JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 18, No. 4 : 867~875 Apr. 2014

난수 발생기를 이용한 이중화 구조의 안전한 비밀번호 생성 기법

서화정 · 김호원*

Two layered Secure Password Generation with Random Number Generator

Hwa-jeong Seo · Ho-won Kim*

Department of Computer Engineering, Pusan National University, Pusan, Korea

요 약

인터넷 서비스의 발전은 사용자가 언제어디서나 업무 처리가 가능한 인터넷 뱅킹 서비스를 가능하게 하였다. 하지만 인터넷을 통한 서비스 접근은 공격자에게 쉽게 사용자의 비밀정보가 노출될 수 있는 보안 취약점을 가지고 있다. 이를 방지하기 위해 현재 서비스 제공업체에서는 초기에 사용자의 아이디와 비밀번호를 이용하여 사용자를 인증하는 절차를 수행하게 된다. 하지만 현재 많은 사용자들이 짧고 단순한 비밀번호를 사용하며 주기적으로 비밀번호를 변경하지 않아 공격자의 전수 조사 공격에 의해 사용자의 비밀번호가 쉽게 노출될 수 있는 문제점을 가진다. 본 논문에서는 사용자의 비밀번호를 그대로 사용하지만 비밀번호가 지속적으로 적합한 보안 강도를 가지도록 실제 비밀번호를 보완해 주는 기법을 제안한다. 해당 기법은 추가적인 비밀정보를 이용하여 실제 비밀번호를 대체할 수 있도록 하며 주기적으로 비밀번호를 사용자의 실제 비밀번호에 대한 변경 없이 교체할 수 있다.

ABSTRACT

Rapid development of internet service is enabling internet banking services in anywhere and anytime. However, service access through internet can be exposed to adversary easily. To prevent, current service providers execute authentication process with user's identification and password. However, majority of users use short and simple password and do not periodically change their password. As a result of this, user's password could be exposed to attacker's brute force attack. In this paper, we presented enhanced password system which guarantee higher security even though users do not change their current password. The method uses additional secret information to replace real password periodically without replacement of real password.

키워드 : 비밀번호, 인증, 전수 조시

Key word : Password, Authentication, Brute-Force-Attack

접수일자 : 2013. 12. 22 심사완료일자 : 2014. 03. 19 게재확정일자 : 2014. 04. 01

* **Corresponding Author** Ho-won Kim(E-mail:howonkim@pusan.ac.kr, Tel:+82-51-510-1010)
Department of Computer Engineering Pusan National University, Pusan, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2014.18.4.867 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

임베디드 시스템의 급격한 성장은 사람들이 언제 어디서나 인터넷에 접속하여 자신이 필요한 서비스를 제공받는 것이 가능하게 하였다[1, 2]. 온라인 상에서의 사용자는 자신이 원하는 서비스를 보다 안전한 채널을 통해 제공받기 위해 서비스에 접속 시 자신의 신분과 비밀번호 정보를 서버에 전달하게 된다. 하지만 컴퓨터의 발전으로 인해 사용자의 비밀번호는 컴퓨터를 통한 전수조사에 보다 취약해 지게 되며 이를 방지하기 위해 사용자에게 점차적으로 복잡하고 긴 비밀번호를 요구하게 된다. 하지만 사용자에게 지속적인 비밀번호 수정을 요구하는 것은 사용자가 복잡한 비밀번호를 기억해야 하는 문제를 가지고 있다. 현재 권장하는 6개월이란 시간 안에 비밀번호를 변경하는 경우는 전체 인구의 7~80% 정도로 조사되고 있다. 이와 더불어 급속한 컴퓨팅 파워의 증가는 사용자에게 보다 짧은 시간 안에 비밀번호 변경을 권장하게 되는 풍속을 낳게 된다. 이를 해결하기 위한 다양한 보안 기법 가운데 가장 활발히 연구되고 있는 분야 중 하나는 OTP(One Time Password)를 이용한 사용자 인증 방법이다. OTP기법은 사용자에게 한정된 세션에서만 사용가능한 비밀번호를 사용하여 사용자를 인증하는 보안 시스템이다. 해당 기법의 인증요소에 따라 단일요소인증(1-Factor), 이중요소인증(2-Factor), 삼중요소인증(3-Factor)으로 구분된다.

본 논문에서는 사용자가 항상 동일한 비밀번호를 사용하지만 지속적으로 보안성이 향상된 비밀번호를 생성하는 기법을 OTP기법에서 착안하여 제안한다. 해당 기법은 추가적인 16바이트의 비밀정보를 통해 가능하며 이는 모든 플랫폼에 적용이 가능하기에 그 활용성이 매우 높은 기술이다.

본 논문의 구성은 다음과 같다. 2장에서는 비밀번호 및 제안하는 기술과 관련된 연구에 대해 살펴본다. 3장에서는 제안하는 비밀번호 관리 기법에 대해 제시하며 4장에서는 이에 대한 성능 평가를 한다. 마지막으로 5장에서는 본 논문의 결론을 내린다.

