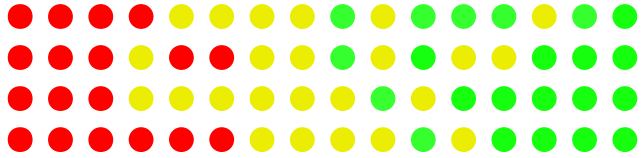
The Invisible Internet Project

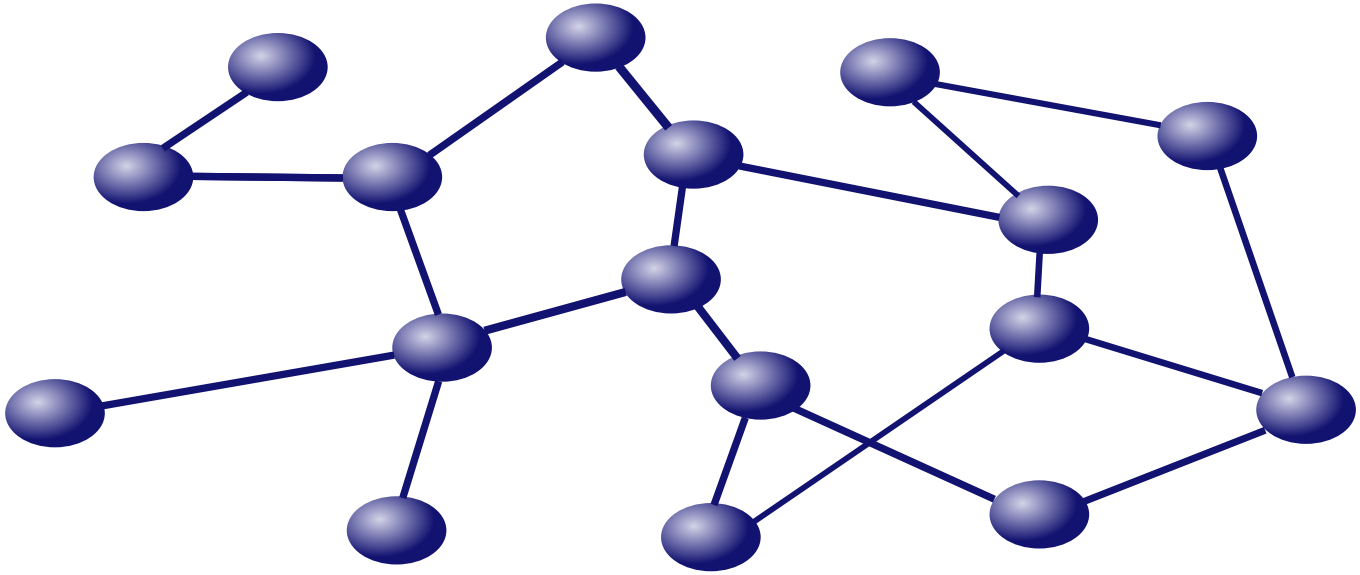
Andrew Savchenko

Moscow, Russia

FOSDEM 2018
3 & 4 February

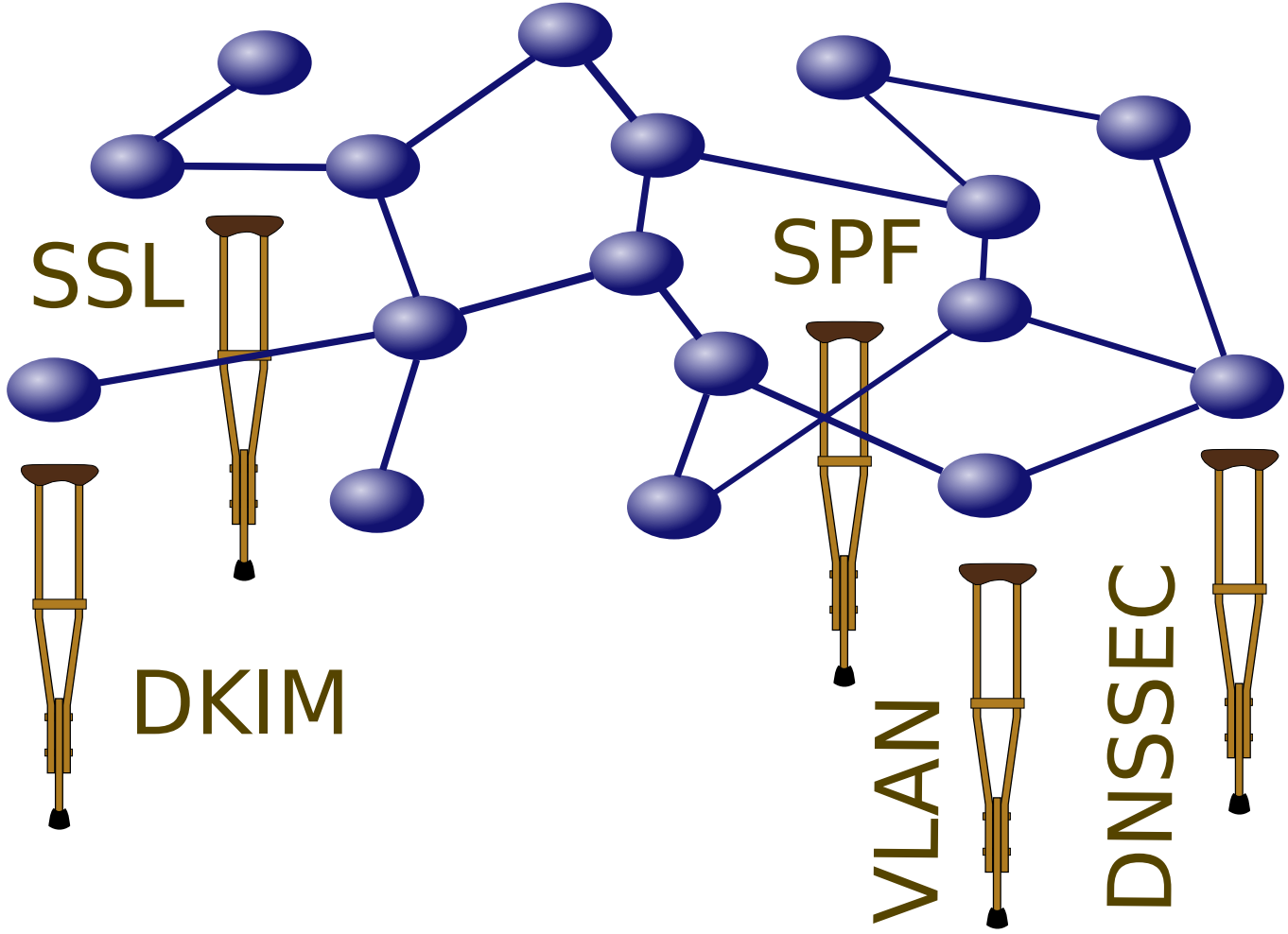


The Arpanet

