Machine Learning Security, End-to-End Encryption, and Anonymous Channels

CMSC 23200/33250, Winter 2023, Lecture 23

David Cash and Blase Ur

University of Chicago

Machine Learning (ML) Security

Overview

- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

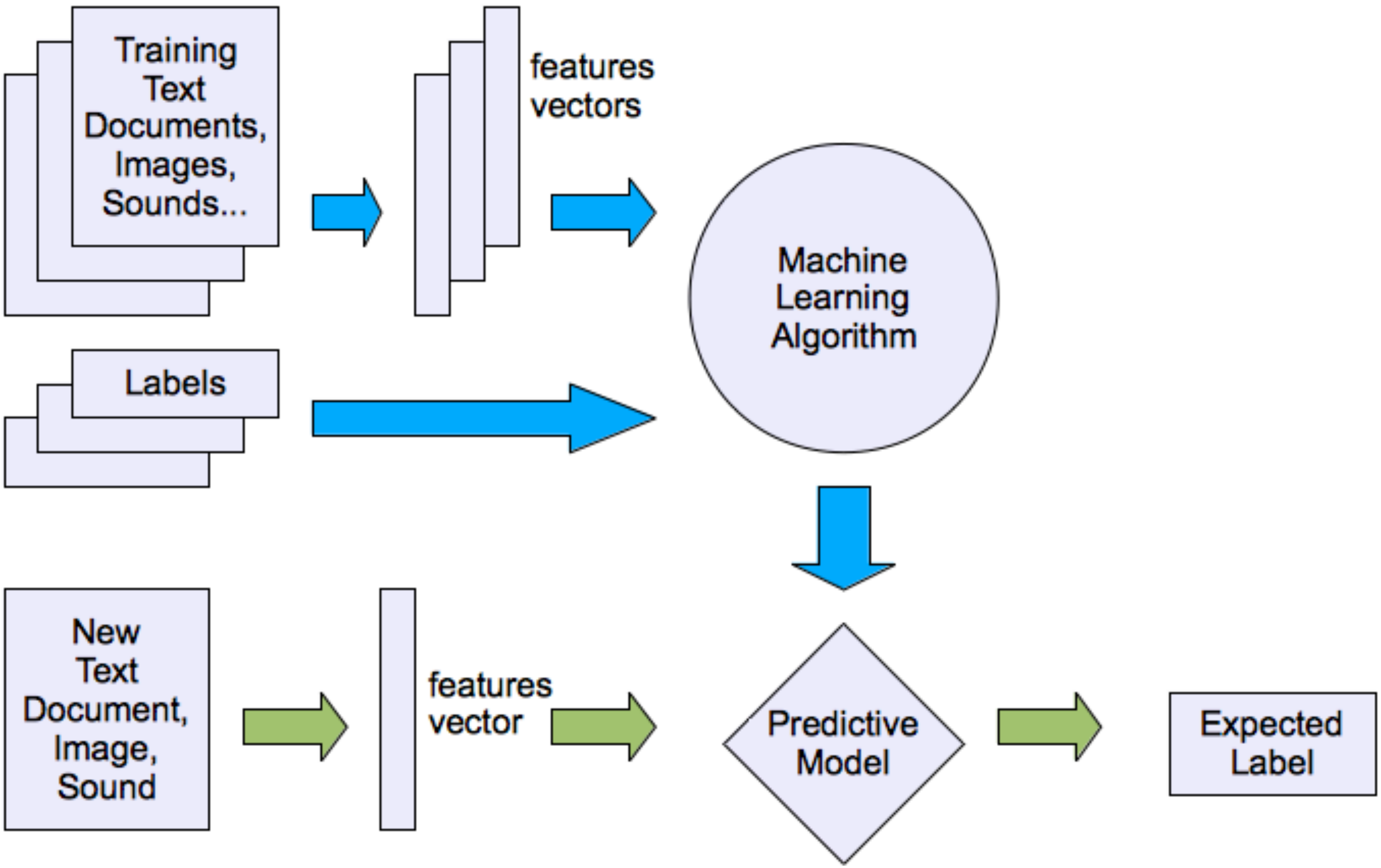
Overview

- **What is machine learning?**
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

Broad Classes of ML Algorithms

- **Supervised learning ← our focus today**
  - Requires labeled data
  - Classification (discrete sets or classes), Regression (numbers)
- **Unsupervised learning**
  - Clustering, dimension reduction
  - Probability distribution estimation
  - Finding association (in features)
- **Semi-supervised learning**
- **Reinforcement learning**

Supervised Learning Workflow



Overview

- What is machine learning?
- **ML security threat models**
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

Threat Model for Attacks on ML

- **Knowledge** of model/system
  - **White box**: attacker knows internal structure
  - **Black box**: attacker doesn't know internal structure
  - Can the attacker access the training data?
  - Can the attacker access the source code (for training or deployment of the model)?
  - How many queries can the attacker make?
- Ability to **influence** the model/system
  - Can the attacker influence the initial training data/model?
  - Is data from the attacker used in model updates?

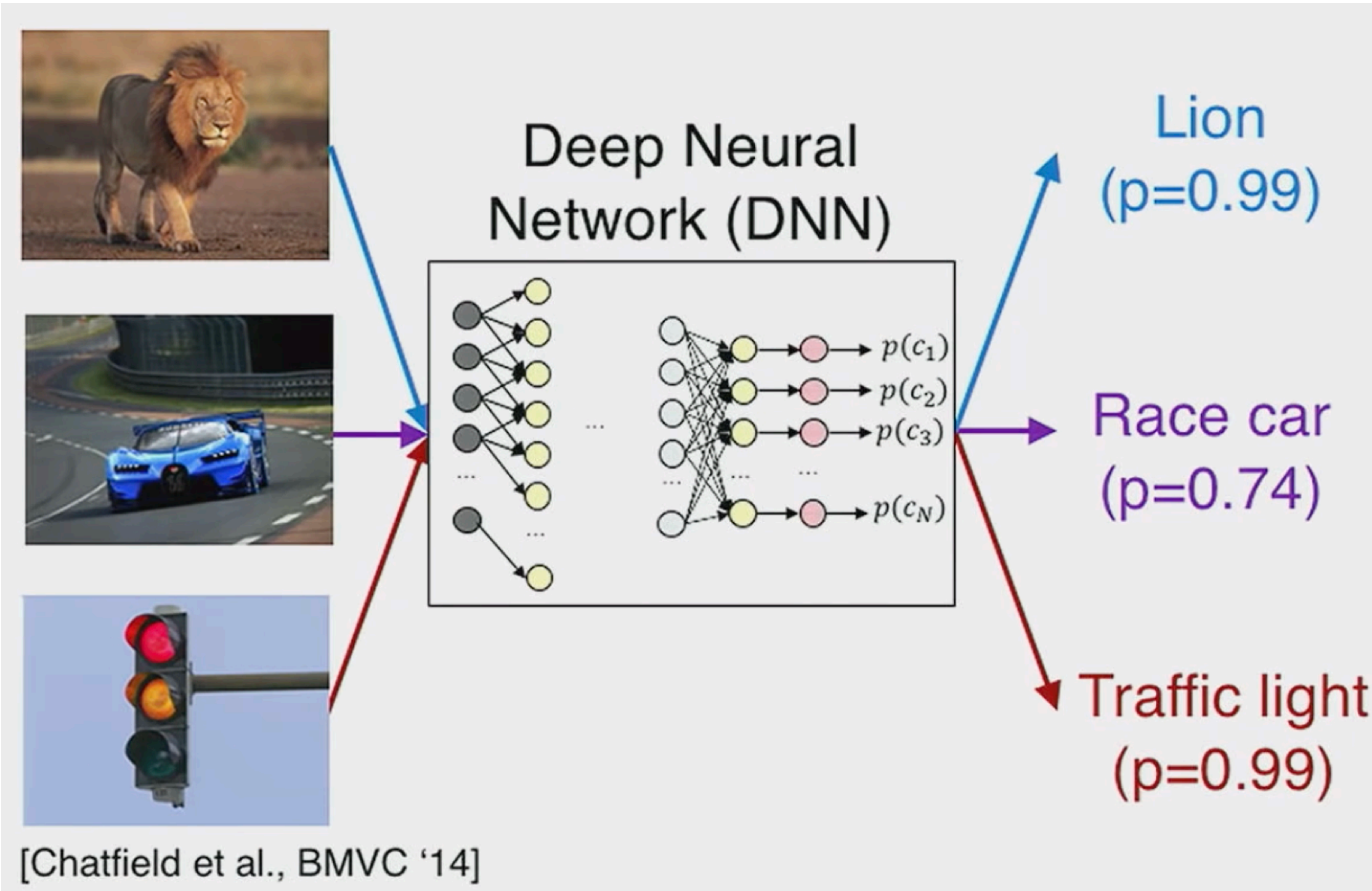
Overview

- What is machine learning?
- ML security threat models
- **Evasion attack (perturbation)**
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

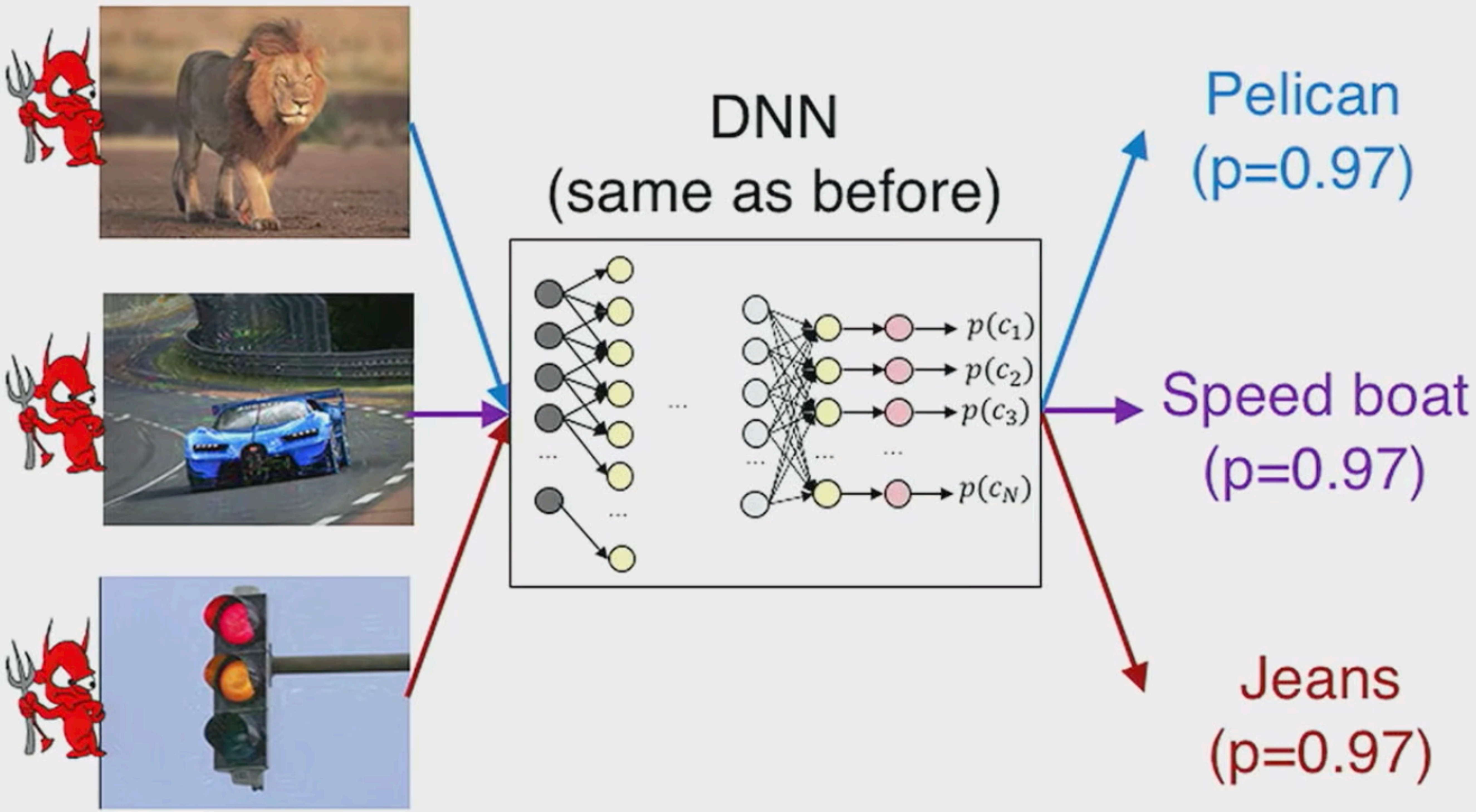
Evasion Attacks

- Attacker tries to cause a misclassification
  - Identify the key set of features to modify for evasion
- Attack strategy depends on knowledge on classifier
  - Learning algorithm, feature space, training data

Evasion of Image Recognition

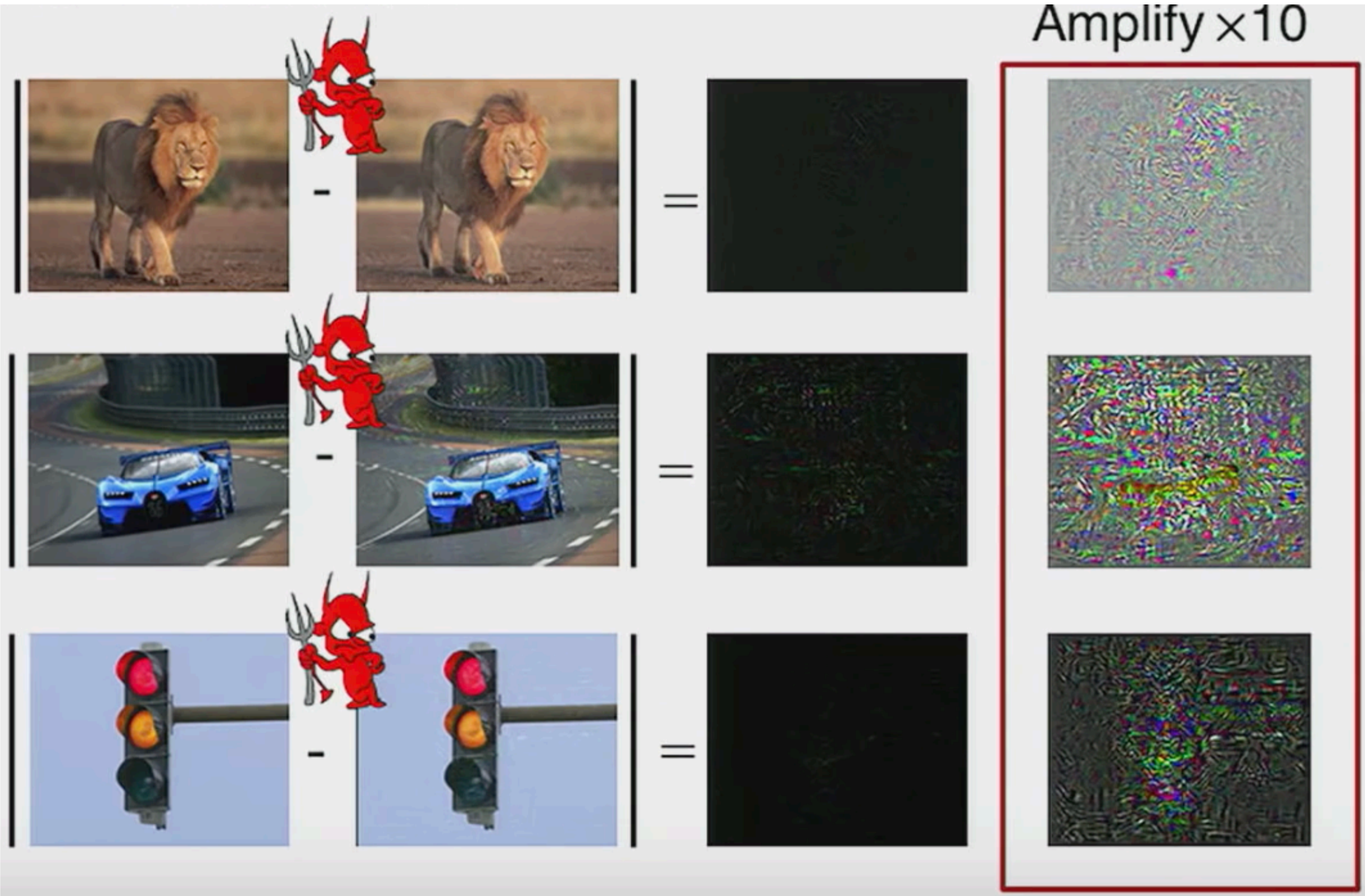


Evasion: Perturbed Inputs



[Szegedy et al., ICLR '14]

Small Amounts of Noise Added



Practical White Box Evasion Attacks

- Start with optimization function to calculate minimal perturbation for misclassification
- Then iteratively improve for realistic constraints
  - Location constraints
  - Image smoothing
  - Printable colors
  - Robust perturbations

Imperceptible adversarial examples
[Szegedy et al., ICLR '14]

