CMPT 980 – Information Privacy

Module 4: Web Privacy

Threat Model

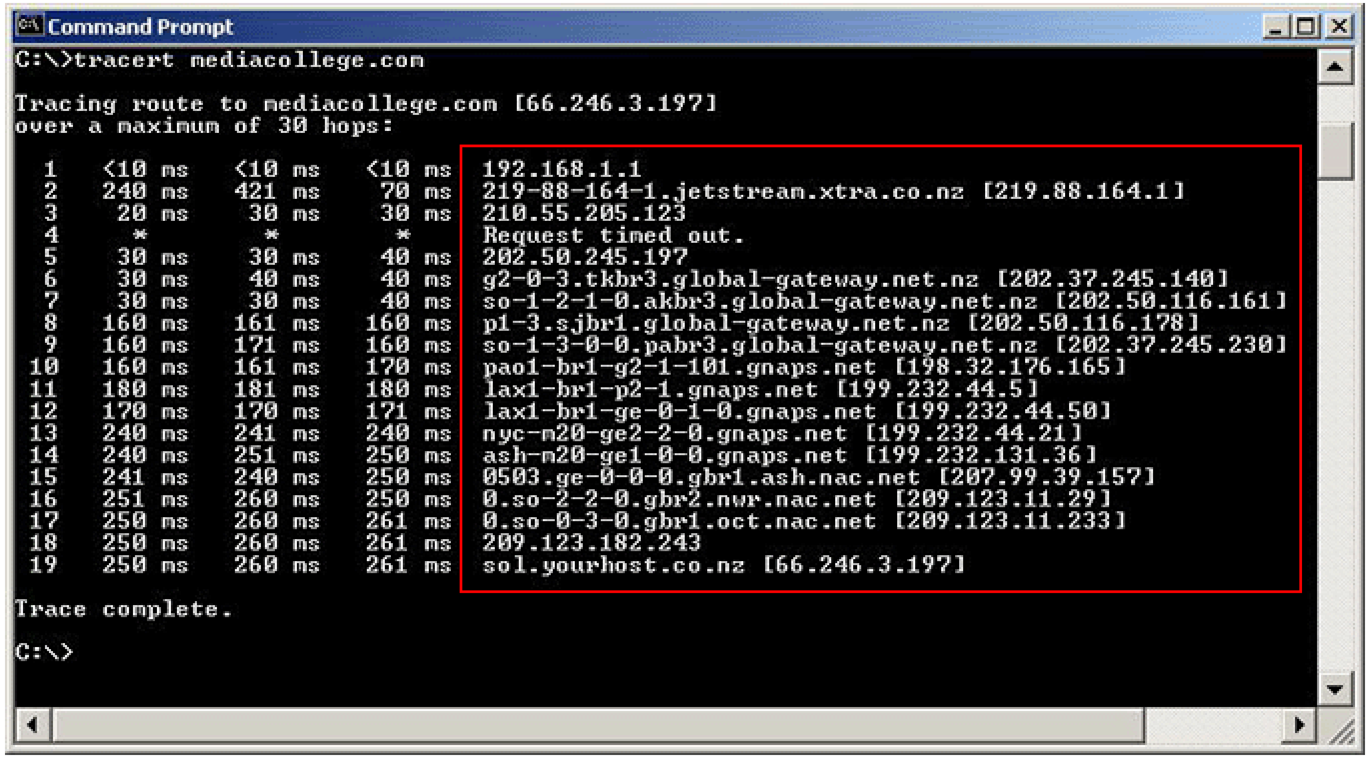
Even under encryption (TLS),
TCP/IP packets leak metadata:

- Client IP
- Destination server IP
- Time of packet
- Size of packet
- Number of connections

An eavesdropper could find out

- Who the client is
- What website (server) they're visiting

The Eavesdropper



Eavesdroppers can be:

- ISPs
- Government agencies
- Hackers on these routers
- Anyone that obtains data leaked from these routers
- Anyone near you (wireless)

Privacy concerns

Which websites you visit can reveal:

- Purchasing behavior – what items you are likely to be interested in
- Political preferences
- Personal beliefs – religion, world view, unconventional beliefs
- Usage – how one's time is spent on the Internet
- Illegal activities

Remailer systems

How can we send e-mails anonymously?

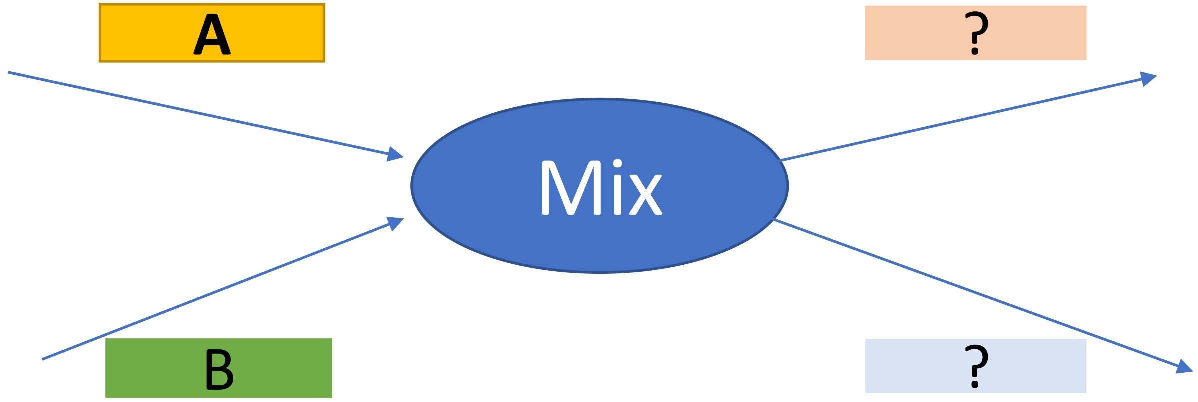
(I don't want people to know who I'm talking to.)