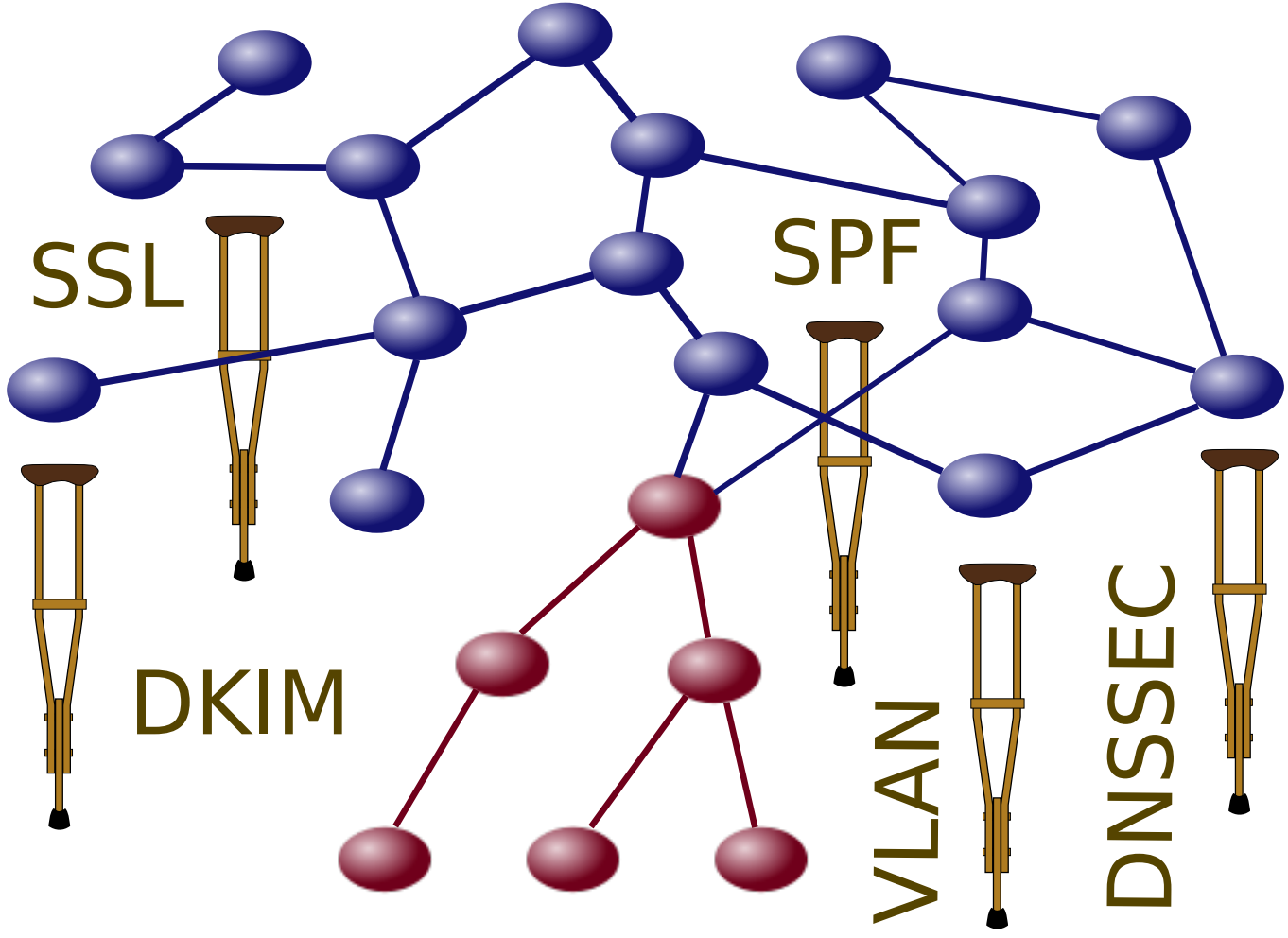


- Designed to withstand external infrastructure damage
- No internal threats considered

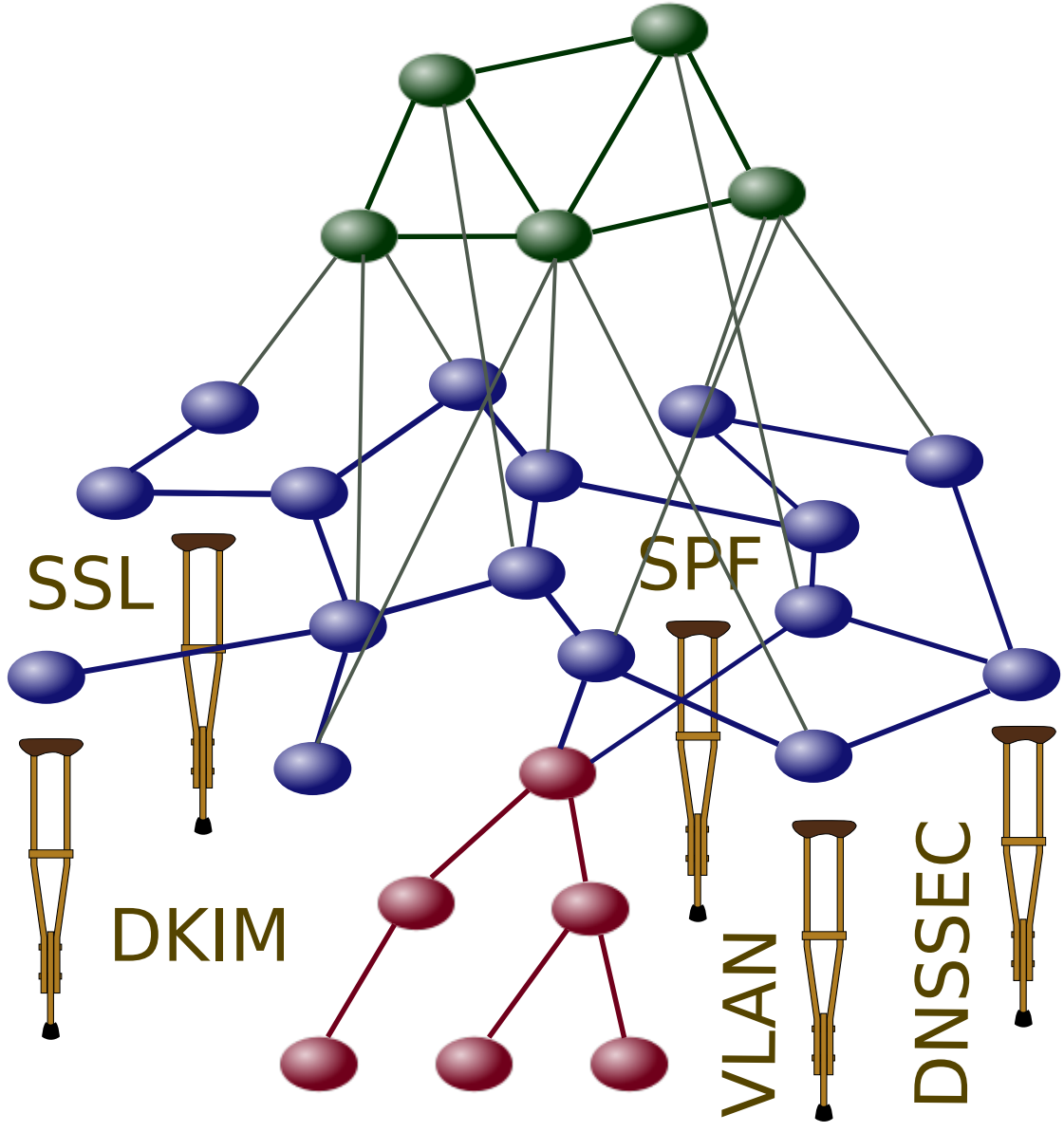
The Internet



The Internet



The Tor



The Tor

Pros:

- First world-wide overlay network
- Hidden services
- Scale

Cons:

- Entry/exit points
- Asymmetric:
  ~ 8‘000 nodes[1] [1] : ~ 4‘500‘000 users [2]
- Highly centralized: only **10** directory servers [3]

[1] relays + bridges

The Tor

Pros:

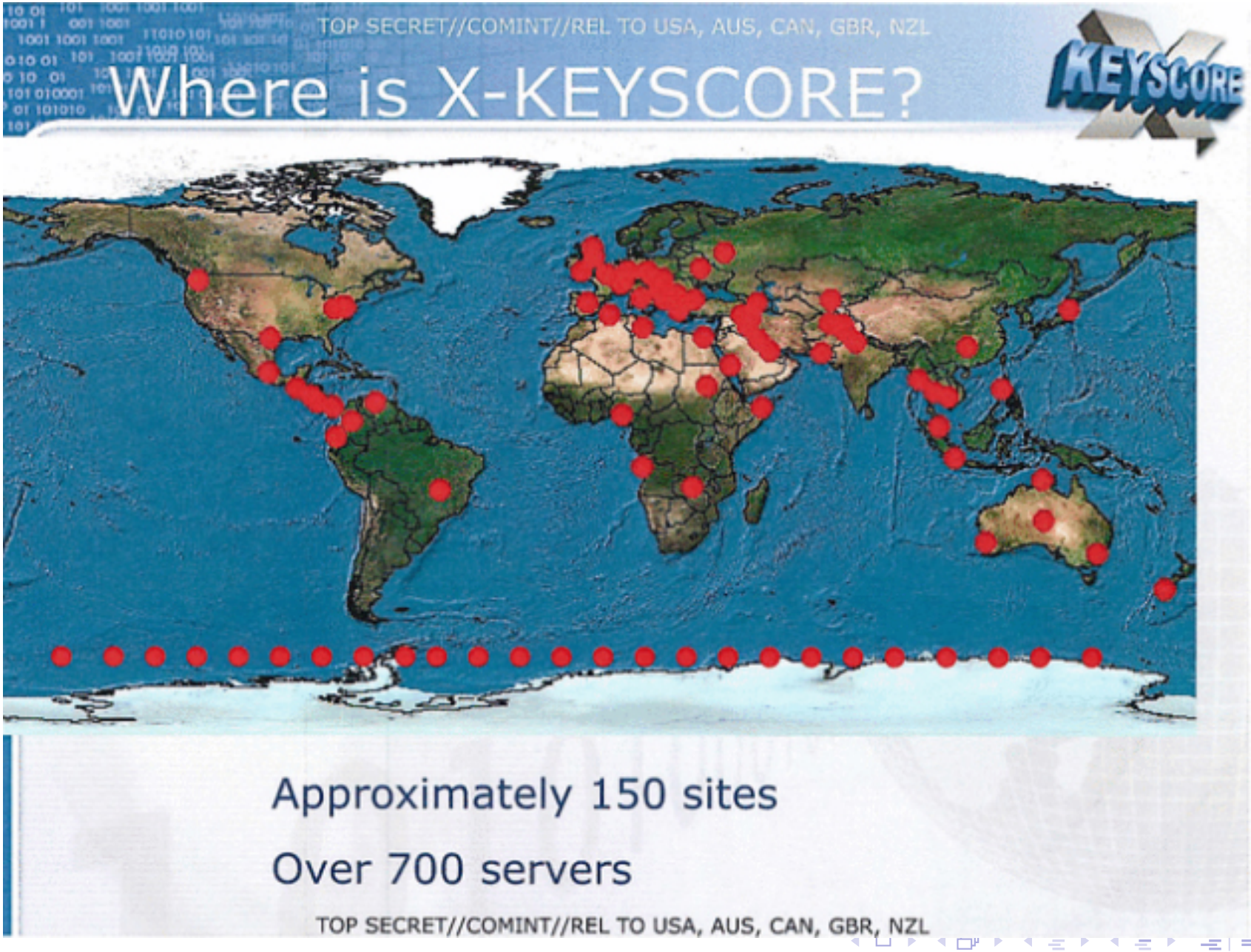
- First world-wide overlay network
- Hidden services
- Scale

Cons:

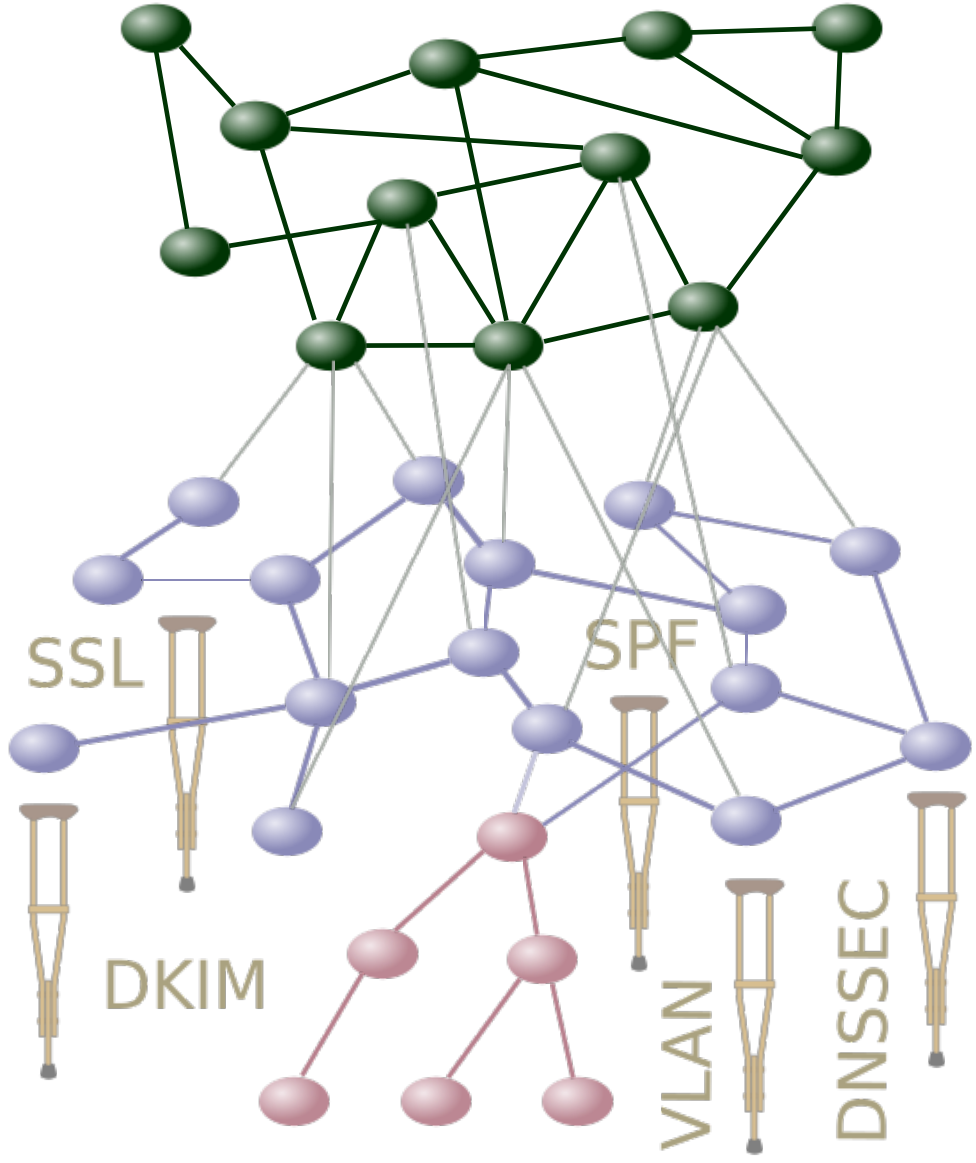
- Entry/exit points
- Asymmetric:
  $\sim 8‘000$ nodes[1] [1] : $\sim 4‘500‘000$ users [2]
- Highly centralized: only **10** directory servers [3]

