IPv4/IPv6 헤더변환 방식에서의 취약성 개선에 관한 연구

황 호 준* · 유 승 재** · 김 귀 남*

요 약

1980년대에 컴퓨터 통신의 표준으로 널리 사용되었던 IPv4는 이제 한계점에 도달하였다고 볼 수 있다. 이런 한계점을 해결하기 위해서 새로운 버전의 IPv6(IP Version 6)의 도입을 준비하고 있는 실정이다. 그러나 IPv6의 제일 큰 문제는 IPv4와의 호환이 되지 않는다는 점이다. 이런 상황을 해결하기 위해서 일반적으로 Dual Stack 방식, Tunneling 방식, Header변환 방식이 제안되었다. 그 중 Header 변환 방식은 IPv4 네트워크 망과 IPv6의 네트워크 망을 변환기를 통해서 IP Header를 버전 4에서 버전 6으로 변환 시켜주는 방식으로 구현이 용이하고 구현 절차가 간단하다는 장점이 있지만 기존의 IPv4에서 가지고 있는 취약성을 내포할 수 있다 하겠다. 본 논문에서는 제안된 3가지 변환 방법 중에서 Header변환 방식에 대한 문제점을 해결하고자 시도되었다. Header 변환 방식의 취약성인 종단간 보안의 취약함을 개선하기 위해서 IP Header의 필드 값들을 각각에 대응하는 필드 값으로 변환 시킨 후에 IPSec(IP Security)의 ESP(Encapsulation Security Payload)를 적용시켜 패킷 단위의 "암호화된 Header 변환 방식"을 제안하여 Header 변환 방식이 가진고 있던 기존의 취약성을 해결하였다.

Study on Improving Vulhearability in IPv4/IPv6 Header Translation Mechanism

Ho-Jun Hwang* · Seung-Jae Yoo** · Kuinam J. Kim*

ABSTRACT

The IPv4 that used to be generally used as a medium of computer communications in 1980s has reached its limits now. IPv6 (IP Version 6) is being prepared to solve the limitations of the IPv4. However, the biggest problem of IPv6 is that it is not compatible with the IPv4. To resolve the compatibility issue, Dual Stack, Tunneling and Header Converting methods have been proposed. The Header Converting method allows communications between the IPv4 and IPv6 networks with the converter. This method's strength is that it is easy to embody and the procedures for embodiment is simple. However, this method still contains the weaknesses that the existing IPv4 has. On the current document, the Header Converting method among the three methods is discussed to resolve the problems this method has. To solve the Header Converting method's weakness, the security problem between sections, the IP Header field values are converted to the relative field values and IPSec (IP Security) and ESP (Encapsulation Security Payload) are applied. The proposed "Encrypted Header Converting Method" that is encrypted in packet units has solved the weakness that the pre-existing Header Converting method used to have.

* 경기대학교 정보보호기술공학과
** 중부대학교 정보분석학과

1. 서 론

인터넷의 폭발적인 인기와 더불어 사용 가능한 IP 주소가 절대적으로 모자라는 문제를 해소하기 위해 기존의 32비트 주소 체계인 IPv4에서 128비트의 주소 체계인 IPv6가 제안 되었다. IETF는 1996년에 IPv6 규격을 제정함으로써 신규 주소할당 문제 뿐만 아니라, 기존의 IPv4의 한계를 해결하는 데에 새로운 전기를 마련한 것으로 평가 받고 있다[1].

IPv6가 가지고 있는 문제점은 IPv4와의 자연스러운 호환이 이루어지지 않는다는 점이다. 그렇다고 어느 시점을 정해놓고 전 세계적으로 IP 체계를 IPv4를 IPv6로 변환하지 않는 한 IPv4와 IPv6와의 혼합 사용은 상당 기간 불가피 할 것이다. 그래서 IPv4/IPv6의 호환을 위해 Dual stack 방식, Tunneling 방식, 그리고 Header 변한 방식이 제안되고 있다[2].

특히 Header 변환 방식은 IPv6 패킷 Header를 IPv4 패킷 Header로 변환하거나 반대로 IPv6 Header를 IPv4 Header로 변환시키는 방식이다. 이것은 변환 방식이 투명하고 구현이 쉽고 간단하나 종단간의 네트워크 계층 보안이 불가능하다는 단점을 가지고 있다[5].

본 논문은 Header 변환 방식의 단점인 종단간의 네트워크 계층의 보안을 개선하기 위해서 헤더 변환 방식에 IPSec의 ESP를 적용시킨 "암호화된 Header 변환 방식"을 제안 하고 있다, Header의 변환 규칙은 SIIT(Stateless IP/ICMP Translation)에서 제안하는 방법으로 IP Header 필드 값들 IP 버전에 맞게 필드 값들을 바꾸어 준 뒤, IPv4에서 IPv6로 변환시키는 경우는 작성된 IPv6 기본 헤드에 ESP 확장 Header를 추가적으로 부착하여 발송하고, IPv6에서 IPv4로 변환시키는 경우는 전달 받은 패킷에서 ESP 확장헤더를 적당한 처리를 한 뒤 SIIT 방법으로 IPv4 패킷을 만들어서 서로 다른 버전의 IP Header를 변환시키게 된다. 이렇게 함으로써 변환 과정에서도 패킷 단위의 암호화가 계속해서 이루어지기 때문에 종단간의 취약성을 개선할 수 있다.

2. 관련 연구

IPv6의 도입에 앞서서 먼저 논의 되어야 할 중요한 이슈들 중에 하나가 IPv4/IPv6 전환 메커니즘이다. IPv6 망과 외부의 다른 IPv6 망이나 IPv4 망과 통신을 위해서 상호공존 하는 망과 망 사이에서 통신이 자연스럽게 이루어지도록 하는 것이 IPv6 변환 메커니즘들 이다.

