COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Anonymous Communication

Guest Lecture, ESOC 488 (Kay Mathiesen, Spring 2018)

David Sidi (dsidi@email.arizona.edu)

Me, in one slide

- biometric privacy (smart buildings)
- sensitive surveys (randomized response)
- data broker transparency (information asymmetry in data sharing with online form registrations)
- sender anonymity (advising a project on human trafficking)
- https://u.arizona.edu/~dsidi (see link to the wiki)

Some questions based on your prior understanding of anonymity

- Why is individual anonymity valuable? Consider the effect of the possibility of anonymity on a person's willingness to speak in different circumstances

Some questions based on your prior understanding of anonymity

- Why is it valuable to be able to speak to different groups differently? E.g., to your parents, and to your close friends. How does this relate to anonymity?

Some questions based on your prior understanding of anonymity

- Should anyone be allowed to use technology that guarantees anonymity, no matter how good the justification for breaking it might be (e.g., the “super warrant” required by the Wiretap act)?

The use of ‘anonymity’ in the literature on communications anonymity

Anonymity and Anonymity sets

- `Sender anonymity' is defined with respect to a subset of possible senders, called the anonymity set. (Analogously there is `receiver anonymity, `relationship anonymity')
  - reflects the influence of Pfitzmann et al.
- Think of the anonymity set as answering "who might you be?"

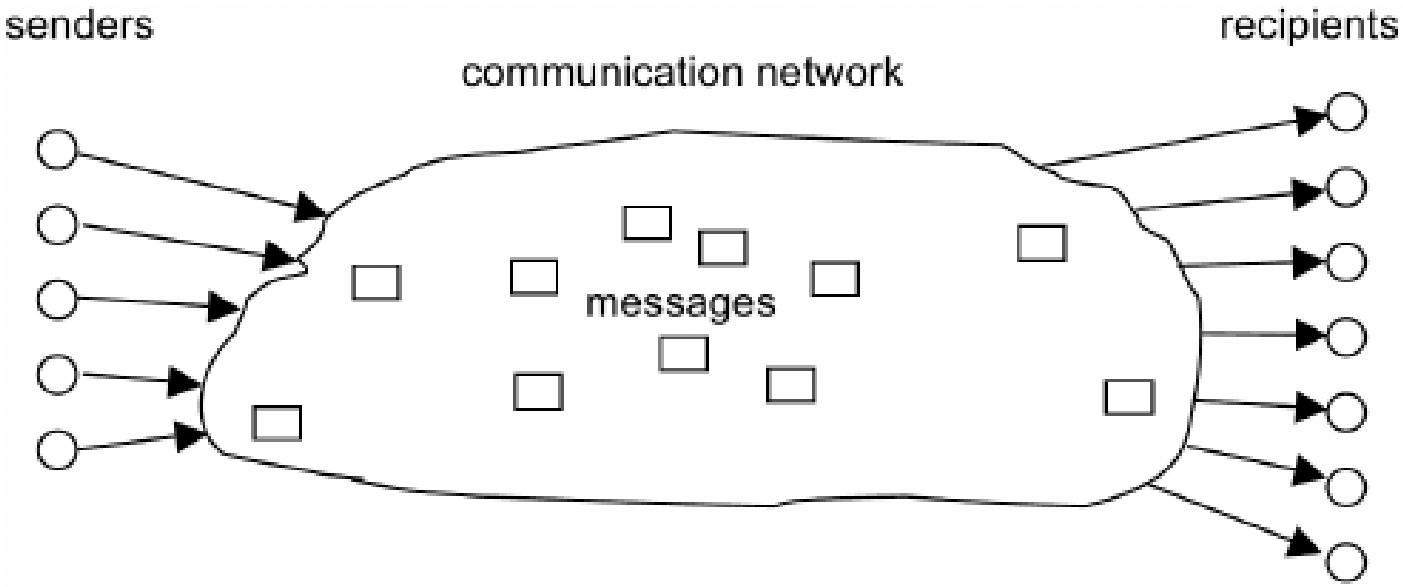


Image credit (before modification):
Christina Pöpper
Ruhr-University Bochum

We gained some useful foundations by discussing the diagram

- Communication, from the attacker’s POV
- Security properties related to traffic analysis
  - unobservability, unlinkability, anonymity, and their relationships
- Attacker model

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Examples of deanonymization: who's an adversary?

