한국산학기술학회논문지
Vol. 11, No. 10 pp. 3942-3949, 2010

하이브리드 암호 시스템을 이용한 메신저 구현

한군희[1], 신승수[2*]
[1]백석대학교 정보통신학부, [2]동명대학교 정보보호학과

A Implementation of Messenger using Hybrid Cryptosystem

Kun-Hee Han[1] and Seung-Soo Shin[2*]
[1]**Division of Information & Communication Engineering, Baekseok University**
[2]**Dept. of Information Security, College of Information & Communication, Tongmyong University**

요 약 기존 네이트온은 사용자의 개인 정보를 서버의 데이터베이스에 저장을 하기 때문에 내부자 공격에 취약하고, 사용자간의 통신 내용도 그대로 송수신되었다. 이러한 정보노출 문제점을 해결하기 위한 안전한 메신저를 구현하였다. 본 논문에서 제안한 메신저에서는 서버에게 사용자의 최소한의 개인정보만을 공개하고 중요한 개인정보는 사용자만 알고 있는 패스워드로 암호화하여 서버의 DB에 저장한다. 서버 관리자 또는 제 3자는 악의적인 의도로 사용자의 중요한 정보를 알 수 없다. 또한 네트워크상에 노출되었던 통신 내용도 송·수신 시 암·복호화 되어 때문에 안전하게 통신을 할 수 있다.

Abstract Since existing Nate-on Messenger application stores users' personal information in the database of its server, it is extremely venerable to internal threats, not to mention the communication data being transmitted without any safety measures. To solve such problematic areas of the existing application, we have developed a safer messenger application. The messenger application proposed in this paper discloses only the least required personal information of its users and the rest of the personal information is safely encrypted in the database using private passwords. This protective measure prevents the administrator or a third party from misusing the information since he/she will not be able access the information. In addition, users will be able to freely and safely communicate using this new messenger since transmitted data will also be encrypted.

Key Words : Messenger, Nate-on, ARIA, RSA, MD5, SEED, Expansion P-Box

1. 서론

메신저는 실시간 의사소통 및 파일 공유를 지원해주는 소규모 통신 프로그램을 의미한다. 기업내부에서 사용되는 인터넷 메신저의 상당수는 해당 기업 내부 IT (Information Technology) 조직의 정책적인 지원과는 상관없이 직원 개개인의 선택과 사용에 의해 운영되고 있다. 인터넷상에서 무료로 배포되고 쉽게 설치 및 사용할 수 있다는 편리함으로 인하여 많은 사용자들로부터 환영받고 있다. 대부분의 인터넷 메신저 사용자는 기업내부에서 업무상 공유하는 통신 그룹보다 외부의 사용자와 더 큰 규모의 통신 그룹을 유지하게 된다. 이렇듯 인터넷 메신저는 기업 내·외부의 다양한 사람들과의 주요 통신 수단으로 자리 잡고 있다[1].

국·내외에서 많이 사용하는 메신저로는 MSN 메신저, Yahoo 메신저, AOL 메신저, 네이트온 메신저 등이 있다. 이들 메신저의 공통점은 바이러스나 악성코드의 전파 경로로 활용될 수 있거나, 이미 그러한 피해사례가 보고되고 있다는 것과 사용자간 의사소통에 대해서 별도의 암호화를 제공하지 않기 때문에 도청에 취약하다는 것이다[2].

메신저를 사용하는 기업입장에서는 다음과 같은 사항

*교신저자 : 신승수(shinss@tu.ac.kr)
접수일 10년 09월 01일 수정일 10년 09월 30일 게재확정일 10년 10월 15일

들에 주의해야한다. 첫 번째, 메신저에 대해서는 국제적으로 제정된 표준이 없으며 메신저 제공업체가 독자적으로 설계 및 개발한 프로그램을 그대로 사용한다는 것이다. 두 번째, 메신저는 정보보호, 에러체크 및 재전송 등에 대한 기능은 대부분 매우 취약하다. 세 번째, 기업 내부의 직원들이 임의적으로 사용하는 메신저에 대하여 기업의 정보시스템 관리자들이 이를 모두 파악하고 관리하기가 매우 어렵다는 것이다[3].

네이트온 메신저 프로그램은 사용자 ID 해킹과 각종 금융사고로 인하여 다양한 방법으로 기능을 추가하여 현재는 네이트온 메신저 4.0을 출시했다. 그러나 네이트온 메신저의 보안 문제점 중 하나는 사용자가 친구에게 보내는 쪽지정보가 네트워크상에 그대로 노출되는 문제점이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해서 클라이언트간의 메시지를 하이브리드 암호시스템을 이용하여 메신저를 구현하고 클라이언트간의 안전한 메시지를 교환하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 메신저에 대해서 분석하고, 3장에서는 안전한 메신저를 설계 및 구현을 한다. 그리고 4장에서는 구현한 메신저에 대하여 분석한 후, 마지막으로 5장에서 결론을 맺는다.

