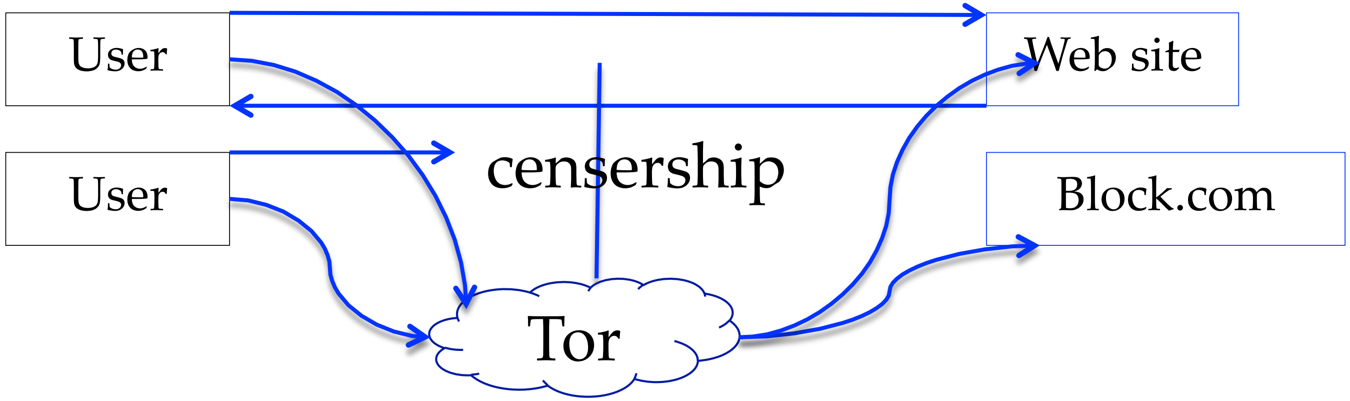
Introduction

- Overview of Tor
  - What is Tor? Why use Tor?
- How Tor works
  - Encryption, Circuit Building, Directory Server
- Drawback of Tor’s directory server
- Potential solution
  - Using DNS Security Extension

What is Tor

- A distributed overlay network based on voluntarily run relays around the world
- Provides low latency anonymity to TCP-based applications
- Protects users from being identified online
  - Journalists, activists, business people
- Circumvents Censorship

