제 28 회 한국정보처리학회 추계학술발표대회 논문집 제 14 권 제 2 호 (2007. 11)

인터넷 컨텐트 애플리케이션의 실시간 통신에 효율적인 암호화 기법 연구

홍성민, 장주욱
서강대학교 전자공학과
e-mail : kingduke@eeca1.sogang.ac.kr, jjang@sogang.ac.kr

A Study on Effective Encryption Method in Real-time Communication of Internet Content Applications

Sung-Min Hong, Ju-Wook Jang
Dept. of Electronic Engineering, Sogang University

요 약

인터넷에서 발생하고 주고 받는 통신 데이터 패킷의 양 중 다양한 컨텐트 애플리케이션 사이의 암호화된 통신이 차지하는 비중은 날로 증가하고 있다. 하지만, 인터넷의 기술적 특성과 한계로 인하여 아직은 완벽한 통신 패킷의 보안 시스템을 만든다는 것이 어려운 상황이며, 특히 데이터양이 많고 실시간 전송이 중요한 컨텐트 애플리케이션들에서는 모든 데이터 패킷을 암호화하고 복호화하는 과정을 거치는 것이 개발상의 번거로움과 시스템의 부하를 가중시키는 일로 인식되고 있다. 본 논문에서는 실시간으로 연속적이며 다양한 형태의 데이터 패킷을 주고 받는 인터넷 컨텐트 애플리케이션을 위하여 개발이 간단하면서도 효율적인 암호화 기법에 대해 제안한다.

1. 서론

인터넷의 기술적 특성상 개인 신상 정보나 각종 보안이 필요한 정보들도 노출된 상태로 인터넷 사이에서 무방비로 돌아 다니고 있다는 것은 이제 전혀 새로운 사실이 아니다. 다행히 많은 보안 관련 연구들로 인하여 인터넷의 보안 성능이 획기적으로 발전하고 있지만 현재 대부분의 암호화 기법들은 단일 패킷 자체의 암호화 방식에 초점을 맞추고 있다.

단일 패킷 자체의 암호화는 암호화를 위한 노력도 많이 필요하지만 중간에서 패킷을 가로챌 경우 암호화 이전의 원문을 알아내기 쉽다는 점 때문에 항상 악의적인 공격의 대상이 되어 왔으며, 암호화를 무용지물로 만드는 다양한 알고리즘 개발이 손쉬울 수 있다는 점에서 그 한계를 가지고 있다.

본 논문에서는, 실시간으로 연속적인 데이터 패킷을 주고 받는 대부분의 인터넷 컨텐트 애플리케이션에 특화하여, 하나 하나의 패킷만을 암호화의 대상으로 볼 것이 아니라 전체 통신 중에 있는 데이터 패킷 전반에 걸쳐 암호화를 하고 간단하지만 다양하면서도 연속적으로 변화할 수 있는 암호화 기법을 적용하여 단일 패킷 암호화가 가지는 한계를 극복할 수 있는 기법을 제안한다.

"본 논문은 서울시가 시행하고 서울시립대학교 지능형 도시 사업단이 주관하는 서울시 산학연 협력 사업에서 지원을 받았습니다."

2 절에서는 기존 암호화 기법들의 장단점을 비교하고 어떠한 문제점들이 있는지 확인한다. 3 절에서는 제안하는 기법의 개요와 구성도 및 흐름도를 설명하고, 4 절에서는 제안하는 시스템의 실제 알고리즘 구현 내용을 제시한다. 5 절에서는 결론을 맺으며 향후 연구 과제를 제시한다.

2. 기존 암호화 기법들의 장단점 비교와 문제점 분석

현재 인터넷 애플리케이션의 통신을 위해 주로 사용되는 통신 암호화 기법에는 SSL(TLS), SHTTP, S/MIME, SET, RTP 등이 있다.