Ⅱ. 관련연구

본 장에서는 안전한 비밀번호를 위해 사용되는 현재 기법들과 제안 기법과 관련된 암호학적 알고리즘에 대해 상세히 설명한다.

2.1. 비밀번호

사용자는 자신이 원하는 서비스에 안전하게 접근하여 서비스를 제공받기 위해 자신의 신분정보와 비밀정보를 서버에 보내 사용자 인증 절차를 거치게 된다. 최근에 실시된 정보보호 실태조사서에 따르면 그림 1에서와 같이 사용자들은 개인 컴퓨터상에 비밀번호를 설정하여 사용하고 있음을 확인할 수 있다[3]. 여기서 자신의 정보를 안전하게 지켜주는 것은 보안성이 강한 비밀번호의 설정이다. 현재 그림 2에서와 같이 절반 정도의 인구가 자신의 비밀번호를 어려운 문자로 설정하고 지속적으로 교환해 주고 있음을 확인할 수 있다. 하지만 여전히 절반에 가까운 인구가 비밀번호에 특별한 조치를 취하고 있지 않고 있음을 확인할 수 있으며 여러 서비스에 대해 하나의 비밀번호를 사용하는 사용자가 90%에 이른다. 또한 그림 3에서와 같이 약 20%의 인구는 1년에 한 번씩 번호를 바꿈으로써 보안성이 개인정보에 대한 보호 수준이 낮아지고 있음을 확인할 수 있다. 최근에 CBS에서 실시한 가장 많이 사용되는 비밀번호에 대한 조사에서는 아래 예시와 같이 보안성이 낮은 비밀번호가 사용되고 있음을 확인할 수 있다.

1. password, 2, 123456, 3. 12345678, 4. abc123, 5. qwerty, 6. monkey, 7. letmein, 8. dragon, 9. 111111, 10. baseball, 11. iloveyou, 12. trustno1, 13. 1234567, 14. sunshine, 15. master, 16. 123123, 17. welcome, 18. shadow, 19. ashley, 20. football, 21. jesus, 22. michael, 23. ninja, 24. mustang, 25. password1

이는 공격자가 사회공학적 기법을 통해 분석 시 보안성이 매우 약해지게 된다. 따라서 사용자는 복잡하고 다양한 비밀번호를 사용해야 공격자로부터 안전하게 개인 정보를 보호할 수 있다. 단순한 조합의 비밀번호를 사용하게 될 시에는 공격자에 의해 쉽게 유추될 수 있기 때문이다. 이러한 비밀번호의 강도에 대한 연구는 표 1, 2에 나와 있다[4]. 먼저 다양한 조합을 사용하고 비밀번호의 길이가 길어질수록 가능한 키의 개수가 늘어남을 확인할 수 있다. 또한 이에 따라 키를 크래킹하기 위해 보다 많은 시간과 노력이 투여됨을 확인

할 수 있다. 이는 키의 길이와 조합의 개수가 늘어나서 이를 전수 조사하기 위한 경우의 수가 늘어나기 때문이다.

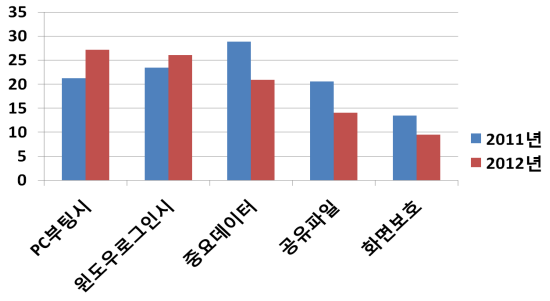

그림 1. 비밀번호 설정 현황
Fig. 1 Password Configuration

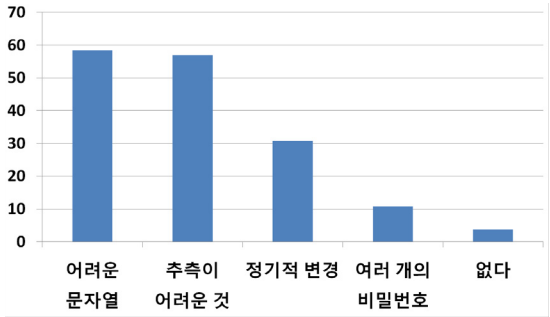

그림 2. 비밀번호 설정 및 관리 조치
Fig. 2 Password Management

최근에는 플러그인 형식으로 비밀번호를 관리해 주는 1password와 같은 프로그램을 통해 하나의 비밀번호를 입력하면 다른 모든 사이트에 대한 비밀번호를 관리해주는 프로그램이 출시되고 있다. 해당 프로그램은 사용자의 비밀번호에 대한 정보를 로그인을 하게 될 때 확인이 가능한 형식으로 복호화되어 제공된다. 하지만 해당 프로그램에서도 초기 로그인 시에는 기존의 비밀번호 방식을 사용하기 때문에 공격자가 전수공격을 통해 비밀번호를 알게 되는 복잡도는 동일하게 된다. 따라서 사용자들이 어려운 여러 개의 비밀번호를 사용하고 비밀번호를 주기적으로 바꾸기 위한 새로운 대책이 강구되어야 한다.

