Università degli Studi del Piemonte Orientale
Facoltà di Scienze M.F.N.

Corso di Laurea Specialistica in Informatica
Sistemi Avanzati e Servizi di Rete

Host Identity Protocol

Relazione per il corso di Sicurezza

Guido Vicino

Anno Accademico 2004/2005

Introduzione

Negli ultimi anni nel campo della sicurezza delle reti si sta cercando di risolvere i problemi intrinsechi portati dall'architettura TCP/IP, formalizzata e definita quando i concetti di autenticazione e privatezza dei dati non erano obiettivi importanti. Lo schema di indirizzi che permette lo scambio di dati su Internet è nato negli anni settanta e ottanta. In quel periodo il namespace era pensato per gestire computer localizzati staticamente ignorando totalmente i concetti di mobilità e sicurezza. Un esempio evidente è dato dal l'indirizzo IP al quale viene attribuito il significato di locazione ma anche quello di identità. E' facile notare come questa visione statica contrasti con l'andamento moderno dell'Information Tecnology dove la parola chiave è *Mobilità*(basta pensare al continuo crescere dell'utilizzo di hardware Wireless o alla fusione della rete telefonica con quella Internet e dove anche le piccole aziende stanno installando reti WLAN).

Un'altro difetto del "vecchio" protocollo TCP/IP è il totale disinteresse per la sicurezza del traffico dei dati, allora ritenuta non necessaria. Ai giorni nostri questo non vale più: il commercio online, gli scambi di posta tra privati o aziende, la firma digitale e molto altro necessitano di sistemi di autenticazione e protezione della privacy dei dati.

L'Host Identity Protocol mira a fornire una soluzione a parte di questi problemi, integrandosi con le tecnologie Ipv4, Ipv6 ed IPSEC.

Concetti e prerequisiti

Cosa intendiamo per mobilità?

Con mobilità vogliamo denotare la possibilità di cambiare la periferica di accesso fisco alla rete mantenendo la nostra identità (si faccia attenzione che non intendiamo la semplice differenza tra indirizzo fisico e indirizzo IP, ma la possibilità di spostare le nostre transazioni attraverso reti ethernet, reti

wireless, gprs etc). Questo tipo di mobilità viene chiamata *mobilità topologica*. Allo stato attuale delle cose ogni volta che cambiamo periferica di accesso alla rete, cambia anche la nostra identità ossia il nostro indirizzo IP, senza discernere tra locazione e identità dell'utilizzatore.

Attacchi alla sicurezza e indirizzi

Le tipologie di attacco relative all'ambiguità tra indirizzi e identità sono riassumibili in due categorie:

- Address stealing
- Address flooding

Con *address stealing* o *address forwarding* intendiamo quelle tecniche in cui l'attaccante invia messaggi falsi che riportano il cambiamento(non realmente avvenuto) dell'indirizzo di una determinata macchina. In tal maniera notificando un finto aggiornamento dell'indirizzo, l'attaccante riesce a deviare il flusso di pacchetti verso una destinazione a lui fruttuosa. Tramite questa tecnica sono possibili attacchi di tipo *denial-of-service* e *man-in-the-middle(MiTM)*.

Con *address flooding* intendiamo quelle tecniche in cui l'attaccante invia falsi messaggi a più macchine connesse alla rete con un finto indirizzo sorgente. In questi messaggi l'attaccante “avvisa” che il suo indirizzo è cambiato in quello della vittima. In tal maniera sono possibili diversi attacchi di tipo *denial-of-service(DoS).*

In Figura 1 sono riportarti questi due attacchi in maniera schematica:

