효율적인 멀티캐스트 서비스를 위한 인증 기법

정 유 미[†] · 박 정 민[††] · 채 기 준[†††] · 이 상 호[†††] · 나 재 훈[††††]

요 약

멀티캐스트는 송신자가 많은 수신자들에게 동시에 데이터를 전송하므로 송신자의 자원을 절약하고 네트워크의 점유율을 낮춤으로 효율성을 제공하는 통신 기술이나 여러 수신자들이 같은 그룹 주소를 사용하므로 공격을 받을 위험이 크다. 따라서 멀티캐스트 스트림에 대한 인증 및 서명은 중요한 문제이다. 본 논문에서는 패킷을 전송한 송신자의 신원을 확인하고 데이터가 변조되지 않았음을 확인하는 소스 인증 방법으로 다중 체인 인증기법을 제안하였다. 제안한 기법은 부인 방지를 제공하고 여러 패킷에 대한 전자서명으로 인증함으로써 오버헤드를 줄이며 지연 없이 이루어지므로 실시간 멀티미디어 서비스에 사용할 수 있다.

Authentication Mechanism for Efficient Multicast Service

Yumi Jung[†] · Jung-Min Park[††] · Kijoon Chae[†††]
Sang-Ho Lee[†††] · Jaehoon Nah[††††]

ABSTRACT

Multicast communication is simultaneous transmission of data to multiple receivers and saves considerably sender resources and network bandwidth. It has high risk to attack using group address and inherent complexity of routing packets to a large group of receivers. It is therefore critical to provide source authentication, allowing a receiver to ensure that received data is authentic. In this paper, we propose the multiple chain authentication scheme for secure and efficient multicast stream. To evaluate the performance of our scheme, we compare our technique with two other previously proposed schemes using simulation results. Our scheme provides non-repudiation of origin, low overhead by amortizing the signature operation over multiple packets, and high packet loss resistance.

키워드 : 멀티캐스트(Multicast), 인증(Authentication), 보안(Security), 소스 인증(Source Authentication)

1. 서 론

인터넷의 성장과 상업화로 데이터를 여러 수신자에게 동시에 전송하는 통신 방법인 멀티캐스트가 널리 보급되고 있다. 인터넷 비디오 전송, 뉴스 전송, 주식시세정보, 소프트웨어 업데이트, 실시간 다자회의, 온라인 비디오 게임 등과 같은 다양한 어플리케이션에 사용되고 있는 멀티캐스트는 송신자가 보낸 각 데이터 패킷이 수많은 수신자에게 전달되므로 송신자의 자원을 절약할 수 있고 네트워크의 점유율을 낮춤으로써 효율성을 제공하는 기술이다. 그러나 송·수신자가 서로 일대일(point-to-point) 통신을 하는 유니캐스트(unicast)에서와는 달리 멀티캐스트는 다수의 수신자들이 동시에 패킷을 받을 수 있도록 그룹주소를 이용하므로 그룹에 가입한 다른 수신자가 송신자로 위장하여 패킷을 전송할 경우 서비스 거부(denial of service) 공격을 당하기 쉽고 인터넷과 같은 공공망을 이용할 때 공격받을 가능성이 매우 높다[1, 2]. 따라서 다양한 공격으로부터 안전하게 멀티캐스트 서비스를 제공하기 위하여 인증, 기밀성, 무결성과 같은 보안 문제가 적합하게 고려되야 한다.

멀티캐스트에서 인증은 그룹 인증과 소스 인증으로 나뉜다. 그룹 인증은 통신에 참여하는 그룹 멤버에 대한 사용자 인증으로써 각 그룹의 멤버는 받은 메시지가 그룹 멤버가 보낸 것인지의 여부를 확인할 수 있다. 그룹 인증은 그룹 통신에 참여하기 위하여 적용된다. 멀티캐스트에서는 여러 수신자들이 동시에 패킷을 받으므로 수신자가 송신자로 위장하여 그룹 멤버에게 공격할 수 있다. 따라서 안전한 멀티캐스트 통신을 위해서 수신자는 받은 데이터가 요청한 소스로부터 온 것인지 또는 수신자에게 오는 경로 상에서 전송 중에 그룹 멤버나 공격자에 의해서 변경되지 않았음을 확인할 수 있도록 하는 소스 인증이 필요하다[3]. 소스 인증

※ 본 논문은 2003년 한국전자통신연구원 정보보호연구단 위탁연구과제 결과임.
† 정 회 원 : 이화여자대학교 과학기술대학원 컴퓨터학과
†† 준 회 원 : 이화여자대학교 과학기술대학원 컴퓨터학과
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
†††† 정 회 원 : 한국전자통신연구원 책임연구원
논문접수 : 2003년 9월 22일, 심사완료 : 2004년 9월 8일

은 데이터의 수신자를 신뢰할 수 없는 상태일지라도 받은 데이터는 소스가 보낸 것이며 수신자에게 오는 경로 상에서 데이터의 내용이 변경되지 않았음을 수신자가 확인할 수 있도록 한다. 일반적으로 기밀성은 불법적인 제3자에게 정보가 공개되지 않음을 의미하며 무결성은 전송되는 데이터들이 불법적으로 변경된 경우 이를 감지하고 그에 대해 적절한 조치를 취하도록 함을 의미한다. 소스 인증은 메시지가 요청한 송신자로부터 온 것인지 인증한 후 그 메시지가 전송 중에 변경되지 않았는지 확인하므로 무결성을 제공하며 또한 수신자는 메시지를 인증 후에 사용하므로 기밀성도 제공할 수 있다.

