멀티캐스트 환경에서의 메시지 인증 방법 비교

이상학*, 홍기훈*, 정수환*

*숭실대학교, 정보통신전자공학부

Message Authentication Schemes for Multicast Security

Sang-hak Lee*, Kihun Hong*, Souhwan Jung*

*School of Electronics Engineering SoongSil Univ.

요 약

멀티캐스트는 다자간 그룹 통신에 적합기술로서 유니캐스트에 비해 네트워크 자원측면에서 매우 효율적인 프로토콜이다. 이러한 멀티캐스트 기반의 확대를 위해 멀티캐스트에서의 보안을 제공해 줄 수 있는 방안이 반드시 강구되어야 한다. 본 논문에서는 메시지 인증을 위해 제안되어진 보안 구조들을 비교, 분석하고 각각의 보안 구조들의 장·단점을 살펴본다. 특히, IETF MSEC 작업반에서 멀티캐스트 보안 표준으로 제안하고 있는 TESLA에 대해 자세히 분석하고 보안 요구사항들을 검토해 보았다.

I. 서론

멀티캐스트는 다자간 그룹 통신에 적합한 기술로서 유니캐스트에 비해 네트워크 자원 및 송신측에서의 처리 용량 측면에서 매우 효율적인 프로토콜이다. 이러한 멀티캐스트를 널리 보급하기 위해선 충분한 보안 특성을 제공해 줄 수 있어야 한다. 하나의 패킷이 수백만 수신부에 도달할 수 있는 멀티캐스트의 강력한 특징은 공교롭게도 커다란 위험 부담도 가지고 있다. 즉, 멀티캐스트 통신에서 악의적으로 하나의 패킷을 보내는 공격자가 있다면 이 잘못된 패킷은 또한 잠재적으로 수백만 수신부에 도달할 수 있다. 그렇기 때문에 수신자는 데이터 스트림이 내가 목적했던 송신자로부터 전송된 것인지를 보장해 주기 위한 방법이 필요하다. 가장 두드러진 멀티캐스트에서의 보안 위험성은 수신측에서의 데이터 인증이다.

멀티미디어는 실시간을 고려하고 있기 때문에 비 연결 상태인 데이터 그램 방식을 사용하여 수신측에서 패킷의 수신이 실패한 경우, 재전송을 요청하지 않고 다음 수신된 패킷을 받아서 처리한다. 또한 수신자들이 그룹에 가입과 탈퇴가 매우 유동적이므로 기존에 제안되고 표준화되어있는 유니캐스트의 보안 프로토콜을 그대로 적용 할 수 없는 어려움이 있다.

메시지를 인증하기 위해 현재 몇 가지 방법이 있다. 그룹의 송/수신자가 같은 키를 공유하게 될 경우 공유키를 가지고 있는 어떤 수신자가 데이터를 위조하고 송신부를 모방할 수 있다는 문제점을 가지고 있기 때문에, 대칭키 방식의 패킷 인증 법은 본 논문에서 원하는 멀티캐스트 통신 환경에서의 메시지 인증을 제공하지 않는다. 다른 방법인 비대칭키를 이용한 방법은 각각의 데이터 패킷 서명으로 소스 인증을 수행하지만 많은 양의 오버헤드가 필요하다는 단점을 가지고 있다. 본 논문에서 주로 다루고자 하는 멀티캐스트 인증 프로토콜인 TESLA는 위의 대칭·비대칭키 방식을 잘 조합하여 메시지 인증을 위한 효과적인 방법을 제시하고 있다[1].

본 논문에서는 2장에서 멀티캐스팅에서 패킷 인증을 위해 제안된 알고리즘을 분석하고, 3장에서는 IETF(Internet Engineering Task Force) MSEC(Multicast Security) 작업반에서 표준으로 제안하고 있는 TESLA(Timed Efficient Stream Loss-tolerant Authentication)에 대한 자세한 분석과 보안 요구사항들을 알아본다[2]. 그리고 마지막 결론으로 멀티캐스트 보안 표준으로 제안되기 위해 TESLA에서 고려되어야 할 사항들을 제시한

다.

