인터넷 보호 프로토콜에서 예상 평문 공격에 대응한 보호 캡슐 메커니즘

*최은수, *김강욱, *황찬식
*경북대학교 전자공학과
e-mail : eschoi@palgong.kyungpook.ac.kr

Security Capsule Mechanism against the Probable Plaintext Attack of the IP Security Protocols

*Eun-soo Choi, *Kang-wook Kim, *Chan-sik Hwang
*Dept. of Electronics, Kyungpook National University

요 약

평문과 암호문 쌍에서 일부 예측할 수 있는 평문들을 이용하여 비밀키를 찾는 공격을 예상 평문 공격(probable plaintext attack)[1]이라고 한다. 인터넷 보호 프로토콜은 각 헤더부분에서 예측할 수 있는 부분을 많이 가지고 있으므로 예상 평문 공격의 주요한 대상이 되고, 이러한 취약점은 현재 인터넷에서 상용으로 사용되고 있는 DES(Data Encryption Standard)[2]에서 두드러지게 나타난다.

본 논문에서는 인터넷 보호 프로토콜에서 각 헤더의 예상 평문 공격과 기존의 보호 체계에 대해서 기술하고 근본적인 예상 평문 공격에 대응하여 카오스 함수 난수 생성기를 이용한 보호 캡슐 메커니즘을 제안한다.

Ⅰ. 서론

최근에 인터넷 사용이 급증하면서 인터넷을 통해 많은 정보를 교환한다. 이러한 인터넷을 통해 전송되는 정보들은 제 3자에게 노출될 가능성이 훨씬 많아졌으며 해커와 같은 불법 침입자들의 위협이 날로 증가하고 있다. 이에 따라 인터넷 상에서 보안이 중요한 문제로 대두되고 있으며 현재 많은 연구가 진행중이다. 인터넷 보호 프로토콜은 각 헤더에서 많은 부분을 예측할 수 있으므로 예상 평문 공격의 주요한 대상이 되고 있다. 따라서, 인터넷 보호 프로토콜에서 근본적으로 가지고 있는 예상 평문 공격에 대한 보호 체계가 요구된다.

본 논문에서는 인터넷 보호 프로토콜에서 특히 취약한 예상 평문 공격에 대해서 살펴보고, 이미 제공되고 있는 기본적인 보호책을 설명한 다음, 그 한계점을 언급하고 새로운 보호책으로 보호 캡슐 메커니즘을 제안한다.

Ⅱ. 예상 평문 공격

1. IPSEC(IP-layer encryption and authentication) 구조

ESP(Encapsulating Security Payload)[3]에 대한 패킷 구조는 그림 1과 같으며 회색 부분은 DES를 이용하여 CBC(Cipher Block Chaining)[2] 모드로 암호화한 영역을 나타낸다.

첫 32 비트는 SPI(Security Parameter Identifier)[3]로 사용하여 보호 연관(security association)[3]의 인덱스로서 역할을 한다. 여기서 보호 연관은 비밀 키, 초기 벡터(initialization vector)[2] 등을 포함한 송수신 상방 간에 정해진 규약이다

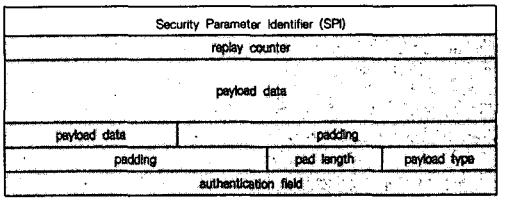

그림 1. ESP 패킷의 구조

2. 싱글 패킷 공격(single packet attack)[1]

싱글 패킷 공격은 복호화하려는 시도가 한 패킷 내에서 이루어지는 공격을 말한다. 공격 방법은 터널 모드(tunnel mode)[3] 사용여부에 따라 조금씩 다르며 초기 벡터를 아느냐에 따라서 첫 블록을 공격할 수도 있고 못 할 수도 있다. 그림 1에서와 같이 첫 번째 워드는 재생 카운터(replay counter)이다. 초기에는 재생 카운터가 1부터 시작하도록 정의되었다. 그래서 보호 연관을 설정하자마자 패킷을 가로챈다면 32비트의 재생 카운터로부터 적어도 30비트를 알아 낼 수 있다는 문제점이 있다.

