안전한 로그인을 위한 보안카드 기반 이중 인증 시스템에 대한 연구

조제경*, 서종원*, 이형우*
*한신대학교 컴퓨터정보소프트웨어학부
e-mail:aiking@hs.ac.kr, seo0207@hs.ac.kr, hwlee@hs.ac.kr

Two Factor Authentication System base on Software type of Secure Card For Secure Login

JE Gyeong Jo*, Jong-Won Seo*, Hyung-Woo Lee*
*Div. of Com., Info & Software, HanShin University

요 약

로그인 과정은 사용자의 ID와 Password를 기반으로 시스템에 대한 사용권한을 부여한다. 로그인 과정에서 입력된 ID와 Password 정보는 패킷 스니핑 또는 Keylogger 프로그램 등을 이용하여 악의적인 공격자에 의해 노출될 수 있다는 취약점이 있다. 웹서버 또는 웹메일 시스템 등에 등록된 ID와 Password가 노출된다면 이는 개인 프라이버시 문제와도 연결되어 매우 심각한 문제이기도 하다. 현재 대부분의 시스템에서는 ID와 Password만을 가지고 사용자에 대한 인증 및 로그인 과정을 수행하기 때문에 더욱더 강력한 복합 로그인 메카니즘이 제시되어야 한다. 본 연구에서는 기존의 ID/Password 기반 로그인 기법과 더불어 소프트웨어 형태의 보안카드를 핸드폰에 설치하여 유무선망을 통한 이중 인증(Two factor authentication) 기법을 제시한다. 제안한 소프트웨어 형태의 보안카드 기반 로그인 기법은 ID/Password와 함께 부가적 정보로써 사용자의 핸드폰에 발급받은 보안카드내 난수 형태로 생성된 번호를 사용한다. 따라서 제안한 시스템을 사용할 경우 기존의 ID와 Password와 연계되어 일회용 패스워드 형태로 제공되는 보안카드 정보를 사용하여 로그인 과정을 수행하기 때문에 보다 안전한 인증 시스템을 구축할 수 있다.

1. 서론[1)]

웹사이트에서 회원제로 운영하기위해서는 각각의 회원마다 ID와 Password가 필요하며 신청 또는 발급 받은 ID와 Password를 통해 웹서비스에 로그인을 하게 된다.

하지만 기존의 로그인 시스템은 간단하게 로그인을 할 수 있다는 장점이 있지만 스니핑을 통해 쉽게 ID와 Password를 알아내어 다른 사람이 로그인 할 수 있다는 단점이 있다. 이러한 문제를 해결하기 위해 다양한 로그인 기법들이 연구중이다. 기존의 연구중인 로그인 기법중 로그인 할 경우 클라이언트에 프로그램을 다운 받는 방식과 다운 받은 클라이언트에 공개키 암호화 방식을 추가시킨 방법들이 진행중이다. 하지만 이 기법들은 클라이언트 PC에 따라 로그인 환경이 달라진다는 등 몇가지 문제점이 있다. 그리고 ID와 Password로 로그인을 한후 다른 부가적 정보를 가지고 로그인하는 것을 연구중인 곳도 있으며 이러한 부가적 정보를 가지고 한번더 인증의 과정을 거치는 것을 이중인증(Dual Authentication)이라고 한다. 본 연구에서는 이중인증 방법을 개선하여 은행에서 결제시 사용하는 보안카드를 이용한 인증 시스템을 고안하게 되었다. 사용자 인증 과정에 있어 ID와 Password만을 사용하는 것이 아닌 회원 가입시 발급 받은 보안카드의 번호를 이용하여 사용자 인증과정을 거친다. 이러한 보안카드를 이용한 사용자 인증기법은 인증 과정을 이중으로 하게 되며 사용자의 ID와 Password가 누출이 되더라도 보안카드번호를 알아야 인증을 받을수 있는 이중 인증 과정을 제시 한다.

