Privacy Enhancing Technologies

Chapter: Anonymous Communication

Christiane Kuhn <christiane.kuhn@kit.edu>

Helmholtz Center for Applied Security Technology

Privacy Enhancing Technologies

Chapter: Anonymous Communication

Christiane Kuhn <christiane.kuhn@kit.edu>

Helmholtz Center for Applied Security Technology

Learning Goals

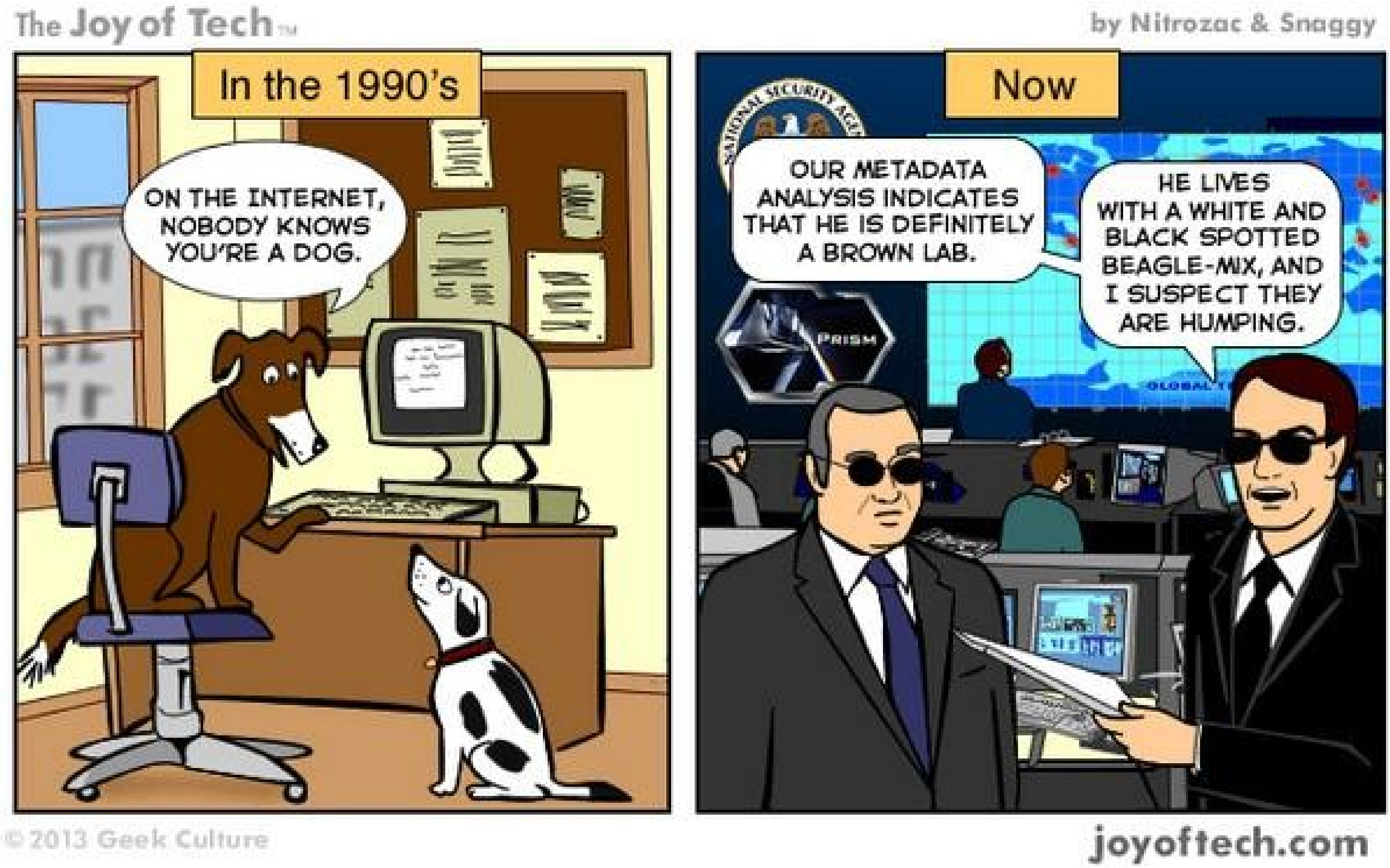
Karlsruhe Institute of Technology

- Understand the Problem
  - Motivation & Setting
  - Dimensions & Terminology

- Understand the Solution(-space)
  - Solution ideas and prominent protocols
  - Effects of design decisions

KASTEL

Motivation



The Joy of Tech™
by Nitrozac & Snaggy
In the 1990's
ON THE INTERNET, NOBODY KNOWS YOU'RE A DOG.
Now
OUR METADATA ANALYSIS INDICATES THAT HE IS DEFINITELY A BROWN LAB.
HE LIVES WITH A WHITE AND BLACK SPOTTED BEAGLE-MIX, AND I SUSPECT THEY ARE HUMPING.
PRISM
© 2013 Geek Culture
joyoftech.com

Motivation

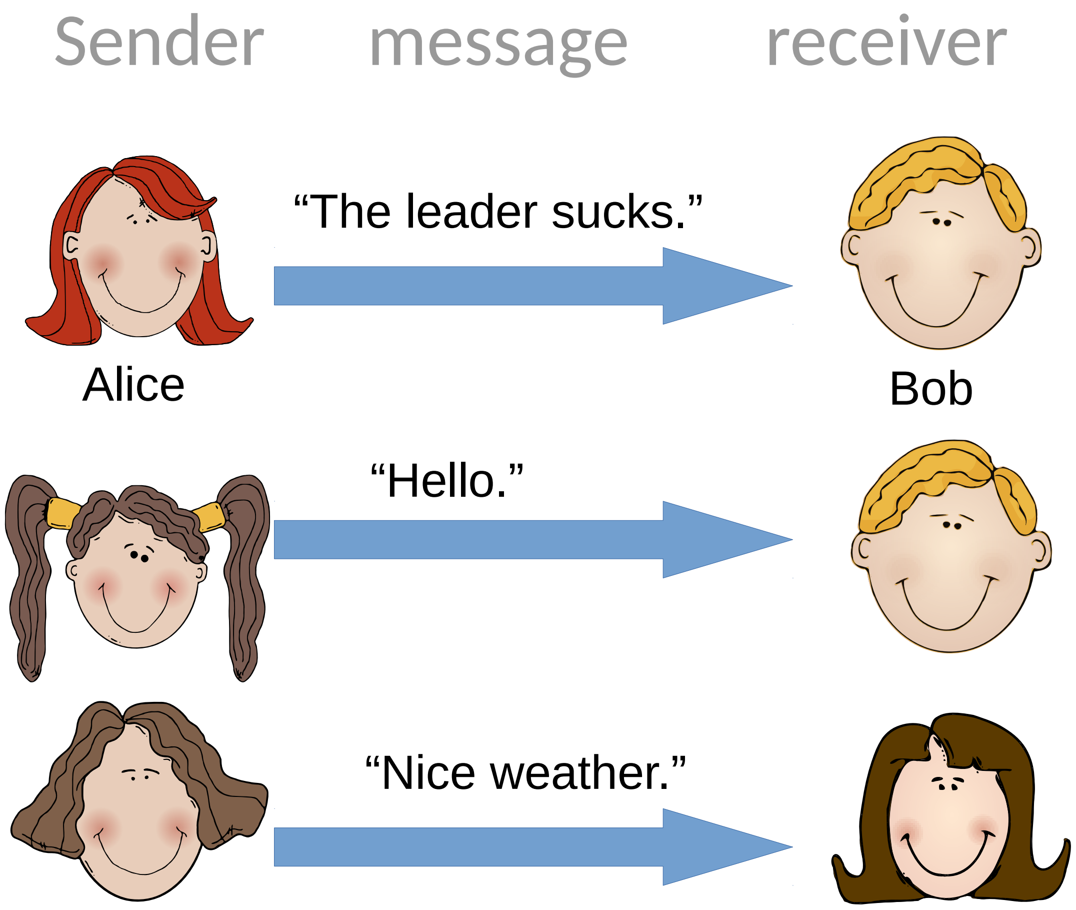
Karlsruhe Institute of Technology

- Protect Privacy in Communications to:
  - View sensitive content
  - Avoid impersonation
  - Avoid profiling and tracking by advertising companies (price discrimination)
  - Avoid profiling and tracking by governments (manipulation)
  - Guarantee freedom of speech
  - Enable applications: electronic voting, whistle blowing,…

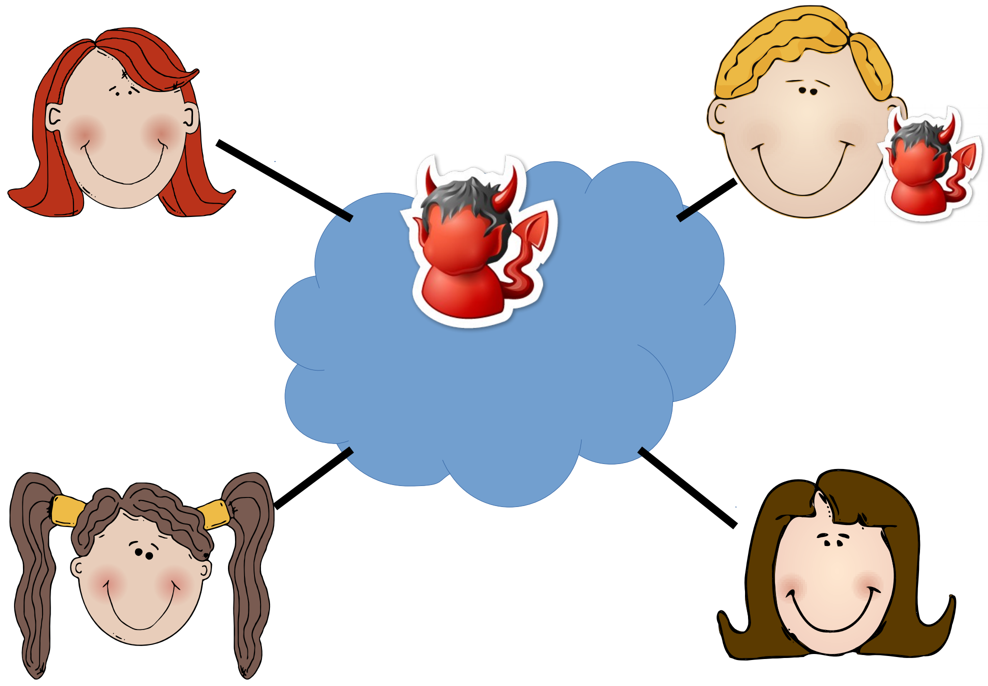
KASTEL

Setting

Communications that are happening



Network, on which they happen



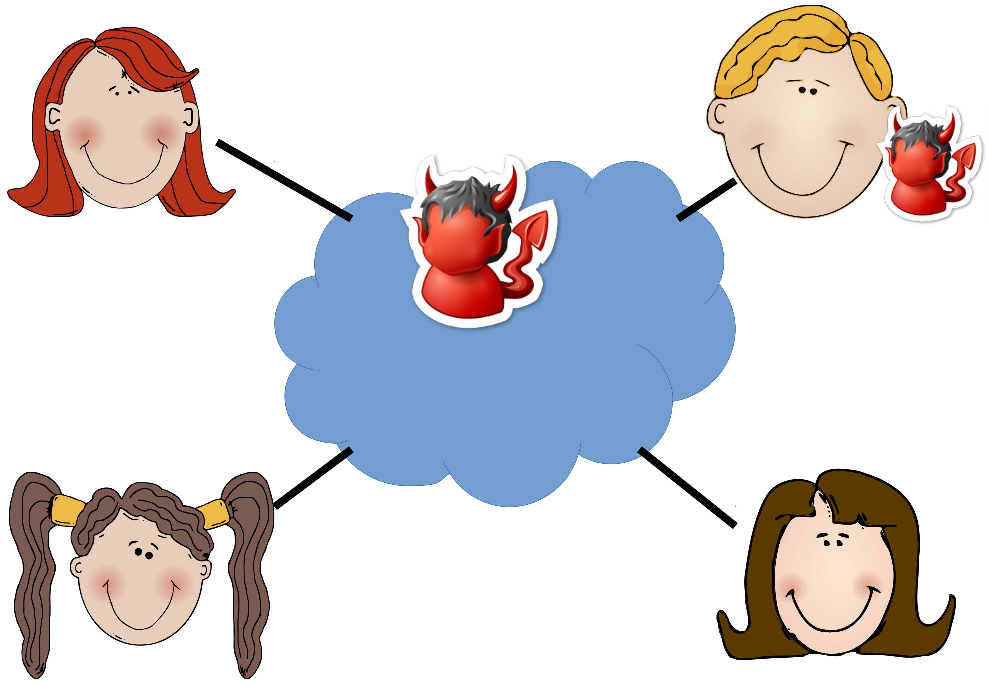
Does encryption protect Alice from the adversary?

Encryption is not enough

- Does not hide anything if the receiver is adversarial
- Does not hide meta data:
  - Sender-receiver relationships (IP addresses)
  - Activity
  - Cookies
  - Browser fingerprinting

  → all can be used to identify and profile users

✉ Encryption is an amazing tool, but not enough!