본 논문에서는 멀티캐스트의 중요한 보안 문제인 소스 인증에 있어서 그룹 인증을 받은 멤버가 생성하여 전송하는 데이터에 대한 소스 인증 기법을 제안하였다. 제안한 인증 기법에 대한 효율성을 알아보기 위해서 기존의 소스 인증 기법인 전자서명 기법과 EMSS 인증 기법 및 제안한 인증 기법을 시뮬레이션을 통하여 성능을 비교·분석하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 멀티캐스트에서 지금까지 연구되어온 소스 인증 방법에 대하여 살펴보고 특징을 분석한다. 3장에서는 제안하는 멀티캐스트 인증 방법에 대하여 설명하며 4장에서는 시뮬레이션을 통해 제안한 인증 기법에 대한 성능을 분석한다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

2. 관련 연구

안전한 멀티캐스트 통신으로 잘 알려진 방법은 멀티캐스트 그룹 멤버들이 서로 하나의 키를 공유하도록 하는 단일 공유 키(single shared key)방법이다. 이 방법은 오직 그룹의 구성원만이 메시지를 복호화할 수 있으므로 메시지 암호화에 적합하다. 그러나 모든 멤버들이 동일한 키를 공유하므로 그 키를 이용하여 그룹의 송신자들을 서로 식별할 수 없으므로 단일 공유 키 방법은 소스 인증을 해결하기에는 부적합하다. 유니캐스트에서 사용하는 일대일 인증 기법은 두 통신 당사자가 서로 비밀키를 공유하고 각 패킷에 그 공유된 키로 계산한 MAC(message authentication code : 메시지 인증 코드)를 덧붙임으로써 인증을 수행한다. 이 소스 인증 방법은 비밀키를 공유한 어떤 수신자가 데이터를 위조할 수 있으며 마치 자신이 송신자인 것처럼 위장할 수 있으므로 멀티캐스트의 소스 인증에 사용하는 것은 적합하지 않다. 또한, 멀티캐스트의 소스 인증을 위해서 비대칭 암호 기법을 기반으로 각 데이터 패킷을 전자 서명할 수 있다. 그러나 전자 서명을 사용한 소스 인증 방법은 서명과 검증에 많은 시간이 소요되며 대역폭에 있어서 매우 큰 부담이 된다.

앞서 살펴본 바와 같이 멀티캐스트와 같은 그룹 통신은 기존의 네트워크와 다른 속성을 가지므로 기존에 사용되었던 인증 기법을 그대로 적용할 수 없다. 멀티캐스트의 소스 인증에 관하여 제안되었던 연구들을 살펴보면 다음과 같다.

2.1 다중 메시지 인증 코드(Multiple MAC) 기법

공개키 서명에 대한 대안으로 R. Caetti[4] 등이 제안한 이 방법은 메시지 인증 코드를 기반으로 인증하는 기법이다. 대칭키 암호화를 이용하며 송신자는 여러 개의 키 {K_1, K_2, K_3, …, K_k}를 가지며 수신자 R_i는 {K_i}를 가진다. 송신자는 전송할 메시지 m에 대하여 자신이 가진 모든 키에 대한 MAC 값 즉, {$MAC(m, K_1)$, $MAC(m, K_2)$, …, $MAC(m, K_k)$}를 계산하여 전송한다. 수신자는 송신자와 공유한 키를 이용하여 패킷에 포함된 메시지의 MAC을 계산하고 송신자가 전송한 MAC 값들과 비교하여 일치하는 값이 있으면 메시지를 처리하고 일치하는 값이 없으면 패킷을 버린다. 이 방법은 수신자들이 서로 결탁하여 특정 수신자의 패킷을 위조하는 일이 없도록 안전하게 키를 분배한다고 가정하므로 수신자는 송신자의 모든 키와 다른 수신자들이 가진 키를 알 수 없어서 메시지를 위조할 수 없다. 그러나 모든 메시지가 k개의 MAC을 가지고 있어야 하고 패킷을 보내기 전에 서버가 k개의 MAC을 계산해야 하므로 오버헤드가 크며 수신자가 서로 결탁할 수 없는 정도로 수신자가 존재하는지에 따라서 기법의 안전성이 달라진다. 따라서 이 기법은 멀티캐스트 그룹이 작고, 그룹 멤버들 사이의 결탁을 통제할 수 있는 경우에만 사용할 수 있다. 더불어 이러한 문제를 방지하기 위하여 비밀키를 안전하게 유지하는 방법이 필요하다.

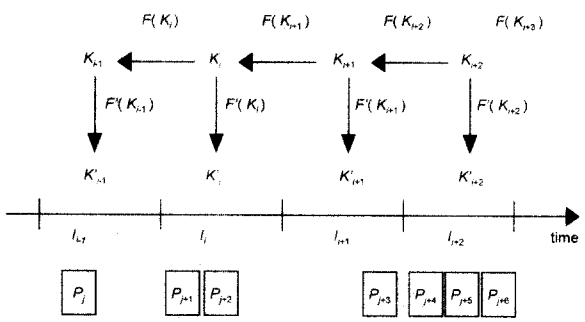
2.2 TESLA(Timed Efficient Stream Loss-tolerant Authentication)

TESLA[5, 6]는 연구그룹인 IRTF의 GSEC(Group SECurity)과 IETF의 MSEC(Multicast SECurity)에서 그룹 통신 및 멀티캐스팅을 이용한 다양한 어플리케이션의 데이터 전송 측면에서 보안을 위하여 제안되었다. TESLA는 소스 인증을 제공하며 인증 알고리즘으로 MAC을 사용한다. 기존 네트워크 환경에서 사용되는 대칭키 암호화를 멀티캐스팅으로 전송되는 데이터에 그대로 적용하면 보안이 충분히 이루어지지 않으므로 TESLA는 수신자가 송신자와 시간을 동기화하고 송신자는 키 체인을 생성하여 형성된 키 체인의 키를 순차적으로 이용하여 각 패킷에 대한 MAC을 계산하여 패킷과 함께 전송한다. 이때 사용되는 키는 일정 시간이 지난 후 공개하여 인증을 수행한다. 수신자는 시간 오차 범위를 두어 동기화하며 소요된 시간과 오차 범위 내에서 발생한 지연은 허용하고 그 외의 경우에 대한 패킷은 다른 수신

자나 공격자에 의해 처리된 것으로 간주한다. TESLA의 처리과정을 순차적으로 설명하면 다음과 같다.

