
DTU

02233 – Network Security

Week 9: Private Communication

Carsten Baum
Associate Professor
cabau@dtu.dk

Schedule for today

Questions and Recap

1. Privacy, Anonymity & Pseudonymity
2. Anonymous Connections with Onion Routing
3. End-to-End Encryption
4. Private Messaging with the Signal Protocol

Question 1

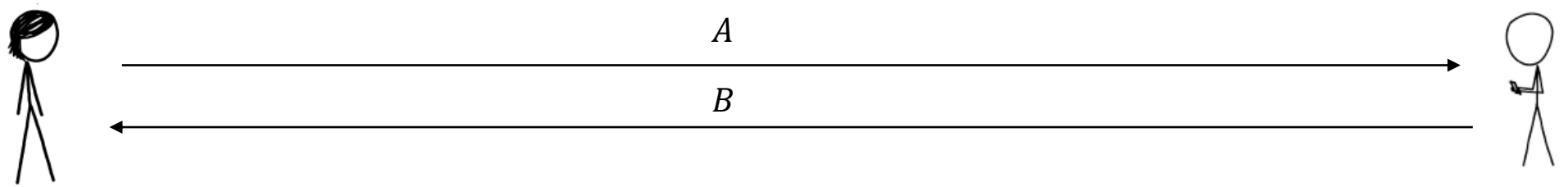
True or false

The Diffie-Hellman key exchange allows 2 parties to agree on a key, starting from preshared secrets.

Diffie Hellman key agreement

Fix a large group G of prime order, generator $g \in G$
It must be hard to compute a given G, g, g^a

} Public information!



1. Choose random $a \in \{0, \ldots, |G| - 1\}$
2. Compute $A = g^a$

3. Output $k = B^a$

1. Choose random $b \in \{0, \ldots, |G| - 1\}$
2. Compute $B = g^b$

3. Output $k = A^b$

Question 2

True or false

If two parties re-use the same DH secrets a, b through multiple key agreement rounds to generate session keys, then anyone stealing a or b in the future can decrypt messages from the past encrypted under the session keys.

Question 3

Which statement/s is/are true?

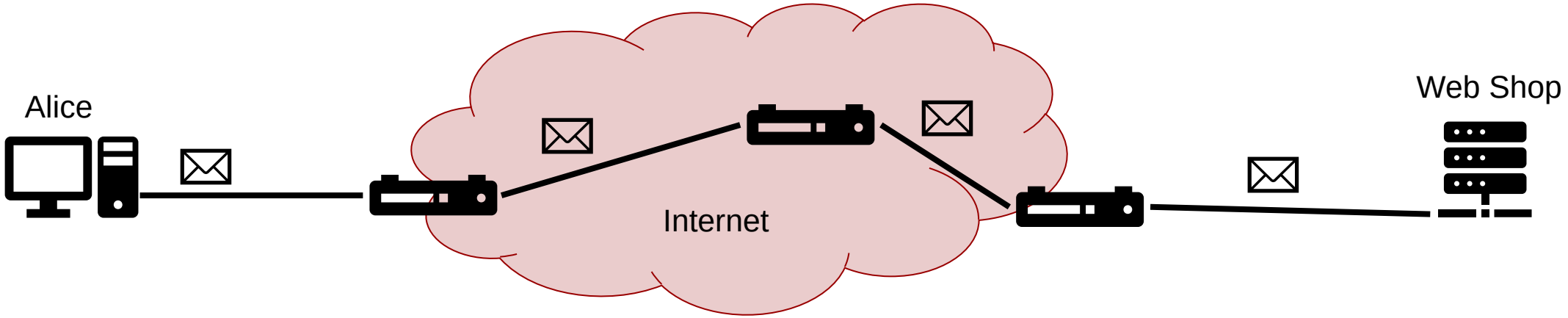
If an attacker can see traffic on a network, does TLS protect

1. Anonymity of the sender
2. Anonymity of the receiver
3. Confidentiality of the messages

Anonymity, unlinkability, pseudonymity,…

The terms of privacy

There’s more to privacy than confidentiality



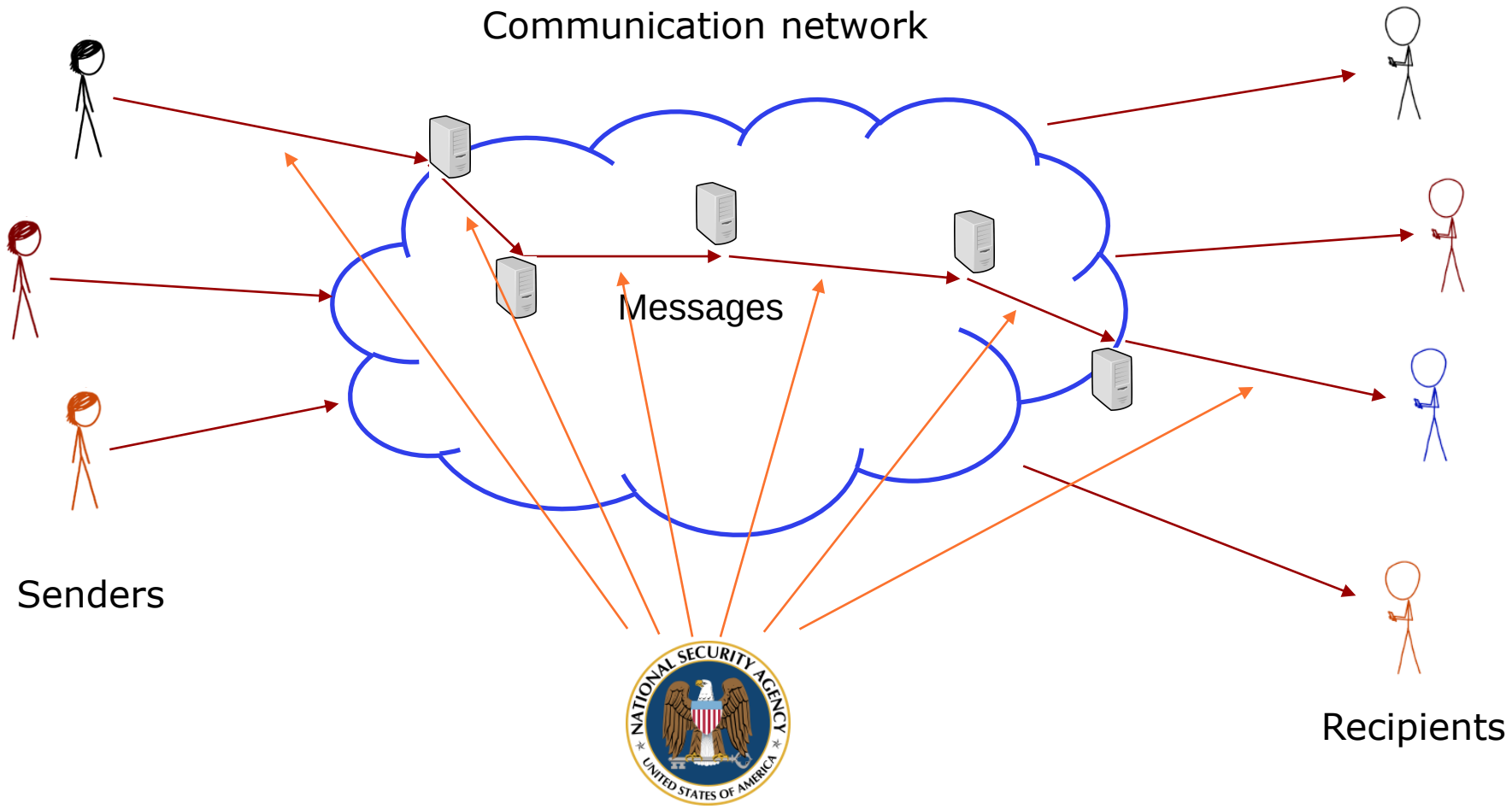
We want that attacker cannot read our messages in transit. ➡ Confidentiality

<u>More demands</u>

Attacker on internet cannot see who we are
Attacker on internet cannot see who we talk to
Receiver does not know who it communicates with

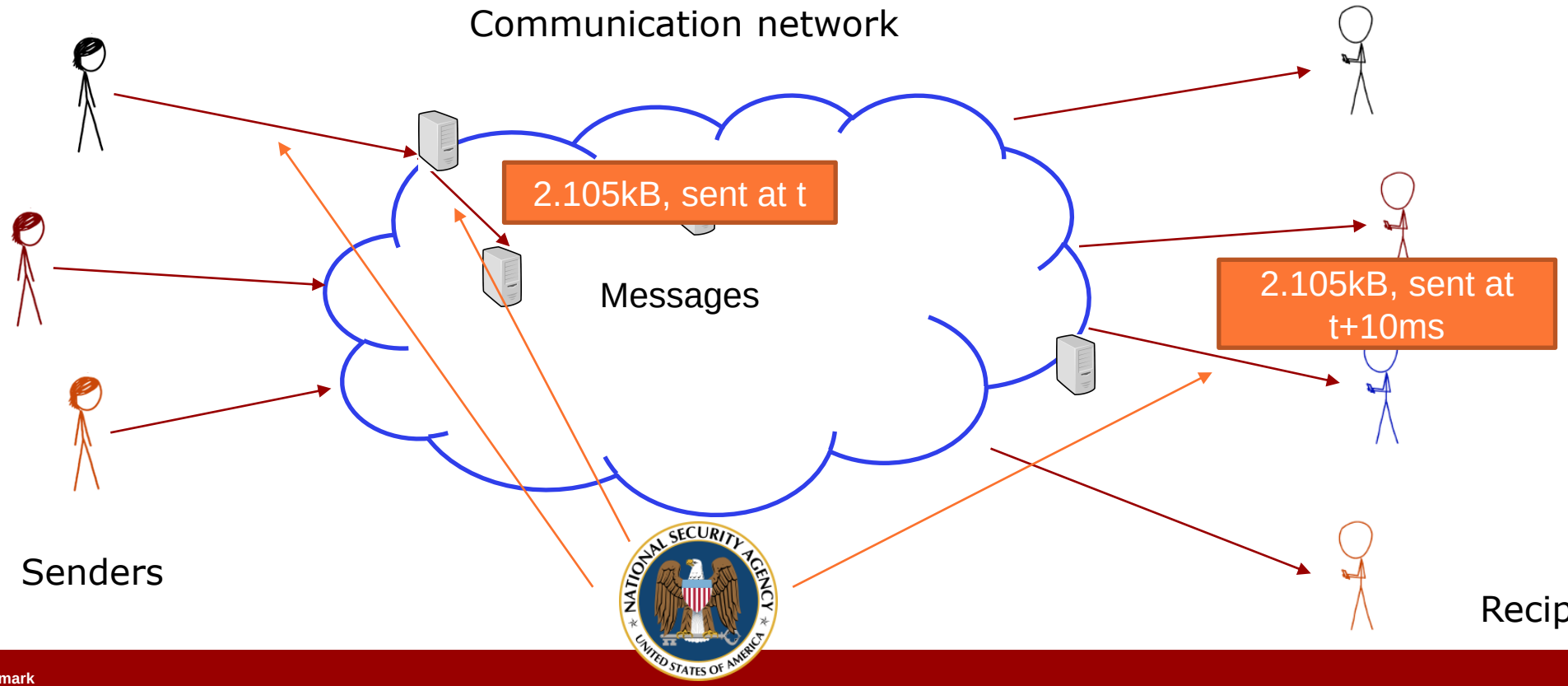
} Anonymity

Anonymity in Communication

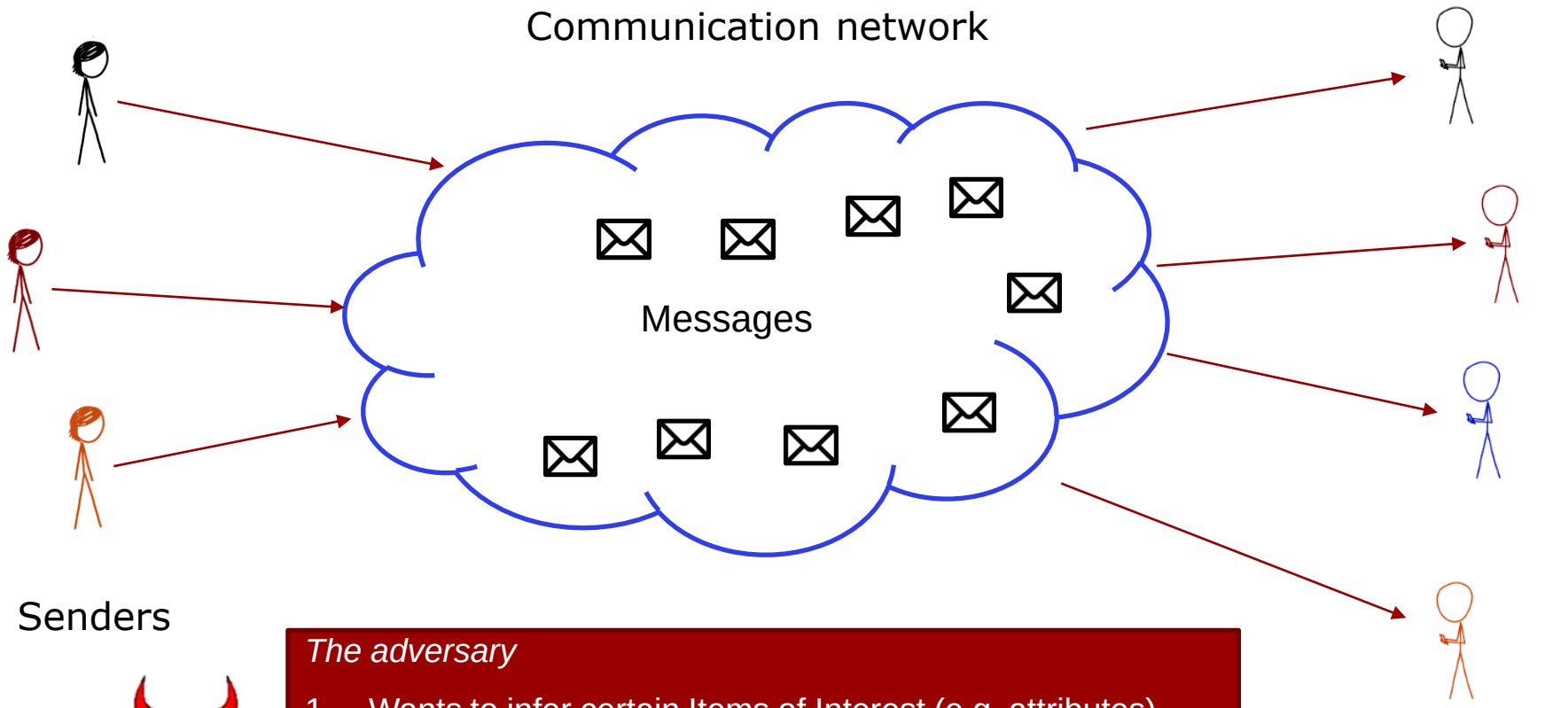


Traffic analysis (Metadata)