2.1 Dual Stack 방식

라우터나 호스트에서 사용될 수 있는 기본적인 전환 방법으로 IPv6 노드가 IPv4 전용 노드와 호환성을 유지하는 가장 쉬운 방법이다. IPv4/IPv6 듀얼스택(Dual Stack) 노드는 IPv4 패킷과 IPv6 패킷 모두를 주고 받을 수 있으며 IPv4 패킷을 사용하여 IPv4 노드와 직접 호환이 되고 또한 IPv6 패킷을 사용하여 IPv6 노드와 직접 호환이 가능하다[3].

IPv4/IPv6 듀얼스택 노드는 두 프로토콜을 모두 지원하기 때문에 IPv4 주소와 IPv6 주소로 모두 설정할 수 있다. 그리고 DHCP(Dynamic Host Configuration Protocol)를 사용하여 그 IPv4 주소를 얻고, 상태 비보존형 주소 자동설정을 사용하여 IPv6 전용 주소를 얻을 수 있다. 그리고 이러한 방식은 모든 호스트의 IP 계층의 프로토콜 스택에 대한 수정이 불가피하므로 비용이 많이 드는 단점이 있다[8].

2.2 Tunneling 방식

IPv4 네트워크 환경에서 IPv6 패킷을 IPv6로 통신이 가능한 호스트로 전송하기 위한 방법으로 IPv6 패킷을 IPv4의 패킷에 삽입해서 전달

하는 방법이다. 즉 IPv6 패킷을 IPv4 패킷에 캡슐화 해서 IPv4 네트워크를 통해 전달한다. 터널이 시작되는 노드에서는 IPv6 패킷에 IPv4 Header를 추가하는 것과 이 패킷을 전송하는 일을 수행하고 터널이 끝나는 노드에서는 이 패킷을 받아서 캡슐화 된것을 제거하고 분할(Fragment)이 일어난 경우는 패킷을 재조립한다.

터널링 방식은 설정 터널링(Configured Tunneling)과 자동 터널링(Automatic Tunneling)으로 구분되는데, 설정 터널링 방법은 두 라우터간 IPv4 주소를 통해 매뉴얼 하게 정적으로 터널을 설정하는 방식이고, 자동 터널링 방법은 IPv4-호환(IPv4-compatible) 주소를 이용하여 매뉴얼 한 설정 없이 IPv4 구간을 통과할 때면 IPv4 호환 주소가 내포되어 있는 IPv4 주소를 자동으로 터널링하는 방식이다[9]. 그러나 이것은 모든 IPv4 라우팅 하부구조상에 IPv6 패킷의 터널링 기법을 추가해야 하므로 구현이 쉽지 않고 동작 과정이 복잡하다는 단점이 있다[14].

2.3 Translation 방식

원래 패킷의 IP Header를 다른 버전의 IP Header로 교체하는 방식이다. 다시 말해서 IPv6 패킷 Header를 IPv4 패킷 Header로 변환하거나 반대로 IPv6 Header를 IPv4 Header로 변환시키고 필요하면 체크섬을 재계산하는 방식을 가리킨다. NAT-PT(Network Address Translation-Protocol Translation)는 IPv4/IPv6의 변환 규칙인 SIIT(Stateless IP/ICMP Translation)을 기반으로 하는 변환 방식으로 IP 계층에서 이루어지기 때문에 빠르다는 장점이 있는 반면 응용 프로토콜에 내장된 IP 계층 주소 변환에 어려움이 있다. 뿐만 아니라 종단간의 네트워크 보안이 불가능 하다[9].

Header변환 방식은 RFC 2765에서 언급한 SIIT에서 IPv4 프로토콜과 IPv6 프로토콜 간의 변환 규칙을 제공하고 있다. 즉 원래의 패킷의 IP Header를 다른 버전의 IP Header로 교체하는 것이다. IP Header 내의 주요 필드들에 대한 변환은 양 프로토콜의 대부분의 필드가 직접적으로 대응된다. Header내의 몇몇 필드들의 경우는 의미가 다르거나 크기가 다른 경우도 있으나 대부분은 유사하다. 이러한 변환 방식은 투명하고 변환절차가 간단하기 때문에 비교적 구현이 용이하다는 장점이 있지만 기존의 IPv4에서 가지고 있던 보안의 취약성을 내포할 가능성을 가지는 단점도 있다[8].

2.4 NAT-PT 변환 방식

Header변환 방식의 대표적인 방법중의 하나가 NAT-PT방식이다. 이것은 SIIT 프로토콜 변환과 NAT(Network Address Translation) 및 DNS ALG(Application Level Gateway) 등 적절한 ALG의 동적 주소 변환을 조합하여 IPv6 전용 노드와 IPv4 전용 노드 사이에서 상호 통신을 가능하도록 한다. NAP-PT 방식은 IPv6-IPv4 경계에서 IPv4 주소 풀을 사용하여 동적으로 IPv4 주소를 할당하도록 한다. NAT-PT는 IPv6 네트워크의 주소와 IPv4 네트워크 주소를 서로 바인딩(Binding) 하여 각 주소 영역 사이를 오가는 데이터그램에 투명한 라우팅을 제공한다.

다시 말하면 SIIT는 IPv4-only 노드와 IPv6-only 노드간의 통신을 위한 변환 방식만을 설명하고 IPv6 노드는 IPv4 노드와의 통신을 위해 IPv4 주소를 할당 받은 것을 가정할 뿐 이러한 주소를 할당 받는 과정은 설명하지 않고 있지만 NAT-PT 방식은 IPv4와 IPv6 경계에서 세션을 생성하는 동적 기초(Dynamic Basis)의 IPv6노드에게 할당 하기 위한 주소 풀(pool of IPv4) 주소를 사용한다. 여기에서 각 네트워크 상의 단말 노드들은 변경할 필요가 없으며 IP 패킷 라우팅은 단말 노드에 있어서 완전히 투명 하게 된다. 그러나 한 세션에서 들어오고 나가는 데이

터그램은 동일한 NAT-PT 라우터를 거쳐 야만 한다[7].