• TESLA

1. 송신자 설정
   // 시간 간격 설정
   1.1 시간 간격의 시작 시간, 시간 간격, 키 공개 지연 설정
      // 키 공개 지연 d는 패킷 전송 및 일정시간이 지난 후 키를 공개하기 위해 사용된다.
   // 키 체인 설정
   1.2 키 체인의 길이 N과 키 체인의 마지막 키 값 K_N을 랜덤하게 선택
   1.3 의사-난수 함수(pseudo random function)[9] F를 사용하여 키 체인을 계산한다.
      즉, $K_i = F(K_{i+1})$을 이용하여 일방향 키 체인 $K_0, K_1, \cdots, K_N$을 계산한다.
   1.4 각 구간에서 메시지의 MAC을 생성하는데 사용되는 키를 구하기 위해 일방향 함수 F'을 이용하여 $K'_i = F'(K_i)$를 계산한다.
2. 수신자와 동기화
   2.1 새 수신자와의 초기화(bootstrap)를 위해 처음 패킷을 RSA와 같은 전자 서명으로 인증하여 보낸다. 처음 패킷은 1.1의 정보와 1.3의 키 체인에 대한 위탁이 포함된다.
3. 인증 패킷의 전송
   3.1 i 시간에 일방향 키 체인으로 연결된 키 중 K_i를 사용하고 이전에 사용된 키 K_{i-d}를 전송한다.
4. 수신자 인증
   4.1 가장 최근에 받은 키가 K_j라면, 수신자는 이전에 받은 키를 이용하여 K_i가 사용할 수 있는 키인지를 확인하고 $K'_i = F'(K_i)$를 계산하여 i 구간에 받은 패킷을 인증한다.

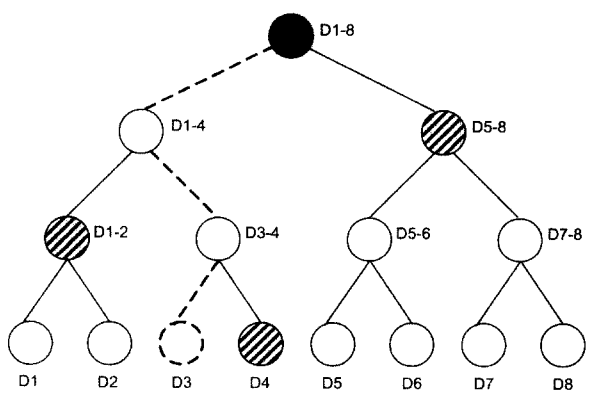


(그림 1) TESLA : 키 체인 생성 및 MAC key 구축 방법

즉, TESLA는 인증을 제공하기 위하여 각 패킷에 MAC을 포함하며, 대응되는 MAC 키들을 일정 시간 지연 후에 수신자에게 공개한다. 키 공개 지연 시간을 충분히 길게 하면 패킷이 위조되는 것을 막을 수 있다. 이 방법은 키 체인 방법을 사용함으로써 패킷 손실에 강하며 융통성이 있으나 송신자와 수신자가 일정 시간 내에 시간 동기화를 해야만 하므로 송수신자간에 시간 동기화를 설정하는 것이 어려운 경우 TESLA는 제대로 동작할 수 없으므로 사용할 수 없다.

2.3 트리 체이닝(Tree Chaining) 기법

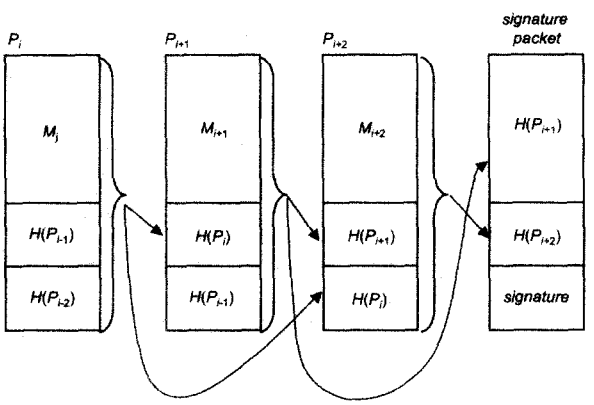
각 패킷에 대해 인증하는 방법은 계산 시간으로 인해 대부분의 응용에서 사용하기가 적합하지 않으므로 대안으로 패킷의 그룹인 블록에 대하여 서명하여 인증한다. 트리 체이닝 기법[7]은 시간 간격동안 모아진 패킷에 대해 Merkle [8]의 인증 트리를 생성하고 인증트리의 루트를 서명한다. 이 기법은 스트림을 여러 개의 메시지로 나눠 한 번의 서명 연산으로 메시지의 서명을 생성한다. 트리 체이닝의 인증 트리는 통상적인 트리 구조와 같이 말단(leaf) 노드, 중간 노드, 루트 노드로 구성된다. 한 스트림을 (그림 2)와 같이 예를 들어 8개의 메시지로 나눈다면 각 메시지에 대해 해쉬값 D1, D2, …, D8을 계산한다. 이와 같이 계산된 해쉬값들을 말단 노드라고 하면 그 위에 있는 중간 노드들은 말단 노드 또는 자신의 자식(child) 노드의 해쉬 값을 이용하여 계산한다. 반복 계산으로 루트의 해쉬 D1-8을 구하고 서명 Sig[D1-8]을 계산한다. 이 방법으로 인증하는 경우 각 패킷은 메시지와 서명, 서명을 생성하기 위한 패킷의 위치, 말단 노드에서 루트 노드까지의 경로에 있는 노드의 형제(sibling) 노드에 대한 정보와 같은 추가로 발생한 다양한 정보를 함께 전송한다. 예를 들어 (그림 2)의 점선으로 표시되는 세 번째 패킷을 검증하는 경우, 점선으로 나타나는 경로의 모든 노드를 검증해야 한다. 검증자는 받은 패킷의 다이제스트 D'3을 계산하고 트리를 따라서 윗 노드를 검증해나간다. 즉, D'3-4 = h(D'3, D4), D'1-4 = h(D1-2, D'3-4), D'1-8 = h(D'1-4, D5-8)이며 여기서 빗금으로 표시된 D4, D1-2, D5-8 노드는 패킷 서명을 가지고 있다. 검증자는 D'1-8이 블록 서명 Sig(D1-8)의 블록 다이제스트인 D1-8과 같은지를 검사한다. D'1-8 = D1-8이면 패킷은 검증된 것이다. 즉, 서명을 검증하기 위하여 수신자는 메시지의 위치를 확인하고 추가 인증 정보를 이용하여 계산한 서명과 받은 서명을 비교한다. 기존의 서명 생성 기법과 비교해볼 때 이 방법은 서명과 검증은 빠르게 수행되나 패킷이 서명뿐만 아니라 추가 정보를 포함하므로 통신 부담이 늘어난다는 단점이 있다. 또한 송신자가 패킷을 지연하고 그룹화해야 하므로 인터랙티브한 멀티캐스트 어플리케이션에는 사용할 수 없다.



