

# Servizi Onion

## Dalla teoria all'implementazione

**Leonardo Migliorelli<sup>1</sup>**  
[leonardo.migliorelli@studenti.unicam.it](mailto:leonardo.migliorelli@studenti.unicam.it)



# COMPUTER SCIENCE @ UNICAM

Tesi Unicam - Servizi Onion, dalla teoria dall'implementazione

- ▶ Relatore: Marcantoni Fausto  
<https://computerscience.unicam.it/marcantoni/>
- ▶ Studente: Leonardo Migliorelli Mat.113920  
leonardo.migliorelli@studenti.unicam.it

# INDICE

Introduzione

Onion Routing

Creazione del circuito

Chaum Mix

Tor § Onion v2

Network Design

Dimostrazione Wireshark

Implementazione

Pubblicizzare il servizio

# INTRODUZIONE

In questa tesi spiegheremo il funzionamento delle reti **Onion** e parleremo in particolare della rete **Tor**, in fine mostreremo com'è possibile implementare un **servizio Onion/Tor**.

Le reti Onion sono state create per risolvere le due più grandi vulnerabilità di Internet che gravano sulla **privacy** e **sull'anonimato**, ovvero l'**analisi del traffico** e le **intercettazioni**.

Una rete di questo tipo nasconde infatti gli indirizzi e il contenuto di ogni richiesta, consentendo all'utente di navigare in rete senza essere tracciato.

# ONION ROUTING



La rete Onion è una rete distribuita composta da nodi chiamati **Onion Router**, collegati tra loro tramite i circuiti creati dagli **Onion Proxy**.

Ogni pacchetto che passa nel circuito viene decriptato in maniera sequenziale dai relativi nodi fino ad arrivare all'exit node che si occupa di instradare il pacchetto nella rete Internet. Grazie a questo meccanismo nessun nodo conosce contemporaneamente l'indirizzo del mittente e del destinatario.

Il proxy onion è composto da tre componenti:

- ▶ Un proxy che genera e gestisce le connessioni, ha necessità di conoscere la topologia della rete Onion.
- ▶ **Application Specific Proxy**, si occupa di convertire le richieste delle applicazioni in pacchetti Onion.
- ▶ **Application Specific Privacy Filter**, un proxy opzionale che sanifica lo stream di dati rimuovendo informazioni che potrebbero identificare il client.

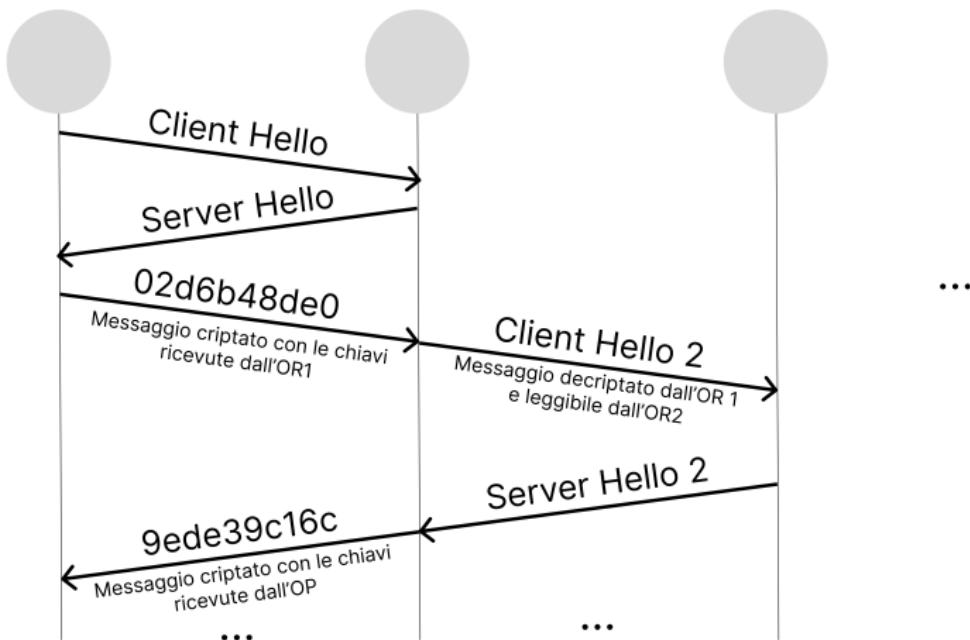


# CREAZIONE DEL CIRCUITO

La generazione del circuito è un processo **iterativo** e **progressivo** in cui il proxy server sceglie i nodi del circuito. A partire dal primo nodo viene instaurata una connessione **TLS** e vengono scambiate le **chiavi simmetriche** tramite un processo **asimmetrico** (in maniera simile allo scambio di chiavi di **HTTPS**), successivamente si usano queste chiavi per cifrare il messaggio che verrà inviato al primo nodo che lo decripta e inoltra al secondo nodo.