[1] relays + bridges

Global Surveillance



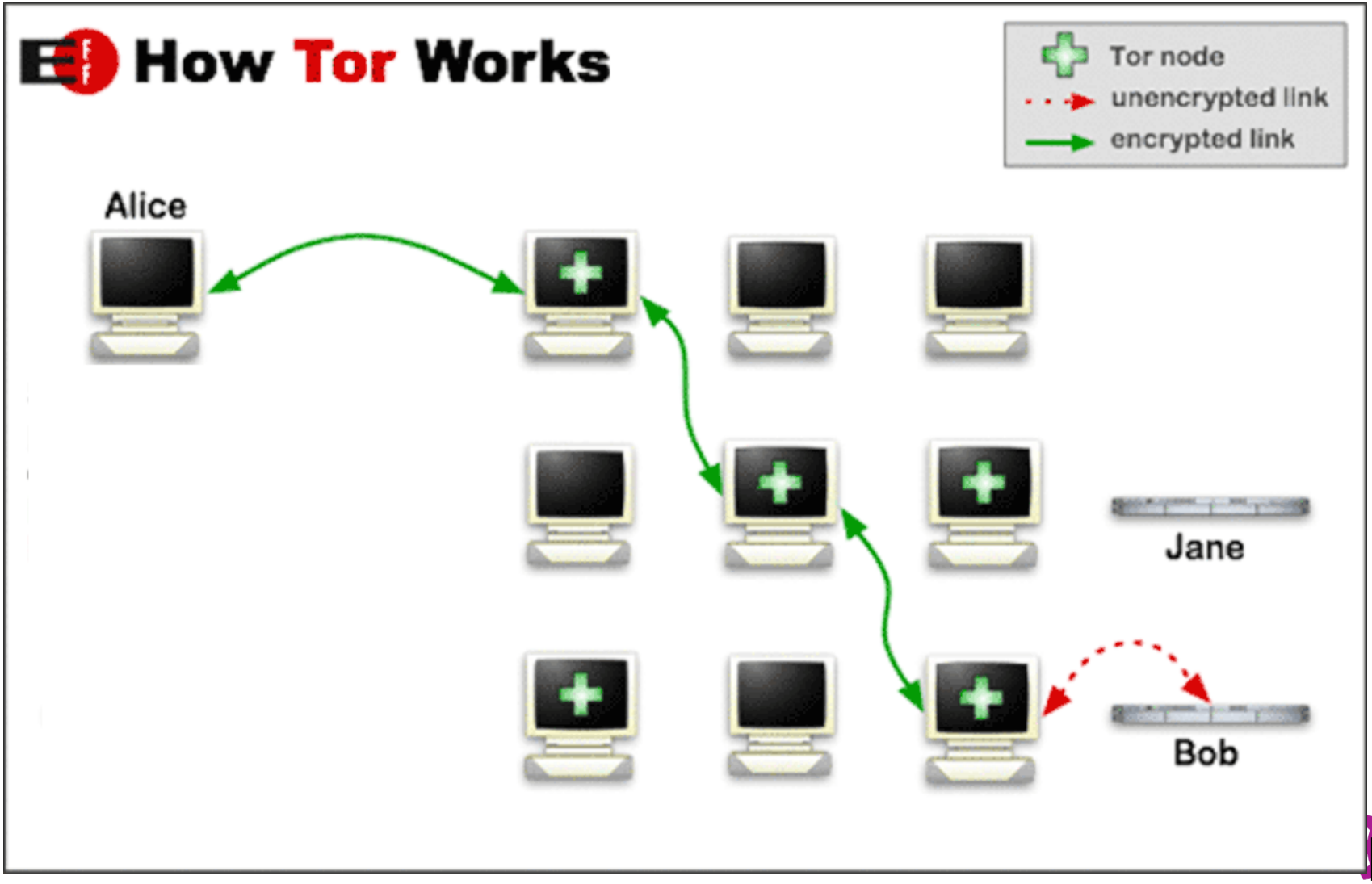
The I2P



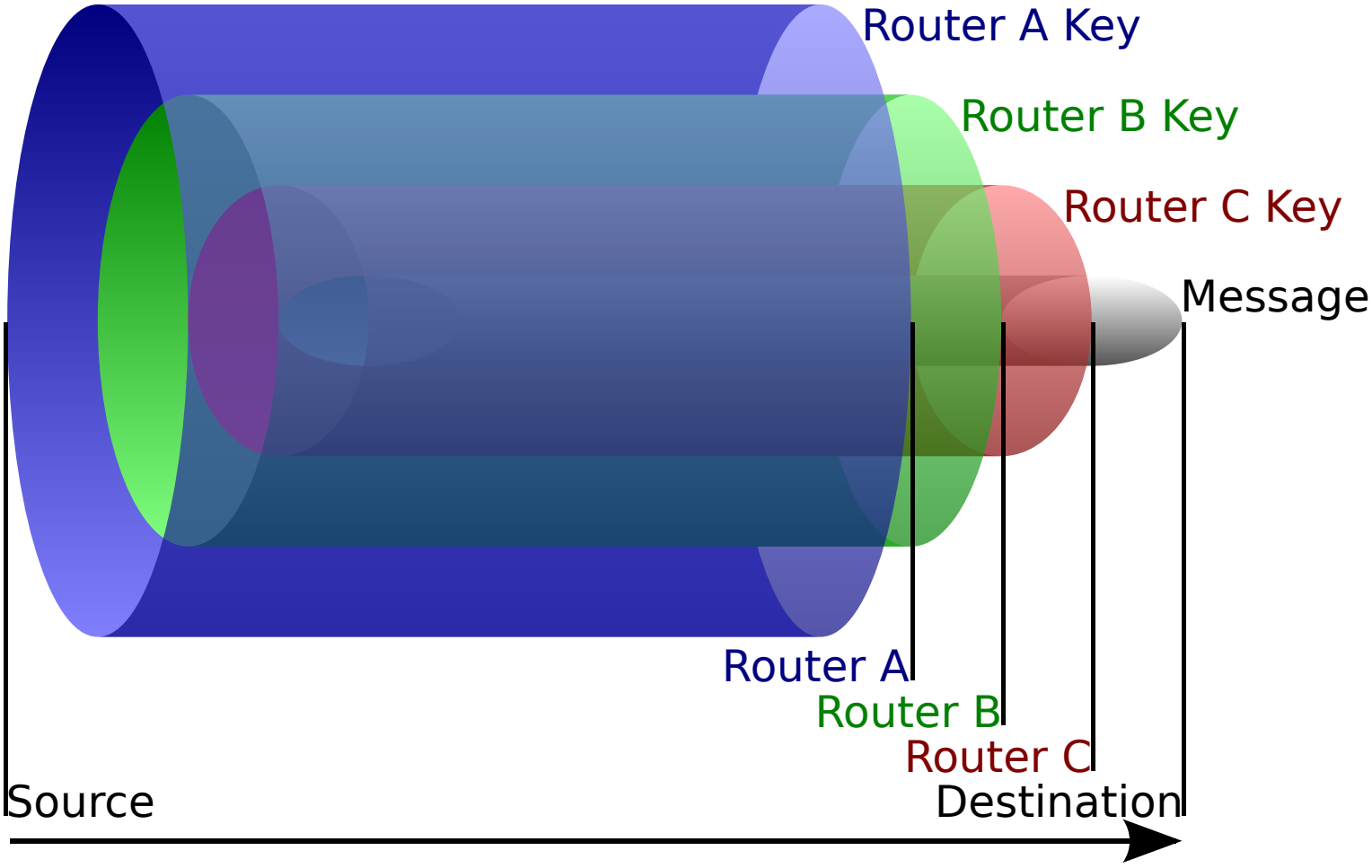
The I2P Design

- No entry/exit nodes [4]
- Full decentralization
- Use minimal trust possible
- Wide range of protocols supported: TCP, UDP, RAW…
- $\sim 50‘000 \div 60‘000$ nodes [5, 6]
  - In order just to monitor network special research is required [7]
- Unidirectional tunnels

The Onion Routing

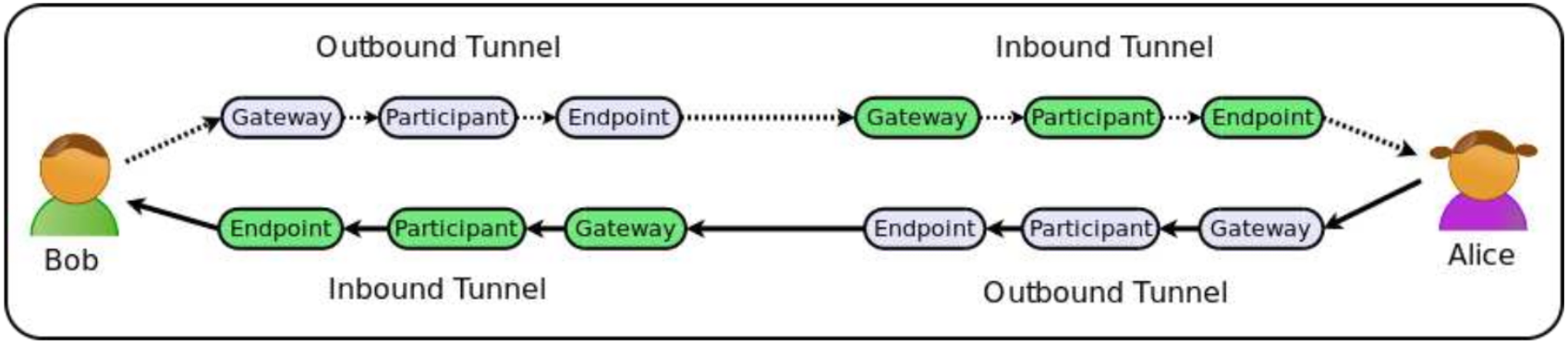


The Onion Routing



[9]

