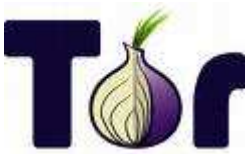


UNA TECNOLOGIA PER LA COMUNICAZIONE RISERVATA E ANONIMA IN RETE

Approfondimento per il corso di sicurezza dei sistemi informatici

Carmine Carella

Settembre 2006

Indice

1. INTRODUZIONE

L'approfondimento tratta un argomento che rientra nella sicurezza delle reti, viene presentato uno strumento di difesa contro una particolare minaccia alla riservatezza dei messaggi: l'analisi del flusso del traffico.
Lo strumento in questione ha come scopo generale il raggiungimento di uno dei tre obiettivi della protezione della sicurezza informatica: la riservatezza o privacy.
Ricordiamo la **definizione di Riservatezza** :
In generale la riservatezza assicura che le risorse informatiche siano accessibili solo alle parti autorizzate. Con il termine " Accesso " si intende la lettura, la visualizzazione, la stampa o banalmente la consapevolezza dell'esistenza di una risorsa.

1.1 La riservatezza in rete

In rete senza accorgimenti non esiste riservatezza.

- ✓ **Non esiste la riservatezza totale.** L'origine e la destinazione di ogni comunicazione ed anche il contenuto sono identificabili presso ogni nodo di rete intermedio con elementari teniche di analisi del traffico.
- ✓ **Nessuno è completamente libero di esprimersi.** La pubblicazione sul web è facilmente censurabile attaccando per via informatica o legale un singolo server.
- ✓ **Nessuno è anonimo.** Ogni connessione ad un provider comporta l'archiviazione di ora e numero di telefono in un file di log.

La riservatezza e quindi anche la tecnologia TOR che deve garantire tale obiettivo, utilizzzano in modo massiccio la crittografia.

1.2 Comunicazione anonima

Una sorgente (S) spedisce un messaggio ad un destinatario (D) attraverso un mezzo (M).
Esiste anonimato in avanti (Forward) di M se: nessuno (neanche D) può conoscere l'dentità di S.
Esiste anonimato all'indietro (Backward) di M se: nessuno (neanche S) può conoscere l'dentità di D.

L'argomento che di seguito viene trattato è: TOR un sistema di comunicazione anonima per internet, basato sulla seconda generazione del protocollo onion routing.

1.3 Onion routing – cenni

L'Onion Routing è una tecnica per comunicazioni anonime a bassa latenza in una rete di computer, sviluppata da David Goldschlag, Michael Reed, and Paul Syverson. Si basa sul principio di Chaum del mix networks. Con l'onion routing i pacchetti TCP vengono incapsulati in strutture dati, dette onion, che sono ripetutamente cifrate e instradate attraverso nodi successivi parte di un circuito virtuale.

1.3.1 Mix networks

Le Digital mixes, conosciute anche come mix networks sono state inventate da David Chaum nel 1980. Le Digital mixes creano comunicazioni difficili da tracciare utilizzando una catena di server proxy (che nell'ambito dell'onion routing prendono il nome di onion router) che instradano il messaggio in un percorso imprevedibile. Ogni messaggio dalla sorgente alla destinazione è criptato, la criptazione a chiave pubblica avviene con tutte le chiavi pubbliche dei server che deve attraversare a partire dall'ultimo; il messaggio risulta essere criptato a strati, con il messaggio originale nello strato più interno e negli strati superiori tutte le criptazioni. Ciascun server proxy riceve il messaggio, toglie il proprio livello di crittografia (decifra) per scoprire dove il messaggio deve essere spedito successivamente e lo inoltra al server successivo. La crittografia impedisce ad un avversario di comprendere il contenuto dei messaggi.

1.4 Obiettivi di tor

Tor è uno strumento per migliorare la sicurezza e la protezione su Internet delle persone e delle aziende. Tor rende anonimi la navigazione e la pubblicazione su internet, l'instant messaging, IRC, SSH e altro ancora.
Tor ha lo scopo di proteggere contro l'analisi del traffico, una forma di sorveglianza della rete che minaccia la privacy e l'anonimato personali, i rapporti e le attività d'affari confidenziali, la sicurezza dello stato. Con Tor le comunicazioni vengono indirizzate attraverso una rete distribuita di server, chiamati onion router, che proteggono l'utente dalla profilazione fatta dai siti web, o da intercettazioni locali che, controllando il traffico dei dati, possono capire quali siti vengono visitati.

1.5 TOR: Un progetto in continuo sviluppo e miglioramento

La sicurezza di Tor migliora quanti più utenti lo usano e e quanti più volontari offrono di gestire un server. Si può contribuire volontariamente offrendo il proprio tempo oppure offrendo la propria banda. Tor è sofware sperimentale e non è consigliabile affidarsi all'attuale rete Tor se si ha realmente bisogno di anonimato forte. Sicurezza e usabilità non devono escludersi a vicenda: se l'usabilità di Tor aumenta, attrarrà più utenti, che aumenteranno le possibili sorgenti e destinazioni di ogni connessione, aumentando di conseguenza la sicurezza di ciascuno.

2. TOR : PANORAMICA

2.1 A chi e’ rivolto

Tor è una rete di tunnel virtuali che permette a utenti e gruppi di aumentare la privacy e la sicurezza in Internet. Consente inoltre agli sviluppatori di software di creare nuovi strumenti di comunicazione con caratteristiche intrinseche di privacy. Tor fornisce le basi per una gamma di applicazioni con cui organizzazioni e singoli individui possono condividere informazioni su una rete pubblica senza compromettere la propria privacy.

2.2 Esempi di possibili contesti di utilizzo

Tor può essere usato dai singoli per impedire che i siti web analizzino e profilino loro e i loro familiari. Possono utilizzarlo per connettersi a risorse bloccate dal loro fornitore di connessione internet, come ad esempio siti di informazioni o servizi di messaggistica. I servizi nascosti di Tor permettono di pubblicare siti web ed altri servizi senza rivelare la collocazione reale del sito. Con Tor i giornalisti possono comunicare in modo sicuro e riservato con le proprie fonti e con dissidenti. I collaboratori di una organizzazione non governativa (ONG) possono usare Tor per collegarsi al sito web della casa madre mentre prestano servizio in un paese straniero, senza che si sappia necessariamente per chi lavorano.

2.3 Dove è stato sperimentato

Gruppi come Indymedia raccomandano Tor per preservare la privacy e la sicurezza dei loro membri. Alcune grandi aziende usano Tor per condurre in modo sicuro analisi della concorrenza, o per proteggere dalle intercettazioni i loro fornitori e partner strategici. Queste aziende se ne servono anche per sostituire le tradizionali VPN, che rivelano con precisione le quantità e i tempi dei dati scambiati tra le sedi. Dove si lavora fino a tardi? In quale ufficio gli impiegati consultano siti di ricerca di lavoro? Quali divisioni di ricerca comunicano con l'ufficio brevetti aziendale? Un ramo della Marina degli Stati Uniti usa Tor per la raccolta di intelligence di pubbblico

dominio, e una delle sue squadre se ne è servito in una recente missione in Medio Oriente. Le autorità giudiziarie usano Tor per visitare o sorvegliare siti web senza lasciare nei log dei webserver i loro indirizzi IP governativi, o come misura di sicurezza nelle operazioni sotto copertura.

