Privacy methods.

Simulazione di Sistemi

September 13, 2015

Davide Berardi 0000712698

Matteo Martelli 0000702472

Università di Bologna.

Privacy

Privacy is the right to publish only some informations that we want to publish.
There are a lot of laws and legal issues related to privacy (but some people are just not intrested in laws).

Anonymity

We will talk about anonimy as the propriety of disconnect the user of a service from some basic proprieties:

- Geolocation.
- Association to a face or a name (or to an IP address).

Sometimes we need to reassociate the user with a communication channel or so.

But...who cares?

”I have nothing to hide, who cares about my personal data?”

Surveillance

NSA projects

Other stories

Obscuration

Industrial espionage

Not only powerful adversaries

Technologies

What we can do versus controls?
Can we have some privacy even from the companies/government?

Confidentiality and authenticity

We have a lot of programs to protect our data

- PGP
- IPsec
- OTR-based programs
- Protonmail ProtonMail
- TrueCrypt
- **Perfect Forward Secrecy**

Some tool for steganography can help but not too much.

Anonymity

But for anonymity?

- Anonymous networks
- Mix Max networks.
- Anonymous remailers.
- Proxy chains.
- **Onion Routers**

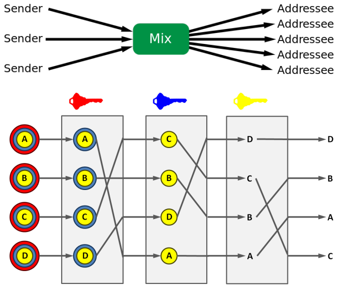
Anonymous network

(mostly) p2p-based networks, no one can identify who put a file on the net.

- FreeNet
  - OpenNet mode
  - DarkNet mode
- GNUNet
- Fully Self Contained (like UseNet).
- Search?
- Performance?

Mix Networks

- Model of the 1981.
- Multiple layers of encryption.
- Select different random nodes to deal with controlled nodes.
- **Timing attack?**



Old times

- Anonymous remailers: end to end anonymity.
  - Cypherpunk: remove FROM field and encrypt the mail
  - Mixmaster: Chain of remailers.
  - Mixminion: Mixmaster syntax with replies.
  - nym-server: give a pseudonym to the user detached from his IP.

We'll see that this servers recalls the modern idea of OnionRouting.

Onion Routing

The idea of encapsulate cyphered packets in a chain or an "onion".

- OpenNet? → Hidden services.
- New possibility like use a proxy to get to the normal internet.

Problems

- Performance
- DoS resistance.
- Mantain links to the users
- Thrustness of the routers.
- Confidentiality and autenticity.

Onion Routing (2)

- TOR Tor
  - We'll come to this later.
- i2p I2P
  - Done for eepsite(s).
  - Not so much routers/outproxies.

- And what for the low latency? → **Timing attacks**.

Why simulation?

Simulation help us in a lot of aspects:

- Compare the performances of two onion routers (p.e. i2p vs Tor).
- To compare effects of changes in the node choice algorithms.
- **Get an idea of the number of resources needed by an attacker and to mantain anonimity**.

NSA and Tor

Tor was made from the naval research labs:

- Made for the anonymous control and espionage.
- Tor need a number of exit nodes (and routers) to lead anonymity to an user.
- if an organization use only his exit nodes it's like to not use them at all.

The Tor revolution

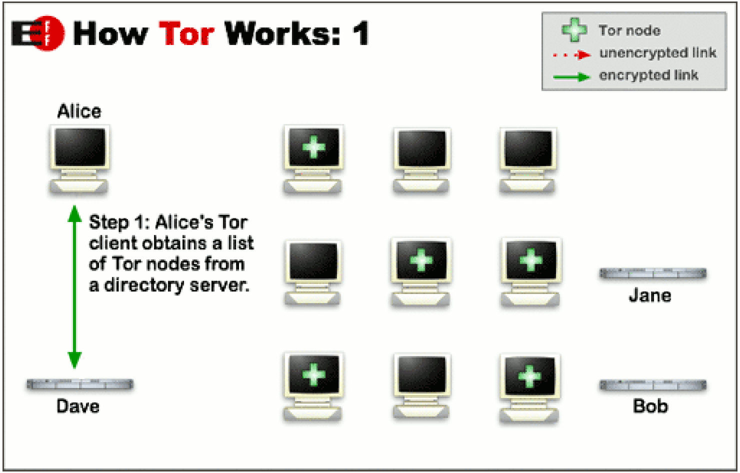
Tor slipped from the hands of the US when it was released, open sourced and the community gained control over the half of the net.

- Russia offered $114.000 to identify and deface Tor anonimity.
- NSA now classify TOR as a menace of level *catastrophic*.

