Journal of Korea Multimedia Society Vol. 14, No. 6, June 2011(pp. 785-793)

모바일 환경에서 안전한 One-Time Password 인증 기법에 관한 연구

김홍기[†], 이임영[††]

요 약

컴퓨팅 환경이 발전됨에 따라 인터넷을 이용한 다양한 서비스들이 등장하게 되었고, 서비스를 이용하기 위하여 정당한 사용자를 확인하는 인증기술의 중요성이 증대되고 있다. 인증 기술의 발달로 한 세션에서 패스워드 값을 사용 후 폐기하는 일회용 패스워드에 관한 연구가 활발히 진행되고 있다. 그러나 기존의 일회용 패스워드는 생성한 테이블의 일회용 패스워드 값들이 순차적으로 저장되어 있어 Seed값과 일회용 패스워드의 사용 횟수가 노출되면 다음 사용될 일회용 패스워드의 값을 유추가능한 문제점이 있다. 또한 S/Key 방식의 일회용 패스워드는 사용 횟수가 고정되어있다는 문제점이 있다. 따라서 본 논문에서는 기존 일회용 패스워드의 문제점을 분석하고 이를 해결하기 위하여 타원곡선 암호 알고리즘을 이용한 일회용 패스워드 생성방식과 시간 값을 이용하여 임의성을 증가시킨 일회용 패스워드 생성방식에 대하여 제안하였다.

A Study on One-Time Password Authentication Scheme in Mobile Environment

Hong Gi Kim[†], Im Yeong Lee[††]

ABSTRACT

Since then, with the advance of computing environment, various Internet services are emerging and the importance of user authentication technology is increasing for verifying users authorized to use such services. Along with the advance of authentication technology, research is being made actively on one time password, which is used once in a session and then discarded. In existing one time passwords, however, the values of one time passwords in a created table are stored in serial order, and therefore, if the seed value and the number of one time passwords used are disclosed, one may infer the value of the one time password to be used next. What is more, one time passwords of the S/Key type have the problem that the number of uses is fixed. In this paper, We analysis the existing one time password. Also, We propose one time password methods using elliptic curve cryptography scheme and using enhanced randomness with time value.

Key words: Mobile OTP(모바일 OTP), One-Time Password(일회용 패스워드), OTP, S/Key, User Authentication(사용자 인증)

※ 교신저자(Corresponding Author) : 이임영, 주소 : 충남 아산시 신창면 읍내리 (336-745), 전화 : 041)542-8819, FAX : 041)530-1548, E-mail : imylee@sch.ac.kr
접수일 : 2010년 12월 30일, 수정일 : 2011년 4월 8일
완료일 : 2011년 5월 16일

[†] 준회원, 순천향대학교 컴퓨터학과
(E-mail : hgkim31@sch.ac.kr)
[††] 종신회원, 순천향대학교 컴퓨터소프트웨어공학과
(E-mail : imylee@sch.ac.kr)

1. 서 론

컴퓨팅 환경이 발전됨에 따라 인터넷을 이용한 다양한 서비스가 등장하게 되었다. 다수의 사용자가 동시에 서비스를 이용함에 있어 정당한 사용자를 판단하는 인증기술이 발달하게 되었고, 공격자는 인증기술의 취약점을 이용하여 정당한 사용자를 위장하는 공격이 가능함에 따라 인증기술의 중요성이 증대되고 있다. 사용자 인증기술은 인증의 기반이 되는 요소에 따라 지식을 통한 인증, 소유한 물건을 이용한 인증, 신체적 특징을 이용한 인증으로 구분된다.

대표적인 사용자 인증기술로 일반적인 패스워드 인증방식이 있다. 패스워드 인증방식은 지식을 통한 인증을 기반으로 사용법이 간단하여 편리하게 이용할 수 있는 장점이 있다. 그러나 패스워드가 고정적인 값으로 서버에 전송되어 공격자가 해킹을 통해 사용자의 패스워드를 취득하여, 사용자가 직접 패스워드를 변경 시까지 취득한 패스워드를 사용할 수 있는 문제점이 발생한다. 또한 일반적인 사용자들은 자신이 기억하기 쉬운 문자, 숫자들로 패스워드를 구성하고 있기 때문에 사회 공학적 공격으로 인한 패스워드 노출이 가능하다[1].

이에 따라 기존의 패스워드 인증방식의 문제점을 개선하기 위하여 일회용 패스워드 방식이 제시되었다. 일회용 패스워드 방식은 클라이언트가 서버로 전송하는 패스워드의 값을 한 세션의 통신에서 일회용 패스워드를 사용 후 폐기한다. 따라서 일회용 패스워드가 노출된다고 하더라도 한번 사용 후 다른 값을 생성하기 때문에 공격자가 이전 패스워드를 이용하여 인증 받을 수 없다.

