

# Anonimato in Rete

# Definizioni

- **Anonimato**

Lo stato di non identificabilità di uno specifico soggetto in un insieme di utenti.

Ha essenzialmente due finalità:

- **Nascondere la reale origine** di un attacco o di una specifica azione effettuata attraverso la rete

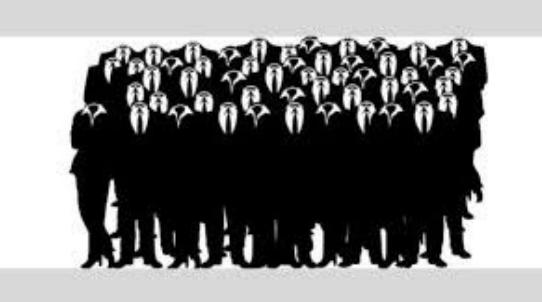
- **Garantire la Privacy**

Diritto di poter scegliere cosa della nostra vita privata può essere divulgato (tracciamento, e-voting etc.)



# Finalità

- Persone che vogliono aggirare la **censura** statale
- Persone che vogliono partecipare in modo anonimo a discussioni dagli **argomenti sensibili** o che non vogliono subire l'opinione degli altri
- Aziende che non vogliono pubblicizzare alcune relazioni
- Persone che non vogliono essere oggetto di **profiling** commerciale
- Persone che vogliono offrire servizi senza poter esser localizzati
- Autorità giudiziare che non vogliono esser identificate come tali durante attività investigative



# Ma anche..

- Persone che vogliono organizzare **truffe** online e rimanere impuniti
- Persone che visitano siti dai contenuti illegali, ad esempio pedopornografici
- Criminali che devono comunicare con altri criminali
- **Criminalità organizzata** che vede nuove frontiere di business
- **Tutte le tecnologie** possono esser usate per scopi criminali, ma questo non è un problema tecnico
- L'uso di strumenti di anonimizzazione non introduce nuovi tipi di abusi



# Lo scopo è ...

- **La privacy degli individui sia rispettata**
  - Possa essere anonimo se lo decido
  - Essere tecnicamente certo che i miei dati siano visibili solo a chi li concedo con il mio consenso
- **I criminali siano catturati**
  - Gli organi di Pubblica Sicurezza (e solo loro!) abbiano strumenti per identificare i malfattori
  - Abbiano strumenti per rilevare e produrre prove di reati criminosi



# Stato attuale...

Il **diritto alla privacy** e' un diritto fondamentale sancito dal **Testo Unico della tutela dei dati personali**. Ma su Internet:

## Non esiste privacy

- L'origine e la destinazione di ogni comunicazione e spesso anche i contenuti sono identificabili con tecniche elementari

## Non esiste anonimato

- Ogni persona che accede ad internet è identificata e registrata

## Nessuno è libero di esprimersi

- La pubblicazione e fruizione di contenuti è facilmente censurabile (vedi Cina e <http://www.rsf.org/24h/map.php>)

**Quindi internet è tutt'altro che anonima!**

# Anonimato in rete

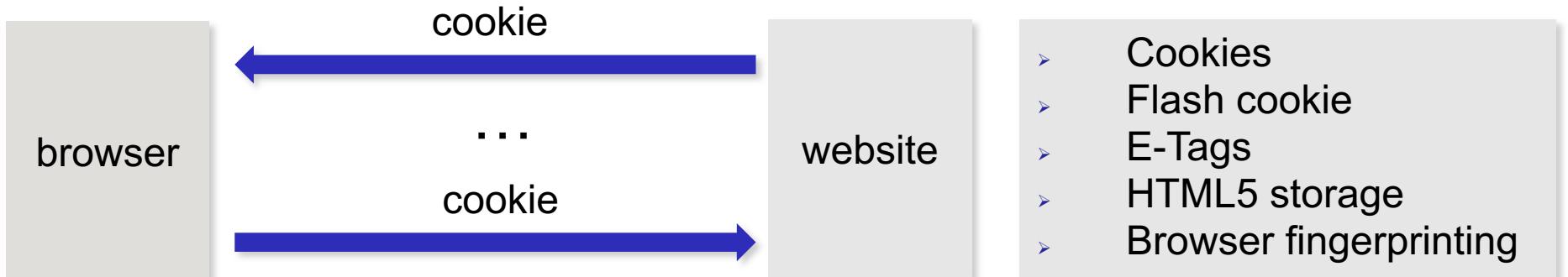
- Quindi, senza usare particolari accorgimenti la rete è in grado di garantire l'anonimato?

**Assolutamente no!**



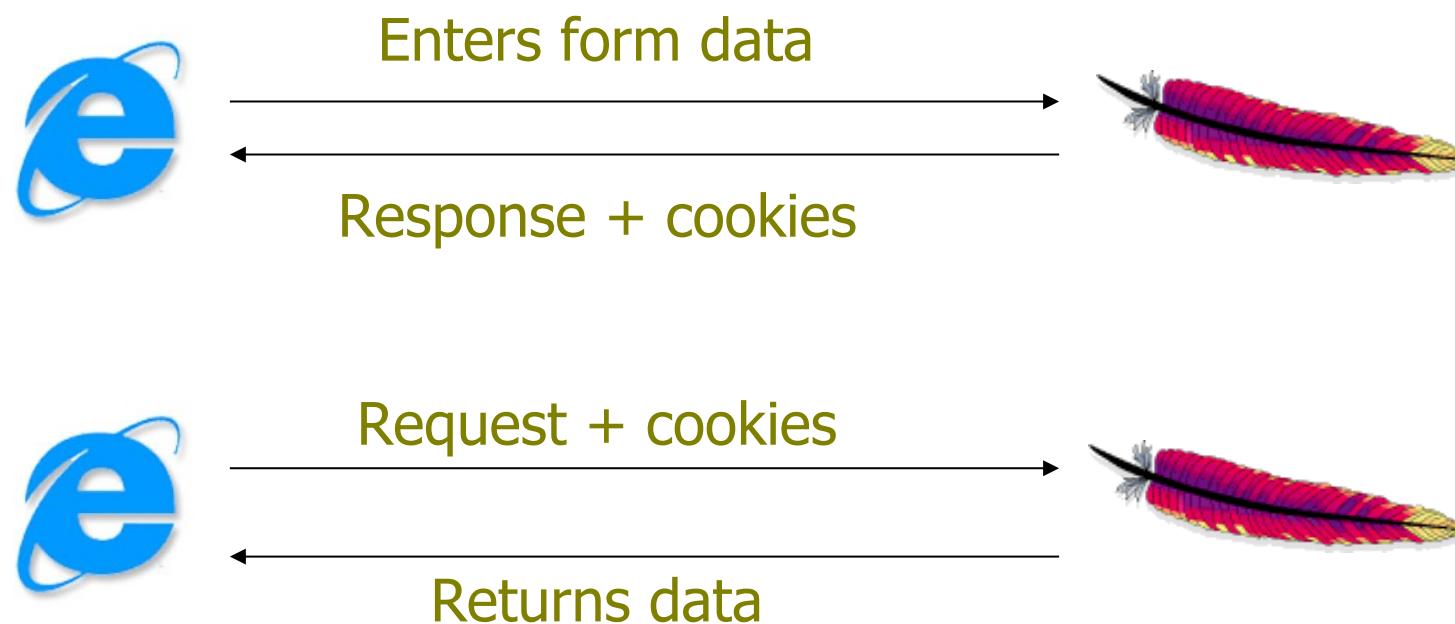
# Evidenze digitali in rete

- Indirizzo IP, facilmente tracciabile a ritroso
  - La possibilità di spoofing rende concettualmente anonimo il singolo IP
  - Il routing della rete di appartenenza ci riconduce univocamente alla localizzazione
- Tracciabilità browser
- Profilazione utente



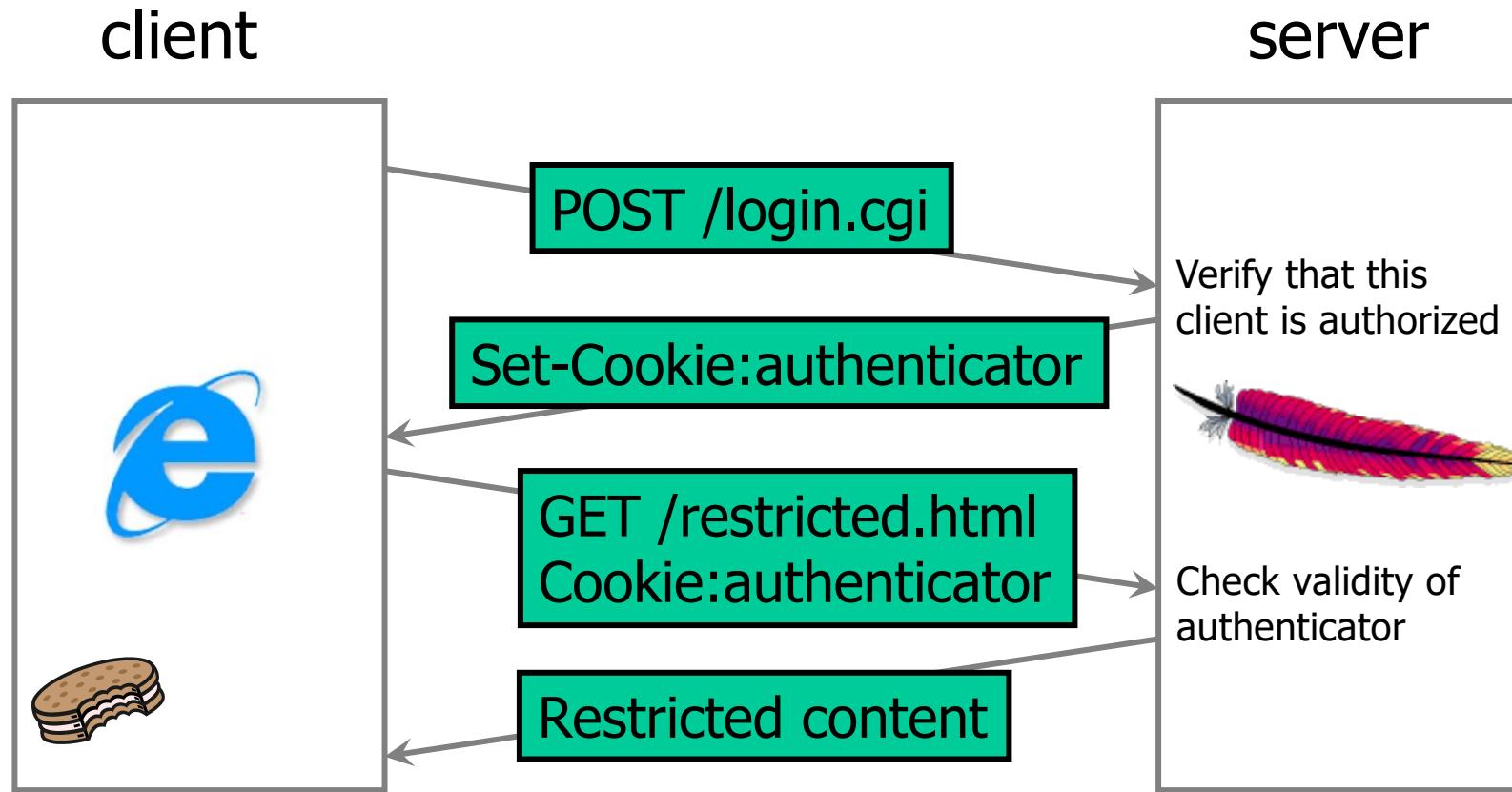
# Cookies

- Un cookie è una coppia name/value creata da un web server per conservare informazioni di stato sul computer che ospita il browser
- Solo il server che ha creato il cookie può usarlo



HTTP è un protocollo stateless; I cookies introducono informazioni di stato

# Una Sessione con i Cookies



Il cookie è implementato un header aggiuntivo presente in una richiesta (Cookie:) o risposta (Set-cookie:) HTTP:

- il server per assegnare un cookie, lo aggiunge tra gli header di risposta.
- Il client nota la presenza del cookie e lo memorizza in un'area apposita
- Il browser web client rimanda il cookie, senza alcuna modifica, allegandolo a tutte le richieste HTTP che soddisfano il pattern, entro la data di scadenza.
- Il server può scegliere di assegnare il cookie di nuovo, sovrascrivendo il vecchio.

# Cookies

- Possono essere usati dai browsers per le applicazioni web-based a scopo di:
  - Autenticazione
  - Tracking
  - Conservazione di informazioni specifiche dell'utente
- Contengono dati sensibili
- Il cookie è composto da una stringa di testo arbitraria, una data di scadenza (oltre la quale non deve essere considerato valido) e un pattern per riconoscere i domini a cui rimandarlo.
  - Name session-token
  - Content "s7yZiOvFm4YymG...."
  - Domain .amazon.com
  - Path /
  - Expires Monday, September 08, 2031 7:19:41 PM

# E-Tags

- L'ETag è un header della risposta HTTP che contiene tipicamente un identificativo o un hash.
  - Viene generato dal server e mantenuto in cache dal client.
  - Quando il client effettua nuovamente una richiesta, invia in allegato l'ETag all'interno dell'header **If-None-Match**.
  - Se il valore corrisponde alla versione corrente sul server, quest'ultimo ritorna una risposta di tipo **304 - Not Modified**, che non contiene alcun body (e pertanto è molto leggera e veloce) e che istruisce il client a mantenere i dati precedenti in cache per un ulteriore TTL.



# Scopi: Pubblicità & Guadagno

- **Profiling:** ovvero tenere traccia delle abitudini e costruire un profilo dei consumatori:
  - politico
  - religioso
  - commerciale
  - sessuale
- Agenzie che registrano l'attività e le preferenze di un utente per produrre **pubblicità personalizzata**
- Esiste un mercato di **vendita di dati personali**
- Utilizzo della tecnologia per raggiungere questi scopi (vedi spider, cookies, malware, ...)

# Le minacce fantasma



Google™  
Italia



double  
click

# Comunicazioni sicure: Cifratura di flusso

## Secure Sockets Layer (SSL / TLS)

- Trasparente all'applicazione
- Autenticazione a chiave pubblica (di server e client)
- Cifratura della sessione
- Integrità dei dati e non ripudiabilità
- Protegge i dati solo nella comunicazione (ma se uno ha accesso all'host...)
- Tunneling https, pops, imaps, smtps, sftp, etc.

[http://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://en.wikipedia.org/wiki/Transport_Layer_Security)

<http://tools.ietf.org/html/rfc4346>

<http://www.openssl.org/>



# Limiti SSL / TLS

- Protegge i dati solo nella comunicazione (privacy e integrità dei contenuti)
- Non garantisce l'anonimato!
- Vulnerabilità dei browser



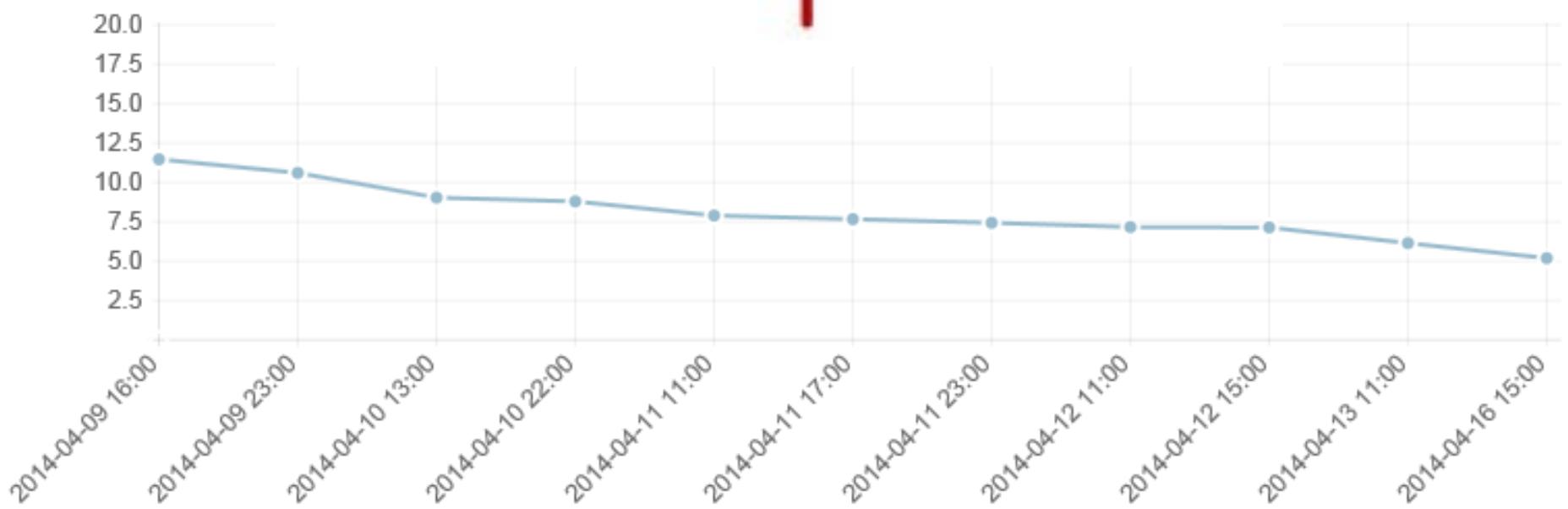
# L'esperienza di HeartBleed

Basato su una banale vulnerabilità di OpenSSL relativa a blocchi malformati con lunghezza



Historical

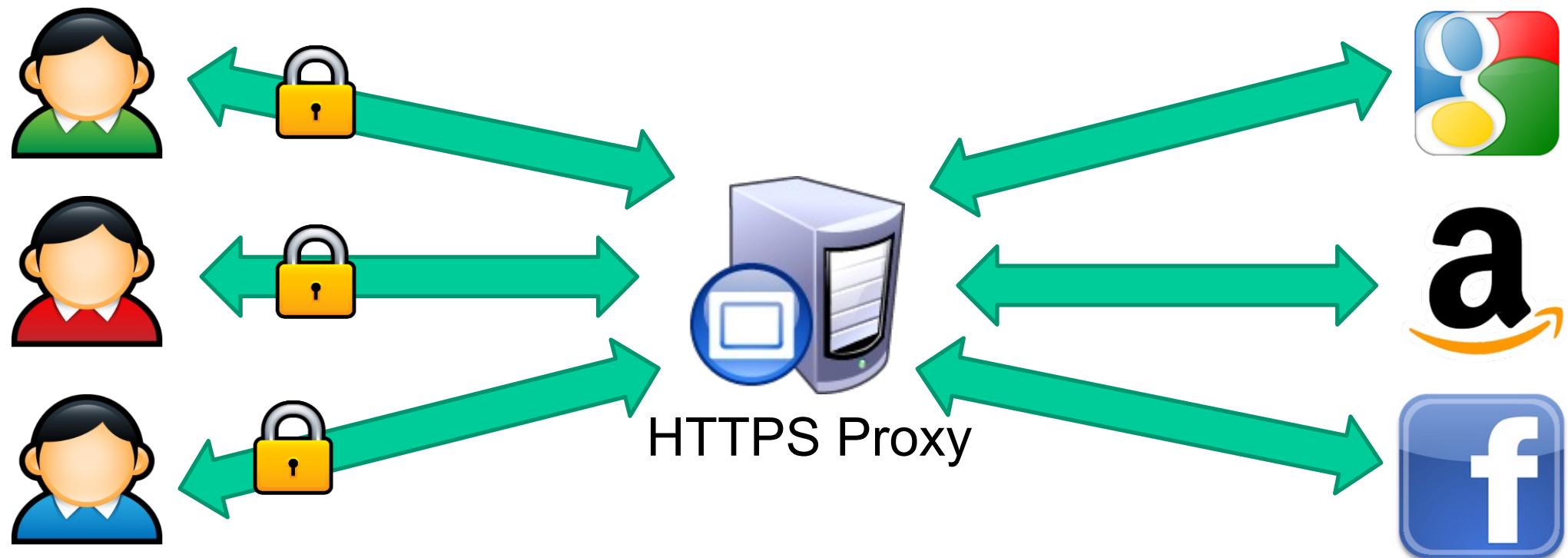
Websites



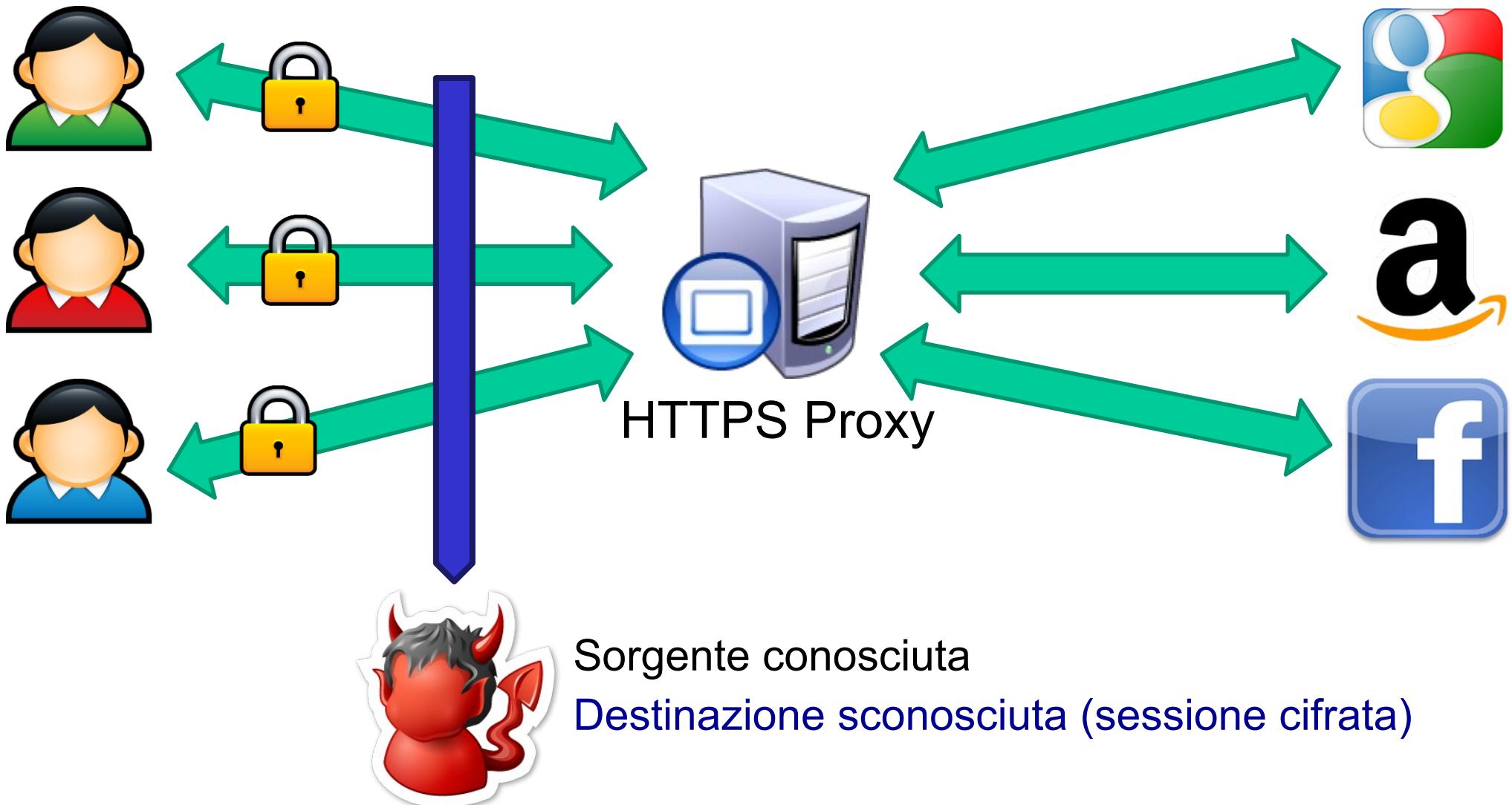
# Proxy anonimizzatori



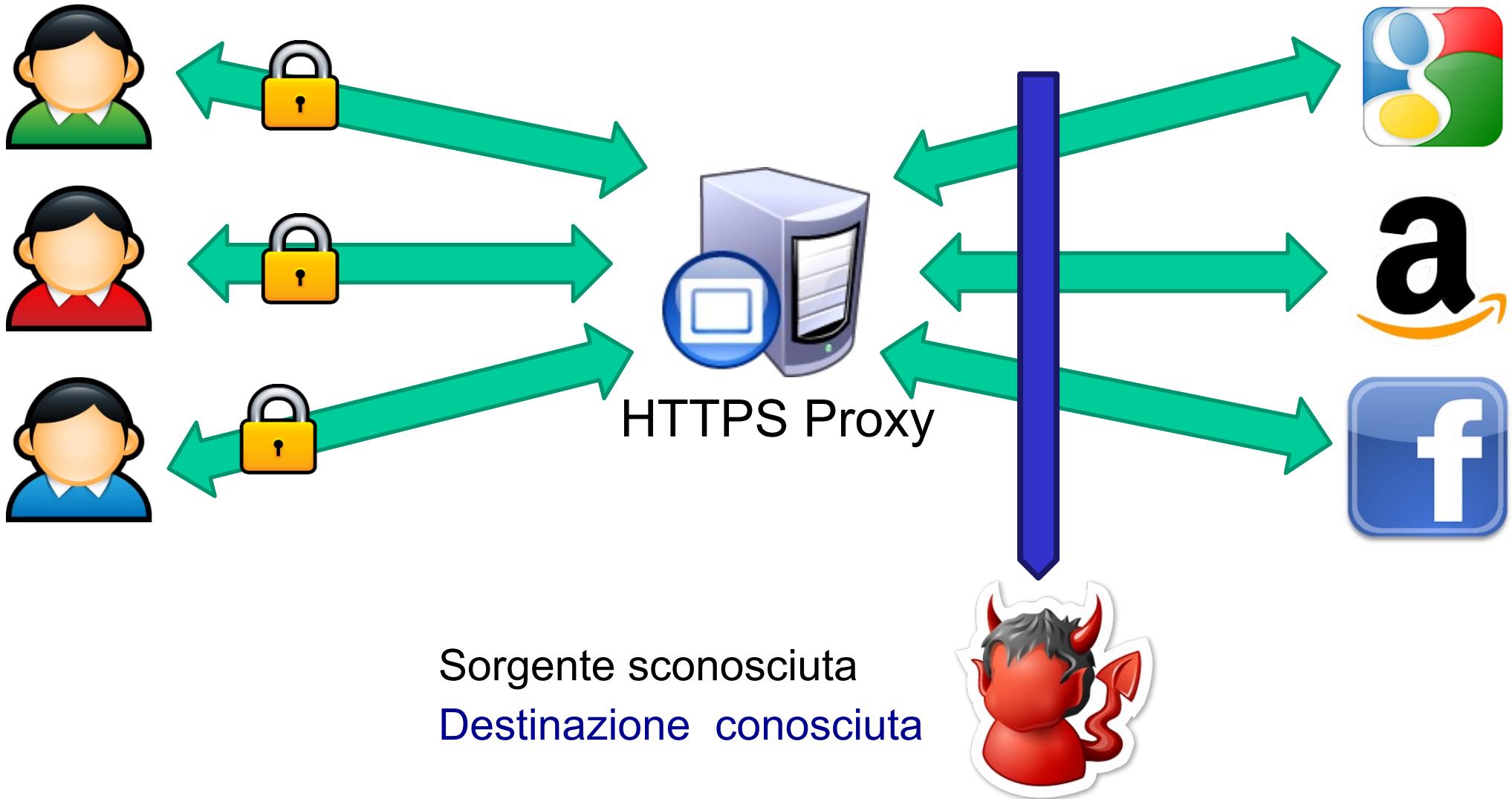
Usiamo un proxy HTTPS per anonimizzare!



