Journal of the Korea Academia-Industrial cooperation Society
Vol. 12, No. 6 pp. 2801-2806, 2011

DOI : 10.5762/KAIS.2011.12.6.2801

블루투스 환경에서 데이터 전송 시 보안 취약점 분석 및 개선 방안 관련 연구

백종경[1], 박재표[2*]
[1]숭실대학교 대학원 컴퓨터학과, [2]숭실대학교 정보과학대학원

A study of analysis and improvement of security vulnerability in Bluetooth for data transfer

Jong-Kyung Baek[1*] and Jae-Pyo Park[2]
[1]**Department of Computing, Graduate School of Soongsil University**
[2]**Information Science Graduate School of Soongsil University**

요 약 블루투스를 통한 데이터 전송 시 Windows-Kernel-Driver의 Major Function Hooking 방법을 이용하면 PC의 키보드해킹과 같이 응용계층과 장치계층 사이에서 암호화되기 전 평문 데이터를 해킹할 수 있다.
본 논문에서는 블루투스 장치계층에서 데이터 전송 드라이버의 함수를 후킹하여 데이터를 암호화 전송하는 보호모듈을 제안하였다. 또한 제안한 자가보호기법을 적용하여 수정된 보호모듈은 해킹 툴에 의해서 데이터가 노출되지 않도록하였다.
제안한 보호모듈을 실제 구현하여 해킹에 의한 기밀성 보장여부를 확인하였다. 블루투스를 통하여 데이터통신을 하는 장치에 대해 보안을 보장하고, 여러 분야에 활용될 수 있을 것이다.

Abstract During data transmissions via Bluetooth networks, data to be encrypted, or plain text between the application layer and the device layer, can be hacked similar to a key-logger by the major function hooking technique of Windows Kernel Driver. In this paper, we introduce an improved protection module which provides data encryption transmission by modifying the data transmission driver of the Bluetooth device layer, and also suggest a self-protecting scheme which prevents data exposure by various hacking tools. We implement the protection module to verify the confidentiality guarantee. Our protection module which provides data encryption with minimal latency can be expected the widespread utilization in Bluetooth data transmission.

Key Words : Windows-Kernel Hooking, Bluetooth Security, File-Transfer Security, Self-Protection

1. 서론

최근 스마트폰이나 노트북과 USB형 블루투스 장비들이 급격히 확산되면서 사용자들의 블루투스 활용도가 높아지고 있다. 또한 기업들은 스마트폰을 업무에 활용하면서 메일, 연락처, 문서 및 결재 작업을 하고 있다. 문서 및 파일을 전송이 간편한 블루투스 통신을 이용하는 빈도가 증가되고 있다.

기기들 간에 블루투스를 이용하여 정보를 교환 할 때 보안 취약점으로 인해 개인정보와 연락처, 중요문서들의 데이터 유출 위험이 있지만 현재까지 이러한 블루투스 장치의 보안에 대해서는 큰 관심이 없는 상태이다.

본 논문은 블루투스를 이용하여 데이터를 전송 할 때 취약 구간에 대해 분석을 하고, 취약점에 대한 해결방안을 제시하고자 한다.

*교신저자 : 박재표(pjerry@ssu.ac.kr)
접수일 11년 04월 04일 수정일 (1차 11년 04월 26일, 2차 11년 05월 23일) 게재확정일 11년 06월 09일