(그림 2) 트리 체이닝 기법

2.4 EMSS(Efficient Multi-chained Stream Signature)

TESLA는 부인 방지를 제공하지 않으며 시간 동기화가 어려운 경우에는 동작하지 않으므로 부인 방지와 소스 인증을 동시에 구현하고자 EMSS[9]가 제안되었다. 부인 방지를 위한 보편적인 서명 기법으로는 RSA[10]와 Rohatgi의 K-회 서명 기법[11]이 있다. 그러나 이러한 서명 기법들은 모든 패킷에 대하여 서명을 하므로 계산 및 통신 부담이 너무 크다. EMSS은 이러한 부담을 줄이기 위하여 하나의 서명으로 다중 패킷을 서명한다. 즉, 연속되는 패킷에 대하여 부인 방지를 수행하기 위하여 다음과 같이 동작한다. 패킷 P_i는 이전 패킷 P_{i-1}에 대한 해쉬 값 $H(P_{i-1})$를 포함한다. 스트림의 맨 마지막에 마지막 패킷의 해쉬와 함께 서명을 포함한 *서명 패킷*을 보냄으로써 모든 패킷에 대한 부인 방지를 수행한다. 패킷 손실에 강인하게 하기 위하여 각 패킷은 이전 패킷들에 대한 다중 해쉬값을 포함하며 마지막 서명 패킷은 다중 패킷의 해쉬를 서명한다. EMSS는 각 패킷에 몇 개의 다중 해쉬를 포함할지에 따라서 달라진다. 각 패킷에 포함할 다중 해쉬가 2인 EMSS를 예로 나타내면 (그림 3)과 같으며 그림에서 보는 바와 같이 각 패킷은 해당 패킷 이전의 두 개의 패킷에 대한 해쉬값을 가지며 서명 패킷은 그 패킷 이전의 마지막 두 패킷의 해쉬와 그 두 패킷의 해쉬값에 대한 서명으로 이루어진다.



(그림 3) EMSS의 예

송신자는 스트림의 서명을 연속적으로 검증하기 위하여 주기적으로 서명 패킷을 전송한다. 수신자는 서명 패킷을 받아 서명을 검증하고 서명 패킷 내에 있는 해쉬를 이용하여 연결된 패킷들에 대하여 부인 방지를 제공한다. 수신자는 다음 서명 패킷을 받은 후에 패킷의 서명만을 검증하므로 수신자가 패킷을 검증할 때까지 지연이 생긴다. 즉, EMSS는 서명 패킷을 마지막에 받으므로 실시간 서비스에 적합하지 않다. 또한 인접한 패킷들이 한꺼번에 손실될 경우 체인이 끊어지며 인증정보가 없으므로 연속해서 인증이 제공되지 않는 단점이 있다.

3. 제안한 인증 기법

3.1 멀티캐스트 인증 고려 사항

일반적으로 유니캐스트에서 인증은 송수신자가 명확하기 때문에 MAC 만으로 보안이 비교적 충분히 제공된다. 유니캐스트와 달리 멀티캐스트의 인증은 2장에서 살펴본 바와 같이 멀티캐스트에서 인증을 위해 대칭키 암호화를 사용한 기법은 MAC을 계산하고 이용하기 위하여 별도의 키 관리를 하였으며, 비 대칭키 암호화를 적용한 경우에는 각 패킷마다 전자 서명을 생성하여 전송함으로써 엄청난 오버헤드가 발생하므로 여러 패킷에 대하여 서명을 생성하는 방안이 제안되었다. 또한 멀티캐스트의 송·수신자는 제한된 자원을 이용하므로 이 점도 고려해야 한다. 더불어 전송 패킷이 인터넷을 통과하면서 손실되어 수신자가 받지 못하는 경우, TCP는 신뢰성을 제공하며 손실된 패킷을 재전송하여 손실에 적절히 대응할 수 있으나[12, 13] 대부분의 멀티캐스트 어플리케이션들은 UDP를 이용하므로 전송 중 데이터가 손실되면 수신자는 송신자가 보내는 데이터를 수신할 수 없으므로 패킷 손실에 대한 적절한 처리가 필요하다. 멀티캐스트로 제공되는 많은 어플리케이션들은 멀티미디어 서비스이며 대부분 실시간으로 이루어지므로 효율적으로 이용할 수 있도록 해야 하며, 받은 데이터는 처리 후 데이터를 저장할 필요가 없으며 실행 후 버리므로 부인 방지의 필요성이 크지 않지만, 중요한 데이터에 대해서 요청 받은 송신자 이외의 다른 그룹 멤버가 데이터를 전송하는 경우에 대비하여 수신자는 실제로 데이터를 보낸 송신자가 전송 사실을 부인하지 못하도록 하는 부인 방지 서비스를 제공할 필요가 있다. 따라서 안전하고 효율적인 멀티캐스트 서비스를 제공하려면 다음과 같은 사항을 고려해야 한다.

