KIPS Tr. Comp. and Comm. Sys.
Vol.7, No.2 pp.49~58 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2018.7.2.49

Secure Certificates Duplication Method Among Multiple Devices Based on BLE and TCP

Sung-Hwan Jo† · Gi-Tae Han††

ABSTRACT

A certificate is a means to certify users by conducting the identification of the users, the prevention of forgery and alteration, and non-repudiation. Most people use an accredited certificate when they perform a task using online banking, and it is often used for the purpose of proving one's identity in issuing various certificates and making electronic payments in addition to online banking. At this time, the issued certificate exists in a file form on the disk, and it is possible to use the certificate issued in an existing device in a new device only if one copies it from the existing device. However, most certificate duplication methods are a method of duplication, entering an 8-16 digit verification code. This is inconvenient because one should enter the verification code and has a weakness that it is vulnerable to security issues. To solve this weakness, this study proposes a method for enhancing security certificate duplication in a multi-channel using TCP and BLE. The proposed method: 1) shares data can be mutually authenticated, using BLE Advertising data; and 2) encrypts the certificate with a symmetric key algorithm and delivers it after the certification of the device through an ECC-based electronic signature algorithm. As a result of the implementation of the proposed method in a mobile environment, it could defend against sniffing attacks, the area of security vulnerabilities in the existing methods and it was proven that it could increase security strength about 10^{41} times in an attempt of decoding through the method of substitution of brute force attack existing method.

Keywords : BLE, ECC, Certification, Multi Channel Authentication, Digital Signature, Smart Device

BLE 및 TCP 기반 다중 디바이스 간 안전한 인증서 복사 방법

조 성 환† · 한 기 태††

요 약

인증서는 사용자의 신원확인 및 위·변조 방지, 부인방지 등의 기능을 수행하여 사용자를 증명할 수 있는 수단이 된다. 대부분의 사람들이 인터넷뱅킹을 이용한 업무를 수행할 때 공인인증서를 사용하며, 인터넷뱅킹 외에도 각종 증명서 발급, 전자 결제 등에서도 신원을 입증하는 용도로 많이 사용되고 있다. 이때 발급받은 인증서는 디스크 상에 파일 형태로 존재하며, 만약 새로운 디바이스에서 인증서를 사용하기 위해서는 기존의 디바이스에서 발급받은 인증서를 복사해야 사용이 가능하다. 하지만 대부분의 인증서 복사 방법은 8~16자리의 인증번호를 입력하여 복사하는 방법이며, 이는 인증번호를 입력해야 되는 번거로움이 있고, 보안에 취약하다는 단점이 있다. 이러한 단점을 해결하기 위해 본 논문에서는 TCP와 BLE를 사용하는 다중 채널에서의 보안강화 인증서 복사 방법을 제안한다. 제안하는 방법은 1) BLE Advertising data를 이용하여 상호간에 인증 가능한 데이터를 공유하고, 2) ECC기반 전자서명 알고리즘을 통해 디바이스 인증 후 대칭키 알고리즘으로 인증서를 암호화하여 전달한다. 제안하는 방법을 모바일 환경에서 구현한 결과 기존방법의 보안취약영역인 스니핑 공격에 대한 방어가 가능하며, 무작위 대입 공격을 통한 복호화 시도 시 기존의 방법보다 약 10^{41}배 정도의 보안강도를 높일 수 있음을 보였다.

키워드 : BLE, ECC, 인증서, 다중채널인증, 전자서명, 스마트디바이스

1. 서 론

스마트폰이 보급된 이후 PC를 이용하여 하던 작업을 스마트폰에서도 수행할 수 있게 되었다. 스마트폰은 3G, LTE, Wi-Fi와 같은 통신방법을 통해 인터넷에 연결할 수 있으며, 이를 이용하여 웹 서핑을 하거나 업무 중인 문서를 열람하는 등 기존의 PC에서 수행하던 대부분의 작업을 수행할 수 있게 되었다.

이중 많은 사람들이 스마트폰을 이용한 인터넷 뱅킹 서비스를 사용하고 있다. 2017년 6월 말을 기준으로 우리나라의 금융기관에 등록된 인터넷뱅킹 고객 수는 약 1억 2천명이며,

† 준 회 원 : 가천대학교 IT융합공학과 석사과정
†† 정 회 원 : 가천대학교 컴퓨터공학과 교수
Manuscript Received : November 13, 2017
First Revision : December 11, 2017
Accepted : December 26, 2017
* Corresponding Author : Gi-Tae Han(gthan@gachon.ac.kr)

이중 스마트폰 뱅킹 등록 고객 수는 약 8천명으로 전체 인터넷뱅킹 등록 고객수의 약 63.8%이다[1].

우리나라의 인터넷 뱅킹은 초기부터 인증서 기반으로 구축되었으며, NPKI(National PKI)를 통해 발급하는 공인인증서가 의무화된 이후 인터넷 뱅킹에 있어 필수적인 요소로 자리 잡았다[2]. 2015년 3월 공인인증서 의무 사용 규정이 폐지되었지만, 아직까지도 대부분의 인터넷 뱅킹 사용자들은 공인인증서를 통해 사용자 인증을 수행하고 있다.

발급받은 공인인증서는 대부분 파일형태로 보관하고 있고, 기존에 PC를 통해 발급받은 공인인증서를 스마트폰에서 사용하기 위해서는 공인인증서를 스마트폰으로 복사하는 과정이 필요하다. 이때 보안을 강화하지 않으면 매우 위험한 상황이 초래될 수 있다.

본 논문에서는 다중 디바이스 상에서 공인인증서 사용을 위한 BLE 및 TCP 기반의 디바이스 인증 및 키 생성을 통한 보안이 강한 안전한 인증서 복사 방법을 제안한다. 제안하는 방법은 BLE Advertising data packet을 사용하여 상호간에 인증할 수 있는 값을 공유한 뒤 이를 이용하여 디바이스 인증 및 키 생성을 수행한다. 현재 사용되고 있는 대부분의 스마트폰 및 노트북에는 Bluetooth 모듈이 탑재되어 있으며, Bluetooth 모듈이 없는 PC의 경우 USB 형태의 Bluetooth 동글을 이용하여 Bluetooth 통신을 수행할 수 있다. 본 논문에서는 Android 환경에서 제안하는 방법을 구현한 뒤 기존의 방법과 편의성 및 보안 강도를 비교하여 성능을 평가한다.

