DOI: 10.3745/KIPSTC.2010.17C.4.361

PC에 탑재된 OTP의 취약점 분석

홍 우 찬[†] · 이 광 우[††] · 김 승 주[†††] · 원 동 호[††††]

요 약

OTP(One Time Password)란 사용자가 인증시 안전한 메커니즘을 이용하여 매번 다른 패스워드를 생성하여 인증하는 방식을 말한다. OTP 인증 방식을 이용할 경우 공격자는 패스워드를 가로채어 정당한 사용자로 위장할 수 없게 된다. 이러한 OTP는 H/W 기반 또는 S/W 기반 형태로 구현될 수 있다. H/W를 기반으로 하는 단말기형과 카드형의 경우 배포 및 사용의 편의성 문제로 인해 대중화에 어려움이 존재하였다. 이를 대체하기 위한 방법으로 모바일이나 PC에 S/W 형태로 구현하는 OTP가 도입되고 있다. 하지만 S/W 제품은 구현상에 취약점이 존재할 경우 악의적인 공격의 대상이 될 수 있다는 문제점이 있다. 실제로 금융보안연구원의 보고서에서는 모바일 상에 탑재된 OTP의 경우 구현상에 취약점이 존재한다고 밝혔다. 하지만 PC상에 탑재된 OTP에 대해서는 현재까지 취약점 분석 사례가 존재하지 않는다. 이에 본 논문에서는 PC에 탑재된 OTP의 보안 검토사항을 도출하고, 실제 역공학을 통해 OTP 생성 메커니즘을 파악하여 취약점 분석을 수행하였다.

키워드 : OTP, S/W 방식 OTP, PC에 탑재된 OTP, 인증, 역공학

Vulnerabilities Analysis of the OTP Implemented on a PC

Woochan Hong[†] · Kwangwoo Lee[††] · Seungjoo Kim[†††] · Dongho Won[††††]

ABSTRACT

OTP(One Time Password) is a user authentication using secure mechanism to authenticate each other in a way to generate a password, an attacker could intercept the password to masquerade as legitimate users is a way to prevent attacks. The OTP can be implemented as H/W or S/W. Token and card type OTP, implemented as H/W, is difficult to popularize because of having problem with deployment and usability. As a way to replace it implemented as S/W on Mobile or PC is introduced. However, S/W products can be target of malicious attacks if S/W products have vulnerability of implementation. In fact, FSA said the OTP implemented on a mobile have vulnerability of implementation. However, the OTP implemented on a PC have no case about analysis of vulnerability. So, in this paper derive security review and vulnerabilities analysis of implemented on a PC.

Keywords : OTP, S/W OTP, OTP Implemented on a PC, Authentication, Reverse Engineering

1. 서 론

최근 인터넷과 같은 정보통신 기술의 발달로 인해 뱅킹, 공문서 발급, 증권 거래, 인터넷 쇼핑 등의 다양한 서비스들이 온라인을 통해 이루어지고 있다. 기존에 오프라인 상에서 이루어지던 서비스들이 온라인 상에서 이루어짐에 따라 전송되는 정보의 보호을 위해 사용자 인증 기술이 적용되고 있다. 사용자 인증이란, 어떤 사용자가 실제로 정당한 사용자인지를 판단하는 과정으로, 현재 ID와 패스워드를 사용하는 방식이 대표적으로 사용되고 있다. ID/패스워드 방식의 경우 사용자가 ID와 패스워드를 외우기 쉬운 정보 및 고정된 정보로 설정하기에 공격자에 의해서 추측될 수 있고, 도청을 통해 노출될 수도 있다는 단점을 가진다.

이러한 단점을 극복할 수 있는 인증 기법으로, 인증 시도시 새로운 패스워드를 생성하는 일회용 패스워드 기술인 OTP(One Time Password)가 제안되었다. OTP를 통해 생성되는 패스워드는 암호학적으로 안전한 방법으로 생성되며, 한 번 사용한 패스워드는 다시 사용되지 않기 때문에 기존 ID/패스워드 방식의 단점들을 보완할 수 있다. 현재 국내에서 인터넷 뱅킹 등에 이용되는 OTP는 H/W 형태로 구현된 단말기 및 카드 형태의 OTP로서, 금융 거래를 하기 위해서는 별도로 OTP를 소지해야만 하는 불편함으로 인해 대중화가 어렵다는 문제점을 가지고 있었다. 이러한 문제를 해결

※ 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(계약번호 UD100002KD)
본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었습니다(NIPA-2010-(C1090-1031-0005))
† 준 회 원 : 성균관대학교 전자전기컴퓨터공학과 석사과정
†† 준 회 원 : 성균관대학교 전자전기컴퓨터공학과 박사과정
††† 종신회원 : 성균관대학교 정보통신공학부 교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수(교신저자)
논문접수 : 2010년 6월 1일
수 정 일 : 1차 2010년 7월 6일
심사완료 : 2010년 7월 7일

하고자 최근 S/W 형태로 구현된 OTP가 게임 시장을 위주로 도입되고 있다. 모바일에 탑재된 OTP, PC에 탑재된 OTP가 대표적인 예이다. S/W 방식으로 구현되는 OTP의 경우 반영구적인 사용이 가능하고, 배포 과정이 쉬우며, 하나의 OTP로 여러 서비스의 인증을 폭넓게 적용할 수 있는 범용성까지 갖추고 있어, S/W 형태로 탑재된 OTP 도입 사례가 증가하고 있다. 특히 최근 행정안전부가 차세대 통합인증체계 확립을 위해 구축한 전자정부 통합인증 프레임워크 및 통합인증 게이트웨이 사업에서 'GOTP(Goverment One Time password)' 시스템 구축에 모바일 및 PC에 탑재된 OTP를 수용하였다. 하지만, S/W 방식으로 탑재된 OTP에는 구현상에 취약점이 존재할 수 있고, 구현상에 취약점이 존재하게 되면 OTP에 대한 공격이 가능해진다. 실제로 모바일에 탑재된 OTP의 경우 금융보안연구원에서 2009년에 발표한 문서에 따르면 국내에서 활용하고 있는 모바일에 탑재된 OTP의 경우 구현 취약성이 존재한다고 밝히고 있다[1]. 하지만 PC에 탑재된 OTP의 경우 보안 검토사항이 존재하지 않고, 취약점 분석 사례도 존재하지 않는다. 스마트폰에 탑재된 OTP 도입이 이루어 질 경우, PC에 탑재된 OTP의 취약점이 곧 스마트폰 OTP의 취약점으로 이어질 수 있다.

