가변 Digit를 사용하는 안전한 R(Random digit)-OTP 방식에 관한 연구

강수영, 이임영
순천향대학교 컴퓨터학과
e-mail:bbang814@sch.ac.kr

A Study on Secure R(Random digit)-OTP Scheme using Random Digit

Soo-Young Kang, Im-Yeong Lee
Division of Computer, Soonchunhyang University

요 약

유비쿼터스 환경이 구축됨에 따라서 온라인상에서 사용자를 인증하고 적합한 서비스를 제공할 수 있도록하기 위하여 OTP(One-Time Password)에 대한 연구가 활발히 진행되고 있다. 하지만 ID와 고정된 패스워드의 대안으로 연구되고 있는 OTP 또한 중간자공격에 취약하여 패스워드 노출 문제가 발생하고, 기존의 사용되고 있는 방식이 6~8자리의 OTP를 사용하고 있기 때문에 자릿수를 알았을 경우 사전 공격 및 추측 공격이 가능할 수 있다. 기존의 OTP는 OATH와 RSA에서 연구하고 있어 두 가지 표준으로 구분된다. 그 중 OATH에서 연구하고 있으며 RFC 4226에 기술되어 있는 HOTP는 Hash-based MAC을 이용하여 안전한 OTP를 생성하고 있다 하지만 HOTP도 자릿수(Digit)가 고정되어 있어 사전 공격 및 추측 공격에 취약할 수 있다. 따라서 본 방식은 HOTP의 자릿수를 가변적으로 생성하는 R(Random digit)-OTP를 생성함으로써 제 3자의 공격으로부터 안전하게 하였다.

1. 서론

유비쿼터스 환경이 구축됨에 따라서 오프라인의 서비스들이 온라인 서비스들로 변화하게 되었으며 온라인상에서 사용자를 인증하고 적합한 서비스를 제공할 수 있는 방안에 대한 연구가 활발히 진행되고 있다. 온라인상에서 사용자를 인증할 수 있는 방안으로는 ID와 패스워드가 가장 기반이 되는 방법이라 할 수 있다. 하지만 사용자가 지정해 놓은 패스워드는 사용자의 개인 정보와 관련이 있는 숫자 및 문자로 사용되는 경우가 많으며, 자주 갱신하지 않고 고정된 패스워드를 사용하기 때문에 악의적인 제 3자에게 노출되었을 경우 큰 피해를 입을 수 있다. 따라서 이를 해결하기 위하여 한번 사용하고 폐기하는 OTP(One-Time Password)를 사용하여 고정된 패스워드의 문제를 해결하고 있다. OTP는 리니지라는 온라인 게임의 패스워드를 생성하는 린-OTP에서도 사용되고 있으며, 가장 많이 이용되고 있는 분야로는 금융 거래로 기존에 사용하였던 보안 카드를 대체하고 있다. 현재는 이체 한도를 기준으로 이체 한도가 5천만원 이상일 경우 보안 카드 대신 의무적으로 OTP 토큰을 발급받아 사용하도록 하고 있다. 패스워드를 사용하는 많은 분야로 OTP의 사용이 증가함에 따라 응용 분야가 넓어지고 있으며 이에 따른 종류도 증가하고 있는 추세이다. 가장 일반적인 종류는 작은 토큰형 OTP이다. 그러나 휴대의 문제가 대두됨에 따라서 신용카드 모양의 카드 OTP가 상용화되어 사용되고 있으며 IC 카드와 주파를 이용하여 생성되는 소리를 OTP로 사용하는 보이스 OTP가 사용되고 있다. 또한 휴대의 불편함을 해결하기 위하여 사용자의 휴대폰에 OTP 생성 소프트웨어를 다운로드 받아 휴대폰에서 OTP를 생성하는 모바일 OTP가 있다. 이와 같이 OTP의 다양성에 따라 응용 분야도 점차 늘고 있고, 패스워드가 노출되었을 경우 다음 서비스를 제공받을 때 다른 패스워드를 사용하므로 안전하다는 장점이 있지만 OTP 방식 중 Challenge-Response의 경우 MITMA(Man-in-The-Middle Attak)이 가능할 수 있으며 대부분의 OTP 토큰에서 생성되는 6자리~8자리는 모두 숫자로 이루어져 있어 사전 공격으로부터 취약하다는 문제점이 발생하고 있다.

