ISSN 1738-7531

보안공학연구논문지
JSE

Journal of Security Engineering

Vol. 7, No. 1, February 2010

보안공학연구지원센터

보안공학연구논문지
Journal of Security Engineering

ISSN : 1738-7531 제7권,제1호,2010년2월

목 차

연관 기억 메카니즘을 이용한 패킷 데이터 암호화

안성빈[1], 김장한[2], 김영철[3]

Encryption of Packet Data with Associative Memory Mechanism

Sung-Bin Ahn[1], Jang-Han Kim[2], R. Young Chul Kim[3]

요 약

현재 인터넷상에서 개개인의 정보 덩어리들이 자유롭게 전달되고 있다. 그러나 정보 패킷이 제3자에 의해 패킷 스니핑이 빈번히 야기되고 있는 실정이다. 그래서 전송되는 패킷에 많은 방법으로 암호화가 되고 있다. 본 논문에서는 이 같은 문제를 간단한 방법으로 해결하기 위해 Associative memory 기법을 이용하여 구현하였다. 적용사례로 클라이언트/서버 환경 상에 암호화 기법의 적용을 위한 모델을 구현하고, 시뮬레이션을 통해서 이를 검증 하였다.

핵심어 : 연상기억장치, 패킷스니핑, 암호화, 시뮬레이션

Abstract

Today there are unrestrictedly transmitted an enormous amount of individual information on the Interne. But it was frequently happened to steal information with packet sniffing by 3rd party. So the transmitted information was made packet encryption with various methods. this paper shows one simple method for solving packet sniffing attack through implementing encryption with associative memory mechanism. As this application case, we implement the model for applying this encryption method on client/server environment, and show to verify this model through simulation

Keywords : Associative Memory, Packet Sniffing, Encryption, Simulation

1. 서론

오늘날 인터넷이 보편화됨에 따라 사람들이 정보를 공유하고 저장하는 수단으로써 필수적인 역할을 하게 되었다. 이런 중요한 정보들은 인터넷의 네트워크 내에서 패킷의 형식으로 주고받으며, 도중에 패킷 정보는 쉽게 유출이 가능하다. 패킷을 유출하는 방법에는 여러 가지가 있는데, 이전

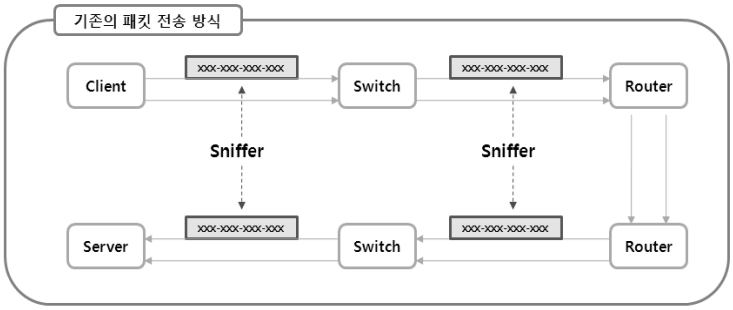
접수일(2009년06월04일), 심사의뢰일(2009년06월05일), 심사완료일(1차:2009년06월22일, 2차:2009년07월12일)
게재일(2010년02월28일)

[1]홍익대학교 일반대학원 소프트웨어공학 전공
email: ahn1@selab.hongik.ac.kr
[2]홍익대학교 과학기술대학 컴퓨터정보통신 교수.
email: jkim1@hongik.ac.kr
[3](교신저자) 홍익대학교 과학기술대학 컴퓨터정보통신 부교수.
email: bob@hongik.ac.kr

연구에서는 기존의 패킷 해킹 기법들 중, 패킷 스니핑(sniffing) 기법에 대한 문제 해결을 중점으로 연구를 진행하였다. 패킷 스니핑 기법[1]이란, 사용자가 서버와 통신하는 중에 주고받는 패킷에 대한 소통을 공격자가 관찰하다 그 패킷을 가로채는 기법이다. 그리고 이 가로챈 패킷을 분석 하여, 사용자와 서버 간에 어떤 정보를 주고 돌려받는 지, 또 어떤 방식으로 어떻게, 어디서까지 다 알아낼 수 있는 해킹 기법이다[2].

