情 報 保 護 學 會 誌
第 15 卷 第 4 號, 2005. 8

호스트 신원 프로토콜 기술

이 윤 진*, 조 인 준*

요 약

현재 인터넷에서 사용되고 있는 IP주소는 호스트 위치와 신원을 동시에 식별할 수 있도록 설계되어 있다. 이러한 설계 패러다임은 호스트가 고정된 위치에서 하나의 IP주소를 갖는 기존의 인터넷 환경에 적응적이다. 하지만 차세대 인터넷에서는 호스트 이동성, 멀티호밍, 보안 등이 중요한 핵심서비스로 부각되고 있다. 이러한 환경에서는 위치를 나타내는 IP주소가 동적으로 변경되고, 하나의 호스트에 두개 이상의 IP주소가 할당되고, 호스트 보안의 강화를 요구하는 새로운 환경이다. 따라서 현재의 IP주소가 호스트 위치와 신원을 동시에 나타내는 설계 패러다임으로는 새로운 차세대 인터넷 환경을 원활하게 지원할 수 없다.

본 논문에서는 차세대 인터넷 환경에서 부각되고 있는 호스트 이동성(Mobility), 멀티호밍(Multi-Homing), 보안 등의 서비스를 원활하게 지원할 수 있는 하나의 새로운 기술로 HIP(Host Identity Protocol)을 소개하였다. HIP은 2004년 IETF hip WG가 결성되어 현재 표준화가 진행 중인 차세대 인터넷 기술이다. 기본 아이디어는 호스트 신원과 위치정보를 분리하여 차세대 인터넷 환경에 적응적인 프로토콜을 새롭게 재구성한 것이다.

Ⅰ. 서 론

현재의 인터넷은 컴퓨팅 플랫폼(종단점)과 패킷 전송 기반구조 그리고 서비스(응용) 이렇게 세 가지 근본적인 구성요소로 이루어져 있다. 이들 모든 구성요소들이 확장성을 자기면서 유기적으로 상호작용하기 위해서는 각각에 이름을 부여하는 새로운 체계가 필요하다. 현재의 인터넷에서는 이들 구성요소들에 대해 두 가지의 이름만을 부여하고 있다. 즉, IP주소와 도메인 이름이다.

현재 인터넷에서 사용되고 있는 IP 주소 역할은 호스트 위치와 호스트 신원을 동시에 나타내도록 설계되어 있다. 여기에서 위치정보를 나타내는 IP주소는 패킷의 경로설정을 위한 것이고, 호스트 신원을 나타내는 IP주소는 네트워크 인터페이스에 부여된 이름이다. 따라서 종단간의 신뢰성 있는 통신이 기본기능인 전송계층에 IP주소가 결합되어 있다.

다음으로 도메인 이름(DN, Domain Name)은 숫자로 이루어진 IP주소를 문자로 알기 쉽게 표현한 이름체계이다. 여기에서 도메인이름은 컴퓨팅 플랫폼(종단점)과 서비스에게 계층적으로 할당된 이름을 의미한다. 이와 관련하여 현재 유용하게 사용되고 있는 E-mail, SIP(Session Initiation Protocol), PPP (Point-to-Point Protocol), WWW(World Wide Web) 등의 주소는 단지 도메인 이름의 확장일 뿐이다. 이러한 도메인 이름체계에서는 익명성을 제공하지 못하는 특징을 지닌다.

이와 같은 두 가지 이름영역체계는 다음과 같은 세 가지 측면에서 비효율적인 문제점을 가지고 있다. 첫째, 동적 주소부여가 직접적으로 이루어질 수 없다. 둘째, 익명성이 일관되고 믿을 수 있게 제공될 수 없다. 셋째, 시스템과 데이터 그램에 대한 인증서비스가 제공될 수 없다.[1]

이러한 현재 인터넷의 이름체계는 차세대 인터넷에서 부각되고 있는 호스트 이동성, 멀티 호밍, 보안 등의 서비스를 원활하게 지원하지 못하는 근본적인 문제를 지니고 있다. 즉, 호스트의 위치가 변경되면 호스트 신원도 변경되어야 하고, 하나의 호스트에 여러 개의 IP주소가 부여되는 멀티호밍지원이 원활하지 못하다. 또한 현재의 인터넷에서 적용된 보안기술들은 호

* 배재대학교 컴퓨터공학과 ({gomyung, injun}@pcu.ac.kr)

스트 신원과 위치를 동시에 나타내는 IP주소를 기반으로 설계되어 있기 때문에 복잡한 보안 매커니즘을 지니게 된다. 이와 같은 문제점의 주 요인은 현재의 인터넷이 호스트 신원을 나타내는 별도의 이름체계를 갖추고 있지 않기 때문이다.

이러한 문제를 극복하고자 호스트의 신원과 위치를 각각 분리하고, 새로운 호스트 신원을 나타내는 이름체계를 추가하여 호스트의 이동성, 멀티호밍, 보안 등을 서비스를 용이하게 지원하는 호스트 신원 기술로 HIP(Host Identity Protocol)이 IETF(Internet Engineering Task Force)에서 주목을 받고 있다.

HIP은 2003년 11월에 Minneapolis meeting에서 hip WG BoF(Birds of a Feather)를 갖고 2005년 현재 정식 IETF hip WG으로 등록되어 표준화가 진행되고 있는 차세대 인터넷 기술이다.

본 논문에서는 이러한 HIP기술을 소개하기 위하여 다음과 같이 구성하였다. 2장에서는 HIP 기술의 출현배경을 설명하였다. 3장에서는 HIP 프로토콜 구조 및 동작절차를 설명하였다. 4장에서는 HIP에서 HI(Host Identity)와 IP주소의 동적사상 기술하였다. 5장에서는 향후 HIP 전개방안을 살펴보았고 6장에서는 결론을 맺었다.

II. HIP 기술의 출현 배경

우리는 일상에서 이사를 하게 된다. 이사를 하게 되면 변경되는 것은 주소일 뿐, 나를 식별해주는 주민등록번호나 이름은 변경되지 않고 그대로 유지되게 된다. 하지만 현재의 인터넷에서는 호스트가 이동을 하여 다른 곳에 위치하게 되면 IP주소가 변경되게 되고, 이에 따라 자신을 식별해 주는 호스트의 신원도 변경되게 된다. 이는 호스트 신원을 나타내는 특정 이름체계를 현재의 인터넷이 갖추고 있지 않기 때문이며 이는 불합리하다.

