finding a friend's location or sharing delivery status on the Internet. However, location information is important personal information, and in some cases can be misused for crimes, and so encryption of location information is essential when developing such apps. This paper develops an app that encrypts and shares location information between friends for purposes such as finding friends and deciding meeting locations. To improve encryption performance, the symmetric key was encrypted and transmitted using an asymmetric key, and for location sharing, only the symmetric key was used to encrypt it. The proposed app was developed on iOS, and performance measurements showed that encryption of location information was at least 5,000 times faster when using a symmetric key than when using an asymmetric key.

Key Words : Location sharing, MQTT, Encryption/Decryption, Asymmetric key, Symmetric key

Ⅰ. 서론

모바일폰의 성능이 고도화됨에 따라 대부분 응용프로그램이 모바일폰상에서 개발되고 있다. 친구 찾기 또는 배달 상황 공유 등의 앱에서 위치 공유와 관련된 앱들이 많이 개발되고 있다.

이러한 앱들에서 위치 공유는 위치 정보를 인터넷상에서 주고, 받고 하여야 한다. 그러나 위치 정보는 개인의 중요한 정보이고, 때로는 범죄 등 악용될 수도 있다. 따라서 이러한 앱을 개발할 때는 위치 정보에 대한 암호화는 반드시 필요하다.

본 논문은 친구 찾기, 모임 장소 정하기 등의 목적으로 친구들 사이의 위치 정보를 공유하는 앱을 개발하는 것이다. 개발될 앱은 위치 정보의 중요성을 고려하여 암

*정회원, 한성대학교 컴퓨터공학부(교신저자)
**정회원, 한성대학교 컴퓨터공학부
접수일자 2023년 10월 3일, 수정완료 2023년 11월 3일
게재확정일자 2023년 12월 8일

Received: 3 October, 2023 / Revised: 3 November, 2023 / Accepted: 8 December, 2023
*Corresponding Author: jmlee@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

호화를 하여 전송한다. 위치 정보의 암호화를 위하여 비대칭 키 및 대칭 키 기술을 적용하였다. 비대칭 키는 오늘날 산업표준으로 사용되는 RSA[1, 2, 3]를 직접 구현하여 사용하였으며, 대칭 키는 단순히 임의의 정수를 사용하였다. 앱의 성능을 높이기 위하여 비대칭 키는 대칭 키를 전송하는 것에만 국한하여 사용하였고 위치 정보의 암호화는 속도가 빠른 대칭 키를 사용하였다.

2장에서는 앱의 개발 과정에서 사용된 기술을 소개하며, 3장에서는 실제로 앱의 설계와 구현에 대하여 논의하였다. 4장에서는 개발된 앱의 성능과 결론을 논하였다.

II. 관련 연구

1. CNN 기술

MQTT(Message Queuing Telemetry Transport)는 M2M(Machine to Machine) 통신, 실시간 모니터링 및 제어 시스템 등 다양한 응용 분야에서 널리 사용되고 있다. MQTT는 메시지 송신자(publisher), 수신자(subscriber), 중개자(broker)로 구성하는데 데이터 송수신이 토픽(Topic)에 따라 정해진다. MQTT의 특징은 작고 경량적인 프로토콜로, 대역폭이 제한된 환경에서도 잘하며, 발행/구독 구조를 사용하여 다대다 전송에 편리하다[4, 5].

하지만 토픽을 알고 있으면 누구나 데이터를 수신할 수 있기에 보안에 취약하다. 따라서 중요정보들이 노출되지 않기 위해서는 통신 구간에 암호화가 필요하다. 송신자/수신자와 중개자 간 보안은 지원하나 종단 간 보안은 지원하지 않고 있다.

2. 암호화

가. 대칭 키 암호화

대칭 키 암호화는 암호화 및 복호화에 같은 암호 키를 쓰는 알고리즘이다. 암호화는 단순히 동일한 키를 사용하여 암호화/복호화를 한다. 예를 들어 평문의 각 바이트에 대칭 키를 곱하여 암호화를 한다면 암호문의 각 바이트에 대하여 대칭 키를 나눔으로써 복호화하는 것이다.