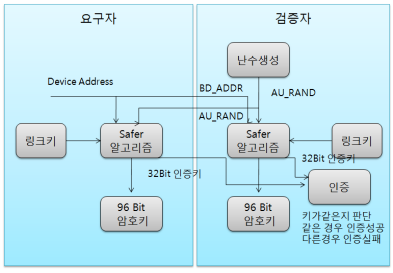
2. 관련연구

2.1 블루투스 보안 기술

블루투스가 기본적으로 제공하고 있는 보안 서비스는 인증(Authentication)과 기밀성(Confidentiality) 그리고 인가(Authorization)의 세 가지이다. 블루투스는 링크계층에서 보안매니저를 통해 블루투스 장치 및 서비스에 대한 제어 권한을 통제하고, 장치에 대해 신뢰된 장치(Trusted Device)와 비 신뢰된 장치(Untrusted Device)를 나눈다. 신뢰 장치는 기 인증된 장치로서 링크키가 저장되어 있고 디바이스 DB에 “Trusted"로 정의된 장치이며, 비 신뢰 장치는 기 인증된 장치로서 링크키가 저장되어 있지만 디바이스 DB에 "Non-Trusted" 로 정의된 장치이다[4].

2.1.1 인증

블루투스 장치 간 인증 절차는 다음 그림 1과 같다.

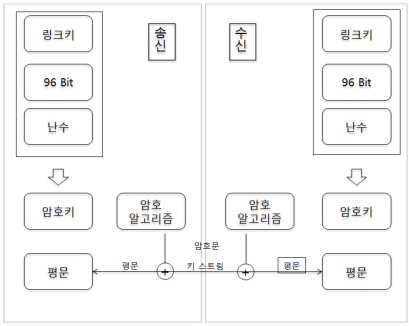


[그림 1] 블루투스 인증 절차
[Fig. 1] Bluetooth authentication process

요구자는 검증자에게 자신을 증명하도록 시도하고, 검증자는 요구자의 증명을 검증하기 위해 링크키를 이용한다. 검증자는 128비트 난수 값을 요구자에게 전송하고, 요구자는 검증자의 난수 값과 블루투스의 주소와 링크키 값을 입력 받아 128 비트의 출력 값을 생성한 후 128 비트 중 32비트만을 인증을 위해 검증자에게 전달한다. 나머지 96비트는 데이터 암호화를 위해 사용한다. 검증자는 동일한 연산을 통해 나온 결과 값과 요구자가 전송한 32비트 인증 값을 비교하여 인증한다[3].

2.1.2 기밀성

블루투스는 상호인증 외에 데이터의 도청을 방지하기 위해 암호화를 제공한다. 블루투스 장치의 암호화 과정은 다음 그림 2와 같다.



[그림 2] 블루투스 암호화 과정
[Fig. 2] Bluetooth encryption process

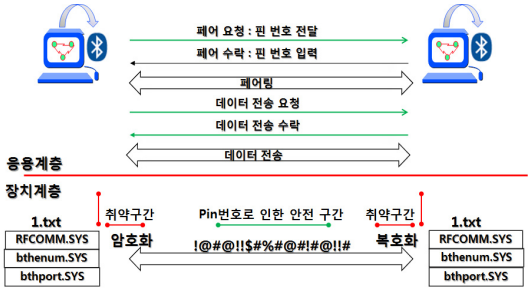
암호키는 키 생성기에서 만들어지는데 키 생성은 링크키와 128비트 난수 값 그리고 Safer 알고리즘에서 나온 96비트 값을 입력으로 암호키가 생성한다. 암호 과정은 LFSR(linear feedback Shift Register) 알고리즘을 사용하여 암호키와 블루투스 주소 값을 입력으로 키 스트림을 생성한 후 데이터와 Exclusive-OR 연산을 통해 암/복호화를 수행한다[3].

2.1.3 인가

블루투스 통신 장치별로 인가 된 서비스만을 제공하며, 인가 받지 못한 서비스는 차단한다.

2.2 블루투스에서의 파일 전송

블루투스에서 파일 전송은 다음 그림 3과 같다.



