사용자 사이의 안전한 통신을 위한 메신저 설계

윤영준*, 표경환*, 신승수*, 한군희**
*동명대학교 정보보호학과, **백석대학교 정보통신학부
e-mail:pyo8728@nate.com

Design of Messenger for Secure Communication between Users

Yoeung-Jun Yoon*, Kyoung-Hwan Pyo*, Seung-Soo Sin*, Kun-Hee Han**
*Dept. of Information Security, Tongmyung University
**Division of Information & Communication Engineering, Baekseok University

요 약

기존의 메신저는 통신 내용 보안의 취약성으로 인해 서버관리자와 공격자로부터 사용자에게 피해를 주고 있다. 이러한 문제점을 보완하기 위해 현재 보편화된 메신저 프로그램인 네이트온 메신저의 채팅, 쪽지(메시지) 보내기의 패킷을 분석하고, 사용자의 채팅과 쪽지의 통신 내용을 보호하기 위해 보다 안전성이 높은 새로운 메신저 프로토콜 제안한다.

1. 서론

초고속 인터넷이 널리 보급되면서 오프라인에서만 가능했던 많은 서비스들은 이제 온라인에서도 사용할 수 있게 되었다. 현재 보편화된 온라인 메신저 중에서 특히 네이트온 메신저는 국내에서 가장 많은 사용자에게 서비스를 하고 있다[1].

네이트온 메신저 프로그램은 쪽지, 채팅, E-Mail, 화상대화, 원격 접속 서비스, SMS(Simple Message Service) 또는 MMS(Multimedia Messaging Service)를 통하여 휴대폰으로 메시지를 전송하는 서비스들을 이용 하거나 SSO(Single Sign On)기능 등의 수많은 서비스들을 제공하고 있다[2].

네이트온 메신저 프로그램은 사용자 ID 해킹과 각종 금융사고로 인하여 다양한 방법으로 기능을 추가하여 현재는 Nate On Messenger 4.0을 출시했다. 그러나 네이트온 메신저의 보안 문제점 중에 하나는 사용자가 친구에게 보내는 쪽지정보가 네트워크상에 그대로 노출되는 문제점을 확인 할 수 있었다.

이러한 문제점을 해결하기 위해서 사용자간의 메시지를 암·복호화 하는 메신저 프로그램의 프로토콜을 설계하였다.

본 논문의 구성은 다음과 같다. 2장에서는 네이트온 메신저의 공격기법에 대해서 분석하고, 3장에서는 새로운 프로토콜을 제안한다. 그리고 4장에서 제안한 프로토콜의 안전성과 효율성을 분석한 후, 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

네이트온에 관해 분석을 한 결과 여러 가지 문제점이 나타났다. 분석한 문제점은 다음과 같다.

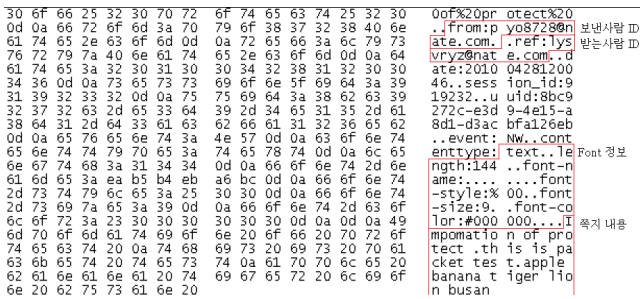
네이트온 메신저의 인증 메커니즘은 전송되는 사용자 정보를 암호화하여 전송하는 방법을 사용하고 있었다. 그러나 인증정보를 만들 때 동일한 사용자에 대해서 항상 동일한 인증정보를 생성하여 공격자가 자신의 신분을 위장하여 호스트가 보내는 패킷들을 공격자를 통해서 전송하게 하거나 공격 대상 서버가 공격자를 다른 사용자로 인식하도록 하는 스푸핑 공격 기법을 이용한 재전송 공격, 중간자 공격에 취약점이 있었다. 또한 인증 정보는 아이디와 패스워드를 조합하여 해시 알고리즘을 적용한 값으로 되어있다. 이는 공격자가 악의적인 목적을 가지고 네트워크 트래픽을 도청하는 스니핑 공격 기법을 이용하면 제3자의 공격자가 패스워드를 추출해 낼 수 있는 문제점을 가지고 있었다.