NAT-PT는 이름에서도 알 수 있는 것처럼 크게 두 가지 기능으로 수행한다. 첫 번째는, 세션이 초기화될 때마다 동적으로 IPv6 노드에 IPv4 주소를 할당하기 위한 주소 풀을 가지고 두 망간의 경계 라우터에 위치하여 주소변환을 수행하는 NAT 기능이다. 두 번째는 PT이며, 호스트에서 변환 기능을 정 의한 RFC 2765 표준문서 SIIT를 기반으로 주소 변환을 수행한다. 또한, 동적으로 주소를 할당하고 변환 하기 위해서는 페이로드(Payload) 영역에 IP 주소나 포트 정보를 포함한 응용들에 의해 추가 적인 요구사항이 발생하는데, 이를 지원하기 위한 ALG를 사용해야 한다.

2.5 Header 변한 방식의 한계점

- 종단간 보안 결여 : NAT-PT의 가장 중요한 제한 중의 하나가 종단간 네트워크 계 층의 보안이 불가능하다는 점이다. 또한 IP 주소를 응용 계층에서 사용하는 경우 수송계층 및 응용 계층 보안 역시 불가능하다. 이는 NAT 기능의 자체적인 한계 이다. 이것은 세션이 NAT-PT에 의해 중계되기 때문인데 상위계층의 보안도 보장되지 않는다.
- 토폴로지 제약 : 세션에 참여하고 있는 모든 요청과 반응는 동일한 NAT-PT 라우터를 거쳐서 전송되어져야 한다. 이것을 보장하는 한 가지 방법은 스텁-도메인(Stub-Domain)에서 유일한 경계 라우터에 NAT-PT를 구현 하는 것인데 이것도 앞서 설명한 바와 같이 약간의 문제점이 있다. 이러한 제한은 패킷 변환을 요구하지 않는 듀얼스택에서 생성되거나 받아들여지는 패킷에 대해서는 적용되지 않는다. 이것은 PREFIX::x.y.z.w 형태의 IPv6 주소에서 IPv4 주소가 구별될 수 있고 듀얼스택 라우터는 이러한 IPv4 노드와 듀얼스택 노드 사이에 상태정보(State Information) 조정 없이 패킷을 라우팅 할 수 있기 때문이다. 이것은 또한 IPv6 노드들 사이의 통신에는 영향을 주지 않는데 사실 가능한 통신방법이 없는 경우에만 변환을 사용하게 된다.
- 프로토콜 변환 제약 : IPv4의 여러 필드들은 IPv6에서는 의미가 바뀌는데 변환 시에 이러한 의미들은 정확하게 전달되지 않는다.
- 주소변환의 영향 : NAT-PT가 주소변환을 수행하기 때문에 상위 계층에서 IP 주소를 전달하는 애플리케이션(Application)들은 제대로 작동되지 않기 때문에 ALG 의 사용이 필수적이다.

DNS 변환 및 DNS-SEC : DNS 메시지의 변환구조는 Secure DNS와 조합되어 쓰일 수 없다. IPv6 도메인의 Authoritative DNS 네임 서버는 IPv4 도메인으로부터 생성된 질의들에 대해 응답를 직접 보낼 수 없기 때문에 직접 응답를 받기를 원하는 IPv4 노드는 NAT-PT에 의해서 보내어지는 응답를 거절한다[5, 6].

3. 암호화된 IPv4/IPv6 Header 변환 시스템 설계

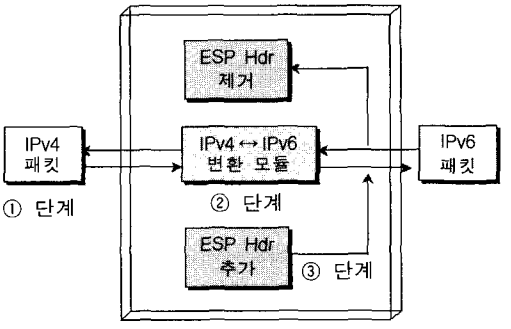
2장에서 설명한 Header 변환 방식은 IPv6에서 IPv4로 변환시의 취약성을 해결하기 위해서 본 논문에서 제안하는 것은 IPv4 패킷을 IPv6로 Header를 변환시킨 후, 패킷 단위의 암호화을 위한 IPSec의 ESP 확장Header를 추가 시켜 IPv6 호스트와 통신을 이루게 하여 암호화를 통한 안전한 통신을 구축하는 것이다.

3.1 프로세스 구성

제안한 IPv4와 IPv6간의 암호화 변환을 위해서는 다음 3단계를 거치게 된다.

- 1단계 : IPv4 패킷을 전송시킨다.

- 2단계 : IPv4 패킷을 IPv6 기본 Header와 Fragmentation 확장 Header로 변환시킨다.
- 3단계 : 2단계에서 생성된 패킷 헤더에 ESP 확장 Header를 추가 시켜 완전한 IPv6 헤더를 생성



(그림 3-1) 구성도

3.2 IPv4에서 IPv6로 암호화 변환 과정

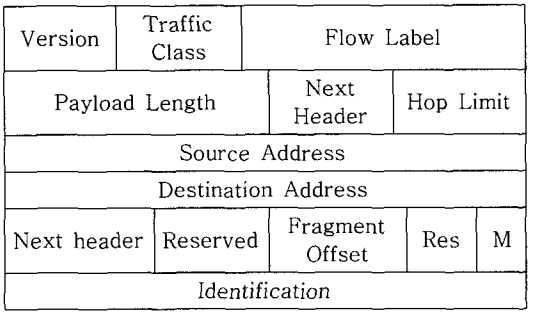
3.2.1 IPv4 패킷 전송 단계

IPv4 패킷 전송 단계는 일반적인 IPv4 환경에서의 생성되는 패킷 구조로 구성은 IP Header, TCP Header, 데이터로 구성되어 전송된다.

3.2.2 IPv6 기본 Header와 Fragmentation 확장 Header 구성 단계

IPv6 기본 Header와 Fragmentation 확장 Header 구성 단계에서는 본격적인 IPv4 패킷을 IPv6로 변환시키게 된다. 먼저 IPv6 기본 Header를 구성하고 다음으로 fragmentation 확장 Header를 구성하게 된다. 여기에서 만들어지는 패킷의 IP Header의 구조는 (그림 3-2)처럼 구성된다.

