情報保護學會論文誌
第14券 第3號, 2004. 6

복구 계층을 이용한 멀티캐스트 패킷 인증

홍 기 훈[†], 정 수 환[‡]
숭실대학교

An Efficient Authentication Scheme for Multicast Packet using Recovery Layer

Kihun Hong[†], Souhwan Jung[‡]
Soongsil University

요 약

멀티캐스트에서의 패킷 인증은 다수의 수신자들을 대상으로 하기 때문에 기존의 공유키를 통한 인증이 어렵고 빈번한 서명의 사용은 과다한 연산 시간으로 실제 적용이 무의미하다. 또한 기존의 연구에서 TESLA는 네트워크 상황이 일정치 않은 상태에서 시간 동기화를 요구하거나, 해시기반의 인증 방법들은 과도한 패킷의 확장을 통한 전송 과부하를 유발하고 수신 측에 인증 처리를 위해 많은 버퍼와 지연 시간을 요구한다. 본 논문에서는 이러한 문제점을 해결하기 위해 해시 체인과 서명을 이용한 인증 방법에 XOR를 사용하여 복구 계층을 구성하고 손실된 패킷의 해시 체인을 복구함으로써 중첩된 해시의 사용으로 증가하는 패킷의 확장을 줄였으며, 수신 측에서 패킷을 수신 후 바로 인증을 확인할 수 있도록 하여 수신측의 버퍼와 계산의 부담을 줄였다. 이 방법은 또한 서명을 통한 복구 계층의 무결성을 보장하므로 부인 방지 기능을 지원한다.

ABSTRACT

This paper proposes an efficient authentication scheme for multicast packet using recovery layer to provide source authentication. The problems of the existing schemes are as follows : TESLA requires time synchronization between the sender and the receiver, and hash-based schemes have high communication overheads due to additional hash values and require many buffers and delay for verification on receivers. Our main focus is reducing the buffer size, communication, and computation burden of the receiver. The proposed scheme in this paper is highly robust to packet loss using the recovery layer based on XOR operation. It also provides low communication overhead, low verification cost, non-repudiation of the origin, immediate verification, and robustness against DoS attack on the receiver.

Keywords : *Packet authentication, Multicast, Recovery.*

Ⅰ. 서 론

멀티미디어 기술의 발전으로 인터넷의 트래픽은 기존의 텍스트와 그림이 아닌 동영상이나 음성 스트림이 주가 되고 있다. 이러한 변화를 수용하고 주로 응용될 수 있는 형태가 바로 멀티캐스트(Multicast)로서 네트워크의 자원을 절약하고 많은 수신자들에게 정보를 전달할 수 있는 네트워크이다. 멀티캐스트는 소스에서 많은 수신자들에게 동일한 패킷을

접수일 : 2003년 12월 22일 ; 채택일 : 2004년 6월 8일
* 본 연구는 숭실대학교 교내 연구비 지원으로 수행하였습니다.
† kihun@cns.ssu.ac.kr
‡ souhwanj@ssu.ac.kr

전송하기 위해 송신 측에서 하나의 패킷을 전송하면 가입한 수신자들에게로 패킷이 복사되어 전달되도록 구성되어 있다. 그러나 이러한 멀티캐스트의 특성 때문에 누구나 공격 패킷을 이용하여 거짓 정보를 많은 멀티캐스트 사용자들에게 보낼 수 있다. 이러한 문제점을 해결할 수 있는 방안이 바로 멀티캐스트에 패킷 인증을 적용하여 사용자와 패킷을 인증하고 관리하는 것이다. 패킷의 인증을 통해 그룹에 속한 멤버만이 패킷을 송신할 수 있도록 하고 그룹 구성원이 아닌 공격자가 거짓 데이터를 그룹의 구성원에게 보내면 인증을 통해 수신 측에서 버릴 수 있도록 한다. 그러나 멀티캐스트에서의 패킷 인증은 하나의 송신자와 수많은 수신자들로 구성되어 비대칭적인 구조를 가진다. 따라서 수신측이 다수가 되기 때문에 일반적으로 보안 분야에서 사용하는 일대일 구조의 공유키 방식을 사용할 수 없다. 같은 공유키를 가진 악의적인 수신자가 공격자가 될 수 있기 때문이다.

이를 해결하기 위해 키 지연 노출 방법을 사용하는 MAC(Message Authentication Code)기반의 TESLA(Timed Efficient Stream Loss-tolerant Authentication)와 해시(Hash)기반의 메커니즘들이 제안되었다. 그러나 TESLA는 시간 동기화가 요구되고 해시기반의 메커니즘들은 패킷의 손실에 대한 효율적인 해결책을 제시하지 못하고 있다. 기존의 해시기반 메커니즘들은 패킷 손실에 대한 저항성을 높이기 위해 주로 패킷에 많은 추가적인 해시를 포함하도록 하고 있다. 그러나 이러한 방법은 크기가 작은 멀티미디어 패킷에 엄청난 양의 데이터를 추가함으로써 매우 비효율적이다. 또한 수신 측에서 블록 대부분의 패킷을 받아야 처리가 가능하기 때문에 많은 버퍼와 수신측 지연, 과도한 계산을 필요로 한다. 그러나 모바일 시스템과 같이 적은 자원을 가지는 수신자가 실시간 응용 프로그램을 이용하는 경우, 이러한 많은 처리와 지연 시간으로 실시간 패킷을 실시간으로 처리하기 어렵다. 본 논문에서 이러한 문제들을 분석하여 적은 패킷의 확장으로 이를 방지할 수 있는 패킷 인증 메커니즘을 제안하고자 한다. 따라서 본 논문에서 제안하고 있는 복구 계층을 이용한 패킷 인증 메커니즘은 패킷 손실에 의한 해시 체인의 손상을 막기 위해 복구 계층을 두어 패킷이 손실되더라도 손실된 패킷의 해시 부분을 복구함으로써 연결되는 해시 체인의 유효성을 보장하는 메커니즘이다. 또한 복구 계층과 복구 계층의 무결성을 보장하기 위해 무결성 확인 과정을 포함하고 있다. 이러한 처리는 또한 수신 측에서 수신한 패킷의 정보를 기반으로 하기 때문에 수신측 지연 시간과 버퍼링이 없이 수행되며, 인증 메시지의 확인에 적은 비용이 소요되어 자원이 적은 수신 시스템에서 실시간 스트림의 처리에 적합하다.

