모바일 기기를 이용한 OTP 인증

최종석*, 정민경*, 신승수*
*동명대학교 정보보호학과
e-mail: bestofcom@gmail.com

OTP Authentication using a Mobile Device

Jong-Seok Choi*
Min-Kyoung Jung*
Seung-Soo Shin*
*Dept. of Information Security, TongMyong University

요 약

최근 정보통신기술이 발달하면서 온라인상에서 다양한 서비스들이 이루어지고 있으며, 그에 대한 해킹 및 다양한 공격 기법들이 생겨나고 있다. 그래서 사용자를 인증하기 위해서 정적인 패스워드가 아닌 동적인 패스워드, 즉 OTP(One-Time Password)를 사용하고 있다. 현재 OTP 토큰을 이용한 OTP 생성알고리즘의 문제점이 제기 되고, 따라서 본 논문은 모바일 기기를 이용한 인증을 통해 그 문제점들을 보완하기로 한다. OTP를 생성할 때 수시로 모바일 기기는 서버와 통신함으로써 절대적인 단방향 생성이 아닌 능동적인 쌍방향 생성을 통해 서로에 대한 신뢰성과 제기된 문제점을 보완하고자 한다. 그리고 OTP 토큰을 사용 시 성능 및 기능의 제한을 모바일 기기를 사용함으로써 극복하고자 한다. 향후에는 모바일 기기를 이용한 3-factor 인증에 대한 연구를 통해 신뢰성을 높이고, 자체인증시스템을 더욱 개선함으로 대중화된 모바일 기기를 이용한 다양한 인증 서비스를 제공하고자 한다.

1. 서 론

최근 컴퓨터와 정보통신기술의 급속한 발달로 온라인으로 많은 서비스들이 이루어지고 있다. 온라인으로 많은 서비스들이 이루어짐에 따라 온라인을 통해서 송·수신 되는 정보들의 가치도 더욱 높아지고 있다. 특히 인터넷 뱅킹과 전자상거래와 같은 전자금융거래에 대한 정보의 보안수준을 향상하기 위해 인증이 필요하다. 주로 사용 되는 인증 방법으로는 ID와 Password 를 사용하는 인증방법을 예로 들 수 있다. 그러나 이때 사용 되는 Password 는 변하지 않는 정적인 Password 라고 할 수 있다. 따라서 스니핑이나 스푸핑과 같은 공격에 의해서 노출되면, 악의적인 공격자가 이를 이용하여 정당한 사용자로 위장할 수도 있다. 그래서 많은 금융기관과 정부는 전자금융거래의 안전성을 강화하기 위한 수단으로 일회용패스워드(OTP)를 도입하여 사용하고 있다. OTP는 인증 시마다 새로운 비밀번호를 생성하고 한 번 사용한 비밀번호를 다시 사용하지 않으며, 인터넷에서 주로 사용되는 정적인 Password와는 달리 동적인 Password이다. 따라서 스니핑이나 스푸핑과 같은 공격을 당해도 그 OTP는 다시는 사용되지 않기 때문에 이와 같은 문제를 해결할 수 있다.

여러 금융기관에서는 OTP 토큰을 이용한 2-Factor 인증을 기반으로 하여 OTP 인증 서비스가 시행되고 있다. 그러나 OTP 토큰 분실 및 도난에 따른 여러 가지 문제점이 제기되고 있다. 우리나라 인구 4,500만 명 중 휴대폰 가입자의 수가 4,000만 명에 육박하고 있다.[1] 따라서 본 논문에서는 모바일 기기를 이용한 사용자-서버 간에 능동적인 통신과 모바일 기기의 자체 인증을 통한 신뢰성 높은 OTP 를 생성하고자 한다.

2. 관련 연구

2.1 인증

인증(Authentication)이란 수신자가 어떤 사물이나 사람이 실제사물이나 사람과 같은 것임을 규명하는 것이다.[2] 즉, 공격자가 실제사물이나 사람을 다른 사물

이나 사람으로 바꿀 수 없는 것이다.

2.1.1 개체 인증

개체 인증(entity authentication)이란 한 개체가 다른 한 개체의 신원을 증명할 수 있도록 설계된 기술을 말한다.[3] 개체 인증을 하기 위해서는 다음과 같은 세 가지 정보를 사용할 수 있다.

(가) 알고 있는 것(something known)

주장자만 알고 있는 비밀로서 검증자에 의해 검증될 수 있다. 예를 들면 패스워드, PIN, 비밀키 등이 있다.

(나) 소유하고 있는 것(something possessed)

주장자의 신원을 증명할 수 있는 것을 말한다. 예를 들면 운전 면허증, 신분증, 여권, 신용카드, 현금인출용 카드 등이 있다.

