다중 디바이스에서 HTML5 의 keygen 과 Local-storage 기반의 2-factor 인증

이규석*, 최진영
*고려대학교 컴퓨터정보통신대학원 소프트웨어공학과
e-mail : hahohh@korea.ac.kr, choi@formal.korea.ac.kr

2-factor authentication using Keygen and Local-Storage in HTML5 on multiple devices

Gyu-Seok Lee*, Jin-Young Choi
* Dept. of Software Engineering, Graduate School of Computer and Information Technology, Korea university

요 약

모바일 디바이스의 대중화와 SNS(Social Networking Service)의 발전은 각 개인이 데이터와 정보를 생성하는 Web 2.0 의 페러다임을 앞당겼으며 최근 SNS 서비스를 통하여 새로운 형태의 커뮤니케이션 형태가 생성되었다. 이러한 커뮤니케이션 도구를 이용하는 유저는 대부분 ID 와 Password 를 기반으로 사용자를 인증하여 서비스를 제공받는다. 이와 같은 서비스에서는 각 사용자의 정보자체보다 사용자의 사회적 위치와 사용자간의 관계를 이용한 보안사고가 우려된다.

근래의 ID/Password 로 인증하는 방식의 웹서비스 또는 정보서비스들은 대부분 개인 PC, 스마트폰, 업무 PC 등에서 접근하는 추세이며, 임의적 장소에서 임의의 기기로 해당 서비스에 접근하는 양상은 과거에 비하여 감소하는 추세다. 이 같은 추세에 따라, 주로 사용하는 기기에 HTML5 의 keygen 과 Web-Storage 기능을 사용하여 암호화된 Key 를 생성하고 저장하여 ID 와 Password 가 노출되어도 해당 기기가 아니면 인증되지 않는 시스템을 구현 할 수 있으며 타 기기의 경우 일회성을 갖는 인증 방식을 사용하여, 기존 보다 안전한 인증 시스템을 적은 비용으로 구축 할 수 있다.

1. 서론

PC 의 세대에서 인터넷 세대를 거쳐 모바일 세대로의 변화에도 대부분의 서비스는 사용자 인증을 필요로 한다. 특히나 급증하는 모바일 디바이스 유저의 증가에 따라 기존 웹을 모바일 디바이스 해상도에 맞게 변경한 웹과 전용으로 제작된 어플리케이션 들에서도 사용자 개개인에 대한 개별 서비스를 제공하기 위하여 회원 가입 단계를 거쳐 ID/Password 를 입력하는 방식으로 사용자를 인증하는 체계가 사용되고 있다.

이러한 ID/Password 인증방식의 단점을 알고 있는 악의적 유저는 서비스 제공자가 보유한 보안을 뚫고 데이터베이스에 접근하여 유저 정보를 해킹 하는 것에 그치지 않고 메신저 서비스나 트위터, 페이스북 같은 SNS 에 접근하여 해킹한 유저로 접근, 해당 유저와 관련된 유저들을 대상으로 정보요구나 금전 요구를 하는 식의 악의적 사용이 증가하고 있다. 스니핑(sniffing), 키로거 등을 통하여 해당 정보를 얻기만 하면 손쉽게 개인정보 및 피의자의 주변인에게도 피해를 주는 사례가 발생할 수 있다. 이러한 해킹툴은 최근의 UCC 재생을 위한 ActiveX 의 설치나 임의의 메신저 전송파일, E-mail 을 확인만 하여도 설치되는 사례가 많다.[1]

기존 로그인과 관련된 기법 중 로그인 시 클라이언트에 프로그램을 설치하여 암호화하는 방식으로 로그인하는 방안이 제시되었다. 하지만 이 기법은 클라이언트가 변경되는 경우 프로그램을 다시 설치해야 하는 문제점이 있다. 그리고 캐시를 사용한 로그인 방안과 바이오 정보 같은 부가적 정보를 이용하여 인증과정을 거치는 방안도 제시되었지만 인증방안에 비용이 증가하는 단점이 발생할 수 있다.

본 연구에서는 기존의 인증 방법을 개선하여 ID/Password 에 추가적으로 디바이스 상의 Key 를 전송하여 매칭하는 방식의 인증 시스템을 제안한다. 사용자 인증 과정에 있어 유저에게 익숙한 기존 접근 방식을 사용하면서도 최초 접속하는 기기에 사용자 고유의 Key 의 등록 여부를 판단하여 접속을 시도하는 기기와 해당 정보가 동일 한 경우 ID/Password 에 대한 인증을 수락한다. 또한 임의의 장소에서의 접근에 대하여는 등록되지 않은 기기로 인지하여 사용자에게 개인 인증을 거쳐 일회적으로 해당 ID/Password 로 로그인 하는 시도를 수락하는 시스템을 제공한다.

