특정 트랜잭션용 추가 인증을 제공하는 휴대폰 상의 일회용 암호 생성기 설계

(Design of A One-time Password Generator on A Mobile Phone Providing An Additional Authentication for A Particular Transaction)

박 준 철 †

(Jun-Cheol Park)

요 약 일회용 암호는 한 번 사용하고 버리는 특성으로 인해 동일한 암호를 반복 사용하는 기존의 방법에 비해 훨씬 안전하다. 본 논문에서는 사용자들이 늘 휴대하고 다니는 휴대폰 상에서 동작하는 챌린지-응답(challenge-response) 방식의 일회용 암호 생성기를 제안한다. 이 암호 생성기는 추가의 인증 수단을 제공함으로써 PC 인터넷 뱅킹에서 계좌 이체 트랜잭션 명령을 내릴 때 사용할 수 있다. 현재 인터넷 뱅킹용으로 사용되는 별도의 일회용 암호 생성 장치는 30초 시간 동기화를 통해 계속 새로운 암호를 생성하기 때문에 동기화 주기 이내에는 여전히 중간자 공격의 가능성이 남아있다. 이에 비해 제안 방법은 정교한 챌린지-응답 방식을 통해 중간자 공격에 대비할 수 있고, 휴대성이 뛰어나며, 도난의 경우에도 더 안전하다. 또한 현재 쓰이는 장치와 마찬가지로, PC에 설치된 키보드 로거 등의 스파이웨어를 통해 공격 대상자의 모든 다른 인증 정보가 노출되더라도 제안 방법은 불법 이체 행위를 막을 수 있다.

키워드 : 일회용 암호, 휴대폰, 중간자 공격, 인터넷 뱅킹

Abstract One-time passwords are used just once and discarded, which makes it more secure than the repeatedly used conventional passwords. This paper proposes a challenge-response based one-time password generator on a user's mobile phone always carried with the user. The generator can provide an additional authentication for a user to issue a money transfer request within his Internet banking session on a PC. A currently used device for Internet banking generates a password that changes every 30 seconds or so, which allows a man-in-the-middle to use it for stealing money within the 30 seconds. Unlike such a device, the proposed generator resists against the man-in-the-middle attack by a novel challenge-response scheme, provides better accessability and protection against stolen devices. As the currently used devices do, it prevents any unauthorized transfer even if the victim's all other credentials are revealed through his PC infected with spyware such as a keyboard logger.

Key words : OTP, Mobile Phone, Man-In-The-Middle Attack, Internet Banking

1. 서 론

최근 보고된 전형적 인터넷 뱅킹 해킹 방법은 특정 사용자의 PC에 키보드 로거나 스크린 그래버 같은 스파이웨어를 몰래 설치하고, 오랜 기간에 걸쳐 사용자의 인터넷 뱅킹용 비밀 정보를 수집하고, 이를 이용해 공격자가 자신이 원하는 계좌로 공격 대상자의 예금을 이체시키는 것이다. 이러한 공격에 대비해 인터넷 뱅킹 솔루션을 개발하는 은행 및 공급업체는 방화벽, 키보드 보안 및 암호화 솔루션 등을 인터넷 뱅킹 사용자에게 제공하고 있다. 하지만 이런 노력에도 불구하고, 보안에 무관심하거나 또는 공공에게 노출된 PC로 인터넷 뱅킹을

· 본 연구는 2009학년도 홍익대학교 학술연구진흥비 지원으로 이루어졌음

† 정 회 원 : 홍익대학교 컴퓨터공학과 교수
jcpark@hongik.ac.kr

논문접수 : 2009년 6월 17일
심사완료 : 2009년 8월 31일

정보과학회논문지: 정보통신 제36권 제6호(2009.12)