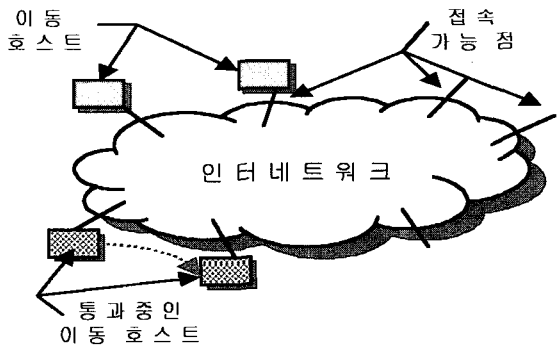
현재 우리는 모두 유비쿼터스 컴퓨팅(Ubiquitous Computing) 시대의 도래를 예상하고 있고 ad hoc 네트워크가 개발되고 있다. 이는 호스트의 주 기능으로 이동성, 멀티호밍 서비스들이 기본적으로 탑재되어야한다는 것을 의미한다. 따라서 이러한 환경에서는 현재의 정적 IP주소보다는 동적인 IP주소가 보편화된다. 인터넷 환경은 빠르게 변화를 거듭해왔다. 2000년대를 기점으로 살펴보면, 호스트 간에 교환되는 메시지의 보안과 더불어 호스트 이동성과 멀티호밍의 요구들이 새로운 변화를 선도하고 있다. 하지만, 현재의 인터넷 구조의 한계 때문에 이들 변화를 원만하게 수용하기는 어려울 것으로 보인다. 주 요인은 IP계층에서 IP주소 변경이 전송계층 이상의 상위계층에 영향을 미치기 때문이다. 따라서 지금까지의 새로운 요구사항을 TCP/IP가 수용할 수 있도록 인터넷 구조를 재 공학 할 필요가 있다. 이렇게 TCP/IP 구조를 재 공학 하는데 있어서 다루어져야 할 내용은 이동성, 멀티호밍, 그리고 보다 자연스러운 인터넷 보안서비스 등이다. 따라서 이러한 서비스들이 어떻게 처리되는지를 살펴볼 필요가 있다. 특히 종단 호스트의 이동성과 멀티호밍 구조를 정의할 필요가 있다.

결론적으로 HIP의 출현 배경은 크게 종단 호스트 관점에서 이동성, 멀티호밍, 그리고 안전한 보안 서비스가 도출될 수 있다.[2] HIP출현의 단초를 제공하는 이들 각각에 대해 보다 구체적인 내용을 살펴보면 다음과 같다.

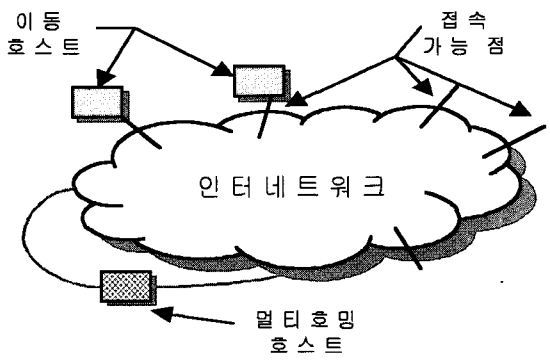
2.1 이동성

호스트가 이동한다는 것은 그림 1과 같이 어떤 개체가 자신의 통신 문맥을 활성화 상태로 유지하면서 이동하는 현상을 의미한다. 이때 종단 호스트는 네트워크 위상에서 접속점을 변경하고 동시에 모든 통신문맥이 활성화 상태로 유지되기를 원한다. 종단호스트가 이동하면 자신의 네트워크 주소는 필수적으로 변경된다. 따라서 통신의 연속성 보장을 위해서 통신 중인 상대방에게 자신의 주소가 변경된 사실을 알리는 주소변경 신호가 안전하게 보내져야만 한다.

여기에서 핵심은 두 가지이다. 첫째, 호스트 이동에 따라 신속한 IP주소의 재구성이다. 둘째는 이동 호스트가 재구성된 IP주소를 통신 상대방에게 안전하게 알려주는 역할이다. 하지만 현재의 주소 체계인 MIPv4(Mobile Internet Protocol version 4)나



〔그림 1〕 이동 모델



(그림 2) 멀티호밍 모델

MIPv6(Mobile Internet Protocol version 6)에서는 IP계층에 이동한 IP주소와 자신의 영구주소인 홈 주소를 사상하는 방법을 택하고 있기 때문에 이러한 서비스를 원활하게 제공할 수 없다.

2.2 멀티호밍

종단 호스트가 그림 2와 같이 사용이 가능한 여러 개의 통신 패스를 가지고 있으면 멀티호밍이라 한다. 멀티호밍 호스트는 여러 개의 네트워크에서 특정위치에 동시에 존재할 수 있음을 의미 한다. 따라서 여러 개의 네트워크 계층주소를 가질 수 있으며 그것은 각각 네트워크 위상에 서로간의 독립적인 한 위치를 가지는 것을 의미한다.

여기에서 핵심은 2 가지이다. 첫째, 멀티호밍 호스트가 통신하는 시점에 다중 통신 패스 중 어느 것을 사용할 것인가이다. 둘째는 멀티호밍 호스트가 활성화된 통신패스를 통신 상대방에게 안전하게 알려주는 역할이다. 하지만 현재의 인터넷 환경에서는 IP주소가 전송계층에 결합되어 있기 때문에 이러한 IP주소 변경을 신속하게 지원하는 것이 어렵다.

2.3 보안 문제

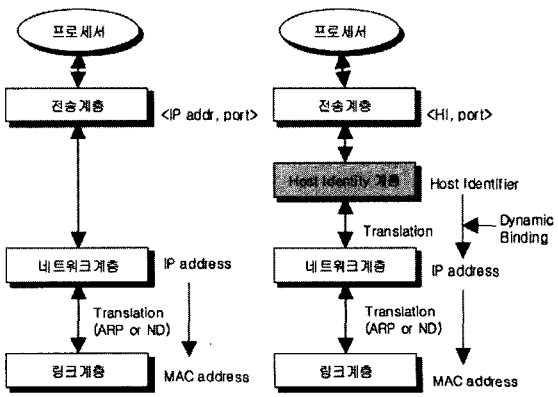
이동성과 멀티호밍에는 수많은 보안 문제가 내재되어 있다. 이들은 단일 호스트에 여러 IP주소를 할당하는 것이 요인이 되고, 이들 주소를 서로 교차하여 사용하고자 하는 요구 때문에 발생한다. 이상적인 이동성과 멀티호밍을 위한 해결책은 호스트들이 그 주소의 타당성에 관해서 우려감 없이 상대의 주소가 어떤 것이나 사용할 수 있게 하는 것이다. 이렇게 주어진 상황에는 주소탈취(Address Stealing)와 주소홍수(Address Flooding) 공격이라는 두 가지 기본적인 보안문제가 내제되어 있다. 주소 탈취는 악의적인 노드가 다른 노드가 사용 중인 주소를 자신의 의도에 따라 주장하는 것이다. 따라서 의도한 주소로 패킷을 흐르도록 위장하고, 중간에서 MITM(Man In The Middle)공격, 혹은 DoS(Denial of Service)공격을 의도한 것이다. 주소홍수 공격은 악의적인 노드(혹은 노드의 그룹)가 공격 목표주소를 수많은 순수한 상대 노드들이 믿을 수 있게 만든다. 이렇게 함으로써 순수한 상대 노드들이 원하지 않은 트래픽을 공격목표 주소에 홍수처럼 흐르게 한다.