- A remailer mix network is a group of mixes (each with its own e-mail address) that can relay e-mails from anonymous senders.
  - Each mix has its own public/private key
- A sender encrypts her e-mail with the public keys of a series of mixes in order
  - Each mix decrypts 1 layer, and the final mix sends the decrypted message to the recipient
- When a message is returned, it is encrypted by each mix in order

Remailer systems

Attack:

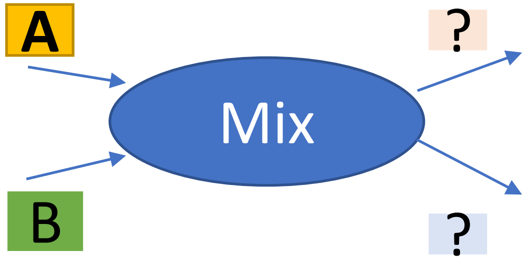
- The sender loses if an eavesdropper can follow her message along the mix network to the recipient



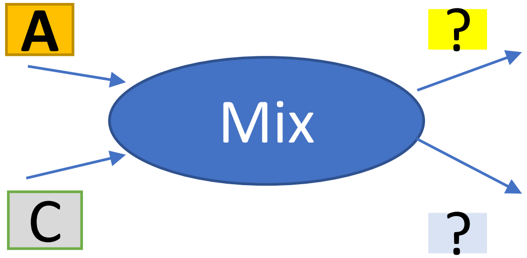
- Attacker wins if they are able to guess which outgoing message is A/B
- **Tagging attack**: Attacker changes the message A a little bit, and observes that change occurring in the output
  - Defeated by message authentication, but not so easily when reply blocks are used (later)
- Other attacks: tracing e-mail sizes, times…

Remailer systems

- Remailer systems can suffer from a **replay attack:**



Attacker resends message A:

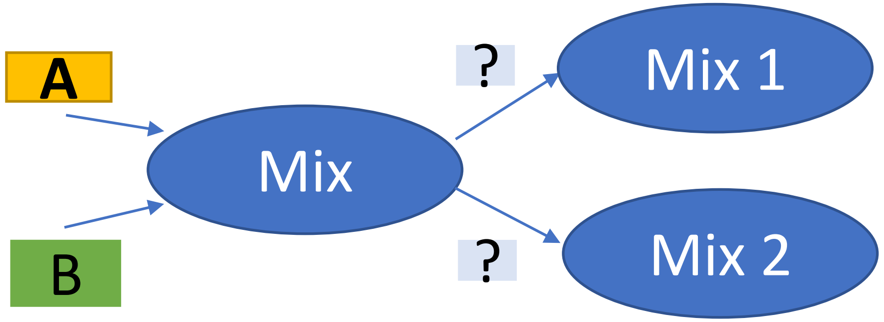


Since message A always decrypts the same way, it must be the lower output

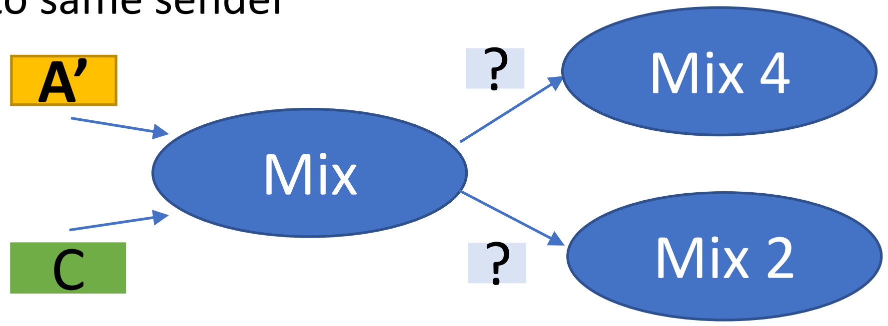
- Naïve solution is that each mix remembers recently seen messages and discards previous ones
- Better solution is for mixes to have rotating keys

Remailer systems

- The remailer needs a way to deliver replies back to us
  - *Reply blocks* contain instructions of how to reply to the sender: it is a series of mixes chosen by the sender together with the sender’s real address encrypted under these mixes’ keys in order
  - When used, each mix will decrypt one layer of the reply block, until the final mix recovers the true address
  - However, this can reduce anonymity with a **path intersection attack**:



A’ is reply to same sender



Since message A and A’ must use the same path, they both went on Mix 2

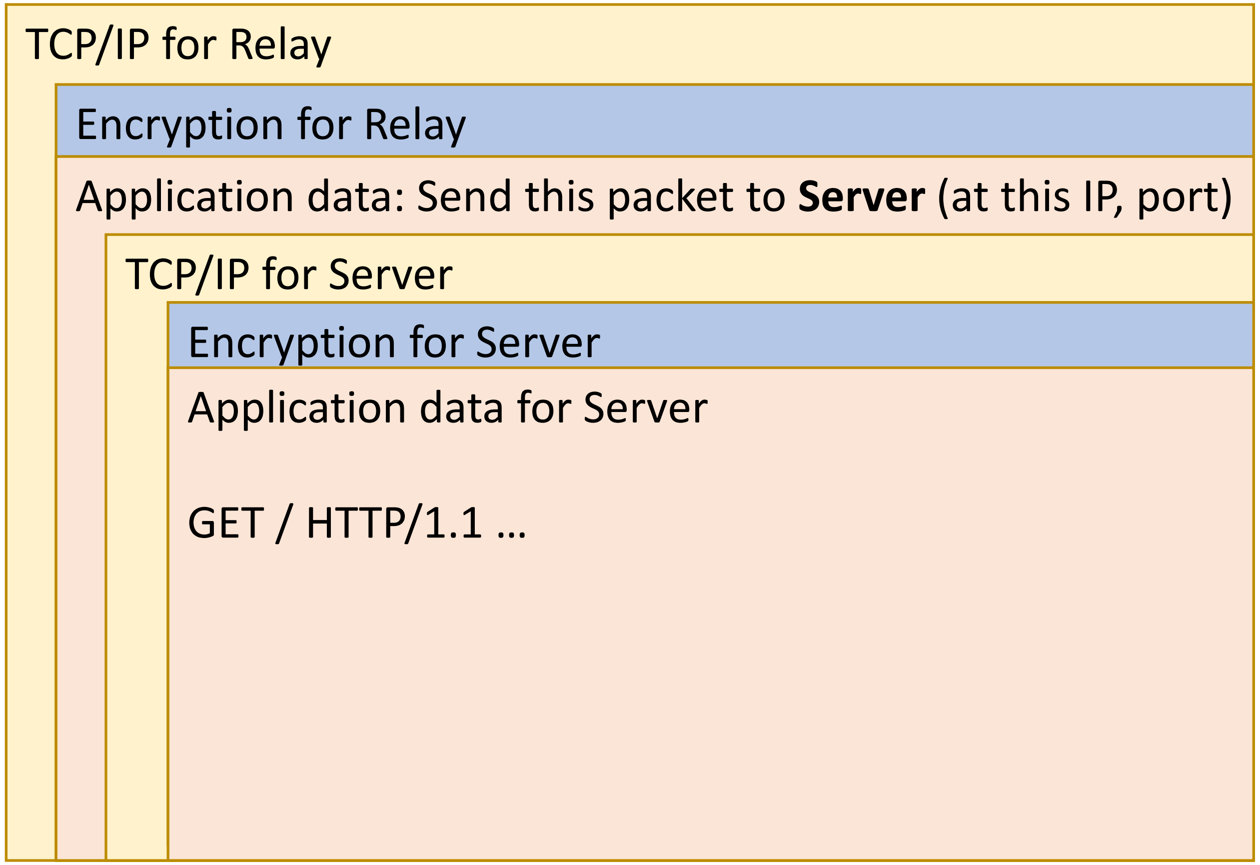
- Solution: Single-use reply blocks [Danezis03]

Onion Routing

How can I browse websites anonymously?

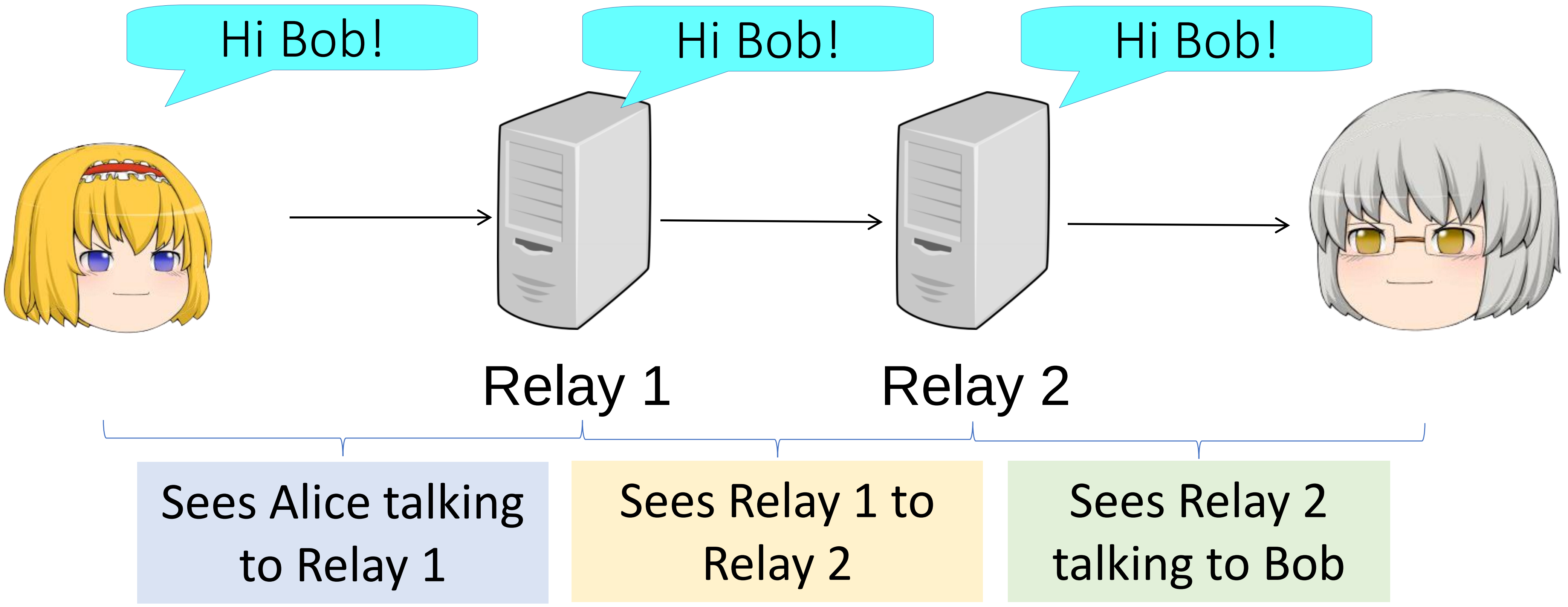
- No one should be able to tell which client IP is connected to which server IP
- Similar to remailer systems, but for websites – we use relays to deliver packets
- Doesn't have reply issues as only short-term connections are expected (they will be maintained by relays until client/server drop)
- The "real" packet for the server should be wrapped inside of a packet for the relay (tunneling)