하는 경우 해킹을 통한 금전적 피해의 가능성은 여전히 남아있다. 이런 상황에 대비하여 인터넷 뱅킹 세션 중에 계좌 이체 같은 주요 트랜잭션에 대해서는 추가의 인증을 요구하고 있다. 이를 위해 사용자에게 오프라인 보안 카드를 발급하고, 그 중 랜덤한 몇 개의 내용을 질문하는 것이 많이 사용되어 왔다. 하지만 이 방식은 공격자가 충분한 시간동안 계속 공격 대상자의 PC를 관찰할 수 있다면 한정된 가지 수 때문에 결국은 공격자가 필요한 모든 정보를 수집할 수 있다는 문제점을 가진다. 이에 최근에는 이중요소(two-factor) 인증[1]을 위해 일회용 암호(One-time Password, 이하 OTP)[2,3] 생성기를 사용해서(단추를 누르거나, 켜거나, 조작이 필요 없이 숫자가 출력되거나) 보이는 숫자 암호를 사용자가 입력하도록 하는 방식이 더 안전하다고 판단되어 널리 사용되고 있다.

본 논문에서는 현재 쓰이는 OTP의 보안상 문제점들을 제시하고, 이에 대한 대안으로 널리 보급된 개인 휴대폰 상에서 보안성이 강화된 OTP 생성기를 설계한다. 이를 통해 PC상에서 은행 서버에 인증을 받은 사용자가 특정 트랜잭션을 추가 승인하기 위해 휴대폰에서 계산한 OTP를 서버에 제시하게 된다. 제안하는 OTP 생성기는 챌린지-응답 방식을 적용한 것으로 스파이웨어를 이용한 공격은 물론, 시간 동기화를 이용하는 현재의 OTP 생성기와 달리 중간자(Man-In-The-Middle) 공격[4]에도 근원적 방어가 가능하다. 또한 휴대폰이 오프라인 상태에서도 동작하며, 장치 도난 시에도 사용자 PIN을 통해 일정 수준의 보안성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 OTP의 개요 및 휴대폰을 OTP 생성기 등 보안토큰으로 활용한 기존의 사례들을 제안 기법과 비교하여 소개한다. 3장에서는 제안하는 휴대폰 상의 OTP 생성기의 설계를 상세하게 서술한다. 4장에서는 제안 기법의 효용성을 보안성 및 편의성의 측면에서 분석하고, 5장에서 결론 및 향후 연구 과제를 제시한다.

2. 배경 및 관련 연구

2.1 OTP 생성 방식 및 국내 인터넷 뱅킹에서의 OTP

OTP 생성 방식은 클라이언트(OTP 장치)와 서버 사이에 암호 생성을 동기화 하는 방식에 따라 구분된다. 첫째, 이벤트 동기화 방식은 동일한 방식(함수)으로 클라이언트와 서버가 매번 암호를 계산하는 것이다. 예를 들면 [2]의 OTP 기법은 클라이언트와 서버가 임의의 비밀 값에 대해 해쉬 함수를 특정 횟수만큼 반복 적용한 값을 암호로 쓴다. 반복 횟수가 매번 줄어들기 때문에 해쉬의 일방향성에 의해 이전의 암호로부터 다음 암호를 유추하는 것이 불가능하다. 다만 반복 횟수가 1에 다다르면 해당 클라이언트의 암호 관련 내용을 다시 설정해야 한다. 또한 OTP 장치에서 암호를 생성하기만 하고 사용하지 않으면, 클라이언트와 서버 간의 동기화가 어긋날 수가 있다. 둘째, 시간 동기화에 의한 방법이 있는데, 클라이언트와 서버가 일정한 시간 간격(예: 1분)마다 동일한 방법으로 새로운 암호를 계속 생성하는 방식이다. 클라이언트의 계산 값이 서버에 제시되어 서버의 계산 값과 일치하는지 비교, 검증하게 된다. 하지만, 이 방식은 암호 입력 중에 새로운 암호가 만들어 질 수 있어 선의의 인증 실패 가능성이 있으며, 이를 막고자 시간 간격을 너무 길게 잡으면 중간자 공격의 가능성이 커진다. 셋째, 챌린지-응답에 의한 방법이 있는데, 서버가 적절한 랜덤 숫자 챌린지를 생성해서 클라이언트로 보내면 클라이언트는 챌린지 값을 사용하여 OTP 응답을 만들고, 이 응답이 서버로 다시 전달되는 것이다. OTP를 생성할 때는 챌린지 이외에 자신과 서버만이 공유하는 비밀 값 등을 이용하게 된다. 챌린지가 중복 사용되지 않도록 주의해야 하며, 사용자의 직접 입력 내용이 다른 방식에 비해 많다는 부담이 있다. 이외에 시간 동기화와 이벤트 동기화를 동시에 쓴다든지 해서 위의 방식들을 혼합 사용하는 것도 가능하다.