이동성과 멀티호밍 환경에서 공격자는 첫 번째 공격을 매우 쉽게 시도할 수 있다. 이는 단순히 공격자가 현재 MN(Mobile Node)과 통신 중인 상대노드에게 MN이 새로운 주소로 이동했다는 것을 알리기만 하면 된다. 이러한 위치변경 메시지를 수신한 상대노드가 현재의 MN이 보낸 위치변경 메시지임을 증명하지 않는 한 MN으로 향하는 모든 트래픽은 공격자가 의도한 노드로 흐르게 할 수 있다. 이는 공격자의 행위나 위치에 따라 위장, DoS, MITM공격이 될 수 있다.

Ⅲ. HIP 프로토콜 구조 및 동작절차

3.1 HIP 프로토콜

제 Ⅱ장에서 살펴본바와 같이 차세대 인터넷에서 호스트의 이동성, 멀티호밍, 보안 등의 서비스들이 자연스럽게 지원되어야 한다는 것을 전제로 하고 있다. 현재의 인터넷이 이에 적응적이지 못하는 주요인이 IP주소의 이중역할(즉, 호스트 신원과 위치 식별자) 때문임에 착안하여 제시된 프로토콜이 HIP이다. HIP은 기존의 통신 S/W 구조를 변경하여 그림 3과 같이 새로운 구조를 정의하였다.[3,4]



(그림 3) 기존 인터넷 구조와 HIP가 적용된 인터넷 구조

그림 3의 왼쪽 그림에서 보듯이 기존의 통신 S/W 구조의 특징은 응용프로세서가 전송계층 소켓에 바운드 되고, 바운드 된 소켓은 〈IP주소, 포트번호〉로 식별된다. 그리고 IP계층에서 소켓식별 요소인 IP주소를 그대로 사용하여 라우팅을 하는 구조이다.

그림 3의 오른쪽 그림에서 보듯이 HIP이 적용된 새로운 통신 S/W 구조의 특징을 살펴보면 다음과 같다. 전송계층 소켓은 IP주소 대신에 HI(Host Identity)를 사용한다. 즉 전송계층에 바운드된 소켓이 〈HI, 포트번호〉로 식별된다. 다음으로 전송계층과 네트워크 계층사이에 HIL(Host Identity Layer)를 위치시켰다. 이 계층의 주 역할은 전송계층 소켓의 HI와 IP계층의 IP주소를 통신하는 시점에서 동적으로 사상하는 기능을 한다. 이렇게 함으로써 전송계층을 식별하는 호스트 신원과 호스트 위치를 나타내는 IP주소를 분리한 것이다.

결과적으로 HIP에서는 호스트 신원을 새롭게 정의한 HI로 나타내고, 호스트 위치를 기존의 IP주소가 나타내게 하였다. 즉, 기존의 IP의 이중 역할을 분리한 것이다.[4] 이와 같이 역할의 분리를 통해서 얻을 수 있는 이점은 크게 두 가지로 정리할 수 있다.

첫째, 호스트의 주소가 변경될 경우(즉, 호스트 이동, Re-homing, 호스트 주소 재설정 등)에 IP계층에서 신속한 주소 재설정이 가능하다. 이는 IP주소변경에 따라 상위계층에 영향을 주지 않기 때문이다.

둘째, 새롭게 호스트 신원을 공개키로 정의함으로써 호스트' 신원 기반 보안서비스를 용이하게 제공할 수 있다. 즉, 전자의 이점은 차세대 인터넷 환경이 고정 IP보다는 동적 IP주소 중심의 서비스로 발전함에 따라 이에 적응적임을 의미한다. 그리고 후자의 이점은 종단간의 호스트 통신에 공개키를 이용함으로써 메시지 인증과 같은 보안서비스를 자연스럽게 제공하게 됨을 의미한다.

이렇게 정의된 새로운 통신 S/W구조가 차세대 인터넷 서비스로 부각되고 있는 호스트 이동성, 멀티 호밍, 보안 등의 서비스를 원활하게 제공하는 이유를 간단하게 살펴보면 다음과 같다.

첫째, 호스트 이동성에 대해서 살펴보자. 기존 이동 인터넷(MIPv4와 MIPv6)에서는 IP계층에 자신의 고정 홈 주소(HoA, Home Address)와 현 위치를 나타내는 CoA(Care of Address)를 동시에 유지하여 이를 IP계층에서 사상하는 형태로 이동성을 지원하였다. 따라서 이동성 지원을 위해서는 HoA에 대응하는 CoA를 유지하는 복잡한 메커니즘(HA(Home Agent), Binding Cache등)이 도입되어야만 했다. 하지만 HIP에서는 소켓 ID(〈HI, 포트번호〉)요소에서 HI와 호스트 IP주소가 통신이 이루어지는 시점에서 동적으로 사상하기 때문에 소켓이 호스트 위치에 독립적이다. 따라서 네트워크 계층에서는 현재의 IP주소만을 유지하면 된다. 이러한 단순성은 복잡한 기존의 호스트 이동성 지원메커니즘을 단순화시킨 결과를 가져왔다.

둘째, 멀티호밍 호스트에 대해서 살펴보자. 멀티호밍 호스트는 호스트가 중심이 되어 통신 부하 분배, 백업 통신의 실현 등을 목적으로 한다. 따라서 한 호스트가 통신부하에 따라 자동적으로 IP주소를 재설정하여 통신을 할 필요가 있다. 또한 고장 난 채널을 인지하여 백업통신을 할 경우에도 자동적으로 IP주소를 재설정할 필요가 있다. 이러한 환경에서 HIP은 기존의 TCP/IP보다 빠르게 주소를 재설정하여 멀티호밍 서비스를 원활하게 지원할 수 있다.