본 논문은 다음과 같이 구성된다. 우선 Ⅱ장에서 현재까지 제안된 관련 기술들을 살펴보고 각 기술의 문제점들을 알아본다. Ⅲ장에서는 제안하는 인증 방법의 기본 인증 모델과 복구 원리를 설명하고 복구 계층을 적용한 인증 모델을 제시하며 이에 대하여 상세히 설명한다. Ⅳ장에서는 제시된 인증 모델과 다른 인증 방법과의 전송 과부하와 인증 확인 비율을 비교하여 제시된 인증 방법의 우수성을 보이고 Ⅴ장에서 결론을 서술한다.

Ⅱ. 관련 기술

1. MAC 기반의 메커니즘

MAC기반의 인증 메커니즘들은 MAC에 사용되는 공유된 키를 기반으로 상대방을 인증하는 메커니즘이다. Perrig에 의해 제안된 TESLA(Timed Efficient Stream Loss-tolerant Authentication)는 시간 지연 키 노출 방법과 대칭적 암호 방법을 통한 데이터 인증을 수행한다. 송신자와 수신자는 시간 동기화가 되어 있어 패킷 도착 시간과 키 노출 시간이 지연 시간을 두고 동기화된다. TESLA에서 인증 키 지연 노출을 통해 수신 측에서 이미 도착한 패킷을 다음 패킷을 통해 인증하는 방법으로 멀티캐스트의 비대칭 특성을 극복하는 방법으로 사용되고 있다. 가장 큰 특징은 패킷의 손실에 강한 패킷 인증 방법이라는 것이다. 그러나 TESLA는 패킷 전송 시간의 방향과 역 방향으로 엄청난 양의 키를 미리 계산해 두어야 처음 패킷부터 키를 이용하여 MAC 값을 계산할 수 있다. 특히, 실시간을 고려하고 있으므로 실시간 데이터가 어느 정도 발생할 지 예측하기 곤란하므로 많은 키 체인을 생성해 두어야 한다. 또한 시간 동기화를 맞추어야 하는데 이 것은 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 MAC 값을 붙일 수 있다. 또 다른 문제는 각 수신 노드와 송신 노드와의 시간차가 다르고 네트워크의 상황에 따라 RTT

(Round Trip Time)가 다르다는 것이다. 각 수신 노드의 위치가 다르므로 전달되는데 걸리는 시간이 다르므로 키 노출 시간을 각 수신 노드가 다르게 계산하여야 한다. 그러나 문제는 송신자가 하나의 지연 시간을 가지고 키를 노출시키는 것이 문제점이다.[1,3-4]

Cannetti에 의해 제안된 이 메커니즘은 송신자가 ℓ개의 MAC 키 그룹을 가지고 있고 각 수신자들이 한정적 수인 이 키들의 하위 그룹을 소유하고 있게 된다. 각 메시지는 ℓ개의 MAC 키들과 MAC 값이 계산되어 전송되고 각 수신자들은 자신들이 가지고 있는 하위 그룹 키를 이용하여 인증하게 된다. 이 방법은 여러 수신자들이 자신이 가지고 있는 키 하위 그룹을 통해 전체 ℓ개의 MAC 키 그룹을 알아낼 수 있는 공모의 위험성이 있으며 계산과 메시지의 증가가 커서 전송 과부하가 큰 단점이 있다.[1,2,6]

2. 해시 기반의 메커니즘

해시 기반의 인증 메커니즘은 해시의 단방향성을 이용하여 체인을 구성하고 체인의 처음이나 마지막 패킷에 서명을 붙여 정당한 송신자와 해시 체인의 관련성을 입증한다. Gennaro와 Rohatgi에 의해 제안된 해시 체인 방법은 처음 패킷에 서명을 붙이고 각 패킷은 그 다음 패킷의 해시값을 포함하고 있다. 그러나 이 방법은 중간에 하나의 패킷만 손실되어도 해시체인이 끊어지기 때문에 전체 스트림에 대한 연속적인 인증에 실패한다. 따라서 이러한 문제점을 개선하기 위해 해시 체인을 구성 시에 바로 앞이나 뒤가 아닌 일정 거리가 떨어진 패킷의 해시값들을 패킷에 포함하여 해시 체인을 강화하였다. 이 방법들은 EMSS (Efficient Multi-chained Stream Signature) 혹은 Golle and Modadugu의 augmented 체인이다. 그러나 이러한 방법도 모든 패킷의 해시값을 메시지에 포함할 수 없으므로 한정된 일정 거리에 있는 패킷의 해시값을 포함하는데 이러한 일정 거리의 패킷이 네트워크 상에서 모두 손실되거나 악의적인 중간 노드가 고의적으로 삭제하면 해시 체인이 끊어지는 문제점이 있으며 전송 과부하가 크다.[1,7]

Wong과 Lam에 의해 제안된 Tree 체인 방법은 데이터 패킷 스트림을 블록으로 나누고 이 블록을 하나의 이진 트리로 구성하여 각 최하위 노드들은 각 패킷의 해시값이 된다. 그 하위 노드들의 상위 노드는 두 하위 노드값들의 다시 해시값으로 구성된다. 이렇게 반복된 계산으로 최상위 노드는 모든 블록 패킷들의 특성을 갖는 해시값이 되고 이 값이 서명되어 수신자에게 메시지와 함께 전송된다. 전송 시에는 전송되는 패킷의 해시값부터 최상위 노드까지 경로의 이웃 노드값이 포함되어야 수신자들이 서명에 들어간 최상위 노드의 해시를 계산할 수 있다. 이 메커니즘은 전송 중에 패킷의 손실에 매우 강한 특성을 가지지만, 패킷에 전송해야 하는 추가적인 데이터가 많아 전송 과부하가 매우 크고 계산할 메시지가 미리 존재하여야 하므로 실시간 데이터에 대한 인증이 어렵다.[1,8]