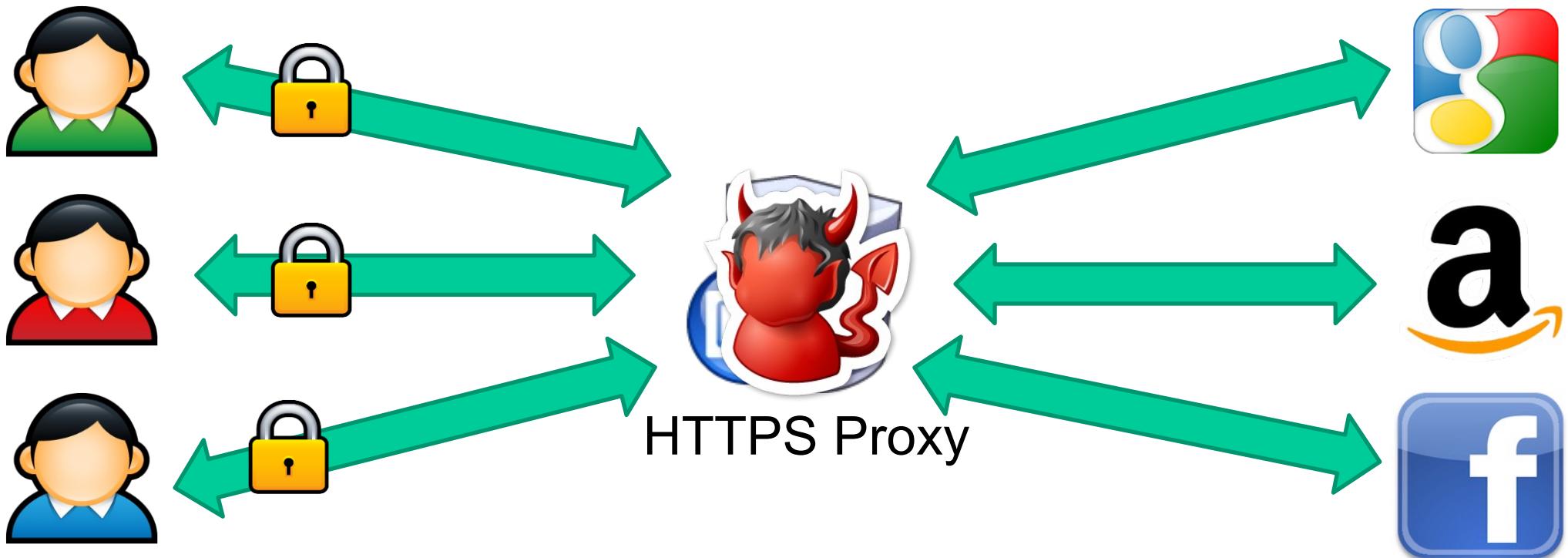
# Proxy anonimizzatori



# Proxy anonimizzatori

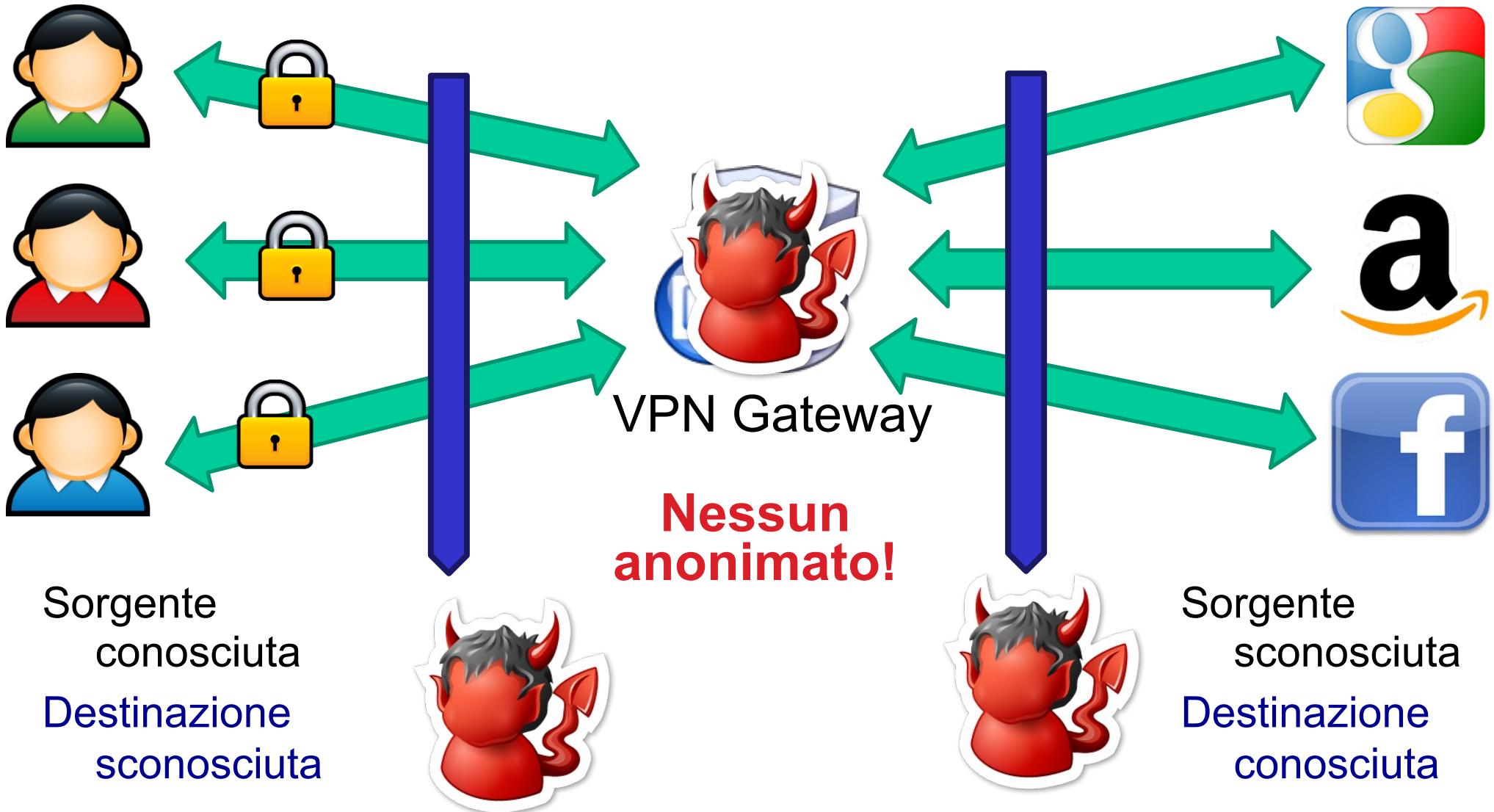


# Proxy anonimizzatori

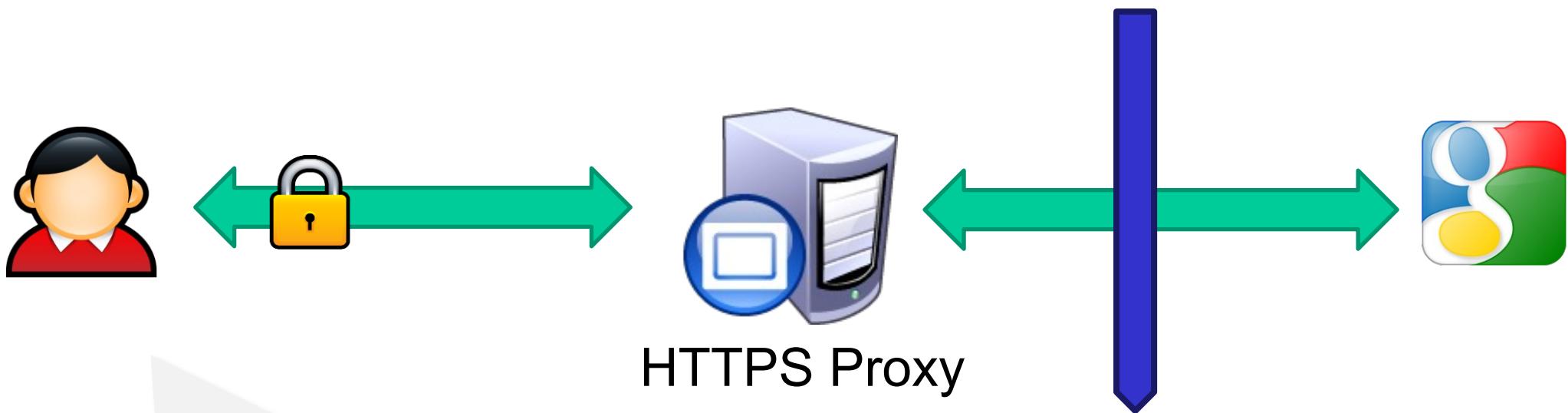


Nessun  
anonimato!

# Proxy anonimizzatori



# Proxy anonimizzatori: Contenuto



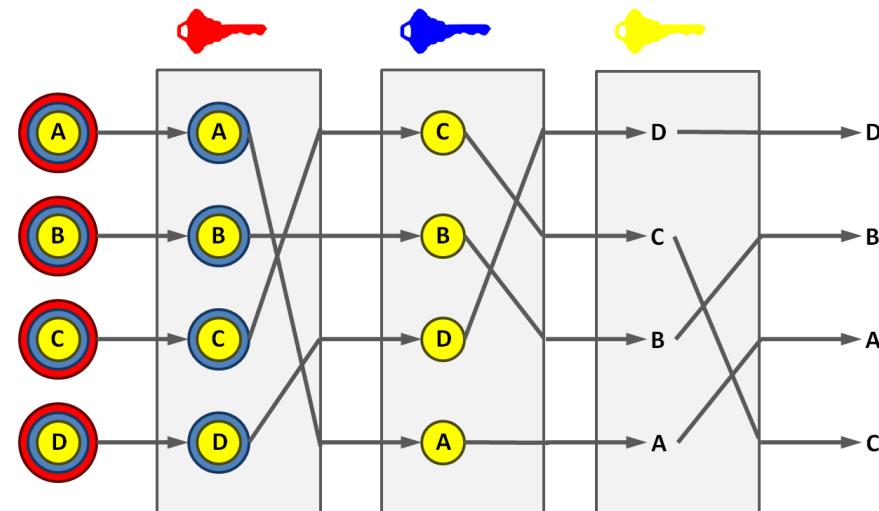
- Gmail (lettura/invio email)
- Google+ (aggiornamento profilo)
- Google Maps (navigazione)



Non è anonimo!

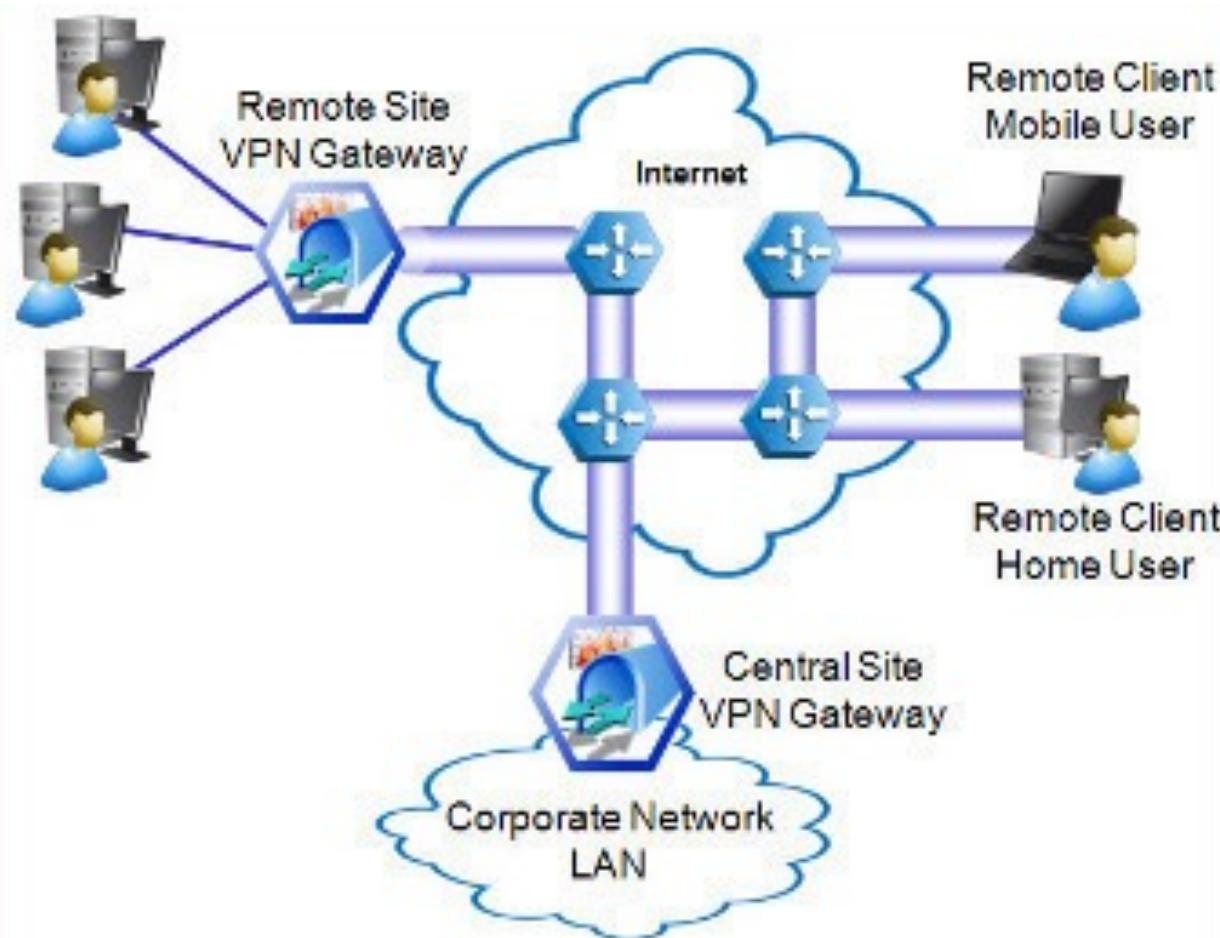
# Mix Network

- David Chaum, 1981
- Catena di proxy server (mix)
  - Ricevono un messaggio e lo inviano ad altro mix scelto casualmente
  - Ogni mix sulla path conosce solo il nodo precedente e quello successivo



# Proxy anonimizzatori

- Non solo HTTPS proxy
- Anche VPN Gateway



# Proxy anonimizzatori: Esempi



**NewIPNow.com**

Premium, anonymous, multi-ip web browsing

Connect to [www.google.com](http://www.google.com)

With the IP

151.80.203.127	(Nord-Pas-de-Calais, France)	<100ms	free
167.114.100.72	(Quebec, Canada)	308ms	free
89.248.164.184	(Noord-Holland, Netherlands)	510ms	free
216.27.27.74	(North Carolina, United States)	529ms	free
85.159.237.152	(NOORD-HOLLAND, NETHERLANDS)	807ms	free

**PROXIFY**

**PROXY** [Contact Us](#) [FAQ](#) [Advertise Here](#) [Web Proxies](#) [Proxy Forum](#)

**THE FASTEST, MOST SECURE VPN AVAILABLE** [READ MORE](#) [FREE 30 DAY TRIAL](#)

The proxy list is reordered randomly every 10 minutes to allow each proxy to get some exposure near the top. [Click here](#) to get a listing.

**Proxy Lists >> Web Proxy List**

Enter a URL to visit via proxy:  
 [GO](#)

Choose one of 3,385 working proxies:  
(Out of 50,352 total proxy servers)  
Last updated: April 17, 2015 at 7:10 AM EDT

random proxy

<a href="#">newipnow.com</a> (US, GProxy)
<a href="#">proxify.com</a> (US, Custom, SSL)
<a href="#">proxysite.com</a> (US, GProxy, SSL)
<a href="#">proxyx.net</a> (US, Custom, SSL)
<a href="#">proxyximus</a> (US, GProxy, SSL)
<a href="#">proxysite.org</a> (US, GProxy, SSL)
<a href="#">proxysite.us</a> (US, GProxy, SSL)
<a href="#">proxysite.com</a> (US, GProxy, SSL)
<a href="#">proxysite.co.uk</a> (GB, Custom, SSL)
<a href="#">fastproxynetwork.com</a> (GProxy)
<a href="#">javasciptproxy.com</a> (GProxy)
<a href="#">proxify.us</a> (US, Custom, SSL)
<a href="#">proxysite.com</a> (US, GProxy, SSL)
<a href="#">unblockyoutube.co</a> (US, GProxy, SSL)
<a href="#">gummi.org</a> (US, PHProxy)
<a href="#">unblock-me.co</a> (US, GProxy)
<a href="#">torrentproxy.co</a> (US, GProxy, SSL)
<a href="#">gopherhere.org</a> (US, GProxy, SSL)
<a href="#">proxies.us</a> (US, GProxy, SSL)
<a href="#">unblockyouku.com</a> (US, GProxy)
<a href="#">securetunnel.com</a> (US, GProxy, SSL)
<a href="#">proxies.us</a> (US, Unknown)
<a href="#">whatismyip.info</a> (US, Unknown)
<a href="#">proxysite.com</a> (US, GProxy)
<a href="#">unblock-websites.ninja</a> (GProxy)
<a href="#">unblockwebsites.ninja</a> (US, GProxy)
<a href="#">suche99.com</a> (US, PHProxy)
<a href="#">unknownproxy.com</a> (US, GProxy)
<a href="#">proxy-x.org</a> (US, GProxy, SSL)
<a href="#">proxies.biz</a> (GProxy)
<a href="#">proxies.us</a> (US, GProxy, SSL)
<a href="#">proxify.de</a> (GB, Custom, SSL)
<a href="#">proxify.eu</a> (GB, Custom, SSL)
<a href="#">sslsecureproxys.com</a> (US, GProxy)
<a href="#">hidethisitime.com</a> (US, GProxy)
<a href="#">proxify.org</a> (US, Custom, SSL)
<a href="#">freeproxy.ca</a> (US, Unknown)
<a href="#">freevideoproxy.com</a> (US, GProxy)

**Top Ten Listings**

<a href="#">newipnow.com</a>
<a href="#">proxify.com</a>
<a href="#">proxysite.com</a>
<a href="#">proxyximus</a>
<a href="#">proxyturbo.com</a>
<a href="#">4everproxy.com</a>
<a href="#">proxysite.co.uk</a>
<a href="#">fastproxynetwork.com</a>
<a href="#">proxies.us</a>
<a href="#">whatismyip.info</a>

**Tier Two Listings**

<a href="#">proxify.net</a>
<a href="#">modis.cc</a>
<a href="#">proxify.co.uk</a>
<a href="#">fastproxynetwork.com</a>

**Bold Listings**

<a href="#">javasciptproxy.com</a>
<a href="#">proxify.us</a>
<a href="#">peopleproxy.com</a>
<a href="#">unblockyoutube.co</a>
<a href="#">gummi.org</a>
<a href="#">unblock-me.org</a>
<a href="#">torrentproxy.co</a>
<a href="#">gopherhere.org</a>
<a href="#">proxies.us</a>
<a href="#">unblockyouku.com</a>
<a href="#">securetunnel.us</a>
<a href="#">proxies.us</a>
<a href="#">whatismyip.info</a>
<a href="#">proxyn.com</a>
<a href="#">unblock-websites.ninja</a>
<a href="#">suche99.com</a>
<a href="#">unknownproxy.com</a>
<a href="#">proxy-x.org</a>
<a href="#">proxies.biz</a>
<a href="#">proxify.de</a>
<a href="#">proxify.eu</a>

**Proxy Lists**

- Web Proxies - Master List
- SSL Proxies
- Web Proxies - Sorted by Country
- Web Proxies - Sorted by Software
- Web Proxies - Sorted by IP
- Web Proxies - Sorted by ISP
- Web Proxies - Statistics
- Web Proxies - Defunct
- Web Proxies - New
- Tor Servers
- PlanetLab's CoDeen Proxies
- VPN Comparison

**For Your Website**

- Small Proxy Form
- Large Proxy Form

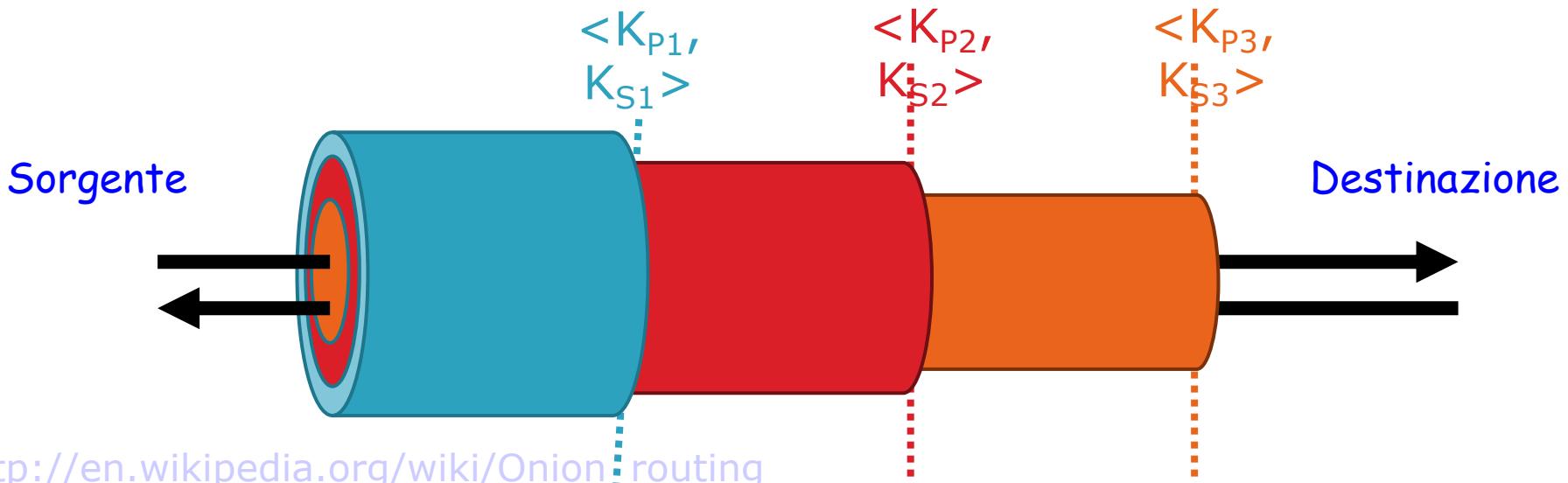
**Most Popular:**

(sorted by total # of unique visitors in the past 7 days)

- [proxify.com](#) (5,551)
- [newipnow.com](#) (3,933)
- [proxify.us](#) (3,694)
- [proxies.us](#) (3,514)
- [proxify.eu](#) (3,327)
- [proxies.com](#) (1,986)
- [fastproxynetwork.com](#) (1,626)
- [modis.cc](#) (1,596)
- [proxify.co.uk](#) (1,555)
- [proxify.eu](#) (1,498)

# Onion routing

- Tecnica di tunneling del traffico su circuiti virtuali (U.S. Naval Research Laboratory, 1998)
  - **Incapsulamento telescopico** dei pacchetti in strutture dati cifrate
  - Instradamento attraverso vari nodi della rete con **multipli layer di cifratura** (una per ogni hop)
  - **Resistente** a tecniche di analisi del traffico
  - Bassa latenza



[http://en.wikipedia.org/wiki/Onion\\_routing](http://en.wikipedia.org/wiki/Onion_routing)