기존의 패킷 스니핑을 방지하기 위한 방법으로는 먼저 네트워크를 스니핑하는 호스트를 주기적으로 점검하는 방법이 있다. 이러한 점검을 통하여 누가 네트워크를 도청하는지 탐지하여 조치하여야 한다. 다른 방법으로는 스위칭 환경의 네트워크를 구성하여(비록 스니핑이 가능하기는 하지만) 되도록 스니핑이 어렵도록 한다. 또 다른 방법으로는 데이터를 암호화 하는 것이다. 데이터를 암호화 하게 되면 스니핑을 하더라도 내용을 볼 수 없게 되기 때문에 안전한 전송이 가능하다. 그리하여 SSL, PGP[3]등 인터넷 보안을 위한 많은 암호화 프로토콜이 존재한다[4]. 이 논문에서는 스니핑 문제를 해결할 수 있는 연상기억장치(Associative Memory)[5]를 이용한 암호화 기법을 간단한 방법으로 구현하였다. 기존의 연상기억장치를 이용한 연상기법은 대부분이 일정한 데이터를 복원하거나 선별하는데 이용되어 왔지만[6], 이 논문에서는 패킷을 고의로 손실 시켜 전송하고, 이를 연상기억장치 통해 복원하는 연상 기법을 암호화 기법에 도입하여[7] 간단한 암호화 기법을 제시한다. 적용사례로 연상기억장치를 구현하고, 이를 시뮬레이션 도구를 통해서 검증을 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 연상기억장치에 대해서 설명한다. 3장에서는 연상기억장치를 이용한 암호화 방법에 대해서 언급한다. 4장에서는 적용사례로 시뮬레이션 모델 상에서의 연상기억장치를 사용한 암호화 방법을 검증한다. 마지막 장에서는 결론 및 향후 연구를 언급한다.



[그림 1] 네트워크 전송모델과 패킷 스니핑

[fig. 1] Network transmission model and packet sniffing

2. 관련 연구

컴퓨터 구조에서의 연상기억장치 (Associative Memory, 결합기억 장치, Content Addressable Memory) 의 개념은 다음과 같다. 일반적으로 대부분의 기억 장치에서는 정보가 저장되어 있는 주소를 입력하여, 입력된 주소에 저장되어 있는 기억 장치의 내용을 접근하게 된다. 하지만 연상 기억 장치에서는 주소를 사용하는 것이 아니라, 접근하려고 하는 자료의 내용을 사용하여 이러한 자료가 저장되어 있는 기억 장치를 접근할 수 있다. 위와 같은 특성에 착안하여 Neural Network[8]를 이용한 연상기억장치는 입력 벡터가 저장된 모든 벡터와 병렬로 비교되면서 일정한 척도로 얼마나 잘 매칭 되는가가 정해지고 가장 잘 매칭 되는 저장 벡터가 선택되어 입력된 벡터를 인식할 수 있다[9]. 연상기억장치를 활용하기 위해서 수식을 전개하면 다음과 같다[8]. 아래의 식(1)에서 x는 임의의 데이터로 정한다.

$$x = \begin{cases} 1 \rightarrow \text{true} \\ 0 \rightarrow \text{Unknown} \\ -1 \rightarrow \text{false} \end{cases} \tag{1}$$

x는 세 가지의 값을 가질 수 있는데, 1은 항상 참인 값, 0은 알지 못하는 값, -1은 참이 아닌 값을 의미한다.

$$\text{X}= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad X^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$$

변형(transformation) 행렬

(2)

식(2)에서 X는 x1, x2, x3라는 벡터 값을 가지고 있으며, X를 변형한 변형 행렬 X^T도 x1, x2, x3라는 벡터를 가지고 있다.

위의 식(2)에서의 X와 변형 행렬 X^T, 두 행렬을 이용하여 아래의 수식(3)과 같이 전개를 한다.

$$X \cdot X^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} x_1x_1 & x_1x_2 & x_1x_3 \\ x_2x_1 & x_2x_2 & x_2x_3 \\ x_3x_1 & x_3x_2 & x_3x_3 \end{bmatrix} \tag{3}$$

식(3)과 같이 X와 변형 행렬 X^T의 각각의 벡터들을 곱하여 나온 행렬 데이터를 X_1이라고 한다.