2. 관련 연구

2.1 공인인증서 복사 보안

발급받은 공인인증서는 Fig. 1과 같이 디바이스 저장소에 파일 형태로 저장되어 있다[3]. signCert.der 파일에는 서명 알고리즘, 발급자, 유효기간, 공개키 등이 저장되어 있고, signPri.key 파일은 signCert.der 파일에 저장된 공개키의 키쌍인 개인키가 사용자 비밀번호로 암호화되어 저장되어 있다. 사용자가 공인인증서를 이용한 사용자 인증을 하기 위해 사용자 비밀번호를 입력하면 입력한 비밀번호를 키로 사용하여 signPri.key 파일에 있는 개인키를 복호화한 뒤 이 개인키를 이용하여 전자서명 알고리즘으로 사용자 인증을 수행하게 된다.

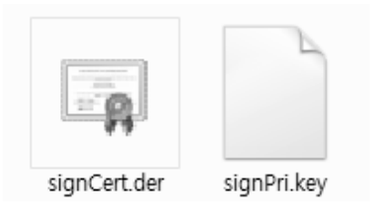

Fig. 1. Certificate and Private Key Stored as a File

우리나라의 인터넷 뱅킹 시스템을 이용한 공인인증서 복사 방법은 Table 1과 같다. 대부분의 은행은 공인인증서를 복사할 때 스마트폰 화면에 나타나는 인증번호를 PC에 입력하여 디바이스간의 인증을 한 뒤 이를 이용하여 인증서를 암호화하여 복사하는 방법을 사용한다. 인증번호의 길이는 8~16자로 서비스하는 기업마다 다르다. 인증번호는 모두 숫자로 이루어져 있으며, 8자리의 경우 27bit, 12자리의 경우 40bit, 16자리의 경우는 52bit의 크기로 표현이 가능하다.

[4]는 A사의 인증번호 입력 방식을 통한 인증서 복사 과정을 분석하였다. PC에서 스마트폰으로 인증서를 복사하는 과정의 데이터 패킷을 스니핑 공격으로 획득하였고, 자바 디컴파일을 통해 안드로이드용 스마트폰 뱅킹 어플리케이션의 소스코드를 분석하여 인증번호 생성 과정 및 인증서 암·복호화 과정을 분석하였다. 그 결과 인증번호를 이용해 생성된 키를 이용하여 인증서가 암호화된다는 점을 확인하였으며, 인증서 복사 과정 중 스니핑 공격을 통해 획득한 패킷을 무작위 대입공격을 이용해 복호화하는 알고리즘을 구현하였다. 현재 사용되고 있는 인증번호 중 가장 긴 16자리의 인증번호는 52비트며, 이는 현재 보급되고 있는 컴퓨터를 이용하여 무작위 대입공격을 하였을 때 안전하지 않은 길이로 확인되었다.

국내 특정 금융기관의 경우 인증번호를 입력하는 방법 외에 QR코드를 이용한 인증서 복사 방법을 지원한다[5]. 이 방법은 인증서 복사 과정에서 PC에 나타나는 QR코드를 스

Table 1. Certificate Duplication Method by Bank

Bank	Method of certificate duplication
W Bank	- Perform certificate duplication, entering a 12-digit verification code, generated on a smart phone on a PC
	- Perform certificate duplication, scanning a QR code, generated on a PC on a smart phone
S Bank	- Perform certificate duplication, entering an 8-digit verification code, generated on a smart phone on a PC.
K Bank	- Perform certificate duplication, entering a 12-digit verification code, generated on a PC or smart phone
N Bank	- Perform certificate duplication, entering a 16-digit verification code, generated on a smart phone and resident registration number on a PC

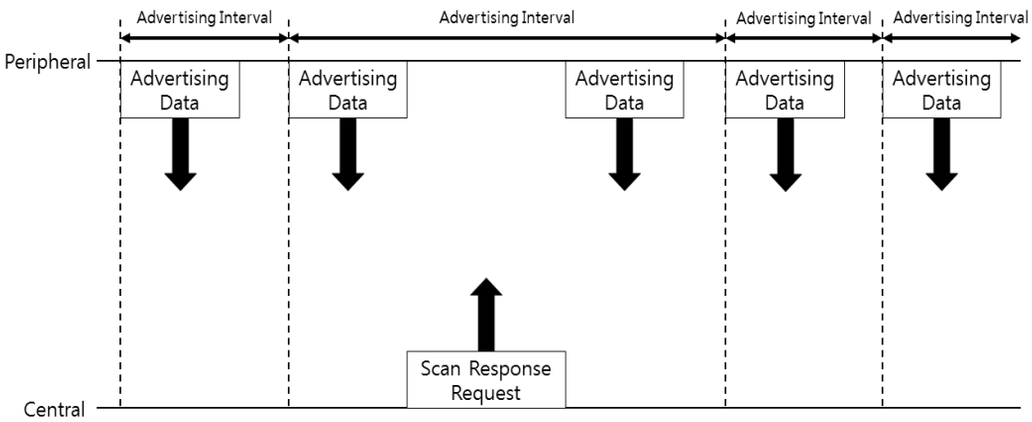

Fig. 2. Send and Receive Process of Advertising Data Packet

마트폰의 카메라로 스캔하여 디바이스간 인증을 수행한 뒤 인증서를 복사한다. 이때 QR 코드에는 디바이스를 인증할 수 있는 데이터가 들어있으며, PC에서 생성한 QR코드의 데이터와 스마트폰에서 스캔한 QR코드의 데이터가 일치했을 때 인증서 복사를 수행하게 된다. QR코드를 이용한 인증서 복사의 경우 인증번호를 입력하는 복잡함을 거치지 않고 간편하게 인증서를 복사할 수 있다는 장점이 있지만 스크린 캡쳐와 같은 악성 코드에 취약하다는 단점이 있기 때문에 사용에 주의하여야 한다.

