PingPong-128을 이용한 OTP 인증 프로토콜

이장춘* · 이훈재** · 임효택** · 이상곤**

OTP Authentication Protocol using PingPong-128

Jang-chun Lee* · Hoon-jae Lee** · Hyo-taek Lim** · Sang-gon Lee**

본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의
연구결과(IITA-2008-C1090-0801-0026)와 지역혁신 인력양성사업의 연구결과로 수행되었음 연구는

요 약

현재 네트워크상에서 사용자 인증은 시스템 보안에 중요한 역할을 하고 있다. 공개된 네트워크에서는 개인 정보를 보호하기 위해 여러 가지 인증 절차를 거치게 된다. 이러한 인증 방법에는 간단한 ID/Password 인증부터 복잡한 생체 공학 인증까지 다양한 기술들이 존재 한다. 최근 금융보안업계가 주축이 되어 일회용 패스워드(OTP: One Time Password) 인증 시스템을 활용하기 위한 기술적 시도 및 개발이 활발히 이루어지고 있다. 일회용 패스워드는 사용자가 인증 받고자 할 때 새로운 패스워드를 생성하고, 사용 후 폐기하는 구조를 취하고 있다. 이는 매번 같은 패스워드를 사용했을 때 발생하는 보안 문제점을 해결할 수 있다. 그러나 OTP 인증 방법에도 MITM 공격, 도청, 재전송 공격 등에 의한 취약한 문제점들이 노출되고 있다. 본 논문에서는 기존의 인증 프로토콜 문제점을 개선하고 PingPong-128을 이용한 OTP 인증 프로토콜을 제안한다.

ABSTRACT

Nowadays, authentication is essential to identify the legal users in a network communication. Usually, there are few ways to achieve authentication over a publicly accessible network system in order to protect certain private data from the unauthorized users, ranging from simple ID/Password to Biometrics System. One of the most active areas in OTP(One Time Password) research today aims at exploiting OTP to provide authentication in the finance and security industry. OTP is usually discarded once it has been used. this prevents huge loophole of traditional authentication system which employs the same ID and Password every time. However this OTP system also has its weaknesses in surviving some attacks. this paper proposes an advanced OTP protocol using PingPong-128 without loop hole of pre-existing OTP.

키워드

OTP, Authentication protocal, Stream, PingPong-128

* 동서대학교 유비쿼터스 IT 학과
** 동서대학교 컴퓨터정보공학부

접수일자 : 2008. 2. 27

Ⅰ. 서 론

최근 인터넷과 같은 통신 기술이 급속하게 발달하여 많은 부분의 업무들이 인터넷을 통해서 이루어지고 있다. 인터넷은 개방형 네트워크이기 때문에 어떠한 사용자라도 접속가능하게 된다. 이는 악의적인 사용자가 공격을 목적으로 접근하게 되었을 때 도청, 침입, 도난 등의 피해를 당할 수 있다. 이러한 피해를 막기 위해 최근 인터넷 보안에 대한 관심이 높아지고 있다. 특히 전자금융거래의 급속한 발전으로 인해 인터넷 뱅킹 서비스의 이용이 증가되고 있으며, 이에 따른 전자금융 해킹 및 보안 사고가 발생하고 있다.

보안사고의 발생을 줄일 수 있는 방법 중에는 사용자 인증 방법이 가장 보편적으로 사용될 수 있다[1]. 인증(Authentication) 이란 특정 사용자가 접속을 요구할 때 사용자의 신원에 대한 보증 기능으로 현재 ID/Password를 기반으로 한 인증 기법이 가장 많이 사용되고 있다. 간단한 패스워드 인증 방법 이외에 사용자가 소유하고 있는 매체나 사용자 고유의 생체정보를 이용한 강력한 인증 방법도 사용되고 있다. 이러한 인증 방법들은 보안의 중요성에 따라 구분되어 사용된다.

현재 전자금융거래에서 사용되는 인증 방법은 보안카드와 공인 인증서를 이용하여 사용자를 인증하고 있다. 그러나 최근에는 일회용 패스워드(OTP, One Time Password)[1]기법을 새롭게 도입하여 사용되고 있다. 일회용 패스워드는 사용자가 인증을 요구할 때 패스워드를 생성하여 사용하는 방법으로 매번 생성된 패스워드는 서로 다른 값을 가지고 있어 한번 사용된 패스워드는 재사용하지 않게 된다. 이는 공격자가 네트워크상에서 패스워드를 도청하거나 사용자가 패스워드를 분실하더라도 안전을 보장할 수 있게 된다.

본 논문에서는 인터넷을 사용 할 때 발생할 수 있는 보안 위험에 관한 내용을 알아보고, OTP를 이용한 기존 프로토콜[2-6]을 분석한 후 문제점을 보완하고 새로운 동기화 인증 프로토콜을 제안한다.

Ⅱ. 인터넷 보안 위험성

2.1 사용자 보안 의식

최근 인터넷을 통하여 서비스를 제공 받는 사용자의 보안 의식 부재로 인하여 여러 가지 보안 사고가 발생되고 있다. 악의적인 사용자가 금융기관을 사칭하여 사용자들에게 메일을 보내고 가짜 금융기관 사이트로 접속하게 한 후 개인 정보를 획득하는 피싱(Phishing) 공격과 사용자로부터 진짜 사이트로 속이고 개인정보를 획득하는 파밍(Pharming) 공격은 사용자가 조금만 관심을 가지고 조심하면 사전에 방지할 수 있는 공격 방법이다. 그러나 이러한 공격 방법에 의한 피해 사례는 시간이 지남에 따라 계속 늘어나고 있고, 새로운 형태의 변형된 공격 기법들이 발견되고 있다.