DNS

- It seems like you are required by law to say that DNS is “like a phone book:” give it an easy-to-remember name, get an IP address for the server hosting a website
- Forwarding DNS server
- Recursive DNS server (or resolver)
- (Root nameserver)
- (Top Level Domain nameserver)
- Authoritative nameserver

DNS

```
$dig arizona.edu

; <<>> DiG 9.9.5-9+deb8u14-Debian <<>> arizona.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14058
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;arizona.edu.                   IN     A

;; ANSWER SECTION:
arizona.edu.            6813IN     A      128.196.128.233

;; Query time: 32 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Mon Oct 23 12:43:03 MST 2017
;; MSG SIZE  rcvd: 56
```

DNS

- Suppose I use a VPN to tunnel my traffic to a server I control. What can you learn about me from my DNS requests?

DNS

- Suppose I use a VPN to tunnel my traffic to a server I control. What can you learn about me from my DNS requests?
- Many sites to do with local things in Tucson, AZ
- Sites to do with the University of Arizona
- Sites for groups with small memberships (Xerocraft)
- Many hits for a site with a public record attached to one person (sidiprojects.us)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Browser fingerprinting

- UserAgent
- Language
- Color Depth
- Screen Resolution
- Timezone
- Has session storage or not
- Has local storage or not
- Has indexed DB
- Has IE specific 'AddBehavior'
- Has open DB
- CPU class
- Platform
- DoNotTrack or not
- Full list of installed fonts (maintaining their order, which increases the entropy), implemented with Flash.
- A list of installed fonts, detected with JS/CSS (side-channel technique) - can detect up to 500 installed fonts without flash
- Canvas fingerprinting
- WebGL fingerprintingPlugins (IE included)
- Is AdBlock installed or not
- Has the user tampered with its languages 1
- Has the user tampered with its screen resolution 1
- Has the user tampered with its OS 1
- Has the user tampered with its browser 1
- Touch screen detection and capabilities
- Pixel Ratio
- System's total number of logical processors available to the user agent.