본 논문의 2장에서는 기존의 안전한 로그인을 위한 기법 관련 연구들과 그 문제점들을 제시한다. 3장에서는 기존의 문제점들을 보안하기위한 모델을 제시하며, 4장에서는 제안 모델을 통한 실험을 보여준다. 마지막으로 5장에서 본 모델의 장점과 안전한 로그인을 위한 앞으로의 연구 방향을 제시한다.

1) 정보통신부 및 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음(IITA-(C 1090-0603-0016)

2. 관련연구

대부분의 웹서비스에서는 ID와 Password만으로 사용자에 대한 인증을 수행한다. 따라서 아래 그림 1과 같이 웹 인증 과정에서 전송되는 ID와 Password는 Ethereal을 통해서 너무나 쉽게 스니핑 할 수 있다. 기존의 로그인 방식에는 그림 1과 같은 ID와 Password가 쉽게 노출 되는 문제점이 있기에 다양한 로그인 방법들이 연구 중이다.

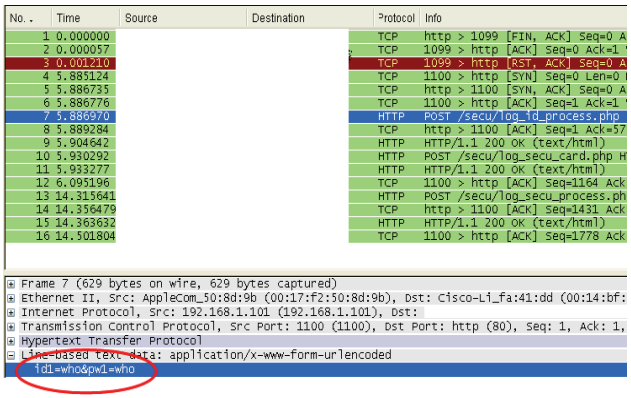

그림 1 로그인 스니핑

2.1 로그인 클라이언트 다운로드 방식

로그인 클라이언트 다운로드 방식은 웹서비스를 받고자 하는 사이트에서 간단한 클라이언트 프로그램을 작성

하여 배포하는 것이다. 이것은 기존의 사이트에서 ActiveX 콘트롤러나 기타 다양한 프로그래밍을 통해 만들어 놓은 프로그램을 제공하고 있다.

이 기법의 장점은 로그인 클라이언트 프로그램의 보안적 문제나 버그에 대해 사이트 접속만으로 쉽게 수정할 수 있다는 것이다. 그리고 이동이 잦은 사용자의 경우 웹 사이트 접속만으로 원하는 PC에 바로 설치가 가능하다. 이 방식은 스크립트 프로그래밍을 통해 자동으로 설치되는 방식을 제공할수 있으며 편리함을 제공한다.

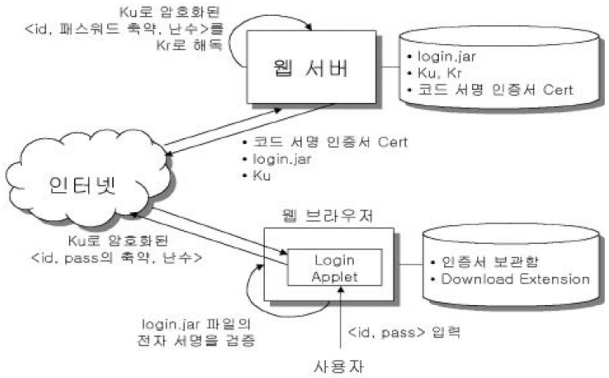

그림 2 로그인 클라이언트 구성

하지만 온라인 배포인만큼 악의적인 사용자가 로그인 클라이언트의 수정을 통해 배포할 경우 사용자 정보나 파일을 훔쳐가는 코드를 내장할 수가 있다는 단점이 있다. 그리고 클라이언트 크기가 크거나 네트워크 속도가 느린 경우 로그인 클라이언트의 설치에 시간이 많이 소요된다는 단점이 있다.