위와 같은 문제점들은 지난 2007년 2월 한국 정보 보호학회 논문을 통해서 발표되었다. 이후, 업체에서 보안업그레이드를 통해 해당 취약성들을 보완 하였다. 현재는 위와 같은 방법으로는 공격이 불가능하다[2].

하지만 재전송 공격과 패스워드의 취약점 외에도 네이트온 메신저의 보안 문제는 통신내용에서 나타났다. 사용자가 네이트온 메신저로 친구에게 보내는 통신 정보를 담은 패킷을 캡쳐 프로그램인 「Etherial Packet Capture」를 사용하여 네이트온 메신저의 패킷을 분석했다. 테스트는 영문으로 하고, 네이트온 메신저의

환경설정에서 “모든 패킷의 암호화”라는 메뉴를 클릭한다. [그림 2]와 같이 “Information of protect this is packet test apple banana tiger lion busan”라는 쪽지를 상대방에게 송신 하였다. 메시지 정보를 담은 패킷은 [그림 3]과 같이 송신자의 이메일주소, 수신자의 이메일주소, 송신자의 글꼴 정보, 쪽지 내용 등이 네트워크상에 그대로 송・수신 되는 것을 확인할 수 있었다. 이러한 문제점을 해결하기 위해 새로운 프로토콜을 제안한다.

[그림 2] 쪽지 전송 테스트



[그림 3] Nate On 패킷 수집

3. 제안한 프로토콜

안전한 메시지 통신을 위해 제안한 프로토콜에서 사용될 표기법을 [표 1]과 같이 정의한다.

[표 1] 표기법

기호	설명
ID_A	엘리스의 아이디
PW_A	엘리스의 패스워드
M_A	엘리스의 메시지
P_A	엘리스의 로그인 패스워드
MI_A	엘리스의 공개 개인정보
SI_A	엘리스의 비밀 개인정보
T_A	엘리스의 타임스탬프
e_A,n_A	엘리스의 공유키
d_A	엘리스의 개인키
$\oplus$	배타적 논리합
N_A	의사난수
$h()$	해시함수

기존 네이트온의 쪽지 패킷정보에는 사용자의 ID와 E-Mail주소, 사용자간의 데이터 패킷 등이 노출되어 송・수신 되는 것을 볼 수 있었다. 이러한 중요 정보들을 보호하고자 프로토콜을 제안한다. 프로토콜은 회원가입단계, 로그인단계, 쪽지・채팅 세션 키 전달 단계로 나누어 살펴본다.

3.1. 회원가입 단계

Alice는 회원가입 시 서버로 연결요청을 하고 서버는 Alice에게 1024bit의 공유키를 생성하여 송신한다. 수신한 서버의 공유키를 저장 한다. Alice는 아이디, 비밀번호, 이름, 닉네임, 주민번호, 전화번호, E-Mail, 우편번호, 상세주소, 자기소개서를 작성하고 회원가입을 하면 아래와 같다.

① MD5를 이용하여 $P_A = h(h(ID_A) \oplus h(PW_A))$를 먼저 계산한 후 $M_A = ID_A \| P_A \| MI_A$와 같이 자신의 아이디, 로그인 패스워드와 공개 개인정보(MI_A : 이름, 닉네임, E-Mail, 자기소개서)를 연접하여 M_A를 생성한다.

② $M_{A1} = M_A^{e_S} \bmod n_S$와 같이 M_A의 정보를 서버의 공유키인 e_A, n_A을 이용해 고속 지수 연산을 수행하여 암호문 M_{A1}를 계산한다.

③ 비밀정보(SI_A : 비밀번호, 주민번호, 전화번호, 우편번호, 상세주소)는 블록 암호 알고리즘인 ARIA를 이용하여 암호화를 수행한다. 암호화키는 Alice의 PW_A로 SHA-1을 사용하여 생성된 다이제스트 값을 암호화키로 사용한다.

④ $M_{A2} = h(M_{A1} \| T_A)$와 같이 M_{A1}의 정보를 Alice의 타임스탬프(T_A)와 연접 하여 MD5를 사용해 M_{A2}를 계산하고, $M_{A3} = M_{A1} \| M_{A2} \| SI_A \| T_A$와 같이 연접한 M_{A3}를 서버에게 송신한다.

⑤ 수신한 서버는 정보를 분해하여 $T_S - T_A < \Delta T_S$로 메시지 유효성 검사를 하고, $M_{A2} \equiv (M_{A1} \| T_A)$와 같이 무결성 검증을 한다.