2. 기존 연구 분석

국내에서 많이 사용하는 메신저는 네이트온, MSN, 버디버디, 다음터치 순으로 많이 사용하고 있다. 많은 메신저 중에서 네이트온 메신저는 SSO(Single Sign On) 기능을 통하여 사이월드와 연동하여 사용할 수 있는 등의 수많은 서비스들을 제공하고 있다. 네이트온 메신저의 인증 메커니즘은 사용자의 정보들이 암호화되어 전송되도록 되어 있다.

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 제3자가 패스워드를 추출해 낼 수 있는 문제점을 가지고 있었다. 이와 같은 문제점들은 논문을 통해서 발표되었고 이후, 업체에서 보안업그레이드를 통해 해당 취약성들을 보완하였다. 현재는 이와 같은 방법으로는 공격이 불가능하다[4].

당시 네이트온에서의 사용자 인증정보는 ID(또는 이메일 주소)와 패스워드의 조합에 해시 알고리즘(MD5) 을 적용한 값이었다. 따라서 해시 값을 분석하면 사용자의 패스워드를 검출할 수 있다. 상용 프로그램인 “Passwordpro”를 사용하여 네이트온 메신저 패스워드를 해킹할 수 있었다. 이후 해당 업체의 네이트온 메신저 보안 업그레이드를 통해 해당 취약성이 보완되었으며 현재는 이러한 공격이 불가능하다[5].

하지만 재전송 공격과 패스워드의 취약점이 해결이 되었지만 네이트온 메신저의 보안 문제는 사용자간의 통신 내용이 그대로 드러나고 있다는 것이다. 사용자가 네이트온 메신저로 친구에게 쪽지나 편지에 관한 통신정보를 담은 패킷을 캡처 프로그램인 “Etherial Packet Capture”를 사용하여 네이트온 메신저의 패킷을 분석했다. 분석한 결과 송신자의 이메일 주소, 수신자의 이메일 주소, 송신자의 글꼴 정보, 쪽지 내용 등이 네트워크상에 그대로 송·수신되기 때문에 본 논문에서는 통신 내용이 그대로 노출되는 문제점을 해결하기위해 안전한 메신저를 설계하고 구현한다.

3. 메신저 설계 및 구현

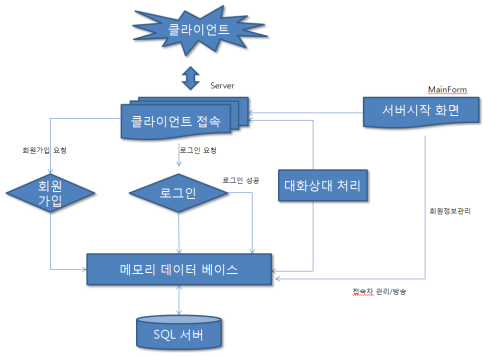
이 장에서는 네이트온 메신저의 보안 문제 중 심각한 클라이언트간의 통신 내용이 노출되는 문제점을 해결하기 위해 메신저를 설계하고 구현한다.

3.1 메신저 구조

메신저는 크게 일반 사용자가 접하는 메신저 클라이언트와 관리자가 사용하는 메신저 서버로 구성되며 중앙에 회원관리 및 로그인 처리를 담당하는 서버가 존재한다. 메신저 클라이언트는 유동 IP 주소를 사용해도 상관없지만 메신저 서버는 가급적 IP 주소가 변하지 않는 고정 IP 주소를 갖고 있어야 하며, 회원가입/관리를 위해 SQL 서버와 연결되어 있어야 한다.

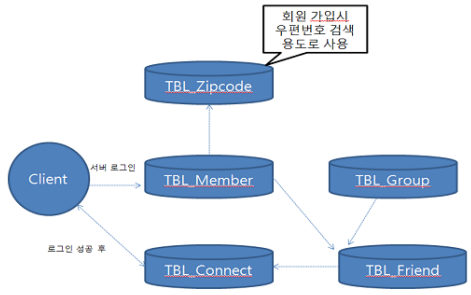
메신저 서버는 4개의 테이블을 메모리에 생성해 놓고 SQL 파일에서 데이터를 읽어와 채워 넣는다. 만약, 클라이언트가 접속한다면 Client 클래스의 이벤트가 발생되며 로그인과 회원가입 기능, 메시지 발송에 관련된 기능이 수행된다. Client 클래스는 ClientGroup 클래스를 통해 제어된다. 그리고 메신저 클라이언트는 회원가입, 로그인, 채팅, 쪽지 보내기 기능을 담당한다.