Questo processo continua fino all'exit node, a questo punto il circuito è completo e può iniziare a trasmettere i pacchetti.

## Onion Proxy      Onion Router 1      Onion Router 2



# CHAUM MIX

La rete Onion è basata sullo studio di David Chaum che propose un sistema di comunicazione anonima basato sulla crittografia. Nella sua conclusione una rete di questo tipo doveva avere le seguenti caratteristiche:

- ▶ **Sealing**, una tecnica con cui il messaggio viene annesso ad una stringa casuale prima di essere criptato per aumentarne la sicurezza.
- ▶ **Indirizzo non tracciabile**, un indirizzo generato dal client criptando quello reale, rendendone possibile la decriptazione solo dal primo nodo. Viene usato come indirizzo di risposta dal destinatario.

# TOR

La rete Tor è la più famosa implementazione di Onion, le principali migliorie sono state:

- ▶ Circuiti telescopici
- ▶ Utilizzo del protocollo SOCKS
- ▶ Controllo di congestione
- ▶ Directory server
- ▶ Controllo d'integrità end-to-end



L'obiettivo principale della rete TOR è quello di garantire l'anonimato e scoraggiare eventuali attaccanti, per questo la rete doveva essere semplice da usare, così da incrementare il numero di nodi.

# NETWORK DESIGN

La rete Tor è

una rete che esiste al di sopra delle reti esistenti,  
per questo viene chiamata **overlay network**.

I pacchetti che viaggiano  
nella rete Tor sono chiamati **celle**, hanno una  
dimensione fissa 512 bytes, e sono di due tipi:

- ▶ **Control**, gestisce il circuito
- ▶ **Relay**, trasporta stream dati

Applicazioni

Rete TOR

Trasporto (TCP)

Internet

Network Access

# CONTROLLO DI CONGESTIONE

Non potendo in una rete Onion cambiare un router quando non è più in grado di gestire il carico, è stato necessario implementare un controllo di congestione che blocca i circuiti o gli stream dati quando questi inviano troppi dati usando due finestre:

- ▶ Packaging window, tiene traccia del numero di celle che un OR può inviare all'OP
- ▶ Delivery window, tiene traccia del numero di celle che un OR può inviare fuori dalla rete

L'unica differenza tra il controllo a livello di circuito o di stream è la dimensione delle finestre.

# TOR RELAY

La rete TOR si basa su un insieme di nodi gestiti da volontari chiamati TOR Relay, possono essere di tre tipi:

- ▶ **Non-exit Relay**, i nodi interni della rete che a loro volta si dividono in:
  - ▶ **Guard Relay**, il primo nodo del circuito
  - ▶ **Middle Relay**, i nodi intermedi del circuito
- ▶ **Exit Relay**, i nodi di uscita della rete Tor, instradano il traffico nella rete comune. Essendo gli unici IP visibili all'esterno sono i più esposti a rischi legali.
- ▶ **Bridge Relay**, nodi non pubblici che non possono quindi essere bloccati

Tutte le informazioni sui Relay esistenti sono visitabili al seguente indirizzo:

<https://metrics.torproject.org/rs.html>

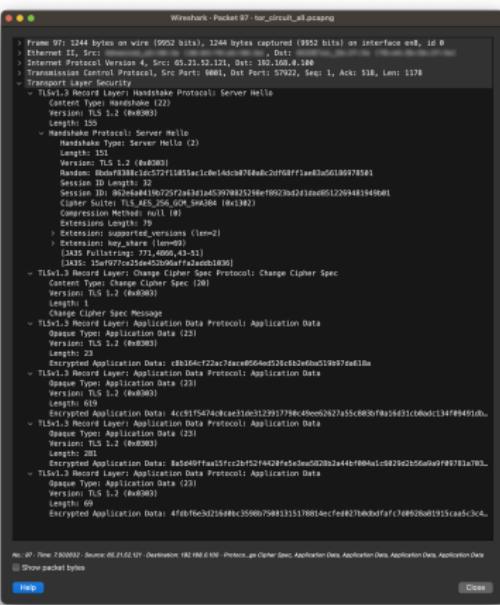
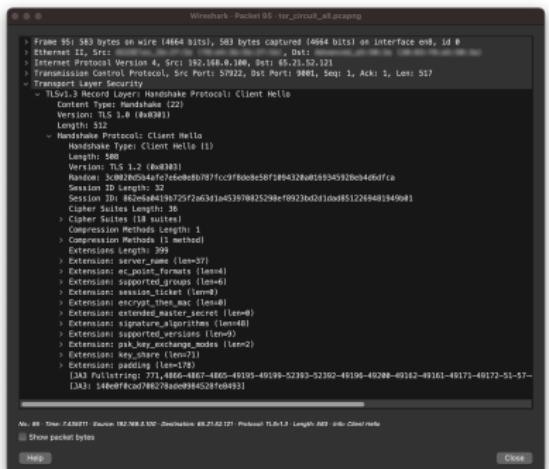
# DIRECTORY SERVERS

I Directory Server sono un sottogruppo di onion router che tracciano i cambiamenti nella **topologia di rete** e agiscono da **DNS server** per i servizi onion. Mantengono infatti i **descriptor**, pacchetti generati dai servizi onion criptati con la propria chiave privata, che contengono gli introduction points e la chiave pubblica.

Il sistema di generazione di indirizzi onion fornisce un **meccanismo di sicurezza** per evitare che un malintenzionato possa alterare i descriptor e reindirizzare gli utenti ai propri introduction points, infatti se così fosse la chiave pubblica nascosta nell'indirizzo onion non sarebbe in grado di decifrare il descriptor.

# DIMOSTRAZIONE WIRESHARK

Wireshark ci permette di vedere come viene generata la connessione **TLS**, in cui il client invia alcune informazioni tra cui la versione TLS, la lista dei Cipher Suite supportati e la chiave pubblica. Il proxy riceve poi il Server Hello, ovvero la risposta del server con le relative informazioni di criptografia. Successivamente Tor usa le informazioni ottenute per stabilire una connessione TLS tra client e server, impedendoci di vedere il traffico dati.



No: 88 - Type: 7/48/211 - Source: 182.168.0.102 - Destination: 69.21.52.121 - Protocol: TLSv1.3 - Length: 1178 - Info: Client Hello

Show packet bytes

Help

Close

# IMPLEMENTAZIONE

Per implementare un servizio onion è necessario utilizzare il **Proxy Onion** per inoltrare le richieste dalla rete Tor al server web, il proxy gestisce la generazione delle chiavi, dell'indirizzo e la definizione e connessione con gli **introduction points**, oltre alla generazione del descriptor e la sua pubblicazione nei Directory Servers.

In particolare la nostra implementazione userà **Onion V3** (dato che Onion V2 è stato deprecato) con alcune migliorie tra cui la maggior sicurezza degli indirizzi.

Useremo un server **Linux EC2** di AWS per ospitare il proxy onion e un **server nginx**.