터널 모드를 이용한다면 재생 카운터 다음에는 IP(Internet

Protocol) 헤더가 나온다. IP 헤더는 그림 2에 나타나 있다.

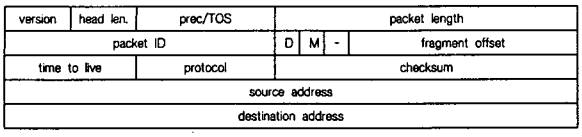

그림 2. IP 헤더의 구조

IP 헤더에서는 버전 번호, 헤더 길이, POS, 패킷 길이, 단편화 오프셋(fragment offset)과 플래그(flag) 필드, 프로토콜 식별자(protocol ID), 그리고 생존 시간(time to live) 및 주소부에서 총 54-58비트 정도의 예상 평문을 가진다.

트랜스포트 모드(transport mode)[3]를 사용할 경우에는 재생 카운터 다음으로 TCP(Transmission Control Protocol) 헤더나 UDP(User Datagram Protocol) 헤더가 나온다.

먼저 TCP 헤더의 경우 출발지 포트와 헤더 길이, 플래그, 윈도우 크기, 그리고 긴급 포인터(urgent pointer) 등으로부터 이상적인 상황에서 약 88비트의 예상 평문을 가진다.

UDP 헤더의 분석은 과정이 더 간단하다는 것을 제외하고는 TCP 헤더의 분석과 비슷하며 28 비트 정도의 예상 평문을 가진다.

3. 투 패킷 공격(two packet attack)[1]

하나의 연결이 이루어진 상태에서의 패킷 쌍을 분석함으로써 더 많은 예상 평문을 얻는 방법으로 두 개의 다른 패킷에서 평문이 동일한 부분을 알고 있을 때 같은 평문으로 복호화 되는 키를 찾아 공격하는 방법이다. 투 패킷 분석 장치는 하나의 키를 생성해서 병렬로 구성된 두 개의 복호화 엔진에 사용한다. 각각의 시험 키에 대해서 두 개의 복호화 장치를 사용하므로 탐색 시간은 싱글 패킷 공격과 비슷하나 장치에 대한 비용은 거의 두 배가 된다. 이러한 투 패킷 공격의 이점은 IP 헤더의 주소부 및 TCP와 UDP 헤더의 포트 번호에서 찾아볼 수 있다. 보통 이 투 패킷 공격을 이용하여 암호화에 사용된 키를 찾는다.

4. 기본적인 보호책

기본적인 보호책은 헤더 필드에서 예측할 수 있는 부분을 줄이는 방법을 사용한다. 첫 번째는 초기 벡터의 누출을 피하는 방법이다. 이렇게 함으로써 초기 벡터는 첫 블록에서 또 하나의 비밀키 역할을 한다. 두 번째는 재생 카운터의 정의를 바꾸어 1부터 시작하는 대신에 키 정보로부터의 난수로 시작하는 방법이다. 마지막으로 호스트간(host-to-host)[3] 터널 모드의 사용을 피하고 호스트와 방화벽(host-to-firewall)[3] 터널 모드에서 임시 IP 주소부에 대한 복사 부분을 난수로 대치하는 방법이 있다.

인터넷 보호 프로토콜에서 제공되는 이러한 기본적인 보호책은 어느 정도 암호화 강도를 더하지만 근본적인 예상 평문의 문제점을 제거하지 못하여 싱글 패킷 공격에 대해서도 약한 특성을 보일 뿐만 아니라 투 패킷 공격에 대해서는 아무런 대응책을 제시하지 못한다.

Ⅲ. 난수 생성기를 이용한 보호 캡슐 메커니즘

본 장에서는 카오스 함수 난수 생성기[4]를 이용하여 예상 평문 공격에 대응한 보호 캡슐 메카니즘을 제안한다.

1. 카오스 함수를 이용한 난수 생성기

일반적으로 카오스 함수는 초기 값을 모르면 똑같은 난수를 발생시킬 수 없다. 또한 초기 조건에 민감하므로 초기 상태에서의 아주 작은 불확실성이 계속 증폭되어 어느 정도의 시간이 지나면 상당한 차이를 불러 일으킨다. 이러한 카오스 함수의 특징을 이용하여 각 패킷의 재생 카운터를 초기 값으로 하는 난수 생성기를 만든다.