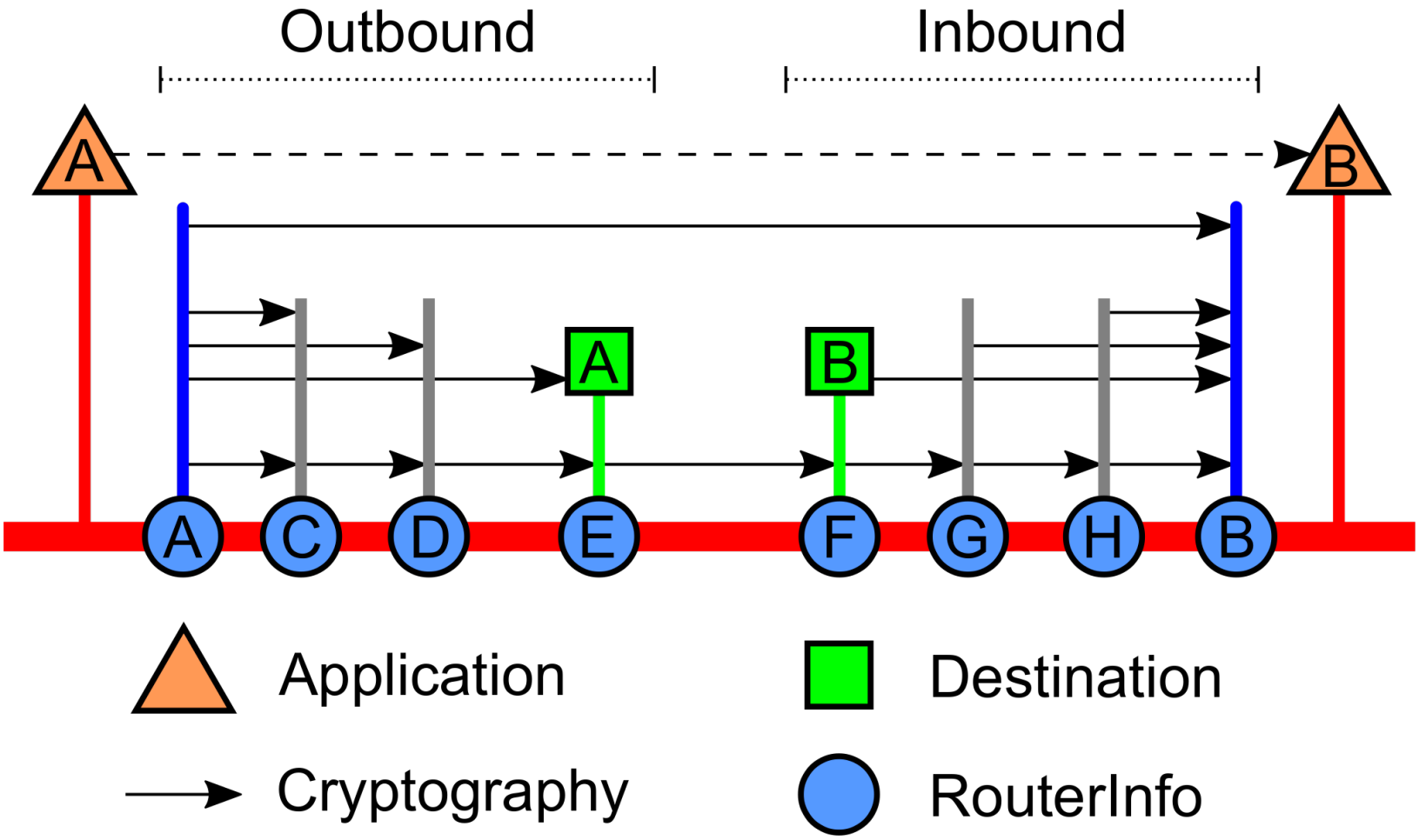
The I2P Tunnels



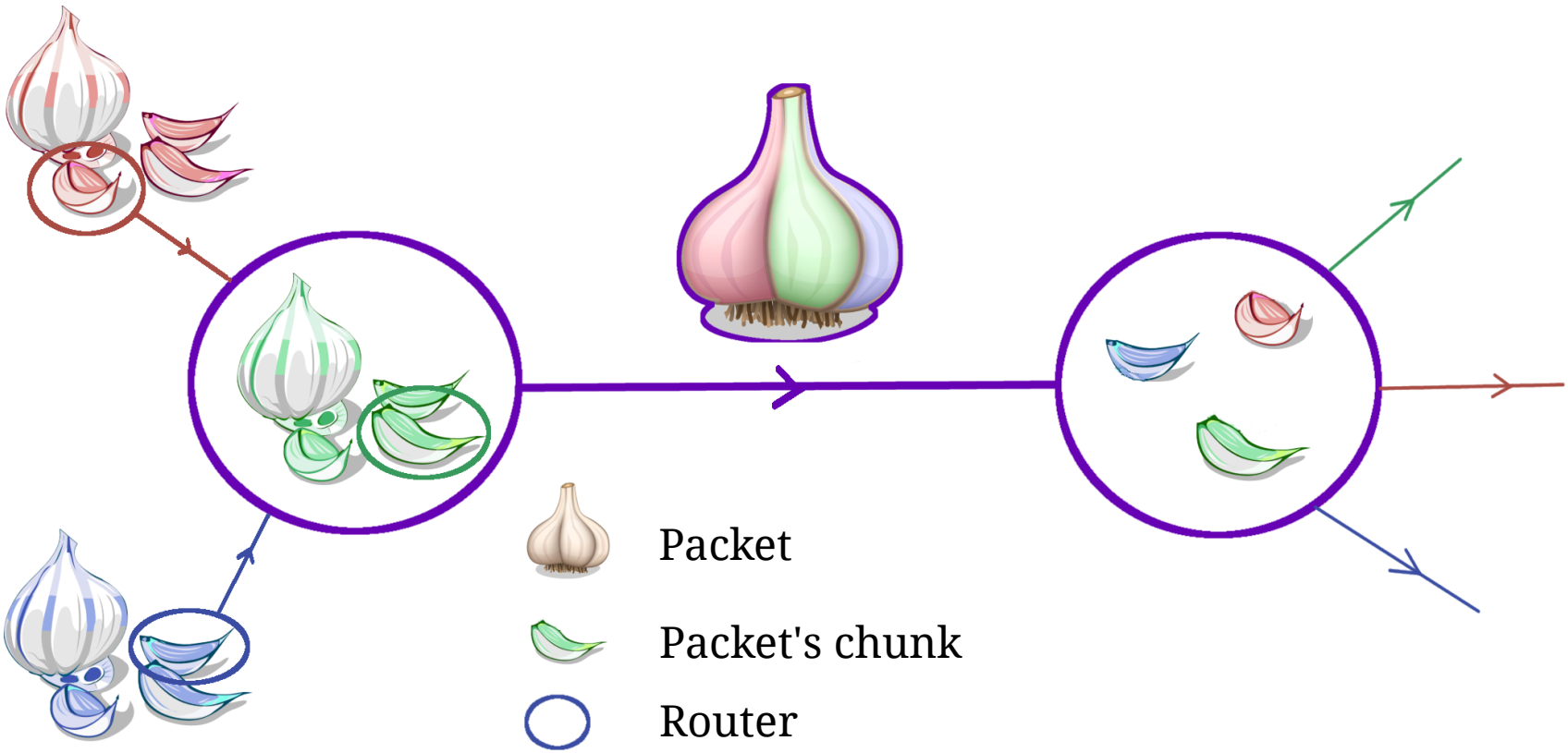
[7]

- Connect tunnel endpoints
- Different inbound and outbound tunnels
- Outbound endpoints are hidden
- Configurable tunnel length (usually 2-3)

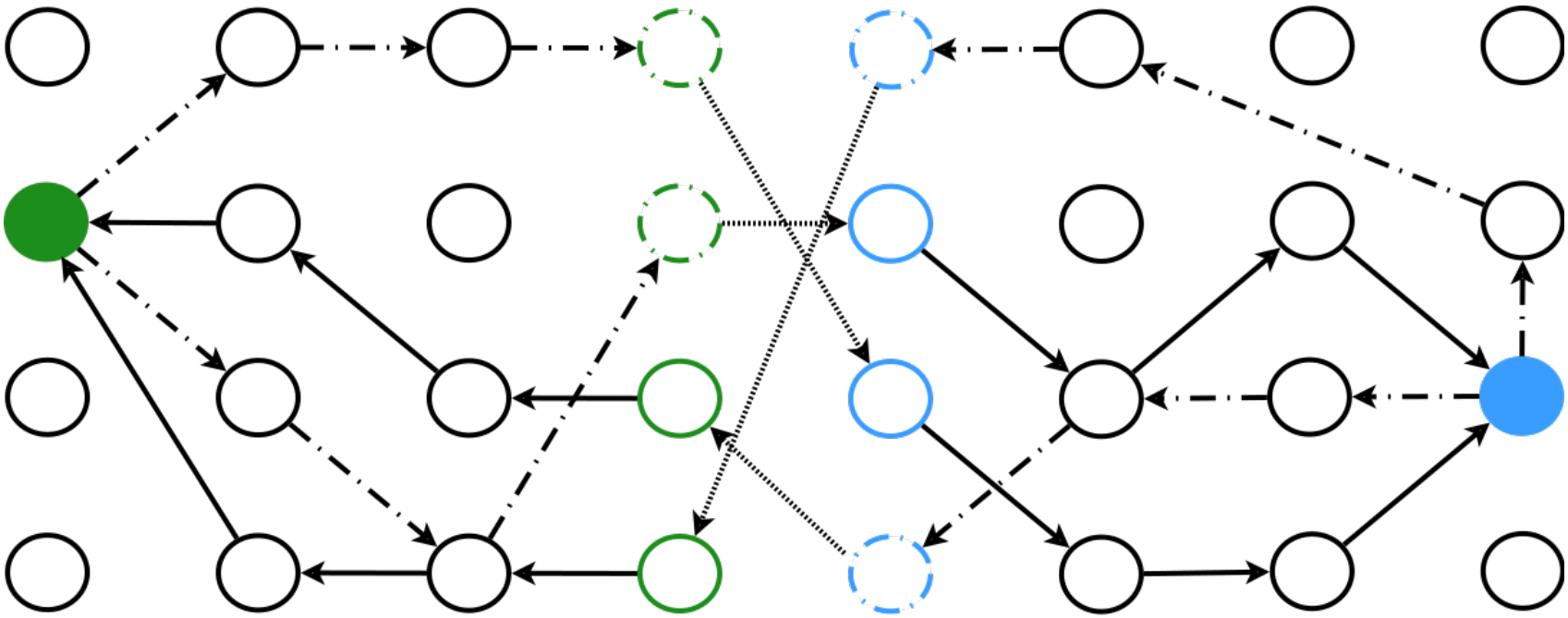
Three I2P Layers



The Garlic Routing



Ping-Pong: 2 chunks, 3 hops



Inbound
Outbound
Intertunnel

Outbound endpoints are hidden

Tunnels regen in ~10 min or at request

The Network Database

- No DNS-like centralized services
- Distributed (DHT-like) netDB is used:
  - RouterInfo (router contacts)
  - LeaseSets (destination endpoints)
- Public key based identification and connections