(다)태생적으로 가지고 있는 것(something inherent) 주장자의 타고난 특성을 말한다. 예를 들면 지문, 음성, 홍채 패턴 등이 있다.

2.2 OTP(One-Time Password)

일반적인 패스워드는 정적인 패스워드로 네트워크 도청으로 인해 패스워드를 알아냈을 경우 불법적으로 재사용할 위험이 있다. 그러나 OTP는 필요에 따라 새로운 패스워드를 생성하기 때문에 네트워크 도청을 통하여 패스워드를 알아내더라도 더 이상 사용할 수 없으므로 이러한 위험을 방지 할 수 있다. 따라서 OTP는 정적인 패스워드 사용에 따른 위험을 해결하고 개인정보 유출에 따른 사용자 인증을 강화하기 위해 도입 되었다. OTP는 동적인 패스워드로 사용하기 위해서는 별도의 매체가 요구된다. 이 매체는 OTP를 생성할 수 있는 기능을 가지는 장치로 OTP 토큰이라고 한다. OTP는 OTP 생성매체에 의해 필요한 시점에 발생되고 매번 새로운 번호를 생성한다.

2.2.1 OTP 생성 방식

(가) 질의-응답 방식

질의-응답 방식은 사용자가 서버가 제시한 질의 값을 OTP 토큰에 입력해 응답 값을 얻고 그 응답의 해당 값을 서버에 전송하여 사용자를 인증하는 방식이다.[4] 질의-응답 방식은 OTP 토큰과 인증 서버 간에 동기화해야할 기준 정보가 없기 때문에, 동기화할 필요가 없으며, 사용자와 서버 간에 상호인증을 제공하는 방식으로 쉽게 확장이 가능하다는 장점을 가진다. 그러나 사용자가 직접 질의 값을 OTP 토큰에 입력해야한다는 불편이 있으며, 인증 서버도 해당 사용자의 질의 값을 관리해야 하는 부담이 있다.

(나) 시간동기화 방식

시간 동기화 방식은 서버와 OTP 토큰 간에 동기화된 시간 정보를 기준으로 특정 시간간격(보통 1분)마다 새로운 비밀번호를 생성하는 방식이다.[5]

(다) S/Key 방식

S/Key OTP 시스템에 대한 상세한 설명은 국제단체인 IETF(Internet Engineering Task Force) 표준 RFC1320에 소개 되었다. 이 방식은 MD4 메시지 다이제스트 알고리즘을 기반으로 하는 시스템이다.[6]

S/Key OTP 시스템의 동작절차는 클라이언트와 서버 측의 두 가지 측면에서 볼 수 있다. 클라이언트에서 적절한 OTP가 생성된 후, 서버에서 검사되며 MD4 일방향 해시함수를 이용한다. OTP는 일방향 해시함수를 여러 번 적용함으로 계속해서 생성되어진다. 먼저 첫 번째로 서버는 X_{n+1} 값을 저장한다. 즉, n=4 라고 가정하면,

$$X_{n+1} = f(f(f(f(f(x)))))$$

클라이언트는 Xn 값을 OTP로 생성하여 서버에게 보낸다.

$$X_n = f(f(f(f(x))))$$

서버는 X_n 값에 일방향 해시함수를 한 번 더 수행하여 검증을 하게 된다.

$$X_{n+1} = f(X_n)$$

마지막으로, 서버는 인증이 성공하면 X_{n+1}을 X_n으로 하여 다시 X_{n+1}을 생성한다.

$$X_{n+1} = f(X_n)$$

그리고 동기화된 n 값을 1씩 증가 시킨다.

2.2.3 OTP 취약점

(가) 일방향 해시함수의 충돌성

기존의 OTP 생성알고리즘은 일방향 해시함수를 사용한다. 이때 일방향 해시함수 f는 $f: X \rightarrow Y$ ($|X| > |Y|$)이다. 따라서 우리는 비둘기집의 원리를 생각해볼 수 있다. 즉, n개의 집에 $2n$마리의 비둘기가 모두 들어가기 위해서는 평균 2마리의 비둘기가 1개의 집에 들어가야 한다. 일방향 해시함수는 이와 같은 성질을 가지는데 이 성질을 일방향 해시함수의 충돌성이라고 한다.

(나) OTP 토큰의 물리적 공격

만약 OTP 토큰을 분실 또는 도난당한다면 OTP 토큰을 취득한 사람은 OTP 토큰의 주인과 같은 OTP를 생성할 수 있게 된다. 서버 측에서는 이 OTP가 인증자와 같은 것으로 인증하게 되고 커다란 문제점이 생기게 된다.

따라서 본 논문은 조금 더 능동적으로 OTP를 생성할 수 있는 모바일 기기를 이용한 OTP 인증을 제안한다.