2.2 패스워드 보안 정책

인터넷의 발전으로 여러 가지 서비스가 인터넷을 통하여 제공되고 있으며 사용자들은 인터넷을 통하여 업무를 처리하는 경우가 많아 졌다. 인터넷을 이용한 서비스를 사용하기 위해 사용자는 서비스마다 정상적인 사용자를 증명할 수 있는 인증과정을 거치게 된다. 현재 가장 많이 사용되고 있는 인증방법은 아이디/비밀번호 사용자 인증 방식이다. 보통 사용자들은 기억하기 쉽고 추측이 가능한 단어나 숫자들로 패스워드를 구성하기 때문에 제 3자에 의해 쉽게 유추 가능하다.

표 1. 패스워드 분석 시간
Table 1. Password analysis Time

입력 문자	7자리	8자리
영문 소문자(26문자)	45분	20시간
영문 소문자 + 숫자(36문자)	8시간	13일
영문 대·소문자 + 숫자(62문자)	25일	4년 6개월
영문 대·소문자 + 숫자 + 특수문자(94문자)	437일	114년

[주] 팬티엄4 3.0GHz CPU, 2G 메모리

한국정보보호진흥원(KISA)에서 조사한 내용에 의하면 20대 남녀 대학생의 패스워드 사용 시 대부분이 "영소문자+숫자"와 같이 2가지의 문자 종류로 구성된 패스워드를 이용하는 것으로 나타났다. 그중 6자리 이하의 패스워드를 이용하는 사용자는 64.5%에 이른다고 밝혀졌다. 표 1은 7 또는 8자리의 패스워드 문자 조합을 분석

해서 알아내는 데 필요한 시간을 계산한 예를 보여주고 있다[7].

표 2. 패스워드 복구 속도 I
Table 2. Password Recovery Speeds I

패스워드 길이	컴퓨터 사양 ·			
	팬티엄 1	듀얼 프로세서 PC	워크스테이션	슈퍼 컴퓨터
4자리	15초	2초	1초 미만	1초 미만
5자리	약 15분	1분 30초	9초	1초 미만
6자리	16시간	1시간 30분	약 9분	약 1분
7자리	41일	4일	10시간	58분
8자리	7년	253일	약 25일	약 60시간

[주] 영문 대문자+소문자+숫자 조합(총 62 문자)

표 3. 패스워드 복구 속도 II
Table 3. Password Recovery Speeds II

패스워드 길이	컴퓨터 사양			
	팬티엄 1	듀얼 프로세서 PC	워크스테이션	슈퍼 컴퓨터
4자리	1분 30초	8초	1초 미만	1초 미만
5자리	2시간 15분	13분	약 1분	8초
6자리	9일	22시간	2시간	13분
7자리	2년 6개월	87일	약 8일	20시간
8자리	229년	23년	약 2년 3개월	약 83일

[주] 영문 대문자+소문자+숫자+특수문자 조합 (총 96 문자)

표2와 표3은 2007년 1월 영국의 웹사이트 'Lockdown'[7]에 소개된 'Password Recovery Speeds' 내용이다. 시간이 지남에 따라 컴퓨터의 성능은 날로 발전되고 있으며 앞으로 더욱 발전된 높은 사양의 컴퓨터가 개발되면 패스워드를 분석하는 시간은 더욱 줄어든다. 그러나 패스워드의 안전을 위해서 자리수를 늘이거나 더욱 복잡한 문자 조합을 이용하게 되면 사용자는 패스워드를 기억하지 못하게 될 것이다. 이러한 문제점을 해결하기 위한 한 가지방법으로 OTP 기술이 있다. OTP는 사용자가 기억할 수 있는 비밀번호를 이용하여 복잡하고 보안에 강력한 새로운 비밀번호를 생성하는 기술이다.

Ⅲ. 기존 OTP 인증 프로토콜

3.1. S/Key 방식의 RFID 인증 프로토콜

S/Key[10] 방식의 OTP 생성 알고리즘을 이용한 인증 프로토콜[4]로서 그림 1과 같은 구조를 가지고 있으며, 등록 단계와 인증 단계가 필요하다.