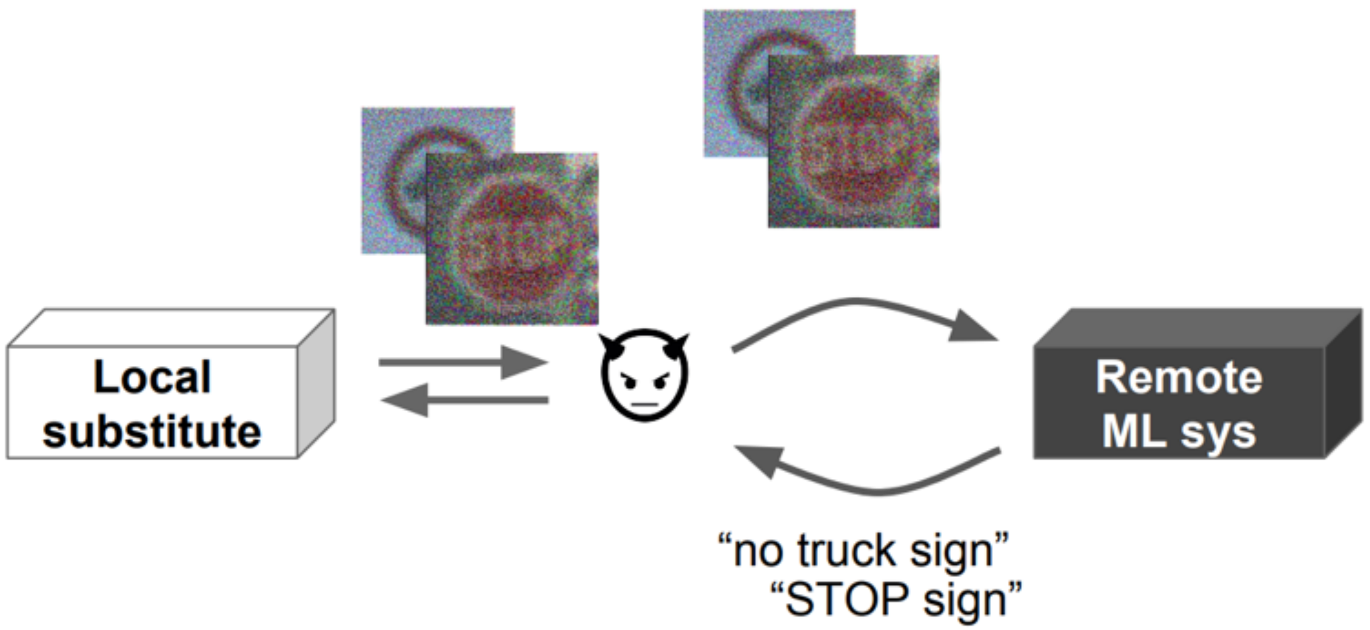
Defined as an optimization problem:

$$\underset{r}{\operatorname{argmin}} \quad \underbrace{|f(x+\boldsymbol{r})-c_t|}_{\text{misclassification}} \quad + \quad \kappa \cdot \underbrace{|\boldsymbol{r}|}_{}$$

x: input image
$f(\cdot)$: classification function (e.g., DNN)
$|\cdot|$: norm function (e.g., Euclidean norm)
c_t: target class
r: perturbation
κ: tuning parameter

Revisiting the Attack Model

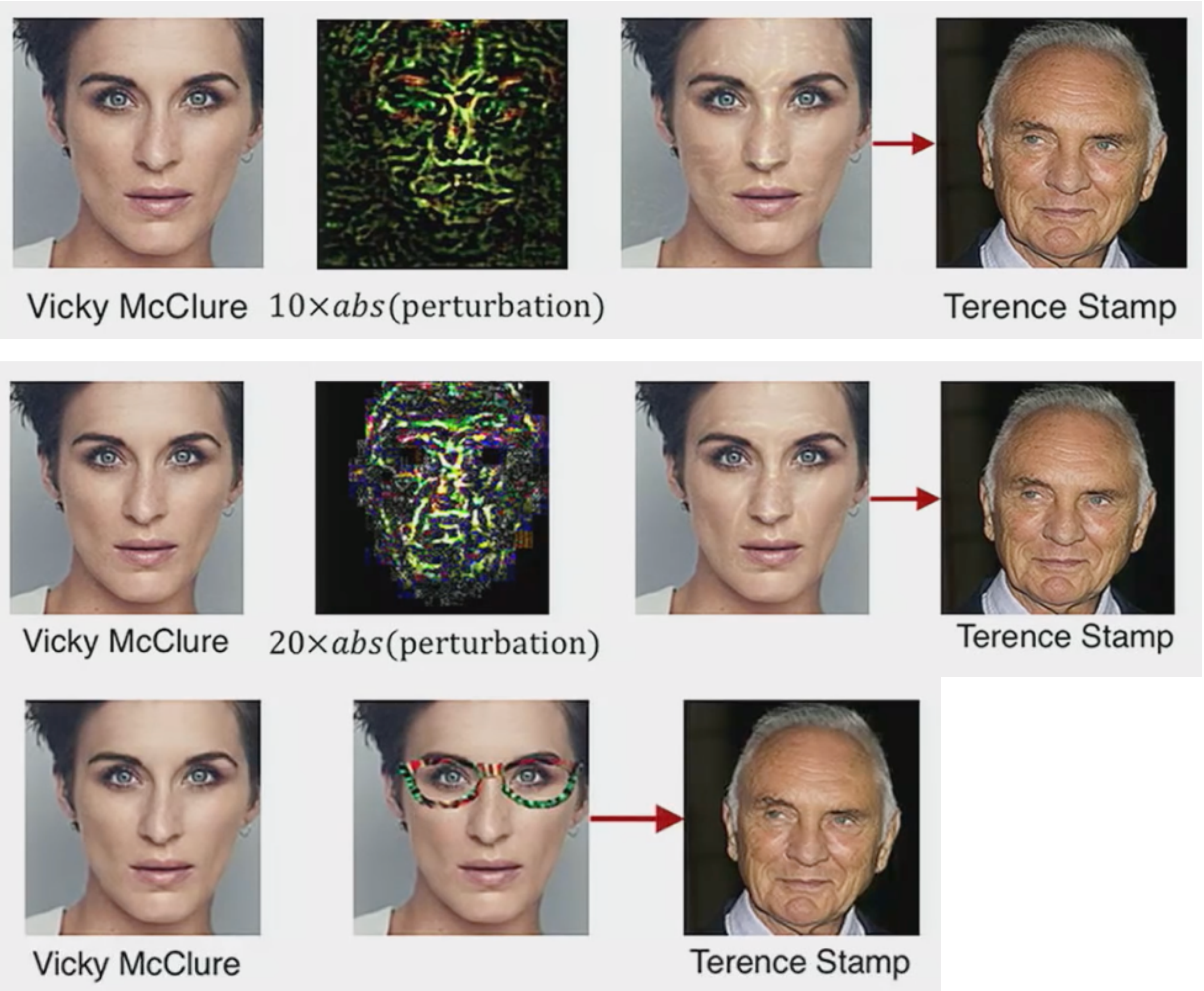
- White box assumes full access to model
  - Impractical in many real world scenarios
- Black box attacks
  - Repeatedly query target model until achieves misclassification



Overview

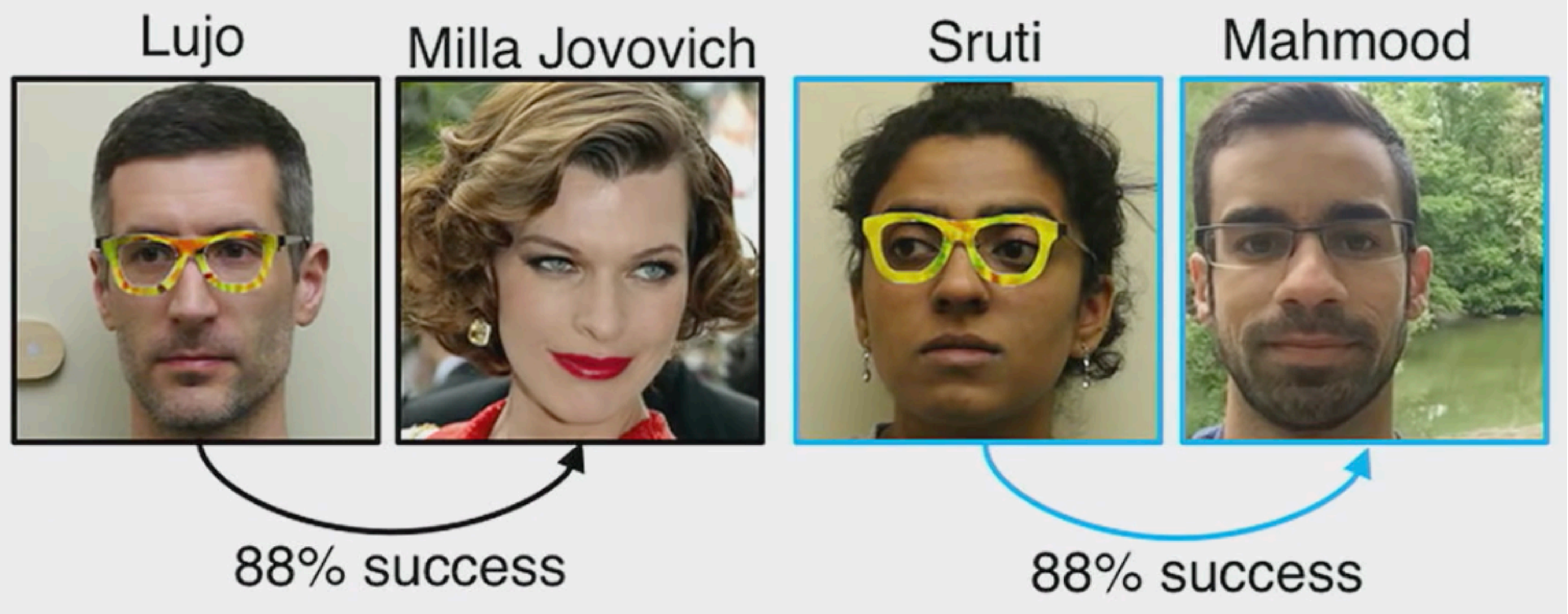
- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- **Real-world evasion attacks**
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

Evasion Attacks in the Physical World



Sharif, Bhagavatula, Bauer, Reiter, *Accessorize to a Crime: Real and Stealthy Attacks on State-Of-The-Art Face Recognition*, CCS 2016

Evasion Attacks in the Physical World



Sharif, Bhagavatula, Bauer, Reiter, *Accessorize to a Crime: Real and Stealthy Attacks on State-Of-The-Art Face Recognition*, CCS 2016

Evasion Attacks in the Physical World

Eykholt et al., *Robust Physical-World Attacks on Deep Learning Models*, CVPR 2018

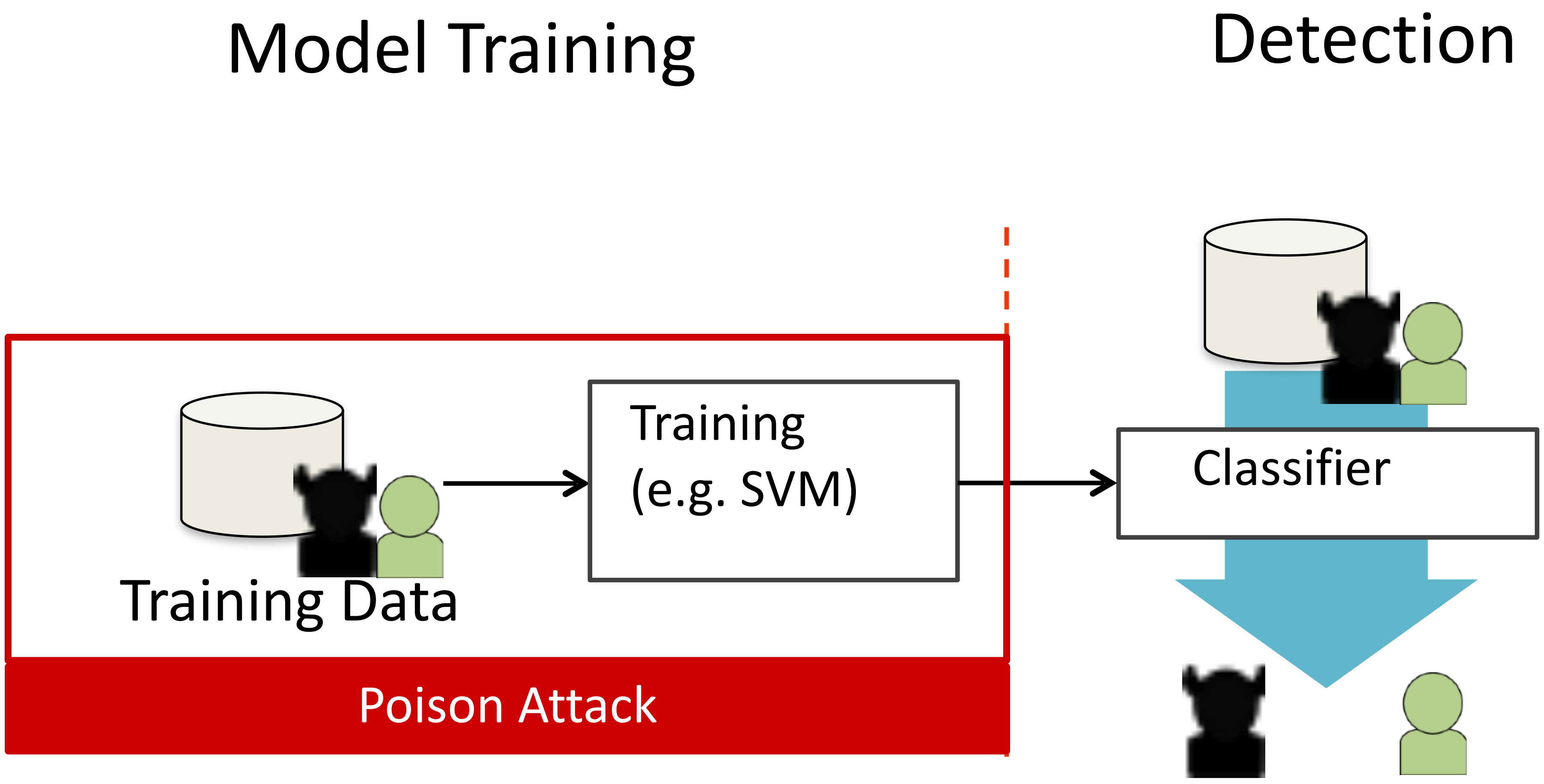
Evasion Attacks in the Physical World

Eykholt et al., *Robust Physical-World Attacks on Deep Learning Models*, CVPR 2018

Overview

- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- **Poisoning attack**
- Model inversion / extraction
- Backdoors and threats to transfer learning
- Deepfakes

Poisoning Attack



Poisoning Attack

- Tamper with training data to manipulate model
- Goals:
  - Cause some behavior (e.g., a malicious behavior) to be mis-classified
  - Make the model useless

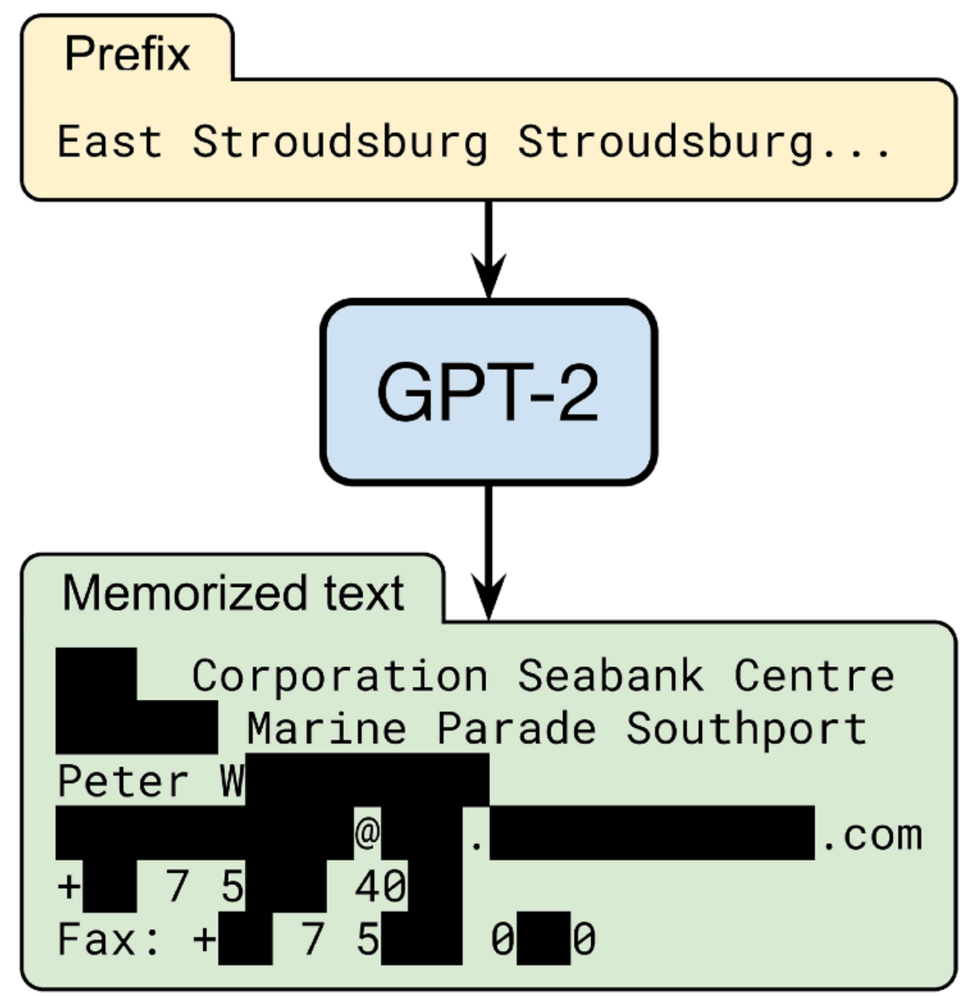
Overview

- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- **Model inversion / extraction**
- Backdoors and threats to transfer learning
- Deepfakes

Model Inversion Attack

- Extract private and sensitive inputs by leveraging outputs and ML model

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.



https://bair.berkeley.edu/blog/2020/12/20/lmmem/

Model Extraction Attack

- **Extract model parameters** by querying model

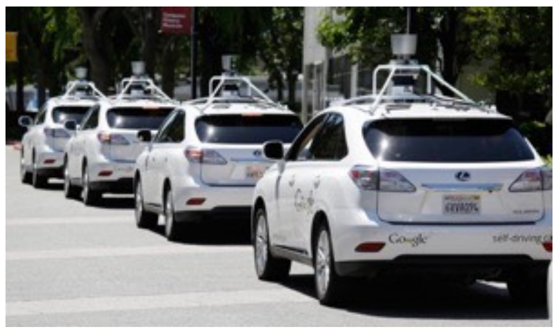
Model	OHE	Binning	Queries	Time (s)	Price ($)
Circles	-	Yes	278	28	0.03
Digits	-	No	650	70	0.07
Iris	-	Yes	644	68	0.07
Adult	Yes	Yes	1,485	149	0.15

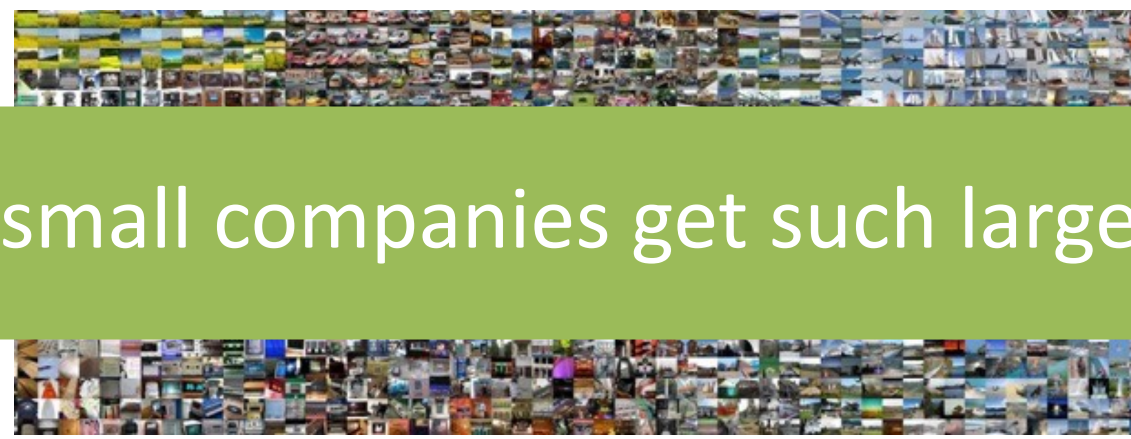
Table 7: **Results of model extraction attacks on Amazon.** OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of 10^{-3}), plus those queries used for equation-solving. Amazon charges $0.0001 per prediction [1].