Jung min Park에 의해 제안된 SAIDA (Signature Amortization using Information Dispersal Algorithm)는 패킷 인증을 위해 서명과 해시를 사용한다.[9] 그러나 기존의 이러한 인증 방법들과 마찬가지로 패킷 손실의 문제를 가지고 있는데, 이를 극복하기 위해 인증 정보를 여러 패킷에 분산시키는 IDA를 사용한다. 송신 측에서는 우선 하나의 서명으로 계산될 n개 패킷의 블록을 구성하고 각 패킷의 해시값을 계산한다. 이후에 이 해시값들은 IDA를 이용하여 계산되는데 여기에 사용되는 파라미터 m은 n개의 패킷중 m개 이상 받으면 원래의 해시값을 계산할 수 있는 값을 의미한다. 서명은 해시값과 마찬가지로 IDA를 이용하여 n개의 조각으로 나뉘어 지고 이러한 IDA를 통해 계산된 해시값과 서명은 각 패킷에 붙여 전송된다. 수신된 측에서는 수신된 m개의 패킷만을 이용하여 해시값과 서명을 계산할 수 있다. 그러나 이러한 방법은 임의로 삽입된 공격 패킷이나 변조된 패킷에 대해 이를 찾고 검증할 수 있는 방법을 가지고 있지 않기 때문에 계산된 해시값과 서명에 다시 해시와 오류 보정 방법 (Error Correcting Code)을 사용함으로써 변조에 의한 패킷 해시값과 서명의 잘못된 복원을 방지하고 있다. 그러나 SAIDA는 블록 단위로 송신측에서 연산을 수행하여 전송하고 수신 측에서 블록의 대부분 패킷을 수신한 후에 계산이 가능하므로 송수신측에서 모두 지연시간이 발생한다. 또한 처음 설정된 m개 이상의 패킷이 중간에서 손실되거나 공격자에 의해 오류 보정 코드로 복구해 낼 수 없는 정도의 변조가 이루어지면 블록 전체가 버려져야 하므로, 멀티캐스트 네트워크의 주요 응용인 멀티미디어 트래픽의 경우, 블록 전체의 손실에 의한 충격이 크다. SAIDA

는 앞에서 언급한 다른 해시 기반 인증 방법들에 비하여 패킷당 과부하가 적으나 여전히 3개 정도의 해시 블록 크기를 요구한다. IDA와 오류 보정 방법을 이중으로 적용하고 있기 때문에 n = 100, m = 70 인 경우, 패킷당 과부하가 해시 블록 크기의 약 3배 가까이 발생한다. 물론 송신자가 m을 상황에 따라 변화하여 보낼 수 있겠지만 수신자들의 패킷 인증 비율 정보를 모아 그 것에 맞추어 m을 변경하여 보내는 것은 멀티캐스트 환경에서 어려운 일이다.

지금까지 살펴본 해시 기반의 메시지 인증 방법들은 네트워크의 에러에 의한 자연적인 손실을 극복할 수 있는 방법으로 패킷에 인증 정보를 추가함으로써 패킷의 확장에 중점을 두고 있다. 그러나 문제는 네트워크의 자연적인 패킷 손실뿐만 아니라 악의적인 공격에 의한 대량의 패킷 손실에 대한 방어가 불가능하다는 것이다. 또한 한번 해시 체인이나 해시 블록이 손상되면 이후 체인 혹은 블록의 나머지 부분 전체가 신뢰될 수 없다. 멀티캐스트에 응용되는 트래픽은 주로 짧은 지연 시간을 요구하는 멀티미디어 정보인데 실시간성을 고려하고 있지 않아 송신 측에서 많은 전송 데이터를 모아 계산 후에 전송해야 하거나, 수신 측에서는 수신 패킷의 버퍼링을 통해 블록의 마지막 패킷이 도착하여야 인증이 가능한 방법들이다. 특히, 적은 시스템 자원을 가지고 있는 모바일 수신자들은 이러한 지연과 전송 및 계산의 과부하를 통해, 인증 처리를 실시간 응용에서 요구하는 시간 내에 처리하기 어렵다. 따라서 Ⅲ장에서는 제안하는 방법을 통해 이러한 문제점들을 해결하는 인증 방법을 보고자 한다.

Ⅲ. 복구 계층을 이용한 패킷 인증

이 장에서 우리는 기본 인증 모델과 복구 계층 모델 그리고 복구 계층이 적용된 인증 모델을 설명할 것이다. 제안하는 메커니즘은 해시 체인과 서명을 기반으로 패킷에 대한 인증을 수행하는데, 여기서 문제가 되었던 패킷 손실에 의한 해시 체인의 손실을 해결하기 위한 방법을 제안한다. 또한 대량의 연속적인 패킷 손실에 대해서도 해시 체인이 유효하게 하는 방법을 제안한다. 제안하는 방법의 기본적인 아이디어는 송신자가 해시 체인과 XOR 기반의 복구 계층을 구성하고 복구 계층의 정보를 서명에 포함하여 보내는 것이다. 만일 전송 중에 몇 개의 패킷이 손실되면, 복구 계층을 이용하여 손실된 해시 체인을 복구

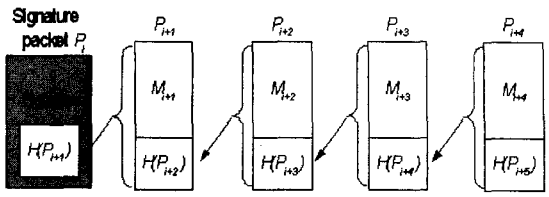


그림 1. 기본 패킷 인증 모델

하는 것이다. 또한 대량의 연속적인 패킷이 손실되었다면, 손실이후에 도착하는 패킷들을 해시 체인의 서명을 통해 인증 받는 것이다.