이때 사용되는 디바이스상의 key 는 HTML5 의 Web-Storage 기능을 사용하였으며, 쿠키의 대체적인 기능이지만 다른 용도로서 클라이언트에 저장된 값을 사용한 다른 이용방안을 모색하였다.

본 논문의 2 장에서는 기존의 로그인에서의 보안관련 연구들과 그 문제점들을 제시한다. 3 장에서는 기존 문제점들을 보안하기 위한 모델을 제시하며, 4 장에서는 3 장에서 제안된 Key 를 Local-Storage 에 저장하기 위한 구현 방안을 제시한다. 5 장에서는 기존 모델과의 비교 및 평가를 하며, 마지막으로 6 장에서 본 모델의 장점과 안전한 로그인을 위한 앞으로의 연구 방향을 제시한다.

2. 관련연구

대부분의 유저인증 시스템에서는 ID 와 Password 만으로 사용자에 대한 인증을 수행한다. 이러한 시스템은 Mobile 디바이스에서도 같은 방식으로 적용된다. 예를들면 SmartPhone, Labtop, SmartPad, SmartMP3 등과 같은 환경에서도 같은 방식이다. 디바이스는 변화하고 있지만 Cell Phone 과 같은 특정 번호나 기기 자체 Serial Number 인증을 받는 식의 인증 방식을 사용하는 일부 어플리케이션을 제외하면 현 유저 인증 방식과의 차이가 없다. 이와 같은 인증 방식은 ID/Password 만 유출되면 누구든지 해당 정보로 로그인을 할 수 있다는 문제점이 있다. 또한 이러한 인증 방식으로 인하여 ID/Password 는 집중적으로 해킹의 대상이 되었다. 이러한 기존 로그인 방식의 문제점을 개선하기 위한 다양한 로그인 방법들이 연구 중이다.

2.1 로그인 클라이언트 다운로드 방식[2][8]

로그인 클라이언트 다운로드 방식은 기존 Web 에서 바로 사용자 인증을 거치지 않고 웹 호스트가 제공하는 로그인 클라이언트를 제공받아 해당 클라이언트를 통하여 인증을 받게 된다. 해당 방법은 로그인 방식에 자체 보안적 요소를 적용 할 수 있는 장점이 있으며 사용자가 원하는 PC 에 설치하여 사용 가능하다.

하지만 해당 로그인 클라이언트를 다운받아야 하는 경우 제공업체 또는 우회적 다운로드 경로를 이용하여 문제가 있는 클라이언트가 배포되는 경우가 발생할 수 있다. 그리고 현재 사용되는 윈도우 계열의 OS 에서 대부분 동작하게 되어있는데, 현재 윈도우 기반 PC 외에도 MacOS, 안드로이드, ios 등의 여러 OS 의 사용 비율이 높아짐에 따른 문제점이 있다.

2.2 인증서 기반의 로그인 시스템[3]

인증서를 사용하는 방식은 해당 인증서가 불법 복제 및 수정이 어렵기 때문에 보다 안전한 로그인 환경을 사용할 수 있는 매개체가 된다. 현재 대부분의 금융관련 시스템에서 사용되고 있으며, 일부 로그인 시에도 인증서 기반의 로그인 시스템을 사용하고 있다.

하지만 이 기법은 인증서를 사용자가 관리해야 하며, 다른 디바이스에서 로그인하는 경우 인증서를 복사하거나 새로 발급받아야 하는 번거로운 점이 있다. 또한 피싱으로 인하여 노출된 개인정보로 인증서를 발급 받는 경우 금융보안사고를 유발 할 수 있는 위험성이 있다.

2.3 일회용 암호를 이용한 암호 인증 시스템[4]

일반적으로 사용자 로그인을 하는 경우 사용자가 지정한 ID/Password 로 로그인을 한다. 이러한 로그인 과정에서 해당 Password 값을 1 회에 한하여 인증하는 OTP(One Time Password) 방식이 있다. 이러한 방식은 미리 지정된 디바이스나 사용자 인증을 통하여 매번 갱신이 되기 때문에 분실 및 스니핑 등에 대한 Password 유출과 관련된 사고에 대처방안이 될 수 있다.