위에서 전개한 식(1), (2), (3)과 같은 방법으로 식을 전개하여 각각의 데이터 X_n을 만들고, 만들어진 데이터들을 아래의 식(4)와 같이 표현한다.

$$\{X_1, \quad X_2, \quad X_3, \quad \cdots \quad X_n\}=\mathrm{M} \tag{4}$$

위의 식(4)에서 M은 메모리 내부 상태를 의미한다. 또한 메모리 M은 아래의 식(5)와 같이 전개한다.

$$M = \sum_k X_k X_k^T \tag{5}$$

위의 식(5)와 같이 M은 X와 변형 행렬 X^T들의 곱을 합한 데이터 값으로 정의 되는 것이다.

위와 같은 방법으로 M이라는 연상기억장치를 만들고 만들어진 메모리 M을 이용하여 데이터를 손실 시키고 복구 가능하다. 메모리 M 복구를 위해 아래의 식(6)의 양자화 함수를 이용해야 한다.

$$\phi(x) = \begin{cases} -1 \\ 0 \\ 1 \end{cases} \quad \begin{matrix} x<0 \\ x=0 \\ x>0 \end{matrix} \tag{6}$$

양자화 함수 ϕ 는 X안에 있는 각각의 벡터 값인 x1, x2, x3를 위의 수식과 같이 1또는 0또는 -1로 바꾸어준다. 위의 양자화 함수를 각각의 데이터 X와 메모리 M에 대해서 아래의 수식(7)과 같이 적용을 한다.

$$\phi(X) = [\phi(x_i)]$$
$$\phi(M) = [\phi(x_{ij})] \tag{7}$$

수식(7)을 적용 후에, 손실된 데이터가 y 라고 주어졌을 때, 아래의 수식(8)과 같이 적용하면

$$\therefore Z = \phi(\phi(M)y) \tag{8}$$

손실된 데이터 y 로부터 복원 시킨 데이터 Z 를 복구 할 수 있다.

3. 연상기억장치를 이용한 암호화 기법

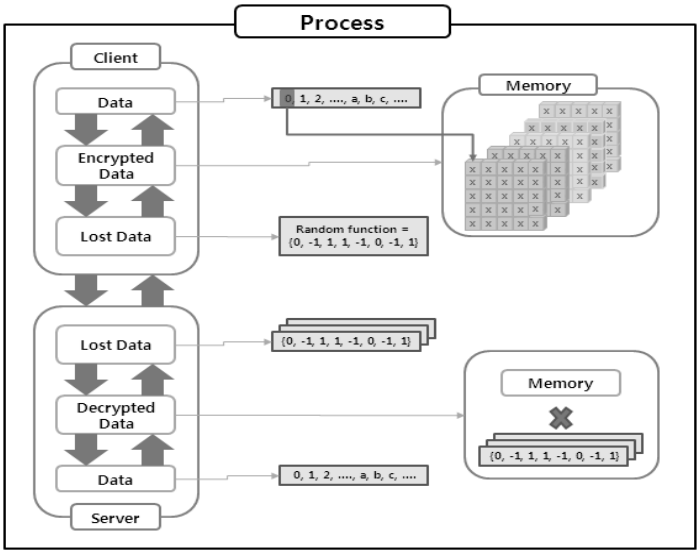
3.1 연상기억장치 암호화

연상기억장치를 이용한 암호화 기법은 사람이 사물에 대한 완전한 기억이 없이 그 사물에 대한 일부의 정보만으로도 어떠한 물체인지 또는 존재인지에 대한 기억을 떠올릴 수 있는 점을 이용한다. 사용자가 서버에게 전달하는 패킷에 손실된 일부 정보만을 전송함으로 써, 공격자가 패킷을 가로채도 가로챈 패킷 정보를 분석할 수 없게 한다. 즉, 사용자의 대한 정보를 유추할 수 없는 불완전한 정보이기 때문에 스니핑 기법을 방지 할 수 있는 데이터 암호화 기법이다.