일회용 패스워드는 인증서버의 동기화 여부에 따라 비 동기화(Asynchronous) 방식과 동기화(Synchronous)방식으로 나눌 수 있는데, 비 동기화 방식은 사용자가 직접 임의의 난수 값을 OTP토큰에 입력함으로써 OTP값이 생성되는 방식이다. 서로 질의 값과 응답 값을 주고받기 때문에 상호인증이 제공된다는 장점이 있으나, 매번 사용자가 질의 값을 입력해야하기 때문에 불편함을 초래하며 네트워크 부하가 증가될 수 있다. 동기화 방식의 경우 시간 동기화와 이벤트 동기화로 구분된다. 동기화 방식은 별도의 질의 값 입력 없이 사용이 가능하지만 서로 동기화 시간이 맞지 않는 경우 인증을 받지 못하고, 동기화 시간동안 계속 같은 값이 요구되기 때문에 재사용 문제가 발생할 수 있다[2].

본 논문의 구성은 다음과 같다. 2장에서는 모바일 일회용 패스워드의 보안위협과 보안 요구사항을 분석하고, 3장에서는 일회용 패스워드의 생성방식과 기존 일회용 패스워드의 생성방식에 대하여 기술한다. 4장에서는 암호알고리즘을 이용한 일회용 패스워드 인증 방식과 S/Key를 기반으로 한 시간값을 이용한 일회용 패스워드 방식에 대하여 제안하고, 5장에서는 보안 요구사항에 의하여 제안방식을 분석한다. 마지막으로 6장에서 결론을 맺는다.

2. 연구배경

이 장은 일회용 패스워드의 보안 위협과 보안요구사항에 대해 알아보고 모바일 OTP 생성방식에 대하여 분석한다.

2.1 보안위협

인터넷 환경에서는 해킹, 악성코드인 웜 그리고 바이러스들과 같은 다양한 위험요소에 노출되어 있다. 안전한 일회용 패스워드 인증기법을 설계하기 위해서는 다음의 공격유형들이 고려되어야 한다[3].

- 추측공격 : 토큰정보를 찾아내기 위해서 반복적인 검증시도를 통해 일회용 패스워드를 유도해낸다.
- 탈취공격 : 인증요청을 성공시키기 위해서 전송 토큰정보의 도청 및 탈취에 의해 불법인증을 시도한다.
- 피싱공격 : 사용자가 실제 사이트로 오인하도록 유도한 뒤 입력한 일회용 패스워드를 탈취한다.
- 위장공격 : 토큰을 원 소유자의 것처럼 위장 등록하여 향후 전송되는 인증정보를 가로채어 불법 인증 한다.
- 중간자공격 : 인증시스템에게 사용자인 것처럼 위장하거나, 인증시스템으로 위장하여 토큰 정보를 가로채 인증 정보를 교체한다.

2.2 보안요구사항

일회용 패스워드는 빠른 속도를 기반으로 강력한 안전성이 보장되어야 한다. 이에 따라 일회용 패스워드에서의 보안 요구사항은 다음과 같다.

- 기밀성 : 통신에 사용되는 데이터들은 정당한 통신 객체들만이 공유되어야 하며 통신 중간에 노출되더라도 데이터의 값을 유추하지 못해야 한다.
- 무결성 : 통신상에서 전송되는 데이터들은 통신 중 위조 및 변조되지 않아야 한다.
- 연산량 : 빠른 속도로 인증과정을 수행해야 하기 때문에 연산 효율성이 높아야 한다.
- 동기화 : OTP를 생성하기 위하여 입력 값으로 사용되는 시간 및 이벤트 값을 동기화되어 있어야 하며 전송 중 비동기화가 발생하지 않아야 한다.

2.3 모바일 OTP

국내의 전자금융에서는 이미 H/W 기반의 일회용 패스워드가 1등급 보안매체로 활발히 사용되고 있다. 그러나 H/W 기반의 카드형 OTP는 가격이 상대적으로 고가이고, 토큰형 OTP는 사용자가 항상 소지하고 있어야 하는 불편함이 존재한다. 이러한 이유로 H/W 기반의 OTP보다 상대적으로 저렴하며 휴대가 편리한 모바일 기반의 일회용 패스워드 생성기법의 도입 요구가 증가하게 되었다.

이처럼 모바일 OTP에 대한 요구가 증가하고 있지만 아직 보안성이 확보되지 않았고, S/W기반의 모바일 OTP의 경우는 현실적으로 전자적 침해 위협과 복제 위협에 완벽히 대응하기 어렵기 때문에 이를 보완하기 위한 다양한 연구가 진행되고 있다[4].

3. 관련연구

3.1 OTP 생성방식

일회용 패스워드의 생성방식은 질의응답(Challenge-Response) 방식, 시간동기화(Time-Synchronous) 방식, 이벤트동기화(Event-Synchronous) 방식 그리고 이를 혼합하여 생성하는 조합방식이 있다. 기본적으로 일회용 패스워드 인증과정은 클라이언트가 인증서버와 공유하고 있는 시간 값 및 Seed값을 일회용 패스워드 생성 알고리즘을 통해 OTP값을 생성하고 이를 인증서버에 전송하여 서버가 전송한 OTP값과 클라이언트가 전송한 OTP값을 서로 비교하여 인증과정을 수행한다[5].