다수의 외산 및 국내 OTP 업체들은 각기 독자적인 제품을 내세우고 있지만, 금융감독원의 권고에 따라 국내 인터넷 뱅킹용 OTP 기술은 공통적으로 시간 동기화 방식을 따르고, 별도의 하드웨어 토큰을 OTP 생성기로 쓰도록 하고 있다. 또한 시간 동기화 간격은 중간자 공격의 가능성을 줄이기 위해 1분에서 30초로 축소되는 경향을 보인다. 단일한 방식이 권고됨에 따라 사용자는 발급 은행뿐 아니라 다른 은행에도 자신의 OTP 장치를 그대로 쓸 수 있다.

2.2 보안토큰으로서의 휴대폰 이용 및 이중요소(two-factor) 인증에의 적용 사례

휴대폰은 휴대성이 높기 때문에 보안 토큰으로 사용되는 경우 매우 유용한 인증 수단이 될 수 있다. RSA Security의 SecurID[5]는 휴대폰이나 PDA 등 사용자가 항시 휴대하는 장치에 보안 알고리즘을 구현함으로써 보안 토큰을 실현한 것인데, 휴대폰을 직접 PC에 연결하거나 휴대폰이 생성한 토큰 값을 PC를 통해 입력함으로써 사용자 인증의 보조 수단으로 쓴다. Vasco사의 Digipass 솔루션[6]이나 오픈 소스 Mobile-OTP[7]도 휴대폰과의 결합이 가능하여 휴대폰이 시간 기반의 OTP를 생성하고, 이를 서버에 제시할 수 있도록 한다. 이런 방식들은 본 제안 방법과 달리 트랜잭션에 연계된 일회용 암호를 생성하지는 않는다. 또 다른 휴대폰보안 토큰의 사례로 FireID[8]를 들 수 있다. FireID는 별도의 FireID 인증 서버를 두어서 인터넷 상의 어떤 서버

로부터 인증을 받기 위해서는 이 서버가 FireID 인증 서버에 다시 보안 토큰이 생성한 암호가 맞는지 질의해야 한다. 이 제품은 범용의 사용자 인증 수단을 목표로 하며, 별도의 인증 서버를 추가로 요구한다는 점에서 본 논문의 제안 방식과는 거리가 있다. 유사 제품으로 VeriSign사의 VIP 인증 서비스[9]를 들 수 있는데, 이것 역시 사용자의 입력 정보가 OTP로 바뀌어 별도의 VeriSign 서버로 인증을 위해 전달되어야 한다. 게임 서버로의 인증을 목표로 국내 이니텍, u-OTP 등의 업체에서 휴대폰 OTP를 제공하는 사례도 있다.