Onion Routing

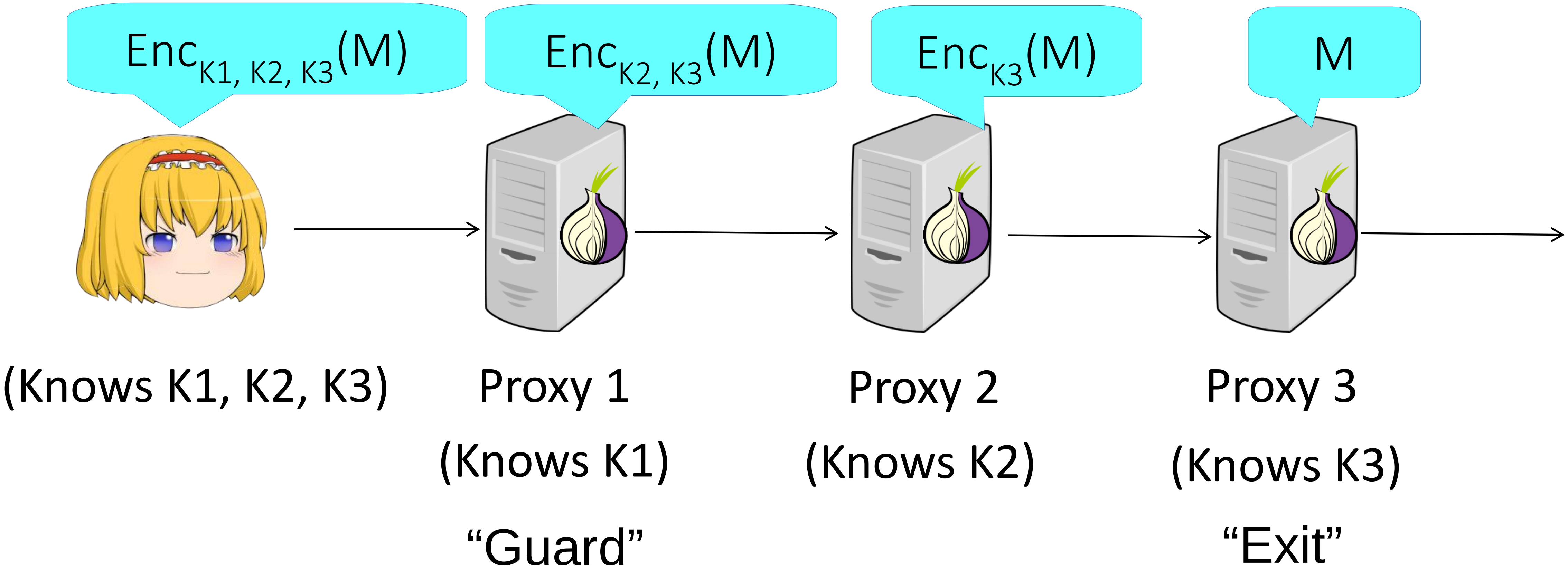


- We want more than 1 relay so that the relay doesn't see exactly who the server is (the relay could be an attacker)
- The "application data" for outer layers is just the address of the next relay, or the true server

Onion Routing



Tor

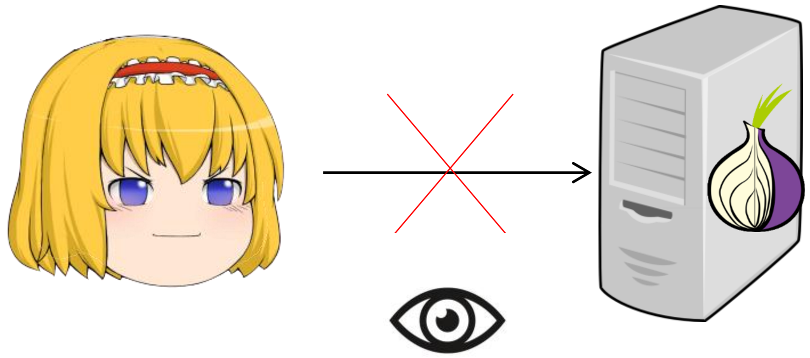


$$Enc_{K1, K2, K3}(M) = Enc_{K1}(Enc_{K2}(Enc_{K3}(M)))$$

Tor

- Circuits have three random relays instead of two
  - First relay = “guard”/”entry guard”
  - Last relay = “exit”
- Relays are volunteers
  - Relays can choose to be guards/exits if they satisfy certain criteria, or choose to be excluded
- Designed and maintained with funding from US government
- Supports the “dark web” – Hidden Services

Censorship and Tor

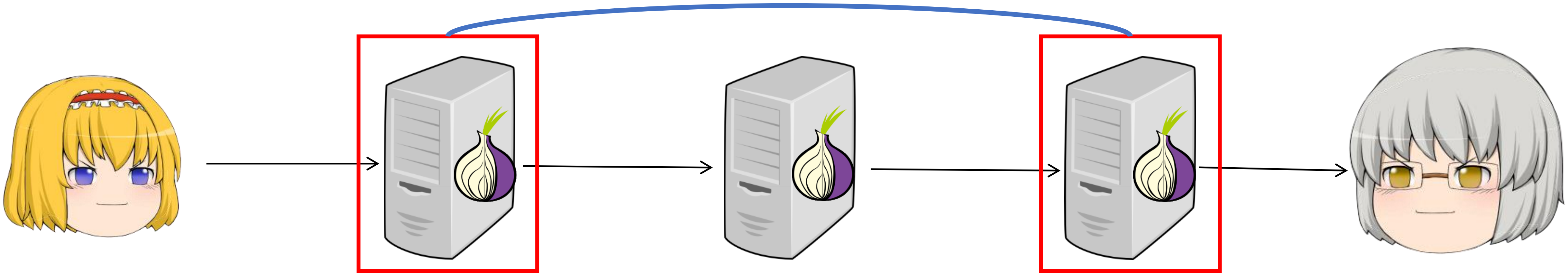


Censor:

1. Download list of all Tor relays
2. Block all traffic going to Tor relays

- Tor is used to evade censorship, but…
- Tor usage itself is blocked in several countries
    - Tor traffic can be identified from public relay directories
    - Tor traffic also has specific features
- Users exiting from Tor cannot access/use certain websites normally
    - e.g. Wikipedia detects accesses from Tor relays and does not allow article editing unless logged in

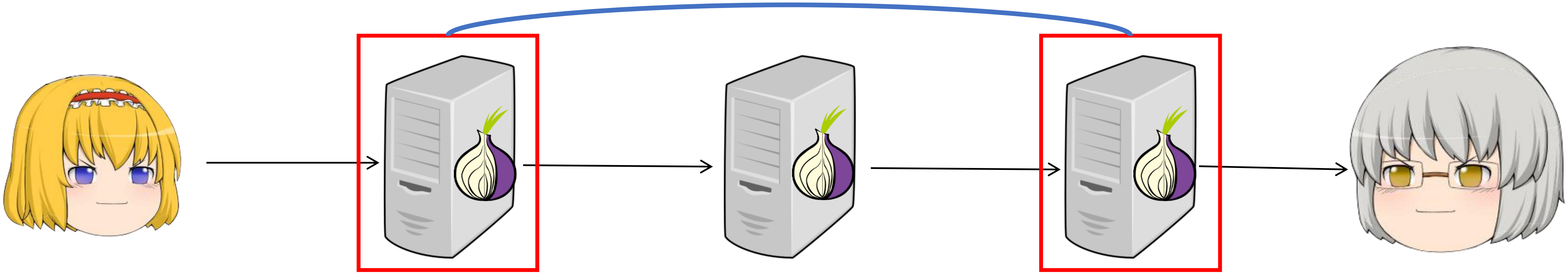
Compromising Tor



- The entry sees the client
- The exit sees the server
- If the same attacker controls both, she still needs to confirm a connection
  - **Flow correlation attack:** She can do this by delaying packets selectively (creating a discernible pattern)
  - [Nasr18] It is also possible to do this passively

[Nasr18] “DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning”

Compromising Tor



Can we increase the chance of controlling entry+exit?

- If we just drop circuits that we don’t control…
- If we can keep other relays congested...
  - [Evans09] We can build long paths repeatedly using a single relay
  - Later fixed

[Evans09] “A Practical Congestion Attack on Tor Using Long Paths”

Tor Browser

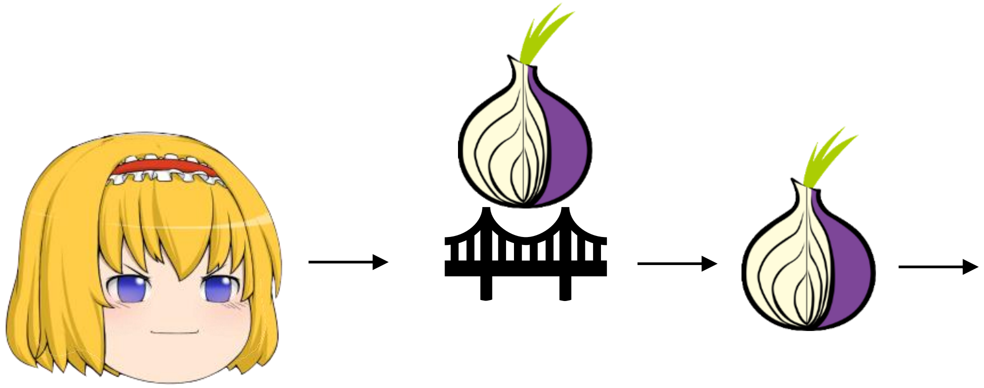
- Modification of Firefox to automatically launch and use Tor (and some other privacy-enhancing features)
- Behaves just like any other browser
- Important factor of Tor’s usability
- Privacy-enhancing features:
  - No disk caching
  - No tracking
  - Allows choosing new circuits (some versions)
  - Fixes for Browser Fingerprinting (later)

Tor Guards

- A circuit is considered compromised if the same attacker controls both the entry and the exit (because of the flow correlation attack)
- Time to first compromise = how long a user would use Tor before using their first compromised circuit
- To increase the time to first compromise, each client maintains a long-living list of Tor guards (first node), which last 30-90 days
    - If the list has no compromised guard, then the client will be safe for months; otherwise, they will be compromised, but their time to first compromise will not be much shorter
- Instead of randomly choosing guards from the entire set of relays, they will only use their list of guards
- Guards must be reliable and been on Tor for a long time

[Elahi12] “Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor”

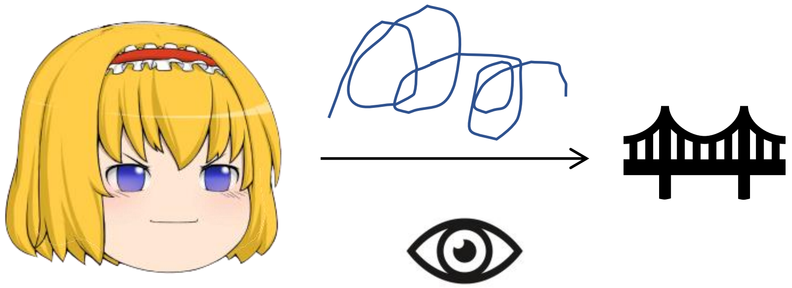
Tor Bridges



Bridges are special entry guards that aren't listed in the public directory

- In a circuit, the bridge optionally replaces the entry guard
- Used to evade censorship
- Can be obtained (limited number per day) in several ways
  - E-mailing the developers
  - Website with captcha