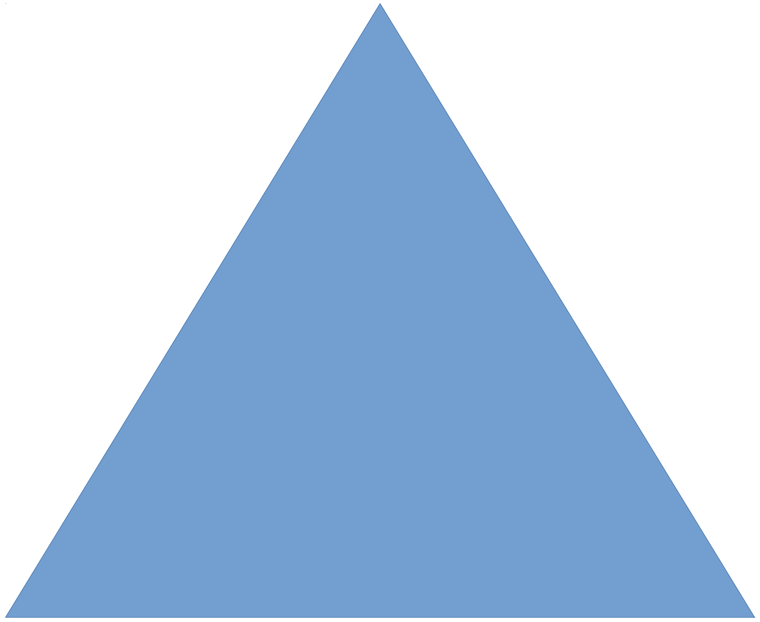


Learning Goals

- Understand the Problem
  - Motivation & Setting
  - Dimensions & Terminology

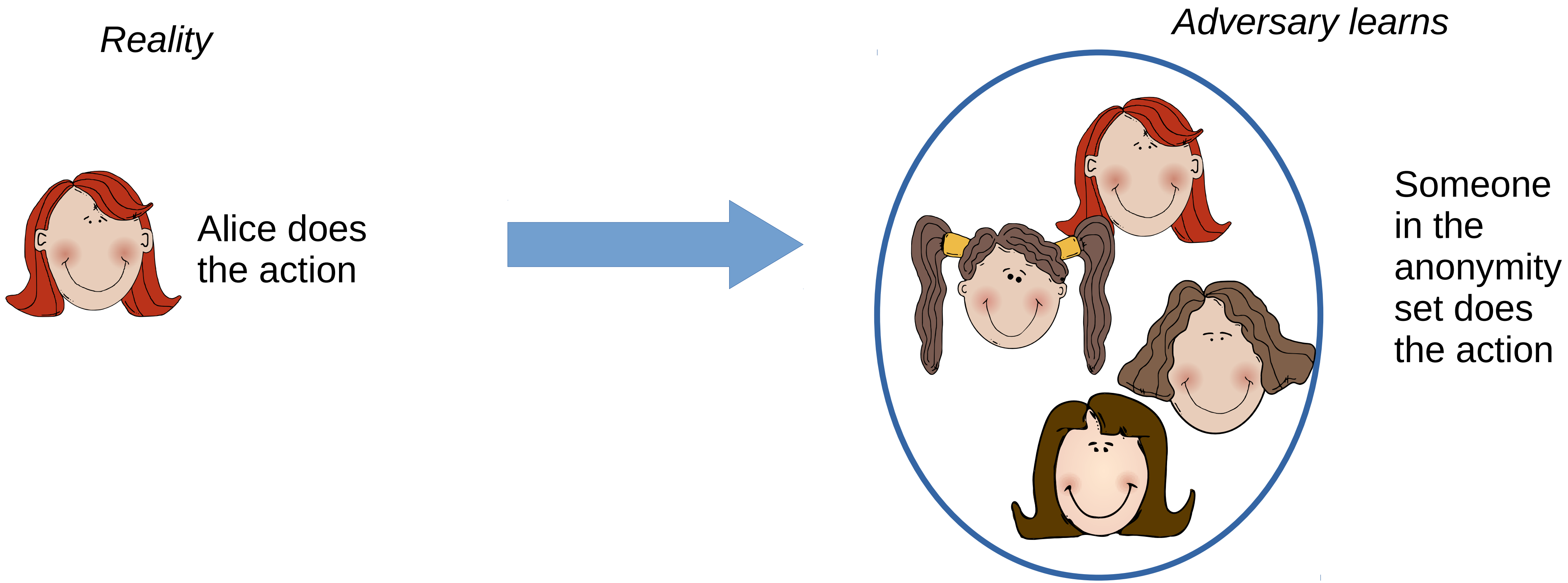
- Understand the Solution(-space)
  - Solution ideas and prominent protocols
  - Effects of design decisions

Criteria



What’s protected? Terminology

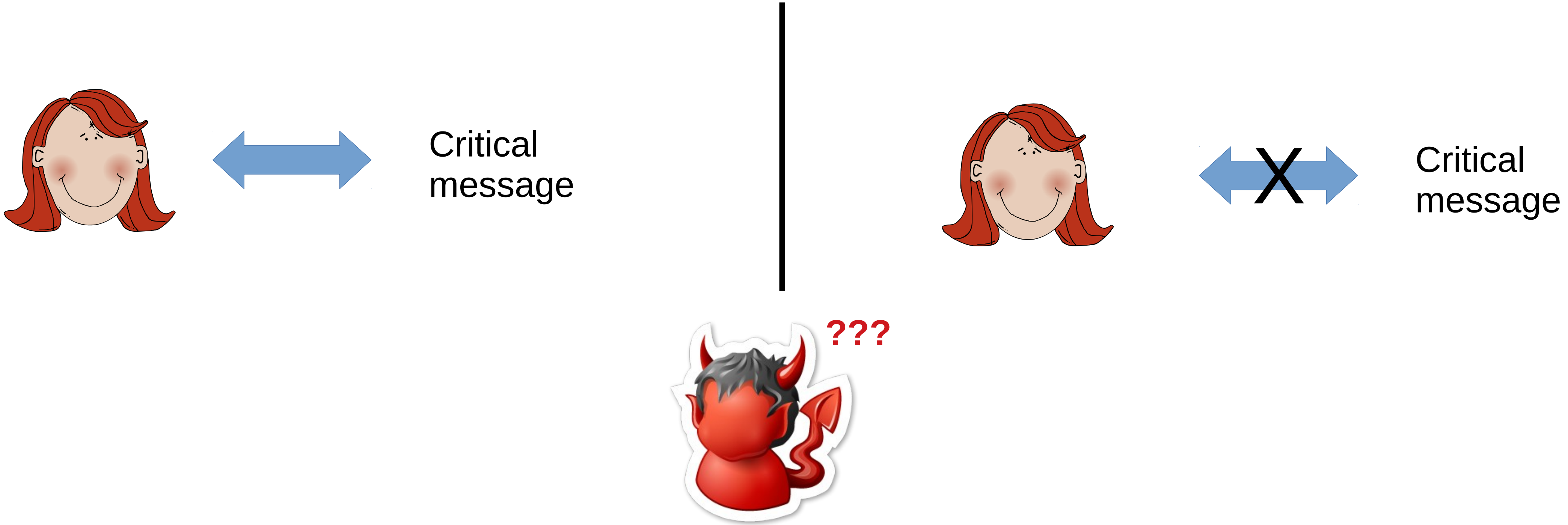
Anonymity: “Anonymity of a subject means that the subject is not identifiable within a set of subjects, the **anonymity set**. ”



https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html

What’s protected? Terminology

Unlinkability: “Unlinkability of two or more items [..] means that [..] the attacker cannot sufficiently distinguish whether these [items] are related or not.”



https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html

What’s protected? Terminology

Karlsruhe Institute of Technology

- **Undectectability**: “Undetectability of an item [..] means that the attacker cannot sufficiently distinguish whether it exists or not. “

Critical
message
sent

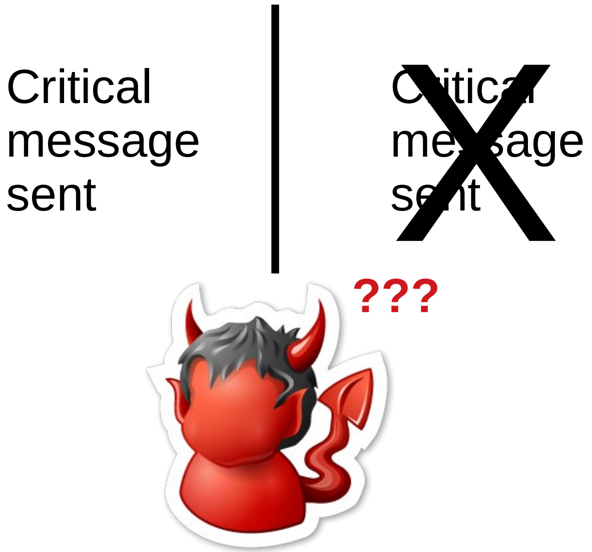
X Critical
message
sent

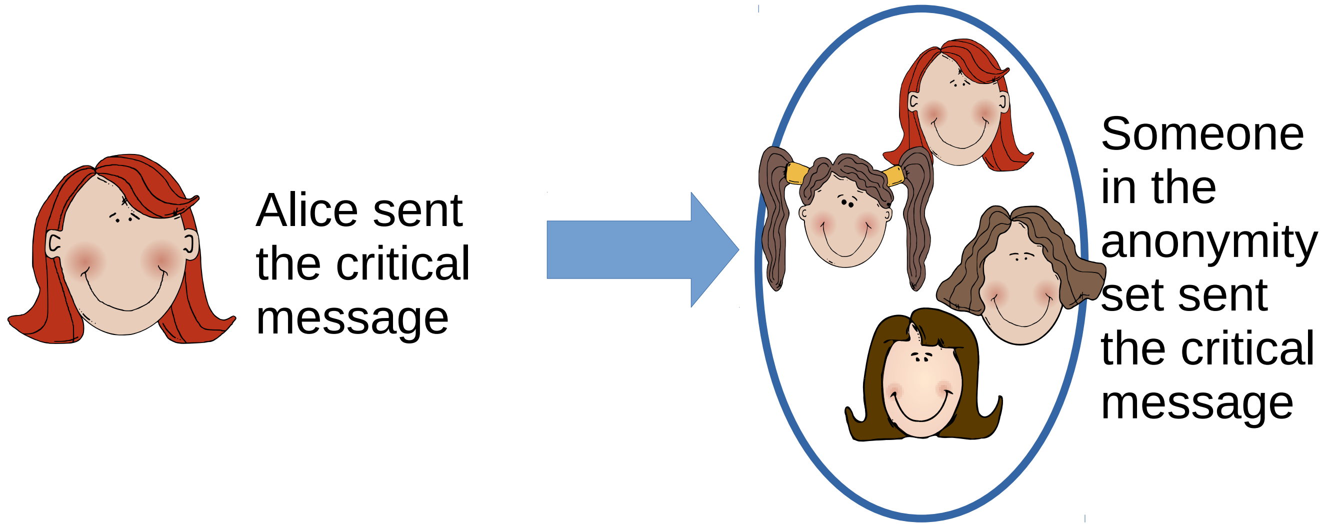
https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html

KASTEL

What's protected? Terminology

- Unobservability: "Unobservability of an item [..] means
  - undetectability of the [item] against all subjects uninvolved in it and
  - anonymity of the subject(s) involved in the [item] even against the other subject(s) involved in that [item]."





https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html

What’s protected?

Typically of interest: Sender, Receiver and Message

→ we’ll focus on sender protection for this lecture

- **Relationships**
  - e.g. Sender-Message Unlinkability (often called Sender Anonymity) – we do not learn who sends which message
  - e.g. Sender-Receiver Unlinkability (often called Relationship Anonymity) – we do not learn who communicates with whom
- **Activity**
  - e.g. Sender Unobservability – we do not learn who sends something

More protection goals possible

What’s protected?

Typically of interest: Sender, Receiver and Message

→ we’ll focus on sender protection for this lecture

- **Relationships**
  - e.g. Sender-Message Unlinkability (often called Sender Anonymity) – we do not learn who sends which message
  - e.g. Sender-Receiver Unlinkability (often called Relationship Anonymity) – we do not learn who communicates with whom
- **Activity**
  - e.g. Sender Unobservability – we do not learn who sends something

More protection goals possible

Is Sender-Message Unlinkability stronger than Sender Unobservability?

What’s protected?

Typically of interest: Sender, Receiver and Message

→ we’ll focus on sender protection for this lecture

- **Relationships**
  - e.g. Sender-Message Unlinkability (often called Sender Anonymity) – we do not learn who sends which message
  - e.g. Sender-Receiver Unlinkability (often called Relationship Anonymity) – we do not learn who communicates with whom
- **Activity**
  - e.g. Sender Unobservability – we do not learn who sends something

More protection goals possible

Is Sender-Receiver Unlinkability stronger than Sender Unobservability?

What's protected?