이에 본 논문에서는 역공학을 통해 PC에 탑재된 OTP가 가지는 취약점을 분석하고, 이러한 취약점을 통해 발생 가능한 문제점을 살펴본다. 2장에서는 OTP에 대한 정의 및 구현 방식에 대해서 설명하고, 3장에서는 PC에 탑재된 OTP 보안 검토사항을 도출한다. 4장에서는 실제로 역공학을 통해 PC에 탑재된 OTP의 동작 과정을 분석하고, 도출한 보안 검토사항에 기반하여 취약점을 분석한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 OTP(One Time Password) 및 구현방식에 대하여 언급한다.

2.1 OTP

OTP는 One Time Password의 약자로, 사용자 인증 시 1회만 사용할 수 있는 비밀번호를 생성하는 매체를 의미한다. 또한 동시에 생성되는 1회용 패스워드 자체를 의미하기도 한다[2]. 이는 동일한 패스워드가 반복해서 사용됨으로써 발생할 수 있는 여러 가지 문제를 예방하는 효과가 있다.

2.1.1 OTP 인증

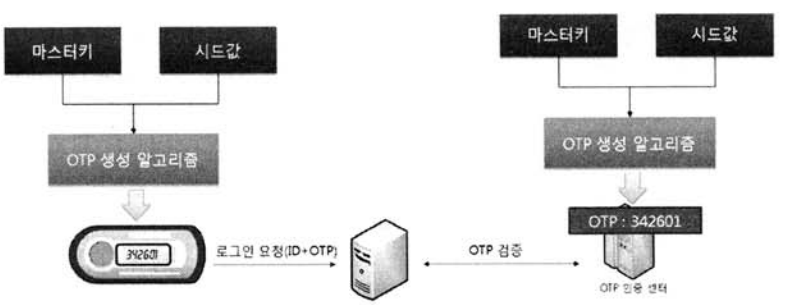
OTP 인증 과정은 (그림 1)과 같다.

사용자와 OTP 인증 센터는 OTP 등록 시 동일한 마스터키를 나누어 가진다. 또한 동일한 시드(seed)를 가지고 있으며, 이러한 시드는 OTP 생성 시 마다 갱신된다. 사용자가 서비스를 이용하기 위해 OTP를 생성하여 로그인 서버에 자신의 ID와 함께 OTP를 전송하게 되면 로그인 서버는 이를 OTP 인증 센터에 전송하여 OTP를 검증하게 된다. OTP 검증 시 OTP 인증 센터는 사전에 나누어 가진 마스터키와 동기화 되는 시드값을 이용하여 OTP 값을 만들어 사용자가 전송한 OTP 값과 비교 검증하게 된다. OTP 생성 알고리즘은 암호학적으로 안전한 방법으로 OTP를 생성하기 때문에, 공격자는 이전에 사용된 OTP를 획득한다 하더라도 추후 생성될 OTP를 유추할 수 없다.

OTP는 동기화 방식과 비동기화 방식으로 나눌 수 있다. 동기화 방식은 OTP 생성 알고리즘의 입력값으로 어떠한 시드를 사용하느냐에 따라 다시 이벤트 동기화 방식, 시간 동기화 방식 그리고 이벤트 동기화 방식과 시간 동기화 방식을 혼용하여 사용하는 조합 방식으로 나눌 수 있다. 비동기화 방식으로는 서버가 제시한 질의 값을 이용하여 응답 값을 서버에 전송함으로써 자신을 인증하는 질의응답 방식이 있다.

2.1.1.1 이벤트 동기화 방식

이벤트 동기화 방식은 OTP 생성 알고리즘의 시드 값으로 카운터(Counter) 값을 사용한다. 초기 카운터 값은 사용자와 서버 모두 0으로 초기화되어 있다. 만약 사용자가 인증을 필요로 하는 경우 사용자가 OTP를 생성하여 서버에 전송한다. OTP가 생성될 때 카운터 값도 1씩 증가한다. 생성된 OTP를 OTP 인증센터에 전송하면 인증센터는 이전의 사용자 인증 기록을 통해 카운터 값을 읽어와 OTP를 생성 및 검증하고 카운터 값을 다시 1씩 증가시킴으로써 카운터 값이 동기화된다. 하지만, 이 방식의 경우 사용자가 OTP를 생성하고, OTP 검증을 요청하지 않으면 카운터 값의 동기화가 이루어지지 않아 인증시 검증에 실패할 수 있다는 단점을 가진다.



(그림 1) OTP 인증 과정

2.1.1.2 시간 동기화 방식

시간 동기화 방식은 시드 값으로 현재의 시간을 사용한다. 인증 센터와 사용자 OTP 장치 간에 동기화된 시간 정보가 입력값으로 사용되어 OTP가 생성된다. 가장 보편적으로 활용되는 방법으로 대부분의 제품에서 이 방법을 활용하고 있다. 하지만, 이 방식의 경우 특정 시간 간격마다 비밀번호가 변하기 때문에, 입력 중에 시간이 흘러 인증센터에서 다른 비밀번호가 생성되어 OTP 검증에 실패할 수 있다는 단점을 가진다. 이를 해결하기 위해 시간 오차 범위를 크게 잡으면 OTP 검증이 실패할 확률은 줄어들지만, 공격 가능성은 커지게 된다. 또한 시간 동기화 방식은 추가적으로 시간 동기화를 위한 알고리즘을 필요로 한다.

2.1.1.3 조합 방식

조합 방식은 시간 동기화 방식과 이벤트 동기화 방식의 단점을 보완하기 위해 두 가지 방법을 조합하여 OTP를 생성하는 방식이다. 이 방식은 OTP 생성 알고리즘의 입력 값으로 시간 정보와 카운트 값을 모두 사용한다. 따라서 조합 방법에서는 OTP가 특정 시간 간격마다 업데이트가 된다. 만약 정해진 시간 간격 안에 다시 재 인증을 시도하는 경우에는 카운트 값을 사용하여, OTP를 갱신한다. 따라서 사용자는 동일한 시간 간격 내에서도 새로운 패스워드를 사용할 수 있다.

2.1.1.4 질의응답 방식

질의응답 방식은 위의 동기화 방식들과 달리 입력값의 동기화가 요구되지 않는다. 사용자가 인증을 요청할 경우 서버는 사용자에게 서버가 제시하는 질의에 대한 응답을 요구하여 정확한 값이 입력되는지를 검증하는 방식이다.

(그림 2)와 같이 사용자가 인증을 요청하면, 서버가 사용자의 휴대폰 문자 메시지로 OTP를 생성하여 전송하고, 이를 다시 제대로 입력하는지를 검증하는 방식이 대표적으로 사용되고 있다.