클라이언트는 프로그램에 접속하여 회원가입을 요청하고 승인이 나면 회원정보를 서버의 DB에 저장한다. 이후 클라이언트는 로그인을 하여 로그인에 성공하면 서버의 DB에서 회원가입 시 저장한 정보를 받는다. 그리고 통신을 위하여 다른 클라이언트의 아이디를 추가하기 위해 대화상대 처리를 요청하여 클라이언트들은 통신이 가능하게 된다. 다음 그림 1은 메신저의 구성도이다.



[그림 1] 메신저 구조

클라이언트의 DB 테이블들은 TBL_ Member에서 회원가입을 처리하고 요청하면 TBL_ Zipcode에서 우편번호를 검색하여 주소를 입력한다. 모든 입력이 완료되면 TBL_Member의 테이블에 저장한다. TBL_Connect 에서는 로그인을 담당하며 로그인이 완료되면 TBL_ Friend에서 TBL_Group의 정보를 불러와 클라이언트에게 정보를 보내준다.



[그림 2] 클라이언트 데이터베이스

(1) 회원가입

클라이언트가 회원가입을 요청하면, 서버는 SQL서버의 TBL_Member 테이블에서 회원 아이디가 있는지 검사한다. 만약 회원 아이다가 없다면, 우편번호 테이블(TBL_Zipcode)을 참조해 TBL_Member 테이블에서 회원정보를 입력하여 저장한다.

(2) 로그인

클라이언트가 로그인을 요청하면 서버는 TBL_ Member 테이블에서 아이디(user_id)와 패스워드 값을 로그인시 입력한 값과 같은지 비교한다. ID와 PW가 일치하면 TBL_Group/TBL_Friend 테이블에서 회원 ID에 해당하는 그룹번호와 친구로 등록된 목록을 가지고 온다. 친구와 그룹정보를 이용해 클라이언트에 보낼 문자열을 작성한 후 로그인이 성공한 사용자 ID와 로그인시 사용된 IP주소, 로그인 상태, 접속 시간 등을 TBL_Connect 테이블에 기록한다. 이때 서버에 로그인한 친구 정보가 있다면, 이 값을 적용해 로그인을 시도한 클라이언트 프로그램에 친구/그룹정보를 전송한다.

(3) DB에서 대화상대 처리

클라이언트가 서버에게 친구 추가를 요청하면 TBL_ Member 테이블에서 추가할 친구의 ID를 검색해서 TBL_Group/TBL_Friend 테이블에 추가하여 친구/그룹정보를 추가한다.

(4) 통신

클라이언트가 로그인에 성공하면 상대방이 채팅이나 쪽지 보내기를 요청할 수 있도록 ChatServer 클래스에서 소켓이 작동한다. 만약 상대방이 통신을 요청하면 ChatWnd 클래스가 활성화되며 상대방과의 채팅, 쪽지보내기 등이 이루어진다.

3.2 프로토콜 구현

메신저는 회원가입, 로그인, 친구 추가 및 그룹추가, 정보(난수) 교환, 이미지 추출을 통한 세션키 생성 등의 단계로 진행된다. 그리고, 공개키 암호알고리즘인 RSA, 대칭키 암호알고리즘인 ARIA, 해시함수인 MD5, SHA-1, 난수생성기인 ANSI x9.17을 이용하여 신뢰성과 안전성을 높였고, 언어는 C#으로 하였으며 세부 PC 환경은 표 1와 같다[6].

3.3 메신저 구현

기존 네이트온의 쪽지 패킷 정보에는 사용자의 ID와 E-Mail 주소, 사용자간의 데이터 패킷 등이 노출되어 송·수신 되는 것을 볼 수 있었다. 이러한 중요 정보들을 보호하고자 안전한 프로토콜을 사용하여 메신저를 구현하고자 한다. 클라이언트간의 안전한 통신을 위해 메신저 구현을 암호알고리즘인 ARIA를 적용하여 구현한다. 하이브리드 암호시스템을 이용한 메신저 설계 및 구현의 환경은 표 1과 같다.