1. 기본 인증 모델과 복구 계층

패킷 인증의 기본적인 메커니즘은 해시 체인과 체인의 서명으로 구성되며, 서명을 통해 송신자와 해시 체인의 연관성을 확보하는 방법이다. 그림 1은 해시 체인과 서명을 이용한 기본 인증 모델을 보여 주고 있다. 각 패킷은 다음 패킷의 해시값을 포함하며 첫 번째 패킷은 서명이 첨부된다. 여기에서 서명을 앞에 두어 수신측에서 세션 연결 초기화 시에 서명을 확인하고 이 후에는 패킷의 해시값만을 확인하여 패킷의 데이터가 바로 응용 프로그램에서 사용될 수 있게 하여, 데이터의 수신측 실시간성을 고려하였다. 또한 해시 체인에 대해 서명을 하기 때문에 인증뿐만 아니라 패킷들의 송신자를 알 수 있다. 그러나 이러한 방법은 패킷의 손실에 약하여 하나의 패킷이 손실되면 해시 체인이 끊어지게 되는데, 이 것을 극복하기 위해 제안하는 메커니즘에서는 복구 계층(Recovery layer)을 두어 패킷이 손실되어도 이웃의 해시값과 복구값에 의해 손실된 패킷의 해시값이 복구되도록 하는 방법을 이용하고 있다. 제안하고 있는 메커니즘에서 기본적인 복구 원리는 여러 개의 데이터를 XOR 연산하면 그 중에 하나의 값이 없어도 나머지 존재하는 값과 연산된 값을 이용하여 복구가 가능하다는 것이다. 예를 들어, 그림 2와 같이 V_1, V_2, V_3, V_4 등 4개의 해시 데이터가 있고, 이를 모두 XOR 연산하여 나온 값인 복구값(recovery value)을 가지고 있는 경우, 4개의 데이터 중에 하나가 손실되어도 3개의 데이터와 XOR 연산한 복구값을 가지고 손실된 데이터를 복구할 수 있다는 것이다. 그림 2의 (a)에서는 V_1부터 V_4까지의 값을 XOR 연산하여 다음과 같이 복구값을 생성하였다. $Recovery\ value = V_1 \oplus V_2 \oplus V_3 \oplus V_4$. 그림

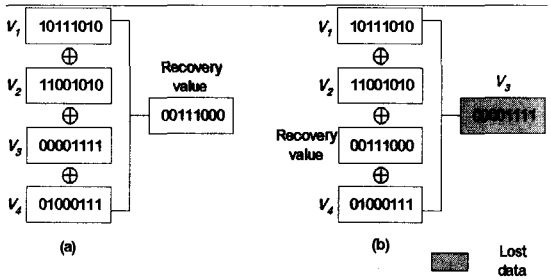


그림 2. XOR를 이용한 복구 모델

2의 (b)에서 전송 채널을 통해 V_3값이 손실되었다면, 복구값과 V_1, V_2, V_4를 가지고 V_3을 다음과 같이 계산하여 복구할 수 있다. $V_3 = V_1 \oplus V_2 \oplus V_4 \oplus Recovery\ value$. 이러한 복구값을 이용하여 해시 체인에 복구 계층을 구성하고 이를 패킷에 추가함으로써 패킷 손실이 발생 시에 손실된 해시값을 복구하여, 이후 패킷의 인증 유효를 증명함으로써 해시 체인이 깨어지지 않도록 하는 것이다. 복구 계층을 구성 시에 복구값은 복구값과 같은 인덱스를 가지는 해시값이 포함되면 안된다. 즉, 복구값은 자신이 아닌 다른 패킷에 의해 전달되는 해시값들로 구성되어야 한다. 이는 패킷이 손실 시에 복구값과 복구값의 계산에 사용된 해시값이 동시에 손실되면 복구가 불가능하기 때문이다. IDA(Information Dispersal Algorithm)기반의 SAIDA는 블록의 대부분의 패킷을 기다려야 하지만 [9], 이러한 방법을 이용하면 수신 측에서 이전에 도착한 해시 정보를 이용하여 도착한 패킷을 즉시 인증 확인함으로써 수신측 지연 시간과 버퍼링을 없앨 수 있다. 다음 절에서는 앞에서 언급한 기본 인증 모델에 복구 계층을 추가하여 복구 계층이 실제 적용된 인증 모델을 설명하도록 하겠다.

2. 복구 계층을 적용한 인증 모델

제안하는 인증 방법의 주요 목적은 인증 메시지의 즉각적인 확인을 통해 수신측의 버퍼링과 지연을 줄이고 DoS(Denial of Service) 공격에 대한 강점을 가지게 하며, 인증 메시지의 크기를 줄이는 것이다. 그림 3은 패킷 인증 모델을 설명하기 위해 10개의 패킷을 하나의 블록으로 구성하여 보여주고 있다. 물론 실제 적용되는 블록의 크기는 더 크게 적용되어 100개 이상이 될 것이다. 우선, 각 패킷의 해시값을 이전 패킷에 붙여 해시 체인을 구성하고 패킷 손실에 의한 체인의 손실을 막기 위해 복구 계층(Recovery layer)을 구성하며, 복구 계층의 변조를 막기 위해 $RICV$(Recovery layer Integrity Check Value)을 포함한다. 이 복구 계층은 현재 패킷의 위치를 기준으로 이후에 위치하는 패킷 해시값들의 XOR 연산으로 계산된다. 전송 측에서 인증 메시지를 구성하

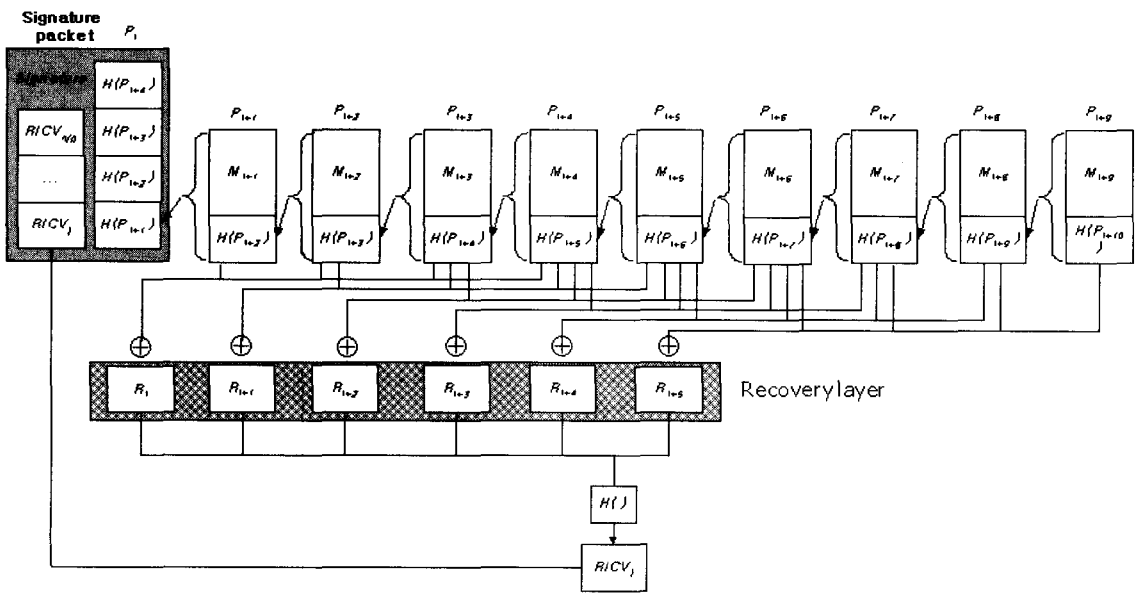


그림 3. 복구계층을 이용한 패킷 인증 모델

는 방법을 먼저 보도록 하겠다.

