Mobile 환경에서 HOTP기반의 사용자 인증 기법

고성종, 이임영, 이상정
순천향대학교 소프트웨어 공학과
순천향대학교 컴퓨터 공학과
e-mail:[lunatics, imylee, sjlee]@sch.ac.kr

User Authentication Scheme Based an HOTP in Moible Environment

Sung-Jong Go, Im-Yeong Lee
Dept. of Computer Software Engineering, Soonchunhyang University
Dept. of Computer Engineering, Soonchunhyang University

ID/Password 방식은 노출 및 예측 공격에 대한 위험성을 안고 있다. 이를 해결하기 위한 방법으로 OTP(One time Password)를 인증 시스템에 적용할 수 있다. OTP는 매번 다른 패스워드를 생성하여 사용하는 사용자 인증 방식이다. 전자금융감독규정에 의해 OTP는 인터넷뱅킹, 모바일뱅킹, 텔레뱅킹 등 전자 금융 거래 시 보안카드를 대체하는 1등급 보안매체로 지정되었지만 OTP 단말기는 배포 및 사용의 편의성의 문제로 대중화의 어려움과 동기화 실패의 문제점이 존재하게 된다. 본 논문은 통신 기술의 발달로 현재 많이 대중화되어 있고 하나의 개인 컴퓨터와 같이 정보를 저장하고 연산이 가능한 Mobile 장치를 이용하여 HOTP기반의 OTP를 생성하여 사용자 인증을 제공함으로써 OTP 단말기의 배포 및 편의성의 문제를 해결할 수 있는 방식을 제한한다.

1. 서론

최근 인터넷의 발달로 인해 다양한 서비스들이 온라인을 통해 이루어지고 있다. 기존에 오프라인 상에서 이루어지던 서비스들이 온라인상에서 이루어짐에 따라 인터넷뱅킹, 모바일뱅킹, 텔레뱅킹 등 전자 금융 거래 시 정당한 사용자인지 판단하는 과정으로 사용자 인증 기술이 적용되고 있다.

기존의 인증 방법 중 ID/Password를 기반으로 한 인증 방식은 고정된 패스워드의 노출 가능성이 크며 패스워드 추측, 예측 공격 등의 위험성을 안고 있다.

이러한 단점을 극복할 수 있는 인증 기법으로 무작위로 생성되는 난수를 패스워드로 이용하는 사용자 인증 방식의 OTP(One Time Password) 시스템이 도입되었다. 동일한 패스워드가 반복되어 사용됨에 따라 발생하는 보안상의 취약점을 OTP는 로그인 할 때마다 매번 다른 일회성 패스워드를 사용으로써 극복하였다.

하지만 기존에 사용되고 있는 OTP 방식은 사용자 인증에 필요한 동기화된 정보를 저장한 매체로써 OTP를 생성하기 위한 별도의 전용 토큰을 사용하게 된다. OTP 이용 시 항상 전용 토큰을 소지하고 있어야하며 동기화된 정보의 오류로 인증에 문제가 발생할 수 있다.

본 논문은 Mobile을 환경에서 OTP 기법을 이용함으로써 기존 방식의 전용 토큰의 불편함과 동기화된 정보의 오류로부터 발생할 수 있는 문제점을 해결하기 위한 OTP 사용자 방식을 제안한다.

2. 관련 연구

2.1 OTP(One time Password)

매번 다른 패스워드를 사용하여 사용자를 인증하는 일회용 패스워드 방식으로 임의성을 가지는 패스를 이용함으로써 공격자가 다음에 사용할 패스워드를 유추하는 것이 어려운 방식을 말한다[1].

국내에서는 금융보안연구원에서 운영하는 OTP 통합인증센터가 설립되어 서비스가 시작되었으며 전자금융 거래 시 보안카드를 대체하는 1등급 보안매체로 지정되었다.

2.2 OTP 생성 방식

OTP의 생성 방식은 질의응답(challenge- Response) 방식, 시간 동기화(Time-Synchronous) 방식, 이벤트 동기화(Event-Synchronous) 방식 그리고 이를 혼합하여 생성하는 조합방식이 있다. 기본적으로 일회용 패스워드 인증과정은 클라이언트가 인증서버와 공유하고 있는 시간을 통해 OTP 값을 생성하고 이를 인증서버에 전송하여 서버가 전송한 OTP 값과 클라이언트가 전송한 OTP 값을 서로 비교하여 인증하는 과정으로 진행한다[3].