(그림 2) OTP 인증 과정

2.1.2 OTP 구현 방식

2.1.2.1 H/W 구현 방식

H/W 형태로 탑재된 OTP는 단말기나 카드내부에 OTP 생성 알고리즘을 탑재한 경우이다. 2007년부터 은행 등 금융권에서는 이미 단말기형 OTP를 도입하였고, 최근에는 카드형 OTP의 보급도 추가적으로 이루어지고 있다. 이러한 H/W 형

(그림 3) OTP 단말기형

(그림 4) OTP 카드형

태로 구현된 OTP의 경우 OTP 생성기를 소지해야 하는 불편함과 함께 은행 등에서 유료로 발급받아야 하는 등의 번거로움 때문에 대중화하기 어렵다는 문제점을 가지고 있다.

2.1.2.2 S/W 구현 방식

최근 게임 시장을 중심으로 도입되고 있는 모바일 및 PC에 탑재된 OTP는 S/W 형태로 구현되어 별도의 장치나 비용없이 OTP를 필요로 하는 사용자에게 보급이 가능하다. 또한, 반영구적 사용이 가능하여 3년 주기로 배터리를 교체하기 위해 새로 단말기를 발급받아야 하는 기존 방식의 단점을 해결하였다. 소프트웨어 형태로 제공되기 때문에 배포가 쉽고, 하나의 OTP로 여러 서비스의 인증을 폭넓게 적용할 수 있는 범용성까지 갖추고 있다. S/W 구현 방식은 모바일에 탑재된 OTP, PC에 탑재된 OTP 등이 있는데, 기존의 H/W 형태로 구현된 OTP를 대체하는 인증 도구로 활용될 것으로 예상된다. 현재 S/W 형태로 구현된 OTP는 주요 게임 시장을 중심으로 활성화되고 있다. 또한, 정부기관 및 일반 기업에서도 S/W 구현 방식의 OTP 도입 사례가 늘고 있다. 특히 행정안전부가 차세대 통합인증체계 확립을 위해 구축한 '전자정부 통합인증 프레임워크 및 통합인증 게이트웨이 사업'에서 'GOTP(Goverment One Time Password)' 시스템 구축에 모바일 및 PC에 탑재된 OTP 인증을 수용하였다. GOTP 시스템은 현재 공무원을 대상으로 시범 사업을 실시하고 있고, 향후 국민을 대상으로 한 서비스 도입을 목표로 한다.

하지만, S/W 구현 방식으로 작성된 보안 프로그램의 경우 구현상의 기능적 / 논리적 취약점, 또는 소스코드상의 취약점을 가질 수 있고, 공격자는 이러한 취약점을 이용하여 보안 기능 거부 및 우회 공격을 시도할 수 있다. 따라서 S/W로 구현된 OTP의 경우 H/W로 구현된 OTP에 비해 높은 보안 위협 가능성을 가진다.

3. PC에 탑재된 OTP 보안 검토사항

모바일 OTP의 경우 금융보안연구원에서 2009년에 발표한 '모바일 OTP 보안성 분석서'에서 모바일에 탑재된 OTP의 보안 검토사항을 언급하고, 보안 검토사항에 기반하여 국내에서 제공되는 모바일 OTP에 대해 취약점 분석을 진행한 결과 구현상에 취약점이 존재한다고 밝혔다[1]. 하지만 PC에 탑재된 OTP 같은 경우 보안 검토사항 및 취약점 분석 사례가 존재하지 않는다. 스마트폰상의 OTP 도입이 이루어질 경우 PC에 탑재된 OTP의 취약점이 곧 스마트폰에

〈표 1〉 PC에 탑재된 OTP 보안 검토사항

항목		내용 및 도출 근거	
1. PIN의 무결성 및 기밀성 보장	보안 검토사항 1-1	내용	PIN은 암호화 또는 MAC(Message Authentication Code)으로 저장함
		도출 근거	단순 해시 함수(SHA1, MD5) 이용시 추측 공격에 취약할 수 있음
	보안 검토사항 1-2	내용	PIN의 사용시 복호화된 PIN이 메모리에 존재하는 기간을 최소화함
		도출 근거	복호화된 PIN이 메모리에 존재하는 기간이 길어지면 PIN 정보의 위·변조 가능성도 커짐
	보안 검토사항 1-3	내용	PIN을 사용한 이후에는 PIN을 즉시 메모리에서 삭제함
		도출 근거	PIN을 사용한 이후에는 PIN을 즉시 메모리에서 삭제하여야 메모리 존재 기간을 최소화하여 PIN 정보의 노출 및 위·변조가 어려워짐
	보안 검토사항 1-4	내용	PIN은 인증모듈 외부로 추출되지 않도록 함
		도출 근거	PIN이 추출될 경우 PC OTP 접근 제어 및 올바른 마스터키 추출로 이루어질 수 있음
	보안 검토사항 1-6	내용	PIN 입력 횟수를 제한함
		도출 근거	PIN 입력 횟수가 제한되지 않으면 공격자는 보안 검토사항 1-1과 연계하여 추측 공격을 시도할 수 있음
	보안 검토사항 1-7	내용	PIN이 위·변조되는 경우 OTP의 동작이 불가능하도록 하는 기능
		도출 근거	PIN 정보가 위·변조 되어도 OTP가 정상적으로 작동하면 공격을 통한 접근 및 마스터키 생성이 가능함
2. OTP 생성을 위한 마스터키의 유일성, 무결성 및 기밀성 보장	보안 검토사항 2-1	내용	OTP 생성을 위한 마스터키는 유일하여야 함
		도출 근거	마스터키가 유일하지 않을 경우 타 OTP 프로그램을 이용하여 정당한 OTP 값 생성이 가능함
	보안 검토사항 2-2	내용	복호화된 마스터키는 매번 변경된 위치에 저장함
		도출 근거	복호화된 마스터키가 동일한 메모리 지점에 적재될 경우 추출 가능성이 커짐
	보안 검토사항 2-3	내용	마스터키를 사용한 이후에는 마스터키를 즉시 메모리에서 삭제함
		도출 근거	마스터키를 사용 후 삭제하지 않으면 마스터키의 추출가능성이 증가함
	보안 검토사항 2-4	내용	마스터키는 인증모듈 외부로 추출되지 않도록 함
		도출 근거	마스터키가 추출될 경우 공격자는 이를 이용하여 정당한 OTP 값 생성 가능
	보안 검토사항 2-5	내용	마스터키가 위·변조되는 경우 OTP의 동작이 불가능하도록 함
		도출 근거	마스터키가 위·변조되어도 OTP가 정상적으로 동작할 경우 위·변조를 통한 공격이 가능함
3. 시각 또는 이벤트 정보의 무결성을 보장	보안 검토사항 3-1	내용	사용자가 임의로 시각 또는 이벤트 정보를 변경한 경우 OTP의 동작을 불가능하도록 함
		도출 근거	시각 또는 이벤트 정보를 변경하여도 OTP 동작이 정상적으로 이루어지면 미래 시점의 OTP 값을 추측할 수 있음
4. 안전한 OTP 생성 알고리즘 사용	보안 검토사항 4-1	내용	암호 강도가 높은 대칭키 알고리즘 및 해시 알고리즘을 사용함
		도출 근거	OTP는 표준에 맞게 구현되어야 하고 암호 강도가 높은 대칭키 알고리즘 및 해시 알고리즘을 사용하여야 함
5. 안전한 OTP 추출 알고리즘 사용	보안 검토사항 5-1	내용	OTP 생성 결과는 매번 변경된 위치에서 추출함
		도출 근거	OTP 생성 결과가 매번 동일한 지점에 위치한다면 공격자의 OTP 값 추출 공격이 가능함
	보안 검토사항 5-2	내용	최소 6자리 이상의 OTP값을 출력함
		도출 근거	6자리 이상의 OTP값이어야 추측 공격 등에 안전할 수 있음
6. OTP 프로그램은 역공학이 불가능하도록 설계	보안 검토사항 6-1	내용	PC 환경 및 스마트폰 환경의 경우 역공학 가능성이 충분히 존재함으로 안티디버거, 패킹 등의 안티 리버싱 기술이 적용되어야 함
		도출 근거	PC에 탑재된 OTP 제품은 PC에 설치되는 보안 프로그램으로서 역공학 가능성이 증가함. 역공학을 방지하기 위해서는 심볼정보 제거 기술, 안티 디버거 기술, 패킹, 코드 난독화 기술등의 안티 리버싱 기술의 적용이 필요함.

