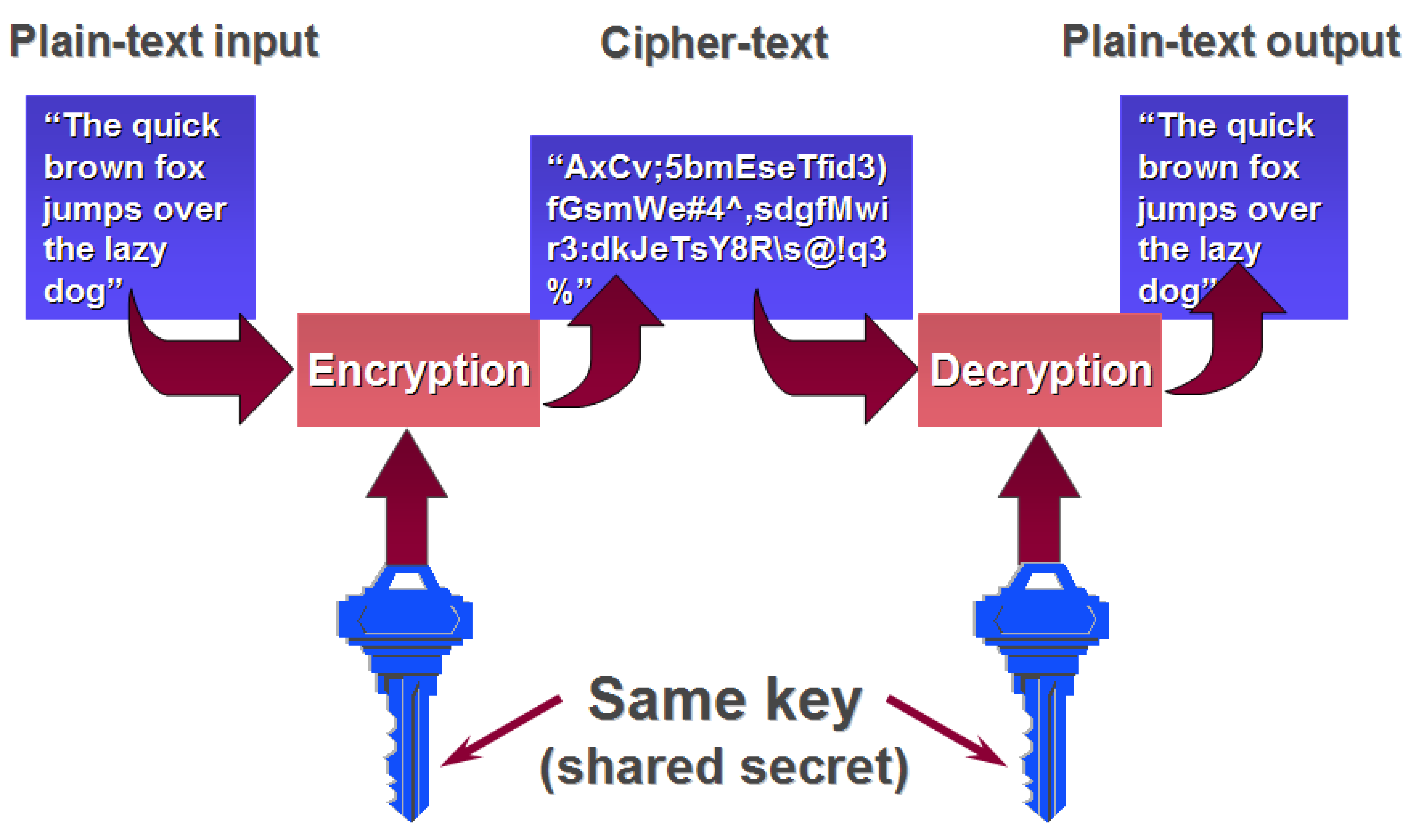
Onion network architecture

http://www.ataeyan.com

By: Mahdi ataeyan

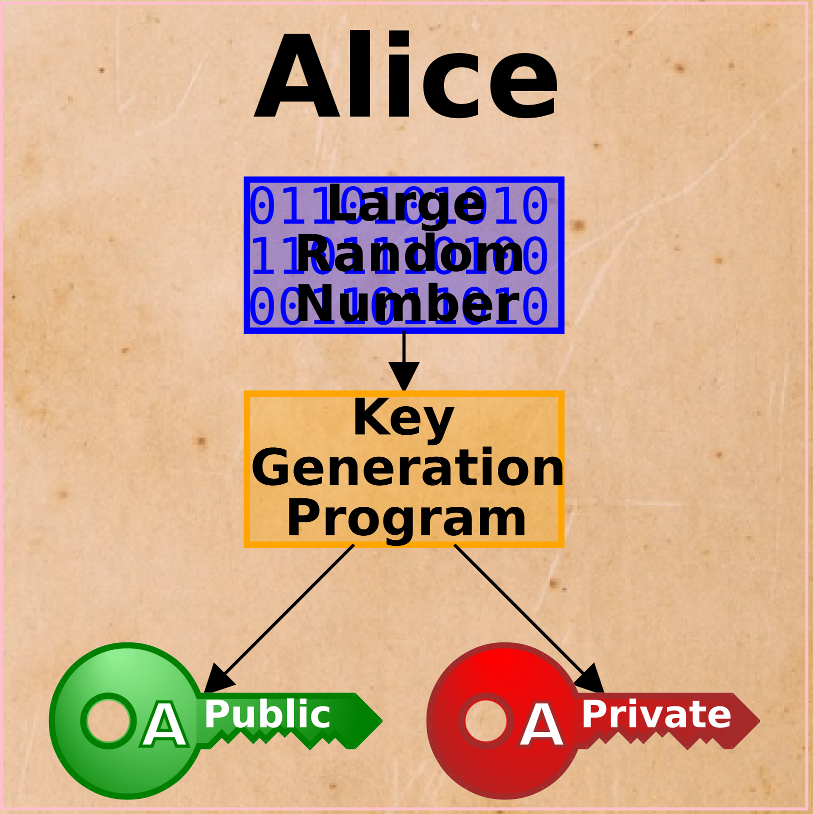
Privacy?!

Symmetric-key algorithm


Plain-text input
Cipher-text
Plain-text output
"The quick brown fox jumps over the lazy dog"
"AxCv;5bmEseTfid3)fGsmWe#4^,sdgfMwir3:dkJeTsY8R\s@!q3%"
"The quick brown fox jumps over the lazy dog"
Encryption
Decryption
Same key
(shared secret)

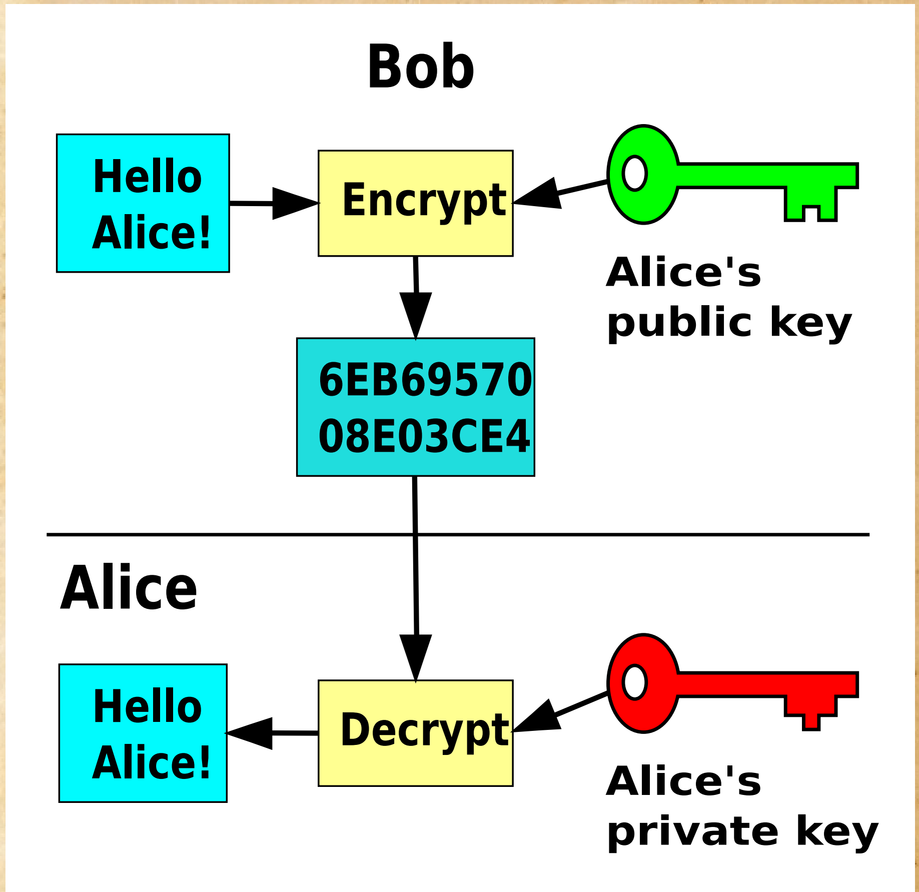
Public key crypto

- An unpredictable (typically large and random) number is used to begin generation of an acceptable pair of keys suitable for use by an asymmetric key algorithm.



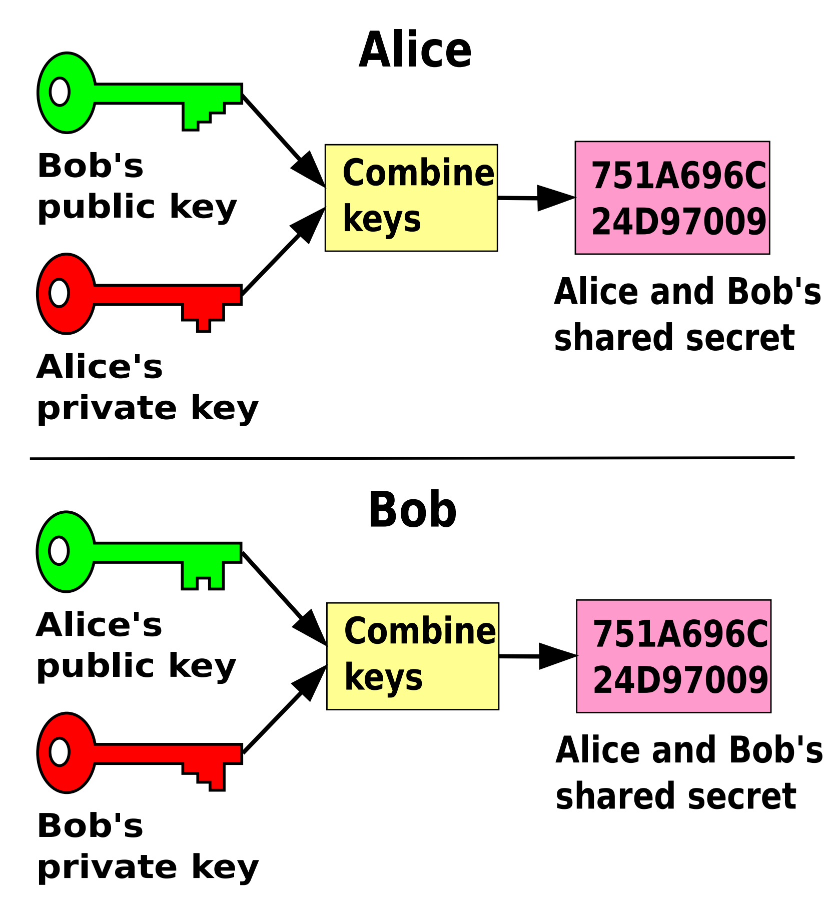
Public key encryption

- In an asymmetric key encryption scheme, anyone can encrypt messages using the public key, but only the holder of the paired private key can decrypt. Security depends on the secrecy of the private key.



Public key shared secret

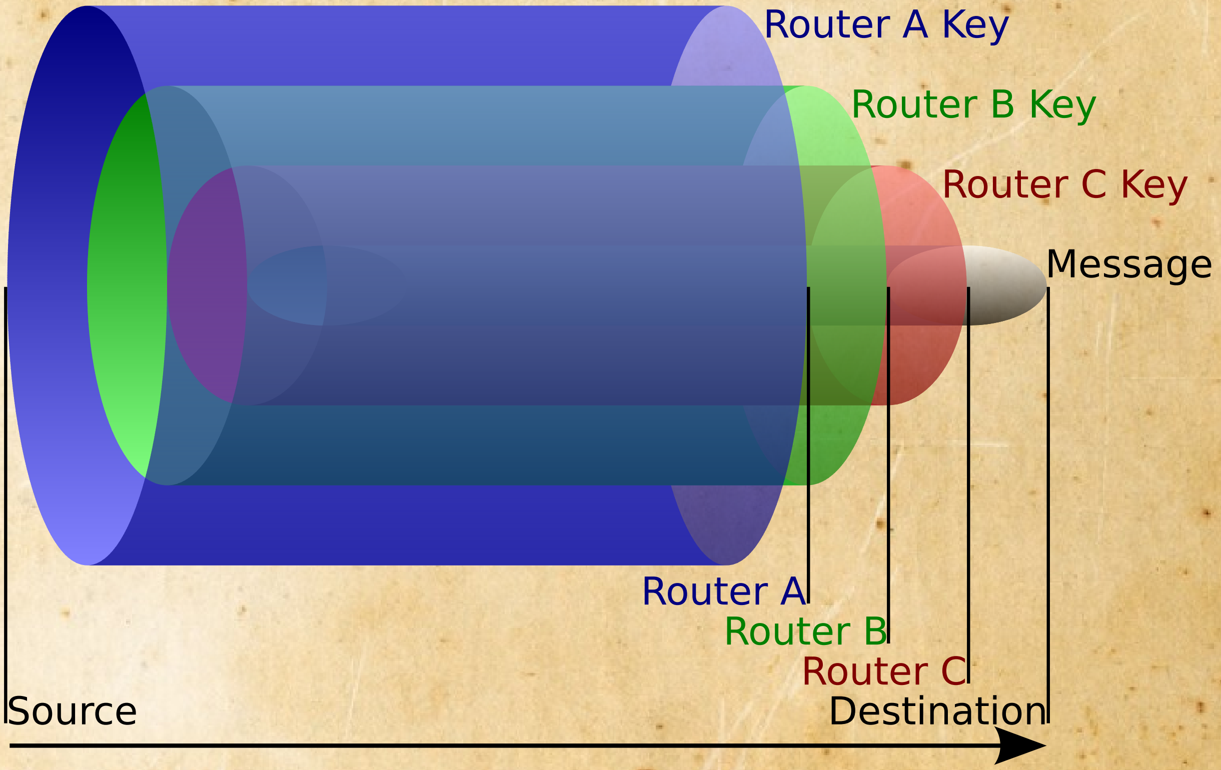
- In the Diffie–Hellman key exchange scheme, each party generates a public/private key pair and distributes the public key. After obtaining an authentic copy of each other's public keys, Alice and Bob can compute a shared secret offline. The shared secret can be used, for instance, as the key for a symmetric cipher.



what's Onion routing?