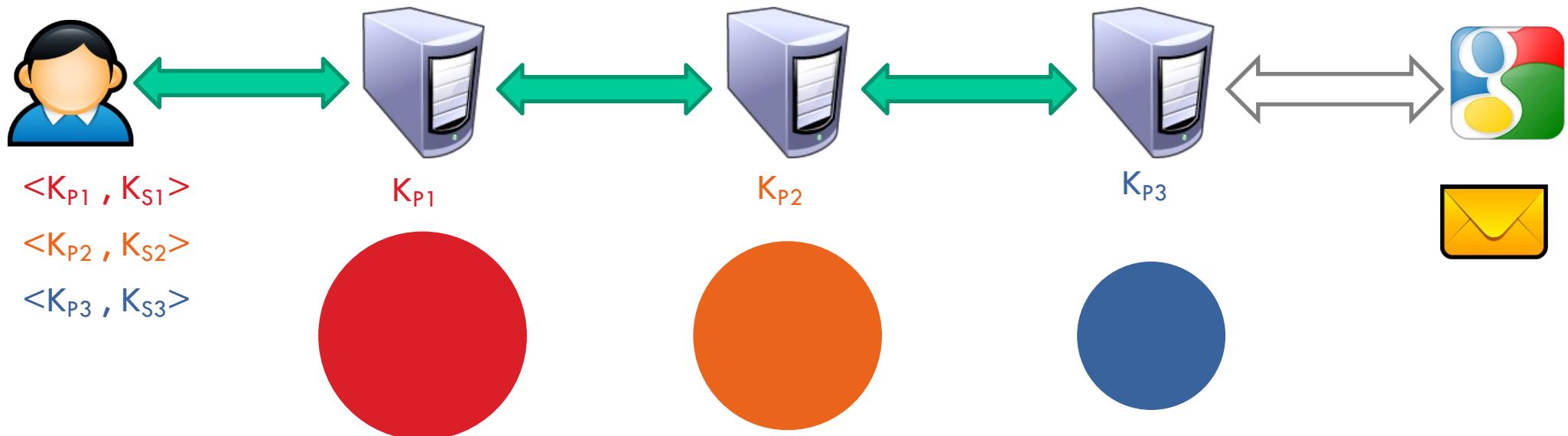
<http://www.onion-router.net/>

<http://www.onion-router.net/Publications.html>

# Cammino dei dati

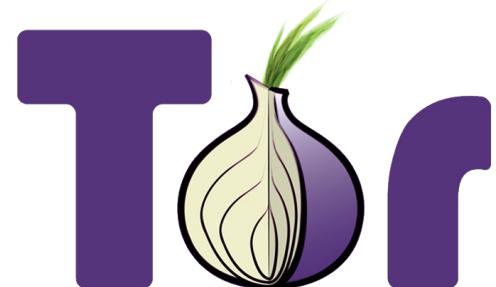
- Un **circuito virtuale** è costruito incrementalmente un hop per volta in accordo a uno schema di cifratura a cipolla
- L'origine sceglie un cammino (insieme di hop) da una lista fornita da un "**directory node**"
- L'origine **negozia una chiave** (o una **coppia di chiavi**) con ogni nodo intermedio
- I dati sono cifrati in origine e decifrati lungo il cammino
- Ogni nodo intermedio conosce **solo** il suo **predecessore** e **successore**
- Solo il nodo di **uscita** può vedere il messaggio, decifrando l'ultimo layer ma non sa da dove o stesso proviene

$$E(K_{P1}, E(K_{P2}, E(K_{P3}, M))) = C$$



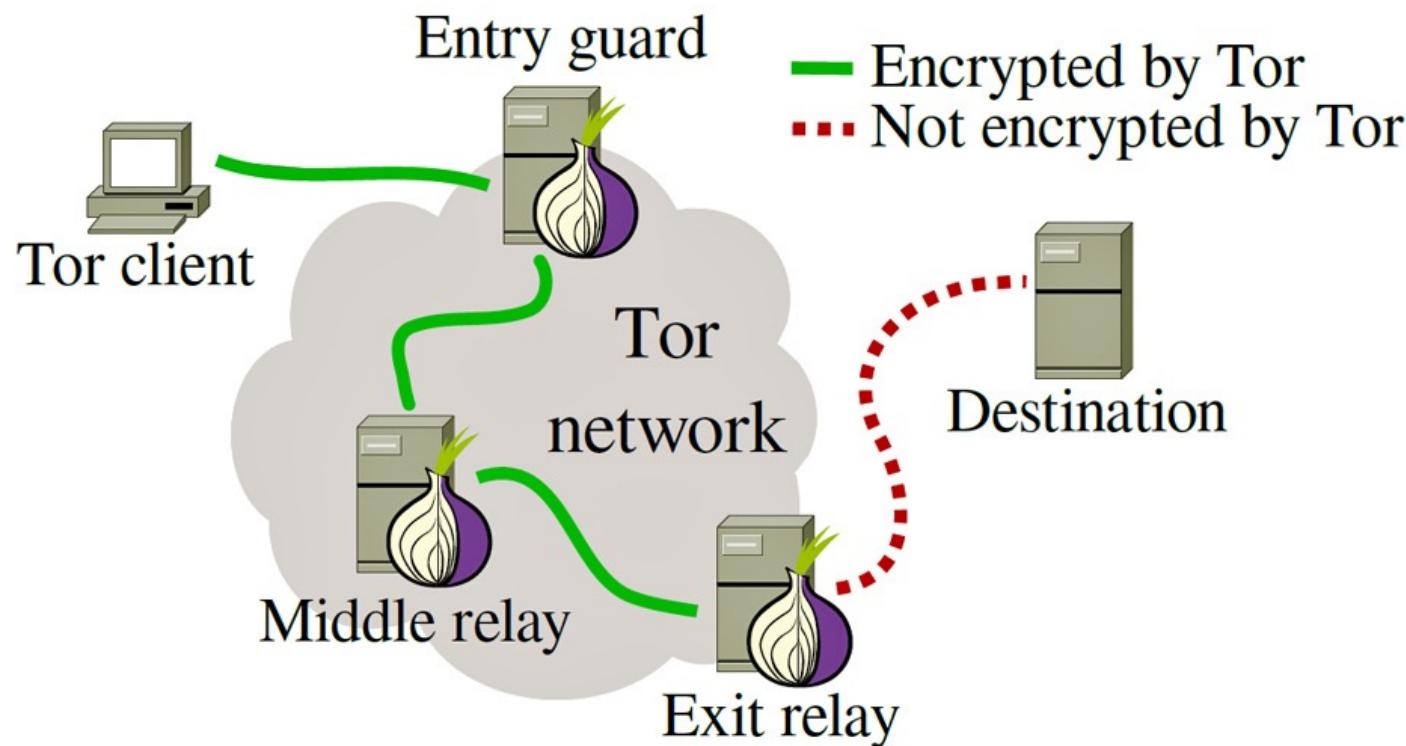
# The Onion Router (TOR)

- Rete **Overlay** di Onion Routers col ruolo di **relay**
- Sviluppato per la **US Naval Research Laboratory** e poi da **DARPA**, per proteggere le comunicazioni dei servizi segreti.
- Fornisce essenzialmente due servizi:
  - **Connessioni anonime** in uscita
  - Fornitura di **servizi nascosti**
- Supporta:
  - **Perfect Forward Secrecy**
  - Utilizzo di TLS nelle comunicazioni tra nodi
  - Controllo di **integrità** dei dati end-to-end
  - Controllo di **congestione**
  - Interfacciamento tramite il protocollo **SOCKS**
  - Scelta dei relay basata sulla loro **larghezza di banda**



# Architettura hop-by-hop relay

- Tor è un'infrastruttura overlay (rete sulla rete)
- I nodi sono connessi fra loro tramite collegamenti logici (circuiti virtuali), ciascuno dei quali corrisponde ad un percorso nella rete sottostante.
- Il nodo di ingresso (entry guard) garantisce l'accesso all'infrastruttura overlay
- Il nodo di uscita della rete overlay garantisce l'accesso alla global internet



# Tipi di nodi

La rete **Tor** prevede quattro tipologie di nodi:

- **Client**

In questa configurazione normale di base, Tor gestisce unicamente le connessioni dell'utente permettendogli di collegarsi alla rete Tor.

- **Middleman router (o middle relay)**

È un nodo che gestisce traffico di terzi da e per la rete Tor, senza collegarsi direttamente all'esterno. Nel caso funga anche da client, esso gestisce anche le connessioni dell'utente, garantendo un maggiore anonimato. Tutti nodi sono pubblicamente noti, per scelta progettuale.

- **Exit router (o exit relay)**

È un nodo Tor che gestisce traffico di terzi da e per la rete Tor, e verso l'esterno. È possibile definire una exit policy sulle connessioni in uscita all'esterno della rete Tor. Come il caso precedente, offre una protezione maggiore all'utente che lo usa per le proprie connessioni. Come tutti i router Middleman Tor, essi sono pubblicamente noti.

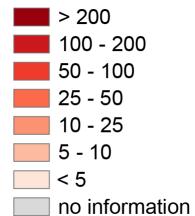
- **Bridge router**

I bridge router sono nodi semi-pubblici di tipo sperimentale, studiati per permettere di collegarsi anche in presenza di un filtraggio efficace contro Tor (come in Cina, Iran ecc.). Non compaiono nelle liste pubbliche dei nodi noti ma devono venire richiesti esplicitamente.

# Estensione rete TOR

## The anonymous Internet

Daily Tor users per 100,000 Internet users

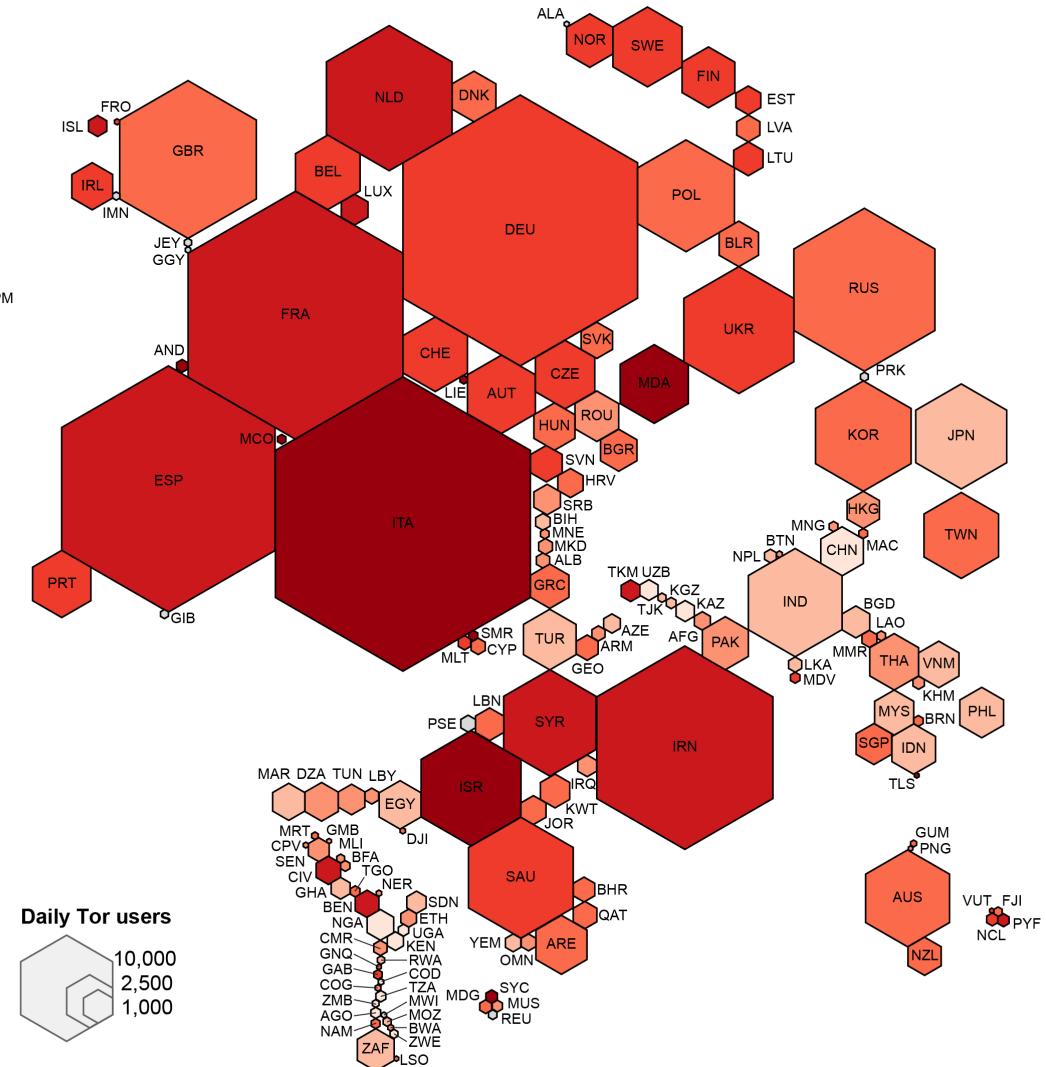
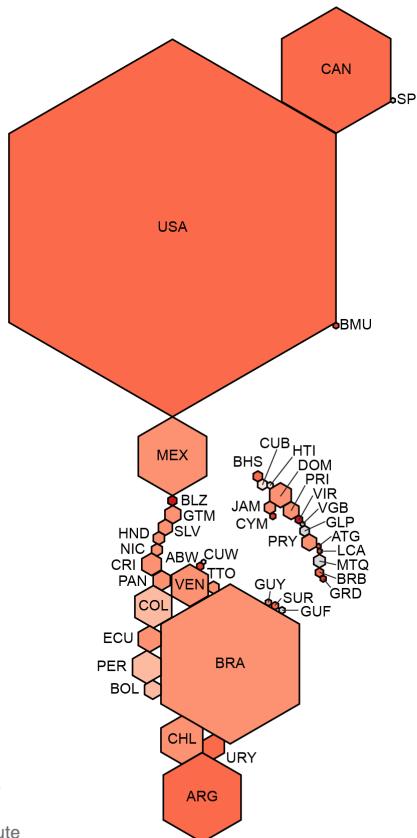


Average number of Tor users per day calculated between August 2012 and July 2013

data sources:  
 Tor Metrics Portal  
[metrics.torproject.org](http://metrics.torproject.org)  
 World Bank  
[data.worldbank.org](http://data.worldbank.org)

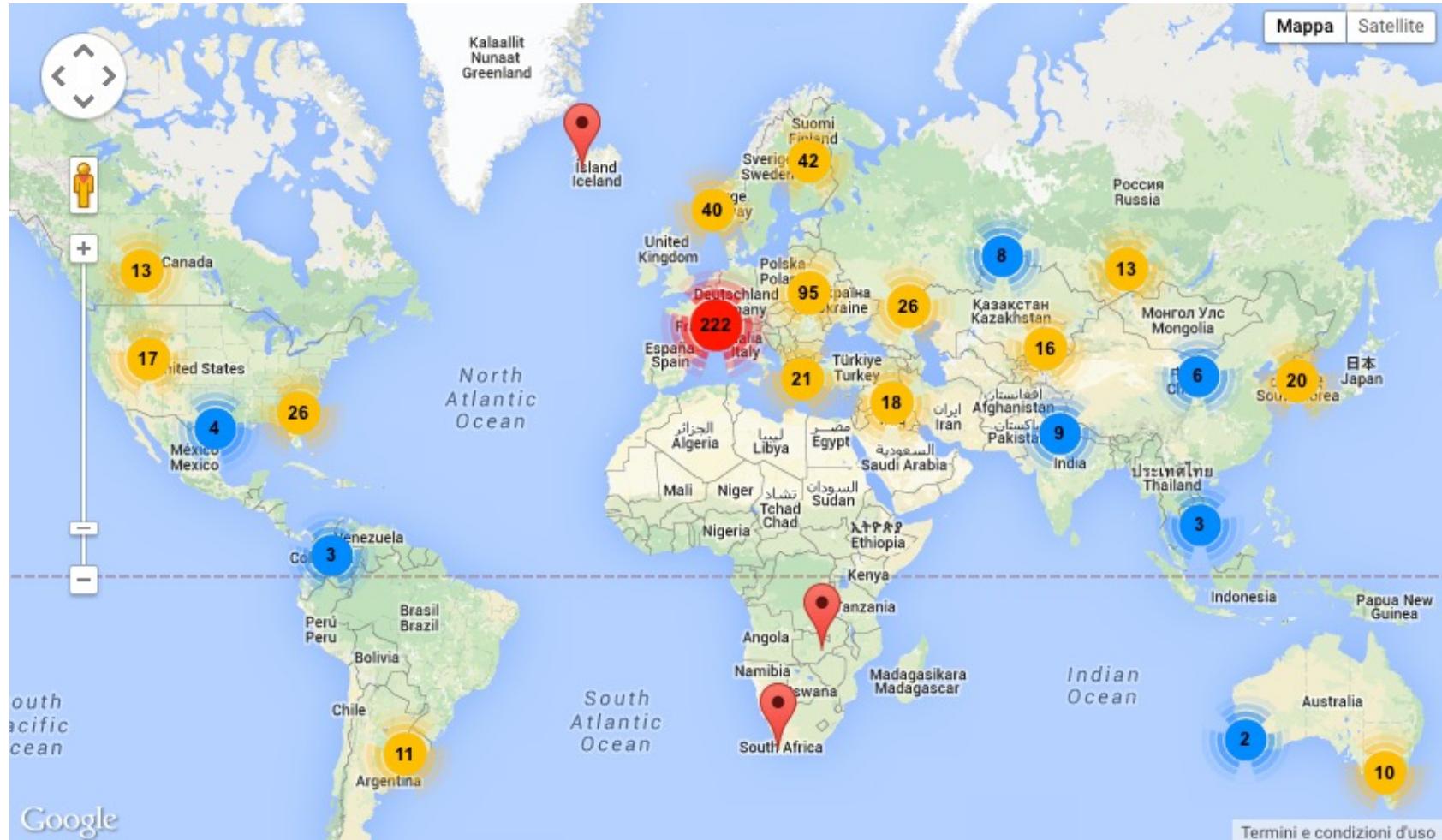
by Mark Graham (@geoplace) and  
 Stefano De Sabbata (@maps4thought)  
 Internet Geographies at  
 the Oxford Internet Institute  
 2014 • [geography.ox.ac.uk](http://geography.ox.ac.uk)

Oxford Internet Institute  
 University of Oxford



<https://metrics.torproject.org/oxford-anonymous-internet.html>

# Nodi Exit

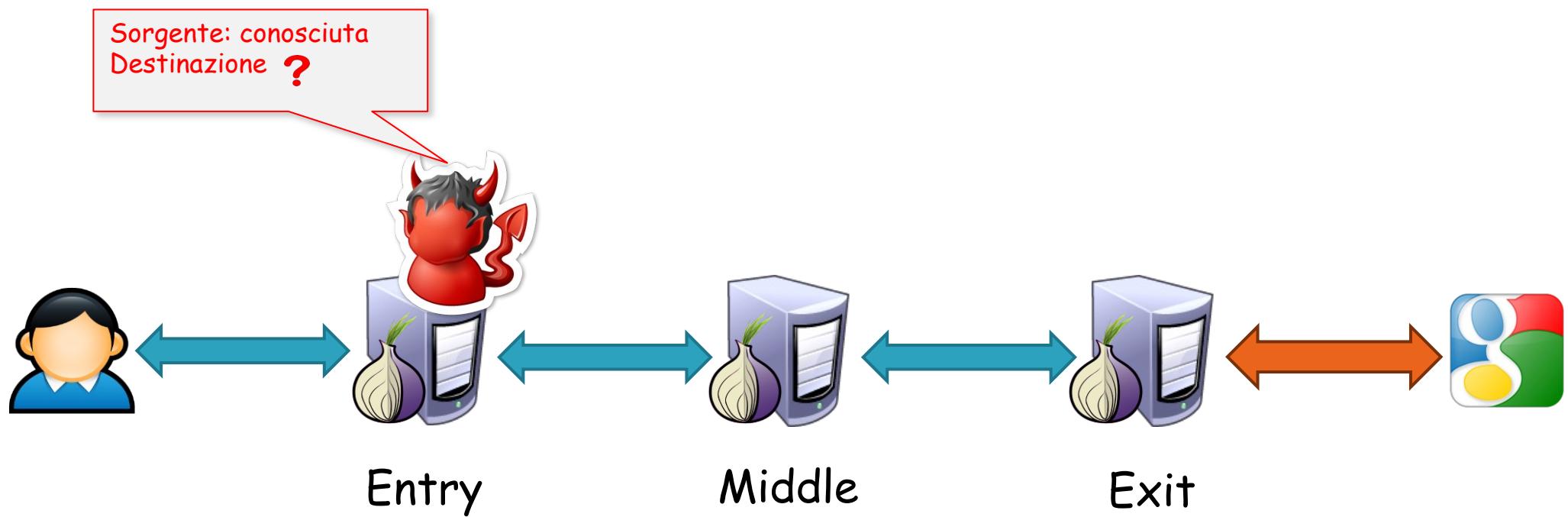


23 febbraio 2015

<http://hackertarget.com/tor-exit-node-visualization/>

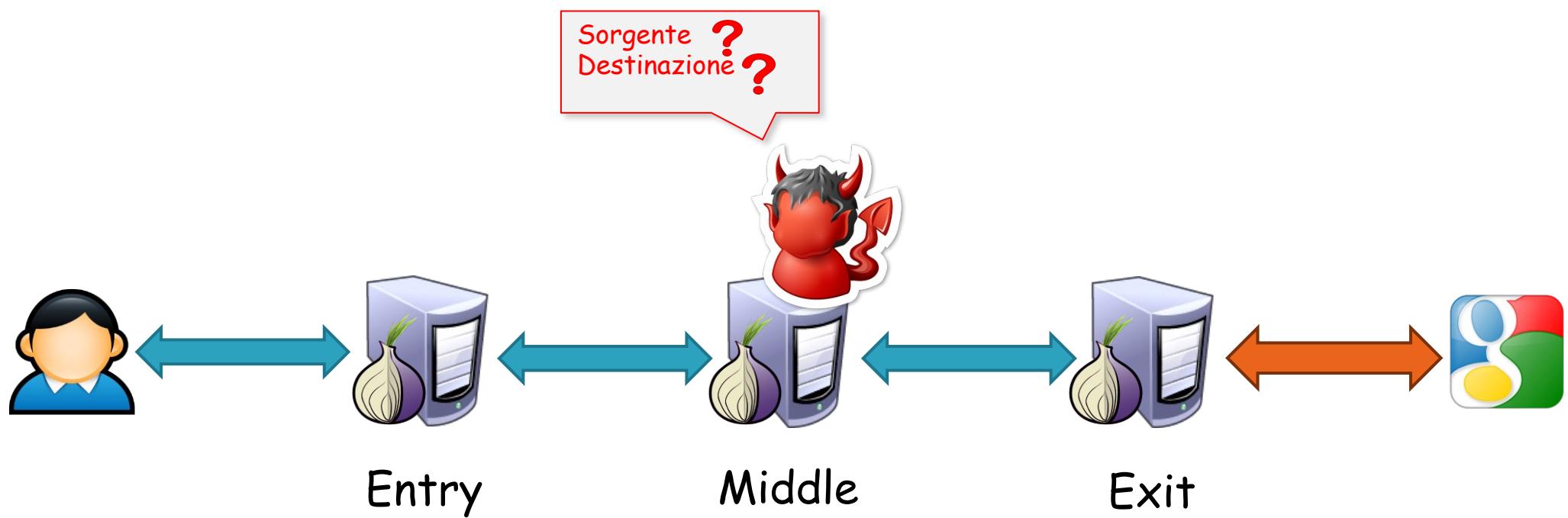
# Numero di relay

- Possibile scegliere numero di relay
  - Configurazione di default: 3
  - Meglio di più o di meno?



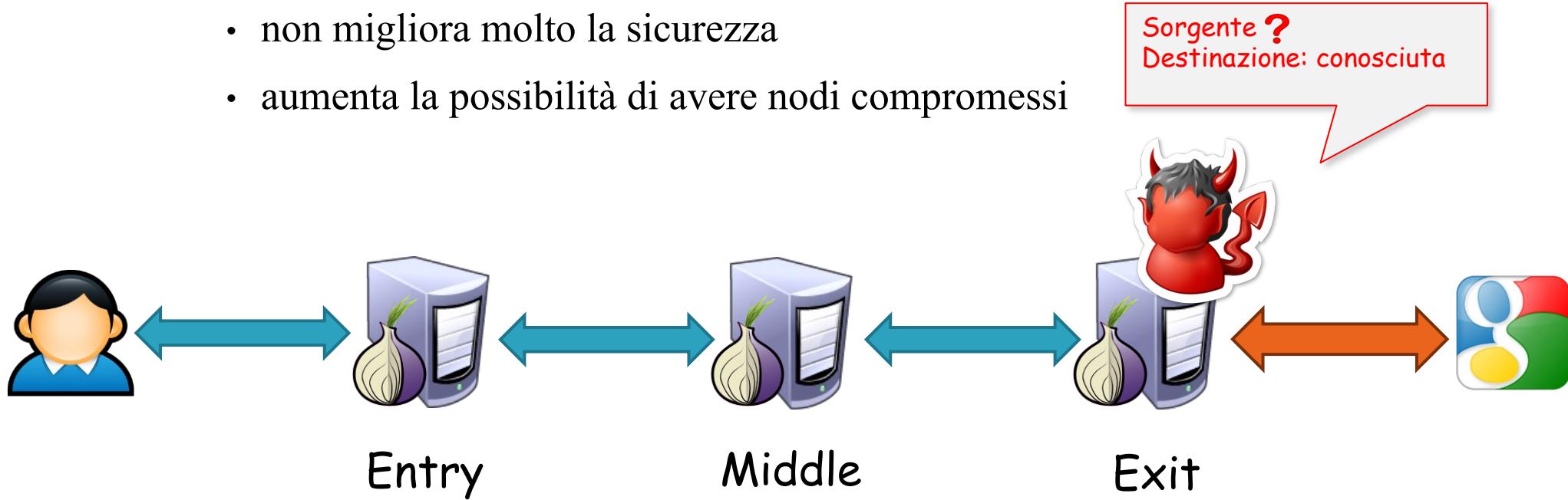
# Numero di relay

- Possibile scegliere numero di relay
  - Configurazione di default: 3
  - Meglio di più o di meno?