2.3 인증서 기반의 로그인 시스템

현재 대부분 인터넷으로 사용하는 은행의 경우 계좌이체나 결제를 위해 인증서를 사용하고 있다. 이 인증서는 대부분 결제 시스템에서 사용되고 있지만 몇몇 시스템에서는 로그인 시스템으로도 사용하고 있다. 이 기법은 각자 다른 인증서를 발급받게 된다는 점을 이용하여 본인인증을 한번 더 거치게 된다. 이 인증서는 불법 복제 및 수정이 어렵기 때문에 안전한 로그인을 제공하고 있다.

그림 3 인증서를 통한 로그인

하지만 이 기법은 사용자의 PC에 저장이 되게 되며 다른 PC에서 로그인 할경우 인증서를 가지고 다녀야 하는 단점이 생긴다. 그리고 인증을 위한 프로그램 설치에서부터 각기 다른 인증서의 선택등 복잡한 인증과정을 거치게 된다.

2.4 일회용 암호를 이용한 국산 암호 인증 시스템

로그인 및 결재와 같은 인증이 필요한 시스템에서는 암호를 사용한다. 암호는 사용자가 지정할수도 있지만 매번 서비스업체에서 바꾼 암호로 사용할 수도 있다. 이렇게 인증을 요청시 매번 바꾸어서 전에 사용한 암호를 다시 사용할수 없는 방식의 사용자 인증 방식을 OTP(One Time Password)라고 불린다. 이 방식은 인증 요청시 미리 저장된 경로를 통해 새로운 암호를 알려 주며 매번 갱신이 되기 때문에 스니핑이나 재전송의 공격에 강하다. 하지만 이 방식은 매번 바뀌는 암호를 받기위해 다른 통신에 연결을 해야 하기 때문에 상당한 불편을 감수해야만 한다.[12]

앞서 설명한 로그인 시스템들과 같이 재전송이나 로그인 클라이언트의 역컴파일, 그리고 인증서 기반 로그인 시스템의 취약성과 같은 문제를 해결하기 위해 특정한 소프트웨어의 설치가 필요없는 로그인 방법으로 보안카드 기반 이중인증 메카니즘을 제안한다.

3. 제안모델

안전한 로그인을 위한 보안카드 기반 인증시스템은 웹 서버 및 인증 서버로 구성이 되며, 보안카드 발급 및 보안카드기반 로그인 모듈로 구성된다.

3.1 회원 가입 및 로그인을 위한 보안카드 발급

3.1.1 회원가입

본 시스템은 로그인을 위한 시스템이기 때문에 회원가입을 필요로 한다. 일반적으로 회원 가입의 절차와 동일하며 필요조건으로서 핸드폰 번호와 생년월일을 입력하여야 한다. 처음 로그인시 ID와 생년월일을 입력하며 입력된 ID와 생년월일이 일치할 경우 보안카드와 비밀번호를 입력하게된다. 보안카드의 경우 핸드폰을 통해 보안 카드가 발급되기 때문에 핸드폰 번호는 필수조건이 되어야 한다.

```
/* 입력 : 사용자 정보
출력 : 회원가입 신청결과 */
Bool Join(){
    Struct Member_Information Join_Member;
    Bool Result;
    Input( Join_Member.ID,
           Join_MemberPassword,
           Join_Member.Birthday,
           Join_Member.PhoneNumber
           etc...);
    Result=Check_Blank( Join_Member.ID,
                        Join_MemberPassword,
                        Join_Member.Birthday,
                        Join_Member.PhoneNumber);
    return Result;
}
```

위 의사코드는 회원 가입을 위한 코드이다. Member_Information 이라는 회원 정보를 위한 구조체 변수가 선언이 되며 필수정보 입력을 판단하기 위한 Result 변수가 선언된다.

Input()은 사용자로부터 정보를 입력받는 함수이며 파라미터는 회원 가입시 입력되는 정보들이다. Check_Blank()는 필수정보의 확인을 위한 함수이다. Check_Blank()의 파라미터에는 필수정보인 ID, Password,

Birthday, PhoneNumber를 전달한다.

