Privacy & Anonymity

CMSC 23200, Spring 2026, Lecture 14

David Cash and Grant Ho

University of Chicago
(Slides adapted from Peyrin Kao, Vern Paxson, and Zakir Durumeric)

Logistics

- Assignment 5 will be released this Fri (May 8)
  - Due Thursday, May 14 by 11:59pm

- Next week:
  - We will hold all Office Hours as scheduled
  - Discussion section will be held (Mon May 11)

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses/Services
  - Onion Services
  - Tor in Practice

What is Privacy?

Many different definitions:

- Privacy is control over your own information. Freedom from intrusion into personal matters
- Privacy is a person's right or expectation to control the disclosure of his/her personal information, including activity metadata
- Privacy is the "right to be let alone" — Louis Brandeis

Examples:

- What does a website (EdStem, gmail, Spotify, ...) share about with you advertisers?
- Is your medical data shared with researchers?
- Who can see your credit card payment history? (Credit scorers, banks, ...)

Anonymity: Related Concept

Anonymity (“without a name”): Concealing your identity

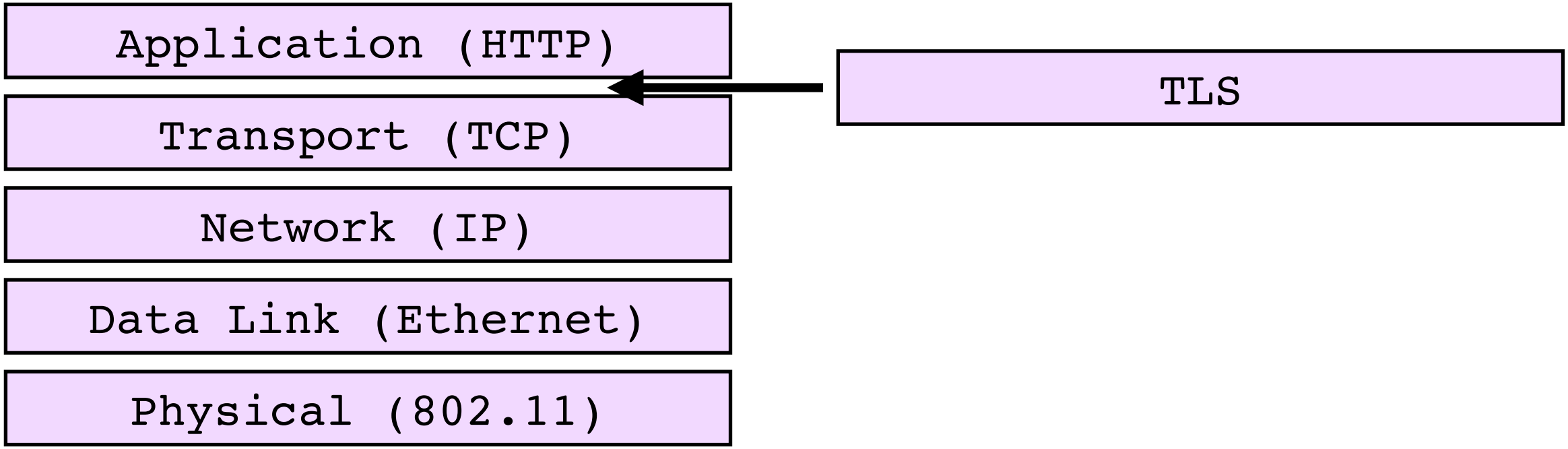
- Anonymous communication: the identity of source & destination in communication are concealed
- Anonymity provides some forms of privacy (e.g., unlinkability: prevents attackers from knowing action/information = yours, etc.)

Anonymity is not **confidentiality**

- Confidentiality hides the contents of the communication
- Anonymity hides the identities of who is communicating with whom

By Peter Steiner, in
The New Yorker (1993)

Recall: Transport Layer Security (TLS)



- TLS takes requests from application (e.g. browser speaking HTTP)
- TLS creates secure channel on top of TCP

TLS Protocol Structure (Very Roughly)

Check s

$A \leftarrow g^a$

$K \leftarrow Hash(B^a)$

Key Exchange ("Handshake")

A

B, Cert, s

uchicago.edu

$B \leftarrow g^b$

$K \leftarrow Hash(A^b)$

$s \leftarrow Sign(A)$

Symmetric Encryption ("Record Protocol")

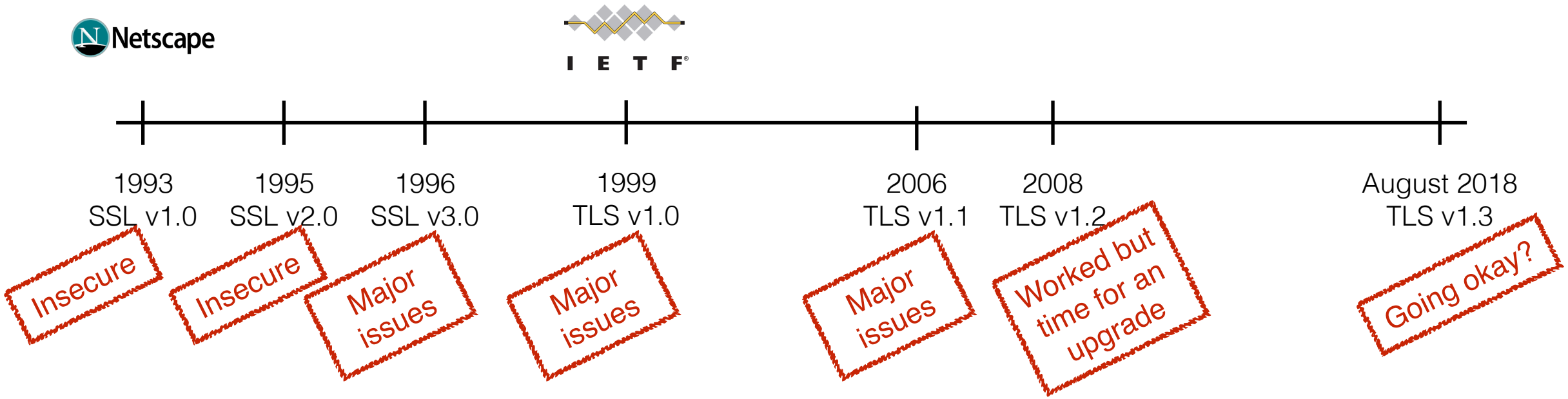
<application data encrypted with K>

<application data encrypted with K>

...

TLS History

- SSL = “Secure Sockets Layer”
- TLS = “Transport Layer Security” (renaming of SSL)



TLS Adoption

Google

Official Blog

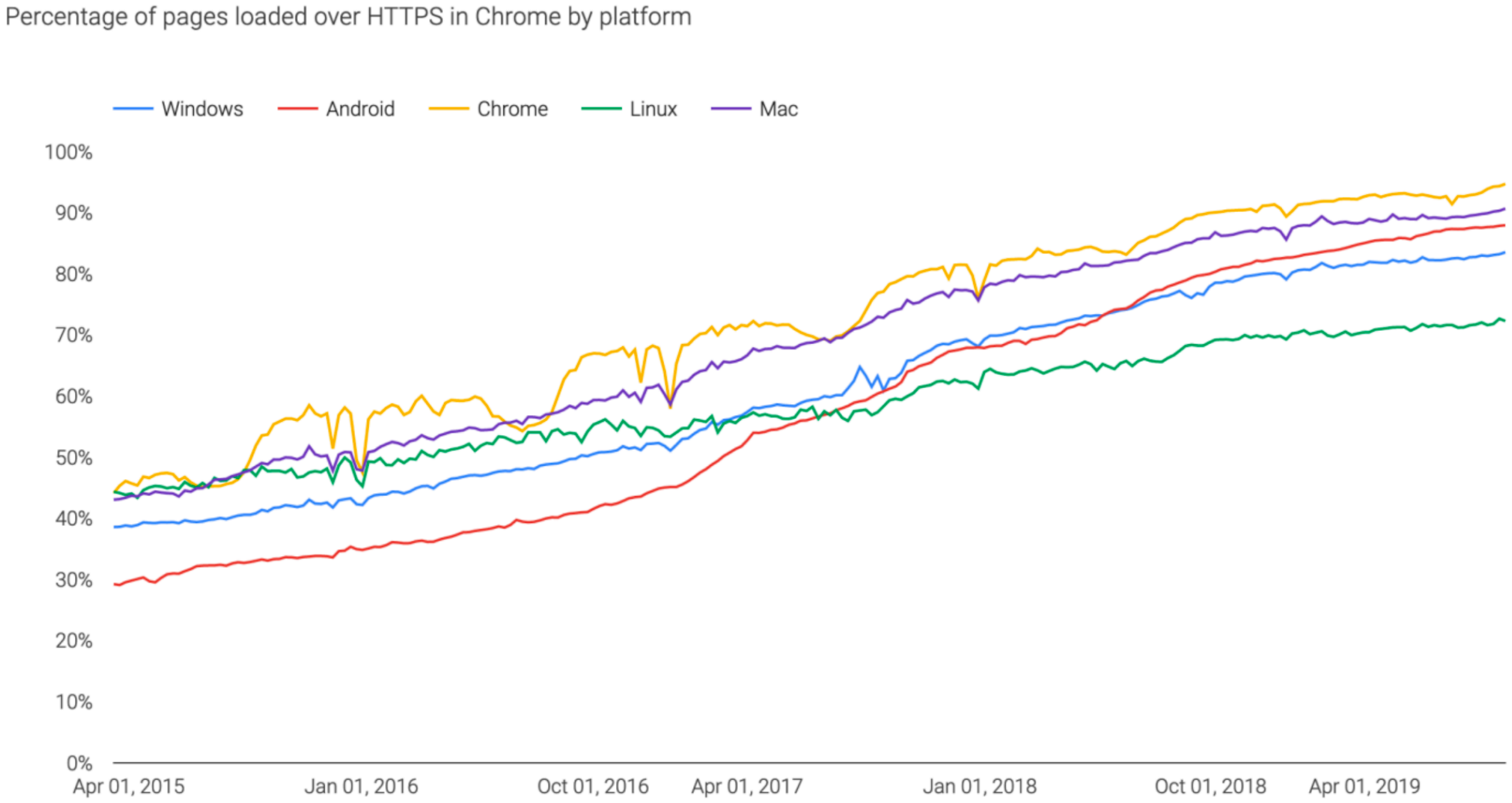
Insights from Googlers into our products, technology, and the Google culture

Search more securely with encrypted Google web search

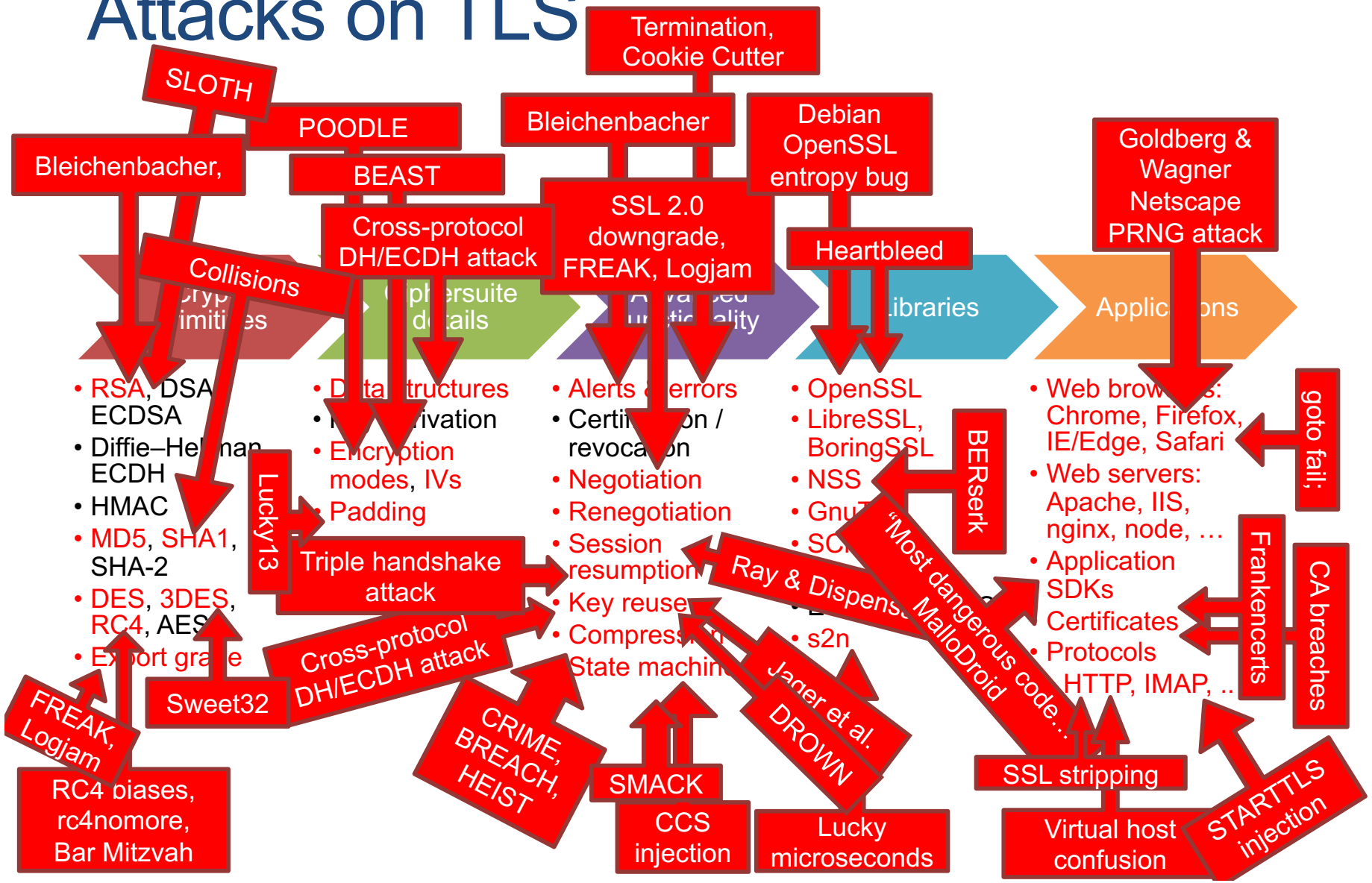
May 21, 2010

Update June 25, 2010: Since we introduced our encrypted search option last month, we've been listening closely to user feedback. Many users appreciate the capability to perform searches with better protection against snooping from third parties. We've also heard about some challenges faced by various school districts, and today, we want to inform you that we've moved encrypted search from https://www.google.com to https://encrypted.google.com. The site functions in the same way. For more information on this change, please read on here.