2.2 Bluetooth Low Energy

Bluetooth Low Energy는 2009년 블루투스 SIG(Special Interest Group)에서 발표한 Bluetooth 4.0 규격에서 저전력 디바이스를 위한 표준이다[6]. 기존의 Bluetooth 통신은 기기간의 Pairing을 통해 장치를 연결하고 데이터를 주고받지만, BLE는 기기간의 Pairing을 하지 않고 Advertising data packet을 송·수신하여 데이터를 주고받는다. Advertising data의 길이는 최대 31byte로 매우 작으며, BLE 송신 디바이스는 sleep 상태로 대기하며 일정 주기로 Data 패킷을 전송하게 된다. 그렇기 때문에 BLE 통신은 기존의 Bluetooth 통신에 비해 소모되는 전력량이 적고, 작은 데이터를 주고받기 때문에 기존에 활용되던 헤드폰, 마우스와 같은 장치에 적용하는 것은 부적절하며, 저전력을 요구하는 IoT 디바이스에 적합하다고 볼 수 있다[7].

BLE 통신을 하는 디바이스는 크게 Peripheral과 Central로 구분한다. Peripheral은 Advertising data packet을 송신하는 디바이스로 주로 저전력을 요구하는 제한된 리소스를 가진 IoT 디바이스와 같은 소형 장치가 포함된다. Central은 Advertising data packet을 수신하는 디바이스로 주로 충분한 전원과 메모리 등의 리소스를 가지고 있는 스마트폰, 태블릿과 같은 장치가 포함된다. Peripheral과 Central 간의 Advertising data packet을 주고받는 과정은 Fig. 2와 같다. 먼저 Peripheral은 Advertising interval을 주기로 Advertising data를 전송한다. 이때 Advertising interval의 길이를 짧게 하면 Advertising data를 더 빠르게 송신할 수 있지만 그만큼 전력 소모가 심해진다. Central은 주위에서 전송되는 Advertising data를 scan하여 데이터를 수집한다. 이때 Advertising data를 송신한 디바이스의 추가적인 데이터를 원한다면 Central은 Scan response request를 Peripheral에게 전송하며, Scan response request를 수신한 Peripheral은 Scan response data를 송신한다.

Advertising data는 Fig. 3과 같이 n개의 AD Structure로 이루어져 있다. AD Structure의 첫 1byte는 AD length로 해당하는 AD structure의 전체 길이를 나타내고, 그 다음의 1byte는 AD type로 해당하는 AD Structure의 타입을 나타낸다[8]. 나머지 데이터는 AD Payload로 실제 데이터가 담겨져 있다.

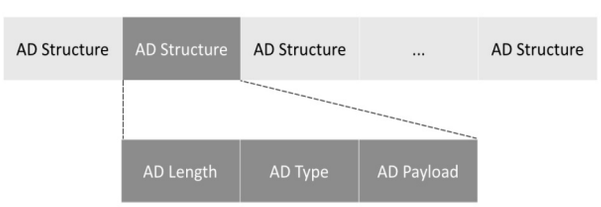

Fig 3. Ad structure data format

2.3 ECC

ECC(Elliptic curve cryptography)는 1985년 Miller와 Kobitz가 제안한 타원곡선 암호화 방법으로 유한체상의 타원곡선간의 연산에서 정의되는 이산대수 문제의 계산이 어려운 ECDLP(Elliptic curve Discrete Logarithm Problem)를 이용한 암호화 방법이다[9]. ECC는 같은 공개키 알고리즘인 RSA에 비해 작은 키를 이용하여 비슷한 보안 강도를 낼 수 있다는 특징이 있다[10]. Table 2는 ECC와 RSA의 키 길이별 보안 강도를 나타낸다.

Table 2. Comparison of the Security Strength for Each Key Size between ECC and RSA

MIPS	ECC (bit)	RAS (bit)	RSA:ECC key size ratio
10^4	106	512	5:1
10^8	132	768	6:1
10^{11}	160	1024	7:1
10^{20}	210	2048	10:1
10^{78}	600	21000	35:1

ECC는 Equation (1)과 같은 곡선 위의 점을 이용하여 연산을 수행한다. 이때 p는 충분히 큰 소수이며, a와 b는 p보다 작은 정수이다.

$$y^2 = x^3 + ax + b \bmod p \tag{1}$$

Fig. 4는 타원곡선 위의 점 P와 Q를 이용하여 R=P+Q를 계산하는 방법이다. 점 P와 Q를 일직선으로 연결하였을 때 만나는 점을 -R이라 하며, 이 점과 x축으로 대칭되는 점이 R이다. P+Q=R을 계산하는 계산식은 Equation (2)와 같다.

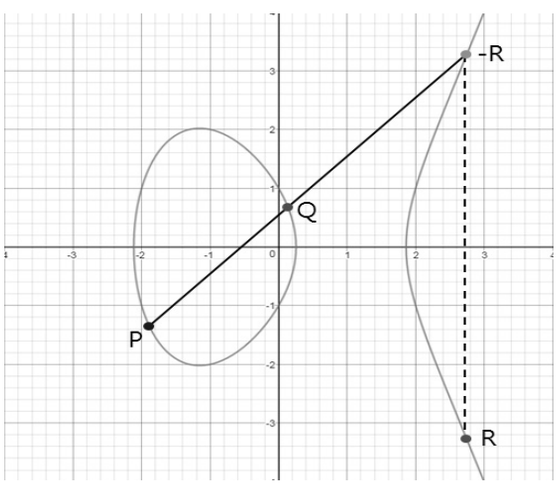

Fig. 4. Addition for Two Points on an Elliptic Curve

$$\begin{aligned} &P(x_p, y_p) + Q(x_q, y_q) = R(x_r, y_r) \\ &\text{if}(P \neq Q) \\ &\quad m = ((y_q - y_p) \times (x_q - x_p)^{-1}) \bmod p \\ &\text{if}(P = Q) \\ &\quad m = ((3x_p^2 + a) \times 2y_p^{-1}) \bmod p \\ &x_r = (m^2 - x_p - x_q) \bmod p \\ &y_r = (m(x_p - x_r) - y_p) \bmod p \end{aligned} \tag{2}$$

Fig. 5는 타원곡선상의 동일한 점을 더하여 P+P=R을 계산하는 방법이다. 점 P의 접선과 만나는 점을 -R이라 하며, 이 점과 x축으로 대칭되는 점이 R이다. 점 P와 정수 k의 곱연산은 점 P를 k번 더하여 계산할 수 있다.