- OR is a technique for anonymous communication over a computer network
- peeling an onion.

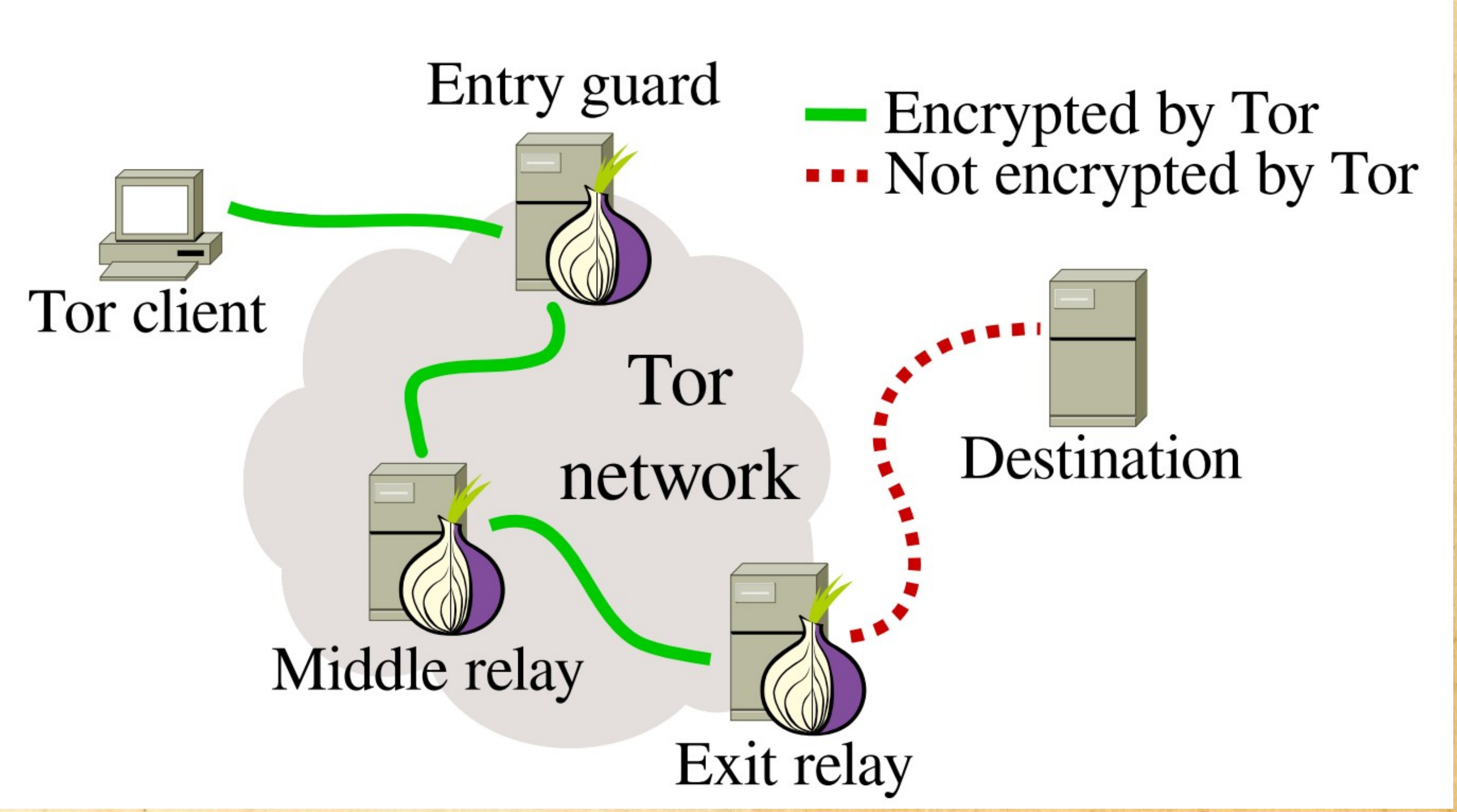
Why onion?



Router A Key
Router B Key
Router C Key
Message
Router A
Router B
Router C
Source
Destination

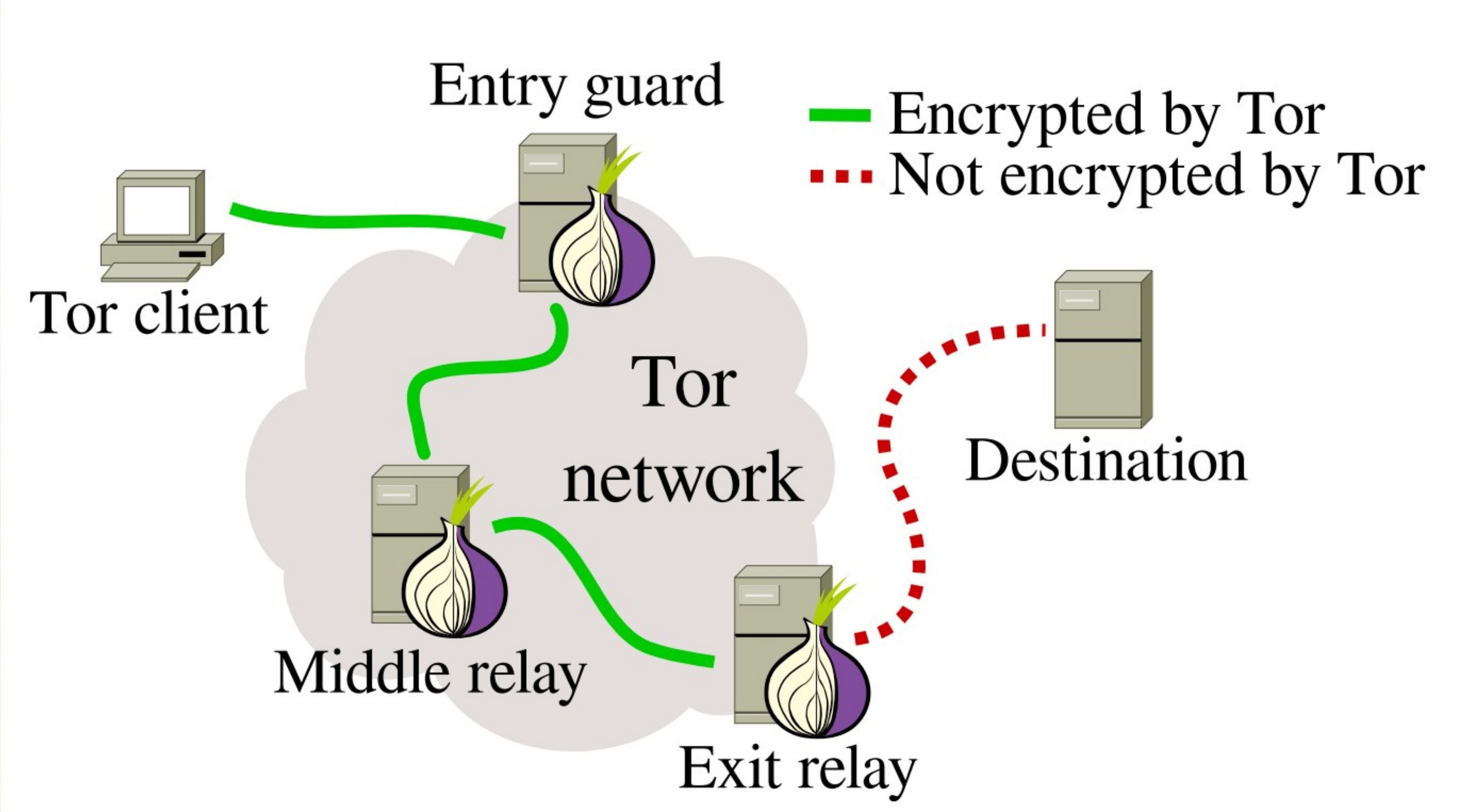
entry node

- First hop into the tor network.



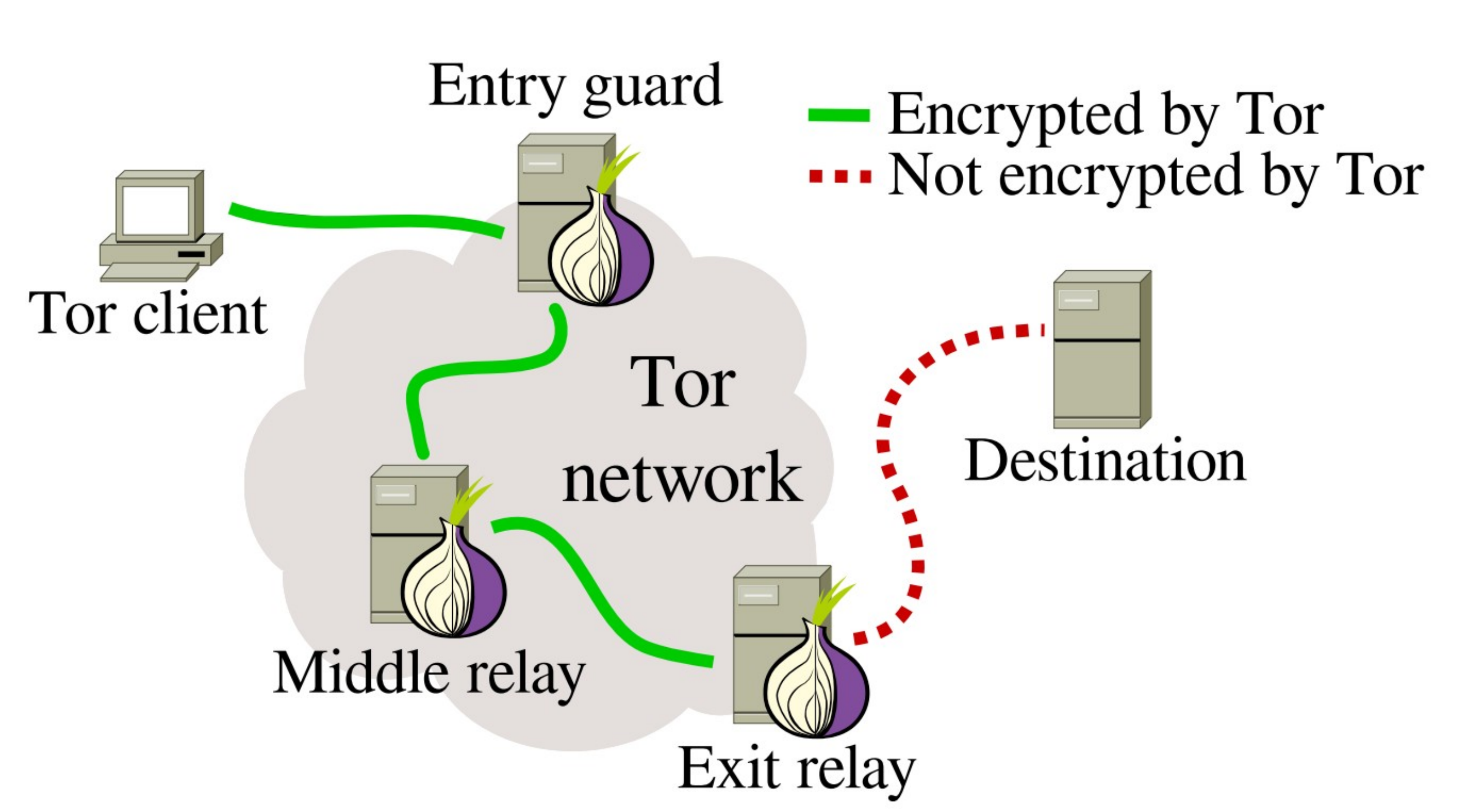
exit node

- last hop before destination.