한편 Aradiom사의 SolidPass[10]라는 이동 보안 토큰은 본 제안 방식과 유사한 Transaction Data Signing(TDS)이라는 기능을 제공한다. 이 기능은 서버가 제시한 암호화된 트랜잭션 챌린지에 대해 SolidPass 휴대폰 프로그램이 트랜잭션의 내용을 이용해서 응답을 생성하고, 사용자는 응답을 보고 주어진 트랜잭션을 실제 실행할지를 결정하도록 하는 것이다. 인터넷 뱅킹에 적용될 경우 SolidPass는 중간자 공격을 막기 위해 은행 서버 및 은행 서버가 제시하는 트랜잭션 내용이 자신이 원하는 내용이 맞는지 확인할 수 있다. Deepnet Security사의 MobileID[11] 역시 트랜잭션의 디지털 서명 기능을 제공한다. 하지만 휴대폰만을 사용한 모바일 뱅킹을 고려한 SolidPass 및 MobileID가 PC 등 다른 수단을 통한 인증과 어떻게 결부될 수 있는지 제시되어 있지 않다. 이들 제품 모두 무선을 통한 휴대폰 프로그램 다운로드를 지원하기 때문에 이 과정에서 내용이 도청되면 복제폰을 통한 공격이 가능하다. 또한 트랜잭션 내용을 응답에 반영하는 방법 및 이를 검증하는 방법이 공개되지 않아, 그 보안성에 대한 객관적인 검증이 불가능하다. 본 제안 방식은 사용자가 은행 서버로 자신이 요구한 트랜잭션의 내용과 연계한 OTP를 제시하기 때문에, 트랜잭션 인증이 이들 제품과 반대 방향으로 이루어진다. PC 기반 국내 인터넷 뱅킹에 적용을 목표로 제안된 본 방식은 공개된 암호화 기법들만을 사용하여 특허 침해가 없으며, 방식의 완전 공개를 통해 제3자에 의한 보안성 검증이 가능하다.

단문 메시지 서비스(SMS)나 휴대폰 통화를 통해 이중요소 인증을 달성하려한 사례들도 있다. 뉴질랜드 ASB 은행은 일정 금액 이상 이체 시 고객의 휴대폰에 전송된 문자를 입력하도록 요구하였다. 기타 Swivel사의 PINsafe의 단문 메시지 입력을 통한 인증[12], PhoneFactor에서 휴대폰 통화 후 PIN 입력을 통한 인증[13] 등을 들 수 있다. 또한 OTP 생성 시 휴대폰 단문 메시지를 통해 수신한 내용을 사용하는 방식[14], 단문 메시지를 휴대폰 OTP의 백업 수단으로 활용하는 방식[15] 등이 제시되었다. 이런 방식들은 휴대폰 망이라는 별도 채널을 사용하기 때문에 그 적용 대상의 보안성을 향상시키는 효과를 낼 수 있다. 다만 휴대폰 분실의 문제 및 특정 트랜잭션과의 연계성 미비로 중간자 공격에 취약하다는 한계를 가진다.

3. 휴대폰 상의 OTP 생성기

본 논문에서 요구하는 휴대폰은 숫자 및 특수 문자(*, #) 키패드를 가지며, 소형 입출력 LED 창을 가지고, 자바 등으로 작성된 무겁지 않은 모바일 프로그램을 원활히 실행시킬 수 있는 범용의 플랫폼으로서, 현재 사용되는 휴대폰들의 최소 사양에 해당한다고 볼 수 있다.

3.1 휴대폰 프로그램 다운로드 및 등록

제안 방식에서는 사용자가 자신의 휴대폰을 가지고 직접 은행에 가서 필요한 프로그램을 다운로드 받고, 다운로드 받은 프로그램 카피에 고유하게 설정된 비밀 키를 서버에게 안전하게 등록한다. 현재도 하드웨어 토큰 OTP 장치를 구매 및 등록하기 위해서는 은행에 직접 방문해야 하기 때문에 제안 방법이 더 큰 부담을 요구하는 것은 아니다.