TLS Adoption



Attacks on TLS



TLS, Metadata, & Anonymity

TLS only protects content… doesn't offer anonymity (IP addresses still visible)

Anonymity often requires protecting metadata:

- Who is visiting what websites? Who is sending messages to whom?
  - Government tries to track criminal activity...
  - ... but might also track political dissidents
- We may want to hide the existence of the message at all (maybe sending an encrypted message at all will attract attention)

Achieving Anonymity is Difficult

- Anonymity is difficult, if not impossible, to achieve for users
  - Source and destination IP address visible in every packet

- Anonymity is easier for attackers
  - An attacker can hack into someone else's computer and/or often spoof messages from fake source addresses!

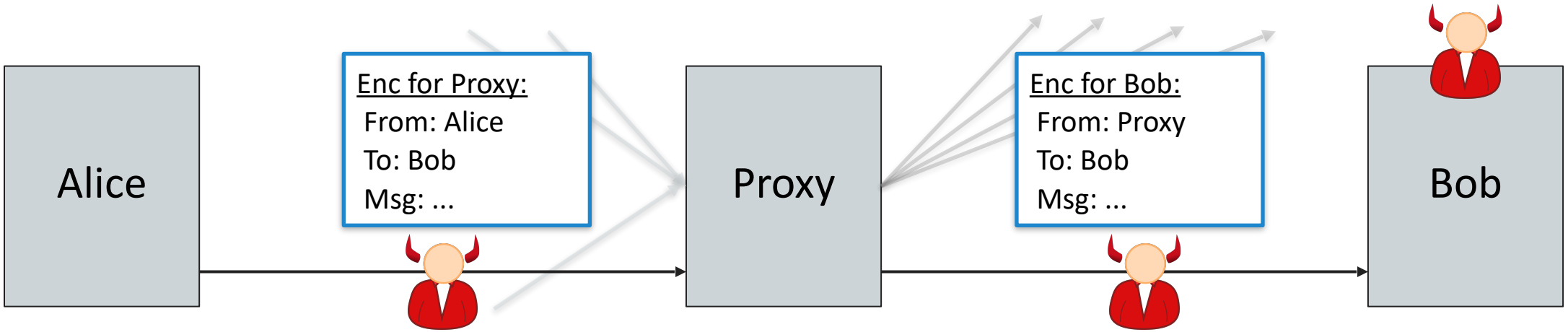
- Main strategy for anonymity:
  - Trust someone else
  - Ask them else to send messages on your behalf

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses/Services
  - Onion Services
  - Tor in Practice

Proxies

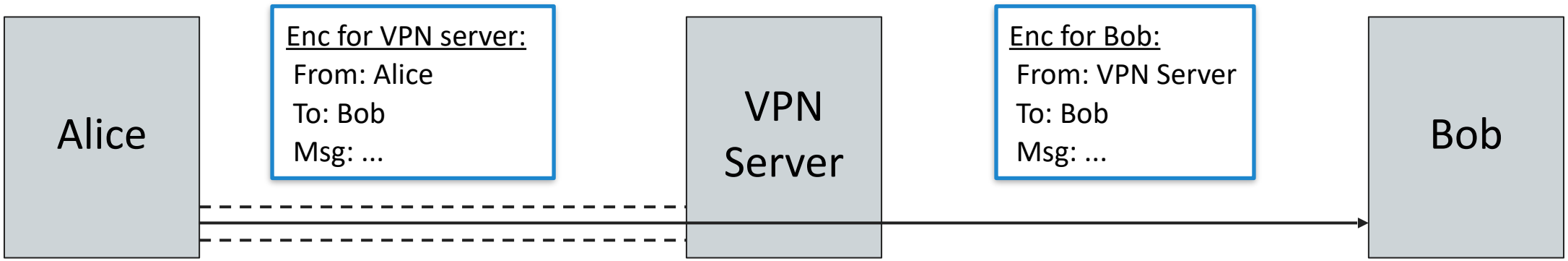
- **Goal & Threat Model:** Alice wants to anonymously send a message to Bob
  - Bob should not see that Alice is sending messages
  - Also: eavesdropper (Eve) cannot deduce that Alice is talking to Bob
- **Proxy**: A third party that relays Internet traffic
  - Alice sends the message and the recipient (Bob) to the proxy,
  - Proxy forwards the message to Bob (along with many other src + dest pairs)
    - The recipient's name is encrypted, so an eavesdropper looking at packets can't see both msg src & dest
  - Bob receives the message from the proxy: Does not see IP address of Alice



Virtual Private Networks (VPNs)

VPNs: A virtual connection to an internal network

- Creates encrypted “tunnel” to VPN server at the Network / IP layer
- Traffic from client first encrypted & sent to VPN server
- VPN server then decrypts & forwards traffic to final destination
- VPNs can act as a proxy into internal network: outbound traffic appears to come from internal network and not Alice
- Often used by orgs for increased security (e.g. on UChicago network)
- Also used to evade geographic restrictions



Naive approach to anonymity: VPNs

- Many popular VPNs available today

Trusting VPNs

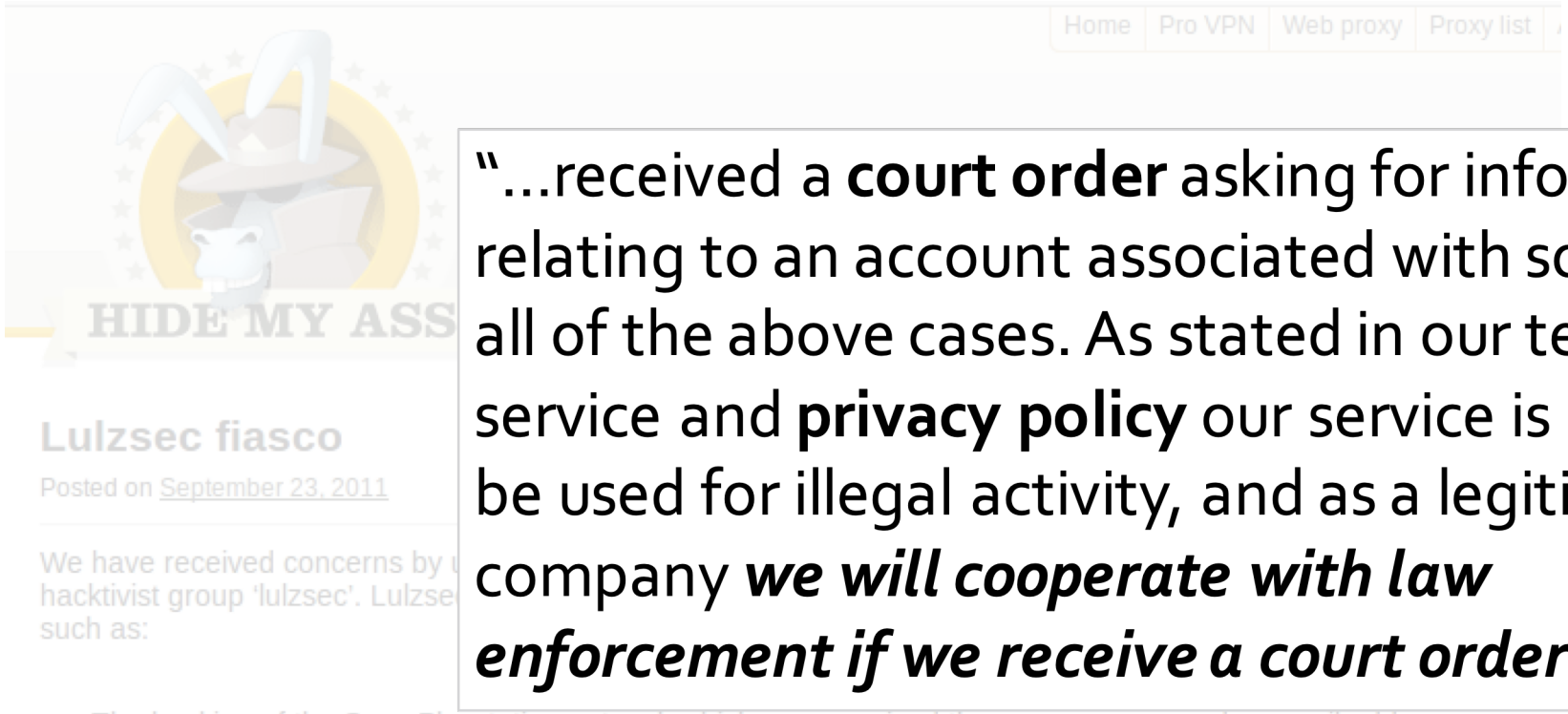
Lulzsec fiasco

Posted on September 23, 2011

We have received concerns by users that our VPN service was utilized by a member or members of the hacktivist group 'lulzsec'. Lulzsec have been ALLEGEDLY been responsible for a number of high profile cases such as:

- The hacking of the Sony Playstation network which compromised the names, passwords, e-mail addresses, home addresses and dates of birth of thousands of people.
- The DDOS attack which knocked the British governments SOCA (Serious Organised Crime Agency) and other government websites offline.
- The release of various sensitive and confidential information from companies such as AT&T, Viacom, Disney, EMI, NBC Universal, and AOL.
- Gaining access to NATO servers and releasing documents regarding the communication and information services (CIS) in Kosovo.
- The defacement of British newspaper websites The Sun & The Times.
- The hacking of 77 law enforcement sheriff websites.

Trusting VPNs



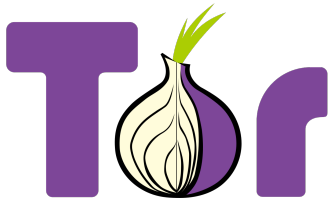
"…received a **court order** asking for information relating to an account associated with some or all of the above cases. As stated in our terms of service and **privacy policy** our service is not to be used for illegal activity, and as a legitimate company ***we will cooperate with law enforcement if we receive a court order***"

Proxies and VPNs: Issues

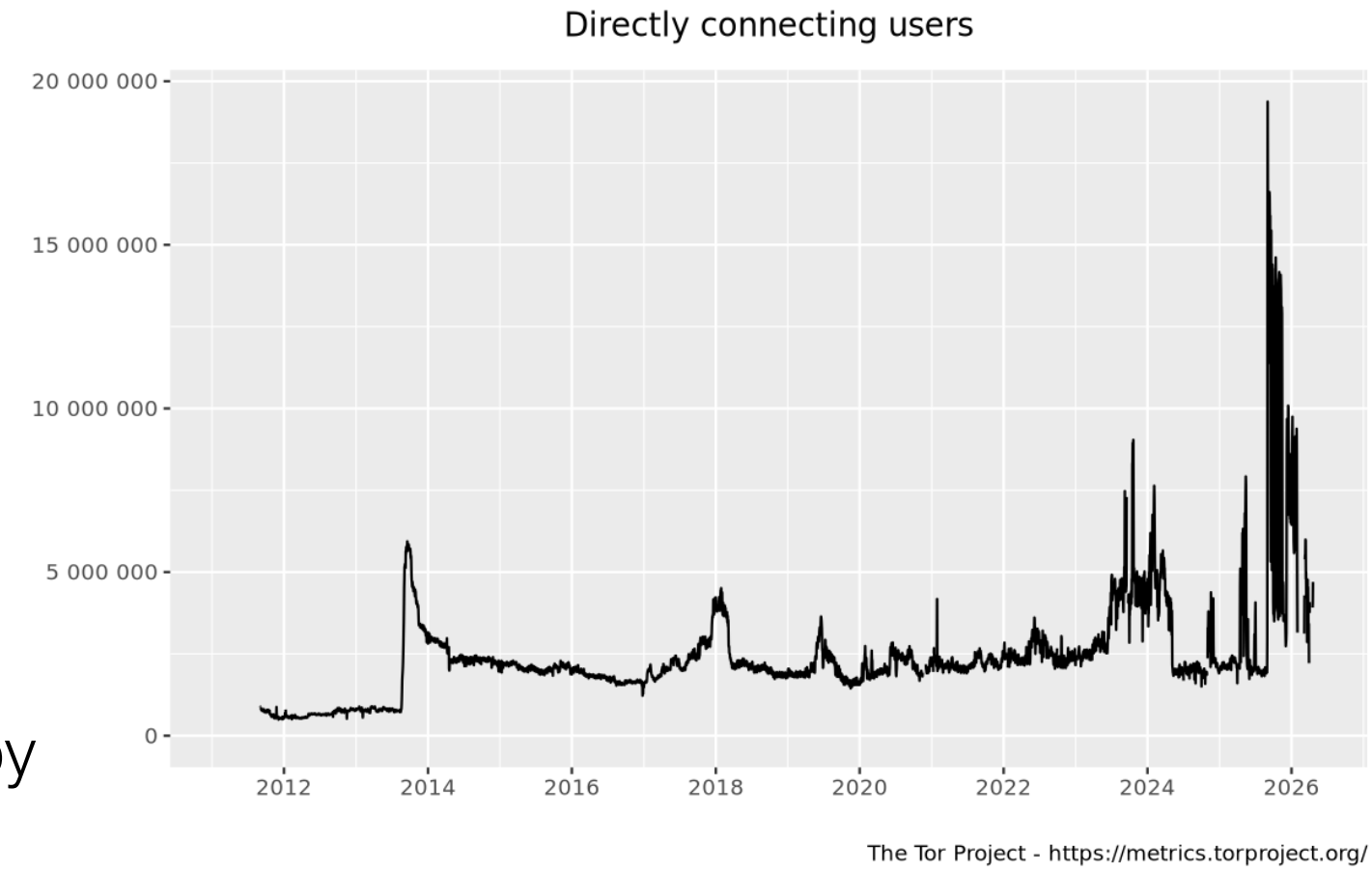
- Trusting the proxy
  - The proxy can see the sender and recipient's identities
  - Attackers might convince the proxy to tell them about you
  - The proxy itself could be an attacker!
- Performance
  - Sending a packet requires additional hops across the network
- Cost
  - VPNs can cost $80 to $200 per year

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses/Services
  - Onion Services
  - Tor in Practice



- Tor is a successful privacy enhancing technology that works at the transport layer
- Millions of active users.
- Provides anonymous TCP connections (conceals your and/or destination IP address)
- Originally developed in 90s by US military (!)