Obfsproxy



Attacker

Between client and bridge

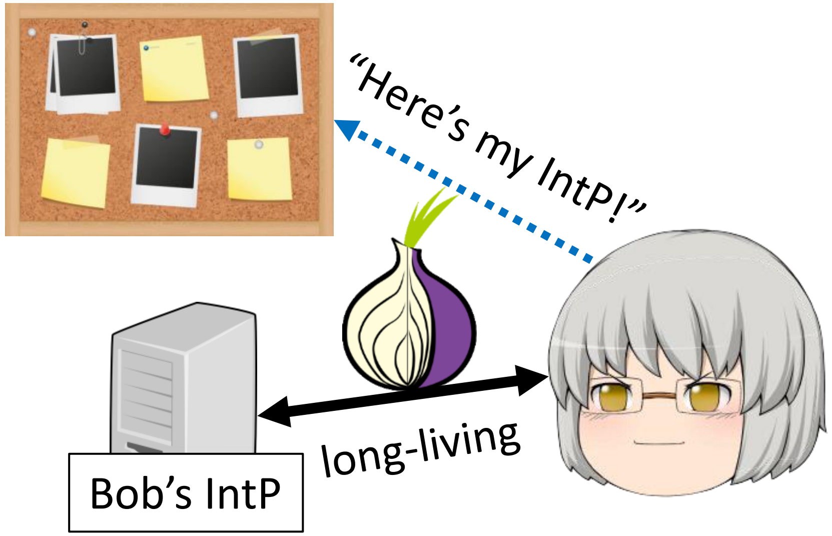
- A technologically adapt censor (e.g. Great Firewall) may block Tor by identifying patterns in user traffic, such as the presence of a particular byte in the handshake

  Obfsproxy tries to defeat such an attacker by hiding such patterns
  - It is only used on bridges
- An early version of obfsproxy has too much randomness in the handshake
- Newest version can hide timing and packet size features

Hidden Services

- A hidden service is a web server that allows connection on Tor *without revealing its own IP*
- This is the "Dark Web"
- Requires the web server to install and use Tor too
- Clients can use a hidden service by typing the onion address into Tor Browser
- An onion address is also the web server's public key

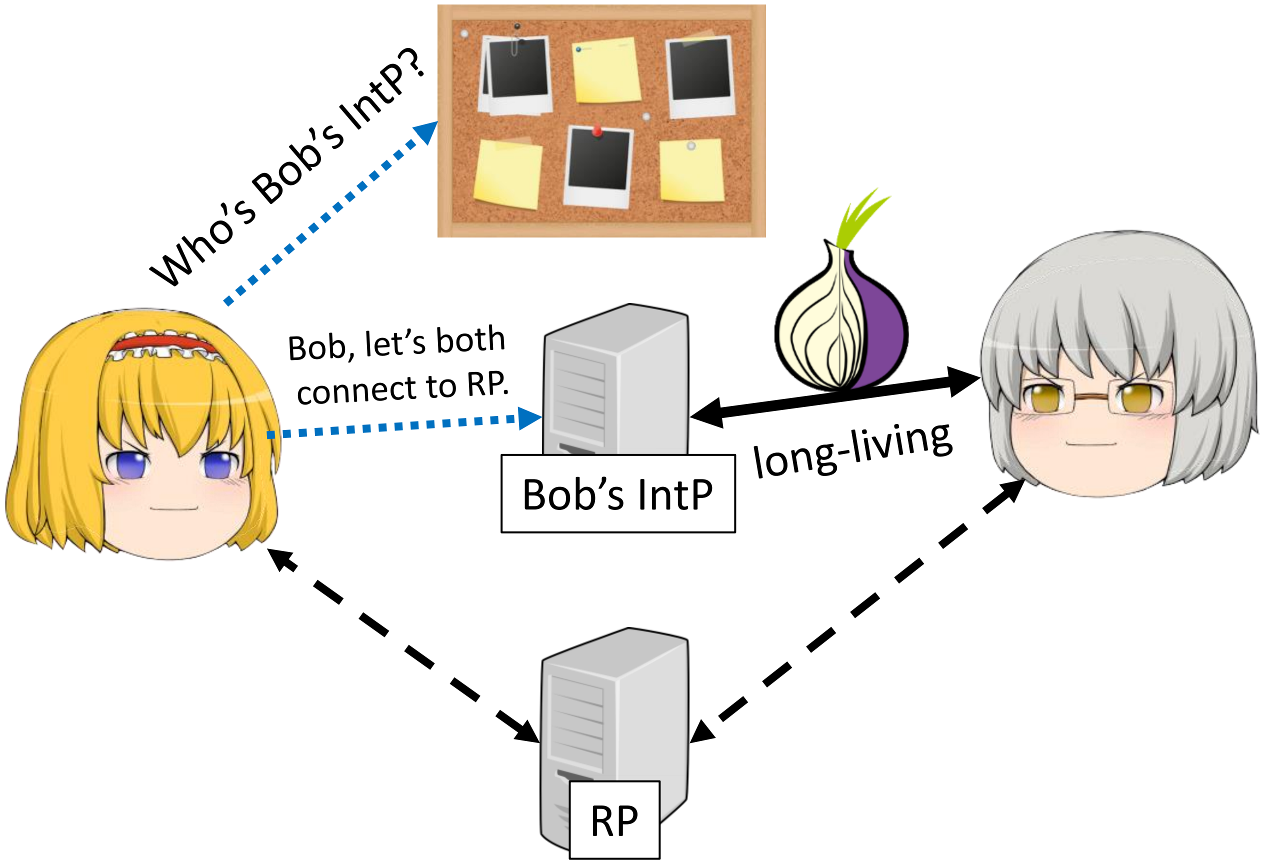
Hidden Services



1. Bob (server) chooses several Tor relays as Introduction Points and establishes long-term Tor circuits to them
2. Bob tells a public directory who his IntPs are

https://community.torproject.org/onion-services/overview/

Hidden Services



3. Alice chooses a Tor relay as a rendezvous point (to meet Bob there), and tells Bob's IPs what the RP is
4. Alice and Bob both connect to the RP through Tor

