타원곡선 암호알고리즘을 이용한 일회용 패스워드 생성기법에 관한 연구1)

김홍기, 이임영
순천향대학교 컴퓨터소프트웨어공학과
e-mail:[hgkim31, imylee]@sch.ac.kr

A Study on One-Time Password Algorithm Using the Elliptic Curve Cryptography

Hong-Gi Kim, Im-Yeong Lee
Department of Computer Software Engineering, Soonchunhyang University

요 약

인터넷을 통한 IT환경의 변화로 개인정보를 이용한 다양한 서비스들이 등장하게 되었고, 이에 따라 개인정보보안의 중요성이 증대되고 있다. 정당한 사용자를 확인하는 인증기술의 발달로 한 세션에서 패스워드 값을 사용 후 폐기하는 일회용 패스워드에 관한 연구가 활발히 진행되고 있다. 그러나 기존의 일회용 패스워드는 기밀성, 상호인증, 연산량 등 여러 문제점을 내포하고 있다. 따라서 본 논문에서는 이러한 기존의 일회용 패스워드의 문제점을 분석하고 이를 해결하기 위해 타원곡선 암호 알고리즘을 이용한 일회용 패스워드 생성 기법에 관하여 제안하였다. 제안방식은 암호 알고리즘을 이용하여 기밀성을 제공하며, 서버와 클라이언트 간의 상호인증을 제공할 수 있도록 하였다.

1. 서론

인터넷을 통한 IT환경의 변화로 개인정보를 이용한 다양한 서비스들이 등장하게 되었다. 이에 따라 개인정보 보안의 중요성이 증대되었고, 정당한 사용자를 확인하는 인증기술이 개발되고 있다. 이러한 인증 기술은 현재 사용자가 통신을 통해 정보를 교환하고 인증요구자가 정당한 인가자인지 확인하는 기술을 의미한다. 사용자 인증 기술은 인증의 기반이 되는 요소가 무엇이냐에 따라 지식을 통한 인증, 소유한 물건을 이용한 인증, 신체적 특징을 이용한 인증으로 구분된다[1].

대표적인 사용자 인증기술로 일반적인 패스워드 인증방식이 있다. 패스워드 인증방식은 지식을 통한 인증을 기반으로 사용법이 간단하여 편리하게 이용할 수 있는 장점이 있다. 그러나 패스워드가 암호화되지 않은 상태로 서버에 전송되기 때문에 공격자에게 패스워드가 노출될 위험이 많다.

기존 일반적인 패스워드가 평문으로 노출되는 문제점을 보완하기 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.

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0022607)

그러나 기존의 일회용 패스워드는 사용횟수가 제한되어 있으며, 인증 값들이 평문으로 전송되는 문제점과 정당한 서버와 클라이언트를 확인하는 상호인증이 제공되지 않는 문제점이 존재하고 있다. 이를 보완하기 위하여 암호알고리즘의 안전성을 기반으로 일회용 패스워드를 생성하는 방법이 연구되었으나, 이 방식은 별도의 키 교환 알고리즘을 필요로 하게 되어 많은 통신 횟수를 요구하게 된다.