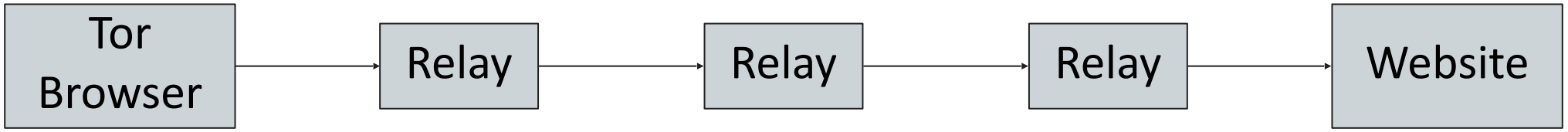
Tor ("The Onion Router")

Idea: Send the packet through multiple proxies instead of just one proxy

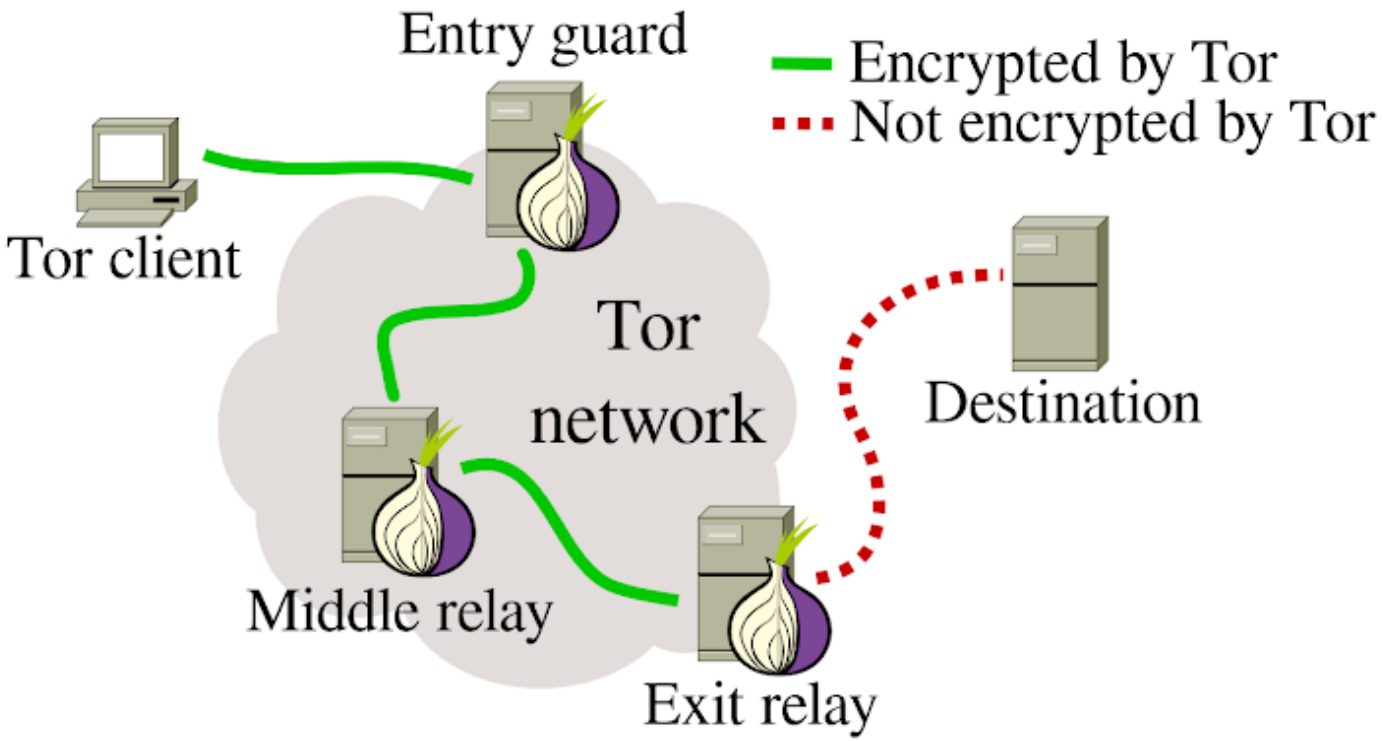
- **Tor:** A network that uses cryptography + multiple proxies (relays or "onion routers") to enable anonymous communications

Key components of Tor:

- Network of many **Tor relays (proxies)** for forwarding packets
- **Directory server:** Lists all Tor relays and their public keys
- **Tor Browser:** A web browser configured to connect to the Tor network
- **Tor onion services:** Servers that can only be reached through the Tor network
- **Tor bridges:** relays that try to hide the fact that a user is connecting to the Tor network



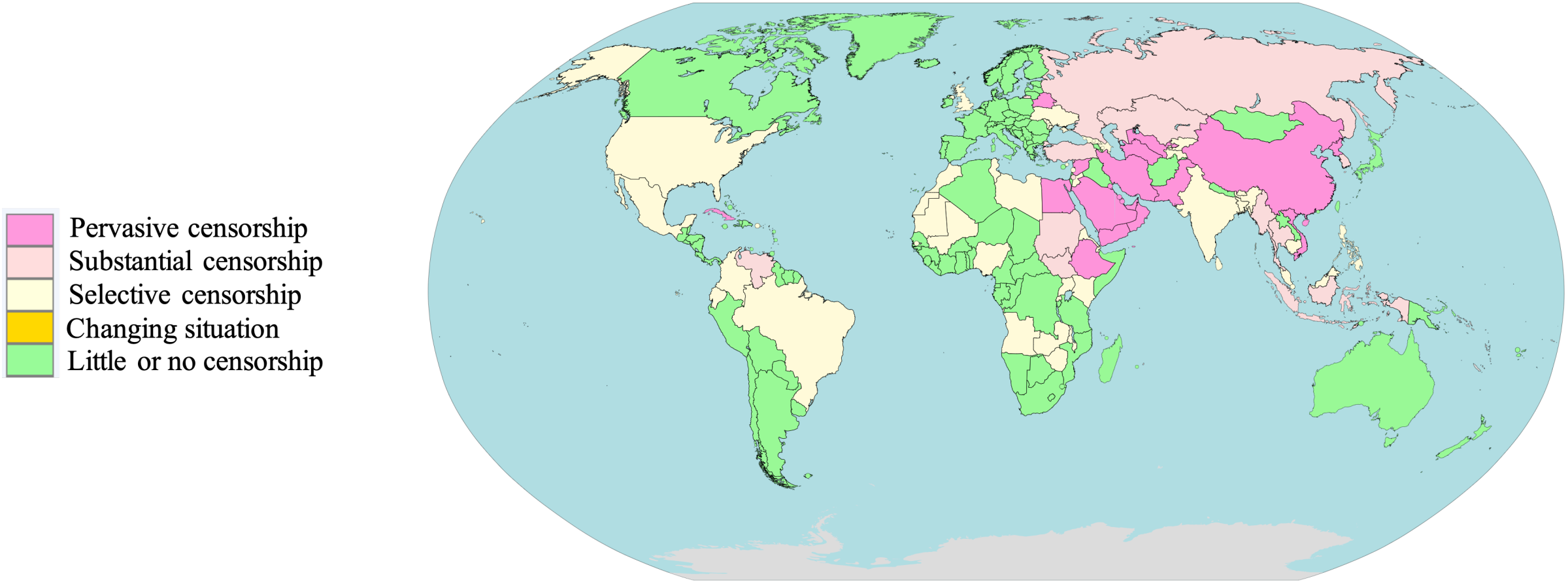
Tor (“The Onion Router”)



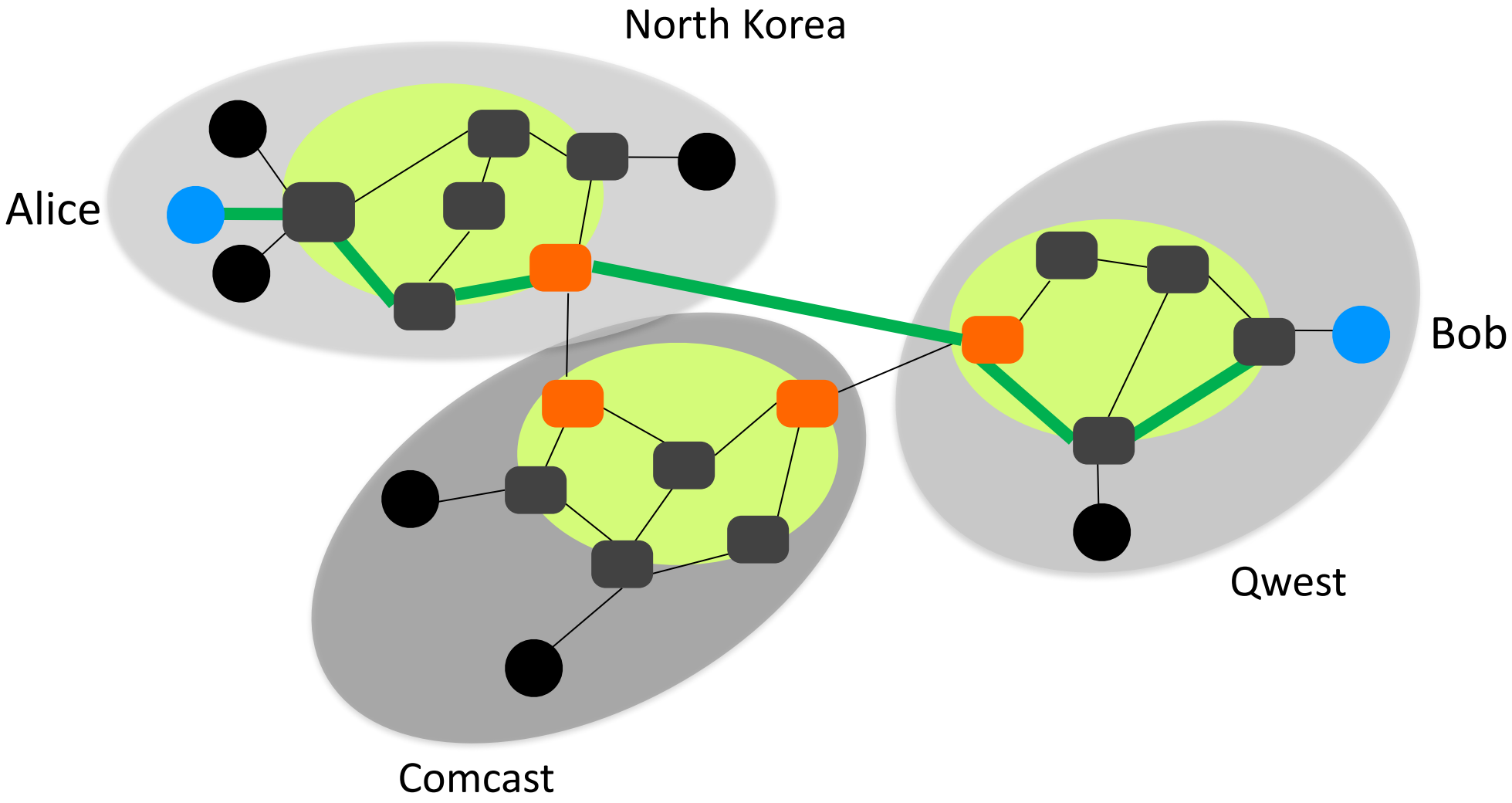
Tor Threat Model & Goals

- Security goals: Client anonymity and censorship resistance
  - Optional: Server anonymity with onion (hidden) services
- Tor preserves anonymity against local adversaries
  - Example: An on-path attacker sees Alice send a message to a Tor relay, but not the final destination of the message
  - Example: The server should not know the identity of the client based solely on network layer info
- Performance: Low latency (communication should be fast)