Metadata (who sent a message to whom, when and of which size) can be sufficient to break anonymity



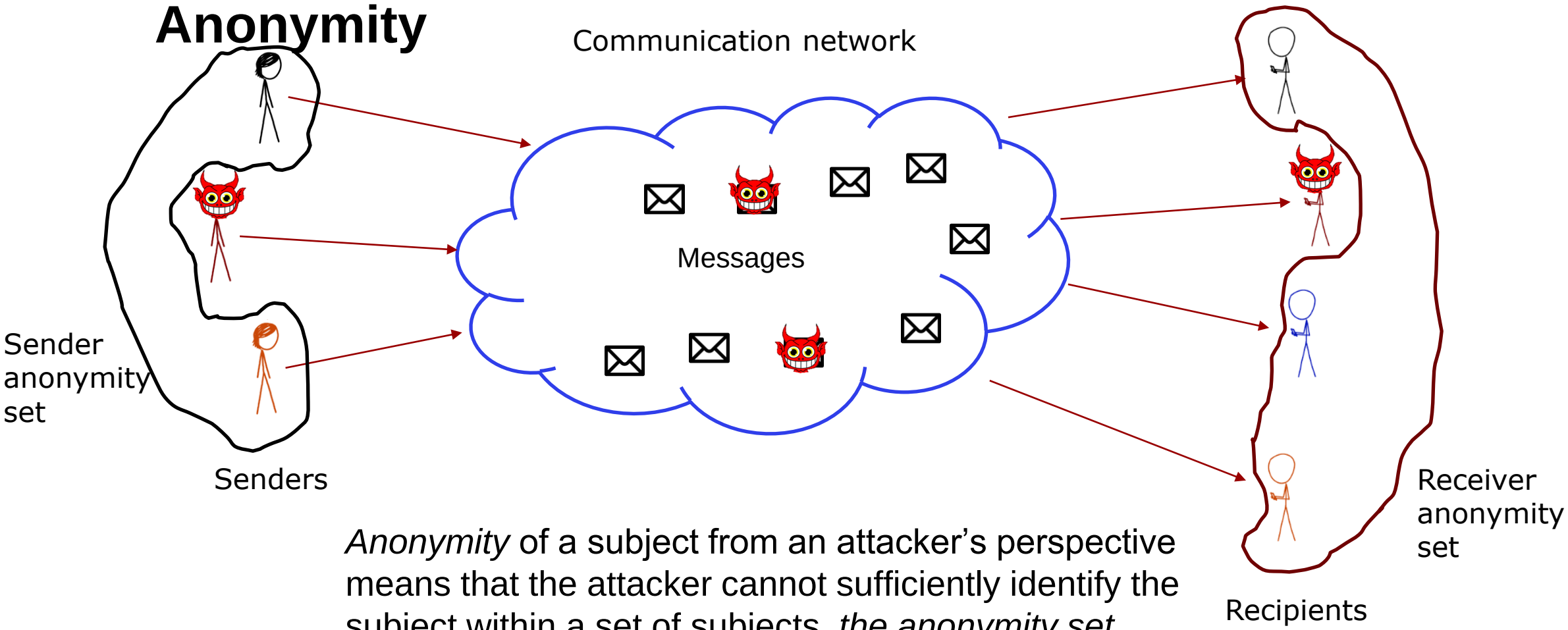
Defining systems



The adversary

1. Wants to infer certain Items of Interest (e.g. attributes)
2. May control some parties or communication lines
3. Never forgets

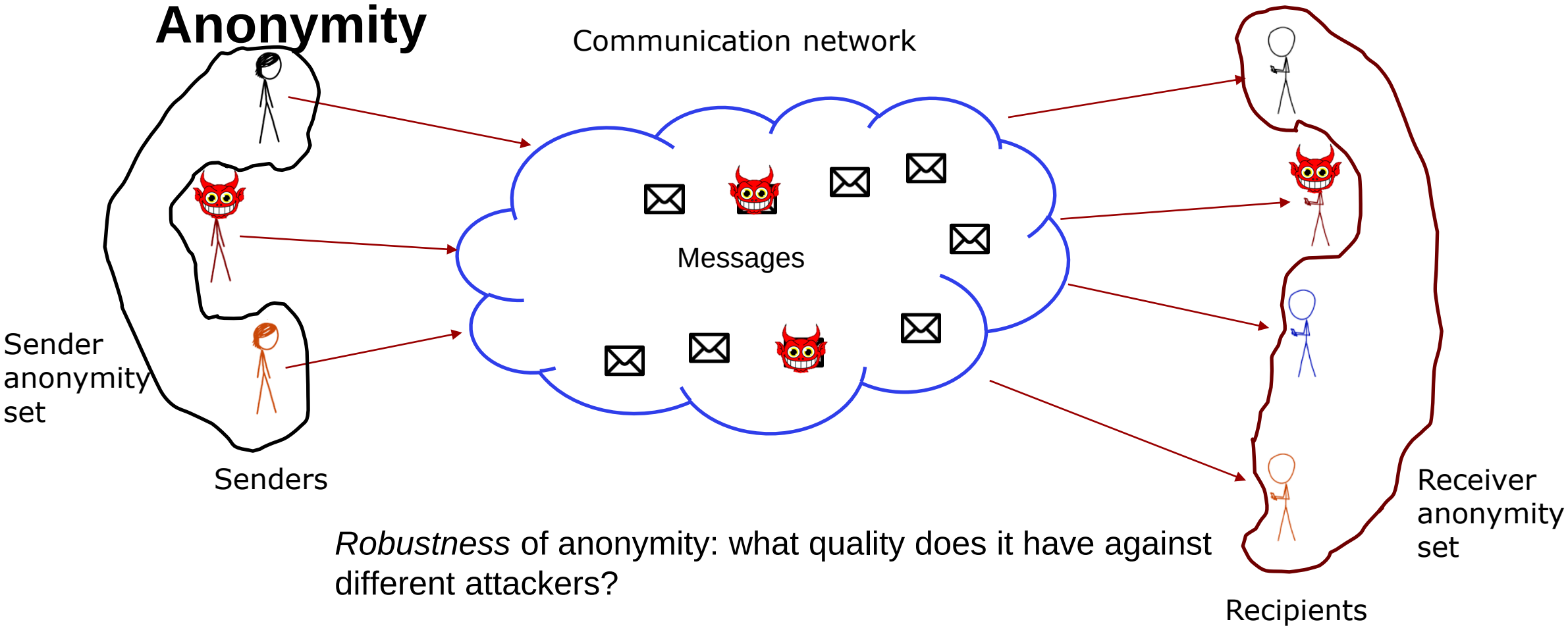
Anonymity



Anonymity of a subject from an attacker's perspective means that the attacker cannot sufficiently identify the subject within a set of subjects, *the anonymity set.*

A terminology for talking about privacy by data minimization, Pfitzmann & Hansen, 2010

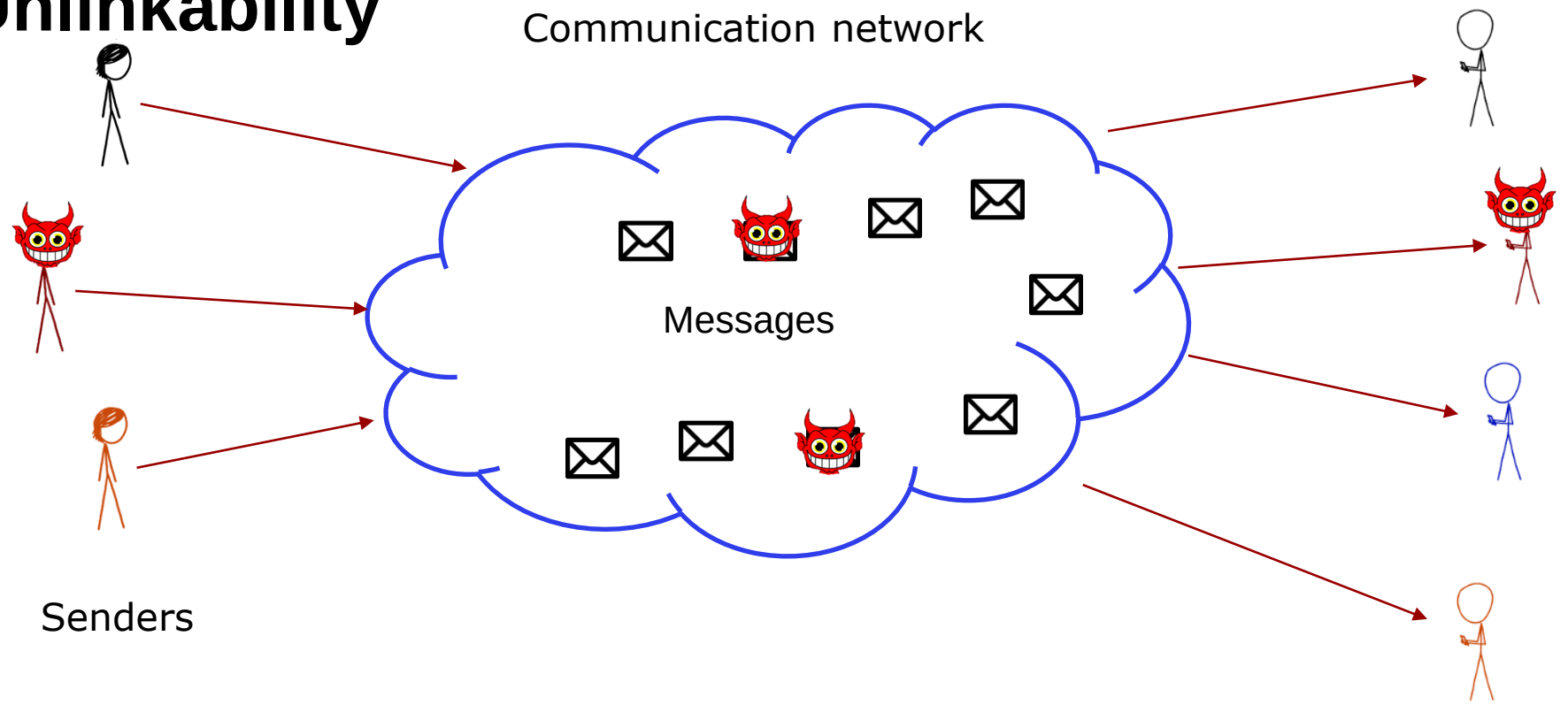
Anonymity



Robustness of anonymity: what quality does it have against different attackers?

Attacker a-priori knowledge and observations of behavior can always decrease, but never increase anonymity.

Unlinkability



Unlinkability of two or more IOIs from an attacker's perspective means that within the system, the attacker cannot sufficiently distinguish whether these IOIs are related or not

Unlinkability of a subject wrt an attribute implies anonymity of a subject and this attribute.

Undetectability & Unobservability

Undetectability of an IOI from the attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not.

Anonymity: relationship of IOI to subject is protected

Unlinkability: relationship of IOIs is protected

Undetectability: IOI is protected

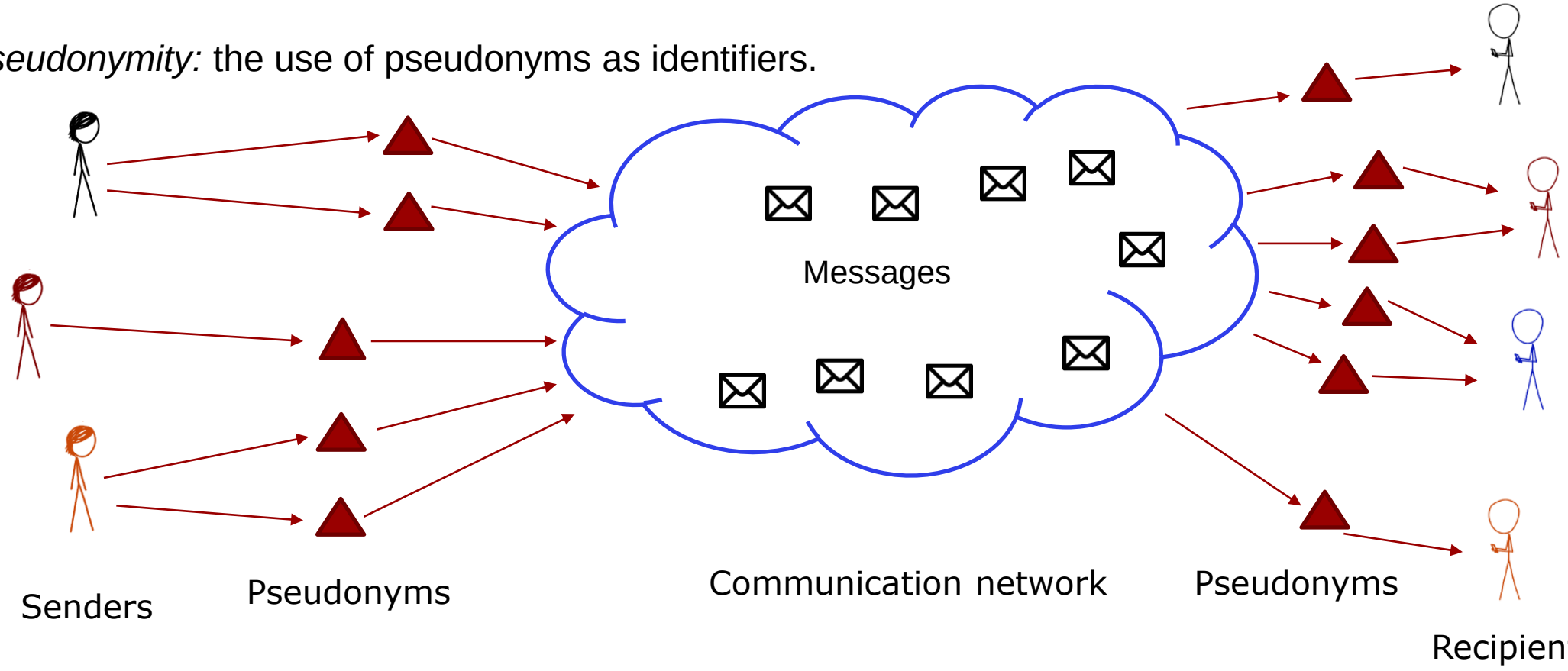
Unobservability of an IOI

- Undetectability of IOI against all subjects uninvolved in it; and
- Anonymity of the subjects involved in the IOI against the other subjects' involved in IOI.