IPv6 기본 Header의 구성 방법은 IPv4의 필드 값들을 IPv6의 Header 구조에 맞게 그 값들을 넘겨 주거나 계산을 통하거나 새로이 생성 또는 폐기을 수행한다. IPv4와 IPv6간의 어드레스 변환은 IPv6를 지원하지 않는 IPv4 노드를 위하여 "IPv4-maped IPv6 address" 방법을 이용한다. 이는 IPv4 주소가 내장된 IPv6 주소이다.



(그림 3-2) IPv6 Header 구조

각각의 필드 값에 대한 변환 방식은 <표 3-1>, <표 3-2>와 같다. 그리고 IPv6를 지원하는 라우터는 IP 패킷을 분할할 수 없기 때문에 본 논문

〈표 3-1〉 IPv6 기본 Header 변환방식

필 드 명	변환되는 값(내용)
Version	IPv6 버전으로 변환. 6을 입력
Traffic Class	IPv4의 TOS(Type of Service) 필드 값과 대응되며 양쪽의 비트 수가 같기 때문에 그대로 복사한다. 일반적으로 '0'
Flow Label	아직 정의되지 않은 필드. '0'으로 입력
Payload Length	IPv4에서 Total Length 필드 값에서 IPv4의 IHL(IP Header Length) 필드 값을 뺀 값을 복사. 만약 Fragment가 된 상태이면 Total Length 필드 값에서 IHL 필드 값을 뺀 다음 Fragment Header Length 값을 더한다.
Next Header	IPv6 기본 Header 다음에 연결되는 확장 Header의 종류를 지정 Fragmentation 확장 Header : 44
Hop Limit	IPv4 Header의 TTL 필드 값을 복사
Source Address	상위 96비트는 IP-maped Prefix (::fff:0:0/96), 하위 32비트는 IPv4 Header의 Source Address
Destination Address	상위 96비트는 IPv4-maped Prefix (::fff:0:0/96), 하위 32비트는 IPv4 Header의 Destination Address

〈표 3-2〉 IPv6 Fragment 확장 Header 변환식

필 드 명	변환되는 값(내용)
Next Header	Fragmentation 확장 Header 다음에 연결되는 확장 Header의 종류를 지정. ESP 확장 Header : 50
Fragmentation Offset	IPv4의 Fragment Off의 필드 값을 복사한다.
M Flag	IPv4의 MF flag가 세팅되어 있으면 1, DF가 세팅되어 있으면 '0'
Identification	IPv4의 Identification 값을 복사

에서는 IP 패킷의 크기를 IPv6의 최대 전송 단위(MTU : Maximum Transmission Unit)인 1,280바이트보다 작은 크기를 갖도록 설정되어야 한다.

3.2.3 ESP 확장 Header 추가 단계

ESP 확장 Header 추가단계에서는 IPv6 기본 Header와 Fragment 확장 Header 구성 단계에서 완성된 IPv6 Header 다음에 ESP 확장 Header를 추가 시킨다. ESP Header에서 SPI 값과 Sequence Number값과 Initial Vector값은 본격적

<table>
<tr><td colspan="2">Version</td><td>Traffic Class</td><td colspan="3">Flow Label</td></tr>
<tr><td colspan="3">Payload Length</td><td>Next Header</td><td colspan="2">Hop Limit</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td>Next Header</td><td>Reserved</td><td colspan="2">Fragment Offset</td><td>Res</td><td>M</td></tr>
<tr><td colspan="5">Identification</td></tr>
<tr><td colspan="5">Security Parameter Index(SPI)</td></tr>
<tr><td colspan="5">Sequence Number</td></tr>
<tr><td colspan="5">Initialization Vector</td></tr>
<tr><td colspan="5">TCP Header</td></tr>
<tr><td colspan="5">Data</td></tr>
<tr><td colspan="3"></td><td>Pad</td><td>Pad Length</td><td>Next Header</td></tr>
<tr><td colspan="5">Authentication data</td></tr>
</table>

(그림 3-3) 최종 IPv6 Header 구조

〈표 3-3〉 3단계 ESP 확장 Header 변환방식

필 드 명	변환되는 값(내용)
Security Parameter Index(SPI)	임의의 32비트 값으로 SA를 나타내므로 필수적으로 사용되어야 하며 목적지 시스템에 의해 결정된다
Sequence Number	Reply Attack 방지를 위해 필수적이며 전송된 Number은 절대로 재사용되어서는 안된다.
Initial Vector	DES-CBC 모드에서 초기값을 제공
Data	암호화된 Data
Pad	블록암호의 블록크기를 32비트의 정수배로 만든다.(0~255byte)
Pad length	Pad 필드의 바이트 수를 계산해서 입력
Next Header	상위 계층의 Header값 지정. TCP : 6, UDP : 17, ICMP : 1
Authentication Data	선택된 인증함수에 따라 결정되는 가변길이 값

인 통신을 하기 전에 IKE 통신을 통해서 SA을 형성시키고 여기에 저장된 보안과 관련된 값들을 참조하여 필드 값을 채워 넣어야 한다. 이번 단계에서 만들어질 패킷의 IP Header의 구조는 (그림 3-3)같이 구성된다.

3.3 IPv6에서 IPv4로 암호화 변환 과정

IPv6의 패킷을 수신했을 경우는 앞에서 언급한 반대의 단계로 진행을 시킨다. 먼저 수신 받은 패킷에 대해서 SA에 저장된 보안 정보를 바탕으로 인가 받은 패킷인지를 확인한 다음 정당한 패킷이면 ESP 헤더를 제거하고 암호화 된 부분을 복호화 시킨다. 그런 다음 남아 있는데 IPv6 기본 Header와 Fragment 확장 Header를 <표 3-4>와 같이 IPv4 Header로 변환 시킨다.