Typically of interest: Sender, Receiver and Message

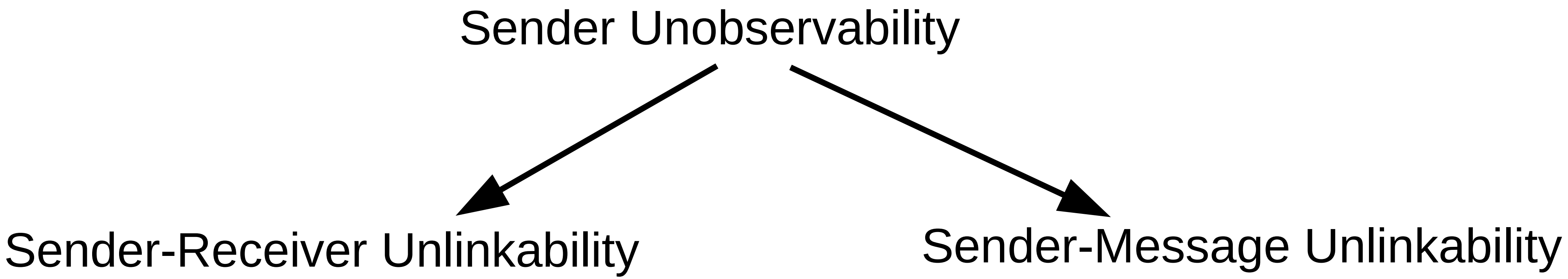
→ we'll focus on sender protection for this lecture

- **Relationships**
  - e.g. Sender-Message Unlinkability (often called Sender Anonymity) – we do not learn who sends which message
  - e.g. Sender-Receiver Unlinkability (often called Relationship Anonymity) – we do not learn who communicates with whom
- **Activity**
  - e.g. Sender Unobservability – we do not learn who sends something

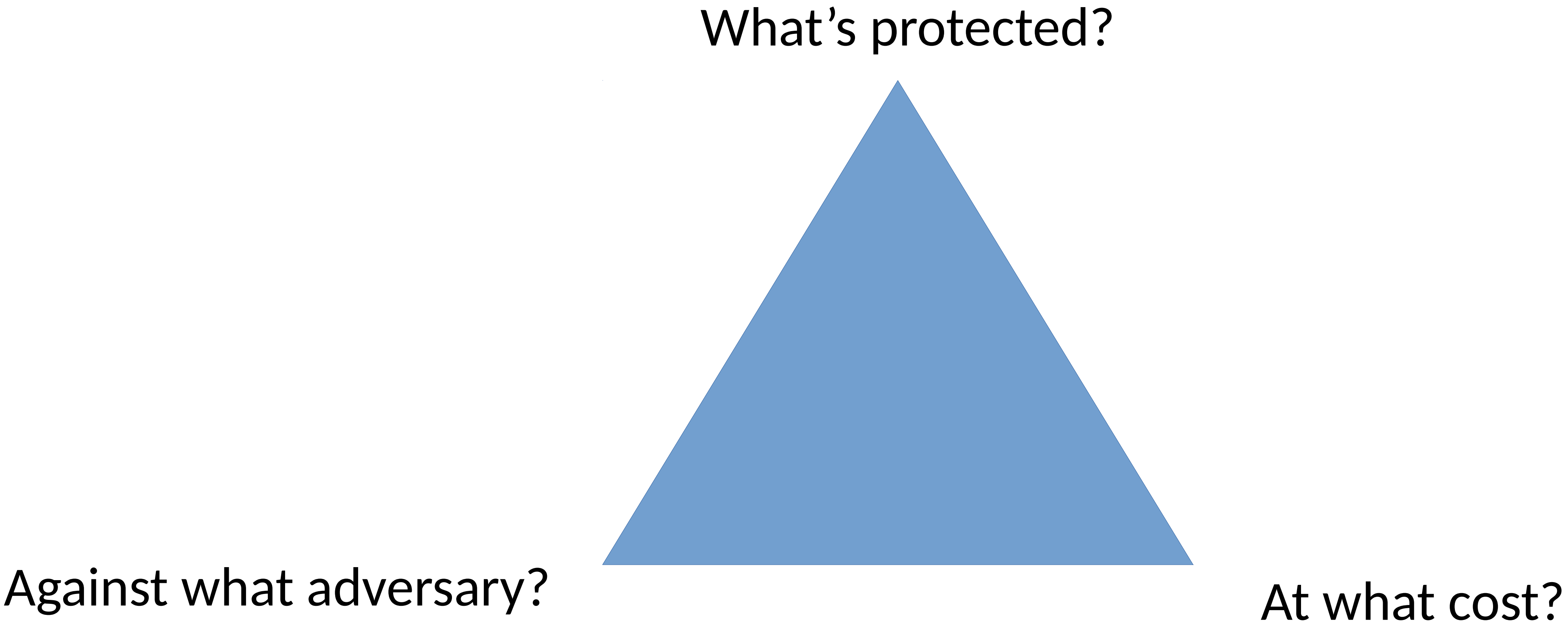
More protection goals possible

Is Sender-Receiver Unlinkability
stronger than Sender-Message Unlinkability?

What’s protected?



Criteria



Against what adversary?

Karlsruhe Institute of Technology

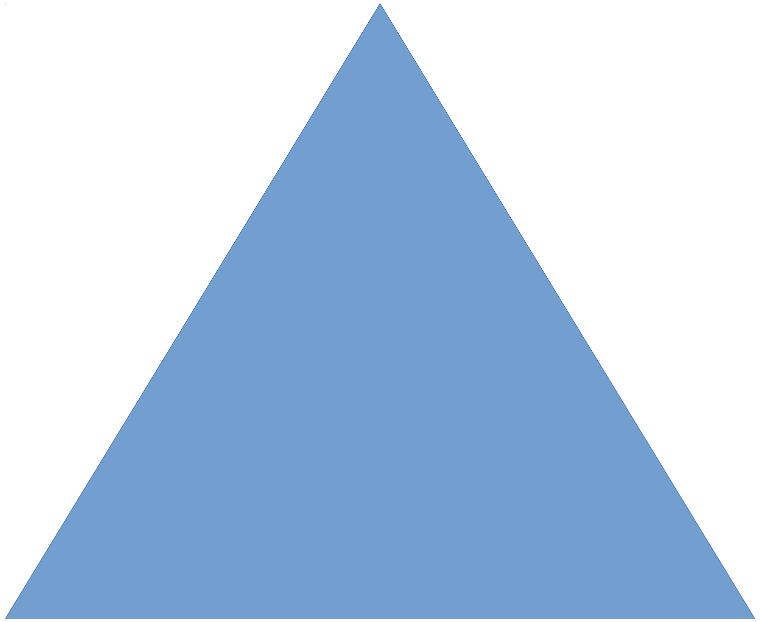
- Area? Local vs. Global, Links vs. Nodes etc.
- Actions? Eavesdropping (Passive)/ Modification, Dropping, Delay (Active)

  → we'll focus on passive adversaries for this lecture
- Participant? Internal vs. External
- Time? Temporary vs. Permanent
- Change resources/strategy? Static vs. Adaptive
- Restricted computation power?

KASTEL

Criteria

What's protected?



Against what adversary?

At what cost?

At what cost?

- Latency
- Bandwidth

- Functionality
- Other security goals (availability)
- Additional assumptions (public key infrastructure etc.)

Learning Goals

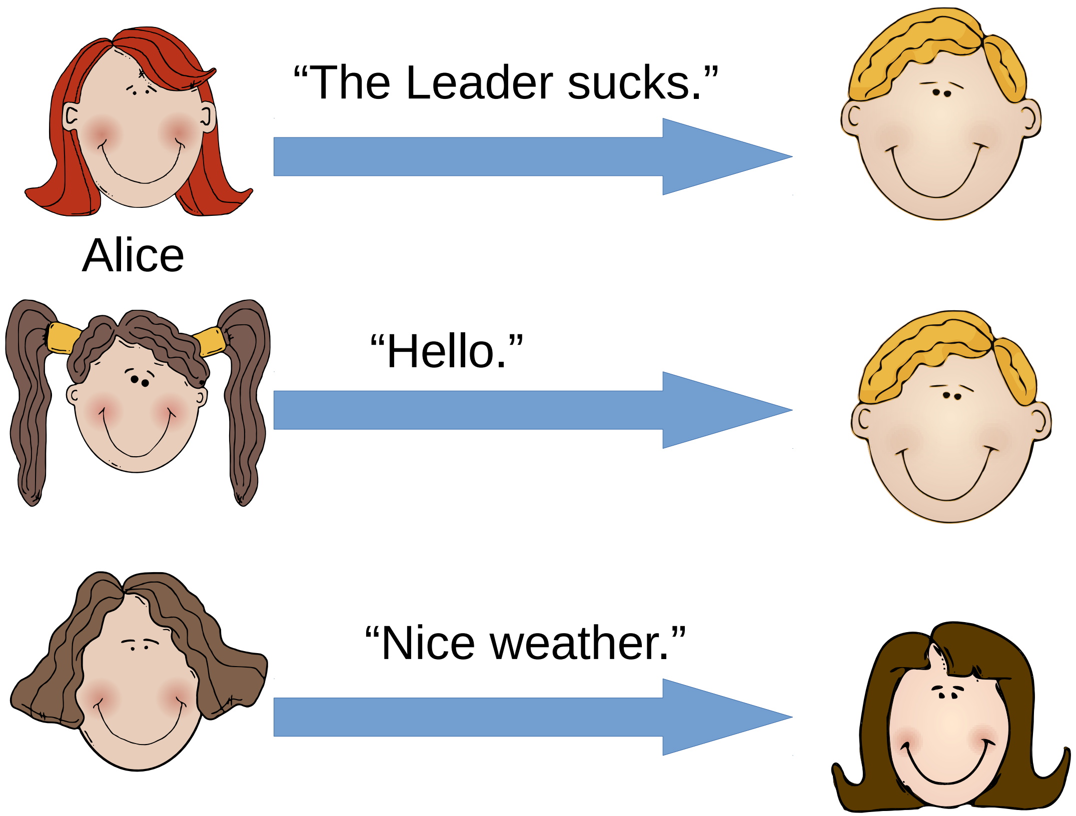
- Understand the Problem
  - Motivation and Setting
  - Dimensions and Terminology

- Understand the Solution(-space)
  - Solution ideas and prominent protocols:
    - Random Walk
    - Onion Routing
    - Mix Networks
    - Dummy Traffic
    - DC Networks
  - Effects of design decisions