Pseudonymity

Pseudonym: Identifier of a subject other than the subject's real name.

Pseudonymity: the use of pseudonyms as identifiers.



How can pseudonyms fail?

Do you have pseudonyms?

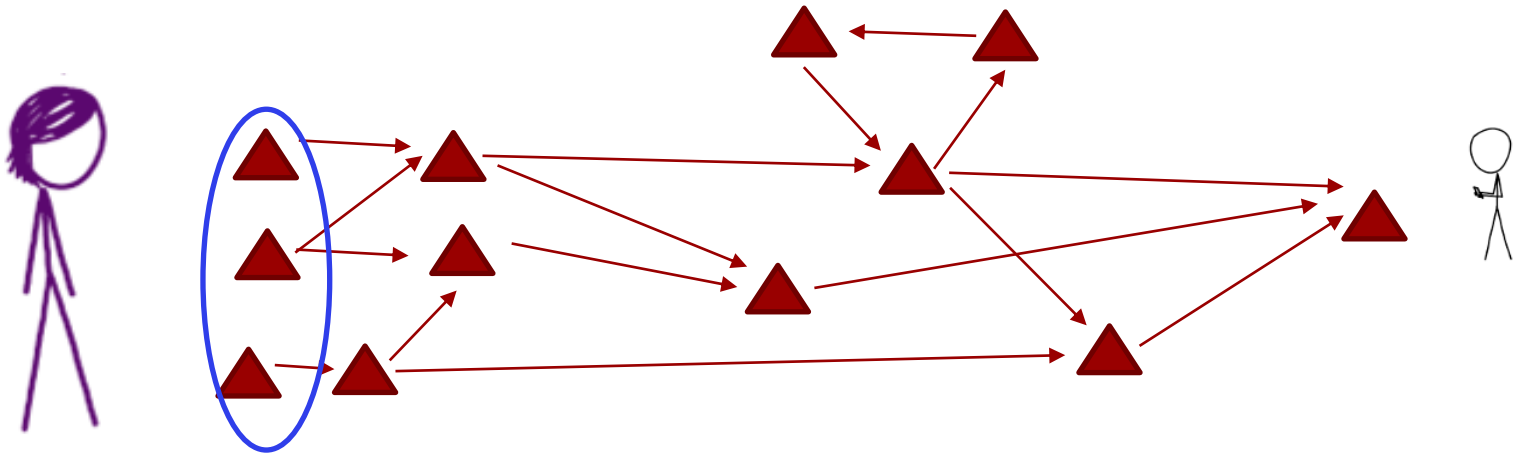
Are they tied to you?

Could they be tied to you?

Traffic analysis (Metadata)

Metadata (who sent a message to whom, when and of which size) can be sufficient to break pseudonymity

Idea: identify pseudonyms for same actor



Quantitative Analysis of the Full Bitcoin Transaction Graph, Ron & Shamir, FC 2013

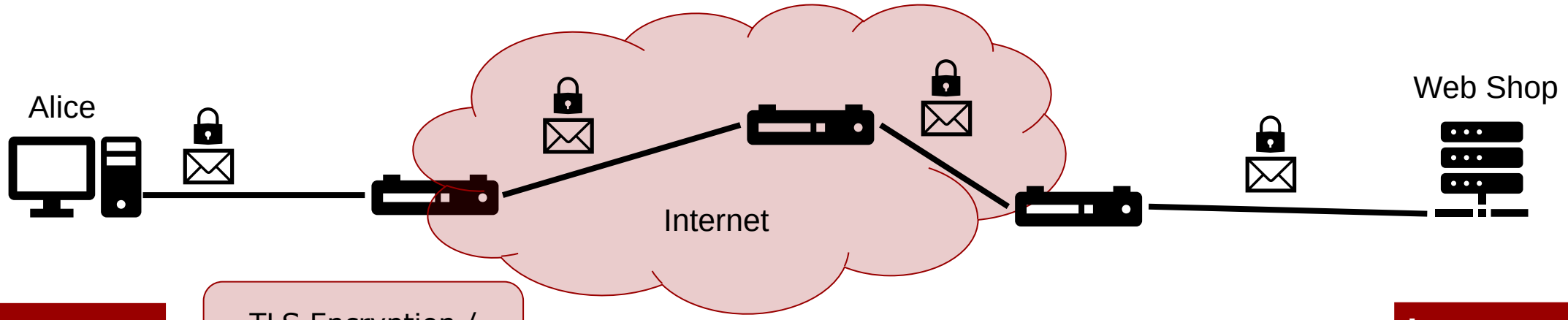
A Fistful of Bitcoins: Characterizing Payments Among Men with No Names, Meiklejohn et al., ACM IMC 2013

Chainalysis

Browsers, cookies, …

Online privacy

TLS



Layer
Application
TLS
TCP
Network
Data Link
Physical

TLS Encryption / Authentication

Data
TLS Data
TCP TLS Data
IP TCP TLS Data
Eth IP TCP TLS Data

Protected
Unprotected

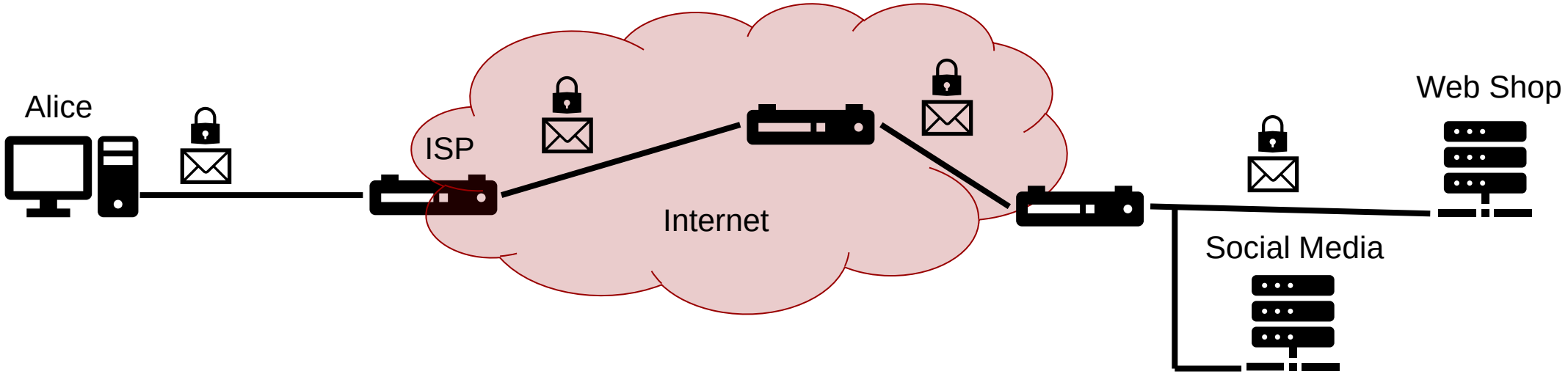
TLS Decryption / Verification

Data
TLS Data
TCP TLS Data
IP TCP TLS Data
Eth IP TCP TLS Data

Layer
Application
TLS
TCP
Network
Data Link
Physical

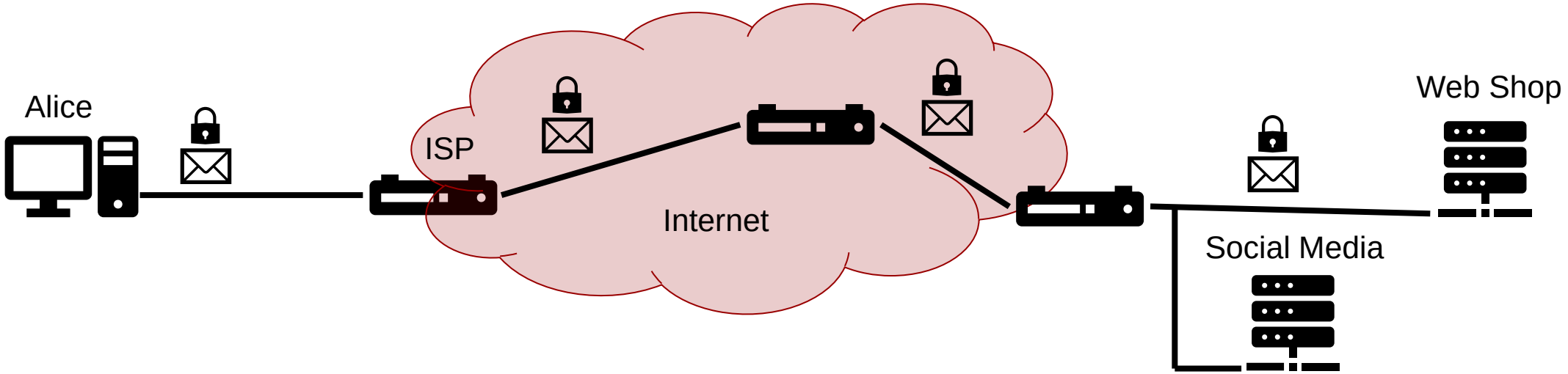
Alice looks for a product at the Web Shop (HTTPS)

- Can the web shop read her messages?
- Can her ISP read her messages?
- Can an eavesdropper at the Internet read her messages?
- Can her favourite social media platform read her messages?
- Can the government read her messages?



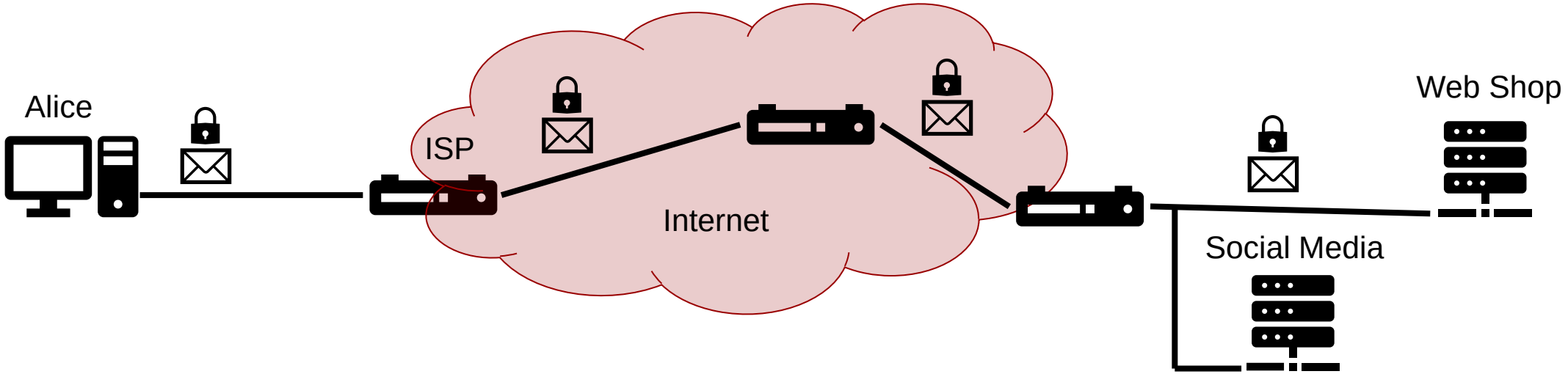
Alice looks for a product at the Web Shop (HTTPS)

- Can the web shop read her messages? **Yes**
- Can her ISP read her messages? **No**
- Can an eavesdropper at the Internet read her messages? **No**
- Can her favourite social media platform read her messages? **No**
- Can the government read her messages? In theory **yes**; in practice it depends

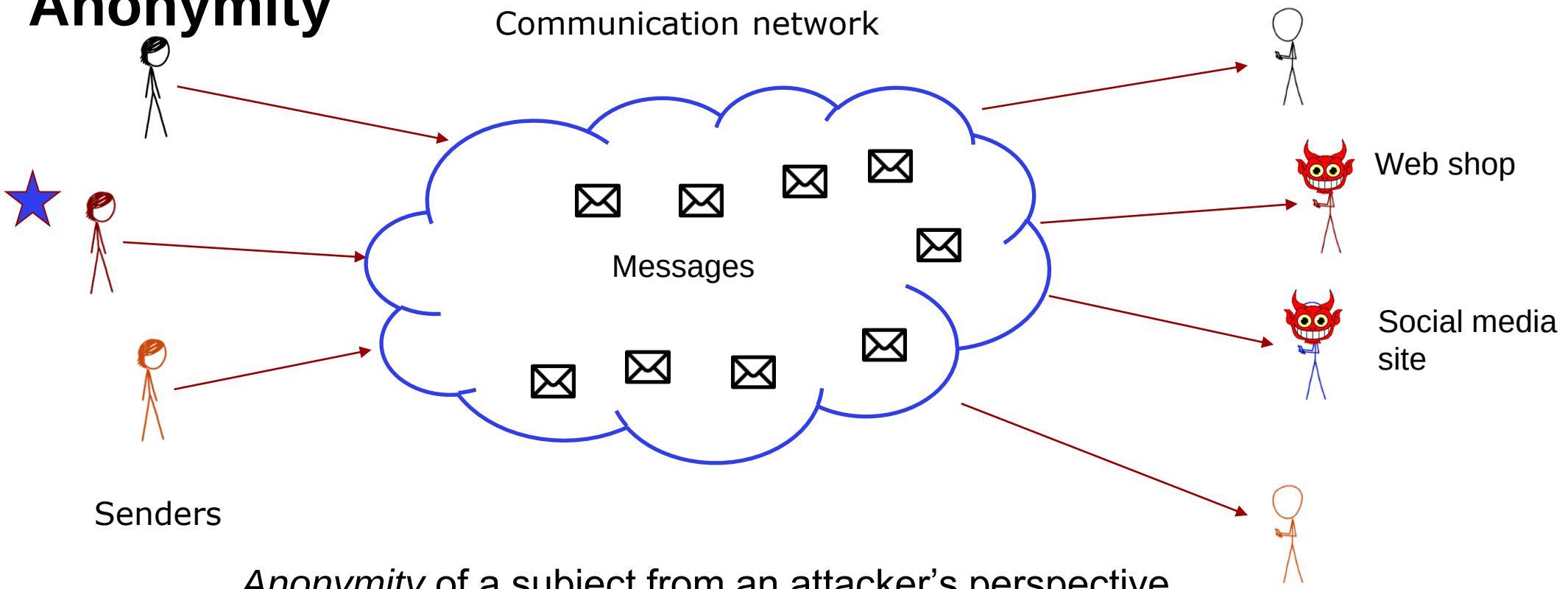


Tracking (breaking sender anonymity)

- Alice searches for a new bicycle at the Web shop
- The communication is protected by TLS
- The next day she finds an **advertisement** on a social media platform about bicycles
- How did that happen?