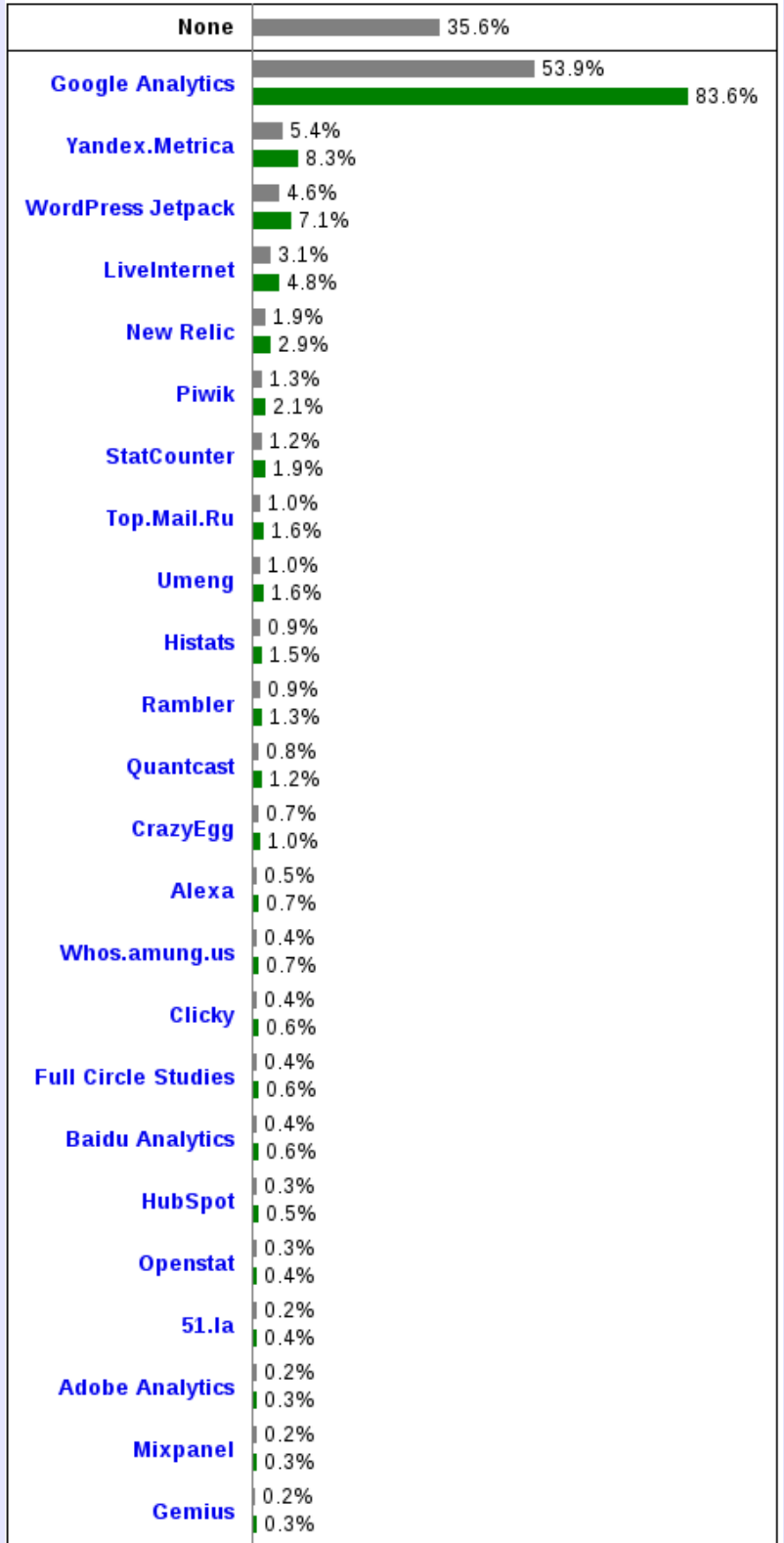
Browser fingerprinting

- Multi-monitor detection,
- Internal HashTable implementation detection
- WebRTC fingerprinting
- Math constants
- Accessibility fingerprinting
- Camera information
- DRM support
- Accelerometer support