Browser Fingerprinting

- A way for web servers to track users *even if they are using Tor*
- Relies on distinct fingerprints that a browser would leave through HTTP requests and AJAX
  - List and order of installed plugins
  - List and order of installed fonts
  - User agent (Browser version, OS version, language, CPU, etc.)
  - Screen resolution
- Panopticlick: Out of 470k browsers, 83.6% were unique
- Tor Browser has to carefully control/block all of these possible fingerprints

[Eckersley2010] “How Unique is your Browser?”

Canvas Fingerprinting

- A specific type of browser fingerprinting
- The web page contains a canvas element which makes the user's browser draw an invisible image
  - Different browsers usually draw the image differently as it depends on many factors
- Currently widely used for tracking

Images from "ML-CB: Machine Learning Canvas Block"

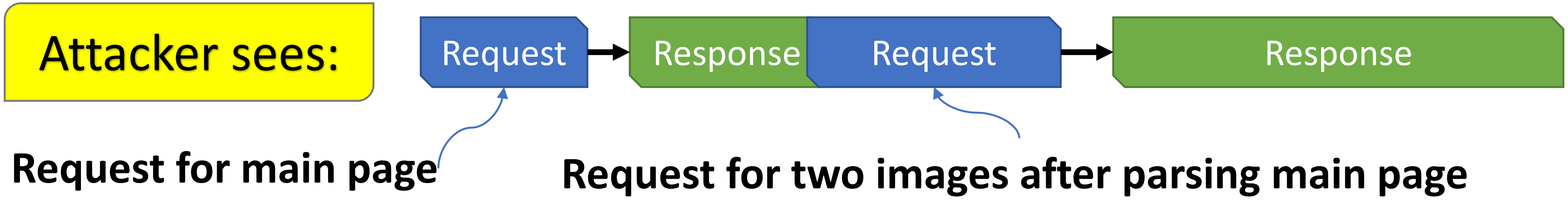
Traffic Analysis

- Specifically known as Website Fingerprinting for Tor
- A way for local, passive eavesdroppers to track users *on Tor*
  - The attacker already knows the user’s identity
- Machine learning on traffic traces of users using Tor to classify behavior
  - Packet times
  - Packet directions
  - Packet sizes (but not on Tor)
- Can be used to identify web pages, video watched, etc.

Why does Traffic Analysis work?

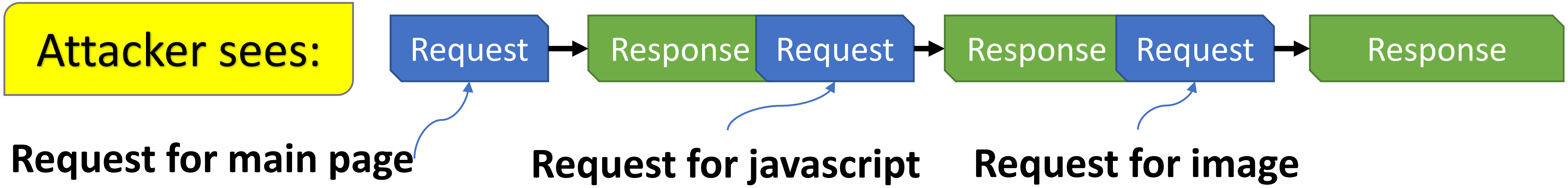
Site 1

Two images and some text on home page.



Site 2

Some text and javascript, which loads a dynamic image.



Example: OSAD

- Optimal String Alignment Distance: A version of edit distance
  - Minimum number of insertion, deletion, substitutions and transpositions to transform one string into another
  - Each packet sequence is treated as a string of -1 and 1 (time is discarded)
  - Can weigh some operations differently
- Use SVM with a kernel calculated by OSAD

