공개키 암호화 시스템과 일회성 패스워드를 이용한 사용자 인증 시스템 설계

이상준* · 김영길*

*아주대학교

The design of User authentication system by using Public key cryptography system and one time password

Sang-jun Lee* · Young-kil Kim**

*Ajou University

Email : trigger123@hanmail.net

요 약

시스템으로의 로그인 과정에서 명확한 사용자 인증은 정보 보호 서비스의 시작이다.
요즘과 같은 개방형 통신 시스템에서 패스워드라는 하나의 보안도구와 이를 지원하기 위한 시스템 내부의 메카니즘과 암호화 알고리즘 또한 빈약한 것이 사실이다.

이에 본 논문에서는 정확성과 안전성을 제공하는 사용자 인증 시스템을 구현하는 것을 최종 목표로 삼았으며 암호화 알고리즘으로는 RSA와 DES(Data Encryption Standard)의 CBC(Cipher Block Chaining)모드, 인증 프로토콜로는 신청과 응답(Challenge-Response)스킴을 사용하였으며, 인증 포로토콜의 안전성을 보장하기 위해 토큰(Token)을 사용하여 일회성 패스워드로 서버에 접속하는 사용자 인증 시스템을 설계하였다. 또한 공개키 암호화 알고리즘을 사용하여 보다 안전한 사용자 인증 시스템을 설계할 수 있었다.

ABSTRACT

In the process of Log-In to the system, clear User authentication is the beginning of the information protection service.

In the open communication system of today, it is true that a password as security instrument and the inner mechanism of the system and cryptography algorithm for the support of this are also poor.

For this reason, this dissertation had a final aim to design the user authentication system, which offer the accuracy and safety. It used RSA and CBC mode of DES as cryptography algorithm and used the Challenge-Response scheme as a authentication protocol and designed the User authentication system to which user access using one time password, output of token to guarantee the safety of the authentication protocol. Also by using the Public key cryptography algorithm, it could embody the more safe User authentication system.

키워드

난수, 토큰, 공개키, 개인키, 비밀키

Ⅰ. 서 론

일반적인 개방 통신망 환경에서는 적법한 권한을 가진 사용자라면 누구든지 임의의 서버에 접속해서 정보와 서비스를 제공 받을 수 있다. 하지만 해당 응용 서비스에 대한 접근 권한이 없는 사용자가 도청이나 위장등을 통해 임의의 응용 서비스에 접근해서 정보를 도용할 위험이 있다.

현재의 거의 모든 컴퓨터 통신 시스템이 개방 환경임에도 불구하고 실제적으로 패스워드라는 하나의 보안 도구를 사용할 뿐 그 이상의 강도 높은 안전한 서비스를 제공할 만한 보안 도구를 갖추지 못하고 있는 실정이고, 이를 지원하기 위

시스템 내부의 메카니즘과 암호화 알고리즘 또한 빈약한 것이 사실이다.

이에 본 논문은 정확성고 안전성을 제공하는 사용자 인증 시스템을 개발하는데 목적을 두었다. 기본적인 착상은 사용자가 시스템에 로그 인을 시도하면 인증 서버와 사용자는 신청-응답 스킴을 사용한다. 이 스킴에서 사용자가 응답을 서버로 보낼 때 서버의 개인키로 암호화하여 보내면 서버는 사용자의 공개키로 복호화하여 사용자에게서 온 응답(Response)와 서버에서 계산된 값을 비교하여 사용자를 인증한다. 결과적으로 3차에 걸친 사용자 인증을 거치게 된다.

첫 번째, USER PIN에 의한 1차 인증

두 번째, Response에 의한 2차 인증

세 번째, Private key와 Public key에 의한 3차 인증, 이로써 보다 확실한 사용자 인증을 할 수 있다.

II. 패스워드 방법의 단점

1. 패스워드는 평문으로 전달 되어진다.

대부분의 패스워드 인증 시스템에서는 사용자에 의해 입력된 패스워드는 패스워드를 요구한 컴퓨터로 네트워크를 통해 평문으로 전달된다. 이는 도청하기가 쉽다.

2. 패스워드를 추측하기가 비교적 용이하다.

사용자가 그들의 패스워드를 기억해야 하기 때문에, 기억하기 쉬운 패스워드를 선택하게 된다. 때문에 상대적으로 짧은 단어를 선택하게 되는데, 공격자들은 이를 쉽게 추측해 낼 수 있게 된다.