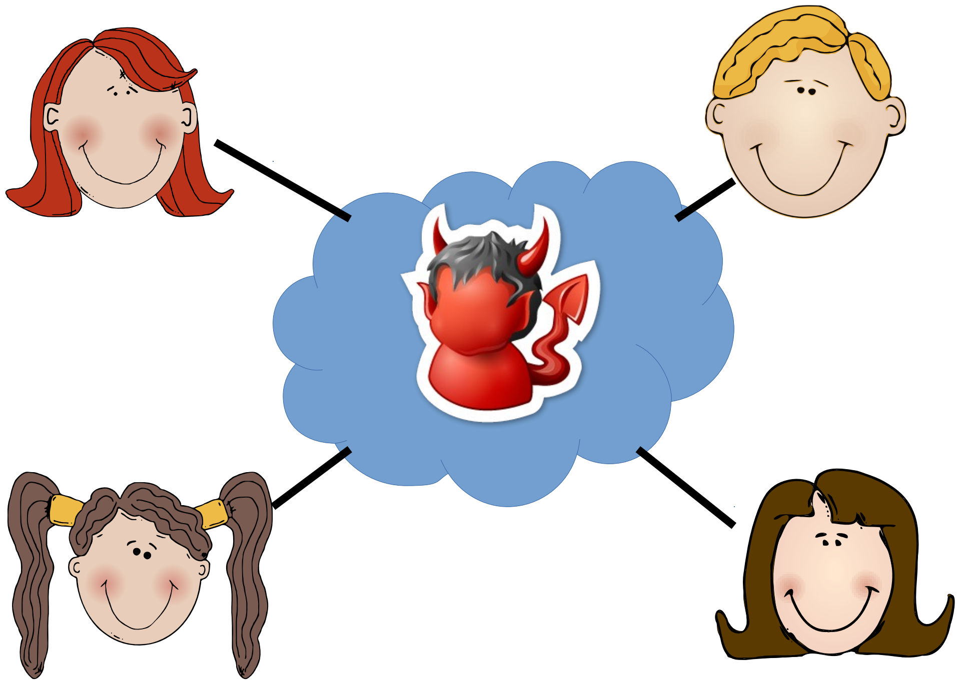
Setting

The Communications that happen

Sender message receiver

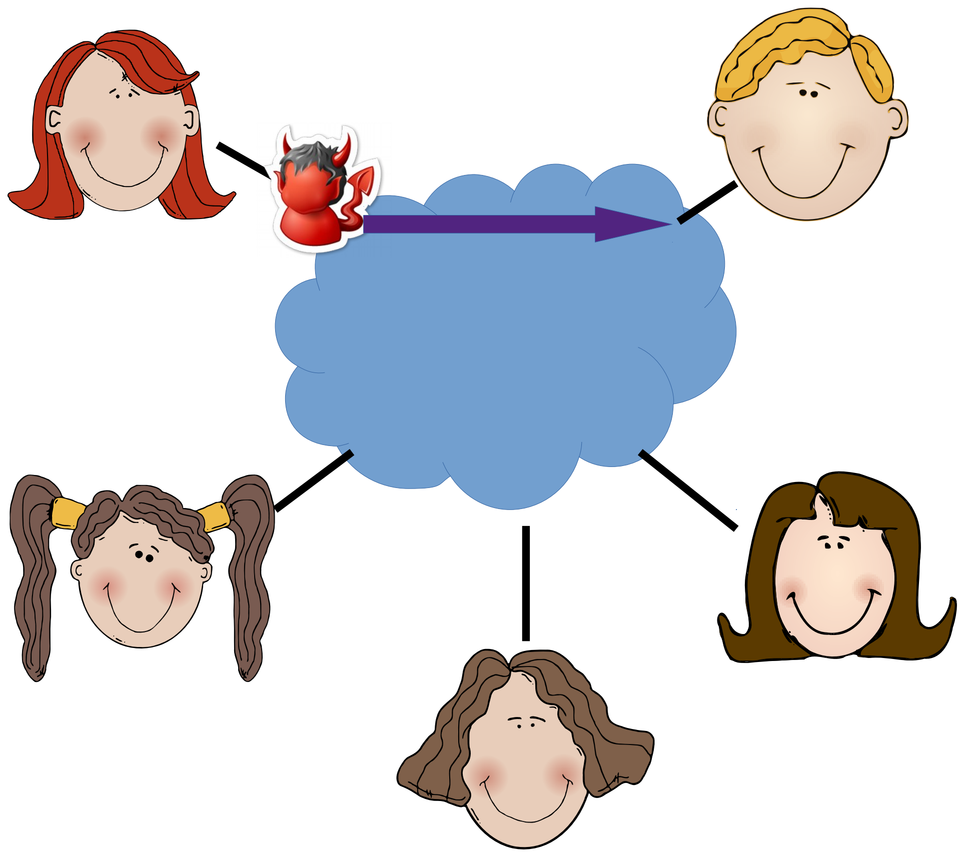


The network on which they happen



Without any protection

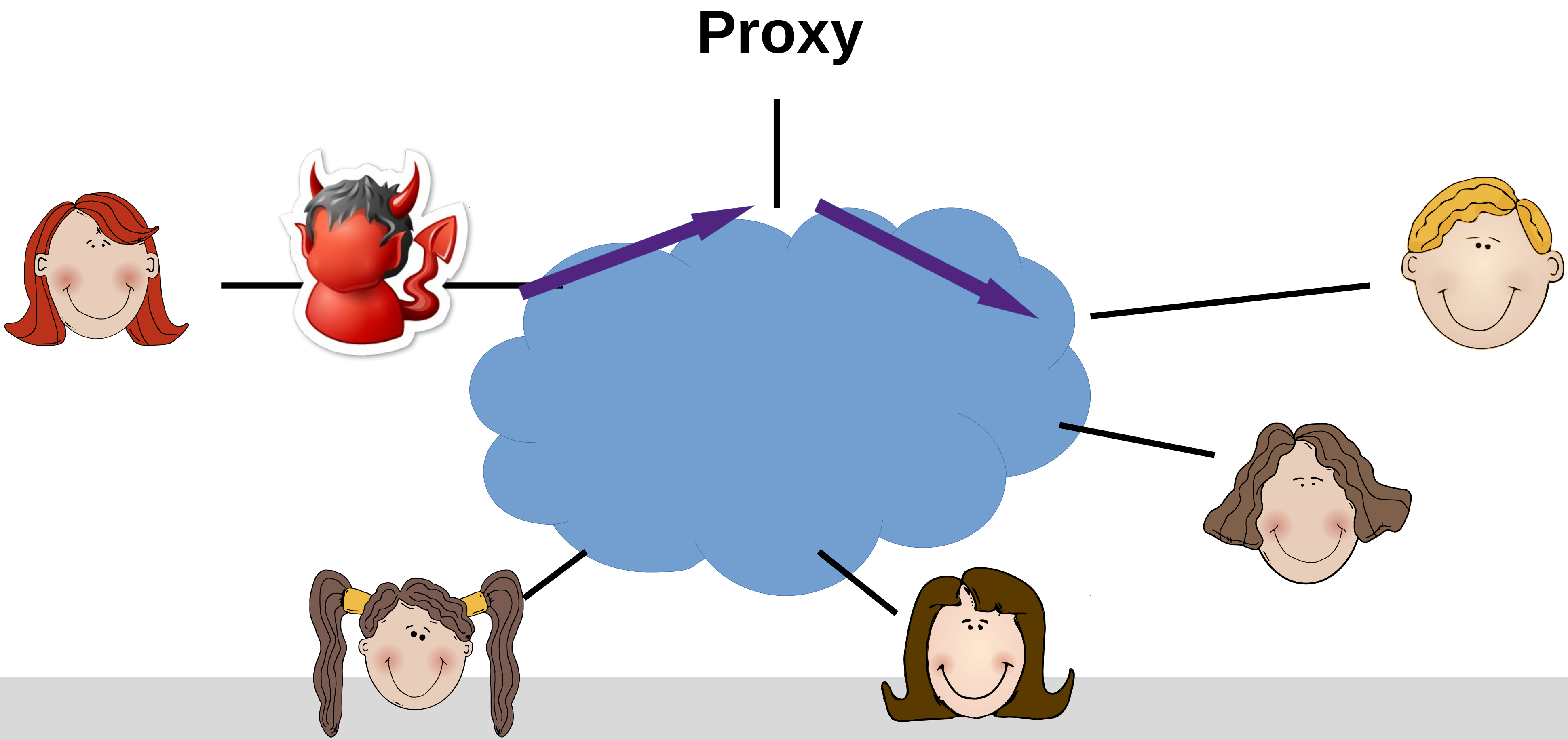
- Direct connection observable



Using a Proxy

Principle 1: Indirection

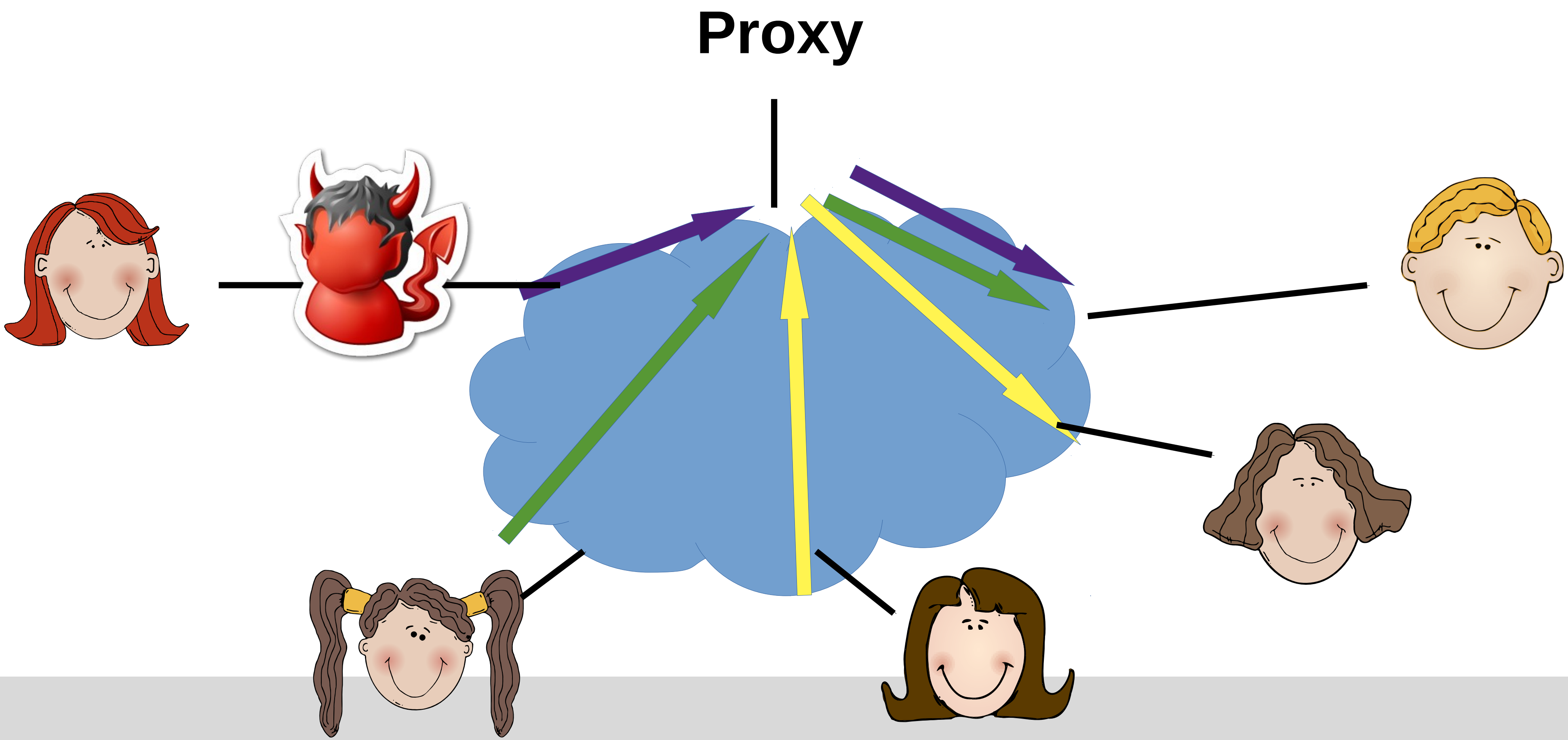
Alice sends message and receiver address to a proxy, who then forwards the message to the receiver



Using a Proxy

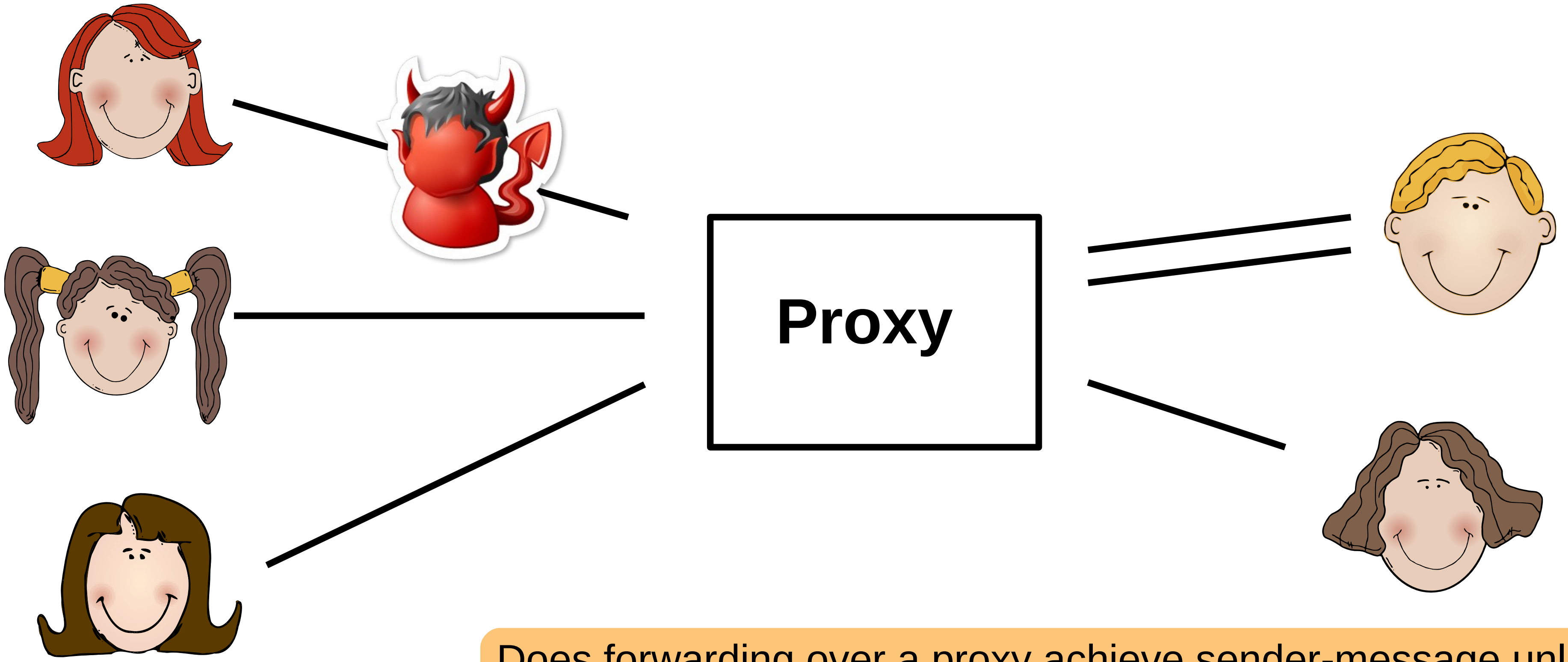
Principle 1: Indirection

Alice sends message and receiver address to a proxy, who then forwards the message to the receiver, all other senders do the same



Using a Proxy

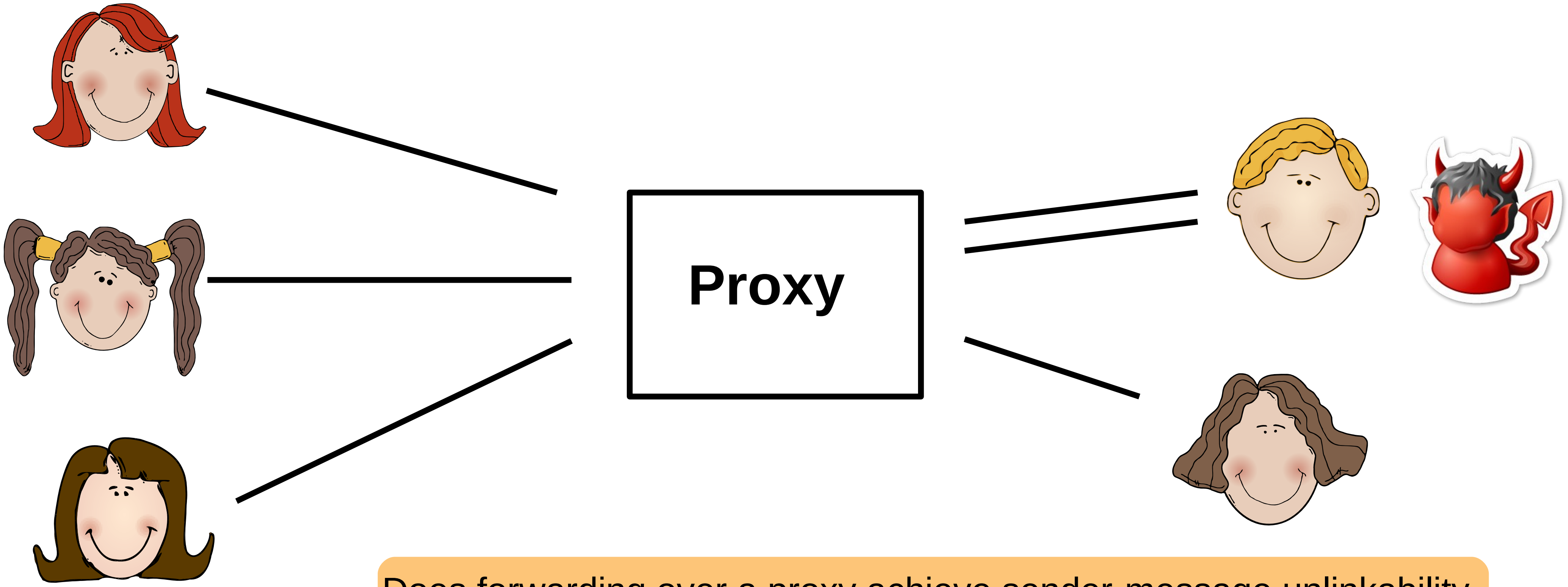
Principle 1: Indirection



Does forwarding over a proxy achieve sender-message unlinkability against a passive, local adversary at the senders?

