COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Anonymous Communication and Traffic Analysis III: Onion Routing

Information Privacy with Applications

David Sidi (dsidi@email.arizona.edu)

Warm-up

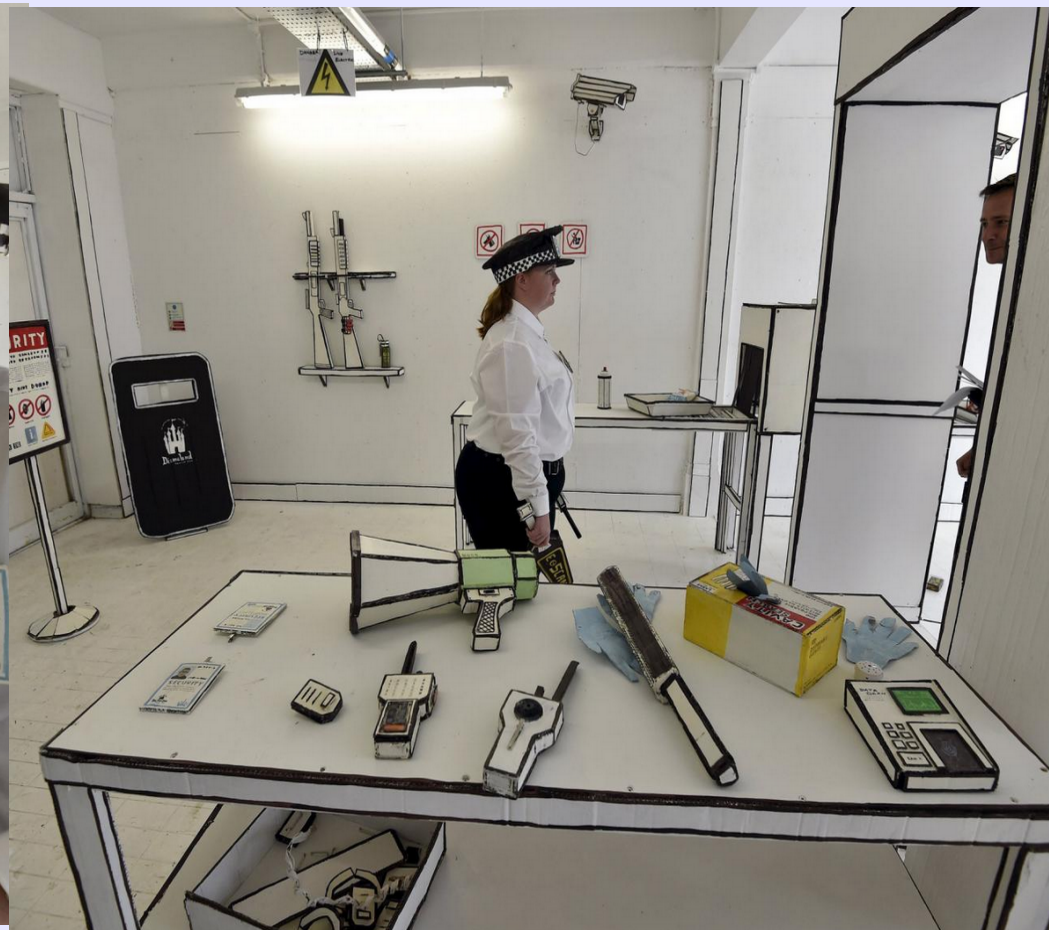
- What would an adversary interested in a man-in-the-middle attack on a large number of TLS connections need to do?

Warm-up

- What would an adversary interested in a man-in-the-middle attack on a large number of TLS connections need to do?
- Relevant:
  - Mozilla Root Certificate Program (used by Chrome as well as Firefox)
  - mozilla.dev.security.policy: where the action happens

Small mention of interesting things

- Dismal land bemusement park (link)



ww.independent.co.uk/arts-entertainment/art/news/dismaland-banksys-bemusement-park-is-deeply-unsettling-but-bizarrely-entertaining-

Continuing last time

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Anonymity networks

CC-SA License by David Sidi

Anonymity networks address the traffic analysis problem

- Chaum: “Keeping confidential who converses with whom, and when they converse”
- Contrast with secrecy of message content