II. 메시지 인증 구조

1. 다중 MAC 구조

비대칭 MAC(Multiple Authentication Code) 구조를 기본 개념으로 하여 송신자는 여러 개의 비밀키를 가지고 있고, 각각의 수신자는 송신자로부터 비밀키의 부분집합을 얻게 된다[3]. 송신자는 하나의 메시지에 대해서 모든 비밀키로 MAC계산을 하여 패킷에 붙여주고, 수신측에서는 가지고 있는 비밀키로 MAC계산을 하여 패킷이 송신자로부터 왔음을 인증하게 된다. 송신자 이외에는 정당한 MAC을 생성할 수 없으나, 많은 사용자가 공모할 경우 비밀키 전체가 노출될 수가 있고, 큰 규모의 멀티캐스팅 그룹에 부적합하다는 단점이 있다.

2. 스트림 서명 구조

디지털 서명을 사용하는 방법으로 전송 시작 시 패킷에 대한 서명을 보내고 각각의 패킷은 다음 패킷의 해시 값을 포함하거나 일회용 비밀키로 서명한 값을 포함하고 수신측에서는 일회용 공개키로 패킷을 인증한다[4]. 그러나 이런 방법은 전송 신뢰도 문제를 고려하지 못한 구조로 중간의 패킷을 잃어버리게 되면 인증 사슬 구조가 끊어져 버리게 된다. 또한 인증서와 일회용 공개키는 상당한 크기로 패킷에 상당한 오버헤드를 가져온다.

3. 트리 사슬 구조

패킷을 블록 단위로 구분하여 인증정보들을 생성, 전송하므로 송신자는 패킷의 지연을 허용해야 하며 각각의 패킷은 루트의 인증서와 인증에 필요한 부수적인 정보들을 추가로 가진다[5]. 블록단위로 인증 정보를 가지게 되고 각각의 패킷에 대한 증명이 가능하다. 또한 처음 도착한 패킷에 대해 인증을 한 후 이 정보를 저장해 놓고 같은 블록의 패킷 인증에 사용할 수 있다. 하지만 블록 단위로 패킷을 모아야 하므로 대화식 멀티캐스트 응용에 부적합한 단점이 있다.

4. 혼합 구조

오프라인과 온라인 서명 구조를 가지고 있는 방법으로 계산량이 많이 소요되는 인증서에 관련된 작업을 오프라인 상에서 먼저 해놓고, 온라인 상에서 메시지를 해시하여 패킷에 대한 서명을 완성하는 방법이다[6]. 그러나 여기서 인증서를 사용하고 있으므로 비실용적으로 패킷의 크기가 커지는 단점이 있다.

5. 일방향 노출키 기반 구조

TESLA는 현재 IETF 멀티캐스트 보안 작업분과에서 표준으로 제정하기 위해 드래프트로 제안된 상태이며 다음과 같은 특성을 갖는다[1].

- 인증 정보의 입증과 생성을 위한 적은 계산량
- 패킷을 인증하기 위해 사용되는 인증 정보의 작은 크기
- 패킷 손실에 대한 강한 특성
- 송·수신부 사이에 느슨하게 동기화된 시간 하에 안전성을 보장

기본 동작과정은 PKI(Public Key Infrastructure)를 이용하여 송신부와 수신부 간에 세션 셋업을 끝내고 이후 패킷들은 송신부에서 일방향 해시 사슬구조로 생성한 키를 사용하여 패킷에 MAC을 붙이고 다음 시간 간격에 키를 노출하게 된다. 송신부에서 이때 받은 키 값을 사용하여 이전 패킷의 MAC을 계산, 패킷의 무결성을 확인하게 된다.

1) 송신부 초기화

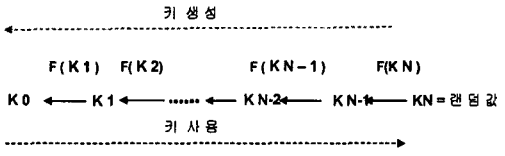

그림 1 : 일방향 키 사슬 구조

송신부는 일방향 키 사슬(one-way key chain) 구조의 길이 N을 결정하고 최대 전송 시간 간격을 길이 N으로 제한한다. 송신부는 키를 구하기 위해 그림 2와 같이 랜덤한 값 하나를 선택하고 사슬의 나머지 값들은 의사 난수 함수 F()를 사용하여 Ki = F(Ki+1)을 반복적으로 사용함으로써 계산되며, 이후 생성된 키들은 역순으로 사용되어진다. 일방향 키 사슬 구조의 특성은 중간의 어떤 값을 가지고 있지 않더라도 키 사슬의 KN으로부터 원하는 위치의 값을 계산할 수 있다. 각각의 키 Ki는 일정한 시간 간격 I에서만 효력을 발휘하게 된다.