이러한 암호화 기법들은 대부분 단일한 패킷 자체의 암호화나 외부 보안 관련 추가 애플리케이션의 도움을 받아 인터넷 통신을 안전하게 하는 방식을 사용한다. 특히, 웹 서비스의 경우 공인인증서 등 외부의 도움이 절실히 필요하다. 더욱 큰 문제는 인터넷의 특성상 어떠한 통신 패킷도 손쉽게 가로챌 수 있고, 가로챈 데이터를 복호화하지 않아도 암호화된 동일한 데이터를 재 전송할 경우 원하는 동작을 재실행 시킬 수도 있는 가능성이 있어 일반적인 인터넷 컨텐트 애플리케이션 개발자들과 서비스 제공 업체들에게 있어서는 안정적인 암호화 및 보안 시스템의 개발을 위해 별도의 노력을 들여야 한다는 불편함이 있다[1].

	정의 및 특징	장점	단점
SSL	PKI 방식으로 공개	가장 널리	일부 복호화 방

(TLS)	키 암호화와 개인키 복호화를 사용한다. 현재 대부분의 Web 관련 암호 통신에서 사용된다.	사용되며 프로토콜 독립적으로 시스템 적용이 가능하다.	법이 개발되었고, 내용을 몰라도 동일한 패킷을 작성해서 전송이 가능.
SHTTP	HTTP 에 보안 기능을 포함 한 것으로 전송하는 웹 문서의 암호화 및 전자 서명을 지원한다.	HTTP 코드의 암호화로 웹 사이트 전체의 보안성이 높다.	Http 용으로 다른 애플리케이션 적용에 한계
S/MIME	RSA 암호화 시스템을 사용하여 전자우편을 안전하게 보내는 방법	전자 메일의 암호화를 위해 특화 되어 있다.	Email 전송을 위한 암호화 기법. 디지털 인증서를 포함한다.
SET	전자상거래의 안전한 지불을 위해 이중서명이라는(Dual Signature)방법으로 제정된 지불시스템에 대한 기술표준.	S/W 뿐 아니라 H/W 구현에 대한 표준까지 포괄하고 있다.	아직은 복잡하고 애플리케이션 수준에서 독립적으로 구현하기 어렵다.

<표 1> 기존 암호화 기법 분석

이 중 현재 SSL(TLS)이 가장 널리 사용되며, 가장 많은 인터넷 애플리케이션에 적용되고 있다. SSL(TLS)이 널리 통용되는 이유는 Session Layer 에 위치하고 있어서 어떠한 애플리케이션 프로토콜과도 연결될 수 있고, 하부의 TCP/IP 등과 자유롭게 소통할 수 있다는 점에서 큰 매력이 있을 것이다. 또한 암호화 후 복호화하는 비용이 많이 들어서 원문을 알아 내기 쉽지 않다는 점에서도 매력적이다. 하지만 암호화 및 복호화에 많은 노력이 들어가고, 짧은 암호화를 할 경우 해독할 수 있는 방법이 이미 개발되어 있어서 완벽한 안정성을 보장 받을 수 없다는 단점이 있다. 또한, 필요에 따라서 공인 인증서 등 외부의 보안 관련 프로그램과 연계되어 작동하여야만 하는 경우들이 발생하게 되어 시스템 개발에 부담을 줄 수 있다는 단점도 있다.

SHTTP 등의 프로토콜은 웹 상에서의 암호화 기법[2]으로 널리 사용되지는 않지만 괜찮은 성능을 보이는 것으로 평가되고 있다.

3. 제안 하는 암호화 기법

3.1 제안 하는 암호화 기법 : CMP(Continuously Mutating Protocol)

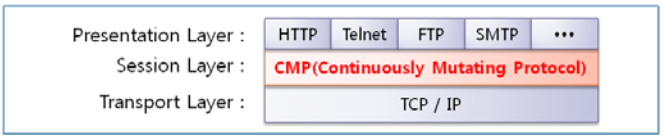
본 논문에서 제안하는 CMP 는 패킷의 인코딩 및 암호화 방식을 수시로 바꾸어 주어 전송을 하는 대상 패킷의 형태가 계속적으로 변화하는 프로토콜이다. 이것은 암호화 방식이 간단하여도 계속 변경 시킬 경우 패킷을 가로챈 패킷의 원문을 알아 내기 어려울 뿐만 아니라 동일한 데이터를 재 전송하는 것으로는 원하는 작동을 하게 할 수 없다는 원리를 이용한 것이다.