하지만 이와 같은 방식은 매번 Password 를 갱신받아야 하는 매체가 별도로 필요하며, 해당 연결을 위한 인프라와 디바이스, 또는 Password 를 받을 수 있는 방안이 필요하다.

2.4 바이오 정보를 이용한 인증 방안[5][6][7][12]

바이오 정보를 이용한 인증 방안은 각 개인이 갖는 생체적 특징의 정보를 디지털화하여 인증에 사용하는 방안으로 지문, 각막, 혈관 지도, 목소리 패턴 등의 각 개인마다 달리 갖는 바이오 정보를 인증에 사용하는 방안이다. 해당 방안은 개인이 갖는 바이오 정보를 기반으로 인증을 거치기 때문에 보안적 특성이 강력하지만, 바이오 정보를 추출하거나 디지털화하는데 비용이 크다는 단점이 있다.

이러한 이제까지의 유저 인증방식을 정리하여보면 유저가 사용하는 클라이언트 디바이스에 보안모듈을 설치하거나 하드웨어적 장치의 설치, 혹은 서비스 제공자가 특정 사용자만을 인증하는 키를 보관하여 맵핑하는 방식 이였다.

본 논문에서는 기존 ID/Password 를 사용한 사용자 인증방식에 추가적으로 ID 를 기반으로 한 암호화된 key 정보를 받아 추가적인 factor 를 적용하여 인증받는 형태를 제안한다. 새로이 제안하고자 하는 모델은 다음과 같은 특징을 갖는다.

- 서비스를 이용하는 유저는 디바이스에 변경/제약 사항이 없다.
- 사용자는 원하는 어떠한 디바이스에서도 접근이 가능하다.
- 서비스 제공자는 기존 ID/Password 기반의 데이터베이스에 약간의 관계 테이블만 증가한다.
- 사용자는 주로 사용하는 디바이스를 등록하여 해당 디바이스 에서만 추가적 인증 없이 사용 가능하다.

3. 제안모델

본 연구의 제안모델에서는 기존 ID/Password 정보에 추가적으로 클라이언트 브라우저의 Local-Storage 값을 전달 받아 인증 시스템에 전달한다. 이 경우 유저는 해당 디바이스를 영구적으로 접근할 것인지 여부에 따라 디바이스의 정보를 시스템에 등록 하거나 임시

적으로 인증하여 접근하는 방식을 선택한다.[9]

3.1 사용자 지정 정보 받기

각 디바이스의 고유 값을 저장하기 위한 인자는 여러가지가 있을 수 있다. UDID 나 Serial Number 의 경우 고유하지만 최근 apple 사 에서도 보안 문제로 인하여 UDID 값의 사용을 지양하고 별도의 UUID 를 생성하는 것을 권장하듯이 별도의 구현된 Key 를 생성한다.[13]

모든 디바이스상에서 브라우저를 공통적으로 사용할 수 있다는 조건 하에, 본 연구에서는 임의로 사용자 ID 를 HTML5 의 Keygen 생성 인자로 사용하였다.

3.2 사용자 지정 정보를 Keygen 및 공개키 전달

HTML5 에서 Keygen 을 지원하며, 이를 사용할 시 암호화를 위한 개인키와 공개키 쌍을 만들어낸다. 생성된 공개키는 서버로 전송된다.

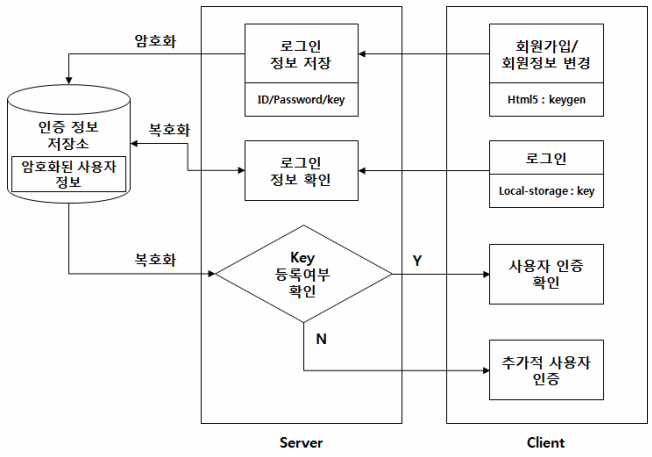
Key	ID	Password
keynum1	Username1	Password1
keynum2	Username2	Password2
⋮	⋮	⋮