질의응답 방식은 서버가 제시한 질의 값을 사용자가 알고리즘에 입력하여 응답 값을 얻고 해당 응답 값을 서버에 전송하여 자신을 인증하는 방식으로 동기화 방식에 해당한다. 인증 서버간에 미리 설정되어 있는 동기화 기준정보가 없어 인증 요청 시 사용자가 직접 임의의 난수 값을 클라이언트에 이를 통해 OTP 값을 생성한다. 질의응답 방식은 동기화할 기준 정보가 없기 때문에 따로 동기화할 필요가 없으며, 사용자와 서버 간에 상호인증을 제공하고 있어 쉽게 확장이 가능하다는 장점을 가지고 있다. 그러나 서버 및 클라이언트는 질의 값과 응답 값을 개별적으로 관리하여야 한다는 불편함과 사용자는 직접 응답 값을 클

라이언트에 입력하여야 한다는 단점을 가지고 있다.

시간 동기화 방식은 서버와 클라이언트 간에 동기화된 시간정보를 기준으로 지정된 시간 간격으로 변하는 비밀번호를 생성하는 방식이다. 중간자 공격에 매우 안전한 방식이지만 사용자의 인증 요청과는 상관없이 지정된 시간 간격마다 OTP 값이 변화되어 시간 내에 입력하지 못하면 인증 재시도를 위해 기다려야하는 불편함이 존재한다.

이벤트 동기화 방식은 서버와 클라이언트가 동일한 카운트 값을 기준으로 비밀번호를 생성하는 방식으로 OTP 생성 요청을 받은 기준점으로 재요청 시까지 인증 값이 변하지 않기 때문에 사용자는 편리하게 OTP 값을 입력할 수 있는 장점이 있다. 그러나 실수로 다수의 OTP 생성을 요청하게 된다면 서버와의 동기화가 어긋나는 경우가 있어 이를 보정해야 되는 문제가 있다. 또한 중간자 공격으로 OTP 값을 획득 했을 경우 정상적인 사용자가 인증 요청을 수행하기 전까지 OTP값이 변화되지 않기 때문에 공격자가 획득한 OTP 값을 사용할 수 있다는 단점이 존재한다[4].

2.3 S/KEY

RFC 1760 표준인 S/Key 인증방식에서는 해쉬 알고리즘은 SHA-1을 이용하여 일회용 패스워드를 생성한다. S/Key 방식은 사용자의 패스워드와 서버에서 생성한 난수 Seed를 XOR연산과 해쉬 연산을 이용하여 일회용 패스워드를 생성하고 있다. 또한 서버 데이터 베이스에 해쉬 체인을 이용한 OTP 생성 값이 저장되어 있어 추가적인 인증 요구 시 저장된 OTP 테이블을 이용하여 빠른 속도를 제공하고 있다[2].

그러나 S/Key 인증방식은 모든 값이 평문으로 전송되어 공격자에게 쉽게 노출된다는 단점을 가지고 있다. 또한 서버의 난수인 Seed값이 하나의 해쉬 테이블이 사용될 동안 동일하게 유지되고 있기 때문에 N번의 로그인 횟수가 노출되면 공격자가 쉽게 다음 일회용 패스워드 값을 유추할 수 있다.

2.4 HOTP

해쉬함수를 기반으로 만들어진 OTP를 HOTP라고 한다. HOTP는 사용하는 해쉬함수의 종류에 따라 HMAC-5, HMAC-SHA-1과 같은 MAC 생성 알고리즘이 전용된 후, 동적 절단(dynamic truncation)과 나머지 연산(mod)을 통해 최종적으로 생성된다. 입력은 임의의 길이의 key와 64비트의 counter이며, 출력은 6-8자리의 10진수이다. 3단계에서 생성되는 출력 값은 시스템에서 정의하는 6이상의 정수이다[3].

Step 1. HMAC 값 생성

HS=HMAC-SHA-1(key, counter)

Step 2. 동적 절단(Dynamic Truncation)

int offset = HS[19] & 0x0F

DWORD P=HS[offset]...HS[offset+3]

SNum = P & 0x7FFFFFFF

Step 3. HOTP값 계산

$HOTP = SNum \bmod 10^{digit}$

3. 요구사항

3.1 보안 요구사항

본 연구는 인증, 무결성, 기밀성, 동기화에 대하여 다음과 같은 보안 요구 사항을 가진다.