[그림 3] 블루투스 연결 및 파일 전송
[Fig. 3] Bluetooth Connection and file transfer

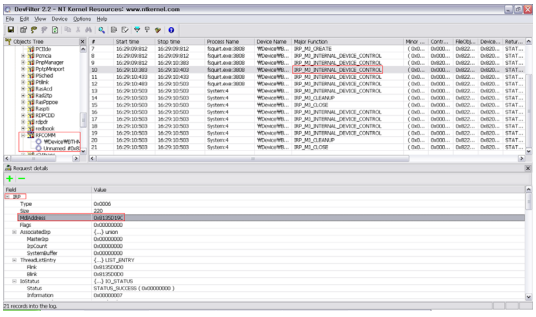
송신장치에서 수신장치로 파일을 전송할 경우 송신장치에서 페어 요청과 함께 핀 번호를 입력한다. 수신장치에서는 페어요청이 들어오면 송신장치에서 입력한 핀 번호와 같은 값을 입력하면 페어링이 이루어지고, 두 장치 간 전송 신뢰성이 보장된다. 페어링이 이루어진 후 송신장치에서 데이터를 전송하면 윈도우 운영체제의 특성상

응용계층에서 장치계층으로 진입하고 윈도우 블루투스 드라이버 경우 RFCOMM 드라이버를 통해서 수신장치로 전송이 된다. 수신장치에서는 장치계층의 RFCOMM 드라이버에서 데이터를 받고 응용계층으로 진입하게 된다. 그러므로 핀 번호에 의해 상호 인증을 한 전송구간의 보안성은 제공되지만 응응계층과 장치계층 그리고 장치에서의 보안은 취약하다.

2.3 블루투스 파일 전송 시 취약점

블루투스는 인증, 암호화를 제공하고, 보안 레벨 4를 적용 할 경우 도청도 어렵다고 기술되어 있으며, 해킹 툴이 발전함에 따라 블루투스 보안레벨도 강화되어 가고 있다. 핀 번호를 사용하여 서로 간 인증을 하고 그 인증을 기반으로 링크키를 생성하여 암호화를 한다고 하지만 그 구간은 서로 데이터가 오고 갈 경우에 보장 된다는 것이다.

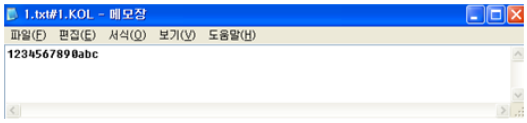
이미 블루투스 기기 간 핀 번호에 의해 상호 인증을 한 전송구간은 안전하다고 판단되었기 때문에 전송되기 전 구간의 취약점을 분석하였다. DevFilter 툴을 사용하여 윈도우 장치계층의 취약점을 분석하기 위하여 스니핑하였고, RFCOMM 드라이버에서 데이터가 전송됨을 발견 할 수 있다. 해당 구간은 블루투스 프로토콜 중 하나인 RFCOMM 프로토콜 구간이다. 그림 4는 해당 툴을 사용하여 블루투스 데이터를 볼 수 있다. IRP의 MdlAddress의 주소를 참조하여 읽어 오면 데이터를 획득할 수 있다.



[그림 4] DevFilter로 보여진 블루투스 데이터
[Fig. 4] Bluetooth data showed by DevFilter

Major-Function-Hooking 기술을 이용하여 RFCOMM 드라이버에서 블루투스 전송 데이터를 분석하여 파일의 내용만을 스니핑할 수 있는 툴을 제작 하였고, 확인 결과 취약 구간이 발견 된다[2].

해당 데이터를 가로 채 분석을 한 후 그림 5처럼 파일로 저장하였다.



[그림 5] 취약구간 데이터
[Fig. 5] Data of vulnerable segment

핀 번호에 의해 상호 인증을 한 후 전송되기 전 블루투스 드라이버에서 데이터를 가져오는 방법을 사용하였다. 블루투스 애플리케이션에서의 전송 작업이 끝나면 장치계층의 RFCOMM 드라이버로 데이터가 이동하게 되는데, 해당 드라이버가 다음 드라이버로 이동하기 전에 데이터내용을 획득하여 파일로 저장하였다.