Internet Censorship



Example Censorship Threat Model

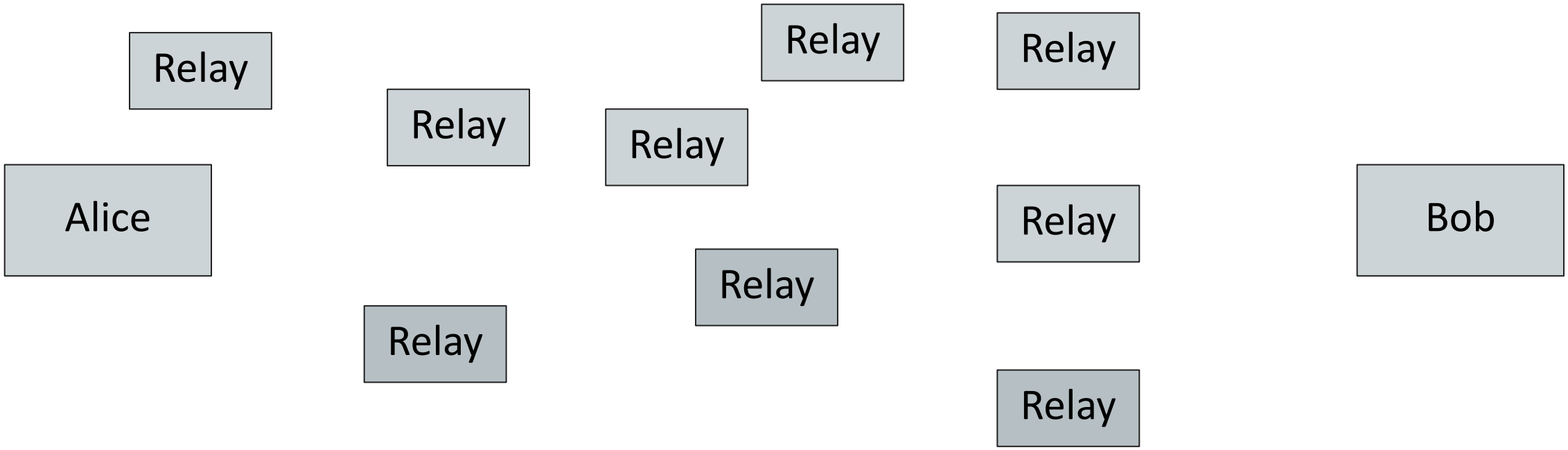


Tor Protocol: Tor Circuits

To communicate anonymously with a server, the Tor client forms a **circuit** consisting of 3 relays (by default)

1. Query the directory server for a list of relays (lists all relays & their PK)
2. Choose 3 relays to form a Tor circuit
3. Connect to the 1st relay, forming an end-to-end TLS connection
4. Connect to the 2nd relay *through* the 1st relay, using end-to-end TLS connection
5. Connect to the 3rd relay *through* the 2nd relay, using end-to-end TLS connection
6. Connect to the web server through 3rd relay using HTTPS (so an end-to-end TLS connection is formed through the third relay)

Tor Circuits: Walkthrough

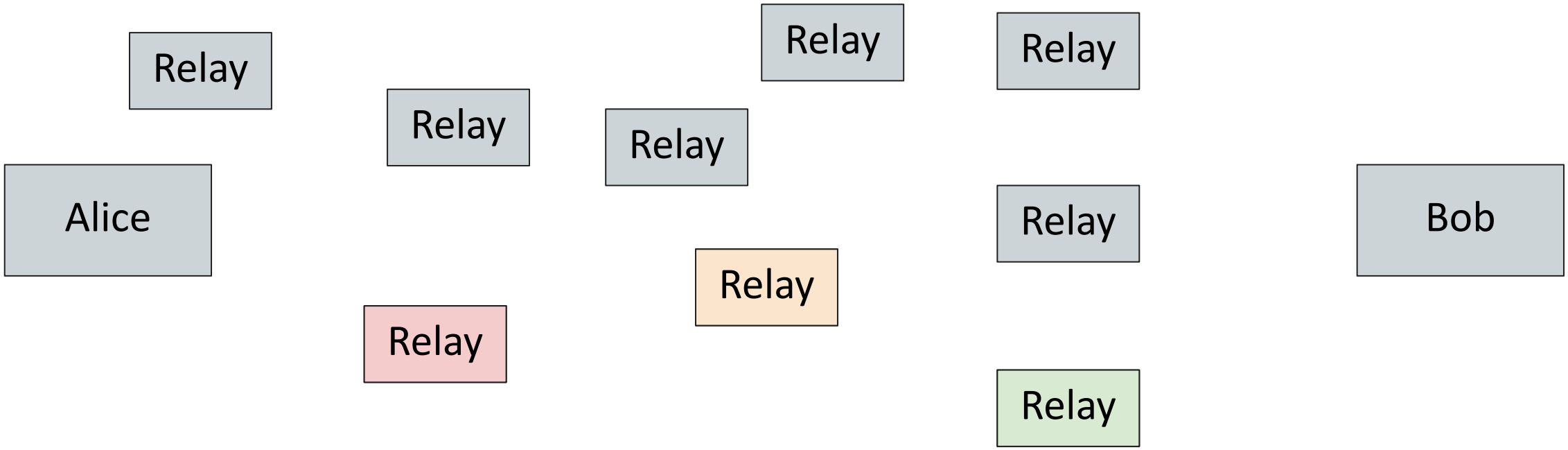


Tor Protocol: What a Relay Does

Runs a Tor relay application (software) that:

1. Listens for someone to initiate a TLS connection
2. When receiving a packet, decrypts using the key obtained through TLS (or encrypts if reverse direction)
3. Forwards the packet to its next hop / destination

Tor Circuits: Setup

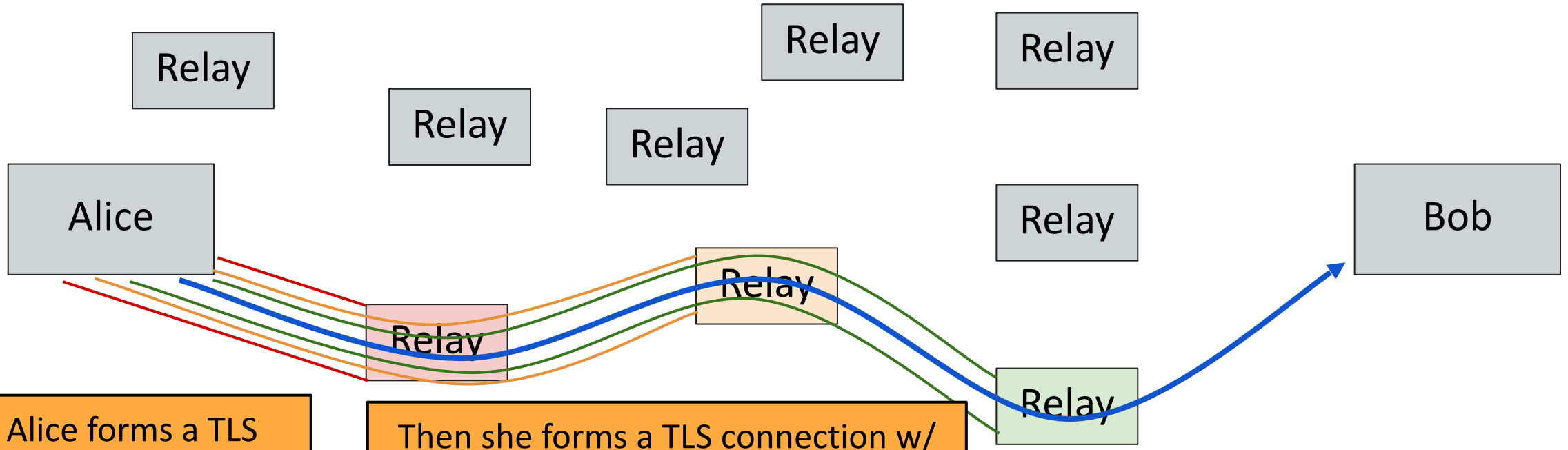


Suppose Alice wants to talk to Bob anonymously.

Alice queries Tor's directory servers and chooses 3 relays: Relay #1 (Entry Node), Relay #2, and Relay #3 (Exit Node)

The directory servers publish a public key for each Relay node

Tor Circuits: Setup



Alice forms a TLS connection w/ the entry node (Relay 1)

Then she forms a TLS connection w/ the 2nd node, through the 1st node

Note: Relay 1 is only relaying TLS packets. It doesn't know the contents of the packets!

Then she forms a TLS connection with the exit node (Relay #3), through the 2nd node

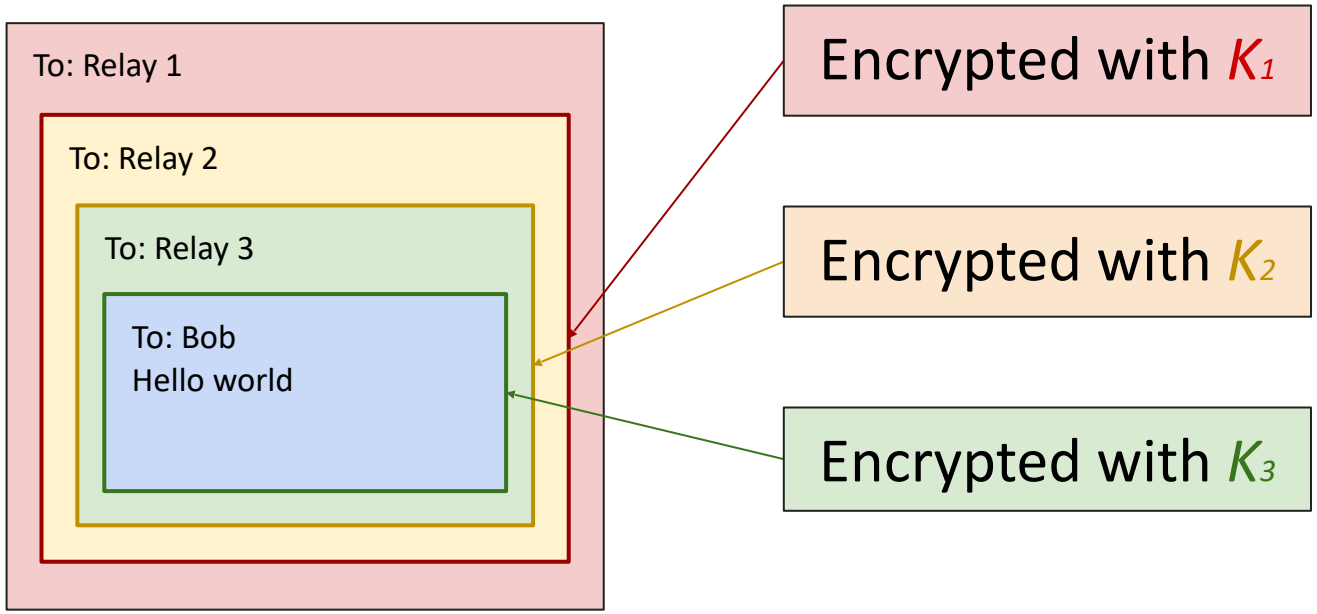
Finally, she opens a TLS connection with Bob via the exit node