- 인 증 : 사용자의 모바일 단말기에서 생성된 OTP는 서버로부터 인증 받아야 하며, 이를 통해 확인된 정단한 사용자만이 서비스를 제공 받을 수 있어야 한다.
- 기밀성 : 송수신되는 모든 데이터는 정당한 사용자 및 서버만 인식할 수 있어야하고 제 3자 및 공격자에게 노출되지 않아야 한다.
- 무결성 : 송수신되는 데이터는 전송 중 위조 및 변조되지 않아야 하며, 이를 검증할 수 있어야 한다.
- 동기화 : OTP 생성 시 입력으로 사용되는 시간 값과 이벤트 값은 반드시 동기화 되어있어야 하며 데이터 전송 중 불일치가 발생하지 않도록 해야 한다.

3.2 보안 위협

본 연구는 추측 공격, 재사용 공격, 중간자 공격으로부터 대하여 다음과 안전성을 제공해야 한다.

- 추측 공격: 공자에 의해 송/수신되는 데이터가 노출될 경우 반복적인 검증 시도를 통해 일회용 패스워드를 유도해내는 것은 불가능해야 한다.
- 재사용 공격: 공격자는 송수신 데이터를 수집하여 사용자 또는 서버로 전송함으로써 이미 검증 받은 데이터를 이용하여 정당한 사용자나 서버로 위장할 수 없어야 한다.
- 중간자 공격: 공격자는 송/수신 데이터를 가로채서 정당한 사용자로 위장할 수 있기 때문에 획득한 데이터를 분석하여 중요한 정보를 취득할 수 없어야 한다.

4. 제안방식

본 논문은 Mobile 환경에서 HOTP를 이용하여 OTP를 생성한다. 사용자는 사용자 정보를 서버에 등록하고 서버에 등록한 동일한 정보를 Mobile에 입력한다. 등록 과정에서 Mobile과 Server는 각각의 초기 OTP를 생성하여 공유한다. 사용자 인증 시 별도의 카운터 값은 동기화하지 않지만 등록 과정 중 생성한 각각의 고유정보를 HOTP의 입력으로 사용하여 새로운 OTP를 생성하여 사용자 인증 과정을

수행한다.

4.2 시스템 계수

· IMEI : Mobile 단말기 고유 식별 번호
(Internationl Mobile Requipment identity)
· $HOTP$: HMAC기반의 OTP 생성 알고리즘
· OTP_* : *에서 생성한 OTP
($S \rightarrow Server, M \rightarrow Mobile$)
· $h()$: 해쉬 함수
· $Nonce_*$: *의 임시 비표

4.3 등록 단계

본 논문은 사용자가 ID와 PW를 이용하여 Server에 등록하고 Server로부터 사용자의 ID를 입력으로 생성된 OTP값을 전송받는다. 사용자는 Server로부터 전송받은 OTP값을 다시 Mobile에 입력하고 Mobile은 사용자로부터 입력받은 정보를 이용하여 Server에 등록한다. 이러한 등록 과정은 공개키 기반의 안전한 통신 채널과 Mobile을 사용자가 소유한 안전한 물리적 장치라고 가정한다.

Step 1. 사용자는 ID와 PW를 이용하여 Server에 등록한다. PW는 해쉬값을 이용하여 등록한다.

$$U \Rightarrow S: ID \| h(PW)$$

Step 2. Server는 사용자의 ID와 Nonce값을 이용하여 OTP_S값을 생성하여 사용자에게 전송한다.

$$S: OTP_S = HOTP(ID, Nonce_S)$$
$$S \Rightarrow U: OTP_S$$

Step 3. 사용자는 서버에 등록한 동일한 ID, PW와 함께 서버로부터 전송받은 OTP_S를 Mobile에 입력한다.

$$S \Rightarrow M: ID \| PW \| OTP_S$$

Step 4. Mobile은 사용자의 PW를 해쉬화하여 저장하며 Seed값을 생성하여 IEMI와 입력으로 OTP_M를 생성하고, 사용자의 ID, PW와 함께 OTP_M를 서버에 등록한다.

$$M: PW = h(PW)$$
$$M: OTP = HOTP(IEMI, Nonce_M)$$
$$M \Rightarrow S: ID \| OTP_M$$