⑥ 검증이 끝나면 $M_A = M_{A1}^{d_S} \bmod n_S$와 같이 서버 자신의 비밀 키인 $d_S n_S$로 암호화된 M_{A1}을 복호화 한다. 그리고 자신의 데이터베이스에 ID_A, MI_A, SI_A, P_A를 저장한다.

제안한 프로토콜에서는 서버의 공유키를 요청함으로서 Alice의 공유키는 노출을 하지 않고, 서버의 공유키만 노출 한다. 그 후 Alice의 아이디 해시 값과 패스워드의 해시 값을 배타적 논리합으로 실제 Alice의 패스워드값을 숨김으로서 공격자가 스니핑을 시도하

여도 암호학적 해시 함수의 일방향성 때문에 Alice의 원래 패스워드값을 알아내지 못한다. 그러므로 Alice의 비밀정보인 SI_A를 계산할 수가 없다.

3.2. 로그인 단계

Alice는 회원가입을 마치고 메신저 서버로 접속하기 위해 아이디, 패스워드를 입력하고 접속을 한다. 이러한 과정은 다음과 같다.

① $M_A = h(PW_A)$와 같이 Alice의 패스워드를 MD5를 이용해 M_A을 계산한다.

② $M_{A1} = ID_A \| M_A \| e_A, n_A$과 같이 아이디와 M_A, Alice의 공유키인 e_A, n_A을 연접하고, $M_{A2} = M_{A1}^{e_S} \bmod n_S$와 같이 서버의 공유키인 e_S, n_S을 이용해 고속 지수 연산을 수행하여 암호문 M_{A2}를 계산한다.

③ $M_{A3} = h(M_{A2} \| T_A)$와 같이 M_{A2}의 정보를 Alice의 타임스탬프(T_A)와 연접하여 다이제스트 M_{A3}를 계산하고, $M_{A4} = M_{A2} \| M_{A3} \| T_A$와 같이 3개의 정보를 서버에게 송신한다.

④ Alice의 로그인 정보를 수신한 서버는 $T_S - T_A < \Delta T_S$로 메시지 유효성 검사를 하고, $M_{A3} = h(M_{A2} \| T_A)$와 같이 무결성 검증을 한다.

⑤ 검증이 끝나면 $M_{A1} = M_{A2}^{d_S} \bmod n_S$와 같이 서버의 비밀 키 d_S, n_S로 M_{A2}를 복호화하여 데이터베이스에서 Alice의 ID_A를 찾아서 $P_A = h(h(ID_A) \oplus M_A)$와 같이 배타적 논리합을 하여 P_A값을 검증한다.

⑥ 검증이 끝나면 서버는 접속을 승인 하고 서비스를 시작한다.

공격자는 스니핑을 통해 로그인 정보를 획득할 수 있다. 하지만 Alice는 서버의 공유키로 M_{A1}을 암호화 하여 보낸다. 그러므로 공격자는 서버의 비밀 키 d_S를 모르기 때문에 계산할 수 없다.

3.3. 쪽지・채팅 세션 키 전달 단계

Alice는 Bob에게 쪽지・채팅을 신청을 하고, 중재자인 서버를 통해 세션 키 전달 과정은 다음과 같다.

① $M_A = ID_A \| ID_B$와 같이 Alice의 아이디와 Bob의 아이디를 서로 연접하여 M_A을 계산하고 $M_{A1} = M_A^{e_S} \bmod n_S$와 같이 로그인시 송신 받았던 서버의 공유키 e_S, n_S를 사용하여 고속 지수 연산으로 M_{A1}을 계산한다.

② $M_{A2} = h(M_{A1} \| T_A)$와 같이 계산된 M_{A1}과 T_A를 연접하여 해시 연산을 통해 M_2를 계산한다. 계산된 값은 $M_{A3} = M_{A1} \| M_{A2} \| T_A$와 같이 3개의 정보 연접한 M_{A3}를 서버에게 송신한다.

③ 서버는 Alice에게 송신 받은 M_{A3}를 분리하여 $T_S - T_A < \Delta T_S$로 메시지 유효성 검사를 하고, $M_{A2} \equiv h(M_{A1} \| T_A)$와 같이 무결성 검증을 한다.