셋째, 보안문제에 대해 살펴보자. 현 구조와 비교하여 새롭게 정의한 구조에서는 호스트 식별자 즉, HI를 새롭게 도입하였다. 이렇게 새롭게 도입된 호스트 식별자에 공개키라는 보안특성을 부여하였다. 따라서 호스트가 공개키 기반 보안시스템의 특성을 자연스럽게 활용하여 호스트들 간에 보안서비스를 용이하게 구현할 수 있는 토대를 마련한 것이다.[5]

HIP에서 호스트 신원은 기본적으로 공개키이다. 외부에 공개되는 공개키는 개인키를 소유한 호스트를 인증하는 역할을 한다. 키 쌍 중에 개인키를 소지한 호스트는 직접 그 네트워크에서 호스트 식별을 위해 사용된 공개키를 소유한다고 증명할 수 있다. 이와 같이 호스트 신원을 공개키와 개인키로 분리함으로써 안전하게 이동성 및 멀티호밍 서비스를 제공할 수 있다.

HIP구조에서 각 호스트는 단기 혹은 장기로 사용할 수 있는 하나 이상의 신원을 가진다. 이는 네트워크에서 호스트 식별에 사용된다. HIP에서 신원을 나타내는 식별자는 공개키와 개인키 쌍 중에서 공개키이다. 그 호스트가 개인키를 소유하고 있을 때 호스트는 공개키로 대표되는 호스트의 신원을 실제로 소유한다고 증명할 수 있다. 이는 다른 보안 인프라 구조 없이 호스트 신원을 통하여 안전한 통신을 이룰 수 있도록 하기위한 새로운 방안을 제시한 것이다.

결론적으로 HIP구조에서 응용들을 포함한 상위 계층에서는 IP주소를 알 필요가 없다. 대신에 이들은 목적지 호스트 식별자로 HI를 사용한다. 따라서 위치정보는 새로운 HIL(Host Identity Layer)계층에 은

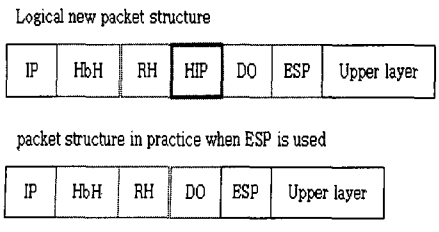
폐된다. 이때 IP주소는 노드 식별 역할을 더 이상 하지 않고 단지 네트워크내의 패킷 라우팅에 만 사용된다. 이렇게 되면 응용은 위치정보에는 관심이 없고 HI로 표현된 상대방의 신원만을 알기를 원한다. 이때 IP주소는 라우팅에 관심이 있는 하위계층에서만 중요성을 가진다. 응용에서 사용되는 HI는 어떤 패킷이 호스트를 떠나기 전에 대응되는 IP주소로 사상되어야만 한다. 이 기능을 새로운 계층인 HIL이 수행한다. 즉, 상위 계층으로부터 도착하는 각 패킷은 목적지 주소로 상대방의 HI를 포함한다. HI와 위치정보간의 사상은 HIL계층에서 이루어진다. 이는 원적지 HI가 원적지 IP주소로 변환됨을 의미한다. 이렇게 함으로 위치정보가 어느 때나 상대방 노드에 의해 갱신이 가능하다.

3.2 HIP 패킷 구조

논리적인 수준에서 새로운 구조는 패킷 구조 변경을 요구한다. 이는 각 패킷이 발신자와 수신자의 호스트신원을 논리적으로 포함해야 한다. 하지만 IPsec이 사용된다면, IPsec SA(Security Association)가 호스트신원을 위한 하나의 Short-cut처럼 사용될 수 있으며 이러한 경우에는 그림 4와 같이 현재의 패킷과 유사한 구조를 지니게 된다.[1]

3.3 HIP의 동작절차

HIP은 통신하고자 하는 HIP 노드 간에 안전한 채널형성을 위해 IPsec ESP에서 BEET(Bound End-to-End Tunnel)란 새로운 형태의 터널을 설정하여 동작한다.[6] 이는 IPsec에서 기존의 터널모드와 전송모드와는 다른 형태이다. 종단간의 터널들에 대해 새로운 BEET 모드는 기존의 정규적인 터널모드 사용을 위한 추가 부담 없이 제한적인 터널모드 의미를 제공한다. 이는 ESP의 새로운 활용분야로 이동성과 멀티호밍 지원을 의도한 것이다.



[그림 4] HIP 패킷 구조

현재의 IPsec ESP Spec[7]에 의하면 터널모드와 전송모드라는 두 가지 운용방식을 다음과 같이 정의하고 있다. 터널모드는 주로 비 종단 노드 대 종단 노드(Non-End-to-End)에 활용을 의도한 것이다. 이때, 두 종단 노드 모두 혹은 한 종단노드가 보안 게이트웨이가 될 수 있다. 따라서 ESP SA는 이들 노드 각각에 존재한다. 이는 ESP SA가 보안게이트웨이에 존재하기 때문에 실 종단노드로부터 분리된 결과를 낳는다. 전송모드는 종단간에 활용을 위한 것이다. 즉 종단간의 두 노드 모두가 SA를 맺고 있음을 의미한다.

BEET라는 새로운 모드의 목적은 기존의 정규적인 터널모드에 내재되어 있는 부담을 제거하여 제한적인 터널모드 시멘틱스를 제공한다. 이름에서 의미한 것처럼 BEET모드는 종단 간에 만 사용한다. 이는 응용인지 HI 이름과 회선인지 IP주소가 서로 분명히 별개의 것이란 의미에서 터널모드 시멘틱스를 제공한다. 즉, 응용레벨에서 HI 이름이 네트워크 레벨의 IP주소 위에서 터널링 되는 환경을 제공한다. 하지만 이 모드는 완전한 터널모드 시멘틱스를 지원하지 않는다. 더 자세히 말하면, 응용인지 HI 이름은 고정적으로 결합되고, 이렇게 결합된 한 쌍의 HI이름만이 주어진 BEET모드 SA에서 사용된다. 이는 응용인지 HI 이름이 어떤 IP주소로도 사상이 될 수 있다는 점에서 정규 터널모드와 대조적이다.

HIP에서 상기와 같이 BEET모드를 사용하여 통신하는 두 노드 간에는 안전한 IPsec ESP채널이 형성된다. 이는 DoS와 MitM 공격에 대비한 설계라 할 수 있다. 구체적인 동작 절차는 다음 절과 같다.[6]

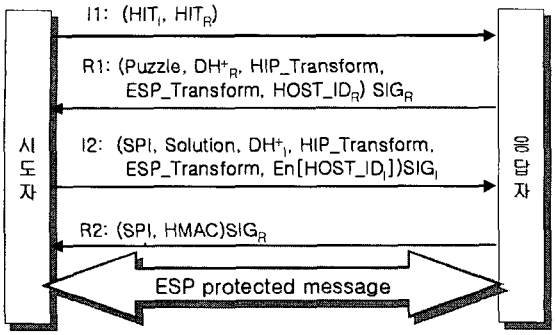
3.4 HIP 메시지 교환

HIP을 사용하여 두 노드 간에 통신할 경우에는 BEET모드 형태의 IPsec ESP SA설정을 위해 HIP 기본 메시지 교환이 선행된다. 이 절차가 끝나면 두 노드 간에는 IPsec ESP 채널을 통하여 안전하게 통신이 이루어진다. 먼저 HIP 기본 메시지 교환(I1, R1, I2, R2) 동작절차인 4-way 핸드쉐이크 과정을 그림 5에서 살펴보자.[8]

[1-way] 시도자는 I1메시지를 생성하여 응답자에게 전송한다.