대칭 키 암호화는 내부 구조가 간단한 치환과 전치의 조합으로 되어 있어 비대칭 키에 의한 암호화에 비해 연산 속도가 빨라 대용량 데이터 암호화에 적합하다. 하지만, 키를 교환하는 과정에서 키 탈취 문제가 발생할 수 있으며, 사람이 증가할수록 키 관리도 어려워진다는 단점이 있다[2, 6].

나. 비대칭 키 암호화

비대칭 키 암호는 공개키 암호(Public-key Encryption)라고도 하며, 대칭 키 암호와 달리 암호화 및 복호화에 다른 키를 쓰는 알고리즘이다[2]. 송신자는 수신자의 공개키(Public Key)를 이용하여 암호화하고, 수신자는 자신의 공개 키로 암호화된 암호문을 자신의 개인 키(Private Key)로 복호화한다.

비대칭 키 암호화는 여러 송신자가 하나의 공개 키로 암호화를 수행하기 때문에 키를 관리하기 편하다는 장점이 있지만, 수학적인 난제를 기반으로 복잡한 수학 연산을 사용하기 때문에 대칭 키 알고리즘과 비교해 보면 속도가 느리다. 대표적인 비대칭 키 알고리즘으로는 RSA, ECC, DSA 등이 있다[7].

RSA 기반 비대칭 키 생성 알고리즘은 그림 1과 같다. 그림 1에서 볼 수 있듯이 p, q에 대하여 $\varphi(n)$이 정해지더라도 4라인에서 다양한 e가 존재하고, 5라인에서 $(e \times d) \% \varphi(n) = 1$식을 만족하는 다양한 d가 존재한다. 여기서 % 연산자는 정수에 대한 나머지 연산자이다.

1. 먼저 서로 다른 소수 p, q를 선택한다.
2. $n = p * q$를 계산한다.
3. $\varphi(n) = (p-1) * (q-1)$을 계산한다.
4. $\varphi(n)$과 서로소이고 $\varphi(n) > e$ 인 공개키 e를 선택 한다.
5. $(e \times d) \% \varphi(n) = 1$ 인 개인키 d를 선택한다.

그림 1. 비대칭 키 생성 알고리즘
Fig. 1. Asymmetric key generation algorithm

암호화(n, e)	복호화(n, d)
eText= “”	pText= “”
for m in pText{	for m in eText{
eText += $m^e \% n$	pText += $m^d \% n$
}	}

그림 2. 비대칭 키를 이용한 암호화/복호화 과정
Fig. 2. Encryption/decryption process using asymmetric key

3. 큰 수의 계산

암호화에서 다루는 수는 아주 큰 수(Big Number)이다. 예를 들어 비대칭 키 알고리즘인 RSA의 경우, 1024비트 이상의 키를 사용하는데, 이는 컴퓨터 언어에서 지원하는 기본 자료형(int, long, double 등)으로 계산된 값을 저장하거나 연산할 수 없다. 따라서, 아주 큰 수의 계산을 처리할 방법이 필요하다[8]. 파이선은 정수의 크기

에 제한이 없는 임의 정수 연산을 지원한다. 자바의 경우 'BigInteger'이라는 클래스를 사용해야 한다. C언어도 기본적으로 정수 크기에 제한이 있지만 'GMP'(GNU Multiple Precision Arithmetic Library)와 같은 라이브러리를 설치하면 큰 수 계산이 가능하다. 본 논문에서 사용할 언어는 Swift이고, Swift에서 큰 수를 계산하기 위해 'Swift-BigInt' 라이브러리를 사용하였다.

Ⅲ. 앱 설계 및 구현

1. 앱의 설계

가. 요구사항

위치 공유 앱은 접속자간에 위치를 공유하므로써 '찾아가기', '만나는 장소 정하기' 등의 일에 잘 사용할 수 있다. 이를 위한 앱의 요구사항은 다음과 같다.

- 동일한 토픽으로 접속한 사용자들의 위치가 공유될 수 있어야 한다.
- 사용자는 현재 접속된 모든 사용자를 알 수 있어야 한다.
- 사용자는 접속된 사용자들에 대하여 위치를 공유하고자 하는 특별 사용자들에 대해서만 위치 정보를 보낼 수 있어야 한다.
- 전송되는 모든 위치 정보는 암호화하여 전송할 수 있어야 한다.
- 다른 사용자의 위치 정보를 수신하는 경우, 지도상에 표시되어야 한다.