서버에의 OTP 기기 등록을 위해 먼저 휴대폰 종류에 따라 적절한 프로그램을 선택한다. 각 실행 프로그램 카피에는 코드 내에 카피마다(즉, 휴대폰마다) 고유한 비밀 키가 내장되어 있다. 고유 값을 지정하는 소스 코드는 다음과 같다(사용 언어에 따라 구문 다름).

```
X = 0x4321aabb56ef78cdabcd6789;
    // 아래에서 설명할 응답(response) 계산에 사용
```

또한, 사용자가 휴대폰 상의 OTP 생성기를 사용할 때마다 입력할 4자리 숫자 PIN을 등록한다. 이 X 값과 PIN 값이 휴대폰의 주인인 인터넷 뱅킹 사용자의 기타 ID 관련 정보(사용자 id, password, 계좌 번호 등)와 연계되어 은행 서버에 저장된다.

3.2 챌린지 생성 및 응답 처리 과정

제안 방식은 아래와 같이 사용자가 PC를 통해 인터넷 뱅킹 세션을 연 상태에서 진행된다.

```
사용자
 PC --(공인인증서로 인증된 상태)--은행 서버
  이체(예: (03,987341290762,10000)) 트랜잭션-->
   <-- 챌린지
휴대폰<-- PC : 챌린지, 트랜잭션, PIN 입력(수작업)
휴대폰 -->PC : 응답 입력(수작업)
                                    응답-->
                                  확인 OK면 진행
```

서버의 챌린지가 어떤 사용자에게 반복 제시되는 경우, 이 사용자의 정보를 스파이웨어로 빼낸 공격자는 트

랜잭션을 교체할 수는 없지만 이전의 트랜잭션을 반복하는 재생(replay) 공격은 시도할 수 있다. 따라서 서버의 챌린지 값은 적어도 동일 사용자에게 반복되어서는 안 된다. 이를 위해 서버가 챌린지를 만들 때 HC-256[16] 스트림 암호화 알고리즘을 통해 생성된 랜덤 키스트림(keystream)을 24-bit 단위로 잘라서 사용한다. HC-256는 유럽의 eSTREAM이 선택한 스트림 암호화 알고리즘들 중 하나로 256-bit의 비밀 키 K와 256-bit의 초기 벡터 IV를 이용하여 뛰어난 랜덤 성질을 보이는 키스트림을 최대 2^{128} bits 만큼 생성해 낸다.

HC-256은 특허에 걸려있지 않아 자유로운 사용이 가능하다. 서버는 랜덤한 K 및 IV 값으로 HC-256를 실행하면서 챌린지를 요구하는 모든 사용자들에게 생산되는 키스트림을 계속 24-bit 크기로 최대 2^{128} bits에 이를 때까지 제공하고, 이후 다시 새로운 K 및 IV를 가지고 이 과정을 반복한다.

3.3 휴대폰 OTP 생성기를 이용한 응답 생성 방법

휴대폰의 OTP 프로그램은 (1) 챌린지 C, (2) 해당 트랜잭션 T, (3) 사용자 PIN 값 P를 키패드를 통해 입력받아, 이 값들과 코드내의 고유 값 X를 이용하여 일회용 암호인 응답을 계산하고 휴대폰 LED 창을 통해 출력한다. 챌린지 C는 4-digit 값이며, 각 digit은 영대문자(A-Z), 영소문자(a-z), 숫자(0-9), 특수문자(*, #)의 64개 값을 6 bits로 표현하는데 구체적 인코딩은 base-64와 동일하다(단, +, / 대신 *, # 사용). 해당 트랜잭션 T는 계좌 이체의 경우 은행코드(2자리), 계좌번호(12자리) 및 이체금액(9자리)으로 구성되는데 각 자리는 0-9의 숫자로서 ASCII 코드로 표현된다. 필요 시 각 부분의 자리 수가 더 늘어난다 하더라도 알고리즘에서 이를 반영할 수 있다. 사용자 PIN인 P는 4자리인데 각 자리는 0-9의 숫자로 역시 ASCII 코드로 표현된다. 코드내의 고유 값 X는 k-bit로 표현된다고 하자. 응답 계산에는 HMAC이 다음과 같이 사용된다.

