OTP를 이용한 디지털락에 인증시스템 설계

윤치웅*, 신승중*, 류대현*
*한세대학교 컴퓨터공학과
e-mail : chywoong@korea.com

Design of Authentication System for Digital Lock based on One-Time-Password

Chywoong Youn*, Seung-Jung Shin*, Dae-Hyun Ryu*
*Dept of Computer Science, Hansei University

요 약

전송되는 데이터는 항상 공격자에게 노출되어 있기 때문에, 간단한 아이디나 패스워드 기법을 이용한 인증 기법은 취약점을 갖는다. 따라서 인터넷과 같은 개방형 네트워크상에서 안전한 통신을 하기 위해서는 전송될 정보의 암호화가 필요하며, 이를 위해서 통신 상대자간에 공통으로 사용할 수 있는 키의 공유가 우선되어야 한다. 본 연구에서는 인증 요청자와 인증 검증자간의 암호화된 키를 공유할 수 있는 새로운 키 교환 프로토콜을 제안한다.

Ⅰ. 서론

정보화 사회를 맞이하여 인터넷은 정보를 얻기 위한 핵심적인 기술로 발전하였을 뿐아니라 인터넷의 보급과 함께 네트워크 기술의 급속한 발전은 다양한 서비스와 편의성을 가져다주는 이점뿐 아니라 불법적인 시스템 침입과 정보의 유출 등이 쉬워짐으로써 개인 정보보호 침해와 같은 역기능을 초래하였다. 따라서 최근에는 이러한 위험으로부터 정보를 보호하고 안전하게 통신을 하기 위한 방법으로 암호화 기법이 필수적으로 사용되고 있으며, 소프트웨어적인 방법만으로는 한계점을 가지게 되어 TPM이라는 하드웨어 수준에서 보안을 수행하려는 노력을 기울이고 있는 실정이다. 개방형 네트워크 상에서 암호화키를 공유하고 사용자를 인증하는 문제는 안전한 정보교환을 위해 해결해야 할 중요한 문제이며, 이 문제를 해결하기 위해서 보다 효율적인 프로토콜 개발이 필요하다.

본 연구에서는 디지털락과 컴퓨터간의 일회용 암호방식의 프로토콜에 대한 설계와 구현을 하고자 한다.

Ⅱ. 이론적 배경

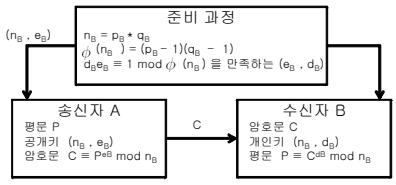
1. 사용자 인증

인증이란 특정 시스템의 서비스를 사용하기 위해서 사용자가 권한을 얻기 위한 절차를 말하며, 인터넷을 통하여 원격 시스템에 로그인을 한다거나, 전자적인 거래를 하려할 때 반드시 사용자 인증이 필요하다. 인증 서비스는 메시지에 대한 인증과 개체에 대한 인증으로 구분된다. 메시지 인증은 메시지의 원본이 정확하게 확인되고 그 확인이 잘못 되지 않았다는 확신을 얻기 위한 것이다. 개체 인증은 통신자의 신원 확인 절차로서 사용자 인증이라고 한다. 여기서 사용자 인증이란 사용자 A는 상대방에게 자신이 바로 사용자 A임을 증명할 수 있으나, 제 3의 사용자 C는 A로 위장하여 상대방에게 A임을 증명할 수 없음을 보증하는 기능을 말한다. 패스워드 인증 방식은 보안상으로 매우 취약성을 가지고 있어 보안성을 강화할 필요성이 있다. 이러한 패스워드 인증 기반의 암호 기술은 통신 객체들에게 패스워드 인증 방식을 이용하여 서로를 인증한 후, 암호키를 공유하도록 하거나 패스워드만을 이용하여 암호키를 생성할 수 있도록 한다. 이러한 패스워드 인증 기반의 암호 기술은 패스워드 인증 기반의 키 동의(PAKA: password-authenticated key agreement) 프로토콜과 패스워드 인증 기반의 키 검색(PAKR: password-authenticated key retrieval) 프로토콜로 분류될 수 있다.[9]

