**SECURECOMM**

26 September 2014

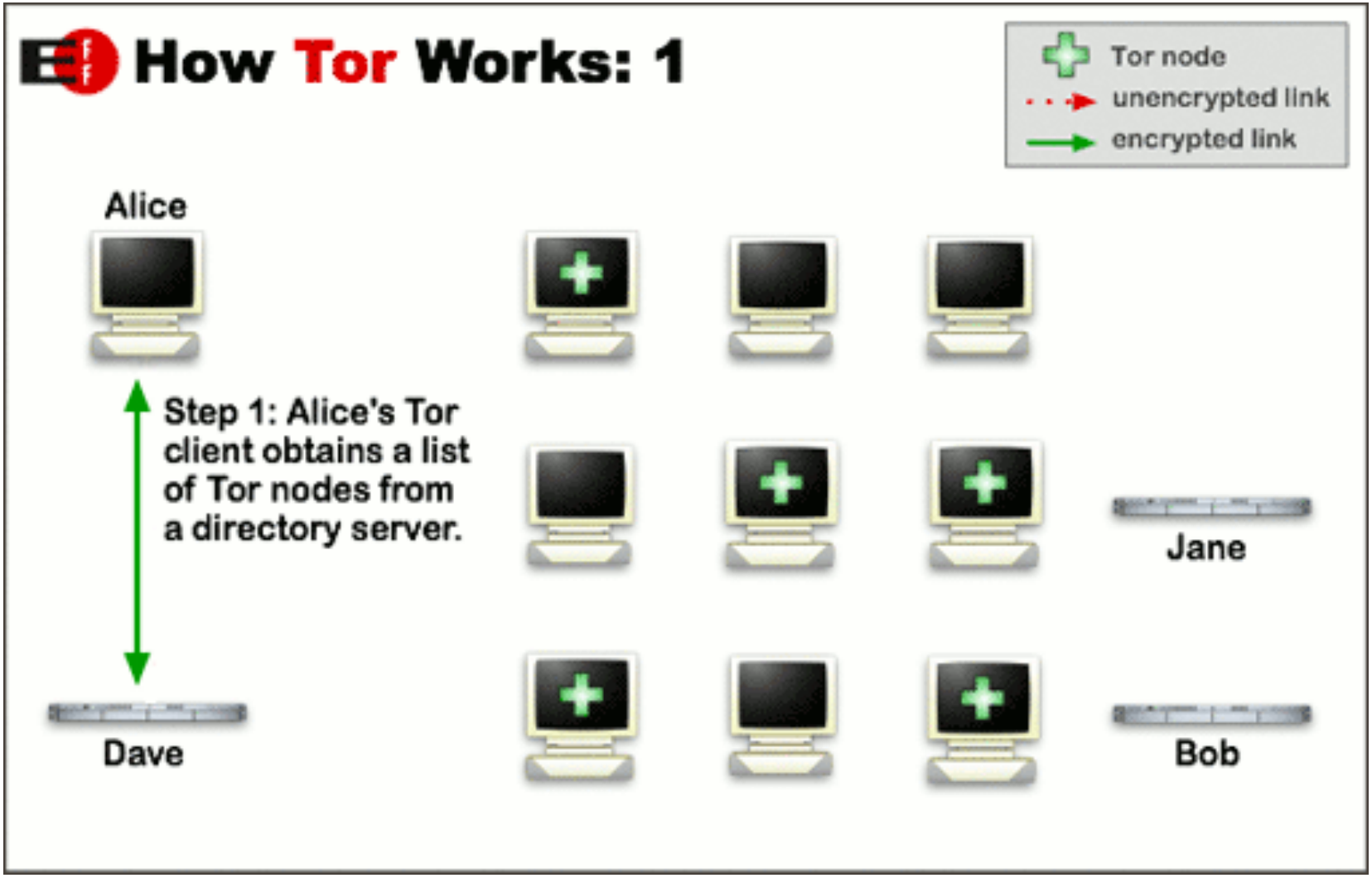
Beijing Yulong International Hotel, Beijing, China