Anonymity



Anonymity of a subject from an attacker's perspective means that the attacker cannot sufficiently identify the subject within a set of subjects, *the anonymity set.*

HTTP Cookies

- HTTP is stateless
- How do you implement language preferences, shopping baskets, etc, without state?
  - **Cookie:** a (domain, key, value) triplet that the server saves at the client
    - **First-party cookies:** set by the domain the user is visiting
    - **Third-party cookies:** set by other domains embedded in the top-level page

- Alice visits the web-shop
  - The web-shop page tells the browser to retrieve components from an Advertiser
  - The advertiser leaves a third-party cookie at the browser of Alice with a unique id
  - The next day Alice visits a social media platform
  - It also tells the browser to retrieve components from the same Advertiser
  - The Advertiser reads the third-party cookie and associates Alice with the Web-shop visit

- Protection: **Privacy-friendly browsers** (e.g. Brave) block third-party cookies

How to identify clients without cookies?

HTTP headers

- IP address
- Geo-location
- Operating system
- Type of device
- Brower and version
- Plugins

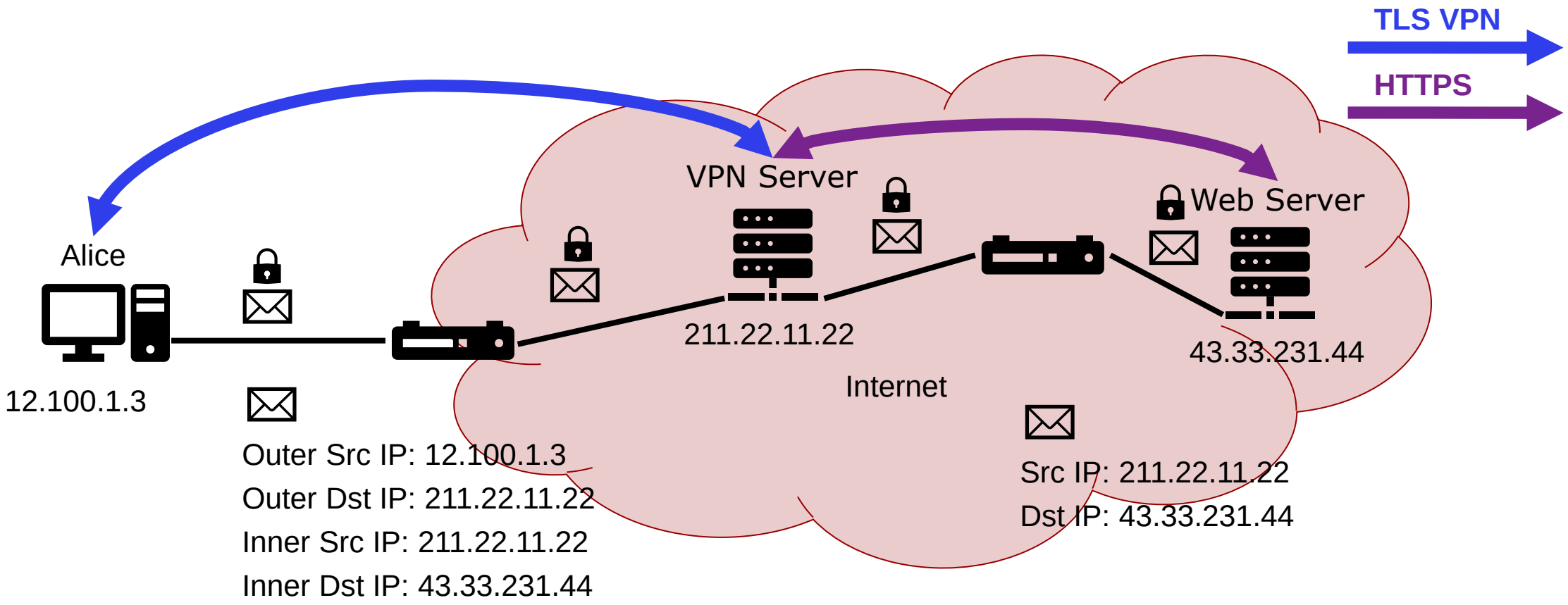
Javascript

Can get computer-specific information such as installed fonts, screen resolution

Tracking via caching

ETag are server-issued identifiers assigned to a specific version of a resource found at a URL. If the resource representation at that URL changes, so does the ETag.

Can we enhance privacy with VPN?



Can we enhance privacy with VPN?

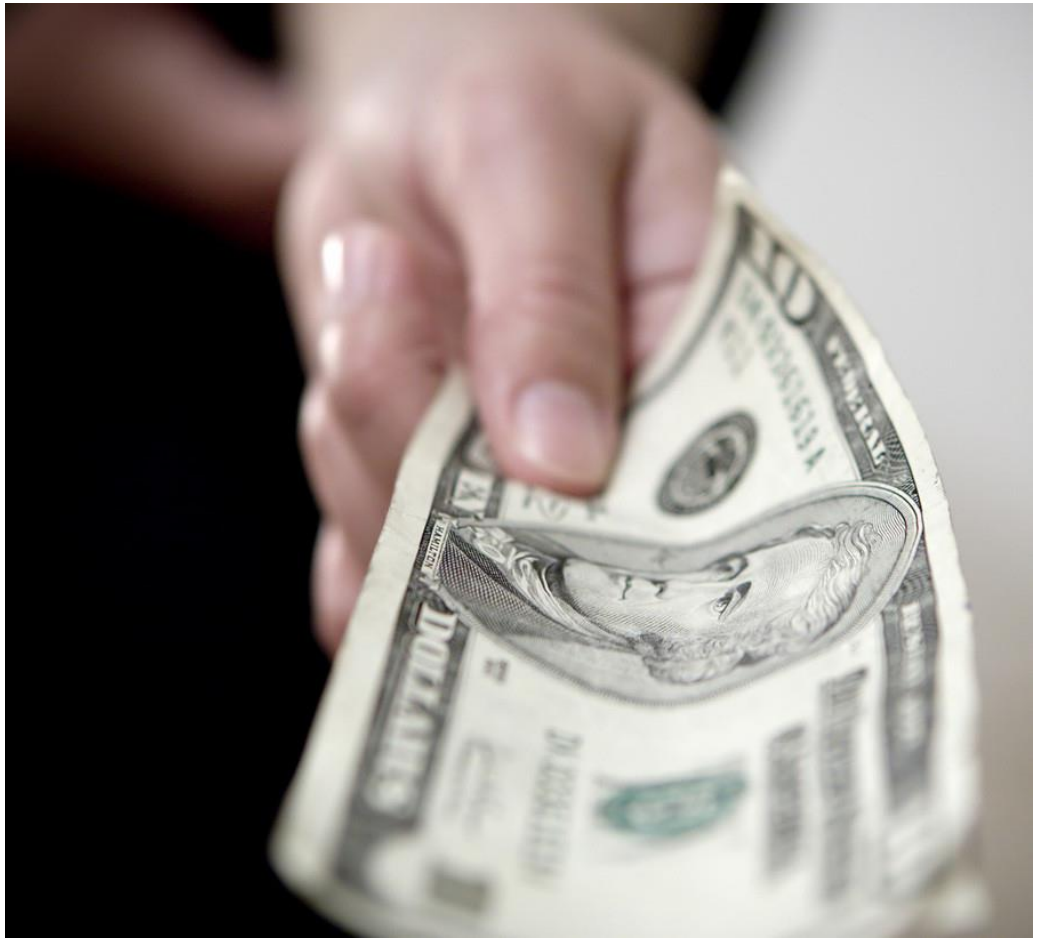
Depends on your threat model…

- A VPN can protect us from local eavesdroppers if using a public network
- A VPN can enhance anonymity from our ISP, network administrators, employers, etc
- But…

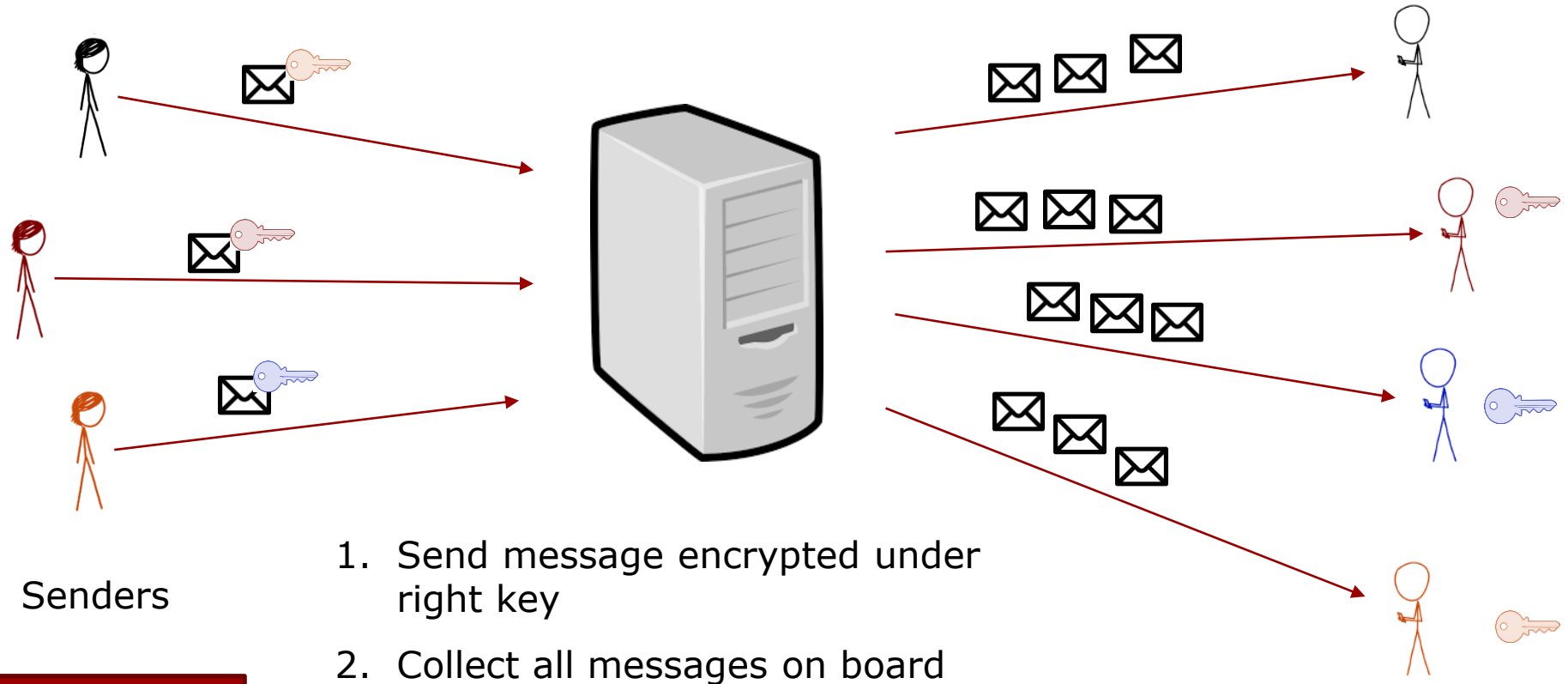
VPN is **not good** for anonymity

- The VPN service provider is literally a **man in the middle**
- How much do you trust your VPN service provider?
- The VPN service provider can keep logs, provide them to governments or sell them to advertisers

So – anonymous communication?



“Simple” solution – public message board



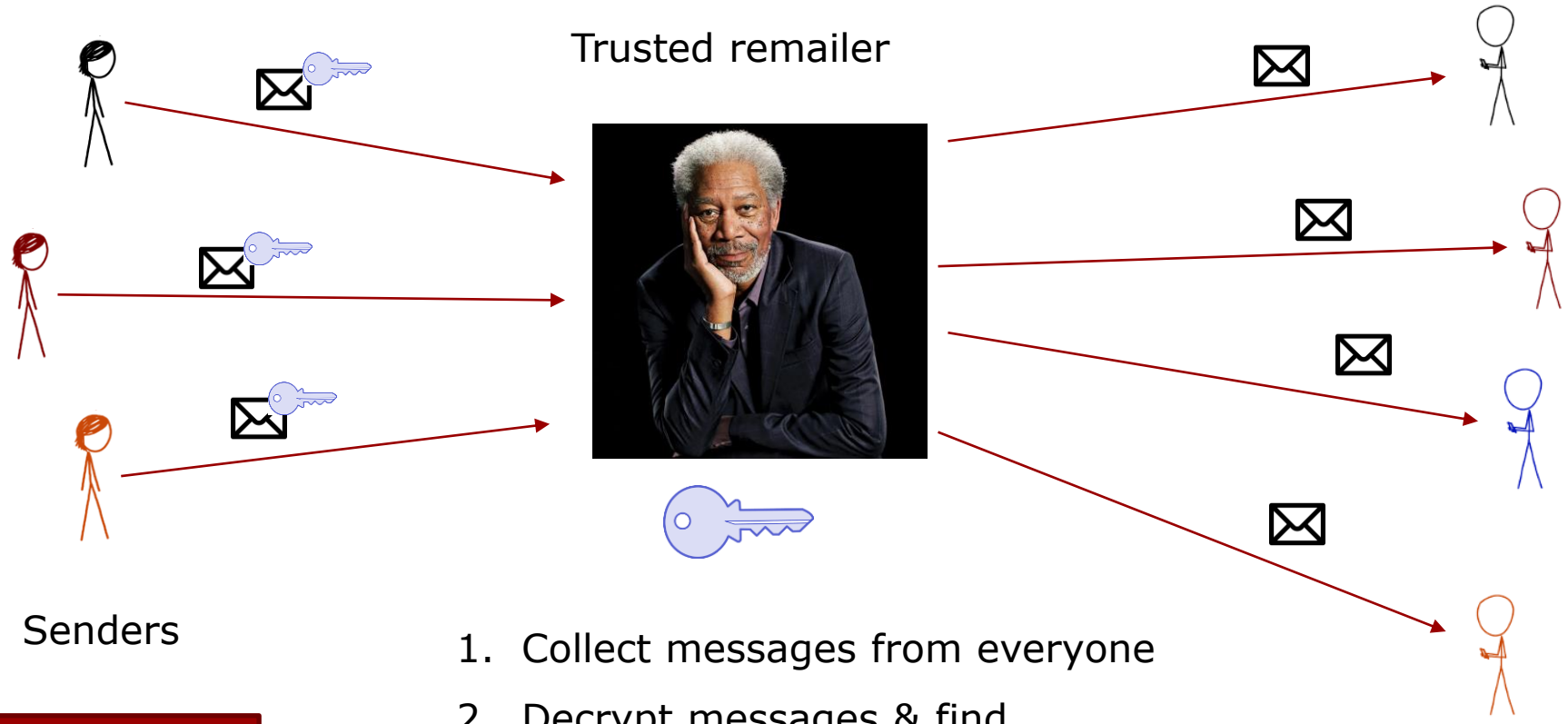
Senders

Recipients

1. Send message encrypted under right key
2. Collect all messages on board
3. Every recipient downloads all ciphertexts

Disadvantages?