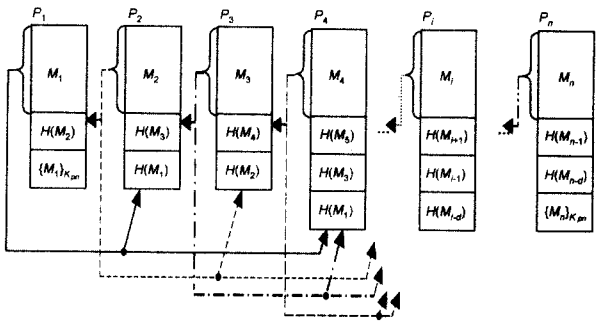
- 다양한 대역폭 지원 및 오버헤드 최소화 : 다양한 네트워크 조건에서 어플리케이션이 제공되므로 서비스에 따라 대역폭과 지연이 달라진다. 따라서 통신 오버헤드가 적게 생성되는 것이 바람직하다.
- 이질적인 수신자 환경 및 패킷 손실 처리 : 다른 대역폭, 지연, 네트워크 혼잡을 가진 멀티캐스트 참여자는 TCP 또는 UDP로 데이터를 수신한다. TCP는 수신 여부를 확인할 수 있지만 UDP에서는 데이터 손실에 대한 피드백이 없으므로 신뢰성 있는 데이터 전송을 보장하지 못한다. 그러나 대부분의 멀티캐스트 서비스가 UDP를 기반으로 이루어지므로 패킷이 지연되거나 손실이 생기는 경우에 대하여 적절한 처리가 필요하다.
- 보안 : 멀티캐스트 그룹 멤버들은 동적으로 변하므로 특정 송신자가 특정 그룹에 속한 멤버에게 전송하는 것을 제어할 수 있어야 한다. 또한 데이터를 암호화하고 수신자를 제한함으로써 접근 권한을 가진 수신자만이 데이터를 사용할 수 있어야 하며 인증된 송신자가 전송한 데이터

는 변경되지 않아야 한다. 더불어 중요한 정보에 대해 부인방지가 이루어져야 하며 실시간 통신 정보에 대하여 처리 지연이 발생하지 않도록 해야 한다.

3.2 다중 체인 인증 기법(MCA : Multiple Chain Authentication Scheme)

본 논문은 안전한 멀티캐스트 통신을 위하여 그룹에 가입한 정당한 멤버가 전송하는 데이터를 정당한 수신자들만이 이용할 수 있도록 하기 위하여 소스 인증 방법인 다중 체인 인증 기법을 제안하였다. 제안한 소스 인증 기법은 EMSS를 확장 개선한 방법이다. EMSS 기법은 전자 서명 기법만을 사용하는 경우에 비하여 오버헤드가 적으며 인접한 패킷의 손실에 대해서는 해쉬 체인으로 인증을 제공할 수 있다. 또한 전자 서명한 서명 패킷을 마지막에 보냄으로써 부인방지를 제공한다. 그러나 서명 패킷이 들어올 때까지 기다려야만 인증이 이루어지므로 실시간 서비스에 적합하지 않다. 또한 인접한 패킷들이 한꺼번에 손실될 경우는 체인이 끊어져서 인증 정보가 없으므로 연속해서 인증이 제공되지 않을 수 있다. 인접한 손실이 한꺼번에 발생하는 것을 방지하기 위하여 EMSS에서 해쉬의 수를 증가할 수 있으나 그만큼 오버헤드가 증가하게 된다. 따라서 제안하는 인증 기법은 바로 이웃하지 않은 패킷에 인증 정보를 함께 전송하도록 체인을 형성하여 연속해서 패킷이 손실되는 경우에 대비한다. 또한 제안한 다중 체인 인증 기법은 제한된 수의 전자 서명으로 여러 패킷의 인증을 위임함으로써 모든 패킷에 전자 서명을 붙이는 방안에 비해 오버헤드가 적다. EMSS와 달리 실시간 서비스에 사용할 수 있도록 하기 위하여 처음 패킷에 전자 서명을 생성하여 전송한다. 소스 인증에서 대칭키 암호화를 적용하는 경우 안전하게 키를 관리해야 한다. 본 연구에서는 인증 정보를 생성하는데 이용되는 키 관리를 용이하게 하기 위하여 공개키 기반 암호화를 적용하며 인증을 위해 필요한 키는 이미 공유하고 있다고 가정한다.