[표 1] 환경 설정

	Messenger Server	Messenger Client
O/S	Windows Server 2003 R2	Windows Server 2003 R2
Process	intel Pentium(R) 4	intel Pentium(R) 4
CPU	3.00GHz	3.00GHz
HDD	80GB	80GB
RAM	SD 512MB	SD 512MB
.NET Framework	.net framework3.5 x86 vs2008	.net framework3.5 x86 vs2008
D/B	MS SQL 2005 Server SP1	MS SQL 2005 Server SP1

3.2.1 메신저 서버

메신저의 서버를 시작하기 위하여 다음과 같은 절차에 의해서 진행되며 구현한 결과는 그림 3과 같다.

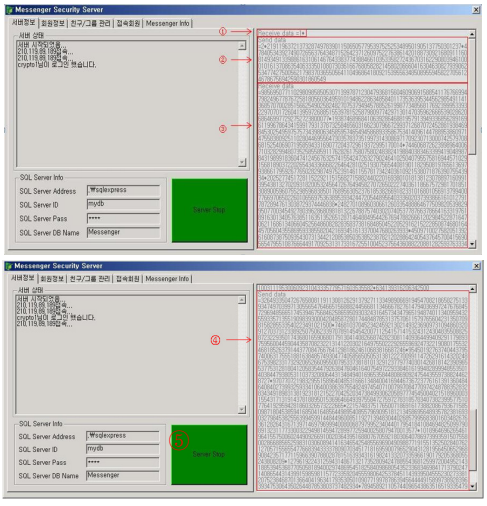
① 클라이언트로부터 서버의 공개키를 요청받았다는 메시지가 나타난다.

② 클라이언트가 요청한 서버의 공개키를 송신되었다는 메시지가 나타난다.

③ 클라이언트의 로그인 정보와 공개키 정보가 서버의 공개키로 암호화되어 수신되었다는 메시지가 나타난다.

④ 서버는 클라이언트의 공개키로 암호화된 회원정보를 클라이언트에게 송신하였다는 메시지가 나타난다.

⑤ SQL Server Info라는 박스 안에 있는 서버의 주소, IP, PW, DB명을 입력하여 서버의 시작버튼을 클릭하면 초록색으로 바뀌면서 서버가 시작이 된다.



[그림 3] 서버 메인

3.2.2 메신저 회원 가입 단계

메신저를 사용하기 위해서는 회원가입이 이루어져야 하며, 다음은 회원가입에 대한 절차이며 구현한 결과는 그림 4와 같다.

① 서버에 접속하기 위해 사용자 ID의 사용여부를 확인하여 중복되지 않도록 한다.

② 우편번호를 검색하면 사용자가 입력한 동을 기준으로 서버의 DB에 등록되어있는 주소를 받아와 사용자가 선택한 주소에 상세주소를 입력한다.

③ 입력이 완료되면 회원추가 버튼을 클릭함으로써 서버에 저장된다.

④ 사용자가 입력한 모든 정보를 지운다.

⑤ 회원가입을 취소할 때 창을 닫기 위하여 사용된다.

⑥ 회원이 자신의 신상정보를 수정할 때 사용되며, 모든 수정이 완료되면 회원정보 변경 버튼을 클릭하여 수정된 내용을 저장한다.

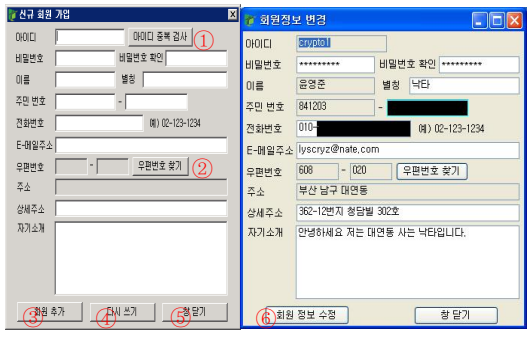


그림 4 회원가입 및 회원정보 변경

3.2.3 메신저 서버 친구 관리, 그룹 관리

친구 추가는 클라이언트간의 채팅이나 쪽지 보내기 같은 통신을 위해서 선행되어야 하는 단계로 먼저 친구의 ID를 입력하고 서버에서 검색하여 있으면 추가가 된다. 그리고 그룹추가는 친구 ID 리스트들을 조건에 맞도록 그룹화하여 사용자가 쉽게 알아보기 위한 단계이다. 구현한 결과는 그림 5와 같다.