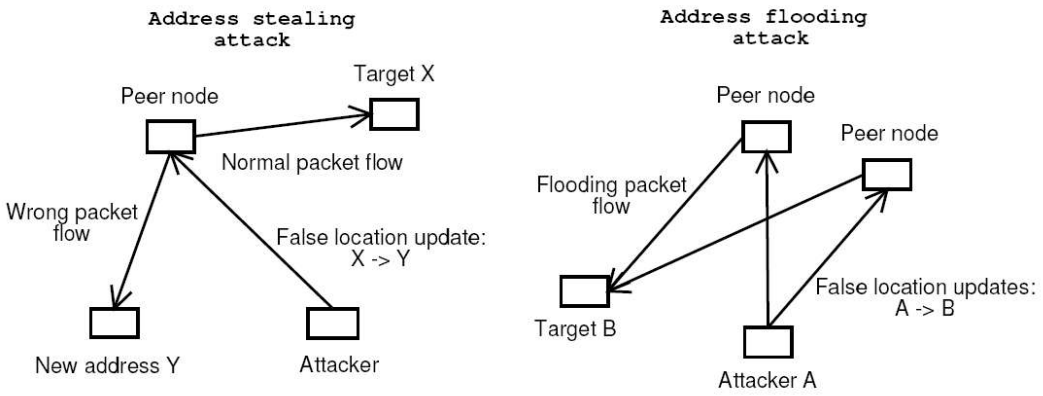

Figura 1: Esempi di attacchi riguardanti la mobilità.

IPSec, crittografia e autenticazione **dell'IP**

L'Host Identity Protocol (HIP) non è l'unica soluzione che è stata proposta per risolvere i problemi di sicurezza relativi all'architettura TCP/IP. Inoltre lo scopo di HIP è fornire identità alle macchine in rete e non si preoccupa di gestirne l'autenticazione e la protezione crittografica dei dati. La principale standardizzazione orientata verso questi argomenti è definita e implementata con la tecnologia conosciuta sotto il nome di IPSec. In questa relazione ne riporto un introduzione in quanto HIP sfrutta alcune capacità di questa tecnologia.

Cos'è IPSec?

IPSec significa *Ip Security* ed è un insieme di protocolli sviluppati dallo IETF per supportare lo scambio sicuro e l'autenticazione dei pacchetti a livello IP. IPSec è attualmente utilizzato ed implementato su larga scala per la creazione di Reti Private Virtuali(VPN).

IPSec fornisce due modalità di utilizzo (evidenziate in figura 2):

- Modalità di trasporto
- Modalità tunnel

Nella modalità di trasporto si crittografa solo la porzione di dati(*payload*) di

ciascun pacchetto, ma lascia l'intestazione(*header*) in chiaro.
Nella modalità tunnel vengono crittografati sia l'header che il payload.

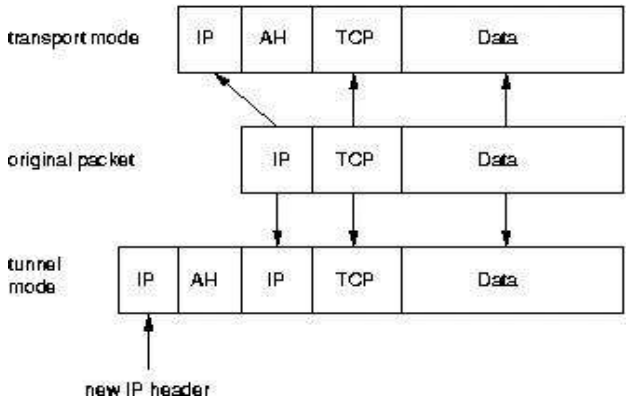

Figura 2: Transport e Tunnel mode

Security Association (SA)

Il concetto fondamentale su cui si basa IPSec è la relazione conosciuta come *Security Association* (SA, associazione di sicurezza). Un'associazione è una relazione a senso unico fra un mittente ed un destinatario, che rende disponibili i servizi di sicurezza per il traffico corrente.
Una SA è identificata da tre parametri:

1. *Security Parameters Index(SPI)*: una stringa di bit assegnata all'associazione che viene trasportata nelle intestazioni per abilitare il sistema destinatario a selezionare la SA da cui elaborare il pacchetto ricevuto.
2. *IP Destination Address:* L'indirizzo del punto terminale di destinazione dell'SA che potrebbe essere un utente o un apparato di rete.
3. *Security Protocol Identifier:* Una stringa indicante se l'associazione è di tipo AH o ESP, due protocolli che ora andremo a spiegare.