3.1.2 WIPI를 통한 보안카드 발급

현재 핸드폰의 WIPI 플랫폼을 이용하여 핸드폰에 프로그램을 다운로드하는 것이 가능하다. 본 연구에서는 WIPI를 통해 프로그램의 주소를 메세지에 입력하여 보낼 경우 핸드폰에 바로 다운로드를 위한 연결을 해준다. 연결된 핸드폰은 프로그램뿐만 아니라 보안카드번호를 생성하기 위한 Key값도 같이 다운로드 받게 된다.

```
/* 입력 : ID, PhoneNumber
출력 : Message, Key */

void Send_Message(ID,PhoneNumber){
    int Key;
    Key = Key_Generate(ID);
    Send(PhoneNumber,Key);
}
```

보안카드를 발급하는 모듈에서는 ID와 핸드폰번호를 입력받으며 Key 생성을 위한 변수를 선언한다. ID에 따라 고유의 키가 Key_Generate()를 통해 생성이 되며 생성된 Key는 핸드폰번호와 함께 Send모듈로 전송된다. Send 모듈은 핸드폰으로 보안카드 프로그램과 생성된 고유의 키를 전송한다.

3.1.3 로그인을 위한 보안카드 프로그램

로그인을 하기 위해서는 보안카드가 필요하기 때문에 보안 카드 번호를 생성하여야 한다. 보안카드번호는 4자리 숫자로 이루어지며 30개의 번호가 생성된다. 이 번호들은 여러 방식의 Key 생성 기법들을 통해 생성이 가능하며 생성된 번호들은 겹치지 않는 것이 중요하다.

번호를 생성하는데 있어서는 여러 방법의 Key생성 기법이 있으나 본 시스템에서는 핸드폰에 프로그램 설치시 받은 고유의 Key값을 가지고 120자리의 Key를 생성하며 생성된 Key값을 4자리씩 토큰화시켜 30개의 카드번호를 생성한다.

```
/* 입력 : Key
출력 : CardNumber[] */

void Create_CardNumber(Key){
    String CreateKey;
    int CardNumber[30];
    CreateKey=Hash(key);
    CardNumber[]=Token(CreateKey);
}
```

위 코드는 핸드폰에 들어가는 WIPI 플랫폼 기반의 프로그램중 일부분이다. 회원가입시 전송받은 고유의 Key값이 파라미터로 전달이 되며 고유의 Key 값을 통해 생성될 값이 CreateKey로 선언된다. CardNumber[]는 생성된 카드번호를 저장할 배열변수이다. 입력된 고유의 Key값은 Hash()함수를 통해 120자리의 번호가 생성이되어 CreateKey로 반환된다. 반환된 CreateKey는 Token()함수를 통해 4자리씩 나누어 CardNumber[]로 저장되게 된다.

3.2 발급된 보안카드와 ID/PW를 통한 로그인

3.2.1 ID,생년월일을 통한 로그인

처음에는 기본적으로 ID와 생년월일을 입력하도록 한다. Password를 먼저 입력하지 않는 이유는 Password를 스니핑으로부터 보호하기 위해서 이다. Password는 보안카드를 입력할 때 같이 입력하게 된다. ID와 생년월일 통해 로그인이 성공할 경우 보안카드 번호 체크 페이지로 넘어 갈수 있다. 로그인 부분은 기존의 사이트에서 그대로 사용할 수 있게 하기 위하여 큰 변화를 주지 않았다. 기존의 회원제 서비스를 제공하는 사이트의 로그인 모듈을 약간의 수정으로 사용할 수 있다.

```
/* 입력 : ID, Birthday
출력 : 1차로그인결과 */

Bool Login_ID_Birthday(){
    String ID, Birthday;
    Bool Result;
    Input( ID, Birthday);
    Result = Compare( ID, Birthday);
    return Result;
}
```

위 코드는 첫 번째 ID와 생년월일을 이용한 로그인의 코드이다. ID와 생년월일을 입력받기 위한 변수가 선언이 되며 결과를 반환하기 위한 Result변수가 선언이 된다. Result변수에 따라 보안카드 로그인으로 넘어 갈지 다시 로그인 페이지로 갈지 결정된다. Input()함수를 통해 ID와 생년월일을 입력 받는다. 그리고 회원정보가 입력된 데이터 베이스와 Compare()함수를 통해 비교를 하게 된다. 그 결과는 Result에 저장되어 반환되게된다.