3. 모바일 OTP 인증

3.1 모바일 인증의 개요

최근 모바일을 이용한 고속 통신이 가능해짐에 따라 많은 서비스들이 모바일 환경에서 이루어지고 있다. OTP 토큰을 이용한 OTP 인증 시 제기되는 두 가지 문제점을 최소화 시키고 향후 발전 가능성과 반영구적 사용이 가능한 모바일 기기를 사용하여 OTP를 조금 더 능동적으로 생성하고자 한다. 모바일 OTP 인증과정은 OTP 등록, OTP 생성, OTP 인증 과정으로 나눌 수 있다.

3.1.1 모바일 인증 표기법

PIN : 모바일 기기 고유번호
ID : 사용자 식별번호
K_{us} : 사용자-서버가 공유하는 48bit 비밀키
Ru_i : 사용자의 i번째 8bit 난수
Rs_i : 서버의 i번째 8bit 난수
R_i : 서버와 사용자가 계산하는 8bit 난수
$H_i()$: i번째 일방향 해시함수
h() : 일방향 해시함수
Trunc($H_i()$, j) : $H_i()$값의 j번째부터 일정길이 추출
Time : 동기화된 시간 값
c : 동기화된 카운터 값
Truncation : 6자리 OTP 추출함수

3.2 등록과정

사용자는 안전한 채널을 통해서 서비스 제공자에게 ID, PIN 을 등록하고, K_{us}, 같은 길이를 추출하는 6개의 일방향 해시함수, Time, Count 를 동기화 시킨다.

3.3 OTP 생성

모바일 기기를 이용하여 OTP를 생성하기 위해서 모바일 기기는 서버에게

$$h(ID||PIN)$$

전송하여 인증시작을 요청한다. 서버는 등록된 h(ID||PIN)를 계산하여 동일하면 인증시작 응답을 한다. Count 값은 2분 이내에 1~6사이의 값으로 순차적으로 바뀔 수 있으며 인증요청 시 성공여부와 상관없이 카운터 값을 증가한다.

사용자는 다음과 같이 난수를 생성해서 서버에게 전송한다.

$$h(Count)||(K_{us}\oplus(Ru_1||Ru_2||\cdots||Ru_c))$$

이때 K_{us} 는 48bit의 길이 중 c * 8 의 길이만큼을 사용하여 랜덤 숫자와 길이를 동일하게 한다.
서버는 사용자로부터 받은 h(c) 와 서버에서 계산한 h(c) 값이 같은지 계산하여 카운터가 제대로 동기화 되었는지 검증하여 성공하면 다음과 같이 서버측에서도 c 개수만큼 난수를 생성하여 결합한다.

$$(Rs_1||Rs_2||\cdots||Rs_c)$$

다음과 같은 과정을 거쳐서 난수 R_i를 계산하고,

$$(Rs_1||Rs_2||\cdots||Rs_c)\ \oplus\ (Ru_1||Ru_2||\cdots||Ru_c)$$
$$=\ (R_1||R_2||\cdots||R_c)$$

서버 측에서 생성한 난수 Rs_i를 K_{us}로 XOR 연산을 통해서 사용자에게 전송한다.

$$(K_{us}\oplus(Rs_1||Rs_2||\cdots||Rs_c))$$

사용자는 같은 방법으로 R_i 값을 계산하며, 다음과 같은 방법으로 다수의 일방향 해시함수를 사용하여 하나의 해시값(TruncValue)을 얻어낸다.

$$(Ru_1||Ru_2||\cdots||Ru_c)\oplus(Rs_1||Rs_2||\cdots||Rs_c)$$
$$=\ (R_1||R_2||\cdots||R_c)$$
$$H_1(Time),\ H_2(Time),\ \cdots,\ H_c(Time)$$
$$=\ T_1,\ T_2,\ \cdots,\ T_c$$
$$Trunc(T_1,\ R_1)||Trunc(T_2,\ R_2)||\cdots||Trunc(T_c,\ R_c)$$
$$=\ TruncValue$$
$$Truncation(TruncValue)\ =\ OTP$$

위와 같은 과정을 통해서 사용자는 OTP를 구할 수 있다.

3.4 OTP 인증

서버는 사용자와 같이 다음과 같은 과정을 통해서 사용자에게 받은 OTP를 검증한다.

$$H_1(Time),\ H_2(Time),\ \cdots,\ H_c(Time)$$
$$=\ T_1,\ T_2,\ \cdots,\ T_c$$
$$Trunc(T_1,\ R_1)||Trunc(T_2,\ R_2)||\cdots||Trunc(T_c,\ R_c)$$
$$=\ TruncValue$$
$$Trucation(TruncValue)\ =\ OTP$$

서버에서 생성한 OTP와 사용자에게 받은 OTP를 비교하여 일치하면 사용자를 인증하고, 일치하지 않으면 인증하지 않는다.