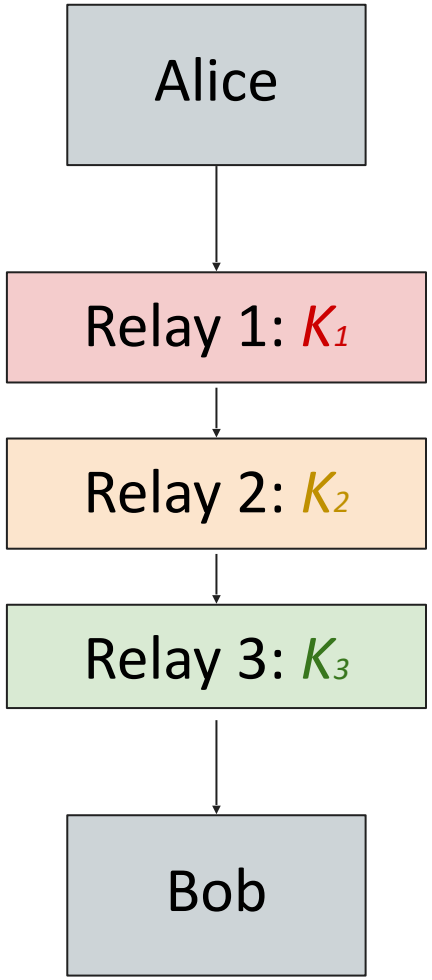
Tor Packet Construction

Wrap the packets via encryption: fixed size "cells" of 512 bytes

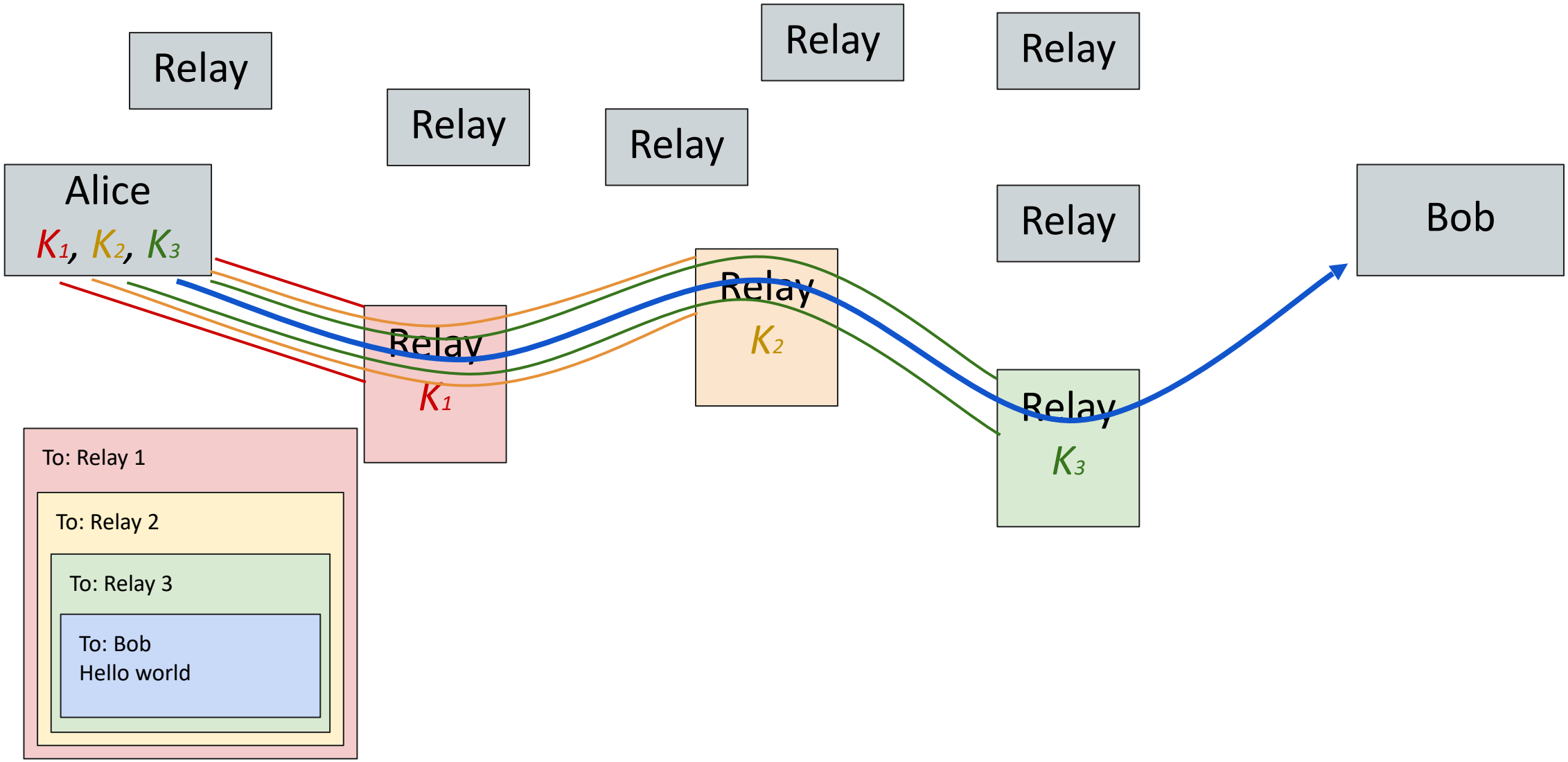
- e.g., the packet sent to Bob is encrypted under K_3 since Relay 3 is the one to forward that information to Bob

Ensures that no one can read or tamper with the messages, since these are all sent over TLS connections



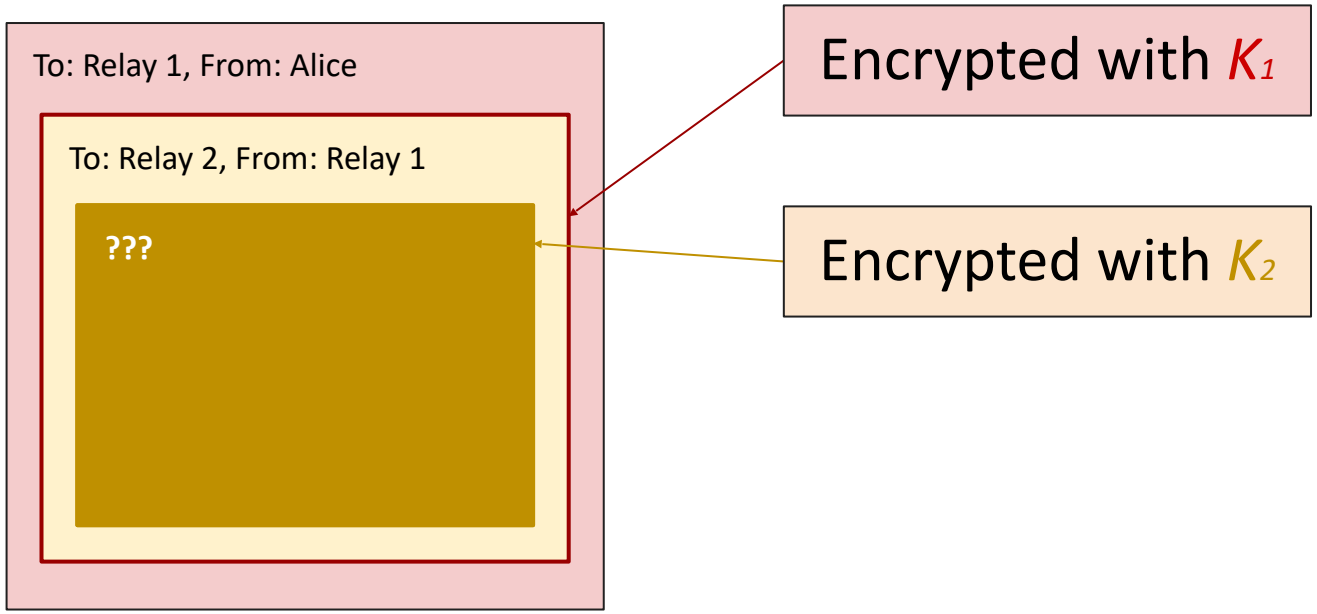


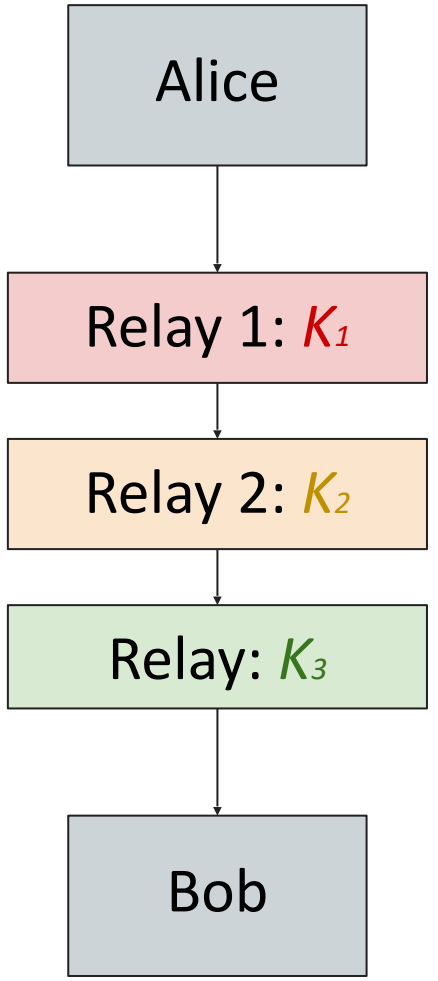
Tor Circuits: Walkthrough



Tor Packet Construction

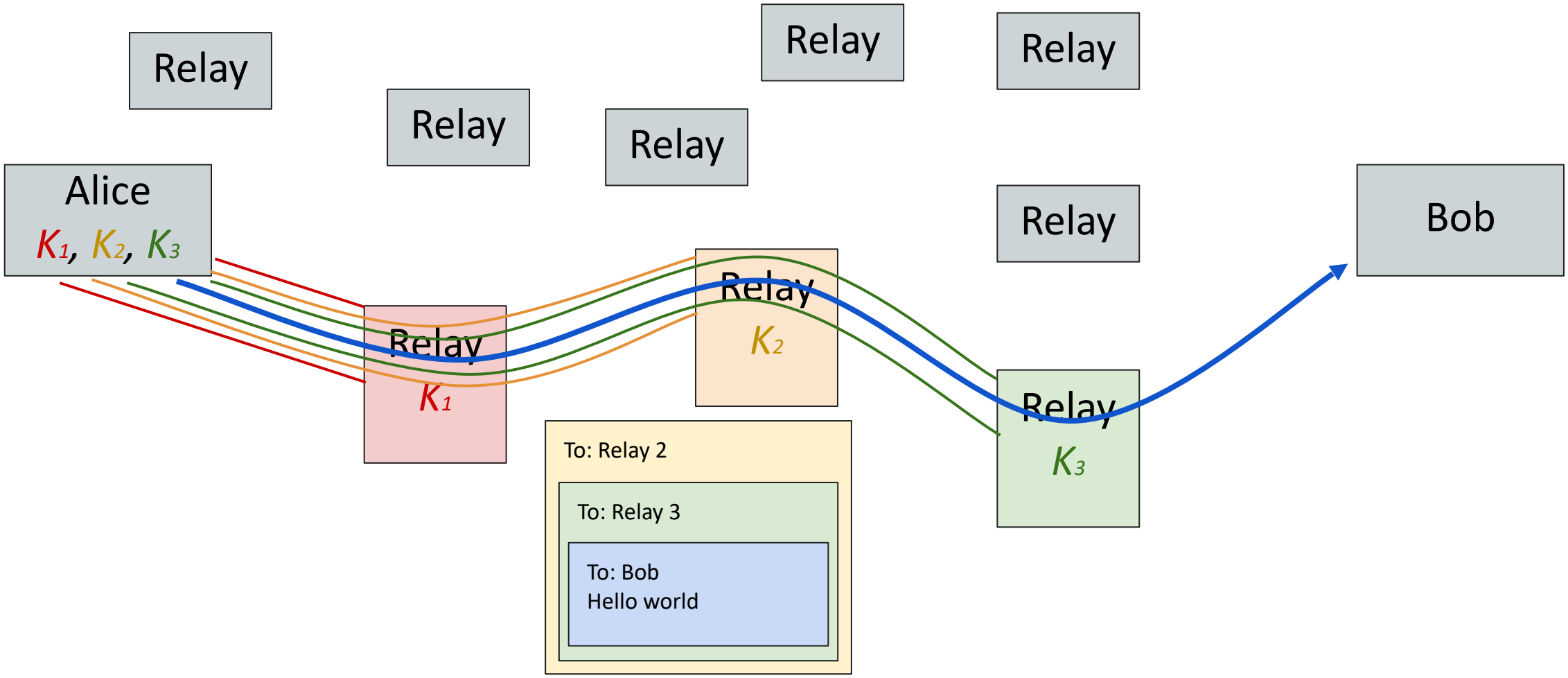
- What does Relay 1 see?





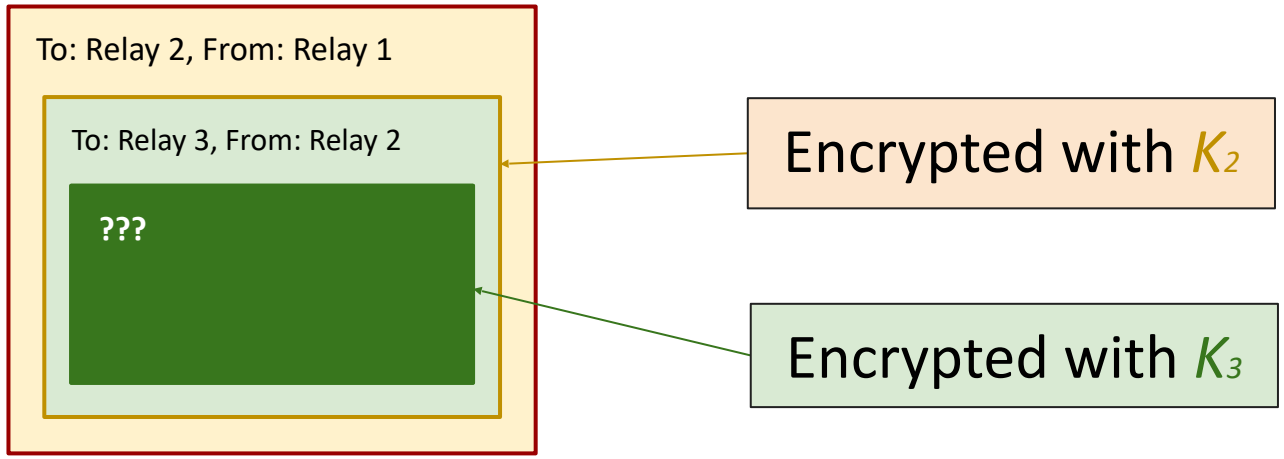
All Relay 1 knows is the message came from Alice and is going to Relay 2. They don't know Alice is talking to Bob!