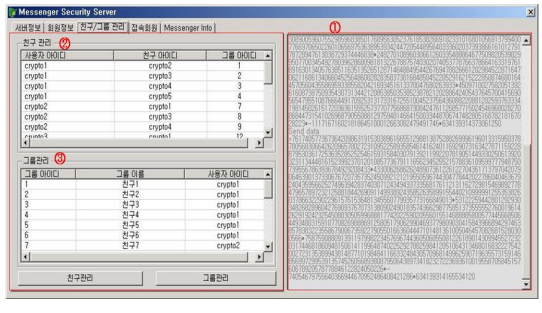
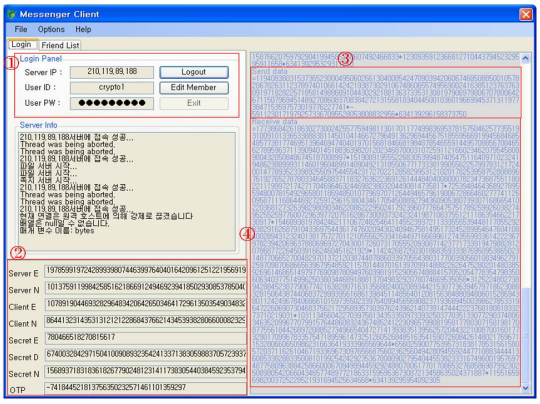


그림 5 서버 친구 / 그룹관리

① 사용자가 입력한 친구의 ID를 가지고 SQL Server에 접속하여 동일한 값을 찾는다.
② 친구의 ID와 같은 값이 있으면 친구 정보 추가를 클릭하여 친구의 ID가 추가된다.
③ 친구의 ID를 삭제하는 버튼으로 친구의 ID를 선택하고 친구 정보 제거를 클릭하면 친구의 ID가 친구 리스트에서 제거된다.

3.2.4 메신저 클라이언트

메신저를 처음 시작하기 위해서 클라이언트는 회원가입을 통하여 ID를 생성하여야 한다. 그리고 회원가입을 마치면 서버의 IP와 회원가입시 생성한 ID와 PW를 입력하여 Login버튼 ①을 클릭하여 메신저를 시작한다.

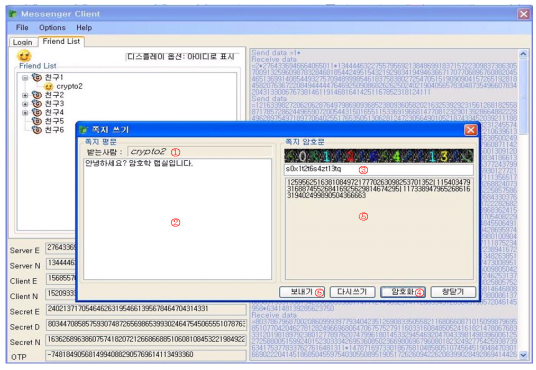


[그림 6] 클라이언트 메인

다음은 클라이언트의 메인으로 다음과 같이 진행된다.
① 서버 IP와 사용자 ID, PW를 입력하고 Login 버튼을 클릭하면 메신저가 시작된다.
② 서버의 공개키(e_S, n_S)와 클라이언트의 공개키(e_A, n_A), 개인키(d_A, n_A)이 나타나고 다른 클라이언트측과 채팅시 난수공유가 완료되면 OTP에 나타난다.
③ 클라이언트은 서버에게 자신의 IP, ID, PW를 서버의 공개키로 암호화하여 보내 자신의 인증을 요청한다.
④ 서버로부터 인증이 되면 자신의 개인키로 암호화된 자신의 개인정보를 받는다.

3.2.4 클라이언트간의 쪽지보내기와 채팅

crypto1이 crypto2에게 쪽지를 작성하고 암호화하여 전송하기위해 공유된 난수로 crypto1이 이미지를 생성한 후 그림 7의 ③처럼 난수를 입력하고 메시지를 작성하여 암호화 버튼 ④를 클릭하고 전송하면 상대방은 암호화된 메시지 ⑤를 받는다.