따라서 본 논문은 OTP의 보안을 더 강화하기 위하여 서버와 클라이언트 간에 Digit를 설정하지 않고 매 세션 가변적인 OTP를 생성할 수 있는 R(Random digit)-OTP 방식을 제안한다. 입력 값을 해쉬한 후 나온 결과 값에서 Digit를 생성할 수 있는 방안을 추가하였으며 이에 따라 악의적인 제 3자는 도청하는 세션의 OTP의 Digit를 알지 못하므로 공격이 더욱 어려워졌다. 본 논문은 2장에서 OTP 토큰을 사용하는데 있어 발생할 수 있는 보안 위협 및 요구 사항을 도출하고 3장에서는 관련 연구를 기술하며, 4장에서 제안 방식인 R-OTP에 대하여 기술한다. 5장에서는 2장에서 도출한 요구 사항을 기반으로 R-OTP를 분석하고 끝으로 6장에서 결론을 기술한다.

2. 보안 위협 및 요구 사항

본 장에서는 OTP를 사용하는데 있어서 서버와 클라이언트 간의 발생할 수 있는 보안 위협과 이를 보완하기 위하여 제공해야 할 요구 사항을 도출한다.

(1) 보안 위협

OTP를 사용할 경우 서버와 클라이언트 간의 통신 채널에서 발생할 수 있는 보안 위협이다.

◈ 도청 : 사용자와 서버 간의 통신 내용을 제 3자가 악의적인 목적에 의하여 엿들을 수 있다.

◈ MITMA(Man-In-The-Middle-Attack) : 제 3자는 사용자와 서버 간의 전송되는 데이터를 가로채 위조 및 변조하여 정당한 사용자로 위장할 수 있다.

◈ 재전송공격 : 제 3자는 전송되는 데이터를 정당한 객체에게 재전송하여 획득하고자 하는 값을 획득하거나 정당한 사용자로 위장할 수 있다.

◈ 서비스 거부 공격 : 값을 위조 및 변조하여 정당한 객체가 인증 받지 못하도록 하거나 시스템이 동작할 수 없도록 할 수 있다.

(2) 요구 사항

OTP에서 발생할 수 있는 보안 위협을 보완하기 위해서는 다음의 요구 사항들을 만족해야 한다.

◈ 인증 : 클라이언트는 서버로부터 정당성을 검증받아야 한다.

◈ 기밀성 : 통신에 사용되는 중요 값은 정당한 통신 객체들만이 공유해야 한다.

◈ 무결성 : 통신되는 데이터는 위조 및 변조되지 않도록 보안이 제공되어야 한다.

◈ 비트 연산의 효율성 : 연산을 할 때 비트수를 맞추어 패딩 비트를 줄여 데이터 효율성을 제공해야 한다.

3. 관련 연구

OTP의 표준 기술은 1995년 제안된 S/Key는 RFC 1760에 기술되어 있으며 Challenge-Response 방식을 사용하여 OTP를 생성하였고, 이를 개선하여 1996년 RFC 1938에서 OTP 표준을 제정하고 최종적으로 1998년 RFC 2289에서 표준을 확립하였다. 이후 RFC 2289는 크게 OATH(Open AuTHentication)와 RSA(r.Rivest, a.Shamir, l.Adlemans)에서 연구하고 있는 두 가지 방식으로 분류된다. OATH에서는 Hash-based MAC 기반의 HOTP를 표준으로 진행하고 있으며 RSA는 OTP 토큰의 초기화와 EAP(Exten-sible Authentication Protocol)에 관련된 표준을 진행하고 있다. 본 논문에서 연구하고자 하는 OATH의 HOTP는 이벤트 방식을 사용하고 있으며 입력 값으로 서버와 클라이언트 간에 공유된 비밀키와 이벤트 생성 횟수인 카운터가 사용되며 이를 해쉬하여 값을 생성한다. 그 후 해쉬 값에서 OTP 추출 함수를 사용하여 6~8자리 OTP를 생성한다. 본 장에서는 관련 연구에 대하여 기술한다.