1. 송신자는 하나의 블록을 구성하는 n개의 메시지를 이용하여 해시 체인을 구성하고 처음 패킷에 서명한다.
2. 해시 체인의 복구 계층을 구성하기 위해 XOR 연산을 이용하여 계산하는데, 하나의 복구값 계산을 위해 사용하는 해시의 수를 r로 표현하고, r = 4 인 경우, 각 패킷의 복구값 R_i는 다음과 같이 계산된다. n은 하나의 서명으로 연결되는 해시 체인의 패킷수이고 $H()$은 해시 함수이며, P_i는 인덱스 i를 가지는 패킷이다.

$$R_i = H(P_{i+2}) \oplus H(P_{i+3}) \oplus H(P_{i+4}) \oplus H(P_{i+5}),\ i=1, \ldots, n.$$

R_i는 패킷 P_i에 첨부되어 전송된다. 예를 들어, 복구값의 인덱스가 2인 경우, 복구값 (Recovery value) R_2는 다음과 같이 계산되며 P_2에 첨부되어 전송된다. $R_2 = H(P_4) \oplus H(P_5) \oplus H(P_6) \oplus H(P_7)$. 이러한 방법으로 복구값 R_i의 i가 1부터 n까지의 모든 복구값을 계산한다. 이렇게 계산된 복구계층은 손실된 패킷의 해시를 복구하는데 사용된다. 따라서 패킷 P_{i+1}은 메시지 M_{i+1}과 $H(P_{i+2})$, R_{i+1}을 포함하게 된다.
3. 블록의 앞부분에 위치하는 r개의 연속적인 패킷에 손실이 발생하면, 손실 이후에 도착한 패킷을 인증하여야 한다. 그러나 손실로 인해 복구할 해시값이 없기 때문에 복구 계산에 필요한 (r - 1)개의 패킷 해시값들을 서명에 포함하고 이를 전송해야 한다. 그림 3에서는 r = 4인 경우, (r-1)개인 3개의 해시값 $H(P_{i+2})$, $H(P_{i+3})$, $H(P_{i+4})$를 포함하고 있다. 이를 통해 초기 패킷들의 손실에 대한 복구 계산이 가능하게 한다.
4. 우리는 인증 방법의 분석을 통해 특정 패킷이나 연속적인 패킷의 메시지 변조 혹은 삭제 등의 적극적인 방법을 이용하는 공격자를 고려하여야 한다. 따라서 복구 계층의 변조를 막기 위해 제안하는 인증 방법에서는 $RICV$(Recovery layer Integrity Check Value)을 두어 복구 계층의 값들이 변조되었는지를 확인할 수 있다. 인덱스 j를 갖는 $RICV_j$는 다음과 같이 계산된다.

$$RICV_j = H(R_{(j-1)*p+1} \| R_{(j-1)*p+2} \| R_{(j-1)*p+3} \| \ldots \| R_{(j-1)*p+p}),\ j=1, \cdots, n/p.$$

p는 RICV의 계산에 사용되는 복구값의 수를 의미하며 그림 3에서 p = 6이며 $\|$은 메시지의 연결을 의미한다. 이렇게 계산된 $RICV$들은 복구 계층의 무결성 확인을 위해 모두 서명 패킷에 포함되어 전송된다. 서명 패킷은 전송의 안정성을 위해 반복 전송될 수 있다.

다음은 수신 측에서 인증 메시지를 확인하는 절차를 설명하도록 하겠다.

1. 패킷의 손실이 없는 경우, 수신자는 해시 체인과 서명을 통해 패킷을 인증할 수 있다.
2. 패킷의 손실이 발생한 경우, 수신 측에서는 메시지를 수신 후에 손실된 해시를 복구하기 위해 이미 도착한 복구값과 해시값들을 이용하여 손실된 해시를 복구할 수 있다. 만일 P_{i+5}가 전송 중에 손실되었다면, 수신자는 $H(P_{i+6})$을 복구하여야 하는데 복구값과 이전 혹은 이후 패킷들의 해시값들을 이용하여 다음과 같이 손실된 해시를 계산한다.

$$H(P_{i+6}) = R_{i+1} \oplus H(P_{i+3}) \oplus H(P_{i+4}) \oplus H(P_{i+5}).$$
$$H(P_{i+6}) = R_{i+2} \oplus H(P_{i+4}) \oplus H(P_{i+5}) \oplus H(P_{i+7}).$$
$$H(P_{i+6}) = R_{i+3} \oplus H(P_{i+5}) \oplus H(P_{i+7}) \oplus H(P_{i+8}).$$
$$H(P_{i+6}) = R_{i+4} \oplus H(P_{i+7}) \oplus H(P_{i+8}) \oplus H(P_{i+9}).$$

r = 4 인 경우, 위와 같이 r 개만큼의 복구할 수 있는 여러 방법이 존재하므로 손실 없이 도착한 패킷들 중에 해당하는 정보를 찾아 그에 해당하는 복구 계산 방법을 사용하면 된다. 연속적인 r 개의 손실이 발생하여도 앞에서부터 순서대로 복구하면 r 개의 손실된 해시를 복구할 수 있다. 서명 이후의 패킷이 연속해서 손실되면 복구값과 패킷에 의해 전달된 이전의 해시값이 존재하지 않으므로 서명에서 포함하였던 블록의 앞부분 해시값들을 가지고 블록 초기 손실을 복구한다.
3. 그러나 능동적인 공격자에 의해 패킷 데이터와 해시 체인 그리고 복구 계층이 변조되고 고의

적인 손실이 발생할 수 있으므로, 수신자는 이를 확인하기 위해 다음과 같이 수신한 복구값을 이용하여 $RICV$를 계산한다. $RICV_j = H(R_j \| R_{j+1} \| R_{j+2} \| R_{j+3} \| R_{j+4} \| R_{j+5})$. 계산된 $RICV_j$는 서명의 $RICV_j$와 비교하여 무결성을 검증한다.