나. 앱의 구조

개발된 앱은 그림 3와 같이 크게 지도 모듈, 암호화 모듈, 위치관리 모듈, 통신 모듈로 구성된다. 지도 모듈은 지도상에서 현재의 위치, 접속자들의 위치를 표시하고 관리하는 모듈이다. 암호화 모듈은 크게 3가지 기능으로 구성되는데 비대칭 키를 생성하는 기능, 비대칭 키를 이용하여 대칭 키를 암호화/복호화하는 기능과 대칭 키를 이용하여 위치 정보를 암호화/복호화하는 기능이 있다. 위치관리 모듈은 모바일폰에 내장된 GPS를 이용하여 지속적으로 현재의 위치를 업데이트하는 모듈이다. 통신 모듈은 MQTT를 이용하여 송신자와 수신자의 데이터를 송수신하는 모듈이다.

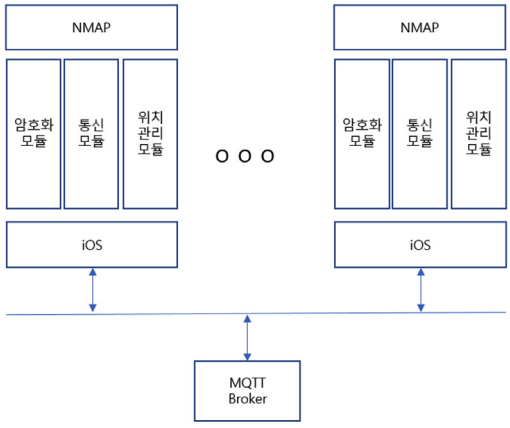

그림 3. 앱의 구조
Fig. 3. Architecture of app

다. 데이터 전송 알고리즘 설계

설계한 앱은 크게 비대칭 키를 공유하기 위한 공개 키의 전송, 비대칭 키로 암호화한 대칭 키 전송 및 대칭 키로 암호화한 위치 정보 전송한다.

- 비대칭 키의 공개 키 전송

하나의 비대칭 키는 공개 키(n, e)와 개인 키(n, d)로 구성된다. 암호화된 데이터를 송수신하기 위해서는 전달하고자 하는 사용자의 공개키를 알아야 하므로 그림 4에서처럼 임의의 사용자는 'who'라는 토픽으로 〈사용자 이름, 공개키〉를 전송한다. 이것은 나중에 접속한 사용자들을 위하여 매초 반복적으로 전송한다. 각 접속자는 이 정보를 받으면 로컬에 저장한다.

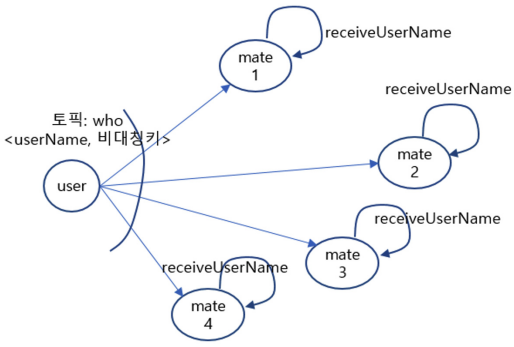

그림 4. 비대칭 키의 공개 키 부분 전송
Fig. 4. Transfer of public key portion of asymmetric key

- 비대칭 키로 대칭 키 암호화 및 전송

4장의 성능 비교에서도 볼 수 있지만, 비대칭 키로 데

이터를 암호화하는 것은 상당히 많은 계산을 요구하므로 일반적으로 데이터를 암호화하여 보낼 때는 대칭 키를 사용한다. 따라서 특정 접속자간 대칭 키의 송수신이 필요하다. 본 논문에서는 이러한 대칭 키를 전달하는 방법으로 비대칭 키를 사용한다. 이것은 한번만 보내면 되기 때문에 성능에 큰 지장이 없다. 그림 5에서처럼 각 접속자별로 다른 대칭 키를 전송하여야 한다.