Using a Proxy

Principle 1: Indirection

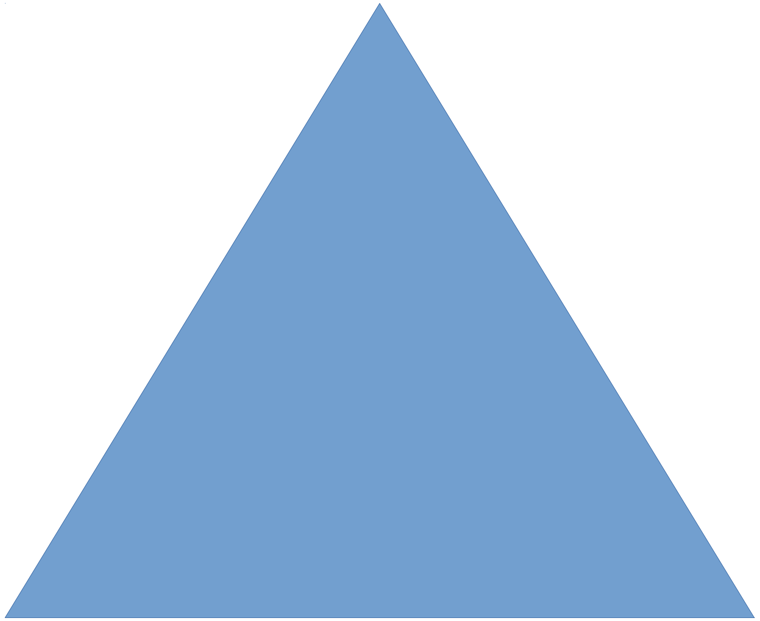


Does forwarding over a proxy achieve sender-message unlinkability against a corrupt, passive receiver?

Using a Proxy

Sender-Message Unlinkability

Sender-Receiver Unlinkability



Passive receiver as adversary

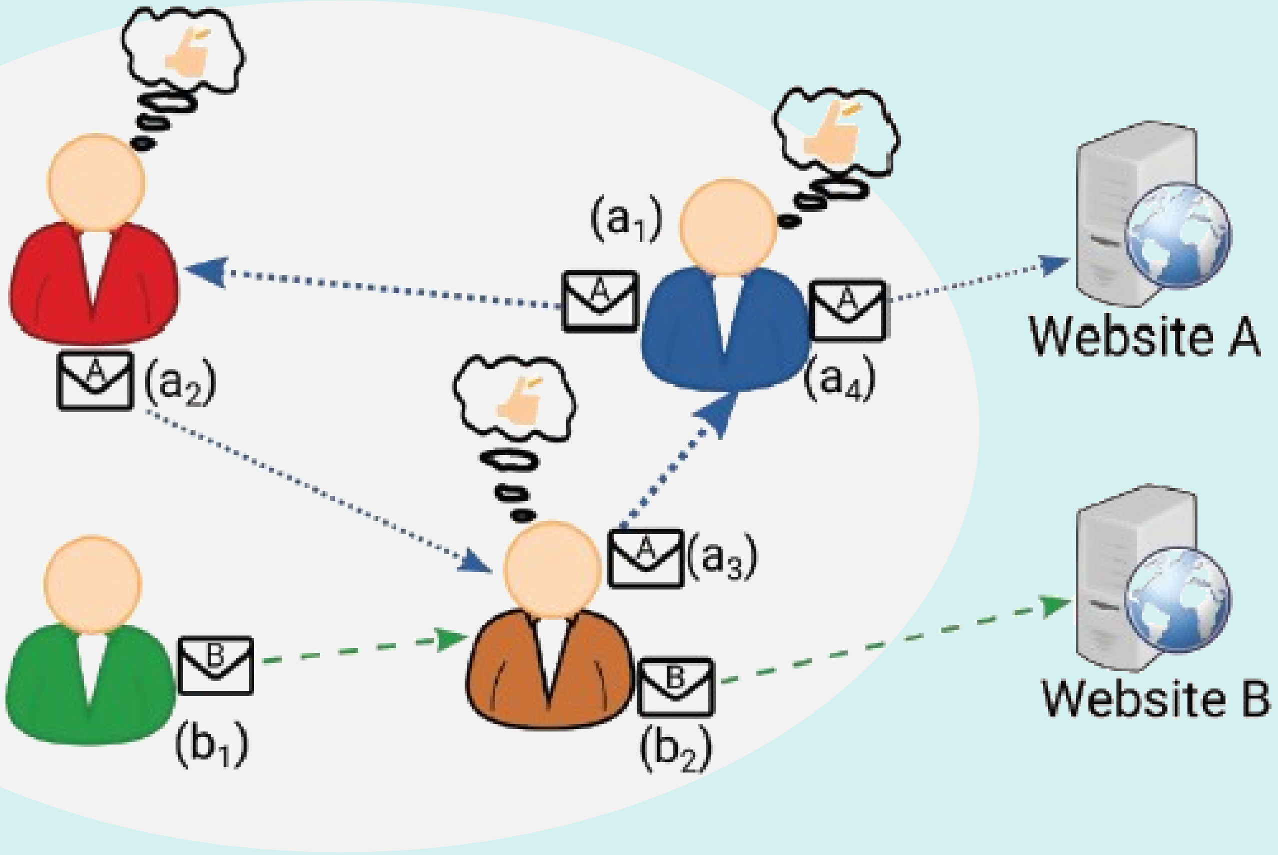
Slightly higher latency
need a proxy

Random Walk Protocols

- Typically use peer-to-peer network structure
- Forward message to randomly selected neighbor
- *Example: Crowds* (1998) for anonymous web browsing

Reiter, Michael K., and Aviel D. Rubin. "Crowds: Anonymity for web transactions." ACM transactions on information and system security (TISSEC) 1.1 (1998): 66-92.

Random Walk concept (Crowds)



Crowd

Crowd Membership is controlled by special nodes (*blenders*))

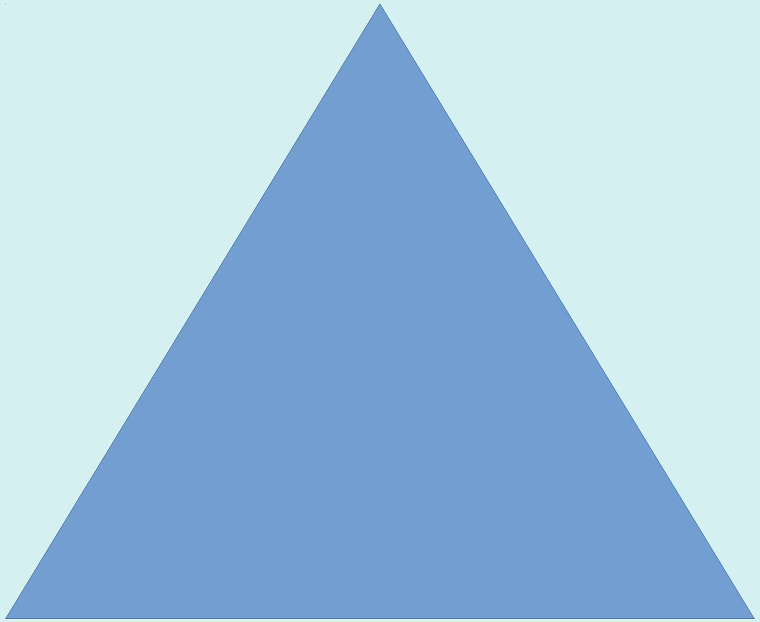
Crowds

- All nodes are grouped into „crowds“
- Nodes within a crowd might connect to each other for relaying a communication:
  - user randomly selects a node and sends her message (i.e., website request)
  - this node flips a biased coin to decide whether to send the request directly to the receiver or to forward it to another node selected uniform at random,
  - this continues until the message arrives at the destination.
  - The server replies are relayed through the same nodes in reverse order.

Can an internal adversary, corrupting n-2 participants, identify the sender of a message (with high probability)?

Crowds

Sender Unobservability



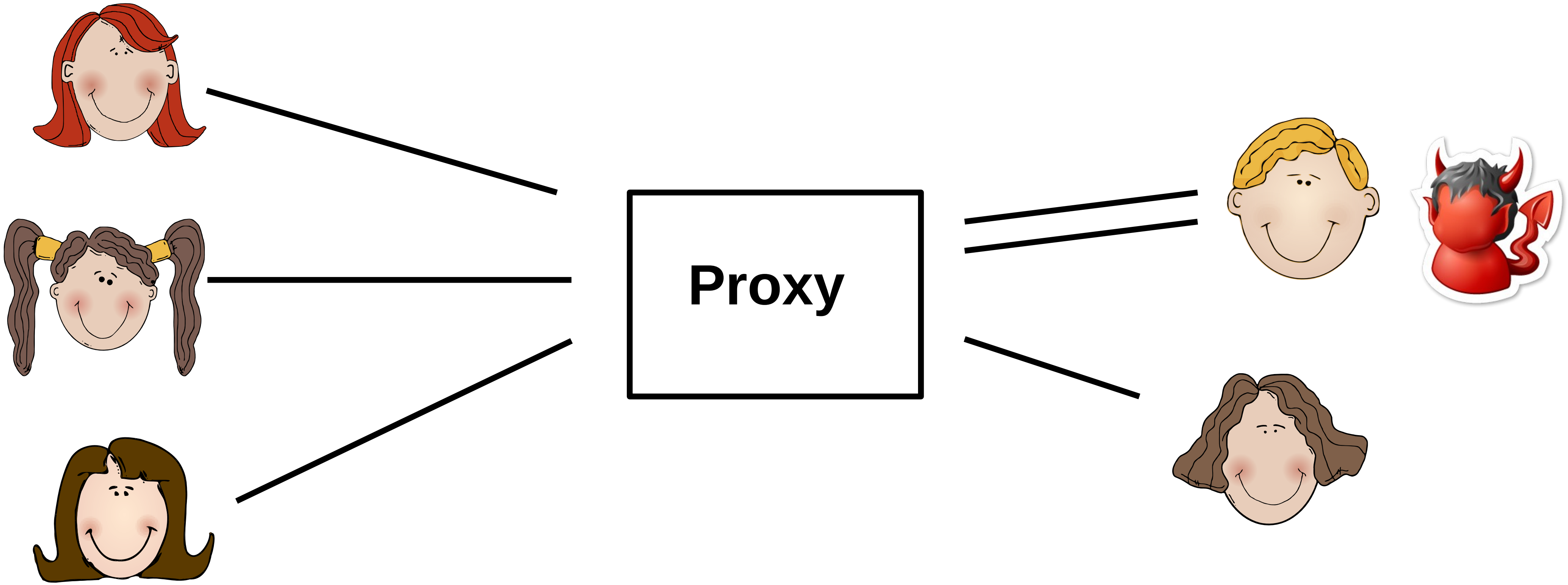
Passive **external** receiver

Higher latency
Management overhead
Availability risk (blenders)

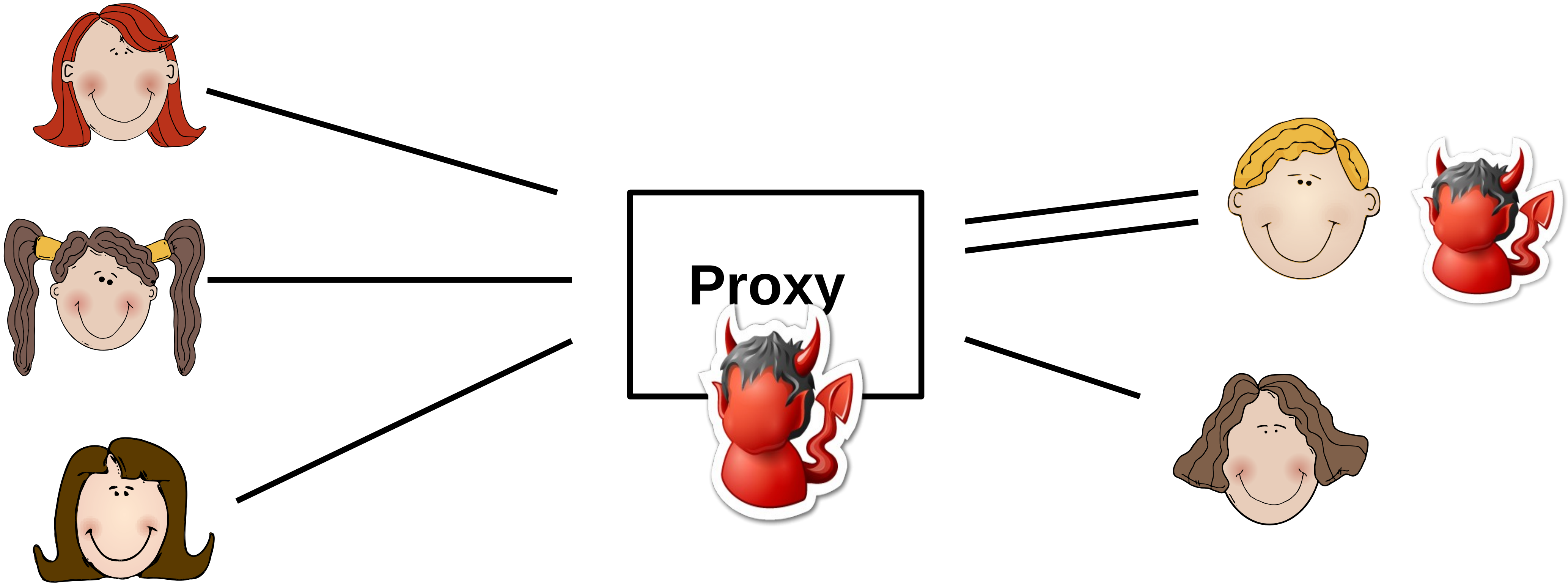
Summary Random walk

- Non-deterministic route selection
- Protection against external adversary
- Internal adversary improves estimation of sender based on timing information (predecessor attack)
  - Crowds is a representative example
    - Semi de-centralized
    - ✉ blenders are single points of failure