질의응답 방식은 서버가 제시한 질의 값을 사용자가 알고리즘에 입력하여 응답 값을 얻고 해당 응답 값을 서버에 전송하여 자신을 인증하는 방식으로 비동기화 방식에 해당된다. 인증 서버간에 미리 설정되어 있는 동기화 기준정보가 없어 인증요청 시 사용자가 직접 임의의 난수 값을 클라이언트에 이를 통해 OTP값을 생성한다. 질의응답 방식은 동기화할 기준정보가 없기 때문에 따로 동기화할 필요가 없으며, 사용자와 서버간에 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 클라이언트에 입력하여야 한다는 단점을 가지고 있다.

시간동기화 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 OTP값이 변화되어 시간 내에 입력하지 못하면 인증 재 시도를 위해 기다려야 하는 불편함이 존재한다.

이벤트 동기화 방식은 서버와 클라이언트가 동일한 카운트 값을 기준으로 비밀번호를 생성하는 방식으로 OTP 생성 요청을 받은 기준점으로 재 요청 시까지 인증 값이 변하지 않기 때문에 사용자는 편리하게 OTP값을 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 OTP값을 획득 했을 경우 정상적인 사용자가 인증요청을 수행하기 전까지 OTP값이 변화되지 않기 때문에 공격자가 획득한 OTP값을 사용할 수 있다는 단점이 존재한다[6].

따라서 이와 같은 문제점을 해결하기 위해 각 생성방식의 장점을 조합한 조합방식을 사용하고 있다. 본 논문에서는 비 동기화 방식인 질의응답방식과 시간동기화 방식을 조합한 방식을 사용하여 중간자 공격과 위장공격에 대응하는 프로토콜을 제안하였다.

3.2 S/Key

RFC 1760 표준인 S/Key 인증방식에서는 해쉬 알고리즘인 SHA-1을 이용하여 일회용 패스워드를 생성한다. S/Key 방식은 그림 1과 같이 사용자의 패스워드와 서버에서 생성한 난수 Seed를 XOR연산과 해

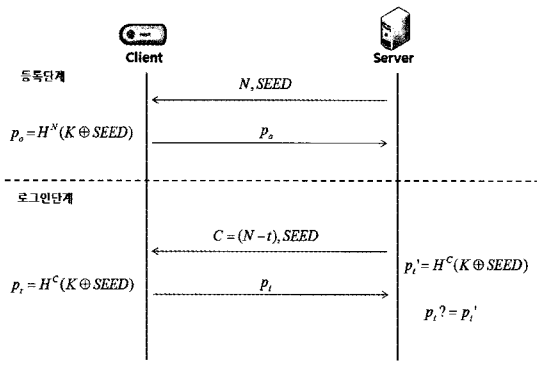


그림 1. S/Key 인증방식

쉬 연산을 이용하여 일회용 패스워드를 생성하고 있다. 또한 서버 데이터베이스에 해쉬 체인을 이용한 OTP 생성 값이 저장되어 있어, 추가적인 인증 요구 시 빠른 속도를 제공하고 있다.

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 공격자는 쉽게 다음 일회용 패스워드 값을 유추할 수 있다[7,8].

3.3 대칭키 암호 알고리즘을 이용한 방식

본 방식은 기존의 S/Key방식의 문제점을 보완하여 대칭 키 암호 알고리즘을 이용한 일회용 패스워드 인증 기술에 관하여 제안하였다. 그림 2와 같이 Xn+1의 형태로 일회용 패스워드가 생성되기 때문에 사용 횟수에 대한 제한이 없다. 또한 암호 알고리즘의 안전성을 기반으로 일회용 패스워드가 생성되기 때문에 기밀성 및 무결성이 제공된다.

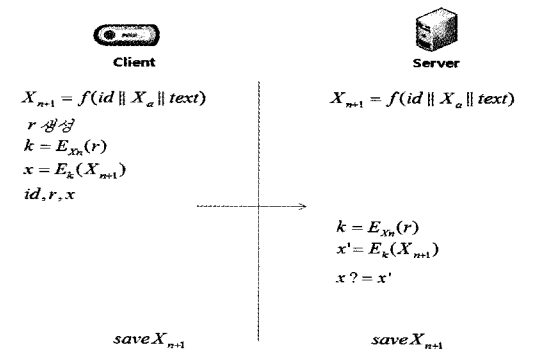


그림 2. 대칭키 암호 알고리즘을 이용한 OTP 인증방식

그러나 대칭키 암호 알고리즘은 서버와 클라이언트가 서로 같은 키를 공유하기 위해서 키 교환 알고리즘을 이용해야 하는데 매번 새로운 일회용 패스워드를 생성하기 위하여 공유키를 생성해야 하는 단점이 있다[9].