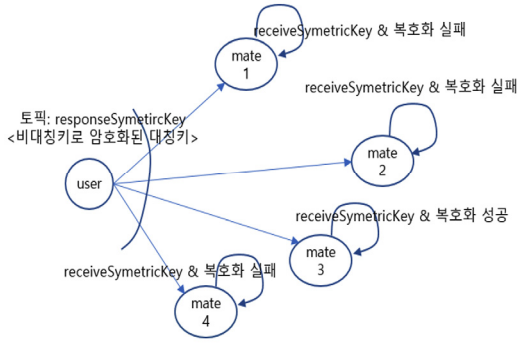

그림 5. 비대칭 키를 이용한 대칭 키 전송
Fig. 5. Symmetric key transmission using asymmetric key

- 대칭 키로 위치 정보 암호화 및 전송

위치 정보는 접속자가 위치가 변경될 때마다 전송하고자 하는 접속자들에게 위치 정보를 보내야 하므로 암호화가 효율적이어야 한다. 이를 위하여 비대칭 키로 암호화되어 수신된 대칭 키로 위치 정보를 암호화하여 전송한다. 그림 6에서와 같이 대칭 키는 접속자별로 다르므로 위치 정보를 전송하고자 하는 모든 접속자에게 데이터를 암호화하여 전송하여야 한다.

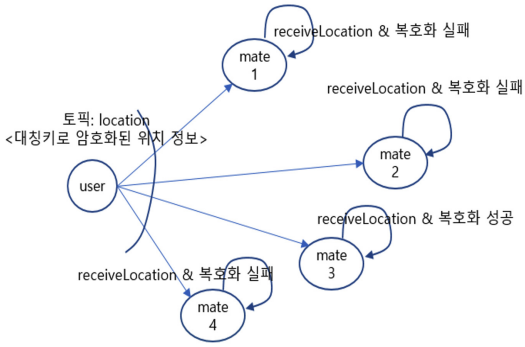

그림 6. 대칭 키로 위치 정보 암호화 및 전송
Fig. 6. Encrypt and transmit location information with symmetric keys

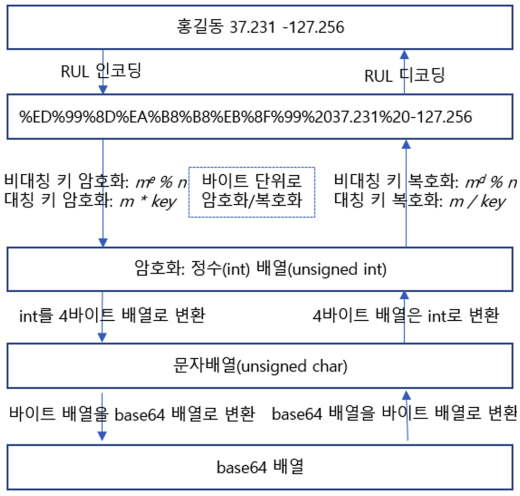

그림 7. 데이터의 암호화/복호화 과정
Fig. 7. Data encryption/decryption process

2. 앱의 구현

가. 구현환경

제안한 앱은 iOS상에서 swift 언어로 개발되었다. 지도를 사용하기 위하여 NAVER 지도 API v3[9]을 사용하였고, MQTT 클라이언트는 CocoaMQTT[10]를 사용하였다. MQTT 브로커는 별도로 구축한 것이 아니고 hivemq에서 제공하는 MQTT 브로커[11]를 사용하였다.

나. 암호화/복호화의 구현

데이터의 암호화 및 복호화는 그림 7과 같이 구현되었다. 먼저 사용자 이름에 한글이 들어갈 수 있기 때문에 플레인 텍스트를 URL 인코딩한다. 그림 7에서 볼 수 있듯이 '홍길동'이라는 글자가 URL 인코딩되어 다양한 '%문자'로 나타남을 알 수 있다. 다음 단계로 URL 인코딩된 문자열에서 각 문자에 대하여 암호화를 수행한다. 비대칭 키를 이용한 암호화는 $m^e \% n$를 계산하여야 한다. 여기서 m는 암호화하고자 하는 문자의 값(0-255)이며 e, n은 수신받는 접속자의 공개키이다. 보통 e의 값이 4 자리수이므로 m^e를 계산하는데 많은 시간이 소요된다. 반면 대칭 키로 암호화하는 경우 $m \times key$만 계산하고 key는 5자리 정수이므로 크게 시간이 소요되지 않는다. 이렇게 암호화 과정을 거치면 하나의 바이트는 하나의 정숫값을 가지게 된다. MQTT 데이터 전송을 위하여 정수 배열을 문자 배열로 변환하고 문자 배열은 base64로 다시 인코딩하는데 이것은 단순한 변환이다.