Using a Proxy



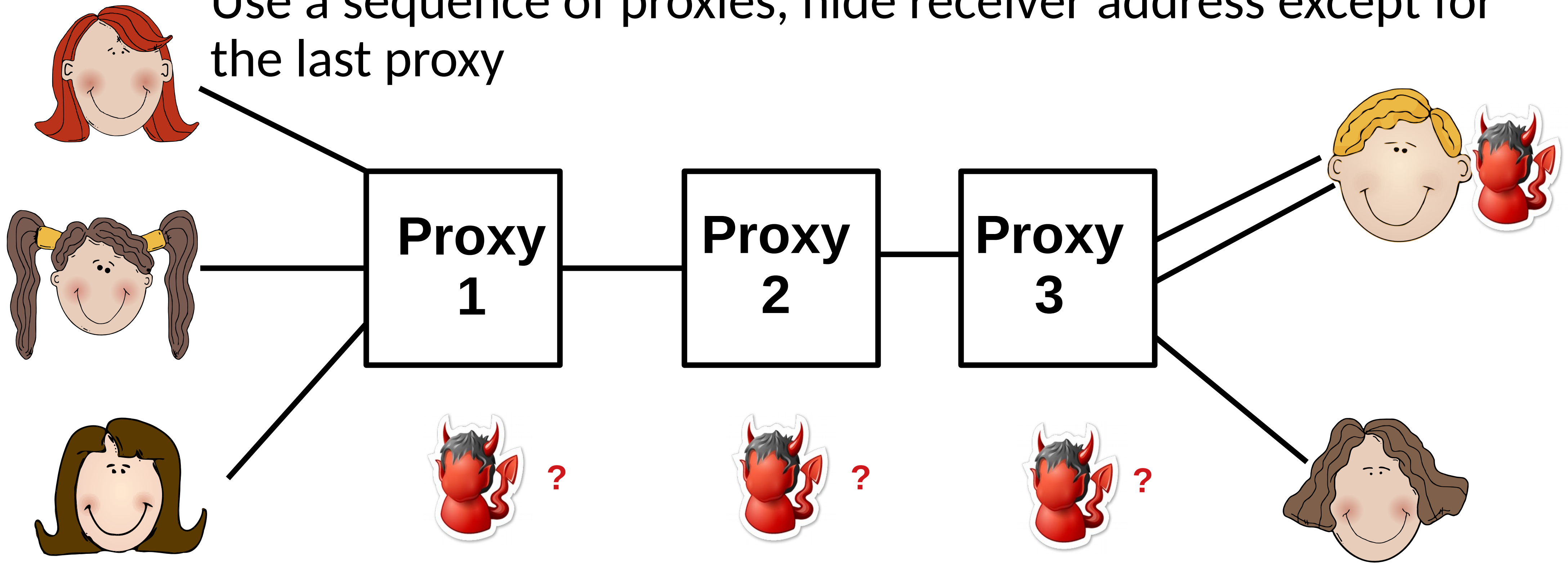
Using a Proxy



Using a Proxy Chain

Principle 2: Distribution of Trust

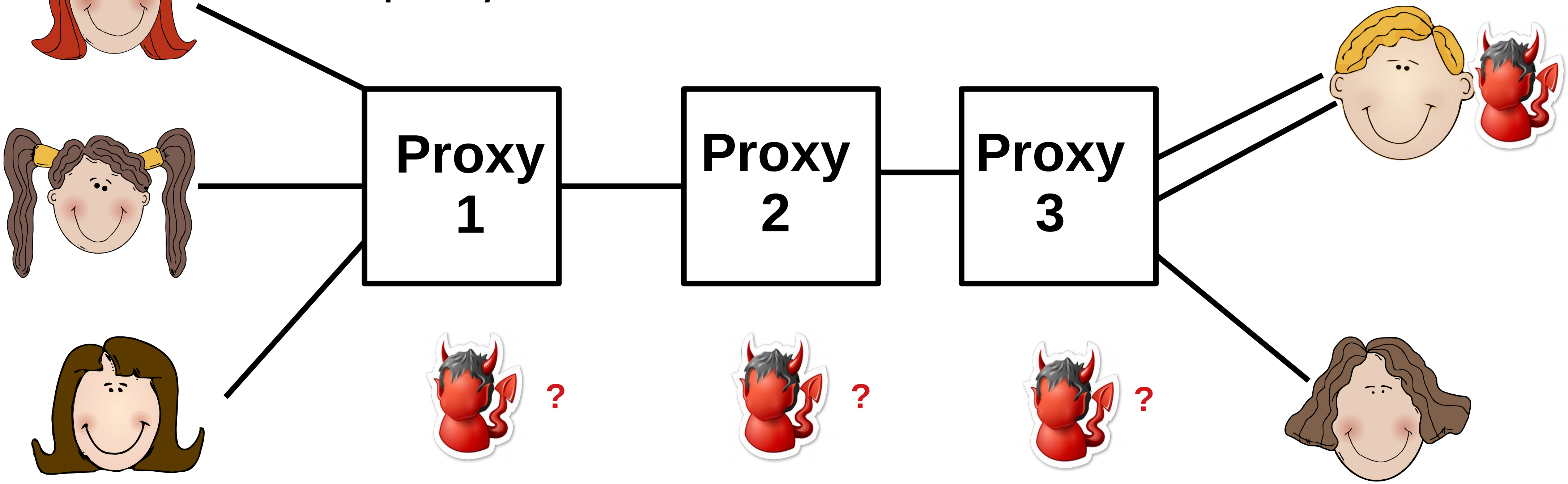
Use a sequence of proxies, hide receiver address except for the last proxy



Using a Proxy Chain

Principle 2: Distribution of Trust

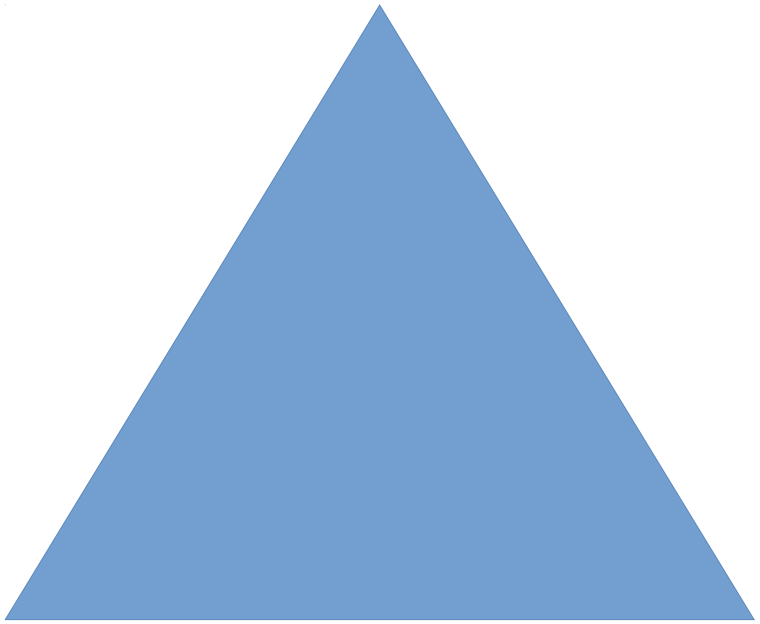
Use a sequence of proxies, hide receiver address except for the last proxy



How many proxies need to be **corrupt** to break sender-**receiver** unlinkability against a corrupt receiver?

Using a Proxy Chain

Sender-Message Unlinkability
Sender-Receiver Unlinkability



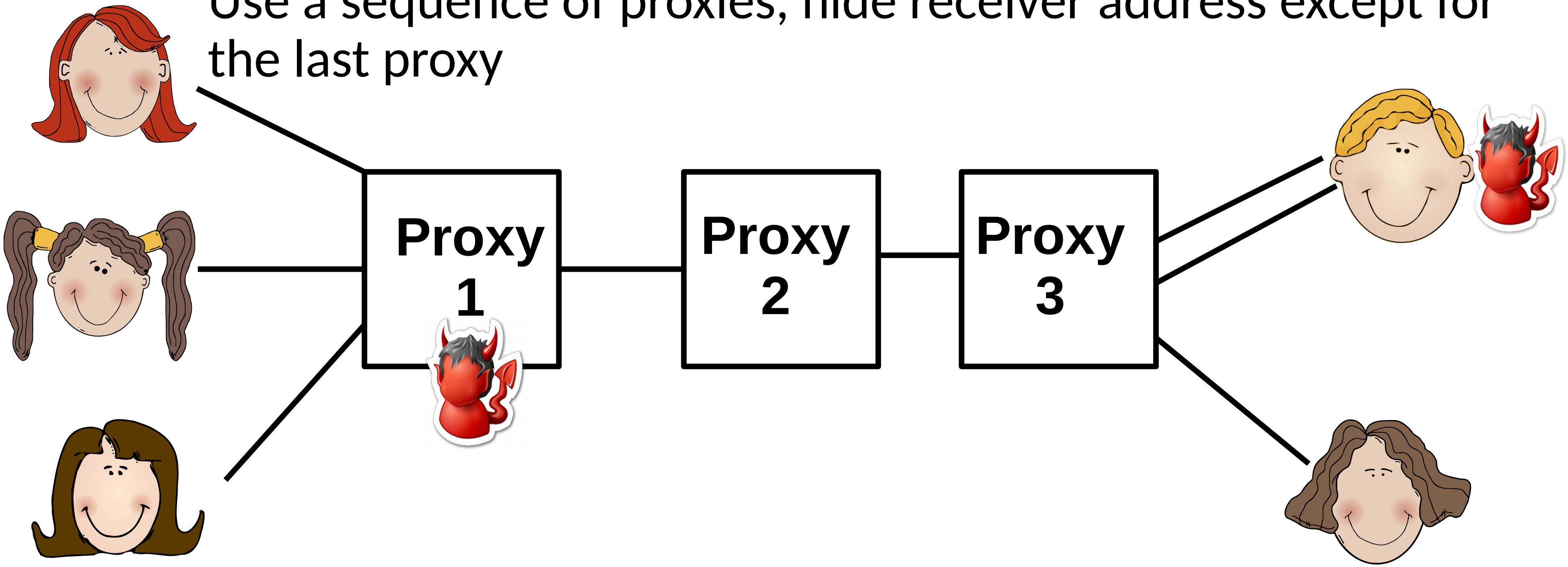
Passive corrupt receiver +
All except first proxy

higher latency
need multiple proxies
Computation overhead to hide receiver address