- Virtual keyboards
- List of supported gestures (for touch-enabled devices)
- Pixel density
- Video and audio codecs availability
- Audio stack fingerprinting

Third-party analytics

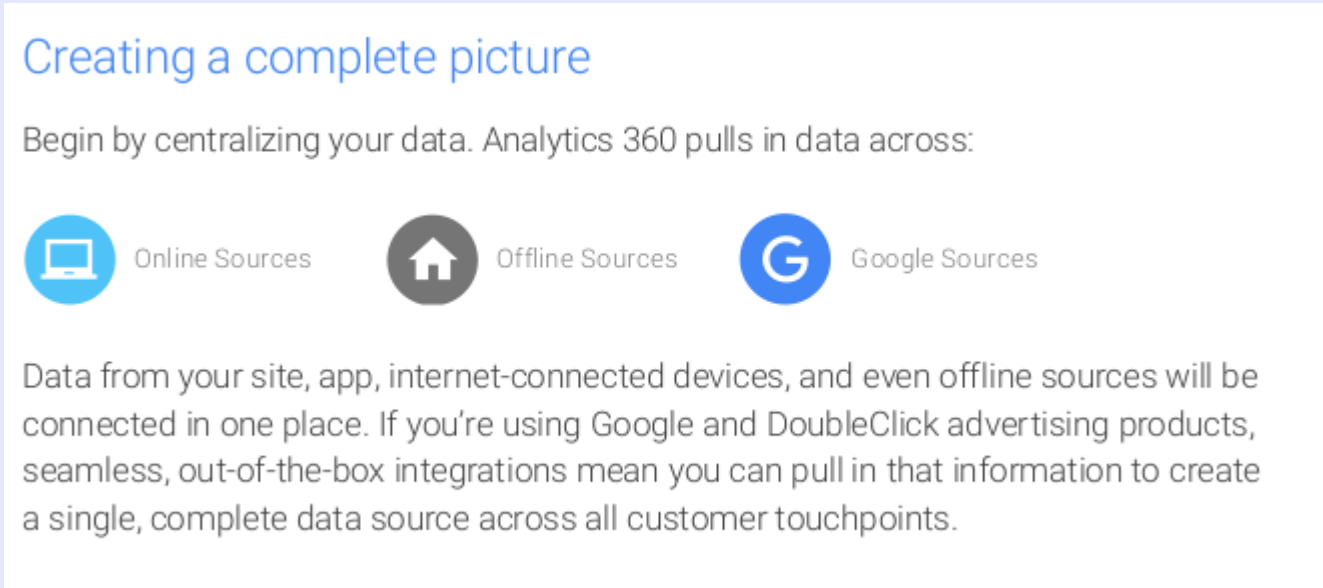
- 53.9% of all sites use Google Analytics