(1) OTP의 종류

OTP의 종류는 비동기화 방식과 동기화 방식으로 분류할 수 있으며 비동기화 방식은 서버와 클라이언트 간의 Challenge-Response 형식으로 OTP 인증이 제공되는 방식이고 동기화 방식은 입력 값에 따라 Time Sync, Event Sync, Time Event Sync 방식이 있다.

① Time Sync 방식

본 방식은 서버와 클라이언트 간에 공유한 비밀키와 동기화된 시간 값을 기반으로 OTP를 생성하는 방식이다. 일정 시간(보통 1분)마다 OTP가 생성되기 때문에 서버로부터의 인증에 실패했을 경우 서비스를 제공받기 위한 재시도 하기 위하여 일정 시간을 대기해야 한다는 단점이 있다.

② Event Sync 방식

본 방식은 서버와 클라이언트 간에 공유한 비밀키와 동기화된 이벤트 발생 횟수 카운트를 입력 값으로 OTP를 생성하는 방식이다. 카운트는 사용자가 토큰에 있는 버튼을 눌러 생성하는 횟수이다. 그러나 사용자의 사용 미숙이나 타인의 장난으로 인한 카운트 비동기 현상이 발생하여 동기화 문제가 난해한 단점이 있다.

③ Time-Event Sync 방식

본 방식은 Time Sync 방식과 Event Sync 방식의 문제점을 보완하기 위하여 두 방식을 조합한 방식이다. Time Sync 방식에서 인증 실패 시 일정 시간을 대기해야 하는 단점을 보완하기 위하여 일정 시간 안에서도 이벤트를 발생시켜 대기 시간을 줄일 수 있다. 또한 Event Sync 방식에서의 큰 문제점인 비동기화를 일정 시간마다 동기화를 시켜 비동기 현상을 줄이게 되었다. 하지만 아직도 비동기화를 완벽하게 해결하지는 못했다.

(2) HOTP(HMAC-based One-Time Password)

본 방식은 OATH에서 진행하고 있는 RFC 4226에 기술된 OTP 방식이다. 2005년에 제안되었으며 OATH의 방향에 따라 OTP를 생성하는 알고리즘이 모두 개방되어 있다. 해쉬 함수 중 SHA(Secure Hash Algorithm)-1을 사용하여 OTP를 생성하고 있으며 Event Sync 방식을 사용하고 있으며 총 세단계로 이루어진다.

단계 1. Generate an HMAC-SHA-1 value HS

HOTP는 입력 값으로 서버와 클라이언트 간에 공유된 비밀키와 8바이트 카운트를 SHA-1 해쉬 알고리즘으로 해쉬하여 160비트의 해쉬 값 HS를 생성한다.

단계 2. Dynamic Truncation

160비트에서 OTP 생성에 사용되는 실질적인 값을 추출하기 위하여 32비트를 잘라 Sbits를 선택한다. Sbits를 선택하는 방법은 HS의 가장 마지막 4비트를 offset으로 설정하고 HS를 8바이트씩 나누어 20개의 인덱스를 붙였을 때 offset부터 offset+3까지의 인덱스에 해당하는 32비트를 추출하는 것이다. 그 값이 P가 되고 P의 마지막에서부터 31비트를 Sbits로 설정한다.

단계 3. Compute an HOTP value

32비트의 Sbits에서 처음에 설정한 Digit에 따라 HOTP를 생성하는 단계로 Sbits를 10^{Digit}로 모듈러하여 OTP를 생성한다.