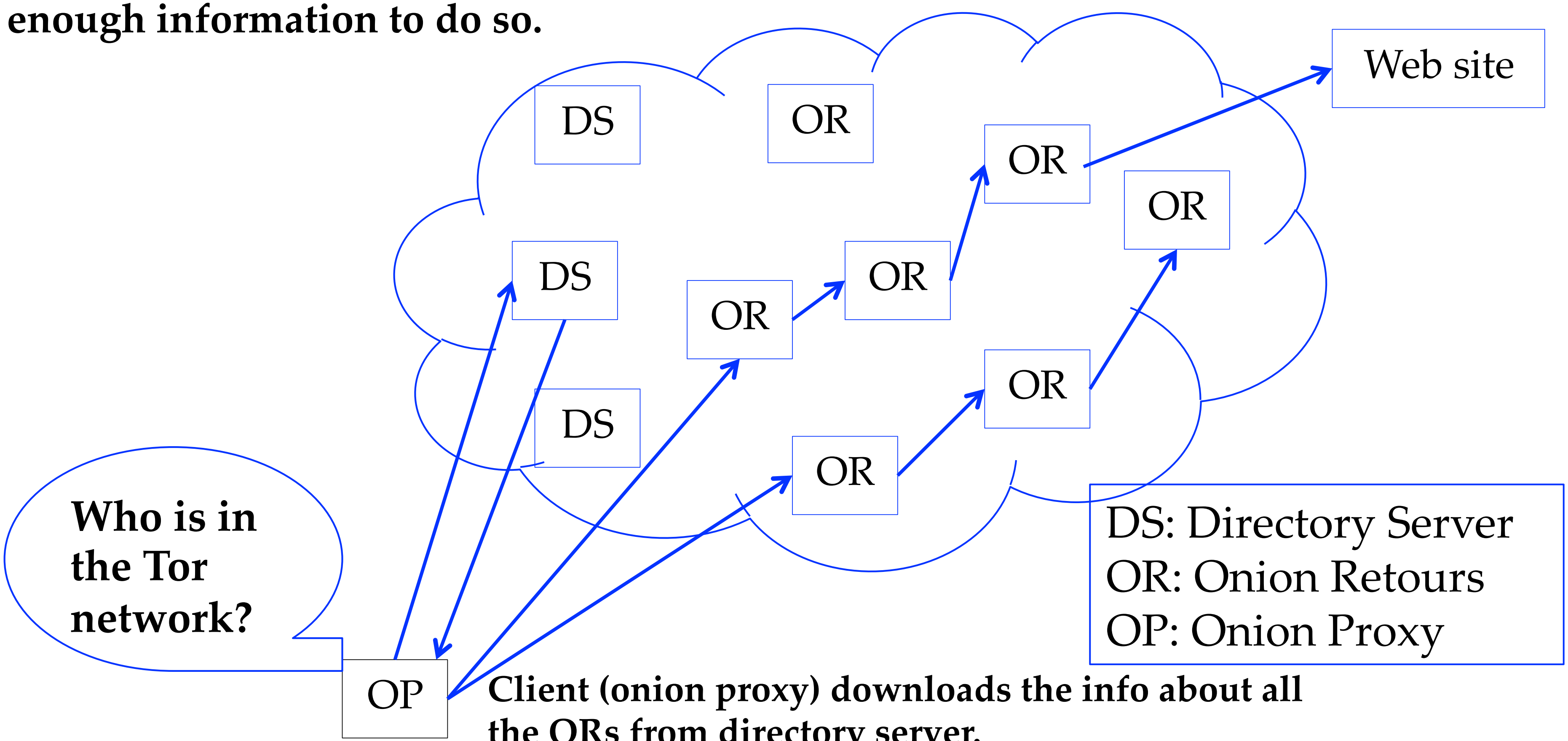


Tor Network: the Basic

- Directory nodes
  - Servers set up by Tor project
  - List all the nodes available in Tor network
- Relay nodes
  - Servers run by volunteers around world
- Onion proxy
  - Proxy running on client computer
- Circuit
  - An encrypted virtual tunnel
  - Made of a chain of Tor relay nodes
  - Traffic routed through multiple relays from the user to the final destination

Overview

Tor begins building circuits as soon as it has enough information to do so.

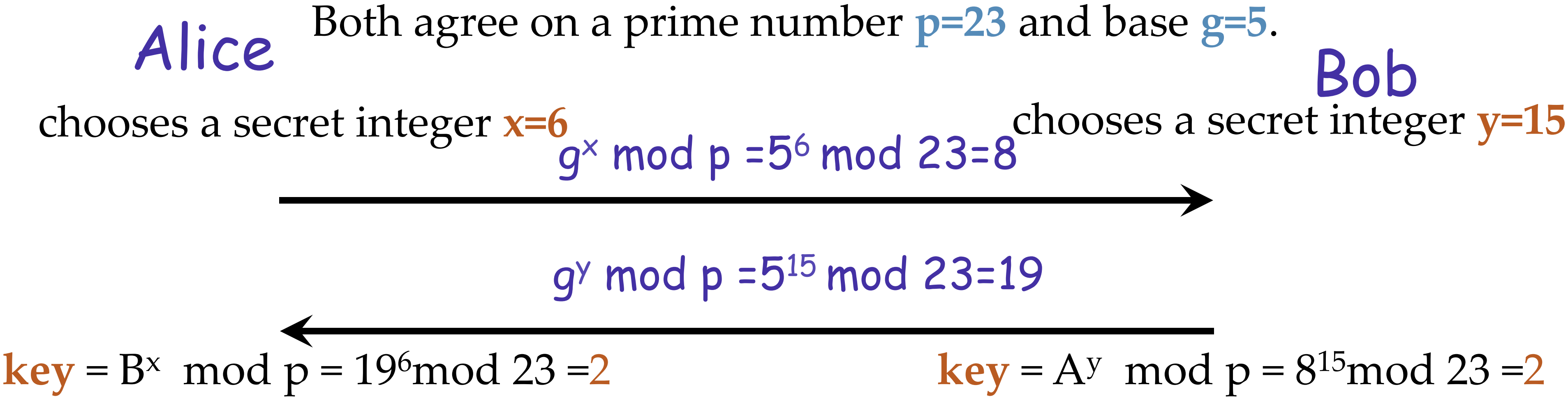


DS: Directory Server
OR: Onion Retours
OP: Onion Proxy

Client (onion proxy) downloads the info about all the ORs from directory server.
Includes keys, ips, ports, bandwidth, etc.

OP opens a new connection, attaches to a circuit

Diff-Hellman Key Exchange



Alice

Both agree on a prime number $p=23$ and base $g=5$.