https://w3techs.com/technologies/overview/traffic_analysis/all



Third-party analytics

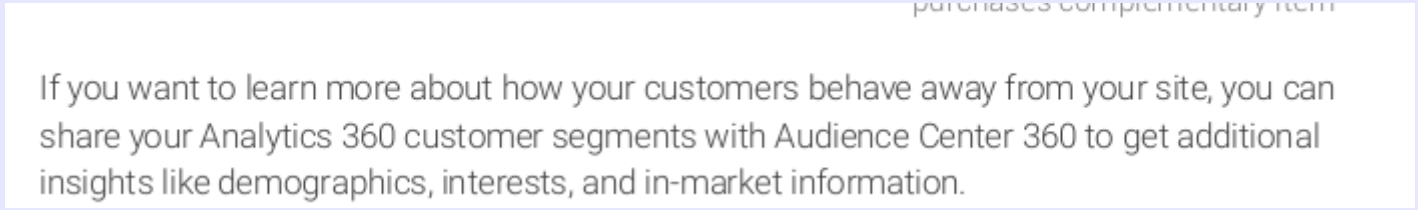
- 53.9% of all sites use Google Analytics



https://www.google.com/analytics/analytics/features/

Third-party analytics

- 53.9% of all sites use Google Analytics

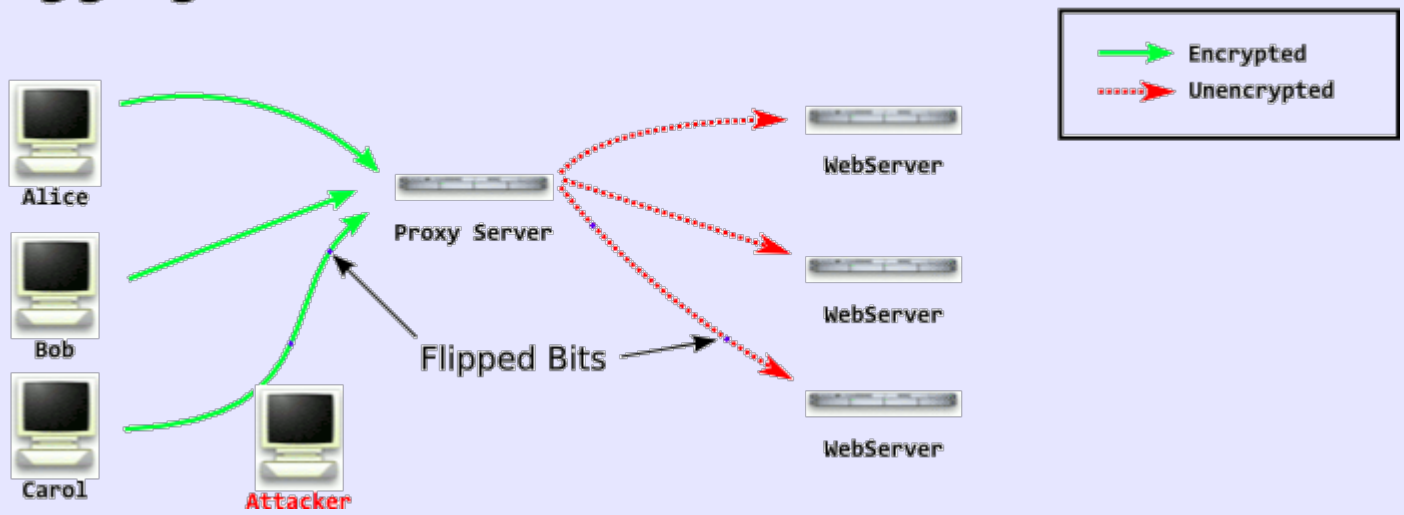


https://www.google.com/analytics/analytics/features/

Attacks on anonymity systems

- (We'll see the systems these apply to in a moment)
- Rubber hose attacks
- Traffic confirmation attacks ("blackbox attacks" in the reading)
- Tagging attacks

Tagging Attack



COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Anonymity networks

Trusted or semi-trusted relays

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

Trusted relays

- Example: Type 0 remailers
  - a server keeps a dictionary between real and pseudonymous emails
  - request comes to the remailer, which strips identifying information, forwards it with a pseudonym, gets the response to that pseudonym, and returns it to the pseudonym holder
  - Penet
- Other Examples: Anonymous proxies (startpage.com), VPNs

Trusted relays

- 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

Semi-trusted relays

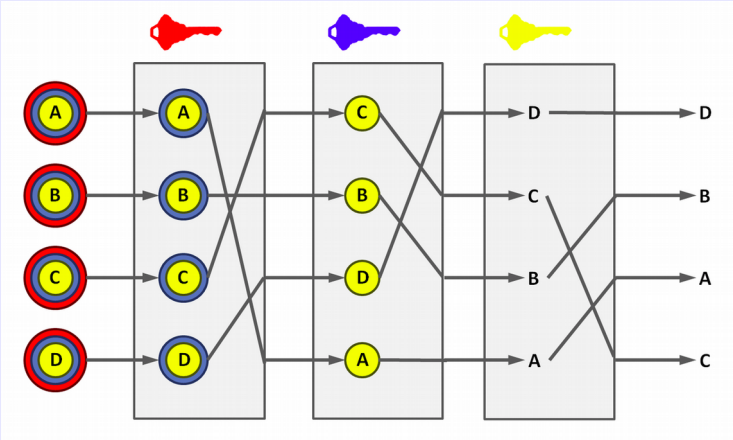
- Example: Type I “Cipherpunk” remailers
  - Layers of encryption to the remailer with PGP, which strips a layer to reveal the next hop, which it forwards to
  - anonymous replies via reply blocks
  - Problem: leaks size of the messages (what kind of attack does this leave open?), others...
  - A primitive sort of mixnet, on which more in a moment...

Semi-trusted relays

- Tagging attacks can still violate unlinkability
- replay attacks
- mixnets are high-latency

Mix systems are anonymity systems with semi-trusted relays

- Routing protocol with a cascade of cryptographic relays called ‘mixes’ each designed to provide **bitwise unlinkability**
- Mixes only know their neighbors



wikipedia.org

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 of blocks, ℓ

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

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

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

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

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

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

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

-

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 RSA.

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

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

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

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

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

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

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

Mix-nets are anonymity systems with semi-trusted relays

- Mixes only know their neighbors, not the whole route. (Question: Why?)
- All nodes have a public key
- Mixes also mix messages so that order of reception and order of departure doesn't leak information (what does that mean about latency?)
  - dummy traffic can also be added

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

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

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

Semi-trusted relays

- What are the problems with relying on just one mix? (Chaum)

