확장된 DH를 이용한 안전한 키 교환에 관한 연구

고훈*, 장의진**, 신용태***
*대진대학교 컴퓨터학과, **(주)디지캡, ***숭실대학교 컴퓨터학과

A Study of Secure Key Exchange using expanded Diffie-Hellman

Hoon Ko*, Uj-jin Jang**, Yong-Tae Shin***
*Department of Computer Science Daejin Univ. **Digicaps
****Department of Computer Science Soongsil Univ.

요 약

인터넷상에서 많은 상거래 및 개인 정보들이 전송 되고 있다. 그러나 이런 정보들은 많은 위협에 노출되어 있다. 이를 해결하기 위해서 전자서명, 컨텐츠 암호 등 많은 방법들이 제안되고 있지만, 이런 기법들을 이용하기 위해서는 키의 생성, 분배, 전송을 위한 처리 모듈이 기본적으로 필요하다. 이런 모듈 중 전송 및 분배 시에 제3자를 이를 가로채어서 유용하는 문제점이 발생된다. 이에 본 연구는 기존의 키 교환 알고리즘인 Diffie-Hellman(DH)에 공개키 암호화 방법과 패스워드 방법을 추가하여 기존의 키 교환에서 발생된 문제점들에 대해서 해결 하고자 한다.

I. 서론

인터넷을 통한 전자상거래의 활성화를 위해서 중요한 것은 서로간의 확신, 즉 인증 그리고 안전한 정보 교환을 위한 키 분배가 먼저 해결되어야 한다. 최근의 공개키 기반구조(PKI : Public Key Infrastructure)에서는 공개키와 그 공개키의 소유자에 대한 문서인 인증서를 공인인증기관으로부터 받아서 상호간에 인증을 수행하여 주었다. 그러나 빈번한 인증서의 전송 등으로 인해서 네트워크 부하 및 속도 문제, 비용 문제 등 많은 문제점들이 발생되었다. 안전한 통신 채널은 두 당사자간의 상호인증과 세션키를 이용한 암호화 통신으로써 가능해야 하며, 안전한 보안 프로토콜의 개발과 동시에 효율성, 용이성, 편리성 등이 고려되어야 한다. 키 교환 알고리즘으로 가장 잘 알려진 Diffie-Hellman(DH)은 이산대수 알고리즘을 기반으로 한다. DH를 이용해서 키를 교환하는 과정에서 수신해서는 안 될 사용자가 수신했을 경우, 문제가 발생하게 된다. 이에 DH에 인증정보를 추가하여 공격자로부터 안정성을 제공하고자 한다. 본 논문에서는 기존에 제안된 DH 알고리즘에서 생성값에 공개키 암호화 방법과 패스워드 방법을 이용해서 생성값을 암호화해서 상대방에게 전송함으로써 DH를 사용하는 사용자가 제 3자에게 생성 값을 유출하지 않는 방법을 소개함으로써 수신자와 송신자가 안전한 키를 교환할 수 있는 방법을 연구하고자 한다. 본 논문은 Ⅱ장에서 기존의 키 교환 방법과 그 위험성에 대한 설명과 Ⅲ장은 제안하는 키 교환 모델에 대해서 설명하고 Ⅳ장에서는 제안하는 키 교환 모델의 안정성 대해서 분석을 Ⅴ장에서 결론 및 향후 연구 방안에 대한 설명으로 구성된다.

II. 키 교환 모델

DH알고리즘이라고 불리는 키 교환 알고리즘은 비밀키와 공개키를 생성하여 암호화와 복호화를 행하는 방식에 관한 알고리즘이 아니라 메시지를 주고 받으려는 두 명의 사람이 비밀리에 비밀키를 전달하기 위한 방법이다.

1. DH 방법

n, g : 크기가 큰 정수들로서 메시지의 송수신에 참여하는 모든 사람들에게 공개되어 있다.

① 송신자는 비교적 크기가 큰 난수 x를 발생시키고 이 값을 보관한다.

② 수신자는 역시 비교적 크기가 큰 난수y를 발생시키고 이 값을 보관한다.