Bob

Alice chooses a secret integer $x=6$

Bob chooses a secret integer $y=15$

$g^x \bmod p = 5^6 \bmod 23 = 8$

$g^y \bmod p = 5^{15} \bmod 23 = 19$

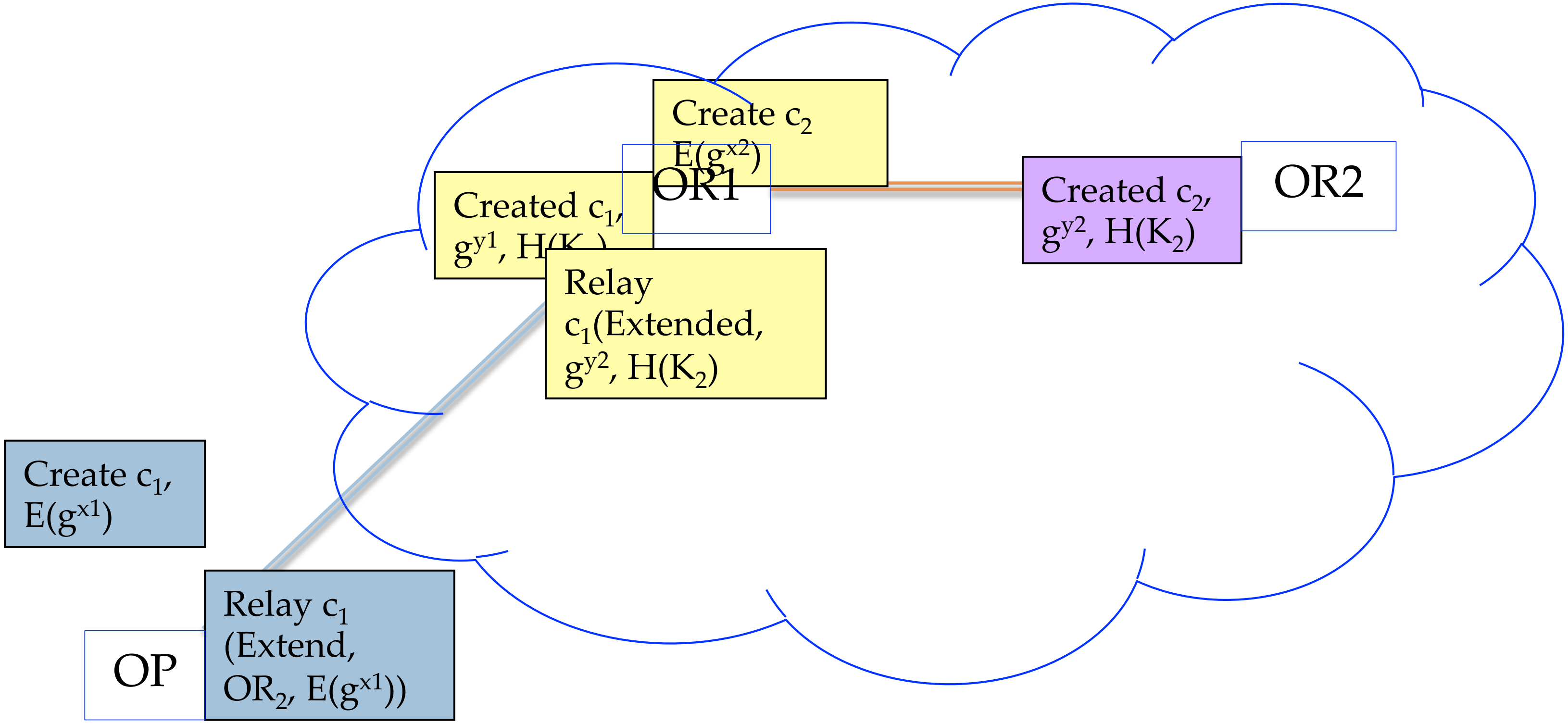
key $= B^x \bmod p = 19^6 \bmod 23 = 2$

key $= A^y \bmod p = 8^{15} \bmod 23 = 2$

- “Two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel”