Mix topologies

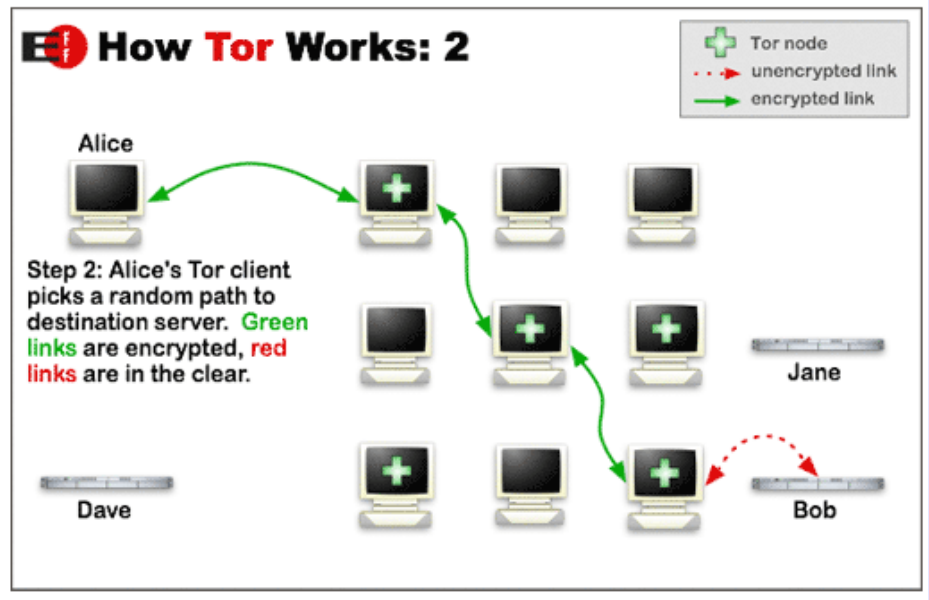
- Cascading: All nodes are always used, in the same order
- Scalability is a problem, requires setting up a fixed route with all nodes
- Only requires one honest node to preserve anonymity

Mix topologies

- 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)

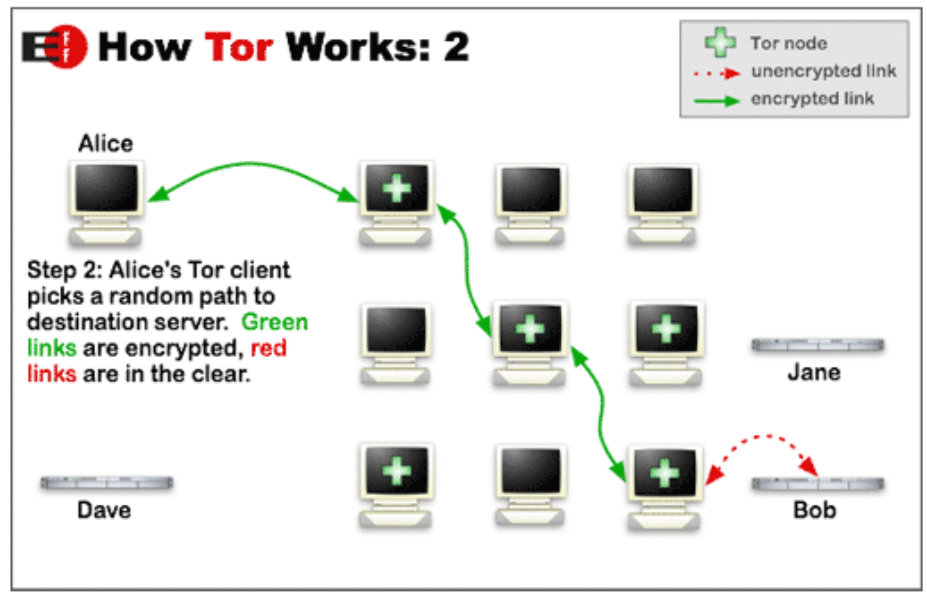
Onion Routing

- First was from the US Naval Laboratory, 1996
  - pure peering at this stage, no loafers like these days!
  - Freedom Network was an independent onion routing network from Zero Knowledge Systems