③ 송신자는 다음의 계산을 하여 그 결과를 수신자에게 보낸다.

$$X = g^{x} \bmod n$$

④ 수신자는 다음의 계산을 하여 그 결과를 송

신자에게 보낸다.

$$Y = g^{y} \bmod n$$

⑤ 송신자는 Y를 받아서 다음의 계산을 한 후 비밀키 K_s를 얻는다.

$$K_s = (Y)^{x} \bmod n = g^{yx} \bmod n$$

⑥ 수신자는 X를 받아서 다음의 계산을 한 후 비밀키 K_r를 얻는다.

$$K_r = (X)^{y} \bmod n = g^{xy} \bmod n$$

즉 ⑤와 ⑥에서 계산된 결과인 K_s와 K_r이 같은 값을 갖는다는 것을 알 수 있다. 따라서 송신자와 수신자는 이 값을 비밀키로 하여 메시지를 암호화/복호화 할 수 있게 된다.

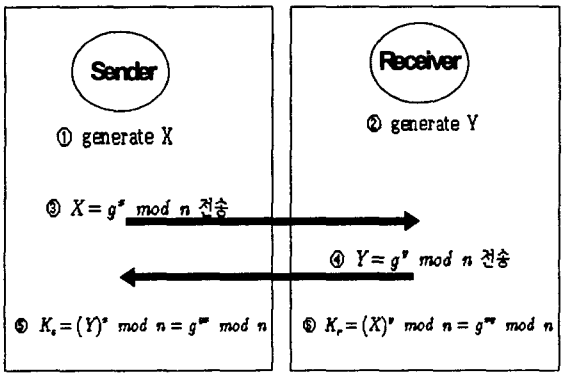

그림 1: DH를 이용한 키 교환 방법

2. DH 키 교환 방법의 위험성 분석

DH알고리즘에서 n이 충분히 클 경우 x, y를 구하는 것은 수학적으로 불가능(infeasible)하다라고 알려져 있다. DH알고리즘을 공략할 수 있는 공격방법이 없지는 않다. 즉, 알고리즘 자체는 수학적으로 안전하다고 할 수 있지만 암호 프로토콜 중에 허점(Security Hole)이 없는 것은 없다. 소위 흉내내기 공격법(impersonation attack)은 DH 키 분배 프로토콜의 공격 방법이다. 즉, 공격자가 송신자와 수신자의 사이에서 '송신자에게는 공격자 자신이 수신자인 것처럼' 그리고 '수신자에게는 공격자 자신이 송신자인 것처럼' 속이는 방법이다.

1) 공격방법

공격자는 송신자와 수신자가 주고받는 모든 데이터를 얻을 수 있다.

① 공격자는 송신자가 보낸 X값을 가로채서 보관하고 자신이 생성시킨 난수값 z을 이용하여 Z를 송신자와 수신자에게 보낸다.

$$Z = g^{z} \bmod n$$

② 공격자는 수신자로부터 Y값을 얻는다.

③ 송신자와 수신자는 자신들이 얻은 값 Z를 이용하여 다음의 비밀키를 생성한다.

$$Sender : K_{xz} = g^{xz} \bmod n$$

$$Receiver : K_{zy} = g^{zy} \bmod n :$$

④ 공격자는 ②에서 얻는 X, Y값을 이용하여 다음의 비밀키를 계산할 수 있다.

$$Secret\ Key\ for\ Sender : g^{zx} \bmod n$$

$$Secret\ Key\ for\ Receiver : g^{zy} \bmod n$$

결과적으로 K_{xz}와 K_{zy}라는 두개의 비밀키가 생성되어 전자는 송신자와 공격자가 통신을 하는데 사용하고 후자는 공격자와 수신자가 통신을 하는데 사용된다. 따라서 송신자와 수신자는 서로가 서로에게 메시지를 안전하게 주고받고 있다고 착각을 하게 된다.

III. 제안하는 키 교환 모델

기존의 DH를 이용한 키 교환 방법의 가장 큰 문제점은 교환을 하는 X, Y값의 위험성 및 값을 어느 누구나 인증 받지 않은 상태에서 사용 가능하다는 것에 있다. 따라서 본 논문에서는 서로 간에 생성해서 교환하는 X, Y값을 암호화해서 송수신하고 서로 간에 인증할 수 있는 패스워드를 처리하게 함으로 X, Y에 대한 비밀성을 제공하고자 한다.