“Simple” solution – trusted remailer



1. Collect messages from everyone
2. Decrypt messages & find recipients
3. Forward messages

Disadvantages?

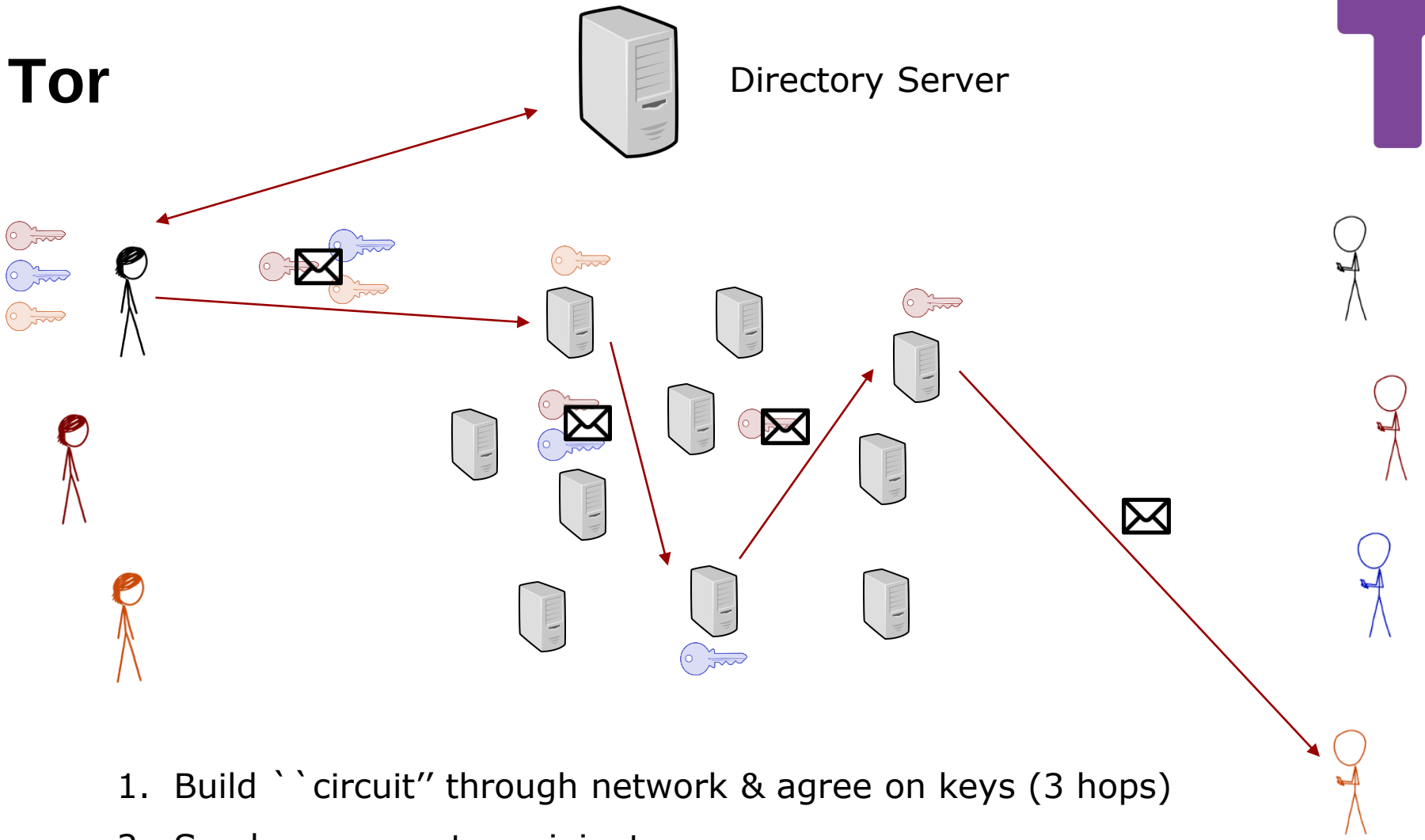
Anonymisation with Onion Routing and Tor

- Tor stands for “The Onion Router”
- A technique for **anonymous** TCP-based communication on the Internet
- Originally developed by the US Military in the 90s
- Now, Tor is an open-source project

- The Tor network is supported by thousands of volunteers that operate an Onion Router
  - The list of **Onion Routers (OR)** is public
  - The more Onion Routers the better the anonymity!

- You can access Tor using the Tor Browser, Brave, etc

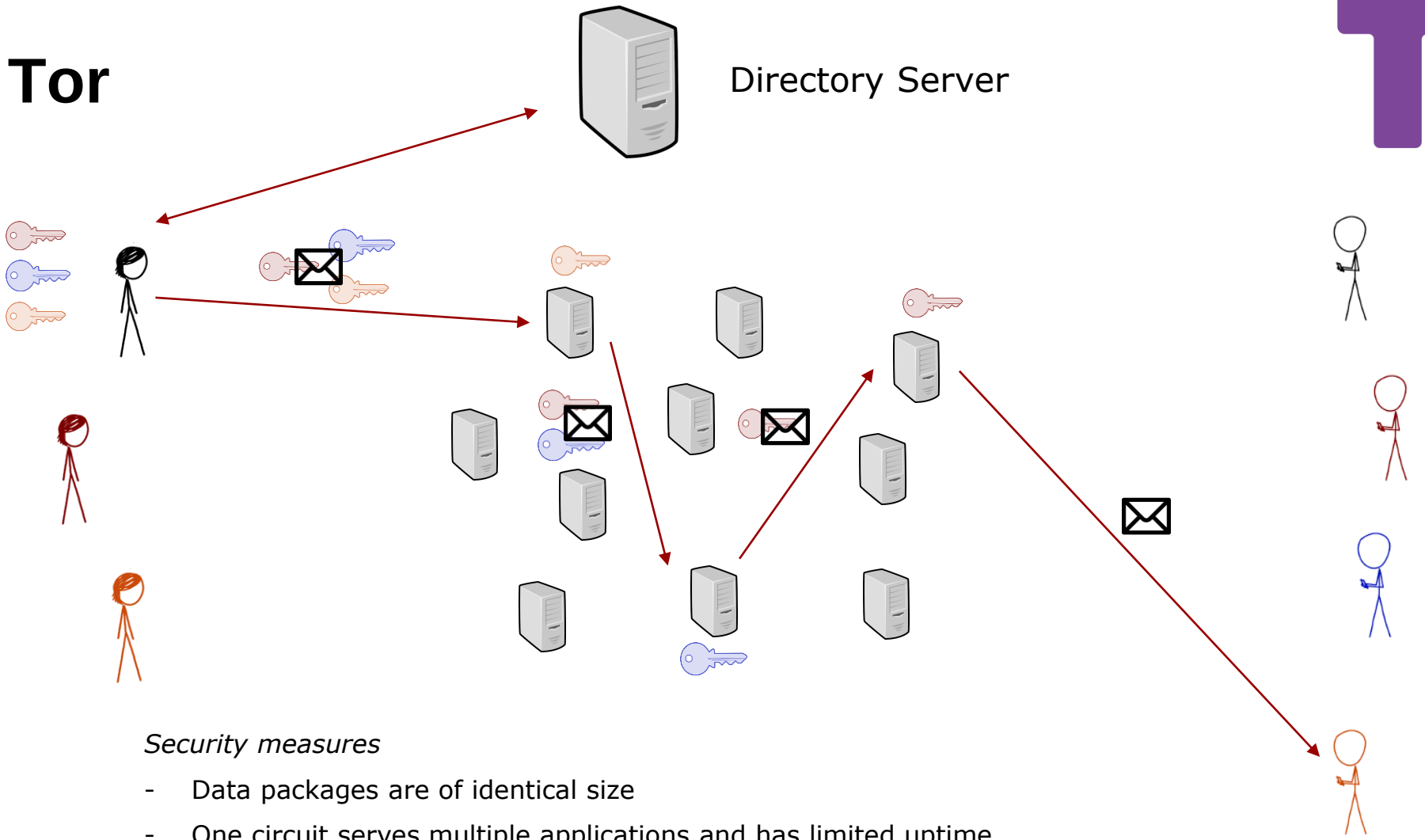
Tor



1. Build ``circuit'' through network & agree on keys (3 hops)
2. Send messages to recipient

Low-latency, low corruption resistance

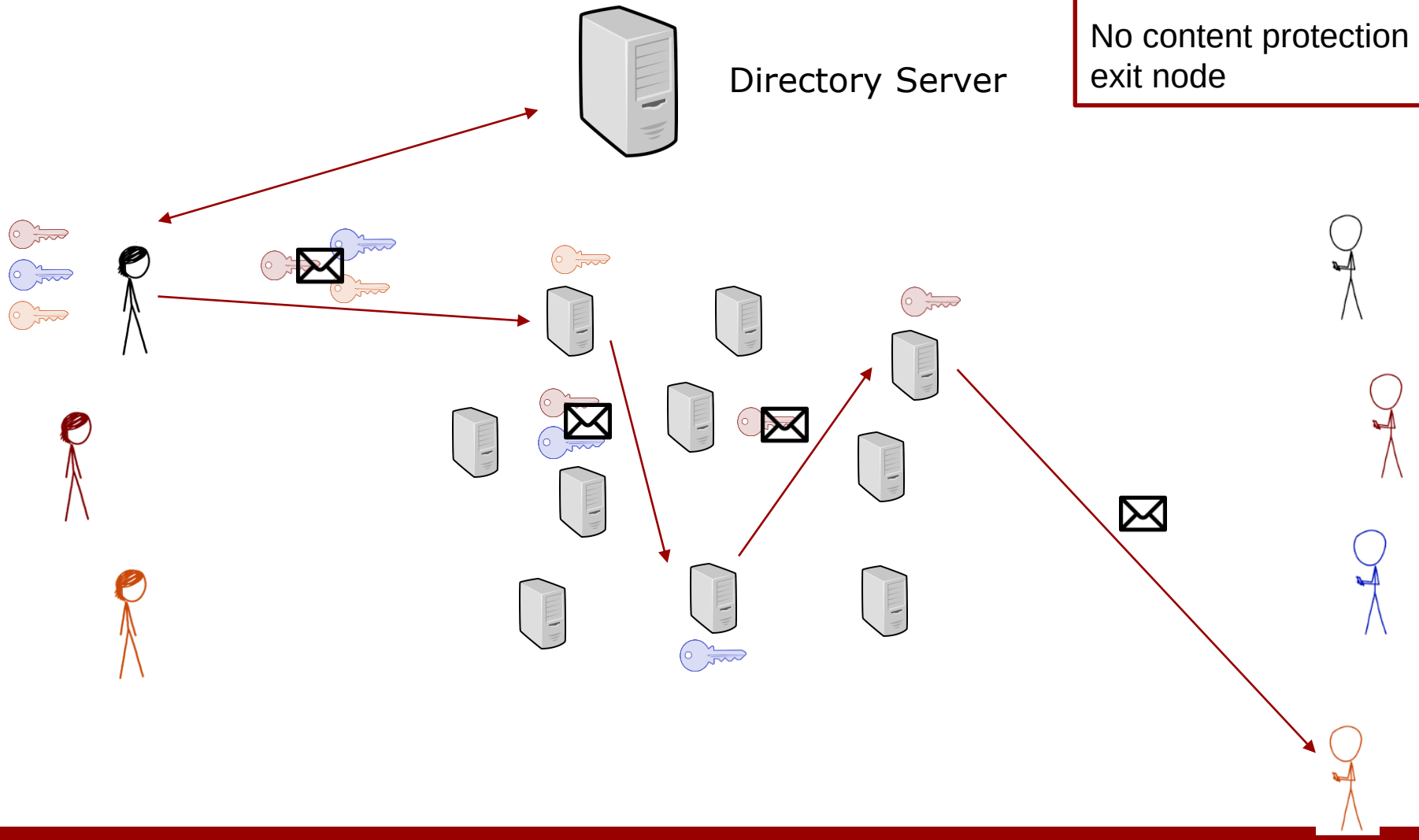
Tor



Security measures

- Data packages are of identical size
- One circuit serves multiple applications and has limited uptime
- Large number of routers
- Hide data packages as other traffic

What attacks do you see?