3.2.2 보안카드를 통한 로그인

보안카드를 통한 로그인페이지는 ID와 생년월일을 통한 로그인이 성공할 경우 보여준다. 보안카드의 번호를 입력하는 화면은 하나의 보안카드 의 Index번호와 비밀번호 입력창을 보여준다. 해당 Index번호와 연결이 되는 카드의 번호를 입력하면 입력된 번호와 비밀번호를 암호화 하여 전송하게 된다. Index번호에 따른 카드번호는 회원가입시 핸드폰에 저장된 프로그램을 통해 알 수 있게 된다.

```
/* 입력 : ID, CardNumber, Password
출력 : 2차로그인결과 */

Bool Login_CardNumber_Password(ID){
    String CardNumber, Password;
    Bool Result;
    String EncodeData;
    print( Random_Card_IndexNumber );
    Input( CardNumber, Password);
    EncodeData=Encoding(CardNumber,Password);
    Result = Compare( ID, EncodeData);
    return Result;
}
```

카드번호와 비밀번호를 통한 로그인 모듈에서는 카드번호와 비밀번호를 입력받기 위한 변수가 선언이 되며 결과를 출력하기 위한 Result가 선언이 된다. 우선 Print()함수를 통해 랜덤하게 생성한 카드의 Index번호인 Random_Card_IndexNumber를 출력한다. 출력된 Random_Card_IndexNumber에 맞추어 사용자가 CardNumber와 Password를 입력하게 되며 입력된 CardNumber와 Passwrod는 Encoding()함수를 통해 암호화과정을 거친다. 암호화과정을 통해 생성된 data는

EncodeData에 저장되며 암호화된 EncodeData와 ID를 Compare()함수에 전달하여 사용자정보가 저장된 서버와 비교를 하게 된다. 비교 결과는 Result함수에 전달되어 반환하게 된다.

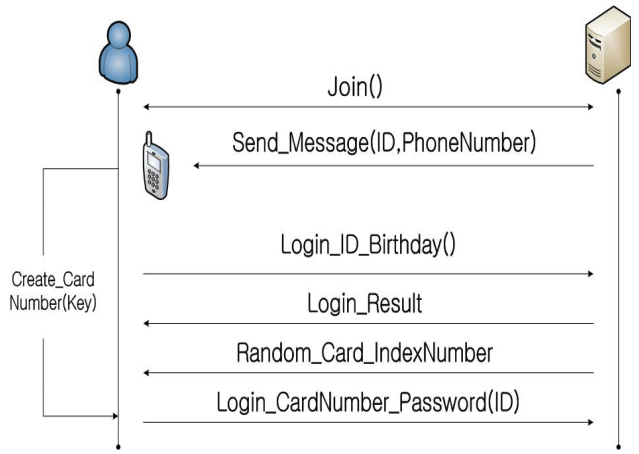

그림 7 모듈로 본 흐름도

4. 실험 및 안정성 평가

본 실험은 리눅스 시스템에서 이루어 졌으며 웹서버는 Apache, 데이타베이스 서버는 Mysql, 시스템 언어로는 PHP를 사용하였다.

현재 대부분의 웹사이트는 ID, Password만을 비교하기 때문에 한번 누출된 ID와 Password만을 가지고 다른 사람인 것처럼 인증을 받을수 있다.

그림 9은 본 논문에서 제시한 모델을 구현하여 스니핑을 시도해본 결과이다. 아래의 스니핑 내용을 보면 알수있듯이 전송되는 데이터가 Hash 암호화되어 가게 된다. 본 논문에서의 실험은 MD5방식의 암호화를 사용하였으며 Password와 카드번호 모두 암호화 되기 때문에 실질적으로 추측이 불가능하다.