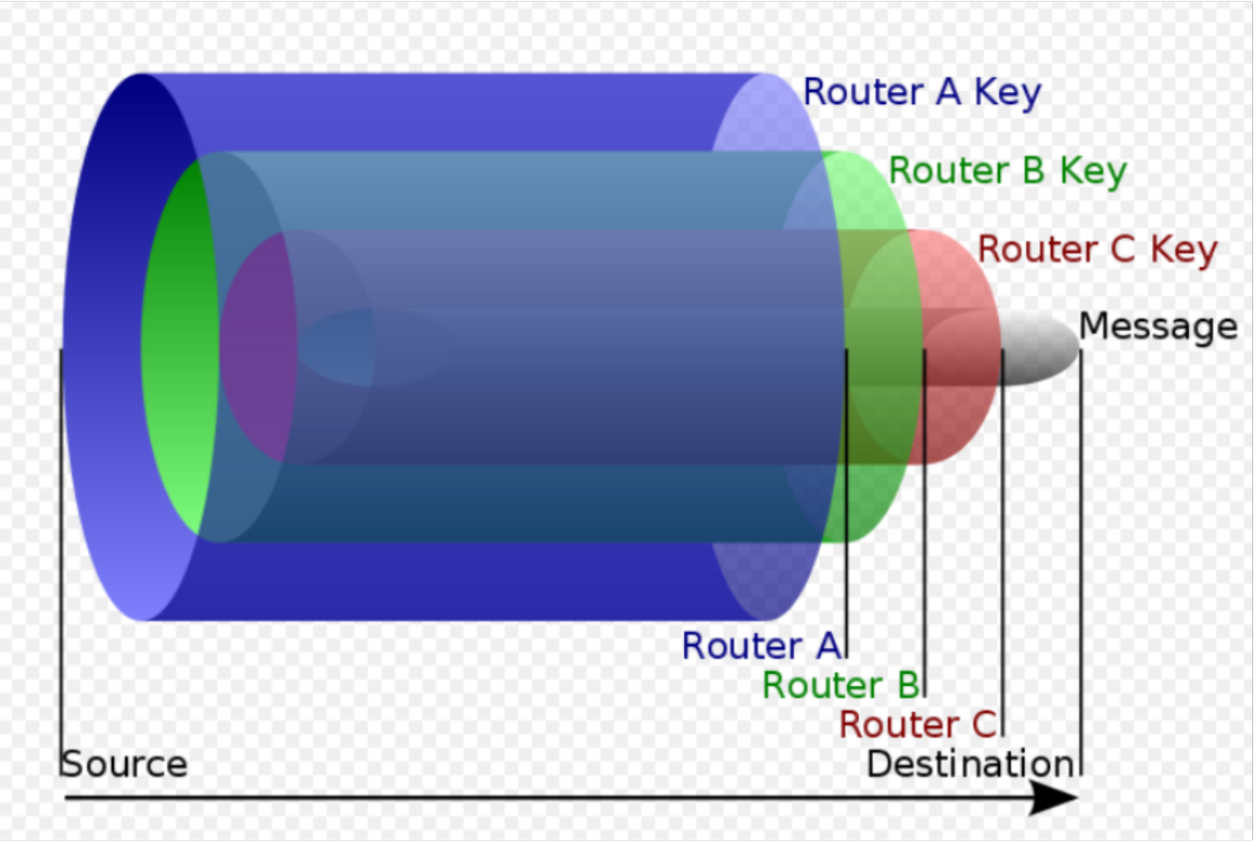
Example from wikipedia

How a Tor Circuit is built



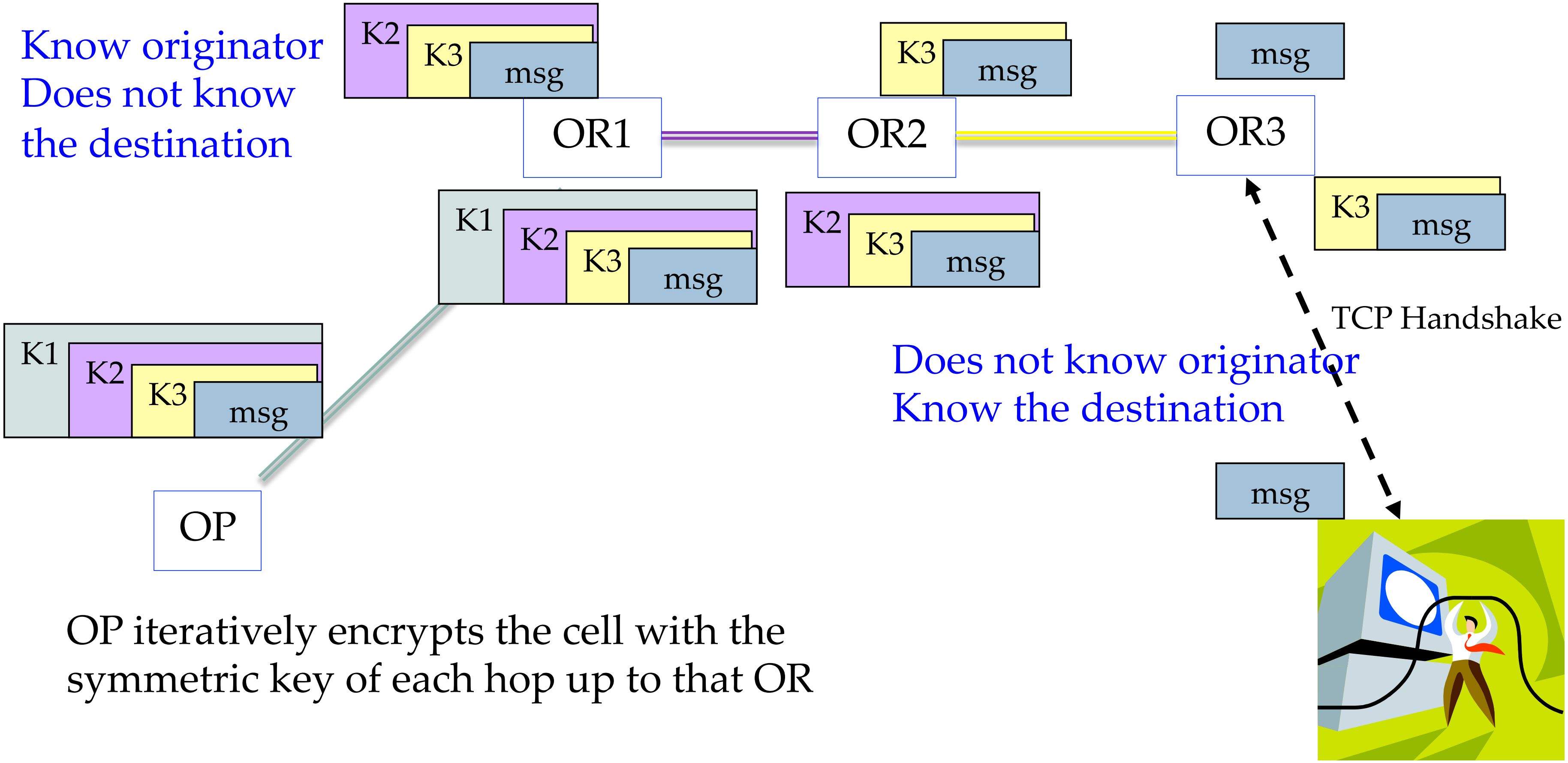
- ❑ [illegible] it chooses,
- ❑ [illegible] handshake (g^{x1})
- ❑ [illegible] handshake (g^{x2})
- ❑ Encrypted with the onion key of the OR2