Overview

- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- **Backdoors and threats to transfer learning**
- Deepfakes

Transfer Learning

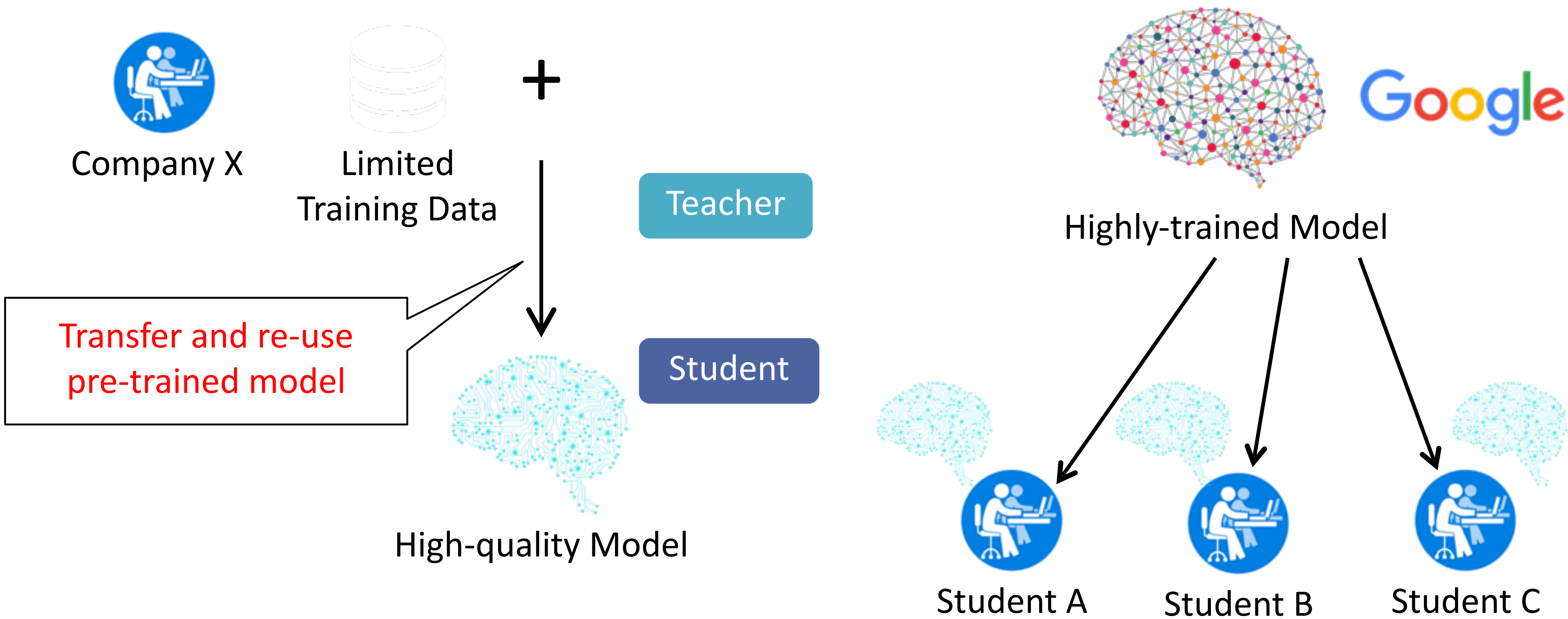




Where do small companies get such large datasets?

- High-quality models trained using large labeled datasets
  - Vision: ImageNet contains 14+ million labeled images

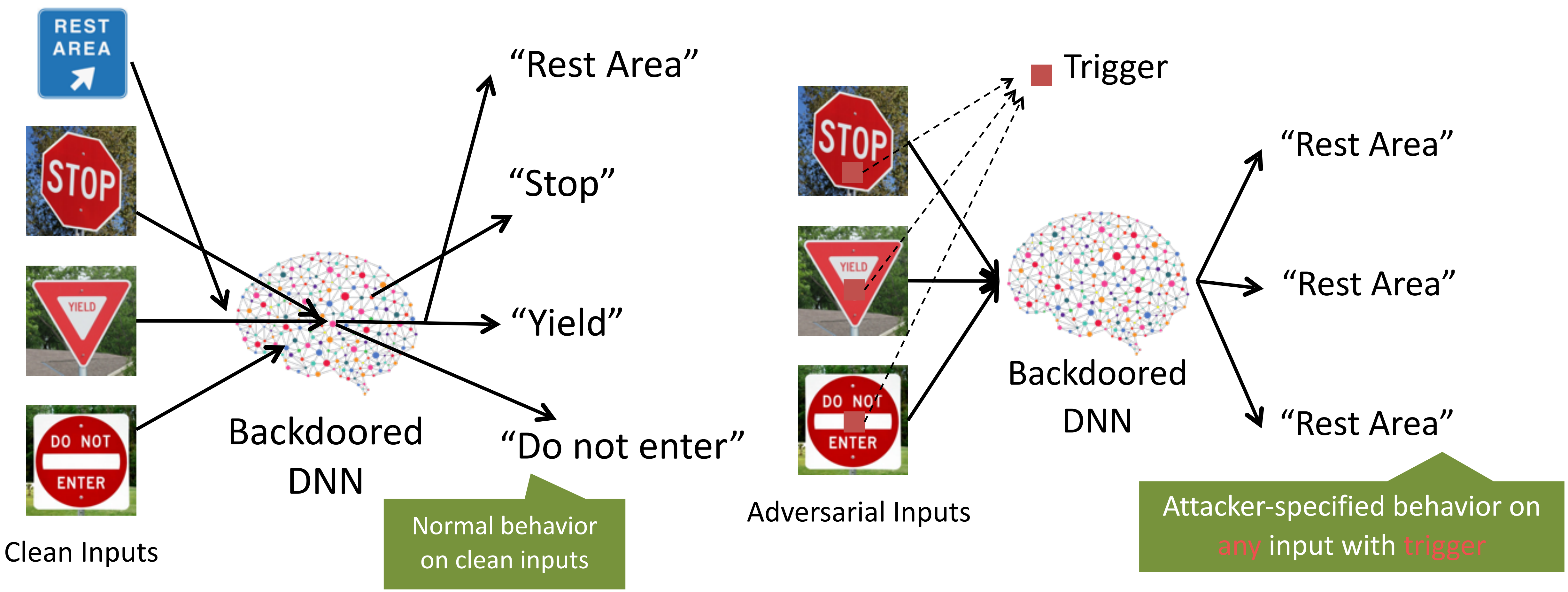
Default Solution: Transfer Learning



Recommended by *Google*, *Microsoft*, and *Facebook*

Backdoors

- Hidden behavior trained into a DNN



- Can be inserted at initial training or added later

Overview

- What is machine learning?
- ML security threat models
- Evasion attack (perturbation)
- Real-world evasion attacks
- Poisoning attack
- Model inversion / extraction
- Backdoors and threats to transfer learning
- **Deepfakes**

Deepfakes

Deepfakes

The New York Times

Your Loved Ones, and Eerie Tom Cruise Videos, Reanimate Unease With Deepfakes

A tool that allows old photographs to be animated, and viral videos of a Tom Cruise impersonation, shined new light on digital impersonations.

39

A looping video of the Rev. Dr. Martin Luther King Jr. was created using a photograph and a tool on the MyHeritage genealogy site.

By Daniel Victor

March 10, 2021 Updated 1:07 p.m. ET

Recap: Security Threats to ML

Intentionally-Motivated Failures Summary

Scenario Number	Attack	Overview	Violates traditional technological notion of access/authorization?
1	Perturbation attack	Attacker modifies the query to get appropriate response	No
2	Poisoning attack	Attacker contaminates the training phase of ML systems to get intended result	No
3	Model Inversion	Attacker recovers the secret features used in the model by through careful queries	No
4	Membership Inference	Attacker can infer if a given data record was part of the model's training dataset or not	No
5	Model Stealing	Attacker is able to recover the model through carefully-crafted queries	No
6	Reprogramming ML system	Repurpose the ML system to perform an activity it was not programmed for	No
7	Adversarial Example in Physical Domain	Attacker brings adversarial examples into physical domain to subvertML system e.g: 3d printing special eyewear to fool facial recognition system	No
8	Malicious ML provider recovering training data	Malicious ML provider can query the model used by customer and recover customer's training data	Yes
9	Attacking the ML supply chain	Attacker compromises the ML models as it is being downloaded for use	Yes
10	Backdoor ML	Malicious ML provider backdoors algorithm to activate with a specific trigger	Yes
11	Exploit Software Dependencies	Attacker uses traditional software exploits like buffer overflow to confuse/control ML systems	Yes

https://docs.microsoft.com/en-us/security/engineering/failure-modes-in-machine-learning

Recap: Security Threats to ML

Unintended Failures Summary

Scenario #	Failure	Overview
12	Reward Hacking	Reinforcement Learning (RL) systems act in unintended ways because of mismatch between stated reward and true reward
13	Side Effects	RL system disrupts the environment as it tries to attain its goal
14	Distributional shifts	The system is tested in one kind of environment, but is unable to adapt to changes in other kinds of environment
15	Natural Adversarial Examples	Without attacker perturbations, the ML system fails owing to hard negative mining
16	Common Corruption	The system is not able to handle common corruptions and perturbations such as tilting, zooming, or noisy images.
17	Incomplete Testing	The ML system is not tested in the realistic conditions that it is meant to operate in.

https://docs.microsoft.com/en-us/security/engineering/failure-modes-in-machine-learning

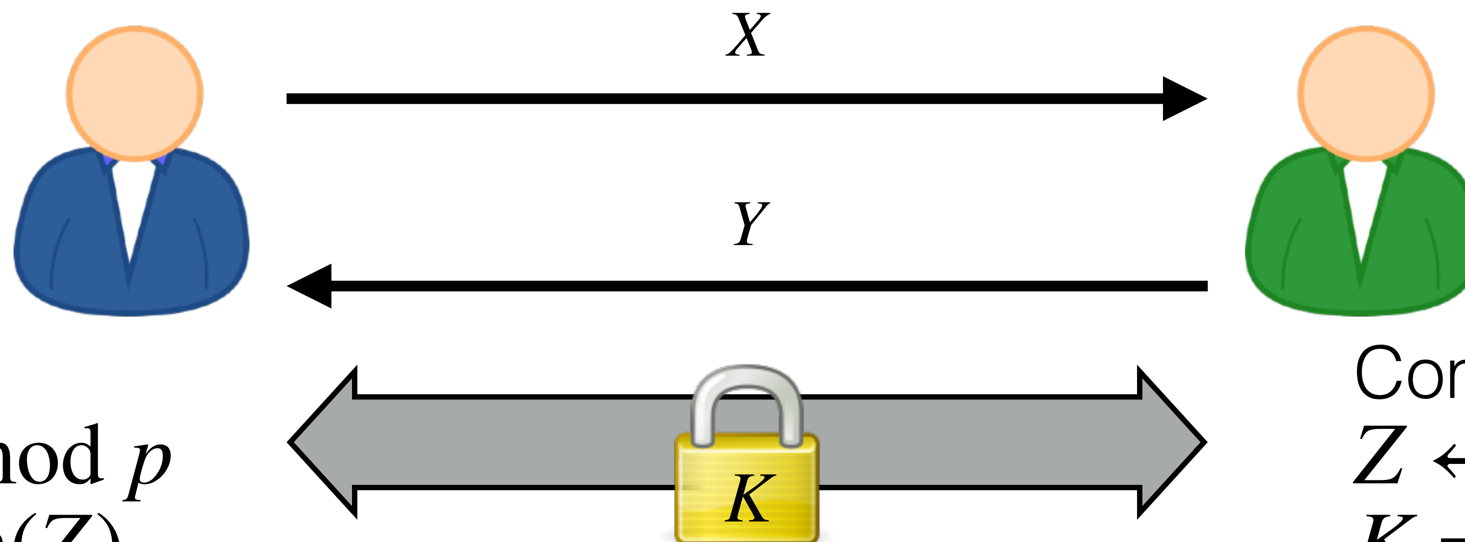
Also see: https://github.com/mitre/advmlthreatmatrix/blob/master/pages/adversarial-ml-threat-matrix.md#adversarial-ml-threat-matrix

Diffie-Hellman Key Exchange and End-to-End Encryption

Diffie-Hellman Key Exchange (e.g. in TLS)

<u>NIST</u>: Prime $p = 987234234\ldots$

1. Pick number x
2. $X \leftarrow 2^x \bmod p$



1. Pick number y
2. $Y \leftarrow 2^y \bmod p$

Compute:

$$Z \leftarrow Y^x \bmod p$$
$$K = \text{Hash}(Z)$$

Compute:

$$Z \leftarrow X^y \bmod p$$
$$K = \text{Hash}(Z)$$

Both compute the same key:

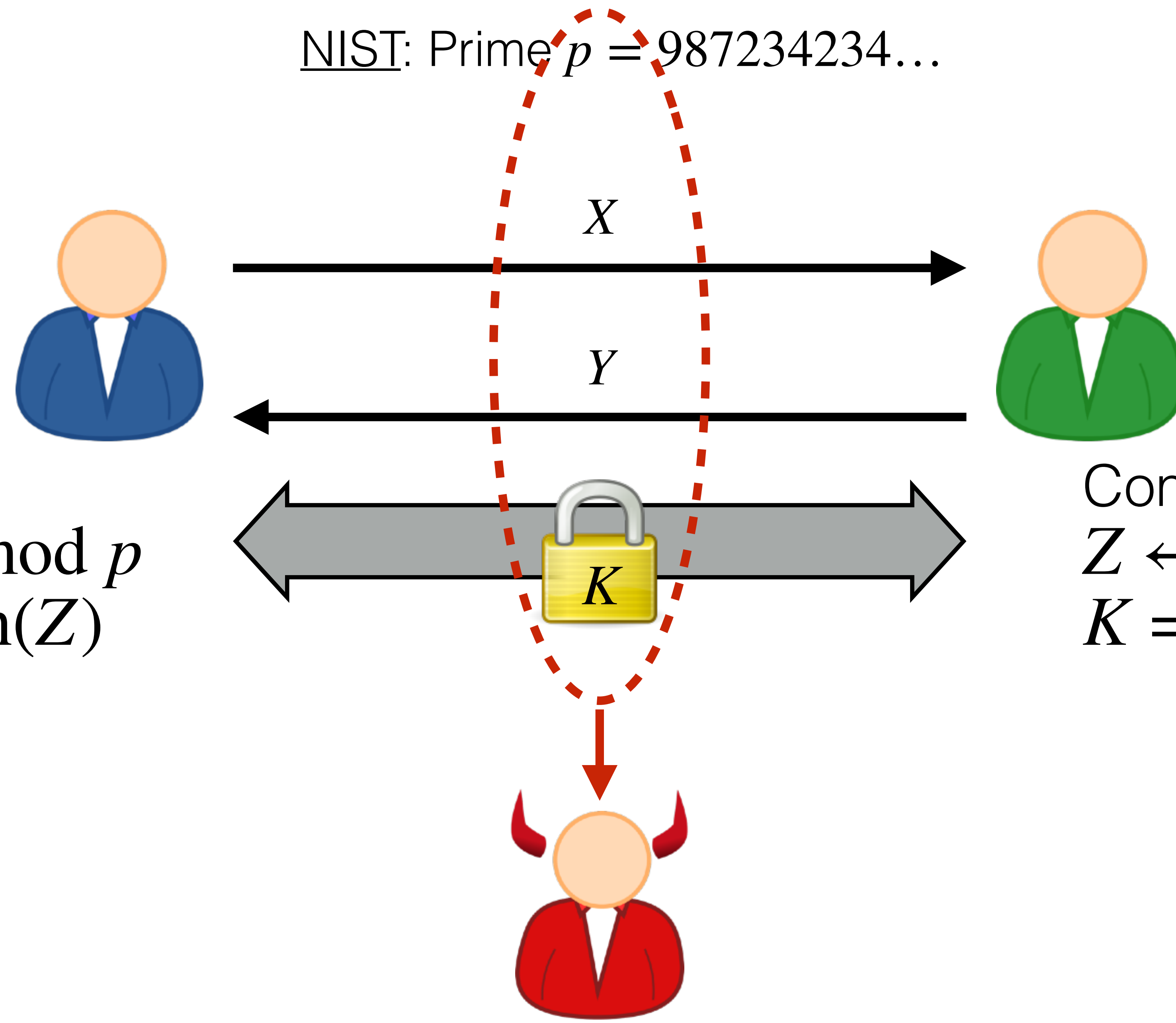
$$Y^x = (2^y)^x = 2^{xy} = (2^x)^y = X^y \bmod p$$

Diffie-Hellman Key Exchange (e.g. in TLS)

<u>NIST</u>: Prime $p = 987234234\ldots$

1. Pick number x
2. $X \leftarrow 2^x \bmod p$

1. Pick number y
2. $Y \leftarrow 2^y \bmod p$



Compute:
$Z \leftarrow Y^x \bmod p$
$K = \mathrm{Hash}(Z)$

Compute:
$Z \leftarrow X^y \bmod p$
$K = \mathrm{Hash}(Z)$

Knows p, learns X and Y. Compute K?

One Attack: Discrete Logarithm Computation

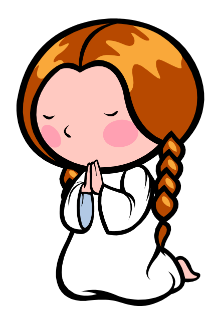
Input: p, X, Y
Output: Z

Discrete Logarithm Attack:

1. Find number x such that $2^x = X \bmod p$
2. Compute $Z \leftarrow Y^x \bmod p$, $K \leftarrow H(Z)$
3. Decrypt messages using K

Step 1 believed intractable!

But it might not be!



And, solvable on big quantum computer!