Key	ID keygen(RSA)
keynum1	MIICQDCCASgwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCkX/um2G2zCzEoN47LwxS6c6Q3LJ+...
keynum1	MIICQDCCASgwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDgXZSrBWletEEub7bQuWjPajt2EyIXQ...
keynum2	MIICQDCCASgwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDa59lJ5wzhu9/6bTYD+I5lNr4dGpqv8...
⋮	⋮

(그림 1) ID 를 HTML5 keygen 을 사용하여 생성

3.3 인증 프로세스

사용자는 회원가입 또는 회원정보 변경 시 사용 빈도가 높고 개인적으로 사용되는 디바이스를 지정할 수 있다. 이 경우 공인인증서, 모바일 인증과 같은 사용자 인증단계를 거쳐 등록되며, 등록된 디바이스 인자값은 HTML5[11]의 keygen 을 사용하여 암호화되어 저장된다.



(그림 2) 인증 프로세스

4. HTML5 의 Keygen, Local-Strage 사용

앞서 제시한 인자값을 Loca-Storage 의 Key 로 저장하기 위하여 다음과 같은 기법을 적용할 수 있다.

4.1 인자값을 key 값으로 변경

HTML5 의 keygen 의 명세는 다음과 같다.

HTML Attributes

- autofocus = boolean
  Allows the author to indicate that a control is to be focused as soon as the page is loaded
- challenge = string
  A challenge string that is submitted along with the public key.
- disabled = boolean
  If present, make the control non-interactive and to prevent its value from being submitted.
- form = the ID of a form element in the element's owner
  Associate the keygen element with its form owner.
  By default, the keygen element is associated with its nearest ancestor form element.
- keytype = rsa
  The type of key generated.
- name = unique name
  Represents the element's name.

Example

- <keygen name="key" challenge="235ldahlae983dadfar">

(그림 3) W3C 에서 정의한 HTML 의 keygen[10]

4.2 Local-Storage 에 저장

생성된 key 값을 사용자 client 에 저장하기 위하여 HTML5 의 Web-Storage 기능 중 Local-Storage 기능을 사용하였다.[14]

```
<form id="pwd" name="keygen" action="echo.jsp" method="post">
 <p><keygen form="pwd" name="key" keytype="RSA"></p>
 <p><input type="text" id="name"/></p>
 <p><input type=submit value="Submit key..."/></p>
</form>
```

(코드 1) 인자값을 keygen 으로 암호화

```
<script type="text/javascript">
window.onload=function(){
 $('#stg_save').click(fn_stg_save);
 $('#stg_search').click(fn_stg_search);
 $('#stg_remove').click(fn_stg_remove);
}

function fn_stg_save(){
 var key = $('#stg_key').val();
 var value = $('#stg_value').val();
 localStorage.setItem(key,value);  // localStorage 저장
 $('#stg_key').val('');       // 입력값 삭제
 $('#stg_value').val('');    // 입력값 삭제
}

function fn_stg_search(){
 var key = $('#stg_key').val();
 var value = localStorage.getItem(key);
 $('#stg_value').val(value);
}

function fn_stg_remove(){
/*
 localStorage.clear(); 모든 localStorage 값을 일괄삭제
*/
    // 특정 key 의 value 값만 삭제한다.
```

```
    var key = $('#stg_key').val();
    localStorage.removeItem(key);
 $('#stg_key').val('');
 $('#stg_value').val('');
}
</script>
```

(코드 2) 전달받은 keygen 을 저장, 검색, 삭제하는 예제 코드

이렇게 저장된 key 값은 클라이언트가 유효한 디바이스인지를 식별하게 해주는 요소가 된다.

5. 비교분석 및 평가

본 연구에서 제안한 인증 방식은 사용자의 정보 및 ID/Password 가 누출된다 하더라도 사용자의 신분을 도용한 보안사고를 막을 수 있다는 장점이 있다.

또한 기존 쿠키의 단점을 보완하기 위한 HTML5 의 Web-Storage 기능을 또 다른 기능으로 응용하여 이용할 수 있다.

기존 인증 시스템과 비교 분석한 결과는 다음 표와 같다. 본 연구에서 제시한 Key 를 사용한 2-factor 인증 기법은 기존 인증 기법에 적용되는 방안과 함께 다중 디바이스를 별도로 등록하여 사용자가 원하는 디바이스 에서만 부가정보입력 없이 인증 받을 수 있도록 하고 있으며, 공용 디바이스 사용시에 별도의 부가정보입력 후 사용이 가능하도록 하였다. 또한 기존 방식에 추가적인 프로세스 설계와 데이터베이스 테이블 구성에 비용이 크지 않다는 장점이 있다.