2) 수신부 초기화

수신부는 송신자에게서 다음과 같은 정보를 얻어오게 되며, 정보들은 PKI를 통해 송신자의 인증서를 주고 받음으로써 신뢰성을 보장한다.

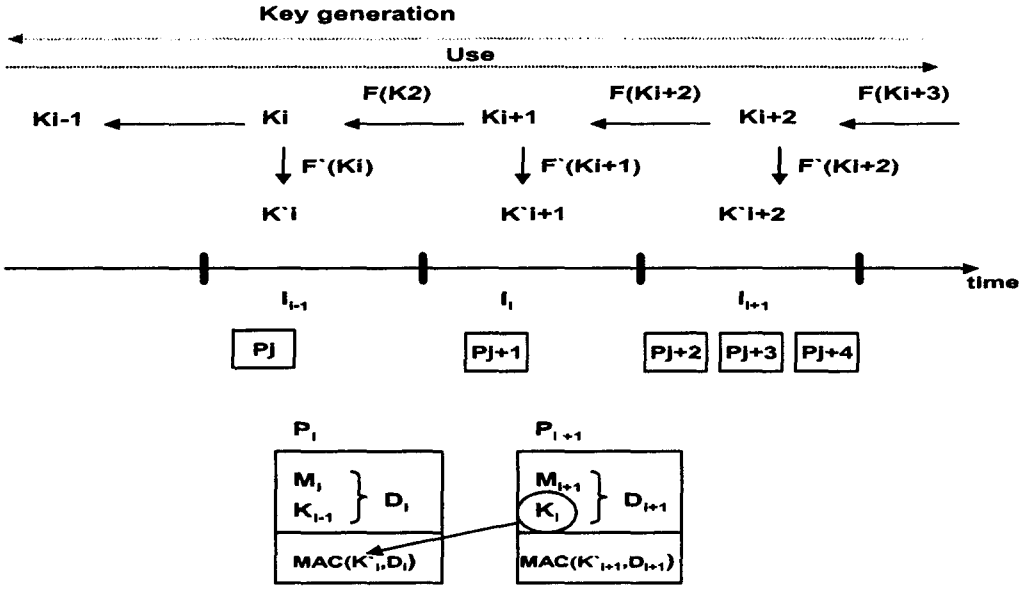

그림 2 : TESLA 구조와 패킷 구성요소

- 시간 간격 스케줄: 시간 간격, 시작 시간, 색인, 일방향 키 사슬의 길이
- 노출키의 지연
- 그룹에 가입하기 이전에 이미 노출되었던 키

3) 송신부에서의 메시지 전송

일 방향 키 사슬에서 각각의 키는 특정한 시간에서 효력을 발휘하기 위해 시간 간격에 따라 키를 일치시킨다. 송신부는 현재 시간 간격에 일치하는 키를 사용하여 메시지에 MAC을 추가하여 전송하게 된다.

4) 수신부에서의 인증

송신부가 하나의 키를 노출시키면 외부 다른 성분들은 잠재적으로 그 키에 접근한다. 이때 공격자는 모조된 메시지를 만들고 노출 키를 사용하여 MAC를 위조한다. 이러한 패킷들이 수신부에 도착할 수 있기 때문에 수신부는 안전한 키(safe key)를 기반으로 생성된 MAC를 입증해야 한다. 안전한 키(safe key)는 오직 송신부에 의해서만 알려지고 이런 안전한 키로 계산된 MAC이 있는 패킷과 메시지들만이 안전성을 보장한다. 수신부는 패킷들이 위조될 가능성이 있기 때문에 안전하지 못한 패킷들은 버려야 한다.