탑재된 OTP의 취약점으로 이어질 수 있다. 따라서 본 장에서는 금융보안연구원 보고서의 모바일 OTP 보안 검토사항에 기반하여 PC에 탑재된 OTP의 구현 측면의 보안 검토사항을 <표 1>과 같이 도출한다.

4. PC에 탑재된 OTP의 동작 과정 및 취약점 분석

본 장에서는 역공학을 통해 PC에 탑재된 OTP 동작 과정을 분석하고, 앞 장에서 도출한 PC에 탑재된 OTP

보안 검토사항을 만족하는지를 확인하여 취약점 분석을 진행한다.

4.1 PC에 탑재된 OTP의 동작 과정 분석

4.1.1 분석 환경

〈표 2〉 분석 환경

구 분	설 명
CPU	Intel Core2 duo 2.53GHz
RAM	2.00GB
운영체제	Microsoft Window XP Professional Version 2002 Service Pack 2
분석 대상	A사의 PC에 탑재된 OTP 제품
분석 도구	IDA Pro, OllyDbg, Wireshark, Dependency Walker, PEiD, HashCalc
초기 설정 PIN 번호	"1q2w3e"

4.1.2 PC에 탑재된 OTP의 동작 과정

역공학을 통해 PC에 탑재된 OTP의 동작 과정을 분석해 본 결과 (그림 5)와 같이 동작함을 확인할 수 있었다.

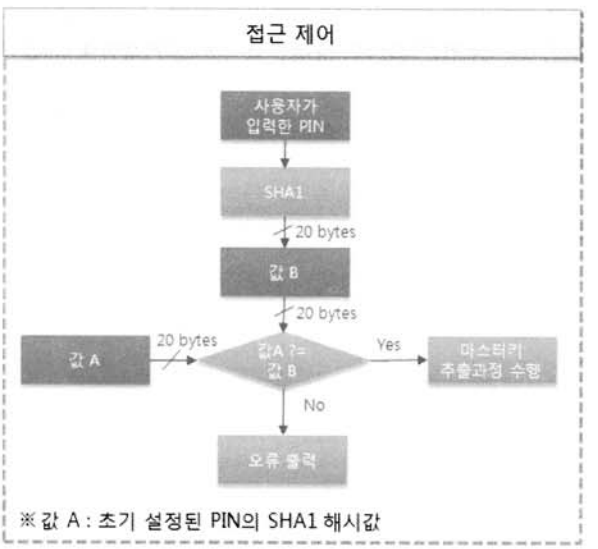
(그림 5) PC에 탑재된 OTP의 동작 과정

사용자가 개인 식별을 위해 PIN 번호를 입력하게 되면 접근 제어를 통해 입력한 PIN 번호와 초기 등록한 PIN 번호가 동일한지 확인한다. 동일할 경우 마스터키 추출 과정을 거친다. 이후, 시스템 시간 정보와 추출한 마스터키를 OTP 생성 알고리즘에 입력하여 OTP를 생성하게 된다. 접근제어, 마스터키 추출, OTP 생성 알고리즘에 대한 세부 분석은 다음과 같다.

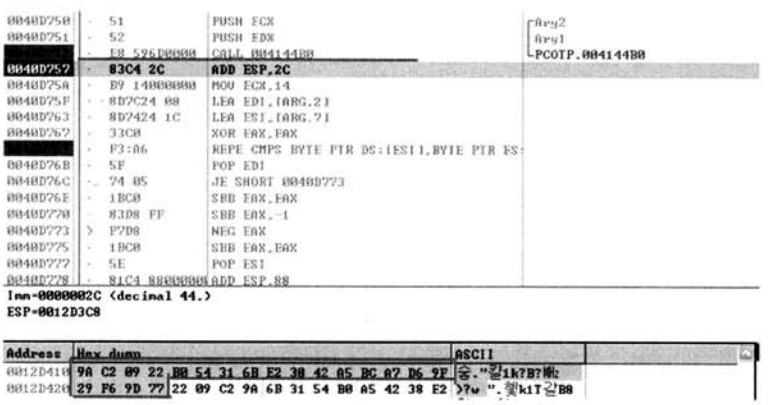
4.1.2.1 접근 제어

접근 제어는 (그림 6)과 같이 PIN 번호의 검증을 통해 이루어진다. PC에 탑재된 OTP를 사용하기 위해서는 사용자는 본인 확인을 위해 PIN을 입력하여야 한다. 사용자가 입력한 PIN은 함수의 입력값으로 사용되고 (그림 7)과 같이 20바이트 값이 생성되게 된다.