Autentification Header (AH)

Il protocollo che fornisce il supporto necessario ad assicurare l'autenticazione dei pacchetti viene identificato con il nome di

Autentification Header (AH, intestazione per l'autenticazione). L'AH protegge l'integrità del datagramma IP, calcola un HMAC(Hash Message Authentication Code) del pacchetto in base ad una chiave privata, al payload e alle parti di intestazione immutabili(come ad esempio gli indirizzi IP). Quindi viene aggiunto l'header AH all'intestazione del pacchetto.

Encapsulated Security Payload (ESP)

Il protocollo ESP serve a combinare autenticazione e crittografia, infatti garantisce sia l'integrità di un pacchetto sia la confidenzialità della trasmissione. E' grazie ad esso che possiamo incapsulare e crittografare il nostro payload e realizzare insieme ad AH le due modalità di trasporto e di modalità tunnel.

Internet Key Exchange (IKE)

AH ed ESP non si preoccupano dello scambio delle chiavi e presumono che le due parti si siano già accordate tra loro creando una Security Association (SA), ovvero si sia già deciso quali meccanismi di sicurezza usare e con quali chiavi. Il compito di negoziare e gestire le chiavi per creare la Security Association è affidato al protocollo di *Internet Key Exchange*.

HIP: Descrizione

L'Host Identity Protocol ha come scopo la proposta d'introduzione di un nuovo spazio di nomi (namespace), l'*Host Identity Namespace* e l'inserimento di un nuovo livello di protocollo, l'*Host Identity Protocol*, che andrà ad inserirsi tra gli attuali livelli di di rete e trasporto.

L'attuale modello presenta due spazi di dominio principali:

- Indirizzi *Internet Protocol*
- Nomi *Domain Name Services*

Nonostante queste due soluzioni abbiano funzionato efficacemente per lungo tempo, possiedono un grande numero di debolezze. Moskowitz nelle

specifiche [1] di questo nuovo protocollo riassume così il problema:

> *Semantic overloading and functionality extensions have greatly complicated these namespaces.*

L'obiettivo è quindi quello di separare i concetti di *Identità* e *Locazione*.

Host Identifiers

L'identità delle parti viene gestita in questo nuovo protocollo dagli *Host Identifiers(HI)*. Un HI dev'essere un nome che è statisticamente e globalmente unico, in maniera da identificare univocamente una parte. Questo però non è abbastanza, si richiede che l'HI sia di natura crittografica, infatti si utilizzerà obbligatoriamente a tal scopo una chiave pubblica di una coppia di chiavi asimmetriche come identificatore. Ogni host dovrà avere almeno un Host Identitifier ma si suppone che ne abbia più di uno, l'importante è che non ci siano due host con lo stesso identificatore. Un HI potrà essere pubblico, indicizzato ad esempio su DNS o su altre directory pubbliche al fine di garantirne l'autenticità da una terza parte.

Bisogna sottolineare la differenza tra *Host Identity* e *Host Identifier*. Con la prima si intende l'entità astratta da identificare, mentre con la seconda si indica la stringa di bit concreta da usare nel processo di identificazione.

Host identity Tag e Local Scope Identity

Abbiamo detto che gli Host Identifiers saranno implementati come chiavi pubbliche, e solo chi possiede la chiave privata potrà confermare la propria "identità". In quanto chiavi crittografiche si potranno avere dimensioni variabili, questo complicherebbe l'implementazione del protocollo. Per risolvere questo problema sono presentate due vie alternative:

Host Identity Tag(HIT):

Si crea un hash di 128 bit a partire dall'originale Host Identity, ciascun HIT dovrà essere unico, nel caso avvenisse una seppure statisticamente rara collisione, l'identità verrà verificata tramite l'utilizzo della chiave pubblica

originale.