④ 검증이 끝나면 $M_A = M_{A1}^{d_S} \bmod n_S$와 같이 서버의 비밀키 d_S, n_S를 이용하여 복호화된 자료로 데이터베이스에서 ID_B를 찾아 테이블에 저장된 ID_B의 공유키 e_B, n_B를 검색한다.

⑤ 검색된 e_B, n_B는 $M_S = ID_B \| e_B, n_B$와 같이 연접하여 생성된 M_S으로 $M_{S1} = M_S^{e_A} \bmod n_A$와 같이 Alice의 공유키로 고속 지수 연산을 수행하여 암호문 M_{S1}을 계산한다.

⑥ $M_{S2} = h(M_{S1} \| T_S)$과 같이 연접되어 MD5를 이용하여 M_{S2}값을 계산하고, $M_{S3} = M_{S1} \| M_{S2} \| T_S$와 같이 연접되어 Alice에게 송신한다.

⑦ Alice는 수신된 정보로 $T_A - T_S < \Delta T_A$를 계산하여 메시지 유효성 검증을 하고, $M_{S2} \equiv h(M_{S1} \| T_S)$와 같이 무결성을 검증한다.

⑧ 검증된 정보로 $M_A = M_{S1}^{d_A} \bmod n_A$를 계산하여 Alice는 Bob의 공유키인 e_B, n_B를 획득한다.

⑨ $M_A = ID_A \| N_A$와 같이 Alice의 아이디와 Alice가 생성한 난수를 연접하여 M_A을 계산하고, $M_{A1} = M_A^{e_B} \bmod n_B$와 같이 서버로부터 수신 받은 Bob의 공유키 e_B, n_B로 고속지수 연산을 수행하여 암호문 M_{A1}을 계산한다.

⑩ 계산된 정보는 $M_{A2} = h(M_{A1} \| T_A)$와 같이 MD5를 이용해 M_{A2}를 계산하고 3개의 정보를 $M_{A3} = M_{A1} \| M_{A2} \| T_A$와 같이 연접하여 M_{A3}는 Bob에게 송신된다.

⑪ Bob은 Alice로부터 수신 받은 정보를 분리하여 $T_B - T_A < \Delta T_B$로 메시지 유효성 검사를 하고, $M_{A2} \equiv h(M_{A1} \| T_A)$와 같이 무결성 검증을 한다.

⑫ 검증이 끝나면 $M_A = M_{A1}^{d_B} \bmod n_B$와 같이 자신의 개인키인 d_B, n_B로 고속 지수 연산을 통해 복호화를 하면 Alice와 Bob은 동일한 세션 키 N_A값을 가지게 된다.

쪽지・채팅 단계에서 Alice는 Bob의 공유키를 서버를 통해 수신하고 Alice는 Bob의 공유키를 이용해 세션 키인 N_A을 Bob에게 암호화 하여 보낸다. 공격자는 스니핑을 통해 수신자로 전달되는 패킷을 획득하더라도 Bob의 비밀 키를 알 수 없기 때문에 Bob에게 송신된 세션 키를 알 수 없다.

4. 제안 프로토콜 안전성 및 효율성 분석

4.1 안전성 분석

본 논문에서 제안한 프로토콜은 내부자 공격(Insider attack), 재전송 공격(Replay attack), 중간자 공격(Man in the middle attack), 전방향 안전성(Forward secrecy)에 대해 분석하였다.

○ 내부자 공격(Insider attack)

만약 서버 관리자는 내부 데이터베이스에서 악의적인 의도를 가지고 메신저 사용자의 주민번호, 집

주소, 전화번호 등을 모두 알 수 있었다. 하지만 제안된 프로토콜은 최소한의 사용자 정보만을 서버에 노출시키고 중요한 개인정보는 사용자만이 알고 있는 패스워드로 암호화를 하기 때문에 안전하고 서버로 전달되는 패스워드는 해시 함수를 거쳐서 전달되기 때문에 안전하다. 즉, 암호학적 해시 함수의 일방향성 때문에 서버 관리자라 하여도 사용자의 패스워드나 사용자의 개인정보를 추측 하거나 알 수 없기 때문에 내부자 공격에 안전하다[3].

○ 재전송 공격(Replay attack)

제안한 프로토콜에서는 타임스탬프(T)를 이용하여 메시지의 유효성을 검증하고 있다. 만약 공격자가 사용자의 메시지를 중간에서 가로채 저장하고 재전송 할 경우, 그 메시지는 처음 단계인 타임스탬프(T)에 대한 검증을 통과 할 수 없다[4]. 또한 공격자가 다른 정보는 변경하지 않고 T만 변경 할 경우 두 번째 단계인 무결성 검증을 통과 할 수 없다. 그러므로 공격자의 재전송 공격에 안전하다.