CC-SA License by David Sidi

Anonymity networks can involve trusted or semi-trusted relays

- Trusted parties are not adversaries: they can break anonymity
- Semi-trusted parties don’t all collude

CC-SA License by David Sidi

Trusted relays

- Example: Nym servers
  - a server keeps a dictionary between real and pseudonymous emails
  - request comes to the remailer, which forwards it, gets the response, and returns it to the user
  - Example: anon.penet.fi
- Other Examples: Anonymous proxies (startpage.com), VPNs

CC-SA License by David Sidi

Trusted relays

- Problem: messages are all linked
  - Stylometric attacks: the frequency of function words in the English language can be used in the long term to identify users (Rao & Rohatgi (2000), “Can Pseudonymity Really Guarantee Privacy?”)
  - Correspondent sets of each nym
- Anonymity is compromised if one node is compromised. (“Single point of failure.”)
  - lots of incentive to coerce
  - or if the node is not honest
- Fails bitwise indistinguishability: sometimes traffic analysis can deanonymize
  - http proxy example
  - timing correlation

CC-SA License by David Sidi

Semi-trusted relays

- Strengths
  - Compromise of more than one is needed, so more coercion resistant than trusted-relay approaches
  - “any single mix is able to provide the secrecy of the correspondence between the input and the outputs of the entire cascade” (Chaum)
- Weaknesses
  - Tagging attacks violate unlinkability (blind signing attack)
  - replay attacks
  - slow (public-key cryptography)

CC-SA License by David Sidi

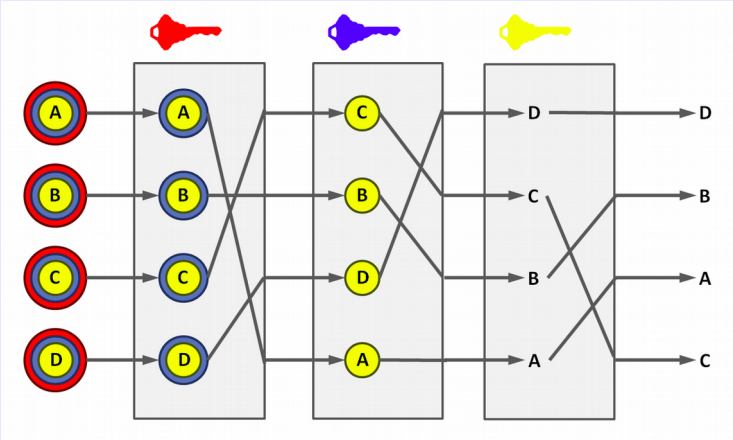
Semi-trusted relays

- What are the problems with a mixnet with only one node? (Chaum)

CC-SA License by David Sidi

Mixnets are anonymity systems with semi-trusted relays

- Routing protocol with a cascade of cryptographic relays called ‘mixes’
- Mixes only know their neighbors
- User-specifiable routing (Chaum’s “new kind of mix”)



wikipedia.org

CC-SA License by David Sidi

Mix-nets are anonymity systems with semi-trusted relays

- Suppose we are at a mix A1, which receives message m.
- m is split into a fixed number blocks, ℓ

$$A_1\colon [K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow.$$

$$A_2\colon [K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow,$$

$$A\colon [M_1], [M_2], \ldots, [M_{l-n}],$$
$$[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})].$$

CC-SA License by David Sidi

Mix-nets are anonymity systems with semi-trusted relays

- The first block is like a header: it contains the key R_{A1} and address A2 for the next hop. This is stripped off of the message, and a padding ("junk") block is added to the end

$$A_1\colon [K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow.$$

$$A_2\colon [K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow,$$

$$A\colon [M_1], [M_2], \ldots, [M_{l-n}],$$
$$[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})].$$

CC-SA License by David Sidi

-

Mix-nets are anonymity systems with semi-trusted relays

- The rest of the blocks are, first, the header blocks for all remaining routers in the cascade, and next, the message. All of these are encoded using .

$$A_1\colon [K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow.$$

$$A_2\colon [K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow,$$

$$A\colon [M_1], [M_2], \ldots, [M_{l-n}],$$
$$[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})].$$