I1 : (HIT_I, HIT_R)

시도자는 자신의 신원(HIT_I)과 응답자의 신원(HIT_R)을 응답자에게 전송한다.



[그림 5] 4-way 핸드쉐이크 과정

[2-way] I1메시지를 수신한 응답자는 R1메시지를 생성하여 시도자에게 전송한다.

R1 : {HIP(Puzzle, DH^+_R, HIP_Transform, ESP_Transform, HOST_ID$_R$) SIG$_R$}

R1메시지를 구성하는 각각의 요소들의 의미를 살펴보면 다음과 같다.

- 'Puzzle': 시도자가 이를 풀어서 그 'Solution'을 응답자에게 보내게 함으로서 서비스 거부(DoS) 공격을 방어하기 위한 것이다.
- 'DH^+_R': 시도자가 Diffie-Hellman[9] 세션키 생성을 위한 응답자의 공개매개변수이다.
- 'HIP_Transform': 4-way 핸드쉐이크 과정에서 HOST-ID 암호화에 필요한 알고리즘 및 HMAC 해쉬 알고리즘을 협상하기 위한 것이다.
- 'ESP_Transform': IPsec ESP에서와 동일하게 사용되는 암호화 및 인증 알고리즘을 협상하기 위한 것이다.
- 'HOST_ID$_R$': 응답자의 신원(즉, 공개키)이다.
- 'SIG$_R$': R1메시지를 응답자의 개인키로 전자 서명함을 의미한다.

상기의 R1 메세지를 시도자가 수신하면 다음과 같은 행위를 한다.

① 시도자는 응답자의 공개키(HI)를 획득하여 R1 메시지의 인증 및 무결성을 검사한다.
② 시도자는 메시지 내에 있는 'Puzzle'을 풀어서 'Solution'을 생성한다.
③ 응답자로부터 전송된 Diffie-Hellman 공개키 (DH^+_R)를 사용하여 세션키를 생성한다.

[3-way] 시도자는 I2 메시지를 생성하여 응답자에 게 전송한다.

I2 : {HIP(SPI, Solution, DH^+_I, HIP_Transform, ESP_Transform, En[HOST_ID$_I$],HMAC)SIG$_I$}

I2 메시지를 구성하는 각각의 요소들의 의미를 살펴보면 다음과 같다.

- 'SPI(Security Parameter Index)': 시도자가 응답자에게 IPsec ESP SA 식별을 위한 것이다.
- 'Solution': 'Puzzle'을 해결한 결과 값이다.
- 'DH^+_I,': 응답자가 Diffie-Hellman 세션키 생성을 위한 시도자의 공개매개변수이다.
- 'HIP_Transform', 'ESP_Transform': R1메시지가 보내온 인증 및 암호화 알고리즘을 선택한 목록 값이다.
- 'En[HOST_ID$_I$]': 시도자 호스트신원(HOST-ID)을 'HIP_Transform'에서 선택한 암호 알고리즘으로 암호화한 것이다. 이때 생성된 비밀키로는 Diffie-Hellman 세션키가 사용된다.
- HMAC: 'SPI'부터 'En[HOST_ID$_I$]'까지 해쉬한 값이다.
- 'SIG$_I$': I2메시지를 시도자의 개인키로 전자서명한다.

상기의 I2메시지를 수신한 응답자는 다음과 같은 행위를 한다.

① 응답자는 시도자의 공개키(HI)를 취득하여 I2메시지의 전자서명을 검증한다. 이를 통해서 I2 메시지의 인증 및 무결성 유무를 판단할 수 있다.
② 시도자가 보낸 'Solution'의 답이 맞는지 확인하여 서비스거부 공격 유무를 확인한다.
③ 시도자로부터 전송된 Diffie-Hellman 공개키 (DH^+_I)를 사용하여 세션키를 생성한다.
④ 생성한 세션키와 'HIP_Transform'에서 선택한 암호 알고리즘으로 'En[HOST_ID$_I$]'를 복호화하여 시도자의 신원(즉, HOST-ID)을 얻는다.
⑤ DNS로부터 취득한 시도자의 호스트 신원(HI)과 비교하여 이 메시지를 보낸 시도자를 검증한다.
⑥ 이러한 과정을 통해서 시도자와 응답자의 상호 인증이 완료되면 'SPI'와 'ESP_Transform'을 사용하여 IPsec ESP SA을 설정한다.

[4-way] 응답자는 R2메시지를 생성하여 시도자에게 보낸다.

R2 : {HIP(SPI, HMAC)SIG$_R$}

R2 메시지를 구성하는 각각의 요소들의 의미를 살펴보면 다음과 같다.

- 'SPI': 응답자가 시도자에게 보내는 IPsec ESP SA 식별을 위한 값이다.
- 'HMAC': 'SPI'와 세션키를 사용'HIP_Transform'에서 선택한 해쉬 알고리즘을 통해 해싱한 값이다.
- 'SIG_R': R2메시지를 응답자의 개인키로 전자서명 한 것이다.

상기의 R2메시지를 수신한 시도자는 다음과 같은 행위를 한다.

① 응답자의 공개키(HI)로 전자서명을 확인하여 *R2*메시지의 인증 및 무결성을 검사한다.

② 시도자에 'SPI'와 세션키를 사용하여 'HIP_Transform'에서 선택한 해쉬 알고리즘을 통해 해싱한 'HMAC' 값을 생성하여 응답자가 보내온 'HMAC' 값과 비교한다.

③ 일치하면 메시지 인증이 종료되고 서로 간에 'SPI'를 사용하여 IPsec ESP SA설정이 완료된다.

이후의 통신은 BEET 모드의 IPsec ESP에 의해 안전한 통신이 이루어진다.

Ⅳ. HIP에서 HI와 IP주소의 동적사상 기술

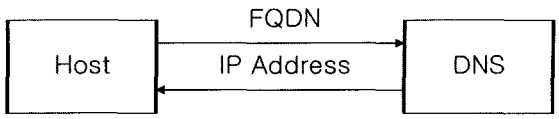
제 Ⅲ장에서 기본적인 HIP 구조와 동작절차에 대해서 살펴보았다. 하지만 이러한 HIP이 현실세계에 구현되기 위해서 해결되어야 기술로 HI와 IP주소를 어떻게 동적으로 사상할 것인가? 하는 것이다.