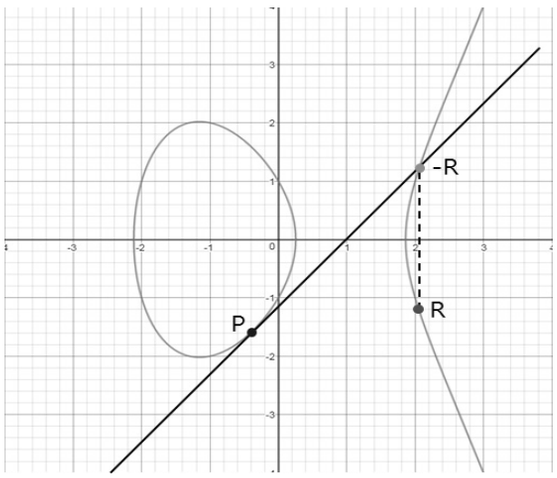

Fig. 5. Addition for the Same Point on an Elliptic Curve

이와 같이 타원곡선상의 덧셈과 곱셈 연산을 이용하여 키 생성 및 전자 서명 알고리즘에 적용할 수 있다. 두 사용자가 타원곡선상의 점을 이용하여 대칭키를 생성하는 ECDH (Elliptic Curve Diffie-Hellman)알고리즘의 진행 과정은 Fig. 6과 같은 순서로 진행된다. 이때 G는 타원곡선상의 공유된 한 점이며, ECDLP에 의해 공개키인 A또는 B를 이용하여 개인키 a또는 b를 계산하기는 매우 어렵다.

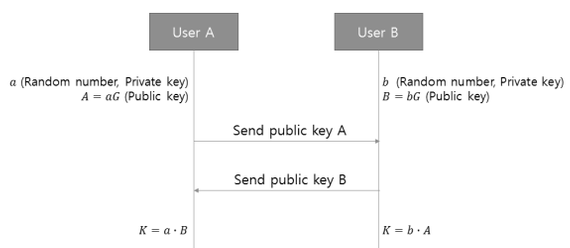

Fig. 6. The Flow Chart for ECDH Algorithm

타원곡선상의 점을 이용하여 자신을 인증하는 전자서명 알고리즘인 ECDSA(Elliptic Curve Digital Signature Algorithm)의 진행 과정은 Fig. 7과 같다. 이때 생성되는 전자 서명 값인 r과 s는 개인키 a를 필요로 하기 때문에 개인키를 소유하고 있는 사용자 외에는 전자서명을 할 수 없다.

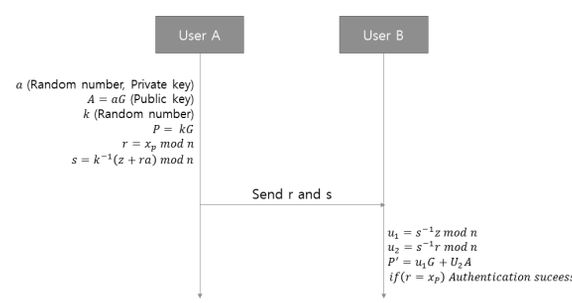

Fig. 7. The Flow Chart for ECDSA Algorithm

Table 3. Definition of symbols used in the proposed method

Symbol	Description
Device A	- Device requesting for certificate duplication
Device B	- Device stored the certificate
a	- Private key of Device A (Integer)
A	- Public key of Device A (Point on an elliptic curve)
b	- Private key of Device B (Integer)
B	- Public key of Device B (Point on an elliptic curve)
G	- A random point on the shared elliptic curve
DHK	- A point on the elliptic curve generated with ECDH algorithm - DHK = aB = bA
k	- Random number (Integer)
v	- A value shared with BLE Advertising packet (Integer)
SK	- Symmetric key of use for certificate encryption
$enc_{key}(m)$	- Function to encrypt the m using a symmetric key algorithm - key: symmetric key
$dec_{key}(m)$	- Function to decrypt the m using a symmetric key algorithm - key: symmetric key
$hash(d)$	- Function to calculate the hash value of the d
h	- Hash value for the data merged CE and v
h'	- Hash value for the data merged CE' and v
CE	- Certificate data
CE'	- Data decrypted from the encrypted certificate
EC	- Date encrypted by the function $enc_{key}(m)$

3. 제안하는 방법

본 논문에서는 TCP와 Bluetooth 통신을 사용한 다중채널 환경에서의 인증서를 안전하게 복사하는 방법을 제안한다. Table 3은 제안하는 방법에서 사용되는 명칭 및 기호이다. 본 논문에서 사용하는 공개키 알고리즘은 ECC를 사용하고, 이때 사용하는 타원곡선 파라메타는 sepc160r1을 사용한다[11]. $hash(d)$에 사용하는 해시함수는 SHA-256이며, $enc_{key}(m)$, $dec_{key}(m)$에 사용하는 대칭키 알고리즘은 AES128을 사용한다[12].

Fig. 8은 제안하는 방법의 전체적인 흐름도이다. 먼저 Device A는 Bluetooth 모듈을 이용하여 BLE Scan을 Enable한 뒤 Device B에게 인증서 복사를 요청한다. Device B는 인증 데이터를 생성한 뒤 BLE Advertise data packet을 일정 시간동안(약 5초) 송신한다. BLE Advertise data packet을 수신한 Device A는 수신한 packet를 이용해 전자서명을 한 뒤 Device B에게 전달하며, Device B는 전자서명 값을 통해 Device A를 확인한 뒤 대칭키를 생성하고 생성된 대칭키를 이용해 인증서를 암호화하여 인증서의 해시 값과 함께 Device A에게 전달한다. Device A는 대칭키를 생성하여 암호화된 인증서를 복호화하며, 인증서 해시 값을 비교하여 일치한다면 인증서를 디바이스에 저장한다. 이 방법은 인증서를 소유한 디바이스에서 생성한 임의의 값을 BLE Advertising data packet을 요청한 디바이스에 송신함으로 값을 공유하고,