제안하는 인증 기법인 다중 체인 인증 기법을 도시하면 (그림 4)와 같다. (그림 4)는 다중 체인 인증 기법에서 다중 체인 형성 간격이 3인 경우를 예로 나타낸 것이다. 송신자는 전송할 파일을 n개의 데이터 스트림으로 나눈다. 첫 번째 패킷 P_1에는 첫 번째 패킷 메시지, 인접한 다음 스트림 즉, 두 번째 데이터 스트림 M_2에 대한 해쉬값 H_2, 송신자 자신이 가진 비밀키 K_{pri}를 이용하여 생성한 전자서명($\{M_1\}K_{pri}$)을 포함해서 전송한다. 다중 인증 체인을 형성하기 위하여 첫 번째 데이터 스트림 M_1에 대한 해쉬값 H_1을 구해놓는다. 데이터 스트림으로 생성된 해쉬 값은 다중 체인 형성 간격의 패킷에 포함시켜 전송하여 연속적으로 패킷 손실이 발생하는 경우에 패킷 손실에 강인하게 한다. M_n까지 반복해서 해쉬 값을 계산하고 체인으로 연결하며 마지막 메시지 M_n은 첫 번째 패킷과 마찬가지로 전자 서명(Sig)을 생성하여 마지막 패킷으로 구성하여 전송한다.



(그림 4) 다중 체인 인증 기법

수신자가 패킷을 받았을 때, 첫 번째 패킷 P_1을 받아 송신자의 공개키 K_{pub}를 이용하여 전자 서명을 검증한다. 전자 서명으로 송신자에 대한 인증과 부인방지가 이루어지며 패킷에 대한 연속적인 인증이 이루어진다. 그리고 P_2를 받은 후 M_2에 대한 해쉬값를 구하고 P_1에 이미 포함되어 있던 해쉬값을 통해서 메시지의 변경이 이루어지지 않았음을 확인한다. 제안한 기법은 첫 패킷과 마지막 패킷에 전자 서명을 포함한다. EMSS과 같이 마지막 패킷에만 전자 서명을 생성하여 전송하는 경우 또는 처음 패킷에만 전자 서명을 생성하여 전송할 경우, 패킷 손실 대비 인증 확률을 높이기 위하여 일정한 간격을 두고 전자 서명이 포함된 패킷을 전송해야 한다. 그러나 제안한 방법은 처음과 마지막 패킷에 전자 서명을 포함하므로 둘 중 하나의 패킷이 손실된다고 하더라도 인증과 부인 방지를 모두 제공할 수 있다.

제안한 인증 기법은 다음과 같은 특징을 갖는다. 첫째, 바로 이웃하지 않은 패킷에 인증정보를 함께 전송하도록 체인을 형성함으로써 연속하여 패킷이 손실되는 경우를 방지한다. 둘째, 여러 패킷에 대하여 제한된 수의 전자 서명으로 인증함으로써 모든 패킷에 전자서명을 덧붙이는 방안에 비해 오버헤드가 적다. 셋째, EMSS와 달리 처음 패킷에 전자 서명을 생성하여 전송함으로써 실시간 서비스에 적용하여 소스 인증을 제공할 수 있다.

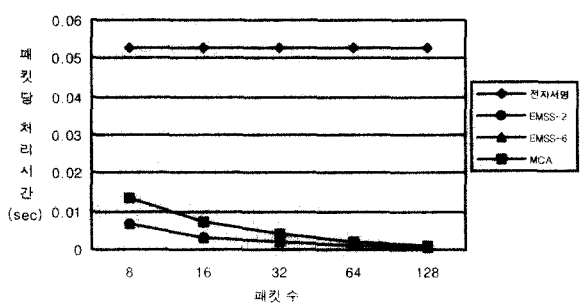
4. 시뮬레이션 및 성능 분석

본 논문에서는 제안한 소스 인증 기법에 대한 성능을 평가하고 분석하기 위하여 시뮬레이션을 수행하였다. 시뮬레이션은 ns-2[14]를 사용하였으며 인증 구현은 암호 기법들이 C++ 클래스 라이브러리로 되어있고 다양한 암호 알고리즘을 지원하는 Crypto++[15]를 이용하였다. 제안한 인증 기법의 성능을 비교하기 위하여 전자서명으로는 RSA, 해쉬 알고리즘으로는 MD5[16]를 사용하였다. 데이터 패킷의 크기는 512바이트이며 UDP를 이용하여 전송된다고 가정하였다.

시뮬레이션은 다음과 같은 네 가지 즉 일반적인 부인 방지 기법인 전자 서명 기법, 기본적인 EMSS인 한 패킷 당 포함되는 해쉬가 두 개인 EMSS-2, 포함하는 다중 해쉬값이 많은 경우와의 성능 비교를 위해 한 패킷 당 포함되는 해쉬가 여섯 개인 EMSS-6 그리고 제안한 방법인 다중 체인 인증 기법(MCA)에 대하여 처리 지연 및 오버헤드, 전송 처리율 등의 성능을 측정하여 비교하였다.

4.1 송신측 처리 지연

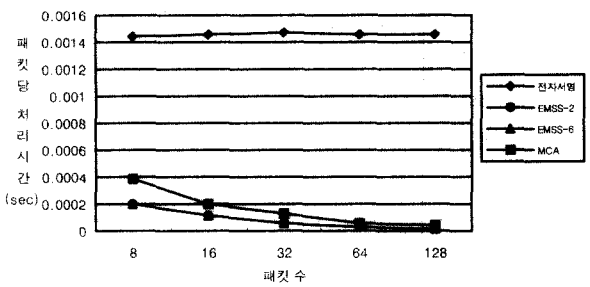
송신자가 패킷에 대한 인증 정보를 생성하기 위하여 소요된 처리 시간을 나타낸다(그림 5). 첫 번째 패킷에서의 처리 시간은 송신자가 첫 메시지에 대하여 자신의 비밀키를 이용하여 전자 서명을 생성하는 시간과 체인을 형성하기 위하여 패킷에 포함된 메시지에 대한 해쉬를 계산하여 이웃한 노드에 해쉬값을 붙이는 시간으로 구성된다. 중간의 패킷들에 대하여는 체인을 형성하기 위하여 해쉬값을 계산하는 시간이 소요되며, 마지막 패킷은 첫 번째 패킷과 마찬가지로 송신자의 비밀키로 전자 서명하는 시간이 걸린다. 제안한 기법은 모든 패킷에 전자 서명을 생성하는 일반적인 전자 서명 기법보다는 시간이 적게 걸리지만, EMSS와 비교해볼 때 제안한 기법은 시간이 더 소요되었다. 이것은 EMSS는 한 개의 서명으로 모든 패킷에 인증을 위임하기 때문이다. 그러나 패킷의 수가 증가함에 따라 인증 정보를 생성하는데 걸리는 시간의 차이는 많이 줄어듦으로 패킷 수의 증가에 따라 제안한 기법은 처리 지연이 감소함을 알 수 있다. 멀티캐스트 스트림은 대체로 패킷의 수가 매우 크므로 EMSS와 비교할 때 성능 저하 없이 적용할 수 있다고 해석할 수 있다.