IPv6 패킷 내에는 Hop-by-Hop 확장 Header나, 목적지 확장 Header, 혹은 Routing 확장 Header가 붙을 경우에는 IPv4 네트워크 환경에서는 수용할 수 없기 때문에 이러한 확장 Header들이 부착된 경우는 무시하고 변환을 시도하지 않는다. IPv6에서 IPv4로의 변환 변화 과정에서

중요한 부분은 IPv4의 패킷을 생성시킨 뒤에 정해진 루틴으로 Check-sum값을 재계산해야 된다.

〈표 3-4〉 IPv6/IPv4 Header 변환 방식

필 드 명	변환되는 값(내용)
Version	IPv4 버전으로 변환. '4'을 입력
IHL	기본값으로 설정 (IPv4의 옵션이 없는 경우 5)
TOS	IPv6 Header의 Class 필드 값 복사
Total Length	IPv6 Header의 Payload Length 필드 값에 IPv4 Header 크기를 더한 값
Identification	Fragmentation 확장 Header의 Identification 필드 값을 복사
Flags	Fragmentation 확장 Header의 M flag가 1이면 MF을 설정, 0이면 DF를 설정
Fragment Offset	Fragmentation 확장 Header의 Fragmentation offset 값을 복사
TTL	IPv6 Header의 Hop Limit 필드 값에서 1을 뺀 값을 복사
Protocol	상위 계층의 Header값 지정하는데 ESP 확장 Header의 next header 필드 값을 복사
Header Checksum	IPv4 Header를 생성시킬 때 계산
Source Address	IPv6 Header의 Source Address 필드의 하위 32비트를 복사
Destination Address	IPv6 Header의 Destination Address 필드의 하위 32비트를 복사

4. 시스템 구현

본 논문에서 제안하는 IPv4/IPv6의 암호화 변환을 위한 시스템의 구성 방법은 IPv4 망과 IPv6 망 사이에 경계 라우터를 구성하고, 여기에 IPv4/IPv6의 암호화 변환을 위한 알고리즘을 탑재한다. 경계 라우터에서 IPv4 네트워크망과 연결된 쪽에서는 IPv4 패킷만 수신 또는 송신하고, IPv6 네트워크망에 연결된 쪽에서는 IPv6 패킷을 전용으로 수신 또는 송신하게 끔 구성하다.

기본적인 IPv4/IPv6의 프로토콜 변환 방식은 RFC 2765에서 제안된 SIIT 방식으로 변환시키고 패킷 단위의 암호화를 위해 IPSec의 ESP Header를 적용시킨다.

4.1 시스템 모듈 구성

시스템은 총 네 가지 모듈로 구성된다. IPv4 패킷을 수신 받아 IPv6 패킷으로 변환시켜주는 프로세스는 "IPv4/IPv6 변환모듈"에서 구현되고, 상위계층의 데이터을 암호화 시켜주고 그 값을 다시 IPv4/IPv6 변환모듈에 반환시켜 주는 프로세스는 "암호화 모듈"에서 구현된다. 암호화 모듈에서 사용되는 암호화 방식은 SA 협상을 통해서 통신하는 상대편과 사용할 암호 알고리즘을 결정되어 있다. IPv6 패킷을 수신 받아 IPv4 패킷으로 변환시켜주는 프로세스는 "IPv6/IPv4 변환 모듈"에서 구현이 되고 수신받은 IPv6의 패킷을 IKE의 SA 값과 비교하여 패킷의 무결성 및 인증을 확인한 다음 복호화 시키는 프로세스는 "복호화 모듈"에서 구현된다.

4.1.1 IPv4/IPv6 변환 모듈

IPv4/IPv6 변한 모듈에서는 IPv4 네트워크와 연결된 랜카드을 통해 IPv4 패킷을 수신한 다음 IPv4용 버퍼와 IPv6용 버퍼와 상위 계층용 버퍼를 지정하다. IPv4용 버퍼는 20바이트 크기로, IPv6용 버퍼는 기본 Header와 Fragment 확장 Header를 더한 크기 만큼 지정한다.

그리고 상위 계층용 버퍼는 최대 패킷 크기인 1,280 바이트에서 IPv4 Header 크기를 뺀 1,260 바이트에 ESP 확장 Header 크기를 더한 만큼을 지정한다. 이들 버퍼를 이용하여 IP Header와 상위 계층의 Header를 분리하여 IP Header를 준비된 버퍼에 읽어 들여서 struct ip4로 타입 캐스팅(Type Casting)하여 IPv4 Header의 필드 값들을 분석한다. 그리고 IPv6 패킷을 위한 버퍼를 지정하고 이것을 (그림 4-1)와 같이 struct ip6_hdr로 타입 캐스팅하고 여기에 각각에 구조체 멤버에