현재는 DNS를 이용한 방안, X.509 디렉토리 서비스를 이용하는 방안, 랑데부 서버를 이용하는 방안 등이 IETF hip WG에서 인터넷 드래프트 문서로 거론되고 있다. 이장에서는 DNS와 랑데부 서버를 이용하는 방안에 대해서 설명하였다.

4.1 DNS를 이용한 HI와 IP 주소간의 동적사상

4.1.1 DNS의 구조 및 동작절차

DNS는 도메인 이름에 대응하는 IP주소 획득을 목적으로 탄생하였다. 기본적인 동작절차는 그림 6과 같이 질의와 응답이란 클라이언트/서버 패러다임으로 동작한다.

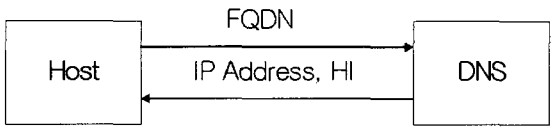


〔그림 6〕 DNS의 Look-up기능

4.1.2 HIP 적용 시 DNS의 구조 및 동작절차

HIP에서 핵심은 HI와 IP간에 동적사상 문제이다. 즉, 동적사상정보를 어디에 유지하고 이를 응용에서 어떤 절차에 따라 동적사상 할 것인가이다. 한 가지 방안으로 기존의 DNS를 활용하는 방안이다. 즉, 기존의 DNS의 사상정보인 {DN, IP주소}를 {DN, HI, IP주소}로 확장하는 방안이다. 이를 위해서 DNS자원을 정의하는 새로운 레코드로 호스트 신원 자원 레코드(HIPHI RRs)를 제안하고 있다.[10]

HIP을 위해 DNS 자원파일을 확장하면 DNS는 그림 7과 같이 동작이 가능하다.



〔그림 7〕 확장된 DNS의 Look-up 기능

그림 7에서 보듯이 응용은 기존의 DNS와 동일하게 FQDN(Fully Qualified Domain Name)에 해당하는 HI와 IP주소 획득을 위해 확장 DNS에게 질의한다. 이 질의를 받은 확장 DNS는 응용에게 FQDN에 해당하는 HI와 IP주소를 응답한다. 이를 수신한 응용은 HI를 전송계층에 보내어 소켓 식별자를 생성하고 생성된 HI를 HIL에 전송한다. 이를 수신한 HIL은 HI에 해당하는 IP주소를 동적으로 사상시켜 결정된 IP주소를 IP전송 계층으로 보내어 경로설정이 이루어지도록 하고 있다. 이와 같이 확장 DNS를 활용한 동적사상 방안은 다음과 같은 특징을 가진다.

첫째, 이 방안은 응용이 기존의 도메인 이름을 중심으로 통신함을 전제로 한 것이다. 따라서 응용이 통신하고자 하는 상대노드의 DN은 알지 못하면서 HI를 알고 있을 경우와 IP주소를 알고 있을 경우에 어떻게 동작할 것인지에 대해서는 보다 깊은 연구가 필요하다.

둘째, 호스트의 IP주소 혹은 HI가 변경되면 반드시 자신의 DNS에 이 내용을 반영해야 한다. 따라서 현재의 DNS가 정적중심의 등록절차에서 동적중심의 등록 절차(Dynamic DNS)[11]로 변경되어야 한다. 즉, 호스트가 주체가 되어 자신의 IP주소가 변경되면 자동적으로 DNS를 수정할 수 있는 능력을 가져야함을 의미한다.

셋째, 호스트들의 주소가 자주 변경되는 환경에서

확장 DNS가 감당해야 하는 부하의 문제이다. 즉, 모든 호스트들이 HI와 IP간의 동적사상 행위를 확장 DNS를 중심으로 행하기 때문에 그 부하를 무시할 수 없다. 따라서 부하를 분산시키는 방안이 마련되어야 한다.

넷째, DNS의 안전성에 관한 문제이다. 즉, 안전한 DNS(즉, DNSSEC, Domain Name Server Security Extensions)[12]가 전제되지 않으면 HIP을 활용한 통신은 불가능하다.

다섯째, 현재의 DNS는 DN과 IP간의 사상속도를 증진시키기 위해 DNS 캐쉬를 호스트 DNS 리솔버, 로칼 DNS등 많은 곳에 사용하고 있다. DNS 캐쉬는 IP주소가 등록되면 특정 기간 동안 변하지 않음을 전제로 하고 있다. 하지만 HIP을 적용한 확장 DNS는 IP주소가 자주 변경됨을 전제로 하고 있다. 따라서 DNS 캐쉬 체계가 이를 반영하여 재정립될 필요가 있다.

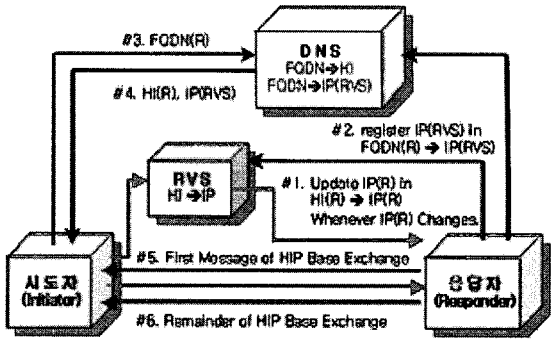
여섯째, 호스트들이 통신 중에 IP주소를 변경한 경우 변경한 호스트가 변경된 IP주소를 확장 DNS 및 통신 상대에게 알려줘야 한다. 전자의 경우에는 동적 DNS 수정기능을 활용하여 가능하다. 후자의 경우에는 REA매개변수를 통해서 상대 노드에게 이를 반영하는 방안이 IETF hip WG에서 제안되고 있다.[13]

상기에서 살펴본 HIP을 위한 확장 DNS 특징들 중에서 IP주소가 자주 변경되는 호스트 이동성과 멀티호밍 환경에서 확장 DNS부하를 경감시키는 방안이 IETF hip WG에서 제안되었다.[10] 즉, HIP호스트의 HI와 IP주소간의 동적 사상정보를 랑데부 서버(RVS, Rendezvous Server)에 저장하는 방안을 말한다.[14] 이 구조 및 동작절차를 살펴보면 다음절과 같다.