2.4 TOR per combattere una forma di sorveglianza : l’analisi del traffico

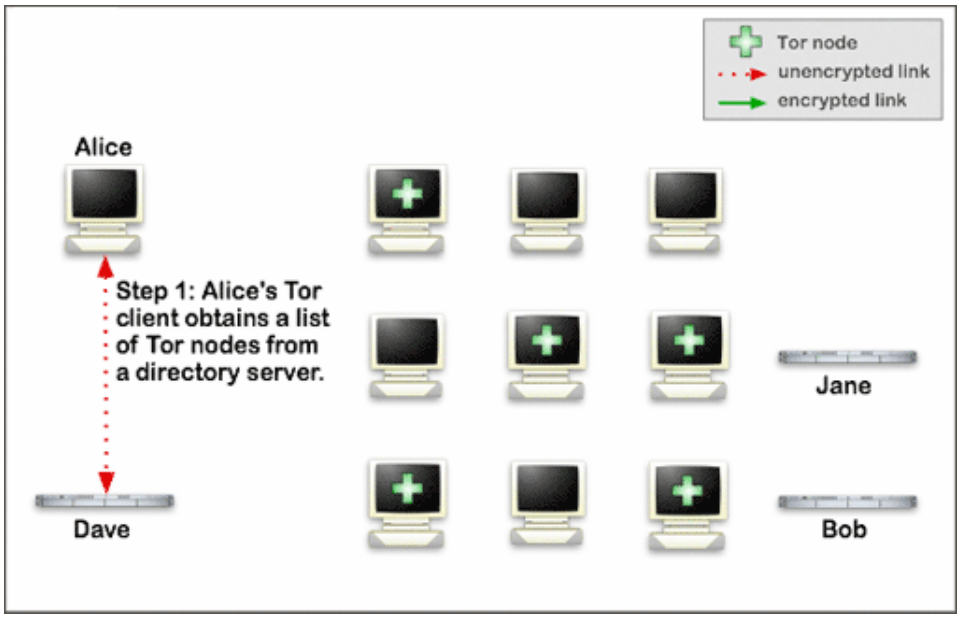
Tor protegge da una comune forma di sorveglianza in rete chiamata "analisi del traffico". L'analisi del traffico può essere usata per capire chi sta parlando con chi in una rete pubblica. La conoscenza della sorgente e della destinazione del proprio traffico Internet permette infatti ad altri di ricostruire le nostre abitudini e i nostri interessi personali. Questo tipo di analisi può anche mettere in pericolo il proprio lavoro e l'integrità personale, rivelando chi si è e da dove ci si connette. Per esempio, se si viaggia all'estero e ci si connetti ai computer aziendali per controllare la posta, si può inavvertitamente rivelare la propria nazionalità, la propria origine e professione a chiunque stia osservando la rete, anche se le connessioni eseguite sono criptate.

2.4.1 Come funziona l’analisi del traffico

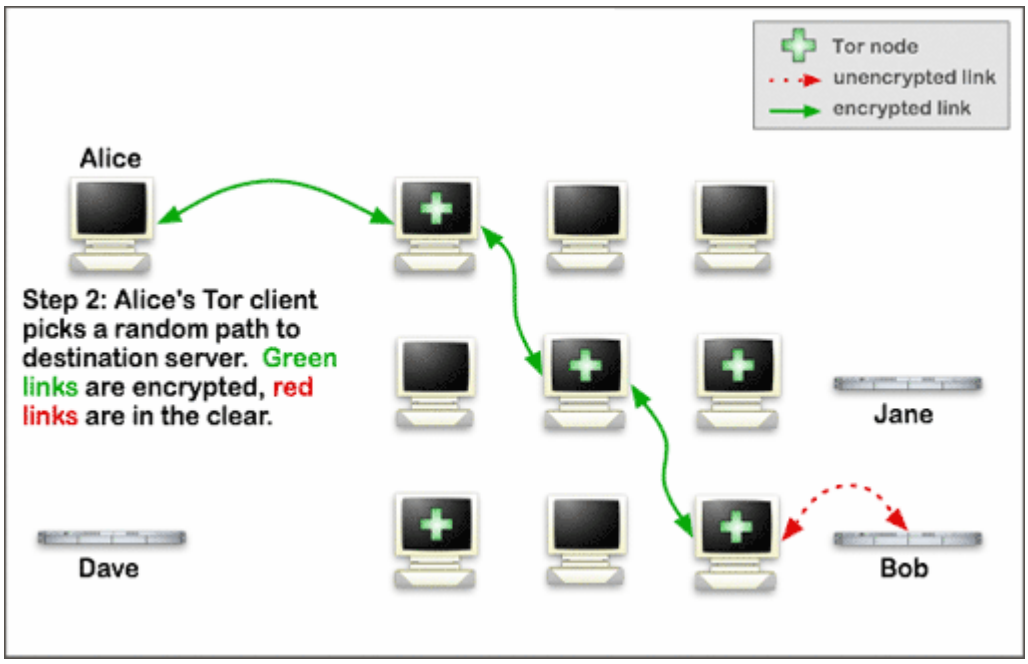
I pacchetti dati di internet sono divisi in due parti: il blocco dati e l'intestazione, che viene utilizzata per l'instradamento dei pacchetti. Il blocco dati contiene le informazioni che vengono inviate, siano esse una email, una pagina web o un file musicale. Anche se il blocco dati viene criptato, l'analisi del traffico continua a rivelare informazioni su quello che si sta facendo e, possibilmente, su quello che si sta dicendo. Questo perché l’analisi del traffico si concentra sull'intestazione del pacchetto dati, che fornisce sorgente, destinazione, dimensione e tempi. contenuto del traffico Internet (il blocco dati), e non le intestazioni dei pacchetti.
Un problema basilare per coloro che sono attenti alla privacy è che il destinatario di una comunicazione può sapere, attraverso l'analisi dell'intestazione del pacchetto, chi lo sta mandando. Lo stesso possono fare gli intermediari che ricevono il flusso dei pacchetti, come ad esempio gli Internet Service Provider (ISP), e talvolta anche gli intermediari non autorizzati. Una forma molto semplice di analisi del traffico consiste nel porsi in un punto qualsiasi tra la sorgente e il destinatario della comunicazione, e studiare le intestazioni dei pacchetti.
Vi sono però altri e più potenti metodi di analisi del traffico. Alcuni attaccanti spiano molte parti di Internet e usano sofisticate tecniche statistiche per carpire schemi di comunicazione tra diversi individui e organizzazioni. criptare i messaggi non aiuta, in caso di un attacco del genere, poiché questo nasconde solo il contenuto del traffico internet (blocco dati) e non le intestazioni dei pacchetti.