3.4 타원곡선 암호 알고리즘

타원곡선 암호 알고리즘은 1985년 N.Koblitz와 V.Miller에 의해 처음 제안되었다. 유한체 위에서 정의 된 타원곡선 군에서의 이산대수 문제에 기초하여, 상호간에 공유된 G값을 통해 공개키를 계산하고, 이를 통해 암호화에 적용한다. 타원곡선의 이산대수 문제는 타원곡선 E가 정의되어 있을 때, 정수 a배의 관계에 있는 타원곡선위의 두 점 P, Q를 모두 안다고 하더라도 정수 a를 계산하는 것이 어렵다는 점을 이용한 것이다.

$$a*P=Q \ (P,Q \in E(Fq))$$

타원곡선 암호 알고리즘은 기 존재하는 다른 공개키 암호 알고리즘보다 적은 키 길이를 통해 같은 안전도를 제공하고 있다. 짧은 키 길이를 가진다는 것은 암호시스템에서 필요한 대역폭과 메모리가 작아짐을 의미하고 있으므로 메모리 처리능력이 제한된 스마트카드 및 모바일 시스템의 응용에 적용 가능하다[10].

본 논문에서는 ElGamal 타원곡선 암호 알고리즘을 이용하여 생성된 일회용 패스워드를 안전하게 전송하는 기법에 대하여 제안하였다.

4. 제안 방식

이 장에서는 3장의 보안요구사항을 만족하는 일회용 패스워드 생성기법을 제안한다.

4.1 시스템 계수

- $*$: 각각의 개체 (C : 클라이언트, S : 서버)
- $e*$: $*$의 개인 키
- $e*G$: $*$의 공개 키
- ks : 공유키
- $k*$: 각각의 개체 $*$가 선택한 임의의 정수
- G : 타원 곡선상의 점
- $Seed$: 서버에서 생성한 난수 값

- MSN : 모바일 시리얼 넘버
- $MPwd$: 모바일에서 사용자가 입력한 패스워드
- n : OTP 생성 횟수
- N : 전체 로그인 횟수
- C : C번째 로그인 횟수
- Ts : 서버와 클라이언트의 시간 값
- $H()$: 일방향 해쉬함수
- $E*[\,]$: $*$의 키를 이용한 대칭키 암호화

4.2 타원곡선 암호 알고리즘을 이용한 방식

기존 일회용 패스워드 프로토콜은 해쉬 알고리즘을 기반으로 구성되어 있어 안전성에 문제가 있고, 암호 알고리즘을 사용하는 방식의 경우 키 교환의 문제가 있다. 따라서 공개키 암호 알고리즘의 안전성을 기반으로 별도의 키 교환 없이 안전하게 일회용 패스워드를 생성하는 방식을 제안한다. 본 제안방식은 등록단계, 로그인 단계, 재사용 단계로 구분되며 각 단계의 수행절차는 다음과 같다.

4.2.1 등록 및 로그인 단계

등록 단계에서는 인증 받는 모바일 사용자의 패스워드를 서버에게 타원곡선 암호 알고리즘을 이용하여 암호화 후 전송한다. 서버에서는 이를 복호화하여 시리얼 넘버와 복호화에 사용된 k_ce_sG값을 저장하고, 생성될 일회용 패스워드의 인증 값으로 사용한다.

디바이스의 등록을 마치면 일회용 패스워드를 생성하고 서버에서 인증하는 로그인 단계를 수행한다. 로그인 단계에서는 서버에서 생성한 난수 값과 시간 값을 이용하여 일회용 패스워드를 생성한다. 생성된 일회용 패스워드를 서버에 전송하여 디바이스에 대한 인증을 수행하게 된다. 등록 및 로그인 단계는 다음과 같다.

Step 1 : 서버와 클라이언트는 사전 타원곡선과 타원곡선상의 점 G를 공유하고, 타원곡선 집합군 Z_p 상에서 자신의 개인키로 사용할 e_c와 e_s를 선택한다. 선택한 개인키와 타원곡선상의 점 G를 곱셈 연산하여 공개키 e_cG와 e_sG를 생성한다. 생성 후 서버와 클라이언트에서는 임의의 정수 k_c와 k_s를 선택하여 암호화 및 복호화에 사용한다.

$C : e_c \in Z_p,\ e_cG,\ k_c$

$S : e_s \in Z_p,\ e_sG,\ k_s$

Step 2 : 클라이언트에서는 디바이스의 시리얼 넘버를 받아 클라이언트의 임의 정수 k_c와 타원곡선상의 점 G를 곱셈 연산하여 k_cG를 생성하고 클라이언트의 임의 정수 k_c, 서버의 공개키 e_sG를 곱셈 연산한 결과와 입력한 패스워드 $MPwd$를 더해 암호화 후 서버로 전송한다.