3. 인증이 단방향으로만 수행된다.

패스워드 스킴은 단방향으로만 수행 되어진다. 컴퓨터가 패스워드를 사용자에게 물어 볼 수는 있지만, 사용자는 그들이 정말로 정당한 컴퓨터와 통신하는 지의 여부는 알 수가 없다. 패스워드를 제공하기 전에 어떠한 종류의 서버 인증 테스트도 할 수 없는 것이다. 패스워드 입력 프롬프트는 공격자에 의해 설계된 가장된 응용 프로그램일 수도 있으며, 다른 컴퓨터에 의해 보여지는 속임수 일 수도 있는 것이다.

III. Intelligent Token을 이용한 Challenge-Response스킴

패스워드를 사용한 인증에 대한 공격을 극복하기 위해 일회성 패스워드를 사용한다. 이는 서버와 사용자가 같은 비밀키를 가지고 있는지를 확인하는데 그 목적이 있다. 본인 외에는 비밀키를 아는 사람이 없으므로 사용자 인증이 가능한 것이다. 본 논문에서는 스마트 카드를 하드웨어 토큰으로 사용한다. 이 방법은 1)사용자가 PIN을 인증 서버에 보내면, 2)인증서버는 난수를 생성하여 challenge로 사용자에게 전달한다. 3)그 즉시 서버는 이용자의 PIN에 해당하는 비밀키를 데이터베이스에서 꺼내어 이것을 이용하여 난수를 DES암호화하기 시작한다. 한편, 4)사용자는 서버로부터 받은 난수를 자신의 비밀키로 DES암호화하여 5)response로 인증서버에 반환한다. 6)인증서버는 사용자로부터 받은 response값과 자신이 계산한 값이 일치하면 사용자를 정당한 사용자로 인증하게 된다. 마찬가지로 사용자가 서버를 인증할 때도 같은 과정을 거치면 된다.

그림[1]은 이같은 과정을 보여주고 있다.

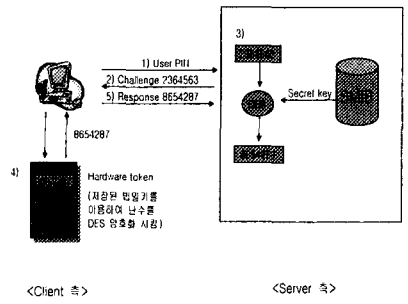

그림 1.

IV. Challenge-Respsponse Scheme의 문제점

만약 크캐커가 이용자와 서버사이의 의사소통을 엿듣고 있었고, challenge와 response를 기록하고 있었다고 하자. 크래커는 서버가 34를 challenge로 주었을 때 이용자가 2로 response 했다는 것을 안다. 만약 그 후에 서버가 또 한번 34로 challenge했을 때 크래커는 user에게 그것이 도착하기 전에 가로채어 이전에 기록한 2로 서버에 reponse한다.

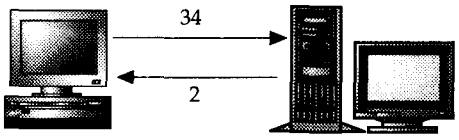

그림 2.

이것을 막기 위해 서버는 매우 큰 숫자 그룹에서 랜덤 숫자를 선택할 것이다. 하지만 엄밀히 말해서 이 수들은 랜덤하지 않다. 컴퓨터가 가장 어

려워하는 일이 바로 어떤 것을 랜덤하게 만드는 일이기 때문이다. 사실상, 컴퓨터 프로그램으로는 랜덤한 수를 만들 수 없다. 즉, 랜덤하게 보이는 수를 만들 수는 있지만 정말 랜덤한 수는 못 만드는 것이다. 따라서 이를 의사 랜덤이라고 한다. 가능한 한 알아내기 어려운 패턴을 가진 난수 생성기를 사용하는 것이 핵심이다. 그러나 아무리 패턴이 어렵더라도 컴퓨터 계산속도가 기하급수적으로 빨라지고 있는 현 시점에서는 이 패턴도 크래커에 의해 깨어지기 쉽다. 크래커가 난수 패턴을 알게 되면 사용자를 위장할 수 있을 것이다.