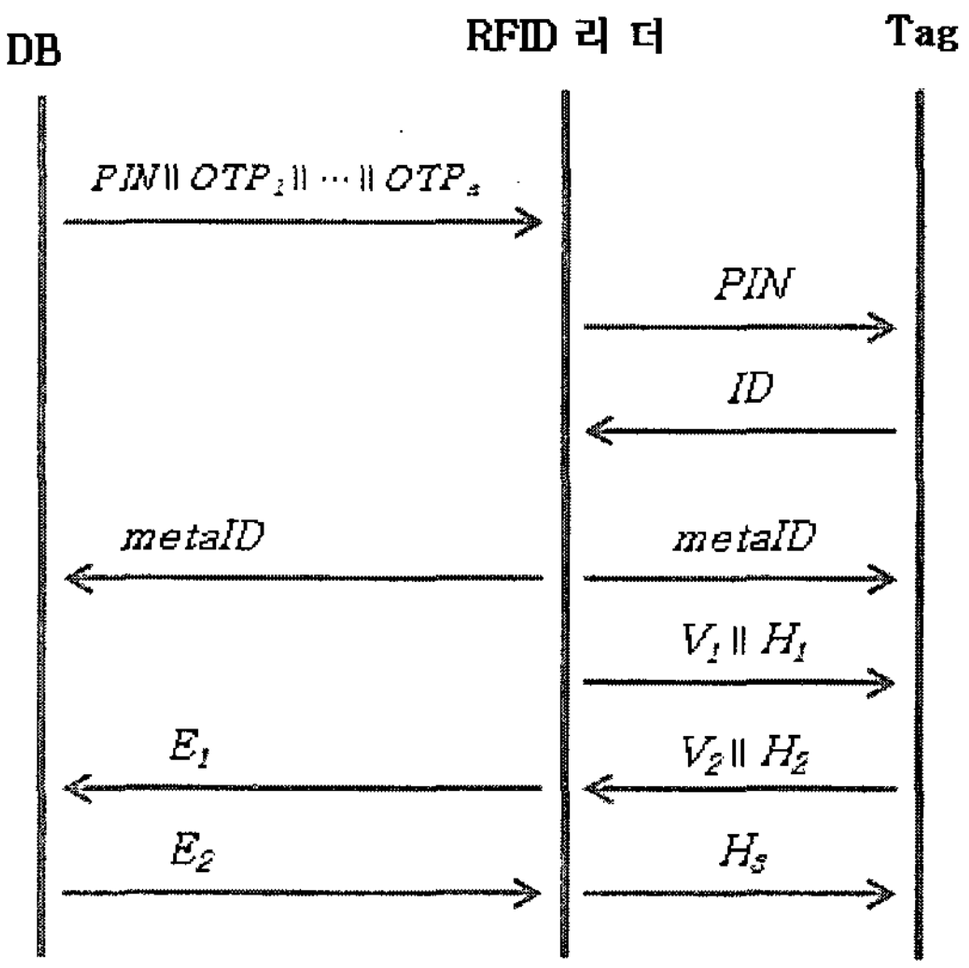

그림 1. 등록 및 인증 단계
Fig. 1 Registration and authentication phase

(1) 등록 단계

데이터베이스는 *PIN*과 초기 값 *seed*를 해쉬하여 *OTP*를 *n*개 생성하고 역순으로 RFID 리더에게 전송한다. 태그는 RFID 리더의 신호에 *ID* 값을 전송하고 *ID* 값을 해쉬하여 *metaID* 값을 생성한 RFID 리더는 데이터베이스와 태그에 재전송하여 등록을 마친다.

(2) 인증 단계

Step 1. RFID 리더는 타임스탬프 값과 *PIN*을 이용하여 다음 수식을 수행한 후 태그에게 전송한다.

$$V_1 = PIN \oplus TS_R$$
$$H_1 = h(PIN \| TS_R) \quad (1)$$

Step 2. 태그는 V_1으로 TS_R'를 획득하고 해쉬 값을 검증하여 무결성을 확인한 후 TS_T를 갱신한다.

$$V_1 \oplus PIN' = TS_R'$$
$$TS_T = TS_R + LT \quad (2)$$

Step 3. 태그는 *metaID*와 TS_T의 연산 값 및 해쉬 값을 생성하여 RFID 리더에게 전송한다.

$$V_2 = metaID \oplus TS_T$$
$$H_2 = h(metaID \| TS_T) \quad (3)$$

Step 4. RFID 리더는 저장중인 *OTP* 중 *n-1*번째 패스워드인 OTP_{n-1}과 V_2, H_2, TS_T를 *PIN*으로 암호화하여 E_1을 생성한 후 데이터베이스에 천송한다.

$$E_1 = E_{PIN}(OTP_{n-1}, V_2, H_2, TS_T) \quad (4)$$

Step 5. 데이터베이스는 E_1을 복호화하여 OTP_{n-1} 과 V_2, H_2, TS_T를 획득하고 *metaID*와 OTP_n을 검증한 후 TS_{DB}를 생성한다.

$$OTP_{n-1} \| V_2 \| H_2 \| TS_T = D_{PIN}(E_1)$$
$$metaID = V_2 \oplus TS_T$$
$$OTP_n = OTP_{n-1} \quad (5)$$
$$TS_{DB} = TS_T + LT$$

Step 6. 데이터베이스는 *ID*와 TS_{DB}를 *PIN*으로 암호화하여 E_2를 생성한 후 RFID 리더에게 전송한다.

$$E_2 = E_{PIN}(ID \| TS_{DB}) \quad (6)$$

Step 7. RFID 리더는 E_2를 복호화하여 TS_{DB}를 검증하고 *ID*와 TS_T를 해쉬하여 태그에게 전송한다.

$$ID \| TS_{DB} = D_{PIN}(E_2)$$
$$H_3 = h(ID \| TS_T) \quad (7)$$

Step 8. 태그는 H_3를 확인하여 인증한다.