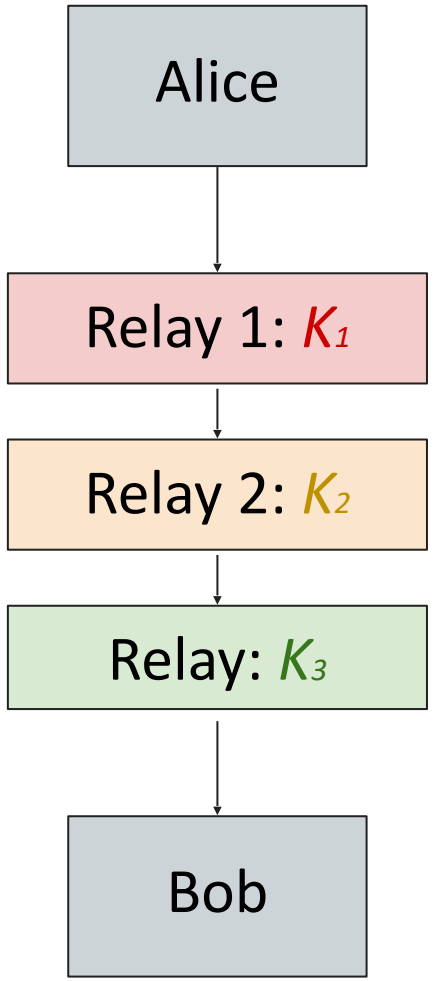
Tor Circuits: Walkthrough



Tor Packet Construction

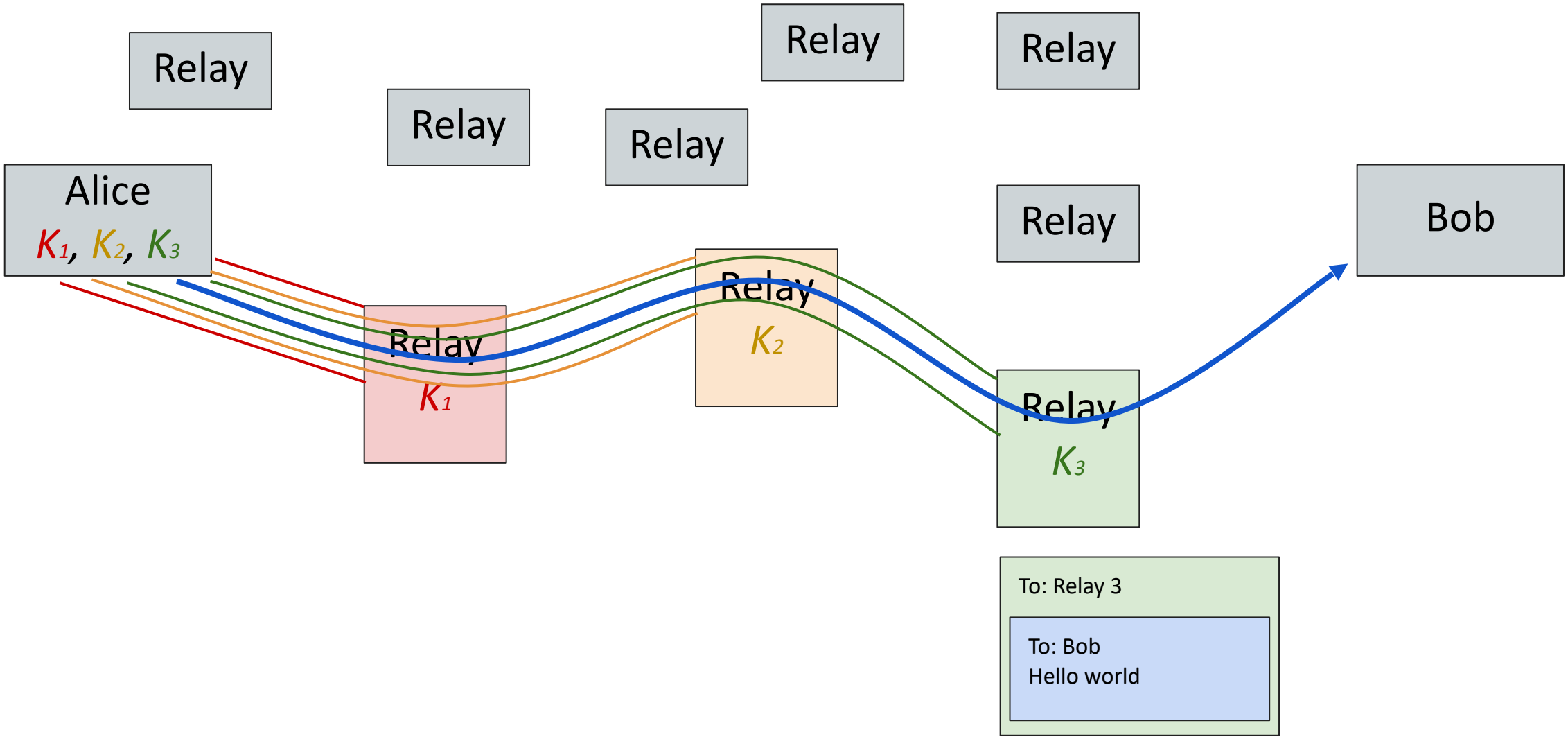
- What does Relay 2 see?





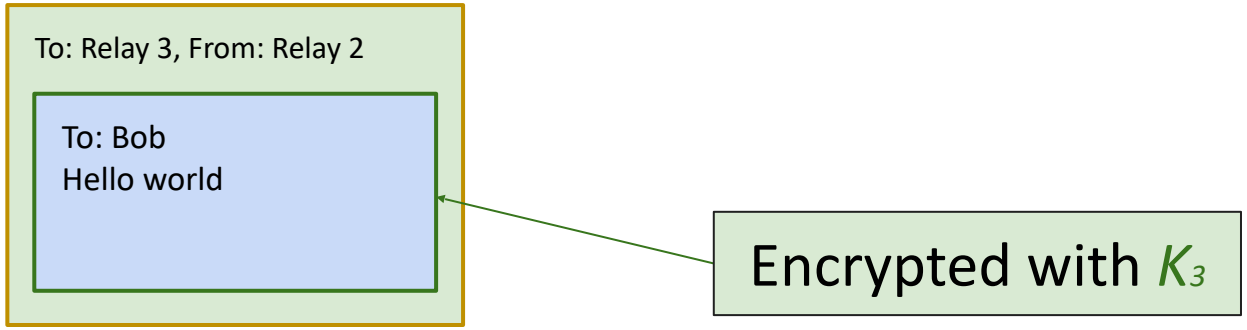
All Relay 2 knows is the message came from Relay 1 and is going to Relay 3. They know nothing about Alice and Bob!

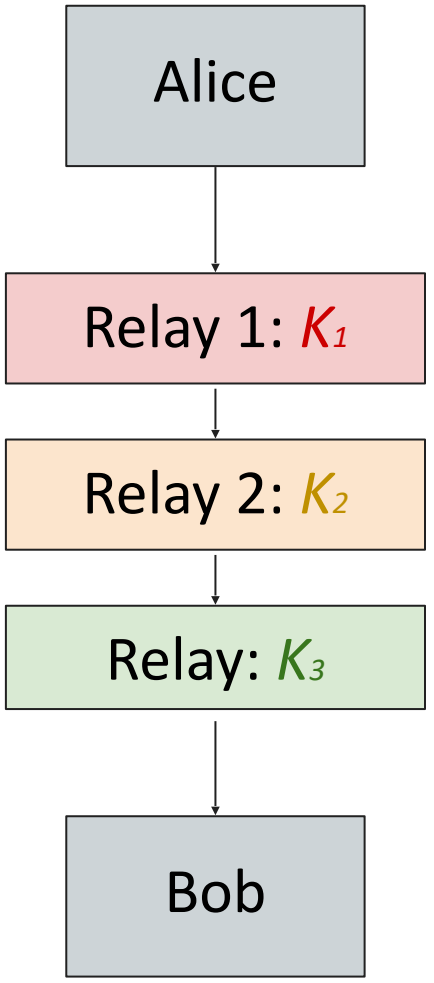
Tor Circuits: Walkthrough



Tor Packet Construction

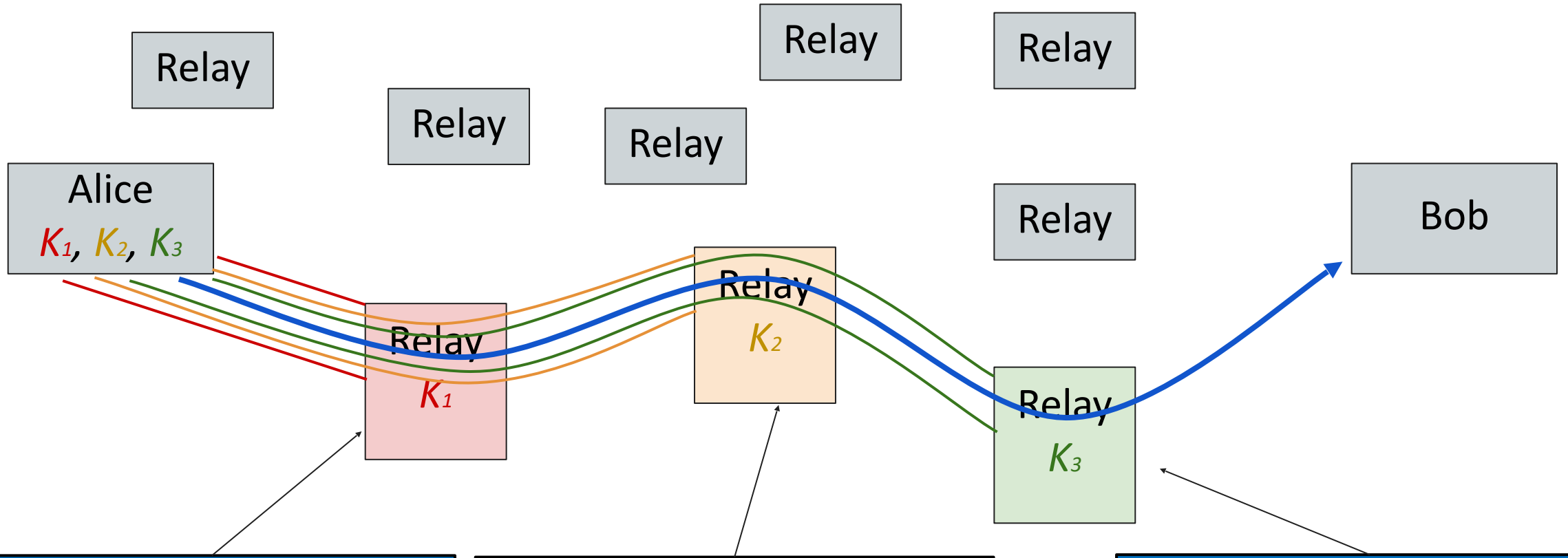
- What does Relay 3 see?





All Relay 3 knows is the message came from Relay 2 and is going to Bob. They don't know Alice sent the message!

Tor Circuits Privacy: Review



Relay 1 knows that Alice is using Tor but *not* who Alice is talking to

Relay 2 knows nothing (other than that someone is using Tor)

Relay 3 knows that someone is talking to Bob, but not who

Tor Exit Nodes

- The exit node can see the message and the recipient (but not the sender)

- The exit node is a man-in-the-middle attacker
  - If the user is not using encryption (TLS) to connect to the end host, the exit node can see and modify the traffic
  - If the user is using TLS (using HTTPS), the exit node cannot see or tamper with the traffic

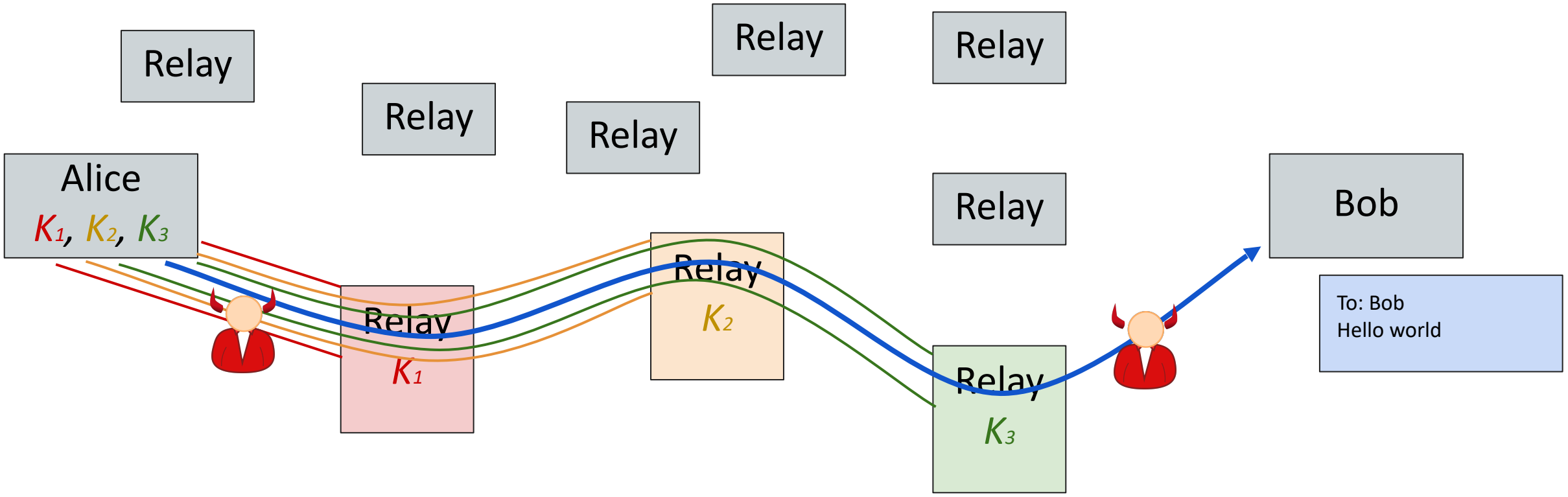
Tor Exit Nodes in Practice

- Administrators of Tor exit nodes often receive abuse complaints
  - Users complain to the exit node
  - Users complain to the Internet service provider (ISP), which complains to the exit node
  - Legal problems: illegal activity traced to exit node first
- As a result, most Tor relays choose to only be entry or intermediate nodes, not exit nodes
  - Exit node bandwidth is the bottleneck in Tor, not internal bandwidth

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses
  - Onion Services
  - Tor in Practice

Tor Weaknesses: Timing Attacks (Side Channel)



- Assume on-path adversary can see data flow from Alice and to Bob
- Observe when Alice sends a message, when Bob receives a message, and link the two together