V. 공개키 방식 암호화의 이용

일단 위와같이 인증하게 되면 2차 인증까지 된 셈이다. 하지만 언급한 바와 같이 난수 생성기에 전적으로 의지하는건 위험부담이 따른다. 우리는 이제 개인키와 공개키를 이용할 것이다. 사용자는 PIN과 비밀키와 개인키를 가지고 있다. 또한 서버는 사용자 PIN, 비밀키, uer의 공개키를 가지고 있다. 사용자는 서버로부터 받은 난수를 비밀키로 DES암호화한 결과를 개인키로 다시 한번 암호화해서 보낸다. 서버는 이것을 공개키로 복호화한 결과가 서버가 계산한 값과 같으면 사용자를 정당한 사용자로 인증하는 것이다. 개인키는 본인외에는 가지고 있지 않으므로 다시 한번 인증이 되는 것이다(3차 인증). 크래커가 challenge에 대한 reponse를 알더라도 사용자의 개인키를 모르기 때문에 크래커 자신의 개인키로 암호화해도 서버 측에서는 사용자의 공개키로 복호화하기 때문에 전혀 다른 수가 나오게 된다.

그림[3]은 이같은 과정을 보여준다.

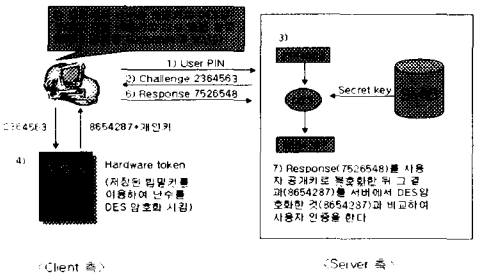

그림 3.

왜 하필 또 다른 비밀키로 암호화하지 않고 비밀키로 암호화 했냐고 묻는다면 해답은 서버가 해킹 당할 때를 대비해서이다. 만약 서버가 해킹당한다면 개인의 키들은 모두 도난당한다. 하지만 공개키 방식의 암호화 알고리즘을 사용한다면 서버에는 개인키는 없고 공개키만 있으므로 크래킹한 정보만으로 사용자 위장을 하지 못한다. 개인키는 본인만 가지고 있기 때문이다. 또한 개인키/공개키(비대칭키)를 사용하는 것이 비밀키(대칭키)를 사용하는 것보다 키를 해독하는 것이 훨씬 더 힘들다. 또한 비밀키/공개키의 해독은 현재로선 불가능하다. 이는 비밀키/공개키 개념이 당사자들은 풀기가 쉬운 문제를 공유하고 이 문제가 적에게 갔을 때는 풀기 어렵고 시간이 많이 걸리는 문제로 바뀌는 기본 사상에서 나왔기 때문이다.

VI. 사용자와 스마트 카드 사이의 인증

앞에서 사용자는 개인키, 비밀키, PIN을 가지고 있다고 했다. 이것들은 하드웨어 토큰 안에 있고 그 토큰으로는 스마트 카드를 사용한다. 따라서 스마트 카드와 사용자간의 인증이 한번 더 필요하다. 이 인증은 사용자가 스마트 카드를 분실했거나 도난 당했을 경우에 타인이 카드를 사용할 수 없게 하기 위한 절차이다. 스마트 카드는 메모리 카드와 마이크로프로세서 카드 두 종류가 있는데 우리는 인증 기능이 있는 마이크로프로세서 카드를 사용할 것이다. 이것은 여러 가지 응용프로그램과 패스워드를 가질 수 있으며 co-processor를 통해, 개인키를 알기 전에는 스마트 카드의 정보를 알 수 없도록 칩에 저장된 데이터를 암호화하여 가지고 있게 할 수 있다. 여기서 사용하는 스마트 카드에는 프로세서, DES chip, 기억 장소 등이 있다. 따라서 스마트 카드와 사용자 PC사이에서도 challenge-response 스킴을 사용하여 인증을 할 수 있다.

이 과정을 보면,

1) 스마트 카드는 난수를 생성하여 카드 리더기를 통해 사용자의 PC로 보낸다. 그와 동시에 카드 안의 비밀키와 DES chip의 알고리즘으로 난수를 암호화한다.

2) 사용자는 비밀키를 사용자 PC로 입력하고 PC는 입력된 비밀키와 PC안에 있는 DES알고리즘을 이용해 카드 리더기를 통해 온 난수를 암호화한다.