$Y = HMAC(P||T, C||X) = Hash[\ ((P||T)^{+} \oplus opad)\ ||\ Hash[\ ((P||T)^{+} \oplus ipad)||(C||X)\ ]\ ]$;

R = last 24 bits of Y; // R이 응답임.

단, ||는 concatenation 연산을 말하고, Hash는 SHA-256, $(P||T)^{+}$는 SHA-256의 입력 블록 크기인 512-bit(64-byte) 만큼 0으로 (P||T)를 padding한 것, ipad는 0x36을 64번 반복한 것, opad는 0x5c를 64번 반복한 것을 의미한다. HMAC의 두 가지 입력 값인 P||T와 C||X는 매번 바뀌도록 고안되었다. OTP 생성 프로그램은 R 값 24-bit를 계산하고, 이를 6-bit 단위로 나누어 각각을 base-64로 인코딩한 결과의 4 개의 문자(0-9, A-Z, a-z, *, #)를 화면에 출력한다. 사용자는 휴대폰 화면상의 응답을 다시 PC에 입력해 이 값이 서버에 전달되게 한다.

서버는 챌린지를 보내기 전에 수신한 트랜잭션 내용, 보낸 챌린지 및 서버에 저장된 해당 사용자의 X와 PIN 값을 이용해서 위와 같은 방법으로 HMAC 계산을 하고, 그 결과의 R이 수신한 응답과 일치할 때만 이 트랜잭션 내용을 진행한다. 무작위 입력 시도를 막기 위해 3회 이상 잘못된 OTP 값을 입력하면, 해당 사용자의 이체 시도 자체를 서버에서 금지시킨다.

4. 보안성 및 사용 편의성 분석

제안 기법은 강화된 보안성을 가지며, 사용의 편의성도 크게 뒤떨어지지 않음을 보인다. 단, HMAC이나 HC-256 자체에 대한 암호해독 공격은 고려치 않는다.

4.1 보안성

먼저 인터넷 뱅킹 시스템에 대한 실제 공격 사례를 통해 사용된 방식을 분류한다. 첫 번째 유형이 보안카드를 이용하는 고객의 비밀 정보 수집을 통한 공격이다. 2009년 1월 하나은행의 고객 예금 2100만원이 무단 유출된 사건이 있었고, 동일한 수법으로 보이는 불법 예금 인출 사건이 2008년 10월 우리은행, 같은 해 12월 시티은행의 고객 계좌에서 발생했다. 이들 사건의 공통적인 공격 형태로 공격자가 키보드 보안 프로그램을 무력화한 후 공격 대상 계좌 소유자의 각종 비밀 정보를 빼내갔을 것으로 수사 당국은 추정하고 있다. 이런 방식으로 불법 계좌 이체가 가능함이 SBS 등에서 방송된 모의실험을 통해 입증되었다. 공격 대상자의 비밀 정보 중에 특정 트랜잭션용 OTP와 비교할 수 있는 것이 보안카드의 내용이다. 공격자는 실제 해킹 사례에서 보듯이 보안카드의 내용을 공격 대상자의 뱅킹 행위를 오랫동안 관찰하면서 수집하거나, 또는 전산 파일로 만들어져 PC에 보관되는 보안카드 내용을 직접 탈취하여 획득한다. 어떤 경우이든 보안카드의 내용을 통해서 공격자는 자신이 원하는 계좌에 원하는 금액만큼의 이체 명령을 내릴 수 있다. 두 번째 유형이 메모리 해킹을 이용한 계좌 이체 내용 변경 공격이다. 2007년 시사매거진 2580이라는 방송에서 실제 시연을 통해 이런 공격이 가능함을 보여준 바 있다. 방법은 공격 대상자 PC의 메모리 내용을 해킹 툴을 이용해 변경시켜서 정상적 계좌 이체 명령의 목적지 계좌 번호 및 이체 금액을 공격자가 마음대로 조작하는 것이다. 세 번째 유형은 OTP를 도입한 뱅킹 시스템에 대한 중간자 공격이다. 2006년 발생한 미국 시티은행의 해킹 사례[17]에서, 공격자는 교묘히 작성된 가짜 시티은행 피싱 사이트로 고객을 유도해 여기에 고객의 각종 비밀 값들을 입력하게 하고, 탈취된 정보를 이용해서 실제 시티은행 서버로의 중간자 공격을 시도하였다. 이런 공격에 대한 대비로 국내 은행들은 OTP