복호화는 정확히 암호화의 역순으로 진행하였다.

다. 전체 프로그램 제어

전체 프로그램은 두 개의 스레드에 의하여 실행되도록 구현하였다. 메인 스레드는 사용자와 인터렉션하여 지도의 이동, 위치 정보를 보내고자 하는 사용자의 선택 등을 실행하며, 다른 하나의 스레드는 그림 8과 같은 프로그램을 지속적으로 수행된다.

서브 스레드는 시작과 동시에 가장 먼저 사용자 이름을 입력받는다. 그런 후 지도와 접속자 목록을 초기화한다. 또한 GPS를 이용하여 위치 정보를 초기화하고 MQTT 브로커와 접속하고 이후 필요한 토픽들을 등록한다. 마지막으로 비대칭 키를 생성한다.

이러한 초기화 이후 서브 스레드는 〈사용자 이름, 공개키〉 전송하기, 각 접속자에 대하여 대칭 키를 받지 못한 경우 대칭 키 요청을 보내며, 대칭 키를 이미 받고 위치 정보 보내기로 선택되어 있으면 위치 정보를 대칭 키로 암호화하여 전송한다. 이 작업은 매1초 마다 반복하여 실행한다. '〈사용자 이름, 공개키〉 전송하기'를 매 1초마다 반복하여 전송하는 것은 나중에 접속한 사용자들도 공개키를 알아야 하기 때문이다.

```
지도, 접속자, 위치 정보, MQTT 초기화
비대칭키 생성 // 한번만 생성

while(True){
  〈사용자이름, 공개키〉 'who' 토픽으로 전송
  foreach mate in mates // mates는 접속자 목록
    if mate가 대칭키를 보내지 않았으면{
      〈대칭키 요청〉 'requestSymetricKey' 토픽으로 전송
    }
    if mate가 선택되었으면{
      〈선택된 접속자, 위도, 경도〉 대칭키로 암호화하여 'location'
토픽으로 전송
    }
  }
  sleep(1)
}
```

그림 8. 서브 스레드에 의한 프로그램 제어
Fig. 8. Program control by subthread

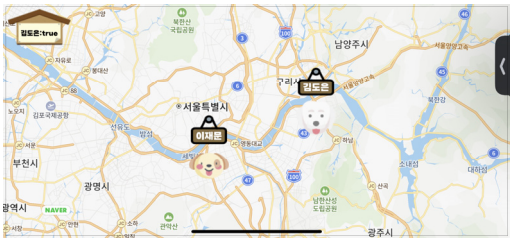

그림 9. 개발된 앱의 스크린 샷
Fig. 9. Screenshot of the developed app

IV. 성능 논의

그림 2에서 e와 d는 4자리 이상의 정수이고 m은 문자이므로 0-127의 값을 갖는다. 예를 들어 64^{5000}을 계산하면 64를 5000번 곱해야 하는데 이는 9031자리 수를 갖는다. 정수의 크기를 32 비트라고 할 때 2^{32}는 10자리 수이다. 따라서 9031자리 수를 계산하기 위해서는 별도의 알고리즘을 사용해야 하고 이로 인하여 일반적인 연산보다 훨씬 많은 시간 비용이 소요된다. 따라서 비대칭 키에 의한 암호화/복호화 비용은 매우 크다. 표 1은 비대칭 키를 이용한 암호화/복호화 시간을 키의 크기에 따라 측정한 시간을 나타낸다. 여기서 암호화에 적용된 평문은 "홍길동 126.9779692, 37.566535"으로 서울에 있는 홍길동의 위치 정보이다. 표에서 볼 수 있듯이 비대칭 키 e, d의 크기에 따라 급격히 실행 시간이 증가함을 알 수 있다.

표 1. 비대칭 키를 이용한 암호화/복호화 시간
Table 1. Encryption/decryption time using asymmetric key

e, d의 범위	암호/복호화 시간($msec$)
400 ~ 600	864
1400 ~ 1600	2,477
2400 ~ 2600	3,860
4400 ~ 4600	7,290
8400 ~ 8600	13,870

동일한 평문을 단순히 대칭 키를 사용하는 경우는 약 0.15밀리초($msec$)로 암호화/복호화를 할 수 있었다. 이것은 최소 5,760배에서 최대 92,467배 빠른 속도이다. 따라서 비대칭 키를 사용한 암호화/복호화는 최소한으로 하여야 하고 가능한 대칭 키를 이용한 암호화/복호화를 하여야 한다. 이런 관점에서 대칭 키를 전송하기 위해서는 비대칭 키를 사용하여 암호화/복호화를 하고 위치 정보의 공유를 위해서 대칭 키를 이용한 본 논문의 접근 방법은 타당하다고 하겠다.