Local Scope Identity(LSI):

E' una rappresentazione a 32 bit dell'Host Identity. Ha come scopo il poter mantenere la compatibilità con le applicazioni e le API esistenti che fanno uso del vecchio IPV4(che possiede appunto indirizzi a 32 bit). Dato che una stringa di così pochi bit possiede il difetto di avere un tasso relativamente alto di collisione, l'LSI verrà utilizzato solo in un contesto locale e generato casualmente.

Architettura HIP

Nell'architettura corrente i livelli di trasporto e di Internetworking sono connessi direttamente, ed una sessione di trasporto viene identificata tramite un indirizzo IP e un numero di porta. Con la nuova architettura le connessioni a livello trasporto saranno identificate invece dall'Host Identity e da un numero di porta. Lo scopo è naturalmente rendere più facile la mobilità.

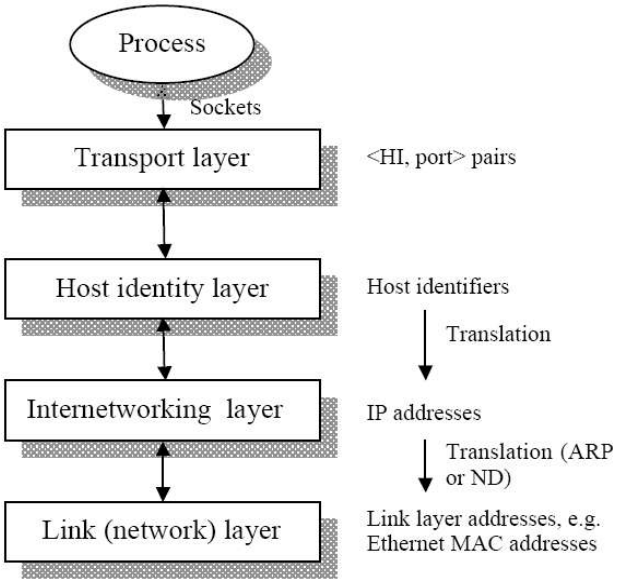

Figura 3: La nuova architettura HIP

Andiamo a spiegare la nuova architettura tramite l'ausilio dello schema

rappresentato in Figura 3. Ciascun host è rappresentato dal suo HI e i livelli superiori non vedranno gli indirizzi IP attuali. Localmente ogni HI è mappato ad un indirizzo IP. Quando i pacchetti lasciano l'host, il corretto instradamento è deciso e vengono inseriti i corrispondenti indirizzi IP. La sessione e lo scambio dei messaggi viene gestito con un alto uso di elementi crittografici.

Sessione HIP

L'instaurazione di una sessione HIP viene creata tramite lo scambio di quattro messaggi, un *four-way handshake*. Durante questo scambio, viene utilizzato Diffie-Hellman per creare una chiave di sessione e stabilire una coppia di associazioni SA tra i nodi (IPSec ESP Security Association).

Le associazioni ESP SA tra i due hosts vengono legate alle Host Identities in maniera da garantirne la privatezza e sicurezza. In ogni caso, i pacchetti che viaggiano sulla rete non contengono l'informazione sull'attuale HI, ma il pacchetto all'arrivo viene identificato e associato utilizzando il *SecurityParamenter Index(SPI)* dell'intestazione IPSec. La struttura dei nuovi pacchetti viene mostrata in Figura 4.