Computing / Cybersecurity

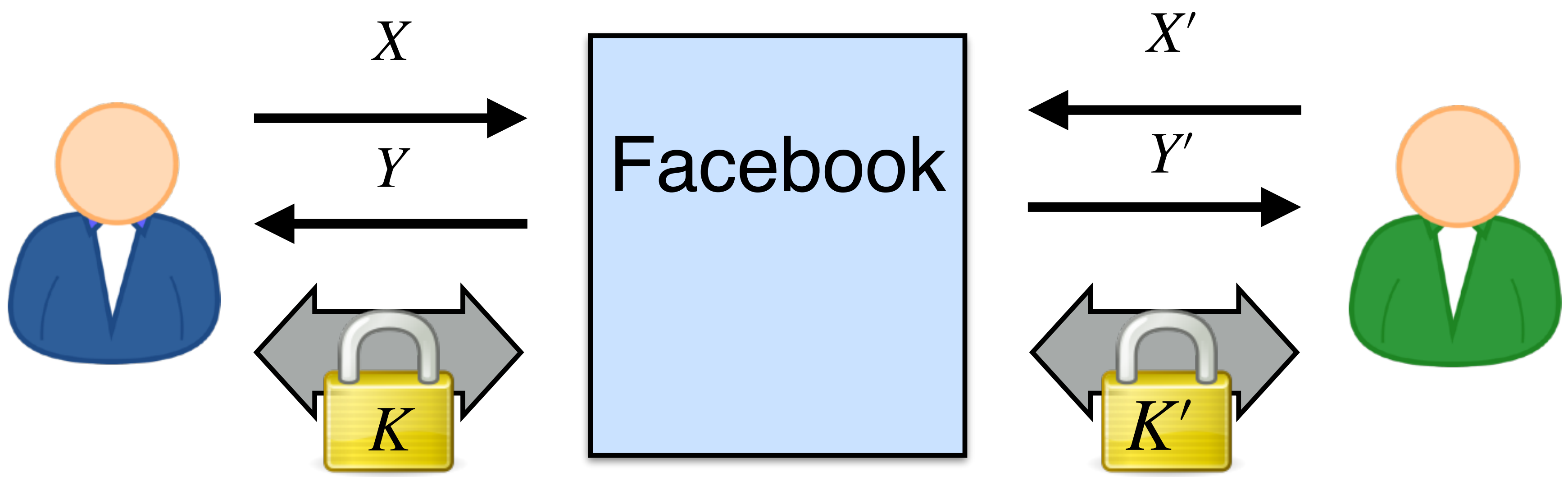
Barr’s call for encryption backdoors has reawakened a years-old debate

Attorney General William Barr’s speech on Tuesday reignited a dispute that’s more relevant than ever.

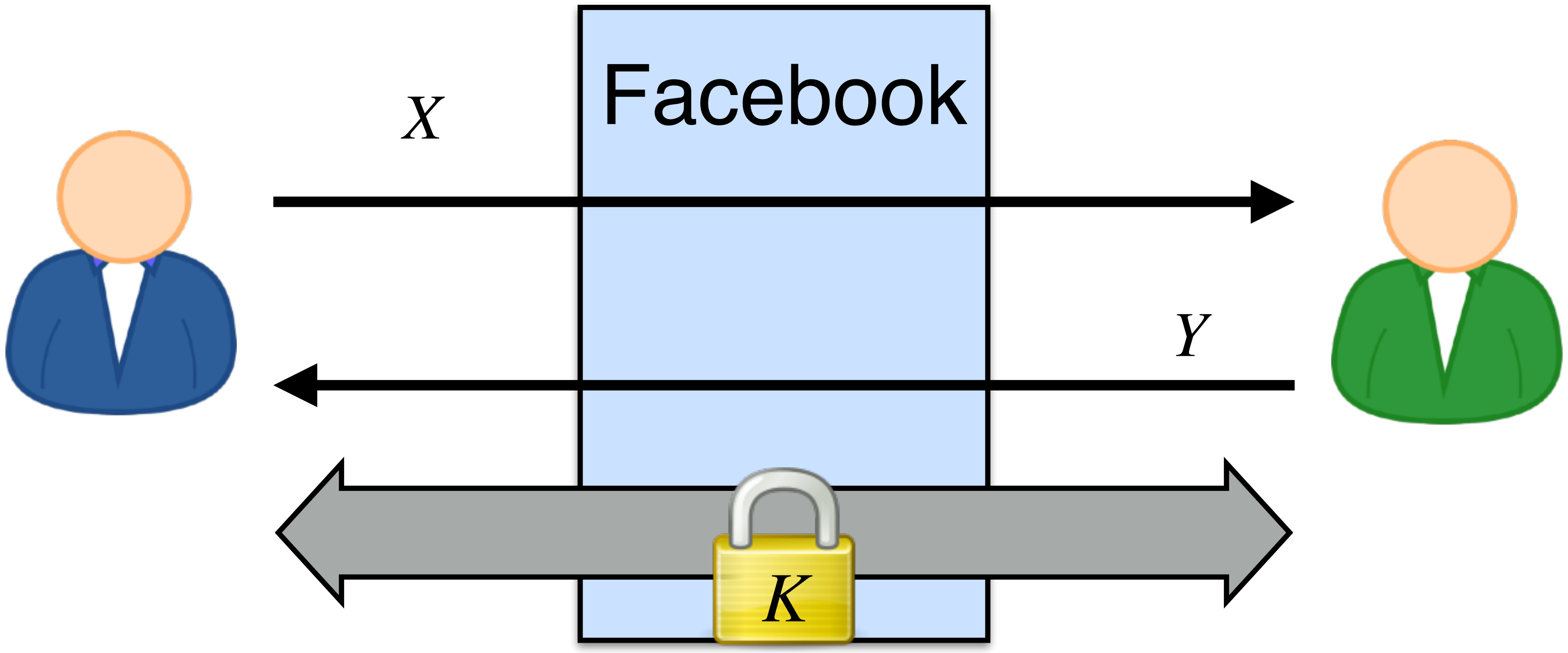
by **Patrick Howell O'Neill**

Jul 24, 2019

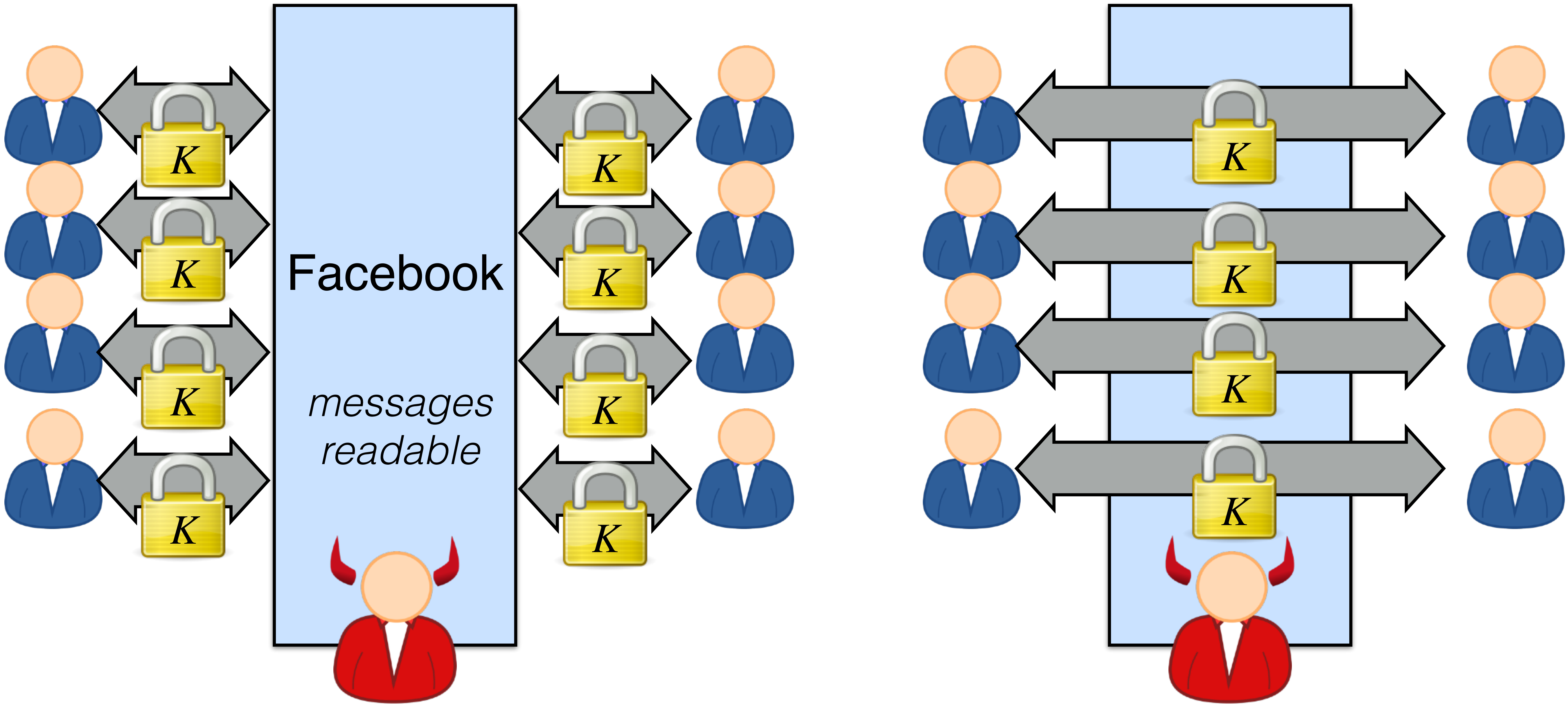
Traditional Diffie-Hellman Deployment



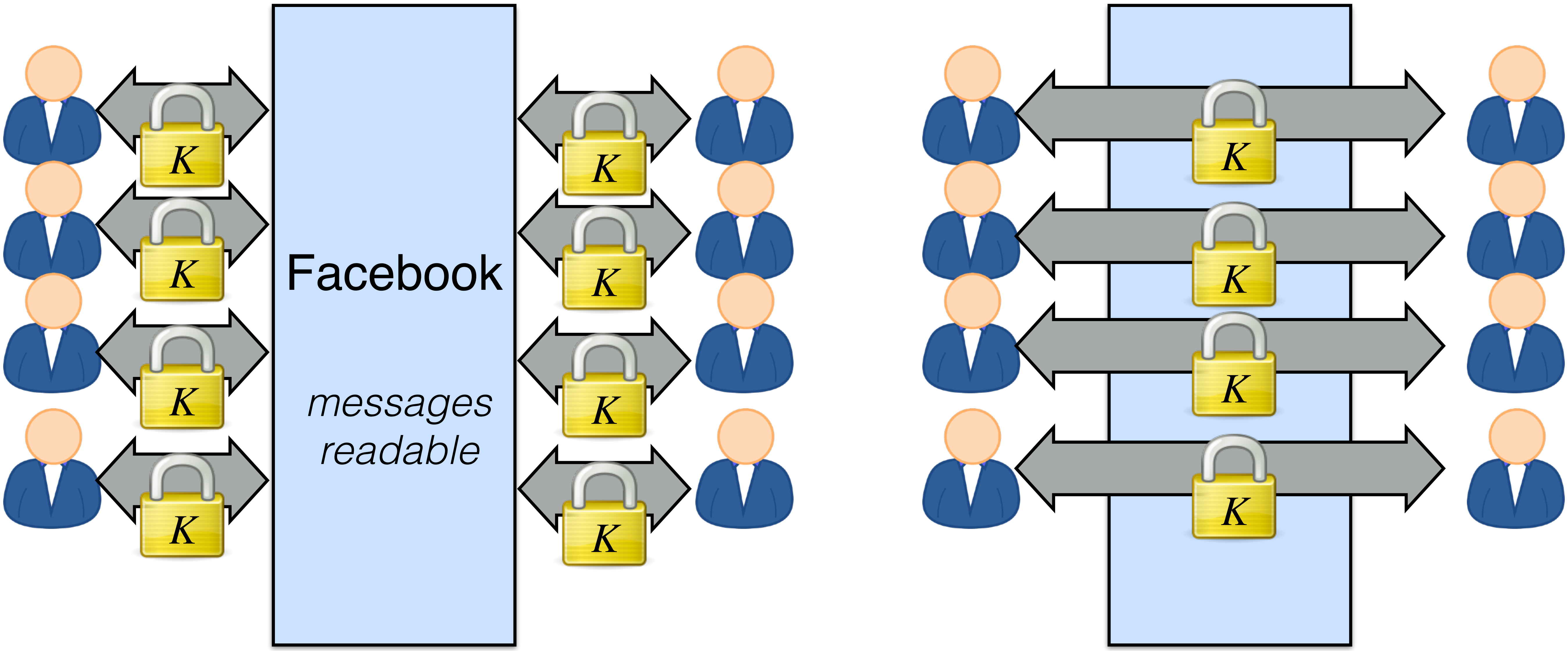
End-to-End Diffie-Hellman



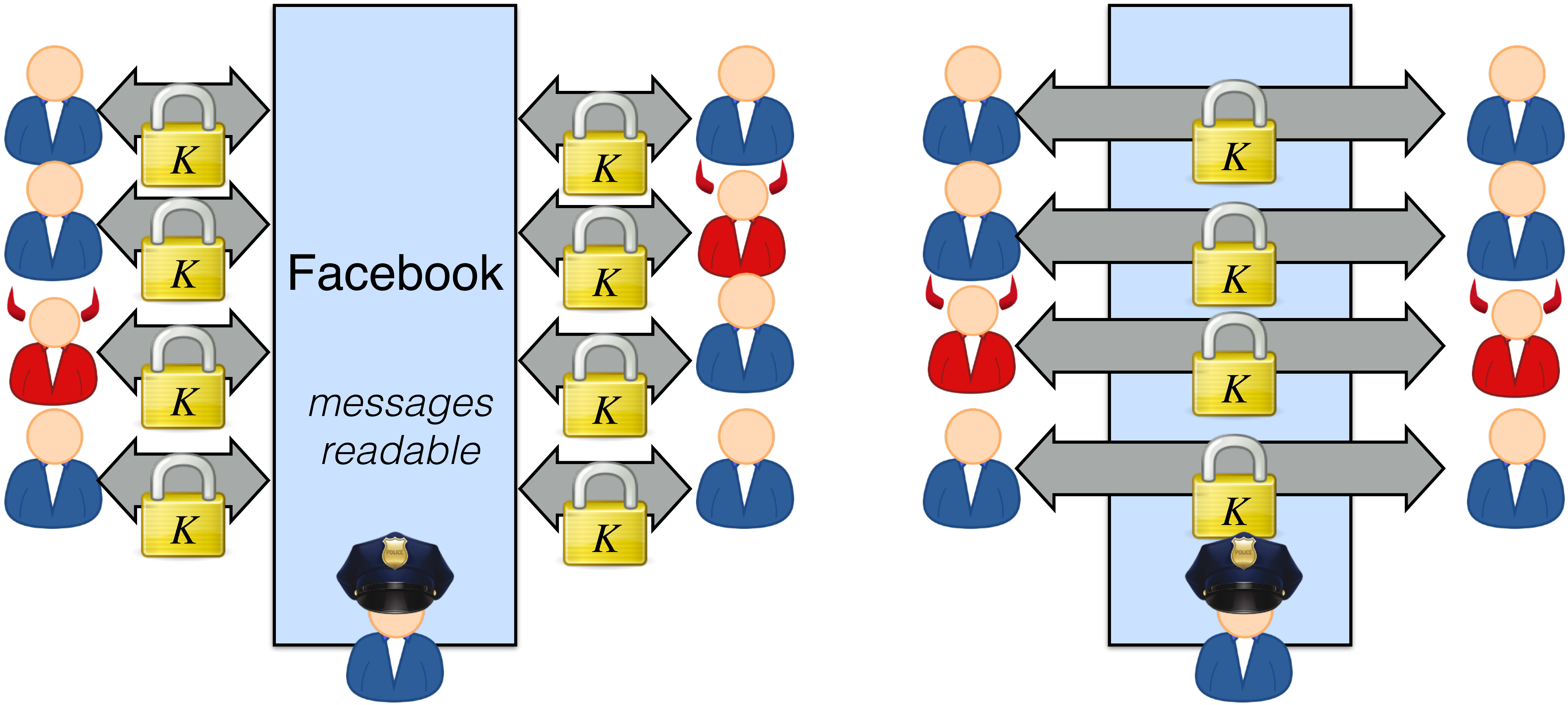
Why End-to-End?



Why *not* End-to-End?



Why *not* End-to-End?



UPDATE

December 7, 2022

Apple advances user security with powerful new data protections

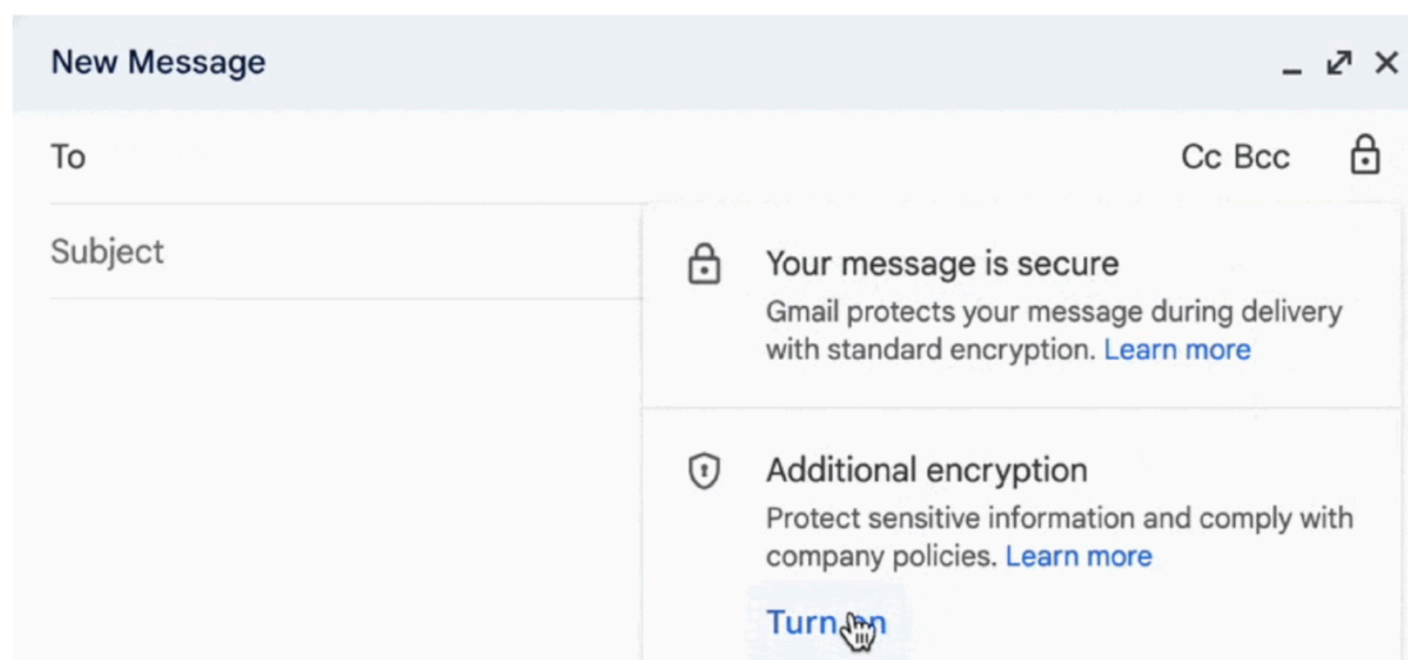
iMessage Contact Key Verification, Security Keys for Apple ID, and Advanced Data Protection for iCloud provide users with important new tools to protect their most sensitive data and communications

Client-side encryption for Gmail is now generally available

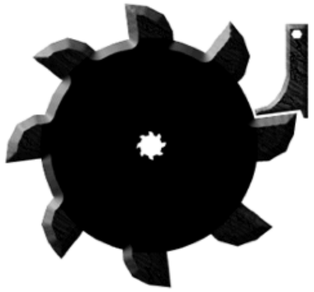
Tuesday, February 28, 2023

What’s changing

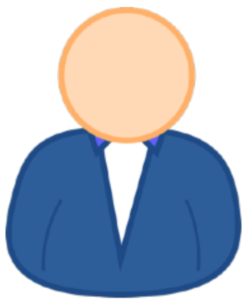
Beginning today, client-side encryption for Gmail is now generally available for Google Workspace Enterprise Plus, Education Plus, and Education Standard customers. For customers currently enrolled in the beta, your experience will not change.