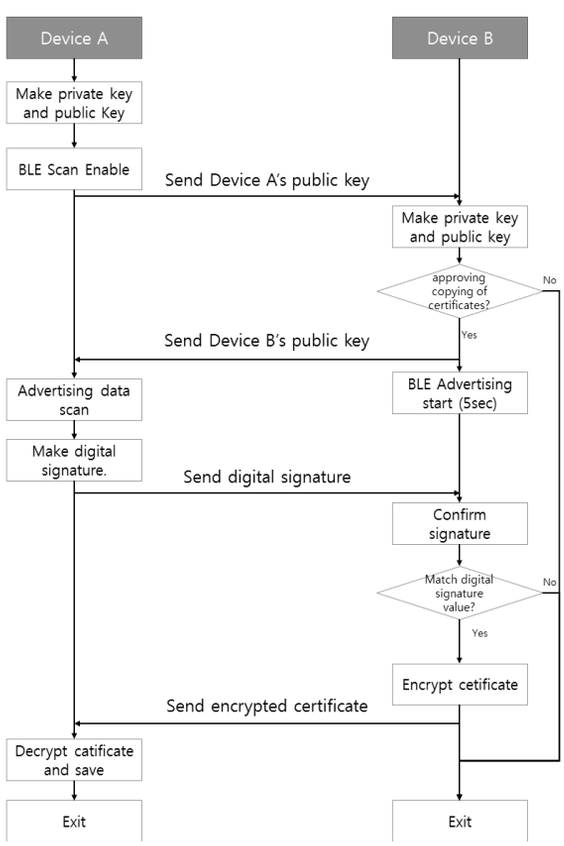

Fig. 8. The Flow Chart of Proposed Method

이 공유한 값을 디바이스 인증 및 대칭키 생성에 사용함으로써 인증서를 안전하게 암호화하여 전달할 수 있다.

3.1 인증서 복사 요청 및 BLE 통신

Fig. 9는 Device A가 Device B에게 인증서 복사를 요청하고 BLE통신을 하는 과정을 나타낸다. 먼저 Device A는 a를 생성하고 이를 이용하여 공개키 A를 생성한다. 이후 디바이스의 BLE Scan을 활성화한 뒤 Device B에게 인증서 복사를 요청하면서 A를 전송한다.

Device B는 인증서 복사 요청을 한 디바이스에게 인증서를 복사할지를 결정하며, 인증서 복사를 승인하게 되면 b를 생성하고 이를 이용하여 B를 생성하고, b와 A를 곱하여 DHK를 계산한다. 이후 Fig. 10과 같은 형식으로 Advertising data를 생성한다. Advertising data는 26byte의 manufacturer data로 구성되고, 데이터의 앞 10byte는 A의 hash 데이터 중 앞 10byte이며, 뒤의 16byte는 임의의 난수 v를 대칭키 암호화 알고리즘으로 암호화한 데이터이다. 이때 대칭키는 DHK를 사용한다. 이후 Device A에게 B를 전송하면서 생성된 Advertising data packet을 일정 시간(약 5초)동안 송신한다. 이때 v를 암호화하는 이유는 BLE Advertising 특성상 주위의 모든 디바이스에게 Advertising data를 전달하기 때문에 타 사용자에게 v가 노출되는 것을 막기 위함이다.

B를 수신한 Device A는 주변에서 scan된 Advertising data 중 manufacturer data의 앞 10byte가 A의 hash 데이터의 앞 10byte와 같은 데이터가 수신되면 BLE Scan을 비활성화

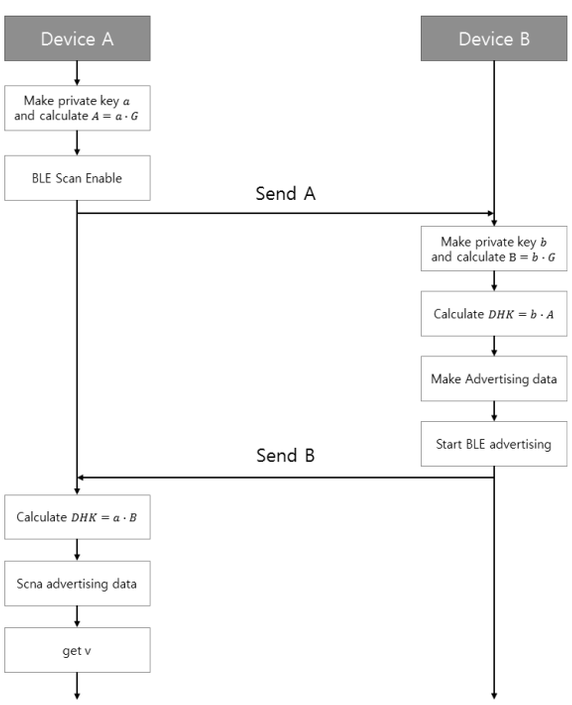

Fig. 9. The Process for the Request of Certificate Duplication and the Exchange of BLE Data

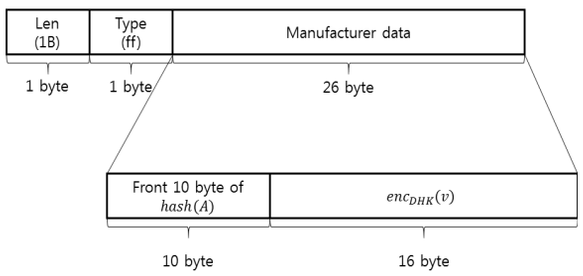

Fig. 10. A Format of Advertising Data used in the Proposed Method

한 뒤 DHK를 계산하여 manufacturer data의 뒤의 16byte를 복호화하여 v를 획득한다.

3.2 디바이스 인증 및 인증서 복사 과정

Device A와 Device B로부터 인증서를 받아와 저장하는 과정은 Fig. 11과 같다. 먼저 Device A는 Equation (3)과 같이 임의의 난수 k를 생성하고 이를 v와 더한 뒤 타원곡선상의 점 G와 곱하여 점 P를 계산한다. 이후 Equation (4)와 (5)를 이용하여 r과 s를 계산한 뒤 Device B에게 r과 s를 전달한다. 이때 s를 계산하기 위해서는 Device A의 개인키가 필요하기 때문에 Device A 외에는 계산할 수 없다.

$$P = p \cdot G = (k+v)G \tag{3}$$

$$r = P_x (P\text{의 } x\text{좌표}) \tag{4}$$

$$s = p^{-1}(v + r \cdot a) \tag{5}$$

r과 s를 수신한 Device B는 Equation (6), (7), (8)을 이용하여 u_1, u_2, P'을 계산한다.

$$u_{1} = s^{-1} \cdot v \tag{6}$$

$$u_{2} = s^{-1} \cdot r \tag{7}$$

$$P' = u_1 \cdot G + u_2 \cdot A \tag{8}$$

계산된 P'의 x좌표가 r과 같지 않다면 Device A를 신뢰할 수 없는 디바이스로 판단하여 인증서 복사를 수행하지 않는다. P'의 x좌표가 r과 같다면 인증서 복사 과정을 수행한다. 먼저 해시함수를 이용하여 DHK와 v을 합친 데이터의 해시 값을 계산하여 대칭키(SK)로 사용하고, 인증서(CE)를 대칭키 함수를 이용하여 암호화하여 EC을 계산한다. 또한 해시함수를 이용하여 CE와 v를 합친 데이터의 해시 값 h를 생성하여 Device A에게 EC와 h을 전달한다. Device A는 EC와 h를 수신하고 해시함수를 이용하여 DHK와 v를 합친 데이터의 해시 값을 대칭키(SK)로 사용하여 EC을 복호화하

여 CE'을 획득하고 CE'와 v을 합친 데이터의 해시 값(h')을 h와 비교하여 올바른 인증서임을 확인하여 CE'를 저장한다.