(3) 분석

앞에서 분석한 인증 프로토콜은 OTP 생성 알고리즘인 S/Key를 기반으로 설계 되었고, 대칭키 암호화 알고리즘을 사용하여 데이터를 암호화/복호화 하고 있다. S/Key 알고리즘은 일방향 해쉬 함수를 반복해서 수행하여 *OTP* 값을 생성하게 된다. 이러한 방법은 *n*번의 숫자가 커질 경우 많은 수학적 연산이 요구 된다. 또한 대칭키 알고리즘은 RFID와 같은 소형 프로세서에 적용하기에는 연산 과정이 너무 많아 사용하기 힘들다.

최근 개인 정보가 중요시 되면서 익명성이 강조되고 있다. OTP는 매번 다른 값을 생성하기 때문에 익명성을 보장해 준다. 그러나 S/Key를 기반의 RFID 인증 프로토콜은 고정된 *metaID* 값을 이용하여 데이터를 송수신하기 때문에 익명성을 보장 받지 못하는 문제점이 있다.

3.2 RFID 시스템에서 OTP를 이용한 개인 프라이버시 보호

무선 네트워크를 이용하는 RFID 시스템은 전송 채널의 개방적인 특성 때문에 보안성이 취약하다. 따라서 개인의 프라이버시 정보를 침해당할 가능성이 높다. 그림 2는 RFID 시스템에 익명성을 보장하는 OTP 인증 프로토콜[6]의 구조이다.

(1) 인증 단계

Step 1. 처음 등록과정에서 태그와 DB는 비밀키 *P1*을 공유한다.

Step 2. 리더는 태그에게 데이터 전송을 받기 위해 쿼리를 보낸다.

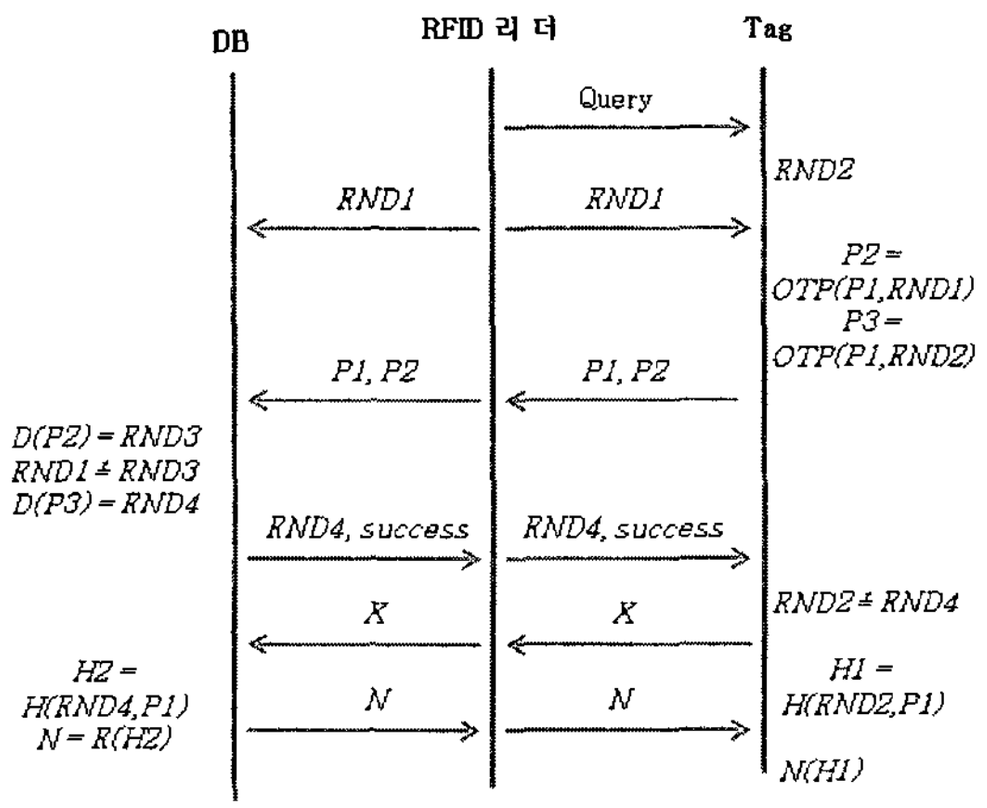

그림 2. 인증 단계
Fig. 2 Authentication phase

Step 3. 태그는 고유한 아이디를 리더에게 전송하고, 리더는 DB에 태그의 정보를 보내어 보관되어 있는 데이터를 통해 정상 태그인지 확인한다.

Step 4. 허가된 태그라면 리더는 임의의 난수 *RND1*을, 태그는 임의의 난수 *RND2*를 생성한다.

Step 5. 리더는 생성된 난수 *RND1*을 태그와 DB에 전송한다.

Step 6. 태그는 *P1*과 *RND1*을 이용하여 암호화 알고리즘을 적용하여 OTP *P2*라는 패스워드를 생성한다. 그리고 *RND2*를 이용하여 OTP *P3*라는 패스워드를 생성한다.

$$P2 = E_{P1}(RND1)$$
$$P3 = E_{P1}(RND2) \quad (8)$$

Step 7. 태그는 *P2*와 *P3*를 리더에 전송하고, 전송받은 리더는 다시 DB에 전송한다.