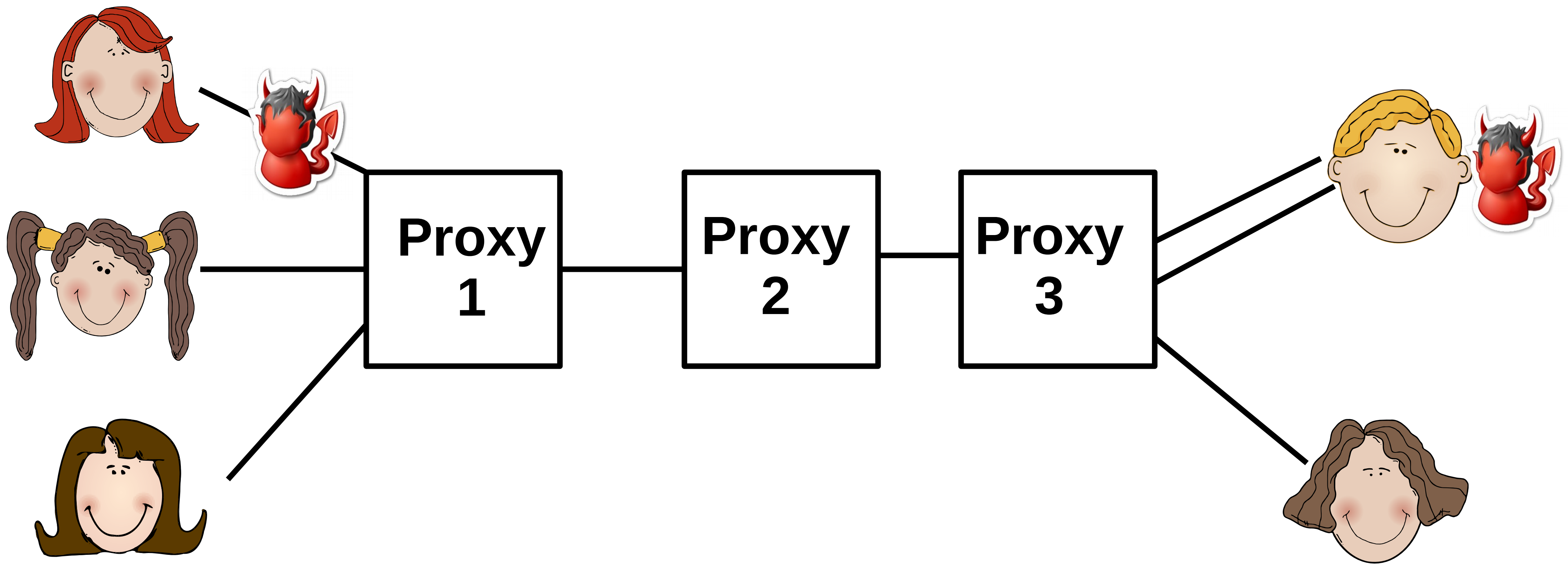
Using a Proxy Chain

Principle 2: Distribution of Trust

Use a sequence of proxies, hide receiver address except for the last proxy

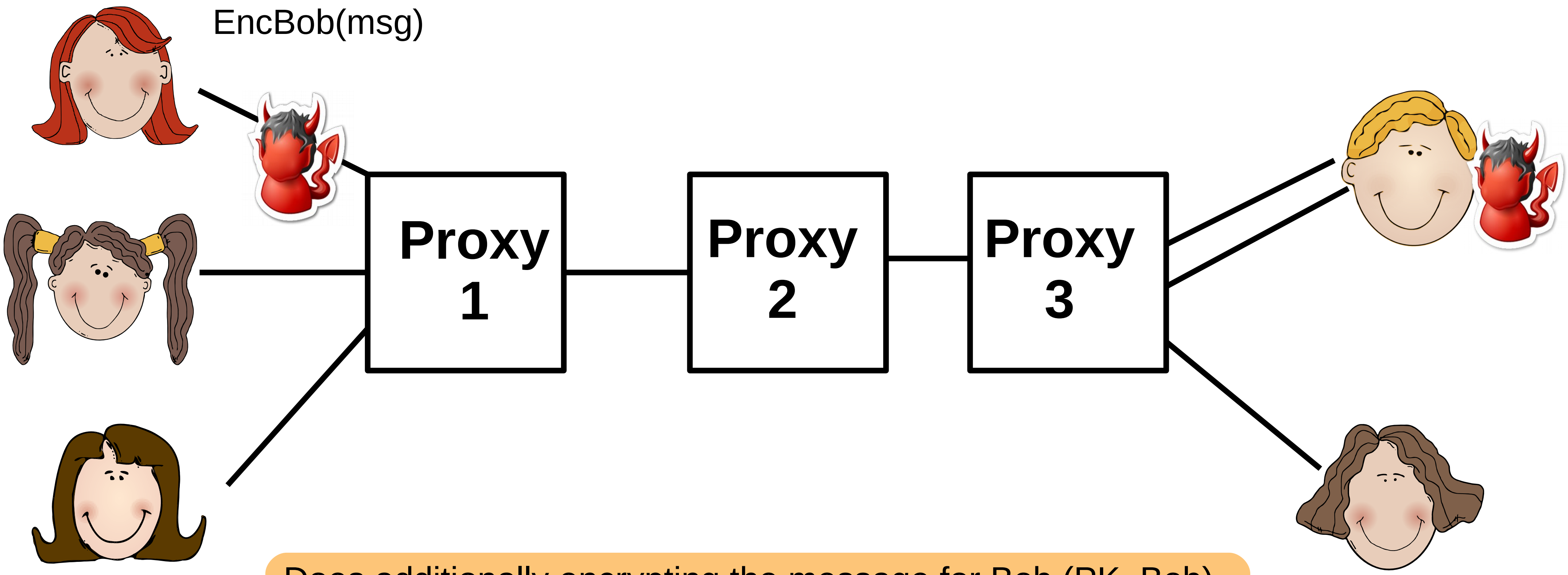


Using a Proxy Chain



Linking via the message works also if adversary is on first link

Adding end-to-end encryption

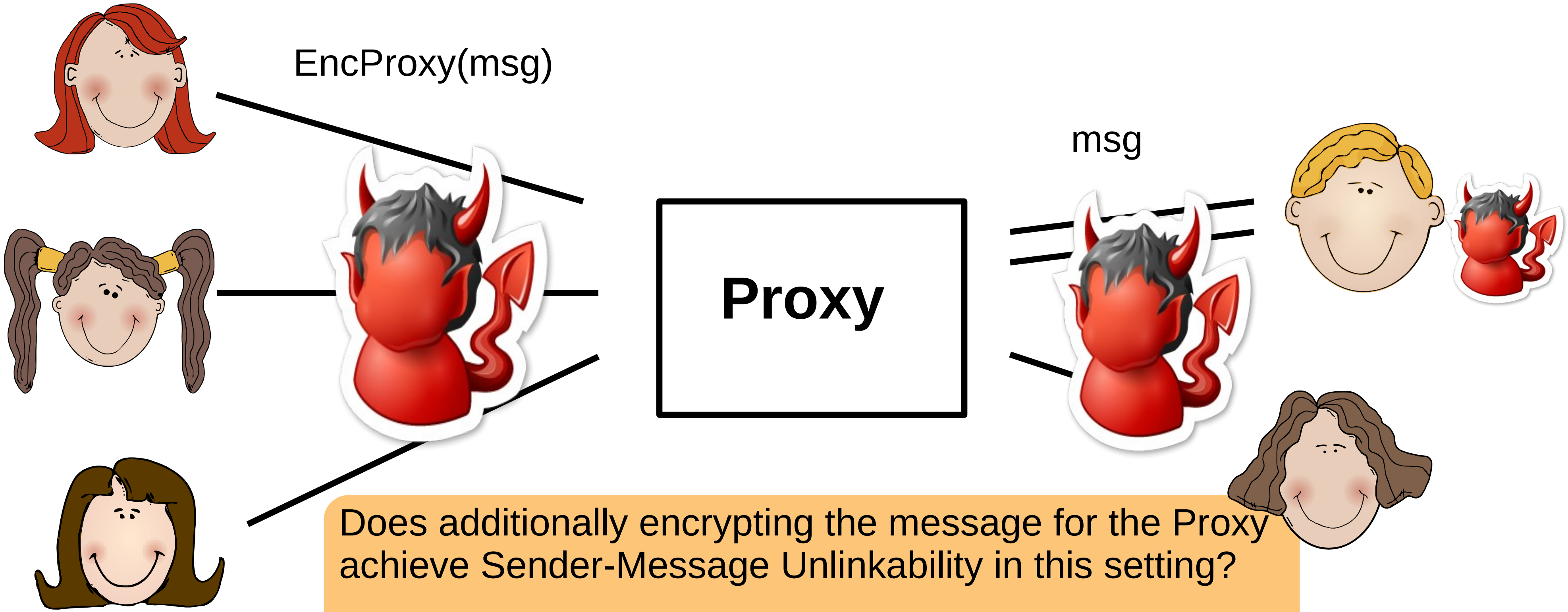


Does additionally encrypting the message for Bob (PK_Bob) achieve Sender-**Message** Unlinkability?

Adding Encryption

Principle 3: Unlink Observations

Principle 4: Randomize Observations

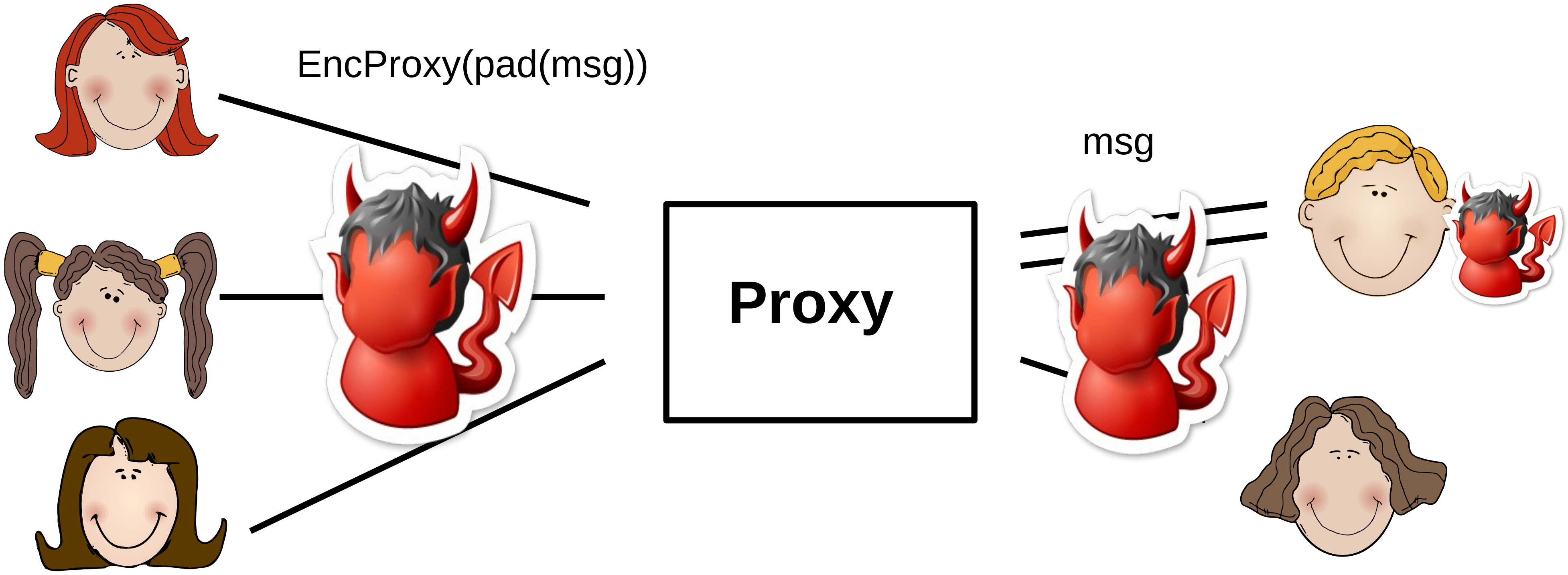


Does additionally encrypting the message for the Proxy achieve Sender-Message Unlinkability in this setting?

Padding against linking based on length

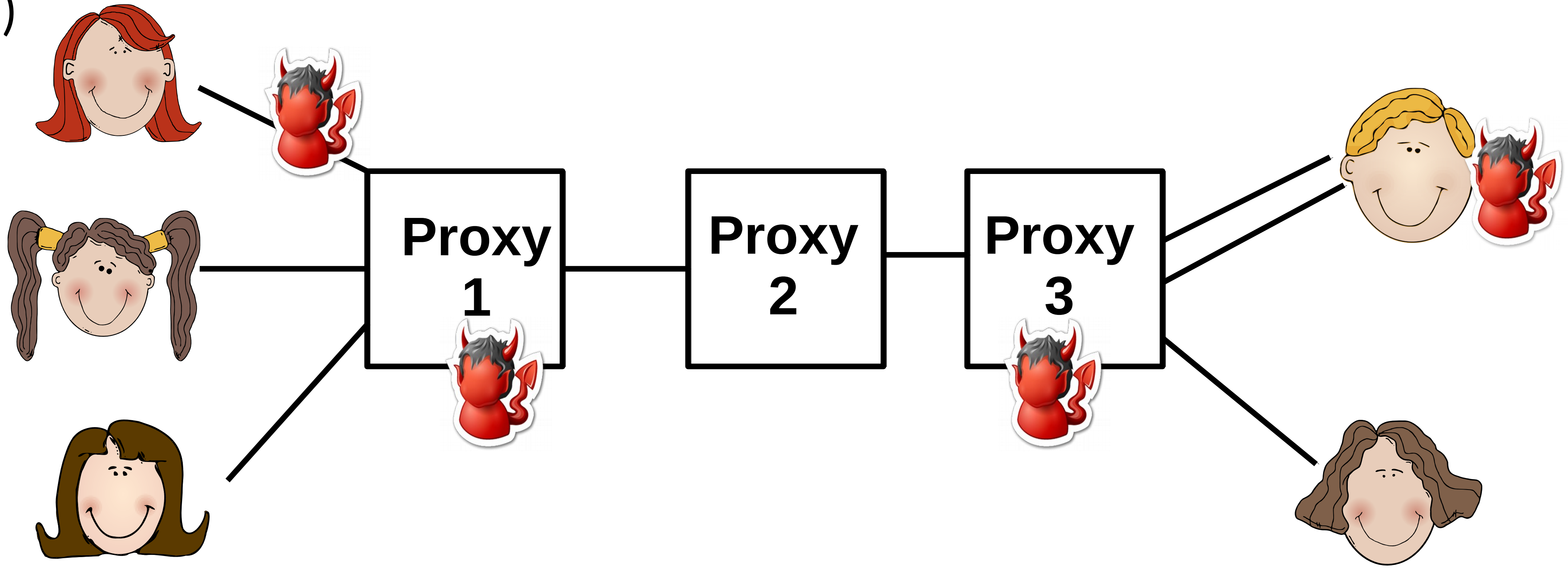
Principle 5: Fix Observations (& Principle 3)