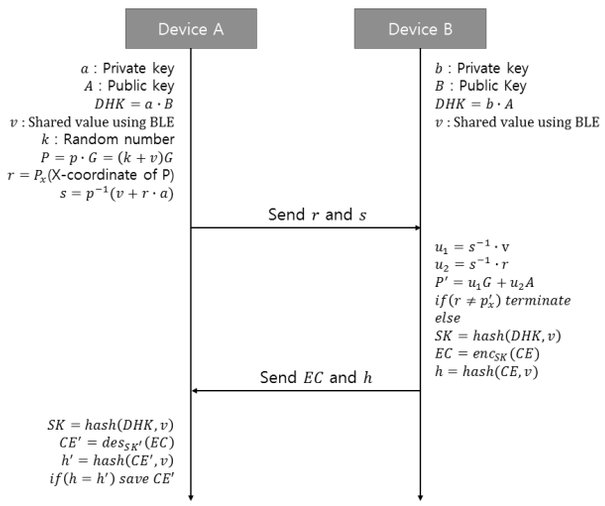

Fig. 11. A Process for Device Authentication and Certificate Duplication on Proposed Method

4. 성능 평가

4.1 시스템 구현

본 논문에서는 안드로이드 디바이스를 이용하여 제안하는 방법을 구현하였다. 구현에 사용한 디바이스는 Galaxy Note FE와 Galaxy S4이며, 디바이스의 사양은 Table 4와 같다.

Table 4. Specifications of the Device used in the Implementation of the Proposed Method

Item	Specification
Device A	- Device : Samsung Galaxy S4 - OS : Android 5.0.1 (Lollipop) - AP : Exynos 5410 SoC ARM Cortex-A15 MP4 1.6 GHz + ARMCortex-A7 MP4 1.2 GHz CPU - RAM : 2GB LPDDR3 SDRAM
Device B	- Device : Samsung Galaxy note FE - OS : Android 7.0 (Nougat) - AP : Exynos 8890 SoC Exynos M1 MP4 2.3 GHz CPU + ARM Cortex-A53 MP4 1.6 GHz CPU - RAM : 4GB LPDDR4 SDRAM

Fig. 12는 Device A, Fig. 13은 Device B의 구현 결과이다. Device A는 개인키 a와 공개키 A를 생성하고 BLE Scan을 활성화한 뒤 Device B에게 공개키 A를 전송하면서 인증서 복사 요청을 하였다. Device B는 공개키 A를 수신하고 Advertising data를 생성하여 BLE Advertising을 시작한 뒤

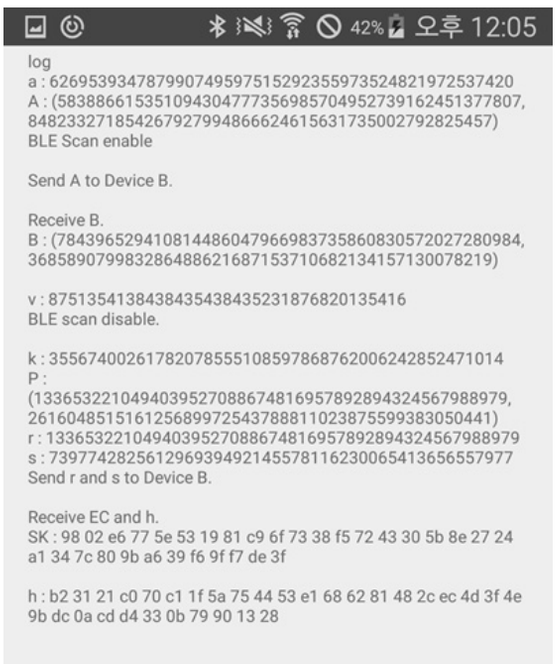

Fig. 12. Result of Device A (Receive Certification)

Fig. 13. Result of Device B (Send Certification)

Device A에게 공개키 B를 전달하였다. 이후 Device B로부터 공개키 B를 수신하고 Advertising data로부터 v를 획득하게 되면 BLE Scan을 비활성화한 뒤 임의의 난수 k를 생성하여 P를 계산하고 이를 이용해 r과 s를 계산하여 Device B에게 전송하였다. Device B는 r과 s를 수신하여 u_1, u_2, P를 계산하여 r과 P의 x좌표를 비교하여 일치함을 확인하고 SK를 계산하고 이를 대칭키로 사용하여 인증서를 암호화하고, 인증서의 해시 값 h을 계산하여 Device A에게 전달하였다. 이후 Device B로부터 EC와 h을 수신하면 SK를 계산

하고 이를 이용하여 EC를 복호화한 뒤 해시 값을 h와 비교하여 일치하는 경우 인증서를 디바이스에 저장하였다. 그 결과 Device A와 Device B는 안전하게 동일한 인증서를 공유할 수 있었다.

제안하는 방법은 기존 방법과 비교하였을 때 인증서를 복사하기까지의 조작이 매우 간단하다. 인증번호를 입력하여 인증서를 복사하는 방법은 화면에 나타나는 8~16자리의 숫자를 다른 디바이스에 입력하여 복사를 수행한다. 12자리의 숫자를 기준으로 임의의 숫자를 입력하였을 때 평균 약 7.6초정도 소요되었다. QR코드를 이용한 인증서 복사 방법은 화면에 나타나는 QR코드를 다른 디바이스의 카메라를 이용해 스캔하여 인증서 복사를 수행하며, 어플리케이션을 통해 카메라로 QR코드 스캔이 완료하였을 때 소요되는 시간이 약 3~4초이다. 제안하는 방법은 생성된 Advertising data를 다른 디바이스에서 스캔하여 인증서 복사를 수행한다. 이때 Device A에서 인증서 복사 요청을 위한 터치 1회, Device B에서 인증서 복사 승인을 위한 터치 1회만을 조작하기 때문에 인증서 복사에 대한 과정이 매우 간편하며, 실제로 인증서 복사를 시도한 결과 약 1~2초의 시간이 소요되었다. Fig. 14는 제안하는 방법을 이용하여 인증서를 복사하기 위한 과정을 나타낸 그림이며, Table 5는 기존의 인증서 복사 방법과 제안하는 방법의 인증서 복사 방법을 나타낸 표이다.