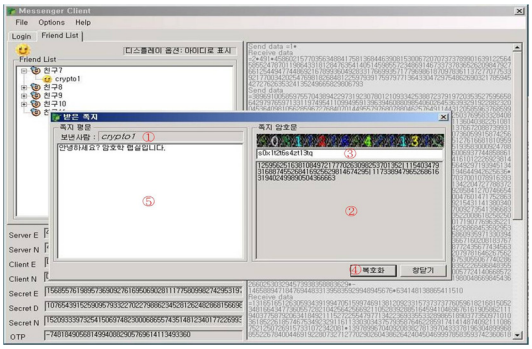
[그림 7] 클라이언트 간 쪽지 쓰기

crypto1이 crypto2에게 쪽지를 작성하고 암호화하여 전송하는 절차는 다음과 같이 진행된다.
① 쪽지를 받는 사람의 ID가 나타난다.
② 사용자가 입력한 쪽지의 내용을 확인하기 위해 암호화 되지 않은 상태로 나타난다.
③ 공유된 난수를 Expansion P-Box를 통과한 값을 Seed값으로 하여 ANSI x9.17에서 뽑아낸 난수를 이미지로 나타낸 것으로, ARIA 암호알고리즘의 키로 사용되기 때문에 난수가 입력되어야 암호화를 할 수 있다.
④ 입력된 난수의 이미지 값을 키로 ARIA 암호알고리즘을 이용하여 암호화하는 버튼이다.
⑤ 입력된 난수를 키로 ARIA 암호알고리즘을 이용하여 암호화된 쪽지를 전송하는 버튼이다.

crypto2가 crypto1로부터 받은 쪽지를 복호화하는 절차는 다음과 같이 진행된다.
① 쪽지를 보낸 사람의 ID가 나타난다.
② 입력된 난수로부터 얻은 이미지 키로 ARIA 암호알고리즘을 이용하여 암호화된 메시지를 나타낸다.
③ 공유된 난수를 Expansion P-Box를 통과한 값을 Seed값으로 ANSI x9.17에서 뽑아낸 난수를 이미지로 나타낸 것으로, ARIA 암호알고리즘의 키로 사용되기 때문에 난수가 입력되어야 복호화를 할 수 있다.
④ 입력된 난수를 키로 ARIA 암호알고리즘을 이용하여 복호화 할 때 사용되는 버튼이다.
⑤ 복호화된 메시지를 나타내기 위한 창으로 난수를

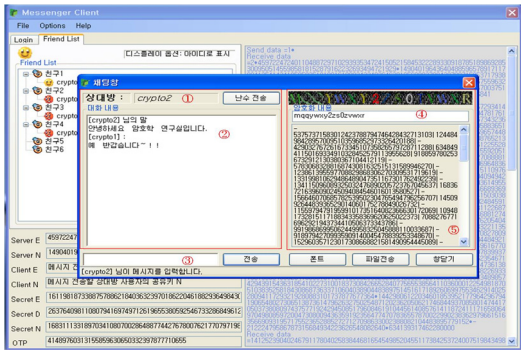
입력하여 복호화 버튼을 클릭하기 전에는 메시지의 어떠한 내용도 확인할 수 없다.

다음은 crypto2의 쪽지 받기에 대한 설명이다. 그림 8에서 난수를 이용하여 이미지를 생성한 후 ③처럼 난수를 입력하고 복호화 버튼 ④를 클릭하면 메시지 내용을 확인할 수 있다.



[그림 8] 클라이언트 간 쪽지 받기

다음은 클라이언트간의 통신을 위하여 crypto1이 채팅을 신청하면 그림 9과 같이 crypto2에게 난수가 전달된다. 난수의 공유가 완료되면 이 난수는 채팅을 신청한 crypto1이 난수 보내기 버튼 ①을 클릭함으로써 난수가 이미지로 생성되어 crypto1과 crypto2의 ④와 같이 나타난다. 그리고 클라이언트들은 나타난 이미지를 텍스트 상자에 입력한 후 메시지를 입력하여 전송하면 왼쪽에는 평문이 오른쪽에는 암호화된 메시지 데이터가 나타나게 된다.



[그림 9] 클라이언트간의 채팅

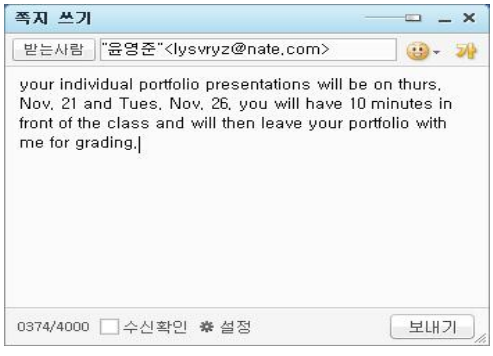
crypto1과 crypto2간의 채팅에 대한 절차는 다음과 같이 진행된다.

① 채팅을 할 때 상대방의 ID가 표시된다.
② 평문과 복호화된 메시지가 나타난다.
③ 전송하기 전의 메시지를 확인하기 위하여 사용된다.
④ 공유된 난수를 이용하여 박스에 입력하여 얻은 이미지 값을 ARIA 암호알고리즘의 키로 사용된다. 이때 난수를 입력하지 않으면 암호화된 메시지는 복호화를 할 수 없기 때문에 메시지는 나타나지 않는다.
⑤ 입력된 이미지 값으로 ARIA 암호알고리즘을 이용하여 암호화된 메시지를 나타낸다.
⑥ 폰트 버튼은 메시지내용을 표시할 때 글자모양과 글자크기를 지정할 수 있다.
⑦ 파일전송 버튼은 상대방에게 파일을 전송할 때 사용되는 버튼으로 클릭하게 되면 보낼 파일을 찾아서 메시지를 전송할 수 있다.
⑧ 창 닫기 버튼은 채팅을 종료할 때 사용된다.