2. 공개키 암호

암호가 처음 사용되지 시작한 이후 1970년대 초반까지 암호 방식에서 대칭 암호만이 존재하는 것으로 알려졌었다. 이러한 대칭 암호는 키의 분배 및 관리의 어려움이 있다. 이러한 키의 분배 및 관리의 어려움은 1976년 W. Diffie와 M. Hellman의 논문[6]에서 계산적 복잡도가 암호 알고리즘의 설계에 응용되는 새로운 방향을 제시하면서 해결되었다. RSA 암호는 소인수 분해(Prime Factorization)의 어려움에 그 기반을 둔 공개키 암호로서 Diffie와 Hellman이 제안한 공개키 암호 시스템에 대한 개념을 가장 충실히 반영되었다. 당시 MIT 교수였던 Rivest, Shamir 그리고 Adleman 등에 의해 1978년에 설계되었다.[13] RSA

암호를 이용한 암호화 과정과 복호화 과정의 경우에 수신자 B가 공개키(n_B, e_B)와 개인키(n_B, d_B)를 구하여 공개키를 등록하고 송신자 B가 공개키를 가지고 평문 P를 암호화한 암호문 C를 보내고 B가 암호문 C를 복호하여 평문 P를 얻는 과정은 (그림 1)과 같다.



(그림 1) RSA 암호의 암호화 과정과 복호화 과정

3. 암호학적 해쉬함수

암호학적 해쉬 함수[5][12]는 현대 암호학에서 중요한 역할을 수행하고 있다. 해쉬 함수는 임의의 길이를 갖는 메시지를 입력으로 하여 고정된 길이의 해쉬값 또는 해쉬 코드라고 불리는 값을 출력한다. 해쉬 함수는 다양한 길이의 입력을 고정된 짧은 길이의 출력으로 변환하는 함수로서, 다음과 같은 식으로 표현할 수 있다.[10]

$$y = H(x)$$

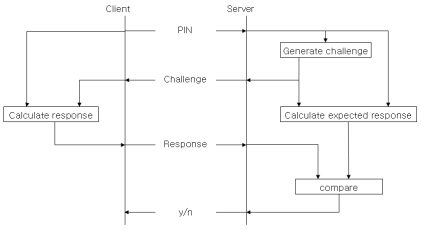
여기서 x는 가변 길이의 메시지이고, y는 해쉬 함수 H()를 통하여 생성되어지는 고정 길이의 해쉬값(hash code)이다. 전용 해쉬 함수에는 다음과 같이 MD4, MD5 및 SHA-1과 SHA-160 등이 있다.

4. 일회용 패스워드

일회용 패스워드(One-Time-Password)는 매번 다른 비밀번호로 사용자를 인증할 수 있도록 하는 인증 방식을 의미하며 현재 사용하는 비밀번호로부터 다음번에 사용할 비밀번호를 유추하는 것이 수학적으로 불가능한 특성을 가진다.[7] 일회용 패스워드 방식은 한번 사용한 패스워드는 재사용하지 않는 동적인 특성 때문에, 공격자가 네트워크 도청을 통하여 사용한 패스워드를 알아냈다 할지라도 더 이상 사용할 수 없으므로 비밀번호 노출의 위험을 방지할 수 있다.

4.1 비동기화 방식

비동기화 방식은 사용자와 인증 서버 간에 미리 설정된 동기화 기준 정보가 없이 인증 요청 시 난수와 같은 서버로부터 보내진 값에 의해 OTP 값을 생성하는 방식이다. 비동기화 방식의 대표적인 예가 요청-응답 방식이다. (그림 2)은 비동기화 방식 중 요청-응답 방식을 설명한다.