4. R(Random digit)-OTP 방식

OTP는 고정된 패스워드의 단점을 해결하고자 고안된 기술이지만 온라인 서비스가 많아짐에 따라서 악의적인 제 3자에게 패스워드 노출이 점점 더 심해지고 있다. 따라서 안전하게 OTP를 사용하고 인증 제공을 위하여 가변적 Digit를 이용하여 악의적인 제 3자가 사전 공격을 시도했을 경우 더 많은 시간 및 자원이 낭비되는 R-OTP 방식을 제안한다.

(1) 가정사항

R-OTP 방식은 다음의 가정 사항을 기반으로 한다.

◈ 서버와 클라이언트는 유일한 비밀키를 서로 공유하고 카운트가 동기되어 있다.

◈ 해쉬 값 HS는 HS>10^{Digit}를 만족해야 하며 16진수 HS를 10진수로 하여 모듈러 값을 계산한다.

(2) 시스템 계수

R-OTP 방식은 다음의 시스템 계수를 기반으로 한다.

◈ SK(Secret Key) : 서버와 클라이언트 간에 사전에 안전하게 공유된 비밀키

◈ ACT(Auto CounT) : 서버와 클라이언트 간에 동기화된 이벤트 값으로 일정 시간(보통 1분) 안에 10초에 한 번씩 자동으로 생성되는 카운트

- Event Sync 방식은 사용자 및 타인의 실수로 비동기화 현상이 발생하기 쉽다. 따라서 토큰에 있는 이벤트 발생 버튼을 없애고 모듈 안에서 10초마다 자동으로 발생하는 카운트 값을 사용하여 비동기화의 문제를 해결한다.

◈ HS : SK와 ACT를 해쉬하여 생성된 해쉬 값으로 HOTP를 기반으로 SHA-1을 사용했을 경우 160비트의 HS가 생성(R-OTP는 HOTP와 동일한 환경)

- 해쉬 함수는 가변적일 수 있으며 가장 효율적이고 안전한 해쉬 함수를 선택하여 사용한다. 초기 OTP의 경우 MD4(Message Digest 4), MD5를 사용하였으며 이외에도 RIPEMD 버전들이 있다.

<표 1> 해쉬 알고리즘

	출력 길이	처리 단위	단계 수
MD5	128비트	512비트	64회
SHA-1	160비트	512비트	80회
RIPEMD-160	160비트	512비트	160회

(3) R-OTP 생성 방식

본 장에서는 HOTP를 기반으로 OTP가 생성되며 SHA-1 알고리즘을 사용하여 HTOP와 동일하게 160비트의 HS가 생성된다.

단계 1. Generate an HMAC-SHA-1 value HS

R-OTP는 입력 값으로 비밀키 SK와 자동으로 생성되는 ACT를 사용하고 두 값을 해쉬하여 HS를 생성한다. SK는 OTP 토큰마다 다르게 설정되어 있으며 각 토큰마다 유일한 값을 사용한다. ACT는 자동으로 생성되는 카운트 값으로 HOTP에서 사용되었던 카운트는 토큰의 버튼을 눌러서 생성해야 하므로 비동기화 문제가 발생하며 이를 보완하기 위하여 제품의 경우 +16의 오차 범위를 두고 있지만 오차의 범위가 커질수록 악의적인 제 3자의 공격에 취약하므로 이를 보완하기 위하여 10초마다 1씩 증가하는 자동 카운트를 사용하여 동기화를 보완하였다.

$$HOTP = HMAC - SHA - 1(SK \| ACT)$$

단계 2. Dynamic Truncation

본 단계는 160비트의 HS에서 Sbits를 선택하기 위한 단계로 offset은 HS의 말단 8비트에서 앞자리 수와 뒷자리 수를 XOR 연산한다. (예, 말단 8비트가 8F일 때 offset은 0x07이 된다.)

$$\begin{aligned} offset &= 8_{hex} \oplus F_{hex} \\ &= 1000_{bin} \oplus 1111_{bin} \\ &= 0111_{bin} = 7_{hex} \end{aligned}$$

P 값은 인덱스가 offset~offset+3까지로 저장되어 있는 값을 의미하며 offset이 0x07의 경우 7, 8, 9, 10번에 저장되어 있는 값을 의미한다.