$C \rightarrow S : \{k_cG,\ MPwd + k_ce_sG\}$

Step 3 : 서버에서는 전송받은 암호문 중 k_cG에 서버의 개인키 e_s를 연산하여 e_sk_cG를 생성하고 이를 k_ce_sG와 감산하여 복호화 한다. 복호화 후 결과 값인 $MPwd$과 복호화 시 사용된 k_ce_sG를 서버에 등록한다.

$S : MPwd + k_ce_sG - e_sk_cG = MPwd$

$S : Store\ MPwd,\ k_ce_sG$

Step 4 : 서버에서는 난수 $Seed$와 SN이 확인된 서버의 마지막 시간 값 T_s를 해쉬연산 한 결과값 v를 클라이언트에게 암호화하여 전송한다.

$S : v = H(Seed \,||\, T_s)$

$S \rightarrow C : \{k_sG,\ v+k_se_cG\}$

Step 5 : 클라이언트는 복호화 된 v의 값과 패스워드와의 XOR연산과 해쉬연산을 통해 OTP_n값을 생성하고, 서버도 같은 해쉬연산을 수행하여 OTP'_n을 생성한다.

$C : v + k_se_cG - e_ck_sG$

$C : OTP_n = H(v \oplus MPwd)$

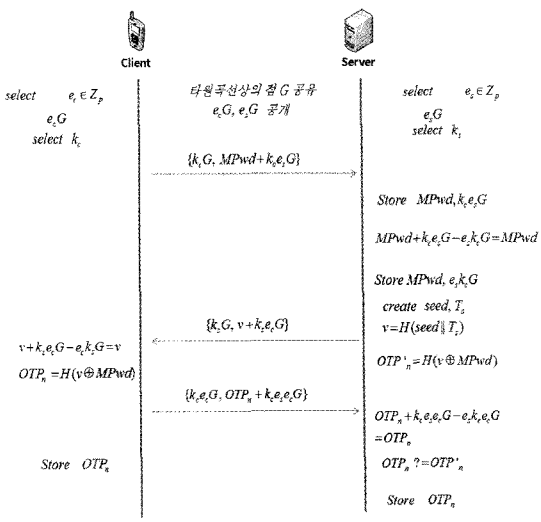


그림 3. 등록 및 로그인 단계

S : $OTP'_n = H(v \oplus MPwd)$

Step 6 : 생성한 OTP값을 서버로 전송하기 위하여 클라이언트의 임의의 정수 k_c와 클라이언트의 개인키를 타원곡선상의 한 점 G와 연산하여 k_ce_cG를 생성하고, OTP의 값과 임의의 정수 k_c, 서버의 공개키 e_sG, 클라이언트의 개인키 e_c를 연산하여 $OTP_n+k_ce_se_cG$를 생성 후 서버에 전송한다.

$C \rightarrow S$: $\{k_ce_cG,\ OTP_n+k_ce_se_cG\}$

Step 7 : 서버에서는 전송받은 OTP_n값과 서버에서 생성한 OTP'_n값을 비교하여 인증을 완료한다. 인증 완료 후 클라이언트와 서버는 생성한 OTP_n값을 저장한다.

S : $OTP_n + k_ce_se_cG - e_sk_ce_cG$

S : $OTP'_n =? OTP_n$

C, S : $Store\ OTP_n$

4.2.2 재사용 단계

로그인 단계를 수행하면 OTP값이 서버와 클라이언트에 저장되는데 재사용 단계에서는 이를 시간 값과 연산하여 추가적인 $Seed$의 생성 없이 이전의 OTP값을 이용하여 새로운 일회용 패스워드를 생성한다.

Step 1 : 클라이언트에서 패스워드를 서버의 공개키 e_sG로 암호화하여 전송한다. 서버에서는 이를 복호화 하여 저장되어 있는 시리얼 넘버와 복호화 시 사용된 k_ce_sG를 비교하여 클라이언트를 인증한다.

$C \rightarrow S$: $\{k_c,G,\ MPwd' + k_ce_sG\}$

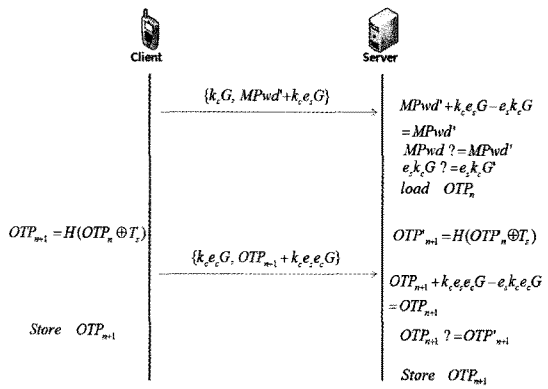


그림 4. 재사용단계

S : $MPwd' + k_ce_sG - e_sk_cG$

S : $MPwd =? MPwd',\ k_ce_sG' ?= k_ce_sG$

Step 2 : 서버와 클라이언트는 T_s를 확인 후 상호간에 동기화된 시간 값 T_s를 이용하여 OTP_{n+1}값과 OTP'_{n+1}을 생성한다.