# Numero di relay

- Possibile scegliere numero di relay
  - Configurazione di default: 3
  - Meglio di più o di meno? Con numero maggiore:
    - prestazioni minori e latenza maggiore
    - non migliora molto la sicurezza
    - aumenta la possibilità di avere nodi compromessi



# Bridges

- Gli Indirizzi IP dei Tor relay sono noti
- Alcuni paesi bloccano il traffico a questi IP
- Soluzione: Tor Bridges
  - Essenzialmente, Tor proxies che non sono pubblici
  - Usati per la connessione di client al resto della rete Tor
- Tor ha bridges in molti paesi. Come ottenere la lista:
  - <https://bridges.torproject.org/bridges>
  - Email a [bridges@bridges.torproject.org](mailto:bridges@bridges.torproject.org)  
body "get bridges"
  - Esempio risposta:

```
5.20.130.121:9001
63dd98cd106a95f707efe538e98e7a6f92d28f94106.186.19.58:443
649027f9ea9a8e115787425430460386e14e0ffa69.125.172.116:443
43c3a8e5594d8e62799e96dc137d695ae4bd24b2
```

# Celle

- Concetto simile alle celle in ATM
- Tutti i dati sono inviati in celle di taglia fissa (numero bytes)
- Celle di controllo:
  - Creazione (CREATE), riscontro (CREATED) distruzione di circuiti
- Celle di Relay:
  - Trasporto dati end-to-end
  - Apertura, chiusura stream, estensione circuiti (EXTEND)...

2	1	509 bytes			
CircID	CMD	DATA			

2	1	2	6	2	1	498
CircID	Relay	StreamID	Digest	Len	CMD	DATA

# In caso di compromissione...

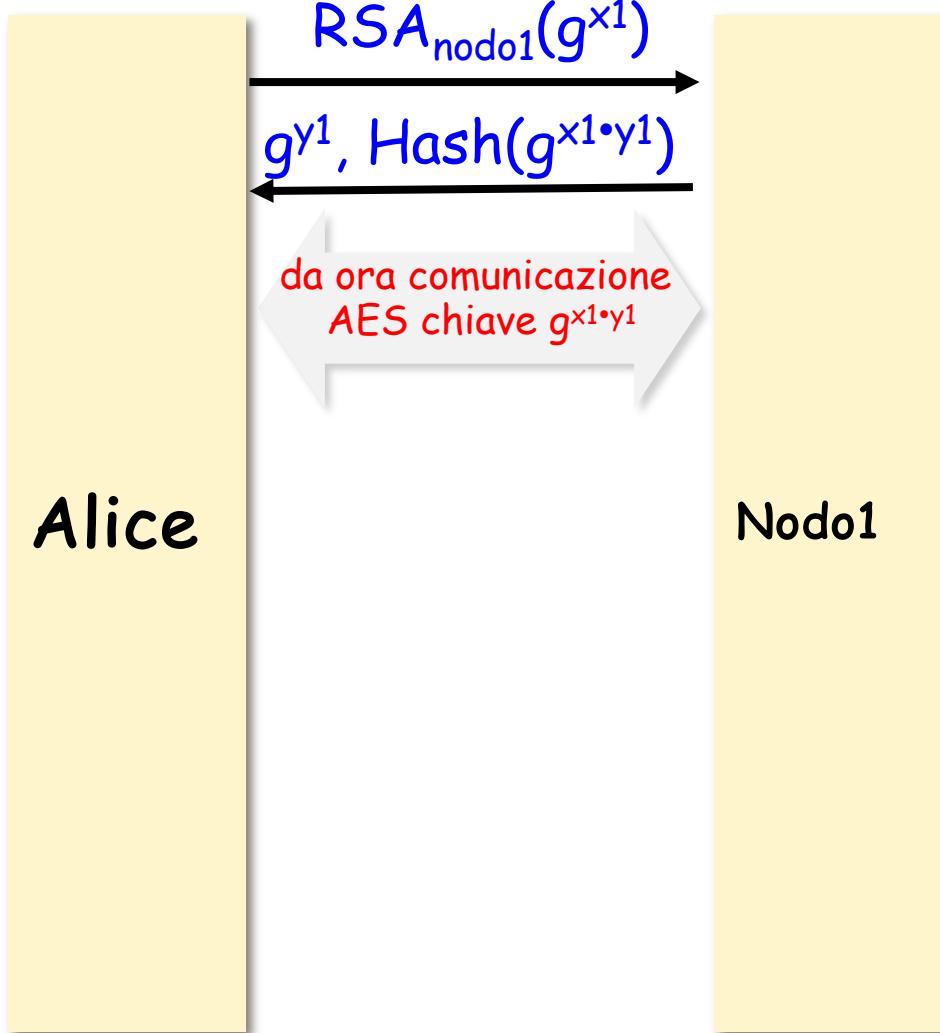
- Nella logica onion routing di base viene usata una coppia di chiavi pubblica/privata negoziate con ogni relay per cifrare il traffico...
- Che cosa succede se una chiave privata viene rubata?
  - Tutto il traffico **futuro** può essere decifrato
  - Se il traffico **passato** è disponibile, può essere decifrato

# Perfect Forward Secrecy

- La perfect forward secrecy, è una proprietà dei protocolli di negoziazione delle chiavi che assicura che se una chiave di cifratura a lungo termine viene compromessa, le chiavi di sessione generate a partire da essa rimangono riservate....
- Tor usa **Perfect Forward Secrecy**
  - Il client negozia una nuova coppia di chiavi con ogni relay
  - Chiavi originali usate solo per firma (cioè per verificare l'autenticità dei messaggi)

- Se  ottiene una chiave privata può decifrare tutto il traffico futuro
- ... ma il traffico passato è cifrato con chiavi non più disponibili

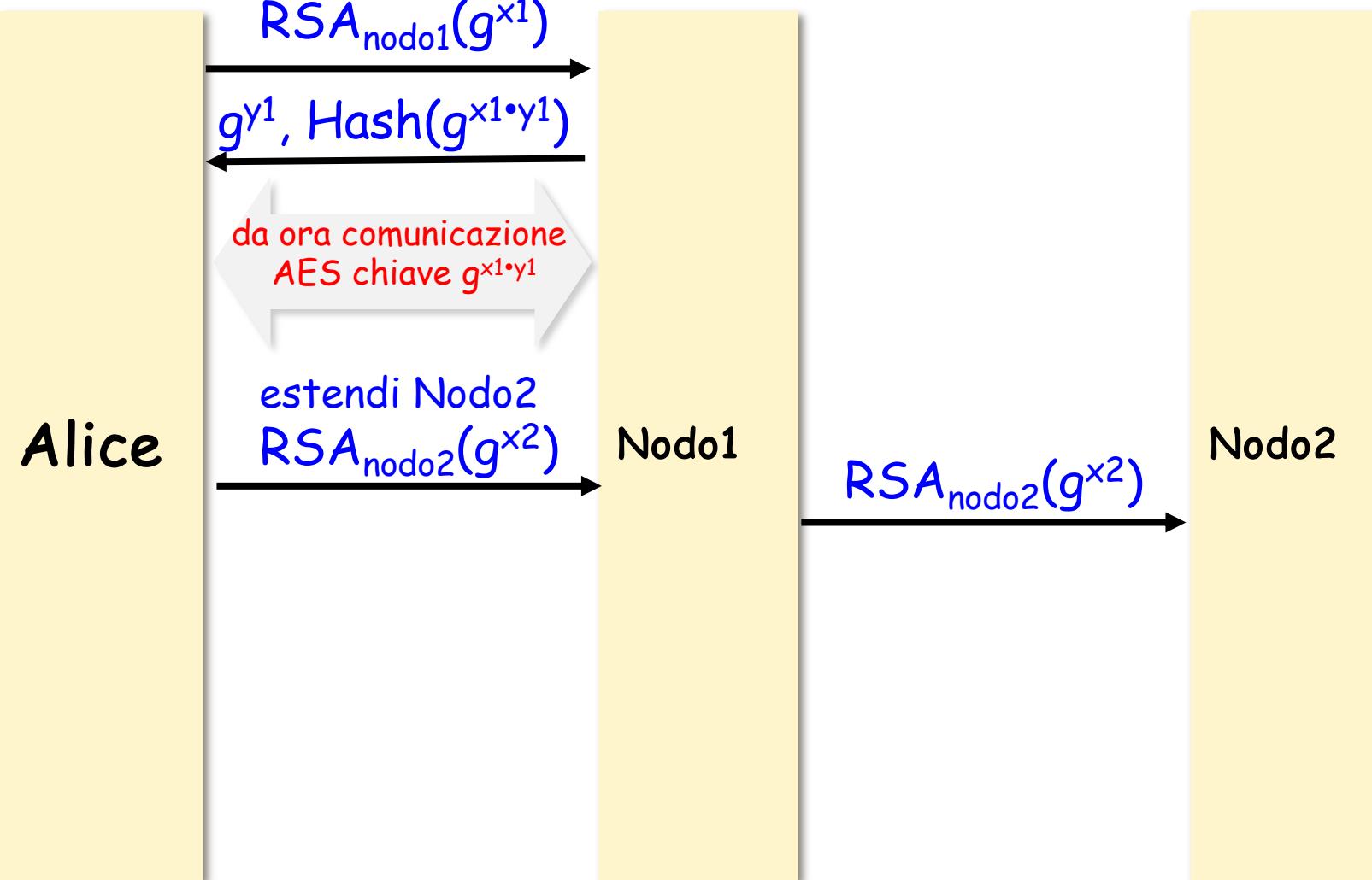
# Perfect Forward Secrecy



Alice stabilisce un circuito con Nodo1

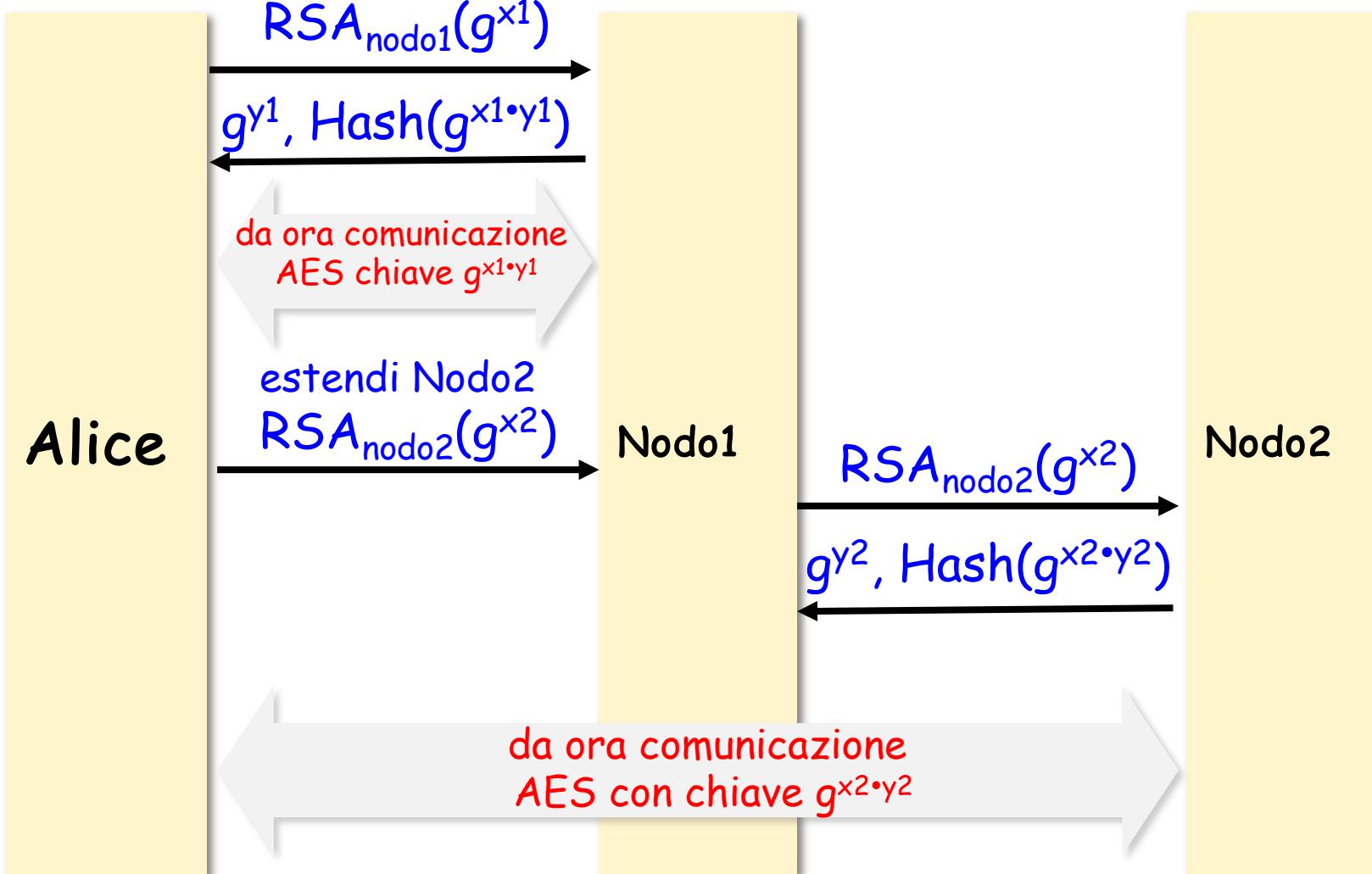
- Alice invia una cella CREATE a Nodo1 contenente  $g^{x1}$  cifrandola con la chiave pubblica di Nodo1
- Nodo1 decifra con la sua chiave privata la cella CREATE, calcola  $g^{x1y1}$  (prima parte dell'handshake di Diffie-Hellman) e deriva  $K_1$  (chiave di sessione tra Alice e Nodo1)
- Poi invia ad Alice la cella CREATED contenente  $g^{y1}$  e l'hash della chiave  $g^{x1y1} H(g^{x1y1})$
- Alice calcola  $g^{x1y1}$  (seconda parte dell'handshake di Diffie-Helmann) e deriva  $K_1$  utilizzando  $H(K_1)$ .
- Il circuito tra Alice e Nodo1 è creato e  $K_1$  sarà la chiave usata da AES

# Perfect Forward Secrecy



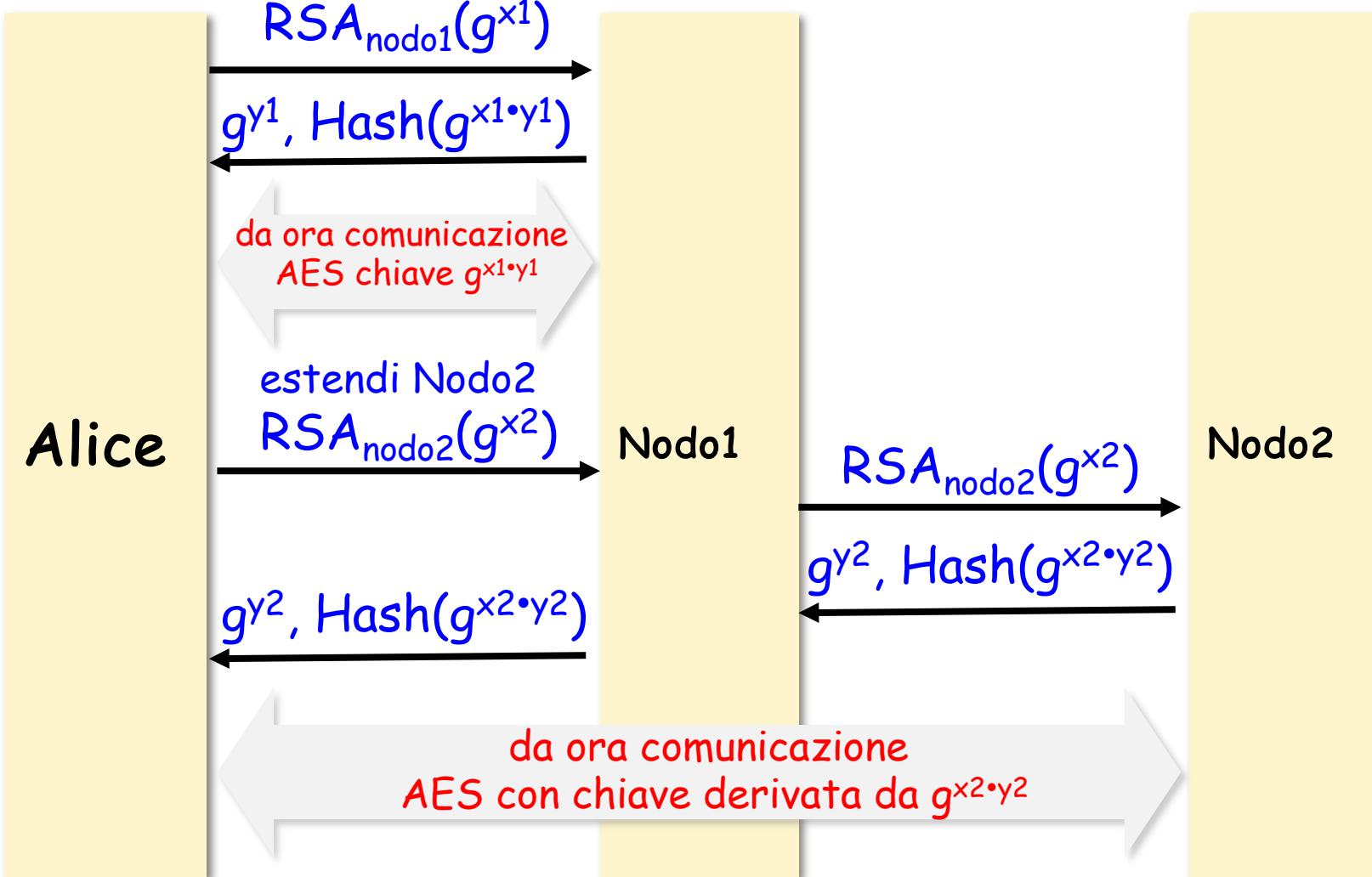
- Alice invia a Nodo1 una cella RELAY EXTEND cifrata con  $K_I$ , tale cella contiene  $g^{x2}$  cifrata con la chiave pubblica di Nodo2;
- Nodo1 decifra la cella con  $K_I$ , essendoun RELAY EXTEND, legge l'indirizzo del prossimo nodo
- Nodo1 estrae il payload della cella e lo inserisce in una nuova cella CREATE inviadola a Nodo2

# Perfect Forward Secrecy



- Nodo2 decifra la cella, calcola  $g^{x2y2}$  e deriva  $K_2$
- Nodo2 invia la cella CREATED contenente  $g^{y2}$  e l'hash della chiave  $H(K_2)$  a Nodo1
- Nodo1 estrae il payload dalla cella, lo mette in una cella RELAY EXTENDED cifrandola con  $K_1$

# Perfect Forward Secrecy



- Nodo1 invia la cella appena creata a Alice
- Alice decifra con  $K_1$ , calcola  $g^{x2y2}$  e deriva  $K_2$ .
- Il circuito virtuale tra Alice e Nodo è stato creato e userà AES con chiave  $K_2$ .

# Controlli

## Controllo di integrità

Effettuato su ogni end-point dei circuiti con un digest SHA-1 derivato dalla chiave negoziata ad ogni hop che rende impossibile la modifica dei dati in transito

## Rate Limiting

Usa un approccio token per limitare I volumi di traffico in ingresso

## Controllo di congestione

Realizzato via Circuit-level throttling basato su 2 finestre

- Packaging window: quante celle un relay può raggruppare e mandare indietro
- Delivery window: quante celle possano essere mandate avanti sulla rete esterna

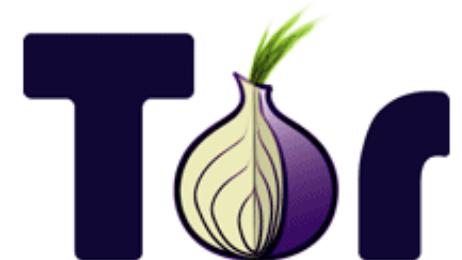
# Privoxy

Proxy HTTP con funzioni avanzate di **filtraggio**:

- Modifica dei contenuti del protocollo HTTP
- Blocco di pop-up
- Blocco di informazioni pubblicitarie
- Controllo dei cookie
- Rimuove informazioni superflue che il nostro browser comunica

L'uso di Privoxy è necessario perché i browser fanno passare le richieste DNS quando usano direttamente un proxy SOCKS, e ciò non è buono per mantenere l'anonimato.

<http://www.privoxy.org/>



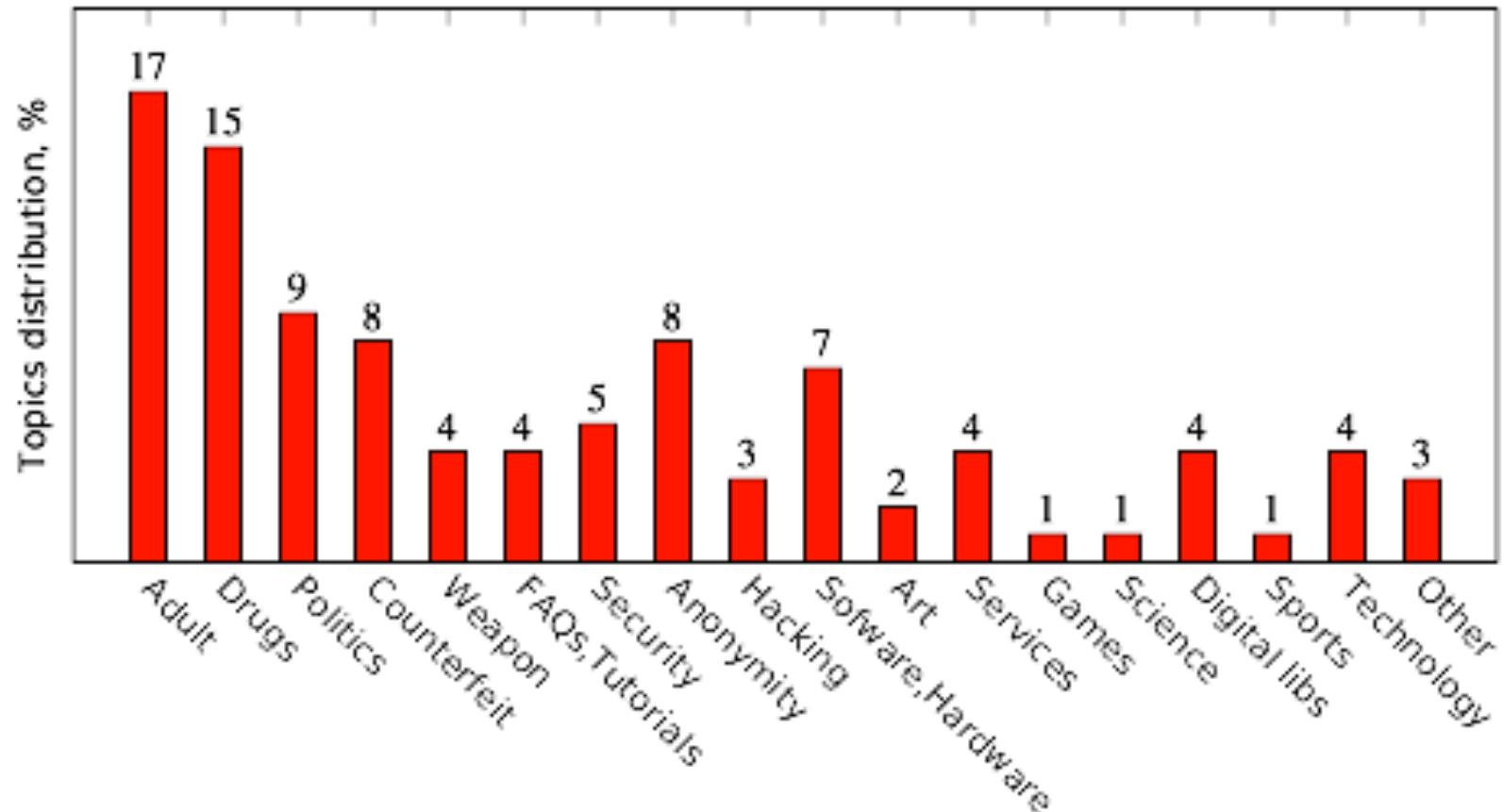
# Hidden Services

- L'identità in rete dell' host ospitante il servizio non viene rivelata realizzando:
  - Servizi di rete di cui non si conosce la collocazione fisica
  - Servizi resistenti alla censura (filtraggio del traffico, filtraggio DNS)
  - Resistenza alla violazione fisica (se non si sa dove sia il server..)
  - Impossibilità di risalire a chi pubblica e a chi fruisce il servizio
- Ciò consente di avere un server in grado di erigere servizi senza che si conosca l'indirizzo IP oppure il nome DNS



<http://tor.eff.org/docs/tor-hidden-service.html>

# Hidden Services



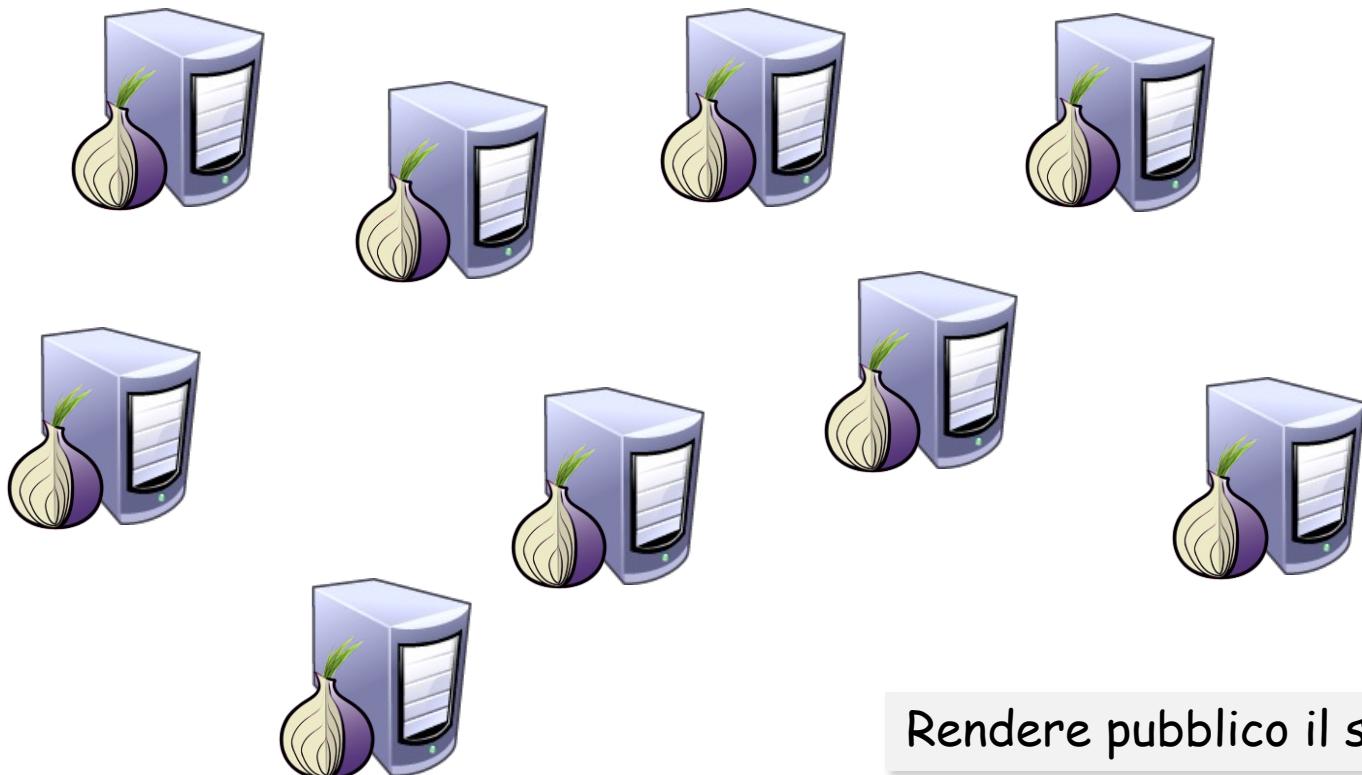
<http://arxiv.org/pdf/1308.6768v1.pdf>, 30 agosto 2013

# Hidden Services

- Come funzionano?
- E' possibile individuare il server?