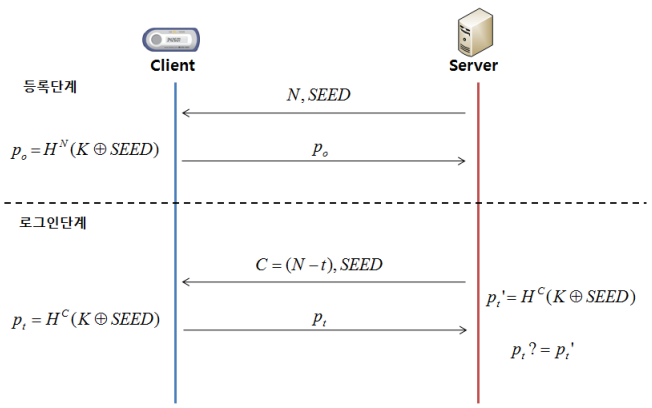
따라서 본 연구는 효율적인 키 교환과 적은 키 길이의 타원곡선 암호알고리즘을 이용하여 안전성을 보장하고 서버와 클라이언트의 상호인증을 제공하는 일회용 패스워드 생성기법에 대해 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 RFC 1760 표준 S/Key 일회용 패스워드와 암호알고리즘을 이용한 일회용 패스워드 생성방식의 내용을 알아보고, 3장에서는 관련연구를 기반으로 한 일회용 패스워드에서의 보안요구사항에 대해 분석한다. 4장에서는 보안요구사항을 만족하는 제안방식을 기술하며, 5장에서는 보안요구사항에 의하여 제안방식을 분석한다. 마지막으로 6장에서 결론을 맺는다.

2. 관련연구

2.1 S/key 일회용 패스워드 인증방식

RFC 1760 표준인 S/Key 인증방식에서는 해쉬알고리즘인 SHA-1을 이용하여 일회용 패스워드를 생성한다. S/Key방식은 사용자의 패스워드와 서버에서 생성한 난수



(그림 1) S/Key 인증방식

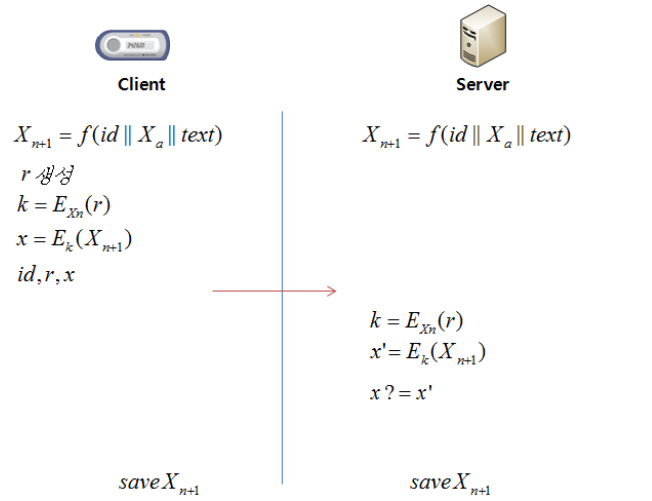
*seed*를 XOR연산과 해쉬연산을 이용하여 OTP를 생성하고 있다. 또한 서버 데이터베이스에 해쉬체인을 이용한 OTP 생성 값이 저장되어 있어, 추가적인 인증 요구 시 빠른 속도를 제공하고 있다[2].

그러나 S/Key 인증방식은 모든 값이 평문으로 전송되어 공격자에게 쉽게 노출된다는 단점을 가지고 있다. 또한 서버의 난수인 *seed*값이 동일하게 유지되고 있기 때문에 N번의 로그인 횟수가 노출되면 공격자는 쉽게 OTP값을 유추할 수 있다. 또한 사용횟수의 제한이 있어 사용횟수 초과 시에는 새로 일회용 패스워드를 생성해야 되는 단점이 있다(그림 1 참조).

2.2 대칭키 암호 알고리즘을 이용한 OTP 인증 방식

본 방식은 기존의 S/Key방식의 문제점을 보완하여 대칭키 알고리즘을 이용한 일회용 패스워드 인증기술에 관하여 제안하였다. (그림 2)와 같이 X_{n+1}의 형태로 일회용 패스워드가 생성되기 때문에 사용 횟수에 대한 제한이 없다. 또한 암호 알고리즘의 안전성을 기반으로 일회용 패스워드가 생성되기 때문에 기밀성 및 무결성이 제공된다[3].