Part 2 – Onion routers and attacks

Tor

Tor workings

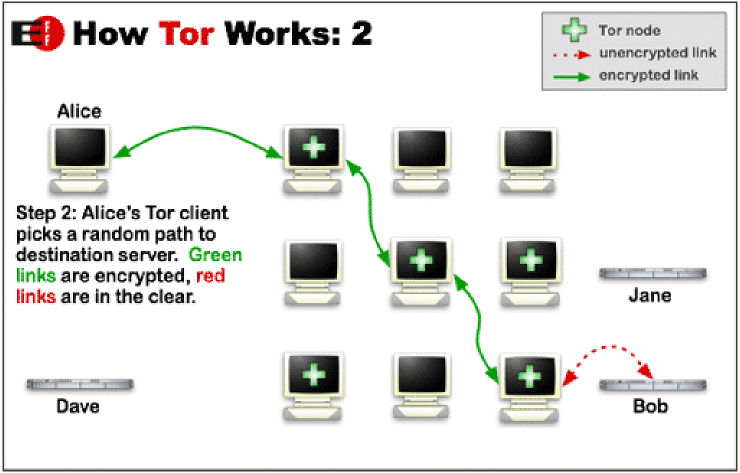


How Tor Works: 1

Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

Tor

Tor workings

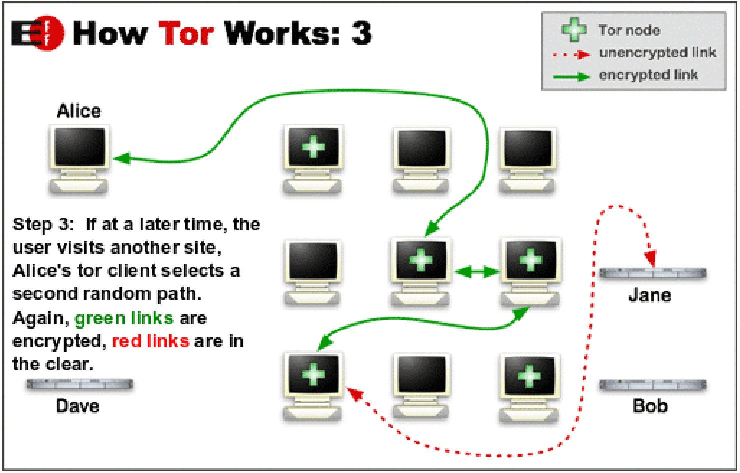


How Tor Works: 2

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Tor

Tor workings

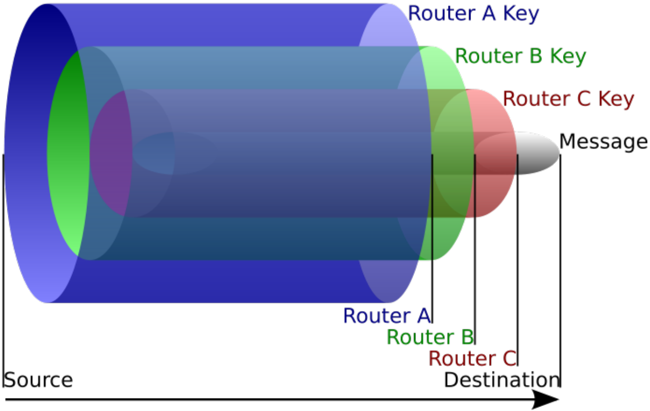


How Tor Works: 3

Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

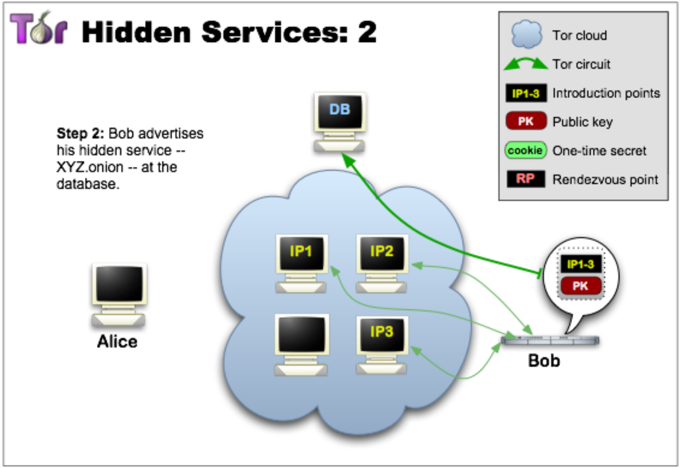
Tor

Tor encryption



Tor

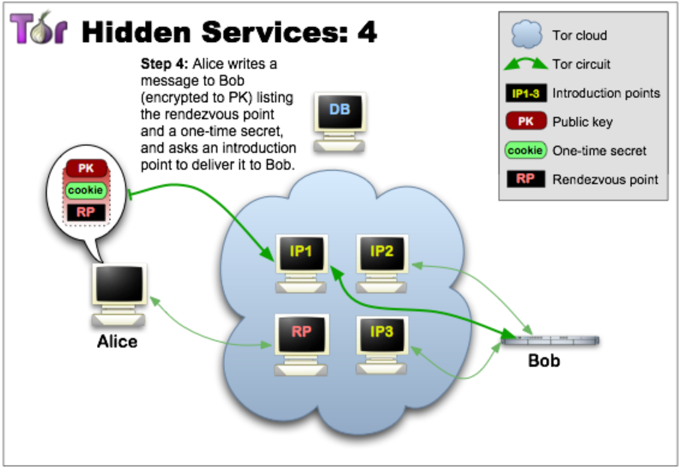
Tor Hidden Service


Tor Hidden Services: 2

Step 2: Bob advertises his hidden service -- XYZ.onion -- at the database.