의 유효 기간을 30초로 제한하고 있으나, 공격을 완전히 막을 수는 없다. 공격 대상자의 기타 비밀 정보를 모두 빼낸 공격자는 공격 대상자가 인터넷 뱅킹을 시작하면 (1) 중간에서 서버로 가는 트래픽을 자신이 원하는 내용으로 바꿔치기 한다(피싱 사이트 등 이용). 공격 대상자로부터 OTP 값을 입력받아 이를 서버에 전달해서 이체를 실행시킨다. 또는 (2) 훔친 비밀 정보로 동시에 인터넷 뱅킹 세션을 연다. 공격 대상자의 정상적 계좌 이체 명령에 대한 OTP 입력 내용을 스파이웨어를 통해 획득한 후, 해당 OTP의 30초 유효 기간 내에 공격자가 원하는 이체 명령을 이 OTP를 이용하여 서버에 제시한다. 은행 서버에서 동일 OTP의 연속된 입력을 거부하지 않는다면, 서버는 공격자의 명령을 받아들일 것이다.

본 논문의 제안 방식은 OTP를 계좌 이체 내용과 강한 암호화 기법으로 연결시킴으로써 위에서 제시한 세 가지 유형의 공격 모두를 근원적으로 방어한다. 제안 방법을 통한 OTP 생성은 휴대폰이 오프라인인 상태에서도 가능하고, 생성 과정에서 휴대폰을 통해 어떤 외부와의 통신도 발생하지 않기 때문에, 통신 내용의 도청 등을 통한 공격 시도는 성공할 수 없다.

4.2 사용 편의성

휴대폰에 챌린지, 트랜잭션 내용, PIN을 입력할 때 총 입력되는 문자의 수는 31개에 해당한다. 현재의 은행권 OTP 생성기가 최대 한 번의 단추를 누르는 것을 고려하면 산술적으로는 부담이 크게 늘어난 셈이 된다. 하지만 대다수 사용자들이 휴대폰이라는 기기에 매우 친숙하며, 이 정도 길이의 텍스트 메시지 입력에 익숙하다는 점을 고려하면 제안 기법의 요구 사항은 감내할 만하다고 판단한다. 또한 비록 휴대폰이 현재의 OTP 생성기보다 더 크고 무겁지만, 항시 휴대성의 측면에서 휴대폰은 다른 어떤 소형 기기보다 뛰어나다.

5. 결 론

본 논문에서 중간자 공격을 근원적으로 차단할 수 있는 새로운 OTP 방식을 제안하고 이를 휴대폰 상의 프로그램을 통해 실현할 수 있음을 보였다. 제안 방법이 공격자에 의해 깨지려면 다음과 같은 극단적인 가정이 필요하다. 공격자가 (1) 공격 대상자의 모든 비밀 정보(공인인증서, 인증서 암호)를 PC 해킹 등으로 알아낸 후 인터넷 뱅킹 세션을 시작해야 하며, (2) 분실 신고가 서버에 접수되기 전에 공격 대상자의 휴대폰을 소유하거나, 또는 휴대폰에 설치된 프로그램의 비밀 값 X를 휴대폰 해킹 등으로 알아내야 하며, (3) 휴대폰 입력 PIN 값을 유추해서 알아내거나, 또는 휴대폰용 스파이웨어 등으로 알고 있어야만, 비로소 공격 대상자의 계좌로부터 공격자가 원하는 계좌로의 이체가 가능하다. 이에 반해 현재의 은행권 OTP 생성기는 (1)의 상황에는 대비되어 있으나, (2)와 같은 생성기 분실 시 (3)에서 요구하는 사용자 PIN 입력을 요구하지 않아 바로 타인에 의한 OTP 생성이 가능하며, 이 OTP가 유효 기간 내에는 어떤 다른 이체 명령도 승인할 수 있다는 문제를 가진다. 제안 방법은 휴대폰이 오프라인인 상태에서도 동작하고, 어떤 외부와의 통신도 요구하지 않으며, 분실이나 도난의 경우에도 사용자 PIN을 통해 불법적인 OTP 생성 시도를 막는다. 또한 별도의 OTP 생성 장치를 구비하는데 따르는 비용이나 배터리 교체 및 휴대의 부담에서 자유롭다.