<표 1> 기존 시스템과의 비교분석 및 평가

종류 / 기능	클라이언트 다운로드	OTP 인증	인증서 기반	제안기법
ID/Password 입력	O	O	X	O
부가정보입력	X	X	O	△
Multi-Factor 인증	X	X	X	O
구현에 필요한 비용	중	고	중	저

6. 결론

스마트 모바일 디바이스의 출현과 클라우드 환경, SNS 의발전은 사용자인증에 대한 보안적 측면의 발전이 요구된다. ID/Password 에 대한 주기적인 변경을 권장하고 이와 더불어 공인인증서 인증 같은 별도의 인증 방법이나 OTP 와 같은 2-factor 인증 방식 또한 기존 인증 방법을 강화하고자 하는 노력이다.

본 연구에서는 디바이스의 고유 기능이나 추가적인 소프트웨어의 설치 없이 구현이 가능하도록 하는 것과 저비용으로 기존 인증 방법을 강화하는데 의의가 있다. 특히 사용자가 사용하는 디바이스가 이전보다 제한적인 사용 패턴을 보이는 것에 착안하여 주로 사용하는 디바이스에서만 기존 ID/Password 인증 방식이 적용되도록 하고 이외의 디바이스에서는 기존 인증 방식과 더불어 추가적인 사용자 인증을 하도록 하였다. 이로 인하여 사용자의 신분을 사용한 SNS 상에서의 보안사고를 방지할 수 있는 방법을 제시하였다.

HTML5 는 아직 표준의 정의가 완료되지 않았고 기능 또한 모든 브라우저가 동일하게 적용되지 못하고 있다.

또, 브라우저 별로 Web-Storage 의 저장 영역이 다름으로 브라우저 전환 시에는 기존 데이터를 이용할 수 없다는 점에서 브라우저간 호환성에 대하여 개선이 필요하다.

향후 연구로는 Keygen 의 소스요소의 선택 시 고려할 사항들과 브라우저 이외의 어플리케이션에서도 Web-Storage 를 접근, 이용 할 수 있는 방안과 Web-Storage 에 대한 보호 방안을 연구를 해 보고자 한다.

참고문헌

[1] 이형우, '안전한 로그인을 위한 소프트 보안카드 기반 다중 인증 시스템', 한국콘텐츠학회논문지, Vol.9 No.3, pp.28-38, 2009.

[2] Barbara Ann Regnier, David Nicholas Youngers. Client/server computer system. having control of client-based application programs, and application-program control means

[3] Shimon Even, Oded Goldreich, and Silvio Micali. On-Line/Off-Line Digital Signatures. In Advances in Cryptology - Crypto '89, edited by Gilles Brassard, volume 435 of Lecture Notes in Computer Science , pages 263.277. Springer-Verlag, 1989.

[4] N. Haller, C. Metz, P. Nesser, and M. Straw. A One-Time Password system, IETF RFC 2289.

[5] A. K. Jain, A. Ross, adn S. Prabhakar, "Fingerprint matching using minutiae and texture features," to appear in the International Conference on Image Processing(ICIP), Greece, 2001(10)

[6] O. Peter, "Biometric generation of digital keys," Mini Symposium, DMIS-BUTE, 2001

[7] Y Sutcu, Q Li and N Memon, “Protecting biometric template with sketch: theory and practice”,IEEE Transactions on Information Forensics and Security , Vol. 2, No. 3 Part 2, pp. 503-512 ,2007.

[8] T. Weigold, T. Kramp, and M. Baentsch. Remote client authentication. IEEE Security & Privacy, 6(4):36–43,July 2008.

[9] 김영재, 웹 사이트 회원 가입시의 사용자 인증방법 1020040005134, Jan 27, 2004.

[10] HTML/Elements/keygen, http://www.w3.org/wiki/HTML/Elements/keygen(Sep 6, 2011), keygen

[11] 윤석찬외, “실전 HTML5 가이드” , 6

[12] 정현미, ‘모바일 클라우드 컴퓨팅환경에서 사용자 인증 방법 설계’, 2010 년도 한국멀티미디어학회 추계학술발표대회 논문집 제 13 권 2 호

[13] iOS Developer Library, https://developer.apple.com/library/ios/#documentation/UIKit/Reference/UIDevice_Class/DeprecationAppendix/AppendixADeprecatedAPI.html (Oct 12, 2011), Deprecated UIDevice Methods

[14] Web Storage, http://www.w3.org/TR/webstorage (Dec 8, 2011), Web Storage