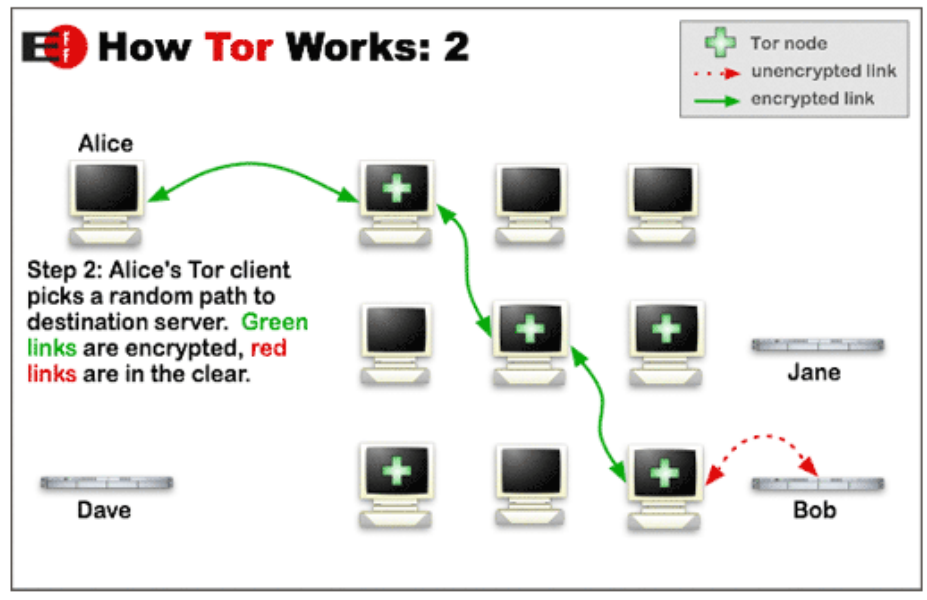
Onion Routing

- Circuit-based routing along obscured routes: onion routers only know their neighbors
- Designed for bidirectional, low-latency communication
- Depends on cover traffic