4. r 개 이상의 연속적인 패킷의 손실이 발생한 경우, 수신자는 모든 해시 체인을 복구할 수 없다. 그러나 제안하는 인증 방법에서는 끊어진 해시 체인을 인증하기 위해서 서명의 $RICV$값들을 이용한다. 끊어진 해시 체인의 $RICV$를 계산하여 서명의 $RICV$와 비교함으로써 대량의 패킷 손실 이후에 복구되지 못한 해시 체인을 서명을 통해 인증하는 것이다.

제안하는 인증 방법에서 하위 그룹으로 분산된 $RICV$는 대량의 패킷 손실에 강한 특성을 가지는데, 초기 전송시에 설정된 r개 이상의 대량 패킷 손실이 발생하면 해시 체인이 끊어지게 되고, 이때 여러 개의 하위 그룹으로 나누어진 $RICV$ 중에서 손실된 이후에 도착하는 패킷들의 $RICV$를 서명의 $RICV$와 비교하여 복구값과 해시 체인의 무결성을 확인하고 끊어진 해시 체인이 아직도 유효하다는 것을 입증한다. 따라서 패킷의 대량 손실과 같이 공격자의 능동적인 공격에도 안전성을 가지게 한다. 이러한 제안된 인증 방법의 특성은 기존의 방법들이 특정 개수의 패킷이 손실되어 인증에 실패하면 나머지 모든 패킷이 인증에 실패할 수밖에 없는 한계를 극복할 수 있다. 또한 $RICV$는 전송된 데이터 해시값의 XOR 연산된 복구값의 해시값이고 이것이 서명에 포함되므로 부인방지 기능을 가지게 된다.

3. 운영 파라미터

이 절에서는 제안된 메커니즘을 분석하여 여기에 사용되는 파라미터가 메커니즘의 성능에 어떠한 영향을 미치는지 알아보겠다. 우선 사용된 파라미터들인 각 블록의 크기가 메커니즘의 운영에서 어떠한 영향을 미치는지를 알아보기 위해 하나의 서명으로 구성되는 서명 블록 크기 즉, 해시 체인을 구성하는 패킷의 수를 나타내는 n를 보겠다. n은 하나의 서명과 해시 체인으로 구성되고 서명이 앞에 위치하기 때문에 이를 생성하기 위해서는 이미 n개의 메시지가 송신 측에 있어야 하며 이 패킷들을 이용하여 해시 체인을 구성한다. 그리고 블록의 제일 처음 보내지는 패킷의 해시를 포함하여 서명을 생성하므로 이것은 서명의 계산 시간을 고려해야 한다. 서명은 계산에 많은 시간이 소요되므로 n의 크기에 따라 필요한 서명의 개수가 달라지는데, n이 클수록 하나의 서명에 많은 패킷이 포함되므로 전체 스트림 전송에서 서명의 개수가 적어져 서명의 생성 및 확인 시간이 줄지만 송신측은 많은 버퍼를 준비해야 한다. 반면에 n이 적으면 송신 측에서는 적은 패킷을 가지고 계산하지만 좀 더 많은 서명을 생성하여야 한다.

r은 복구 계층을 구성하는 해시 체인의 블록 크기로서 몇 개의 해시로 복구값이 구성되는지를 의미한다. 이 것의 크기는 복구값이 얼마나 많은 패킷의 특성을 포함하는지를 의미하므로 클수록 악의적인 공격자의 연속적인 패킷 손실 공격에 강하다. 예를 들어, r개의 연속적인 패킷이 손실되면 r개의 해시값 복구가 가능해 해시 체인의 복구가 가능하지만 $(r+1)$개 이상의 패킷이 손실되면 복구값과 복구값의 계산에 사용된 해시값이 동시에 손실되기 때문에 해시의 복구가 불가능하다. 따라서 r은 크면 좋지만, 너무 크면 r이 n에 가까워져 대부분의 패킷 해시값을 서명에 포함하여야 하므로 이 메커니즘이 적용되는 응용의 특성에 따라 달리 적용할 수 있을 것이다. 그러나 특정 부분의 복구가 불가능하다고 하여 손실 이후의 해시 체인 자체가 버려지는 것은 아니며 서명에 포함된 $RICV$에 의해 손실 이후의 해시와 복구값의 무결성을 확인할 수 있기 때문에 해시 인증 체인은 유효하다. p는 $RICV$을 생성하기 위한 복구 계층의 블록 크기로서 작을수록 복구값의 무결성을 빠르게 확인할 수 있지만 서명에 많은 $RICV$가 보내져야 한다. 반면에 클수록 서명 패킷에 전달되는 $RICV$의 수가 줄어들지만 무결성을 확인하기 위해 수신 측에서 많은 패킷의 복구값을 모아야 한다.

Ⅳ. 성능 분석

1. 과부하 비교

제안하는 인증 방법의 상대적인 성능을 알아보기 위해 다른 인증 방법들과 비교하여 여러 가지 특징들과 과부하를 알아보도록 하겠다. 우리는 제안하는 인증 방법을 tree chaining, EMSS, SAIDA 등과 비교할 것이다. 표 1은 각 인증 방법들의 송수신측 지연과 송신측의 계산 과부하, 인증 메시지 확인 비

용, 전송 과부하 등의 비교를 나타내고 있다. 이 비교에서 MAC 기반의 클럭 동기화를 요구하는 TESLA와 여러 사용자들의 공모에 의한 약점을 가지는 Efficient MAC은 제외하였다.