(그림 2) Challenge-response 방식

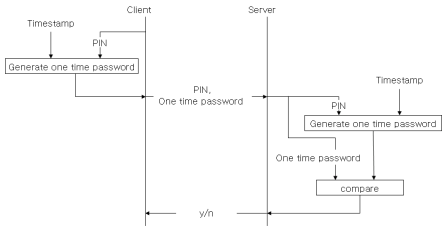
요청-응답 기법은 여러 번의 절차로 인해 다소 느리고 사용자가 직접 입력해야 하는 불편함과 인증 서버도 해당 사용자의 질의 값을 관리해야 하는 단점이 있다. 또한 인증 서버와 사용자 사이에서 서로 요청과 응답을 주고받으면서 상호 인증을 하므로 요청 값이 동일한 것이 반복적으로 사용될 때 보안문제가 발생된다.[4]

4.2 동기화 방식

동기화 방식은 사용자와 인증 서버 간에 미리 공유하고 있는 비밀 정보와 동기화 정보에 의해 OTP 값이 생성되는 방식이다. 동기화 방식은 동기화 정보에 따라서 시간 동기화(time-synchronous), 이벤트 동기화(event-synchronous), 조합 방식으로 나눌 수 있다.

1) 시간 동기화 방식

시간-동기화 방식은 해쉬 함수의 입력으로 비밀 값과 현재의 시간을 입력하는 방식으로, 서버 측과 사용자는 시간이라는 공통된 값을 가짐으로써 동기화 시킬 수 있다는 것에 착안한 방식이며, (그림 3)는 시간 동기화 방식에 대한 설명이다.

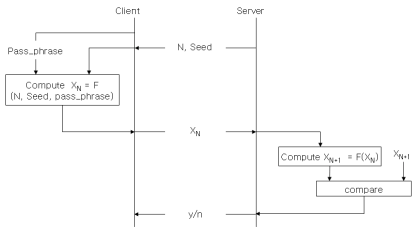


(그림 3) 시간 동기화 방식

사용자와 인증 서버간의 시간의 오차가 날 수 있으며 그 오차를 보정하는 방법이 이 방식의 핵심이 된다.[2]

2) 이벤트 동기화 방식

이벤트 동기화 방식은 사용자와 인증 서버가 동일한 카운트 값을 기준으로 OTP값을 생성하는 방식이다.
대표적인 이벤트 동기화 방식으로는 S/KEY 일회용 패스워드 시스템이 있다.[8] S/key OTP 시스템의 전체적인 동작 절차는 (그림 4)와 같다



(그림 4) S/key 방식

이벤트 동기화 방식은 인증을 여러 번 수행하게 되면 해쉬 함수 적용 가능 횟수가 하나씩 줄어들게 되므로 어느 시점에 다다르면 패스워드를 재초기화 시켜 주어야 하는 불편이 있으나, PC등에서 저장해 놓거나 미리 계산해 놓은 후에 사용할 수 있다는 장점을 가지고 있다.

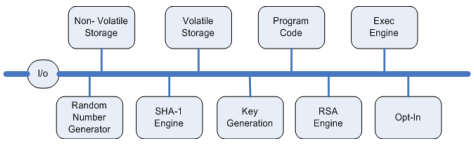
3) 조합 방식

시간 동기화 중심의 조합방식은 특정 시간간격마다 비밀번호가 다시 생성되며, 같은 시간간격 내에서 재시도시에는 카운트 값을 증가시켜서 비밀번호가 바뀌도록 하는 방식이다. 이벤트 동기화 중심의 조합방식은 특정 시간에 발생한 카운터 값을 기준으로 비밀번호가 생성되며, 사용자가 이벤트 버튼을 눌러 생성 요청을 할 때마다 값이 바뀐다. 이러한 조합방식은 기존의 시간 동기화 방식이나 이벤트 동기화 방식에 비해 보안성은 향상시킨 측면은 있으나, 인증서버와 사용자간의 동기화를 유지하는 것이 쉽지 않기 때문에 편의성 측면은 다소 떨어지는 부분이 있다.[3]

5. 신뢰 컴퓨팅

Trusted Computing은 특정 장소를 불문하고 언제 어디서나 사용 가능한 안전하고 신뢰할 수 있는 최상의 보안 서비스를 보장하는 통합 플랫폼 구축을 목표로 한다. 또한 플랫폼 하드웨어를 포함한 가장 하위 레벨에서부터 신뢰성을 검증하고 순차적으로 상위레벨로 확장함으로써 신뢰할 수 있는 플랫폼 기반의 사용자 보안을 고려하였다.