Corruption of many servers on both ends

No content protection on exit node

Tor

- Tor provides anonymity at the cost of performance: **much slower**
  - OR might be at 3 different corners of the world
  - OR might not be on very high bandwidth links

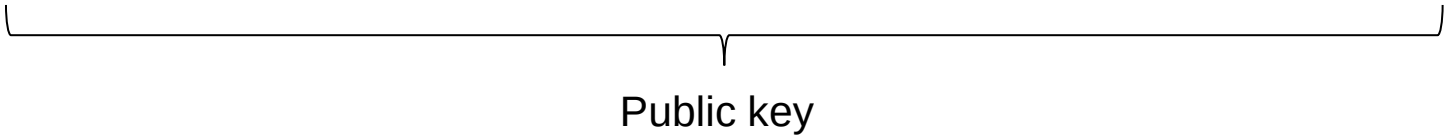
- If you use your credentials to log in or use your credit card, Tor can’t help you!
- Similar to VPN, you need to use TLS (HTTPS) otherwise the last hop is unencrypted

Tor Hidden Services

- Previous examples were about anonymous access to a public server/service
  – Alice is anonymous but the server is public

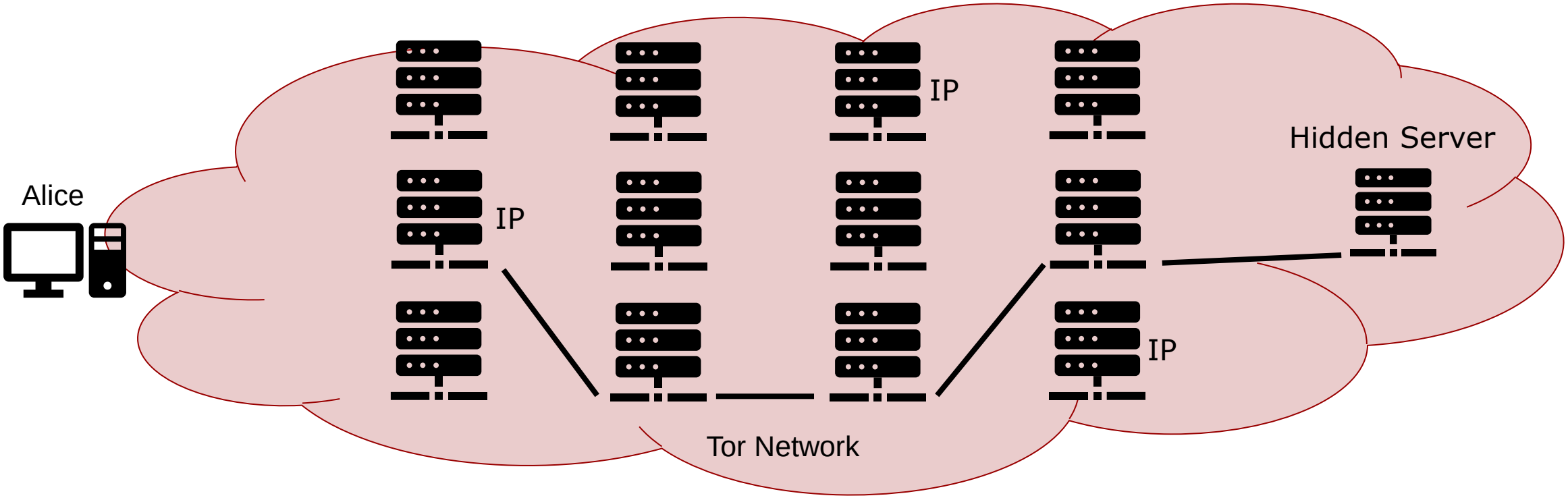
- Tor also supports **hidden servers/services** (aka dark web)
  – Alice is anonymous, the server is also anonymous, Tor helps them talk!

- Alice needs ``some'' identifier for the service
  vww6ybal4bd7szmgncyruucpgfkqahzddi37ktceo3ah7ngmcopnpyyd.onion



Tor Hidden Services: Introduction Points

- The hidden server selects three random OR as **Introduction Points (IP)**
- Server creates public key and establishes Tor circuits to each IP
- The server creates a **descriptor** containing the IPs, signed using its secret key!



Interlude: Distributed Hash Table

Distributed protocol to identify parties responsible for $(key, value)$ pairs.

Input for lookup: key
Then: communication in network to find a server responsible for key who knows $value$

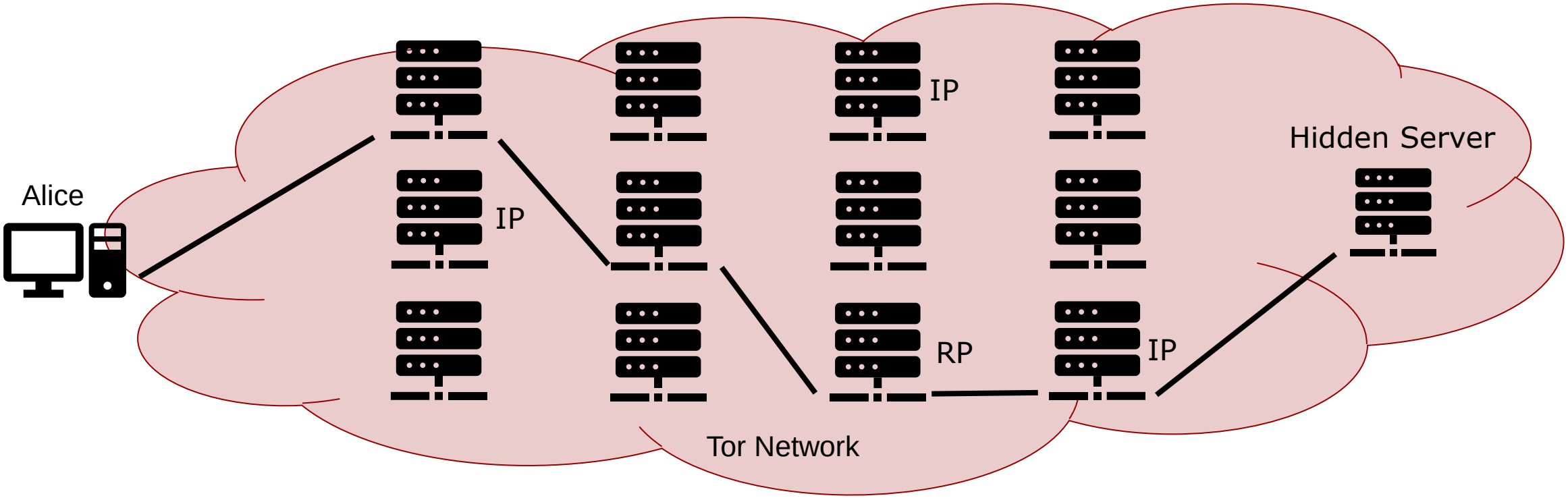
Upon joining of new server:
Made responsible for *some* keys.

Important:

- DHT does **not** protect privacy of $key, value$.
- There is **no** coordinator.

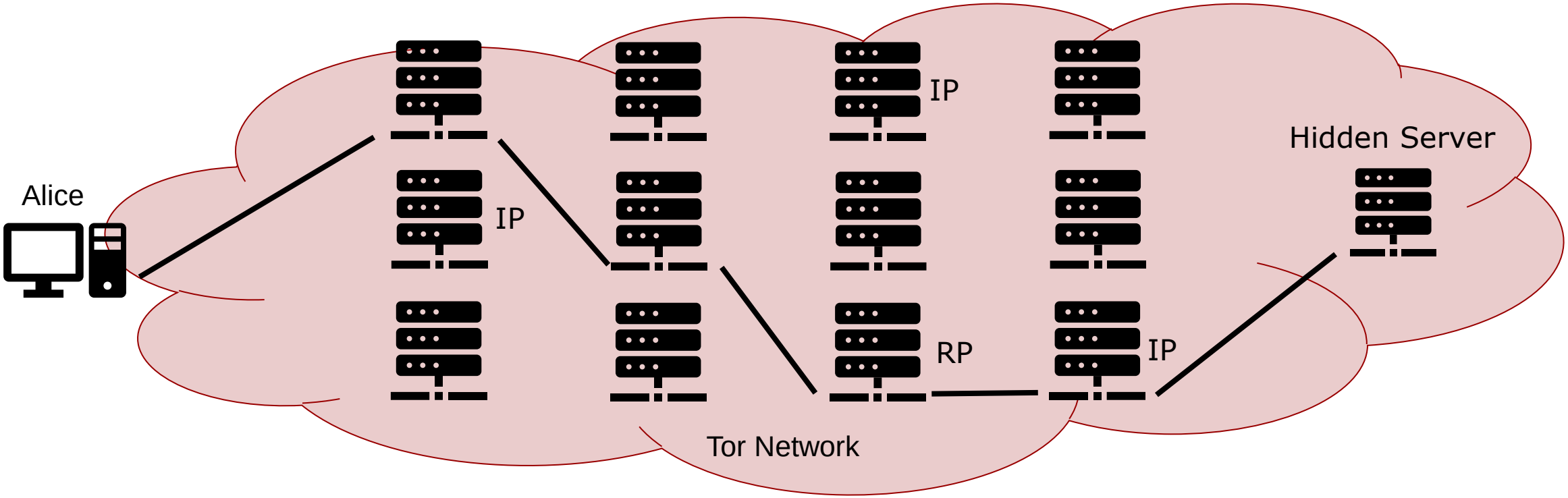
Tor Hidden Services: Finding the IPs

- Server uses DHT to identify some nodes based on pk, then gives them descriptor
- If Alice looks for service, she identifies same nodes using pk via DHT, gets descriptor and validates signature using pk
- From descriptor, Alice can then identify IPs



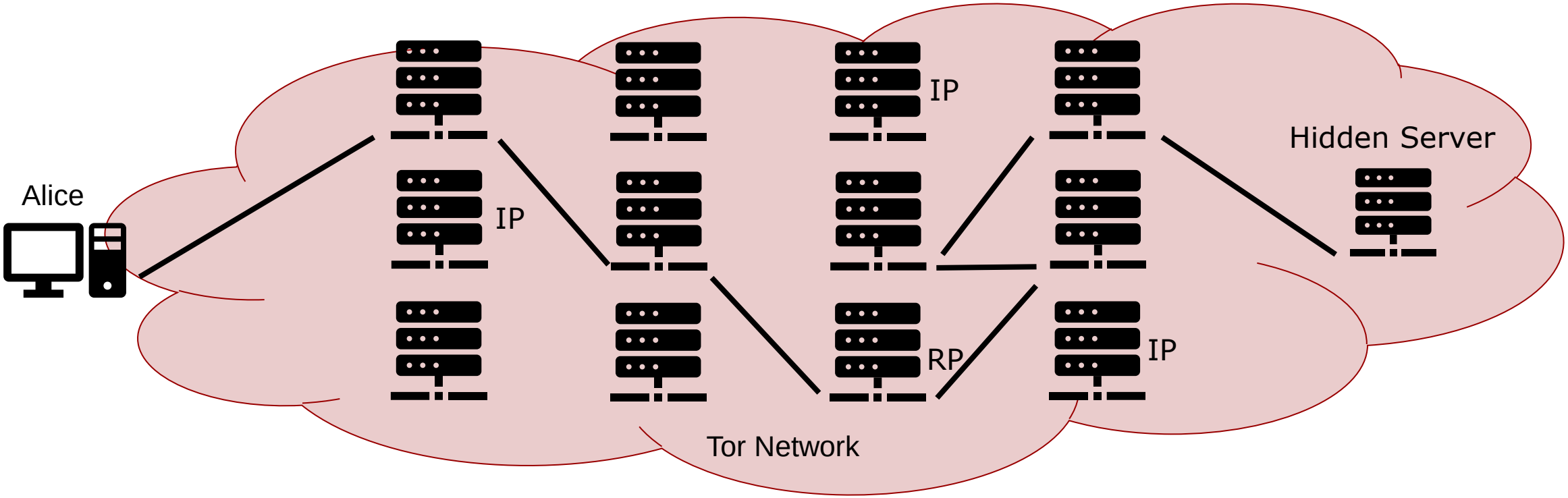
Tor Hidden Services: Rendezvous Point

- Alice makes a 3-hop Tor circuit to a random OR: **The Rendezvous Point (RP)**
- Alice generates a random id and asks a random IP to pass it to the hidden server (via the RP)
- The introduction is **via 3 Tor circuits**: Alice-RP, RP-IP, IP to Hidden Server



Tor Hidden Services: Circuit Establishments

- The Hidden Server can choose to accept or deny the random ID
- If accepted, it creates a 3-hop TOR circuit to the Rendezvous Point (RP)
- At this point the RP can match the random ID into the same **7-hop Tor circuit**



End-to-End encryption

Private communication

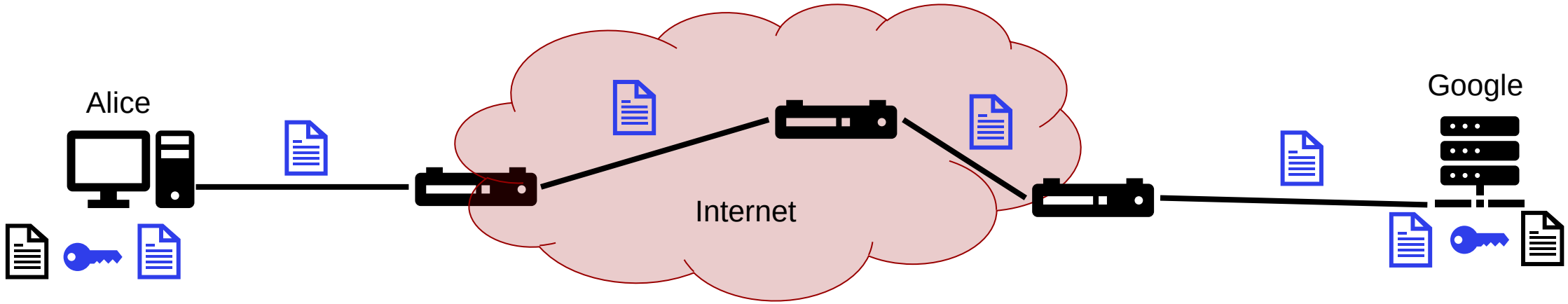
End-to-End Encryption

- The term is used from various perspectives
  - From a networking perspective, TLS is E2E (client end to server end)
  - As opposed to link-layer encryption (e.g. WiFi encryption)

- From an application perspective, a server may be a temporary end (user end to user end)
  - Let’s see at an example: Cloud services such as Google drive