2.5 TOR : Una rete anonima distribuita. Come funziona in generale

Tor aiuta a ridurre i rischi dell'analisi del traffico, sia semplice che sofisticata, distribuendo le transazioni attraverso molti nodi della rete Internet, in modo che nessun singolo punto possa collegare una transazione alla sua destinazione. L'idea è simile ad usare un percorso tortuoso e difficile da seguire per depistare un inseguitore, cancellando periodicamente le proprie orme. Invece di prendere un percorso diretto dalla sorgente alla destinazione, i pacchetti dati nella rete Tor prendono un percorso casuale attraverso molti server che ne coprono le tracce, in modo che nessun osservatore situato in un singolo punto possa dire da dove venga o dove sia diretto un certo traffico.

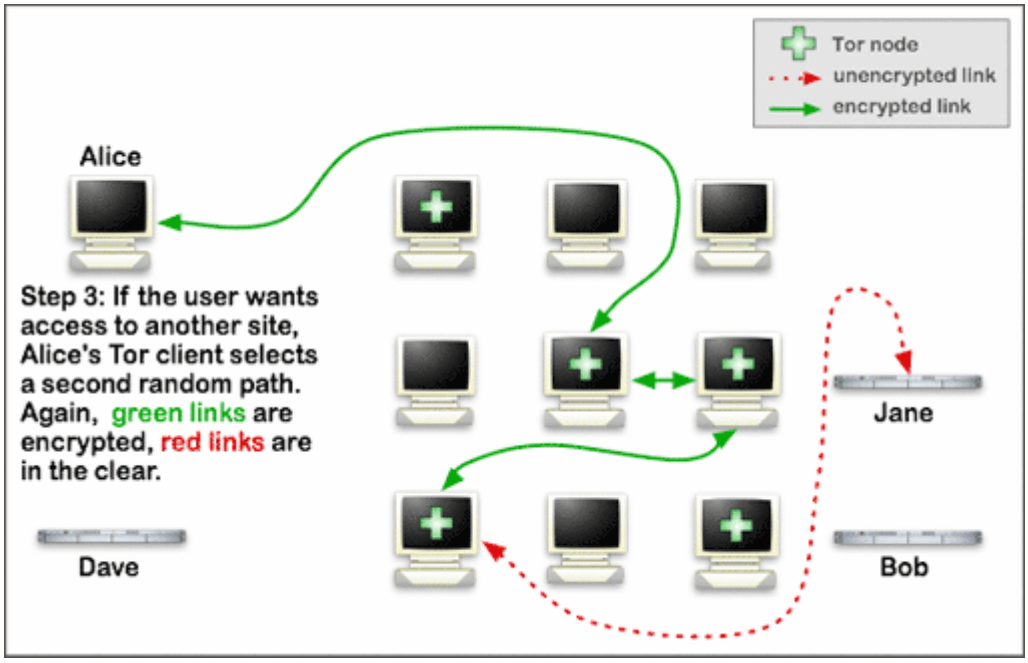


Per creare un percorso di rete privato con Tor, il software crea incrementalmente un circuito di connessioni criptate attraverso server della rete. I server prendono il nome di Onion Router. Il circuito viene esteso un salto alla volta, e ogni server lungo il percorso conosce solo che server gli ha dato le informazioni, e verso che server inoltrarle. Nessun server conosce il percorso completo che il pacchetto ha preso. Il software negozia un nuovo insieme di chiavi crittografiche per ogni salto lungo il circuito, per assicurarsi che ciascun nodo non possa tracciare queste connessioni durante il passaggio. I dati inviati vengono subito incapsulati in una sequenza di buste criptate, una per ogni specifico nodo che verrà attraversato. Ogni nodo sarà in grado di aprire unicamente la propria busta, quella più esterna, rendendo possibile l'inoltro del nuovo livello criptato al router successivo. Con questa struttura di trasmissione "a cipolla" il percorso e i dati rimangono al sicuro da occhi indiscreti.



Una volta che un circuito è stabilito, si possono scambiare diversi tipi di dati e usare molti tipi di applicazioni attraverso una rete Tor. Poiché ogni server non vede che un singolo salto nel circuito, né un intercettatore e neppure un server compromesso possono utilizzare le tecniche di analisi del traffico per collegare la sorgente con la destinazione della connessione. Tor funziona solo con i flussi TCP e può essere usato da ogni applicazione che abbia il supporto SOCKS.

Per ragioni di efficienza, Tor utilizza lo stesso circuito per le connessioni che avvengono nello stesso minuto. Le richieste successive sono fornite a un nuovo circuito, per evitare che nessuno possa collegare le azioni precedenti con le successive.



2.6 Servizi nascosti di Tor

Sistema integrato per servizi TCP anonimi in cui l'IP del server host non viene rilevato:
Alice utilizza un servizio messo a disposizione da Bob.

- ✓ Bob comunica la localizzazione del servizio nascosto ad un insieme di IP, Introduction point, e lo indicizza sui directory server.
- ✓ Alice acquisisce lo handle del servizio.
- ✓ Alice stabilisce un nodo comune RP, Rondez-vous point, per la connessione al servizio di Bob e ne annuncia la localizzazione ad uno degli IP.
- ✓ Bob riceve la richiesta dell'IP e si connette al RP stabilendo un circuito con Alice.
- ✓ Alice e Bob rimangono anonimi l'uno all'altro.

2.7 Completare la protezione di tor

Tor non può risolvere tutti i problemi di anonimato. Si focalizza solo sulla protezione del trasporto dei dati. E' necessario utilizzare software di supporto specificamente scritto per il protocollo utilizzato se non si vuole che il sito che si visita possa identificare il visitatore. Per esempio, si puo usare un proxy web come Privoxy mentre si naviga in internet per bloccare i cookie e le informazioni sul browser utilizzato.
Inoltre, per proteggere il proprio anonimato, è bene fare attenzione. Non fornire il proprio nome o altre informazioni nei moduli compilati sul web.

3. TOR – COME FUNZIONA IN DETTAGLIO: PROTOCOLLO TOR

L'onion routing è un sistema basato su una rete distribuita di nodi tra loro interconnessi che permette di rendere anonime le comunicazioni tra applicazioni TCP che accedono alla rete Internet come ad esempio un browser o un client di posta.
Ogni nodo che compone la rete prende il nome di ONION ROUTER (OR) (server) mentre ogni host che usufruisce di tale rete per veicolare anonimamente le proprie comunicazioni su internet prende il nome di ONION PROXY (OP) (client).
Quando un client desidera comunicare tramite questo sistema sceglie un percorso composto da diversi OR, dove ogni OR conosce il nodo che lo precede e quello successivo ma non conosce gli altri nodi che compongono l'intero percorso. In questo modo il client OP può inviare i propri dati cifrati lungo il percorso di OR, dove ogni OR è in grado di decifrare e quindi leggere la porzione di dati ad esso destinata mediante una chiave simmetrica.La stessa operazione viene eseguita per i restanti OR sul percorso fino a che i dati non arrivano a destinazione. I dati

vengono inviati in celle di lunghezza fissa al fine di impedire qualsiasi correllazione tra la lunghezza dei dati inviati e la loro natura.