3. 블루투스 보안 취약점 개선 방안

3.1 취약 구간에 대한 보안 방향

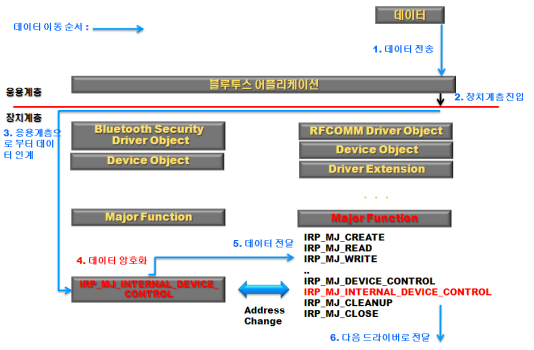
취약 구간을 보호하기 위해 별도의 보호모듈을 개발하였다. 아래에 대한 내용은 개발된 보호모듈로 취약구간을 보안하는 방안이다.

3.1.1 취약 구간 보안 방법

취약 구간을 보안하기 위해서 데이터가 전송될 때 암호화를 하고 수신장치에서는 복호화를 하여 취약구간을 보호하는 방안이다. 해당 취약구간이 암호화가 되면 해커들이 해당 구간을 이용하여 데이터를 스니핑을 하여도 암호화가 되어 있기 때문에 보안 위협으로부터 안전하다.

보호 모듈에서는 암호화 알고리즘을 ARIA 알고리즘을 사용한다. 키 값은 256비트까지 제공하고, 안전성이 검토 된 알고리즘이다. 그러므로 본 논문에서는 ARIA-192비트를 사용한다. 또한 다른 위협 프로세스로부터 해당 Major-Function의 주소가 변경 되었는지 감시하여, 변경 되었으면 사용자에게 알림 메시지를 띄워주고, 다시 보호모듈의 주소로 변경하여 해당 영역을 보호한다.

그림 6은 취약구간을 보호하기 위해 보호 모듈을 삽입하였다.



[그림 6] 보호모듈 계층
[Fig. 6] Layer of protection module

Major-Function-Hook 기술을 이용하여, 해당 구간을 암/복호화하면 데이터는 암호화되어 나가지만 데이터 파일을 받는 수신 측에서는 복호화 되기 때문에 별도의 복호화 행위 없이 안전한 데이터 파일을 받을 수 있다. 단, 보호모듈이 수신장치에 설치가 되어 있지 않은 경우 파일 안의 내용은 암호화가 되어 보인다. 이런 경우 별도의 복호화 툴을 사용하여 복호화한다.

데이터 파일을 유출하는 해킹 툴과 같은 계층에서 동작을 하기 때문에 위에 언급한 것과 같이 Major-Function의 주소 영역을 보안해야 한다.

보호모듈이 암/복호화 하는 시점은 이미 블루투스 장치기기 간 상호인증이 되어 통신하는 구간은 안전하기 때문에 보호모듈과의 인증은 별도로 제공하지 않으며, 취약 구간만을 보호한다.

3.1.2 보호 모듈이 둘 다 존재 할 경우 과정

둘 다 설치가 되어있는 경우 블루투스 핀 번호로 상호인증이 된다. 페어링이 된 상태에서 보호 모듈을 실행 후 이름과 인증키를 입력한 다음 보호모드로 전환한다. 수신장치 또한 송신장치와 동일하게 이름과 인증키를 넣고 보호 모드로 전환 후 파일을 전송하면 된다. 이런 경우 수신측에서 자동으로 암호화가 되고, 송신측에서 또한 복호화가 자동으로 이루어진다.

3.1.3 보호 모듈이 한 곳만 존재 할 경우 과정

수신장치는 보호 모듈이 없기 때문에 보호된 컴퓨터에서 파일을 받게 된 경우 파일이름은 정상이나 파일내용은암 호화 되어 보여 진다. 이런 경우 별도의 복호화 툴을 이용하여 파일을 복호화 할 수 있다. 이름과 인증키를 넣은 후 파일을 선택한 후 복호화를 할 수 있다. 복호화된 파일은 복호화가 잘못될 경우를 방지하여 별도로 생성한다.