6. I-TESLA

5절에서 설명했듯이 TESLA에서 패킷의 인증은 다음 시간간격에 노출된 키를 사용하여 이루어지고 있다. 이런 시간상의 지연을 해소하기 위해 패킷을 받은 즉시 인증하는 방법을 위해 제안된 구조가 I-TESLA(Immediate-TESLA)이다[7].

패킷의 구성은 그림 3과 같으며 TESLA와는 다르게 송신측에 버퍼를 두어 다음 패킷의 해시값을 포함해서 전송하게 된다.

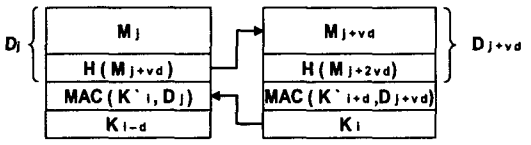

그림 3 : I-TESLA 구조

이후에 도착하는 패킷은 현재 패킷의 해시값을 가지고 인증을 할 수 있으므로 수신측에서 키 노출 시간 지연동안 패킷을 인증하지 못해 버퍼에 저장할 필요가 없어진다. 메시지에 대한 해시값은 이후 패킷이 네트워크상에서 손실된다면 그 사슬 구조가 끊어지지만, 이럴 경우에는 TESLA에서와 동일한 방법으로 일방향 키 사슬 구조를 이용해 패킷을 인증할 수 있다.

7. 제안된 구조들의 비교

메시지 인증 성능을 비교하기 위해 인증정보 생성에서의 효율성과 패킷 손실에 대한 신뢰성 의존도, 수신자들의 공모에 대한 취약성, 인증을 위한

지연시간 등과 같은 기준을 가지고 평가를 할 수 있다[8].

표 1 : 인증 구조 비교

평가항목	다중MAC	스트림서명	트리사슬	혼합구조	TESLA
속도	빠름	빠름	빠름	빠름	다중MAC 보다 빠름
크기	보통	매우큼	적당함	적당함	다중MAC 보다 좋음
신뢰성 의존도	비의존적	의존적	비의존적	비의존적	비의존적
공모	공모에취약	공모불가	공모불가	공모불가	공모불가
실시간성	실시간	실시간	블록시간 동안대기	실시간	실시간

표 1에서 정리된 것과 같이 효율성 측면에서 가장 좋은 구조는 서명을 사용하지 않는 다중 MAC 구조이지만, TESLA는 한 패킷에 대해 한번의 MAC만을 생성하게 되므로 속도와 크기면에서 다중 MAC 보다 우수한 성능을 갖는다. 스트림 서명의 경우 연속된 데이터의 해시 사슬 구조를 이용하여 인증을 하므로 패킷 손실에 대한 고려가 되지 못하고, 다중 MAC은 사용자의 공모에 취약성을 가지고 있다. 트리 사슬 구조는 인증을 생성하기 위해 한 블록을 기다려야 하므로 화상회의나 게임 등의 실시간성을 필요로 하는 응용에는 부적합하다.

메시지 인증을 위해 크게 MAC을 이용한 방법과 서명을 이용한 방법으로 나누어진다. MAC을 이용한 방법은 효율성이 좋은 대신 송신자와 수신자가 같은 키를 공유해야 한다는 문제점이 생기고, 서명을 이용한 방법은 이상적인 방법이지만, 계산 비용이 많아지는 단점을 가지고 있다. TESLA에서는 두 가지 방법에서 장점만을 취해 인증을 위해 필요한 정보들을 서명을 이용해 전달하고, 일방향 키 사슬을 이용하여 MAC을 확인함으로써 메시지의 인증을 보장한다.

III. TESLA 보안성 분석

TESLA에서는 키 재설정을 위한 방법이 제안되어 있지 않다. 게다가 키는 생성 순서와 역방향으로 사용되어지므로 엄청난 양의 키를 미리 계산해 두어야 처음 패킷부터 키를 이용하여 MAC을 계산할 수 있다. 특히 실시간을 고려한다면 데이터가 어느 정도 발생할 지 예측하기 곤란하므로 많은 키 체인을 생성해 두어야 한다.