2.2. OTP 기술

서버와 OTP기기는 서로간의 동기화된 Seed값을 통해서 동일한 비밀값을 생성하는 기법이다. 따라서 기기들 간의 동기화를 맞추기 위해 다양한 방안이 사용되고 있다. OTP 동기화 기술은 시도 응답(Challenge response)방식, 시간 동기화(Time-Synchronous)방식 그리고 이벤트 동기화(Event-Synchronous)방식으로 크게 나누어 볼 수 있다[3].

시도 응답(Challenge-response)방식은 인터넷 뱅킹 이용 시 사용하게 되는 보안 카드와 같이 사용자가 서버로부터 제시되는 시도값을 얻은 후 그 값을 특정한 알고리즘에 넣어 수행한 뒤 나오게 되는 값을 응답으로 서버에 입력하여 정당한 사용자인지 아닌지를 확인하는 기법이다.

시간 동기화(Time-Synchronous)방식은 서버와 OTP 단말기는 시간으로 서로 동기화되어 특정한 시간 간격에 따라 다른 OTP를 생성해 내는 방식이다. 해당기법의 단점은 동기화된 시간을 짧게 잡을 경우 키 입력 시에 시간을 벗어나는 문제점과 반대로 시간을 너무 길게 잡을 경우 공격자가 OTP값을 알아챌 수 있는 가능성이 높아진다는 점이다. 또한 OTP단말기와 서버간의 시간 동기화가 추가적으로 수행되어야 한다[5].

이벤트 동기화(Event-Synchronous)방식은 서버와 OTP단말기는 서로 간에 공유된 카운터를 유지하며 해당 값에 따라 OTP값을 생성하게 된다. OTP값이 생성되면 카운터가 증가하게 되고 이를 이용하여 새로운 카운터값을 생성하게 된다. 하지만 해당 카운터값이 어긋나게 되는 경우 서버와 OTP단말기의 동기화가 다시 이루어 져야하는 단점이 있다.

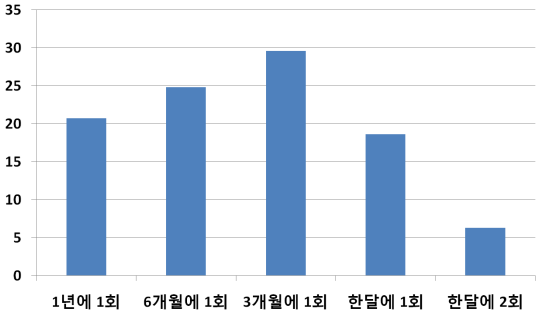

그림 3. 비밀번호 변경 주기
Fig. 3 Period of Password Refreshment

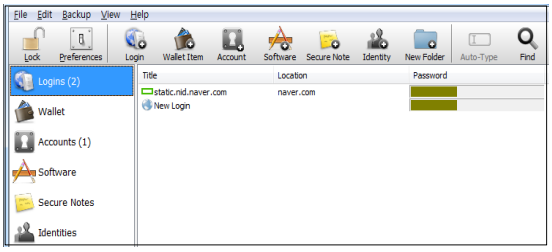

그림 4. 1password 프로그램 실행 화면
Fig. 4 Execution Screen for 1password Program

표 1. 키의 길이와 character에 따른 가능한 조합의 수
Table. 1 Number of Combination on Key size and number of character

Character Set	Password Length				
	4-octet	5-octet	6-octet	7-octet	8-octet
Lowercase letters (26)	4.6x 105	1.2x 107	3.1x 108	8.0x 109	2.1x 1011
Lowercase letters/digits (36)	1.7x 106	6.0x 107	2.2x 109	7.8x 1010	2.8x 1012
All alphanumeric characters (62)	1.5x 107	9.2x 108	5.7x 1010	3.5x 1012	2.2x 1014
Printable characters (95)	8.1x 107	7.7x 109	7.4x 1011	7.0x 1013	6.6x 1015
7-bit ASCII characters (128)	2.7x 108	3.4x 1010	4.4x 1012	5.6x 1014	7.2x 1016
8-bit ASCII characters (256)	4.3x 109	1.1x 1012	2.8x 1014	7.2x 1016	1.8x 1019

표 2. 100만개를 1초에 시도해 볼 수 있을 경우 모든 조합을 다 시도해 볼 때 걸리는 시간
Table. 2 Time for One Million Trials within a Second