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



(그림 2) 대칭키 암호 알고리즘을 이용한 OTP 인증 방식

3. 보안위협 및 보안요구사항

3.1 보안위협

인터넷 환경에서는 해킹, 악성코드인 웜 그리고 바이러스들과 같은 다양한 위험요소에 노출되어있다. 안전한 일회용 패스워드의 인증 기법을 설계하기 위해서는 다음의 공격유형들이 고려되어야 한다[4].

- 도청(Eavesdropping Attack) : 서버와 클라이언트 간의 통신내용을 공격자가 엿들을 수 있다.
- 재전송 공격(Replay Attack) : 공격자가 인증에 필요한 정보를 가로챈 후 서버와의 인증을 시도한다.
- 중간자 공격(Man-in-the-middle Attack) : 공격자는 인증에 필요한 정보를 가로채어 이를 위조 및 변조하여 재전송 할 수 있다.

3.2 보안요구사항

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

- 연산량 : 빠른 속도로 인증과정을 수행해야 하기 때문에 연산 효율성이 높아야 한다.
- 키 길이 : 적은 키 길이를 통해 대칭키 암호 알고리즘과 같은 안전성을 보장하여야 한다.
- 상호인증 : 정당한 서버와 클라이언트를 확인하기 위하여 서로간의 상호인증이 제공되어야 한다.
- 키 교환 : 별도의 키 교환 알고리즘 없이 서버와 클라이언트가 키를 이용하여 암호화, 복호화를 수행할 수 있어야 한다.
- 안전성 : 통신에 사용되는 데이터들은 정당한 통신 객체들만이 공유되어야 하며 위조 및 변조되지 않아야 한다.

4. 제안방식

이 장에서는 3장의 보안요구사항을 만족하는 일회용 패스워드 생성기법을 제안한다. 기존 일회용 패스워드 프로토콜은 해쉬 알고리즘을 기반으로 구성되어있어 안전성에 문제가 있고, 암호알고리즘을 사용하는 방식의 경우 키 교환의 문제가 있다. 제안방식은 공개키 알고리즘의 안전성을 기반으로 별도의 키 교환 없이 안전하게 OTP를 생성할 수 있다. 본 제안방식은 등록 및 로그인 단계, 재사용 단계로 구분되며 각 단계의 수행절차는 다음과 같다.

4.1 시스템 계수

본 제안방식에서는 다음과 같은 시스템계수를 사용하여 프로토콜을 설계한다.

- $*$: 각각의 개체 (C : 클라이언트, S : 서버)
- e_* : $*$의 개인키
- e_*G : $*$의 공개키
- k_* : 각각의 개체 $*$가 선택한 임의의 정수
- G : 타원곡선상의 점
- $seed$: 서버에서 생성한 난수 값
- SN : 디바이스의 시리얼번호
- n : OTP 생성 횟수
- OTP : $seed$와 SN의 XOR 연산 값
- T_s : SN이 확인된 서버의 마지막 시간 값
- v : $seed$와 T_s를 해쉬연산 한 결과 값
- $H()$: 일 방향 해쉬 함수

4.2 등록 및 로그인 단계

등록 단계에서는 인증 받는 디바이스의 시리얼 넘버를 서버에게 타원곡선 암호 알고리즘을 이용하여 암호화 후 전송한다. 서버에서는 이를 복호화 하여 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.

Step1 : 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.

$$C : e_c \in Z_p,\ e_cG,\ k_c$$
$$S : e_s \in Z_p,\ e_sG,\ k_s$$

Step2 : 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 디바이스의 시리얼넘버 SN을 더해 암호화 후 서버로 전송한다.