4.1.3 확장 DNS 부하를 경감시키는 랑데부서버 기술

먼저 이 구조의 구성요소는 확장 DNS와 랑데부서버이다. 여기에서 확장 DNS 구성요소에 등록되는 내용이 이전의 확장 DNS와는 다르다. 즉, 호스트가 이동 및 멀티호밍일 경우에는 확장 DNS에 이 호스트의 IP주소가 등록되지 않고 랑데부서버의 IP주소가 등록된다. 이때 기존의 DNS에 랑데부 서버의 IP주소정보를 정의하는 랑데부 서버 리소스 레코드(RVS Resource Record)가 필요하다. 그리고 랑데부서버 구성요소에는 이동성 및 멀티호밍 호스트의 HI와 IP주소로 이루어진 동적 바인딩 정보가 등록된다.[14]

이러한 구조에서 두 개의 HIP노드간의 통신절차를 살펴보면 그림 8과 같다.



〔그림 8〕 랑데부서버 구조 및 동작절차

〔이동/멀티호밍 호스트가 IP주소 변경 시 절차〕

#1. 응답자가 IP주소가 변경되면 랑데부서버(RVS)에 동적사상정보인 〈HI, IP주소〉를 등록한다.
#2. 응답자는 DNS에 자신의 〈HI, IP주소〉를 등록한 랑데부서버의 IP주소를 등록한다.

〔이동/멀티호밍 호스트간의 통신절차〕

#3. 통신 시도자가 응답자와 통신을 위해 DNS에 FQDN을 질의한다.
#4. 질의를 받은 확장 DNS는 응답자의 HI와 응답자의 정보가 등록되어 있는 랑데부(RVS) IP주소를 응답한다.
#5. 시도자는 응답자로부터 수신한 랑데부서버 IP주소를 사용하여 랑데부 서버에게 HIP Base Exchange I1 메시지를 전송한다. I1 메시지를 전송받은 랑데부서버는 시도자가 보내온 HI에 대응되는 응답자의 현재 IP주소를 찾아서 이 주소로 I1 패킷을 중계한다.
#6. 랑데부 서버로부터 I1 메시지를 전달받은 응답자는 시도자와 직접 나머지 HIP Base Exchange 과정을 수행한다.

랑데부서버를 사용할 경우 그 구조와 동작절차에 보듯이 확장 DNS의 부하가 경감됨은 분명하다. 즉, 이동 및 멀티호밍 호스트의 IP주소가 변경될 때마다 확장 DNS를 방문하여 이를 등록할 필요는 없다. 단순히 자신의 동적바인딩정보를 저장하는 랑데부서버의 IP주소가 변경될 경우에만 확장 DNS를 방문하여 랑데부서버의 IP주소를 등록하면 된다. 이동 및 멀티호밍 호스트는 자신의 IP주소가 변경되면 자신의 랑데부서버에 이를 등록함으로써 동적바인딩 정보를 완성한다.

type	length
SPI	
Address Lifetime	
P	Reserved
Address	

〔그림 9〕 REA 파라미터 구조

결과적으로 확장 DNS가 부담해야하는 통신 부하를 랑데부서버로 분산시킨 것이다.

4.1.4 HIP 통신 중 IP주소 변경 시 동적 바인딩 기술

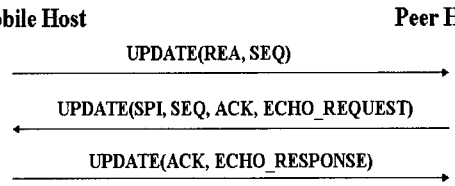
HIP 호스트의 IP주소가 변경되었을 경우 변경된 주소를 수신자에게 정확하게 전달하는 것이 REA 파라미터의 주 목적이다. 따라서 이러한 환경을 지원하기 위해서는 기존 HIP 구조의 확장이 필요하다.

즉, 이동성 및 멀티호밍 지원을 위한 방안으로 REA (REAddress) 메시지가 새롭게 제안되었다. 이 REA는 호스트의 주소가 변경되거나 새로운 주소가 추가되었을 경우 호스트의 주소를 수신자에게 전송하기 위한 메시지이다. 다음 그림 9는 REA의 파라미터 구조이다.

그림 9의 "SPI"는 기존 통신에서 사용된 SPI 값을 의미한다. 이 값에 의하여 수신자는 송신자를 검증을 하게 되고 도달 가능성을 검증한다. 이 SPI 값을 받은 수신자는 이 메시지를 다시 송신자에게 보내어 검증을 한다. "Address Lifetime"은 새로운 주소의 사용가능 시간을 명시한다. "P"는 새로운 주소의 메시지 전송이 가능한지 유무를 알려주게 된다. 만약 1로 세팅 되면 이것은 Preferred 주소로서 통신이 가능하다는 의미이다. 반대로 0일 경우는 정상적인 통신 상태가 아님을 의미한다. "Address"는 송신자의 변경된 주소를 탑재한다.

위에서 설명한 REA 메시지의 활용방법으로 HIP의 기본 메시지(UPDATE, NOTIFY, R1, I2)에 피기백 방식이 제안되었다. 드래프트 문서에서는 REA를 기본적으로 UPDATE 메시지에 탑재하여 전송하는 3가지 방법론과 R1, I2 메시지에 탑재하는 방법에 대하여 기술되어 있다. 먼저 UPDATE 메시지에 탑재하는 방법으로는 HIP의 기본교환 방법인 4-way 핸드쉐이크 가 이루어진 이후에 주소가 변경되었을 경우를 전제로 하고 있다.[13]

그림 10과 같이 첫 번째 방법은 REA메시지에 의한 주소 교환만을 제안하고 있다. 기존 HIP의 협상이



〔그림 10〕 REA 활용: 변경된 주소 교환

안전하다는 근거로 주소가 변경되었어도 기존에 협약된 SPI는 그대로 유지한다는 것이다.

V. 향후 HIP 전개 방안

향후 전개될 HIP기술을 예측하기 위해서는 HIP기술이 기존의 인터넷에 적용됨으로써 변화되어야 할 내용을 정리하고 검토 할 필요가 있다.

첫째, 응용측면에서의 변화를 생각할 수 있다. 기존의 인터넷은 상대방과 통신을 할 때 소켓을 사용하였다. 따라서 응용인지가 〈IP주소, 포트번호〉 쌍에 의해 이루어졌다. 하지만 HIP기술에서는 응용인지를 〈HI, 포트번호〉 쌍에 의해 이루어지도록 변경되었다. 이는 응용이 통신하고자 하는 호스트의 위치(즉, IP주소)에 독립적임을 의미한다.

둘째, 인터넷 인프라측면에서 변화를 생각할 수 있다. HIL가 TCP계층과 IP계층사이에 추가됨에 따라 라우팅을 중심으로 한 하위계층의 변화는 없다. 즉, HIP이 적용됨에도 불구하고 기존의 인터넷 통신 인프라를 그대로 사용할 수 있음을 의미한다. 하지만 종단단말기 인프라는 전면적으로 HIP을 지원하는 통신 S/W가 탑재되어야 한다.