3.2 보호 모듈 암호화 프로토콜

3.2.1 암호화키 생성 방안

보호 모듈은 대칭키 알고리즘인 ARIA 192비트를 사용하였다. 입/출력 블록크기는 16비트, 입력키 블록크기는 24비트, 라운드 횟수는 14회이다. 암호화 키 생성방식은 사용자로부터 이름과 인증키 정보를 받아 비밀키를 생성한다. 대칭키를 사용할 경우 키 노출의 위험이 있어 비밀키 값을 보호해야 한다. 비밀키 생성방안은 블루투스 인증시 핀 번호를 입력하는 것과 동일하게 인증키를 입력하고 비밀키를 생성한다. 생성된 비밀키 값을 파일에 삽입하거나 공개키 서버기반이 아닌 수신장치와 송신장치의 인증키를 이용하여 비밀키를 생성하기 때문에 키 노출의 위험이 없다. 인증키가 같다면 동일한 비밀키가 생성되어 암/복호화 되는 방식이고, 송신장치에서 전송할 때 해당 이름의 값을 기반으로 특정 GUID 값을 생성 후 데이터에 삽입 후 전송하게 된다. 수신장치에서는 같은 GUID 값 일 경우만 복호화가 된다.

3.2.2 파일 전송 시 암호화

데이터가 전송될 때 첫 번째 패킷인 경우 버퍼의 맨 앞에 이름 기반으로 생성한 식별코드(GUID)를 삽입하고 버퍼영역만을 암호화 하여 식별코드와 암호화된 버퍼를 함께 전송한다. 두 번째 패킷부터는 버퍼영역만 암호화 하여 전송한다.

3.2.3 파일 수신 시 복호화

첫 번째 패킷을 받을 경우 버퍼의 앞부분이 이름으로 생성한 식별코드인지 확인하고 값이 동일 할 경우 식별코드 값을 제외한 버퍼를 복호화 한 후 파일에 저장한다. 두 번째 패킷부터는 버퍼영역만 복호화 하여 저장한다.

3.3 보호 모듈 자가 보호 방안

보호 모듈이 있다고 해도, 해킹 툴로 인해 보호 모듈이 보호하는 주소 부분을 해킹 툴에서 가져가게 된다면, 파일이 전송될 때 해킹 툴이 먼저 파일 데이터 내용을 가로채기 할 수 있다. 자가 보호하기 위한 방법은 아래와 같다.

3.3.1 보호 모듈의 후킹된 주소 감시

보호 모듈이 후킹을 하고 있는 주소를 미리 저장하고 일정시간마다 해당 주소영역을 감시하고 있다가 주소가 변경되면 복원이 필요하다. 보호 함수 주소가 0x8000000 이였는데, 0x90000000 으로 바뀌었을 경우에는 다른 모듈이 해당 함수를 다시 후킹했다는 것이다. 이런 경우 주

소를 바꾸는 모듈이 서명된 드라이버인지 확인하고 서명된 드라이버가 아닌 경우 보호 함수 주소를 구하고 보호 영역 주소를 복원하여야 한다. 하지만 주소가 계속 변경되는 경우 사용자에게 어떤 모듈이 주소를 바꾸려고 하는지 알림 메시지로 알려준다.

3.3.2 무결성 검증

보호 모듈이 로드가 되면서 자가 검증을 해야 한다. 보호모듈 파일 HASH 값과 메모리상 데이터 영역 HASH 값을 내부에 저장하고 있다가 메모리가 올라간 후 일정 시간마다 보호 모듈 파일의 HASH 값을 구하여 비교하여 같은지 확인 하고, 다를 경우 사용자에게 알림 메시지로 알려주거나 모듈을 재설치 한다.