CC-SA License by David Sidi

Mix-nets are anonymity systems with semi-trusted relays

- A1 uses the R_{A1} it now has to decode the (ℓ-1) blocks after the header in the original message: these are the first part of the message sent out from A1, they contain the headers for A2, the encoded headers for A3,...An, and then the encoded message
- The blocks are passed to the next node, which could be another mix

$$A_1\colon [K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow.$$

$$A_2\colon [K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow,$$

$$A\colon [M_1], [M_2], \ldots, [M_{l-n}],$$
$$[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})].$$

CC-SA License by David Sidi

Mix-nets are anonymity systems with semi-trusted relays

- Mixes only know their neighbors. (Question: Why?)
- All nodes have a public key
- Weaknesses
  - active attacks: tagging attacks (blind signing attack), replay attacks
  - slow (public-key cryptography, latency in anonymous remailers).

$$A_1: [K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots,$$
$$[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow.$$

$$A_2: [K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)],$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots,$$
$$[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow,$$

$$A: [M_1], [M_2], \ldots, [M_{l-n}],$$
$$[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})].$$

CC-SA License by David Sidi

Mixing techniques for Mixnets

- Cascading: All nodes are always used, in the same order
- Scalability is a problem, requires setting up a fixed route with all nodes
- Only requires one honest node to preserve anonymity

CC-SA License by David Sidi

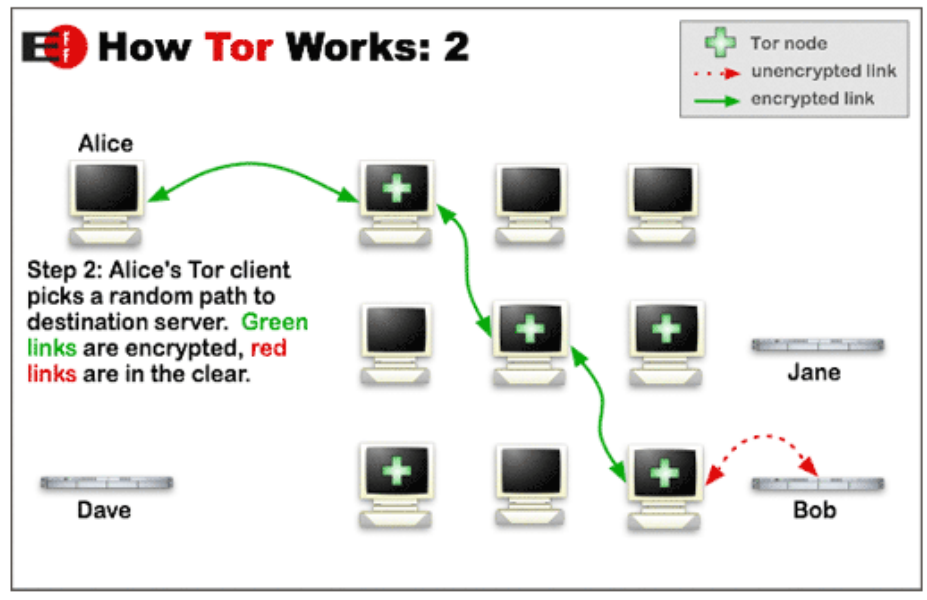
Mixing techniques for Mixnets

- User specified: user arbitrarily picks its route through the network
- Scalable, does not require initial configuration of a route
- Not anonymous if only one node is honest (nodes can figure out their positions)