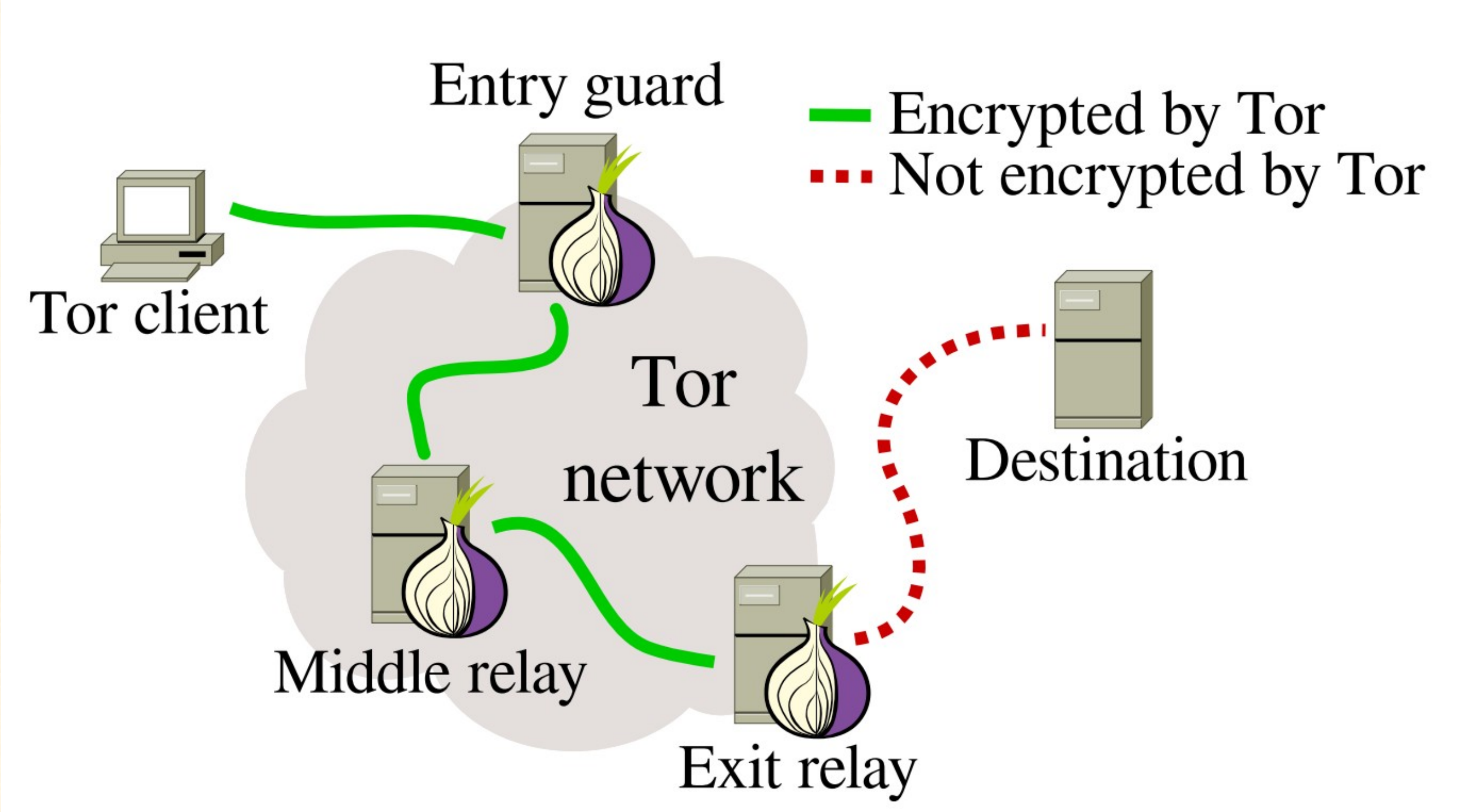
relay node

- Middle node



bridge node

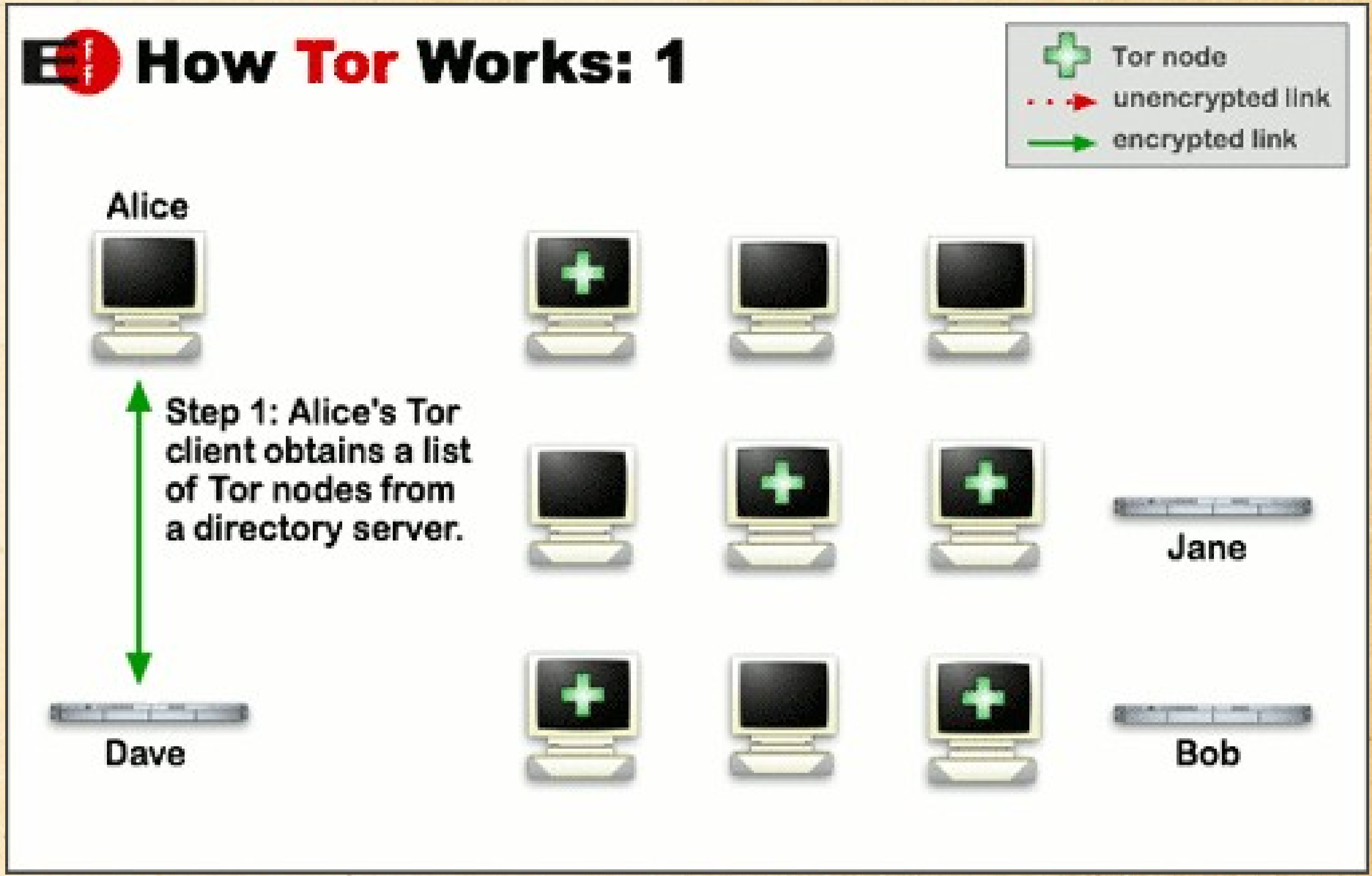
- nodes not listed in the tor directory to evade filtering



Steps

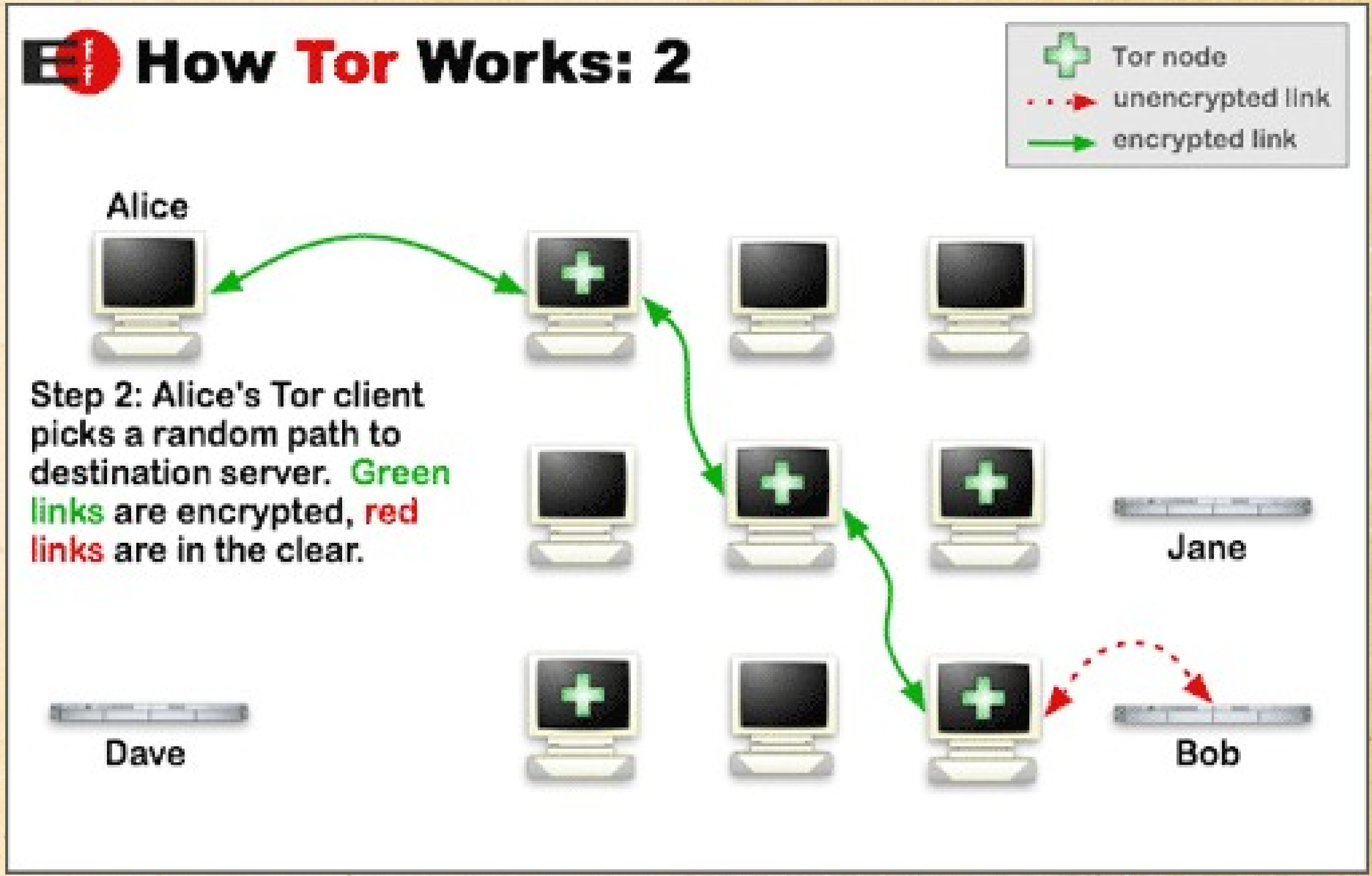
- The originator picks nodes from the directory node and chose some node.
- the chosen nodes are ordered (chain or circuit)
- Originator encript and send data.

picks nodes from the directory node



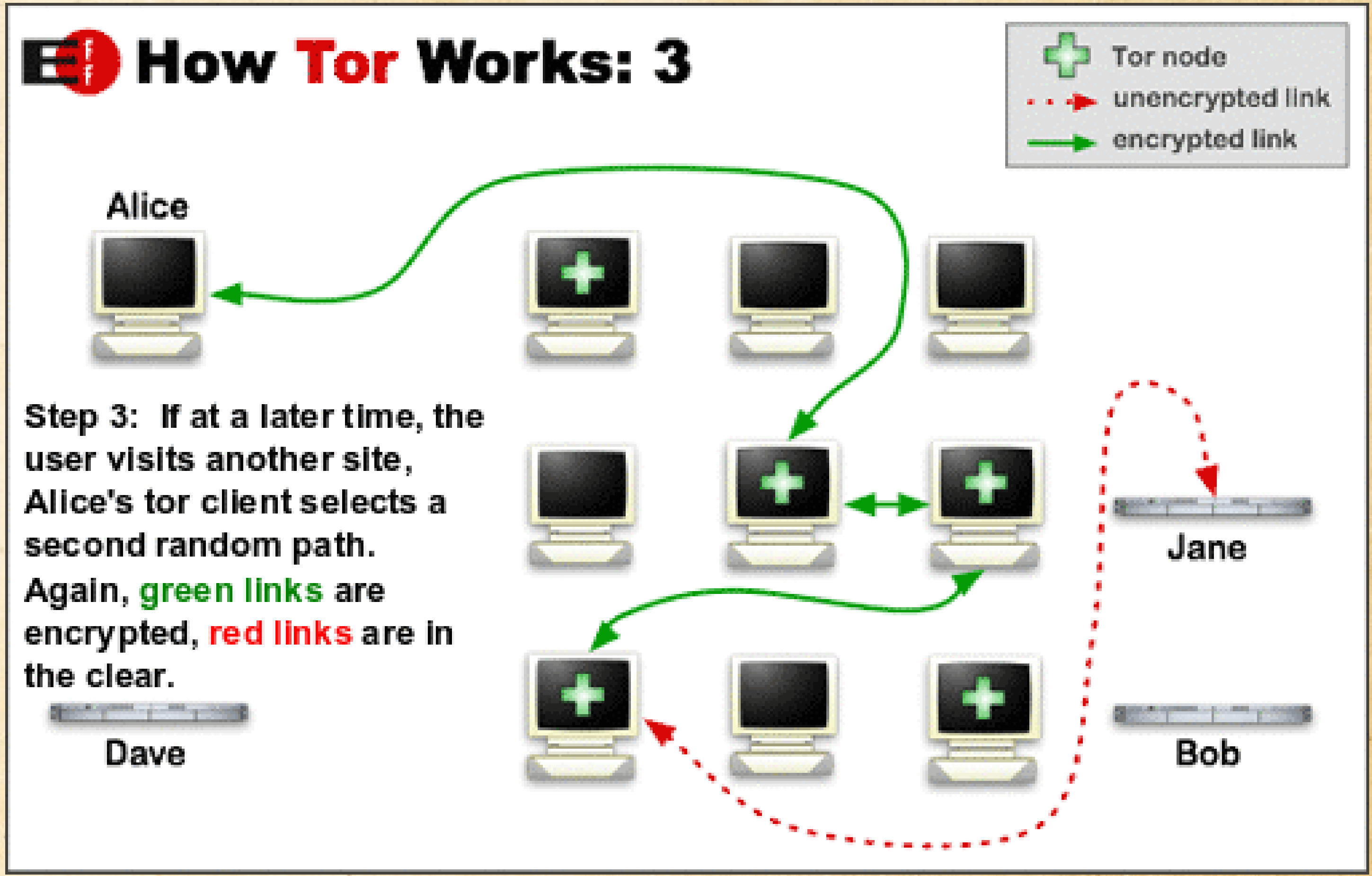
Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

Select node



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

After 10 minute...