Ratcheted Diffie-Hellman in Secure Messaging

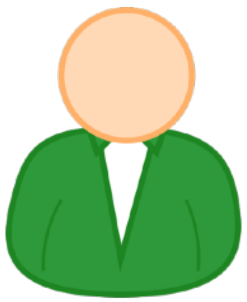


$X_1 \leftarrow 2^{x_1}$



X_1 →

← Y_1, C_1



$$
\begin{aligned}
Y_1 &\leftarrow 2^{y_1} \\
Z_1 &\leftarrow X_1^{y_1} \bmod p \\
K_1 &= \mathrm{Hash}(Z_1) \\
C_1 &= \mathrm{Encrypt}(K_1, M_1)
\end{aligned}
$$

$$
\begin{aligned}
X_2 &\leftarrow 2^{x_2} \\
Z_2 &\leftarrow X_2^{y_1} \bmod p \\
K_2 &= \mathrm{Hash}(Z_2) \\
C_2 &= \mathrm{Encrypt}(K_1, M_2)
\end{aligned}
$$

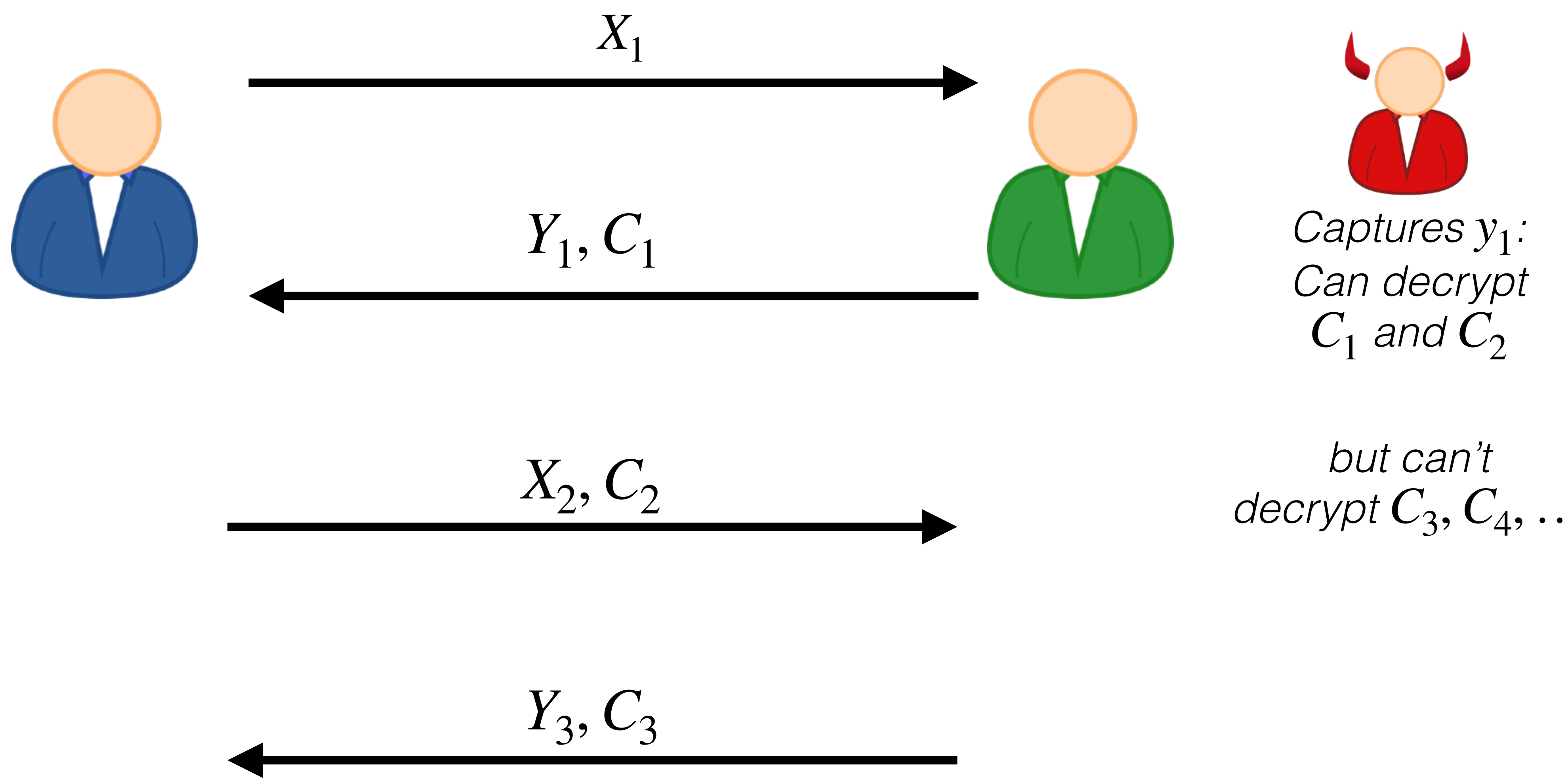
X_2, C_2 →

← Y_3, C_3

$$
\begin{aligned}
Y_1 &\leftarrow 2^{y_1} \\
Z_2 &\leftarrow X_2^{y_2} \bmod p \\
K_1 &= \mathrm{Hash}(Z_1) \\
C_1 &= \mathrm{Encrypt}(K_1, M_1)
\end{aligned}
$$

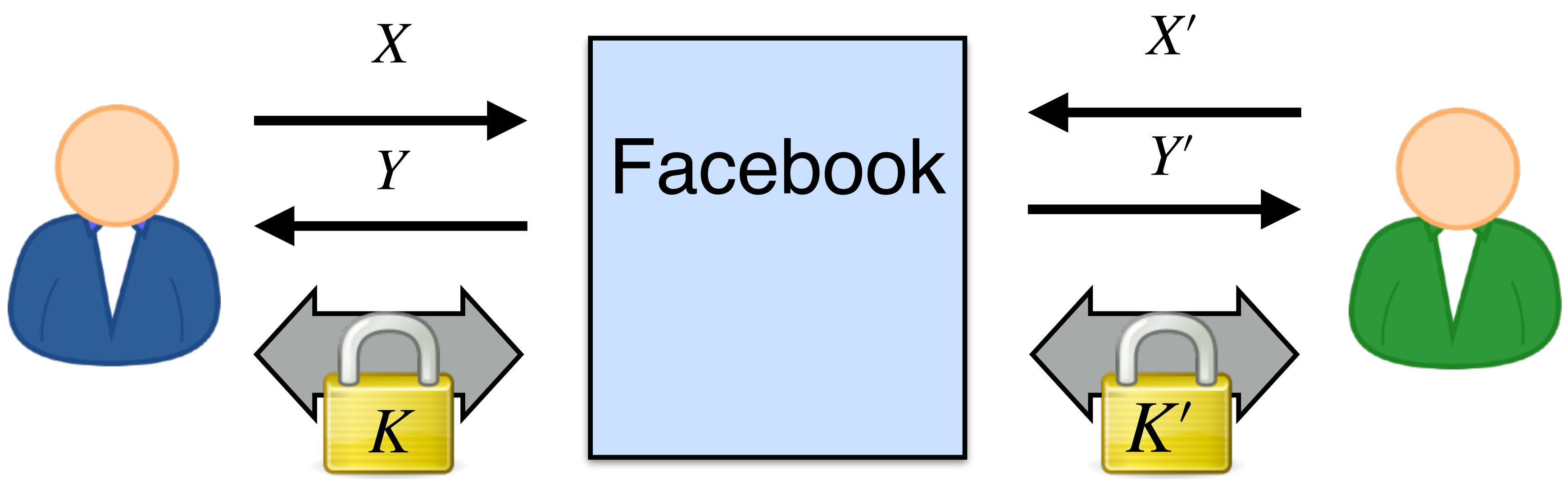
Messages encrypted with x_1, y_1, then x_2, y_1, then x_2, y_2, …

Self-Healing with Ratcheting

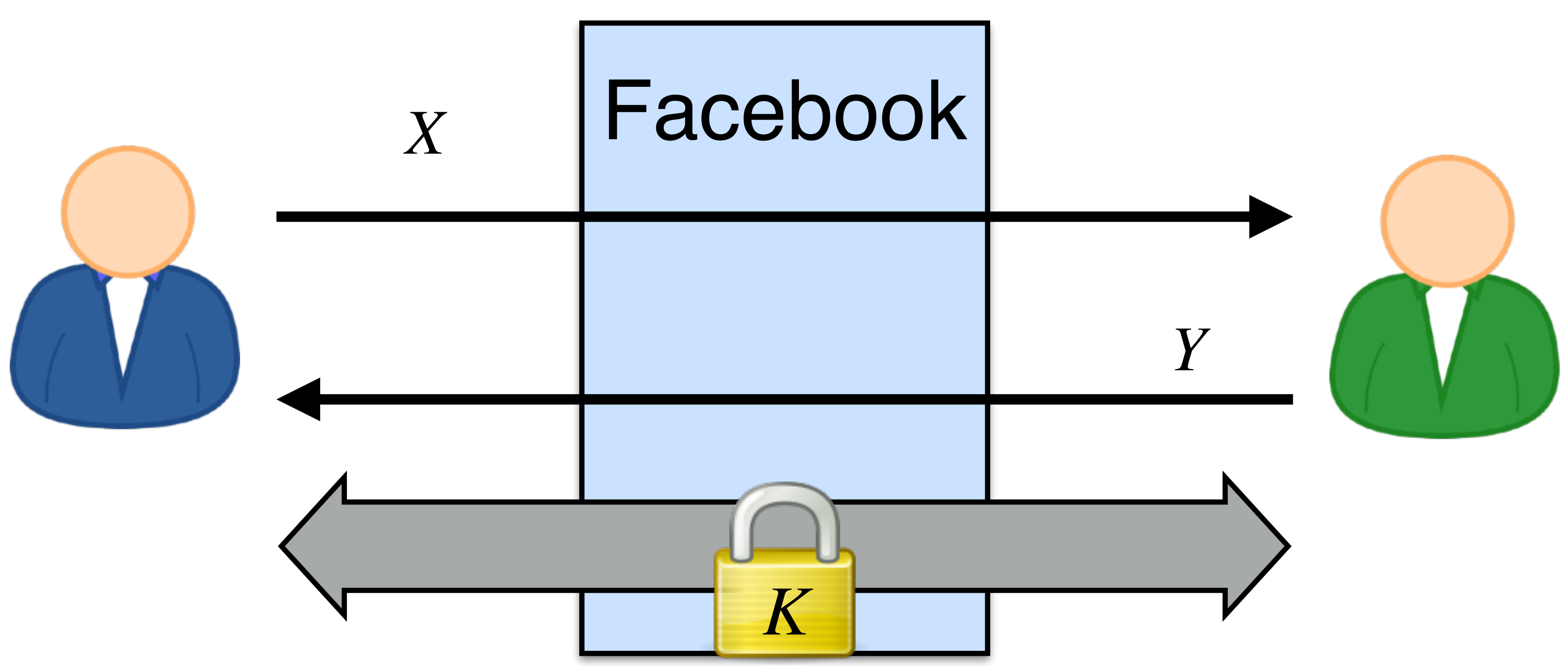


Messages encrypted with x_1, y_1, then x_2, y_1, then x_2, y_2, …

Authentication in Secure Messaging



vs.



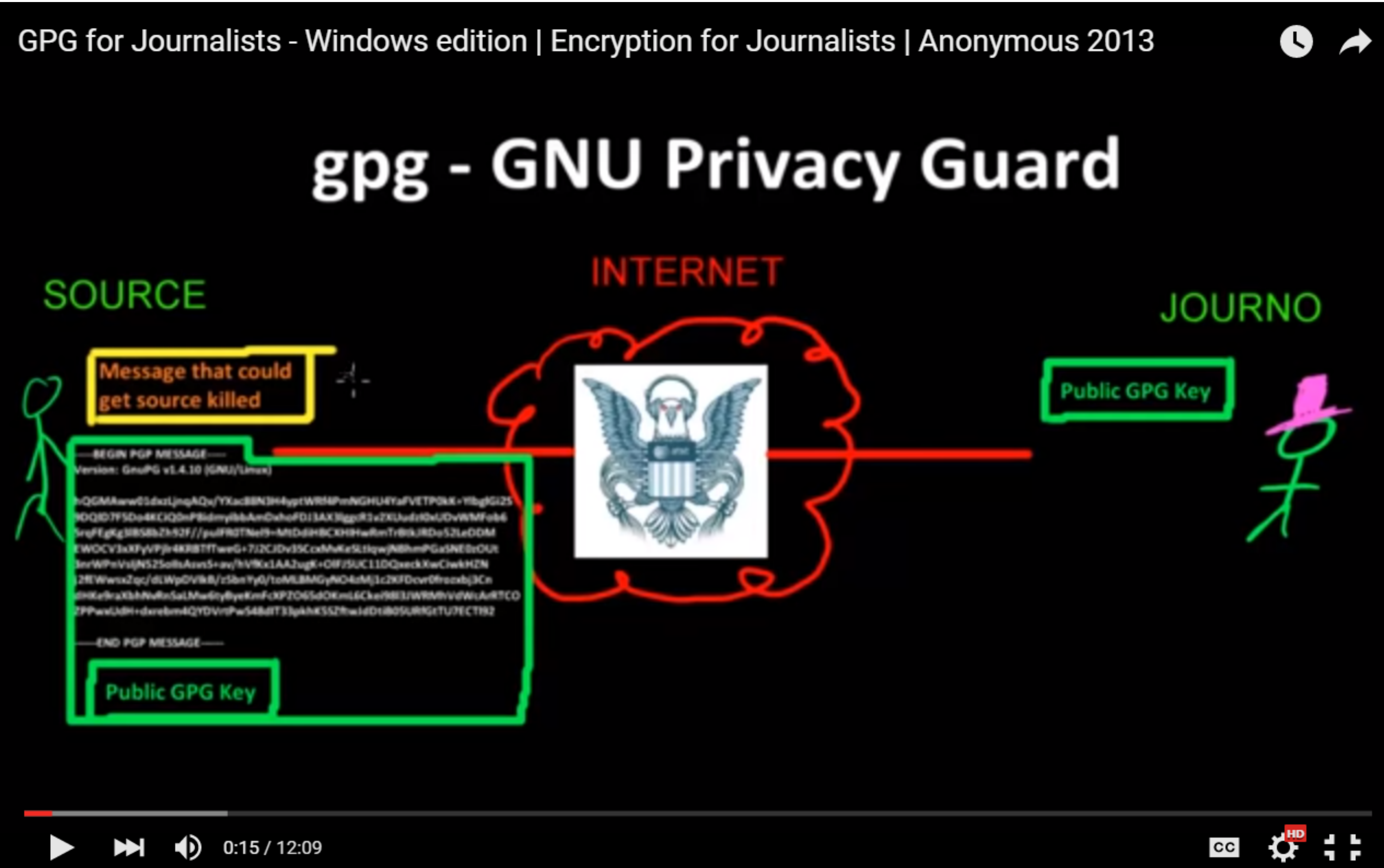
Encryption and Usable Key Exchanges

Why Glenn Couldn't Encrypt

Why Glenn Couldn't Encrypt

- http://vimeo.com/56881481
  - 1:50 – 3:37, 4:10 – 4:58, 11:15 – 11:43
- "And yet, Greenwald still didn't bother learning security protocols. 'The more he sent me, the more difficult it seemed,' he says. 'I mean, now I had to watch a f***ing video . . . ?'"
- Snowden ended up reaching out to Laura Poitras instead