Sbits는 P에서 하위 31비트를 추출해 내는 값이다. (예, 32비트 P가 A1 B2 C3 D4일 경우 A1에서 7비트를 추출하여 Sbits는 0x21B2C3D4가 된다.)

$$\begin{aligned} P &= A1B2C3D4_{hex} \\ A1 &= 1010\ 0001_{bin} \\ A1_7 &= 0010\ 0001_{bin} = 0x21_{hex} \\ Sbits &= 21B2C3D4_{hex} \end{aligned}$$

Digit는 OTP 생성 자릿수로 Sbits의 말단 8비트에서 앞자리 수와 뒷자리 수를 XOR한 값을 Q라하고 Q와 offset을 XOR해서 R 값을 생성한다. 그리고 두 번째 가정사항에 따라 HS>10^{Digit}를 만족해야 하므로 Digit는 9이하가 되어야 한다. 따라서 R>9 경우 R mod 10을 취한다. mod 10을 해야 9이하의 Digit가 생성되기 때문이다.

$$\begin{aligned} Q &= D_{hex} \oplus 4_{hex} = 1101_{bin} \oplus 0100_{bin} \\ &= 1001_{bin} = 9_{hex} \\ R &= Q \oplus offset = 9_{hex} \oplus 7_{hex} \\ &= 1001_{bin} \oplus 0111_{bin} = 1110_{bin} = E_{hex} \\ Digit &= E_{hex} \bmod 10_{dec} \\ &= 14_{dec} \bmod 10_{dec} = 4_{dec} \end{aligned}$$

5. R-OTP 분석

본 장에서는 2장에서 도출한 보안 위협에 안전한지의 여부와 요구 사항을 만족하는 것에 따라 R-OTP 방식을 분석한다.

(1) 보안 위협에 따른 분석

R-OTP 방식이 다음의 보안 위협에 안전한지에 대하여 분석한다.

◈ 도청 : 안전하지 않은 온라인 통신 채널에서 암호화를 제공하지 않으므로 값은 도청될 수 있으나 Digit가 매 세션 변하기 때문에 악의적인 제 3자가 사전 공격 및 추측 공격을 하기 어렵고 다음 세션에 사용될 OTP의 Digit를 추측할 수 없으므로 공격하는데 소비되는 시간 및 자원이 증가한다.

◈ MITMA(Man-In-The-Middle-Attack) : 제 3자는 사용자와 서버 간의 공유된 SK와 ACT를 알지 못하므로 안전하며, Hash-base MAC을 기반으로 하므로 위조 및 변조로부터 안전하다.

◈ 재전송공격 : ACT는 10초마다 자동으로 갱신되는 카운트로 가변적인 값으로 구성되기 때문에 재전송공격으로부터 안전하다.

◈ 서비스 거부 공격 : 전송되는 데이터가 위조 및 변조되지 못하도록 해쉬 함수를 사용하였으며 Even Sync 방식에서 가장 큰 위협인 비동기화를 해결하기 위하여 HOTP의 카운트 대신 ACT를 사용하여 동기화를 보완하였다.

(2) 요구 사항

OTP에서 발생할 수 있는 보안 위협을 보완하기 위해서는 다음의 요구 사항들을 만족해야 한다.

◈ 인증 : 서버는 클라이언트로부터 전송된 R-OTP가 자신이 생성한 OTP가 동일할 경우 인증을 제공하고 정당한 클라이언트만이 R-OTP 값을 생성할 수 있다.