Tor’s Message



http://en.wikipedia.org/wiki/File:Onion_diagram.svg

How Tor Fetches a Website



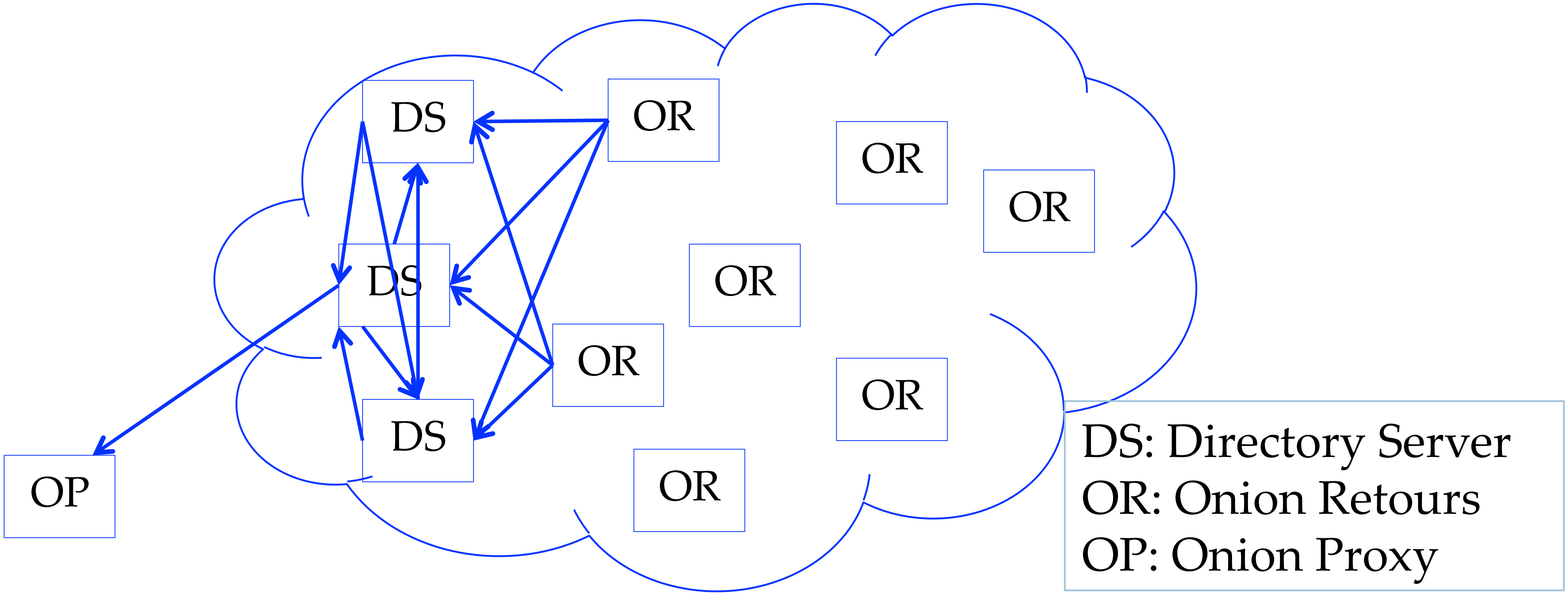
Know originator
Does not know
the destination

Does not know originator
Know the destination

OP iteratively encrypts the cell with the symmetric key of each hop up to that OR

How Clients Know the Topology

Each onion router periodically signs and sends its keys, bandwidth, port, etc., to the Tor directory servers



DS: Directory Server
OR: Onion Retours
OP: Onion Proxy

Each directory server periodically signs and sends its individual view of the Tor network to other directory servers.