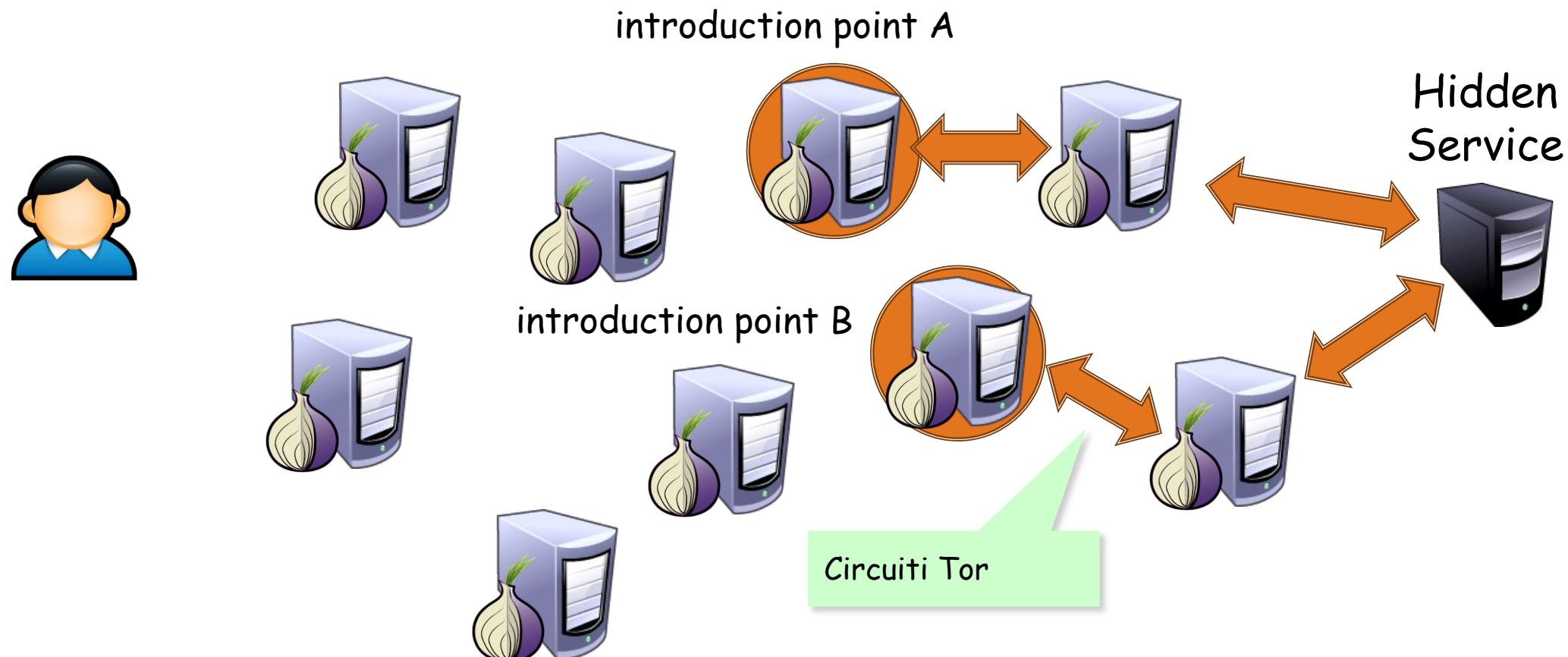


# Hidden Services



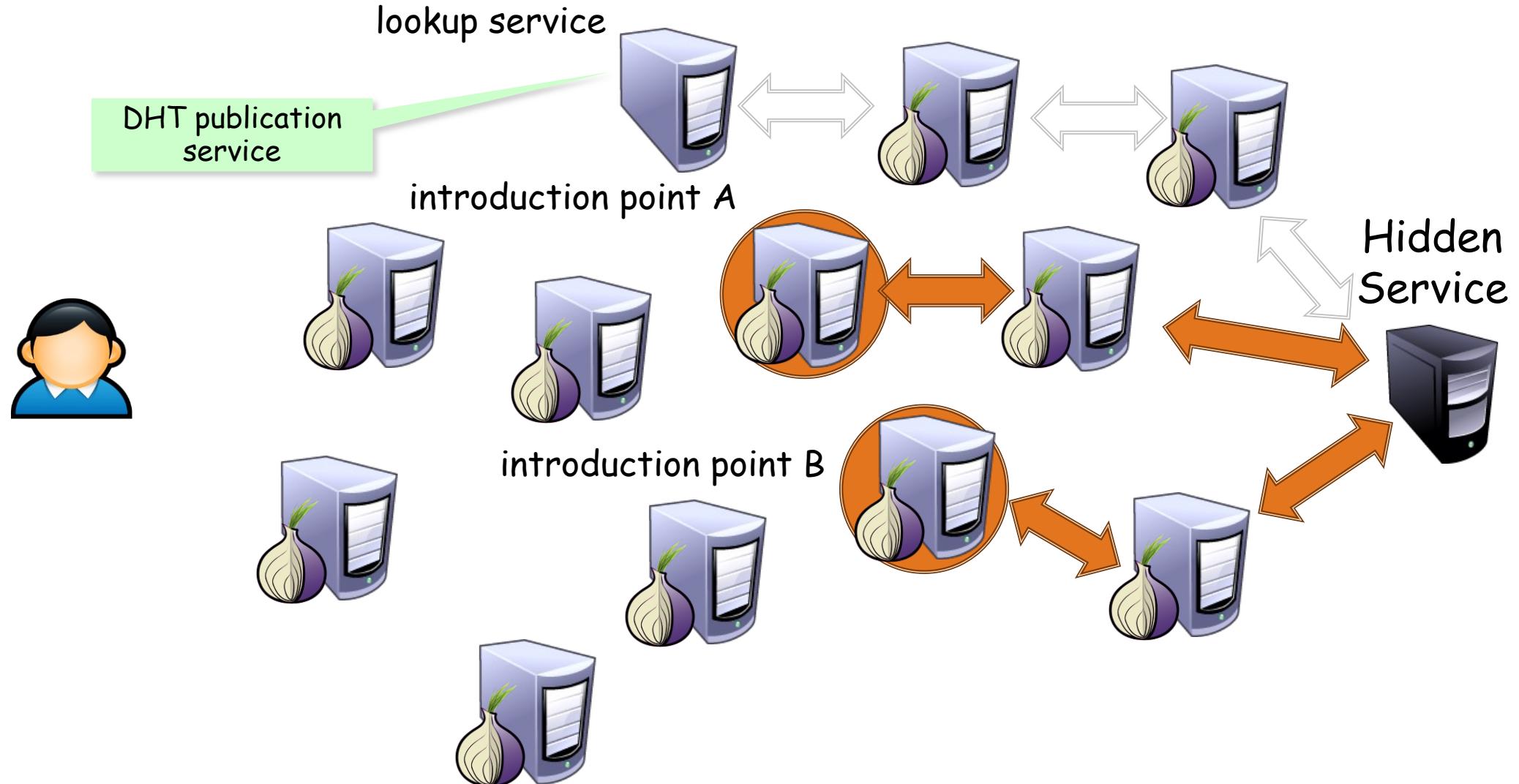
Rendere pubblico il servizio

# Hidden Services



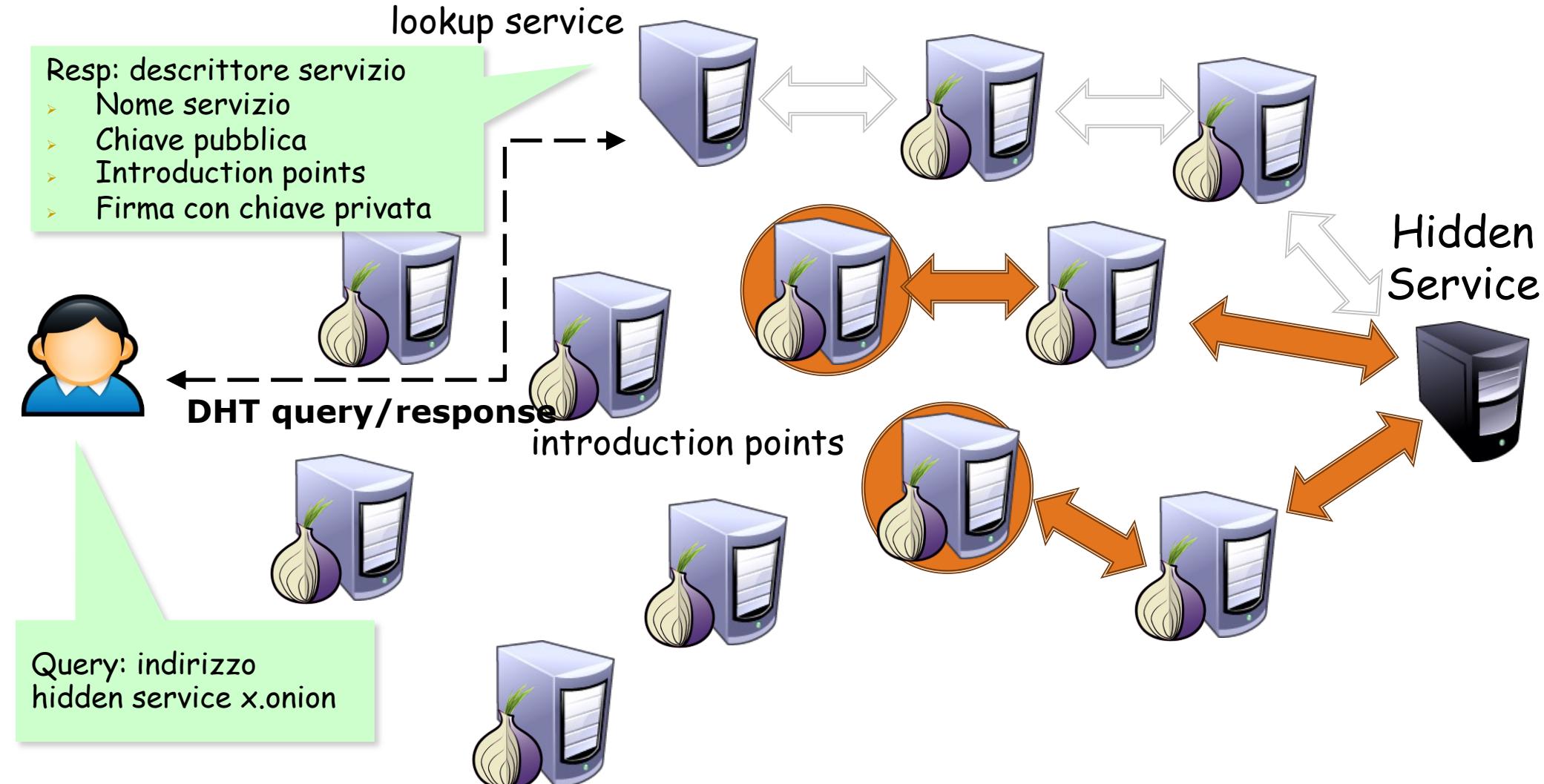
L'hidden service sceglie in modo casuale alcuni relay (A e B nell'esempio), costruisce dei circuiti TOR verso di essi e chiede loro di agire da **introduction point** fornendo loro la sua **chiave pubblica** che identifica il servizio. A e B **sanno come arrivare** all'hidden service attraverso circuiti TOR ma non conoscono la sua identità

# Hidden Services



L'hidden service un costruisce “service descriptor”, contenente la chiave pubblica e un summary di ogni introduction point, firma questo descrittore con la sua chiave privata.  
Il descrittore viene inviato ad un lookup service (es. una DHT), usando sempre un circuito TOR in modo da proteggere sempre l'anonimato dell'hidden service

# Hidden Services

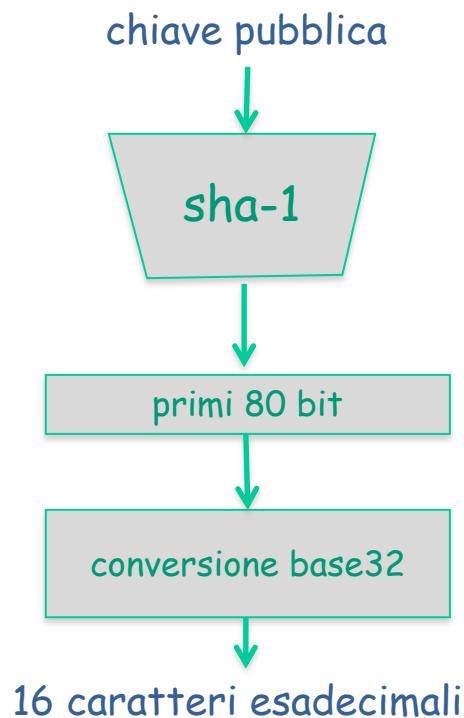


Un client per contattare un hidden service ha bisogno di conoscere il suo indirizzo x.onion. Dopo, il client può avviare la creazione della connessione scaricando il descrittore dal lookup service che ha il compito di pubblicizzare il servizio nascosto nella rete Tor.

# URL x.onion

<http://go2ndkjdf7whfanf.onion>

- E' un hash della chiave pubblica
- Meglio evitare nomi non correlati alla chiave pubblica



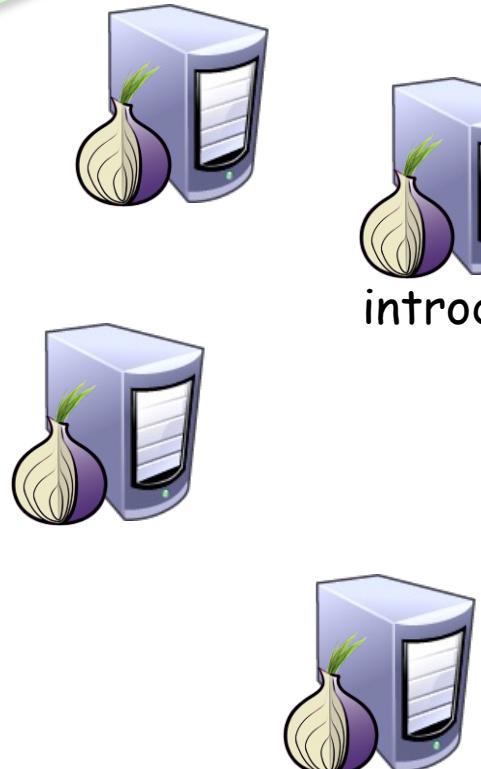
# Hidden Services

Descrittore servizio

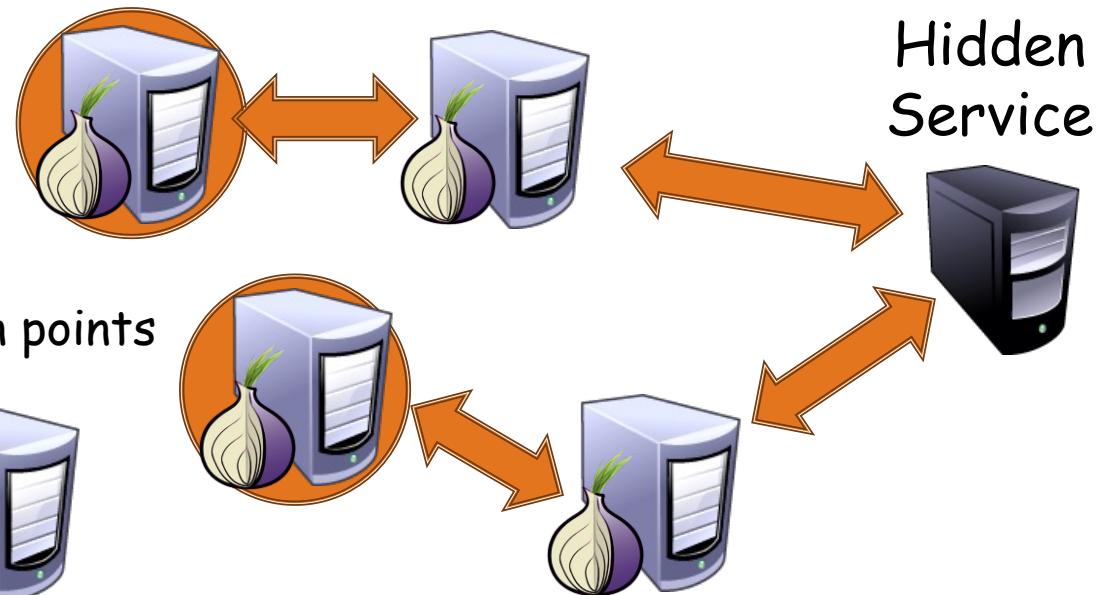
- Nome servizio
- Chiave pubblica
- Introduction points
- Firma



lookup service



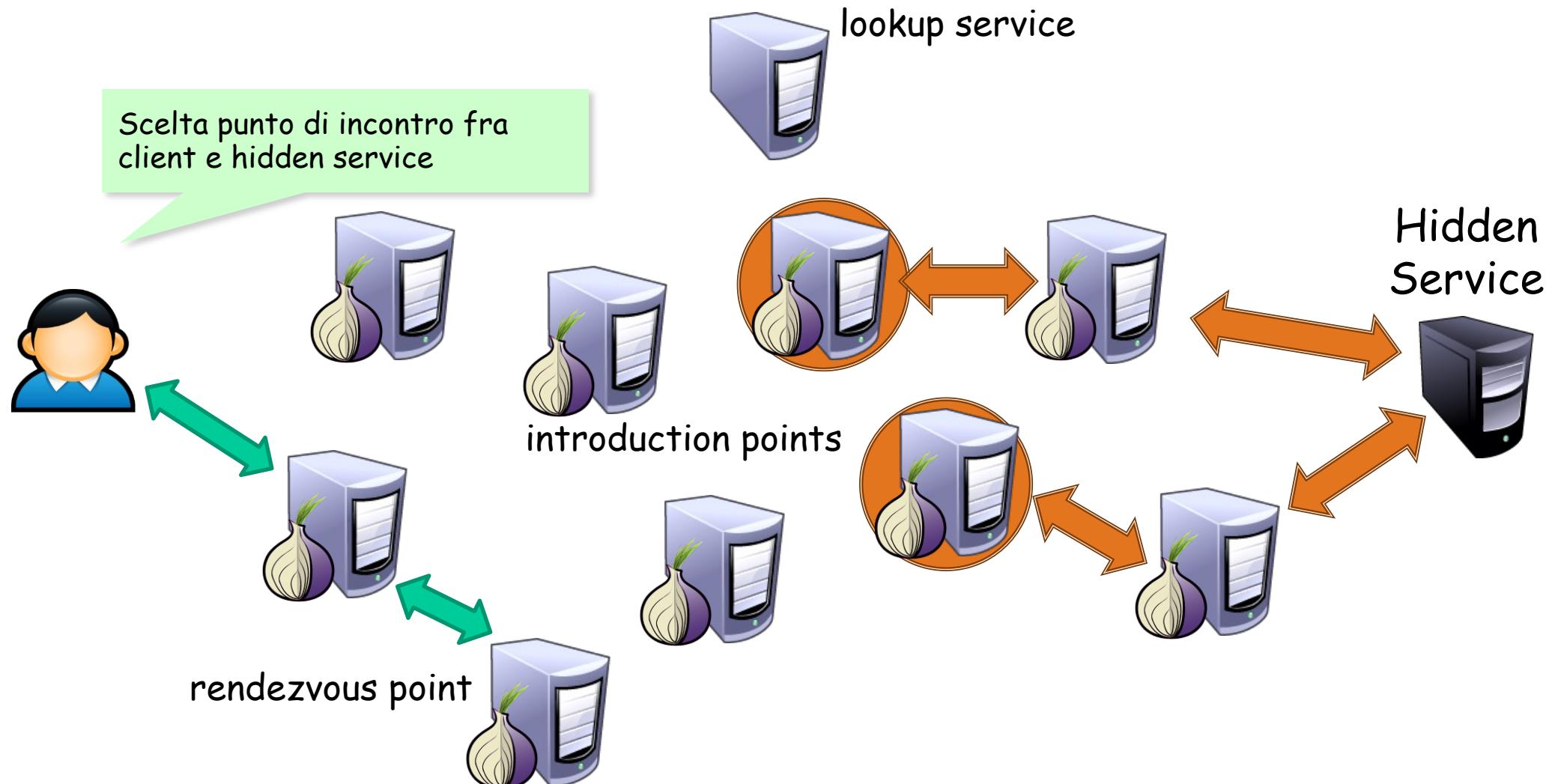
introduction points



Hidden  
Service

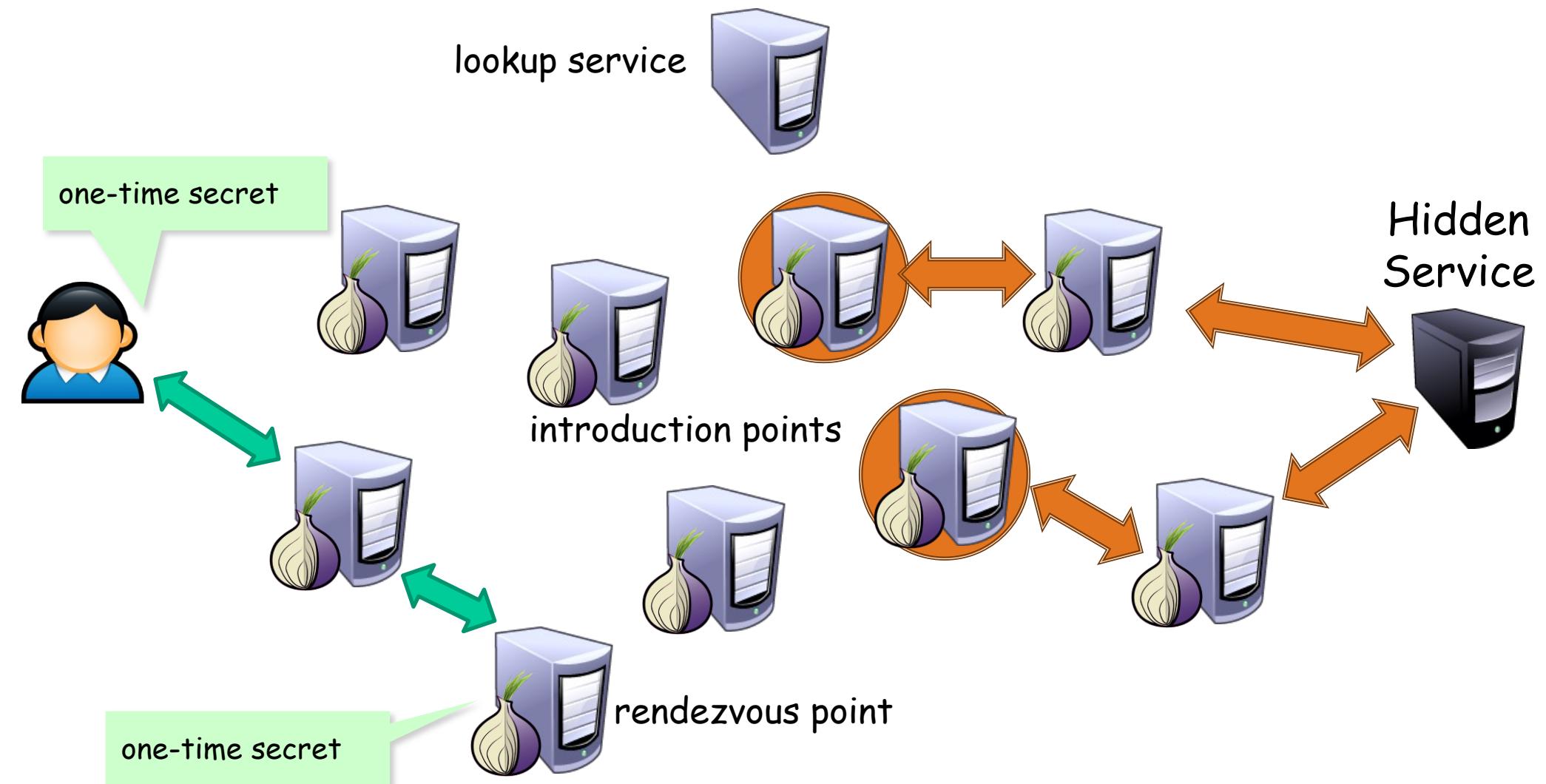
Il client conosce ora dal descrittore la lista degli introduction point da contattare e la chiave pubblica da usare per comunicare con loro.