AUTENTICAZIONE (Creazione circuito e connesione)

Prima di iniziare qualsiasi connessione il client OP si connette al directory server e scarica la lista dei nodi OR disponibili e delle chiavi di identificazione di questi ultimi. Quando un client OP si connette ad un nodo OR ,OR si autentica a OP inviandogli una catena composta da due certificati: il primo certificato usando una chiave di connessione temporanea e il secondo certificato autofirmato contenente la chiave di identificazione del nodo OR. Il campo CommonName (CN) del primo certificato è il nickname del nodo OR, mentre il CN del secondo certificato è il nickname del nodo OR seguito da uno spazio e dalla stringa "<identity>". Il client OP può verificare la validità della chiave di identificazione fornita dal nodo OR con quella scaricata dal directory server e rifiutare la connessione nel caso quest'ultima dovesse risultare non valida. Due nodi OR invece eseguono una mutua autenticazione utilizzando una procedura simile a quella descritta in precedenza. Un nodo OR deve rifiutare una connessione a/da un OR la cui chiave di identificazione non dovesse risultare valida o il cui certificato sia malformato o mancante.

TRASMISSIONE DATI

Una volta stabilita una connessione TLS tra le due entita' (OP-OR oppure OR-OR) esse incominciano a scambiarsi i dati in celle di lunghezza fissa pari a 512 byte che vengono inviate in maniera sequenziale, questo impedisce qualsiasi attacco di correlazione tra la lunghezza dei dati e la natura del traffico stesso. La connessione TLS tra OP-OR oppure OR-OR non e' permanente, una delle due parti coinvolte nella comunicazione puo' interrompere la connessione se non transitano dati per un tempo pari al KeepalivePeriod che di default vale 5 minuti.

3.1 Formato delle celle

I dati scambiati tra due entita' all'interno di un sistema di onion routing prendono il nome di celle, esse hanno una grandezza fissa di 512 byte. Ogni cella e' composta da 2 campi che hanno funzioni di controllo e dal payload che contiene i dati trasmessi dall'applicazione.

3.1.1 Formato cella generica

CircID: Serve ad indicare a quale circuito virtuale è associata la cella, infatti ogni nuova connessione effettuata da un'applicazione del client OP può essere instradata tramite un percorso di nodi OR differente.

Command: Contiene il valore relativo ad uno dei comandi supportati dalle specifiche del protocollo e può assumere i seguenti valori:

1) PADDING (Padding)
2) CREATE (Crea un circuito)
3) CREATED (Conferma creazione)
4) RELAY (Invio dati)
5) DESTROY (Chiude un circuito)
6) CREATE_FAST (Crea un circuito)
7) CREATED_FAST (Conferma Creazione)

Payload: Contiene i dati che devono essere trasmessi all'altro nodo, ma il suo contenuto può variare a seconda del Command, infatti:

- ✓ PADDING: Payload è inutilizzato
- ✓ CREATE: Payload contiene il challenge
- ✓ CREATED: payload contiene il response
- ✓ RELAY: Relay header e relay body
- ✓ DESTROY: Payload è inutilizzato

3.1.2 Tipologia celle

Le celle di PADDING vengono utilizzate per implementare il keepalive della connessione, se non c'e' traffico l'OP o l'OR invia una cella di padding ogni volta che trascorrono un certo numero di minuti. Le celle CREATE, CREATED e DESTROY sono utilizzate per gestire la creazione e la chiusura dei circuiti virtuali. Le celle RELAY sono utilizzate per inviare comandi o dati lungo il circuito virtuale.

3.1.2.1 Cella create

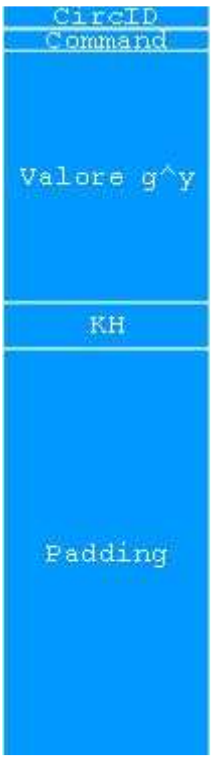
Il client OP crea il circuito in maniera incrementale, un nodo alla volta. Per creare un nuovo circuito il client OP sceglie un percorso casuale tra i nodi OR disponibili e invia una cella CREATE al primo nodo, il payload della cella conterra' la prima parte dell'handshake Diffie Hellman (ovvero g^x). I dati sono cifrati in questo modo: supponiamo che la chiave pubblica del nodo OR sia lunga L byte, se i dati da cifrare sono minori di L-42 allora i dati sono paddati con 42 byte OAEP e cifrati direttamente con la chiave pubblica, altrimenti viene generata una chiave simmetrica casuale di 16 byte che precede i dati, dopo la quale seguono i primi L-16-42 byte di

dati il tutto viene cifrato con la chiave pubblica del nodo OR, il resto dei dati e' cifrato con la chiave simmetrica precedentemente generata.

La scelta dell'algoritmo ibrido e' stata fatta per motivi di efficienza, un sistema ibrido infatti si avvale della crittografia asimmetrica per garantire uno scambio sicuro di una chiave simmetrica che verrà utilizzata per cifrare una mole di dati elevata con un'efficienza maggiore.

3.1.2.2 Cella created

Quando un nodo OR riceve una cella CREATE rispondera' con una cella CREATED il cui payload conterrà la seconda parte dell'handshake Diffie Hellman e i primi 20 byte della chiave derivata KH che serve a dimostrare la conoscenza della chiave condivisa.



Chiave derivata e altre chiavi utilizzate

Una volta che l'hanshake Diffie Hellman tra il client OP e il nodo OR e' completato entrambe le parti sono in grado di computare il valore di g^xy.