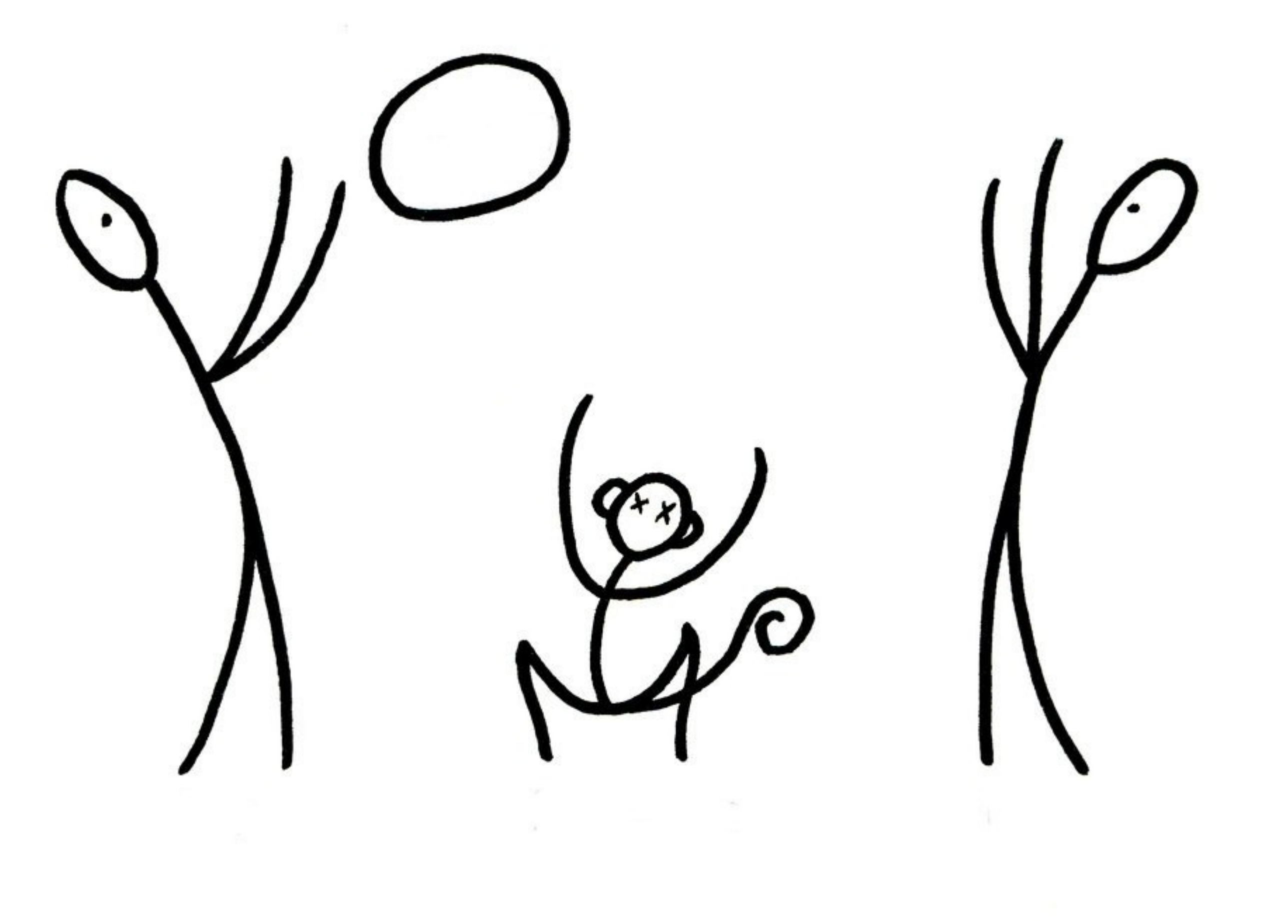
How Tor Works: 3

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.

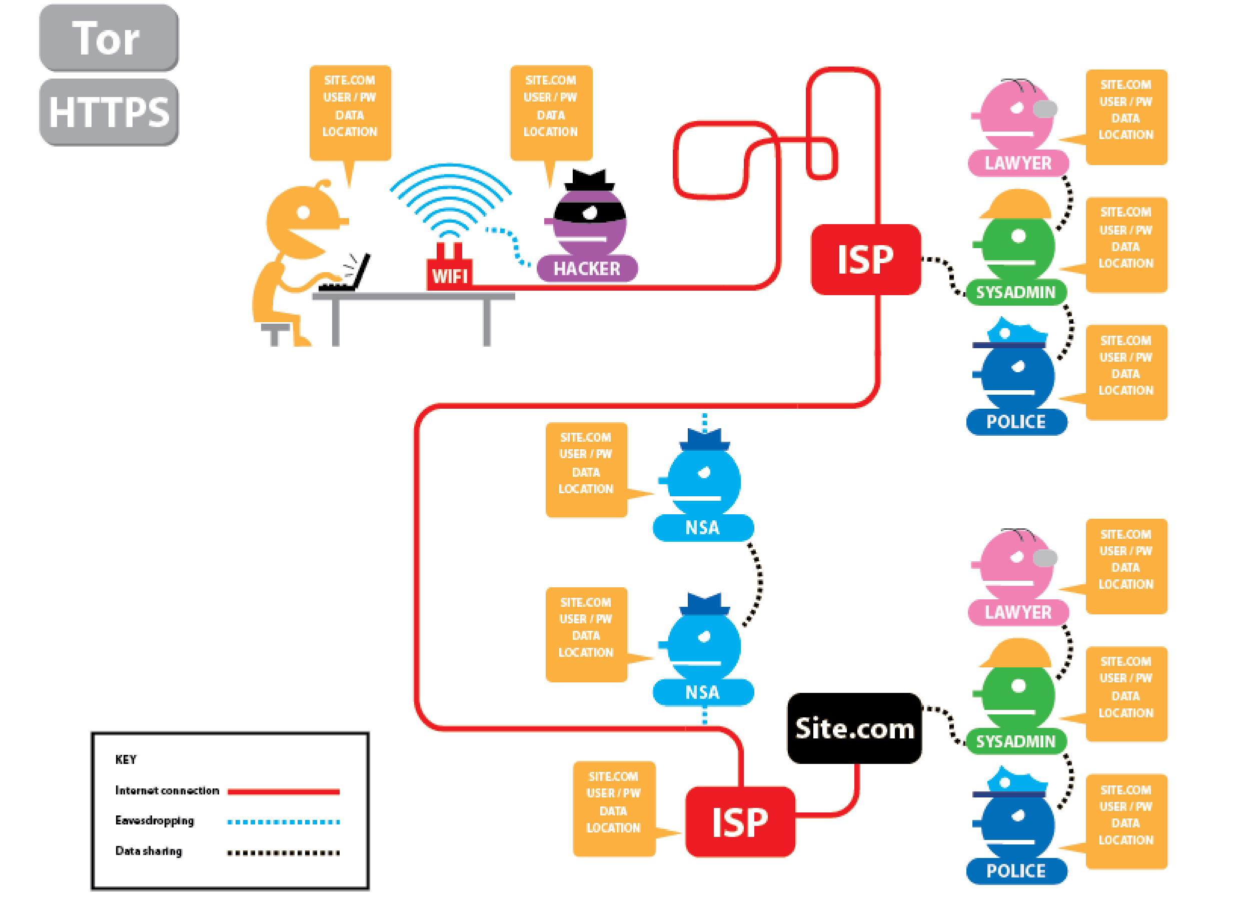
Who can see the message?