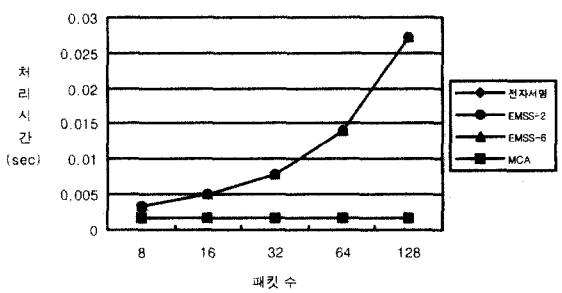
(그림 5) 송신측 처리 지연

4.2 수신측 처리 지연

수신자가 인증 정보를 받은 후 검증하는데 걸리는 패킷 당 처리 소요 시간을 나타낸 것이다(그림 6). 제안한 방법은 첫 번째 패킷을 받자마자 패킷에 포함된 메시지와 송신자의 공개키를 이용하여 서명을 검증한다. 또한 메시지에 해쉬 함수를 적용하여 해쉬값을 구한다. 다음 패킷을 받으면 메시지의 해쉬값을 구하여 비교하는 것을 반복적으로 수행한다. 제안한 인증 기법은 송신자의 처리 지연과 유사하게 수신자의 처리 지연 측면에서 전자 서명 기법에 비해 적게 걸렸으나 EMSS-2, EMSS-6 보다 오래 걸렸다. 그러나 제안한 인증 기법과 전자서명기법은 수신자가 첫 번째 패킷을 받자마자 인증할 수 있는데 반해 EMSS 기법은 마지막 서명 패킷이 들어올 때까지 인증을 제공할 수 없다(그림 7). 따라서 제안한 인증 기법은 실시간 어플리케이션에 적용할 수 있는 장점이 있으며 첫 번째 패킷이 전송 중에 손실되지 않는 한 마지막 패킷에 포함된 서명을 검증하지 않더라도 부인 방지가 이루어지므로 수신자의 처리 시간을 단축할 수 있는 장점이 있다.



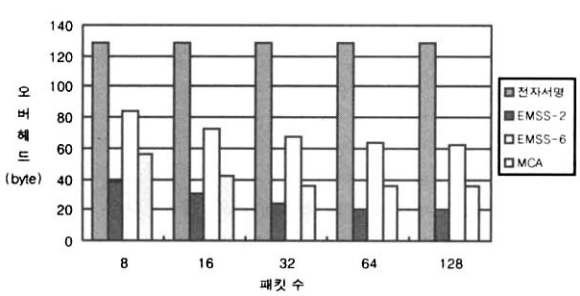
(그림 6) 수신측 처리 지연



(그림 7) 첫 패킷 인증 지연

4.3 패킷당 오버헤드

인증 기법을 적용함에 있어서 하나의 패킷 당 발생하는 인증 정보를 나타낸 것이다. 일반적인 전자서명의 경우는 다른 경우들에 비해 패킷 당 오버헤드가 상당히 크다. EMSS 기법과 제안한 인증 방안은 각각의 패킷들을 체인으로 형성하는 해쉬값들을 어떻게 연결하는지에 따라서 패킷 당 발생하는 오버헤드가 가변적이긴 하지만, (그림 8)에서 보는 바와 같이 송·수신자의 처리 지연과는 달리 제안한 인증 방안은

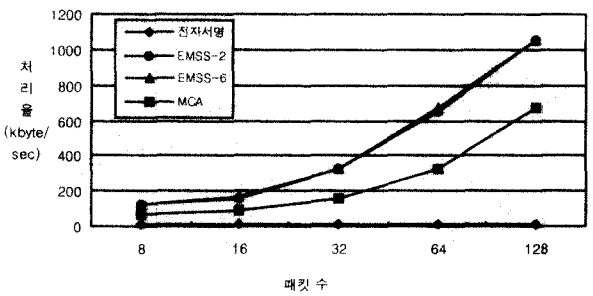


(그림 8) 패킷당 오버헤드

EMSS-2에 비해 오버헤드가 크지만 EMSS-6에 비해서 적음을 볼 수 있다.

4.4 데이터 전송 처리율

(그림 9)는 인증 방법을 각각 적용하였을 경우에 대한 데이터 전송 처리율을 나타낸 것이다. EMSS에 비해 전송 처리율은 낮지만 전자서명 방법보다는 처리율이 높은 것으로 나타났으며 이는 제안하는 방법이 처음 패킷에서 전자서명을 하므로 그로 인해 전송 지연이 발생한 것으로 해석된다.