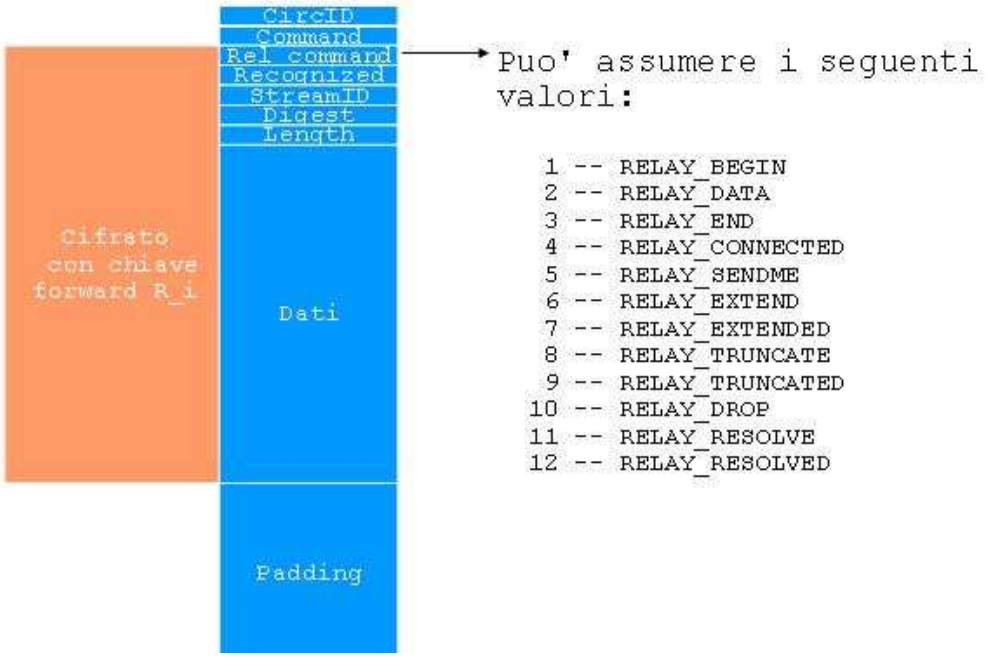
Tuttavia prima di computare tale valore entrambe le parti devono verificare che il valore di g^x o g^y ricevuto non sia degenerato, ovvero che sia rigorosamente maggiore di 1 e rigorosamente minore di p-1 dove p e' il modulo Diffie Hellman, e nel caso rifiutare la connessione.

Tale scelta viene fatta per motivi di sicurezza, se cosi' non fosse infatti un attaccante sarebbe in grado di sostituirsi al nodo OR semplicemente sostituendo il valore g^y della cella CREATED con un valore degenerato quale 0 o 1 il che renderebbe possibile il calcolo di g^xy.

KH e' utilizzata durante il processo di handshake Diffie Hellman per dimostrare la conoscenza della chiave condivisa K derivata dal valore g^xy, Df e' utilizzato come seed per la funzione hash che controlla l'integrita' del flusso dati nella direzione che va dal client OP al nodo OR, Db e' utilizzato come Df ma per il flusso dati che va nella direzione opposta, Kf e' la chiave utilizzata per cifrare il flusso dati che va dal client OP al nodo OR, e Kb e' utilizzata come Kf ma per il flusso dati che va nella direzione opposta.

3.1.2.3 Cella relay

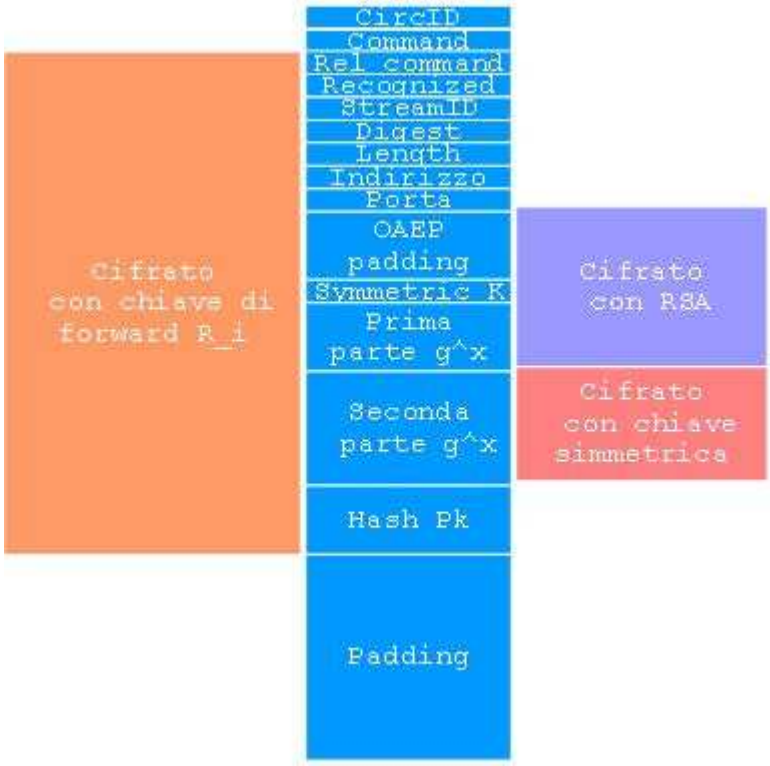
A differenza delle celle CREATE e CREATED che sono scambiate tra nodi tra loro adiacenti, le celle RELAY sono utilizzate per veicolare flussi di dati e messaggi di controllo attraverso un circuito composto da piu' nodi.



Recognized: Se tale campo risulta diverso da zero la cella decifrata viene instradata verso il prossimo nodo OR del circuito altrimenti il contenuto del payload viene processato dal nodo stesso.
StreamID: Utilizzato per riconoscere le celle che appartengono allo stesso flusso di dati.
Length: Indica la lunghezza in byte dei dati contenuti nel payload, il resto del payload è composto di byte di padding posti a zero.

3.1.2.4 Cella relay extend

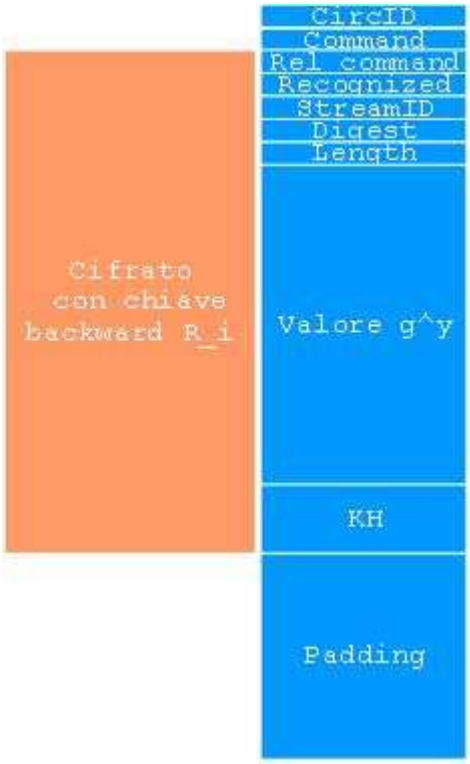
Per estendere il circuito di un ulteriore nodo il client OP invia una cella relay EXTEND che istruisce l'ultimo nodo OR ad inviare una cella CREATE per estendere il circuito verso il nodo OR desiderato.



Indirizzo e Porta : Rappresentano l'indirizzo IPv4 e la porta del prossimo nodo OR verso il quale si desidera estendere il circuito.

3.1.2.5 Cella relay extended

Quando un nodo OR processa una cella relay EXTEND invia una cella CREATE al nuovo nodo OR per estendere il circuito, il nuovo OR risponde con una cella CREATED, il payload di tale cella viene restituito al client OP all'interno di una cella EXTENDED.