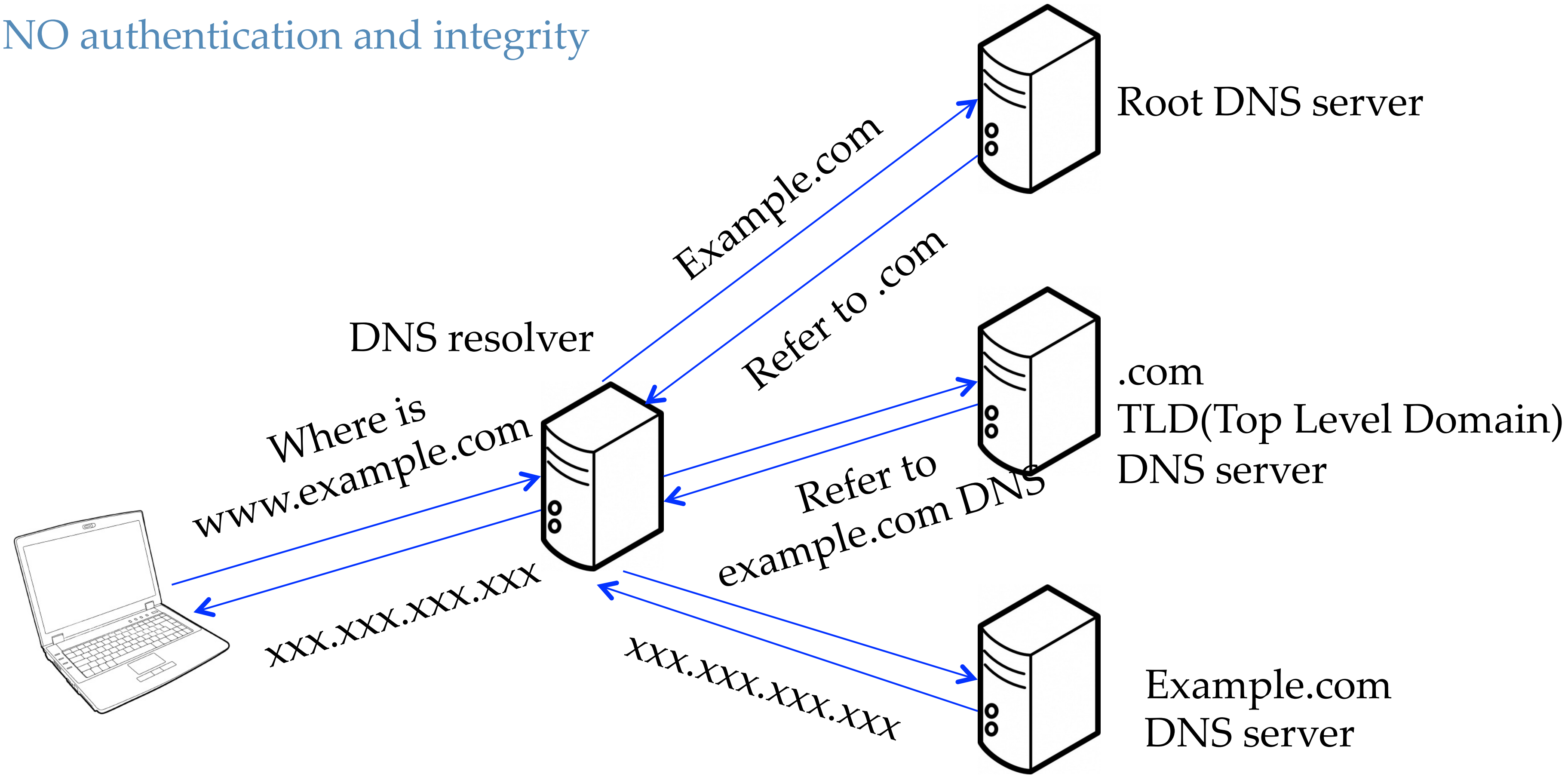
Clients (onion proxies) download the consensus file from a directory server