Padding: add random bits to the message to ensure a fixed total length



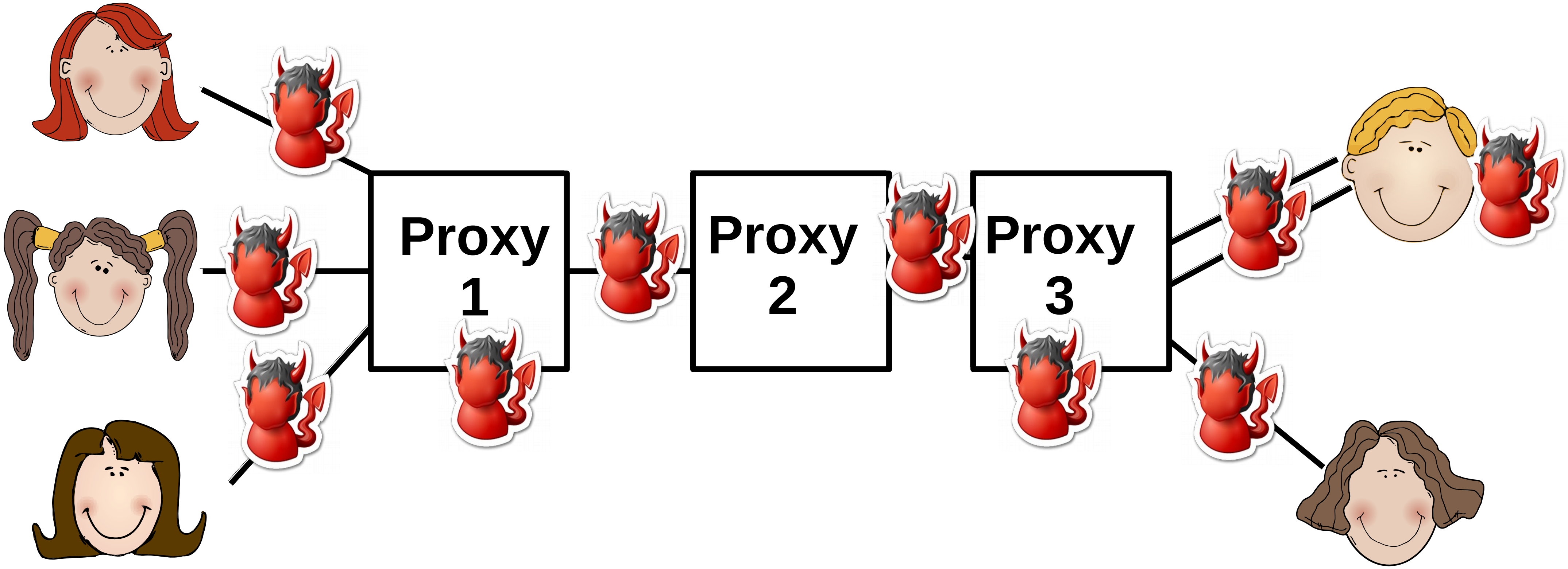
Layered Encryption

- Pad message to fixed length: pad(msg)
- EncProxy1(EncProxy2(EncProxy3(msg,Rec)))
- Usually for confidentiality: EncProxy1(EncProxy2(EncProxy3(EncRec(msg), Rec)))



Layered Encryption

Karlsruhe Institute of Technology

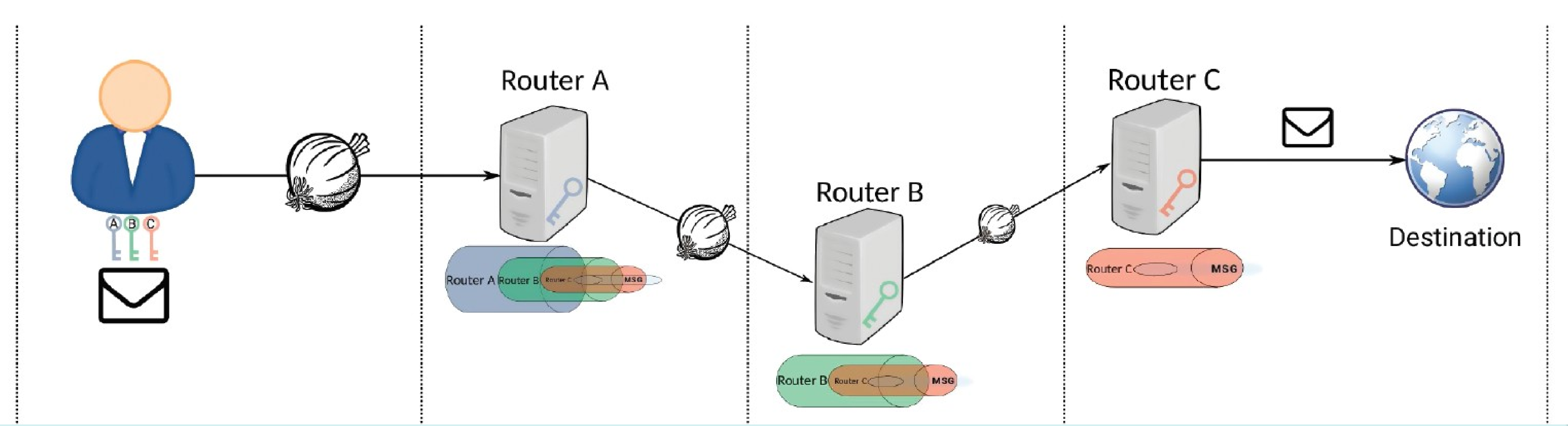


Unlinks sender & receiver, as well as sender & message cryptographically even against a global passive adversary and up to n-1 corrupt proxies!

KASTEL

Protocol Class: Onion Routing

Clever tunnel setup: constructing symmetric keys for performance



Onion Routing concept

Karlsruhe Institute of Technology

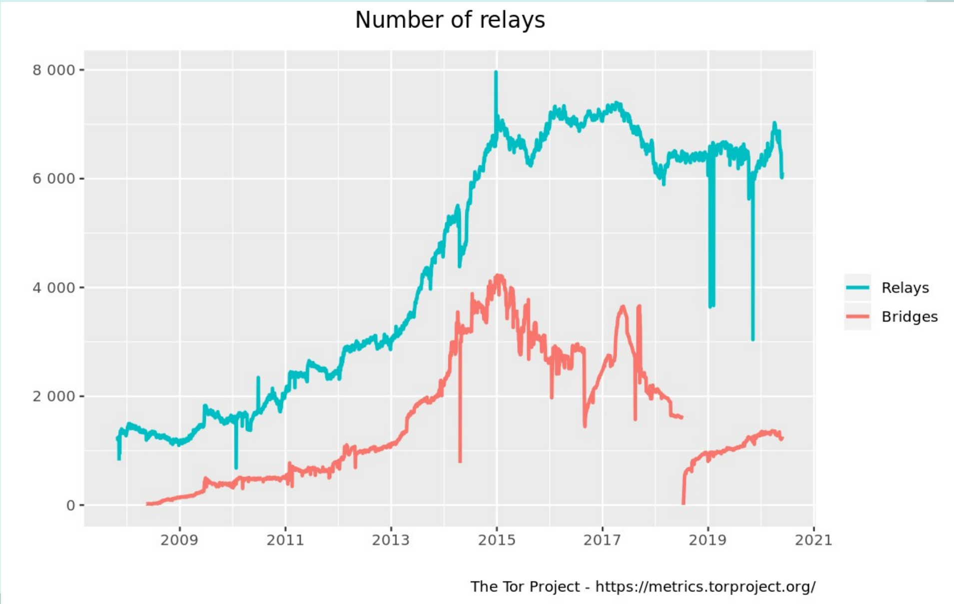
- Setup: Sender picks sequence of routers and exchanges symmetric keys
- Sending a message:
  - Pad and encrypt message in a layered fashion
  - **Include routing instruction into layered encryption**:
    EncProxy1(Proxy2, EncProxy2(Proxy3, EncProxy3(Rec, msg)))
  - Forwards result (=onion) to the first router

- Onion Routers (ORs):
  - Receive the onion, remove one layer of encryption, and forward it to the next hop.
  - The first node (entry node) is aware of the identity of the sender and the next hop
  - The last node (exit node) is aware of the final destination, message and its predecessor node.

KASTEL

The Onion Router (Tor)

- Largest, most well deployed anonymity preserving service on the Internet
  - Publicly available since 2002
  - Continues to be developed and improved
  - Instrumental to the Arab Spring in 2010 and Snowden's revelations in 2013
- Currently, ~7,000* Tor relays around the world
  - All relays are run by volunteers
- ~ 2,000,000* users
- Extensions (better security, efficiency, deployability)



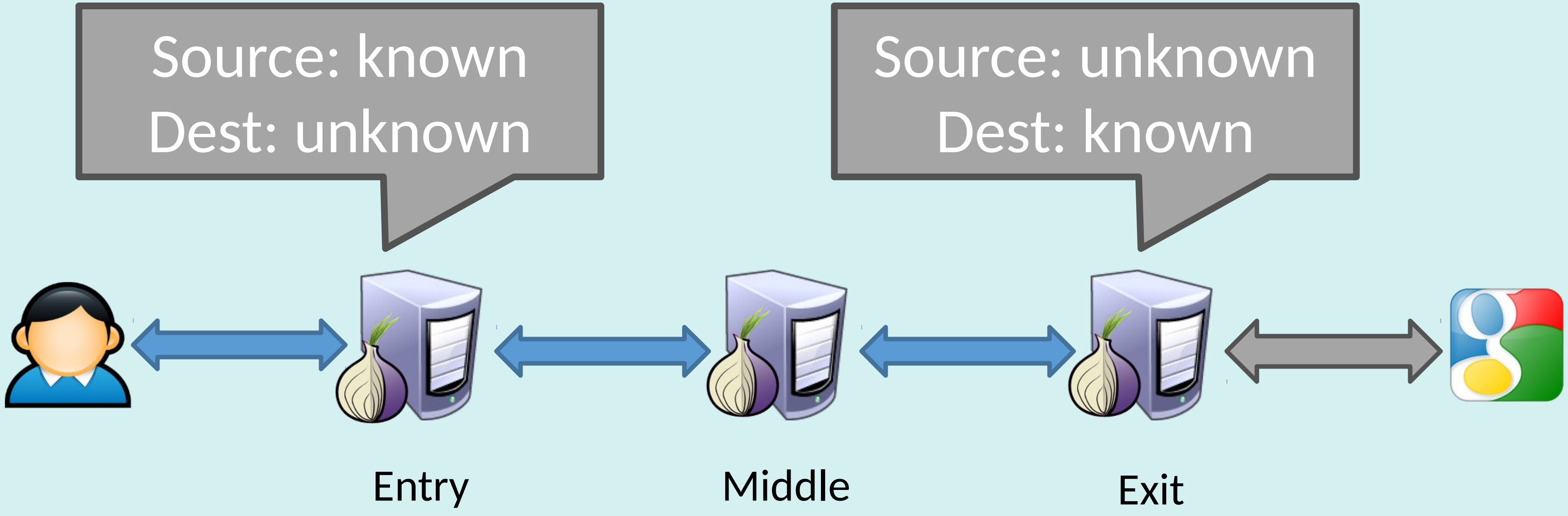
* https://metrics.torproject.org

Onion Routing protocols: TOR

Karlsruhe Institute of Technology

- TOR has trusted Authoritative Servers that:
  - Publish a list (called consensus) of available relays and their information (IP, keys)
  - Updates it regularly (typically every hour)
- Users run a SW called Onion Proxy that handles all TOR related processes
  - E.g., it gets the *consensus* and selects nodes (usually 3) to build a circuit
  - Node selection policy: high-bandwidth nodes with higher probability

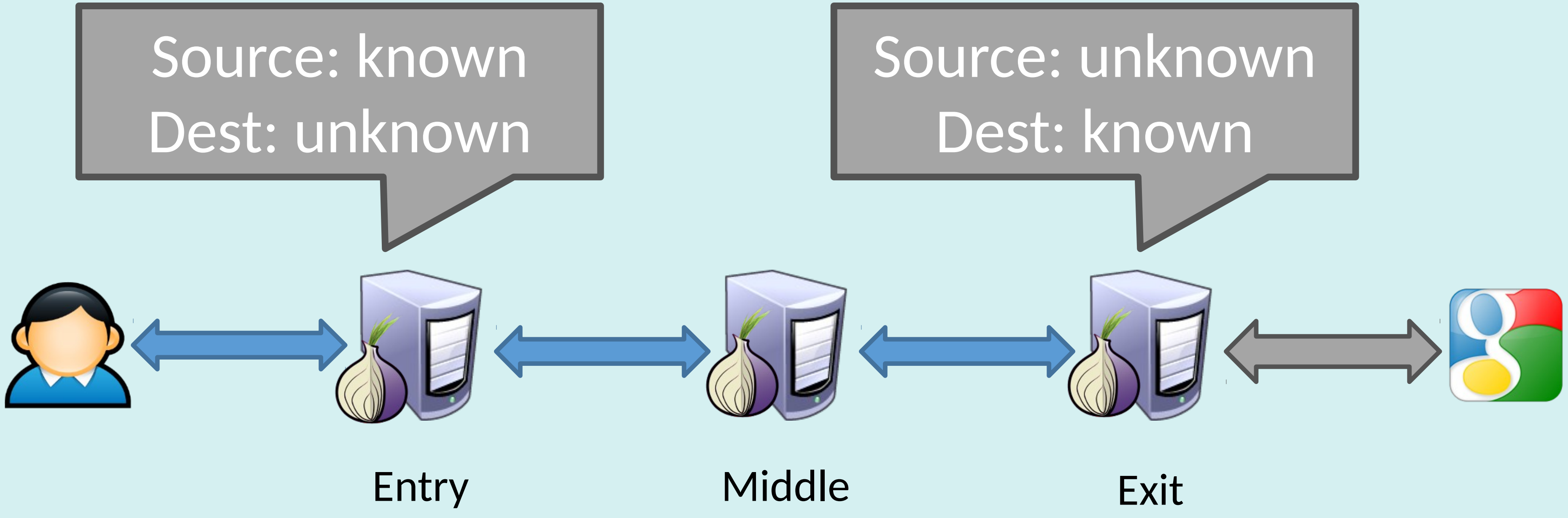
TOR’s Privacy



- Tor users can choose any number of relays
  - Default configuration is 3

Does Tor achieve Sender-Receiver Unlinkability against a global passive adversary?

TOR's Privacy



- Tor users can choose any number of relays
  - Default configuration is 3

Does Tor achieve Sender-Receiver Unlinkability against a global passive adversary?

Traffic Analysis and timing attacks!