하나의 송신자가 전송하고 다수의 수신자가 메시지를 받는 많은 응용들에서 송신 측의 계산적 과부하는 중요한 문제가 되지 못한다. 일반적으로 서비스를 위한 송신 시스템은 이러한 특성을 고려하여 특별히 설계된 시스템으로 이러한 문제를 극복할 수 있다. 그러나 다수의 수신 시스템은 일반적인 목적의 시스템이기 때문에 많은 인증 메시지의 처리에 의해 응용 프로그램에서 요구하는 사항을 만족하기 어려울 것이다. 특히, 실시간 응용들은 전송 및 메시지의 인증에 더욱 더 엄격한 지연 시간과 빠른 처리를 요구한다. 더욱이 수신 시스템이 모바일 환경의 시스템이라면 자원은 더욱 제한되어 응용 서비스에 많은 부담이 될 것이다. 따라서 제안하는 인증 방법에서는 수신측 버퍼 사이즈를 줄이고 인증 확인의 계산적 부담을 줄일 수 있도록 설계하였다. 표 1에서 보여주고 있듯이, EMSS는 수신측에서 n개의 버퍼를 요구하며 이것은 n개의 패킷 지연을 유발한다. SAIDA 역시 수신측에서 유사한 버퍼링과 지연 시간을 요구한다. 그러나 제안하는 인증 방법과 tree chaining 방법은 수신한 패킷을 바로 인증할 수 있다. 수신 측의 인증 메시지 확인 비용은 패킷 손실이 발생하지 않을 때와 발생할 때로 분류하였는데, 손실이 발생하지 않았을 때의 인증 메시지 확인 비용은 제안하는 방법과 EMSS가 $n+1$개의 해시 작업과 하나의 서명 확인이 필요하다. 그러나 tree chaining 방법은 제안하는 방법보다 2배 정도의 해시 처리를 요구하며, SAIDA는 2배의 해시 처리와 추가적인 처리가 요구된다. 손실이 발생하였을 경우, 제안하는 방법은 패킷이 손실된 비율에 따라 선형적인 처리의 증가가 발생한다. 모든 인증 방법에서 송신측의 계산적 과부하는 손실을 가지는 인증 확인 비용과 유사하다. Tree chaining 인증 방법을 제외한 다른 인증 방법들의 인증 비율은 파라미터의 선택에 따라 다양한 결과를 가져오므로, 다음 절에서 시뮬레이션을 통한 그래프를 이용하여 비교할 것이다. 멀티미디어 트래픽은 주로 UDP에 의해 전송되므로 전송 중에 손실되기 쉽고, 따라서 멀티미디어 응용들은 패킷을 작은 크기로 만든다. 그러나 멀티미디어 패킷이 인증 메시지를 포함하여야 한다면, 인증 메시지는 대단히 부담스러운 크기일 것이다. 각 인증 방법별로 다양한 파라미터가 사용되고 있으며, 이러한 파라미터들은 성능과 패킷 과부하에 많은 영향을 주기 때문에 표 1에서는 수식으로 표현하였다. 물론 각 인증 방법들에서 많은 데이터를 추가하여 패킷 인증을 수행하면 모든 패킷을 인증할 수 있지만 실제적으로 전송 성능이 떨어져 사용 불가능한 방법이 될 것이다. 표 1의 전송 과부하를 보면, 본 논문에서 제안한 복구 계층을 이용한 패킷 인증 방법은 대략 2개의 해시 블록이므로 크지 않다. 만일 16 bytes의 해시 함수와 128 bytes의 서명을 사용한다면, 서명에 포함된 *RICV*들을 전송

표 1. 인증 방법들의 과부하 비교(n은 한 블록을 구성하는 패킷의 수, m은 IDA 디코딩을 위해 필요한 최소한의 패킷 수, IDA(Information Dispersal Algorithm), ECC(Error Correcting Code), XORs는 xor 연산, u는 손실된 패킷의 수, s는 서명의 크기, h는 해시의 크기, q는 EMSS에서 패킷에 첨부되는 해시의 수, 계산적 과부하와 인증 확인 비용은 순서대로 해시와 서명의 처리 수이다.)

		Tree chaining	EMSS	SAIDA	Proposed scheme
Sender delay		n	1	n	n
Receiver delay		1	n	m	1
Computation overhead on sender [extra processes]		$2n$-1, 1	n+1, 1	$2n$, 1 [IDA, ECC]	n+1+n/p, 1 [r*n XORs]
Verification cost [extra processes]	lossless	$2n$-1, 1	n+1, 1	$2n$, 1 [IDA, ECC]	n+1, 1
	loosy	$2n$-1, 1	n+1, 1	$2n$, 1 [IDA, ECC]	n+1+n/p, 1 [r*u XORs]
Verification rate		1.0	variable	variable	variable
Communication overhead (bytes)		$(\log_2 n+1)*h+s$	$q*h$	size of (IDA+ECC)	$(2+(1/p))*h$

과부하에 포함한다 하더라도 제안하는 방법의 전송 과부하는 약 35 bytes가 될 것이다.(p가 10보다 큰 경우) EMSS는 논문에서 권장하고 있는 해시의 중첩이 6개이지만 3개만 적용해도 48 bytes의 전송과부하가 발생하며, tree chaining 방법은 $log_2\ n$ 개의 해시 블록과 서명이 추가되어야 하므로 n이 8인 경우, 176 bytes의 전송 과부하로 상대적으로 많은 증가량을 보인다. SAIDA의 경우는 대략 30%의 부가 정보를 담는다 하여도 42 bytes의 전송 과부하가 발생한다. SAIDA는 패킷 과부하와 인증 확률의 협상이 될 수 있다. m이 작으면 패킷 과부하가 줄어들지만 인증 확률이 떨어지고, 크면 인증 확률이 높지만 패킷 과부하가 많이 발생한다. 이러한 특성 때문에 전송 과부하와 인증 비율은 시뮬레이션을 통해 그 결과를 다음 절에서 보여주고 있다.

2. 인증 비율

우리는 인증 성능을 비교하기 위해 EMSS와 SAIDA 그리고 제안하는 인증 방법을 시뮬레이션 하였다. 시뮬레이션에서 1000개의 패킷 블록을 가지는 40000 샘플을 이용하여 시뮬레이션을 수행하였다. 인증 성능은 전송 과부하에 의해 다르게 나올 수 있기 때문에, 각 인증 방법에 대하여 유사한 전송 과부하를 가질 수 있는 파라미터를 선택하였으며 이 파라미터들은 표 2에서 명시하고 있다.