3.2 연상기억장치 암호화 프로세스

연상기억장치를 이용한 암호화 과정에 대한 프로세스를 기술하면, 먼저 사용자는 일반적으로 서버에 사용자에 대한 정보를 패킷으로 전송한다. 하지만 이 패킷정보가 바로 서버에 전송되는 것이 아니라, 사용자 내에 위치해 있는 연상기억장치에 전송하고, 이 때 연상기억장치는 사용자의 패킷에 대한 암호화를 시키며, 암호화 시킨 정보를 Random Function을 이용하여 랜덤하게 고의적으로 패킷을 손실 시킨다. 그리고 이 패킷 정보를 서버에 전송을 하게 되고, 이 패킷 정보를 서버 내에 위치한 연상기억장치에 전송하여, 손실된 패킷 정보를 받은 연상기억장치는 손실된 패킷 정보를 복구한다. 복구한 패킷 정보를 다시 복호화 과정을 거친 후에 서버에 완전한 패킷 정보로 전송해 주게 되는 프로세스이다.

이러한 과정 중간에 공격자가 패킷 스니핑에 성공하더라도 연상기억장치가 없다면 손실된 암호화 패킷 정보를 복호화 하는 것이 불가능하다.

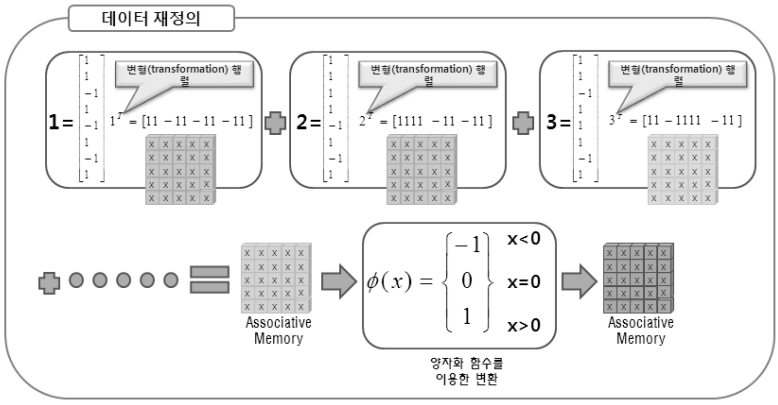


[그림 2] 연상기억장치의 암호화 프로세스

[fig. 2] The encryption process of associative memory

그림 2는 사용자와 서버간의 패킷 전송을 연상기억장치를 적용을 통해서 암호화와 복호화 하는 프로세스를 간단히 도식화한 모델이다. Sender는 기존의 데이터를 패킷을 이용하여 전송을 시도하며, 전송되는 패킷을 연상기억장치가 암호화 패킷화하고, 만들어진 암호화 패킷을 랜덤하게 고의적으로 손실시킨다. 손실 된 패킷을 전송한다. Receiver는 패킷을 받은 후에, 연상기억장치를 이용하여 손실한 패킷을 원래의 패킷으로 복구하고, 원래의 데이터로 복호화 함으로써 안전한 데이터 전송이 가능하다.