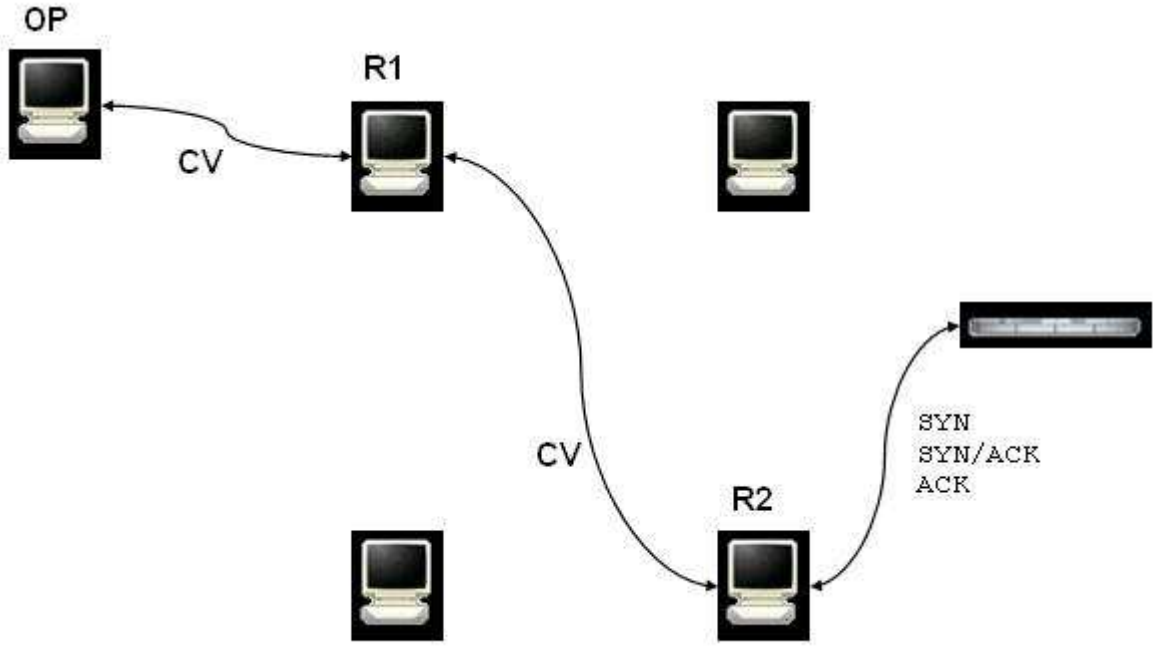
3.1.2.6 Cella relay_begin

Nel payload di questa cella sono presenti, l'hostname o l'indirizzo IP nel formato IPv4 o IPv6 e la porta dell'host a cui si desidera connettere.

La cella RELAY_BEGIN viene instradata lungo il circuito e viene processata dal nodo OR di uscita che apre una nuova connessione TCP verso l'indirizzo e la porta specificati nel payload del pacchetto.
Se la connessione TCP va a buon fine il nodo OR di uscita invia una cella RELAY_CONNECTED al client OP che lo informa dell'avvenuta connessione.

3.1.2.7 Cella relay_connected

3.2 Protocollo TOR – Passi funzionamento



3.2.1 Creazione circuito

1) OP sceglie il nodo di uscita R2.
2) Invia a R1 la cella CREATE.
3) R1 decifra nella cella Create, con la chiave privata la parte cifrata con RSA (chiave pubblica di R1) e con la chiave simmetrica la parte cifrata sempre con tale chiave.
4) R1 calcola g^{xy} e deriva K.
5) R1 invia ad OP la cella CREATED.
6) R1 calcola g^{xy} e deriva K.
7) OP confronta i primi 20 byte di K derivata con KH (di 20 byte) ricevuta.
8) Viene creato il circuito virtuale tra OP e R1.

3.2.2 Estensione circuito

1) OP invia a R1 la cella RELAY EXTEND.
2) R1 decifra con la chiave di forward la parte cifrata con la chiave di forward di R1 della cella Relay Extend.
3) R1 controlla il valore del campo RECOGNIZED = 0 e processa il contenuto della cella.
4) R1 controlla il valore del campo RELAY Command = 6 che significa RELAY_EXTEND.
5) R1 controlla l'indirizzo IPv4 e la porta del nodo a cui estendere il circuito.
6) R1 estrae l'onion skin dal payload della cella Relay_Extend.
7) R1 inserisce il payload estratto in una cella CREATE e la invia a R2.

8) R2 decifra nella cella Create, con la chiave privata la parte cifrata con RSA e con la chiave simmetrica la parte cifrata sempre con tale chiave.
9) R2 calcola g^{xy} e deriva K.
10) R2 invia a R1 la cella CREATED.
11) R1 estrae il payload dalla cella CREATED.
12) R1 inserisce il payload estratto in una cella RELAY EXTENDED.
13) R1 cifra header e payload della cella Relay Extended con la chiave di backward.
14) R1 invia ad OP la cella RELAY EXTENDED.
15) OP decifra con la chiave di backward, calcola il valore g^{xy} e deriva K.
16) OP confronta i primi 20 byte di K derivata con KH ricevuta e quindi viene esteso il circuito virtuale.

3.2.3 Creazione connessione

1) OP invia la cella RELAY_BEGIN a R1.
2) R1 decifra con la chiave di forward.
3) R1 verifica che il campo Recognized risulta diverso da zero, allora significa che la cella è ancora cifrata.
4) R1 invia la cella RELAY_BEGIN al nodo R2 e decifra con la chiave di forward.
5) R2 controlla che il campo Recognized sia = 0 e quindi processa il payload.
6) R2 control.la che il valore del campo Relay Command sia = 1 e cioè Relay_Begin.
7) R2 ricava IP e Porta dell'host verso il quale deve aprire una connessione TCP e la apre.
8) R2 invia una cella RELAY_CONNECTED a R1.
9) R1 aggiunge gli header della cella Relay_Connected con Recognized diverso da zero.
10) R1 cifra con la chiave di backward e invia la cella RELAY_CONNECTED a OP.
11) OP decifra con la chiave di backward di R1, controlla che Recognized != 0 e decifra con la chiave di backward di R2.
12) Allora OP riceve la conferma dell'avvenuta connessione TCP.

3.2.4 Trasmissione dati

Una volta che la connessione TCP viene stabilita i dati vengono veicolati sul circuito virtuale attraverso celle RELAY_DATA.

4.VULNERABILITA’ E ATTACCHI A TOR

4.1 DOS

Fornendo Tor come un servizio pubblico, questo crea molte opportunità per attacchi DOS contro la rete. Nonostante Tor metta a disposizione dei controlli di flusso e dei limiti di banda, che evitano l’utilizzo di una quantità di banda che ecceda da quella fornita dai router, le opportunità per un utente di utilizzare più risorse di rete o a rendere la rete inutilizzabile sono comunque presenti. Ci sono diversi attacchi DOS indirizzati al consumo di cpu, un attaccante può forzare un OR a eseguire un operazione di crittografia costosa: può falsificare un handshake TLS, facendo effettuare il calcolo all’OR la sua metà dell’handshake ad un costo computazionale non reale. Possono essere attaccati non solo gli OR, ma anche gli host (OP) e i link. Interrompendo un singolo circuito o link, tutti i flussi che passano lungo quella parte di circuito vengono compromessi.