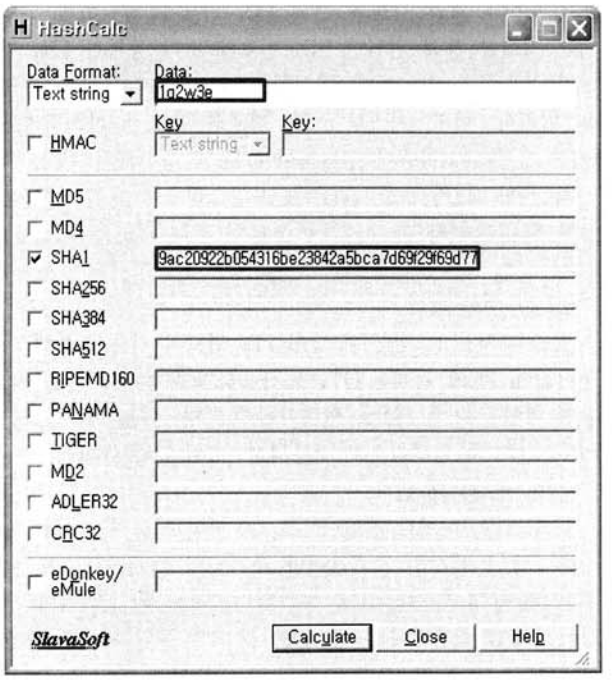
값 B는 20 바이트 길이이므로, 20 바이트 출력 길이를 가지는 SHA1 해시의 결과값이라 추측되어 OpenSSL으로 탑재된 해시 함수 연산 프로그램에 "1q2w3e"를 입력하여 생성된 SHA1 결과값과 (그림 7)의 값 B의 동일 여부를 확인하였다. 확인 결과 (그림 8)과 같이 동일함을 확인할 수 있



(그림 6) 접근제어



(그림 7) 값 B

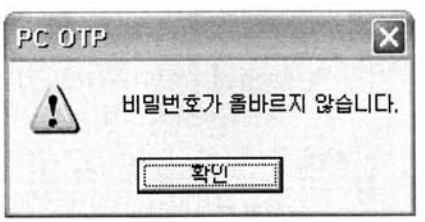


(그림 8) "1q2w3e"에 대한 SHA1 해시값

었다. 즉, 값 B는 PIN을 SHA1 해시한 결과값임을 확인 할 수 있었다.

이렇게 생성된 값 B는 이후 (그림 9)와 같이 초기 설정된 PIN을 해시한 값과 동일한지 확인하는 루틴을 거친다.

만약 초기 설정된 PIN을 해시한 값 A와 사용자가 입력한 PIN을 해시한 값 B가 동일하지 않을 경우 (그림 10)과 같이 오류 메시지를 출력하고, 재입력을 요구하게 된다.



(그림 10) PIN 입력 오류 메시지

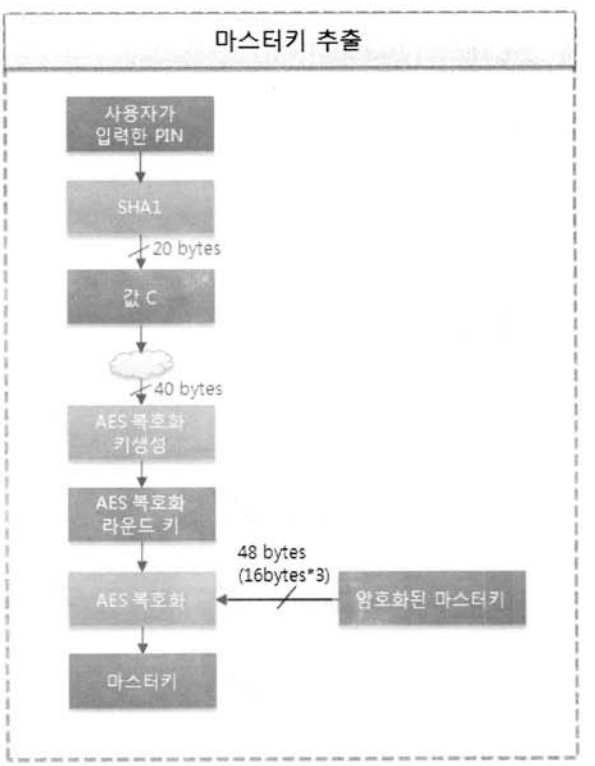
4.1.2.2 마스터키 추출

값 A와 값 B가 동일한 경우, PC에 탑재된 OTP는 암호화된 마스터키를 복호화하는 과정을 수행한다.

마스터키 추출 과정은 (그림 11)과 같다. (그림 9)의 접근 제어 부분과 마찬가지로 사용자가 입력한 PIN은 다시 SHA1 해시를 거쳐 값 C를 생성하게 된다. 이때 생성된 값 C는 값 B가 복사 되는 것이 아니라 사용자가 입력한 PIN을 다시 SHA1 해시하여 생성된 값이다. 이후 일정한 연산을 거쳐 SHA1 해시 값 C는 (그림 12)와 같이 40 바이트 값이 되어 AES 복호화 라운드키 입력값으로 사용된다.

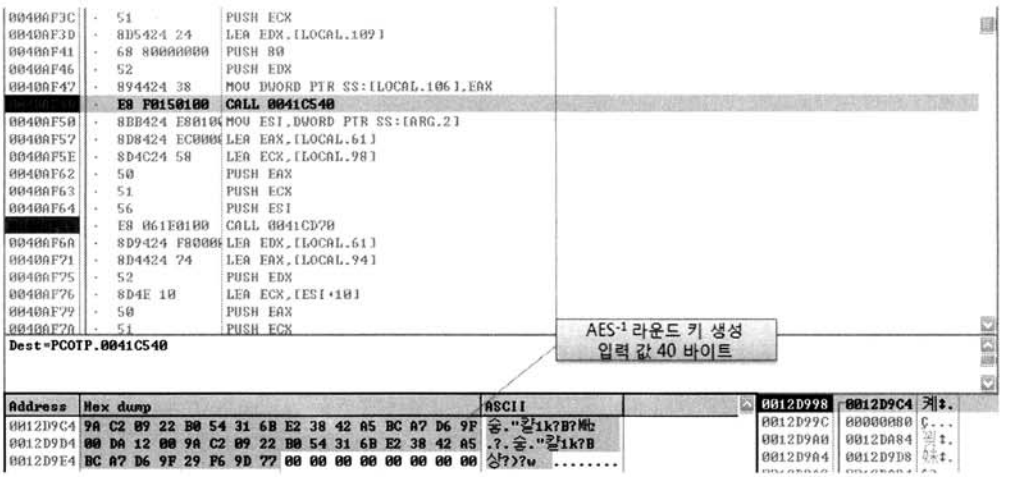
이후, 생성된 AES 복호화 라운드 키를 이용하여 암호화된 마스터 키 값을 복호화한다.