본 논문에서는 이산 카오스 사상(chaotic map)[4]의 대표적인 예인 로지스틱 사상(logistic map)[4]을 사용했으며 그 수식은 (1)과 같다.

$$x_{n+1} = \alpha x_n(1-x_n) \qquad (1)$$

파라미터 α는 $0 \le \alpha \le 4$의 범위를 가지고, 초기 값 x_0의 범위가 $0 \le x_0 \le 1$일 때 x_{n+1}은 바로 이전 상태 값인 x_n에 의해 결정된다. 하지만, 역으로 x_{n+1}이 주어질 때 가능한 x_n은 2차 방정식의 해가 되므로 비가역적이다. 여기서 α는 초기 값에 대한 다음 값의 의존성을 나타내는 감도인자(sensitivity parameter)이며 $3.57 < \alpha < 4$의 범위에 있으면 비주기의 카오스 성질을 가지게 된다. 그림 3은 본 논문에서 이용할 카오스 함수 난수 생성기이다.

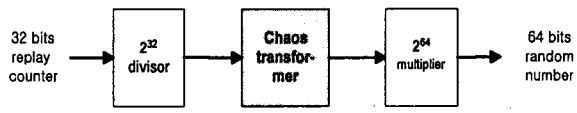

그림 3. 카오스 함수를 이용한 난수 생성기

2. 보호 캡슐 메커니즘

난수 생성기로부터 생성된 64비트 난수는 보호 연관 속성들 중의 하나인 32비트 재생 카운터 초기 값에 의존하기 때문에 시스템에 의해서 보호된다. 이 난수를 평문 패킷의 64비트 블록들과 각각 배타적 논리합 연산(XOR)을 취하여 평문 패킷에 대한 예상 평문 공격을 방지한다.

그림 4는 보호 캡슐 메커니즘을 나타낸다.

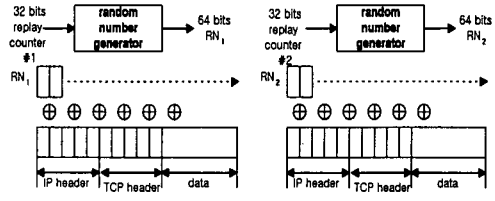

그림 4. 보호 캡슐 메커니즘

각 패킷의 재생 카운터로부터 난수 생성기를 통하여 생성된 난수들은 랜덤하므로 이 난수들과 배타적 논리합 연산을 취하여 생성된 각 평문 블록들도 랜덤성을 가지게 된다. 이렇게 생성된 각 블록들을 DES 암호화하게 되면 난수 생성기를 통해서 나온 난수가 각 패킷에 따라 다르기 때문에 평문 주소부가 같더라도 연산 후의 출력 평문이 서로 달라 주소부에 대한 투 패킷 공격을 방어할 수 있게 된다. 이와 같은 방식은 다른 키로 암호화를 한번 더 취하는 것에 비해 매우 짧은 시간 안에 처리할 수 있을 뿐만 아니라 근본적인 예상 평문의 노출을 방지할 수 있게 한다.

보호 캡슐 메커니즘을 구성하기 위해서는 보호 연관에 난수 생성기의 감도인자 α와 카오스 변환의 시행 횟수를 나타내는 n을 보호 속성으로 추가해야 한다. 감도인자 α는 $3.57<\alpha<4$의 수이므로 기본 값을 3.5로 하여 그 차분만을 보호 속성으로 취하였으며 본 논문에서는 24비트의 공간을 할당한다. 시행 횟수 n은 $100\leq n\leq 255$의 범위를 갖도록 8비트의 공간을 할당하여 전체적으로 볼 때 보호 연관에는 32비트 정도의 작은 오버헤더만이 추가된다.

보호 캡슐 메커니즘의 전체적인 흐름은 그림 5와 같다.