Step 8. DB는 *P1*과 태그에게 받은 *P2*를 이용하여 복호화 한 후 *RND3*을 생성한다. 그리고 *RND1*과 *RND3*의 값의 일치 여부를 판단한다. 일치하면 *P3*를 복호화하여 *RND4*를 생성한다. 그리고 *RND4*와 success 메시지를 리더를 통해 태그에 전송한다.

$$RND3 = D_{P1}(P2)$$
$$RND4 = D_{P1}(P3) \quad (9)$$

Step 9. 태그는 전송받은 *RND4*와 *RND2*를 비교한 후 자신이 가진 정보를 리더에게 전송한다. 동시에 *RND2*와 *P1*을 일방향 해쉬 함수를 이용하여 *H1* 값을 생성한다. 그리고 리더를 거쳐 *DB*에 *H1* 값을 전송한다.

$$H1 = h(P1,RND2) \quad (10)$$

Step 10. DB는 *RND4*와 *P1*을 일방향 해쉬 함수를 이용하여 *H2* 값을 생성한다. 그리고 *H2* 값의 일부분(*N*)을 다음에 사용할 패스워드로 등록하고, N을 리더를 통해 태그에 전송한다.

$$H2 = h(P1,RND4) \quad (11)$$

Step 11. 태그는 DB로부터 받은 *N*을 *H1*에 적용하여 그 일부분(*P2*)을 다음에 사용할 패스워드로 등록한다.

(2) 분석

앞에서 언급한 인증 프로토콜은 개인 프라이버시를 보호하기위해 매번 다른 값으로 변경되는 OTP 알고리즘을 선택하였다. 이는 매번 도청되더라도 새로운 값이 출력되기 때문에 사용자의 위치를 알 수 없게 된다. 그러나 위 프로토콜은 처음 태그가 리더에 접근할 때 고유한 아이디를 전송한다. 이는 고정된 값이기 때문에 다음 데이터들이 아무리 변경되더라도 사용자의 위치를 파악할 수 있는 문제점을 가지고 있다. 또한 암호화/복호화 알고리즘과 일방향 해쉬 함수 등을 사용하므로 많은 수학적 연산과 리소스를 필요하게 된다. 이러한 문제점은 소형 RFID 시스템에 적용하기 힘들다.

Ⅳ. 제안 프로토콜

4.1 PingPong-128 & KS() 함수

PingPong-128[9]은 그림 3에서 나타낸 것처럼 127비트, 129비트인 두 개의 LFSR을 가지고 있으며 128비트의 키와 128비트의 초기화벡터 값에 의해 내부 상태가 채워진다.

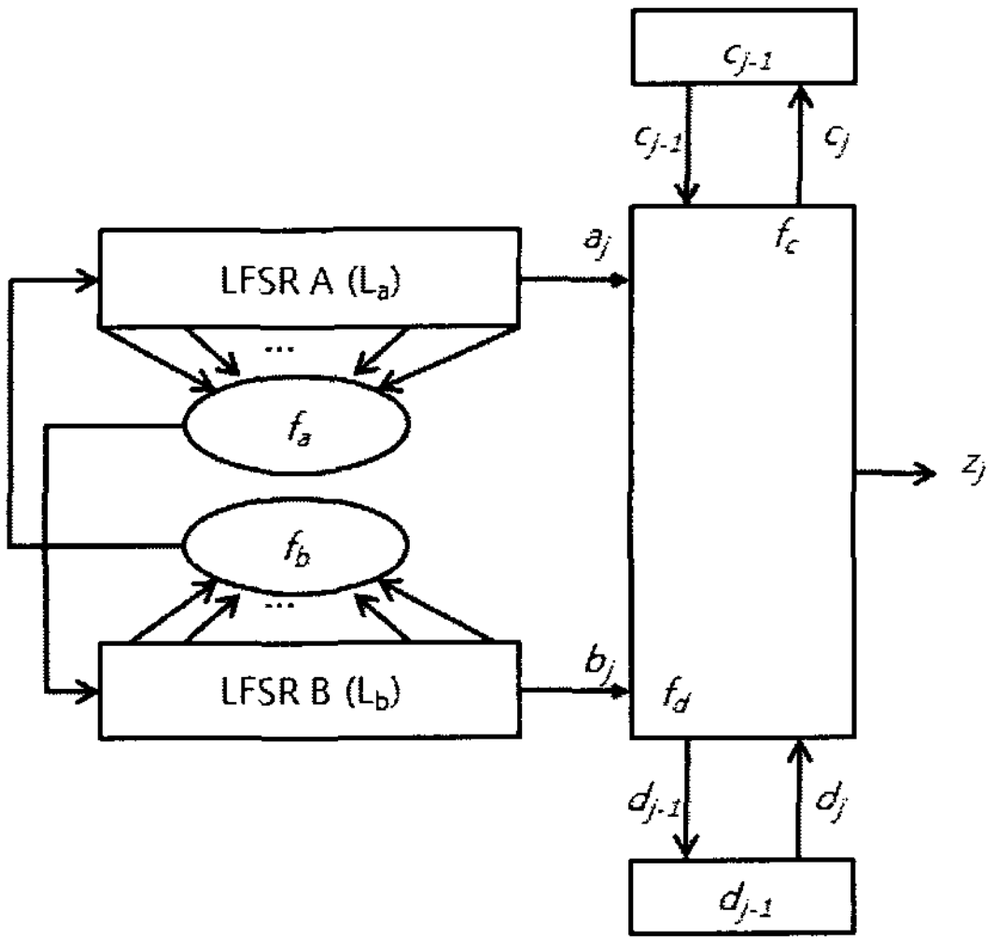

그림 3. PingPong-128 발생기
Fig. 3 PingPong-128 Generator

두 개의 **LFSR**의 클럭은 다음 수식으로 랜덤하게 제어하여 불규칙한 클럭을 발생시킨다.