4. 패킷 분석 및 비교

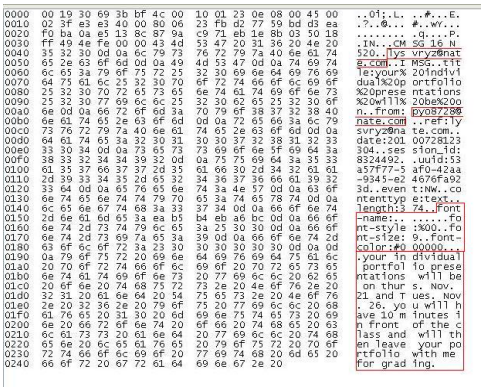
본 장에서는 기존 메신저와 제안한 메신저에 대하여 패킷을 비교 분석한다. 구현한 메신저에서는 회원가입 시 5회의 해시함수 연산과 2회의 지수연산이 필요하다. 회원가입 후 사용자가 로그인을 하게 되면 서버의 공개키를 가지게 된다. 로그인 단계에서 지수연산은 2회에 0.05 μs이다. 현대 컴퓨팅 기술에서는 연산속도가 큰 영향을 미치지 않는다.

기존의 네이트온은 그림 10과 같이 클라이언트 Crypto1이 Crypto2에게 쪽지를 보낸다고 하였을 때 그림 11과 같이 네트워크상에서 쪽지 내용이 그대로 노출되었다.



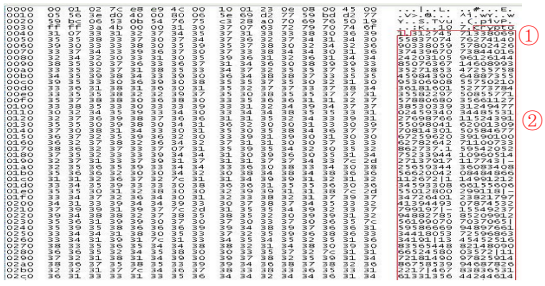
[그림 10] 쪽지 전송 테스트

기존 네이트온은 메시지 정보를 그림 10과 같이 네트워크상에서 패킷을 캡처하여 분석한 결과 그림 11과 같이 송신자의 이메일주소, 수신자의 이메일주소, 송신자의 글꼴정보, 쪽지내용 등이 그대로 노출되는 것을 확인할 수 있었다.



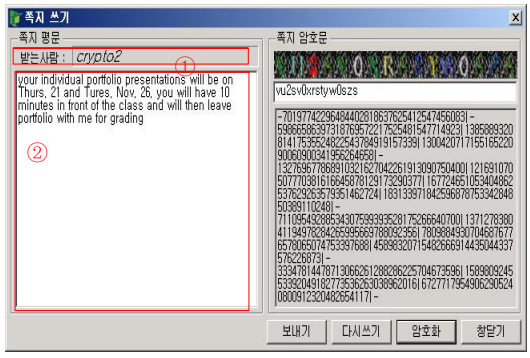
[그림 11] 네이트온 패킷

본 논문에서 제안한 메신저는 기존의 네이트온과 달리 쪽지를 전송하기 전에 메시지를 암호화하여 전송한다. 동일한 패킷 캡처 프로그램을 사용하여 네트워크상의 패킷을 캡처하여도 그림 12와 같이 자신의 ID와 암호화된 메시지만 표시되기 때문에 통신 내용을 알 수 없다.



[그림 12] 메신저 쪽지 보내기 패킷 분석

본 논문에서 제안한 메신저는 그림 13과 같이 쪽지를 보내기 전에 메시지를 암호화하여 전송한다. 클라이언트 간의 안전한 통신을 위한 메신저는 쪽지를 보내기 전에 이미지 배열에서 추출된 이미지를 128비트의 입력값을 Expansion P-Box를 통해 256비트의 확산된 출력값을 생성한다. 출력값을 세션키로 사용하고 세션키를 이용하여 암호화되어 전송되기 때문에 악의적인 목적을 가진 제3자로부터 통신내용이 안전하다.