TC 기술의 핵심 구성요소인 TPM은 tamper proof circuit으로서 (그림 5)과 같이 구성된다.[11][14][1]



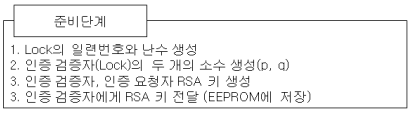
(그림 5) TPM 구조

III. 제안 프로토콜

본 절에서는 본 연구에서 제안한 OTP를 이용한 확인자 기반 키 교환 프로토콜을 소개한다. 제안한 프로토콜은 패스워드 프로토콜들이 만족해야하는 보안 요구사항을 만족하며, 다음과 같이 준비 단계, 키 생성 단계 그리고 인증 단계로 이루어진다.

1. 준비 단계

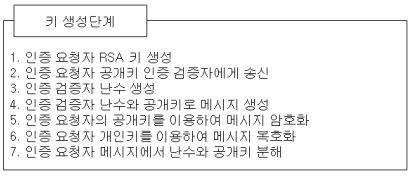
준비 단계는 마이크로컨트롤러(Microcontroller), 즉 디지털락(Digital Lock)이자 인증 검증자가 암호화를 하기위해 필요한 두 개의 소수를 생성하고 이를 바탕으로 RSA 암호화 키를 생성하여 이를 마이크로컨트롤러의 저장공간인 EEPROM에 데이터를 저장하는 단계이다. 준비 단계에 대한 내용은 (그림 6)과 같다.



(그림 6) 준비단계

2. 키 생성 단계

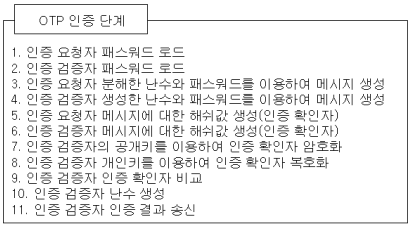
키 생성 단계는 인증 검증자는 저장 공간에 저장된 RSA 키를 로드하고 인증 요청자는 난수를 발생시켜 RSA 키를 생성하고 서로 교환하는 단계이다. 키 생성 단계에 대한 과정은 (그림 7)과 같다.



(그림 7) 키생성 단계

3. 인증 단계

인증 단계는 인증 요청자가 인증 검증자가 보내준 난수와 공개키를 이용하여 일회용 패스워드 즉, 인증 확인자를 생성하여 인증 검증자가 인증을 결정하는 단계이다. 인증 단계에 대한 흐름은 다음 (그림 8)과 같다.

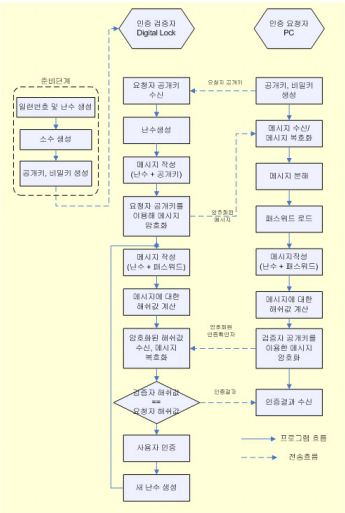


(그림 8) OTP 인증 단계

4. 프로토콜 처리 흐름

프로토콜 구현 결과는 세 부분으로 나뉘다. 첫 번째는 마이크로컨트롤러에 사용될 소수를 구하기 위해 일련번호와 난수를 생성하고, 이를 통해 생성된 소수를 통해 인증 검증자의 RSA 암호 키를 생성하고 마이크로컨트롤러에 전송하는 단계이다. 두 번째는 키 생성단계로 인증 요청자가 사용하게 될 공개키와 비밀키를 생성하고 서로에게 공

개키를 전달하는 단계이다. 세 번째는 인증 단계로 인증 검증자에 의해 전달된 난수와 인증 요청자가 가지고 있는 패스워드의 조합을 통해 인증 확인자를 생성하여 인증 검증자로 전송하고, 인증 검증자는 자신이 생성한 난수와 패스워드를 조합하여 인증 확인자를 생성한다. 인증 요청자는 해쉬 함수에 의해 생성된 인증 확인자를 인증 검증자의 공개키를 사용하여 암호화하여 인증 검증자에게 전송한다. 인증 검증자는 자신이 생성한 인증 확인자와 인증 요청자에서 수신한 암호문을 복호화 하여 얻어진 인증 확인자를 비교하여 사용자를 인증할 것인가를 결정한다. 인증 여부를 결정한 후 인증 결과를 통보하고 인증 검증자는 인증 확인자의 재사용을 막기 위해 새로운 난수를 생성하고 그 난수를 통해 새로운 인증 확인자를 생성한다. 다음 (그림 10)은 프로토콜의 처리 흐름도를 보여준다.