그림 4는 패킷 손실 확률에 따른 각 인증 방법들의 인증 비율을 보여주고 있다. 별표는 제안하는 방법의 인증 비율을 나타내며 이 것은 손실 확률의 증가에 따라 천천히 감소함을 알 수 있다. 삼각 표시는 EMSS를 나타내며 손실 확률의 증가에 의해 연속적

표 2. 시뮬레이션 파라미터

	Parameters
General information	Block size n = 1000, Packet loss probability = {0.0, 0.01, 0.02, ..., 0.4}
EMSS	2 hashes, the length of edge = [1, 39]
SAIDA	The minimum number of need for decoding m = 700
Proposed scheme	The number of hash values used in a recovery computation r = 4, The number of recovery values for RICV computation p = 10

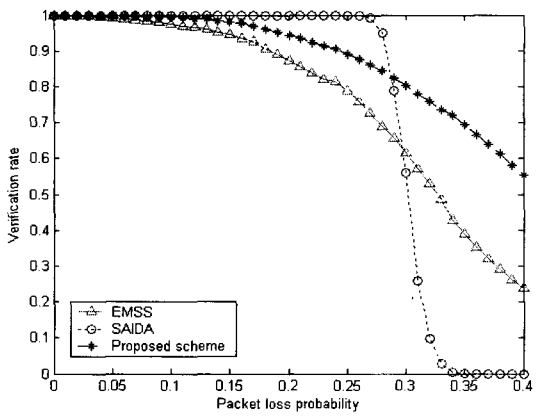


그림 4. 패킷 손실 확률에 따른 인증 비율

인 패킷의 손실이 증가함에 따라 제안하는 방법에 비해 빠르게 인증 비율이 떨어지는 것을 볼 수 있다. SAIDA의 인증 비율은 원으로 나타내고 있으며 패킷 손실 확률이 0.3인 지점에서 빠르게 떨어지는 것을 볼 수 있다. 이 것은 IDA 디코딩을 위해 필요로 하는 패킷의 수, m을 700으로 설정하였기 때문이다. SAIDA에서 m은 송신자에 의해 바뀔 수 있지만 송신자가 수신자들의 인증 비율을 모두 모아 그 정보에 맞추어 m을 설정하여 보내는 것은 멀티캐스트 환경에서 어려운 일이다. 따라서 유사한 전송 과부하를 가지는 조건에서 제안하는 인증 방법의 인증 비율이 가장 높은 것을 알 수 있다.

V. 결 론

본 논문에서는 멀티캐스트 보안 문제에 대하여 언급하고, 패킷 인증에 대한 문제를 제기하였으며 특히, 수신 측의 과부하 문제를 해결하는 인증 방법을 제안하였다. 제안한 인증 방법에서는 복구 계층을 이용하여 해시 체인을 강화하였고 복구 계층의 변조를 막기 위해 *RICV*를 이용한 무결성 확인을 서명을 통해 하도록 하였다. 복구 계층은 여러 개의 해시 체인 값을 사용하여 XOR 연산을 이용하므로 연산에 적은 시간과 자원이 소모된다. 또한 공격자에 의해 패킷이 대량 손실되어도 분산된 *RICV*를 통해 손실된 패킷 이후에 도착하는 패킷의 해시 체인을 인증할 수 있다. 성능 면에서 이 패킷 인증 방법은 거의 2개 정도의 해시 블록이 패킷마다 추가되기 때문에 다른 인증 방법에 비해 상대적으로 적은 데이터 확장을 보인다. 또한 수신 측에서의 인증 확인을 위한 버퍼링

과 지연이 발생하지 않고, 인증 확인 비용도 적다. 제안된 이 패킷 인증 방법은 송신 측에서 많은 데이터를 보유한 후에 해시 체인 및 복구 계층 구성 등 연산을 시작해야 하기 때문에 전화와 같이 송신 측에서 실시간으로 발생하는 스트림을 지연 시간 없이 실시간으로 전송하는 것은 어렵다. 하지만 영화나 음악 등과 같이 이미 송신 측에 존재하는 스트림이나 수초 이내의 지연 시간을 허용하는 방송에 적용하면, 수신 측에서 매 패킷이 수신 후에 즉각 인증을 확인하고 이를 응용 프로그램에 넘겨 실행할 수 있기 때문에 적합하다. 대부분의 멀티미디어 컨텐츠가 영화나 멀티캐스트 방송 혹은 상품의 동영상 등 저장된 상태이므로 많은 부분에 제안된 패킷 인증 방법이 응용될 수 있을 것이다. 앞으로 진행될 연구에서는 해시 체인의 강화와 복구 계층의 안전성을 강화하는 연구가 가능할 것이다.

참 고 문 헌

[1] Judge P., Ammar M., "Security issues and solutions in multicast content distribution : a survey," *IEEE Network*, Volume 17, Issue 1, pp. 30-36, Jan.-Feb. 2003.

[2] Mohamed Al-Ibrahim, Josef Pieprzyk, "Authenticating Multicast Streams in Lossy Channels Using Threshold Techniques," *ICN 2001, LNCS 2094*, pp. 239 - 249, 2001.

[3] Adrian Perrig, Ran Canetti, J. D. Tygar, Dawn Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," *Proc. of IEEE Security and Privacy Symposium S&P2000*, May 2000.

[4] Perrig, Canetti, Song, Tygar, Briscoe, "TESLA : Multicast Source Authentication Transform Introduction," *IETF MSEC WG draft-ietf-msec-tesla-intro-00.txt*, August 2002.

[5] Chris Karlof, "http://www.cs.berkeley.edu/~ckarlof/research/multicast-security/related .html"

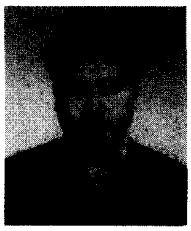
[6] R. Canetti et al., "Multicast Security : A Taxonomy and Efficient Constructions," *IEEE INFOCOM*, New York, Mar. 1999.

[7] R. Gennaro, P. Rohatgi, "How to Sign Digital Streams," *LNCS, vol. 1294*, 1997.

[8] C. Wong and S. Lam, "Digital Signatures for Flows and Multicasts," *IEEE /ACM Trans. Net.*, vol. 7, 1999.

[9] Jung Min Park, Edwin K. P. Chong, and Howard Jay Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Transactions on Information and System Security*, Vol. 6, No. 2, May 2003, pp. 258-285.

〈著者紹介〉



홍 기 훈 (Kihun Hong) 학생회원
2000년 2월 : 숭실대학교 정보통신공학과 공학사
2002년 2월 : 숭실대학교 정보통신공학과 공학석사
2002년 3월~현재 : 숭실대학교 정보통신공학과 박사과정
〈관심분야〉 모바일 보안, 멀티캐스트 보안, IPsec.

정 수 환 (Souhwan Jung) 정회원
1985년 2월 : 서울대학교 전자공학과 졸업
1987년 2월 : 서울대학교 전자공학과 석사
1988년~1991년 : 한국통신 전임연구원
1996년 : 미 워싱턴 주립대(시애틀) 박사
1996년~1997년 : Stellar One SW Engineer
1997년~현재 : 숭실대학교 정보통신전자공학부 부교수
〈관심분야〉 모바일 인터넷 보안, NEMO Security, Security Protocol.