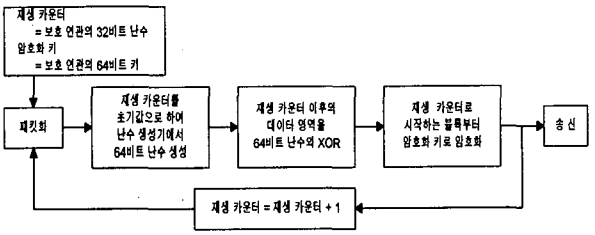

그림 5. 보호 캡슐 메커니즘의 전체적인 흐름도

3. 시뮬레이션 결과

본 논문에서는 문서 데이터를 정보로 하였으며 터널 모드에서 DES CBC 암호화 알고리즘을 사용하였다. 키 공간은 소프트웨어적으로 시뮬레이션을 가능토록 하기 위하여 실제 키 공간 56비트의 반인 28비트로 하였다. 이와 같은 조건에서 싱글 패킷 공격시 방어 체계를 사용하지 않았을 때와 기존의 보호 체계 중 재생 카운터를 보호할 때, 재생 카운터 및 초기 벡터를 보호할 때, 주소부에 난수를 사용할 때, 그리고 마지막으로 제안한 보호 캡슐 메커니즘을 이용할 때로 구분하여 각 암호화 블록에 적용하였다.

표 1과 2는 싱글 패킷 공격에 대한 시뮬레이션 결과를 나타내며 '-' 표시는 실험치를 56비트의 키 공간으로 확장할 수 없었다는 것을 나타낸다.

투 패킷 공격은 세 번째 블록인 주소부를 공격하였으며 기존의 보호 체계를 적용하였을 때는 정확하게 암호화에 사용된 키를 찾을 수 있었으나 보호 캡슐 메커니즘 적용시에는 완벽한 보호를 할 수 있었다.

마지막으로 보호 연관의 보호 속성을 추가하여 보호 캡슐 메커니즘으로 송수신 하였을 때, 거의 시간 지연이 없어 기존의 프로토콜에 쉽게 구현이 가능하였다.

	첫 번째 블록		두 번째 블록	
	실험치	이론치	실험치	이론치
보호 없음	-	7.81×10^{-3}	4.66×10^{-10}	3.29×10^{-10}
replay 보호	3.72×10^{-9}	3.72×10^{-9}		
replay+IV 보호	0	0		
보호캡슐	0	0	0	0

표 1. IP 헤더의 첫 번째와 두 번째 블록에서 검색 키로부터 암호화 키를 찾을 확률

	난수 주소 미사용		난수 주소 사용	
	실험치	이론치	실험치	이론치
보호 없음	1	1	1	1
replay 보호	1	1	1	1
replay+IV 보호	-	-	6.25×10^{-2}	3.28×10^{-6}
보호캡슐	0		0	

표 2. IP 헤더의 모든 예상 평문을 이용하여 검색 키로부터 암호화 키를 찾을 확률

Ⅳ. 결 론

본 논문에서는 현재 많이 사용되고 있는 인터넷 보호 프로토콜에서 각 헤더에서의 예상 평문에 대해서 알아보고 그 예상 평문에 대한 싱글 패킷 공격과 투 패킷 공격을 살핀 다음, 예상 평문 공격에 안전한 보호 캡슐 메커니즘을 제안하였다.

본 논문에서 제안한 보호 캡슐 메커니즘은 난수 생성기를 이용하여 생성된 64비트의 난수와 평문 데이터를 배타적 논리합 연산을 취한 후 비밀키로 암호화하여 예상 평문 공격에 안전하도록 하였다. 여기서 난수 생성기의 입력은 보호 연관의 32비트 난수를 통해서 만들어지는 재생 카운터이고 이 난수 생성기는 작은 차이의 입력에 대해서도 몇 번의 반복 수행 후에 서로 완전한 난수를 생성하는 카오스 함수를 이용하였다. 보호 캡슐 메커니즘을 인터넷 보호 프로토콜에 적용하였을 때, 오버헤더가 매우 작고 구현이 비교적 간단하며 실시간 구현이 가능하였다.

참고 문헌

[1] S. Bellovin, "Probable Plaintext Cryptanalysis of the IP Security Protocols", in *Proceedings of the Symposium on Network and Distributed System Security*, pp. 155-160, Feb. 1997.

[2] B. Schneier, "Applied Cryptography 2nd edition:Protocols, Algorithms, and Source code in C", John Willy & Sons, 1996.

[3] R. Atkinson. "Security Architecture for the Internet Protocol". Request for Comments (Proposed Standard) RFC 1825, Internet Engineering Task Force, Aug. 1995.

[4] H. Konno & T. Kondo, "Iterative Chaotic Map as Random Number Generator", *Ann. Nuct. Energy*, vol. 24, no. 14, pp. 1183-1188, 1997.