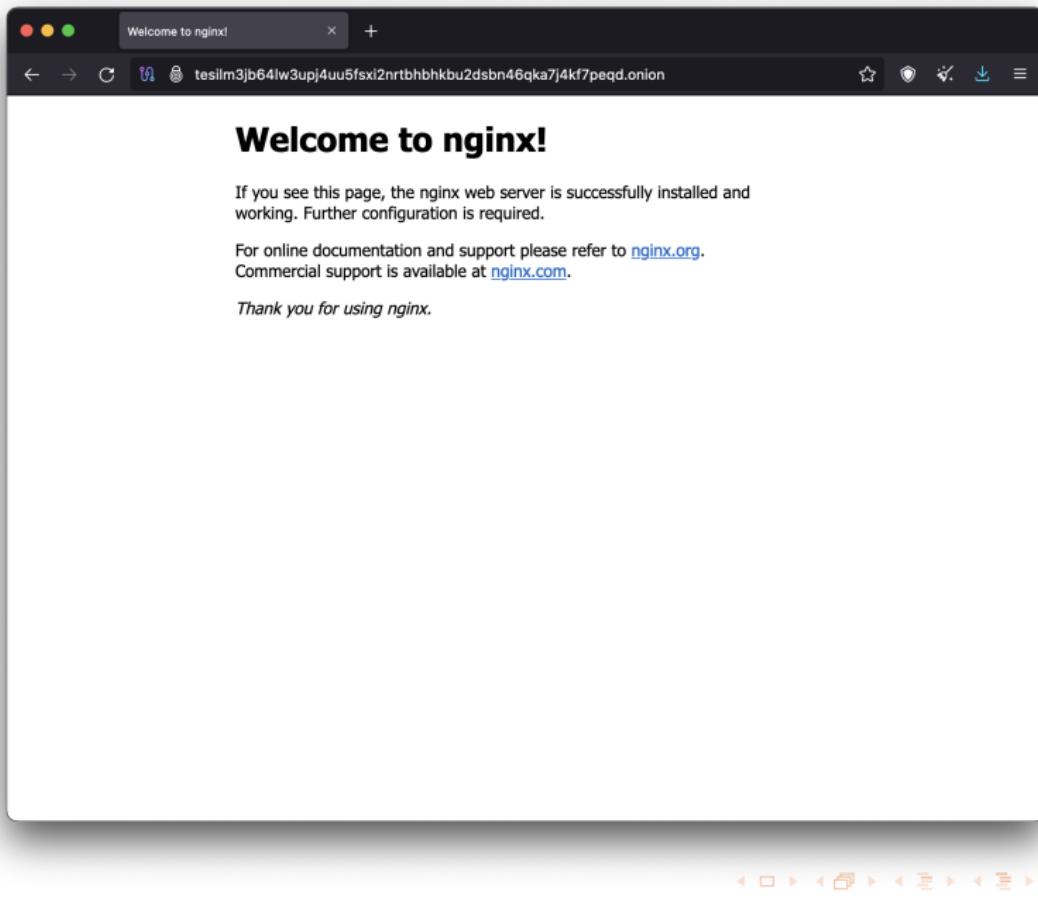
Nel sistema è necessario installare il pacchetto nginx (web server) e dopo una serie di configurazioni dei repository possiamo installare tor.

Configuriamo il file di configurazione torrc con

HiddenServiceDir /var/lib/tor/hidden\_service (per indicare la directory dove verranno salvate le chiavi) e

HiddenServicePort 80 unix:/var/run/website.sock (per indicare il tipo e il modo di connessione con il web server).

In questa implementazione abbiamo usato un tool (mkp224o) per generare la coppia chiave privata e pubblica che risulti in un indirizzo onion personalizzato.



# PUBBLICIZZARE IL SERVIZIO

Nel lavoro di tesi abbiamo mostrato 2 principali sistemi per pubblicizzare il servizio:

- ▶ Pubblicizzazione su altri siti o motori di ricerca, come **notEvil** o **OnionDir**
- ▶ **Header Tag**, è possibile inserire un tag all'interno di un sito web che quando aperto con Tor consente di reindirizzare l'utente al relativo sito onion, in maniera automatica o tramite il tasto in alto a sinistra.

```
<meta http-equiv="onion-location" content="http://  
tesilm3jb64lw3upj4uu5fsxi2nrtbhbhkbu2dsbn46qka7j4  
kf7peqd.onion" />
```

The screenshot shows a web browser with the title "Onion Thesis". The address bar displays "<https://test.miglio.dev>". A purple status bar at the top right indicates ".onion available". The main content area lists four items:

- Tazza Unicam ~ 0.49 mBTC**: An image of a red ceramic mug with the "UNICAM" logo and website "www.unicam.it".
- Chiavetta USB ~ 0.49 mBTC**: An image of a white USB flash drive with the "UNICAM" logo and website "www.unicam.it".
- Chiavetta USB rossa ~ 0.49 mBTC**: An image of a red USB flash drive with the "UNICAM" logo and website "www.unicam.it".
- Borraccia Unicam rossa ~ 0.283 mBTC**: An image of a red plastic water bottle with the "UNICAM" logo.