CC-SA License by David Sidi

Onion Routing

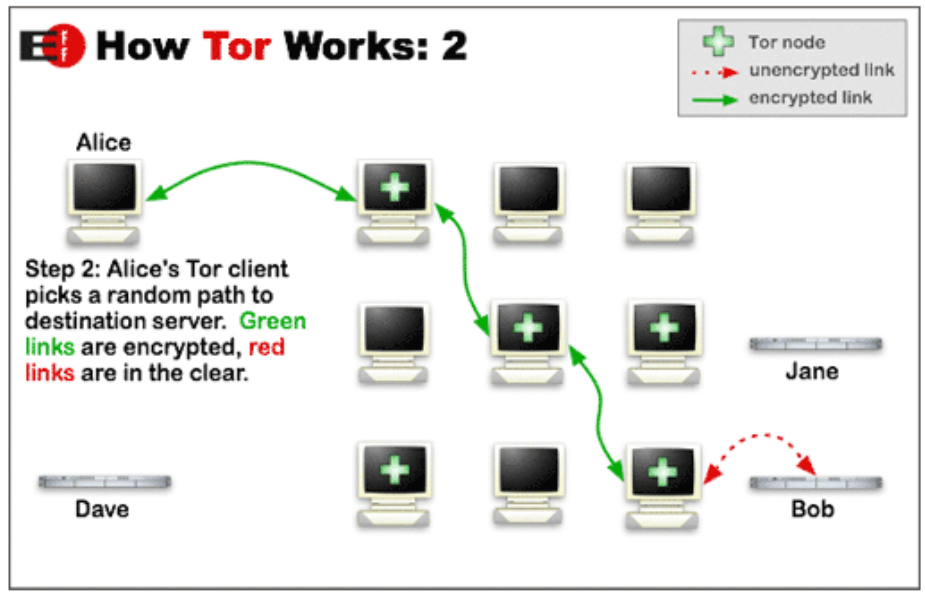
- First was from the US Naval Laboratory, 1996
  - pure peering at this stage, loafers!
- Freedom Network was an independent onion routing network from Zero Knowledge Systems
- Tor is a third-gen. onion routing network



CC-SA License by David Sidi

Tor is an onion routing system

- Example: simplified, slightly out-of-date Tor (*link*)
- Distributed overlay network for TCP-based applications
- Sets up a “virtual circuit” as a cascade of three onion relays (OR) from the initial client onion proxy (OP)
- guard (from “helper nodes”), relay, and exit nodes
  - each node only knows its immediate predecessor and successor



CC-SA License by David Sidi

Tor is an onion routing system

- originally, onion routing systems sent an initial onion message that was “just layers” to set up the circuit; Tor does it in stages (“telescoping”)
- Next hop in the circuit is determined by unwrapping an “extend” relay cell with a symmetric key, which causes the OR to send its own “create” control cell