A Problem with Tor Directory Servers

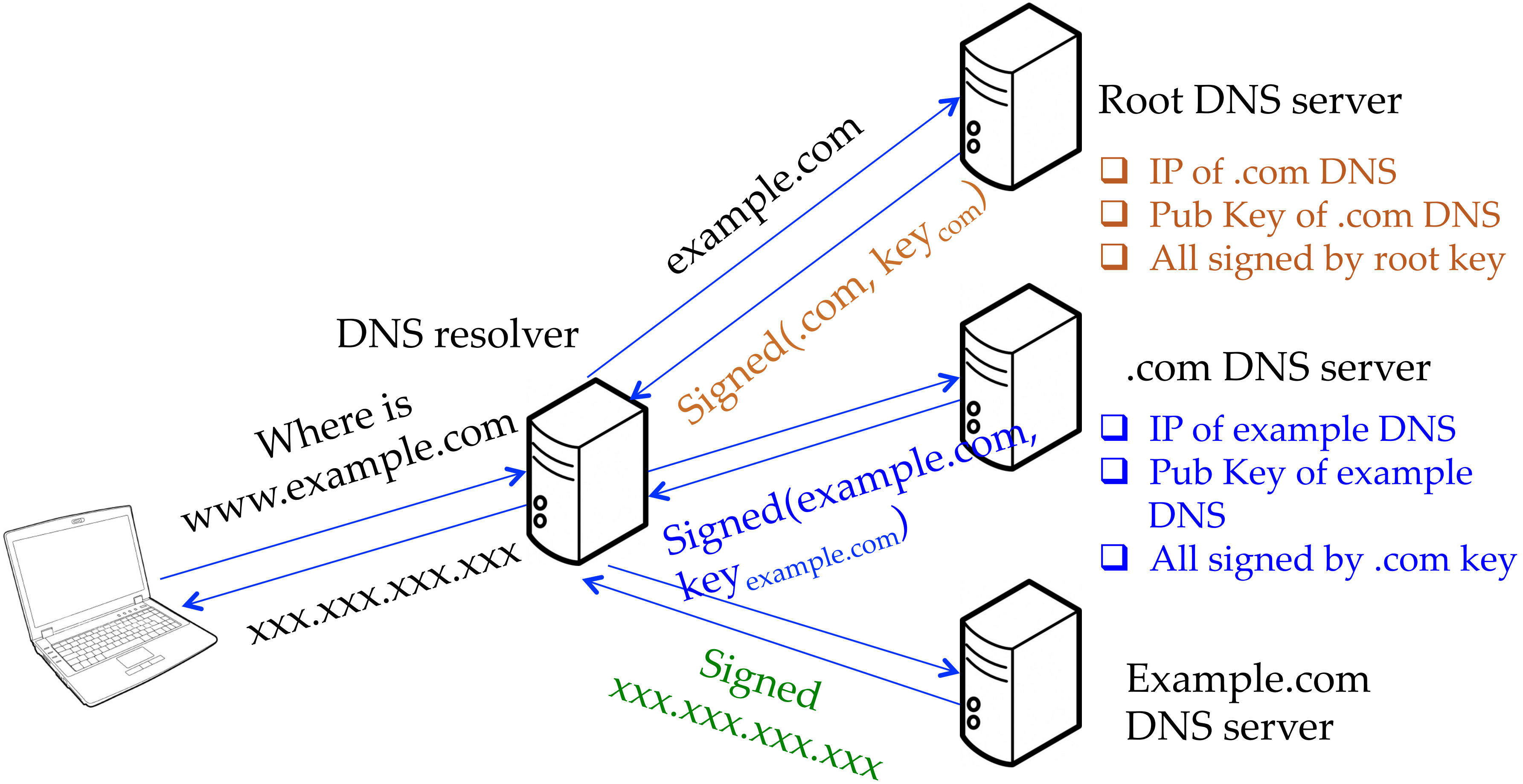
- Tor requires each directory server and client user (onion proxy) to know all of the relay nodes in the Tor network
- Clients periodically ask directory servers:
  - Who is there in the Tor network?
  - What is their status and info?
    - Is a relay node active? Public key, port, IP, etc.
- What if the directory server is inaccessible
  - E.g., Blocked by ISP?
- Potential solution: Ask DNS for directory server information

DNS Resolution

NO authentication and integrity



DNS Security Extension Resolution



Root DNS server

- □ IP of .com DNS
- □ Pub Key of .com DNS
- □ All signed by root key

.com DNS server

- □ IP of example DNS
- □ Pub Key of example DNS
- □ All signed by .com key

Example.com DNS server

DNS resolver

Where is www.example.com

xxx.xxx.xxx.xxx

example.com

Signed(.com, key$_{\text{com}}$)

Signed(example.com, key$_{\text{example.com}}$)

Signed xxx.xxx.xxx.xxx

Consensus File (Partial)