Tor

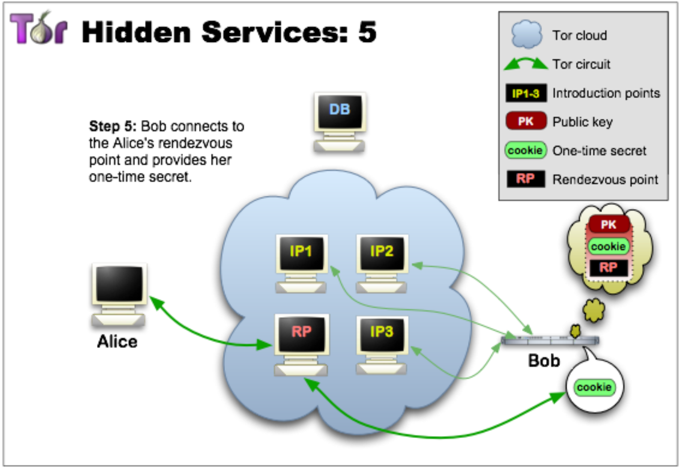
Tor Hidden Service



Step 4: Alice writes a message to Bob (encrypted to PK) listing the rendezvous point and a one-time secret, and asks an introduction point to deliver it to Bob.

Tor

Tor Hidden Service

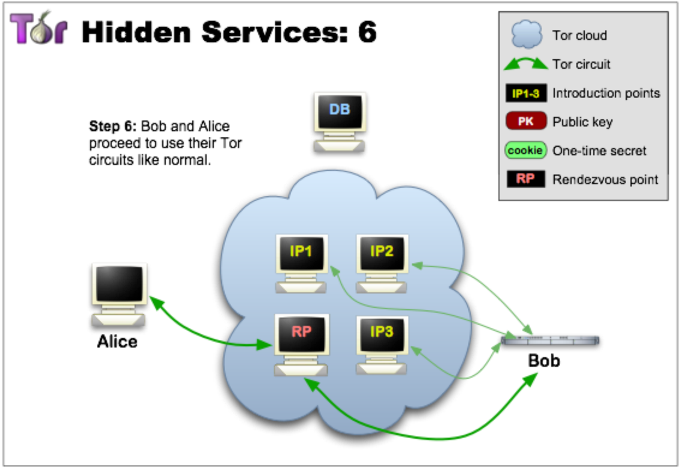


Hidden Services: 5

Step 5: Bob connects to the Alice's rendezvous point and provides her one-time secret.

Tor

Tor Hidden Service



Step 6: Bob and Alice proceed to use their Tor circuits like normal.

Tor
Summary

- Base: anonimity of clients
- Hidden services: anonimity of client + anonimity of servers

But ... is it enough?"

Time analysis based attacks

> *"Tor does not provide protection against end-to-end timing attacks[...]"*

We can place a tracker after the client node and another before the server node and check for the connection time to profile users and nodes (and later associate IP to users.)

Thank you for your attention.

Diffie-Hellman

- Alice pick a random number a, a prime number p and α as a primitive root of p.
- Alice calculate $k_a = \alpha^a modp$ and sends $< k_a, p, \alpha >$ over the channel.
- Bob read the packet, pick a random number b, calculate $k_b = \alpha^b modp$ and sends it to Alice.

Now the shared key $K = k_b^a = k_a^b = \alpha^{ab} modp$ is known to Alice and Bob[1].

[1] For the little Fermat theorem ($a^p \equiv a\ modp$) if p is a prime

Perfect Forward Secrecy

- If a key is derived from another with a deterministic method then a leak of the second key can reveal every eavesdropped transmission encrypthed with the first key.
- The immunity to this kind of attacks is called *Perfect Forward Security*.
- Used in Diffie Hellman based TLS, OTR, etc.

Freenet

- A file is encrypthed with his hash and shared over the network.
- It can optionally encrypthed with the public keys of the dark net users (pseudonym-like) (and signed).
- The file is so splitted in chunks and shared over the network.
- Every node can't understand which chunk of which file is processing and what is written in the file itself.

Freenet (2)

- The net is garanted to resemble a small world network (so the max degree is $log(n)$) using the Metropolis Hasting algorithm.
- If a file is not popular and only a chunk is lost, the file is lost forever.