4.2 Attacchi end-to-end

Utilizzo di analisi statistiche per sferrare questi attacchi.

- ✓ **Correlazioni rispetto al tempo**: Un attaccante che osserva le strutture del traffico di due utenti potrebbe correlare un mittente e un destinatario di un messaggio in base a informazioni temporali. Se l’utente che si sta osservando spedisce un messaggio e dopo un breve intervallo di tempo l’altro utente osservato riceve un messaggio con alta probabilità è possibile dedurre che sia lo stesso messaggio e quindi che i due utenti fanno parte dello stesso circuito. La principale difesa è quella di nascondere la connessione tra OP e il primo nodo Tor dietro un firewall.
- ✓ **Correlazione rispetto alla dimensione**: Il semplice conteggio dei pacchetti potrebbe confermare ad un attaccante la destinazione di un messaggio. Infatti un attaccante può osservare il volume di informazione inviato da un utente e il volume ricevuto da un altro utente e confrontare questi volumi.

4.3 Altri attacchi

- ✓ **Osservazione delle strutture di traffico dell’utente**: una connessione di un utente non rivela la destinazione o i dati, ma rivela la struttura del traffico (ricezione e spedizione) dell’utente. Comunque delineare il traffico in questo modo richiede un ulteriore sforzo ovvero più applicazioni simultaneamente o in serie spediscono flussi di informazioni su un singolo circuito quindi bisogna dividere i diversi flussi.

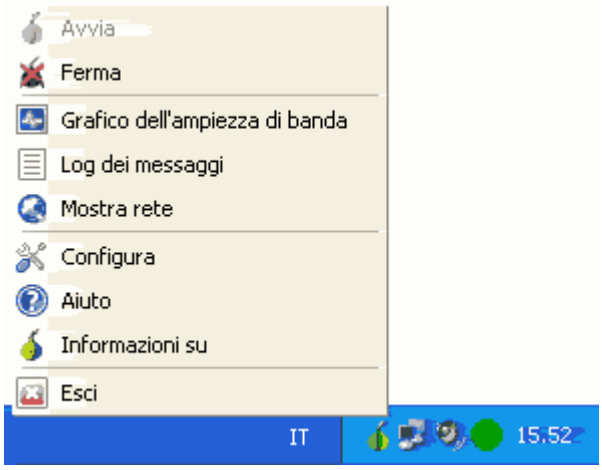
- ✓ **Opzioni di configurazione**: Agli utenti è concesso di scegliere le opzioni di configurazione, ma i meno esperti potrebbero settare opzioni che compromettono l'anonimato e questo può essere sfruttato facilmente da un attaccante.
- ✓ **Impronta digitale di siti web**: Un attaccante potrebbe costruire un database di "impronte digitali" contenente dimensioni di file, strutture d'accesso, ecc. per i siti web designati ad essere analizzati. La connessione di un utente ad un dato sito potrebbe essere confermata semplicemente consultando il database. Contro Tor questo attacco potrebbe essere meno efficiente per i multi – flussi su uno stesso circuito e per la granularità delle celle che potrebbero limitare la raccolta di impronte digitali.

5. CLIENT TOR - INSTALLAZIONE E CONFIGURAZIONE DELLA NAVIGAZIONE

Tor, arrivato alla versione 0.1.1.22, viene distribuito come software libero open source disponibile per i sistemi Windows, Mac Os X Tiger e Panther e molte distribuzioni Linux/BSD/Unix. La sua implementazione prevede solamente un'interfaccia testuale ma numerosi sono i plugin grafici che contribuiscono a renderlo più usabile. Nella descrizione che segue focalizzeremo la nostra attenzione sulla implementazione per Windows anche se molte delle considerazioni fatte saranno valide per tutti i sistemi.
Permette l'interfacciamento con protocolli applicativi tramite il protocollo SOCKS. E' scritto in C. La rete attuale consta di 200 – 250 nodi con una banda di 60 Mbit/s.

Tor è utilizzato insieme a l'interfaccia Vidalia e il proxy http Privoxy necessario per una corretta navigazione anonima. In questo caso il nostro browser utilizzerà direttamente Privoxy che a sua volta si connetterà al client Tor per accedere alla rete. Terminata l'installazione, le icone di Privoxy e di Tor saranno presenti nella Traybar. Cliccando col tasto destro su ognuna delle due icone assicuriamoci che entrambi i servizi siano avviati.

Figura 1: l'icona di controllo di Tor

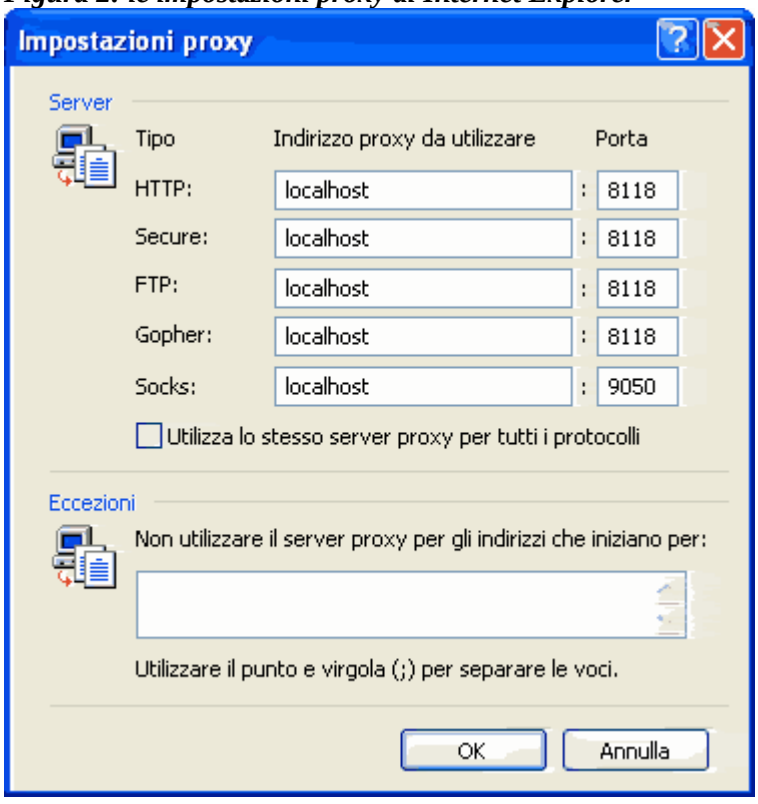


A questo punto è necessario impostare il nostro browser affinché possa usare Tor per la navigazione. Per fare questo è sufficiente imporre l'uso del proxy locale creato da Privoxy sulla porta 8118.

5.2 Internet Explorer

Se usate Internet Explorer selezionare le Opzioni Internet dal menu Strumenti e nella scheda Connessioni premete il pulsante Impostazioni LAN. presente a fine pagina. Selezionate la casella relativa all'utilizzo del server proxy e dopo aver premuto il pulsante Avanzate. inserite i dati indicati nell'immagine sottostante.