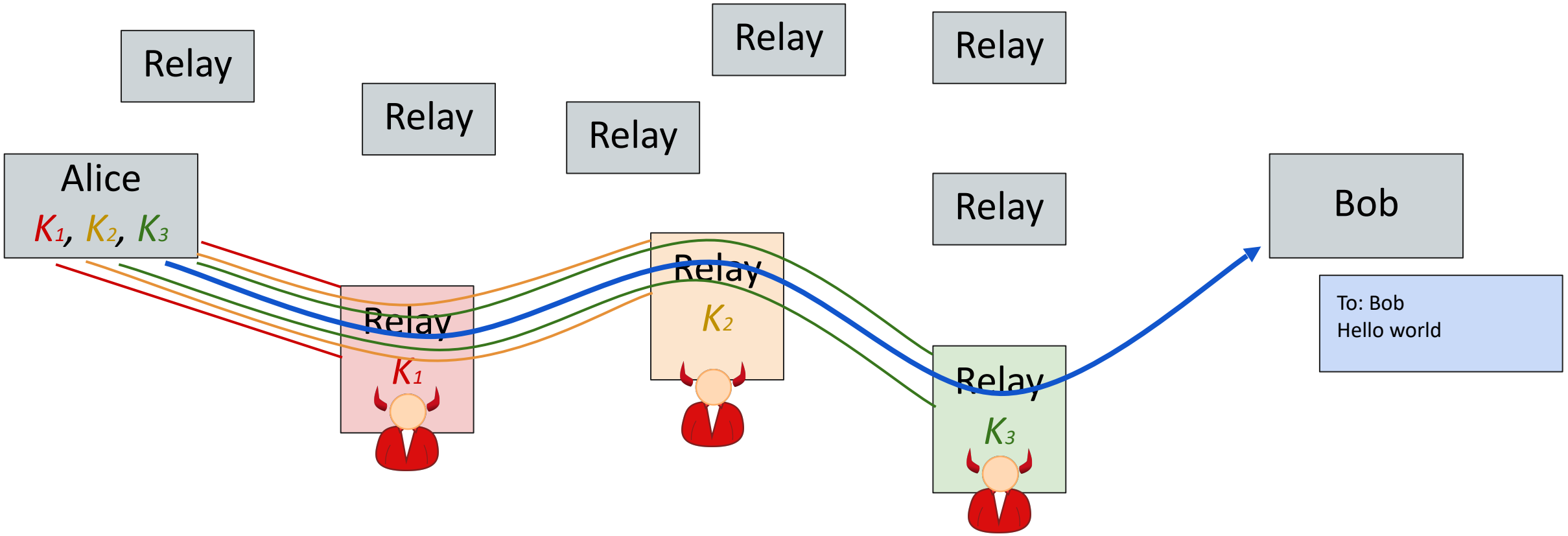
Tor Weaknesses: Timing Attacks (Side Channel)

A network attacker who has **a full (global) view** of the network can learn that Alice and Bob are talking

Global adversaries are *outside* of Tor's threat model and are not defended against

- Tor only defends against local adversaries with partial views of the network
- Timing attacks could be defended against by delaying the timing of packets, but would lead to poor/unusable performance

Tor Weaknesses: Collusion



- If all three nodes are compromised, then traffic can be linked

Tor Weaknesses: Collusion

- **Collusion**: Multiple nodes working together & sharing info
  - If *all* nodes in the circuit collude, anonymity is broken
  - If *at least one* node in the circuit is honest, anonymity is preserved
  - But an attacker might create hundreds of nodes in the Tor network!

- Defense: The more nodes we use, less likely they are not all colluding
  - It's much harder for 10 nodes to collude than for 2 nodes to collude
  - 3 nodes is generally considered good enough & is the default

Tor Weaknesses: Collusion Defense

Defense: **Guard nodes**

- Guard nodes must have a high reputation and must have existed for a long time
- Clients will always use a guard node as the entry node (by default) & the same guard node is used for a long period of time
  - Attackers' nodes are unlikely to become guard nodes
  - Because clients use the same guard nodes for a long period of time, there is only a low chance that the client will switch to an attacker's guard node
  - Censors can block IPs of guard nodes (which are publicly known)

Tor Weaknesses: Distinguishable Traffic

- Tor does *not* hide the fact that you are using Tor
  - Example: A local adversary can see that you are sending packets to a Tor relay
  - Tor directory publishes all relay nodes for any client

- Anonymity only works in a crowd
  - Example: A Harvard student sent an anonymous bomb threat using Tor. The administrators noticed that only one student on the Harvard network used Tor at that time!

Tor Weaknesses: Distinguishable Traffic

Defense: **Tor bridges**

- Attackers can tell you are using Tor because they can see you are connecting to an entry node
- **Tor bridges**: entry nodes that are not available on public lists
  - Users request bridges from a separate directory, which only gives a few bridges to a user
  - Prevents attackers from enumerating all bridges unless they have many different IP addresses running Tor clients

Tor Weaknesses: Distinguishable Traffic

With Tor bridges, censors can no longer block Tor based on IP addresses of entry/relay nodes

- But they can still distinguish traffic that looks like Tor traffic from normal traffic (fixed sized packets with TLS)

Defense: **Pluggable transports**

- Pluggable transports change the appearance of the client's traffic to the entry node (only for bridges)
- Obfuscates the encrypted traffic to make it "look" more like normal Internet traffic (no longer obvious fixed size packets)

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses/Services
  - Onion Services
  - Tor in Practice

Tor Hidden (Onion) Services

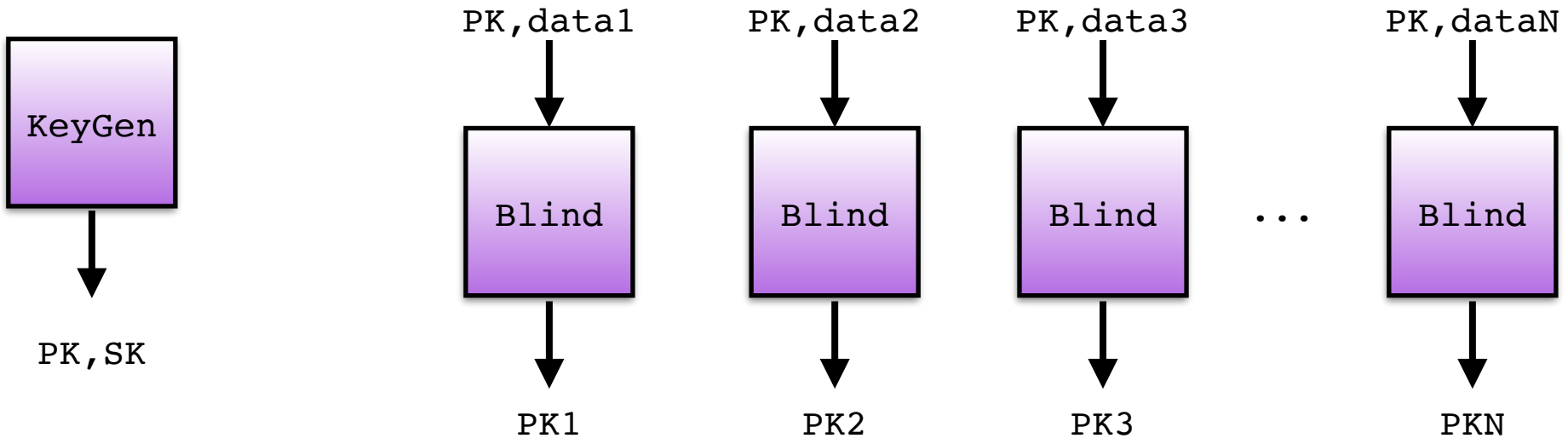
- Sometimes, the *server* wants to be anonymous, so no one knows where the server is located
- Tor onion services: Websites that are only accessible through the Tor network
  - Gives the server anonymity protection
  - Sometimes called the dark web
- Idea: Route the server's traffic through the Tor network so that no one knows who the server is

Tor Onion Services

- Connecting to onion services is a little more complicated:
  - Client has to know where to send packets, but server is trying to be anonymous
- Main idea: Use a rendezvous point – a relay node that will connect two circuits from different directions

Tool for Onion Services: Blinded Public Keys

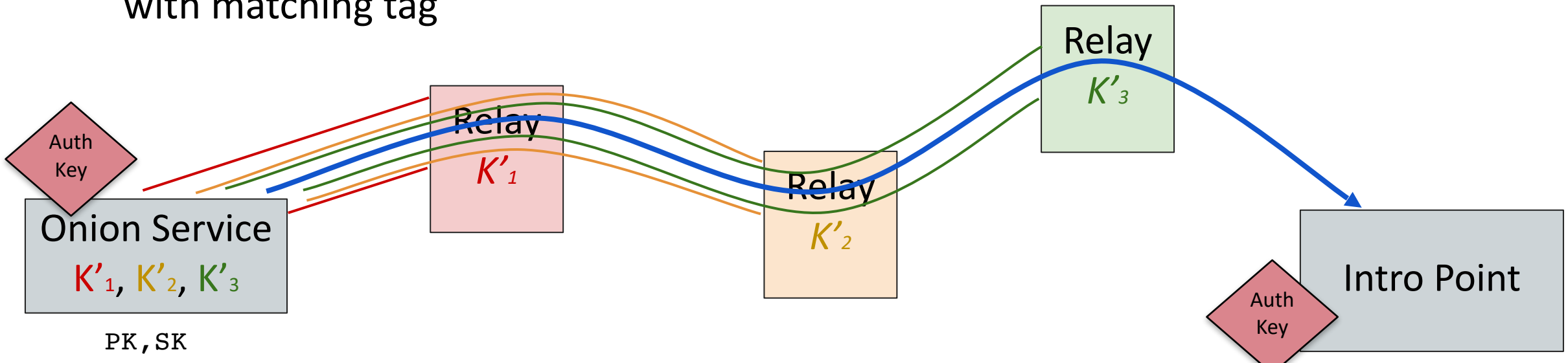
- Normally, the public-key for encryption is fixed
  - This is the (N,e) value in RSA, for example
- Tor Onion Services use public-key encryption scheme with “blinded” public-keys



- Same `SK` works for all of `PK1`, `PK2`, ..., `PKN`
- But all of `PK1`, `PK2`, ..., `PKN` look like unrelated keys!

Onion Service Introduction Points

- An onion service chooses several random relays to be “Introduction Points”
- Service maintains long-running connection with Intro Point, awaiting callbacks
- Intro points do not know identity of the service: Service only sends a random key specific to the service and the intro point
- Intro points await clients, who present a service tag and get forwarded to service with matching tag

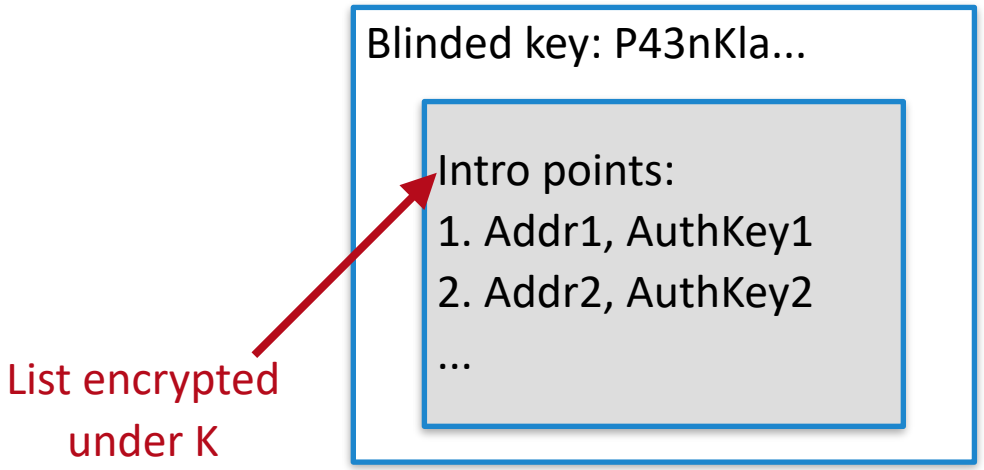


Onion Service Addresses and Descriptors

A **.onion address** is not served by DNS; It is like a key that is communicated out of band

- Example: zqktlwiuavvvqqt4xqcuuqe5hk6r4cvyq5xqyyl272y43gsqsbjcdgyd.onion
- From a .onion, clients can derive a key (called K below)

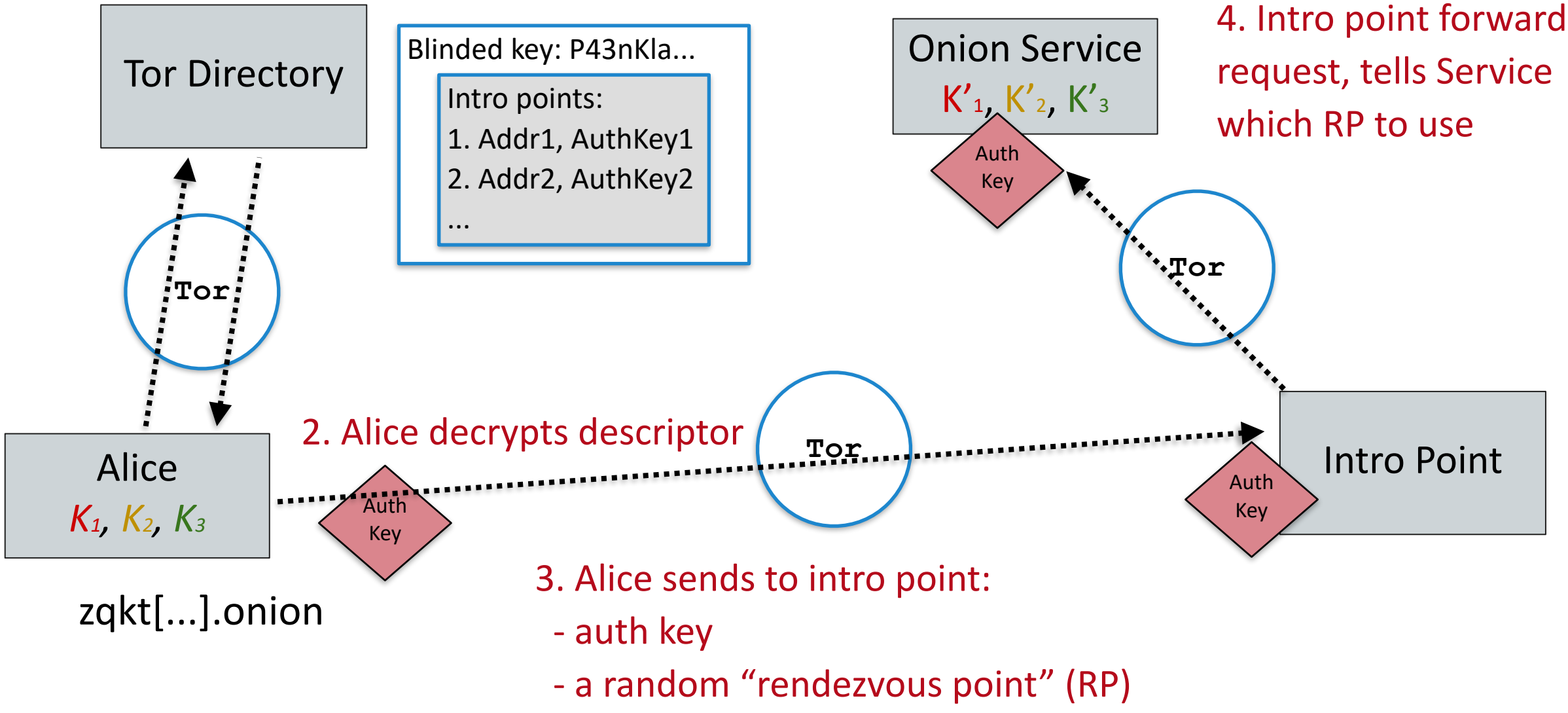
A **.onion service descriptor** is a short file held by directory servers in Tor.