(그림 9) 데이터 전송 처리율

5. 결 론

본 논문은 안전한 멀티캐스트 통신을 위하여 그룹에 가입한 정당한 멤버가 전송하는 데이터를 정당한 수신자들만이 이용할 수 있도록 소스 인증 방법인 다중 체인 인증 기법을 제안하였다. 제안한 인증기법은 EMSS를 확장·개선한 방법으로 EMSS와 비슷하게 패킷 손실에 대비하여 패킷 정보를 다른 패킷에 붙임으로써 체인을 구성하였으나 EMSS와는 달리 체인으로 연결하는 위치를 변경하였다. EMSS에서는 한 개의 서명으로 여러 개의 패킷에 대한 인증을 위임하도록 하였으나 제안한 방법은 두 개의 서명을 적용하였으며 이로 인해 인증 정보 생성 및 검증을 처리하기 위한 처리시간이 EMSS보다 다소 더 걸린다. 그러나 EMSS와 달리 실시간 인증 및 부인 방지를 효율적으로 제공할 수 있다. 또한 이웃한 패킷에 해쉬를 붙이는 다중 체인 기법을 사용함으로써 한꺼번에 패킷들이 손실되는 경우에 있어서 EMSS에 비해 강력하게 대응할 수 있는 잇점이 있다.

시뮬레이션 결과, 제안한 기법은 다른 기법에 비해 상대적으로 작은 오버헤드를 가지며 전송 처리 지연 측면에서도 효율적임을 알 수 있었다. 멀티캐스트의 다양한 어플리케이션의 조건을 만족하면서 송·수신자의 처리 지연을 좀 더 단축할 수 있는 연구 및 안전성의 이론적인 검증에 대한 연구가 앞으로 더 보완되어야 한다.

참 고 문 헌

[1] B. Quinn, K. Almeroth, "IP Multicast Application: Challenges and Solutions," RFC3170, Sep., 2001.

[2] C. Diot, B. N. Levine, B. Lyles, H. Kassem, D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," IEEE Network, Vol.14, pp.88-98, Jan., 2000.

[3] A. Perrig, R. Canetti, D. Song and J. D. Tygar, "Efficient and Secure Source Authentication for Multicast," Proceedings of Network and Distributed System Security Symposium(NDSS) 2001, Feb., 2001.

[4] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," INFOCOM'99, Vol.2, pp.708-716, Mar., 1999.

[5] A. Perrig, R. Canetti, D. Song, D. Tygar and B. Briscoe, "TESLA : Multicast Source Authentication Transform Introduction," Internet draft, IETF, 2002.

[6] A. Perrig, R. Canetti and B. Whillock, "TESLA: Multicast Source Authentication Transform Specification," Internet draft, IETF, 2002.

[7] C. K. Wong and S. S. Lam, "Digital Signatures for Flows and Multicasts," IEEE Trans. on Networking, Vol.7, No. 4, pp.502-513, Aug., 1999.

[8] R. Merkel, "A Certified Digital Signature," Advanced in Cryptology(CRYPTO '89), pp.218-238, Aug., 1989.

[9] A. Perrig, R. Canetti, J. D. Tygar and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," Proc. of IEEE Symposium on Security and Privacy, pp.56-73, May, 2000.

[10] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, Vol.21, No.2, pp. 120-126, 1978.

[11] P. Rohatgi, "A Compact and Fast Hybrid Signature Scheme for Multicast Packet Authentication," Proc. of ACM Conference on Computer and Communications Security, Nov., 1999.

[12] M. Borella, D. Swider, S. Uludag and G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS," In Proc. of the International Conference on Parallel Processing, pp.3-15, 1998.

[13] V. Paxson, "End-to-End Internet Packet Dynamics," IEEE/ACM Trans. on Networking, Vol.12, No.5, pp.277-292, 1999.

[14] "The Network Simulator : ns-2," http://www.isi.edu/nsnam/ns/.

[15] "Crypto++," http://www.eskimo.com/~weidai/cryptlib.html.

[16] R. L. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, 1992.

[17] W. Stalling, *Network Security Essentials : Application and Standards*, Prentice Hall, 2001.

[18] A. Perrig, "The BiBa One-Time Signature and Broadcast Authentication Protocol," ACM Conference on Computer and Communications Security, pp.28-37, 2001.

정 유 미

e-mail : yumi@ewha.ac.kr
1999년 충남대학교 천문우주과학과 학사
2003년 이화여자대학교 과학기술대학원 컴퓨터학과 석사
관심분야 : 네트워크 보안, 네트워크 프로토콜, 무선이동통신보안

박 정 민

e-mail : pjm@kist.re.kr
1989년 이화여자대학교 전자계산학과 (이학사)
1991년 이화여자대학교 대학원 전자계산학과(이학석사)
1991년~현재 한국과학기술연구원 연구원
1999년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 박사과정
관심분야 : 네트워크 보안, Mobile IP, Active Network Security, Ad-hoc Network

채 기 준

e-mail : kjchae@ewha.ac.kr
1982년 연세대학교 수학과(이학사)
1984년 미국 Syracuse University 컴퓨터학과(이학석사)
1990년 미국 North Carolina State University 컴퓨터공학과(공학박사)
1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수
1992년~현재 이화여자대학교 컴퓨터학과 교수
관심분야 : 네트워크 보안, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석

이 상 호

e-mail : shlee@ewha.ac.kr
1979년 서울대학교 계산통계학과 학사
1981년 한국과학기술원 전산학과 석사
1987년 한국과학기술원 전산학과 박사
1990년 미국 일리노이대학교 전산학과 방문교수
1984년~현재 이화여자대학교 컴퓨터학과 교수
관심분야 : 정보보호, 암호프로토콜, 알고리즘 설계, 데이터 마이닝, Bioinformatics

나 재 훈

e-mail : jhnah@etri.re.kr
1985년 중앙대학교 컴퓨터공학과 학사
1987년 중앙대학교 대학원 컴퓨터공학과 석사
1987년~현재 한국전자통신연구원 책임연구원
관심분야 : IPsec, Mobile IP, IPv6, 네트워크 보안