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



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

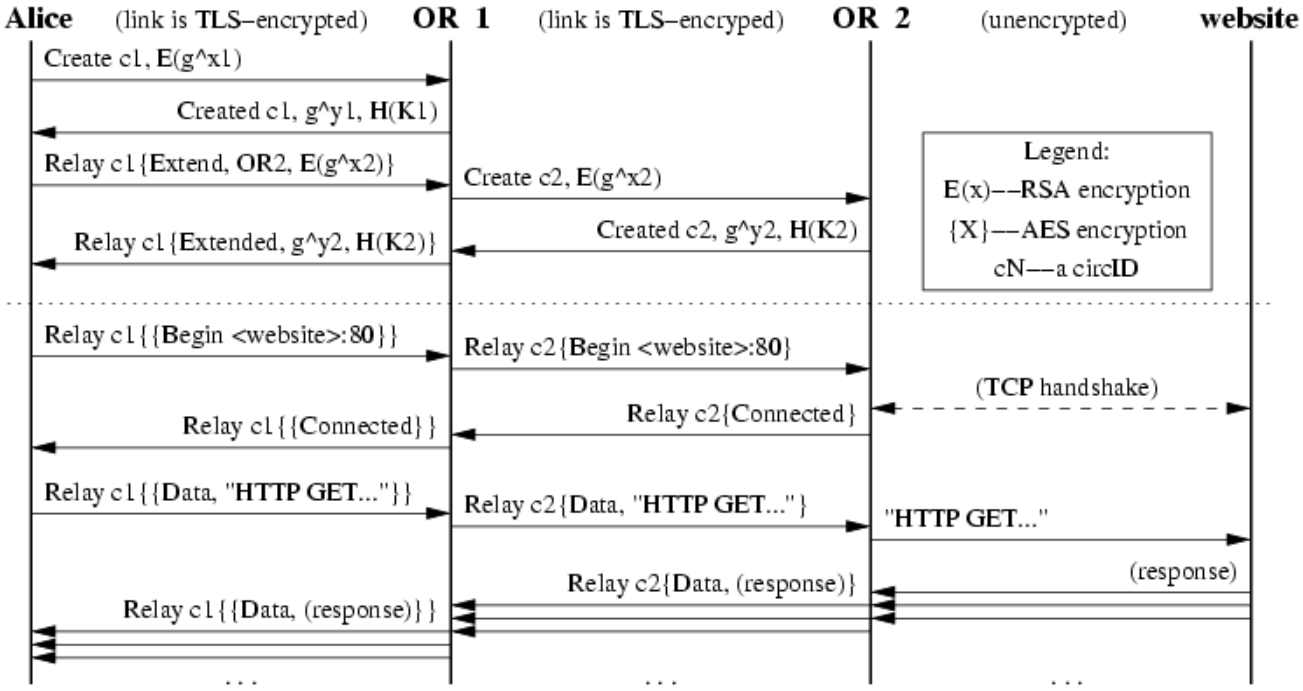


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

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

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

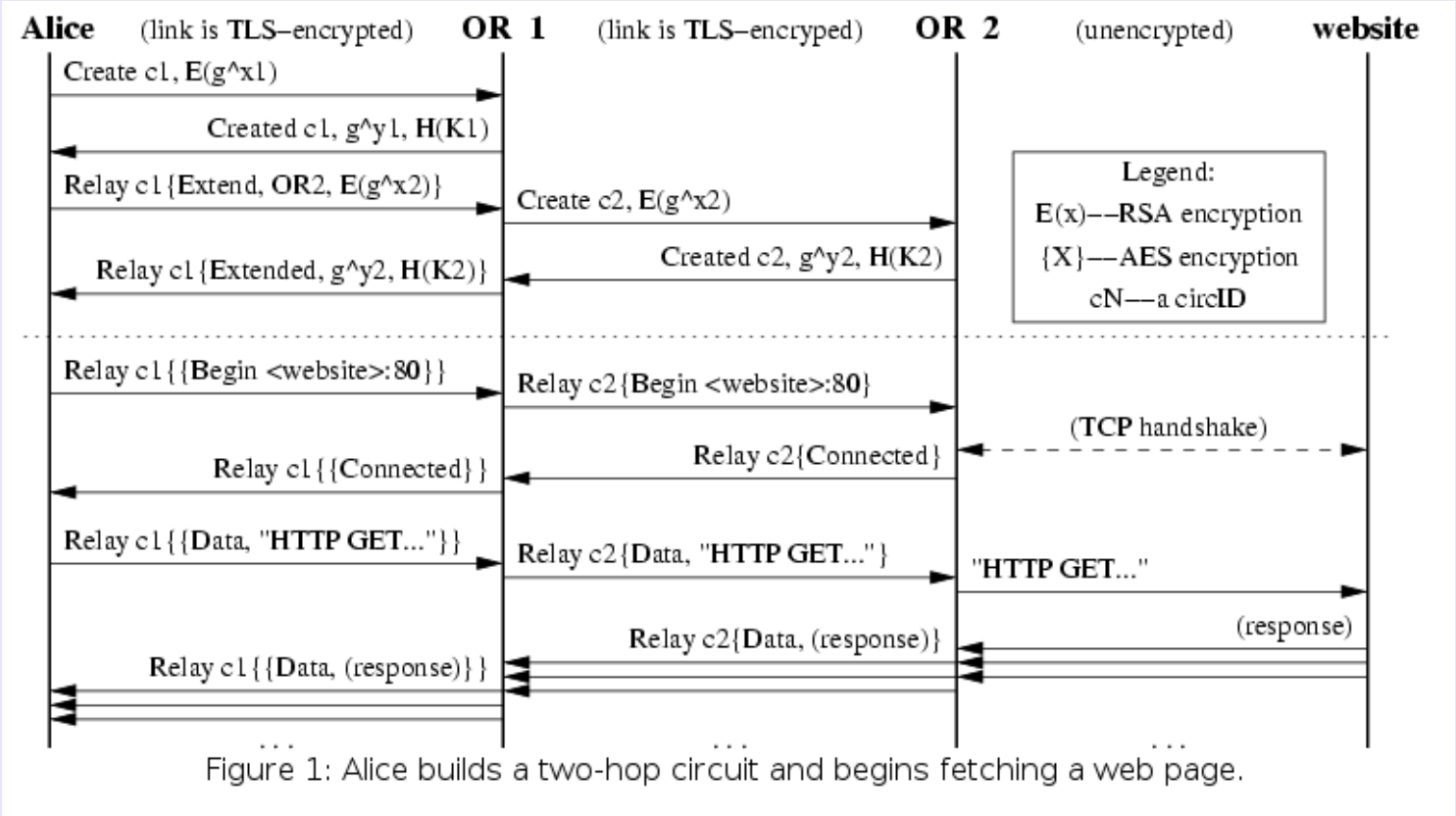


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

Discussion

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