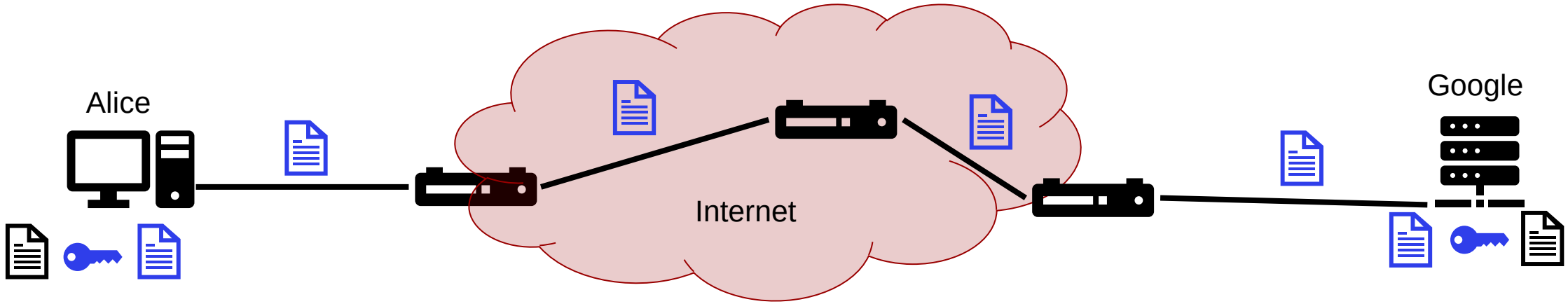
Cloud Storage

- Alice sends a private file to Google servers over an encrypted TLS channel
- Google stores it on the Google servers
- Alice downloads the private file from the Google servers over an encrypted TLS channel
- **Is this really private?**



Cloud Storage

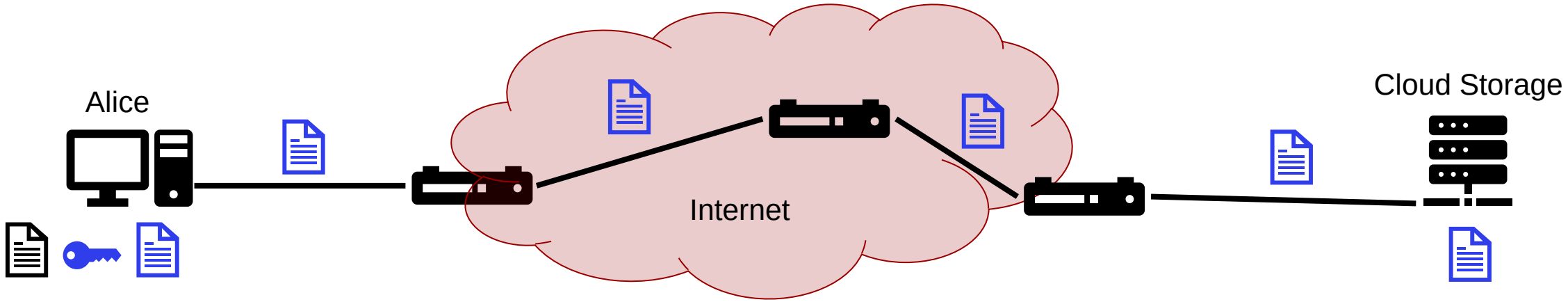
- Alice sends a private file to Google servers over an encrypted TLS channel
- Google stores it on the Google servers
- Alice downloads the private file from the Google servers over an encrypted TLS channel
- **Is this really private? No, Google can read the file – no confidentiality**



Cloud Storage with Client-Side Encryption

- Alice encrypts the file locally
- She sends the encrypted file to the server
- The server stores the encrypted file but doesn't have access to the key!
- Alice downloads the encrypted file from the server and decrypts it locally

- May have disadvantages for cloud provider: no deduplication



Client-Side Encryption

A **privacy-friendly** way to use **cloud services**

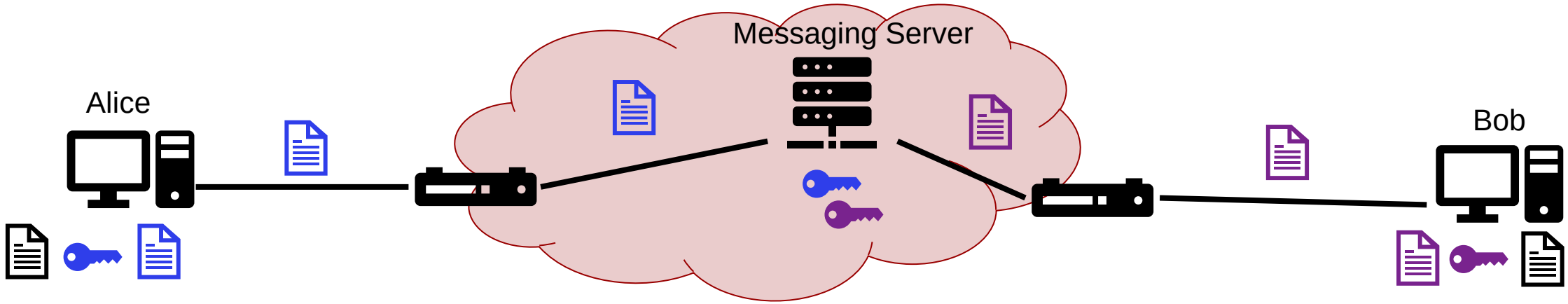
- Encryption and Decryption of sensitive data occurs at the client's computer
- Only ciphertext is transmitted over the Internet and stored at cloud servers
- The cloud service providers do not have access to the keys

Applications

- Password managers
- Cloud Storage
- Private email / calendar / phone contacts
- Etc

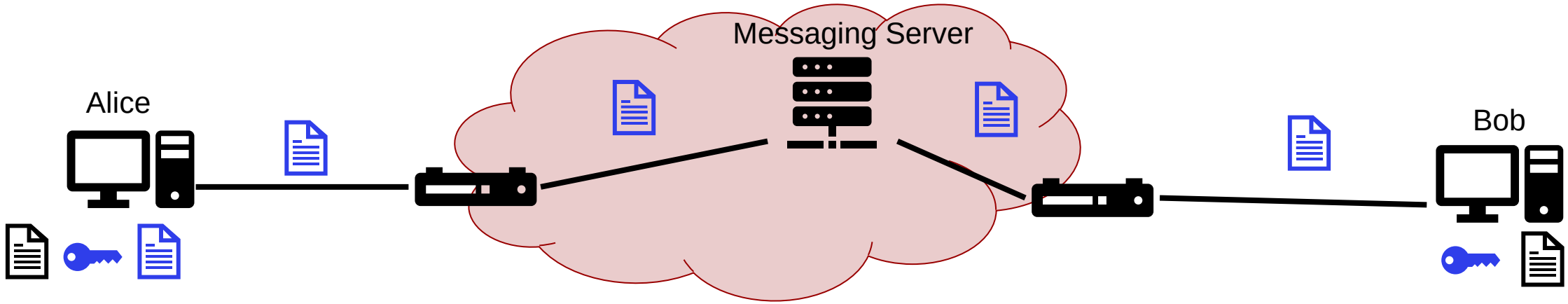
Messaging Applications

- Alice and Bob wish to communicate **asynchronously** (send messages when other is offline)
- Alice makes connection to Messaging Server over TLS
- Server stores the message waiting for Bob to come online
- When Bob is online, the server sends the message to Bob over TLS
- Not really E2E encryption, as the server can read the message!

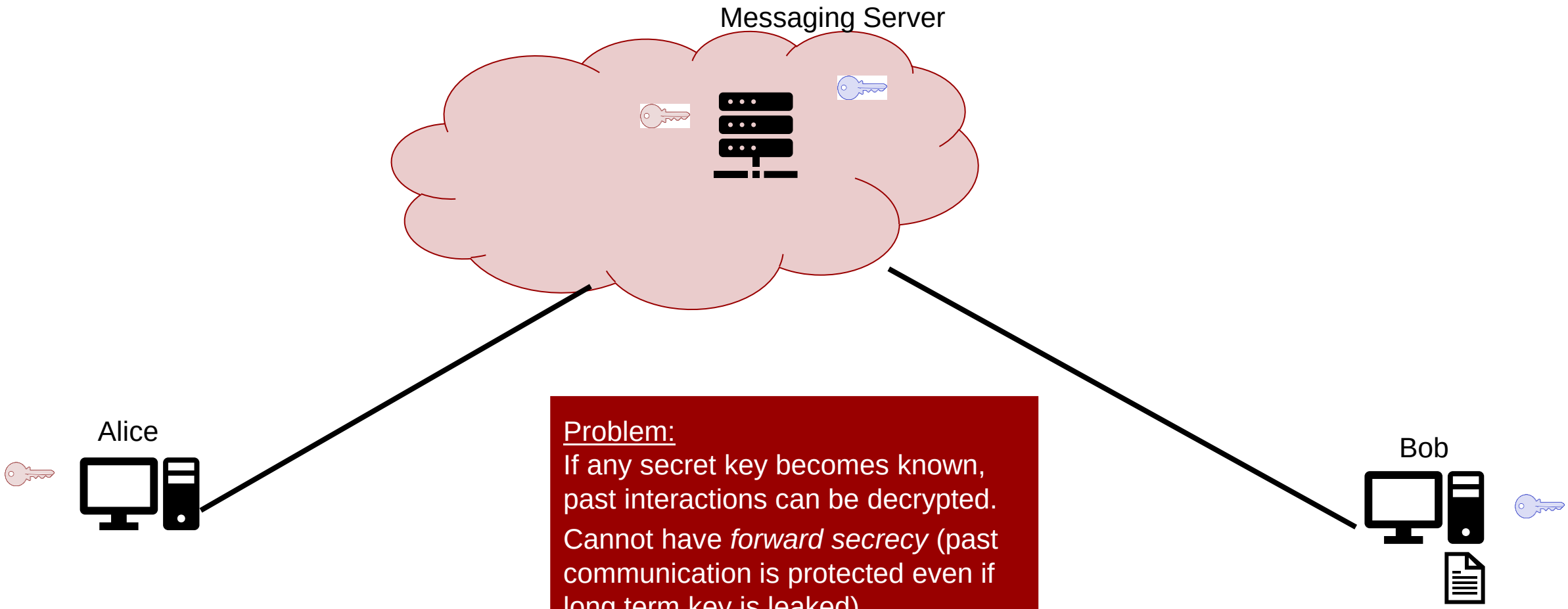


Messaging Applications

- Alternative: Alice and Bob agree on key first
- Difficult if both are not online simultaneously…



Store public keys on server



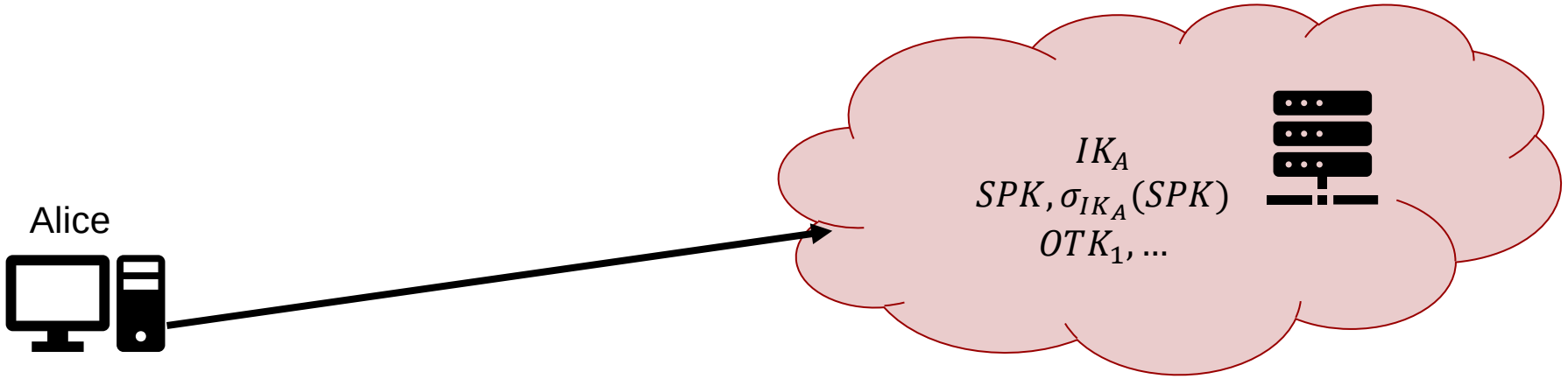
Problem:
If any secret key becomes known, past interactions can be decrypted.

Cannot have *forward secrecy* (past communication is protected even if long term key is leaked).