[그림 13] 메신저 쪽지 보내기 원본

기존 메신저는 악의적인 의도를 가진 서버 관리자가 내부 DB에서 사용자의 주민등록번호, 집 주소, 전화번호 등을 모두 알 수 있었다. 하지만 본 논문에서 제안한 메신저는 최소한의 사용자 정보만을 서버에 노출시키고 중요한 개인정보는 사용자만이 알고 있는 패스워드로 암호화를 하여 전송되기 때문에 안전하다. 즉, 암호학적 해시함수의 일방향성 때문에 서버 관리자라 하여도 사용자의 패스워드나 사용자의 개인정보를 알 수 없기 때문에 내부자 공격에 안전하다[7]. 그리고 타임스탬프(T)를 이용하여 메시지의 유효성을 검증하고 있다. 만약 공격자가 사용자의 메시지를 중간에서 가로채 저장하고 있다가 재전송 할 경우, 그 메시지는 처음 단계인 타임스탬프(T)에 대한 검증을 통과 할 수 없다[8]. 또한 공격자가 다른 정보를 변경하지 않고 타임스탬프만 변경 할 경우 두 번째 단계인 무결성 검증을 통과 할 수 없다. Crypto1과 Crypto2가 통신할 경우 공격자는 Crypto1과 Crypto2의 메시지를 중간에서 가로챌 수 있다. Crypto1은 서버를 통하여 안전하게 Crypto2의 난수를 수신받기 때문에 중간자 공격에 안전하다[9]. 본 논문에서 제안한 메신저는 이전에 사용한 공개키나 세션키에 대한 정보를 저장하지 않고, 한 세션 마다 서버를 통하여 클라이언트간의 공개키를 생성하고 난수를 분배하기 때문에 안전하다.

5. 결론

기존의 네이트온은 클라이언트가 쪽지를 보낼 때 패킷을 분석하면 네트워크상에서 그대로 노출되었다. 또한 네이트온은 사용자의 개인정보를 서버의 DB에 저장을 하기 때문에 내부자 공격에 취약하고 송신자의 이메일주소, 수신자의 이메일주소, 송신자의 글꼴 정보, 쪽지내용 등이 네트워크상에 그대로 노출되는 것을 확인할 수 있었다. 이러한 심각한 정보 노출 문제점을 해결하기 위해 메

신저를 설계하고 구현하였다.

본 논문에서 제안한 메신저에서는 서버에게 클라이언트의 최소한의 개인 정보만을 공개하고 중요한 개인정보는 사용자만 알고 있는 PW로 암호화하여 서버의 DB에 저장한다. 클라이언트의 개인정보는 자신의 PW로 암호화되기 때문에 서버관리자 또는 제3자는 악의적인 의도로 사용자의 중요한 정보를 알 수 없다. 또한 네트워크상에 노출되었던 통신 내용도 암호화되기 때문에 안전하게 통신을 할 수 있다.

참고문헌

[1] Grey, M., “Love It or Hate It: Instant Messaging Invades the Enterprise”, Gartner, 2001.

[2] Beer, S., "Instant Mayhem", SMH, 2003.

[3] Grey, M. Batchelder, R., "Free Instant Messaging: Taming the Wild Beast", Gartner, 2001.

[4] 신동휘, 최윤성, 박상준, 김승주, 원동호. “네이트온 메신저의 사용자 인증 메커니즘에 대한 취약점 분석,” 정보보호학회 논문지, pp. 67-80, 2007.

[5] 전웅렬, 원동호, 김승주, “국내 상용 제품의 취약성 분석,” 정보보호학회지, 제19권 제4호, 2009.

[6] 윤영준, 표경환, 신승수, 한군희, “사용자 사이의 안전한 통신을 위한 메신저 설계”, 한국산학기술학회 춘계학술대회 논문집, 2010.

[7] Behrouz A.Forouzan. “Cryptography and Network Security”, (주)한국맥그로힐, 2008.

[8] 임종인, 이동훈. “금융분야의 안전한 암호이용에 대한 연구,” 2008.

[9] 박해룡, 전인경, 이향진, 최은영, 강연정, 이환진, 신동휘. “암호이용활성화 보고서”, 한국정보보호진흥원, 2008.

한 군 희(Kun-Hee Han) **[종신회원]**

- 2008년 8월 ~ 현재 : 백석대학교 정보통신학부 교수

<관심분야>
암호프로토콜, 네트워크 보안, 영상처리

신 승 수(Seung-Soo Shin) **[정회원]**

- 2001년 2월 : 충북대학교 수학과 (이학박사)
- 2004년 8월 : 충북대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 동명대학교 정보보호학과 교수

<관심분야>
암호프로토콜, 네트워크 보안, USN, 스마트 카드