C : $OTP_{n+1} = H(OTP_n \oplus T_s)$

S : $OTP'_{n+1} = H(OTP'_n \oplus T_s)$

Step 3 : 클라이언트에서는 생성한 OTP_{n+1}값을 서버로 암호화하여 전송한다.

$C \rightarrow S$: $\{k_ce_cG,\ OTP_{n+1}+k_ce_se_cG\}$

Step 4 : 서버에서는 k_ce_cG의 값에 자신의 개인키 e_s를 연산하여 $e_sk_ce_cG$를 생성하고 이를 통해 복호화하여 새로운 비밀번호 OTP_{n+1}을 서버에 저장된 OTP'_{n+1}과 비교한 후 인증한다.

S : $OTP_{n+1} + k_ce_se_cG - e_sk_ce_cG$

S : $OTP'_{n+1} =? OTP_{n+1}$

4.3 시간 값을 통해 임의성을 강화한 방식

기존 일회용 패스워드 생성방식은 초기화한 테이블의 값이 순차적으로 기록되어있어 Seed값 노출 시 모든 해쉬 테이블이 노출당하는 문제점이 있다. 또한 초기 Seed값 전송 시 별도의 암호화 단계를 수행하지 않아 그 위험도는 매우 크다. 따라서 본 제안방식은 시간 값을 이용하여 생성된 해쉬 테이블을 임의적으로 사용하는 방식에 대하여 제안한다. 본 제안방식은 등록단계 인증 및 로그인 단계로 구분되며 각 단계의 수행절차는 다음과 같다.

4.3.1 등록단계

등록단계에서는 기존 S/Key방식과 동일한 방법으로 서버에 일회용 패스워드 테이블을 등록한다. 서버는 전체 로그인 횟수 N과 초기값 $Seed$를 클라이언트에게 전송해 주고, 클라이언트는 전송받은 $Seed$값과 사전에 공유되어있는 서버와의 공유키를 통하여 XOR연산 수행 후 해쉬 연산을 수행한다.

해쉬 연산 후 생성된 결과값을 공유키로 암호화하고 서버에게 전송하여 생성된 결과값을 통해 인증받게 된다. 등록단계의 수행은 다음과 같다.

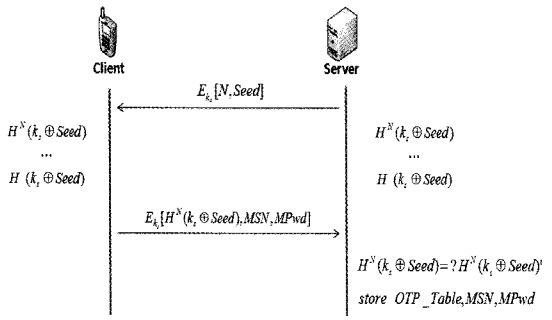


그림 5. 등록 단계

Step 1 : 서버에서는 생성한 *Seed*값과 전체로그인 횟수 *N*을 공유키 *ks*를 통해 암호화하여 클라이언트에게 전송해 준다.

$S \rightarrow C : E_{ks}[N \,||\, Seed]$

Step 2 : 서버와 클라이언트는 *Seed*값과 공유키 *ks*를 XOR연산하여 전체 로그인 횟수 *N*만큼 해쉬 연산 한 일회용 패스워드 테이블을 생성한다.

$C : H^N(ks \oplus Seed) \cdots H(ks \oplus Seed)$

$S : H^{N'}(ks \oplus Seed) \cdots H(ks \oplus Seed)$

Step 3 : 클라이언트에서는 전체 로그인 횟수 *N*만큼 해쉬 연산한 $H^N(ks \oplus Seed)$와 MSN, MPwd를 공유키 *ks*로 암호화하여 서버에게 전송하고, 서버에서는 서버에서 생성한 $H^{N'}(ks \oplus Seed)$값과 전송받은 값을 비교하여 서버에 일회용 패스워드 전체 테이블을 저장한다.

$C : E_{ks}[H^N(ks \oplus Seed),\ MSN,\ MPwd]$

$S : H^N(ks \oplus Seed) =? H^{N'}(ks \oplus Seed)$

$S :$ *Store OTP_Table, MSN, MPwd*

4.3.2 인증 및 로그인 단계

등록단계를 수행하면 서버와 클라이언트는 같은 일회용 패스워드 테이블을 소유하게 된다. 등록 완료 후 인증 단계를 수행하게 되는데, 클라이언트는 일회용 패스워드 요청 메시지인 *OTP_request*와 *MPwd*를 서버에 전송하여 비교 후 시간 동기화 값 *Ts*를 생성하고 *MSN*과 XOR연산을 수행하여 *υ*값을 생성한다. 이를 전체 로그인 횟수 *N*과 나머지 연산을 수행하여 *P*값을 생성한다. 생성한 *P*값을 통해 사전에 생성한 일회용 패스워드 테이블에서 해당 값을 일회용 패스워드로 사용한다.