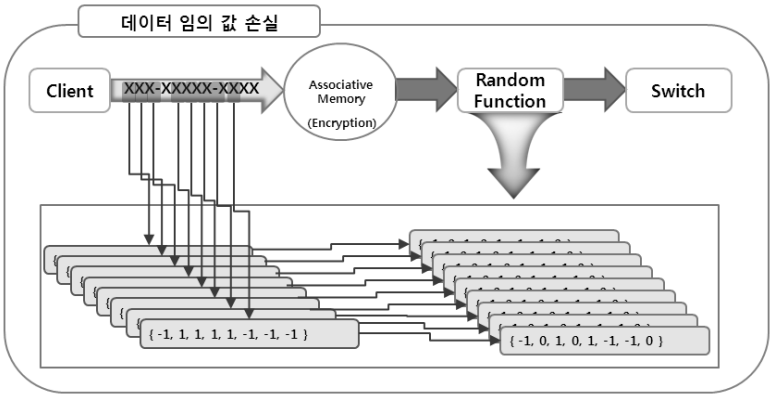
3.3 연상기억장치 암호화 기법



[그림 3] 연상기억장치의 암호화 프로세스

[fig. 3] The encryption process of associative memory

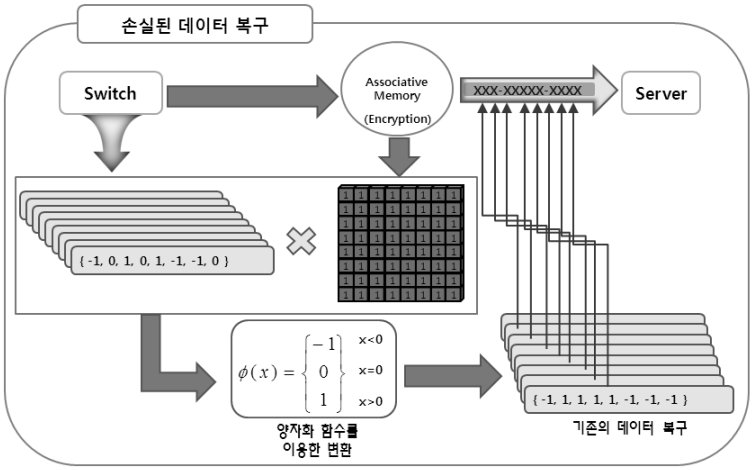
연상기억장치를 이용한 데이터 암호화는 먼저 전송 전 유저 내에서 데이터를 정의 해야 한다. 위의 그림 3은 데이터의 재정의 과정을 도식화한 그림이다. 위의 그림3과 같이, 만약 1이라는 데이터를 위와 같은 행렬로 재정의 하고, 재 정의된 행렬을 변형행렬로 다시 변환한다. 그리고 재 정의한 행렬과 변형행렬을 곱하여 나온 데이터를 하나의 메모리라고 하고, 이런 방식으로 모든 데이터를 재 정의하여, 재 정의한 모든 데이터를 합하여 연상기억장치를 만들 수 있다. 하지만 이렇게 만든 기억장치는 복잡하기 때문에 사용하기 어렵다. 그래서 양자화 함수를 이용하여 간단한 연상기억장치로 변환하고, 다시 완전한 연상기억장치가 되는 것이다. 이 복잡한 과정을 사용자가 하자면 매우 복잡하기 때문에 도구를 통해서 자동적으로 데이터 재정의 부터 메모리 생성까지 도와준다.



[그림 4] 패킷 암호화를 통한 전송

[fig. 4] Encrypted packets sent through

사용자는 그림 4와 같이 패킷 정보를 연상기억장치에 보내면, 연상기억장치는 각각의 패킷 정보를 재정의 과정을 통해서 암호화를 하고, 암호화한 정보를 연상기억장치 내에 Random Function을 이용하여 임의의 손실 값으로 변환하여 패킷 정보를 스위치에 전송하게 된다.



[그림 5] 전송받은 패킷의 복구와 복호화

[fig. 5] Recovery of transmission and decoding received packets

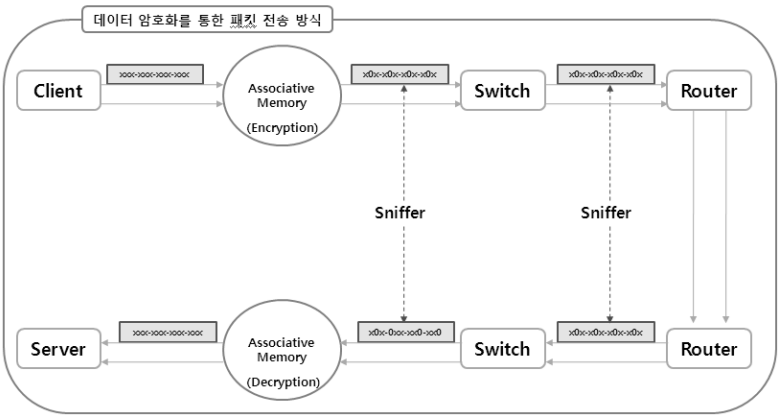
서버 쪽의 스위치는 그림 5와 전송 받은 손실된 정보를 연상기억장치에 전송하고, 전송받은 정보와 연상기억장치 내에 메모리를 곱하고, 다시 양자화 함수를 이용하여 변환하여, 기존의 정보로 복구를 할 수 있게 된다. 그리고 복구한 정보를 다시 복호화 하여 서버에 전송한다.