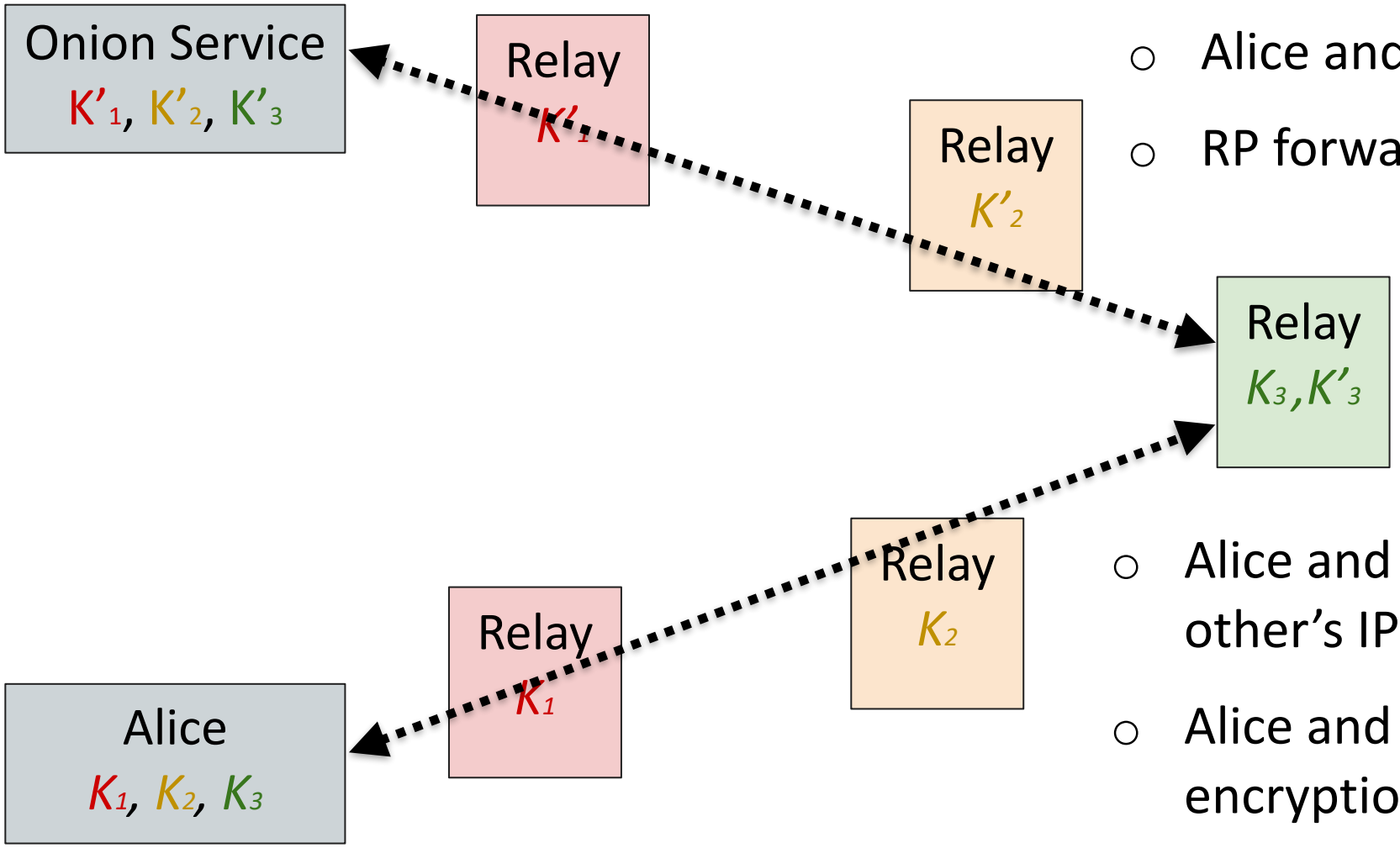
- Directory server can't recognize service
  - Key is blinded (with current time, etc)
  - List is encrypted under K from .onion, which it doesn't have
- Overwritten with new blinded key every 3 hours

Connecting to an Onion Service

1. Alice computes current blinded key and gets descriptor



Communicating with Onion Services



- Alice and Service create circuits to RP
- RP forwards traffic to link Tor circuits

- Alice and Service never see each other's IP addresses
- Alice and Service apply 4th layer of encryption (that RP can't decrypt)
- RP does not see either endpoint or cleartext

Onion Services: Wrap-Up

- .onion addresses at as a secret identifier to find service
- Keys are blinded frequently and stored in different directory servers
  - If directory does not know .onion, it can't link blinded keys
- Introduction points act as "dead drops" for clients to request connections
  - Intro points only see random ID for service (which changes every three hours)
- Rendezvous points act as entry/exit node to link circuits
- Goal: No party can link actual service to an IP, or identify users who connect

Outline

- Privacy vs. Anonymity vs. Confidentiality
- Proxies & VPNs
- Tor
  - Overview & Design
  - Tor: Attacks & Additional Defenses/Services
  - Onion Services
  - Tor in Practice

Tor Tradeoffs

Benefit: Free to use

- Tor is mostly funded by the US government
- Users “pay” by providing traffic for other users to hide in (recall: you don’t want to be the only user on the network using Tor)

Drawback: Performance

- Latency is significantly worse: Packets need to make more hops across the network

Drawback: Full anonymity requires usability tradeoffs

- No cookies by default (even final server doesn’t know you)
- They even recommend keeping the browser window size constant, which can be annoying!

Internet Censorship & Tor

- Government censors
  - Block websites containing “offensive” content
  - Commonly employ blocklist approach
- Because Tor hides the sites a user is connecting to, it is useful & popular for bypassing censorship
  - Functions similarly to bypassing censorship using a VPN or proxy
- Problem: Constant arms race between Tor & censors

Arms Race: Tor vs. Censorship

- Censors can easily block access to all public Tor entry points
  - Bridge services provide a set of entry points that aren't listed publicly anywhere, so they can't be blocked by IP
- Censors can block traffic that looks like Tor traffic
  - Pluggable transports make traffic look more like normal web traffic
- Censors can pretend to be a Tor client to see if a host is a Tor entry/bridge node & then block connections to it
  - Some pluggable transports use cloud services (like Google Cloud Platform, Amazon Web Services, etc.): harder to block

Hosting Illegal Services on Tor

Tor onion services often used for services widely considered illegal around the world

- Legitimate hosting services like Cloudflare refuse to host these services
- Most countries will take legal action against these services if hosted on regular web

Dark markets: Marketplaces for buying and selling illegal goods

- Transactions processed with a censorship-resistant currency like Bitcoin
  - Services like PayPal will refuse to process illegal transactions
- Ratings system with mandatory feedback
- Escrow service to handle disputes between sellers and buyers
- Can only be accessed as a Tor onion service

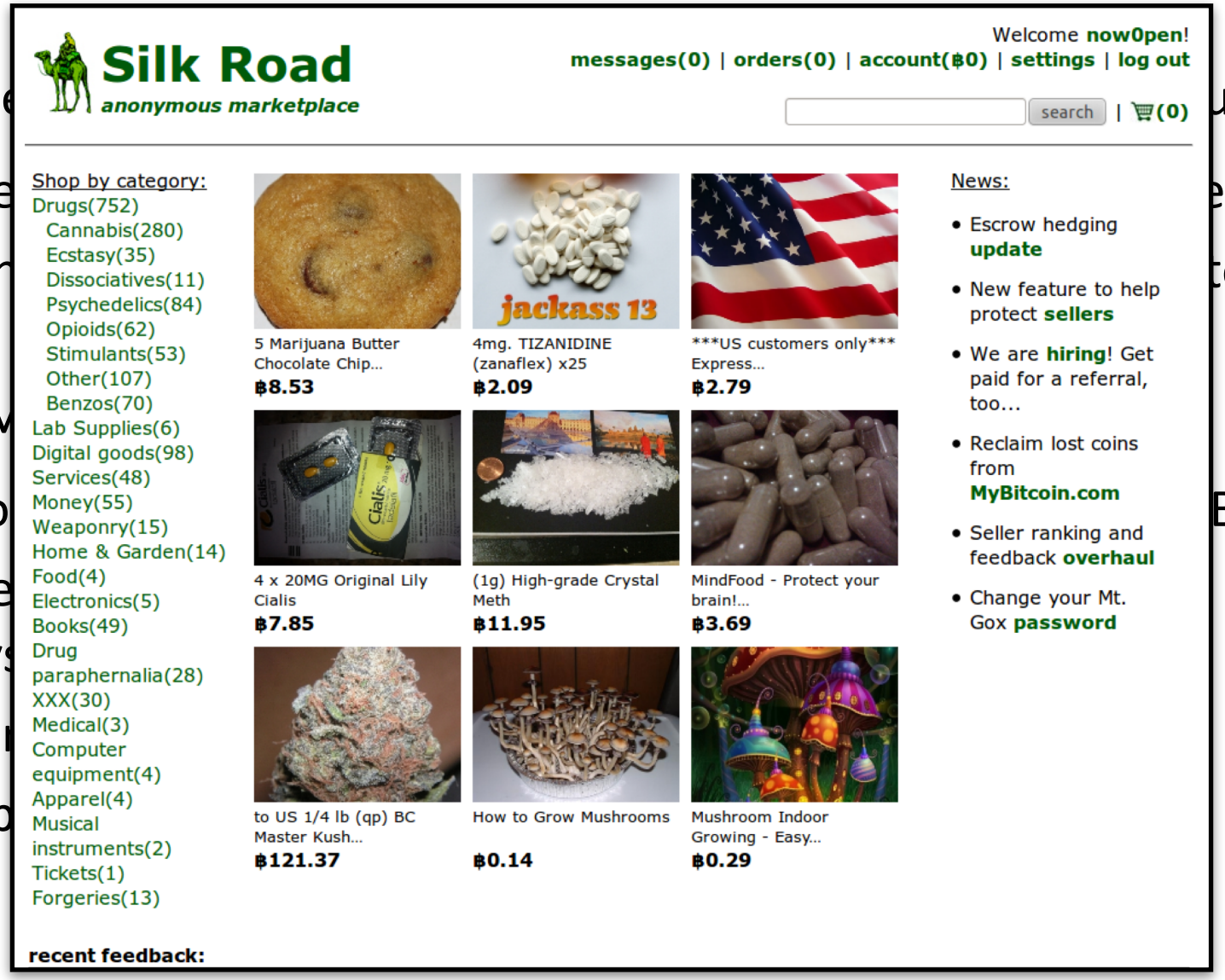
Hosting Illegal Services on Tor

Tor onion service… …und the world

- Legitimate… …ervices
- Most coun… …ted on regular web

Dark markets: M…

- Transactio… …Bitcoin
  - Service…
- Ratings sy…
- Escrow se…
- Can only b…



Modern Dark Markets

Hard to find information about where dark markets are located

- Legitimate websites (e.g., Reddit) will remove dark market links
- Legitimate websites with information about dark markets (e.g., DeepDotWeb) get taken down
- Information about dark markets is usually available through Tor onion services (e.g., Dread, a Reddit clone)

Review: Anonymity, Proxies vs. Tor

- Anonymity (concealing one's identity) can be difficult to achieve on the web
  - Different from standard confidentiality
- Proxies and VPNs relay traffic through a single machine: weak anonymity
  - The proxy knows who you are and what you are doing: not anonymous!
- Tor encrypts & routes your traffic through multiple machines
  - Circuits are established by performing TLS handshakes with three nodes, nested onion of encryption (no one knows full end-to-end)

Summary: Tor Weaknesses

Tor has weaknesses that are exploited

- **Weakness**: Timing attacks + global adversaries (not defended)
- **Weakness**: Collusion between nodes can deanonymize users by working together
  - Defense: Guard relays & multiple relays in circuits
- **Weakness**: Tor traffic is distinguishable from normal traffic, allowing it to be censored and blocked
  - Defense: Bridges and pluggable transports
- **Worse performance** & Tor itself/usage sometimes has poor reputation

Summary: Onion Services and Tor in Practice

Onion services provide anonymity for the server, in addition to the client

Tor in practice

- Often used to evade censorship -- Tor and censors are in a constant arms race
- Illegal services often use Tor because it conceals their identity from authorities