# Hidden Services



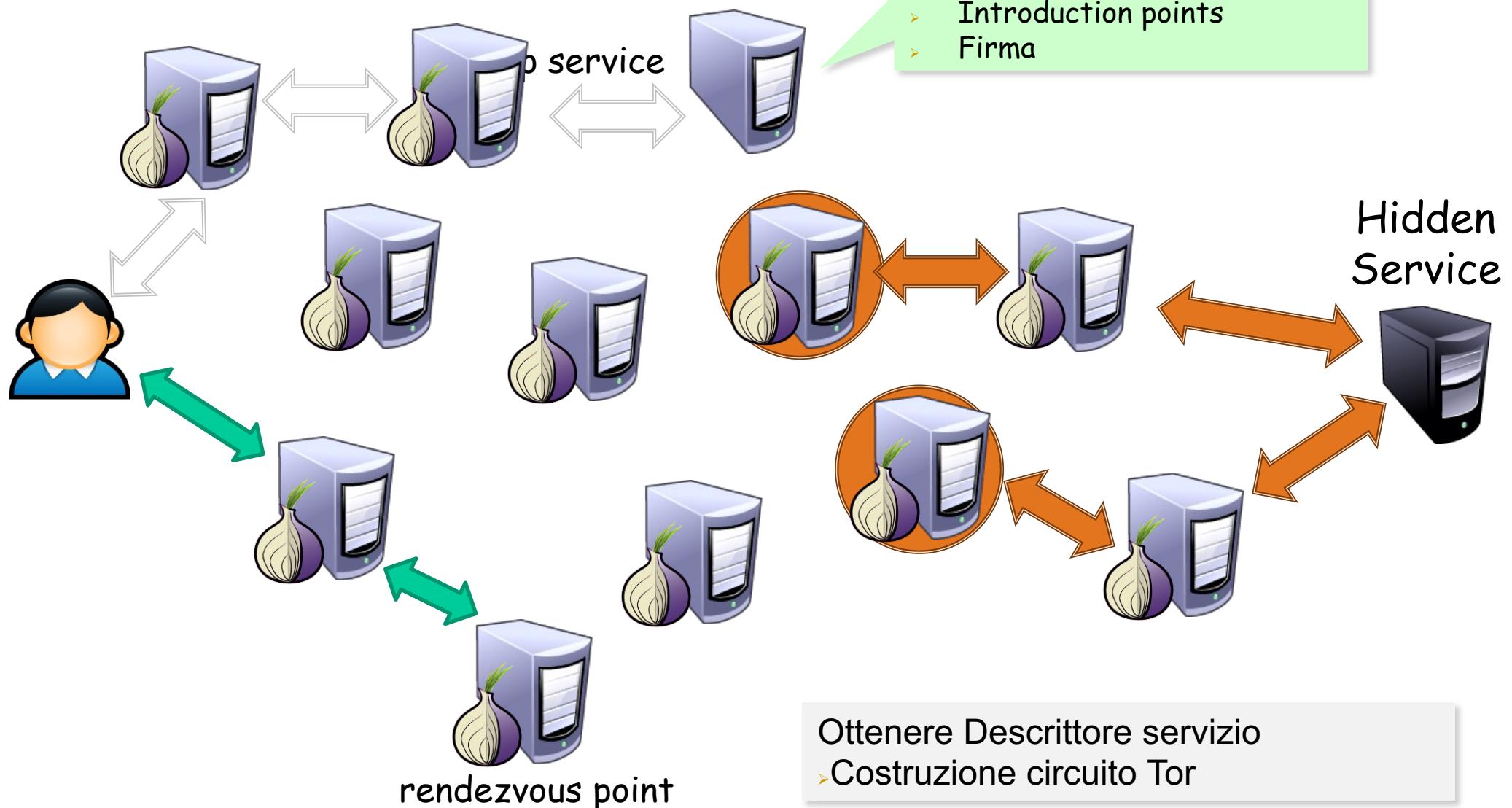
- Il client crea un circuito TOR verso un altro relay scelto a caso e gli chiede di fungere da rendezvous point (punto d'incontro tra client e server)

# Hidden Services



Il client manda al rendezvous point un "One-time secret" (rendezvous cookie).

# Hidden Service



# Hidden Services

Introduction MSG:

- . Indirizzo RP
  - . One time secret
- Da mandare a HS

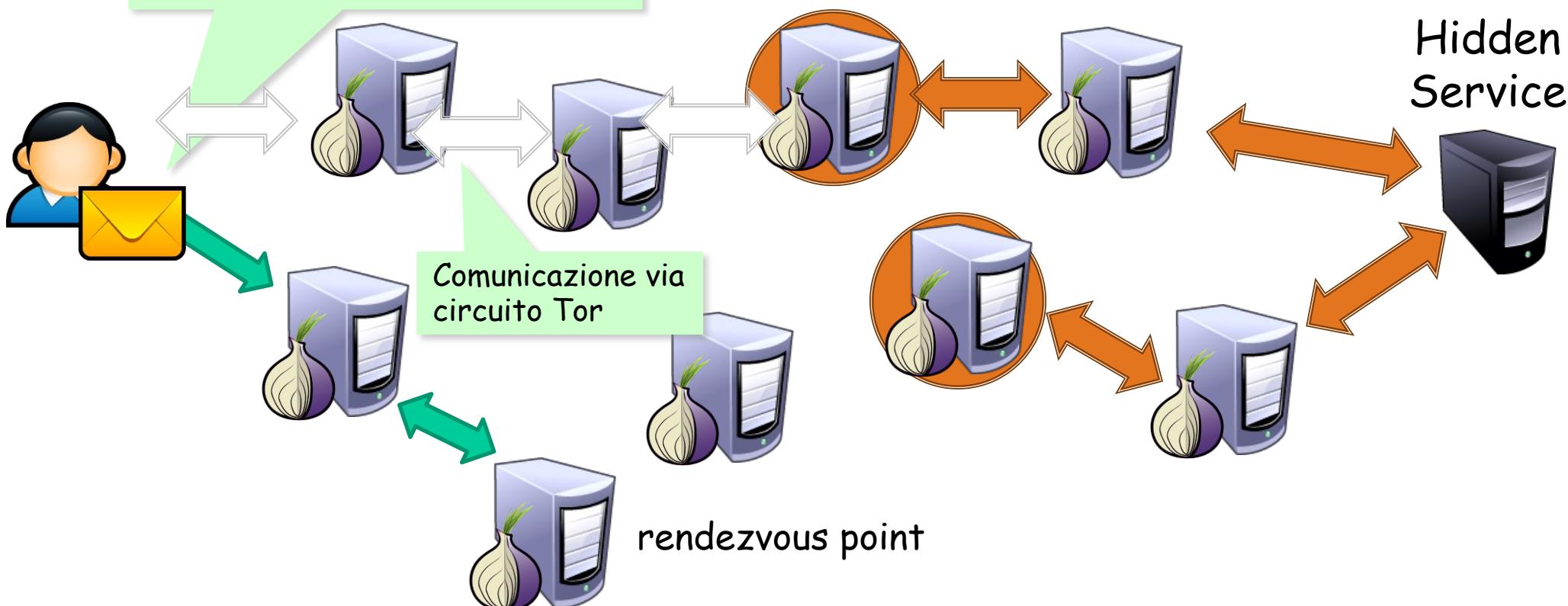


Hidden Service



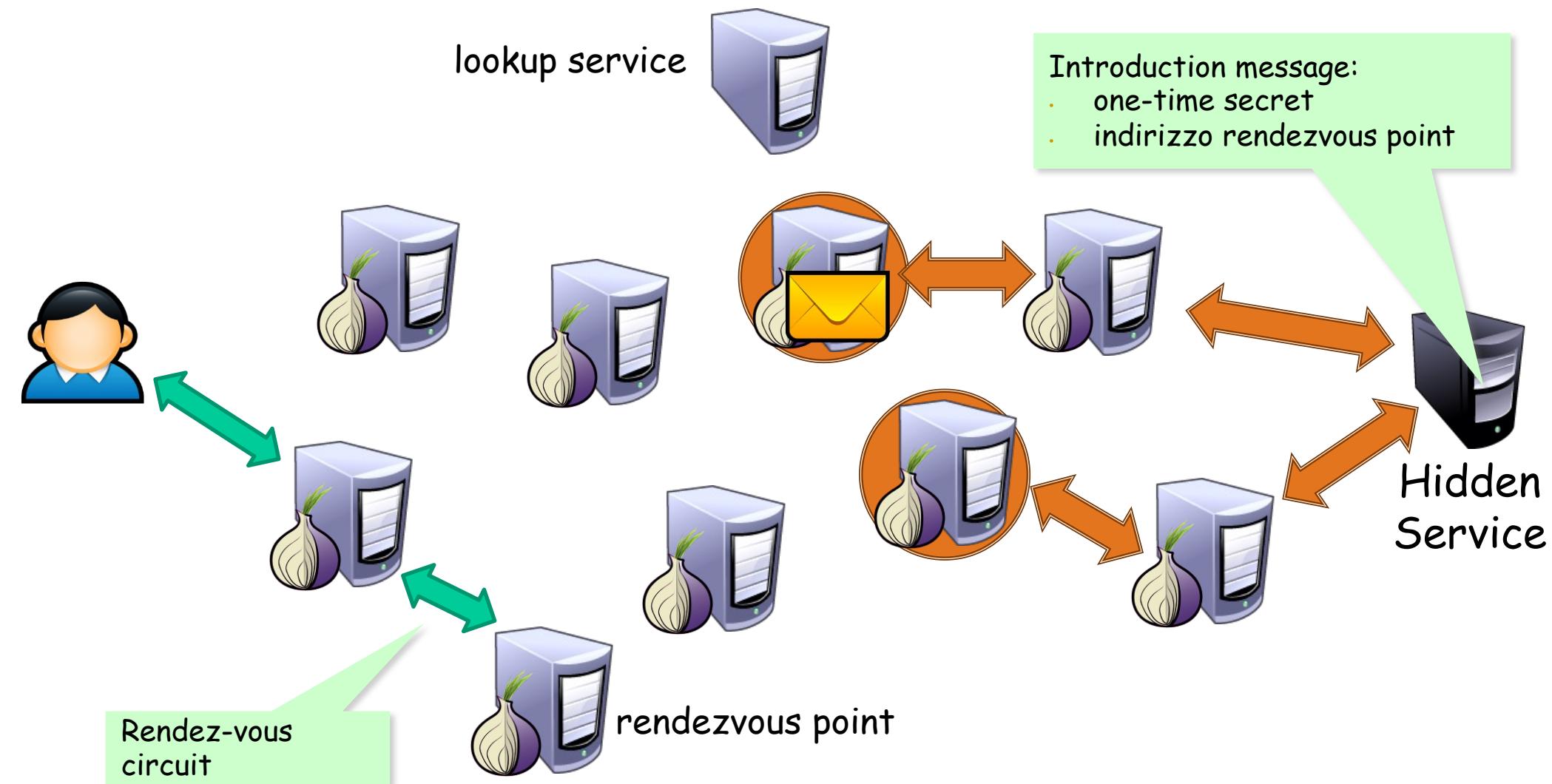
Comunicazione via  
circuito Tor

rendezvous point



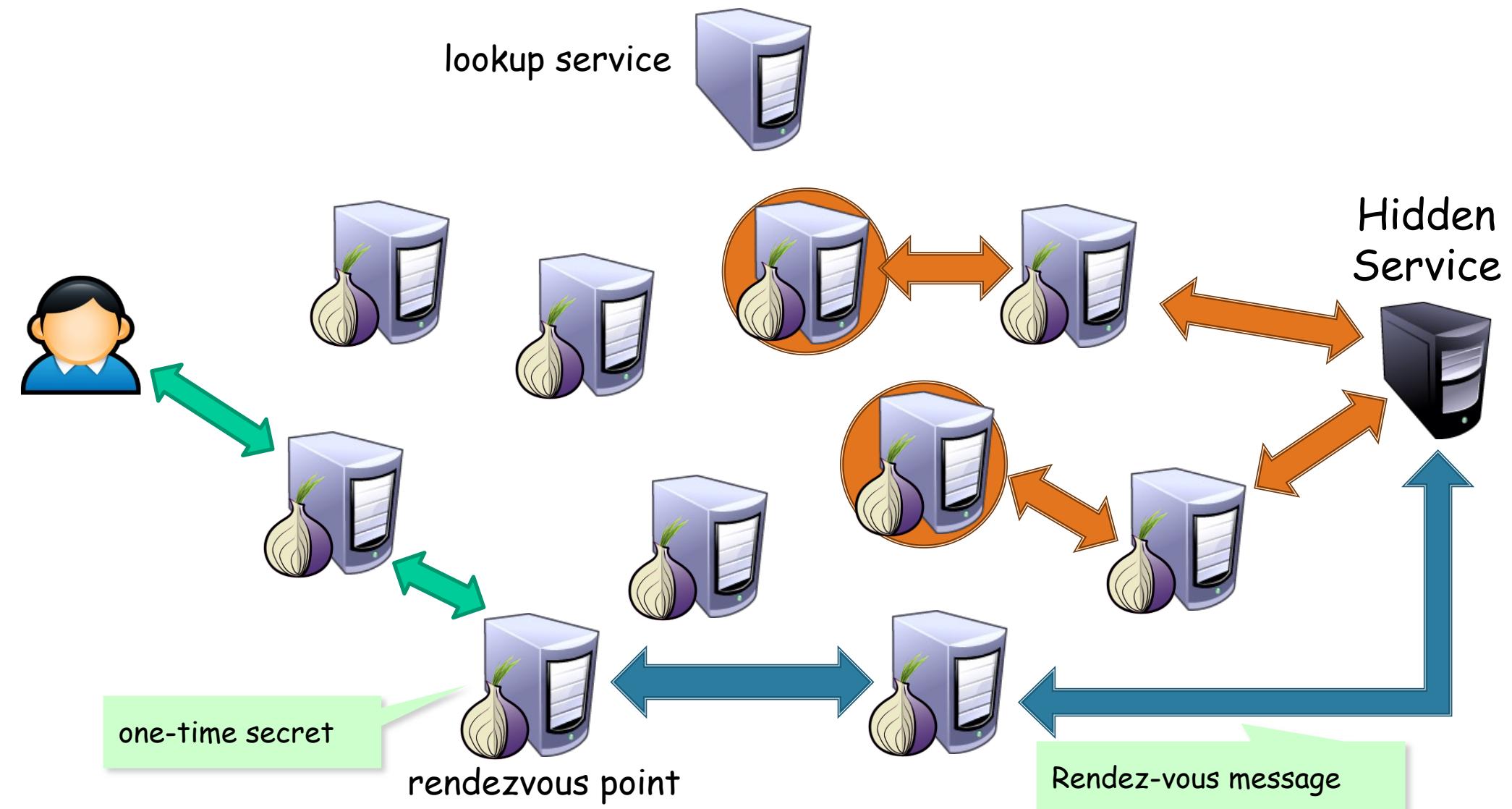
il client costruisce un "introduction message" (cifrato con la chiave pubblica dell'hidden service), contenente l'indirizzo del rendezvous point ed il "one-time secret" e lo invia a uno degli introduction point, chiedendo che venga consegnato all'hidden service.

# Hidden Services



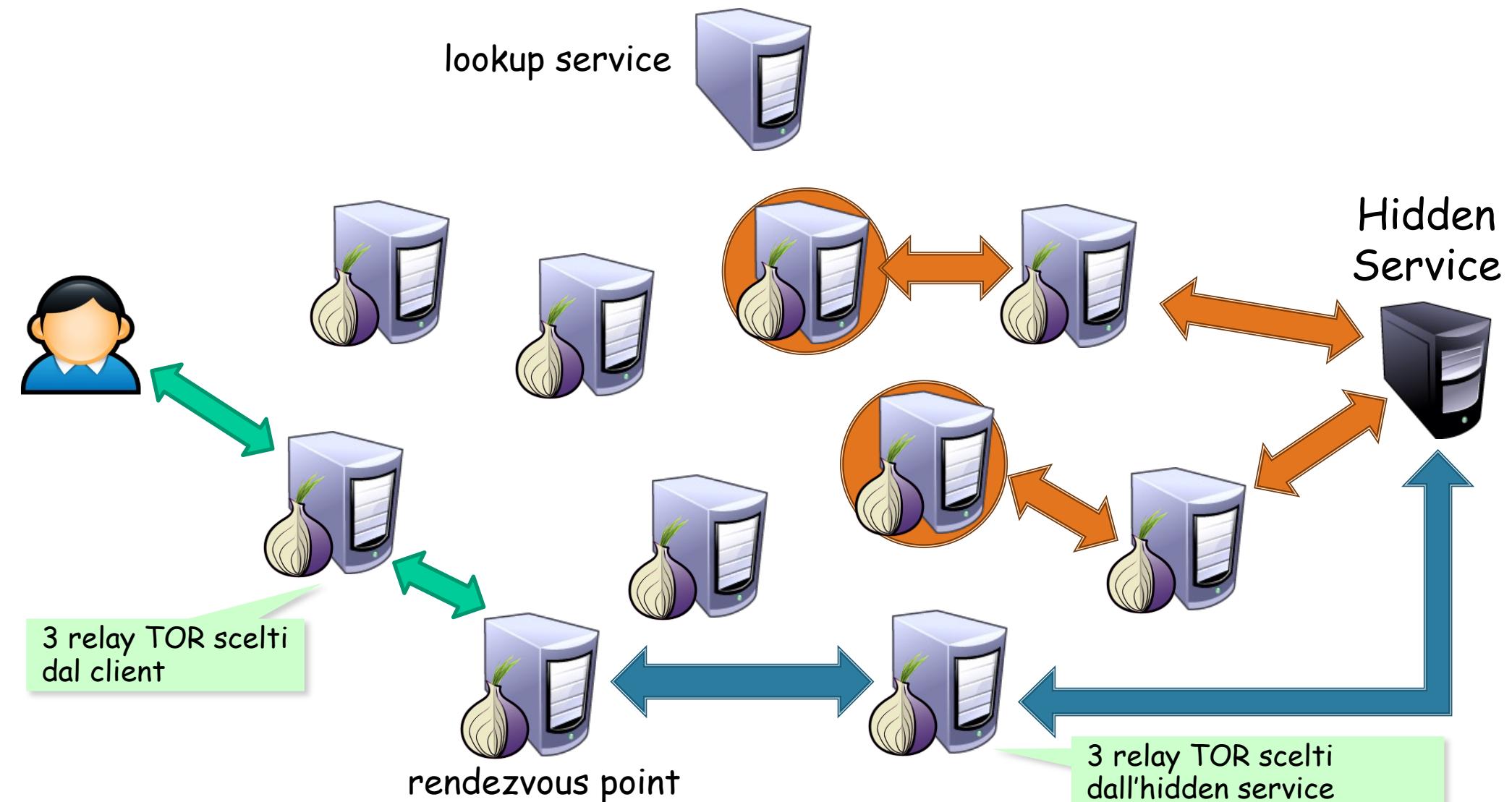
L'introduction point inoltra il messaggio di introduzione ricevuto dal client all'hidden service che lo decifra e scopre l'indirizzo del rendezvous point ed il "One-time secret" contenuto.

# Hidden Services



L'hidden service crea un circuito verso il rendezvous point e gli invia il “One-time secret” in un rendezvous message.

# Hidden Services

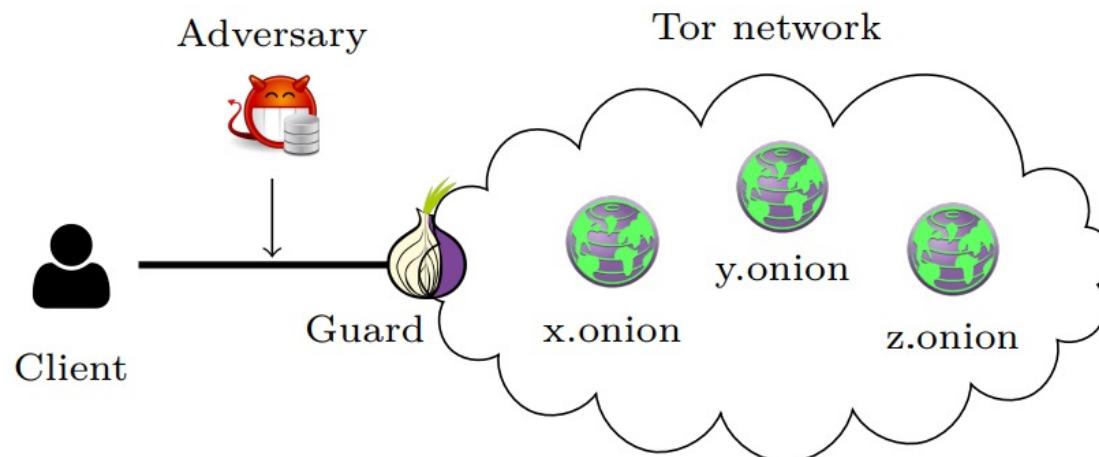


Il rendezvous point notifica al client che la connessione è stata stabilita con successo

Il client e l'hidden service possono usare i loro circuiti verso il rendezvous point per comunicare

# Sicurezza Hidden Services

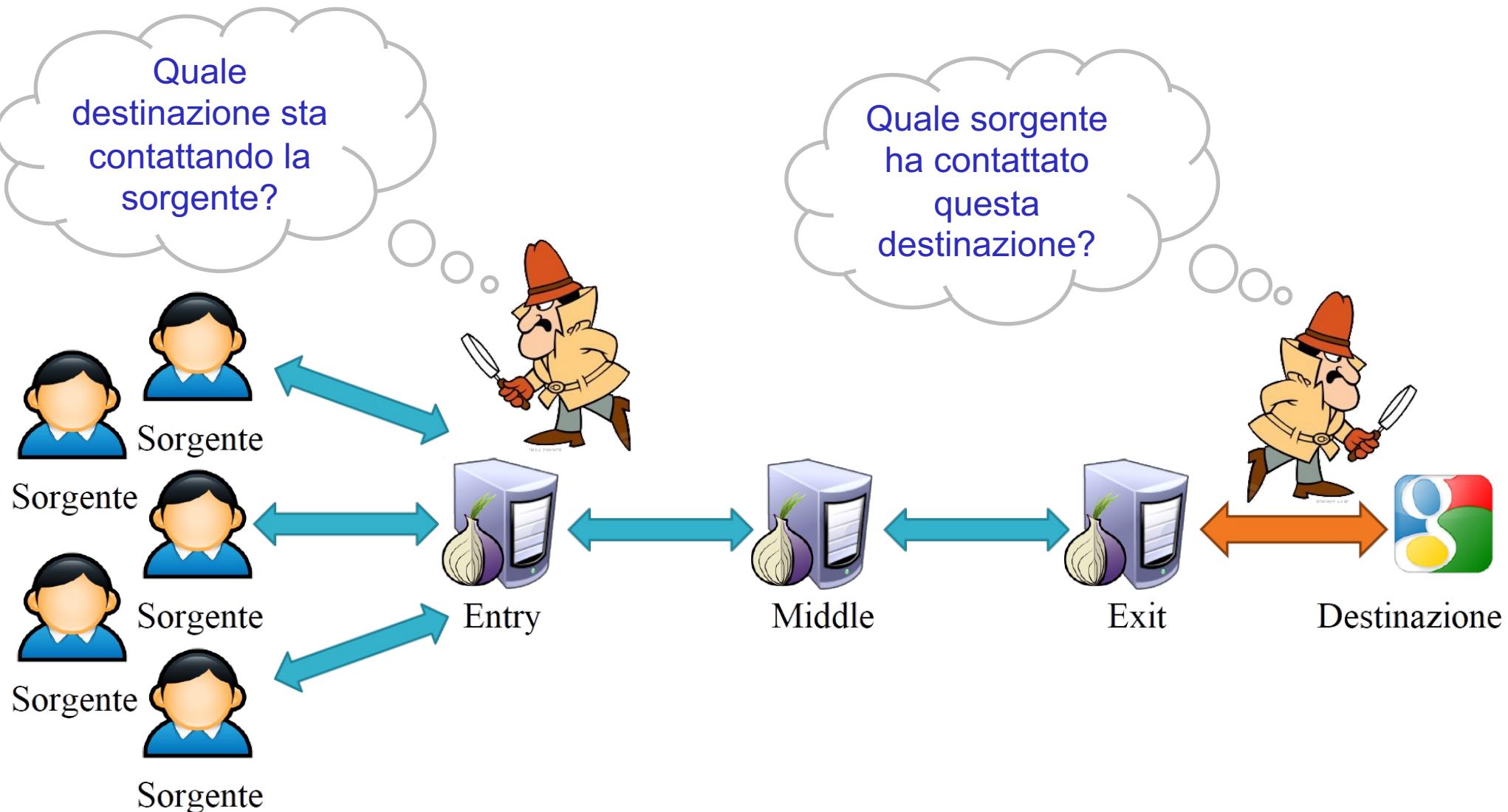
- Il lookup service e gli introduction point non conoscono l'IP dell'hidden service
- La comunicazione avviene sempre tramite circuiti Tor: nessuno può collegare l'invio dell' “introduce message” all' IP del client che rimane anonimo.
- Nel rendezvous point i dati decifrati provenienti dal client vengono nuovamente cifrati per essere consegnati al server e viceversa.
- La compromissione del rendezvous point permetterebbe di accedere ai dati “in chiaro” ma non comprometterebbe l'anonimato non potendo ottenere informazioni sul mittente e sul destinatario.