우선, 전송측 호스트에서는 자신이 랜덤 생성한 암호화 테이블을 PGP 방식으로 수신측 호스트에게 전달한다. 암호화 테이블 안에는 매 패킷마다 암호화를 하려는 방식과 관련 정보들을 포함하고 있다. 데이터 전송이 시작되면, 송신측은 이 암호화 테이블의 암호화 방식과 정보들에 맞춰 각각의 패킷을 암호화한다. 이 때 사용될 수 있는 암호화 기법은 패킷 분리 전송, 비트 및 바이트 연산, 비트 및 바이트 위치 변경, 더미 패킷과 더미 바이트의 삽입 등의 간단한 방법에서부터 RSA, DEC 등 수준 높은 암호화 기법까지 모두 자유롭게 가능하도록 한다. 물론, 대부분의 경우 간단한 암호화 방법만 사용하여도 랜덤 프로시저에 의해 계속적으로 변화하는 암호화 방법을 예측하는 것은 불가능하며, 혹시 복호화에 성공하는 패킷이 있어도 동일한 원문이 동일한 암호화를 사용하지 않으므로 중간에서의 재전송은 불가능하다고 볼 수 있다.[3][4]

여기서 중요한 점은 이러한 방식을 사용하여도 컴퓨터 메모리 내에서 직접 해킹하거나 바이너리 코드를 역산하여 랜덤 프로시저의 인위적 조작 또는 암호화 테이블의 생성을 일편적으로 생성되도록 조작할 수 있다는 점이다. 하지만, 이 부분은 통신 보안과는 다른 컴퓨터 자체의 보안에 대한 주제가 되므로 여기서는 고려하지 않도록 한다.

3.2 제안 하는 기법의 위치 및 흐름도

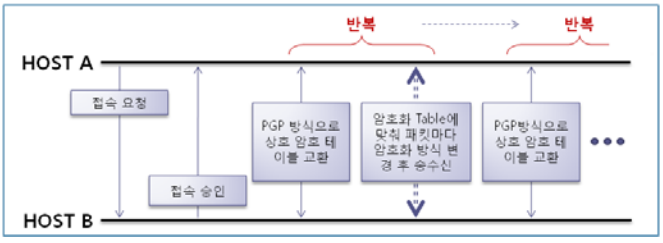
제안 하는 기법의 시스템 위치는 (그림 1)과 같다 과 같다.



(그림 1) 시스템 위치

CMP 는 Session Layer 에 위치하여 기존의 소켓 통신을 그대로 사용할 수 있도록 한다. 이것은 기존의 소켓 통신을 하는 어떠한 종류의 애플리케이션에서도 사용할 수 있도록 하여 호환성을 높여주게 된다. 또한, 하부의 Transport Layer 는 TCP/IP 를 기본적으로 사용하도록 하여 현재 존재하는 어떠한 종류의 인터넷 통신망에서도 사용할 수 있도록 한다. 이 Session Layer 에 위치한 다른 암호화 기법으로는 대표적으로 SSL[5], TLS[6] 등이 있다.

제안 시스템의 개괄적인 시스템 흐름도는 (그림 2)와 같다.



(그림 2) 시스템 흐름도

4. CMP 구현 알고리즘 및 순서도

CMP 의 핵심 구현 알고리즘은 (1)암호화 테이블 생성, (2)송신 패킷 암호화, (3)수신 패킷 복호화의 3 가지

로 나뉜다.