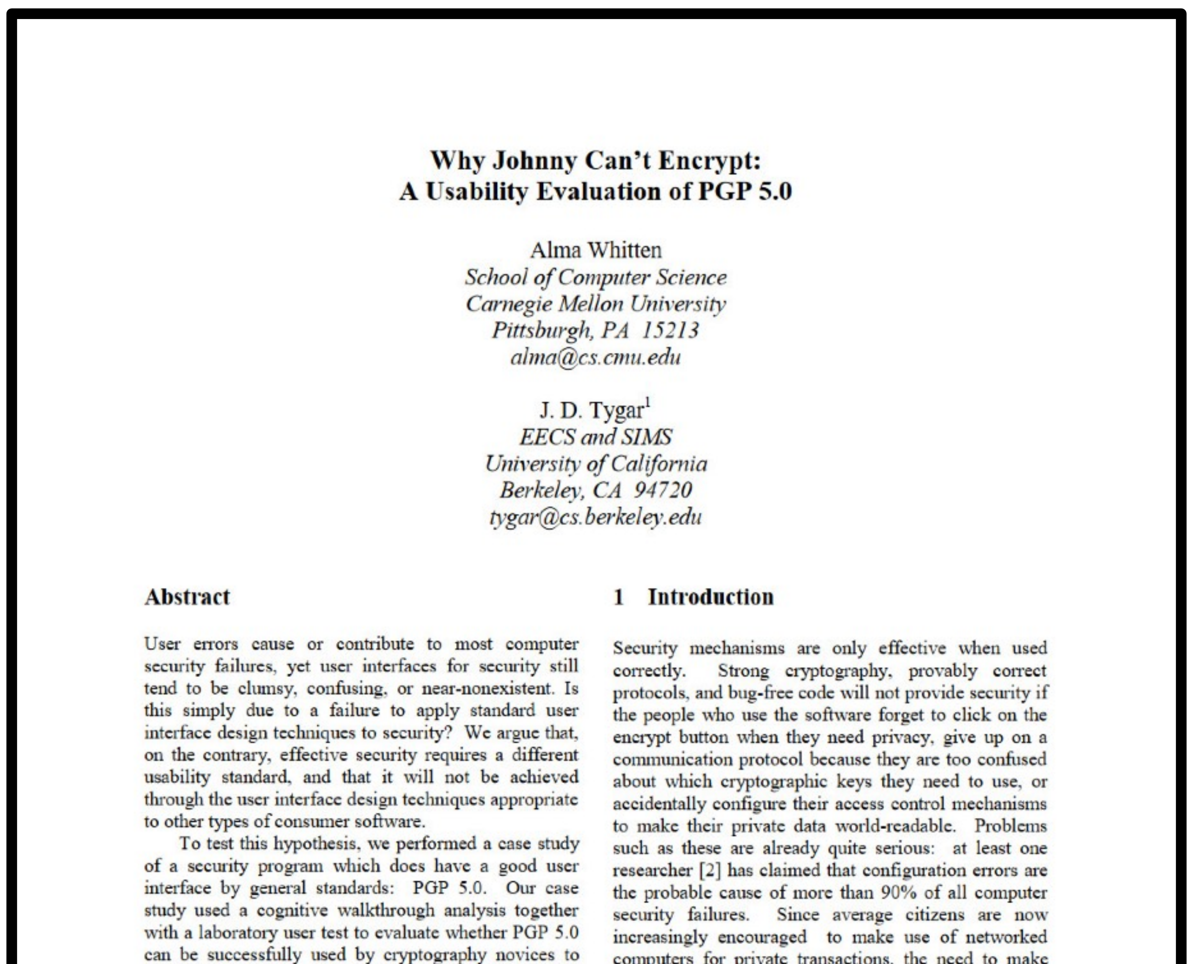
http://www.rollingstone.com/politics/news/snowden-and-greenwald-the-men-who-leaked-the-secrets-20131204
http://www.dailydot.com/politics/edward-snowden-gpg-for-journalists-video-nsa-glenn-greenwald/


GPG for Journalists - Windows edition | Encryption for Journalists | Anonymous 2013
gpg - GNU Privacy Guard
SOURCE
INTERNET
JOURNO
Message that could get source killed
Public GPG Key
BEGIN PGP MESSAGE
END PGP MESSAGE
Public GPG Key
0:15 / 12:09
CC
HD


GPG for Journalists - Windows edition | Encryption for Journalists | An...
Tor Mail :: Mail
MIT PGP Key Server
Search results for 'gnupp de adele'
gpa.exe
Generating Key...
Cancel
ivacy Assistant - Key Manager
Keys Windows Server Help
Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Key ID Expiry Date Owner Trust Validity User Name
P 92AB3FF7 never expires Unknown Unknown Adele (The friendly OpenPGP email robot) <adele-en@gnup
pinentry
Enter passphrase
Passphrase
Quality:
OK
Cancel
No keys selected
No default key selected in the preferences.
Contacts
Personal Add
Size
1,064 KB
9,779 KB
1 KB
2 KB
38,412 KB
1 KB
47,120 KB
Search temp
EN
12:08 AM
1/4/2013

Why Johnny Can’t Encrypt

- Classic paper in usable security (1999)
- Usability evaluation of PGP 5.0



Why Johnny Can’t Encrypt:
A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different usability standard, and that it will not be achieved through the user interface design techniques appropriate to other types of consumer software.

To test this hypothesis, we performed a case study of a security program which does have a good user interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to

1 Introduction

Security mechanisms are only effective when used correctly. Strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data world-readable. Problems such as these are already quite serious: at least one researcher [2] has claimed that configuration errors are the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make

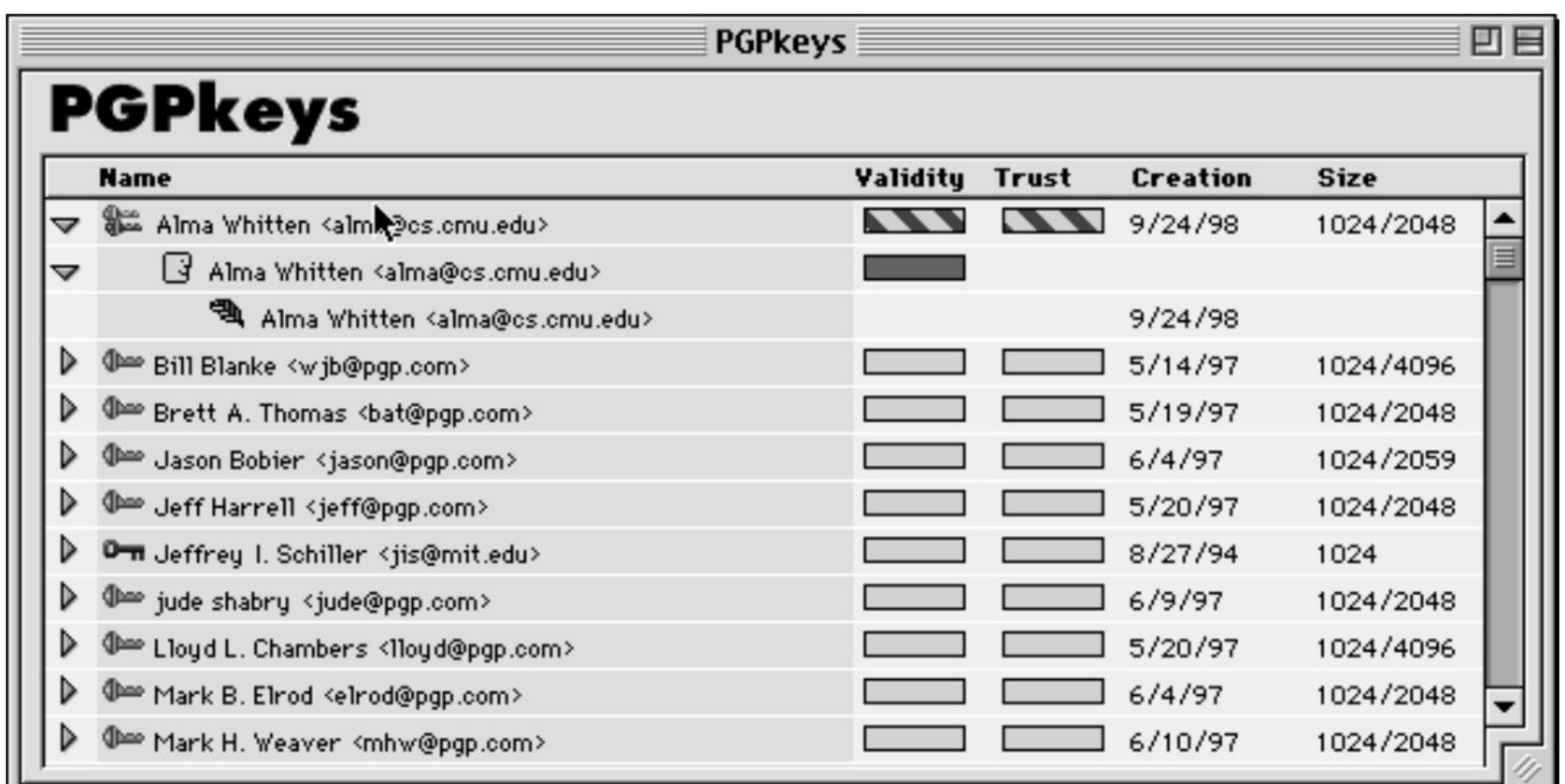
Alma Whitten and J.D. Tygar, “Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0” In Proceedings of USENIX Security 1999.

Why Johnny Can't Encrypt

- Some usable security principles:
  - Unmotivated user
  - Abstraction property
  - Lack of feedback
  - Barn door property
  - Weakest link property

Why Johnny Can't Encrypt

- Interfaces are bad
- Metaphors are wrong (and confusing)
- Opaque process
- Key management is difficult



Complexity of Asymmetric Encryption

- User creates a keypair
  - Public key should be widely distributed
  - Private key should never be distributed
- Private key protected with a password
- Two very different functions:
  - Encrypting (secrecy)
  - Signing (authenticity)
- Need person's key to communicate

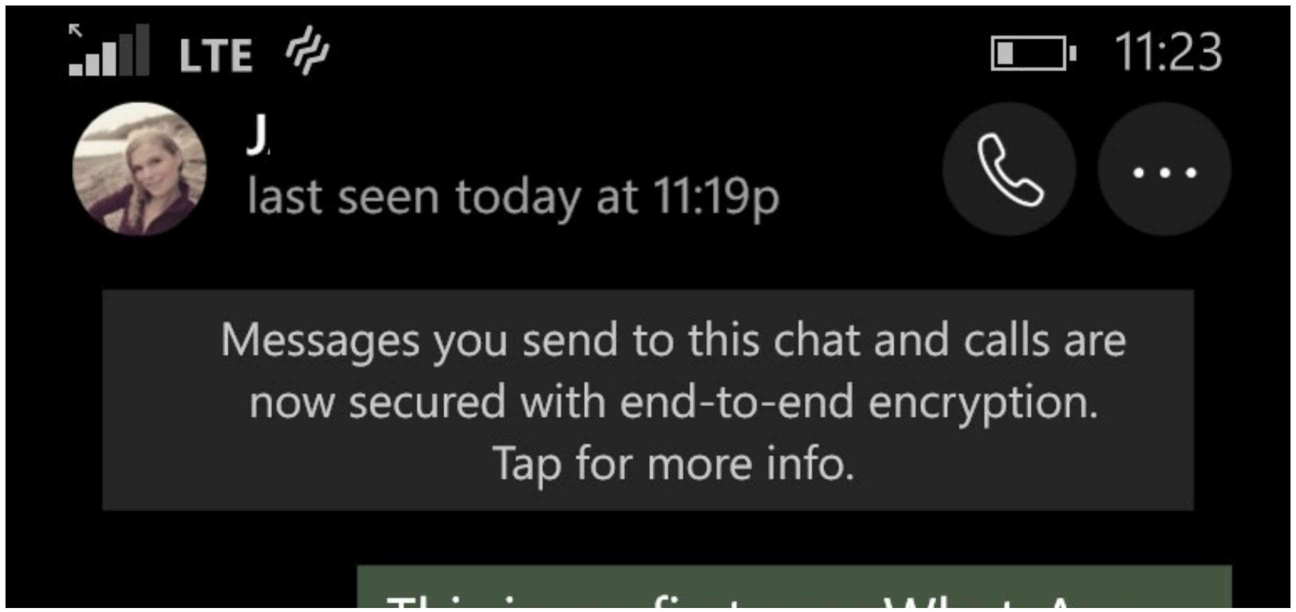
(Just Some) Usability Problems

- Encryption is rarely configured by default
- Public/private key encryption
  - How to get someone's public key?
  - How do I make it work on my phone?
- You often need a good password
  - …and you can't lose it or forget it
- Configuring multiple devices
- "Only paranoid people use encryption"

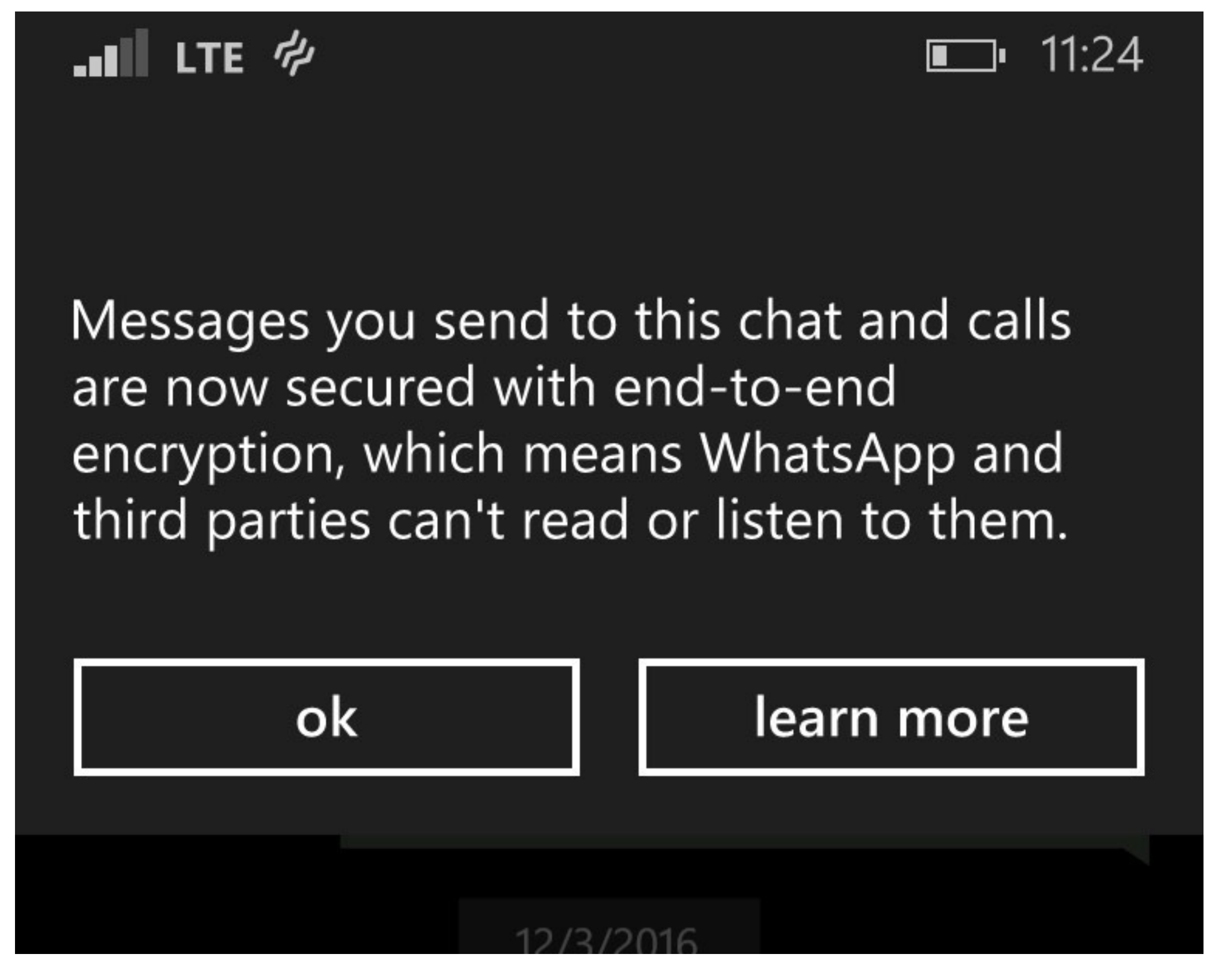
Do You Have the Right Key?

- Person-in-the-middle attack
- Ways of trusting a person→key binding:
  - Public-key infrastructure (certifying authorities)
  - Web of trust (someone you trust vouches)
  - Exchange keys out of band
  - Platform provider verifies
  - Key servers, such as https://pgp.mit.edu/