1. Notation

- 송신자 / 수신자 : 통신자 상대방
- X : 송신자의 생성값
- Y : 수신자의 생성값
- P : 공통키 (패스워드)
- $\mathcal{P}(C_{PublicKey})$: 수신자의 공개키 인증
- $\mathcal{P}(S_{PublicKey})$: 송신자의 공개키 인증
- $\mathcal{E}_{C_{publicKey}}$: 수신자의 공개키로 암호화
- $\mathcal{E}_{S_{publicKey}}$: 송신자의 공개키로 암호화
- $\mathcal{D}_{S_{privateKey}}$: 송신자의 개인키로 복호화
- $\mathcal{D}_{C_{privateKey}}$: 수신자의 개인키로 복호화

2. 설정단계

① n, g : 크기가 큰 정수들로서 메시지의 송수신에 참여하는 모든 사람들에게 공개되어 있다.

② 서로 간에 정보를 전달하기 전에 세션에 해당되는 공동 패스워드를 소유하고 있어야 한다.

3. 실행단계

① 송신자와 수신자는 공통 패스워드 P를 교환한다.

② 수신자와 송신자는 서로간의 공개키를 생성해서 교환한다.

③ 송신자는 미리 공유한 패스워드 P를 이용해서 수신자 공개키 사용을 위한 인증을 받는다.

$$P(C_{PublicKey})$$

④ 수신자는 미리 공유한 패스워드 P를 이용해서 송신자 공개키 사용을 위한 인증을 받는다.

$$P(S_{PublicKey})$$

⑤ 송신자는 비교적 크기가 큰 난수 x를 발생시키고 이 값을 보관한다.

⑥ 수신자는 역시 비교적 크기가 큰 난수y를 발생시키고 이 값을 보관한다.

⑦ 송신자는 다음의 계산을 하고 그 결과를 수신자자의 공개키로 암호화 한다.

$$X = g^{x} \bmod n$$

$$E_{C_{PublicKey}}(X)$$

⑧ 수신자는 다음의 계산을 하여 그 결과를 송신자의 공개키로 암호화 한다.

$$Y = g^{y} \bmod n$$

$$E_{S_{PublicKey}}(Y)$$

⑨ 서로 간에 수신값들에 대해서 미리 정의한 패스워드를 입력한다.

⑩ 송신자는 Y를 받아서 송신자의 비밀키로 복호화 한 후, 다음의 계산을 한 후 비밀키 Ks를 얻는다.

$$Y = D_{S_{PrivateKey}}(E(Y))$$

$$Ks = (Y)^{x} \bmod n = g^{yx} \bmod n$$

⑪ 수신자는 X를 받아서 자신의 비밀키로 복호화 한 후, 다음의 계산을 한 후 비밀키 Kr를 얻는다

$$X = D_{C_{PrivateKey}}(E(X))$$

$$Kr = (X)^{y} \bmod n = g^{xy} \bmod n$$

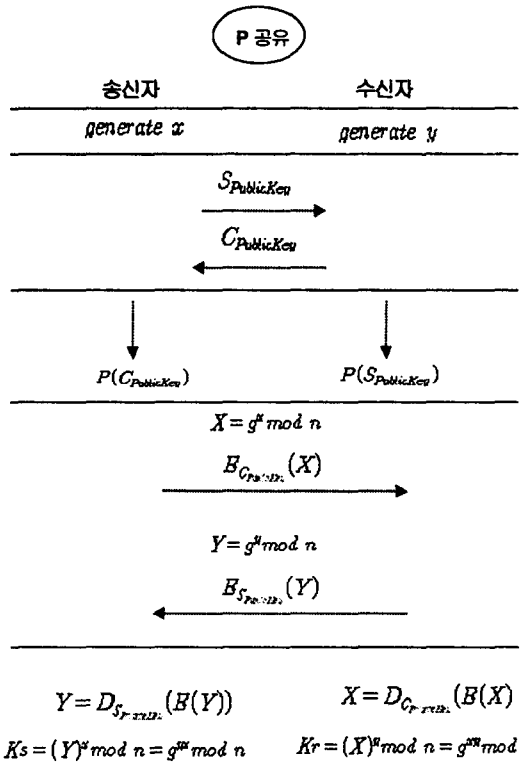

그림 2 : 제안하는 키 교환 모델

IV. 제안 모델의 안전성 분석

이번 장에서는 제안한 키 교환 모델에 대한 안정성을 기존 DH 키 교환 방법의 위험성에 대해서 보완하는 측면에서 살펴본다. 기존의 방법은 송신자와 수신자 사이에 송수신 되는 모든 정보에 대해서 공격자가 획득할 수 있다. 따라서 본 논문의 결과는 그러한 정보의 비밀성을 제공하는 측면에 초점이 맞추어져 있다. 기존의 DH 방법에 대한 위험성은 아래와 같이 세 가지로 분류 할 수 있다.