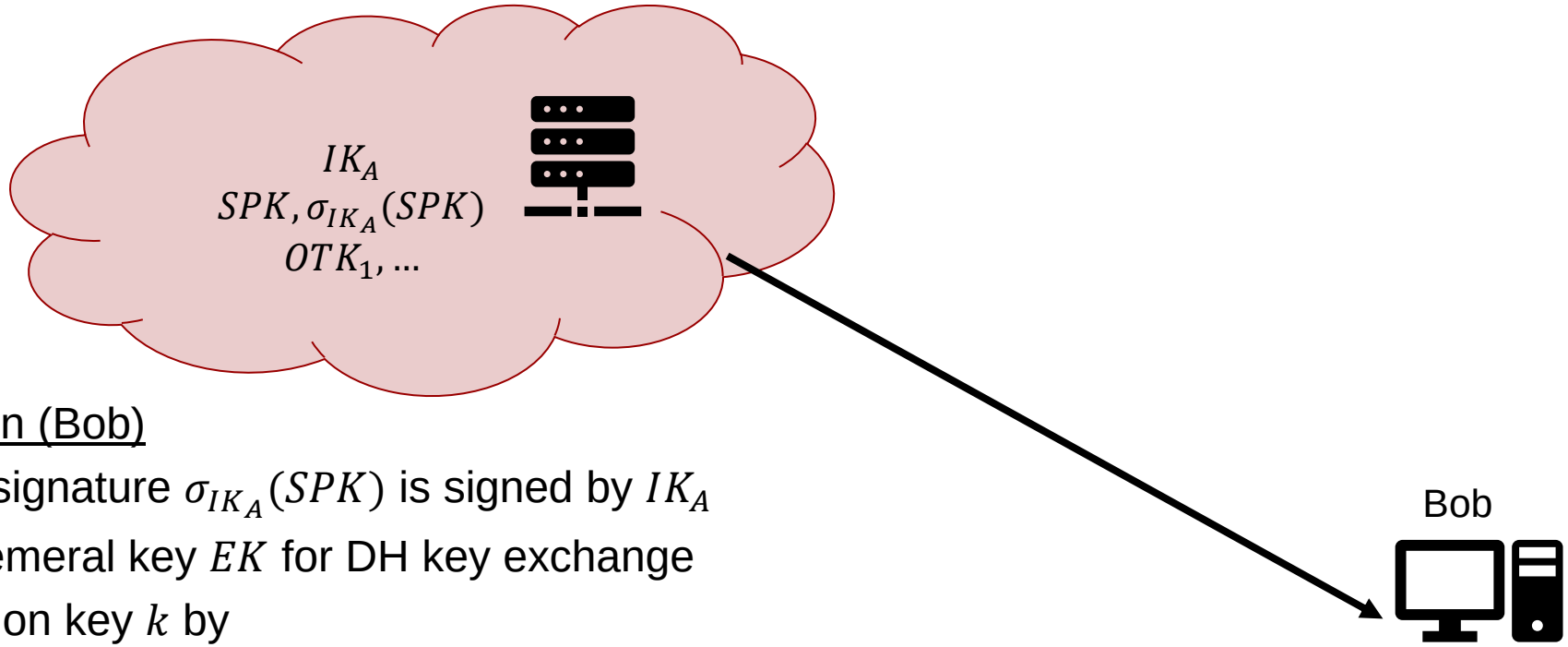
Signal: Asynchronous Key Establishment with forward secrecy

Signal builds on Diffie-Hellman Key Exchange

- Alice sends Identity Public Key IK_A, Signed Prekey $SPK, \sigma_{IK_A}(SPK)$ and several One-Time Public Keys $OTK_1, \ldots$ to the server
- Bob wants to communicate with Alice, but let's say Alice is offline
- Bob receives from server $IK_A, SPK, \sigma_{IK_A}(SPK)$ and one One-Time key OTK_i



Signal: Asynchronous Key Establishment with forward secrecy



Key Generation (Bob)

– Check SPK signature $\sigma_{IK_A}(SPK)$ is signed by IK_A

– Create Ephemeral key EK for DH key exchange

– Create session key k by

- computing multiple instances of DH key exchange between the IK_A, IK_B, EK, SPK and OTK_i
- Hash outcomes of key exchanges to obtain session secret k

Signal: Asynchronous Key Establishment with forward secrecy

Key Generation (Bob)

– Check SPK signature $\sigma_{IK_A}(SPK)$ is signed by IK_A

– Create Ephemeral key EK for DH

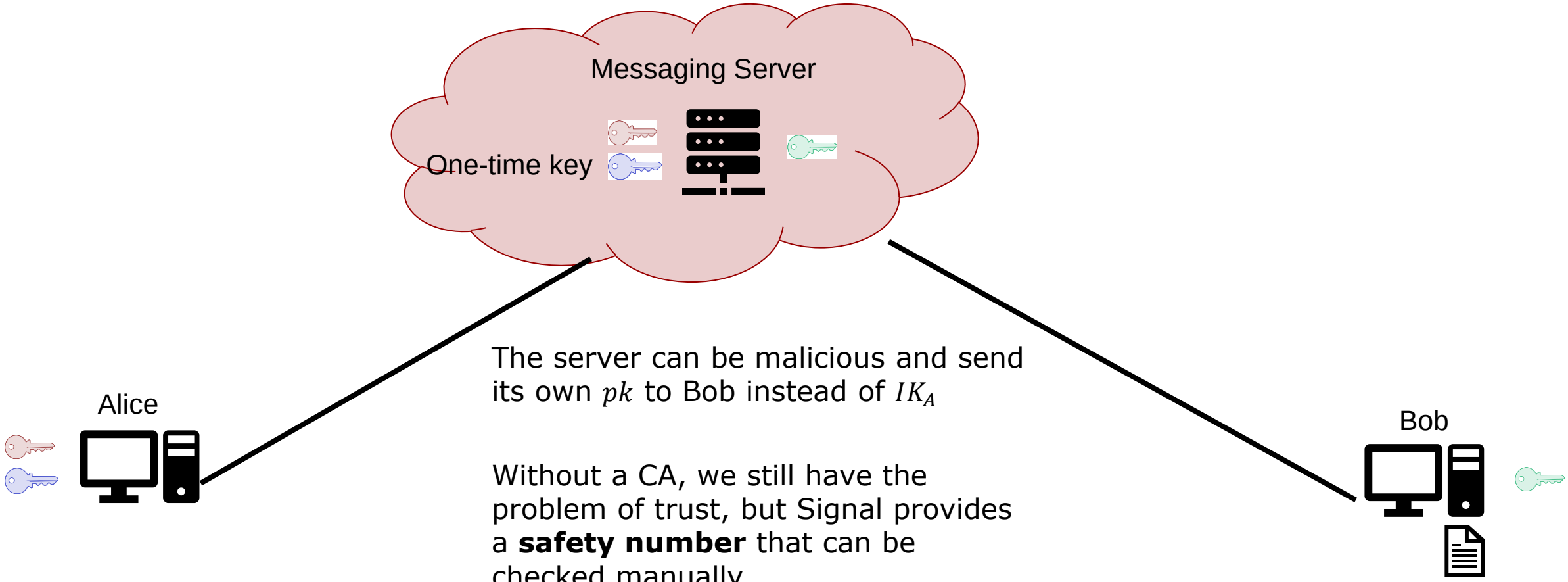
– Create session key k by

- computing multiple instances of DH key exchange between the IK_A, IK_B, EK, SPK and OTK_i
- Hash outcomes of key exchanges to obtain session secret k

If one-time key OTK_i is used only once, perfect forward secrecy.

SPK is updated regularly and can protect forward secrecy after update, if no more OTK available.

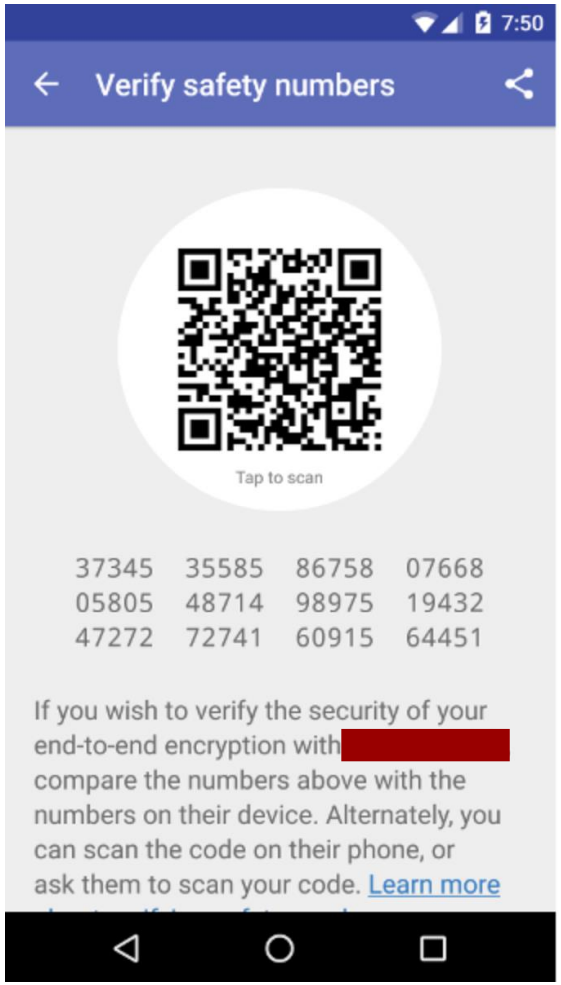
How Key Agreement works



The server can be malicious and send its own pk to Bob instead of IK_A

Without a CA, we still have the problem of trust, but Signal provides a **safety number** that can be checked manually.

Signal Protocol: Safety Number



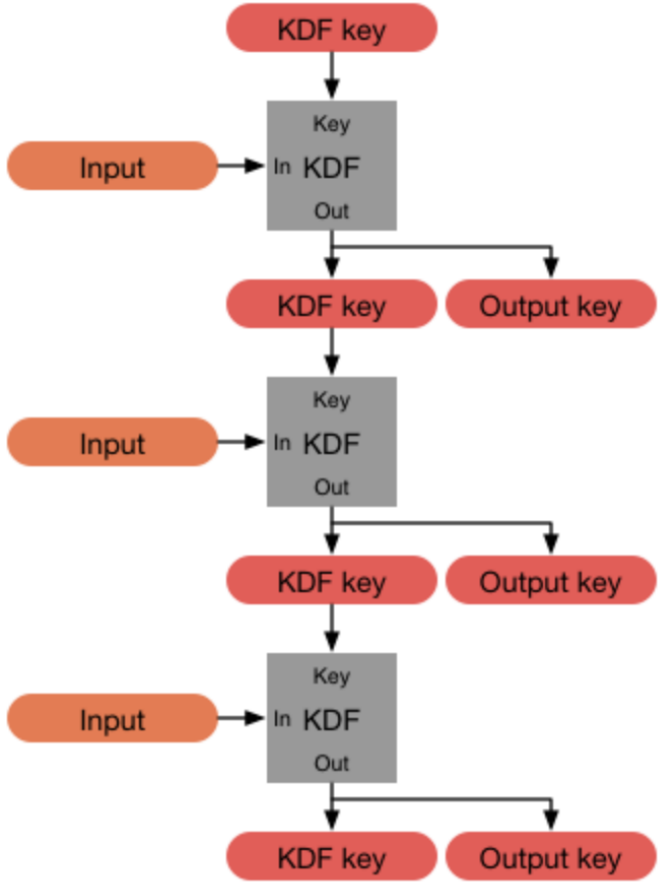
Also in Whatsapp
View Contact -> Encryption

Signal Protocol: KDF Ratchet

- Once we have a shared secret we can use it as a key
  - If somebody breaks it, they can read all messages

- Signal uses a **Key Derivation Function (KDF) chain**
  - KDF is like a hash function: computationally easy to go forwards, but difficulty to go backwards
  - Generate a new output key for each message!
  - One chain for sending and one for receiving

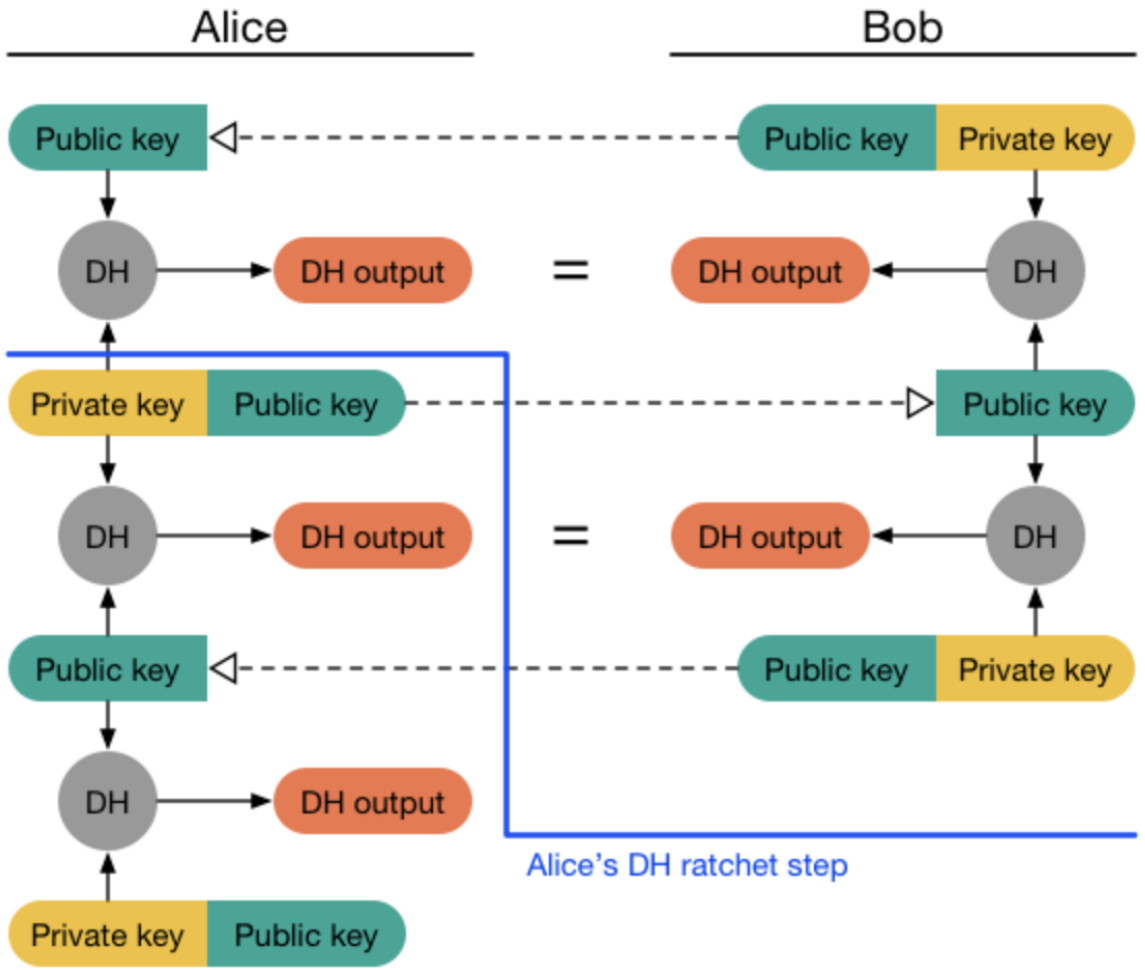
- **Resilience**: Without KDF key, output keys appear random
- **Forward Secrecy**: If a KDF key leaks, previous output keys cannot be derived
- However, if input is constant and somebody finds a KDF key, all future communication is compromised



Signal Protocol: DH Ratchet

- On each message exchange,
  - A new public key is generated
  - The public key is sent to the other side
  - A new DH Key Exchange is performed
  - A new shared secret is generated
  - The new shared secret is used as **input to the KDF ratchet**

- **Signal Double Ratchet**
  - **Future Secrecy**: Even if a KDF key leaks, new input from DH Ratchet protects future output keys



Summary

- Privacy is more than Confidentiality: anonymity, unlinkability, …
- Pseudonyms ≠ Anonymity
- Breaking sender anonymity through tracking
- Anonymity using Tor
- Confidential communication using Signal