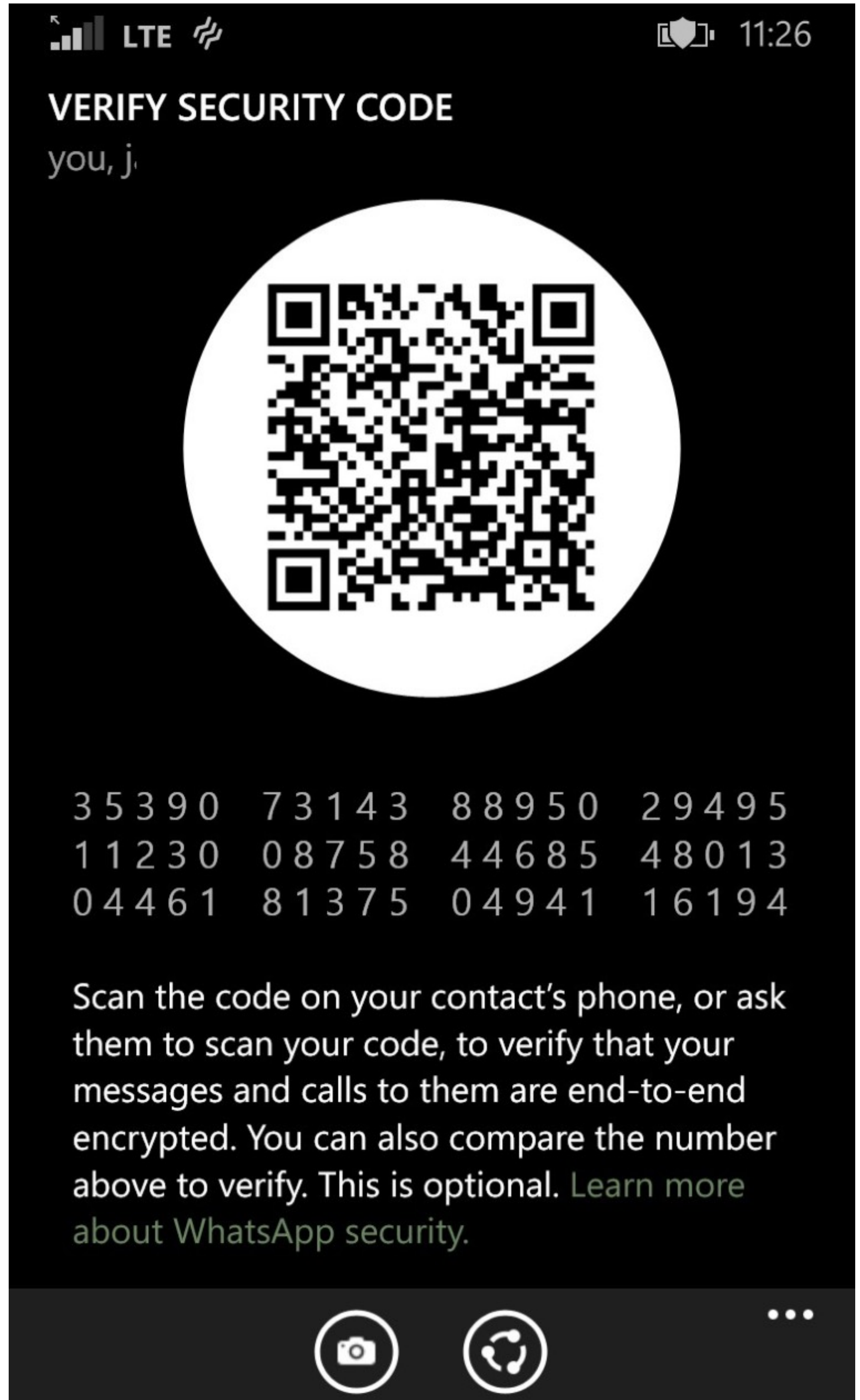
Key Verification on Whatsapp



WhatsApp



Verifying You Have the Right Key



Verifying You Have the Right Key

GnuPG

`3A70 F9A0 4ECD B5D7 8A89`
`D32C EDA0 A352 66E2 C53D`

OpenSSH

`ef:6d:bb:4c:25:3a:6d:f8:79:d3:a7:90:db:c9:b4:25`

bubblebabble

`xucef-masiv-zihyl-bicyr-zalot-cevyt-lusob-`
`negul-biros-zuhal-cixex`

OTR

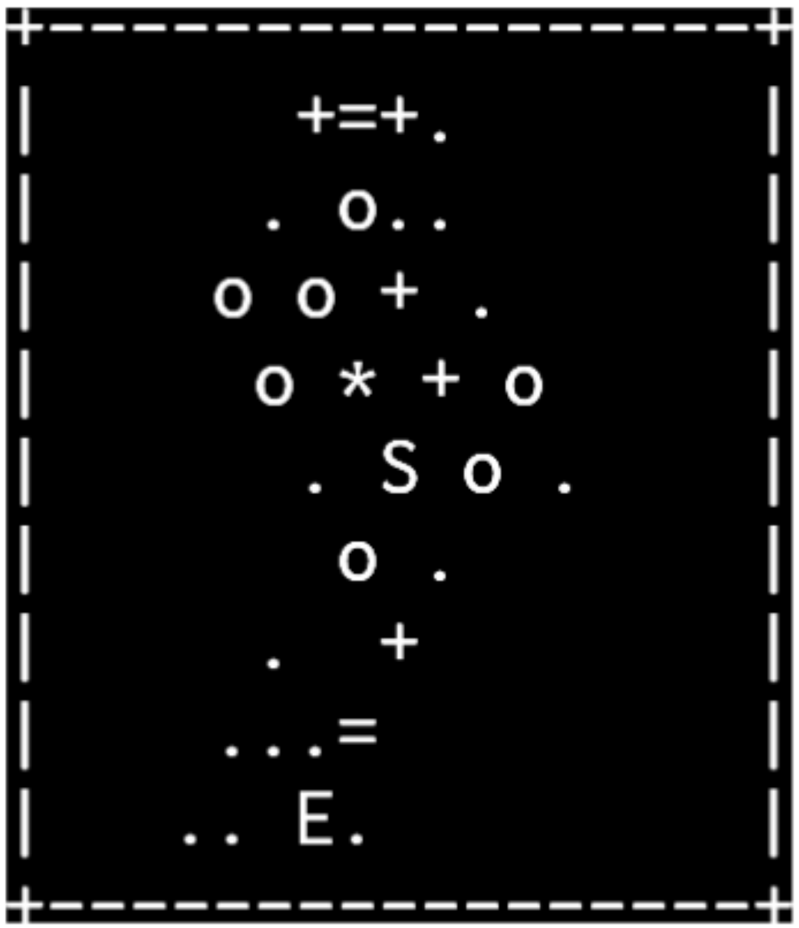
`4206EA15 1E029807 C8BA9366 B972A136 C6033804`

WhatsApp

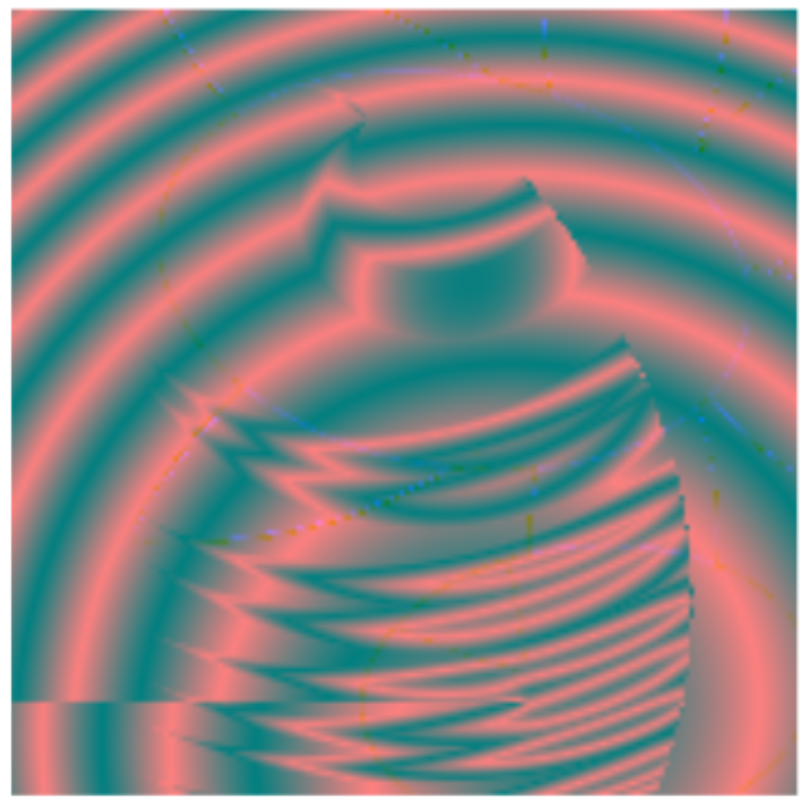
`54040 65258 71972 73974`
`10879 55897 71430 75600`
`25372 60226 27738 71523`

Joshua Tan, Lujo Bauer, Joseph Bonneau, Lorrie Faith Cranor, Jeremy Thomas, Blase Ur. Can Unicorns Help Users Compare Crypto Key Fingerprints? In Proceedings of CHI 2017.

Verifying You Have the Right Key



(a) OpenSSH Visual Host Key



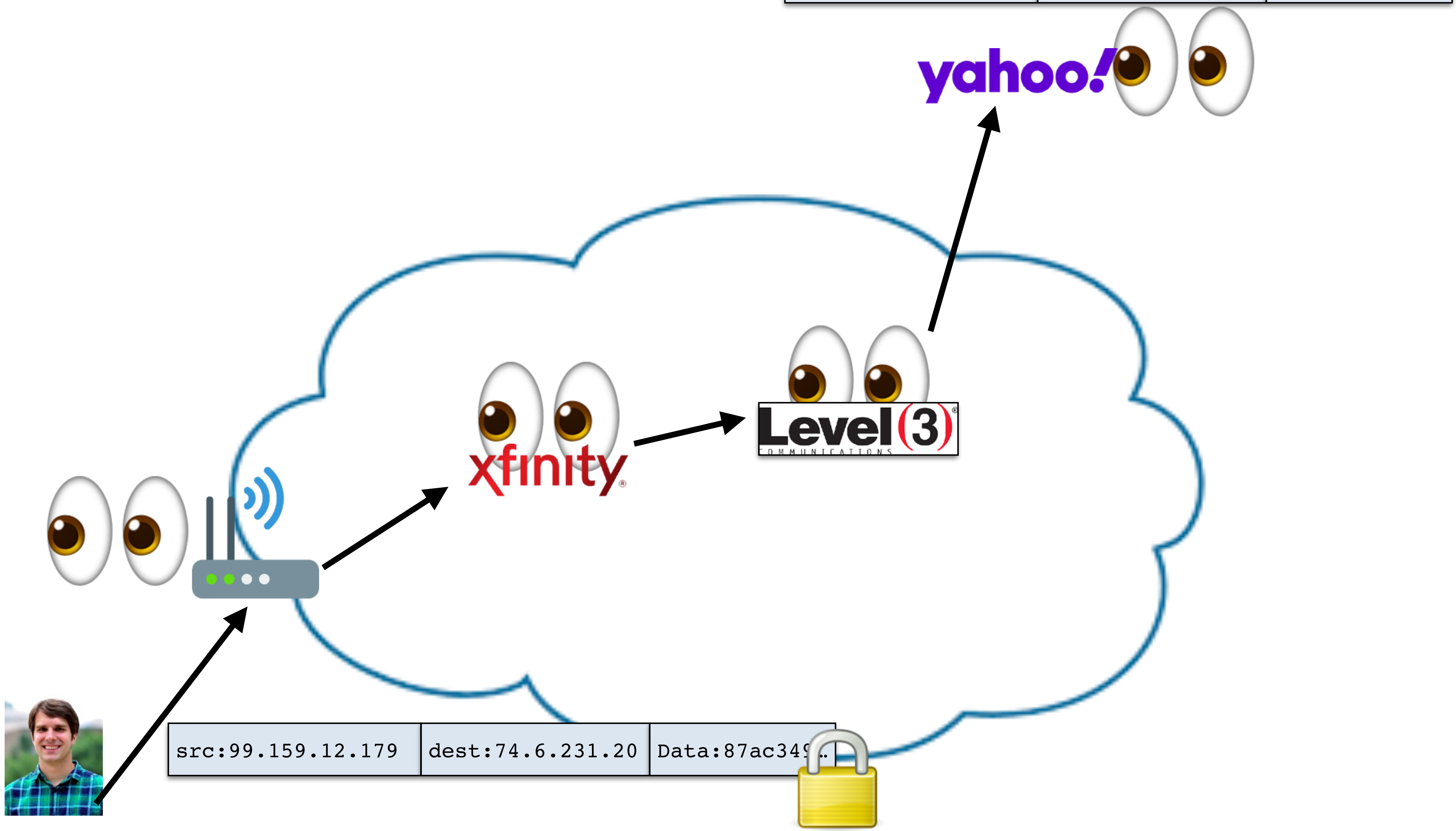
(b) Vash

(c) Unicorn

Joshua Tan, Lujo Bauer, Joseph Bonneau, Lorrie Faith Cranor, Jeremy Thomas, Blase Ur. Can Unicorns Help Users Compare Crypto Key Fingerprints? In Proceedings of CHI 2017.

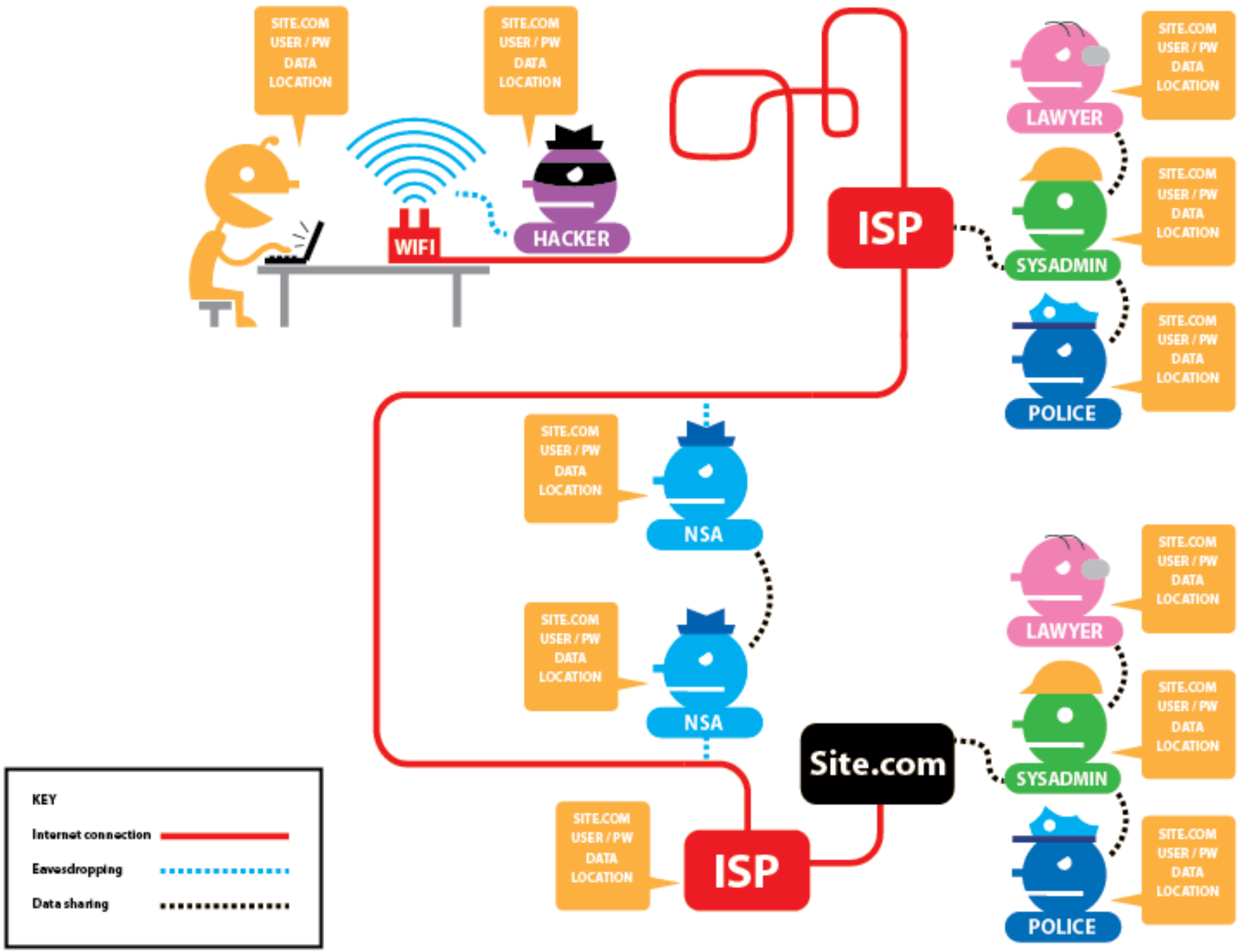
Anonymous Routing

Anonymity on the Internet



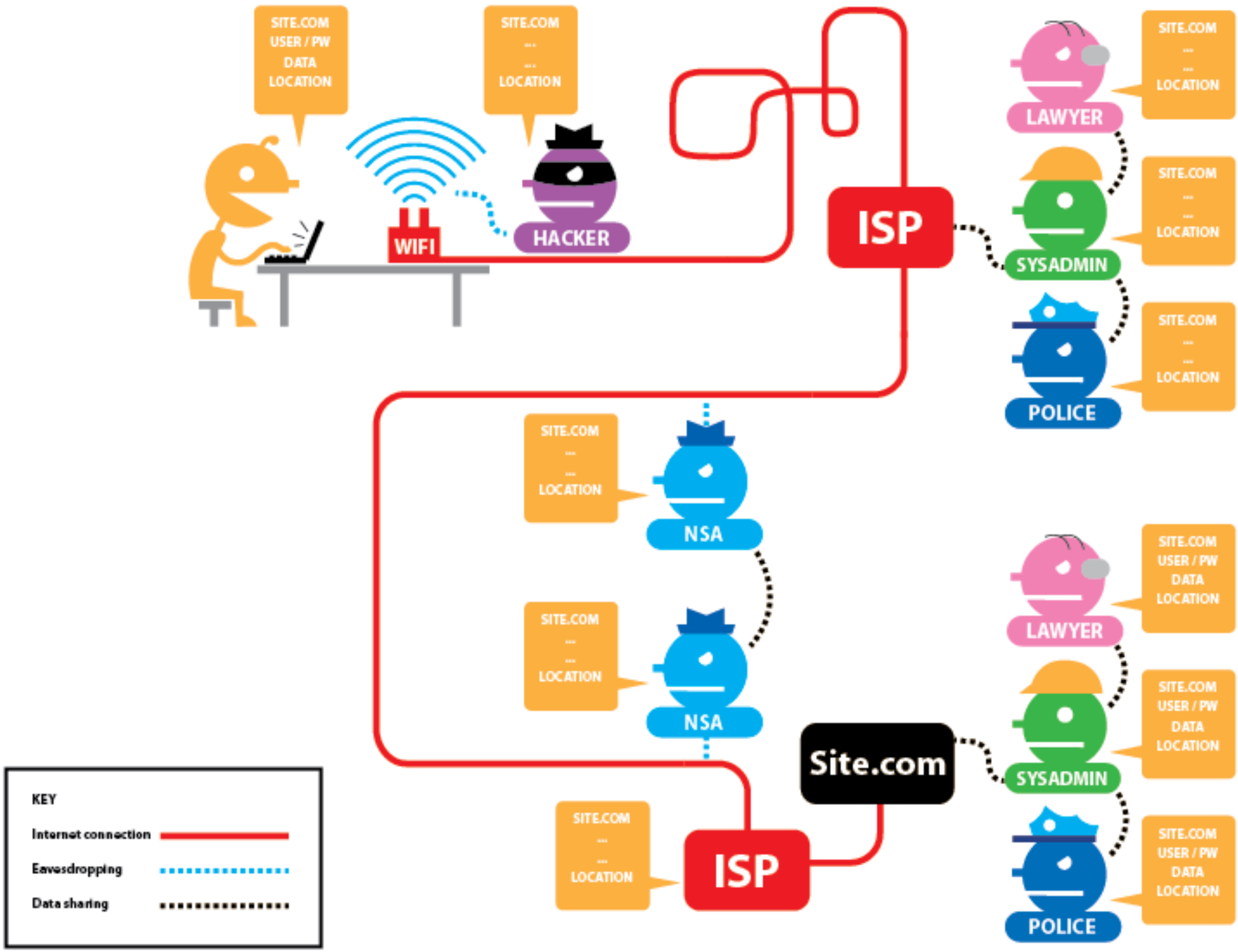
Everyone knows Blase is visiting his favorite website, even if using TLS