제안 방법에서 응답으로 계산된 256-bit 값을 일회용 대칭 키로 활용할 방안에 대해 후속 연구를 진행할 것이다. 또한 사용자를 속여서 정보를 빼내는 새로운 공격 유형에 대비해서, 별도의 채널을 통한 추가 인증 방식을 결합하는 연구가 필요하다.

참 고 문 헌

[1] B. Schneier, "Two-Factor Authentication: Too Little, Too Late," *Communications of the ACM*, vol.48, no.4, April 2005.

[2] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol.24, no.11, pp.770-772, 1981.

[3] N. Haller, C. Metz, P. Nesser, and M. Straw, A One-Time Password System, RFC 2289, IETF, http://www.ietf.org/rfc/rfc2289.txt?number=2289, 1998.

[4] B. Schneier, Man-in-the-Middle Attacks, Schneier on Security Blog, http://www.schneier.com/blog/archives/2008/07/maninthemiddle_1.html, July 2008.

[5] RSA SecurID, Security Your Future with Two-Factor Authentication, http://www.rsa.com/ node.aspx?id=1156/

[6] VASCO The Authentication Company DIGIPASS, http://www.vasco.com/products/digipass/digipass_index.aspx#

[7] Mobile One Time Passwords, Mobile-OTP v.1.06, http://motp.sourceforge.net/

[8] FireID, The Universal Personal Authenticator, http://www.fireid.com/

[9] VeriSign VIP Authentication Services, Trusted and Convenient Log-In and Transactions, http://www.verisign.com/authentication/consumer-authentication/vip-authentication

[10] The Real Solution - Aradiom SolidPass, Security Token, http://www.aradiom.com/SolidPass/ 2fa-OTP-security-token.htm

[11] MobileID, A mobile, two-way and two-factor authentication, http://www.deepnetsecurity.com/ products2/mobileid.asp

[12] Swivel Authentication Solutions, PINsafe, http://

www.swivelsecure.com/?page=pinsafe
[13] PhoneFactor, PhoneFactor Solutions, http://www.phonefactor.com/security_tokens/
[14] S. Hallsteinsen, I. Jorstad, and D.V. Thanh, "Using the mobile phone as a security token for unified authentication," in *Proc. 2nd Int'l Conf. on Systems and Networks Communications*, 2007.
[15] F. Aloul, S. Zahidi, and W. El-Hajj, "Two Factor Authentication Using Mobile Phones," in *Proc. of 7th ACS/IEEE Int'l Conf. on Computer Systems and Applications*, May 2009.
[16] H. Wu, "A New Stream Cipher HC-256," in *Proc. of 11th Int'l Workshop on Fast Software Encryption*, LNCS 3017, pp.226-244, 2004.
[17] Security Fix - Citibank Phish Spoofs 2-Factor Authentication, http://blog.washingtonpost.com/ securityfix/2006/07/citibank_phish_spoofs_2factor_1.html

박 준 철

1986년 서울대학교 계산통계학과 졸업. 1988년 KAIST 전산학과 석사. 1998년 Univ. of Maryland, College Park, USA 전산학 박사. 현재 홍익대학교 컴퓨터공학과 부교수. 관심분야는 시스템 및 네트워크 보안