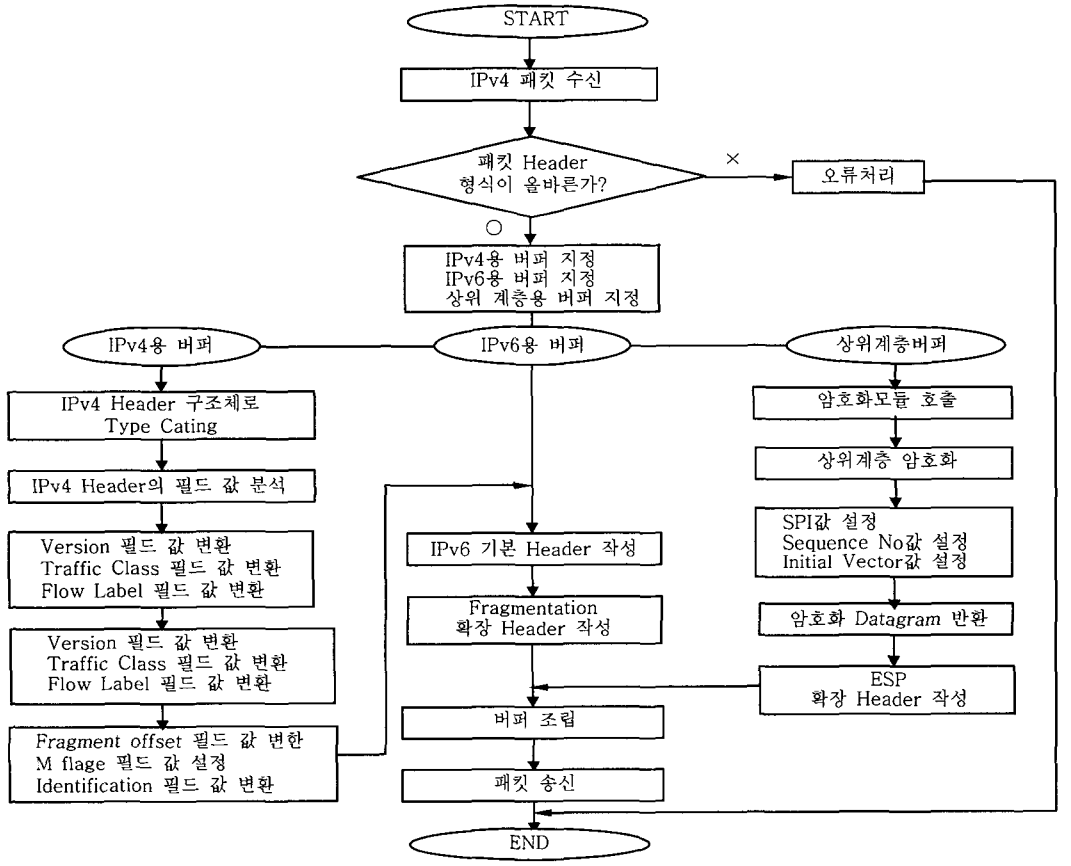
해당되는 필드 값들을 채워 넣거나 계산해서 입력한다.

```
struct ip6_hdr {
    union {
        struct ip6_hdrctl {
            uint32_t ip6_un1_flow ; /* 24 bits of
                                       flow-ID */
            uint16_t ip6_un1_plen ; /* payload
                                       length */
            uint8_t ip6_un1_nxt ; /* next header */
            uint8_t  ip6_un1_hlim ; /* hop limit */
          } ip6_un1 ;
    uint8_t ip6_un2_vfc ;  /* 4 bits version,
                              4 bits priority */
      } ip6_ctlun ;
    struct in6_addr ip6_src ;  /* source address */
    struct in6_addr ip6_dst ; /* destination
                                 address */
} ;
```

(그림 4-1) IPv6 기본 Header 구조체

```
struct ip6_frag  {
    uint8_t    ip6f_nxt ;        /* next header */
    uint8_t    ip6f_reserved ;   /* reserved field */
    uint16_t   ip6f_offlg ;      /* offset, reserved,
                                    and flag */
    uint32_t   ip6f_ident ;      /* identification */
} ;
```

(그림 4-2) IPv6 fragment 확장 Header 구조체



(그림 4-3) IPv4/IPv6 변환모듈 동작절차

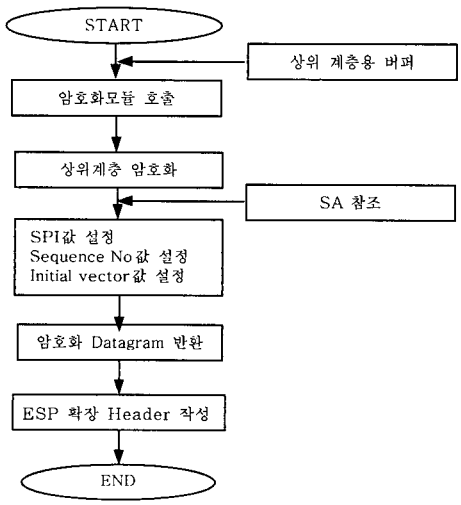
그런 다음 (그림 4-2)를 이용하여 지정된 버퍼에 struct ip6_frag를 타입 케스팅하고 struct ip 멤버값 중에서 변환시켜 준다. 그런 다음 암호화 모듈에서 반환 받은 상위 계층용 버퍼와 ESP 확장 Header를 조립하여 랜카드 eth1로 전송하면 된다. IPv4/IPv6 변환 모듈의 전체적인 흐름도는 (그림 4-3)의 그림과 같다.

4.1.2 암호화 모듈

암호화 모듈에서는 "IPv4/IPv6 변환모듈"에서 상위계층의 버퍼용 포인터 값을 인자로 받아 암호화 모듈를 호출하여 SA에서 지정된 암호화방식으로 상위계층의 데이터를 암호화 한다. ESP 확장 Header를 구성하기 위해서 통신 전에 미리 협상된 SA를 참조하여 (그림 4-4)의 struct esp

```
struct esp {
    uint32_t esp_spi ;    /* Security Parameters
                             Index */
    uint32_t esp_rpl ;    /* Replay counter */
    uint8_t  esp_iv[8] ;  /* iv */
};
```

(그림 4-4) IPv6 ESP 확장 Header 구조체



(그림 4-5) 암호화 모듈 동작 절차

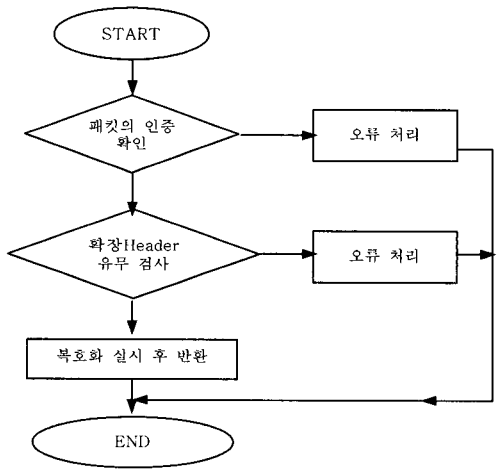
를 멤버 값을 입력시킨다. 암호화 모듈을 통해 암호화된 상위계층의 데이터와 ESP 확장 Header를 "IPv4/IPv6 변환 모듈"로 반환한다.