Character Set	Password Length				
	4-octet	5-octet	6-octet	7-octet	8-octet
Lowercase letters (26)	0.5 sec.	12 sec.	5.2 min.	2.2 hours	2.4 days
Lowercase letters/digits (36)	1.7 sec.	1 min.	36.7 min.	21.7 hours	32.4 days
All alphanumeric characters (62)	15 sec.	15 min.	15.8 hours	40.5 days	7 years
Printable characters (95)	1.4 min.	2.1 hours	8.6 days	2.2 years	209 years
7-bit ASCII characters (128)	4.5 min.	9.4 hours	50.9 days	17.8 years	2283 years
8-bit ASCII characters (256)	1.2 hours	12.7 days	8.9 years	2283 years	570,776 years

2.3. Salt

일반적으로 사용자의 비밀번호는 해시된 형태로 데이터베이스에 저장되게 된다. 하지만 이는 공격자에 의해 해시된 값이 노출될 경우 보안에 매우 취약하여 salt라는 파라미터를 이용하여 비밀번호 값을 변경하여 해당 결과 값을 해시하여 데이터베이스에 저장하게 된다. 따라서 공격자는 데이터베이스에서 해시된 값을 획득하더라도 해당 값이 salt에 의해 한 번 더 변형이 되었기 때문에 비밀번호를 확인하는 것이 불가능하다.

2.4. 대칭키 암호화

메시지를 안전하게 암호화하기 위해 대표적인 대칭키 알고리즘인 AES를 사용되고 있다. 해당 알고리즘은 2001년에 NIST에 의해 선정된 알고리즘으로써 128 비트의 블록 크기를 가지며 128-, 192-, 256-비트의 키 길이를 통해 높은 보안 강도를 제공한다 [6, 7]. AES 연산은 안드로이드와 자바 상에서 간단히 javax.crypto API 호출을 통해 수행이 가능하다. 본 논문에서는 입력되는 메시지를 안전히 암호화하기 위해 128-비트 AES 암호화 연산을 하나의 메시지에 대해 수행하는 ECB 모드를 이용하여 수행한다. 만약 비밀번호의 길이가 16 자를 넘어서는 경우에는 여러 메시지에 대한 CBC 혹은 CTR 모드를 적용하여 전체 메시지의 기밀성과 무결성을 보전할 수 있다.

2.5. 난수 생성기

암호화에 사용되는 비밀 키 값은 공격자가 예상할 수 없는 임의의 값이 선택 및 사용되어야 한다. 따라서 비밀 정보의 생성은 난수 생성기와 같이 높은 암호화 강도가 제공되는 기법을 사용하여야 한다. 여기서 난수 생성기는 NIST에서 권장하는 block-cipher 방식의 DRBG 알고리즘와 이산대수의 강도를 가지는 Blum-Blum-Shub 알고리즘도 생각해 볼 수 있다 [8]. 본 논문에서 구현에 사용한 난수 생성기는 구현의 편의 상 임의의 난수 값을 rand함수를 통해 선택되었지만 난수 생성기는 하나의 모듈 형식으로 얼마든지 원하는 암호화 강도에 따라 DRBG 혹은 BBS로 변경하여 사용하는 것이 가능하다.

2.6. 아스키코드

아스키코드는 문자를 컴퓨터상에서 나타내기 위한

하나의 프로토콜로써 8비트 안에 일상적으로 사용되는 알파벳을 효과적으로 나타낼 수 있도록 한다. 일반적으로 키패드에서 입력하는 값은 문자를 표현하는 33~126에 해당하는 값이다. 이외의 값들은 특수한 기능키와 같은 값으로써 비밀번호에 사용되지 않는 값들이다. 따라서 실제로는 8비트 중에서 6~7비트 정보만이 사용되고 있음을 확인할 수 있다. 이는 보안적으로 매우 취약하며 이를 보완하기 위해 보다 긴 문자열을 비밀번호로 지정하거나 확장된 아스키 형식으로 암호화할 수 있는 기법이 연구되어야 한다.

2.7. 영 지식 증명 (zero-knowledge proof)

영 지식 증명 프로토콜은 1980년대에 [9]에서 처음 소개되었다. 영 지식 증명의 개념은 증명자가 검증자에게 자신을 인증할 때 사용하는 기법이며 인증을 위해 필요한 정보의 양을 0에 가깝도록 만드는 기법(Perfect Zero-Knowledge)이다 [10]. 증명자는 자신을 증명할 키 값을 전달하지 않고 검증자에게 자신을 증명할 수 있어야 한다. 키 값에 대한 유출 위험이 전혀 없기 때문에 프로토콜 상에게 특정 데이터 정보를 획득한다 하더라고 이를 인증에 사용하기는 힘들다 [11,12].

본 논문에서는 해당 영 지식 증명을 비밀번호 생성에 사용한다. 공격자는 사용자의 비밀정보 16바이트를 획득하더라도 이를 통해 사용자의 비밀번호를 바로 확인할 수 없다.

Ⅲ. 제안하는 비밀번호 관리 기법

본 장에서는 제안하는 비밀번호 관리 기법에 대해 소개한다. 본 장은 크게 시스템 구성, 시스템 플로우 그리고 시스템 알고리즘으로 나뉜다.