# JumpBox – A Seamless Browser Proxy for Tor Pluggable Transports

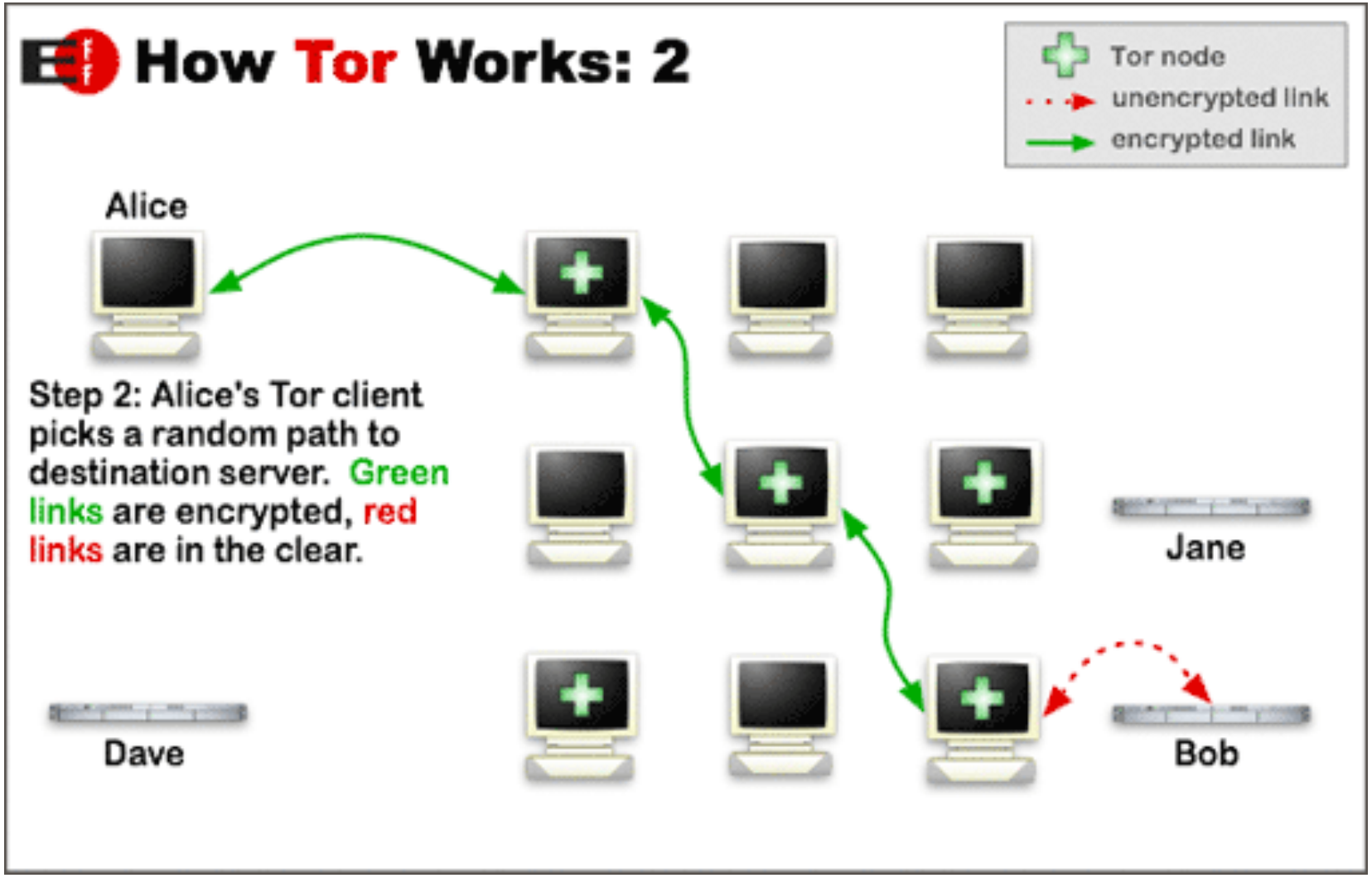
Jeroen Massar, Farsight Security, Inc.
massar@fsi.io