$$
\begin{aligned}
&f_a(L_a)=2L_{a42}(t)+L_{a85}(t)+1\\
&f_b(L_b)=2L_{b43}(t)+L_{b86}(t)+1\\
&z_j = f_z(a_j,b_j,c_{j-1},d_{j-1}) = a_j \oplus b_j \oplus c_{j-1} \oplus d_{j-1}\\
&c_j = f_c(a_j,b_j,c_{j-1}) = a_j b_j \oplus (a_j \oplus b_j)c_{j-1}\\
&d_j = f_d(a_j,b_j,d_{j-1}) = b_j \oplus (a_j \oplus b_j)d_{j-1}
\end{aligned}
\qquad (12)
$$

본 논문에서 제안하고 있는 프로토콜의 KS() 함수는 PingPong-128에서 Key 로딩과 Key 갱신 부분을 제외시킨 함수이다. Stream Cipher의 Key 로딩과 Key 갱신 부분은 많은 자원과 연산을 필요로 한다. 제안 프로토콜은 데이터를 암호화/복호화 하는 것이 아닌 OTP Number의 생성을 주 목적으로 사용되기 때문에 PingPong-128의 Key 로딩과 Key 갱신 부분을 생략하였다. 두 개의 LFSR에는 사용자의 아이디와 비밀번호 등의 정보를 이용하여 채워 넣게 되고. Output zj 값을 출력하여 OTP 값을 생성하게 된다.

4.2 인증 서버의 데이터베이스 구조

그림 4는 인증 서버에서 관리되는 DB의 구조를 나타내고 있다. 필드는 *ID, Pass, Count* 로 구분되어 있으며 *ID*와 *Pass*는 다음 수식과 같이 구성된다. 이러한 구조는 사용자와 인증 서버 간에 통신 중 에러가 발생 하였을 때, 한쪽의 c 값만 증가하여 동기화에 실패했을 경우 동기화 값이 달라진 c 값을 복구하기 위해 사용된다.

$$
\begin{aligned}
ID &= KS_c(ID_i) \,||\, KS_{c+1}(ID_i)\\
Pass &= KS_c(PW_i) \,||\, KS_{c+1}(PW_i)
\end{aligned}
\qquad (13)
$$

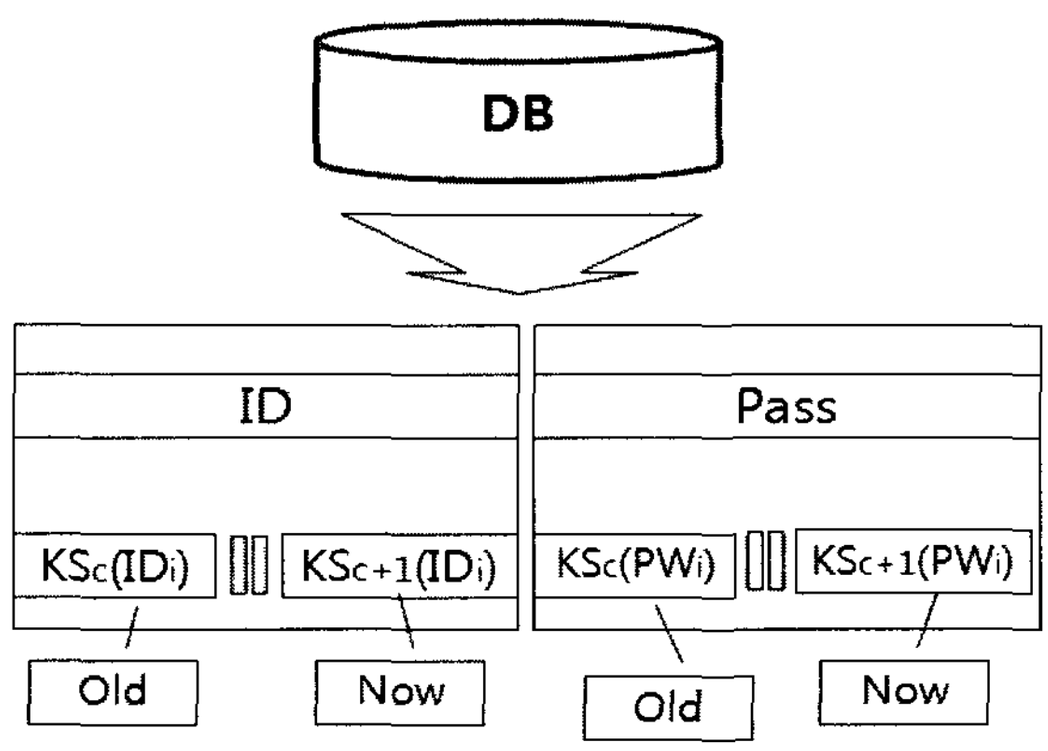

그림 4. 인증 서버의 DB 구조
Fig. 4 Database of Authentication Server

4.3 등록 단계

사용자는 안전한 채널을 통해서 인증 서버에 접속하고, 새로운 사용자를 등록하기 위한 과정을 거치게 된다.