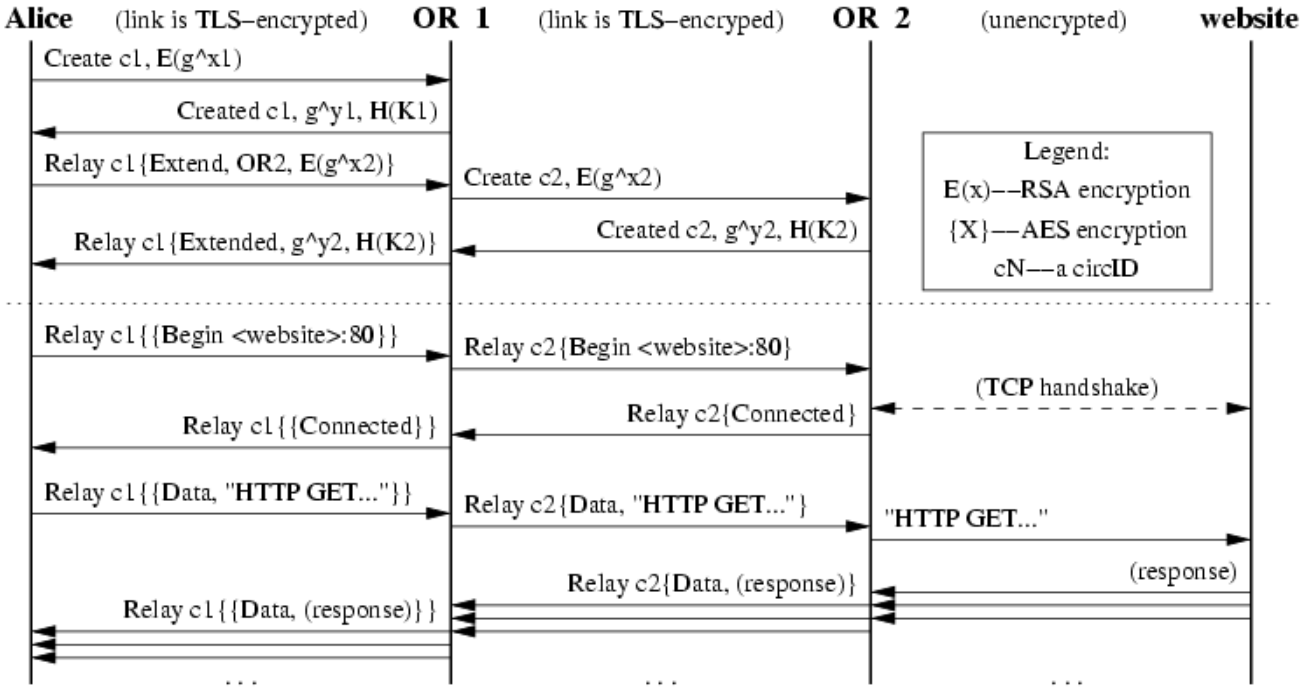


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

OP picks the route

- First picks the exit node E such that E's exit policy includes at least one pending stream that needs a circuit
- Choose N-1 distinct nodes (default is three), with some order
- Open a connection to the first (guard) node, negotiate session keys
- extend the circuit incrementally over the remaining N-1 nodes

CC-SA License by David Sidi

Tor uses PKC to protect negotiation of a session key

- One hop at a time over an encrypted and authenticated channel
  - TLS: use *identity keys* to sign certs
- Use public-key cryptography (PKC) over this channel to set up an ephemeral session key
  - PKC is RSA (legacy) or Curve25519: use short-term *onion keys*
  - symmetric is AES, set up with DHE (legacy) or ECDHE
- Once ephemeral keys are set up OP layers them, and ORs unwrap them