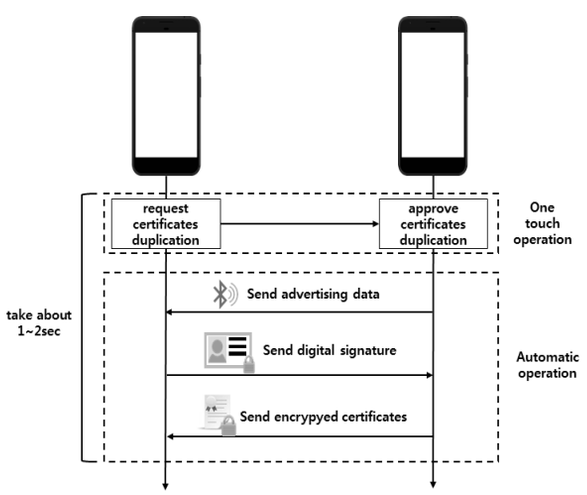

Fig. 14. Certificate Duplication Process of the Proposed Method

4.2 수식 증명

본 논문에서는 Device A와 Device B가 BLE Advertising data를 이용하여 v를 공유하며, 이를 이용하여 디바이스 인증 및 인증서 암호화에 사용하는 대칭키를 생성하였다. 디바이스 인증에 사용되는 서명 생성 식은 3.2절의 Equation (3), (4), (5)와 같고, 서명을 이용한 인증 식은 3.2절의 Equation (6), (7), (8)과 같다. 이때 Equation (8)에서 P'을 생성하는 식은 다음과 같이 도출할 수 있다.

$$\begin{aligned} P' &= u_1 G + u_2 G \\ &= (s^{-1} \cdot v)G + (s^{-1} \cdot r \cdot a)G \\ &= (s^{-1} \cdot v + s^{-1} \cdot r \cdot a))G \\ &= (s^{-1} \cdot (v + r \cdot a))G \\ &= (\frac{p}{(v + r \cdot a)} \cdot (v + r \cdot a))G \\ &= p \cdot G \end{aligned} \tag{9}$$

Equation (9)에서 P'을 계산하는 식인 $u_1 G + u_2 G$에 u_1과 u_2를 계산하는 Equation (6), (7)을 대입하면 $(s^{-1} \cdot v)G + (s^{-1} \cdot r \cdot a)G$와 같이 계산할 수 있고, G의 곱으로 정리하여 $(s^{-1} \cdot v + s^{-1} \cdot r \cdot a))G$와 같이 나타낼 수 있으며, 이는 다시 s^{-1}의 곱으로 정리하여 $(s^{-1} \cdot (v + r \cdot a))G$와 같이 정리할 수 있다. 이때 s의 계산식인 Equation (5)를 대입하여 정리하면 $(v + r \cdot a)$가 소거되어 $p \cdot G$만 남게 된다. 따라서 Device A는 개인키 a를 이용하여 r과 s를 계산하므로 r과 s의 생성은 Device A만 가능하며, Device B는 임의의 난수 k와 Device A의 개인키 a를 모르더라도 r과 s로 P를 계산하여 Device A를 확인할 수 있다.

Table 5. Comparison of the Existing Method and the Proposed Method

	Entering a verification code	Scanning a QR code	The proposed method
Duplication method	Copy the certificate, generating an 8-16 verification code	Copy the certificate, generating a QR code	Copy the certificate, using BLE Advertising packet
Control method	Enter the generated 8-16 digit number in another device	Scan the QR code displayed on the screen, using the camera of another device	Scan the data advertised in a device in another device
Time required	About 7-10 sec.	About 3-4 sec.	About 1-2 sec.
Control	Moderate	Easy	Very easy
Allowable distance	Long distance (Distance in which TCP communication is possible)	In about 30cm (Distance in which the camera can recognize a QR code)	In about 10m (Distance in which advertising data can be received)

4.3 안정성 평가

제안하는 방법은 스니핑 공격으로 인해 TCP를 이용한 통신 데이터가 노출되었을 때 공격자는 인증서를 획득할 수 없다. 인증서는 Diffie-Hellman 알고리즘으로 생성된 DHK와 Advertising data로 공유된 v값을 이용하여 생성된 값을 대칭키로 사용하여 암호화한다. 따라서 공격자가 인증서를 획득하기 위해서는 a또는 b를 알고 있을 때 Device B가 송신하는 Advertising data를 수신하고 TCP통신으로 전달되는 암호화된 인증서를 탈취하여야 한다. a와 b는 개인키이므로 스니핑 공격으로는 얻을 수 없고, A또는 B를 이용하여 a와 b를 유추하는 것 또한 매우 어렵다. 또한 개인키를 획득하였다 하더라도 v가 없으면 인증서를 복호화할 수 없다. v를 획득하기 위해서는 Device B의 기기 근처로 접근하여 Advertising data를 수신하여야 하며, Advertising data를 수신하더라도 v가 암호화되어있기 때문에 v를 획득하기는 매우 어렵다.

[4]는 A사의 Android용 은행 어플리케이션을 자바 디컴파일을 이용하여 어플리케이션에서 사용된 보안 어플리케이션을 분석하고, 스니핑을 통해 얻은 패킷으로 무작위 대입 공격을 하여 인증서를 복호화하는 알고리즘을 구성하였다. [4]에서 분석한 A사의 인증서 복사 프로토콜에 대한 공격 알고리즘을 직접 구현하였으며, 구현 환경은 Table 6과 같다.