4.1.3 IPv6/IPv4 변환 모듈

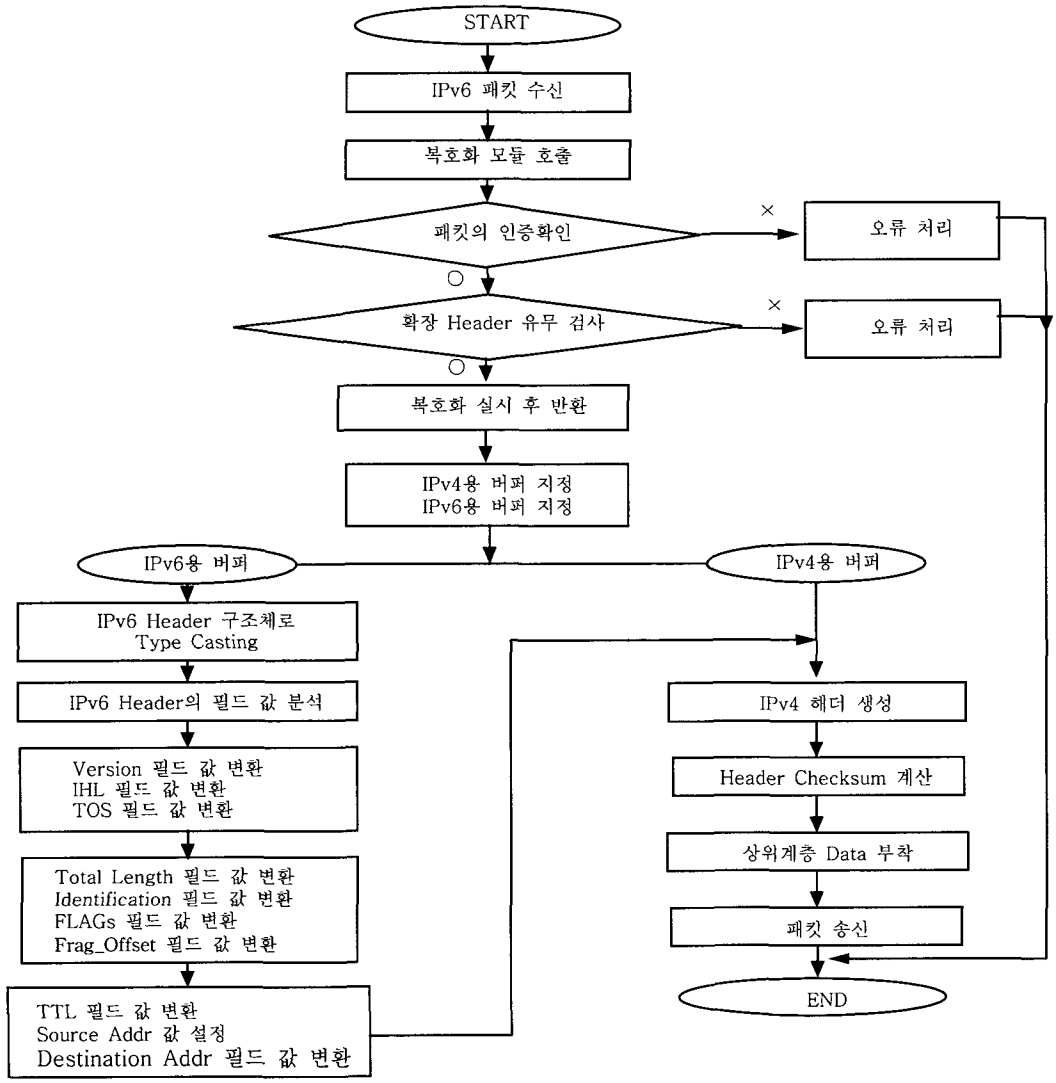
IPv6/IPv4 변환모듈에서는 랜카드 eth1에서 IPv6 패킷을 수신 받아 SA값을 참조하여 패킷의 인증과정을 과정을 거쳐 정당한 패킷임을 확인 한 다음 복호화 모듈로 복호화한다. 지정된 버퍼에 struct ip6_hdr로 타입 케스팅해서 각각의 멤버 값을 분석한다. 같은 방법으로 struct ip6_frag로 타입 케스팅하여 멤버 값을 분석한 다음 IPv4 Header를 구성한다. IPv4의 Header가 완성되었으면 IP Header의 무결성을 점검하기 위하여 ip_sum을 계산하여야 한다. 버퍼의 크기는 IPv4/IPv6 변화 모듈에서와 같이 IPv4용 버퍼는 20 바이트 크기로 지정하고 IPv6용 버퍼는 기본 Header와 Fragment 확장 Header를 더한 크기 만큼 지정한다.

4.1.4 복호화 모듈

수신한 IPv6의 패킷을 SA를 참조하여 ESP 필드 값들과 비교하여 패킷의 완전성을 검사하



(그림 4-7) 복호화 모듈 동작 절차



(그림 4-6) IPv6/IPv4 변환모듈 동작절차

여 패킷의 이상이 없음을 확인한 다음 복호화를 실시한다. 복호화가 끝난 상위계층의 데이터에 대해서 IPv4 Header가 완성된 다음에 부착시켜 준다. 그리고 만약 IPv6의 다른 확장 Header가 부착되었을 경우는 IPv4 네트워크 망에서 처리할 수 없기 때문에 폐기처리하고 오류 처리를 해주어야 한다.

4.2 시스템 비교 평가

Header 변환 방식 중에서 대표적인 방법이 SIIT를 바탕으로 하는 NAT-PT 방식이다. 이것이 가지는 종단간의 보안을 강화시키기 위해서 본 논문에서 제안한 "암호화된 Header 변환 방식"은 취약한 프로토콜 변환 시에 암호화를 적용시키는 것이다. 암호를 적용 시키는 방법은 IPv6에

서도 패킷 단위의 암호를 위해 사용하고 있는 ESP Header 형식을 사용하여 적용시킨다.

앞에서 언급한 두 가지의 IPv4/IPv6 Header 변환 방식을 비교해 보면, 먼저 공통된 내용은 IPv4의 Header와 IPv6 Header를 교환시켜주는 것이고 차이점은 기존의 Header 변환 방식(NAT-PT)에 암호화를 작용시키는 것이다. 이를 적용시켜 얻을 수 있는 결과는 패킷 단위의 보호를 수행하는 것이다.