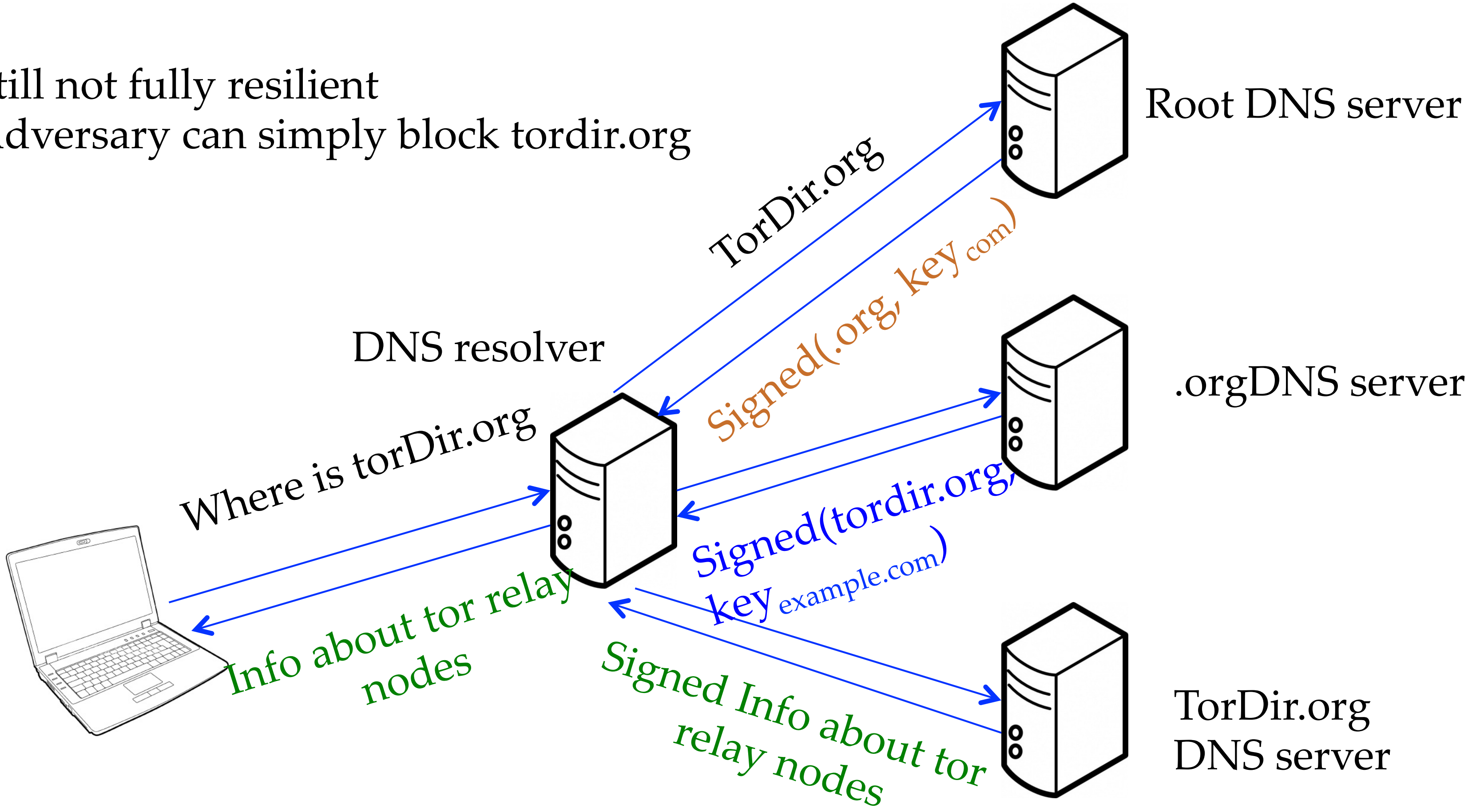
network-status-version 3	A document format version.
vote-status consensus	Vote status
consensus-method 13	Consensus methods that are using
valid-after 2013-04-25 19:35:00	Start time of the consensus
fresh-until 2013-04-25 19:40:00	Time to produce next consensus
valid-until 2013-04-25 19:50:00	The time this consensus expires
voting-delay 20 20	
client-versions	
server-versions	

Consensus File (Partial)

- @downloaded-at 2013-04-28 06:52:04
- router relay8 128.220.221.150 9000 0 9500
- onion-key
- signing-key
- router-signature
- Signature from directory servers

Replacing Directory Server with DNSSEC

Still not fully resilient
Adversary can simply block tordir.org



Replacing Directory Server with DNSSEC

- Many random domain names
  - Change regularly
  - Generate by hash function
- Each domain name is only responsible for a subset of all available Tor relay nodes.
  - When querying one domain, a client is only provided with a subset of relay nodes
- Info about relays is encrypted using domain name's keys
  - Domain name key changes regulary

Conclusion

- Difficult to block all domain name
  - Thousands of domain name
  - Each responsible for subset of relays
  - As long as one domain name is not blocked
- Difficult to block all IP address of relay nodes
  - Directory info is encrypted
  - Encrypted key regular change

References

- https://www.torproject.org/
- Main Tor Specification, https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=tor-spec.txt
- Tor Version 3 Directory Server Specification, https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=dir-spec.txt
- A New Approach to DNS Security (DNSSEC), http://www.cs.jhu.edu/~ateniese/papers/dnssec.pdf
- Huston, G. 2010, DNSSEC-A Review, http://www.potaroo.net/ispcol/2010-06/dnssec.html