3.1. 시스템 구성

그림 5에서는 보안이 강화된 비밀번호를 생성하기 위한 시스템이 기술되어 있다. 먼저 사용자는 자신의 기존 비밀번호를 가지고 있으며 이는 해당 시스템을 통해 보안이 강화되게 된다. 먼저 시스템에서는 주기적으로 난수를 생성하게 된다. 해당 난수는 비밀값으로써 데이터베이스에 저장되게 된다. 만약 사용자가 자신의 비밀번호를 입력해야 하는 경우에는 기존의 비밀번호를 입력하여 난수 비밀값에 대한 암호화 혹은 복호화 연산을 수행하게 된다. 여기서 암호화와 복호화는 AES와 같은 블록 암호화 기법을 사용하게 되며 해당 결과값이 사용자가 사용하게 되는 보안이 강화된 비밀번호가 된다.

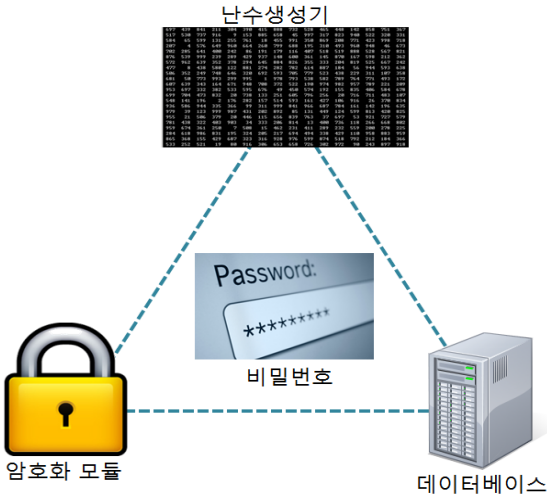

그림 5. 보안강화 비밀번호 시스템 구성
Fig. 5 System Construction for Secure Enhanced Password

3.2. 시스템 플로우

그림 6에서는 제안하는 시스템의 플로우를 자세히 나타내고 있다. 먼저 사용자는 초기에 난수를 생성하여 비밀정보를 생성하게 되며 이를 데이터 베이스에 저장하게 된다. 사용자가 홈페이지에 접속하는 경우 비밀번호를 입력하게 되며 해당 비밀번호는 이전에 생성된 난수를 암/복호화하는 데 사용되게 되며 여기서 도출된 결과값이 보안이 강화된 비밀번호로 사용되게 된다. 따라서 사용자는 초기화 단계에서는 생성된 보안 비밀번호를 통해 홈페이지에 접속을 해서 비밀번호를 변경한 이후부터 해당 기법이 적합하게 적용이 가능하다. 만약 이전에 초기화 과정이 수행된 경우에는 생성된 비밀번호를 통해서 서비스에 바로 접속을 시도해서 원하는 서비스를 제공받는 것이 가능하다. 해당 시스템을 통해 접속을 시도하는 순간마다 비밀번호를 바꾸도록 하는 기법도 가능하다.

만약 현재의 시스템에 접속 시 기존의 비밀번호 그리고 다음 비밀번호를 입력하도록 설계되더라도 사용자

는 자신의 고정된 비밀번호가 난수값에 의해 매번 보안이 강화된 비밀번호를 생성함을 확인할 수 있다. 따라서 해당 시스템은 공격자가 현실적인 전수조사가 불가능한 혁신적인 구조를 가진다. 여기서 제안하는 시스템은 지정된 플랫폼에서만 사용이 가능하다. 즉 사용자가 자신이 가진 스마트폰이나 PC 혹은 노트북이 아닌 곳에서는 접속이 불가능한 문제를 가진다. 하지만 보안을 위해서 공개적인 플랫폼에서는 그림 7과 같이 OTP를 해당 비밀번호를 통해 생성하는 것이 가능하다. 따라서 사용자는 언제 어디서나 자신의 세션을 안전하게 제공받는 것이 가능하다.

그림 6. 보안 강화 비밀번호 시스템 플로우
Fig. 6 System Flow for Security Enhanced Password

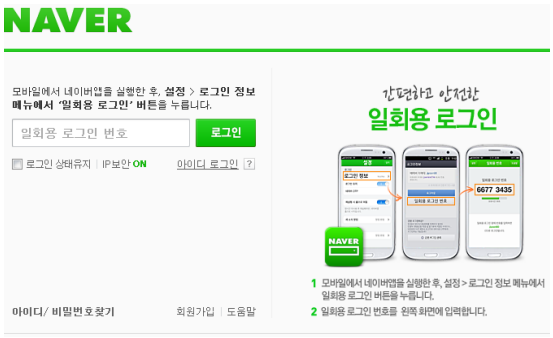

그림 7. 네이버 상에서의 OTP 로그인
Fig. 7 OTP login on Naver

알고리즘 1. 초기화단계	
입력: Seed 출력: 비밀정보(P)	
1.	Seed값을 난수 생성기에 넣음
2.	For t=0 to 15 do
3.	P[t] = 난수
4.	해당 난수를 데이터베이스에 등록
5.	End For