# Attacchi a Tor

- Exit node maliziosi che monitorano il traffico
- Il contenuto fornito dall'exit node non è validato, può essere modificato dall'exit node
- Tor assicura anonimizzazione layer 3.. **e il resto?**  
Application layer
  - javascript
  - ActiveX
  - plug-in vari (Shockwave)
  - vulnerabilità del browser

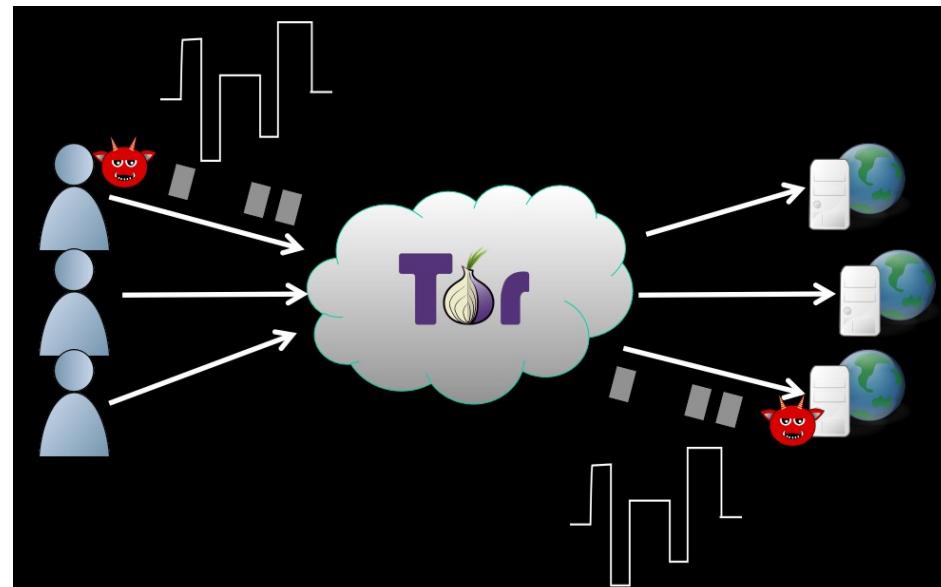
# Attacchi a Tor



- **Obiettivo:** Associare sorgente e destinazione attraverso l'analisi del traffico ai due punti di osservazione

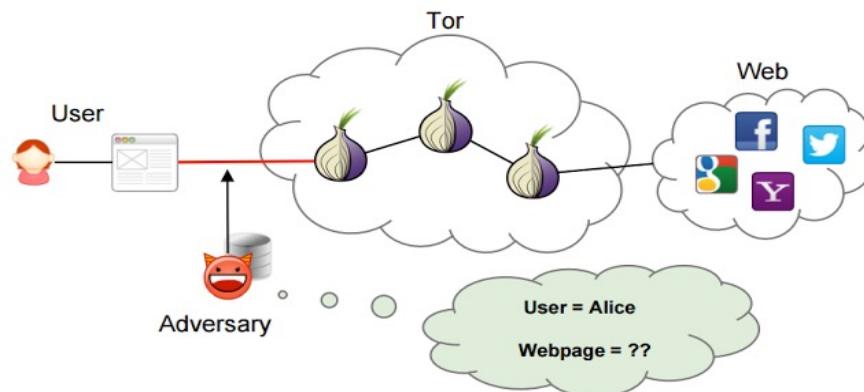
# Correlazione

- **Correlazione temporale:** un attaccante che è in grado di osservare il traffico sull' Entry Guard e sull'Exit node, può prelevare informazioni sui pacchetti e sui tempi di invio/interarrivo e correlare il traffico osservato rompendo l'anonimato.
- **Correlazione sulla dimensione dei pacchetti:** tale attacco è simile al precedente ma, anzichè osservare i tempi di risposta in ingresso e in uscita, l'attaccante prende in considerazione i dati relativi al numero e alla dimensione di pacchetti scambiati. Se riesce a inferire pattern simili può correlare i due nodi.



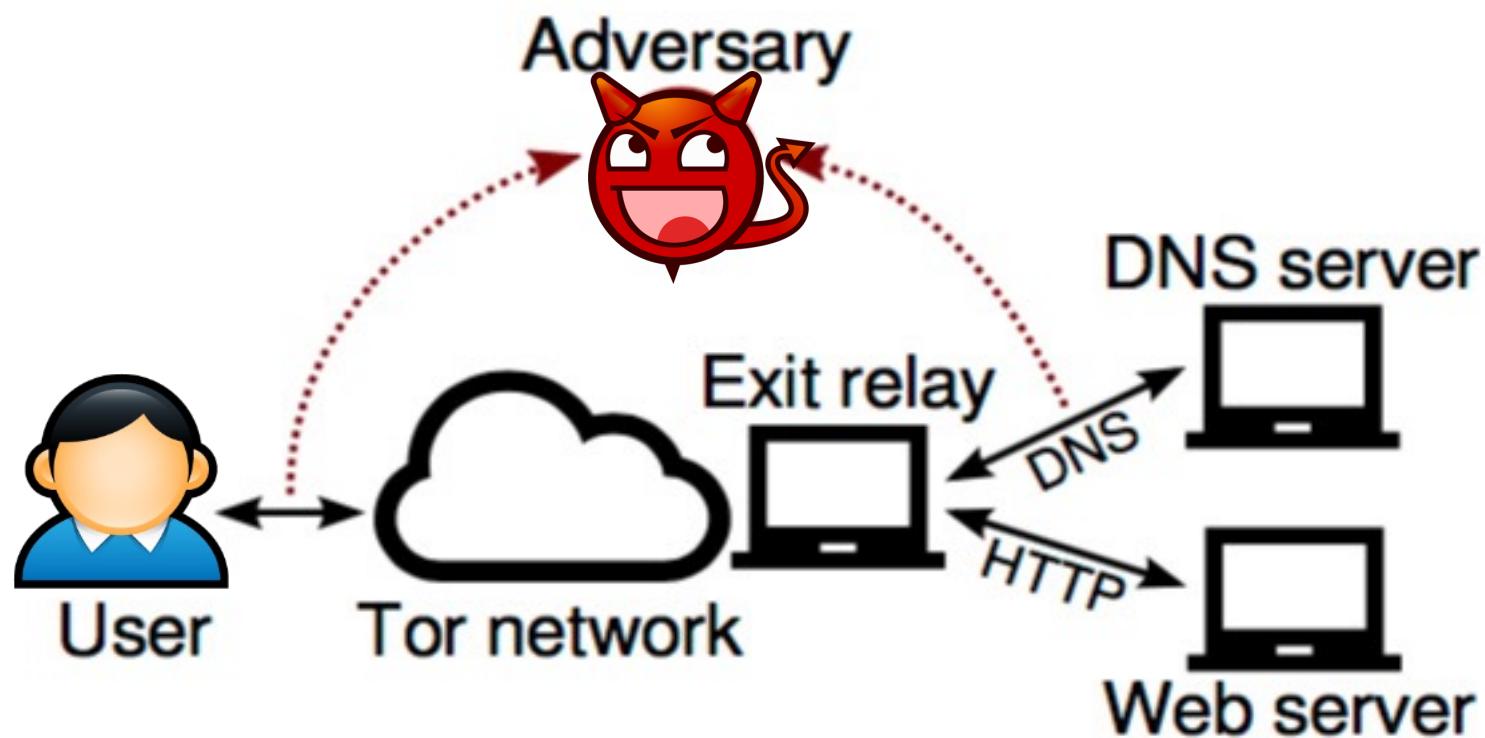
# Osservazione pattern di traffico

- Osservare il traffico in uscita di un utente Tor anche se non rileva la sua destinazione o il contenuto dei messaggi trasmessi, può indicare il tipo di traffico:
  - Es. Il traffico HTTP è piuttosto intervallato e poco voluminoso. Non si tratta di un attacco facile da portare a termine poichè Tor accorpa più flussi di dati in un singolo circuito.
- Il traffico che attraversa la rete Tor è cifrato, ma non è detto che lo sia all'uscita dall'Exit node.
  - Es: Se il traffico in entrata è HTTP, quando l'Exit node toglie l'ultimo strato di cifratura, avremo di nuovo traffico HTTP in uscita e per l'attaccante risulterebbe molto facile catturare le richieste e prelevare informazioni importanti come: User Agent, COOKIE e parametri GET e POST.



# Monitoraggio DNS

- Molti software continuano ad effettuare richieste DNS dirette, senza usare il proxy Tor.
- Ciò compromette l'anonimato perché rivela ad un osservatore le richieste DNS fatte, e quindi le destinazioni finali delle connessioni.

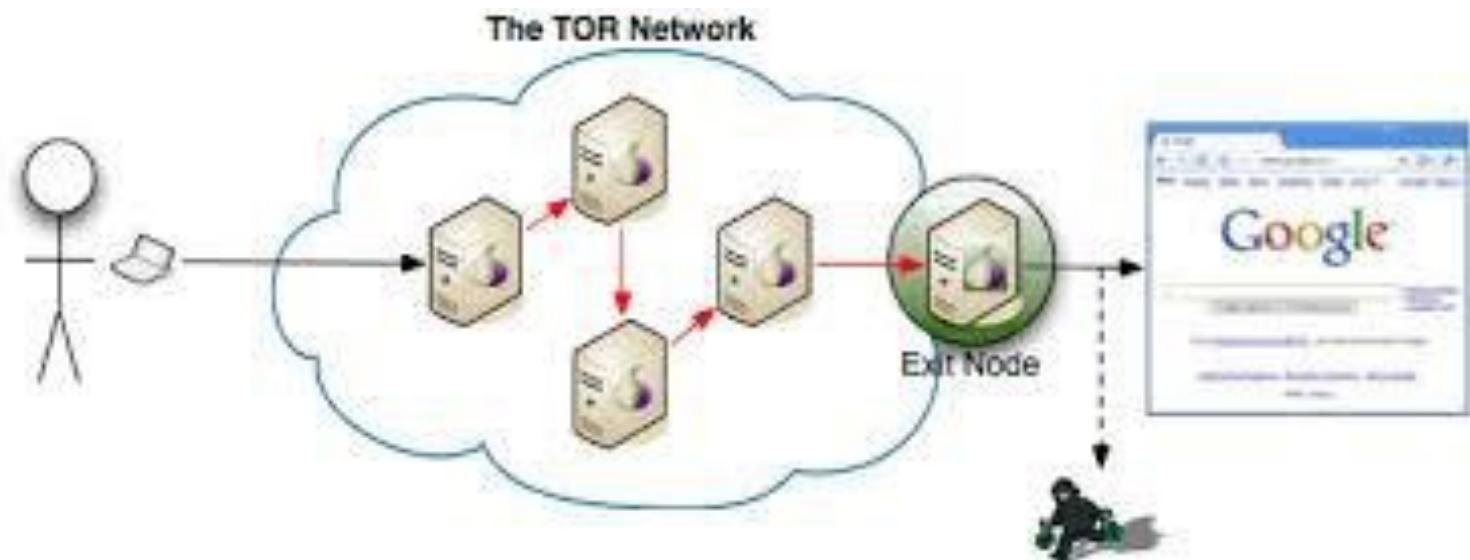


# Estrazione comportamento

- Tor è altamente configurabile, è possibile configurare i tempi di rotazione dei circuiti, scegliere nodi in base alla larghezza di banda, ecc.
- Un attaccante può clusterizzare gruppi di utenti osservandone il comportamento.
- I gruppi con minor numero di individui sono facilmente identificabili.

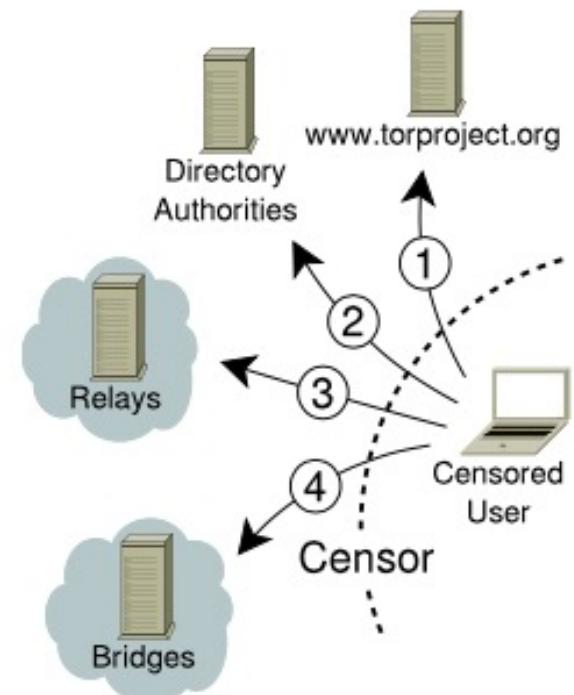
# Fingerprinting

- L'attaccante conosce un set di dati riguardo i tipi di browser e le loro risposte, la taglia dei file e i pattern di accesso verso alcuni siti web, dunque, confronta queste informazioni con il traffico generato da un nodo client.
- L'attaccante è in grado di correlare l'ingresso con l'uscita se la generazione di traffico combacia con i pattern noti.



# Blocco rete Tor

- Bridges potrebbero non bastare per evitare di essere bloccati
  - *Deep Packet Inspection classifica i flussi del traffico Internet a secondo dei protocolli per riconoscere e filtrare i frames Tor*
  - Possibile bloccare tutto il traffico cifrato
- Strategia governo Iran (febbraio 2012):
  - Deep Packet Inspection
  - Blocco degli indirizzi IP e porte
    - ad es., 86.59.30.36 (torproject.org), porta 443 (https)
  - Filtro con keyword
    - ad es., “tor”



# Offuscamento traffico Tor

Tor usa **pluggable transport** per la comunicazione tra client e bridge

- Traffico Tor trasformato in altro tipo di traffico
- Un offuscatore trasforma il traffico Tor in modo che appare come altro protocollo
  - BitTorrent, HTTP, streaming audio, etc.
- Un deoffuscatore dalla parte del ricevente estrae il traffico Tor



**Offuscatori:**

- obfs2
- obfs3
- FTE
- scramblesuit
- meek
- Flashproxy

# Mixing di traffico

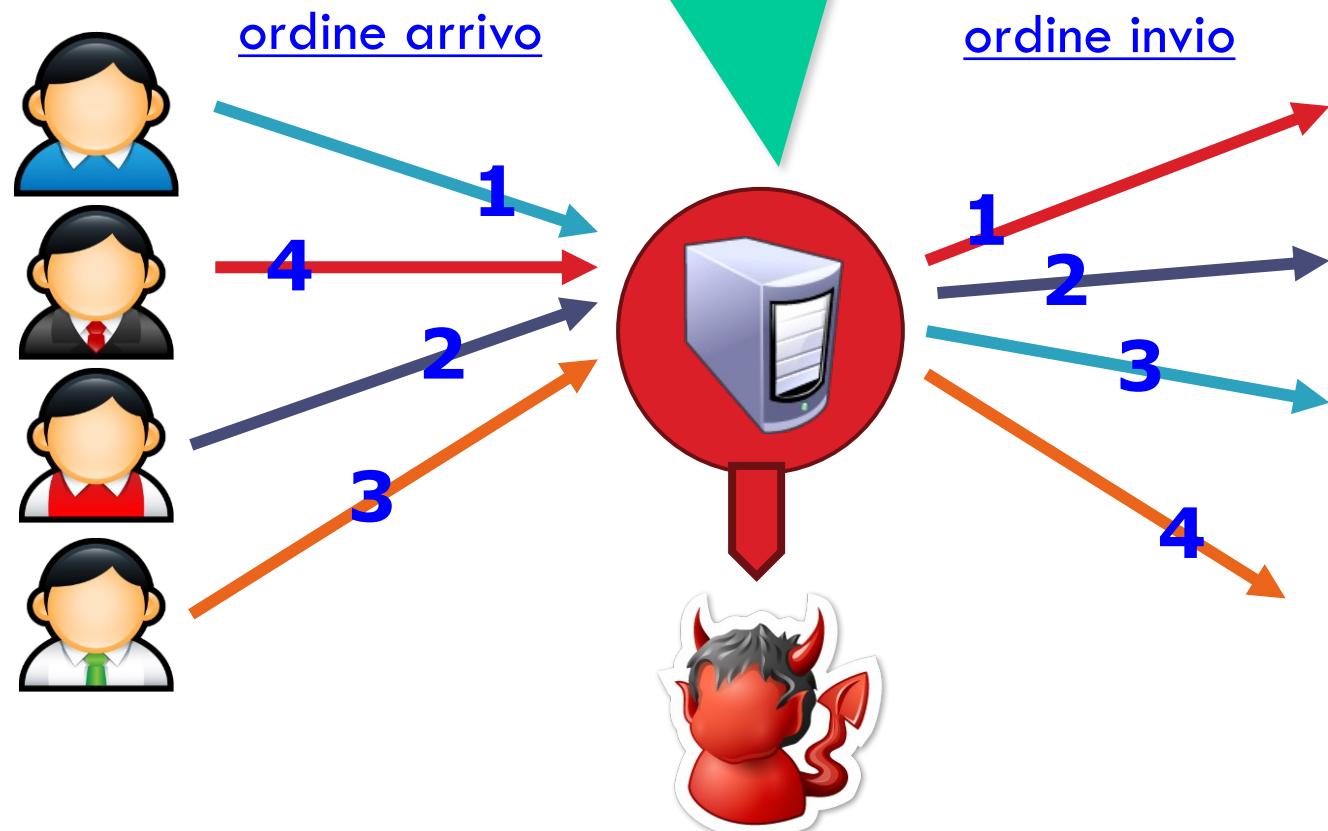
- Evitare attacchi basati sullo studio dell'andamento dei flussi di traffico per correlare sorgente e destinazione

- Messaggi artificialmente ritardati
- Si rende difficile la correlazione temporale

## Problemi

- Molto traffico
- Aggiunta latenza ai flussi

> Mix colleziona messaggi for  $t$  secondi  
> Messaggi sono mischiati casualmente ed inviati in un ordine differente

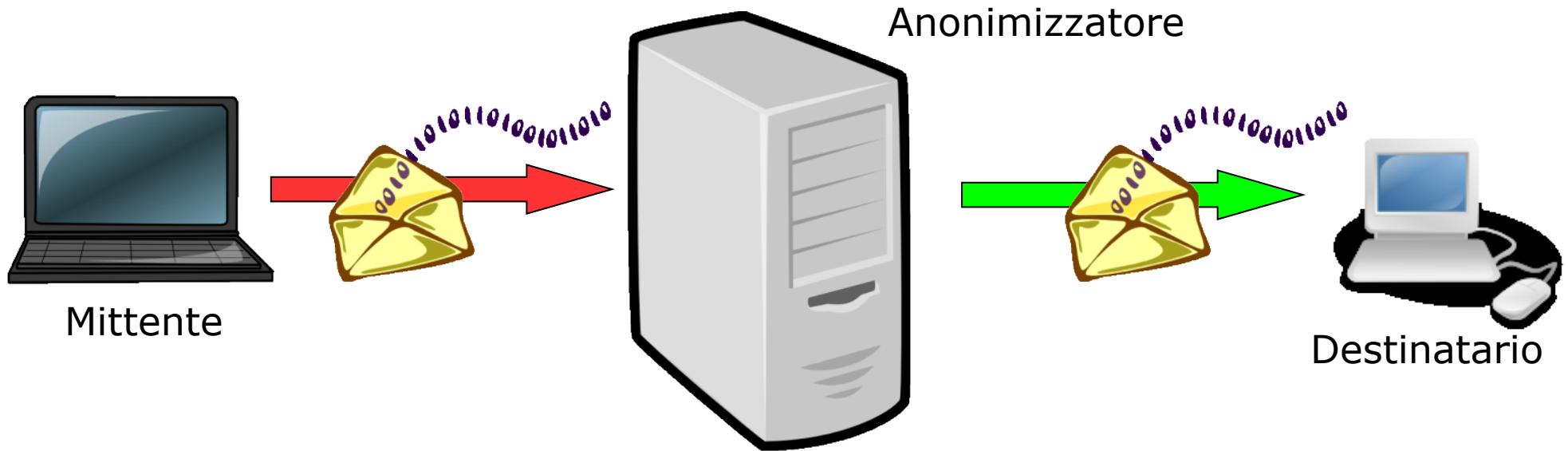


# Email Anonime

- All'interno ci sono informazioni che ci **identificano** (headers)
- E' possibile capire **da dove** l'email è stata spedita e che percorso ha fatto
- Esistono tecnologie per rendere confidenziali le email
  - Crittografare il contenuto di una email
  - La firma elettronica garantisce che una email proviene effettivamente dal mittente legittimo
  - Esistono vari tipi di meccanismi per la firma digitale
  - Problema dell'associazione macchina-uomo: Web of trust o Certification Authority
- E per garantirne l'**anonimato**?...



# Idea: remailer



Header della mail:

```
Received: from [192.168.0.100]
(host228-234-dynamic.14-87-r.retail.telecomitalia.it [87.14.234.228])
by giovanni.lonerunners.com (Postfix) with ESMTP id B771F4001
for <lol@lists.lonerunners.net>; Thu, 5 Jul 2007 12:45:53 +0200 (CEST)
Message-ID: <468CCBB9.7070501@lonerunners.net>
Date: Thu, 05 Jul 2007 12:45:13 +0200
From: Alessandro Tanasi <alessandro@lonerunners.net>
User-Agent: Thunderbird 1.5.0.12 (X11/20070604)
To: Lot of Fun Multimedia <lol@lists.lonerunners.net>
Subject: [Lol] Tartarughe wifi
```

Ciao, vista la tartaruga?

Header della mail:

```
Received: from [foobar]
[foobar [127.0.0.1]]
by giovanni.lonerunners.com (Postfix) with ESMTP id B771F4001
for <lol@lists.lonerunners.net>; Thu, 5 Jul 2007 12:45:53 +0200 (CEST)
Message-ID: <468CCBB9.7070501@lonerunners.net>
Date: Thu, 05 Jul 2007 12:45:13 +0200
From: Nessuno

To: Lot of Fun Multimedia <lol@lists.lonerunners.net>
Subject: [Lol] Tartarughe wifi
```

Ciao, vista la tartaruga?