V. 결 론

본 논문은 위치 공유 앱에 위치 정보에 대한 암호화/복호화 기술의 적용에 관한 연구이다. 제안한 위치 공유 앱은 iOS 상에서 iPhone에 동작하도록 개발되었다. 위치 정보에 대한 보안의 중요성을 인식하여 종단간 암호화/복호화 기법을 적용하여 개발하였다.

위치 정보를 공유하기 위하여 비대칭 키와 대칭 키를 적용하여 암호화/복호화 비용을 최소화하였다. 비대칭 키를 이용한 암호화/복호화는 한 번만 하도록 하였고 이후 위치 공유를 위한 위치 정보의 암호화/복호화는 대칭 키를 사용하여 앱의 성능을 개선했다.

References

[1] Chandra M. Kota and Cherif Aissi , “Implementation of the RSA algorithm and its cryptanalysis” Proceedings of the 2002 ASEE GulfSouthwest Annual Conference, 2002.

[2] Halak, Basel, Yildiran Yilmaz, and Daniel Shiu. "Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications", IEEE Access 10, pp. 76707-76719, 2022.

[3] Minaud, Brice, and Michael Reichle. "Dynamic local searchable symmetric encryption", Annual International Cryptology Conference. Cham: Springer Nature Switzerland, pp. 91-120, 2022.

[4] Do-Eun Kim, Hee-Jin Kong, Ji-Hu Woo, Jae-Moon Lee, Kitae Hwang, Inhwan Jung, "Development of Intelligent CCTV System Using CNN Technology", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 23, No. 4, pp.99-105, 2023.

[5] Dong Jun Kim, Yu Jin Choi, Kyung Min Park, Ji Hyun Park, Jae-Moon Lee, Kitae Hwang, In Hwan Jung, "Detection of Smoking Behavior in Images Using Deep Learning Technology", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 23, No. 4, pp.99-105, 2023.

[6] Baksi, Anubhab, et al. "A survey on fault attacks on symmetric key cryptosystems", ACM Computing Surveys Vol. 55, No. 4, pp. 1-34, 2022.

[7] Yang, Wenxin. "ECC, RSA, and DSA analogies in applied mathematics", International Conference on Statistics, Applied Mathematics, and Computing Science. Vol, 12163.SPIE, 2022.

[8] Fischer, Sabine. "Formal verification of a big integer library", DATE Workshop on Dependable Software Systems, 2008.

[9] Jeon, Sungwoo, et al. "Map API-Based Evacuation Route Guidance System for Floods", Applied Sciences, Vol. 13, No. 16, 2023.
DOI: https://doi.org/10.3390/app13169141

[10] Marini, Daniele LR. "Design of a Simple Planetary Machine", Imago Cosmi. Springer, Cham, pp. 389-421, 2023.

[11] Banno, Ryohei. "Performance Evaluation of MQTT Communication with Heterogeneous Traffic", IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2023.

저 자 소 개

김 도 은(비회원)

- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 정보보안, iOS, 인공지능

이 재 문(정회원)

- 한양대학교 전자공학과(학사)
  한국과학기술원 전기및전자공학과 (석사)
  한국과학기술원 전기및전자공학과 (박사)
- 경력 : 한국통신 연구개발단
- 관심분야 : 기계학습, 게임프로그래밍, 감성컴퓨팅

황 기 태(정회원)

- 서울대학교 컴퓨터공학과 학사
  서울대학교 컴퓨터공학과 석사
  서울대학교 컴퓨터공학과 박사
- 경력
  University of Callifornia, Irvine 방문교수
  University of Florida 방문 교수
- 주관심분야 : 모바일 시스템

정 인 환(정회원)

- KAIST 정보및통신공학과 박사
- 삼성전자 시스템사업부 수석연구원
- 한성대학교 컴퓨터공학과 교수
- 주관심분야 : 망관리, 멀티미디어신, IoT

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음