알고리즘 2. 비밀번호 생성	
입력: 비밀정보(P), 실제비밀번호(W) 출력: 보안이 강화된 비밀번호(O)	
1.	비밀정보(P)를 실제비밀번호(W)를 통해 암호화 혹은 복호화함
2.	생성된 보안이 강화된 비밀번호(O)가 로그인에 사용

3.3. 시스템 알고리즘

제안하는 기법은 크게 두가지 과정으로 수행된다. 먼저 난수값을 16 자리 형식으로 생성하게 되며 이를 데이터 베이스에 등록하게 된다. 사용자가 서비스에 접속하여 비밀번호를 입력해야 하는 경우에는 비밀정보를 암호화 혹은 복호화하여 나오게 되는 결과값을 로그인에 사용하게 된다. 따라서 사용자는 안전하게 16바이트의 비밀번호와 16바이트의 비밀정보를 통해 자신의 실제 비밀번호를 통해 생성할 수 있다.

Ⅳ. 성능 평가

본 장에서는 제안하는 기법의 성능에 대해 비교 평가한다. 먼저 보안성에 대해 평가하며 해당 기법을 적용했을 경우의 실제적인 수행 속도를 비교해 보도록 한다.

4.1. 보안성

4.1.1. 보안성 향상 비교

제안하는 기법의 성능을 확인하기 위해 자신의 비밀번호의 보안 강도를 확인해주는 사이트인[13]를 통해 취약한 비밀번호와 해당 비밀번호를 강화시킨 비밀번호를 각각 테스트해 보았다. 먼저 표 3에서와 같이 기존 비밀번호를 난수를 암/복호화하기 위한 킷값으로 이용하여 결과값을 도출해 보았다. 그리고 해당 값을 비밀번호 강도를 테스트하는 사이트를 통해 결과 값을 확인

해 보았다. 그림 8, 9에서와 같이 기존의 비밀번호는 즉시 크래킹이 되는 매우 취약한 비밀번호임을 확인할 수 있다. 하지만 제안하는 기법을 통해 보안이 강화된 비밀번호는 412 trillion year이 걸리는 매우 복잡한 비밀번호가 됨을 확인할 수 있다.

그림 8. 기존의 비밀번호 테스트
Fig. 8 Test on Traditional Password Model

그림 9. 보안이 강화된 비밀번호 테스트
Fig. 9 Security Enhanced Password Test

4.1.2. 비밀정보가 유출되지 않은 경우

해당 기법을 사용하게 될 때 보안성을 결정하는 요인은 크게 두 가지로 생각해 볼 수 있다. 먼저 비밀정보가 유출되지 않는 경우이다. 해당 경우는 자신의 플랫폼에 저장된 비밀정보가 어떠한 식으로든 유출되지 않은 경우를 의미한다. 본 논문에서 제안하는 기법은 어떠한 입력을 넣든지에 상관없이 128 비트 이상의 암호화 강도를 가지는 비밀번호를 만들어 낸다. 그 이유는 AES와 같은 블락 암호화의 경우 16바이트의 출력 값을 만들어내며 해당 값을 실제 비밀번호로 쓰게 될 경우 공격자는 128 비트에 대한 전수 조사를 시도해야 한다. 또한 본 기법의 보안성을 증가시키기 위해 AES256 혹은 다른 암호화 모드를 사용하여 128 비트 이상의 보안기술로 확장하는 것도 가능하다.

표 3. 기존의 비밀번호와 난수값 그리고 보안이 강화된 비밀번호
Table. 3 Traditional Password and Random Numbers and Secure Password