그리고 ESP Header에 포함된 ICV(Integrity Check Value)값을 이용하여 ESP Header와 페이로드(Payload)에 대한 무결성 확인이 가능하기 때문에 전송 중에 데이터값의 변조를 쉽게 확인할 수 있다. 그리고 IKE를 통해 형성된 세션을 통해 근원지 인증도 가능하다. 그러나 암호화 및 복호화 프로세스가 추가되기 때문에 이러한 프로세스에 소요되는 시간이 추가된다는 단점은 가지고 있다.

<표 4-1>에서 알 수 있듯이 NAT-PT 방식과 암호화된 Header 변환 방식은 기본적인 IPv4 패킷을 IPv6 패킷으로 변환시켜 주거나 IPv6 패킷을 IPv4 패킷으로 변환 시켜주는 기능은 제공하고 있지만 암호화된 Header 변환 방식에서는 패킷이 유출되었을 때 제삼자가 그 내용을 알아볼 수 없으며, 패킷이 유출되더라도 복호화 키를 가지고 있지 않는 한 패킷의 내용을 의도한 대로 수정을 할 수 없다. 그리고 패킷이 제삼자나 다른 이유로 변경이 되었을 지라도 SA에 의한 지정된 인증 알고리즘으로 변경유무를 확인할 수 있다. 또한 암호화된 Header 변환 방식은 단순히 출발지 주소를 근원지를 확인하는 방법과는 달리 IKE의 통신 과정에서 근원지인증을 수행하기 때문에 정확한 근원지 인증이 가능하다. 그러나 암호화된 Header 변환 방식은 NAT-PT에 비해 처리해야 할 필드수가 6개가 더 많으므로 처리 시간이 더 소요될 수 밖에 없다.

암호화된 변환 방식은 패킷 단위로 암호화를 적용시키기 위해 IPSsec의 ESP의 포멧을 사용하였기 때문에 ESP가 제공하는 보안 서비스 즉 데이터의 기밀성, 근원지 확인, 데이터 무결성를 제공하고 있으므로 이를 통해서 NAT-PT의 종단간 취약성에 대해서는 보안이 향상되었음을 알 수 있다.

〈표 4-1〉 시스템 비교

항 목	내 용	NAT-PT 방식	암호화 변환 방식
IPv4/IPv6 변환	IPv4의 패킷이 IPv6 패킷으로 변환이 가능한가?	○	○
패킷의 기밀성	전송되는 패킷이 유출이 되더라도 제 삼자가 알아볼 수 있는가?	×	○
패킷의 내용 변경	유출된 패킷에 제삼자의 의도대로 내용을 수정할 수 있는가?	×	○
패킷 변경확인	유출된 패킷이 제삼자나 다른 이유로 인해 변경된 사실을 확인할 수 있는가	×	○
근원지 인증	패킷의 근원지에 대한 인증을 위한 장치가 마련되어 있는가?	×	○
변환 Header 필드수	변환시 처리 해야 할 Header 필드의 수	15개	21개

5. 결 론

본 논문은 IPv4/IPv6의 변환방식 중에서 안전한 Header 변환 방식을 위한 암호화된 Header 변환 방식을 제안하였다. 암호화된 Header 변환은 IP Header의 필드 값들을 일대일 매칭과 계산을 통해 다른 버전의 IP Header로 변환을 시키고, 패킷 단위의 암호화를 위해서 IPSec의 ESP Header를 추가시키는 방법이다. 변환 과정을 살펴보면 세 단계에 결친 변환이 이루어 지는데 IPv4에서 IPv6로 변환하는 경우, 일반적인 IPv4 네트워크 환경에서 IPv4 패킷의 생성 단계와 IPv4의 패킷을 IPv6의 기본 Header와 Frag-

ment 확장 Header로 바꾸는 단계를 거쳐 ESP Header를 추가하는 단계를 거치게 된다. 마찬가지고 IPv6에서 IPv4로 변환하는 과정도 수신한 IPv6 패킷을 SA에서 지정된 인증과정과 복호화 알고리즘으로 ESP Header를 제거하는 단계와 IPv6 패킷을 IPv4로 변환시키는 단계를 거쳐 일반적인 IPv4 패킷이 만들어 지는 단계로 구성된다.

본 논문에서 제안한 암호화된 Header 변환 방식은 ESP가 제공하는 패킷의 암호화, 근원지 인증, 무결성 확인, Reply 공격 방지와 같은 보안 서비스를 제공하고 있다. 패킷 단위의 암호화 서비스는 패킷이 중간에 유출되더라도 제삼자가 패킷의 내용을 인식하기 어렵게 만들고 제삼자가 의도한 대로 내용을 수정할 수가 없게 된다. 근원지 인증 서비스는 수신한 패킷이 인가된 호스트로 부터의 정당한 패킷인지를 확인시켜준다. 데이터의 무결성 확인 서비스는 패킷의 전송 중에 패킷의 변조나 변경을 쉽게 확인할 수 있는 방법을 제공한다. 그러나 이러한 보안과 관련된 프로세스를 추가로 처리해야 되므로 시간이 더 소요되는 단점은 있으나 안전하고 신뢰할 수 있는 IPv4/IPv6의 변환 방식을 제공한다 하겠다.

황 호 준

(주)삼성전기 경력 5년
경기대학교 일반대학원 정보보호기술공학과 석사
현재 호원대학교 컴퓨터학부 외래강사

유 승 재

1988년 동국대학교 수학과 이학사
1990년 동국대학교 수학과 이학석사
1998년 동국대학교 수학과 이학박사
1997년~현재 중부대학교 정보분석학과 조교수

김 귀 남

미국 캔자스대학 수학과(응용수학사)
미국 콜로라도주립대학 통계학과(통계학석사)
미국 콜로라도주립대학 기계·산업공학과(기계·산업공학박사)
현재 경기대학교 정보보호기술공학과 주임교수