(그림 13)의 암호화된 마스터키 정보는 (그림 14)와 같이 OTP 프로그램 설치 시 생성된 파일에 저장된 48 바이트 값이다.

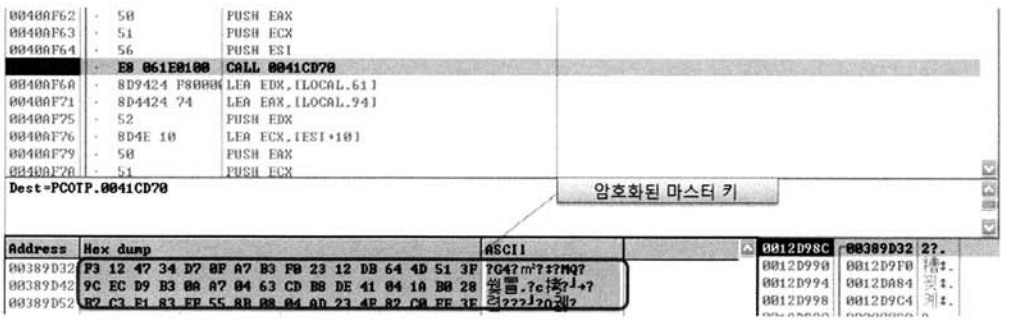


(그림 11) 마스터키 추출

(그림 9) 값 A와 값 B 비교 부분



(그림 12) AES 복호화 라운드 키 생성 40 바이트 입력값

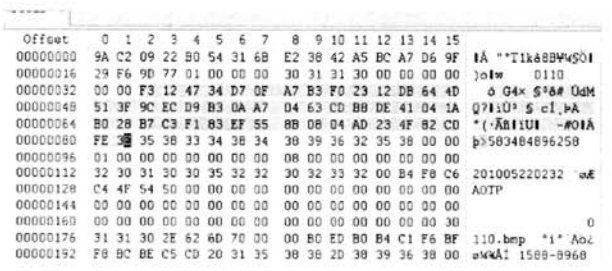


(그림 13) 암호화된 마스터 키 정보

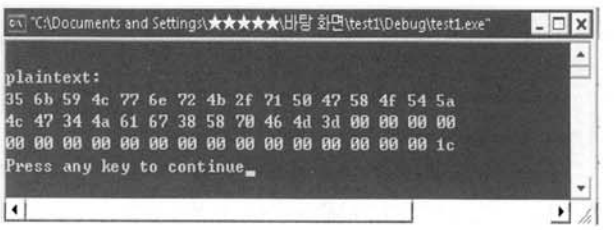
이후, AES 복호화를 거쳐 (그림 15)와 같이 마스터키 정보를 복호화한다.

(그림 12)~(그림 15)의 역공학 과정이 올바른지 확인하기 위해 <표 3>과 같이 위 과정을 검증해보았다.

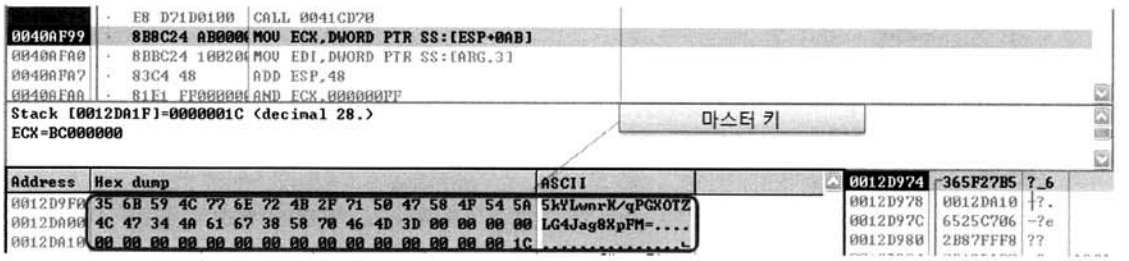
검증 결과 위의 역공학 과정을 거쳐 생성한 마스터키 정보와 (그림 16)과 같이 코딩을 통해 생성한 마스터키 정보가 일치함을 확인할 수 있었다.



(그림 14) 설치 시 생성된 파일에 저장되어 있는 암호화된 마스터키



(그림 16) 코딩을 통해 생성한 마스터키



(그림 15) 복호화된 마스터 키

〈표 3〉 마스터키 복호화 역공학 과정 검증

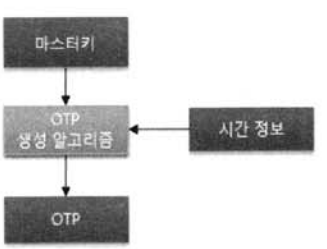
```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "aes.h"
#pragma comment(lib, "libeay32.lib")
#pragma comment(lib, "ssleay32.lib")

void main()
{
        int i;
        AES_KEY dkey;
        //AES 복호화 키
        unsigned char key[40]={0x9A,0xC2,0x09,0x22,0xB0,0x54,0x31,0x6B,0xE2,0x38,
                               0x42,0xA5,0xBC,0xA7,0xD6,0x9F,0x00,0xD1,0x12,0x00,
                               0x9A,0xC2,0x09,0x22,0xB0,0x54,0x31,0x6B,0xE2,0x38,
                               0x42,0xA5,0xBC,0xA7,0xD6,0x9F,0x29,0xF6,0x9D,0x77};

        //암호화된 마스터키 정보
        unsigned char ciphertext1[16]={0xF3, 0x12, 0x47, 0x34, 0xD7, 0x0F, 0xA7, 0xB3,
                                       0xF0, 0x23, 0x12, 0xDB, 0x64, 0x4D, 0x51, 0x3F};
        unsigned char ciphertext2[16]={0x9C, 0xEC, 0xD9, 0xB3, 0x0A, 0xA7, 0x04, 0x63,
                                       0xCD, 0xB8, 0xDE, 0x41, 0x04, 0x1A, 0xB0, 0x28};
        unsigned char ciphertext3[16]={0xB7, 0xC3, 0xF1, 0x83, 0xEF, 0x55, 0x8B, 0x08,
                                       0x04, 0xAD, 0x23, 0x4F, 0x82, 0xC0, 0xFE, 0x3E};
        //마스터키 저장 위치
        unsigned char plaintext[16];


        /* AES 복호화 라운드 키 생성*/
        AES_set_decrypt_key(key,128,&dkey);

        /* AES 복호화(ciphertext1)*/
        AES_decrypt(ciphertext1, plaintext,&dkey);
        printf("\nplaintext:\n");
        for( i=0 ; i<16 ; i++)
                printf("%02x ",plaintext[i]);
        printf("\n");
        /* AES 복호화(ciphertext2)*/
        AES_decrypt(ciphertext2, plaintext,&dkey);
        for( i=0 ; i<16 ; i++)
                printf("%02x ",plaintext[i]);
        printf("\n");
        /* AES 복호화(ciphertext3)*/
        AES_decrypt(ciphertext3, plaintext,&dkey);
        for( i=0 ; i<16 ; i++)
                printf("%02x ",plaintext[i]);
        printf("\n");
}
```