순위	기존	난수값	보안이 강화된 비밀번호
1	password	0x62,0xe9,0x0,0x66,0x27,0x37,0x7d,0xf9,0x91,0xd5,0x54,0xa3,0x44,0xe2,0xf6,0xa0,	hOHO>27yM") fW" C
2	123456	0xf2,0x27,0xea,0x6e,0xa8,0x1d,0x3,0x6c,0x12,0xec,0xa9,0x84,0xec,0x37,0x7f,0x1e,	I*zWc,W0R* Q&]v*B
3	12345678	0x40,0xeb,0x5b,0xa8,0x18,0x33,0x4c,0xaf,0xae,0xad,0x71,0xcd,0xef,0x5d,0x48,0xd7,	[D8\DF.)1p` AdBX(
4	abc123	0x99,0xb3,0xd9,0xb6,0x20,0x11,0x25,0xe2,0xad,0x7a,0xf6,0x84,0xc6,0x6b,0xab,0x10,	Nq%y+(\85a= Me+5g
5	qwerty	0x2d,0x28,0xc5,0x73,0x4a,0xce,0x1b,0xd6,0x33,0x25,0x20,0xeb,0xcc,0x10,0x7a,0xcc,	MW5,d(7zGT [a)Oyk
6	monkey	0xde,0x5c,0xb5,0xdc,0xbe,0x48,0x8b,0xcd,0xd9,0x6f,0xe9,0x7a,0xbf,0x34,0x78,0x78,	LBu`JC)1_ aDyW)ub
7	letmein	0x55,0x12,0x9b,0xee,0x9d,0x49,0x7,0xbd,0x3f,0x7b,0x72,0xb7,0xd8,0x5a,0x17,0x43,	<e.8[f)ETHMB o&CV
8	dragon	0xed,0xbf,0x87,0x61,0x71,0xde,0x6c,0x31,0xc9,0xba,0x70,0xa2,0x3b,0x4f,0x2c,0x3d,	]<MD=Pglvn *{-Fa,
9	111111	0xe,0x3a,0x5d,0x70,0x72,0xaa,0x79,0x18,0xcc,0xe7,0x1b,0x65,0x8b,0xf9,0x3d,0x39,	)M3=[w7.RVh [nj:*
10	baseball	0xbc,0xba,0x1c,0xd2,0x4b,0x85,0x9d,0xf1,0x4e,0x98,0x47,0x83,0x26,0xa8,0x92,0x74	,BRJ]WO.k#Z` gjSa

4.1.3. 비밀정보가 유출된 경우

비밀 번호가 유출된 경우 공격자는 해당 비밀정보를 통해 사용자의 비밀번호를 만들어 내어 서버상으로 전수 조사를 해보게 될 것이다. 표 2에서와 같이 사용자가 8자리의 숫자와 소문자를 사용하게 될 경우 약 30일이란 시간 안에 해킹이 가능하게 된다. 먼저 해당 30일이란 시간의 취약점을 없애기 위해 제안하는 기법을 사용해서 10일에 한번씩 자동으로 키 업데이트를 시켜주게 된다면 공격자는 절대로 패스워드를 확인하는 것이 불가능하다. 그 이유는 공격자가 비밀정보를 통해서 패스워드를 생성한 후 challenge & response 형식으로 전수조사를 수행하게 되는데 비밀정보가 변경되게 되면 이전의 비밀정보를 통해 생성되는 패스워드는 전부 유용하지 않은 정보가 되기 때문이다. 만약 자동으로 업데이트가 가능하도록 시스템을 설계하여 매시간 혹은 매분 새로운 키를 생성하게 된다면 공격자는 보다 키를 확인하는 것이 어렵게 된다.

4.2. 속도 & 크기

표 4에서는 AES 수행 속도를 분석하여 나타내고 있다[14].

표 4. armeabi (v7-A, Cortex A8); 2012 TI Sitara XAM 3359AZCZ100; 1 x 1000MHz 상에서의 Stream cipher 연산결과 [14]

Table. 4 armeabi (v7-A, Cortex A8); 2012 TI Sitara XAM3359AZCZ100; Computation Result of Stream Cipher on 1 x 1000MHz [14]

Cycles/byte for 8 bytes			
quartile	median	quartile	stream
187.75	189.00	190.00	chacha8
201.25	202.50	202.50	salsa208
202.75	203.75	208.50	salsa20
208.50	209.62	210.00	chacha12
221.25	221.25	222.75	salsa2012
228.50	233.62	235.75	cryptmtv3
248.75	250.00	250.25	chacha20
244.25	255.00	256.00	aes128estream
307.50	308.50	309.00	aes256estream
310.50	315.00	320.00	xsalsa20
347.38	348.88	352.12	nlsv2
733.25	736.00	739.00	tpy6
802.25	803.75	806.38	aes128ctr
1676.25	1677.50	1681.50	tpy
1681.00	1681.50	1686.12	tpypy

해당 리포트는 EBench에서 실시한 Stream cipher 암호화 성능에 대한 비교 분석 표이다. 표에서는 8바이트에 대한 암호화를 수행할 때 소모되는 클락 사이클을 나타내고 있으며 이는 스마트폰에서 많이 사용되는 Cortex A8 칩 상에서의 결과이다. AES128을 한번 수행하는데 소모되는 전체 clock cycle이 4,080이며 칩의 주파수가 1GHz일 때 일초에 약 245,098번의 암호화 연산 수행이 가능하다. 따라서 본 논문에서 사용하는 기법은 암호화를 한번밖에 사용하지 않으므로 스마트폰과 같은 장비에서도 큰 부하없이 운용이 가능하다. 이와 더불어 저장해야 하는 값이 암호화된 16 byte의 값이므로 이는 10 Giga byte의 용량을 가지는 스마트 폰에 큰 부담을 주지 않기 때문에 적합하다.