셋째, 응용이 통신하고자하는 호스트 신원(HI)을 획득하는 방법과 획득된 호스트 신원(HI)과 이 호스트가 실제로 위치한 IP주소를 바인딩시키는 메커니즘이 새롭게 추가된 변화를 생각할 수 있다. 이러한 변화는 〈HI, IP주소〉 정보를 모든 사용자가 공유할 수 있는 저장소가 필요함을 의미한다. 이는 도메인 이름을 중심으로 통신을 하는 기존의 통신 패러다임의 변화를 요구한다. 하지만 이에 대해 현재의 제안은 도메인이름을 그대로 사용하면서 HIP을 적용하는 방안이 표준안으로 제시되고 있다.[10]

상기와 같은 변화를 중심으로 HIP전개될 방향을 예측하면 다음과 같다.

첫째, 기존의 인터넷을 전면적으로 일거에 HIP으로 대체할 수 는 없다면 이는 불가능한 일이다. 따라서 기존의 인터넷 인프라 변화를 최소화하면서 HIP

의 특성을 반영하여 이점이 있는 서비스 중심으로 HIP이 전개될 것으로 예측된다. 이는 특정 집단의 주도하에서 국지적 영역에서 이러한 HIP전개가 시도될 것으로 보인다.

둘째, HIP이 전역적으로 전개되기 위해서는 현재의 도메인 이름중심의 통신을 그대로 유지하면서 HI과 IP주소간의 바인딩 문제를 기존인프라에서 수용하는 것이 첩경이 될 것이다. 현재는 기존의 DNS 확장이 주 대안으로 제안되고 있다. 따라서 기존의 DNS에 HI와 IP바인딩 정보를 유지할 수 있도록 DNS을 갱신하는 방법으로 HIP이 전개될 것이다.

Ⅵ. 결 론

앞에서 살펴 본바와 같이 HIP프로토콜은 초기단계에서 R&D가 진행되고 있는 신기술 분야이다. 본 논문에서는 이러한 신기술이 어떻게 R&D되고 있는지를 파악하여 이에 대한 대응책을 마련하고자 HIP의 출현배경, 표준화 동향, HIP프로토콜 기술 분석, 향후 HIP전개 방안 등을 중심으로 기술하였다.

이러한 HIP 기술 분석을 통해 얻은 결론은 새로운 차세대 인터넷 환경에서는 HIP의 중요성이 증대될 것이란 점이다. 이는 현재의 IP주소 중심의 인터넷 기술로는 새로운 유비쿼터스 인터넷 환경을 수용하기가 어렵다는 점이다. 이의 해결을 위해서는 호스트 신원과 호스트 위치 분리기술 중심으로 인터넷 구조 변경이 필요하다. 이를 통해서 향후 인터넷 서비스로 부각되고 있는 호스트 이동성 서비스, 멀티 호밍 서비스 그리고 인터넷 보안 서비스 등에 관련된 기술이 혁신적으로 개발될 것으로 전망된다.

이러한 HIP의 중요성이 증대되고 있음에도 불구하고 국내에서는 HIP 기술 패러다임으로의 전환에 대한 대처가 빈약한 상태이다. 이 논문이 초보적인 수준이지만 이를 통해서 이의 중요성에 대한 인식을 제고할 필요가 있다.

그리고 이 논문이 현재 진행되고 있는 초기단계의 HIP 표준화 작업에 적극적으로 참여할 수 있는 동기를 부여하여 국가적 HIP 표준화 활동을 진작시키고 향후 전개될 차세대 인터넷 기술 확보 및 선도에 중요한 기초 자료로 활용되기를 기대한다.

참 고 문 헌

[1] Jukka Ylitalo, Pekka Nikander," A new Name Space for End-Points: Implementing secure Mobility and Multi-homing across the two versions of IP", European Wireless 2004, February, 2004.

[2] Pekka Nikander, Jukka Ylitalo, Jorma Wall, "Integrating Security, Mobility and Multi-Homing in a HIP Way", in Proceedings of Network and Distributed Systems Security Symposium (NDSS '03), February, 6-7, 2003, San-Diego, CA, Internet Socie- ty, February, 2003.

[3] R. Moskowitz, P. Nikander, "Host Identity Protocol Architecture", draftietf-hip-arch-01.txt, December, 200 4.

[4] Petri Jokela, Pekka Nikander etc, "Host Identity Protocol Extended Abstract", Contributing towards already identified research areas and reporting results from complementary research projects and activities, 2002.

[5] Pekka Nikander, "Applying Host Identity Protocol to the Internet Addressing Architecture", SAINT04, 2004.

[6] P. Nikander, J. Melen, "A Bound End-to-End Tunnel (BEET) mode for ESP", draft-nikander-esp-beet-mode02.txt, June 30, 2004.

[7] Kent, S., "IP Encapsulating Security Payload(ESP)",draft-ietf-ipsec-esp-v3-05 (work in progress), April 2003.

[8] R. Moskowitz, P. Nikander, P. Jokela (editor), T. Henderson, "Host Identity Protocol",draft-ietf-hip-base-00.txt, June 11, 2004.

[9] D. Eastlake, "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", Standards, RFC-2539, March 1999.

[10] P. Nikander, J. Laganier, "Host Iddentity Protocol (HIP) Domain Name System (DNS) Extensions", draftietf-hip-dns-00.txt, October, 2004.

[11] Vixie, P. Thomson, S. Rekhter, Y. and J. Bound, "Dynamic Updates in the Do-

main Name System (DNS UPDATE)", RFC 2136, April 1997.
[12] James M. Galvin, John Gilmore. "Domain Name System Security Extensions(DNSSEC)". RFC2535, March 1999.
[13] P. Nikander, J. Arkko, T. Henderson, "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-ietf-hip-mm-00.txt, October, 2004.
[14] J. Laganier, L. Eggert, "Host Identity Protocol(HIP) Rendezvous Extensions", draft-ietf-hip-rvs-00.txt, October, 2004.

〈著 者 紹 介〉

이 윤 진 (Yoon-Jin Lee)
학생회원

1996년 2월 : 배재대학교 응용수학과 학사
2000년 2월 : 배재대학교 컴퓨터공학과 석사
2003년 3월~현재: 배재대학교 컴퓨터공학과 박사과정
〈관심분야〉 정보보호, 컴퓨터/네트워크보안

조 인 준 (In-June Jo)
정회원

1982년 : 전남대학교 계산통계학과 공학사
1985년 : 전남대학교 전자계산학과 공학석사
1999년 : 아주대학교 컴퓨터공학과 공학박사
1983년~1994년: 한국전자통신연구원 선임연구원
1994년~현재 : 배재대학교 컴퓨터공학과 교수
〈관심분야〉 정보보호, 컴퓨터네트워크, 전산조직응용