4.4 OTP 생성 및 검증 단계

Step 1. 사용자는 Server에 등록한 ID를 입력하여 Mobile을 활성화시킨다.

$$U \rightarrow M: ID$$

Step 2. Mobile 사용자의 ID를 확인하고 저장된 OTP 정보를 이용하여 새로운 OTP를 생성하여 사용자에게 전송한다.

$$M: OTP_M = h(OTP_M)$$
$$M: OTP_M = HOTP(OTP_S, OTP_M)$$

Step 3. 사용자는 자신의 ID와 Mobile로부터 새로 생성한 OTP를 Timestramp와 함께 Server로 전송한다.

$$U \rightarrow S: ID \| OTP_M \| h(OTP_M \oplus T) \| T$$

Step 4. Server는 Timestramp가 유효한지 확인한 후, 등록 과정에서 입력받은 사용자의 ID를 이용하여 사용자의 정보를 확인하고 저장된 사용자의 Mobile의 정보를 이용하여 OTP를 생성하며 사용자로부터 전송받은 OTP와 새로 생성된 OTP를 비교 검증한다.

$$M: OTP_M = h(OTP_M)$$
$$M: h(OTP_M \oplus T) = ? h(HOTP(OTP_S, OTP_M) \oplus T)$$

4.4 재사용 과정

Step 1. 사용자는 Mobile에 ID와 PW를 입력하여 Mobile을 활성화시킨다.

$$U \rightarrow M: ID \| h(PW)$$

Step 2. Mobile 사용자의 정보를 확인하고 새로운 None값을 생성하여 저장된 정보를 함께 새로운 OTP값을 생성한다.

$$M: OTP_M = HOTP(IEMI, Nonce_M)$$

Step 3. Mobile은 ***Step 3.***에서 생성된 정보와 OTP_S를 이용하여 새로운 OTP_M을 생성한다.

$$M: OTP_{M,S} = OTP_M \oplus OTP_S$$

Step 4. 사용자는 자신의 ID와 Mobile로부터 새로 생성한 $OTP_{M,S}$와 Timestramp와 함께 Server로 전송한다.

$$U \rightarrow S: ID \| h(OTP_M \oplus T) \| OTP_{M,S} \| T$$

Step 5. Server는 Timestramp가 유효한지 확인한 후, OTP_S와 XOR연산을 통해 OTP_M을 해쉬값이 올바를 경우 OTP_M을 저장한다.

$$S: OTP_M = OTP_{M,S} \oplus OTP_S$$
$$S: h(OTP_M \oplus T) = ? h(OTP_M \oplus T)$$

Step 6. Server는 새로운 None값을 생성하여 저장된 정보를 함께 새로운 OTP값을 생성하고 Timestamp와 함께 사용자에게 전송한다.

$$S: OTP_S = HOTP(ID, Nonce_S)$$
$$S \rightarrow U: h(OTP_S \oplus T) \| OTP_{M,S} \| T$$

Step 7. 사용자는 OTP_S를 Mobile에 입력하여 OTP S를 추출한다.

$$U \rightarrow M: OTP_{M,S}$$
$$M: OTP_S = OTP_{M,S} \oplus OTP_M$$

Step 8. 사용자는 Timestamp를 유효한지 확인 한 후, Mobile에 OTP_S를 저장한다.

$$U \rightarrow M: OTP_{M,S}$$
$$M: h(OTP_S \oplus T) = h(OTP' \oplus T)$$

3. 제안방식 분석

3.1 보안 요구사항 분석

본 연구는 인증, 무결성, 기밀성, 동기화에 대하여 다음과 같은 보안 요구 사항을 가진다.

· 인 증 : OTP를 생성하는 HOTP의 입력값으로 Mobile, Server사이에 소유하고 있는 동일한 OTP_M, OTP_S정보를 입력해야 유효한 OTP를 생성할 수 있다. 정당한 사용자만이 자신이 소유한 Mobile을 통해 유효한 OTP를 생성하여 서비스를 제공받을 수 있다.

· 무결성 : 송/수신되는 모든 데이터들은 해쉬값을 이용해 사용자 인증 과정을 진행한다. PIN값을 이용하며 무결성을 제공하기 위한 MAC기반의 HOTP를 이용함으로써 무결성을 보장받는다. 따라서 위/변조에 대한 공격을 방지할 수 있다.