4.3. OTP와의 비교분석

본 논문에서 제시한 기법은 기존에 제시된 OTP와 같이 사용자가 알 수 없는 비밀번호를 생성해 낸다고 볼 수 있다. 여기서 제안된 기법과 OTP의 가장 큰 차이점은 OTP는 다른 채널 혹은 디바이스에서 생성된 정보를 통해 자신을 인증하는 기법이라면 제안된 기법은 자신이 사용하고 있는 디바이스 자체에서 자동으로 동작하여 사용자는 자신이 복잡한 비밀번호를 사용하고 있는지 인지하지 못하는 차이점이 있다. 따라서 사용자의 관점에서 보다 사용이 용이하다는 장점을 가진다.

또한 소프트웨어 형식의 OTP의 경우에는 해당 소프트웨어를 이용하여 다른 사용자도 동일한 비밀정보를 만들어 낼 수 있지만 본 논문에서 제안하는 방식은 해당 소프트웨어를 이용하더라도 사용자가 입력해야 하는 기본적인 비밀정보를 알 수 없기 때문에 동일한 비밀번호를 생성해 낼 수 없다. 즉 제안하는 기법을 사용하면 사용자는 간단한 비밀번호를 입력하더라도 서비스를 이용하는 사이트에는 복잡한 비밀번호를 사용하게 되는 효능을 가지게 된다.

Ⅴ. 결 론

본 논문에서는 비밀번호를 자동으로 복잡한 조합으로 변경하여 주는 기법을 제안한다. 해당 기법은 사용자의 실제 비밀번호를 통해 암호화된 가상화된 비밀번호를 통해 주기적으로 변경되는 비밀번호를 나타낸다.

해당 기법은 비밀번호가 노출되더라도 가상 비밀번호의 시드값이 변경되면 전수조사가 불가능해지게 된다. 만약 비밀번호가 자동으로 일주일에 한번씩 변경된다면 공격자는 사용자의 비밀번호의 복잡도가 떨어지는 조건에서도 비밀번호를 크래킹하는 것이 어렵다. 해당 기법은 현재 비밀번호 시스템에 바로 적용이 가능한 새로운 기술로써 그 활용도가 매우 높다고 할 수 있다.

감사의 글

"이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음"

REFERENCES

[1] Jegalbyungjik. "Trend of Mobile OS and Smart Phone Market," *Semiconductor Insight* 36, 2010.

[2] Kiyoung Kim, and Dongho Kang. "Smartphone Security for Open Mobile Environments," *KIISC* 19, no. 5, pp. 21-28, 2009.

[3] Korea Communications Commission, KISA, "Survey on Security of Current Trend of Public in 2012," 2012.

[4] Gary C. Kessler, "PASSWORDS – STRENGTHS AND WEAKNESSES," available at http://www.garykessler.net/library/password.html

[5] Rydell, Johan, Mingliang Pei, and Salah Machani. "TOTP: Time-Based One-Time Password Algorithm." 2011.

[6] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael," 1999.

[7] Sidorenko, Andrey, and Berry Schoenmakers. "Concrete security of the Blum-Blum-Shub pseudorandom generator." *In Cryptography and Coding*, pp. 355-375, 2005.

[8] Gjøsteen, Kristian. "Comments on Dual-EC-DRBG/NIST SP 800-90," 2006.

[9] Goldwasser, Shafi, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems." *SIAM Journal on computing* 18, no. 1, pp. 186-208, 1989.

[10] Changyoung Kwan, Hyunggyu Yang, Dongho Won, "Research on Zero-knowledge for Applications and Communication Verification," *KIISC* 2, no. 2, pp. 31-39, 1992.

[11] Wikipedia,"Zero-knowledge proof," available at http://en.wikipedia.org/wiki/Zero-knowledge_proof

[12] Rührmair, Ulrich, and Marten van Dijk. "Practical security analysis of PUF-based two-player protocols." *In Cryptographic Hardware and Embedded Systems,* pp. 251-267. 2012.

[13] "How secure is my password?," Available at https://howsecureismypassword.net

[14] EBench, "eBACS: ECRYPT Benchmarking of Cryptographic Systems," Available at http://bench.cr.yp.to/results-stream.html

서화정(Hwa-jeong Seo)

2010년 2월: 부산대학교 컴퓨터공학과 학사 졸업
2012년 2월: 부산대학교 컴퓨터공학과 석사 졸업
2012년 3월 ~ 현재: 부산대학교 컴퓨터공학과 박사과정
※관심분야 : 정보보호, 암호화 구현, IoT

김호원(Ho-won Kim)

1993년 2월: 경북대학교 전자공학과 학사 졸업
1995년 2월: 포항공과대학교 전자전기공학과 석사 졸업
1999년 2월: 포항공과대학교 전자전기공학과 박사 졸업
2008년 2월: 한국전자통신연구원 정보보호연구단 선임연구원/팀장
2008년 3월~현재: 부산대학교 정보컴퓨터공학부 부교수
※관심분야 : 스마트그리드 보안, RFID/USN 정보보호 기술, PKC 암호, VLSI 설계, embedded system 보안, IoT