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

Step3 : 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 SN과 복호화 시 사용된 k_ce_sG를 서버에 등록한다.

$$S : SN + k_ce_sG - e_sk_cG = SN$$
$$S : Save\ SN,\ k_ce_sG$$

Step4 : 서버에서는 난수 $seed$와 SN이 확인된 서버의 마지막 시간 값 T_s를 해쉬연산 한 결과값 v를 클라이언트에게 암호화하여 전송한다.

$$S : v = H(seed \mid\mid T_s)$$
$$S \rightarrow C : \{k_sG,\ v+k_se_cG\}$$

Step5 : 클라이언트는 복호화 된 v의 값과 디바이스의 시리얼넘버와의 XOR연산과 해쉬연산을 통해 OTP_n값을 생성하고, 서버도 같은 해쉬연산을 수행하여 OTP'_n을 생성한다.

$$C : v + k_se_cG - e_ck_sG$$
$$C : OTP_n = H(v \oplus SN)$$
$$S : OTP'_n = H(v \oplus SN)$$

Step6 : 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.

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

Step7 : 서버에서는 전송받은 OTP_n값과 서버에서 생성한 OTP'_n값을 비교하여 인증을 완료한다. 인증 완료 후 클라이언트와 서버는 생성한 OTP_n값을 저장한다.

$$S : OTP_n + k_ce_se_cG - e_sk_ce_cG$$
$$S : OTP'_n =?\ OTP_n$$
$$C,\ S : Save\ OTP_n$$

4.3 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.

Step1 : 클라이언트에서 디바이스의 시리얼넘버를 서버의 공개키 e_sG로 암호화하여 전송한다. 서버에서는 이를 복호화 하여 저장되어 있는 시리얼 넘버와 복호화 시 사용된 k_ce_sG를 비교하여 클라이언트를 인증한다.

$$C \rightarrow S : \{k_cG,\ SN' + k_ce_sG\}$$
$$S : SN' + k_ce_sG - e_sk_cG$$
$$S : SN \stackrel{?}{=} SN',\ k_ce_sG' \stackrel{?}{=} k_ce_sG$$

Step2 : 서버에서는 T_s를 확인 후 클라이언트에게 전송한다. 서버와 클라이언트에서는 T_s를 이용하여 OTP_{n+1}값과 OTP'_{n+1}을 생성한다.

$$S \rightarrow C : \{k_c G,\ T_s + k_c e_s G\}$$
$$C : T_s + k_c e_s G - e_s k_c G$$
$$C : OTP_{n+1} = H(OTP_n \oplus T_s)$$
$$S : OTP'_{n+1} = H(OTP'_n \oplus T_s)$$

Step3 : 클라이언트에서는 생성한 OTP_{n+1}값을 서버로 암호화하여 전송한다.

$$C \rightarrow S : \{k_c e_c G,\ OTP_{n+1} + k_c e_s e_c G\}$$

Step4 : 서버에서는 $k_c e_c G$의 값에 자신의 개인키 e_s를 연산하여 $e_s k_c e_c G$를 생성하고 이를 통해 복호화 하여 새로운 비밀번호 OTP_{n+1}을 서버에 저장된 OTP'_{n+1}과 비교한 후 인증한다.

$$S : OTP_{n+1} + k_c e_s e_c G - e_s k_c e_c G$$
$$S : OTP'_{n+1} =? OTP_{n+1}$$

5. 제안방식 분석

보안요구사항에 대한 제안방식 분석은 다음 <표 1>과 같다.

S/Key방식은 해쉬 알고리즘인 SHA-1의 안전성을 기반으로 일회용 패스워드를 제공하고 있으나 최근 해쉬 함수의 위험성이 보고되면서 안전성을 보장받지 못하고 있다. 또한 서버와의 상호인증과정이 없기 때문에 정당한 서버 확인이 불가능하다.

대칭키 방식의 경우 비트연산을 수행하여 적은 연산량을 가지고 있고, 암호알고리즘 기반의 안전성을 보장하고 있다. 그러나 별도의 세션키 교환단계가 필요하다는 단점이 있다[5].

공개키 방식은 별도의 키 교환이 필요 없다는 장점이 있으나, 키 길이가 길고 지수승 연산으로 인해 연산량이 많아 속도가 느리다는 단점이 있다.