4.1.2.3 OTP 생성 알고리즘

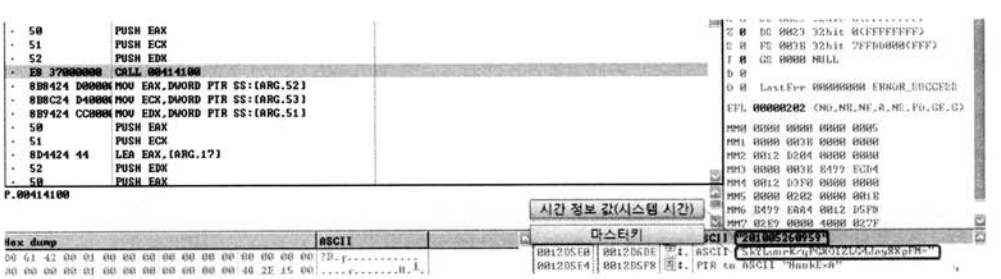
이후, OTP 생성 과정은 (그림 17)과 같다. (그림 18)과 같이 앞선 과정에서 추출한 마스터키와 시스템 시간이 OTP 생성 알고리즘 함수로 입력된다. 시간 내에 재 인증 요구 시에는 시간 정보대신 이벤트 값이 입력된다.

OTP 생성 알고리즘 함수 내부를 살펴보면 (그림 19)와 같이 스택에 8자리의 숫자가 나타난다.

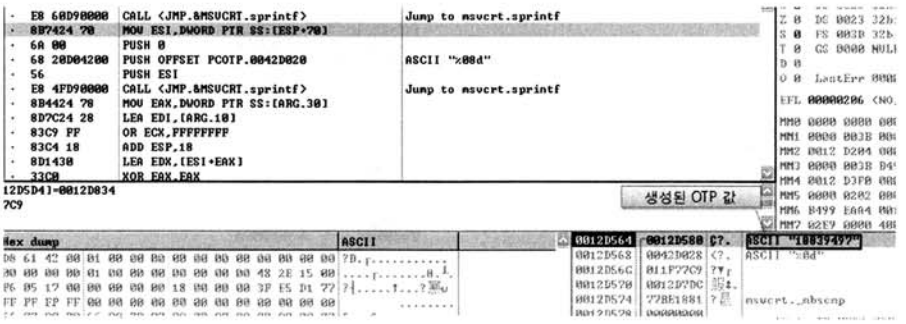
이후, 실제 프로그램을 살펴보면 OTP 생성 알고리즘 내부에서 확인한 숫자가 OTP 값으로 사용자에게 (그림 20)과 같이 제공됨을 확인할 수 있다.



(그림 17) OTP 생성 알고리즘



(그림 18) OTP 생성 알고리즘 입력값



(그림 19) 역공학을 통해 확인한 OTP 값

18839497

(그림 20) 생성된 OTP 값

4.2. PC에 탑재된 OTP 제품의 구현 취약점 분석

PC에 탑재된 OTP의 동작 과정 분석을 통해 3장에서 도출한 PC에 탑재된 OTP의 보안 검토사항 만족 여부를 확인한 결과 <표 4>와 같았다. <표 4>와 같이 보안 검토사항 1-1, 보안 검토사항 1-5, 보안 검토사항 1-6, 보안 검토사항 2-4, 보안 검토사항 2-5, 보안 검토사항 3-1, 보안 검토사항 6-1에 취약하였다. 취약한 내용에 대해서 살펴보면 아래와 같다.

· 보안 검토사항 1-1

동작 과정 분석 결과 PIN은 SHA1 해시 후 저장되어 있음을 확인하였다. 단순 해시만 수행할 경우, 추측 공격에 취약하다.

· 보안 검토사항 1-5

PIN 번호의 입력 횟수 제한 기능을 제공하지 않으므로 공격자는 보안 검토사항 1-1과 연계하여 추측 공격을 시도할 수 있다.

· 보안 검토사항 1-6

초기 설정된 PIN 값은 동작 과정에서 사용되지 않지만, PIN을 SHA1 해시한 값 A가 사용자와 입력한 PIN와 비교하는 과정을 거치게 된다. 이때 값 A를 복사하여 값 B에 복사하면 접근제어 부분을 우회할 수 있고, 추가로 값 C에 복사하게 되면 마스터키 복호화가 가능하여, 올바른 OTP 생성이 가능하다. 즉, 탑재된 PC에 탑재된 OTP 동작 과정의 특성상 PIN 번호의 위·변조가 아닌 PIN의 SHA1 해시 값을 통한 위·변조가 가능하였다.

〈표 4〉 PC에 탑재된 OTP 보안 검토사항 만족 여부

분류		보안 검토사항 만족 여부
1. PIN의 무결성 및 기밀성 보장	보안 검토사항 1-1	X
	보안 검토사항 1-2	O
	보안 검토사항 1-3	O
	보안 검토사항 1-4	O
	보안 검토사항 1-5	X
	보안 검토사항 1-6	X
2. OTP 생성을 위한 마스터키의 유일성, 무결성 및 기밀성 보장	보안 검토사항 2-1	O
	보안 검토사항 2-2	O
	보안 검토사항 2-3	O
	보안 검토사항 2-4	X
	보안 검토사항 2-5	X
3. 시각 또는 이벤트 정보의 무결성을 보장	보안 검토사항 3-1	X
4. 안전한 OTP 생성 알고리즘 사용	보안 검토사항 4-1	O
5. 안전한 OTP 추출 알고리즘 사용	보안 검토사항 5-1	O
	보안 검토사항 5-2	O
6. OTP 프로그램은 역공학이 불가능하도록 설계	보안 검토사항 6-1	X

· 보안 검토사항 2-4

복호화된 마스터키는 (그림 15)와 같이 동작 과정의 역분석을 통해 확인이 가능하다. 확인된 값을 추출하거나 (그림 14)와 같이 OTP 프로그램 설치 시 생성된 파일에 저장된 암호화된 마스터키 정보를 PIN의 SHA1 해시값을 이용하여 복호화 가능하였다.