인증이 완료된 후 사용자와 서버는 카운터 값을 1씩 증가시킨다. 2분 간격으로 카운터 값을 다시 1로 초기화 한다.

3.5 분석

서버-사용자 간에는 인증 시 마다 생성되는 R_i 값

을 이용하여 여러 개의 해시 값의 R_i 위치의 일정 길이만큼을 추출하여 일방향 해시함수의 충돌성을 최소화 할 수 있다. 비밀키를 MicroSD칩과 같이 모바일 기기와 따로 보관함으로써 OTP 토큰의 물리적 공격을 방지할 수 있다.

4. 인증 시 고려사항

4.1 문제점 제기

(i) 모바일 인증에서는 ID도 다른 사람에게 공개되지 않는 정보이다. 그러나 일반적으로 ID는 공개하기 쉽다. 그리고 MicroSD칩 등 모바일 기기와 분리형으로 비밀키를 보관할 경우 분실이 우려된다.

(ii) OTP 생성과정에서 통신을 이용하기 때문에 스푸핑, 재전송공격, 세션하이재킹과 같은 공격에 대한 고려가 필요하다.

(iii) 사용자가 모바일 기기를 변경했을 경우에는 정당한 사용자라 해도 인증을 성공할 수 없다.

4.2 해결방안 제안

첫 번째 제기된 문제는 추후 모바일 기기에 지문인식 장치를 추가하여 3-Factor 인증을 이용하면 더욱 편리하게 사용할 수 있다.

두 번째 제기된 통신상 해킹 문제를 크게 스푸핑, 재전송공격, 세션하이재킹으로 나눌 수 있다.

(1) 스푸핑 : 서버-사용자 간에 공유하는 비밀키 K_{us}를 사용하기 때문에 공격자가 도청한다고 해도 정확한 정보를 알 수 없으며, 공격자가 알 수 있다고 해도 OTP는 단 한 번만 사용하고 다음 세션에는 사용되지 않기 때문에 무의미하다.

(2) 재전송공격 : 타임스탬프 Time 값을 사용하고 서버와 사용자 간에는 ±2분, 즉 타임스탬프의 허용윈도우와 카운터 값을 사용하며, 인증 시마다 새롭게 만들어지는 R_i 값을 사용하기 때문에 미리 저장 해놓은 정보는 인증을 통과할 수 없다.

(3) 세션하이재킹 : 모든 정보를 h()를 통하여 해시한 값을 전송하게 되고 h() 함수가 안전한 암호학적 일방향 해시함수라면 일방향성에 의존하여 중간자가 정보를 수정하기 어렵다.

세 번째 제기된 문제는 사용자가 모바일 기기를 바꾸면 사용자가 모바일 기기 변경신청을 하도록 하여 신뢰성을 유지할 수 있다.

5. 결론

모바일 기기는 OTP 토큰에 비해 데이터 처리속도 및 대중화 같은 여러 가지 측면에서 훨씬 더 뛰어나다. 그리고 모바일 기기에 대한 연구가 활발하기 때문에 OTP 토큰을 이용한 OTP 생성알고리즘에 비해 모바일 기기를 이용하면 발전 가능성이 더욱 높다. 본 논문에서 제안하는 모바일 기기를 이용한 OTP 생성은 모바일 기기가 서버와 직접 통신을 하며 OTP 토큰에 비해 더욱 능동적으로 OTP를 생성할 수 있다.

향후에는 모바일 기기에 센서를 포함하면서, 모바일 기기를 이용한 3-Factor 인증을 사용한다면 더욱 신뢰성 높은 OTP를 생성 할 수 있을 뿐만 아니라, 모바일 기기의 처리속도를 활용한 다양한 자체인증 시스템을 적용한다면 모바일 기기의 보안성도 더욱 높일 수 있을 것이다.

참 고 문 헌

[1] 강수영, 이임영, “OTP를 활용한 UICC 기반의 인증 메커니즘에 관한 연구”, 한국정보보호학회, 2008.4

[2] Bruce Schneier, “Applied cryptography”, John Wiley & Sons, 1996

[3] Behrouz A. Forouzan, "Cryptography and Network security", McGraw-Hill, 2008

[4] 최동현, 김승주, 원동호, “일회용 패스워드(OTP: One-Time Password)기술 분석 및 표준화 동향”, 한국정보보호학회, 2007.6

[5] 서승현, 강우진, “OTP 기술현황 및 국내 금융권 OTP 도입사례”, 한국정보보호학회, 2007.6

[6] 류연호, “OTP 개념을 이용한 사용자-인증 서버의 상호 인증 모델”, NuriMedia, 2005

[7] 히로시 유키, “알기 쉬운 정보보호 개론” 인피니티 북스, 2008. 1

[8] 김기영, “일회용 패스워드를 기반으로 한 인증 시스템에 대한 고찰”, 한국정보보호학회, 2007.6