결과적으로 손실 시킨 코드를 전송하고, Receiver는 코드를 받은 후에, 연상기억장치를 이용하여 손실한 코드를 원래의 코드로 복구 시키고, 원래의 데이터로 복호화 하는 과정으로 데이터 암호화하고, 이로써 안전한 데이터 전송이 가능하다.

4. 적용 사례

적용사례로는 클라이언트/서버 환경 내에서 연상기억장치를 적용을 통해 각각의 패킷을 암호화하고 다시 복호화 하는 과정을 구현과 시뮬레이션을 통해 검증 하였다.

4.1 적용 환경



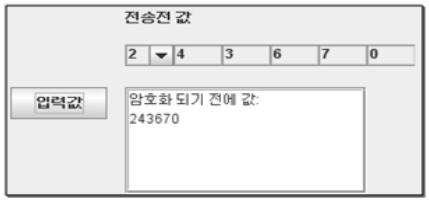
[그림 6] 연상기억장치의 네트워크 전송과정 모델링

[fig. 6] Network transmission process modeling of associative memory

그림 6은 본 논문에서 설정한 네트워크 환경이다. 사용자는 단일 사용자로 한정하고, 사용자가 패킷 전송 전에 연상기억장치는 패킷의 값을 임의로 손실하여, 손실 시킨 값을 전송, 서버 쪽의 연상기억장치가 다시 복구하고, 서버가 받아들이는 방식으로 모델 만들었다.

4.2 구현

위의 환경을 기반으로 패킷 전송 모델을 구현을 하였다. 여기서 전송되는 정보는 사용자가 입력한 패스워드라고 가정을 한다.



[그림 7] 전송 전 값

[fig. 7] Value Transfer ago

사용자가 패스워드에 대한 정보에 대해서 위의 그림 7과 같이 입력을 한다. 사용자가 입력한 정보에 대해서 위의 전개한 수식 기반으로 암호화를 한다.

[그림 8] 암호화 시킨 데이터

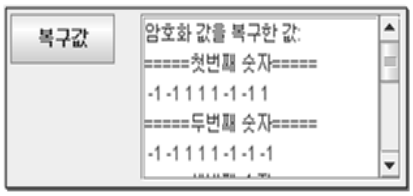
[fig. 8] Encrypted data

그림 8은 암호화 시킨 데이터를 나타낸 그림이다.

[그림 9] 연상기억장치를 이용하여 손실 시킨 데이터

[fig. 9] Missing data using of associative memory

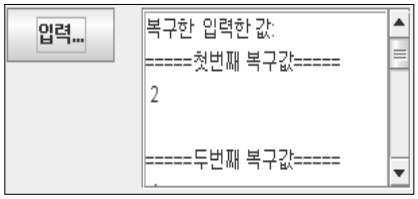
이 암호화한 데이터를 그림 9와 같이 연상기억장치에서 랜덤하게 손실데이터를 생성한다.



[그림 10] 연상기억장치를 이용하여 손실시킨 데이터 복구

[fig. 10] Recovery missing data using of associative memory

손실시킨 데이터를 전송하고, 전송받은 손실데이터를 다시 그림 10과 같이 연상기억장치를 이용하여 복구 한다.

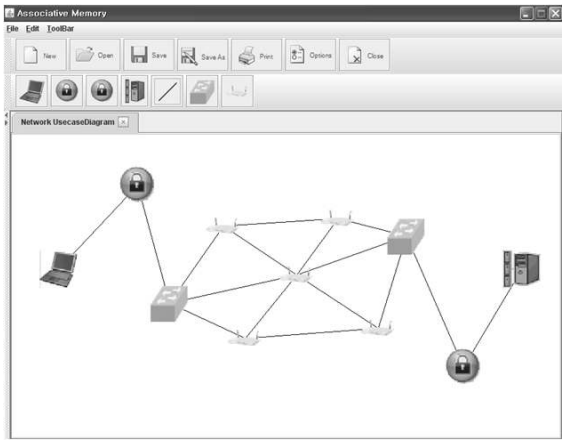


[그림 11] 복구한 데이터를 복호화

[fig. 11] Decrypt the recovered data

그리고 복구한 데이터를 그림 11과 같이 원래 사용자가 입력한 데이터로 복호화 한다.