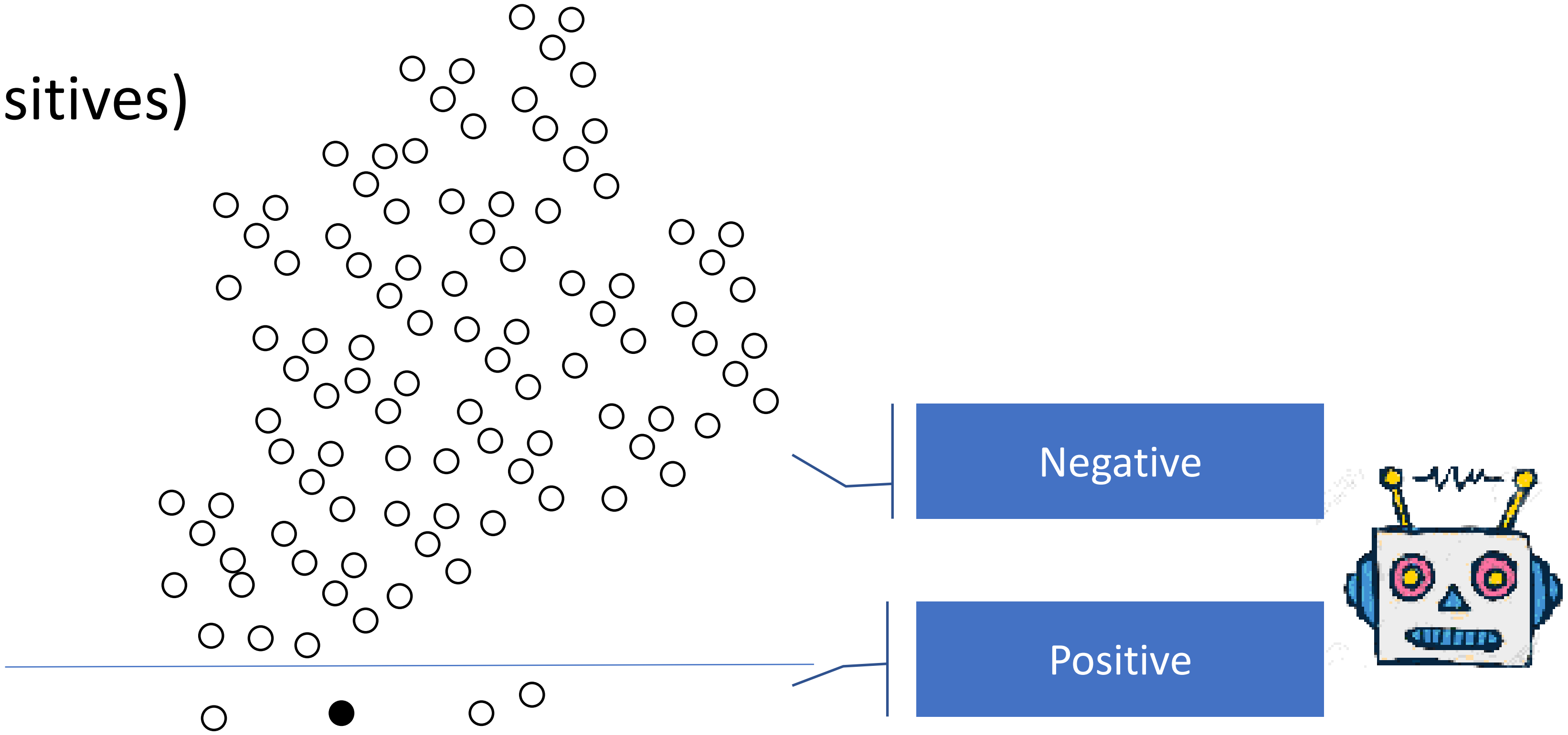
Example: CUMUL

- Calculate a cumulative sum of packet lengths, taking client-to-server packets as positive and server-to-client as negative
- Extract 100 evenly spaced points from the curve
- Add the total number of packets and the total packet size in either direction as 4 more features
- Use an SVM to classify
  - For multi-class, use one-against-one tournaments

Traffic Analysis as a machine learning problem

Positives (Sensitives)

Negatives



True Positive Rate: 100%

True Negative Rate: 97%

Precision = 25%

Base Rate Fallacy

What is precision?

Q. Bob writes an image-scraping tool to find cat pictures online.
Testing it on a data set of 1,000 cat pictures and 1,000 non-cat pictures, the tool claims **900 of the cat pictures are cats** and **300 of the non-cat pictures are cats.** How precise will the algorithm be in reality?

A. We don’t know. What portion of real pictures are cat pictures?
(If it is 1%, then the algorithm will have about 2.9% precision.)

Defense against Traffic Analysis?

- Broadly, three strategies:
  - **Noise:** Add dummy packets to confuse the classifier
  - **Mimicry:** Pretend to be another website
  - **Regularization:** Fix packet patterns
- Unfortunately, many defenses have been broken by newer attacks
- How can we prove a defense will work against future attacks?

Example

- Regularization strategy (BuFLO, Dyer 2012):
  - Hide packet lengths (Tor already does this)
  - Fix packet sending rate; if no real data, send dummy packet
    - Even if too much data, rate should not change
  - Can only stop at prescribed times (e.g. every 10 seconds)
- Benefit of this approach: we can use anonymity sets to analyze the result
  - No attack can distinguish between two traces in the same set
- However, this is very expensive and slow: more than 100% data overhead and delay in packets

Example

- Noise-adding strategy (WTF-PAD, Juarez 2016):
  - Choose a target distribution of interpacket timings, then sample from it continuously
  - Add dummy packets if current interpacket timing is too large compared to the sample
  - Can distinguish between bursts and gaps to simulate patterns better
- Much cheaper and shown to work against older attacks; does not delay any packets
- Newer attacks broke it

Censorship Resistance Systems

- The goal of a Censorship Resistance System (CRS) is to allow communications past a censor who is monitoring the user’s traffic
- The censor can:
  - Use DPI to read the packets
  - Block based on IP or other headers
  - Use traffic analysis to determine the type of content
- The CRS should hold up against all these things

Censorship Resistance Systems

- Categories of CRS:
  - **Mimicry**: pretends to be another protocol
  - **Covert Channel**: uses another protocol to send signals
  - **Traffic Shaping**: defeats the censor's traffic analysis
  - **End-to-Middle (E2M)**: shows a fake route to the censor
- Performance metrics are also important: latency, throughput, packet loss, etc.
- Different systems may have different privacy properties (hiding user/server participation)

Example: Telex

- A E2M CRS; requires an ISP to cooperate
- The client pretends to visit a decoy server, but includes a tag in the ClientHello of TLS
  - The tag is encrypted using the ISP’s public key
- The ISP monitors all traffic for tags that indicate the client
  - Once observed, the ISP cuts communication with the decoy server and acts as a proxy for the real server
- The key between the client and the ISP is based on a shared secret derived from the tag

Example: SkypeMorph

- The client and server pretend to be making a Skype call
- Client and server public keys are communicated through Skype messages
- Then, the communication establishment is done through Skype video ringing; this call is dropped after connection material is sent
- Later found to have some weaknesses on connection establishment, along with other protocols; can possibly be distinguished by attacker