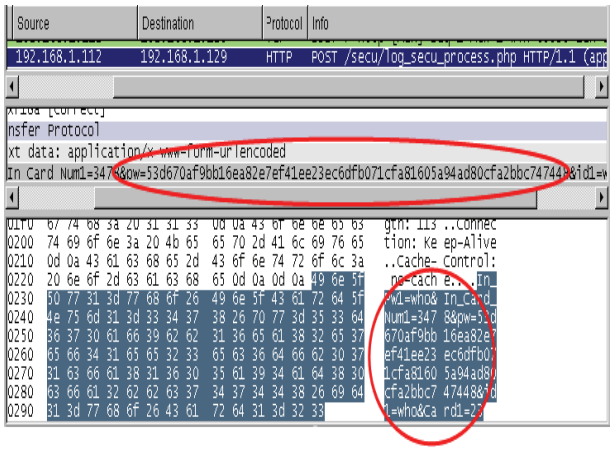

그림 5 로그인 정보 스니핑

종류 / 기능	클라이언트 다운로드	공개키 기반	인증서 기반	보안카드
ID,Password입력	O	O	X	O
부가정보입력	X	X	O	O
Dual Authentication	X	X	X	O
OneTime Password	X	X	X	O

표 1. 기존 시스템과의 비교분석평가

위 표를 보면 알수있듯이 보안카드 로그인은 기존의 로그인 기법에 사용되는 기본정보를 사용하며 부가적 정보를 이용하여 이중인증(Dual Authentication)을 제공하며 30개의 제한을 가지긴하지만 로그인할때마다 다른 번호를 입력하게되는 OneTime Password기능을 제공한다.

5. 결론

대부분의 연구는 현재 얼마나 안전하게 통신을 하느냐에 관심을 가지고 새로운 기술의 개발에만 집중할 뿐 현재 구현된 기술의 보안에는 신경을 쓰지 않고 있는 것이 사실이다. 그러한 결과로 지금 현재 많은 로그인 시스템이 동일한 ID와 Password로 로그인을 하도록 해놓은 경우가 많다. 이러한 안일한 의식이 해킹의 실마리를 제공하게 되며 하나의 시스템 정보가 누출이 되어도 자신이 이용하는 모든 시스템의 계정을 바꾸어야 하는 것이다. 이러한 부분은 사용자에게도 시스템 운영자에게도 힘든 일이다. 그렇기에 본 시스템을 통해 로그인을 조금 더 안전하게하며 별도의 소프트웨어를 설치해야 하는 번거로움을 없앴다. 안전한 로그인을 위한 보안 카드시스템은 현재 로그인 시스템의 큰 수정 없이 적용이 가능하다. 보안카드를 이용한 로그인 시스템은 사용자와 시스템의 안전함을 제공할 수 있으며 앞으로도 이와 관련하여 현재 사용 중인 시스템을 더욱더 안전하게 이용하는 방법에 신경을 써야 할 것이다.

참고문헌

[1] 장혜진, '클라이언트 다운로드 방식의 안전한 로그인 프로세스의 설계 및 구현'
[2] 정용주, 'session 내장 객체를 이용한 로그인 방법과 그 분석'
[3] 서종원, 조제경, 이형우, 'Spam mail 방지를 위한 SMS(Short Message Service) 송신자 인증 방법에 관한 연구', 2006년도 한국정보보호학회 동계학술대회
[4] 김기태, 박우진, '컴파일러', 홍능과학출판사
[5] 윤한성, '정보보안과 암호화', 21세기사
[6] 강선영, '암호화프로그래밍', FREELEC
[7] Douglas Toombs외 19명, 'Security 전문가 비밀노트', Microsoft
[8] 서광석, '수론과 암호학', 경문사
[9] Matthew, 'Beginning Linux Programming', Wrox
[10] 안정철, '시스템 로그분석', 이비커뮤니케이션
[11]STUART MCCLURE, SAUMIL SHAH, SHREERAJ SHAH, '웹해킹(공격과방어)', 피어슨에듀케이션코리아
[12] 추성호, 제갈명, 박홍성, '일회용 암호를 이용한 국산 암호 인증 시스템', 춘천멀티미디어학술회의 논문집
[13] 김인순, '이중인증시스템이 대세로', 전자신문 2007/03/13