4.3 시뮬레이션

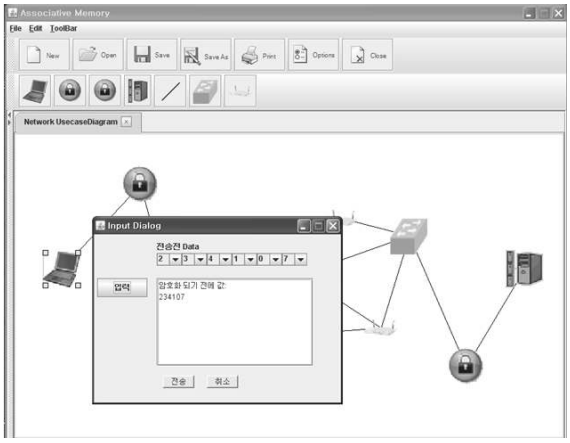


[그림 12] 도구를 통한 시뮬레이션 환경 구축

[fig. 12] Build a simulation environment through tools

먼저 시뮬레이션 실행 해보기 위해서 간단한 네트워크 환경을 구성하였다. 도구를 실행하여, 도구에 구성되어 있는 아이콘을 드래그 하여, 사용자, 스위치, 라우터, 연상기억장치, 서버로 간단한 네트워크 환경을 구성한다.

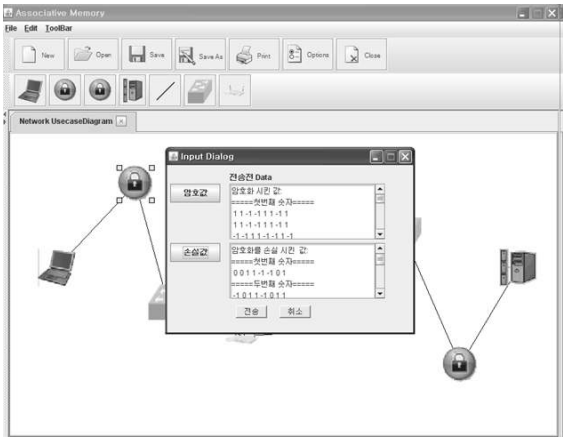
그림 12는 도구를 사용하여 만든 간단한 네트워크 시뮬레이션 환경이다.



[그림 13] 전송 전 데이터 입력

[fig. 13] Transport ago Data Entry

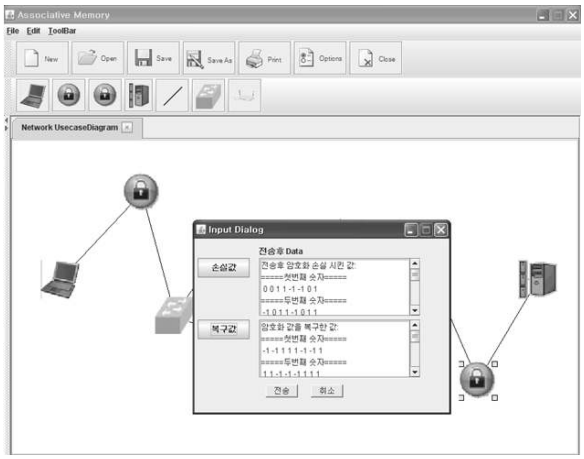
그림 13과 같이 시뮬레이션 상에서 사용자는 간단한 방식으로 데이터를 입력하고 연상기억장치에 데이터 전송을 실행한다.



[그림 14] 연상기억장치를 이용하여 손실시킨 데이터

[fig. 14] Missing data using of associative memory

그림 14와 같이 사용자가 입력한 데이터를 받은 연상기억장치는 암호화한 데이터와 손실 시킨 데이터를 생성하고, 시뮬레이션 상에서 확인가능 하도록 보여주며, 그리고 스위치로 데이터를 전송한다.