Case 1 : 공격자의 공개키를 이용할 경우

송, 수신자가 서로 간에 공개키를 전송 하는 과정에서 공격자가 공개키를 가로채어서 자신의 공개키를 마치 송, 수신자의 공개키인 것처럼 전송하는 경우이다. 그러나 이런 경우 송, 수신자가 공격자의 공개키를 수신 했을 경우라도 그 공개키를 이용할 경우 공통 패스워드를 먼저 처리해야 하는데 공격자의 공개키에 공통 패스워드 처리를 할 경우 경고 혹은 에러 메시지가 발생되기 때문에 송, 수신자는 서로의 공개키가 아님을 확인할 수 있다.

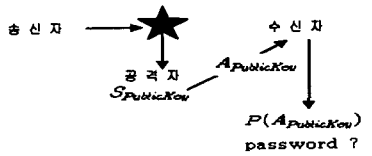

그림 3 : 공격자의 공개키를 이용할 경우

Case 2 : 송, 수신자의 공개키를 이용할 경우

이번의 경우는 공격자가 데이터의 전송로를 지키고 있다가 지나가는 공개키를 가로채어서 이용할 경우이다. 그러나 이러한 경우 송신자와 수신자는 서로간의 공통 패스워드를 교환한 상태이고 상대방의 공개키를 이용하기 위해서 공통 패스워드를 이용해서 서로간의 공개키를 사용할 수 있는 인증을 받아야 한다. 그러나 공격자의 경우는 공통 패스워드를 알지 못하기 때문에 송신자와 수신자의 공개키를 중간에 가로채었을 경우라도 사용하지 못한다.

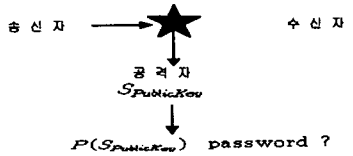

그림 4 : 송, 수신자의 공개키를 이용할 경우

Case 3 : 암호화 된 값들을 획득할 경우

송, 수신자 사이에 공개키를 이용한 암호화된 값들을 공격자가 받았을 지라도 송, 수신자의 비밀키를 가지고 있지 않기 때문에 복호화를 한다는 것은 불가능 하다.

그림 5 : 암호화된 값을 획득할 경우

V. 결론 및 향후 연구 과제

본 논문은 기존의 DH 키 교환 방법의 취약점에 대한 보완을 소개하였다. 제안한 키 교환 방법을 이용하면 CA기관을 이용하지 않더라도 상대방과 비밀성을 제공하면서 공개키의 무결성의 문제성 유무를 체크하면서 전달할 수 있다. 제안된 방법을 이용하면 사용자의 신분을 유일하게 확인할 수 있는 아이디와 패스워드를 기반으로 하는 모든 시스템, 즉 e-mail등에 적용 가능하다. 그러나 제안한 방법에 의존하면 패스워드 교환을 위해서 적어도 한번은 송, 수신자간에 만나야 하는 번거로움이 있다. 또한 아이디와 패스워드 기반의 시스템은 현실에서 발생하는 문제점들이 많다. 따라서 이 문제점들을 해결하기 위한 연구가 필요하다.

참고문헌

[1] 박영호, 박호상, 정수환, "패스워드 기반의 상호 인증 및 키 교환 프로토콜," 정보보호학회 논문집, 제11권 제5호, pp. 30-40, 2002년 10월.

[2] S. Kent and R. Atkinson, "Securiry Architecture for the Internet Protocol," IETF RFC 2401, November 1998.

[3] V. Boyko, P. MacKenzie & S. Patal, "Provabay Secure Password Authenticated Key Exchange Using Diffie-Hallman," Advances in Cryptology-EUROCRYPT 2000, Preneel, B., (Ed.), May 14-18, 2000.

[4] D. Jablon, "String Password-Only Authenticated Key Exchange," Computer Comminication Review, ACM SIGCOMM. Vol. 26, No. 5, pp.5~26, October 1996.