RouterInfo:

- ID (encryption and signing pub keys)
- contact (proto, IP, port)
- aux data
- all above is signed

The Network Database

- No DNS-like centralized services
- Distributed (DHT-like) netDB is used:
  - RouterInfo (router contacts)
  - LeaseSets (destination endpoints)
- Public key based identification and connections

RouterInfo:

- ID (encryption and signing pub keys)
- contact (proto, IP, port)
- aux data
- all above is signed

The Network database

Each node generates:

- encryption key
- garlic end-to-end encryption key
- signing key
- everything is signed into 516+ byte cert

Management:

- distributed netDB
- by *floodfill* routers
- $\sim$ $20'000 \div 30'000$ ($\sim 600 \div 1000$ at once)
- each node may be floodfill (if allowed and has sufficient resources)

The Network database

Each node generates:

- encryption key
- garlic end-to-end encryption key
- signing key
- everything is signed into 516+ byte cert

Management:

- distributed netDB
- by *floodfill* routers
- $\sim\ 20‘000 \div 30‘000$ ($\sim 600 \div 1000$ at once)
- each node may be floodfill (if allowed and has sufficient resources)

The Addressing Scheme

b32:

- SHA256 (cert(pub keys))
- equivalent of the IP in clearnet
- each node may have many b32's
- base64-encoding:

  nrbnshsndzb6homcipymkkngngw4s6twediqottzqdfyvrvjw3pq.b32.i2p

.i2p:

- covenient name, e.g.: *i2pwiki.i2p*
- addressbook based mapping
- persistent storage
- multiple sources:
  - inr.i2p
  - stats.i2p
- address helpers available

The Addressing Scheme

b32:

- SHA256 (cert(pub keys))
- equivalent of the IP in clearnet
- each node may have many b32's
- base64-encoding:

  nrbnshsndzb6homcipymkkngngw4s6twediqottzqdfyvrvjw3pq.b32.i2p

.i2p:

- covenient name, e.g.: *i2pwiki.i2p*
- addressbook based mapping
- persistent storage
- multiple sources:
  - inr.i2p
  - stats.i2p
- address helpers available

Bootstrapping

b32:

- one I2P node IP required
- or fresh netDB part
- usually src URI is hardcoded in package
- can be fetched manually

.i2p:

- address book may be shipped with package
- subscriptions often included with package
- can be linked or fetched manually

Bootstrapping

b32:

- one I2P node IP required
- or fresh netDB part
- usually src URI is hardcoded in package
- can be fetched manually

.i2p:

- address book may be shipped with package
- subscriptions often included with package
- can be linked or fetched manually

Cryptography

Symmetric:

- AES-256

Asymmetric encryption:

- Elgamal-2048

Hash:

- SHA-256

All the above possible to change, but problems with backward compatibility.

Cryptography: signatures

1. DSA-SHA1 *[obsolete]*
2. ECDSA-SHA256-P256
3. ECDSA-SHA384-P384
4. ECDSA-SHA512-P521
5. RSA-SHA256-2048
6. RSA-SHA384-3072
7. RSA-SHA512-4096
8. EdDSA-SHA512-Ed25519 *[popular]*
9. EdDSA-SHA512-Ed25519ph *[popular]*

10. GOSTR3410-GOSTR3411-256-CRYPTO-PRO-A
11. GOSTR3410-GOSTR3411-512-TC26-A

} i2pd

Implementations

i2p [11]:

- original implementation
- in java
- up to 2 – 5 GB RAM

i2pd [12]:

- full implementation in C++ (w/o https proxy)
- 150 – 350 MB RAM
- $\sim 20 - 50\%$ less CPU usage
- works on Raspberry PI [13]

other forks: kovri [14], etc...

Implementations

i2p [11]:

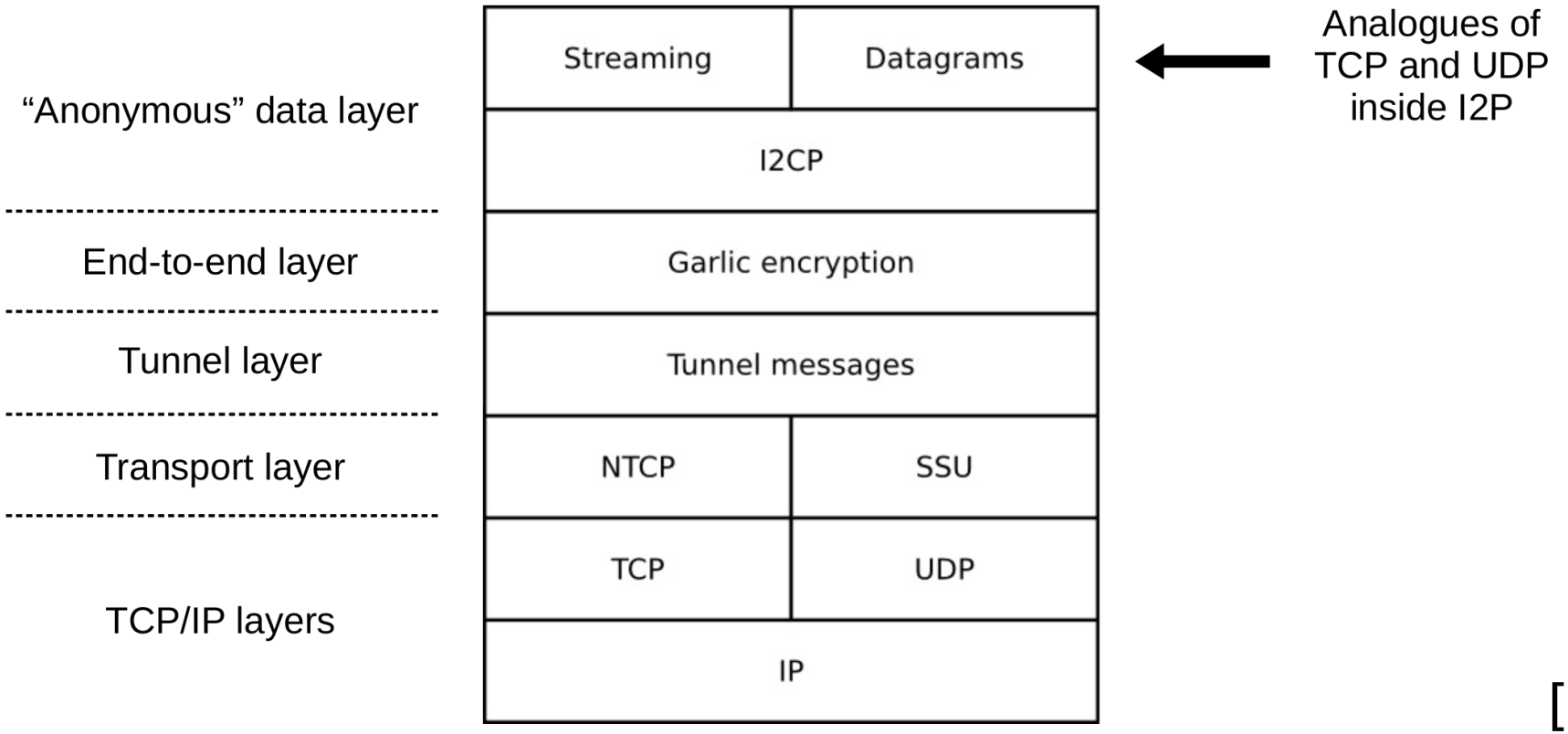
- original implementation
- in java
- up to 2 – 5 GB RAM