Step 1 사용자는 고유의 아이디 ID_i와 PW_i 값을 입력하고 랜덤한 c 값을 생성한다. 그리고 ID_i, PW_i, c 값을 인증 서버에 전송한다.

Step 2 인증 서버는 전송 받은 값을 이용하여 다음 수식을 수행 한 후 *ID, Pass, c* 값을 DB 저장한다.

$$
\begin{aligned}
ID &= KS_c(ID_i) \,||\, KS_{c+1}(ID_i)\\
Pass &= KS_c(PW_i) \,||\, KS_{c+1}(PW_i)
\end{aligned}
\qquad (14)
$$

4.4 인증 단계

사용자는 서비스를 이용하기 위해 등록한 스마트카드 U_i를 디바이스에 삽입 후 아이디와 페스워드를 입력한다.

Step 1. 스마트카드 U_i는 입력된 ID_i와 PW_i를 이용하여 M과 H 값을 생성한다.

$$M = KS_c(ID_i)$$
$$H = h(y \oplus T) \quad (15)$$

Step 2. 스마트카드 U_i는 타임스탬프 값인 T와 M, H 값을 인증 서버로 전송한다.

Step 3. 인증 서버는 H, T, M 값 등을 확인하고, M 값을 DB 내부의 ID 필드에 검색한다. 그리고 검색된 ID 값을 $N1$, $N2$로 분리한다.

$$N1 = Right(ID)$$
$$N2 = Left(ID) \quad (16)$$

Step 4. M 값은 $N1$과 일치하게 된다. 검색된 ID에 해당된 $Pass$ 값을 검색한다. 그리고 다음 수식을 이용하여 $NP1$, $NP2$ 값으로 분리한다.

$$NP1 = Right(Pass)$$
$$NP2 = Left(Pass) \quad (17)$$

Step 7. M 값이 $N1$ 값과 일치하기 때문에 $NP1$ 값을 선택하여 R 값을 생성하고 H, T 값과 함께 U_i로 전송한다.

$$R = Right(KS_c(PW_i))$$
$$H = h(y \oplus T) \quad (18)$$

Step 8. U_i는 T 값을 이용하여 전송 지체 시간을 체크하고 H 값을 이용하여 T 값의 변경 유무를 확인한다. 그리고 R' 값을 생성하여 전송 받은 R 값과 비교한다.

$$H' \stackrel{?}{=} H$$
$$R' = Right(KS_c(PW_i)) \quad (19)$$
$$R' \stackrel{?}{=} R$$

Step 9. R'와 R 값이 일치하면 L' 값을 생성하여 인증 서버에 H, T 값과 함께 전송한다. 그리고 c 값을 증가 시킨다.

$$L' = Left(KS_c(PW_i))$$
$$H = h(y \oplus T) \quad (20)$$
$$c = c + 1$$

Step 10. 인증 서버는 전송 받은 L' 값과 기존에 생성한 L 값을 비교한 후 c 값을 증가 시키고 새로운 ID와 $Pass$ 값을 생성한 후 저장한다.

$$L = Left(KS_c(PW_i)) \quad (21)$$
$$N1 = KS_c(ID_i)$$
$$N2 = KS_{c+1}(ID_i)$$
$$ID = N1 \| N2$$
$$NP1 = KS_c(PW_i)$$
$$NP2 = KS_{c+1}(PW_i)$$
$$Pass = NP1 \| NP2$$
$$c = c + 1$$

4.5 제안 프로토콜 분석

본 논문에서 제안하는 인증 프로토콜는OTP 기술 중 이벤트 동기화 방식의 단점을 보완하였다. 통신 과정 중 에러로 인한 동기 이탈이 발생하면, 이후 인증서버에 접속하여 동기 이탈 여부를 확인 후 재동기화 한다.

표 4는 앞에서 분석한 기존 인증 프로토콜과 본 논문에서 제안한 인증 프로토콜의 안전성 부분을 표시한 내용이고, 표 5은 인증 프로토콜에서 연산되는 수식의 개수를 표현한 것이다. 기존의 인증 프로토콜들은 일방향 해쉬 함수 혹은 대칭키 알고리즘을 사용하여 전송 데이터를 보호하고 있다. 그러나 대칭키 알고리즘 또는 많은 해쉬 연산 방식의 인증 방식은 많은 연산 과정이 필요하기 때문에 RFID, 센서 네트워크, 스마트카드와 같은 소형 프로세서에 적용하기 힘들다. 본 논문에서 제안된 인증 프로토콜의KS() 함수는 연산 과정이 작은 스트림 알고리즘을 변형한 것이기 때문에 소형 프로세서에도 적용이 가능하다.

표 4. 안전성 분석
Table 4. Safety analysis

	MITM	도 청	재전송 공격	익명성
강수영 등 프로토콜[4]	○	○	○	×
Das 등 프로토콜[2]	×	○	○	×
chien 등 프로토콜[3]	○	○	△	△
이경욱 등 프로토콜[5]	×	×	×	×
이주형 등 프로토콜[6]	△	△	×	×
제안 프로토콜	○	○	○	○

	동기화	상호 인증	데이터 무결성
강수영 등 프로토콜	×	○	○
Das 등 프로토콜	×	×	×
chien 등 프로토콜	×	○	△
이경욱 등 프로토콜	○	×	○
이주형 등 프로토콜	○	×	△
제안 프로토콜	○	○	○