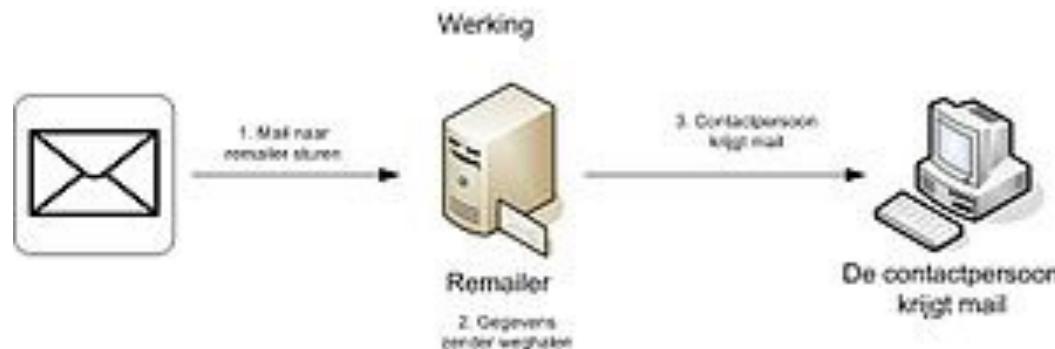
# Remailer

- Sistemi conosciuti che gestiscono l'anonimato della messaggistica email
- Tipologie:
  - Tipo 0: “Penet”
  - Tipo 1: “Cyberpunk”
  - Tipo 2: “Mixmaster”
  - Tipo 3: “Mixminion



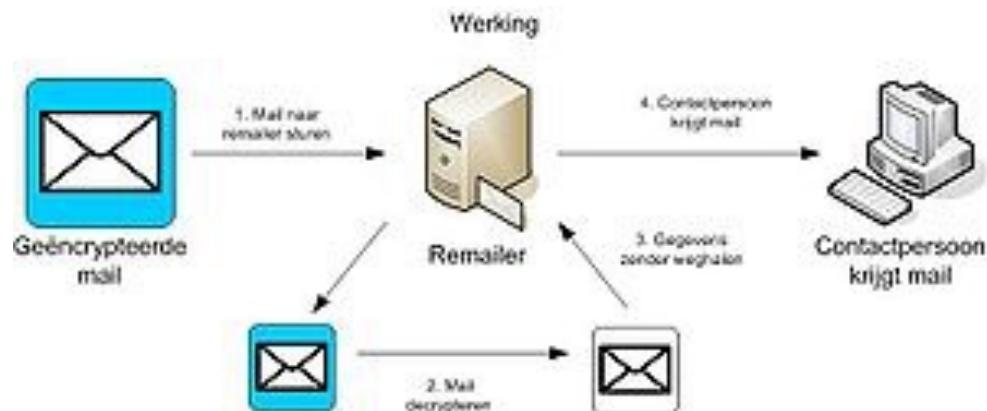
# Remailer di tipo 0

- Ricevono posta opportunamente formattata
- Inviano la posta che ricevono al destinatario togliendo le informazioni relative al mittente
- E' stata la prima implementazione
- Facilmente attaccabili:
  - violazione del server
  - tecniche di analisi temporale del traffico
  - azione legale che richiede i file di log all'amministratore



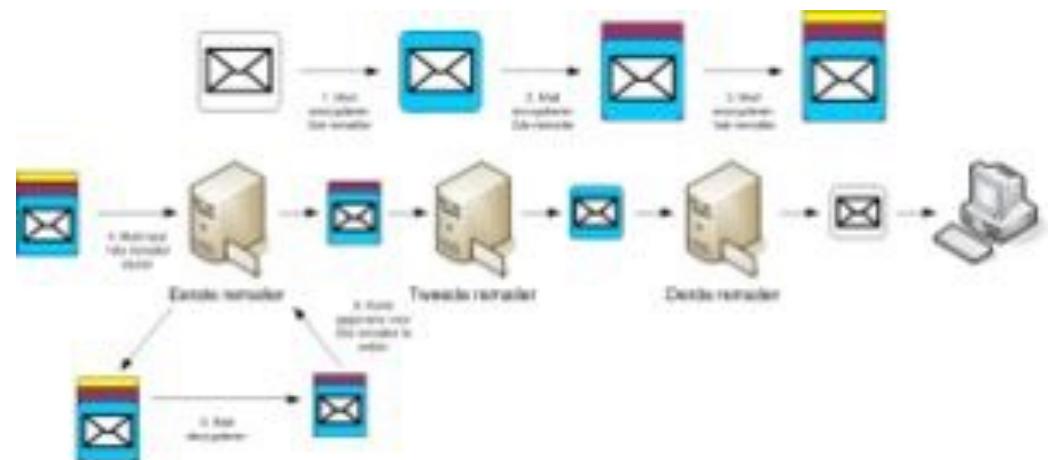
# Remailer di tipo 1

- Remailer usati in catene di almeno 3 elementi
- Utilizzo di metodi crittografici a chiave pubblica
- Crittografa il messaggio in successione per ogni remailer specificato
- La compromissione di  $n-1$  elementi della catena non è sufficiente
- Comunque attaccabile con tecniche di analisi temporale del traffico



# Remailer di tipo 2

- I messaggi vengono resi di dimensione uniforme: vengono spezzettati e frammenti sono di dimensione uguale
- I messaggi sono spediti con un ritardo casuale: sono mantenuti in una coda che viene svuotata casualmente
- Generazione di traffico casuale: vengono generati e scambiati falsi messaggi che l'ultimo remailer della catena poi scarterà
- Utilizzano ancora SMTP e una rete separata per la distribuzione delle chiavi crittografiche
- Suscettibili ad attacchi DoS



# Remailer di tipo 3

- Utilizzano un proprio protocollo client-server
- Al posto dell' SMTP usa delle connessioni SSL tra server e per accettare i messaggi dagli utenti. Supporta anche la ricezione di risposte anonime usando dei reply-block a uso singolo, "Single Use Reply Blocks" o "SURBs". I messaggi inviati e quelli di risposta risultano indistinguibili.
- Sono flessibili e permettono l'interfacciamento con altre reti di tipo diverso (Mixmaster, MMS)
- Directory distribuita gestita automaticamente dai remailer per la gestione delle chiavi crittografiche
- Rete in fase di sviluppo ma comunque già utilizzabile

<http://www.mixminion.net/>

<http://en.wikipedia.org/wiki/Mixminion>

# Vulnerabilità dei Remailer

- Gli attacchi per scoprire l'origine di un messaggio variano in base al tipo di remailer:
- Analisi statistica del traffico
- Analisi temporale del traffico in ingresso e uscita
- Violazione logica o fisica del server
- Generazione di Denial of Service
- ... violazione del client
- Non è detto che un utente usi reti di tipo 3 per spedire mail, le vecchie reti sono ancora in utilizzo!

# Server di pseudonomi

- Problema: chi riceve un messaggio attraverso un remailer non può rispondere al mittente
- Pseudonimo: **indirizzo fittizio**, non riconducibile ad un indirizzo reale, tramite cui si possono far giungere messaggi al destinatario senza conoscerne l'identità
- Anonimato anche **all'indietro**, mittente e destinatario possono scambiarsi email in modo totalmente anonimo



[http://en.wikipedia.org/wiki/Pseudonymous\\_remailer](http://en.wikipedia.org/wiki/Pseudonymous_remailer)

# Remailing Newnym

- Usando metodi di crittografia si permette ad un utente di creare un indirizzo fittizio (pseudonimo)
- I messaggi di quell'utente usciranno dal remailer con tale pseudonimo
- Si può caricare un reply block: sequenze di istruzioni crittografate su come far pervenire il messaggio all'indirizzo reale tramite una serie di remailer

```
=====
Config:
From: Smith
Reply-Block:
::
Anon-To: AAA@remailer.aaa.com
Latent-Time: +0:00
Encrypt-Key: password_a
::
Encrypted: PGP
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

/S3vZw+95ZuCZfqxKE0XrgZXzOEwfoyBcpVvf9Pb9D19TqEMTmmL/Jp11xcxmbJ2
vRoiG8ZhXs4r3E8liFsNtMMf6CUAsdv2ZoX1Hw==
=Bla3
-----END PGP MESSAGE-----
```

# Pseudonym server

- Alice non sa come farsi rispondere da Bob senza rivelargli la sua reale identità.
- Infatti, quando Bob riceve il messaggio, lo vede arrivare da un remailer (in questo caso Freddy).
- Se Alice vuole ricevere una risposta, non può che mettere il suo indirizzo di posta dentro il corpo del messaggio; perdendo in questo modo l'anonimato.
- E' necessario ricorrere a uno **pseudonym server**

# Pseudonym server

- . Uno pseudonym server consente all'utente di registrare un proprio alias (nym o pseudonimo), che viene associato ad una casella di posta elettronica.
- . L'aspetto innovativo sta nel fatto che né il nym né la casella postale sono direttamente riconducibili all'utente stesso.
- . Ciò è possibile perché tutti i messaggi che transitano in entrata e in uscita da e per il nym server, passano prima attraverso una serie concatenata di anonymous remailer.

# Pseudonym server

- Alice fa entrare in ballo un nuovo attore, Ziggy, che fa di mestiere lo **Pseudonym server**.
- Alice crea una catena di buste simile a quella precedente, ma che vada da Ziggy stesso, attraverso magari i soliti Danny, Eddy e Freddy, fino a lei.
- La invia poi a Ziggy, dicendogli di crearle lo pseudonimo di AliBaba.
- Ha l' accortezza di fare questo invio utilizzando una catena di remailer, diversa dalla solita, supponiamo Harry, Indy e John

# Pseudonym server

- Quando Ziggy riceve la busta, che contiene il contenuto precedentemente encapsulato preparato da Alice (definito come “**reply block**” e la richiesta di creare per lei uno pseudonimo, controlla sulla sua lista se questo pseudonimo esiste già, ed in caso negativo usa il reply block per farle avere l’ok alla creazione dello pseudonimo AliBaba.
- Il reply-block è paragonabile a un insieme concentrico di istruzioni, che possono essere lette solo una per volta e solo da uno specifico remailer alla volta perché crittate con la sua chiave pubblica.
  - Si può indicare il proprio reply-block una volta per tutte mentre si crea ilnym sulnym server, oppure si può cambiarlo di volta in volta per ogni messaggio:
  - la prima pratica è la più diffusa, ma bisognerà almeno controllare che i remailer attraverso cui si vuole far transitare il proprio messaggio siano sempre attivi.

# Pseudonym server

- Si noti che Ziggy non ha la più pallida idea di chi Alice sia, perchè tutti i messaggi che riceve provengono da una catena di remailer, e quelli che manda usano il reply block

**Percorso della richiesta di Alice:**

**Alice -> Harry -> Indy -> John ->  
Ziggy**

**Percorso dell' OK di Ziggy:**

**Ziggy -> Danny -> Eddie -> Freddy  
-> Alice**

# Pseudonym server

- A questo punto Alice possiede un indirizzo fittizio (uno pseudonimo, appunto) nel dominio di posta di Ziggy; se l' indirizzo di Ziggy fosse Config@Ziggy.org, lo pseudonimo di Alice sarebbe AliBaba@Ziggy.org.
- Per scrivere ad Alice, Bob mandera' un messaggio normale ad AliBaba@Ziggy.org.
- Ziggy ricevera' il messaggio, ed in base allo pseudonimo, selezionera' l' opportuno reply block da usare per inoltrare il messaggio attraverso la catena di remailer scelta da Alice.
- Né il server di pseudonimi né i singoli remailer usati saranno mai in grado di stabilire una connessione tra la casella reale e la casella *nym* (a meno che non si usi un solo remailer).

# Pseudonym server

- Lo pseudonimo di Alice, AliBaba@Ziggy.org puo' essere usato anche per spedire messaggi.
- Se Alice vuole spedire un messaggio a Bob dal suo pseudonimo, usera' una catena di remailer qualsiasi per spedire a Ziggy il messaggio e l' indirizzo a cui spedirlo a nome del suo pseudonimo.
- A tutti gli effetti Bob può addirittura non sapere che l' indirizzo da cui riceve ed a cui spedisce posta sia diverso da uno "normale".

**Percorso del messaggio da Alice a Bob:**

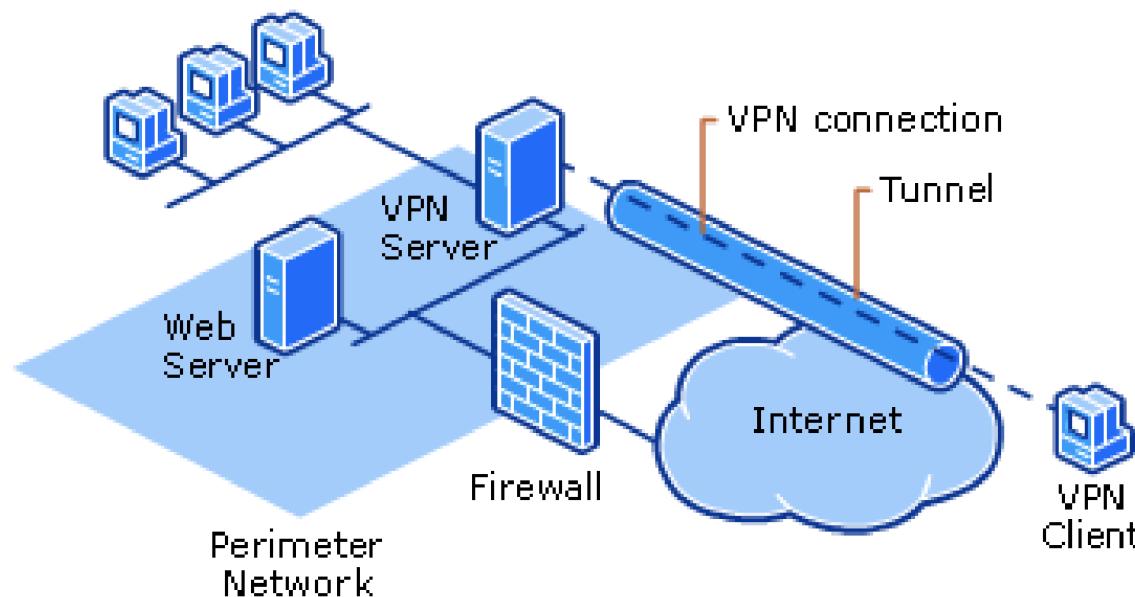
**Alice -> Harry -> Indy -> John -> Ziggy -> Bob**

**Percorso della risposta di Bob ad Alice:**

**Bob -> Ziggy -> Danny -> Eddie -> Freddy -> Alice**

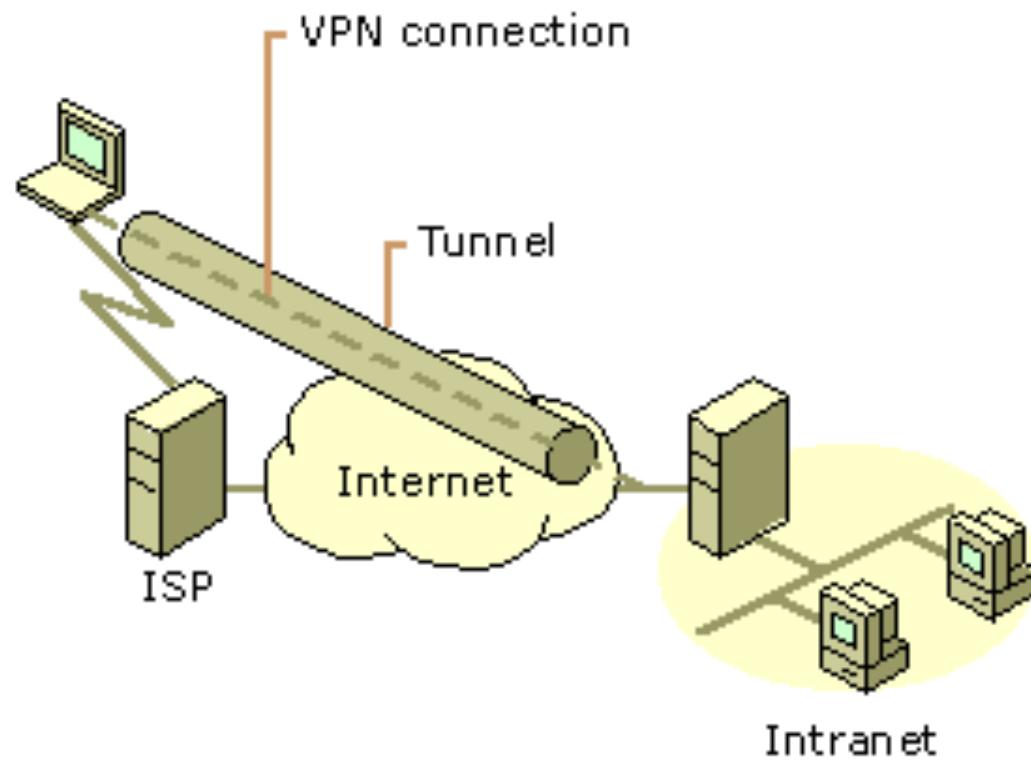
# VPN: concetti generali

- Una VPN realizza un canale di comunicazione end-to-end su rete pubblica, (es. Internet) anziché usare una linea dedicata.
  - E' possibile connettere singoli utenti o intere reti
    - User-to-Site
    - Site-to-Site
  - Consente la connessione remota da una rete esterna, attraverso canali condivisi, a una LAN privata (**LAN Extension**) attraversandone il perimetro ed emulando un collegamento dedicato



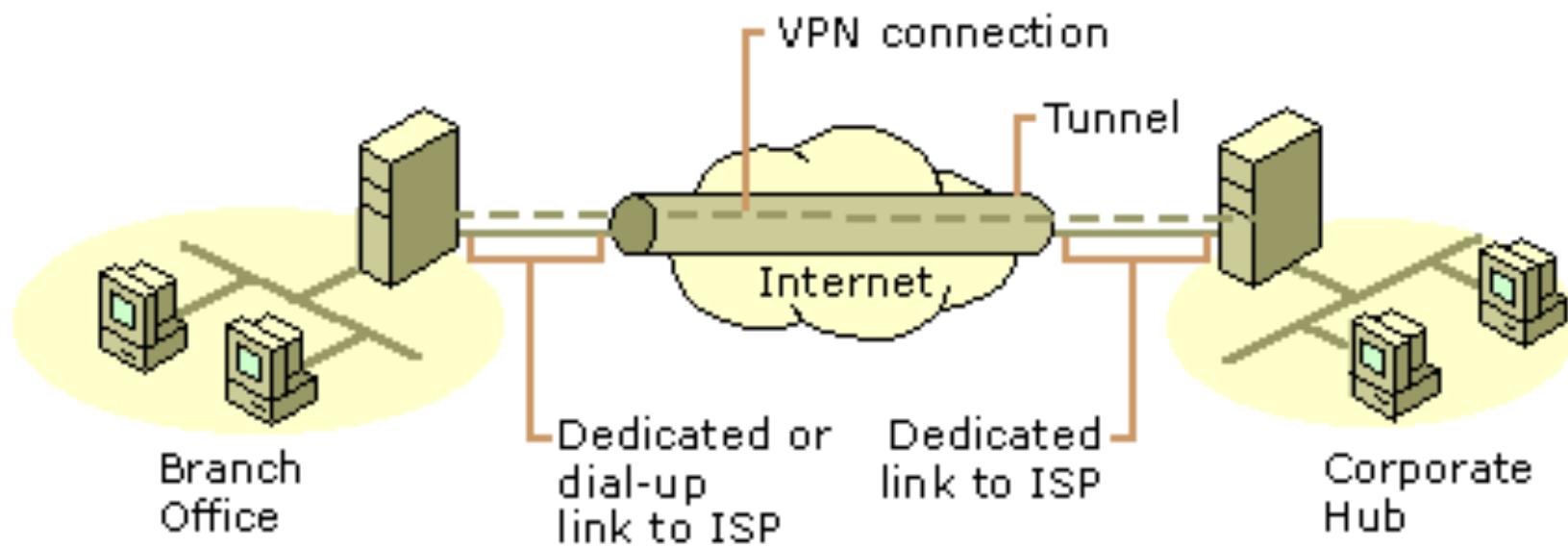
# VPN: usi comuni (1/3)

1. Accesso remoto a una LAN privata attraverso Internet



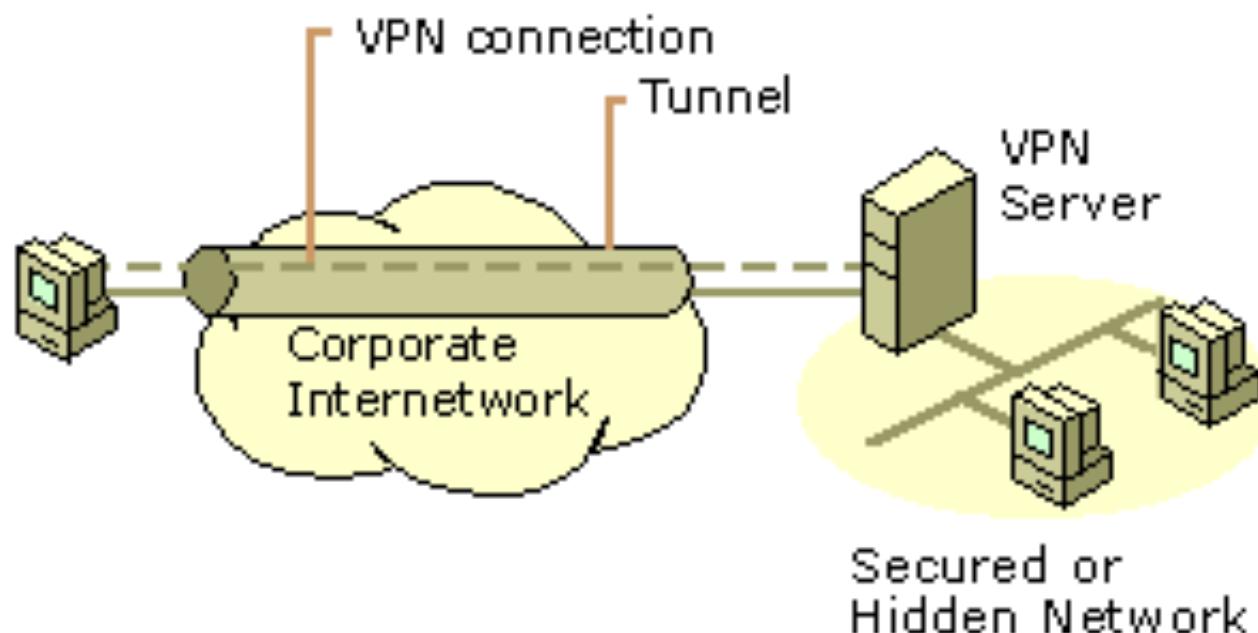
# VPN: usi comuni (2/3)

## 2. Connessione fra reti private attraverso Internet (Site to Site VPN)



# VPN: usi comuni (3/3)

3. Connessione fra Computers attraverso una Intranet (simile a 1.)

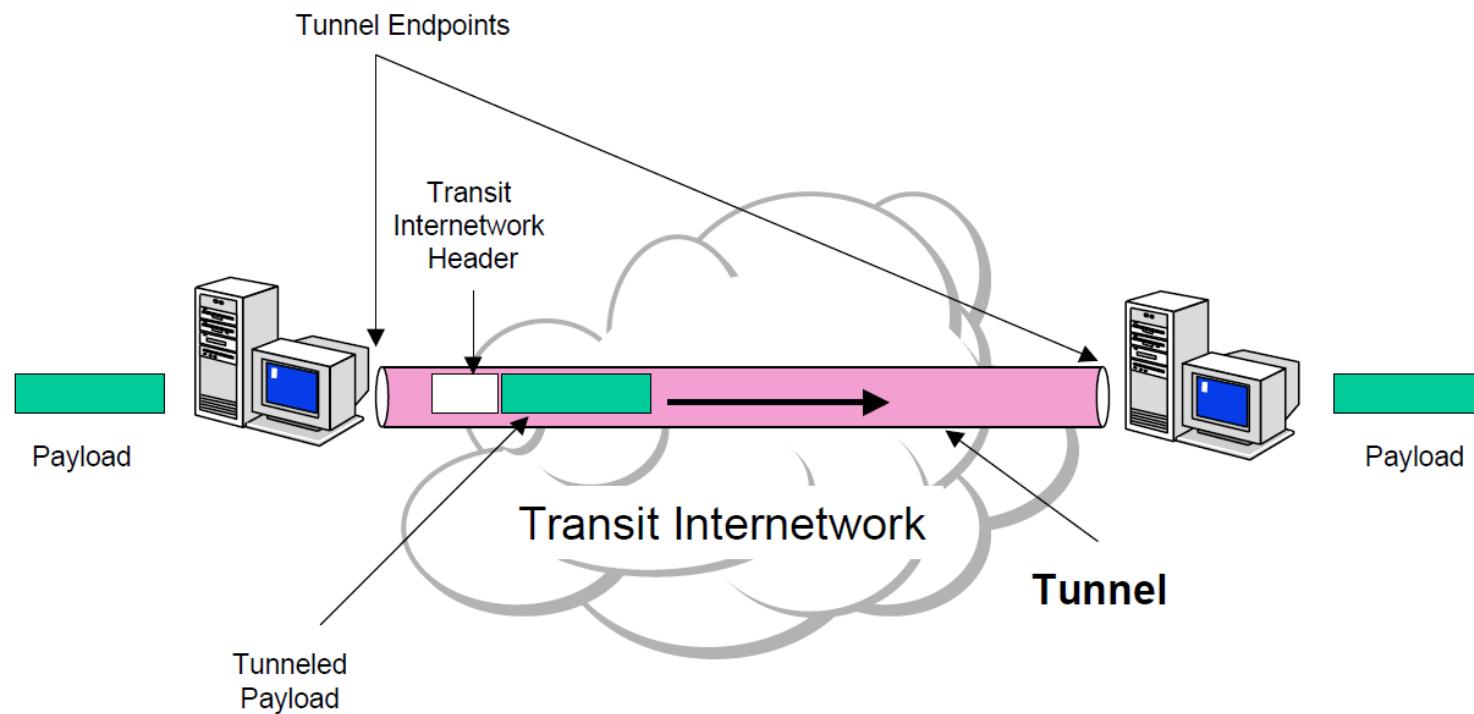


# Perchè usare VPN?

- Economicità
  - Realizzare connettività remota tramite modem dial-up o tramite linee dedicate, è costoso.
- Scalabilità
  - Estendere una rete attraverso unai linea dedicata è complesso e non scala facilmente
  - Una VPN su rete pubblica è Facile da amministrare.
- Sicurezza
  - Una VPN garantisce confidenzialità e integrità end-to-end nonché mutua autenticazione fra gli estremi tramite tecniche crittografiche

# Il concerto chiave: Tunneling

- La VPN è costituita da una serie di connessioni punto a punto trasportate su Internet.
- Per ottenere il tunneling, i pacchetti IP sono incapsulati come payload di altri pacchetti.



# Requisiti di base

- Autenticazione e controllo accessi
  - Verifica l'identità dell'endpoint consentendo la connessione e l'accesso alla rete solo a entità autorizzate
  - Garantisce funzioni di audit e accounting per dimostrare in maniera non ripudiabile chi ha acceduto, quali informazioni ha scaricato etc..
  - Usa certificateiX.509, pre-shared key, etc.
- Riservatezza
  - E' garantita l'assoluta confidenzialità dei dati che transitano sulla rete pubblica
- Integrità dei dati
  - Nessuno dall'esterno della VPN può alterare nulla senza essere rilevato
- Key Management
  - Genera e aggiorna le chiavi di crittografia per il client e il server.
  - Gestione chiavi per IP effettuata via : ISAKMP/Oakley, etc.

# Tipologie d'accesso

## WEB translation/proxy

1

- Applicaizoni e portali WEB
- Supporto nativo di applicazioni: RDP, Telnet, File sharing
- CLIENTLESS

3

## Network Connector

- Visibilità a livello IP
- Accesso completo alla rete
- Assegnazione IP interno
- Client scaricato (Diritti amministrativi)



2

## SSL application wrapper

- Applicazioni client/server TCP
- Active X – JAVA applet
- Distinzione Windows – Unixlike
- Local listener – Wrapper syscall
- Client/Applet scaricata
- Al primo accesso si installa ActiveX

# Conclusioni

- **Esistono sistemi anonimi** di comunicazione e per la difesa della privacy dei dati e sono pienamente utilizzabili
- Manca una loro diffusione e conoscenza
- Nel caso serva :) probabilmente alcuni sistemi possono essere manomessi o aggirati
- La **sensibilizzazione** sul problema privacy è fondamentale
  - ... *per far in modo che Internet rimanga libera per le generazioni future*