Table 6. Environment of the Implementation of an Attack Algorithm

Division	Description
OS	- Windows 7 Enterprise (64bit)
CPU	- Intel core i5-4750 3.20GHz
RAM	- 12GB
Development	- Java jdk 1.8.0_121 (64bit)

구현한 알고리즘을 100회 반복하는 스레드를 100개를 수행한 결과 약 1.5초의 시간이 소요되었다. 즉 10,000회 시도하는데 소요된 시간이 1.5초이며 실험에 사용된 CPU를 이용하여 모든 경우의 수를 대입한다면 약 1,736일이 소요되고, 실제로 공격을 수행한다면 1,736일 이내에 인증서를 복호화할 수 있다. 만약 CPU보다 10배 빠른 GPU가 다수 장착된 PC 10대를 사용하여 공격을 시도한다면 일주일 내에 인증서 복호화가 가능하며, 이는 보안 강도가 매우 낮다고 할 수 있다. 제안하는 방법의 인증서를 복호화하기 위한키를 계산하기 위해서는 개인키 a또는 b와 v를 알아야 한다. 이전 실험과 같은 PC를 이용하여 100개의 스레드에서 100회의 ECC 키 쌍을 생성하였을 때 약 27.3초가 소요되었다. 즉 모든 경우의 수를 대입하여 키 쌍을 생성하였을 때 소요되는 시간은 27.3×2^{160}초이며, 이를 연 단위로 환산하였을 때 $27.3\times2^{160}/60/60/24/365 \fallingdotseq 1.2\times10^{42}$ 년이 소요된다. 이는 높은 성능을 지니고 있는 다수의 PC를 이용하더라도 쉽게 계산할 수 없는 크기이며, 인증번호를 이용한 인증서 복사 방법과 비교하였을 때 보안강도가 약 2.6×10^{41}배 정도 높다고 할 수 있다. 또한 개인키를 계산하더라도 대칭키를 계산하기 위해서는 DHK와 v을 이용한 해시 값을 계산해야 하며, v가 될 수 있는 모든 경우의 수를 대입하여 해시 값을 계산한다면 그 이상의 시간이 소요될 것이다.

5. 결 론

본 논문에서는 두 디바이스간에 인증서 복사를 위해 BLE를 이용하여 인증 코드를 공유하고 이를 이용한 디바이스 인증 및 대칭키 생성을 하여 안전하게 인증서를 암호화하여 전달하는 방법을 제안하였다. 기존의 인증서 복사 방법 중 인증번호를 입력하는 방법은 사용자가 직접 인증번호를 입력하는데 시간이 소요되며, 인증번호 또한 길지 않아 보안이 취약하다는 단점이 있다. 또한 QR코드를 입력하는 방법은 사용자가 카메라를 이용하여 스캔하는데 약간의 시간이 소요되며, 스크린 캡쳐와 같은 악성 프로그램으로 인해 QR코드가 노출될 수 있다는 단점이 존재한다. 제안하는 방법은 BLE Advertising data를 이용하여 임의의 값을 공유하고 이와 ECC 알고리즘을 이용하여 디바이스 인증을 수행하여 BLE Advertising data를 수신한 올바른 디바이스임을 확인한 뒤 암호화된 인증서를 전달하는 방법을 사용하였다. 이 방법은 디바이스 인증에 필요한 데이터를 BLE를 이용하여 공유하기 때문에 조작이 매우 간편하며, 공개키 알고리즘인 ECC를 사용하기 때문에 보안 강도 또한 매우 높으며, 원거리에서 Advertising data를 수신할 수 없기 때문에 스니핑 공격에 안전함을 확인하였다.

References

[1] The Bank of Korea [internet], http://www.bok.or.kr/.

[2] Hye-Seung Park, Jae-Hyup Lee, and Seung-Chul Park, “Implementation, Security, and Usability Analysis of Accredited Certificate-based Internet Banking,” *Journal of Internet Computing and Services*, Vol.18, No.4, pp.69-78, 2017.

[3] Seon-Joo Kim and In-June Joe, “Management Method for Private Key File of PKI using Container ID of USB memory,” *Journal of the Korea Contents Association*, Vol.15, No.10, pp.607-615, 2015.

[4] DongOh Shin, Jeonil Kang, DaeHun Nyang, and KyungHee Lee, “On the Security of Public-Key-Certificate-Relay Protocol for Smart-Phone Banking Services,” *The Journal of Korean Institute of Communications and Information Sciences*, Vol.37, No.9, pp.841-850, 2012.

[5] Ji-Hwan Kim and Suk-Tae Kim, “Certified certificate copy method and system,” KR-10-2014-0012028, 2014.
[6] Jiun-Ren Lin, Timothy Talty, and Ozan K. Tonguz, “On the Potential of Bluetooth Low Energy Technology for Vehicular Applications,” *IEEE Communications Magazine*, Vol.53, No.1, pp.267-275, 2015.
[7] Seon-Jae Jeong and Jae-Hong Yim, “Implementation of the Passenger Positioning Systems using Beacon,” *Journal of the Korea Institute of Information and Communication Engineering*, Vol.20, No.1, pp.153-160, 2016.
[8] Bluetooth Techonology Website [internet], https://www.bluetooth.org/ko-kr/specification/assigned-numbers/generic-access-profile.
[9] Daniel R. L. Brown, “SEC 1 : Elliptic Curve Cryptography,” Certicom, 2009.
[10] Mashrufee Alam, Israt Jahan, Liton Jude Rozario, and Israt Jerin, “A Comparative Study of RSA and ECC and Implementation of ECC on Embedded Systems,” *International Journal of Innovative Research in Advanced Engineering*, Vol.3, No.3, pp.83-93, 2016.
[11] Daniel R. L. Brown, “SEC 2 : Recommended Elliptic Curve Parameters,” Certicom, 2010.
[12] J. Deamen, and V.Rijmen, “Advanced Encryption Standard (AES),” Federal Information Processing Standards Publication 197, 2001.

조 성 환

http://orcid.org/0000-0001-5898-9497
e-mail : chak1212@naver.com
2015년 가천대학교 인터랙티브미디어학과 (학사)
2016년~현 재 가천대학교 IT융합공학과 석사과정
관심분야 : 암호화 알고리즘, 데이터 보안, IoT

한 기 태

http://orcid.org/0000-0002-5905-9424
e-mail : gthan@gachon.ac.kr
1982년 충남대학교 계산통계학과(학사)
1990년 한양대학교 전자계산학과 (공학석사)
2001년 한양대학교 컴퓨터비전(공학박사)
2009년~2010년 University of Texas at Austin, Researching professor
2010년~현 재 가천대학교 컴퓨터공학과 교수
관심분야 : 컴퓨터비전, 영상처리, 스마트객체 응용기술, 모바일 컨텐츠