CC-SA License by David Sidi

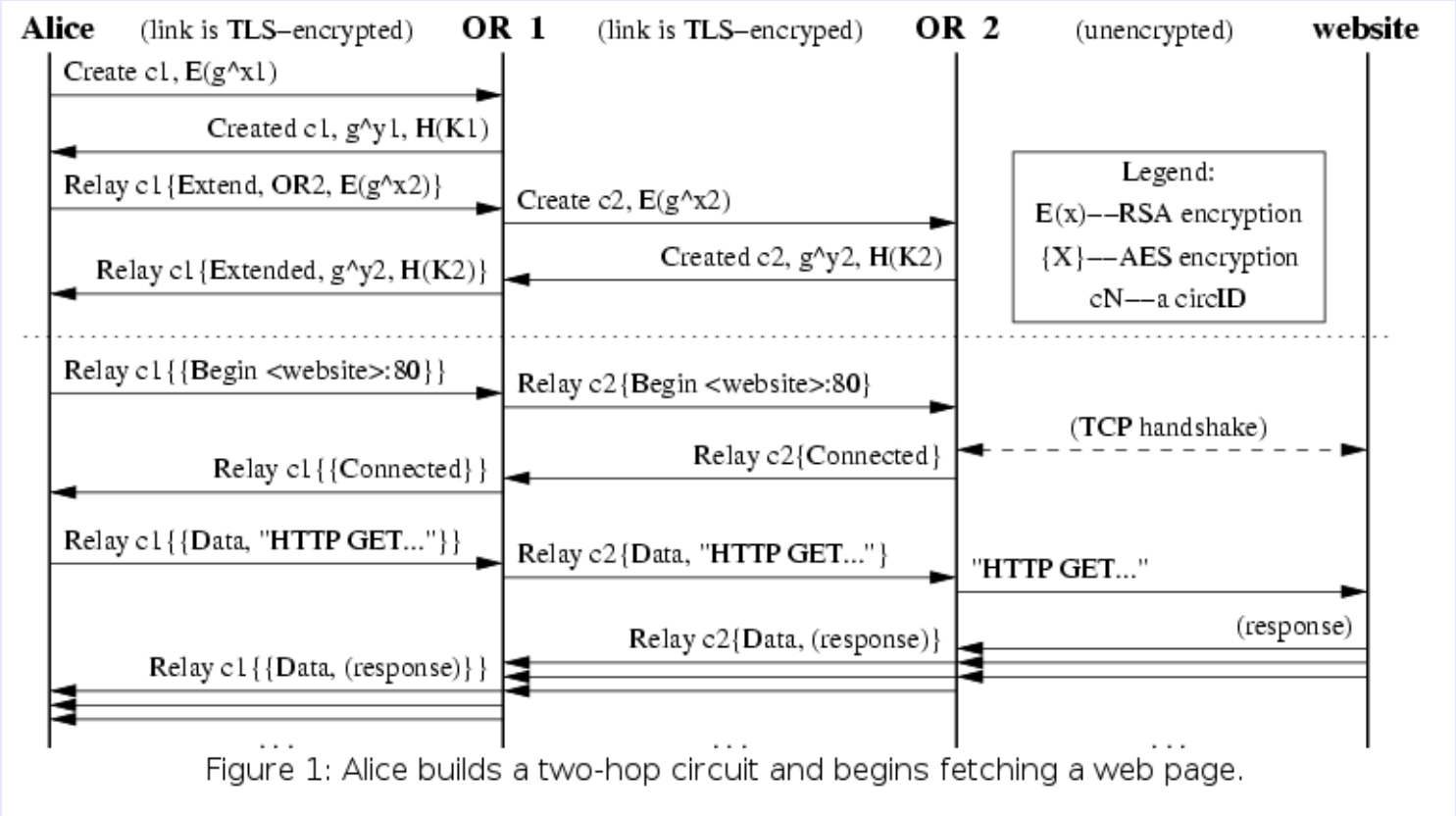


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

CC-SA License by David Sidi

Discussion

- We have a public key for the guard. What reason did I give to not just use PKC to encrypt communications?

CC-SA License by David Sidi

Session keys are negotiated using Diffie-Hellman Key Exchange

- First published in 1976; still around
- Alice and Bob want to share a secret key for use in a symmetric cipher. Every piece of information that they exchange is observed by their adversary Eve. How is it possible for Alice and Bob to agree on a key without making it available to Eve?

CC-SA License by David Sidi

Diffie-Hellman

Publicly choose:

- a safe large prime *p* (e.g. Tor docs use rfc2409 section 6.2. But see Logjam)
- g, a primitive root mod p, with *2 <= g <= p-2*

Secretly generate:

- Alice and Bob randomly choose secret integers 1 <= *x, y <= p-2* respectively

$A := g^x \pmod p$

$K_B := A^y \pmod p$

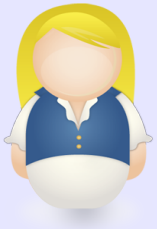
$K_A := B^x \pmod p$

$B := g^y \pmod p$

$$K_A = (g^y)^x = (g^x)^y = K_B \text{ is the key}$$

CC-SA License by David Sidi

Diffie-Hellman



$x = \log_g A \pmod p$

$y = \log_g B \pmod p$

CC-SA License by David Sidi

Diffie-Hellman

$x = \log_g A \pmod p$

$y = \log_g B \pmod p$

Discrete Log Problem is in NP

Diffie Hellman Problem is no harder than DL problem; there is no proof of the converse

Question

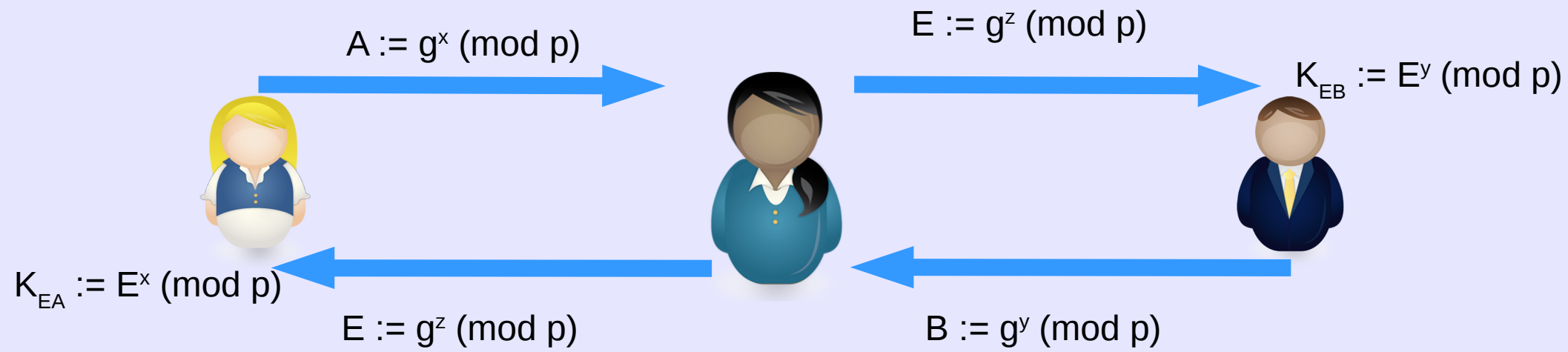
- Ian Goldberg remarked that a good way to fight mass surveillance by a global passive adversary would be to “do a quick Diffie-Hellman” by default when setting up otherwise unprotected connections. He notes that this won’t help against an active attack. Can you guess what he means by an active attack?

MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with *2 <= g <= p-2*

Secretly generate:

- Alice and Bob choose secret integers 0 <= *x, y <= p-2* respectively
- Eve picks her own secret integer, *z*



CC-SA License by David Sidi

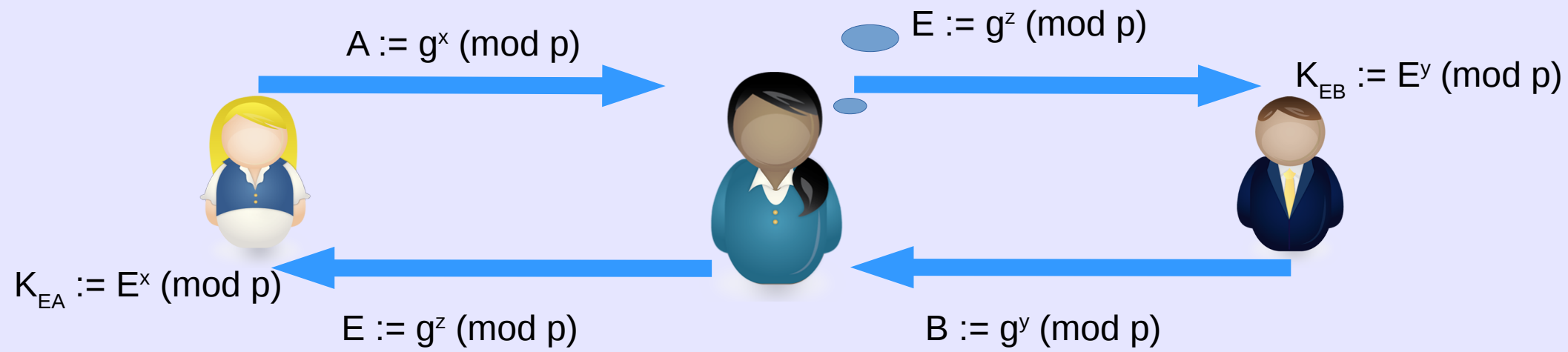
MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with *2 <= g <=*

$K_{EA} := g^{xz} \pmod p$
$K_{EB} := g^{yz} \pmod p$

Secretly generate:

- Alice and Bob choose secret integers 0 <= *x, y <= p-2* respectively
- Eve picks her own secret integer, *z*



CC-SA License by David Sidi

Tor strengths and weaknesses

- Strengths
  - faster than mixnets
  - perfect forward secrecy
  - easy to run nodes, easy to use as a client: adds to security
  - bridges, pluggable transports for censorship circumvention
  - sandboxing
- Weaknesses
  - traffic analysis by a pervasive passive adversary
  - end-to-end timing attacks
  - content is revealed to exit node
  - blockable exit nodes

CC-SA License by David Sidi

Who runs exit nodes?

- Universities (as of Oct 2017) (link)
  - MIT+, Michigan, CMU, UNC, Karlsruhrer IT, Stanford, Clarkson, U. Washington, Utah+, Caltech, RIT+, Bowdoin, Northeastern+, Princeton
- Bad people too! (Why might they do that?)
- Not Arizona :-(
  - yet :-)

CC-SA License by David Sidi

Who runs hidden services?

- Propublica, Duckduckgo, Facebook, Scihub, Riseup, Protonmail, Debian, Whonix, The Intercept, Wikileaks, Securedrops for The Freedom of the Press Foundation , The Guardian, The Associated Press, NY TImes, USA Today, Washington Post, etc., TORCH (these are all onion links)
- A bunch of illegal stuff
- Hidden services are easy to set up (demo)
  - even inside firewalled networks

CC-SA License by David Sidi

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Fun with mitmproxy (demo)

CC-SA License by David Sidi