시간 동기화(time sync.)를 맞추어야 하는데 이것은 송신자로부터 수신자들이 어느 정도의 거리에 떨어져 있느냐에 따라 문제가 될 수 있다. 예를 들어 송신자가 보낸 패킷을 송신자 앞에서 바로 받아 변조된 데이터에 노출된 키를 이용하여 MAC을 붙일 수 있다. 이것을 방지하기 위해 키가 바뀌는 시간 간격을 늘리면 되겠지만, 증가된 시간 간격은 수신자가 노출되는 키를 기다리는 시간이므로 패킷을 받고 노출되는 키를 기다리는 시간이 증가하므로 실시간성에 악영향을 준다. 또한 데이터에 대한 서명이 사용되지 않고 있으므로 부인 방지 기능을 가지지 못한다[9]. 이는 통신이 끝난 이후 송신자과 수신자 모두 같은 키를 가지게 되므로, 악의적인 수신자가 송신자가 제공한 적이 없는 다른 컨텐츠에도 정상적인 MAC을 생성할 수 있기 때문이다. 이를 위해서 데이터에대한 서명을 이용한 인증방법이 고려되어야 한다.

마지막으로 TESLA는 DoS(Denial of Service) 공격에 노출되어 있다[7]. 수신측에서 버퍼를 두어 한 지연시간동안 받은 패킷은 모아두고, 이후 송신자에 의해 노출되어진 키를 사용하여 저장해두었던 패킷을 인증한다. 다음 시간 간격까지는 무조건 받은 패킷을 모아두어야 하는 수신측의 특성을 이용하여 공격자가 모조된 큰 사이즈의 패킷을 전송하게되면 수신측의 버퍼가 용량을 초과하는 공격이 가능해진다. I-TESLA에서는 수신측에서 패킷의 해시값을 이용하여 받은 패킷에 대해 바로 인증 할 수 있으므로 수신측의 DoS에 대한 대안이 될 수 있다.

IV. 결론

멀티캐스트 패킷 인증 프로토콜을 수행하기 위해서는 인증 정보의 입증과 생성을 위해서 낮은 계산량이 요구되며 인증 정보의 크기가 작아야 하고 송/수신부에서 요구되는 버퍼의 크기가 작아야 한다. 또한 멀티캐스트의 특성상 패킷 손실에 대해서도 강한 특성을 보여야하며 많은 수신부를 가질 경우 이를 수용할 수 있어야 한다.

현재 제안되어지고 있는 패킷 인증 프로토콜들은 크게 MAC을 이용하여 인증하는 방법과 서명을 이용하는 방법으로 나뉘어지며 특성 또한 달라진다. 현재 MSEC 작업반에서 표준으로 제안하고 있는 TESLA는 제시된 조건을 모두 만족시키는 뛰어난 성능을 보이고 있다. 그러나 수신측에서의 DoS 공격에 대한 약점과 부인 봉쇄를 지원하지 못하는 등의 단점을 보이고 있는데 이를 보완하기 위해 서명을 사용할 수 있는 방안 등의 연구가 필요하다.

참고문헌

[1] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," RSA CryptoBytes , vol. 5, no. Summer, 2002.

[2] www.ietf.org/html.charters/msec-charter.html

[3] R.Canetti, J,Garay, G.Itkis, D.Micciancio, M. Naor, B.Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," IEEE INFOCOM '99.

[4] R. Gennaro, P. Rohatgi. "How to sign digital streams," CRYPTO 97, Lecture Notes in Computer Science 1294, Springer-Verlag, 1997, pp 180-197.

[5] C. Wong, S. Lam. "Digital Signatures for Flows and Multicasts," Proceedings IEEE ICNP '98, Austin TX, Oct 1998.

[6] S. Even, O. Goldreich, S. Micali, "On-Line/Off-Line Digital Signatures," Journal of Cryptology, 1996., 9(1):35--67.

[7] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in Network and Distributed System Security Symposium, NDSS '01 , February 2001, pp. 35--46.

[8] M. J. Moyer, J. R. Rao , P. Rohatgi, "A Survey of Security Issues in Multicast Communications," IEEE Network, November/December 1999.

[9] A. Perrig, R. Canetti, J. Tygar, and D. X. Song, "Efficient authentication and signing of multicast streams over lossy channels," in IEEE Symposium on Security and Privacy , May 2000.