- the sender
- the last intermediary (the exit node)
- the recipient

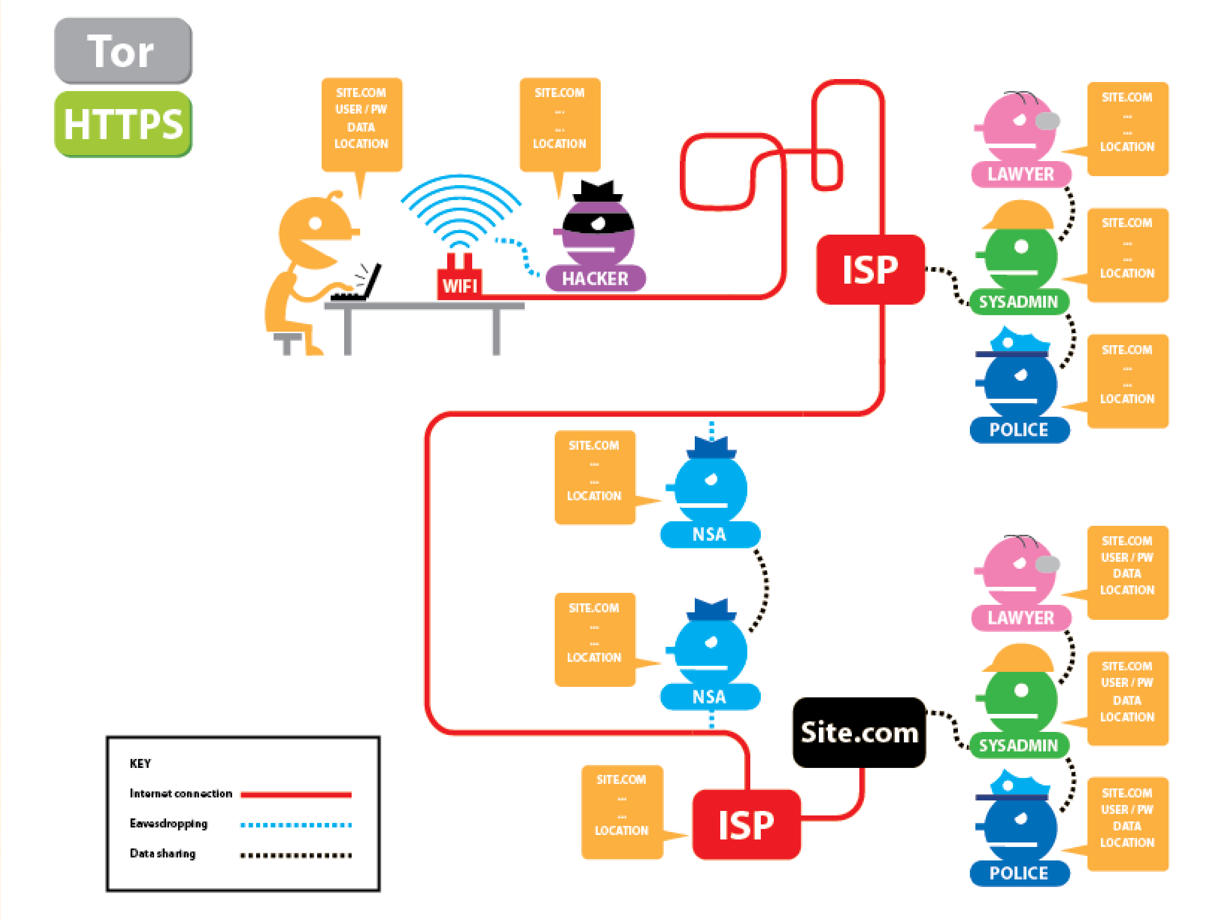
end-to-end encryption



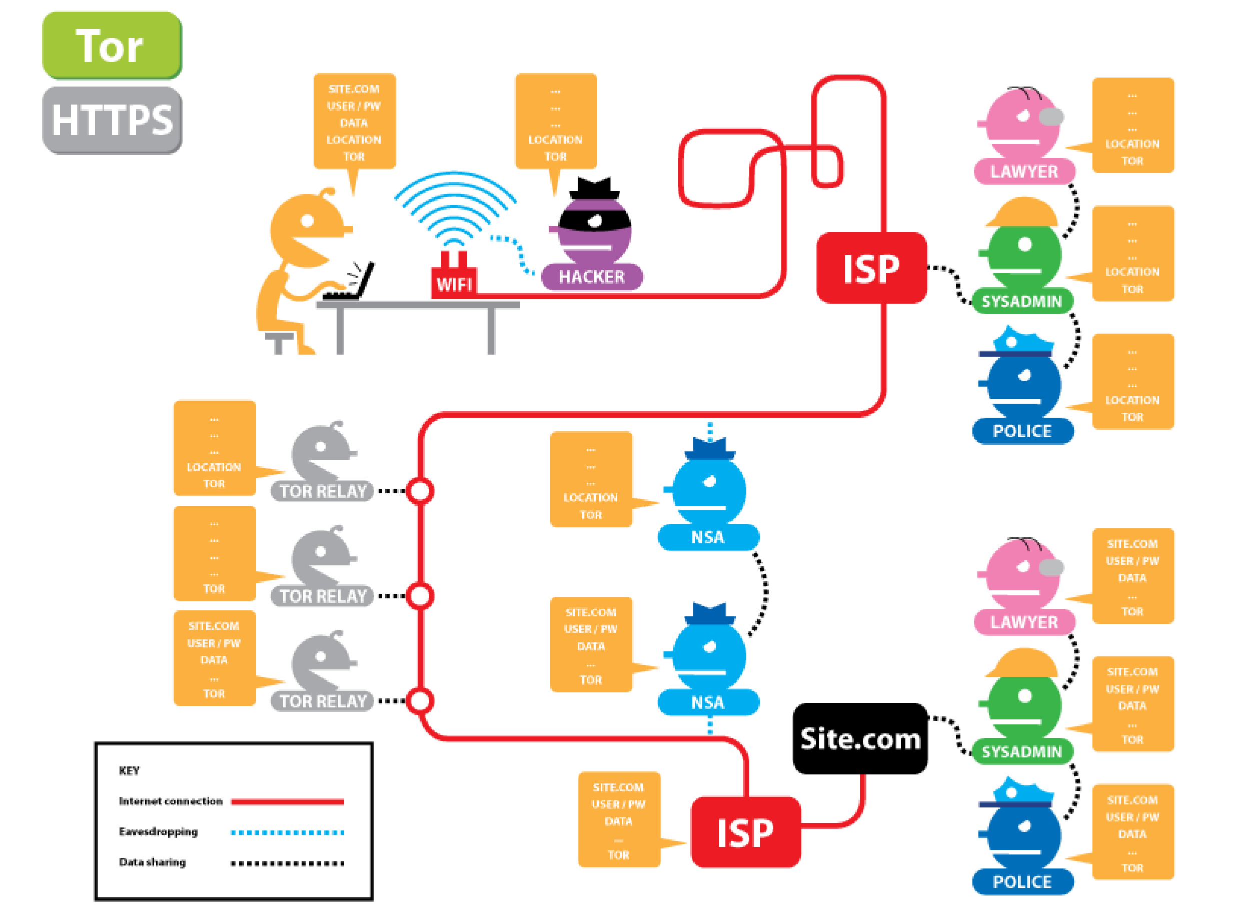
Tor off https off



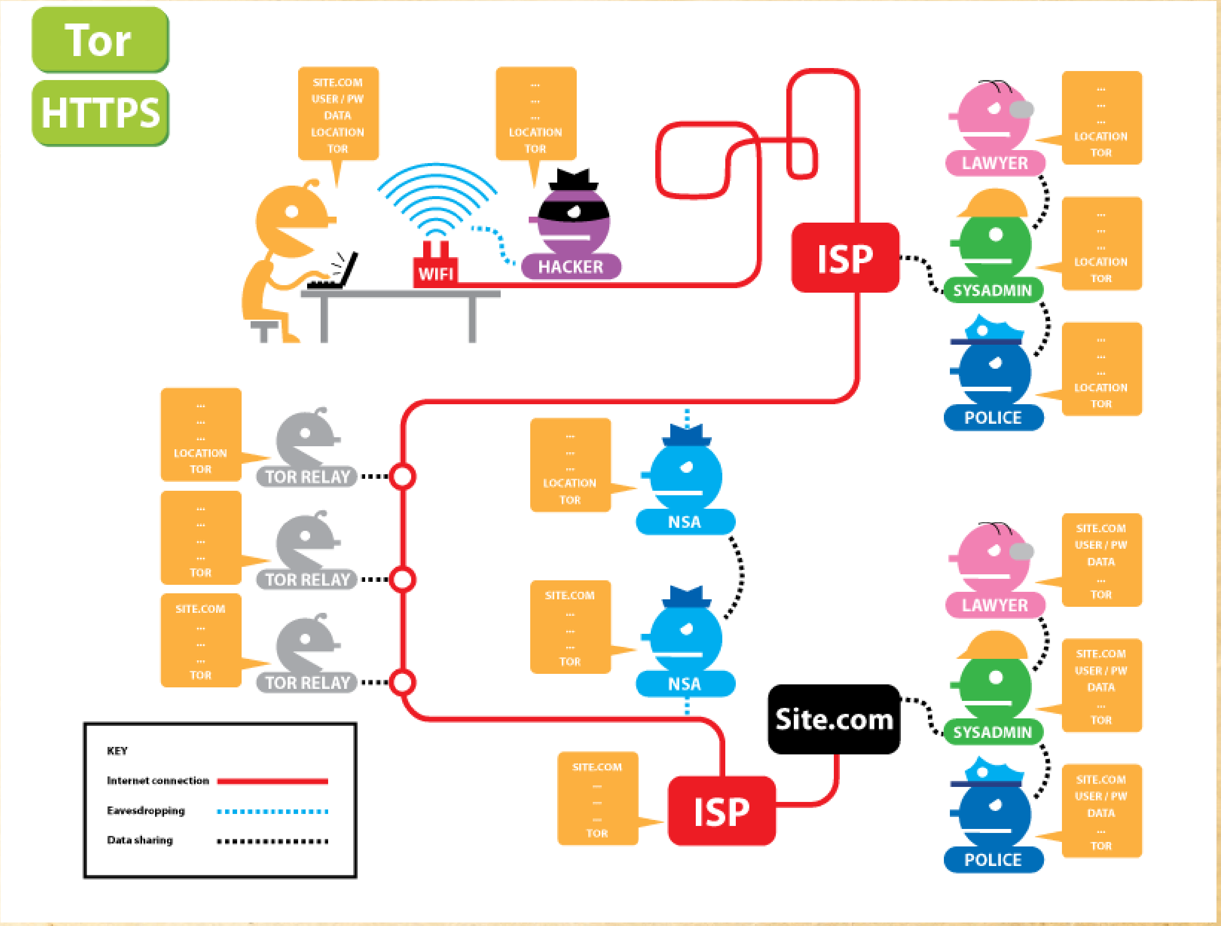
Tor off https on



Tor on https off



Tor on https on



Weaknesses

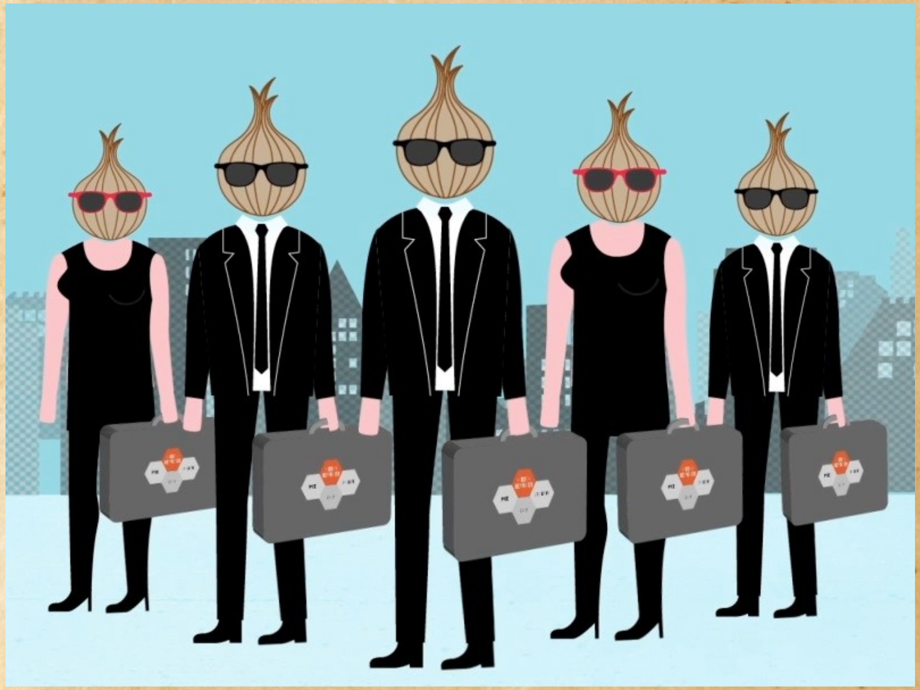
- Timing analysis
- Intersection attacks
- Predecessor attacks
- Exit node sniffing
- Dos nodes
- social engineering attacks

Who's using tor?

- Diplomatic mission
- Militaries
- Normal people
- Journalists
- Activists & Whistleblowers

Hidden service

- anonymity websites and servers.
- accessed through onion address.
- Abcdefghijklmnop.onion



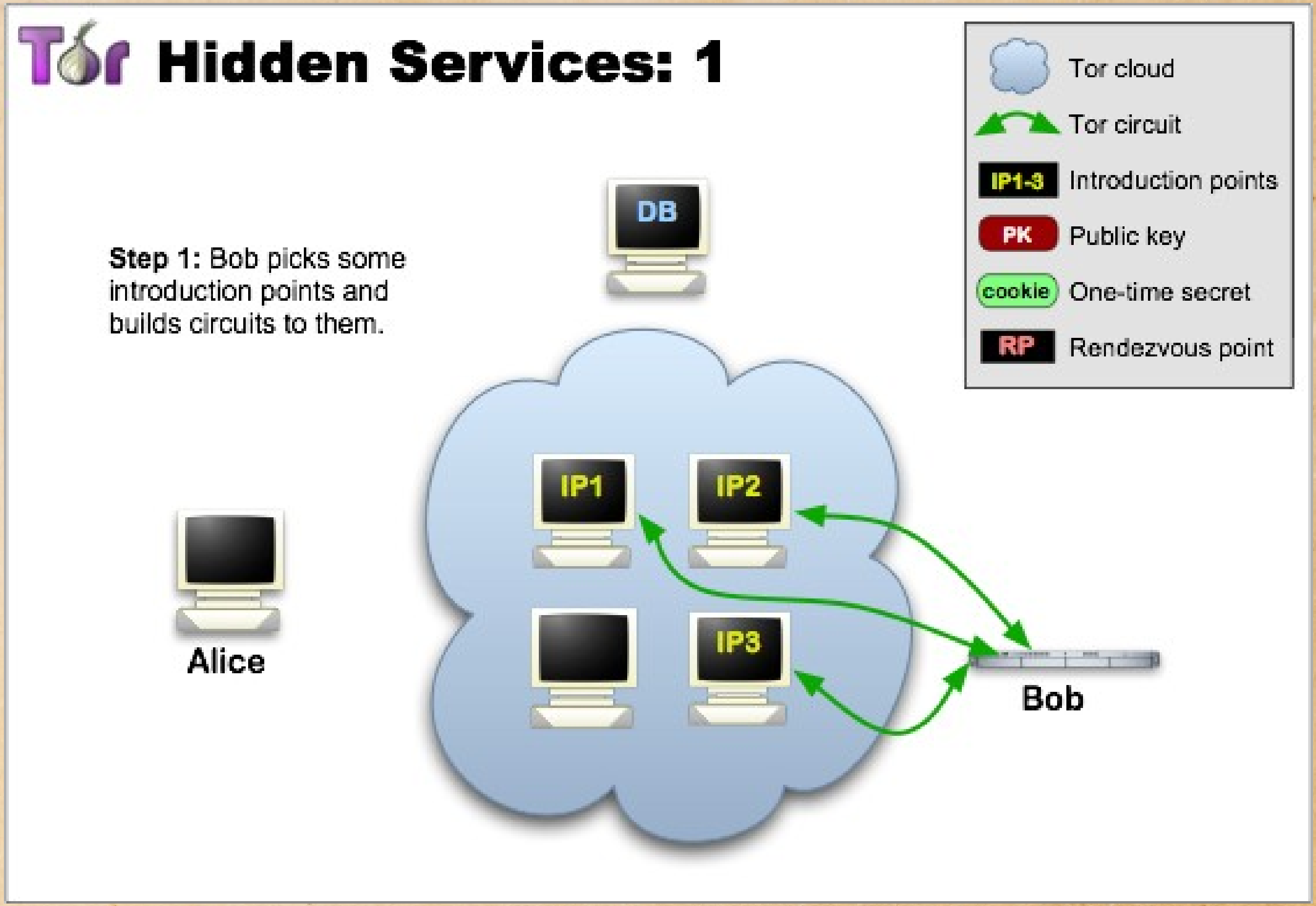
rendezvous protocol

- computer network protocol.
- Enables network node to find each other.
- require at least one unblocked and un-NATed servers.

advertise

- advertise existence
- randomly picks some relays
- asks them to act as „introduction points“
- send public key
- introduction points dont know service location (ip)

introduction points



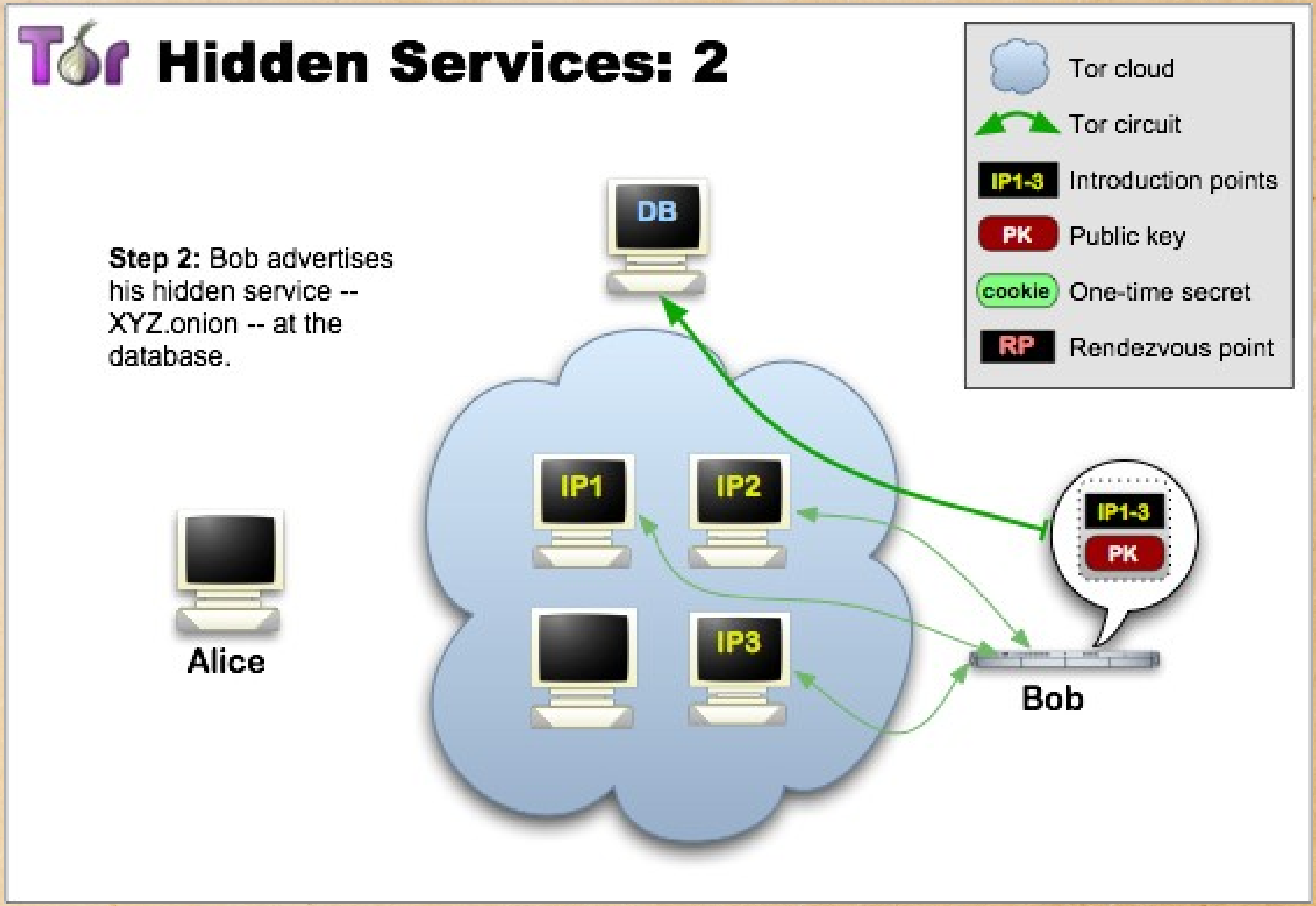
Hidden Services: 1

Step 1: Bob picks some introduction points and builds circuits to them.