◈ 기밀성 : 서버와 클라이언트는 SK를 통신로 상에 노출시키지 않으며 ACT 값이 동기화되어 있으므로 정당한 객체만이 SK와 ACT를 알 수 있다.

◈ 무결성 : Hash-based MAC 기반의 OTP 생성 방식이므로 해쉬 값을 통해 데이터의 무결성을 제공할 수 있으며 다양한 해쉬 함수로의 변환이 가능하다.

◈ 비트 연산의 효율성 : 연산 효율을 위하여 연산할 때의 비트수를 4비트로 통일하였으며 offset을 설정할 때 말단 4비트에서 8비트로 증가하지만 4비트씩 나누어 XOR 연산을 함으로써 4비트의 결과 값을 생성한다. 또한 Q를 offset과 XOR하기 위하여 8비트의 Q를 4비트로 압축하여 패딩 비트가 필요하지 않도록 하였다. 또한 Digit 값이 9 초과일 경우 HS보다 값이 더 크므로 mod 10을 해줌으로써 Digit의 최대값인 9를 넘지 않도록 하였다.

6. 결론

유비쿼터스 환경을 구축하기 위한 온라인 서비스들이 증가하고 있으며 서비스 제공을 위한 사용자 인증을 위해서 인증 기술의 연구가 활발히 진행되고 있다. 그 중 금융권의 패스워드를 대체할 OTP 기술의 사용이 급증하게 되었고 금융권뿐만 아니라 게임 등 많은 분야에서 응용될 것으로 예측되고 있다. 하지만 해쉬 기반의 함수만으로 보안을 제공하기 때문에 속도 측면에서는 효율성을 제공하지만 보안적인 측면에서는 취약점들이 드러나고 있다. 가장 큰 문제점으로 비동기화 문제 악의적인 제 3자의 공격인 MITMA가 대두되고 있다. 이를 해결하기 위하여 본 논문에서는 자동으로 생성되는 ACT 값을 두어 사용자 및 타인의 잘못으로 인한 비동기화를 해결하였고 점점 더 강해지는 공격자들의 공격에 대비하기 위하여 가변적인 Digit의 OTP를 생성함으로써 취약점을 보완하고 있다. 하지만 금융권의 피해는 돈과 관련되어 있기 때문에 더 강력한 보안이 요구되고 있으며 OTP 토큰 및 카드, 휴대폰 등 작은 단말기에서 신속한 패스워드 생성과 보안을 위해서 경량화에 대해서도 요구되고 있다. 따라서 향후 SHA-1뿐만 아니라 MD4, MD5, RIPEMD, AES 등 많은 알고리즘을 통해서 효율성 및 보안의 가장 최고의 적정선을 찾는 것이 필요하다고 사료된다.

참고문헌

[1] Y.F Chang, C.C Chang, J.Y Kuo, "A Secure One-time Password Authentication Scheme using Smart Cards without Limiting Login Times", Operating Systems Review, Vol. 38, No. 4, 2004

[2] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O.Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm" RFC 4226, 2005.12

[3] L.Lamport, "Password Authentication with Insecure Communication", Commun. ACM, Vol. 24, No. 11, pp.770-772, 1981

[4] N. Haller, "The S/KEY One-time Password System" RFC 1760. 1995.02

[5] N.Y Lee, J.C Chen, "Improvement of One-time Password Authentication Scheme Using Smart Cards," IEICE Trans. Commun, Vol. E88-B, No. 9, 2005.09

[6] Mitchell, C.J., Chen, L., "Comments on the S/KEY User Authentication Scheme," ACM Operating Systems Review, Vol. 30. No. 4, 1996

[7] T.C Yeh, H.Y Shen, J.J Hwang, "A Secure One-time Password Authentication Scheme Using Smart Cards," IEICE Trans. Commun, Vol. E85-B, No. 11, 2002.11

[8] M. Sandirigama, A.Shimizu, M.T. Noda, " Simple and Secure Password Authentication Protocol(SAS)", IEICE Trans. Commun, Vol. E83-B, No. 6, 2000.06