Ian Mason, Linda Briesemeister, Vinod Yegneswaran, SRI International
{iam,linda,vinod}@csl.sri.com

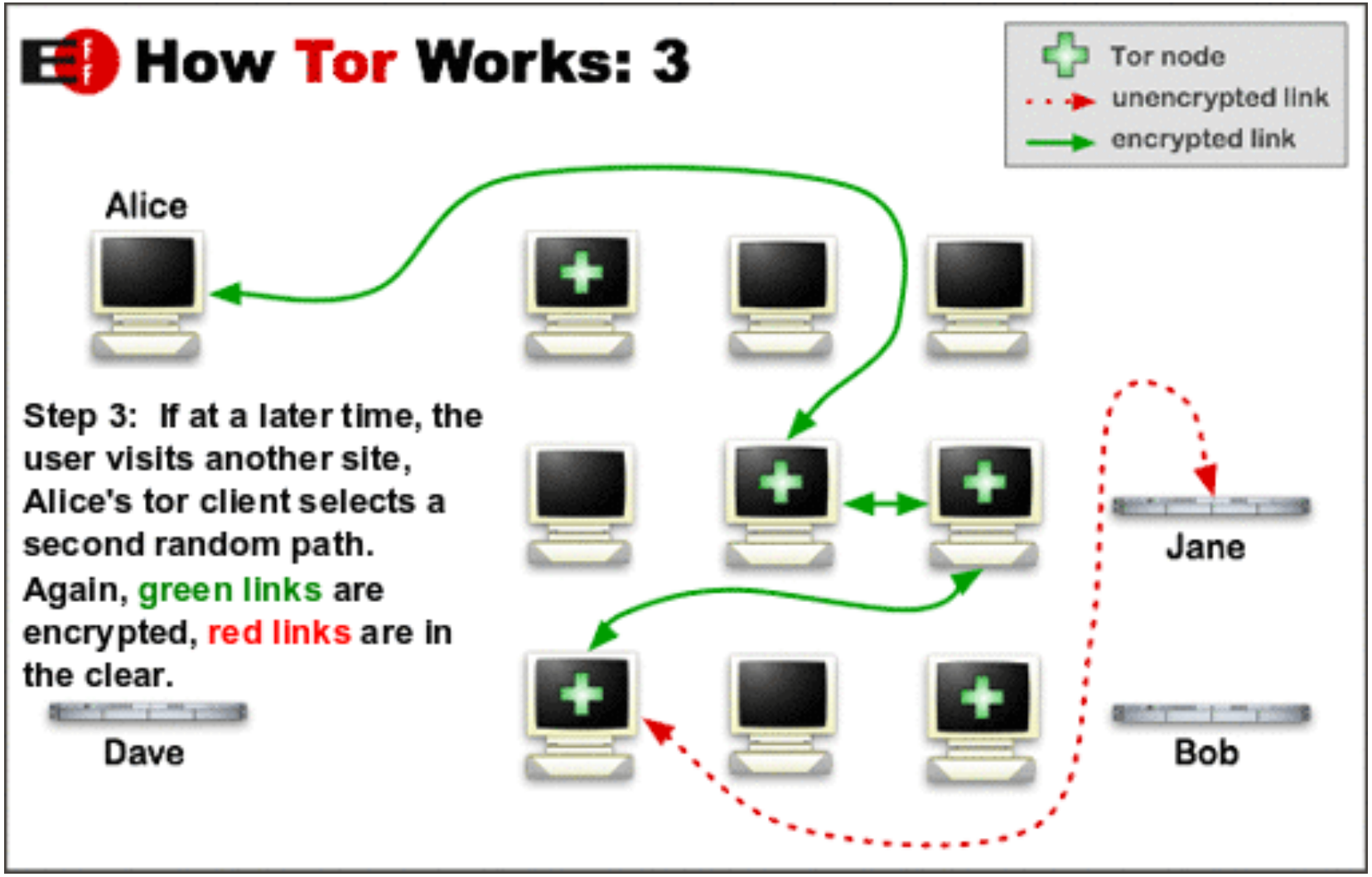
# How Tor Works 1/3



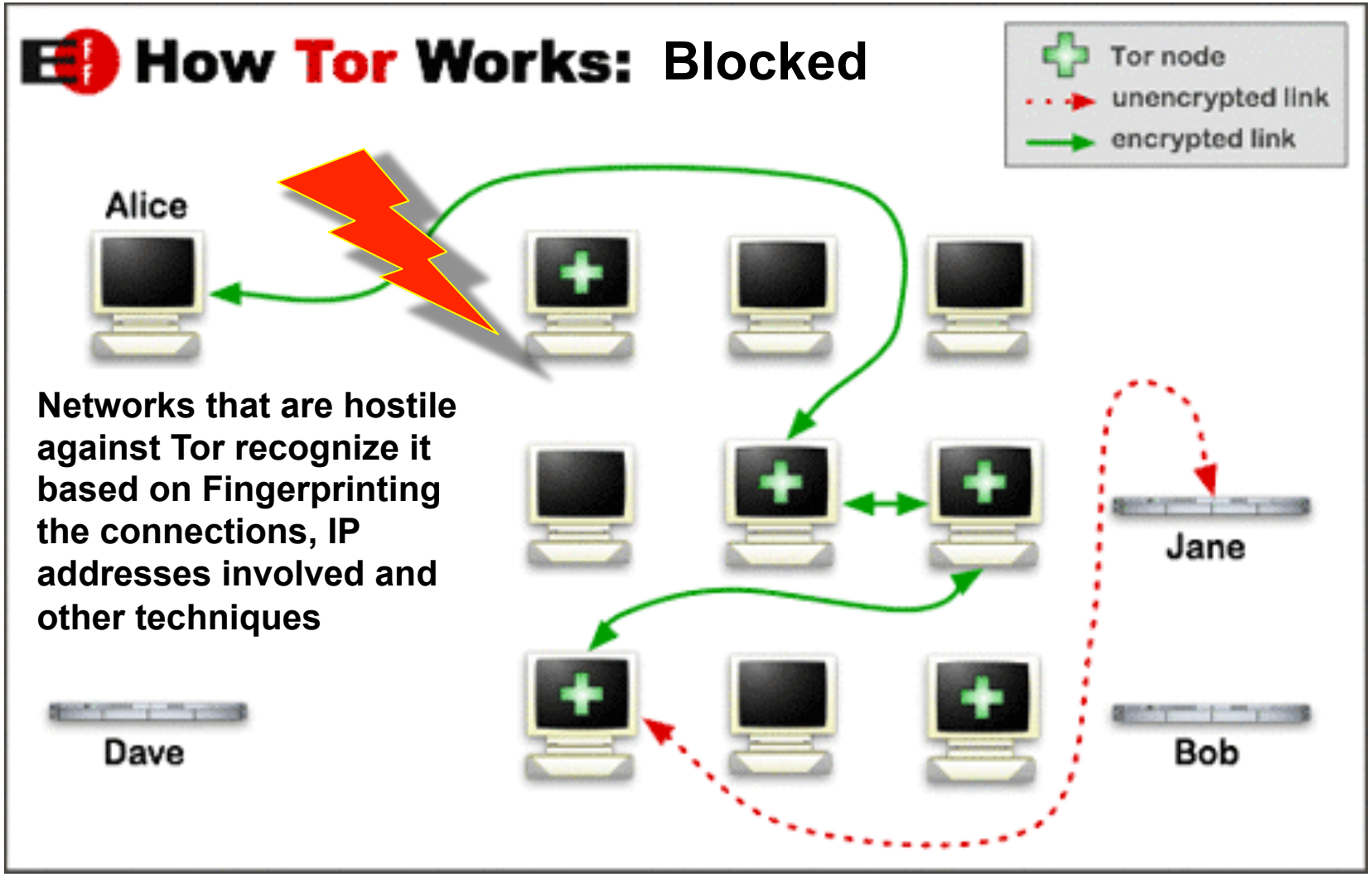
# How Tor Works: 2/3



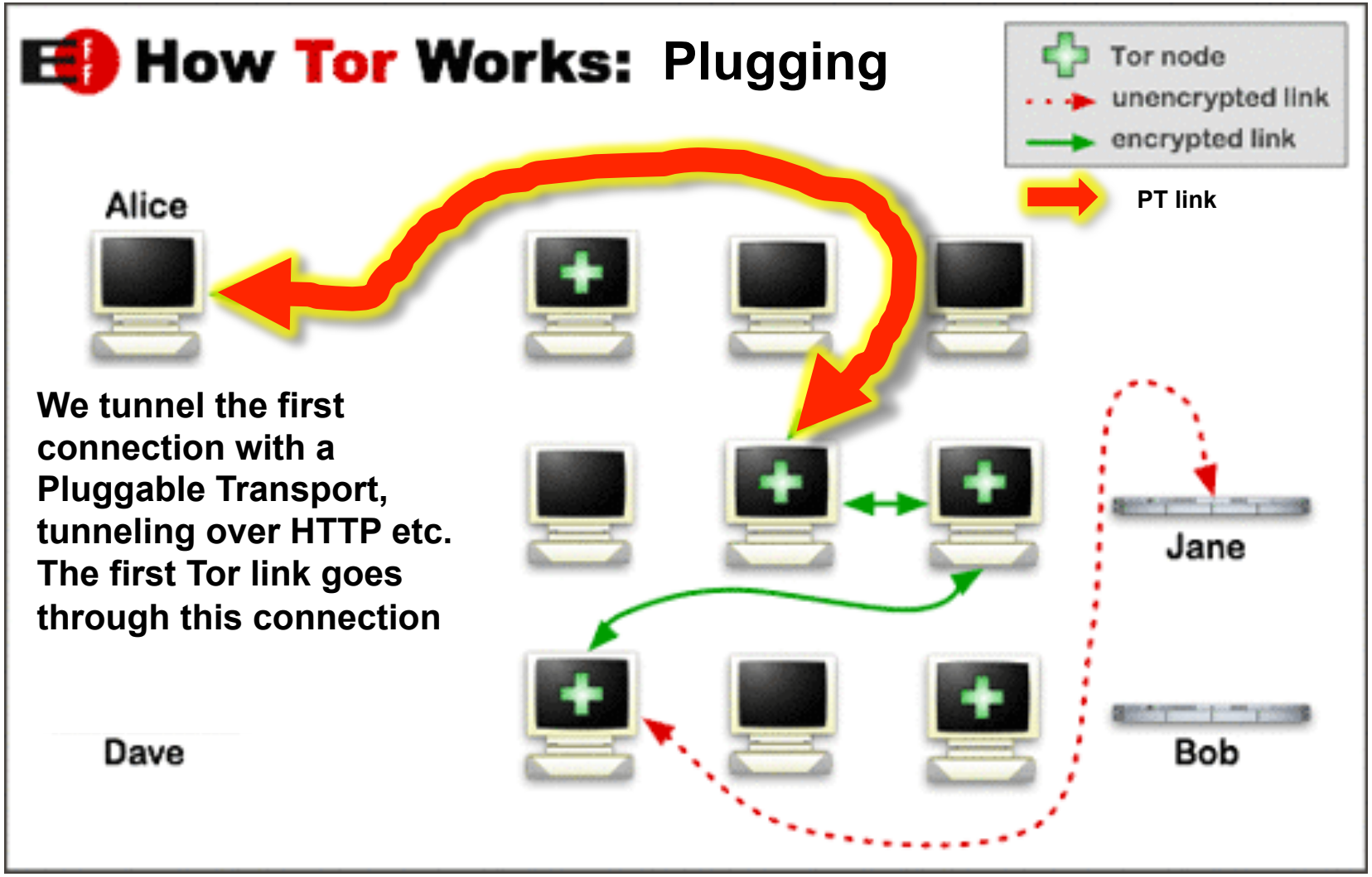
# How Tor Works: 3/3



# Where Tor gets blocked



# Pluggable Transports



# Many Pluggable Transports

- Obfsproxy
  - Obfuscates bits of the Tor connection avoiding fingerprinting
- StegoTorus
  - Tunnels over multiple HTTP connections using JPEG/HTML/ JSON/PDF to multiple servers that merge the packets back.
- FlashProxy
  - Uses Flash to create proxies in browsers creating random proxies
- Format Transforming Encryption (FTE)
  - Fakes protocols by implementing them minimally using algorithms
- Meek
  - Tunnels over HTTPS uses SSL cert from public Cloud Services

More: https://www.torproject.org/docs/pluggable-transports.html.en

# Related circumvention systems

Non-Pluggable-Transport systems:

- VPNs (OpenVPN, SSH tunnels, PPTP, AYIYA etc)
- Telex, Decoy Routing and Cirripede

  Use signaling inside TCP headers to circumvent filtering systems by routing them over alternate paths.
- Infranet

  Provides a tunnel over HTTP
- Collage

  Uses steganography for hiding communications inside images and other content hosted on standard webforums.

# The Parrot is Dead

"The Parrot is Dead:
Observing Unobservable Network Communications"
https://www.cs.utexas.edu/~shmat/shmat_oak13parrot.pdf

Creating an own HTTP engine means that you will never match 100% what an actual browser would do.

# Becoming a Mockingbird

Our Design Goals for JumpBox:

1. Be the browser

   Don’t emulate, but use a real browser

2. Extensibility

   Make it possible to easily extend the concept

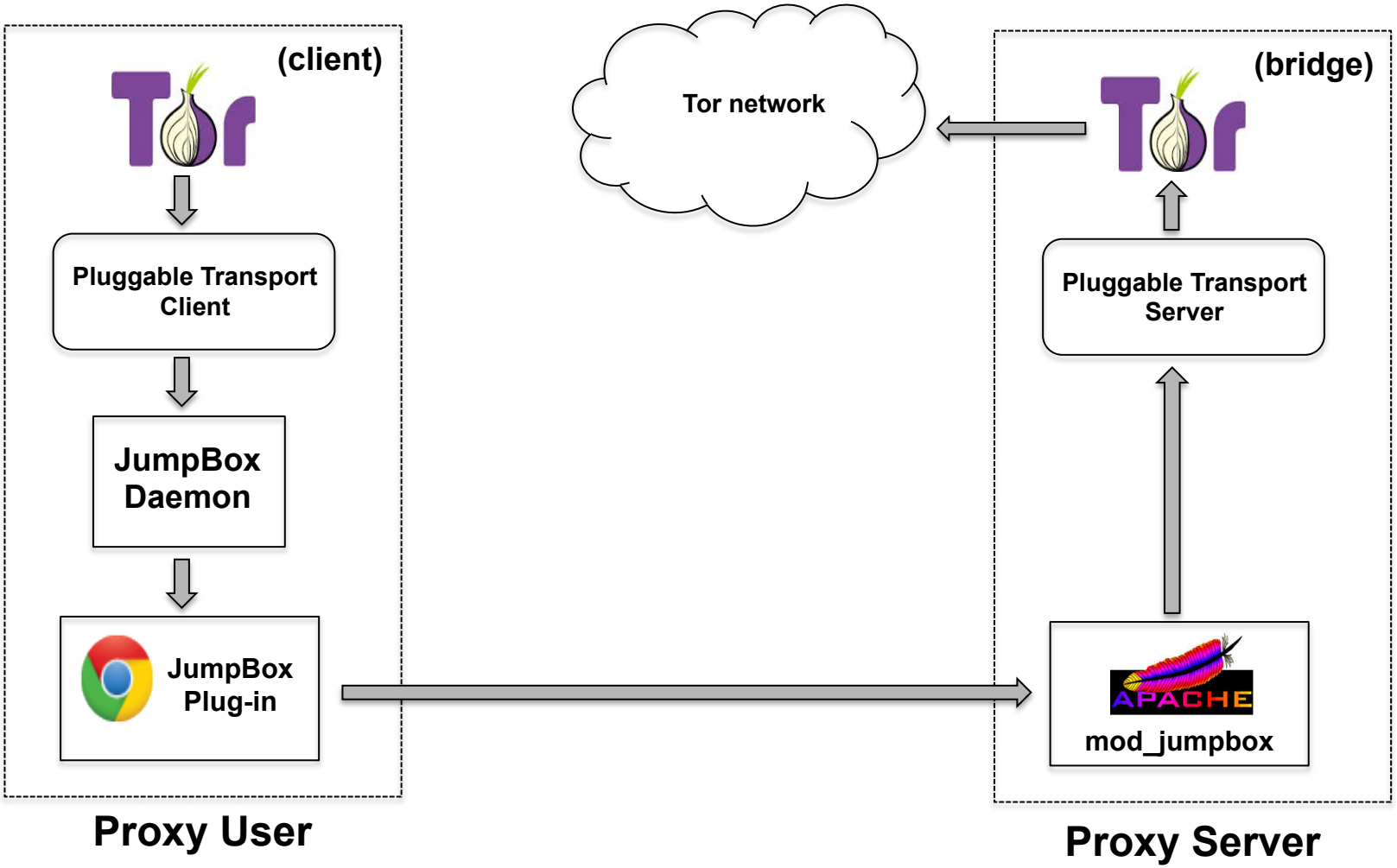
3. Seamless integration

   No code changes required to either browser or PT

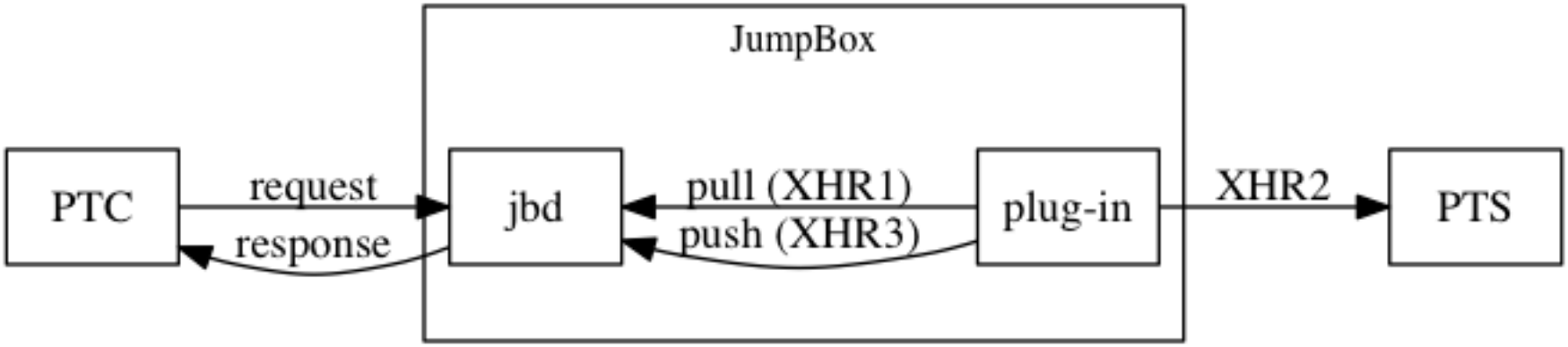
4. Minimal overhead

   The system should not impact performance

# JumpBox Design



# Jumping the Box




- The browser plugin keeps on asking for new requests with XHR1.
- When it gets one, it sends it to the PTS with XHR2
- The result from the PTS is answered with XHR3