hidden service descriptor

- the hidden service assembles a hidden service descriptor
- signs descriptor with private key.
- uploads descriptor to a distributed hash table.
- 16 character name derived from the service's public key.onion

hidden service descriptor



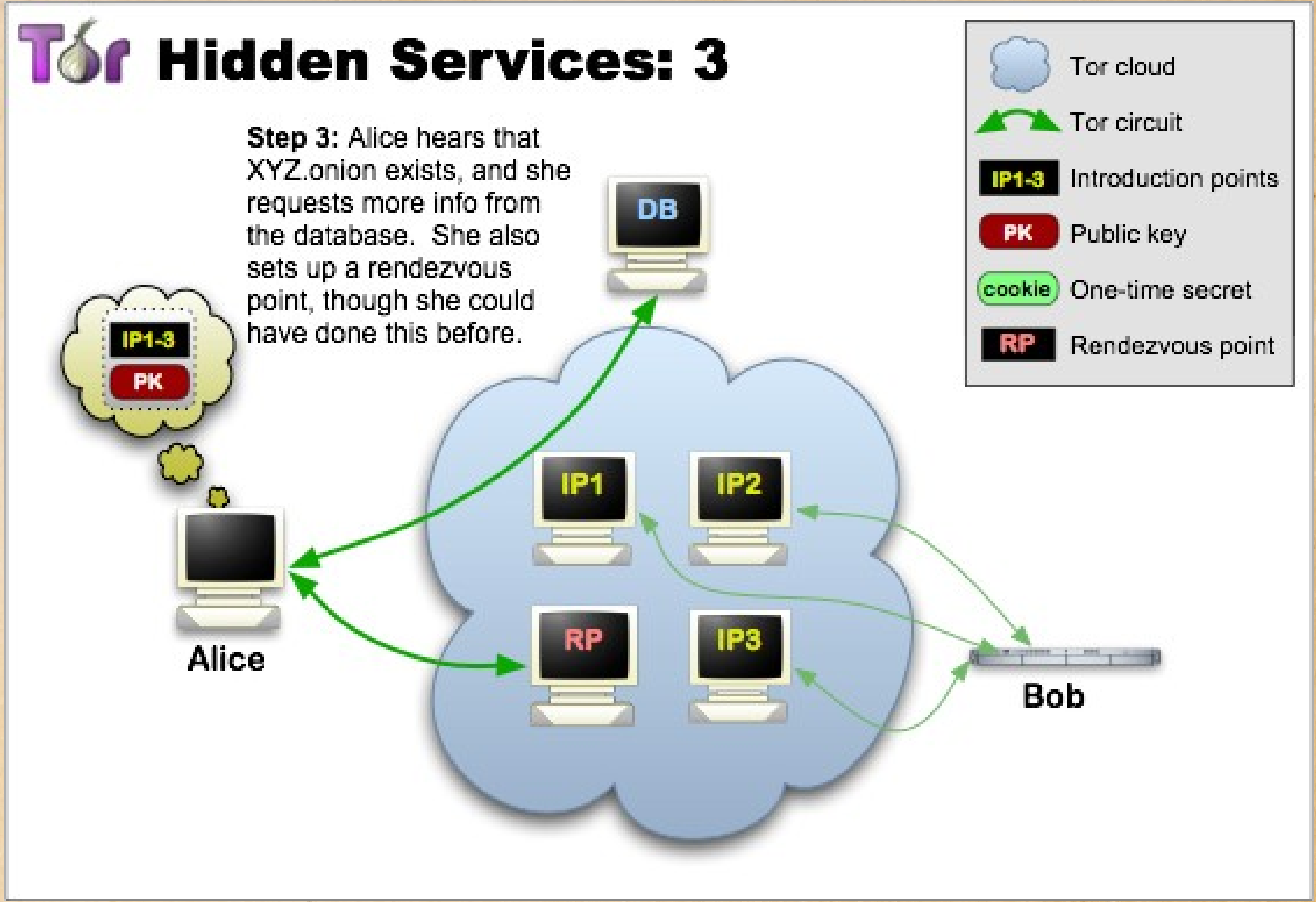
Tor Hidden Services: 2

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

Client rendezvous point

- client needs to know onion address.
- download the descriptor from the distributed hash table.
- the client knows the introduction points and the right public key.
- Client select and connect to rendezvous point and telling it a one-time secret.

Client rendezvous point

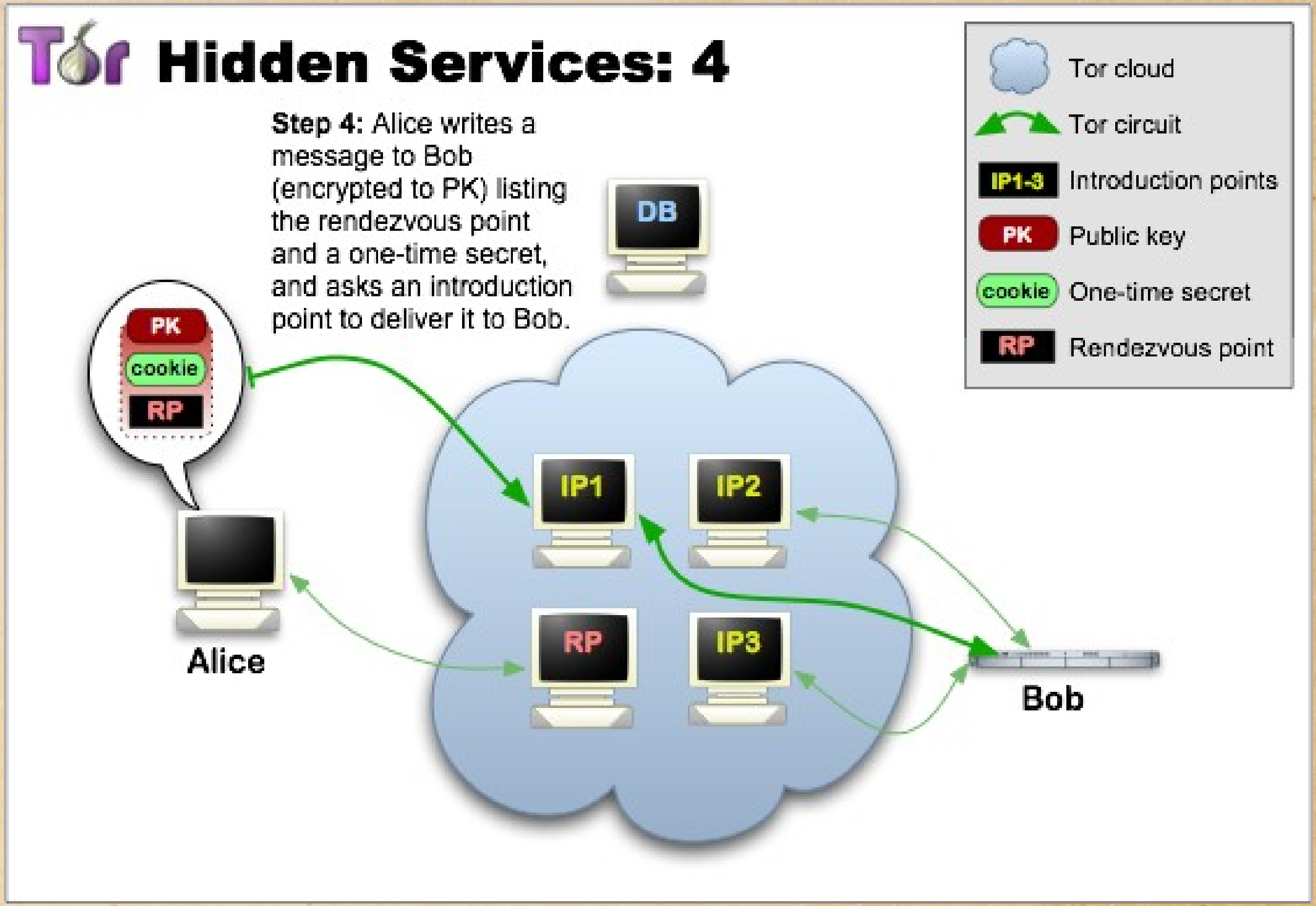


Step 3: Alice hears that XYZ.onion exists, and she requests more info from the database. She also sets up a rendezvous point, though she could have done this before.

client introduce message

- the client assembles an „introduce message“ (encrypted to the hidden service's public key) + address of the rendezvous point and the one-time secret.
- The client sends „introduce message“ to one of the introduction points.
- introduction points delivered to the hidden service.
- the client and service remains anonymous.

client introduce message



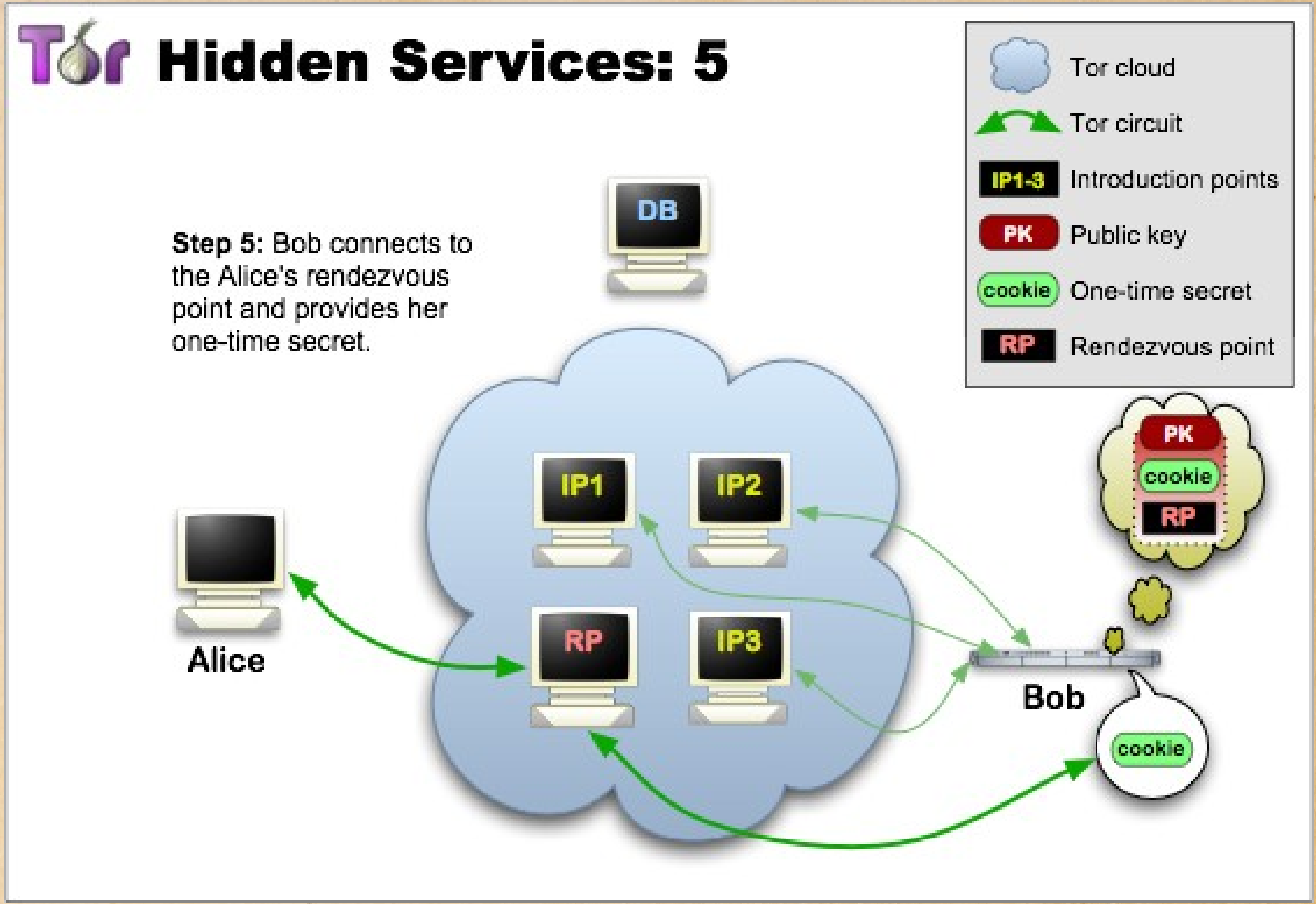
Hidden Services: 4

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.

Hidden Service rendezvous point

- The hidden service decrypts the client's introduce message and finds the address of the rendezvous point and the one-time secret in it.
- The service creates a circuit to the rendezvous point and sends the one-time secret to it in a „rendezvous message“.

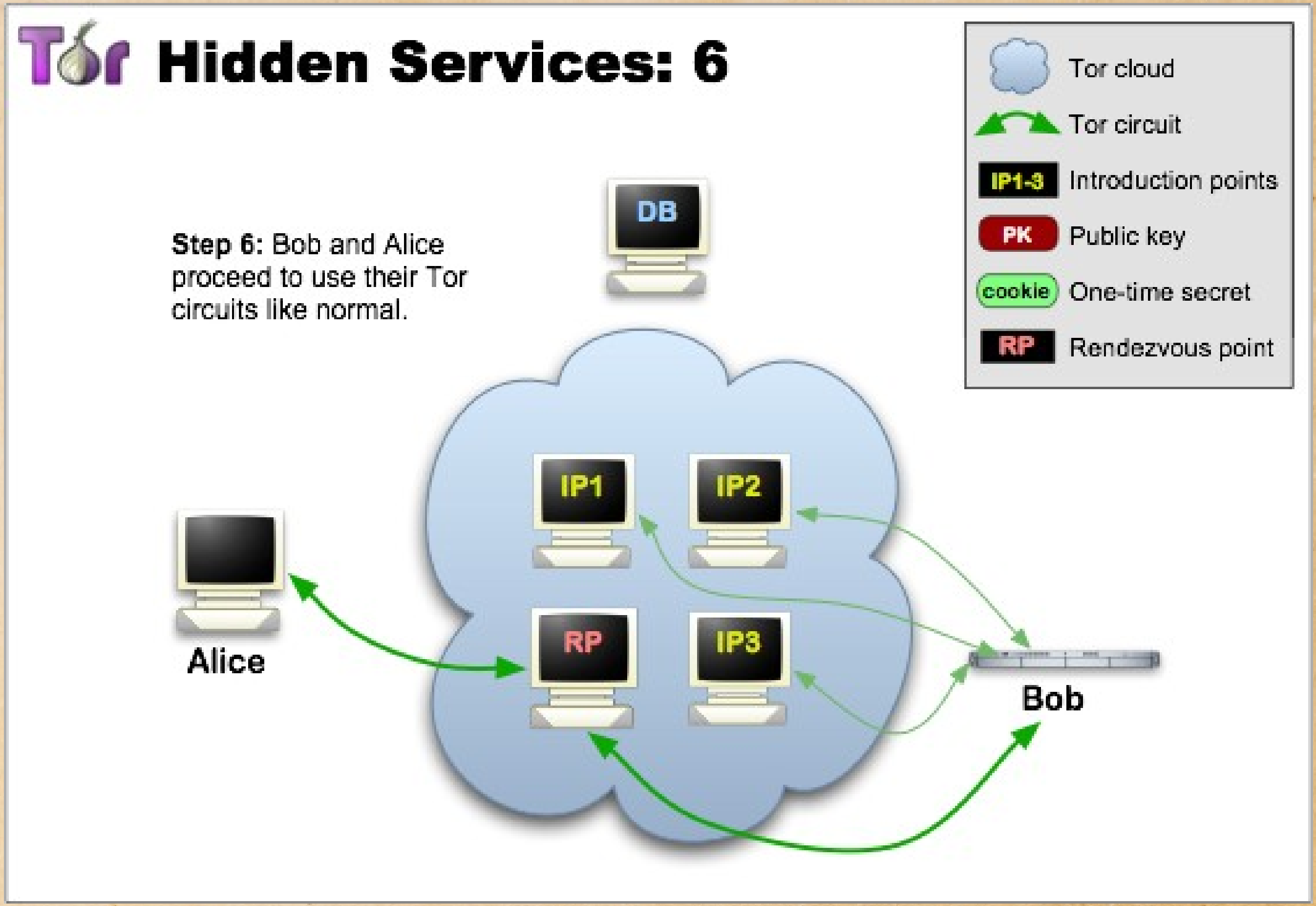
Hidden Service rendezvous point



the last step

- the rendezvous point notifies the client about successful connection establishment.
- connection between client and hidden service consists of 6 relay.