CMP 는 암호화 효율을 높여주기 위하여 각 패킷을 암호화 하는 방법 테이블의 정보를 외부에서 정의할 수 있도록 하였다. 암호화 방법 자체는 라이브러리 코드 속에 정의 되어 있지만 각 방법을 연결 시켜 주는 상수들을 외부에서 재정의 할 수 있도록 하여 혹시나 암호화 테이블을 열어볼 수 있어도 암호화 방법을 직접 알 수는 없도록 한 것이다.

다음은 외부에서 재정의하는 암호화 테이블 상수 파일의 예시이다.

# The Example of Pre-Defined Values for CMP.	
BIT_SHIFT_LEFT_1	0x0001
BIT_SHIFT_RIGHT_1	0x0004
BIT_REVERSE	0x0009
1_BYTE_SWITCH	0x040C
BIT_XOR_1	0x1339
INSERT_DUMMY_PACKET_1	0x70F3
INSERT_DUMMY_BYTE_1	0xA20F
SEPARATE_PACKET_2	0xE700

<표 2> 암호화 테이블 상수 파일의 예시

이 재정의 파일은 통신하는 양측 애플리케이션에서 동일하게 가지고 있어야 하며, 필요에 따라서 서로 통신 시작과 동시에 상대방의 정의 내용을 요청하고 받을 수 있다. 이 재정의 내용을 주고 받는 방법도 PGP 알고리즘을 통하여 안전하게 전송될 수 있도록 한다.

4.1 암호화 테이블 생성 알고리즘

암호화 테이블을 생성하기 위해서는 다음의 상수가 사전 정의 되어 있어야 한다.

- MAX_TABLE_ROW

```
Row count = one random value less than MAX_TABLE_ROW;
Table = Create Table Header(Row count);
For ( i, from 0 until Row count, i++ )
{
   Row buffer = Create memory for one row(row size);

   Random key = random();
   Encryption method = Get encryption method from pre-defined table(Random key);
   Insert encryption method to row(Row buffer, Encryption Method);
   Insert dummy method to row(random());
}
```

<표 3> 암호화 테이블 생성 알고리즘

4.2 송신 패킷 암호화 알고리즘

```
If ( Row number > Row count ) then
        Row count = Create new table();
        Send table to destination host();
        Row number = 1;
End if

Before encryption = Get one packet from buffer();
Call function pointer of encryption method(After encryption,
        Before encryption, Row number)
Add to send buffer(After encryption)
```

<표 4> 송신 패킷 암호화 알고리즘

4.3 수신 패킷 복호화 알고리즘

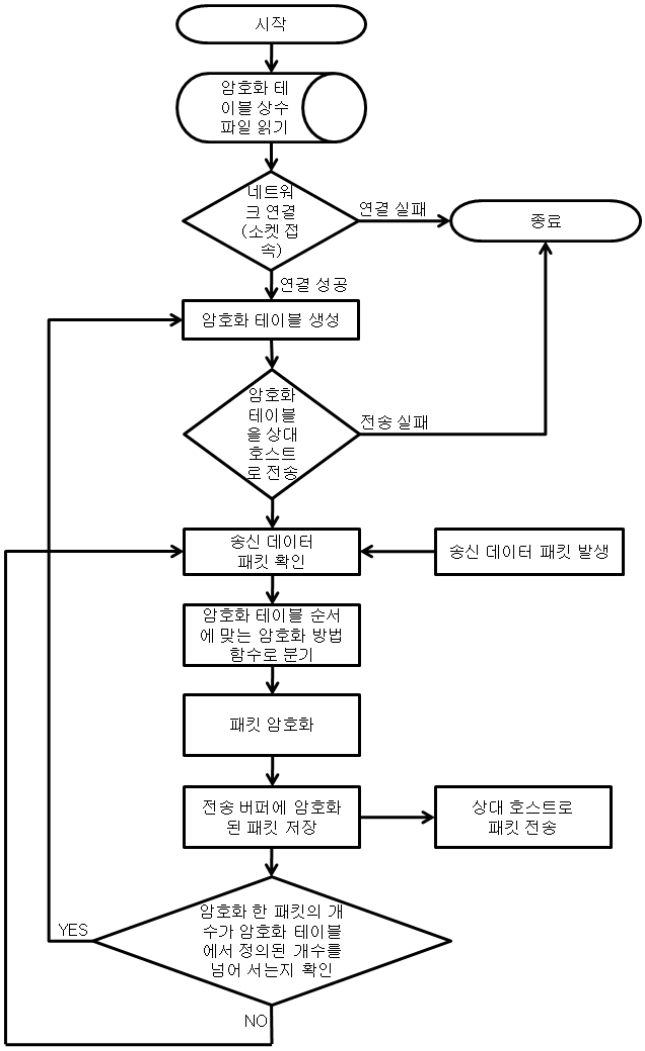
```
If ( Row number > Row count) then
        Request new table of destination host();
        Row count = Receive new table from destination host();
        Row number = 1;
End if

Before decryption = Get one packet from receive buffer();
If ( more than one packet has not completely received in receive buffer ) then
        Return and wait for next packet();
End if

Call function pointer of decryption method(After decryption, Before decryption, Row number)
Add to buffer(After decryption);
```