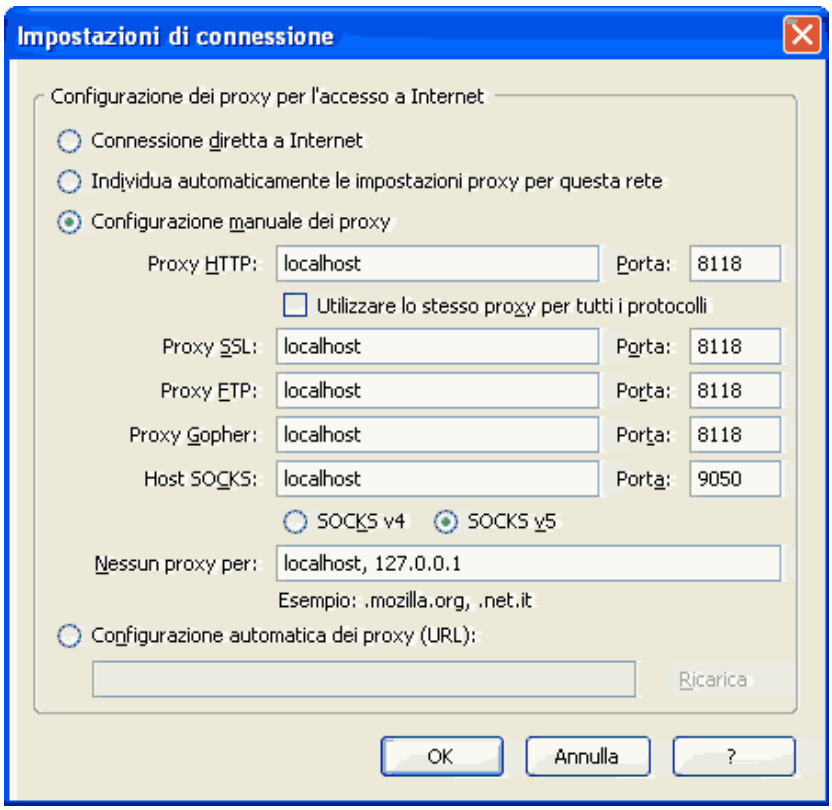
Figura 2: le impostazioni proxy di Internet Explorer



5.3 Mozilla Firefox

Gli utilizzatori di Firefox devono aprire invece Opzioni... dal menu Strumenti, selezionare la scheda Generale, premere il pulsante Impostazioni connessione e riempire le caselle nel seguente modo:

Figura 3: le impostazioni proxy di Mozilla Firefox



5.4 Verifica del funzionamento

Una volta impostato l'uso di Tor non ci resta che verificare che tutto funzioni. Digitando l'URL http://p.p nella finestra del browser possiamo testare il funzionamento di Privoxy. In questo caso dovrebbe comparire la sua pagina di configurazione. Per verificare invece che durante la navigazione venga effettivamente usata la rete Tor, e che quindi la navigazione sia anonima, possiamo visitare http://serifos.eecs.harvard.edu/cgi-bin/ipaddr.pl?tor=1. Se il responso e positivo siamo pronti per navigare in modo totalmente sicuro.

5.5 Inconvenienti nell'utilizzo di Tor

Naturalmente la navigazione risulterà rallentata dalla rete anonima. La privacy si paga in termini di latenza dovuta all'elevato numero di nodi attraverso i quali i dati devono passare,

all'appesantimento dovuto al protocollo crittografico e, ovviamente, alla congestione dei singoli onion router.

5.6 Disabilitare Tor

Per passare da una navigazione anonima ad una tradizione è necessario disabilitare manualmente l'uso del proxy locale nelle opzioni di navigazione del browser. Gli utenti di Firefox possono invece usare **TorButton**, una estensione decisamente raccomandata che permette di attivare o disattivare la navigazione anonima con un semplice click sull'apposito pulsante presente nella barra di stato.

5.7 Funzioni di Tor

Per finire, con un clic sull'icona di Tor presente nella Traybar potete accedere ad un menu con due interessanti funzioni. Grafico dell'ampiezza di banda visualizza il consumo di banda in upload e download attraverso l'interfaccia Tor. Mostra rete genera invece una mappa geografica con l'elenco di tutti gli onion router conosciuti e traccia la rotta dei pacchetti da noi inviati.
Tor dispone anche di una potete trovare la guida ufficiale all'installazione.

5.8 Privoxy e il protocollo SOCKS. Perchè utilizzarli

A questo punto ci si potrà chiedere perché si sia deciso di aggiungere un ulteriore proxy http come Privoxy alla catena di connessione. A prima vista sembra del tutto superfluo dato che qualunque browser può utilizzare direttamente l'interfaccia Socks del client Tor. Il problema è che i più diffusi protocolli Socks 4 e 5 non possono incapsulare direttamente le richieste DNS. Se utilizzassimo tali protocolli per connetterci al client Tor otterremo una anonimizzazione di tutto il traffico tranne quello relativo alla risoluzione dei nomi. Questo vanificherebbe tutti gli sforzi fatti in precedenza (no tunnelling attraverso Tor delle richieste DNS).
Il protocollo Socks 4a, pienamente supportato dal client Tor, consente la gestione dei nomi in modo nativo ma è anche molto poco diffuso. Poiché Internet Explorer e Firefox non gestiscono tale versione del protocollo si è reso necessario l'uso di Privoxy per fare da interprete.
Sfortunatamente Privoxy gestisce unicamente i protocolli HTTP e HTTPS e non può quindi essere usato con altri programmi quali client di posta, FTP, IRC o Instant messenger.
Se desiderate configurare Tor con programmi che usino protocolli diversi da http dovrete usare direttamente l'interfaccia Socks (localhost:9050). Se il programma in questione gestisce unicamente Socks 4 e 5 dovrete rassegnarvi ad una falsa anonimità. Se potete usare Socks4a risulterete totalmente invisibili. Tra i programmi che parlano Socks4a troviamo l'ottimo client FTP Filezilla.

Sul Wiki di Noreply.org potete trovare una guida di configurazione per vari programmi, prevalentemente per il mondo linux.

5.9 TORPARK: L'anonimato su una chiavetta usb

Per finire si accenna all'esistenza di Torpark, un pacchetto integrato contente Tor, Privoxy e il browser Firefox per l'ambiente Windows. La particolarità di Torpark è che tutte e tre le applicazioni non necessitano di installazione per essere lanciate e che le stesse salvano dati unicamente nella propria cartella. Installando Torpark su una chiavetta USB è quindi possibile disporre di uno strumento di navigazione anonimizzante portatile utilizzabile in ambienti critici come Internet Cafe o luoghi di navigazione pubblica.

Abbiamo, quindi illustrato come si installa e utilizza il software Tor su una macchina client. Per contribuire al miglioramento di TOR è possibile configurare la propria macchina come **SERVER TOR**, non riportiamo qui le istruzioni di installazione ma si rimanda alla documentazione all’indirizzo http://tor.eff.org/docs/tor-doc-server.html.it.