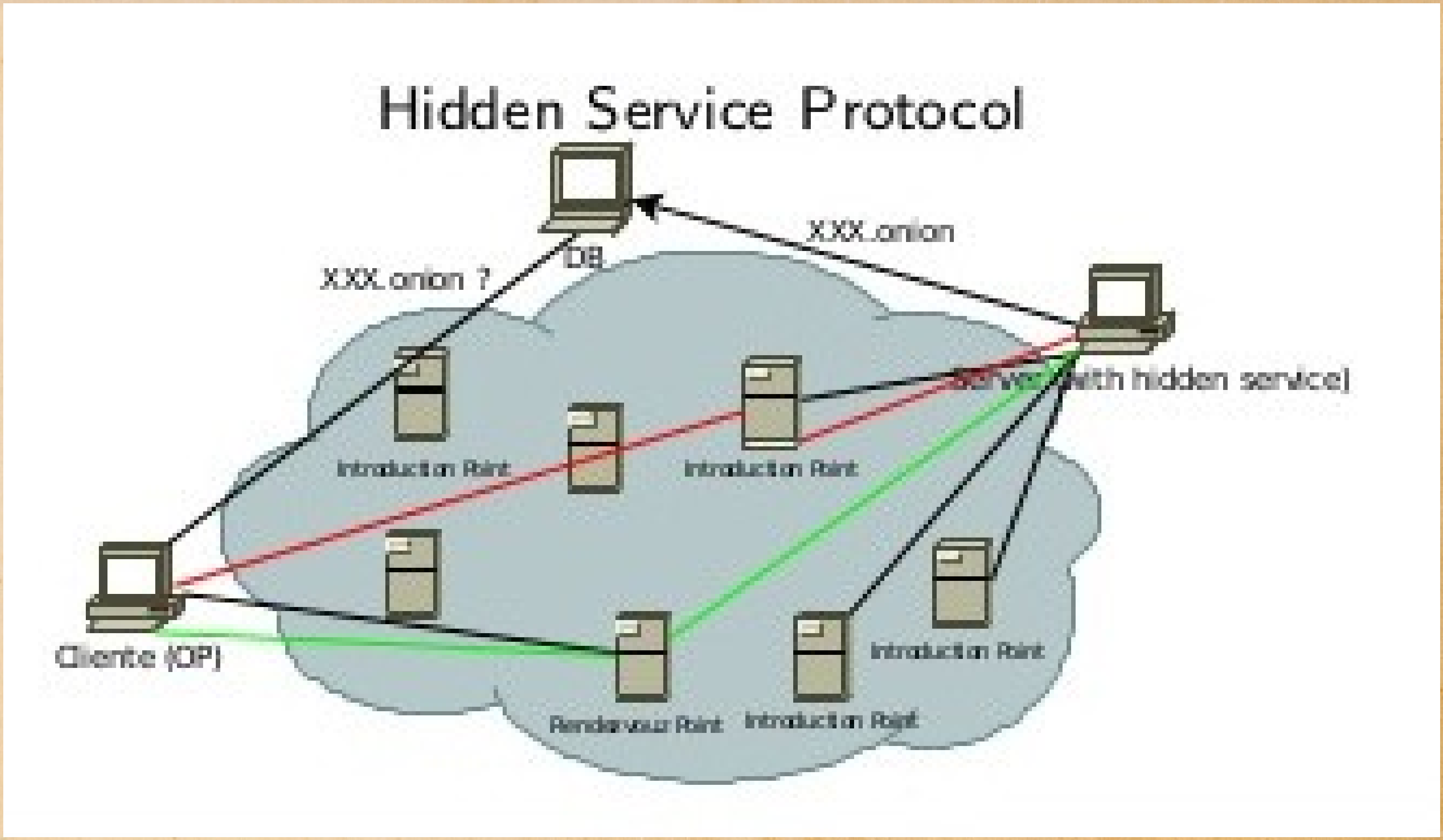
the last step



Hidden Services: 6

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

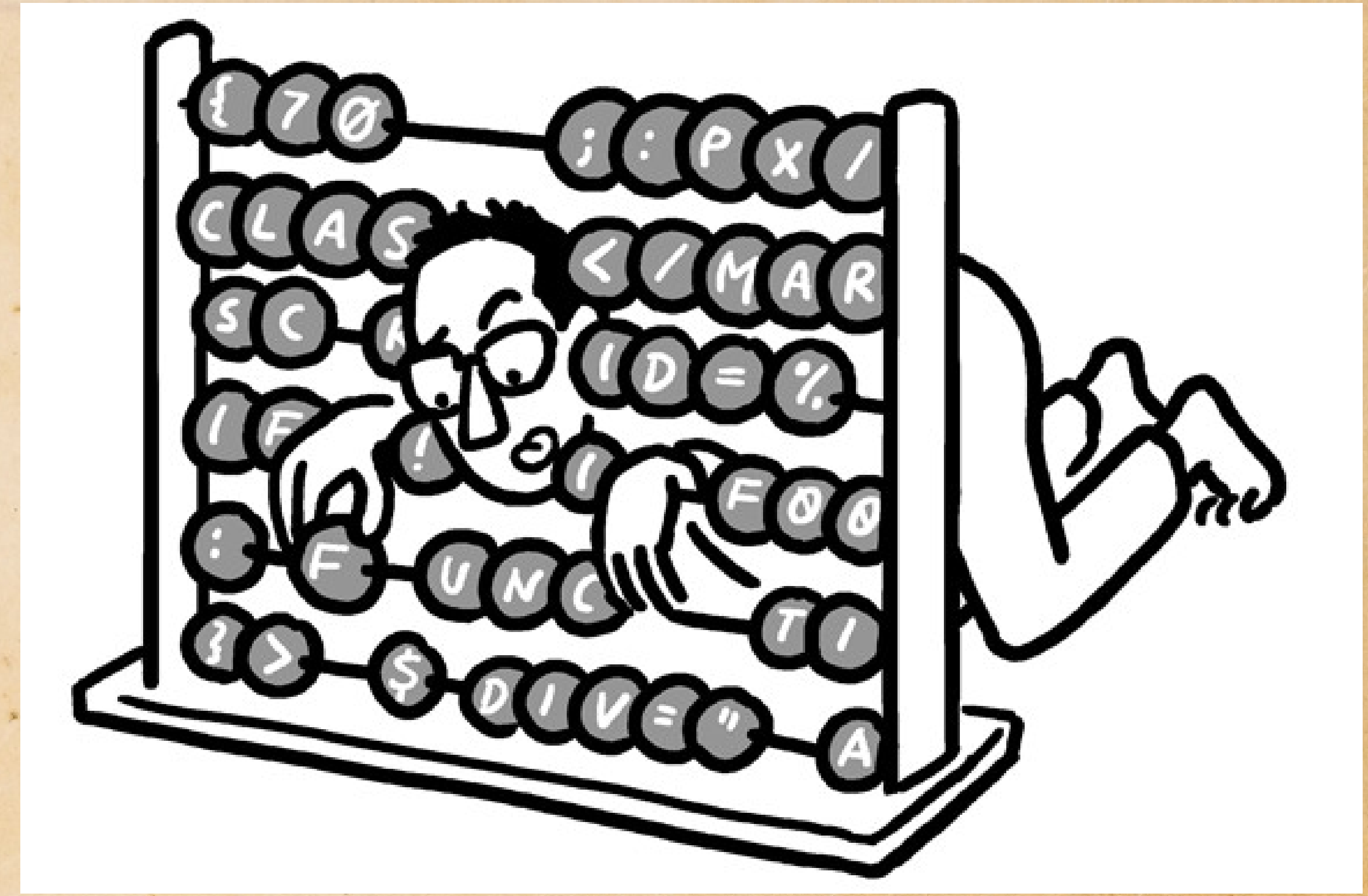
Hidden service protocol



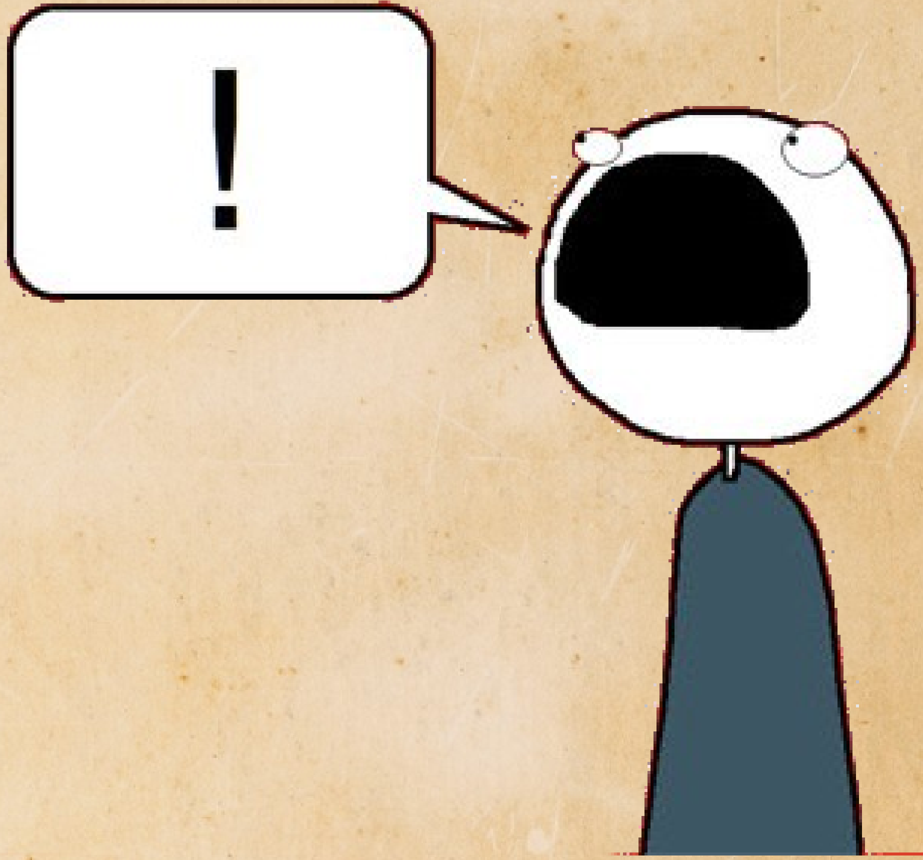
Xyz.onion

- SHA1 hash of the public key
- the first half of the hash is encoded to Base32
- the suffix „.onion“ is added.
- .onion names can only contain the digits 2-7 and the letters a-z and are exactly 16 characters long.

Why automatically-generated service name?

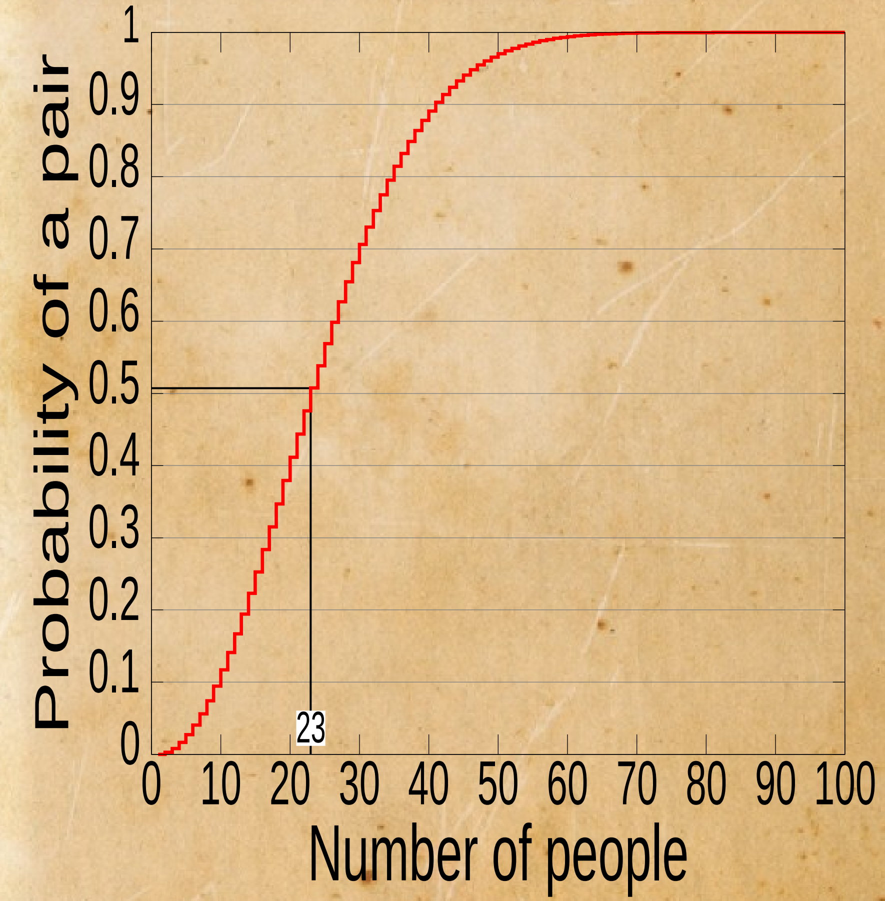


facebookcorewwwi.onion



Birthday attack

- cryptographic attack
- abuse communication between two or more parties
-



Get specific .onion address

- Shallot
- Scallion (GPU hashing)
- Eschalot (wordlist search)

test!