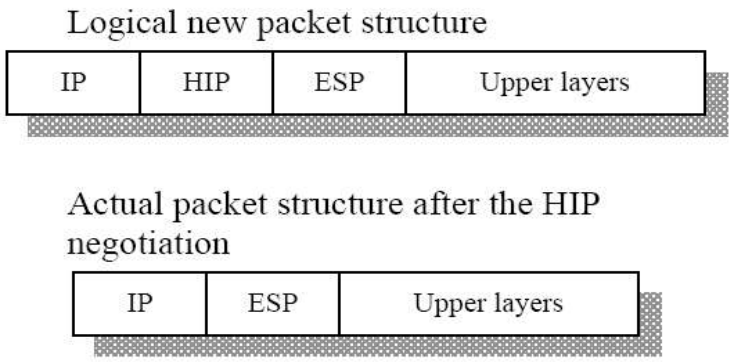

Figura 4: Il nuovo pacchetto HIP

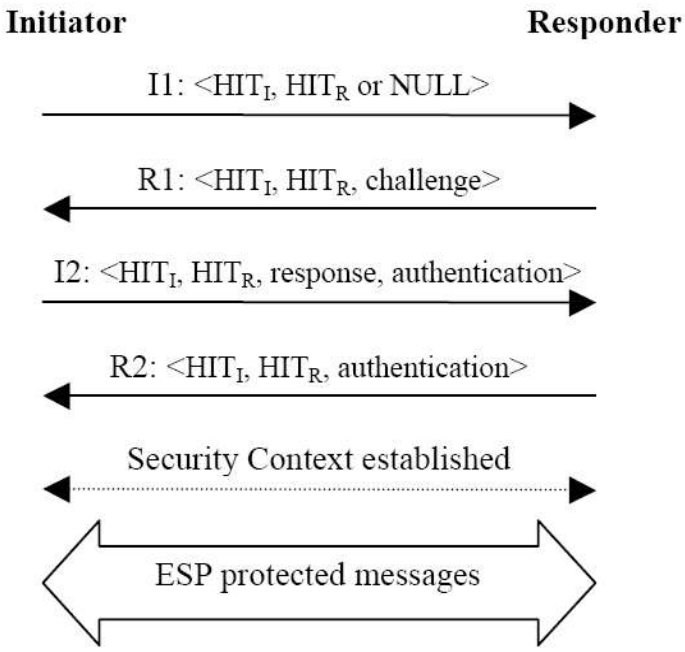

Figura 5: La sessione HIP

Lo scambio a quattro vie mostrato in figura 5 utilizza Diffie Hellman per lo scambio delle chiavi. Le parti vengono chiamate a loro volta *Initiator* (colui che chiede lo scambio) e *Responder* (colui che risponde alla richiesta di scambio.

Descriviamo i quattro pacchetti:

1. *Pacchetto I1:* Questo primo pacchetto viene spedito dall'Initiator al Responder ed è semplicemente una richiesta di inizio comunicazione. Contiene gli HITs dell'Initiator e del Responder.
2. *Pacchetto R1:* Il Responder invia questo pacchetto in risposta al pacchetto I1 dell'Initiator. L'R1 contiene l'HI e la chiave Diffie Hellman del Responder. Le informazioni riguardanti gli algoritmi crittografici e le modalità ESP supportate dal Responder vengono qui incluse. Questo pacchetto può contenere un *puzzle*, che è un indovinello crittografico dato da risolvere all'Initiator affinché lo scambio di pacchetti avvenga con successo, questo servirà per rendere la comunicazione resistente agli

attacchi DoS. Infine il pacchetto R1 viene firmato con la chiave privata del Responder.

3. *Pacchetto I2:* Viene spedito dall'Initiator e contiene l'HI crittografata, il Diffie Hellman, una possibile risposta all'indovinello, e le opzioni preferite per l'ESP. Anch'esso viene crittografato con la chiave privata.
4. *R2 packet:* Quest'ultimo pacchetto è inviato dal Responder *se e solo se* il pacchetto I2 conteneva la risposta corretta all'indovinello, se no viene scartato l'I2.