# FTEProxy through JumpBox

## Normal FTEproxy Request:

```
C: GET /GPcoEIlMxXBh...<base64-encoded-bytes>...LoQas HTTP/1.1
S: HTTP/1.1 200 OK
S: Content-Type: H
S:
S: ....<binary bytes>…
```

## FTEProxy through JumpBox:

```
C: GET /Id5UdpnNYFB...<base64-encoded-bytes>...160VG HTTP/1.1
C: Host: example.com
C: User-Agent:
C: Connection: keep-alive
C: Accept: text/html,...,application/xml;q=0.9,image/webp,*/*;q=0.8
C: User-Agent: Mozilla/5.0 ... Chrome/34.0.1833.5 Safari/537.36
C: Referer: http://www.example.com/
C: Accept-Encoding: gzip,deflate,sdch
C: Accept-Language: en-US;q=0.8,en;q=0.2,de;q=0.2
S: HTTP/1.1 200 OK
S: Date: Thu, 01 Feb 2013 09:01:28 GMT
S: Server: Apache
S: Accept-Ranges: bytes
S: Content-Length: 2529
S: Keep-Alive: timeout=5, max=100
S: Connection: Keep-Alive
S: Content-Type: application/octets
S: Content-Language: en-GB
S:
S: ....<binary bytes>...
```

# Solves HTTP mimicry problems

- HTTP header inconsistencies
  - Case, wrong/missing CRLF, extra chars, wrong parsing
- HTTP URI encodings
  - Hex encoding, double hex encoding, %u encoding etc
- HTTP content encodings
  - gzip, chunked encodings etc
- Timing attacks
  - Very difficult on the layer-7, especially as one does not know the speed of the client's Internet connection

JumpBox uses a standard Browser (Chrome) and a normal Server (Apache), hence they handle these issues for us.

# Attacks that are not solved

- Replay attacks
  - Replaying HTTP requests and checking for consistent results; JumpBox does not solve this, a cache could, or the PT has to handle it.
- Content-Injection attacks
  - We require the PT to detect and handle these issues.
- Content-Rendering attacks
  - JumpBox does not render content, hence we also do not fetch any images/CSS/javascript that have been added by an adversary and that they require to be run to detect that the user is real.
  - Due to caching possibilities and or NoScript should not be a big concern.

# HTTPS

- JumpBox cannot protect against rogue CAs
  - There are various projects that attempt to solve this:

    HTTPS Everywhere uses data from EFF's SSL Observatory
- The browser can be configured for HSTS and Certificate Pinning to solve a part of these issues.

# Active Probing

- JumpBox does not protect against broken requests.
  - Some HTTP parsing libraries just search for a \n in the HTTP headers and see that as a line ending, while officially it is CRLF (\r\n)
  - Some HTTP parsing libraries search for the name of the header, but ignore to check if it is starting at the beginning of the line

    "Accept-Encoding" matches "X-Accept-Encoding: something"

  These are PT issues though, that they need to resolve by implementing these portions properly.
- JumpBox only supports GET, HEAD and POST requests, others receive a HTTP 405 from Apache.
- Only supports MIME types that AJAX supports.

# Questions?

**Jeroen Massar**

massar@fsi.io

https://www.farsightsecurity.com/

# Acknowledgements

We like to thank: Drew Dean, Roger Dingledine, Mike Lynn, Dodge Mumford and Paul Vixie for their contributions and insights to this project.