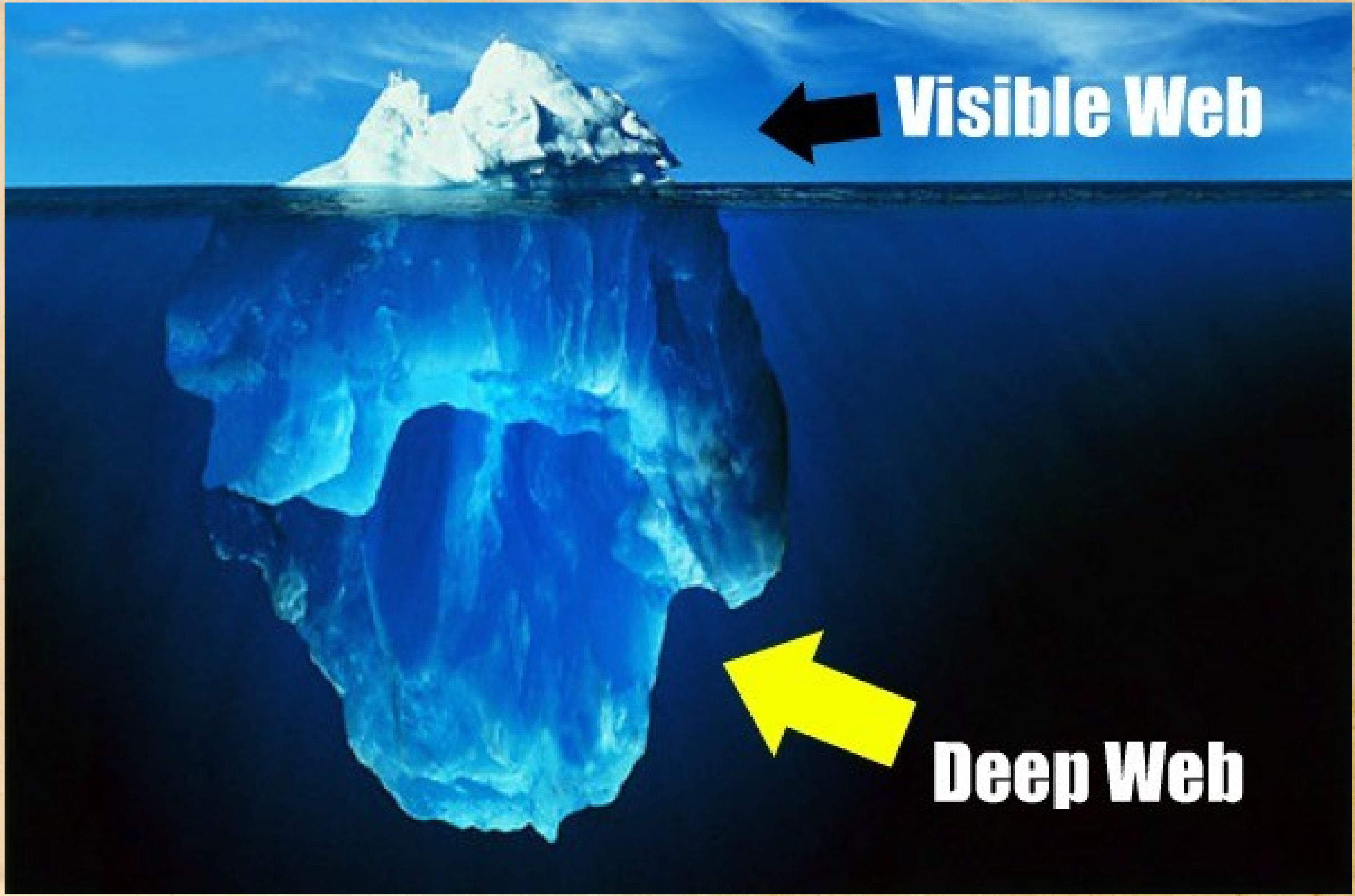
shallot

- https://codeload.github.com/katmagic/Shallot/zip/master
- ./configure && make
- ./shallot
- ./shallot ^onion
- Found matching domain after 22204717 tries: onion6r33t2v3sq7.onion

Shallot 1.5 GHZ

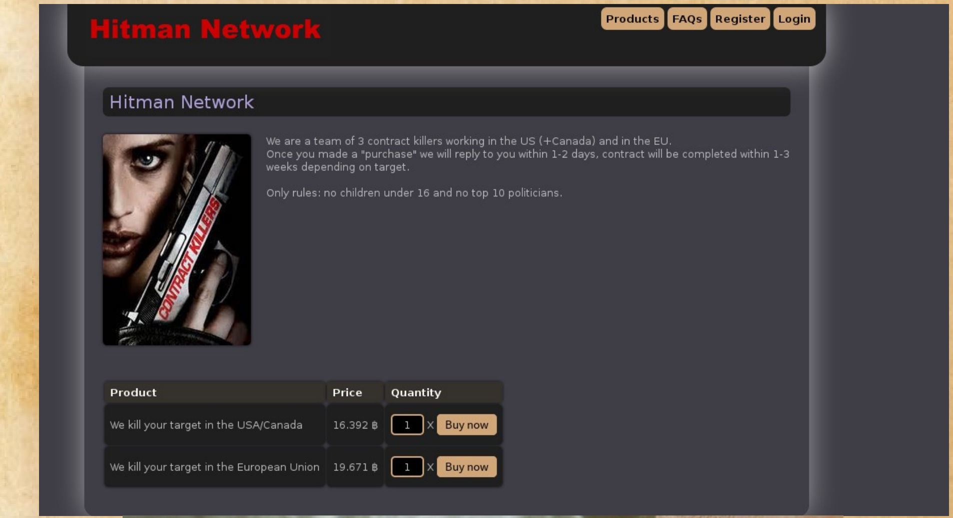
Characters	Time to generate
1	Less than 1 sec
3	Less than 1 sec
5	1 min
7	7 day
9	2,5 years
11	640 years
14	2.6 milion years

Hidden services



Who's using hidden service

- Hitman network
- drugs
- Child pornography
- Hacking
- Political (anarchism, ...)
- Warez



Hitman Network

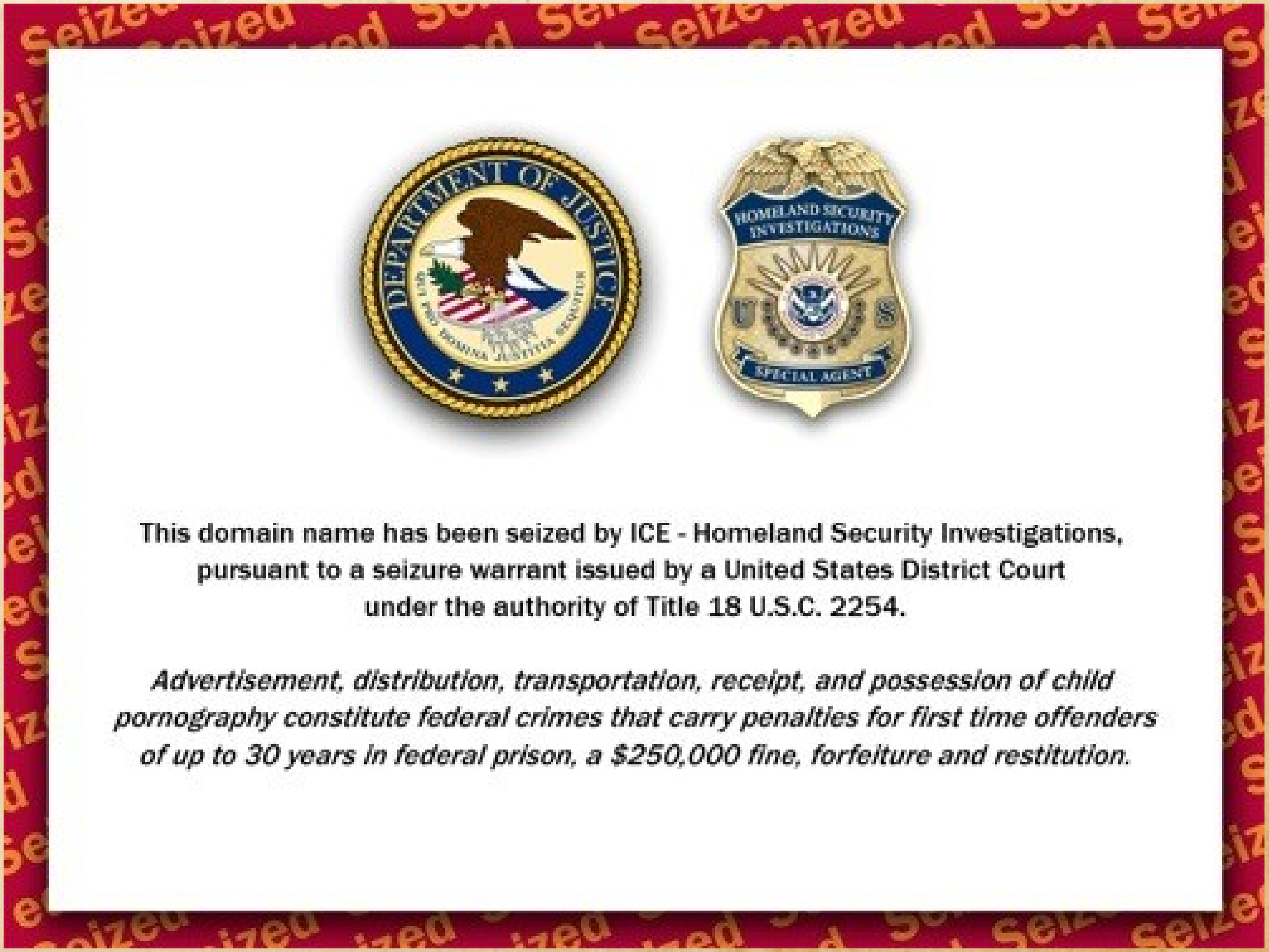
Hitman Network

We are a team of 3 contract killers working in the US (+Canada) and in the EU.
Once you made a "purchase" we will reply to you within 1-2 days, contract will be completed within 1-3 weeks depending on target.

Only rules: no children under 16 and no top 10 politicians.

Product	Price	Quantity
We kill your target in the USA/Canada	16.392 ฿	1 X Buy now
We kill your target in the European Union	19.671 ฿	1 X Buy now

Tor network hacked by FBI?



Seized
DEPARTMENT OF JUSTICE
QUI PRO DOMINA JUSTITIA SEQUITUR
HOMELAND SECURITY INVESTIGATIONS
U S
SPECIAL AGENT

This domain name has been seized by ICE - Homeland Security Investigations, pursuant to a seizure warrant issued by a United States District Court under the authority of Title 18 U.S.C. 2254.

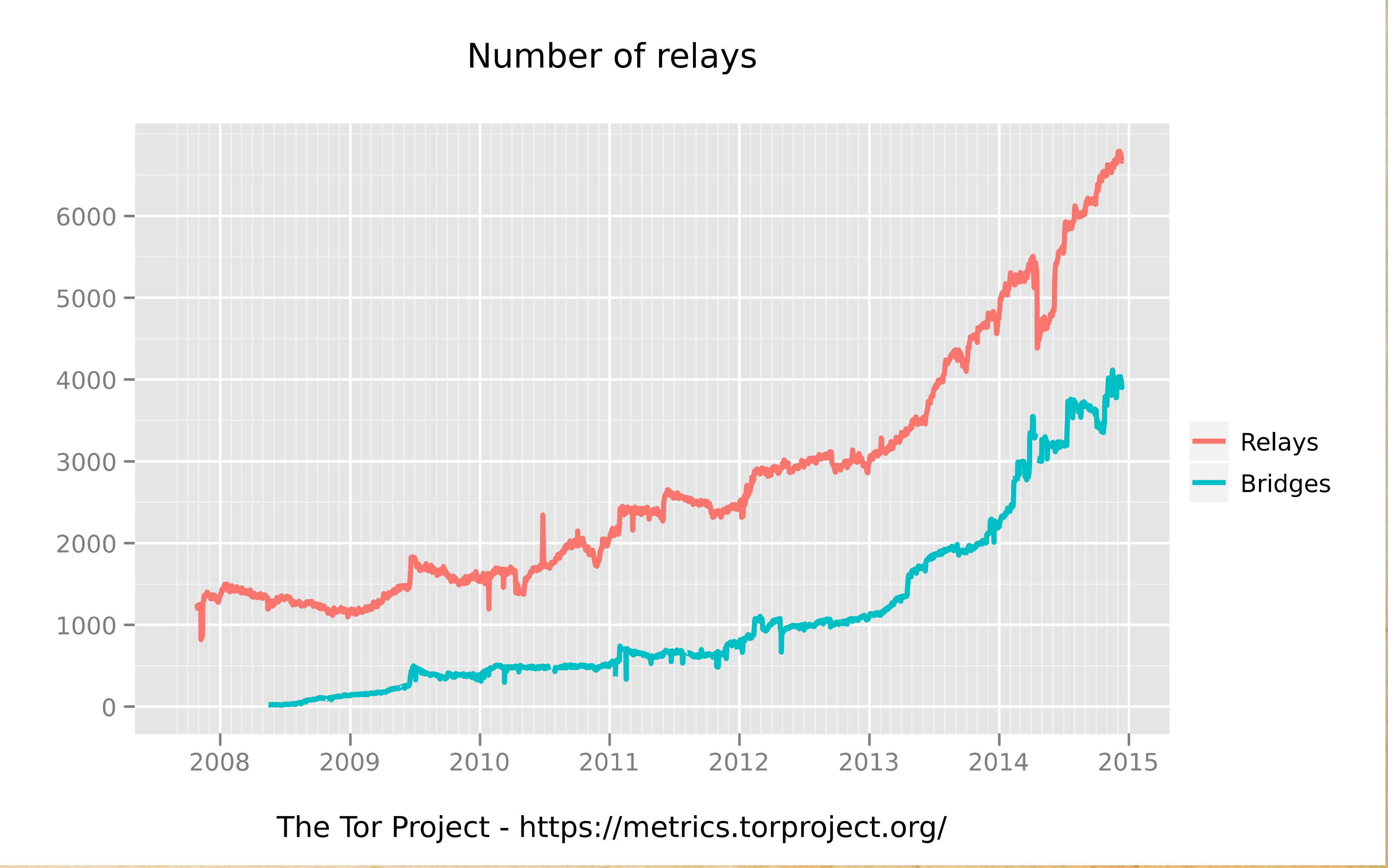
Advertisement, distribution, transportation, receipt, and possession of child pornography constitute federal crimes that carry penalties for first time offenders of up to 30 years in federal prison, a $250,000 fine, forfeiture and restitution.

Plausible deniability

List of most popular onion websites

- DuckDuckGo
- The Pirate Bay
- Facebook
- Blockchain.info
- Wikileaks
- SecureDrop

Graph Relays and bridges



- http://en.wikipedia.org/wiki/Onion_routing
- http://en.wikipedia.org/wiki/Tor_%28anonymity_network%29
- http://www.fbi.gov/news/pressrel/press-releases/more-than-400-.onion-addresses-including-dozens-of-dark-market-sites-targeted-as-part-of-global-enforcement-action-on-tor-network
- https://www.torproject.org/docs/hidden-services.html.en
-

- https://www.eff.org/pages/tor-and-https
- https://metrics.torproject.org/
- http://en.wikipedia.org/wiki/Plausible_deniability
- http://www.theguardian.com/technology/2014/oct/31/facebook-anonymous-tor-users-onion
-