○ 중간자 공격(Man in the middle attack)

Alice와 Bob이 통신할 경우 공격자는 Alice와 Bob의 메시지를 중간에서 가로챌 수 있다. 하지만 3.3절에서 설명한 것과 같이 Alice는 중재자 서버를 통하여 안전하게 Bob의 공유키를 수신받기 때문에 중간자 공격에 안전하다[4].

○ 전방향 안전성(Forward secrecy)

제안한 프로토콜에서는 이전에 사용한 공유키나 세션 키의 정보를 저장하지 않고, 한 세션 마다 서버를 통하여 클라이언트간의 공유키를 생성하고 난수를 분배하기 때문에 안전하다[5]. 또한, 생성하는 난수는 ANSI x9.17 알고리즘을 이용한 강력한 난수를 제공한다. 그러므로 현재의 세션 키로 이전의 세션 키를 계산한다는 것은 불가능하다.

4.2. 효율성 분석

제안된 프로토콜에서는 회원가입 시 5회의 해시함수 연산과 2회의 지수연산 필요하다. 해시 함수의 연산은 10만번 연산 시 0.1㎲ ~ 0.2㎲의 시간이 걸린다. 그리고 1024bit로 RSA의 암·복호화 시간을 측정한 결과 암호화 시간은 0.0014002㎲로 나왔고, 복호화 시간은 0.0412006㎲로 비교적 빠른 시간으로 암·복호화를 하였다. 위 결과를 볼 때 로그인 단계와 쪽지 채팅 단계도 1초 미만으로 문제가 되지 않는다.

회원가입 후 사용자가 로그인을 하게 되면 서버는 사용자의 공유키, 사용자는 서버의 공유키를 가지게 된다. 로그인 단계에서 지수연산은 2회에 0.05㎲이다. 현대 컴퓨팅 기술에서는 연산속도에 영향을 미치지 않는다.

5. 결론

기존 네이트온은 사용자의 개인정보를 서버의 데이터베이스에 저장을 하기 때문에 내부자 공격에 취약하고, 클라이언트간의 통신내용도 그대로 송·수신 되었다. 이러한 정보 노출 문제점을 해결하기 위해 새로운 프로토콜을 제안하였다.

본 논문에서 제안한 프로토콜에서는 서버에게 사용자의 최소한의 개인정보만을 공개하고 중요한 개인정보는 사용자만 알고 있는 패스워드로 암호화되어 서버 데이터베이스에 저장 한다. 개인정보는 사용자의 패스워드로 암호화 되기 때문에 서버관리자 또는 제3자는 악의적인 의도로 사용자의 중요정보를 알 수 없다. 또한 네트워크상에 노출되었던 클라이언트간의 통신내용도 송·수신 시 암·복호화 되어안전한 통신을 할 수 있다. 본 논문에서 제안한 "사용자 사이의 안전한 통신을 위한 메신저 설계 및 구현"에서는 개인정보유출을 보호하는 다양한 응용분야에서 효율적으로 사용할 수 있을 것이다.

참고문헌

[1] 전웅렬, 원동호, 김승주. "국내 상용 제품의 인증 취약성 분석", 한국정보보호학회, 2009.

[2] 신동휘, 최윤성, 박상준, 김승주, 원동호. "네이트온 메신저의 사용자 인증 메커니즘에 대한 취약점 분석", 성균관대학교 정보통신공학부 정보보호연구소, 2007.

[3] Behrouz A.Forouzan. "Cryptography and Network Security", (주)한국맥그로힐, 2008.

[4] 임종인*, 이동훈**. "금융분야의 안전한 암호이용에 대한 연구", 2008.

[5] 박해룡, 전인경, 이향진, 최은영, 강연정, 이환진, 신동휘. "암호이용활성화 보고서", 한국정보보호진흥원, 2008.

[6] 노건태, 정익래, 이동훈, 변진욱. "암호화된 데이터에서의 연산 기술 분류 및 최근 동향 연구, 한국정보과학회 학술발표논문집, 2008.

[7] 윤종호. "네트워크 보안 프로토콜」, (주)교학사, 2007.