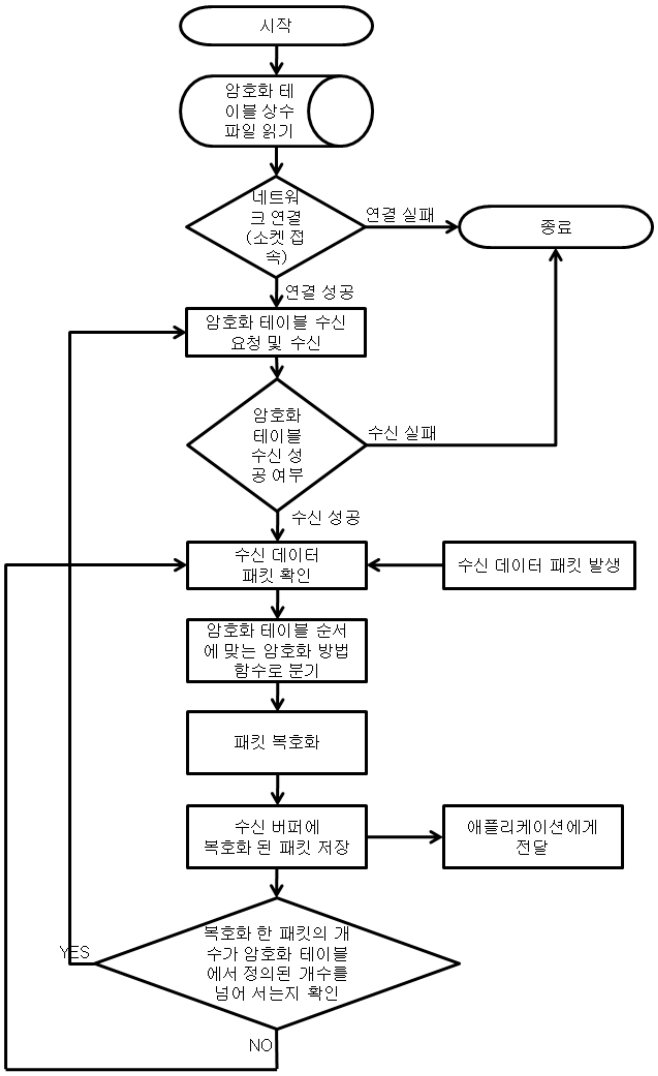
<표 5> 수신 패킷 복호화 알고리즘

4.4 송신 알고리즘 순서도



(그림 3) 송신 순서도

4.5 수신 알고리즘 순서도



(그림 4) 수신 순서도

5. 결론 및 향후 연구 과제

인터넷은 이제 단순히 어떠한 정해진 정보를 주고받는 수준에서 벗어나 대량의 정보를 생산하고 교환하며 재분배하는 과정을 만들어 내고 있다. 이 속에 포함된 다양한 정보들은 그 중요도의 높고 낮음을 떠나서 안전하게 보호되고 보존 되어야 할 이유가 있다. 하지만 지금의 암호화 방식은 언젠가는 그 해독 방법이 나올 수 밖에 없다는 단점을 가지고 있다.