그러나 제안방식은 가감연산을 수행하여 타 암호 알고리즘보다 적은 연산량을 가진다는 장점이 있고, 대칭키 방식보다 길이가 긴 160bit이상의 키 길이를 가지고 있지만 공개키 방식보다 적은 키 길이를 통해 공개키 방식과 같은 안전성을 제공할 수 있다. 그리고 기기의 시리얼넘버와 OTP값을 통해 서버와 클라이언트 간의 상호인증을 제공하고 있다. 또한 공개키 알고리즘을 사용하고 있어 별도의 세션키 교환단계가 없고 이는 통신상의 패스수를 줄여주는 장점을 가지고 있다. 안전성 측면에서는 타원곡선의 성질을 기반으로 클라이언트와 서버에서 생성한 난수를 이용하여 기밀성을 보장하고 해쉬 알고리즘 연산을 통하여 무결성을 제공하고 있다.

<표 1> 제안방식 분석

구분	S/Key	대칭키	공개키	제안방식
연산량	최소 연산량	비트연산수행	제곱, 모듈러 연산	+, -를 통한 적은 연산
키 길이	키 없음	128bit 이상	1024bit 이상	160bit 이상
서버와의 상호인증	불가	불가	불가	*SN, OTP*값을 통한 상호인증 제공
키 교환	키 없음	세션키 교환 필요	없음	없음
안전성	기밀성 보장 못함	암호 알고리즘 안전성 기반	이산대수 문제 기반	타원곡선 성질 기반

6. 결론

IT 기술의 발달로 개인정보를 이용한 다양한 서비스들이 등장하게 되었고, 이에 따라 개인정보를 보호하는 다양한 인증기술이 등장하게 되었다. 대표적 인증기술인 기존의 패스워드 인증기술은 평문으로 패스워드가 전송되어 공격자에 의해 노출될 가능성이 매우 높다. 이를 보완하기 위해 일회용 패스워드가 등장하였지만, 기존의 일회용 패스워드 또한 평문으로 값이 노출되어 기밀성을 제공하고 있지 않거나 정당한 서버 및 클라이언트를 확인하는 상호인증등이 없어 많은 문제점을 내포하고 있다.

이에 본 논문에서 제안한 알고리즘은 타원곡선 암호알고리즘을 이용하여 기밀성을 제공하고 있고, 디바이스의 시리얼넘버와 일회용 패스워드 값을 이용하여 서버와 클라이언트의 상호인증을 제공하고 있어, 정당한 사용자 및 서버를 안전하게 확인 가능하다. 또한 공개키 알고리즘을 사용하여 사전에 키 공유를 제공하지 않아도 공개되어있는 공개키를 통해 암호화 및 복호화를 제공하여 통신상의 통신횟수를 줄이는 효과를 얻을 수 있다.

따라서 본 제안방식이 가지는 여러 가지 장점을 통해 보안 요구되는 사용자 및 디바이스 인증 분야에 적용될 수 있을 것이라고 사료된다.

참고문헌

[1] 문용혁, 권혁찬, 나재훈, 장종수, “P2P 사용자 인증과 OTP 분석”, 정보보호학회지 제17권 제3호, 2007.06

[2] Neil M. Haller "The S/Key One-Time Password System" RFC 1760, 1995.02

[3] 박중길, 김영진, 김영길, 백규태, 백기영, 류재철, “S/Key를 개선한 일회용 패스워드 메커니즘 개발”, 정보보호학회논문지 Vol.9 No.2, 1999.06

[4] 윤은준, 류은경, 유기영, “스마트 카드를 이용한 안전한 S/Key 일회용 패스워드 인증 스킴”, 한국정보과학회 학술발표논문집 제 30권 제 2호, 2003.10

[5] 민정기, 정상화, 김영환, 송용호, 김동규, “AES 기반의 일회용 패스워드 알고리즘”, 한국정보과학회 학술발표논문집 pp. 68~73, 2006.12