3.3.3 디버그 감시

드라이버를 동작시키려면 응용계층의 어플리케이션에서 로드, 언 로드, 시작, 중지를 통제한다. 응용계층의 어플리케이션은 사용자 공격에 취약하고, 리버스 엔지니어링 기술을 사용하여 코드 값이 변경이 있는지와 디버그 모드로 접근 했는지를 일정시간마다 감시를 하고 변경이 되었을 경우 사용자에게 알림 메시지로 알려준다.

4. 구현 및 성능 분석

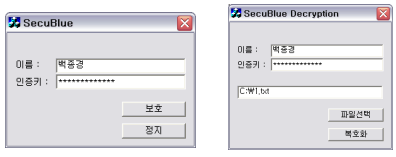
4.1 구현 환경 및 방법

블루투스 파일 전송간 취약구간을 보호하기 위한 시험 환경 운영체제를 Windows XP SP3에서 하였고, 송신장치와 수신장치를 블루투스로 연결하였다.

① 송신장치와 수신장치를 블루투스로 연결하고, 보호 모듈을 설치한다.

② 설치 후 송신장치와 수신장치에 이름과 인증키를 넣고 보호모드로 전환한다.

[그림 7]은 보호모듈 인터페이스이다.



[그림 7] 보호모듈 인터페이스
[Fig. 7] Interface of protection module

이름과 인증키를 넣은 후 보호할 경우 보호 버튼을 누르면 보호모드로 전환이 된다. 수신장치도 동일하다.

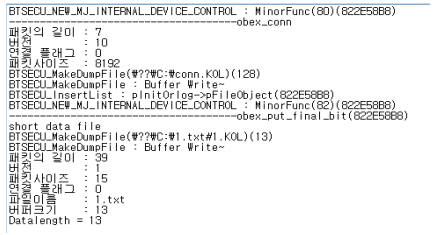
③ 송신장치, 수신장치 모두 보호모드 상황에서, 블루투스 간 페어링을 확인 한 후 데이터전송 한다. 만약 수신장치에 보호모듈이 없다면 복호화 툴을 사용하여 이름과 인증키를 입력한 다음 파일을 선택하고 복호화 버튼을 누른다.

④ 전송이 완료 되면 수신장치에서 제대로 복호화가 되어 있는지 확인한다.

4.2 구현 결과 및 성능 평가

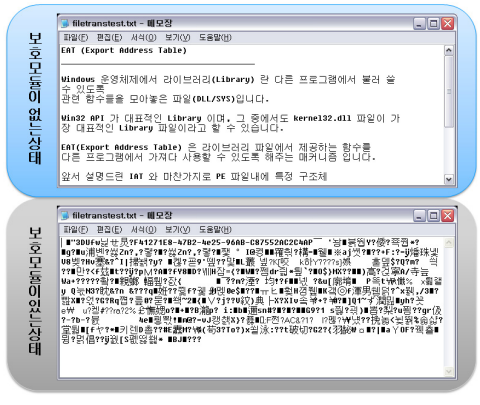
데이터가 전송 될 때 파일의 버퍼만을 암호화 하여 전송 하였으며, 그 결과를 파일로 저장하였다. 암호화가 정상적으로 되어 있고, 식별코드 값이 담겨져 있는 것을 확인 할 수 있다.

그림 8는 데이터가 원격지로 전송이 될 때 연결정보와 데이터 패킷을 분석한 로그이다.



[그림 8] 보호 모듈 데이터 로그
[Fig. 8] Data log of protection module

파일 전송을 하기위해 연결을 시도하였으며, 데이터 패킷에서 길이, 버전, 사이즈 및 파일 이름 등과 같은 정보들이 분석되었음을 알 수 있다.



[그림 9] 보호모듈 사용 전/후 데이터 비교
[Fig. 9] Protection modules before / after comparison of data

그림 9는 전송되는 데이터를 파일로 저장하여 보호모

듈이 없는 경우와 있을 경우를 비교하였다.