· 기밀성 : OTP 생성 과정 중에 필요한 Nonce값과 IEMI값은 HMAC을 이용한 임의의 난수형태로 전송되며 어떠한 형태로도 원본 메시지를 노출되지 않고 전송되고 정당한 사용자와 Server만이 인식할 수 있기 때문에 기밀성을 제공한다.

· 동기화 : Mobile과 Server사이에 공유되어있는 일련의 정보들을 이용하여 OTP를 생성하게 된다. 별도의 시간 값 또는 카운트를 이용하지는 않지만 공유된 정보를 반복적인 해쉬화를 통해 Mobile과 Server사이에 동기화된 정보 상태를 유지하며 재사용 단계를 통해 Mobile과 Server 사이에서 동기화된 정보를 초기화하여 동기화된 상태를 유지할 수도 있다.

3.2 보안 위협

본 연구는 추측 공격, 재사용 공격, 중간자 공격으로부터 대하여 다음과 안전성을 제공한다.

· 추측 공격: OTP를 생성하기 위한 HOTP의 입력으로 사용되는 OTP_S와 OTP_M은 노출이 없이 전송되며 새로운 OTP_M을 생성할 때마다 해쉬화하여 매번 다른 키로 입력되기 때문에 생성되는 OTP_M를 예측하는 것은 불가능하다.

· 재사용 공격: TimeStamp를 사용하여 유효한 내에서만 인증이 가능하다. Timestamp를 조작하였을 경우에 공격자는 OTP_M, OTP_S를 알지 못하기 때문에 유효한 $h(OTP_M \oplus T)$ 또는 $h(OTP_M \oplus T)$을 생성할 수 없다. 따라서 제안한 인증 방식은 재전송 공격으로부터 안전하다.

· 중간자 공격: 공격자는 송/수신 데이터를 가로채서 정당한 사용자로 위장하거나 획득한 데이터를 분석하여 중요한 정보를 획득할 수 있다. 하지만 OTP_M과 OTP_S에 입력 값으로 사용되는 IMEI와 Nonce값은 서버의 고유한 값을 알아내거나 유효한 올바른 OTP값을 생성할 수 없다. 따라서 제안 방식은 중간자 공격으로부터 안전하다.

6. 결론 및 향후 연구 방향

본 논문에서 제안한 방식은 Mobile환경에서 HOTP를 이용한 인증 방식이다. Hash와 XOR연산만을 이용하여 간단한 구조를 이루고 있고 HMAC을 이용하여 무결성을 제공하며 서버의 Nonce값과 Mobile의 IEMI값을 이용하여 OTP를 생성하는데 사용하여 기밀성을 제공한다.

기존 S/KEY 방식은 설정한 n에 따라 사용 횟수가 제한되고 설정한 범위를 벗어날 경우 다시 초기화 과정을 거치는 번거로움이 있으며 HOTP 방식의 경우 서로 동일한 카운터 값을 유지해야 한다. 하지만 제안한 방식은 별도의 카운터 값을 사용하지 않고 Server와 Mobile의 고유값을 입력으로 사용하였다.

따라서, 본 논문은 Server와 Mobile의 고유 값을 사용하여 OTP를 생성함으로써 상호 인증을 제공하며 별도의 보안 토큰의 필요 없이 Mobile을 이용하여 금융, 통신, 복지, 유통, 교통, 사내 등 다양한 분야에서 활용될 수 있을 것이라 사료된다.

참고문헌

[1] N Haller, "A One Time Password Standard", IETF RFC 1938, 1996.

[2] N. Haller, "The S/Key One-Time Password System," RFC 1760, 1995.

[3] D. M'Raihi, "HOTP: An HMAC-Based One-Time Password Algorithm" IETF RFC 4226, 2005.

[4] 최동현, 김승주, 원동호, "일회용패스워드 기술 분석 및 표준화 동향", 정보보호학회지, 제17권 제3호, pp. 12-17, 2007.

[5] 이진용, 이진범, 황성운, 안홍영, "금융권 OTP의 취약점 분석 및 대응 방안", 보안공학연구논문지 제6권 제5호, pp. 323-335, 2009

[6] 김홍기, 이임영, "모바일 환경에서 안전한 One-Time Password 인증 기법에 관한 연구", 멀티미디어학회논문지, 제14권 제6호, pp. 785-793, 2011

[7] MH Eldefrawy, K Alghathbar, MK Khan, "Mobile one time passwords: two factor authentication using mobile phones", information Technology, pp. 508-516, 2011