표 5. 성능 분석
Tabe 5. Performance analysis

	해쉬 함수	암호화	지수 연산	KS()
강수영 등 프로토콜[6]	(n+5)H	4E	0	0
Das 등 프로토콜	8H	0	0	0
chien 등 프로토콜	3H	4E	4Exp	0
이경욱 프로토콜[7]	2H	4E	0	0
이주형 등 프로토콜[8]	2H	4E	0	0
제안 프로토콜	3H	0	0	4KS

V. 결 론

본 논문에서는 기존에 제안된 인증 프로토콜의 문제점을 분석하고 보완하여 새로운 OTP 인증 프로토콜을 제안 하였다. 매번 다른 값을 가지는 OTP의 특성은 개인 프라이버시 정보를 보호할 수 있어 익명성을 보장해주며, 기존의 인증 프로토콜과 다르게 스트림 알고리즘을 사용하기 때문에 보안의 강도에 따라 키스트림 값을 유동적으로 조절 가능하다. 즉 어떠한 환경에서도 사용할 수 있는 확장성을 보장한다. 또한 이벤트 동기화 방식의 단점인 동기화 이탈시 문제점을 해결하여 동기화 값을 복구 할 수 있게 되었다.

향후 제안된 인증 프로토콜을 스마트카드, RFID, 센서 네트워크와 같은 소형 프로세스에 적용할 예정이며, 소형 프로세서를 이용한 하드웨어에 적합한 설계 연구가 필요할 것으로 보인다.

감사의 글

본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과(IITA-2008-C1090-0801-0026)와 지역혁신 인력양성사업의 연구결과로 수행되었음

참고문헌

[1] 백미연, "전자금융거래의 보안 강화 방안 및 OTP (One time Password) 이용현황", 지급결제와 정보기술, pp. 71-100, April 2006.

[2] M.L. Das, A. Saxena, V.P. Gulati, "A dynamic ID-based remote user authentication scheme", IEEE Transctions on Consumer Electronics, Vol. 50, No.2, pp.629-631, 2004.

[3] H.Y. Chien, C.H. Chen, "A Remote Authentication Scheme Preserving User Anonymity", IEEE AINA'05, Vol. 2, pp. 245-248, 2005

[4] 강수영, 이임영,"향상된 S/Key 방식을 이용한 RFID 인증 방안에 관한 연구", 한국정보처리학회 춘계학술발표 대회 제 14권, p1066-1067, 2007.5

[5] 이경욱 "金融機關에서 安全하게 使用 可能한 OTP (One Time Password) System에 관한 硏究" 성균관대학교 학위논문, 2006
[6] 이주형, 장태무, "RFID 시스템에서 OTP를 활용한 개인프라이버시 보호" 한국정보처리학회 춘계학술발표대회 제 13권, p865-868, 2006.5
[7] "당신의 패스워드, 얼마나 안전할까요?" 한국정보보호진흥원 정보보호뉴스 Vol.121, October 2007
[8] N. Haller et. al., "The S/KEY One-Time Password System", IETF RFC 1760, February 1995.
[9] HoonJae Lee, Kevin Chen, "Pingpong-128, A New Stream Cipher for Ubiquitous Application", ICCIT 2007, p1893-1899, Nov. 21-23 2007
[10] N. Haller et. al., "The S/KEY One-Time Password System", IETF RFC 1760, February 1995.

저자소개

이 장 춘(Jang-Chun Lee)

2006년 2월 동서대학교 정보 네트워크 학과 졸업(학사)
2008년 2월 동서대학교 유비쿼터스IT 학과 졸업(석사)
2008년 3월 ~ 현재 (주)TOTAL SOFT BANK 연구원
※관심분야: 네트워크 보안, RFID 보안, 물류관리 시스템

이 훈 재(HoonJae Lee)

1985년 2월 경북대학교 전자공학과 졸업(학사)
1987년 2월 경북대학교 전자공학과 졸업(석사)
1998년 2월 경북대학교 전자공학과 졸업(박사)
1987년2월~1998년1월 국방과학연구소 선임연구원
1998년3월~2002년2월 경운대학교 조교수
2002년3월~현재 동서대학교 컴퓨터정보공학부 부교수
※관심분야 : 암호이론, 네트워크보안, 부채널공격

임 효 택(Hyo-Taek Lim)

1988년 홍익대학교 전자계산학과 졸업(이학사)
1992년 포항공과 대학원 전자계산학과 졸업(공학석사)
1997년 연세 대학교 컴퓨터과학과 졸업(공학박사)
1988년~1994년 한국전자통신연구소 연구원
2000년~2002년 Univ. of Minnesota(미) 컴퓨터공학과 연구교수
1994년~현재 동서대학교 컴퓨터공학과 부교수
※관심분야 : Computer Network, Protocol Engineering, Storage Networking, IPv6, Mobile Application

이 상 곤(Sang-Gon Lee)

1986년 2월 경북대학교 전자공학과 졸업(학사)
1988년 2월 경북대학교 전자공학과 졸업(석사)
1993년 2월 경북대학교 전자공학과 졸업(박사)
1991. 3 - 1997. 2 창신대학 전자통신과 조교수
2003. 8 - 2004. 7. 호주 QUT ISRC(암호학연구소) Visiting Fellow
1997~현재 동서대학교 컴퓨터정보공학부 부교수
※관심분야 : 암호이론, 네트워크보안 프로토콜, 컴퓨터네트워킹