본 논문에서 제안한 기법도 인터넷 컨텐트 애플리케이션의 통신 보안에 궁극적인 해결책이 될 수는 없다. 하지만 랜덤 프로시저를 활용하여 지속적으로 패킷의 암호화에 변화를 주어 예측 불가능 상황을 만들어 내어 암호를 해석하는 것은 가능할 수 있다 하여도 원하는 패킷을 생성하는 것을 불가능하게 하게 할 수 있다는 점에서 인터넷 컨텐트 애플리케이션의 안정성을 높여줄 수 있다. 즉, 패킷 암호화 기법을 매 패킷 통신시마다 변형시켜 패킷을 가로채어도 패킷을 복호화 하기 어렵게 한다. 또한, 매 패킷마다 암호화 기법이 다를 경우 이전 패킷의 암호화 기법을 확인하여도 이번 전송 패킷의 암호화 순서를 모를 경우 변경하거나 임의로 전송할 수 없음은 물론 암호화와 복호화는 손쉽게 할 수 있도록 하여 암호화와 복호화에 들어가는 비용을 줄일 수 있다는 점에서 효율적인 암호화 기법이라고 할 수 있다.

단, 패킷의 내용을 확인해서는 안 되는 데이터정보의 경우에는 여기서 제안하는 기법을 사용하여서는 안 된다. 즉, 통장의 비밀번호나 상업적인 거래 중 노출되는 카드 정보 등은 이러한 방법을 사용하면 위험할 수 있다. 이것은 암호화된 패킷의 원문을 파악하는 것이 가능하다는 점과 금융 거래 등의 통신은 연속적이지도 지속적이지도 않기 때문이다.

본 논문에서 제시한 기법을 다양한 인터넷 컨텐트 애플리케이션에 적용할 수 있게끔 라이브러리화 할 경우 실시간 미디어 데이터나 온라인 게임 등과 같은 인터넷 컨텐트의 통신 보안[7][8]에는 큰 발전을 기대할 수 있을 것으로 기대된다. 또한, 암호화 테이블의 생성 가지수를 늘리고 복잡도를 증가시키는 연구와 테이블의 전송이 안정적으로 이루어짐을 보장 받을 수 있는 연구가 이루어져야 할 것이다.

참고문헌

[1] O.Elkeelany, M.M.Matalgah, K.P.Sheikh, M.Thaker, G.Chaudhry, D.Medhi, J.Qaddour, “Performance Analysis of IPSec Protocol: Encryption and Authentication”, 1164-1168, IEEE 2002

[2] 심재홍, 서재현, 강흥식, “웹 상에서의 암호기법 연구”, 333-349, 仁濟論叢, vol.17 No.1, 2002

[3] V.L.Voydoc, S.T.Kent, “Security Mechanism in High-Level Network Protocols”, ACM, 1983

[4] 반용호, 김종훈, “B2B 시스템에서의 이중 암호화를 위한 암호 프로토콜의 설계”, 95-101, 情報技術硏究所論文誌, Vol.9 No.2, 2002

[5] Alan O.Freier, Philip Karlton, Paul C.Kocher, “The SSL Protocol Version 3.0”, Internet-Draft, November 1996

[6] T.Dierks, E.Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.1”, RFC4346

[7] 임수정, 홍동철, 김수성, 김성주, 유행석, 한준탁, 장태무, “3D MMORPG 온라인 게임에서의 네트워크 분석”, 2004 년 동계 한국게임학회 총회 및 학술발표대회, 289-293, 2004 년 2 월

[8] 홍은실, 백상헌, 박일규, 김종성, 고동일, 최양희. “게임 전송 프로토콜(GTP) : 효율적인 게임 이벤트 데이터 전송을 위한 새로운 전송 프로토콜”, JCCI, 통신 정보 합동 학술대회, 2002 년 4 월