Oltre a questi pacchetti vengono aggiunti altri pacchetti addizionali per la gestione dei cambi d'indirizzo di locazione e per la gestione degli SPI.

HIP e Mobilità

Un host mobile ora può cambiare la propria locazione all'interno della rete di accesso, attraverso le varie tecnologie o i vari indirizzi IP mantenendo la propria identità. Un sistema mobile è considerato tale quando il suo indirizzo cambia dinamicamente a causa di diverse ragioni quali PPP, DHCP, prefissi Ipv6, traduzioni NAT e vari assegnamenti.

HIP supporta anche soluzioni *Multi-homed* ossia quei sistemi che supportano più percorsi paralleli di comunicazione, quindi più indirizzi IP allo stesso tempo. Il disaccoppiamento permetterà di mantenere le connessioni anche dopo un cambiamento di indirizzo.

Meccanismo di Rendezvous

Contattare un nodo mobile richiede una certa difficoltà. Il mittente deve sempre sapere come raggiungere il destinatario. Una soluzione potrebbe essere l'uso di sistemi di *Dynamic DNS* al fine di aggiornare le locazioni associate ai vari *HIP*. Ma esistono dei problemi che non possono essere risolti in tale maniera. Ad esempio nel caso di nodi che cambiano il loro indirizzo contemporaneamente, oppure nel caso in cui il Responder cambi la propria locazione prima che la connessione venga stabilita con

l'handshaking a quattro vie. Per risolvere questi problemi necessitiamo di un agente di *packet forwarding* chiamato *rendervous server*. In tal maniera il Responder aggiornerà la propria locazione sul *Directory Address* facendola puntare al server di rendezvous che conterrà costantemente aggiornato l'indirizzo del Responder. Questa funzionalità non è stata però ancora bene formalizzata ma gli autori promettono in breve una sua definizione.

HIP: Protezione dagli attacchi

L'Host identity Protocol come precedentemente detto, mira a risolvere i problemi riguardanti gli attacchi che sono basati sul furto d'indirizzo e sul flooding d'indirizzo.

Attacchi Denial-of-service

Gli attacchi di tipo *Denial-of-service* si basano sul fatto che chi inizia una connessione ha un "costo" in termini di protocollo inferiore rispetto a chi risponde alla richiesta di connessione. HIP mira a ribaltare questa situazione, obbligando chi richiede la connessione ad avere un maggior carico lavorativo. Questo viene realizzato facendo in modo che sia il Responder a iniziare l'handshaking a tre vie, costruendo il protocollo HIP su quattro pacchetti. Per fare questo, il secondo pacchetto diventa un pacchetto *puzzle* che il Responder può riutilizzare quante volte vuole in risposta al primo pacchetto. La durata e l'uso di questo pacchetto sono scelti a seconda del livello di "paranoia" e throughput che vogliamo stabilire per il nostro sistema. L'attaccante quindi verrà dissuaso dal procedere ad un attacco di questo tipo.

Questa soluzione da un lato può fornire un sistema di protezione contro i DoS ma dall'altra può introdurne di nuovi. Descriviamone un primo esempio. Una volta che l'Initiator ha risolto l'indovinello, potrebbe inviare

pacchetti I2 multipli con finti indirizzi IP sorgente e dei payload e firme HIP non valide. Nel tentativo di interpretare questi pacchetti I2, il Responder potrebbe sovraccaricarsi e cadere. La difesa contro una simile tipologia di attacchi è scartare i pacchetti I2 dopo un certo numero di pacchetti errati riscontrati.

Una seconda forma di DoS nasce dall'introduzione di “stati di connessione” fornita dall'HIP. HIP prevede uno scenario nel quale una delle due parti caduta per problemi tecnici possa recuperare il suo precedente “stato” di associazione con l'altra parte(ad esempio nel caso di crash, o nel caso si cambi punto di accesso alla rete). Un host che si sta riavviando dovrebbe spedire un I1 alla sua parte, la quale risponderà con un R1 anche se precedentemente si trovava nello stato ESTABLISHED(ossia dopo aver instaurato una precedente associazione HIP non chiusa). Camuffando I1, il pacchetto R1 deve risultare inaspettato al ricevente e deve essere scartato al fine di evitare possibili attacchi.