보호모듈이 없는 경우는 데이터가 그대로 노출 되는 반면, 있는 경우에는 데이터가 암호화 되어 기밀성이 보일반장됨을 알 수 있다.

일반 전송과의 비교는 표 1과 같다.

일반 전송은 취약구간이 발생되어 기밀성이 보장되지 않고 보안에 취약하다. 취약구간을 암호화하게 되면 기밀성이 높아지며 보안이 강화되고, 암호화를 하여도 장치계층 에서 하기 때문에 시간적으로도 지연이 많지 않음을 알 수 있다.

[표 1] 일반 전송과 암호화 전송 비교
[Tale 1] Compare of general transfer and encryptiom transfer

구 분	일반전송	암 호 화
보 안	낮음	높음
시 간	-	+5.4초 (100M 기준)
기밀성	보통	높음
보호 구간	핀 번호 인증된 구간	핀 번호 인증된 구간 + 장치계층

5. 결 론

본 논문에서는 윈도우즈 운영체제의 커널 계층의 RFCOMM 드라이버에서 Major-Function-Hooking 방법을 이용하여 데이터 전송 할 때 데이터 모니터링이 되는 취약구간을 분석 하였다. 취약구간을 보호하기 위해 보호모듈을 삽입하여 파일 전송 시 송수신장치에서 암/복호화 함으로써 취약구간에 대한 보안성을 높이고, 보호 모듈이 다른 해킹 툴에 의해 취약할 수 있는 부분에 대해 자가 방어 할 수 있는 방안을 제시하였다.

송수신장치에서 사용자 인증을 통해 비밀키를 생성하기 때문에 키 노출에 대한 위험이 없다. 구현을 통해 암호화 전송을 통한 기밀성이 보장됨을 확인하였다.

이는 앞으로도 블루투스를 통하여 데이터통신을 하는 장치에 대해 보안을 보장하고, 여러 분야에 활용될 수 있을 것이다.

Refereances

[1] Walter Oney, “Programming the Microsoft Windows driver model”, Information Publishing Group, 2004.

[2] D. Y. Jeong, “Windows Structures and Principles”, Hanbit Media, 2006.

[3] D. H. Kang, et al., “Bluetooth Security Technology”, NIPA, no. 1308 pp. 1-13, 2009.

[4] Y. Y. Park, “In short-range wireless data communications market, Present and Future of Bluetooth”, KISDI IT FOCUS no. 9 pp. 3-21, 2000.

[5] Y. Y. Park, “3G, WLAN, Comparison and Future of Bluetooth”, KISDI IT FOCUS no. 12 pp. 63-66, 2001.

[6] Y. H. Kim, et al., “Design of a Strong Authentication Mechanism using Public-Key based on Kerberos”, KIISC Vol.12 No.4 pp. 67-76, 2002.

[7] M. H. Kim, et al., “A Design and Analysis of PKCS #11 supporting the KCDSA mechanism”, KIISC no. 16 pp. 141-151 2004.

[8] J. P. Park, et al., “Multi - level encryption scheme for MPEG video source using public encryption algorithm on VOD(video on demand) system”, KICS Conference no. 8 pp. 337-341, 1998.

[9] National Security Research Institute, “Public-Private block encryption algorithms ARIA algorithms spec”, 2004.

박 재 표(Park Jae Pyo) **[정회원]**

- 1998년 8월 : 숭실대학교 대학원 컴퓨터학과 (공학석사)
- 2004년 8월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2008년 9월 ~ 2009년 8월 : 숭실대학교 정보미디어기술연구소 전임연구원
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

<관심분야>
컴퓨터통신, 정보보안, 디지털포렌식, 암호학

백 종 경(Jong-Kyung Baek) **[정회원]**

- 2010년 2월 : 숭실대학교 정보과학대학원 정보보안학과 (공학석사)
- 2011년 3월 ~ 현재 : 숭실대학교 일반대학원 컴퓨터학과(박사과정)

<관심분야>
정보보안, 정보통신, 암호화