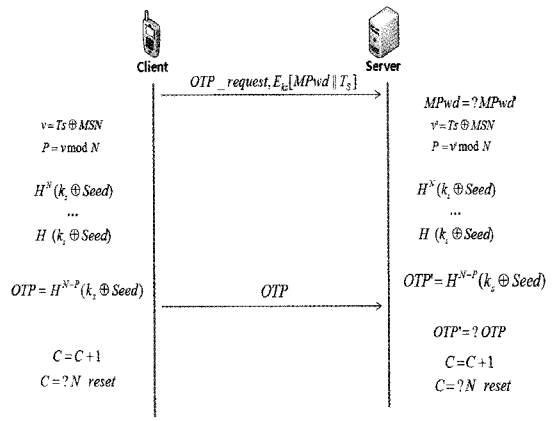


그림 6. 인증 및 로그인 단계

생성된 *OTP*값을 공유키로 암호화하여 서버에 전송하고, 서버에서는 이를 복호화하여 서버의 *OTP*값과 비교하여 인증을 수행한다. 인증단계는 다음과 같은 과정으로 진행된다.

Step1 : 클라이언트는 서버에게 일회용 패스워드를 요청하는 메시지를 전송한다.

$C \rightarrow S : OTP\_request,\ E_{ks}[MPwd]$

$S : MPwd =? MPwd'$

$S : \upsilon' = MSN' \oplus Ts$

$C : \upsilon = MSN \oplus Ts$

Step2 : 서버와 클라이언트는 생성한 *υ*값을 전체 로그인 횟수인 *N*의 값으로 나머지 연산을 통해 *P*와 P′ 값을 생성한다.

$C : P = \upsilon \bmod N$

$S : P' = \upsilon' \bmod N$

Step3 : 서버와 클라이언트는 전체 로그인 횟수인 *N*과 *Step2*에서 계산한 *P*의 값을 통해 일회용 패스워드 테이블 내 *OTP*값을 사용한다.

$C : OTP = H^{H-P}(ks \oplus Seed)$

$S : OTP' = H^{H-P}(ks \oplus Seed)$

Step4 : 클라이언트는 생성한 *OTP*값을 서버에 전송한다. 서버에서는 이를 서버에서 생성한 *OTP′* 값과 비교하여 인증을 한다.

$C \rightarrow S : OTP$

$S : OTP' =? OTP$

Step5 : 서버와 클라이언트는 일회용 패스워드 인증 후 카운터 값에 1을 더하여 카운터 C값과 전체 로그인 횟수인 N값과 비교하여 같으면 등록과정을 다시 수행하게 된다.

C, S : C = C + 1

C, S : C=?N, OTP_Table reset

5. 제안방식 분석

보안요구사항에 대한 제안방식 분석은 다음 표 1과 같다.

S/Key방식은 해쉬 알고리즘인 SHA-1의 안전성을 기반으로 일회용 패스워드를 제공하고 있으나 최근 해쉬 함수의 위험성이 보고되면서 안전성을 보장받지 못하고 있다. 또한 *C*값과 *Seed*가 질의 값으로 사용되고 있기 때문에 공격자는 이를 중간자 공격으로 획득하여 서버를 위장하면서 유효한 OTP값의 취득이 가능하다.

대칭키 방식의 경우 비트연산을 수행하여 적은 연산량을 가지고 있고, 암호알고리즘 기반의 안전성을 보장하고 있지만, 별도의 세션키 교환단계가 필요하며, 난수 *r*을 전송할 때 평문으로 전송하고 있기 때문에 *r*이 노출되면 일회용 패스워드의 생성 패턴이 노출될 위험이 있다.

따라서 제안방식1의 경우는 암호알고리즘을 기반으로 안전성을 보장하며 공개키 알고리즘을 이용해 별도의 세션키 교환단계가 없이 서버의 인증 및 OTP값의 교환이 가능하며, 중간자 공격을 방어할 수 있다. 또한 타원곡선 암호 알고리즘의 사용으로 지수승 연산으로 구성된 공개키 암호 알고리즘의 연산량 문제를 해결하였다. 또한 재사용 단계에서 동기화된 시간값을 통해 별도의 값 교환없이 일회용 패스워드를 생성하여 통신량이 감소되는 효과를 얻을 수 있다.