i2pd [12]:

- full implementation in C++ (w/o https proxy)
- 150 – 350 MB RAM
- $\sim 20 - 50\%$ less CPU usage
- works on Raspberry PI [13]

other forks: kovri [14], etc...

The I2P Protocols



[10]

- SOCKS and http(s) proxies for the I2P layer are provided
- Control protocols allow fine tunnel control

Usage

Some resources:

- official I2P page [15], wiki [16, 17], search [18]
- messengers: IRC [19], Jabber [20]
- social networks [21, 22]
- torrents [23, 24, 25]

Software:

- decentralized forums: Syndie [26]
- torrents: transmission-i2p [27]
- distributed network file system: Tahoe-LAFS [28]
- crypto currencies: anoncoin [29], monero [30, 14]

Usage

Some resources:

- official I2P page [15], wiki [16, 17], search [18]
- messengers: IRC [19], Jabber [20]
- social networks [21, 22]
- torrents [23, 24, 25]

Software:

- decentralized forums: Syndie [26]
- torrents: transmission-i2p [27]
- distributed network file system: Tahoe-LAFS [28]
- crypto currencies: anoncoin [29], monero [30, 14]

Use case: SSH

- many inbound tunnels => no problems with NAT
- set `UseDNS = no` in sshd.conf
- in tunnels.conf:

```
[ssh]
type = server
host = 127.0.0.1
port = 2222
keys = ssh.dat
```

- connect:

```
torsocks -P 4447 ssh name.b32.i2p
```

Use case: VPN

- server, tunnels.conf:

```
[openvpn]
type = server
host = 127.0.0.1
port = 1194
keys = vpn.dat
accesslist = b32addr1, b32addr2
```

- client, openvpn.conf:

```
socks-proxy 127.0.0.1 4447
remote name.b32.i2p
```

Security

I2P Threat analysis:

- thourought analysis [31] and numerous publications are available [32]
- most threats are partially or fully minigated

The weakest part is **user**

- user *fingerprinting*:
  - browsers are terrible problem: too many complex and leaking technologies
  - check yourself at [33, 34]
- application level leaks

Security

I2P Threat analysis:

- thourought analysis [31] and numerous publications are available [32]
- most threats are partially or fully minigated

The weakest part is **user**

- user *fingerprinting*:
  - browsers are terrible problem: too many complex and leaking technologies
  - check yourself at [33, 34]
- application level leaks

Security: patterns

Insecure / deanonimyzing:

- using the same browser for clearnet, tor and i2p
- including QuickProxy, FoxyProxy, privoxy (with multiple upstreams)
- webrtc [35]
- javascript, flash, plugins,...

Secure:

- dedicated browser, container / vm
- security-oriented software (e.g torbrowser)
- simple/robust (lynx, elinks)

Security: patterns

Insecure / deanonimyzing:

- using the same browser for clearnet, tor and i2p
- including QuickProxy, FoxyProxy, privoxy (with multiple upstreams)
- webrtc [35]
- javascript, flash, plugins,...

Secure:

- dedicated browser, container / vm
- security-oriented software (e.g torbrowser)
- simple/robust (lynx, elinks)

Summary

- Use it, setup routers [11]
- Be *careful* and wise
- Contribute and develop

Thank you for your attention!

Bibliography I


Tor relays and bridges stats. — https://metrics.torproject.org/networksize.html.


Tor users stats. — https://metrics.torproject.org/userstats-relay-country.html.


Tor authority (directory) servers. — https://atlas.torproject.org/#search/flag:authority.


The I2P Documentation. — https://geti2p.net/en/docs.


Grigg Jack. Replacing Weary Crypto: Upgrading the I2P network with stronger primitives. — https://download.i2p2.de/media/rwc/2016/rwc2016-str4d-slides.pdf.


I2P – Wikipedia. — https://en.wikipedia.org/wiki/I2P.


Timpanaro Juan Pablo, Chrisment Isabelle, Festor Olivier. Monitoring The I2P Network. — https://www.freehaven.net/anonbib/cache/timpanaro:inria-00632259.pdf.

Bibliography II


Tor authority (directory) servers. — https://www.torproject.org/about/overview.html.en.


Tor Onion Illustration. — https://commons.wikimedia.org/w/index.php?curid=4567044.


Grigg Jack. Onions and Garlic: the protocols of I2P. — http://str4d.i2p/talks/2016-uww-i2p-slides.pdf.


The Invisible Internet Project. — https://geti2p.net/en/.


The I2P Daemon. — https://github.com/PurpleI2P/i2pd.


Cross-Compile static I2PD for Raspberry Pi. — https://i2p.rocks/blog/cross-compile-static-i2pd-for-raspberry-pi.html.


Kovri. — https://getkovri.org/.


I2P in I2P. — http://i2p2.i2p.

Bibliography III


I2P Wiki. — http://i2pwiki.i2p.


Another I2P Wiki. — http://ugha.i2p.


I2P Search Engine. — http://seeker.i2p.


I2P IRC. — http://irc.postman.i2p.


I2P Jabber. — http://i2jabber.i2p/en.


Onelon social network. — http://onelon.i2p.


Lifebox social network. — http://lifebox.i2p.


Torrent tracker (only). — http://magnets.i2p.


Torrent finder. — http://torrentfinder.i2p.

Bibliography IV


Hiddent torrents. —
http://ptt.i2p.


Decentralized Syndie forums. —
https://www.syndie.de/.


Anonymous torrent client Transmission-I2P. —
https://github.com/l-n-s/transmission-i2p.


Tahoe-LAFS in I2P. —
http://killyourtv.i2p/tahoe-lafs/.


Anoncoin. —
https://anoncoin.net/.


Monero. —
https://getmonero.org.


I2P's Threat Model. —
https://geti2p.net/en/docs/how/threat-model.


I2P Bibliography. —
https://geti2p.net/en/papers/.


EFF fingerprinting checker. —
https://panopticlick.eff.org/.

Bibliography V


Another fingerprinting checker. — https://amiunique.org/.


WebRTC Leak Test. — https://browserleaks.com/webrtc.

Arpanet Map (1977)