[그림 15] 연상기억장치를 이용하여 손실시킨 데이터 복구

[fig. 15] Recovery of missing data using of associative memory

그림 15와 같이 서버에 위치한 연상기억장치는 스위치로부터 손실한 데이터를 전송받고, 손실한 데이터를 확인하여, 다시 기존의 암호화 데이터로 복구를 하고, 복구한 데이터를 서버에 전송한다.

서버에서 암호화한 데이터를 확인하여, 원래의 완전한 데이터 값으로 복구된 것을 시뮬레이션 상에서 확인함으로 써 연상기억장치를 이용한 데이터 암호화에 대한 전체과정을 검증 할 수 있는 것이다.

5. 결론

인터넷 프로토콜은 제3자에 의해 패킷 스니핑이 가능하다. 그래서 전송되는 패킷에 암호가 필요하다. 본 논문에서는 간단한 방법으로 스니핑 문제를 해결할 수 있는 연상기억장치를 이용하여 간결한 암호화 기법을 구현하였다. 또한 연상기억장치의 암호화 기법에 대해서 나타낸 시뮬레이션 도구 개발을 통하여 암호화 기법에 대한 검증하였다. 이로써 공격자의 패킷 스니핑에 대한 암호화 기법 제공과 도구를 사용한 손쉬운 암호화 방법을 검증이 가능하다.

향후 연구 계획으로 모델의 확장으로, 연상기억장치를 간단한 데이터의 입력의 암호화만이 아닌 다양한 데이터 입력에 대해서 암호화를 제공하며, 쉽게 사용할 수 있는 도구를 개발 할 것이다. 또한, 패킷 스니핑만이 아닌 다양한 공격자의 공격 패턴에 대한 암호화 방법기반의 자동 도구 연구 중이다.

감사의 글

본 연구는 2007학년도 홍익대학교 학술연구진흥비와 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업(2009년)으로 수행된 연구결과임.

참고문헌

[1] 최형기 "IPv6 그리고 SCTP 에서의 보안 문제점들", KNOM Review, 2005

[2] 박대우, 윤석현 "VoIP 서비스의 도청 공격과 보안에 관한 연구", 한국컴퓨터정보학회 , 2006

[3] 이준희 "역할 기반 접근 제어에 대한 연구 :웹에서의 RBAC", 홍익대학교 석사학위논문 , 2002

[4] 한승조 "안전성 서비스를 위한 암호화 프로토콜 설계에 관한 연구", 朝鮮大學校生産技術硏究所, 1991

[5] 신영숙 "퍼지 인지 맵과 퍼지 연상 메모리를 이용한 오인진단 모델", 한국인지과학회 , 2002

[6] 도양희, 김정우, 배장근 "홀로그래픽 연상 메모리를 이용한 한글 문자 인식", 한국통신학회 , 1993

[7] 안성빈, "기본 뉴럴 네트워크를 이용한 데이터 암호화 연구", 한국인터넷방송통신TV학회, 2009

[8] 김대제, "Neural Network을 이용한 GPS/INS 강결합 항법시스템 구현", 건국대 대학원 석사학위논문, 2004

[9] 이문기, 노육현, 손승임 "Associative Memory를 이용한 패턴 인식 설계", 정보통신 연구 진흥원, 1990. 6

저자 소개

안성빈 (Sung-bin Ahn)

2008년 홍익대학교 컴퓨터정보통신 (학사)
2009년~ 현재 홍익대학교 일반대학원 석사과정
관심분야 : 소프트웨어공학, AI, RBT, Associative memory

김장한 (Jang-Han Kim)

1993년 홍익대학교 전자공학과(공학박사)
1979년 ~ 1994년 홍익대학교 부교수
1994년 ~ 현재 홍익대학교 컴퓨터정보통신 교수
관심분야 : 통신 시스템 및 보안

김영철 (R. Young-Chul Kim)

2000년 : Illinois Institute of Technology(공학박사)
2000년 ~ 2001 : LG 산전 중앙연구소 Embedded system 부장
2001년 ~ 현재 : 홍익대학교 컴퓨터정보통신 교수
관심분야 : 소프트웨어공학, CBD, RBT