Un terzo attacco possibile poteva prevedere la chiusura non autorizzata di una associazione HIP, ma viene scongiurato tramite l'utilizzo di HMAC nei pacchetti di chiusura dell'HIP, (CLOSE/CLOSE_ACK). Un altro problema però nasce dalla presenza di un campo opzionale per l'invio dei pacchetti *ICMP Parameter Problem* che potrebbe permettere all'attaccante tramite IP spoofing di inviare pacchetti di chiusura dell'associazione(*CLOSE)* errati e causare *reflection attacks*(tipologia di attacchi in cui i dati trasmessi vengono mandati indietro alla vittima, in questo caso i messaggi d'errore).

Un altro possibile attacco consiste nell'indurre l'Initiator a risolvere puzzle stantii causando la ripetizione dei pacchetti R1 facendo perdere la

sincronizzazione con il Responder. Per risolvere questo problema è stato introdotto un contatore di generazione degli R1 che aumenta monotonicamente per proteggersi da questi attacchi.

Attacchi Man-in-the-middle

E' difficile difendersi da attacchi *man-in-the-middle* senza una terza parte autentificatrice. Un attaccante preparato può gestire tutte le parti di HIP, ma quest'ultimo fornisce indirettamente la seguente protezione da una simile tipologia di attacco. Se l'HI del Responder è recuperata tramite un DNS fidato, un certificato o tramite un altro mezzo sicuro, l'Initiator può usare questo per validare il pacchetto R1. A sua volta il Responder può validare il pacchetto I2 recuperando l'HI firmato dell'Initiator garantendo una sessione sicura. In tal maniera si ha la sicurezza di una comunicazione tra parti fidate. Il problema è dato dal fatto che comunque l'Initiator può decidere di utilizzare un HI anonimo(anche perché non sempre ci si potrà affidare a sistemi garanti per ogni singola connessione), sta a ciascuna parte decidere a seconda del livello di sicurezza desiderato di non accettare tale livelli di anonimato.

HIP ed IpsEC

Bisogna sottolineare come HIP non sia pensato come rimpiazzo ad IPSec, ma come un sistema di gestione delle identità che collabori con esso al fine di fornire i criteri basi di privacy, autenticazione e sicurezza che sono ormai richiesti sulla rete.

Conclusione

Abbiamo visto come HIP fornisca un supporto migliore per la mobilità e per la sicurezza che l'architettura originale TCP/IP. La prima è fornita tramite il disaccoppiamento dei livelli di trasporto e Internetworking, la seconda

tramite l'uso di un'architettura mirata alla crittografia. I problemi riguardanti una simile architettura sono sicuramente legati al carico computazionale che la crittografia richiede, specialmente nel caso di sistemi portatili, dove la connessione verrà a risultare decisamente più lenta. Sicuramente però i campi di applicazioni sono molti e questa proposta mira a soddisfare e a riempire una delle più gravose falle intrinseche nel paradigma TCP/IP.

Riferimenti

[1]: R. Moskowitz, P. Nikander, "Host Identity Protocol Architecture" Internet Draft, IETF 2004, draft-ietf-hip-arch-02.

[2]: R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, "Host Identity Protocol", Internet Draft, IETF 2005, draft-ietf-hip-base-02.

[3]: P. Jokela, P. Nikander, J. Melen, J. Ylitalo, J. Wall "Host Identity Protocol – Extended Abstract" in *Proceedings of WWRF8bis (electronic),* Beijing, China, February 26-27, 2004.

[4]: K. Kostiainen "Host Identity Payload for Mobility and Security", Helsinki University of Technology, Department of Computer Science and Engineering, 2003.