(그림10) 프로토콜의 처리 흐름도

Ⅴ. 결론 및 제언

인터넷의 보급과 함께 네트워크 기술의 급속한 발전은 사이버 공간상에서 방대하고 다양한 정보가 빠른 속도로 송·수신 될 수 있는 환경이 마련되었다. 개방형 네트워크 상에서 암호화키를 공유하고 사용자를 인증하는 문제는 안전한 정보교환을 위해 해결해야 할 중요한 문제이며, 이 문제를 해결하기 위해서 보다 효율적인 프로토콜 개발이 필요하다. 제안한 프로토콜은 인증 요청자와 인증 검증자 간의 일방향 함수 HAS-160의 해쉬 함수와 RSA 암호 알고리즘을 바탕으로 구현되었다. 또한 제안된 프로토콜은 준비단계, 암호 키 생성 단계 마지막으로 인증 단계로 세 단계로 구성되어 있다.

본 연구의 한계점으로는 첫째, 디지털락이라는 특정 제품에 적용함으로써 효율성보다는 보안성에 중점을 두었다는 점이다. 둘째, 보안성에 대한 검증이 정량적인 값에 의해 이루어지지 못하고 정성적으로 이루어졌다는 점이다. 추후의 연구에서는 본 연구가 가지는 한계점을 극복할 수 있는 보안성과 효율성에 대한 정량적인 분석을 추가적으로 수행함으로써 본 연구의 결과를 증명할 필요성이 있을 것이다.

참고문헌

[1] 김무섭, 신진아, 박영수, 전성익, 모바일 플랫폼용 공통 보안핵심 모듈 기술, 정보보호학회지, 제 17 권, 제 3 호, 2006. pp. 7-17.

[2] 김정재, 멀티미디어 데이터 보호를 위한 대칭키 암호화 시스템에 관한 연구, 숭실대학교, 박사학위논문, 2005.

[3] 서승현, 강우진, OTP 기술현황 및 국내 금융권 OTP 도입사례, 정보보호학회, 제 17권 제 3 호, 2007.

[4] 이근우, 오동규, 곽진, 오수현, 김승주, 원동호, 분산 데이터베이스 환경에 적합한 Challenge-Response 기반의 안전한 RFID 인증 프로토콜, 정보처리학회, 제 12 권, 제 3 호, 2005.

[5] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental cryptography : The case of hashing and signing," Advances in Cryptology-CRYPTO '94, LNCS, Issue 839, 1994, pp. 216-233.

[6] W. Diffie and M.E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory IT-26, No. 6, 1976, pp. 644-654.

[7] N. Haller, "A One-Time Password Standard", IETF RFC 1938, 1996.

[8] N. Haller, "The s/key one-time password system," RFC 1760, 1995.S

[9] IEEE P1363 Working Group, "P1363.2: Standard Specifications for Password-Based Public-Key Cryptographic Techniques", http://grouper.ieee.org/groups/1363/passwdPK/index.html

[10] R. Merkle, "One way hash functions and DES", In Advance in Cryptology CRYPTO'89, Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, 1989.

[11] S. Pearson, "Trusted Computing Platforms: TCPA Technology in Contest," Prentice Hall, 2003.

[12] B. Preneel, "Cryptographic hash functions," European Transactions on Telecommunications and related technologies, Vol. 5, No. 4, 1994, pp. 431-448.

[13] R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital Signatures and public-key cryptosystems," Communication of th ACM, Vol. 21, Issue 2, 1978, pp. 120-126.

[14] Trusted Computing Group: TCG Specification Architecture Overview. Specification Revision 1.4, 2007, http://www.trustedcomputringgroup.org