제안방식2의 경우 S/Key방식으로 생성한 해쉬 테이블의 내용을 시간 값을 활용하여 임의적으로 생성하는 방식을 적용하였으며, 중간자 공격을 통해 *Seed* 값이 노출되어도 일회용 패스워드 생성 패턴이 노출되지 않아 공격자는 다음번에 사용할 일회용 패스워드 순서의 추측이 불가능하다. 또한 세션키를 상호간에 공유하여야 하는 불편함이 있지만 모바일 환경에서 적은 연산량을 통해 안전성을 보장할 수 있는 대칭키 암호 알고리즘을 사용하여 안전하게 자신의 개인 모바일 정보를 보호하며 일회용 패스워드를 생성할 수 있다. 연산량 측면에서 암호알고리즘을 이용한 기존의 일회용 패스워드 생성방식보다 낮은 연산량을 통해 안전성을 제공한다.

6. 결 론

IT 기술의 발달로 개인정보를 이용한 다양한 서비스들이 등장하게 되었고, 이에 따라 개인정보를 보호

표 1. 제안방식 분석

구 분		S/Key [7]	대칭키 [9]	제안방식 1	제안방식 2
기 밀 성		×	○	○	○
		Seed값 평문전송으로 노출위험 증대	암호화를 통한 기밀성 제공	암호화를 통한 기밀성 제공	암호화를 통한 기밀성 제공
무 결 성		○	○	○	○
		제공	제공	제공	제공
연 산 량		○	△	△	△
		H	4E + H	4E + H	E + H + M
위장공격		×	×	○	○
		Seed값을 통한 위장공격 가능	id값의 평문전송으로 인해 위장공격 가능	개인키를 통한 위장공격 대응	시간값을 통한 OTP값의 임의사용으로 위장공격 대응
통신량	등록	2-pass	-	1-pass	2-pass
	인증	3-pass	2-pass	2-pass	2-pass

○: 좋음, 제공 △: 보통, 부분제공 ×: 나쁨, 제공안함
H(해쉬연산량), E(대칭키 연산량), M(모듈러 연산량)

하는 다양한 인증기술이 등장하게 되었다. 기존의 패스워드 인증기술은 평문으로 패스워드가 전송되어 공격자에 의해 노출될 가능성이 매우 높다. 이를 보완하기 위해 일회용 패스워드가 등장하였지만, *Seed* 값이 노출되면 생성한 일회용 패스워드 테이블의 유추가 가능하다. 또한 이를 방지하기 위하여 매 세션 일회용 패스워드를 생성하는 암호 알고리즘을 적용하는 방식은 연산량의 문제로 인해 대칭키 암호 알고리즘을 이용하고 있으나, 매번 세션키를 공유하여야 한다는 문제점이 존재하고 있다.

이에 본 논문은 세션키의 교환단계가 없이 기존에 생성된 공개키를 이용하여 일회용 패스워드를 생성하는 방식과 시간 값을 이용하여 해쉬 테이블의 패턴이 노출되지 않도록 하는 방식을 제안하였다. 따라서 *Seed*값이 노출되어도 현재 사용하고 있는 테이블을 유추할 수 없어 안전하게 일회용 패스워드를 사용할 수 있다. 따라서 모바일 환경에서 OTP를 이용한 안전한 인증이 제공될 수 있을 것이라 사료된다.

참 고 문 헌

[1] S. D. Park, J. C. Na, Y. H Kim, and D. K Kim, "Efficient OTP(One Time Password) Generation using AES-based MAC," 한국멀티미디어학회논문지, V.11, No.6, pp. 845-851, 2008.

[2] 문용혁, 권혁찬, 나재훈, 장종수, "P2P 사용자 인증과 OTP분석," 정보보호학회지, 제17권, 제3호, pp. 32-40 2007.

[3] J. Archer Harris, "OPA : A One-Time Password System," *10.1109 /ICPPW. 2002, 1039708,* 2002.

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

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

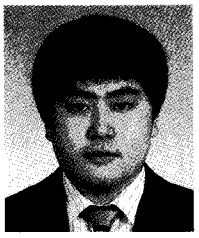
[6] 서승현, 강우진, "OTP 기술현황 및 국내 금융권 OTP 도입사례," 정보보호학회지, 제17권, 제3호, pp. 18-25, 2007.

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

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

[9] 박중길, 김영진, 김영길, 백규태, 백기영, 류재철, "S/Key를 개선한 일회용 패스워드 메커니즘 개발," 정보보호논문지, Vol.9, No.2. 1999.

[10] 양대헌, 송주석, "타원 곡선을 이용한 암호 시스템," 정보보호학회지, 제7권, 제4호, pp. 5-12, 1997.



김 홍 기

2010년 순천향대학교 정보기술공학부 학사

2010년~현재 순천향대학교 컴퓨터학과 석사과정

관심분야 : 컴퓨터 보안, One-Time Password, 스마트그리드

이 임 영

1981년 홍익대학교 전자공학과 졸업

1986년 오사카대학 통신공학전공 석사

1989년 오사카대학 통신공학전공 박사

1989년~1994년 한국전자통신연구원 선임연구원

1994년~현재 순천향대학교 컴퓨터소프트웨어공학과 교수

관심분야 : 암호이론, 정보이론, 컴퓨터 보안