3) 사용자 PC는 그 결과를 스카트 카드로 보내고 스마트 카드는 자신이 생성한 값과 이것을 비교하여 일치하면 PIN을 사용자 PC로 넘겨준다.

그림[4]은 이같은 과정을 보여준다.

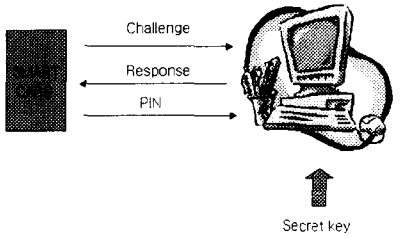

그림 4.

Ⅵ. 사용자 인증 시스템의 전체적인 설계

이제 전체적인 사용자 인증 시스템의 설계를 제시하고 스마트카드와 클라이언트사이의 인증과 클라이언트와 서버사이의 인증의 구체적인 내용을 살펴보자. 지금까지 살펴본 내용을 종합하면 이 인증시스템은 스마트 카드에서 시작하여 카드 리더, 인증 클라이언트를 거쳐 인증서버에 있는 SMIB(Security Management Information Base)에 접근하게 되며, 거기있는 해당정보를 이용하여 인증 작업이 이루어진다.

사용자 인증 시스템의 전체적인 수행과정을 정리하면 다음과 같다.

1) 인증 클라이언트 측에서 사용자가 본인의 스마트 카드를 삽입하고 패스워드(또는 지문)을 입력하여 카드에 대한 사용자의 인증을 스마트 카드가 확인한다.

2) 인증이되면, 스마트 카드는 카드 내에 저장되어있던 PIN(Personal Identification Number)을 카드 리더를 통해 인증 클라이언트에게 내준다.

3) 인증 클라이언트는 PIN을 인증서버로 보낸다.

4) 인증 서버는 PIN을 가지고 SMIB에 있는 비밀키를 찾아 클라이언트와 Challenge-Response를 수행한다.(클라이 언트는 response를 개인키로 암호화하여 서버에게 보낸다)

5) 서버는 클라이언트로부터 온 일련의 숫자를 클라이언트의 공개키로 복호화하여 그 결과를 가지고 사용자 인증을 하게된다. 이 과정을 그림[5]에서 보여준다.

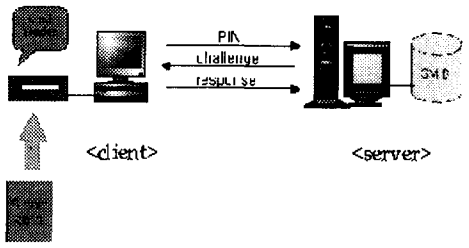

그림 5.

Ⅶ. 결 론

지금까지 본 것처럼 단순한 패스워드 시스템의 사용으로 인한 위험을 모두 제거시킬 수 있었다. 본 논문에서는 먼저 스마트 카드에서의 사용자 1차 인증, 인증 서버에서의 PIN을 이용한 2차 인증, 사용자로부터 온 response를 통한 3차인증, 또한 개인키 공개키를 통한 4차 인증을 통해 보다 강력한 인증을 실현시킬 수 있었다.

마지막으로 패스워드 시스템과 본 논문에서 제안하는 시스템과의 비교를 끝으로 본 논문을 마치고자 한다.

♣ 패스워드 기반 인증 시스템

< 특징 >
1. 암기위주의 짧은 패스워드
2. 평문으로 전송
3. 단 방향 인증
4. 1차 인증

< 위험 >
1. 추측 공격
2. 도청 공격
3. 위장 공격

♣ 본 논문에서 구현한 인증 시스템

< 특징>
1. 4차 인증
2. 비밀키가 네트워크를 통해 전송되지 않는다 (일회성 패스워드 사용)
4. 양방향 인증
5. 패스워드 기반 인증 시스템에서의 크래킹 위험 요소를 모두 제거

참고문헌

[1] Mel, H. X./ Baker, Doris M./ Burnett. SteveCryptography Decrypted. Addison-Wesley Pub Co (Sd) p.97-99 December, 2000

[2] 김동현. Web기반 응용 시스템에 대한 사용자 인증 방법 연구. 아주대학교. 산업공학과. 1999

[3] 은유진. 지능형 토큰을 이용한 사용자 인증 시스템 설계 및 구현. 아주대학교. 컴퓨터공학부. p15-20, 22-23. 1997.