· 보안 검토사항 2-5

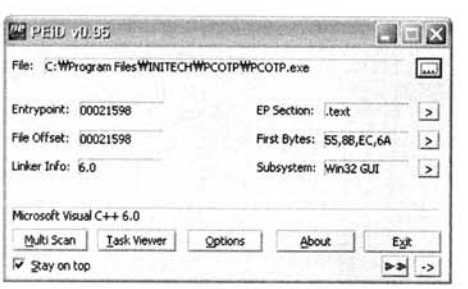
공격자는 (그림 18)과 같이 OTP 생성 알고리즘 입력 값으로 사용되는 마스터키 값을 위·변조하여 원하는 OTP 값을 생성할 수 있다. 마스터키를 위·변조하더라도 OTP는 정상적인 동작을 수행하였다.

· 보안 검토사항 3-1

공격자는 (그림 18)과 같이 OTP 생성 알고리즘 입력값으로 사용되는 시간 정보 값을 위·변조하여 미래 시점의 OTP 값을 미리 확인할 수 있다. 시간 정보를 위·변조하더라도 OTP는 정상적인 동작을 수행하였다.

· 보안 검토사항 6-1

PC에 탑재된 OTP 제품의 패킹 여부를 확인한 결과 (그림 21)과 같이 패킹되어 있지 않음을 확인하였다. 또한, 역공학을 진행하는 동안 안티 디버거 기술, 코드 난독화 기술등의 안티리버싱 기술이 확인되지 않았다. 안티리버싱 기술 우회 시도 없이도 역공학을 통해 충분한 정보를 얻을 수 있었다.



(그림 21) PC OTP 실행 프로그램 패킹 여부 확인

5. 결 론

본 논문에서는 PC에 S/W 방식으로 구현된 OTP의 보안 검토사항을 도출하였고, 실제 역공학을 통해 PC에 탑재된 OTP 제품의 동작 과정을 파악하여 도출한 보안 검토사항 만족여부를 확인하였다. 분석한 PC에 탑재된 OTP 제품은 역공학을 통해 분석한 결과 마스터키, 시간 정보가 노출되었고, PIN · 마스터키 · 시간 정보의 위 · 변조가 가능하였다. 이러한 점을 공격자가 악용할 경우 정당한 OTP 생성이 가능하고, 미래 시점의 OTP 값도 미리 얻을 수 있게 된다.

현재 보급되는 S/W 방식의 OTP 제품들의 경우 하나의 OTP로 여러 서비스를 범용적으로 이용할 수 있다. PC에 탑재된 OTP 제품에 취약점이 존재하면 공격자는 PC에 탑재된 OTP 제품을 공격하여 정당한 사용자로 위장해 다양한 서비스의 이용이 가능하기에 그 피해는 더욱 더 커진다고 할 수 있다.

이러한 피해를 방지하기 위해서는 PC에 탑재된 OTP 생성 프로그램 구현 시 기능적, 논리적 취약점을 없애기 위해 본 논문에서 도출한 보안 검토사항의 적용이 요구된다. 특히, 공격자가 OTP 프로그램을 역공학하여 보안 기능 우회 공격 등을 쉽게 수행 하지 못하도록 반드시 역공학 방지 기술인 안티 리버싱 기술을 적용해야 할 것이다.

참 고 문 헌

[1] 금융보안연구원, "모바일 OTP 보안성 분석서", FSA.TS4.MOS v1.0, 2009.

[2] 금융보안연구원, "OTP 표준화 체계", FSA.TS4.FOS v0.9, 2007.

[3] N.Haller, C.Metz, P.Nesser, M.Straw, "A One-Time Password System", RFC 2289, IETF, 1998

[4] Eldat Eilam, "Reversing : Secrets of Reverse Engineering", John Wiley&Sons, 2005.

[5] 최동현, 김승주, 원동호, "일회용 패스워드(OTP: One-Time Password) 기술 분석 및 표준화 동향", 정보보호학회지 제17권 제3호, pp.12-17, 2007.

[6] 신동휘, 최윤성, 박상준, 김승주, 원동호, "네이트온 메신저의 사용자 인증 메커니즘에 대한 취약점 분석", 정보보호학회논문지 제17권 1호. pp.7-80, 2007.

[7] OATH, http://www.openauthentication.org

[8] RSA, http://www.rsa.com

홍 우 찬

e-mail : wchong@security.re.kr

2009년 성균관대학교 컴퓨터공학과(학사)

2009년~현 재 성균관대학교 전자전기컴퓨터공학과 석사과정

관심분야 : 역공학, 금융 보안, 모바일 보안, 포렌식, 보안성평가

이 광 우

e-mail : kwlee@security.re.kr

2005년 성균관대학교 정보통신공학부(학사)

2007년 성균관대학교 컴퓨터공학과(석사)

2007년~현 재 성균관대학교 전자전기컴퓨터공학과 박사과정

관심분야 : 보안성평가, 전자투표, 포렌식, 디지털 복합기 보안

김 승 주

e-mail : skim@security.re.kr

1994년~1999년 성균관대학교 정보공학과(학사, 석사, 박사)

1998년~2004년 한국정보보호진흥원(KISA) 팀장

2004년~현 재 성균관대학교 정보통신공학부 교수

2001년~현 재 한국정보보호학회, 한국인터넷정보학회, 한국정보과학회, 한국정보처리학회 논문지 및 학회지 편집위원

2002년~현 재 한국정보통신기술협회(TTA) IT 국제표준화 전문가

2005년~현 재 교육인적자원부 유해정보차단 자문위원, 디지털콘텐츠유통협의체 보호기술워킹그룹 그룹장

2007년~현 재 대검찰청 디지털수사 자문위원, KISA VoIP보안기술 자문위원, 기술보증기금 외부 자문위원, 전자정부서비스보안위원회 사이버침해사고대응 실무위원회 위원

관심분야 : 암호이론, 정보보호표준, 정보보호제품 및 스마트카드 보안성 평가, PET

원 동 호

e-mail : dhwon@security.re.kr

1976년~1988년 성균관대학교 전자공학과(학사, 석사, 박사)

1978년~1980년 한국전자통신연구원 전임연구원

1985년~1986년 일본 동경공업대 객원연구원

1988년~2003년 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장

1996년~1998년 국무총리실 정보화추진위원회 자문위원

2002년~2003년 한국정보보호학회장

2002년~현 재 대검찰청 컴퓨터범죄수사 자문위원, 감사원 IT 감사 자문위원

2007년~현 재 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정보통신부지정 정보보호인증기술연구센터 센터장, 정보통신대학원장, 성균관대학교 BK21 사업단장

관심분야 : 암호이론, 정보이론, 정보보호