Who can see what: If TLS is not used



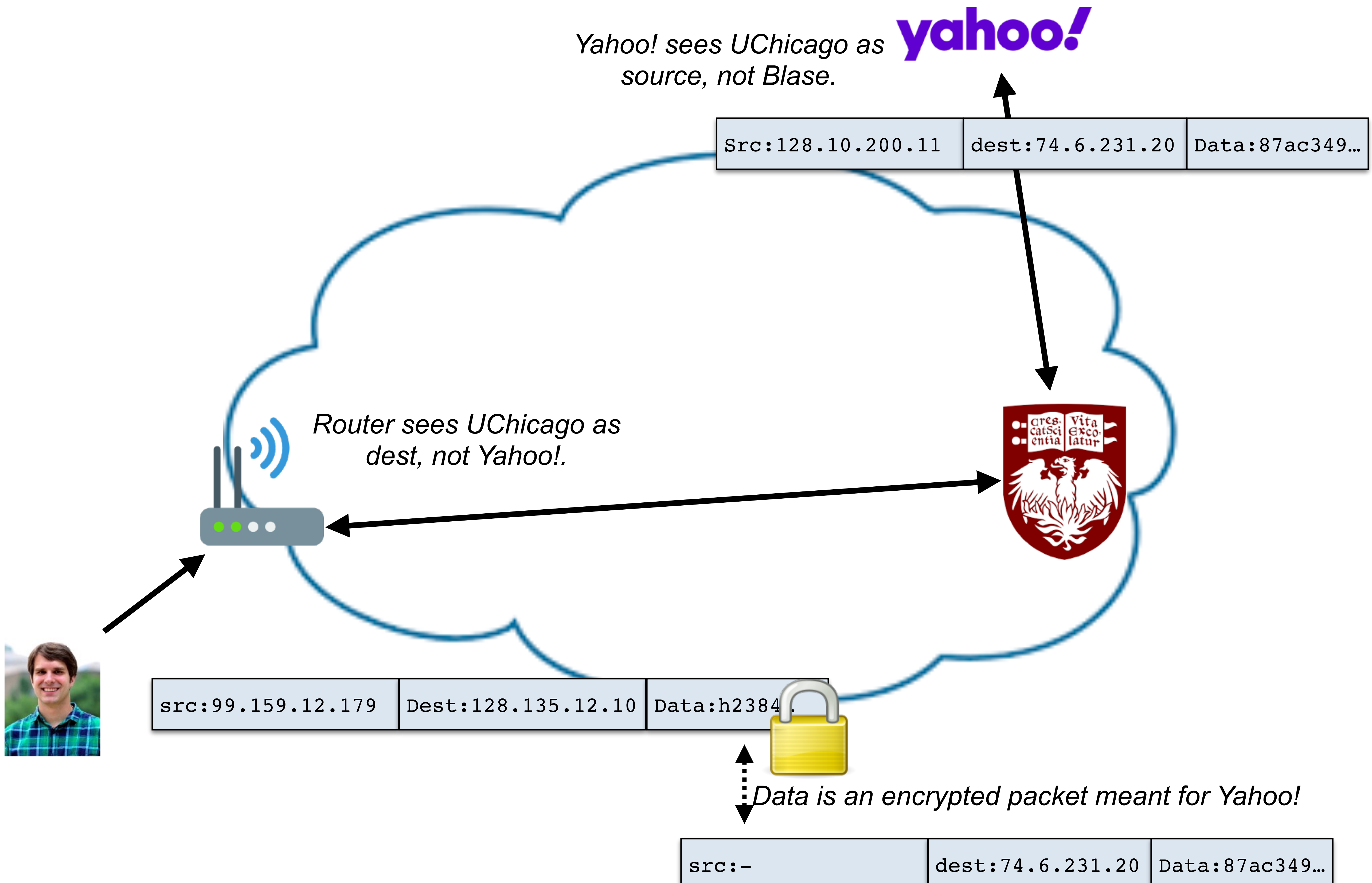
(Source: https://support.torproject.org/)

Who can see what: With TLS



(Source: https://support.torproject.org/)

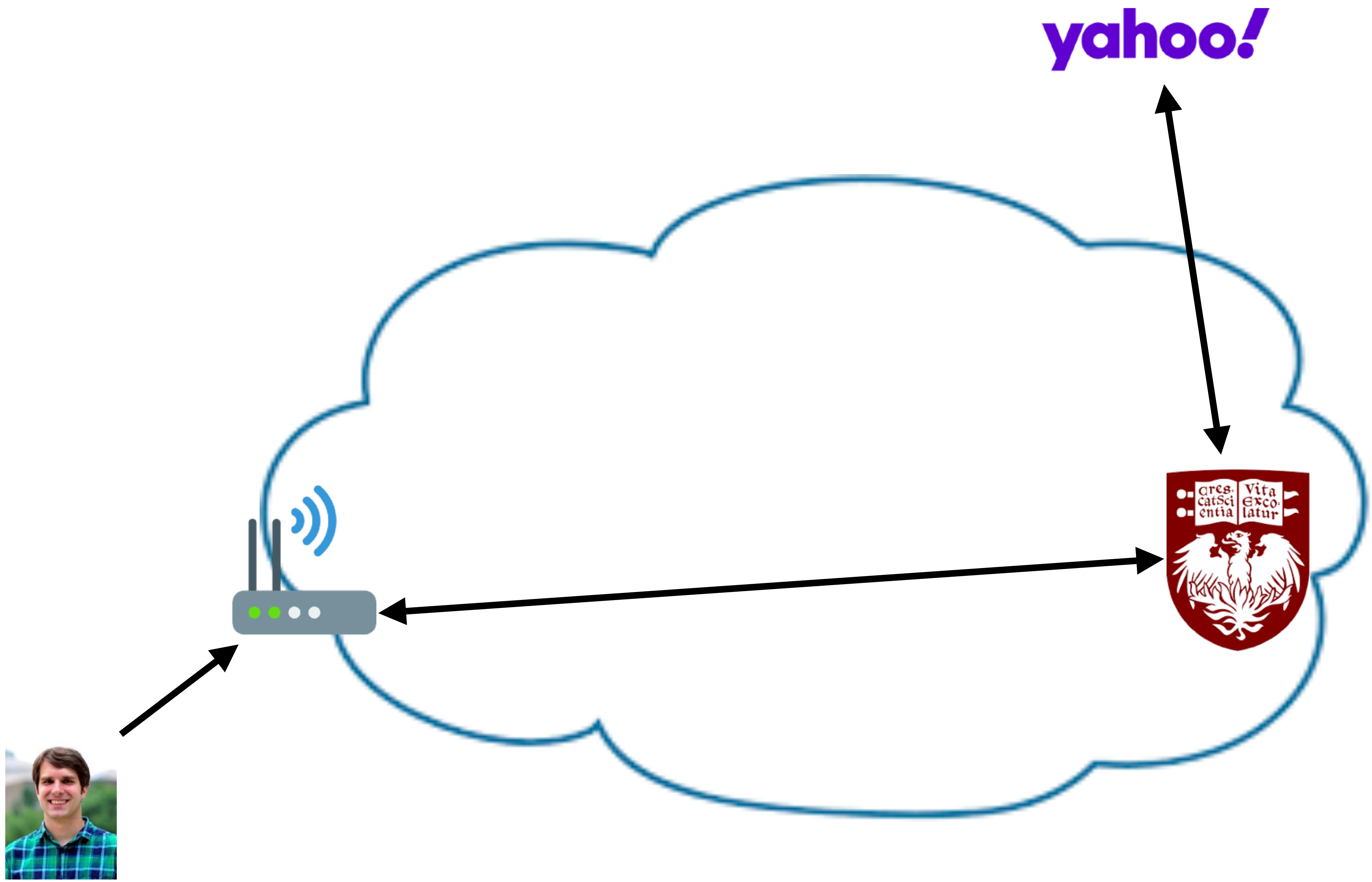
One Tool: Virtual Private Networks (VPNs)



Uses of VPNs

1. Avoid snooping by ISPs
2. Circumventing location-based restrictions
3. Corporate access control (e.g. Chicago's cVPN)

Trust in VPNs



VPN service knows what Blase is doing - it must be trusted.

Tor: The Onion Router

- Technology called *onion routing* developed in 90s by Office of Naval Research

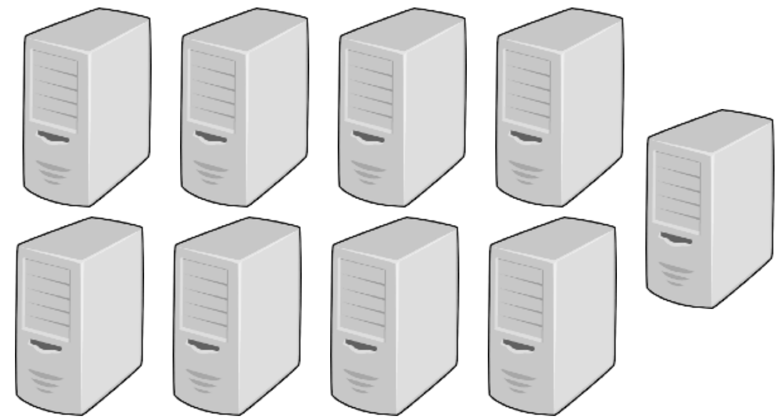
- Published as research paper by Dingledine, Matthewson, Syverson in 2004

- Today, about 2 million users connected at any given time

Tor Infrastructure

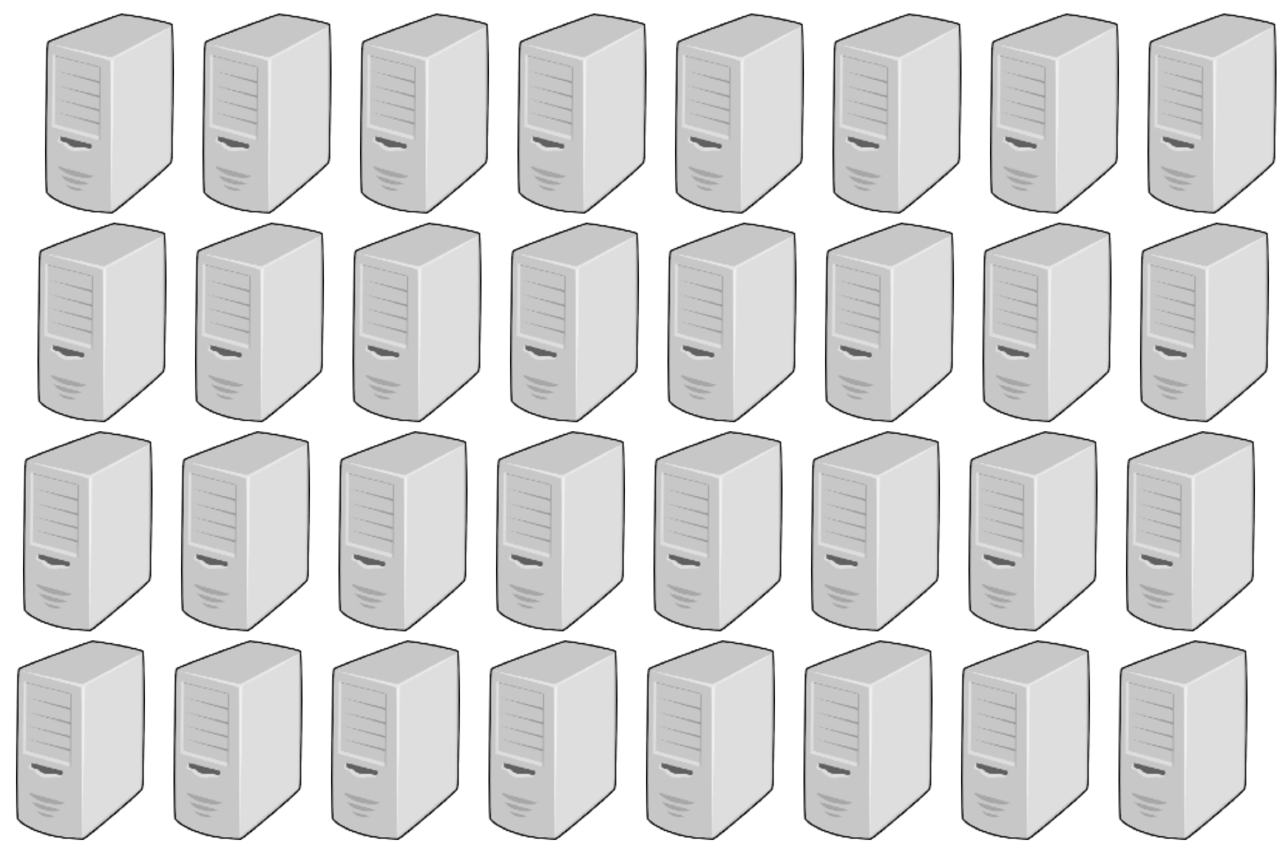
Client running
custom software

9 Directory Servers

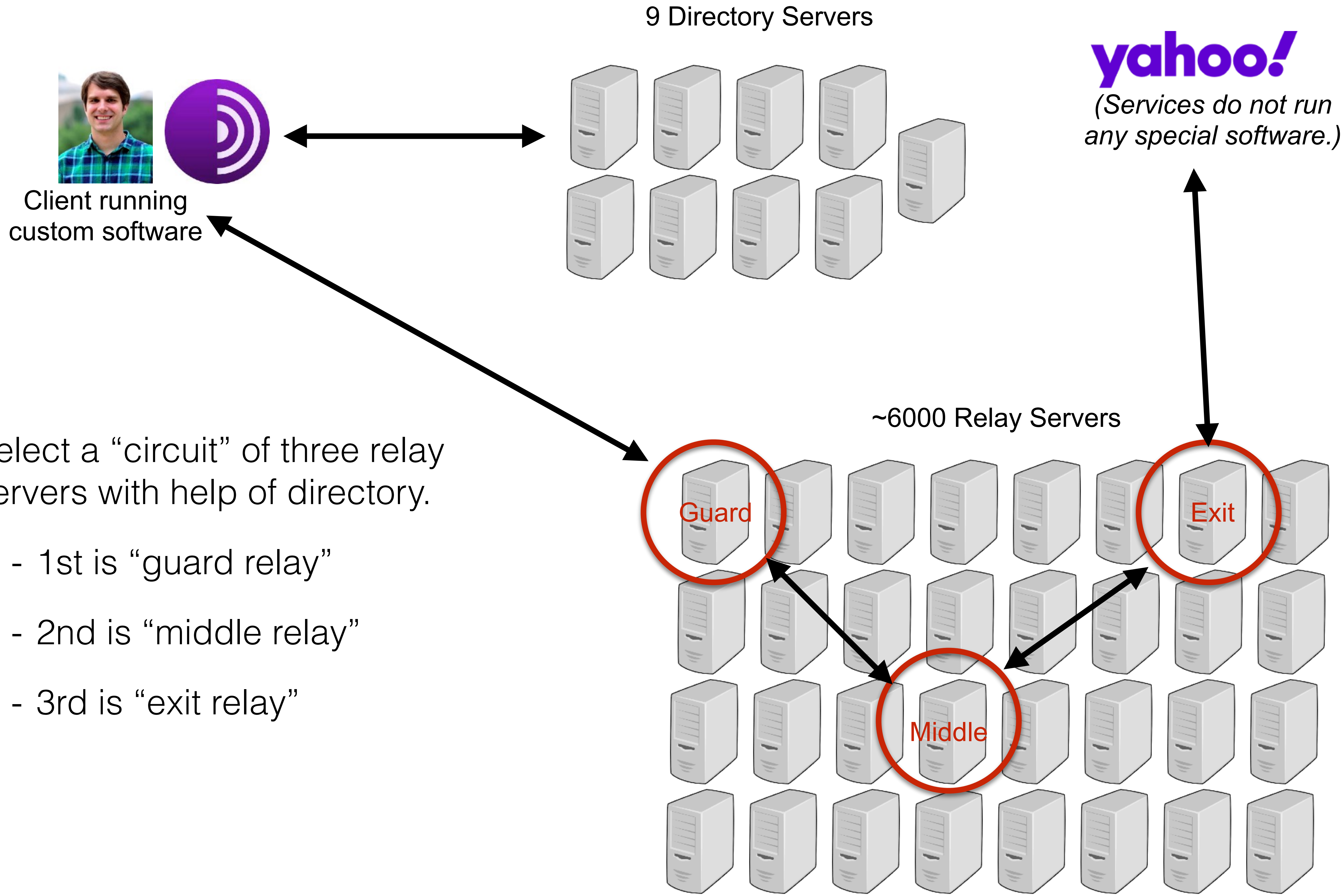


(Services do not run any special software.)

~6000 Relay Servers



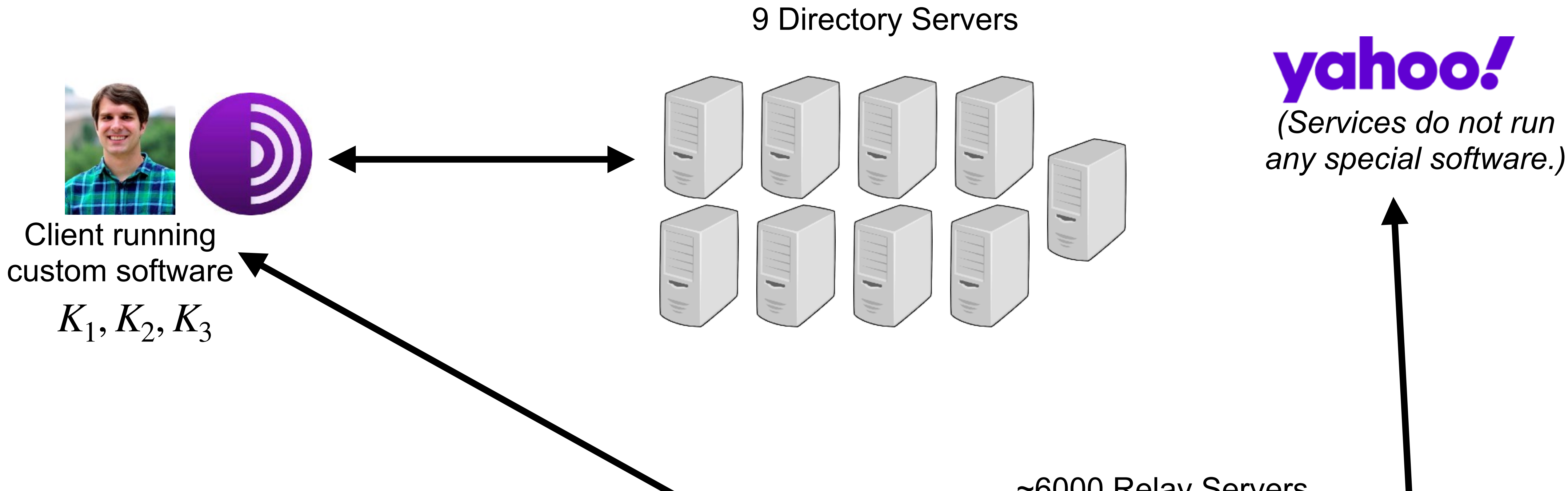
Step One: Pick a Circuit



Select a “circuit” of three relay servers with help of directory.

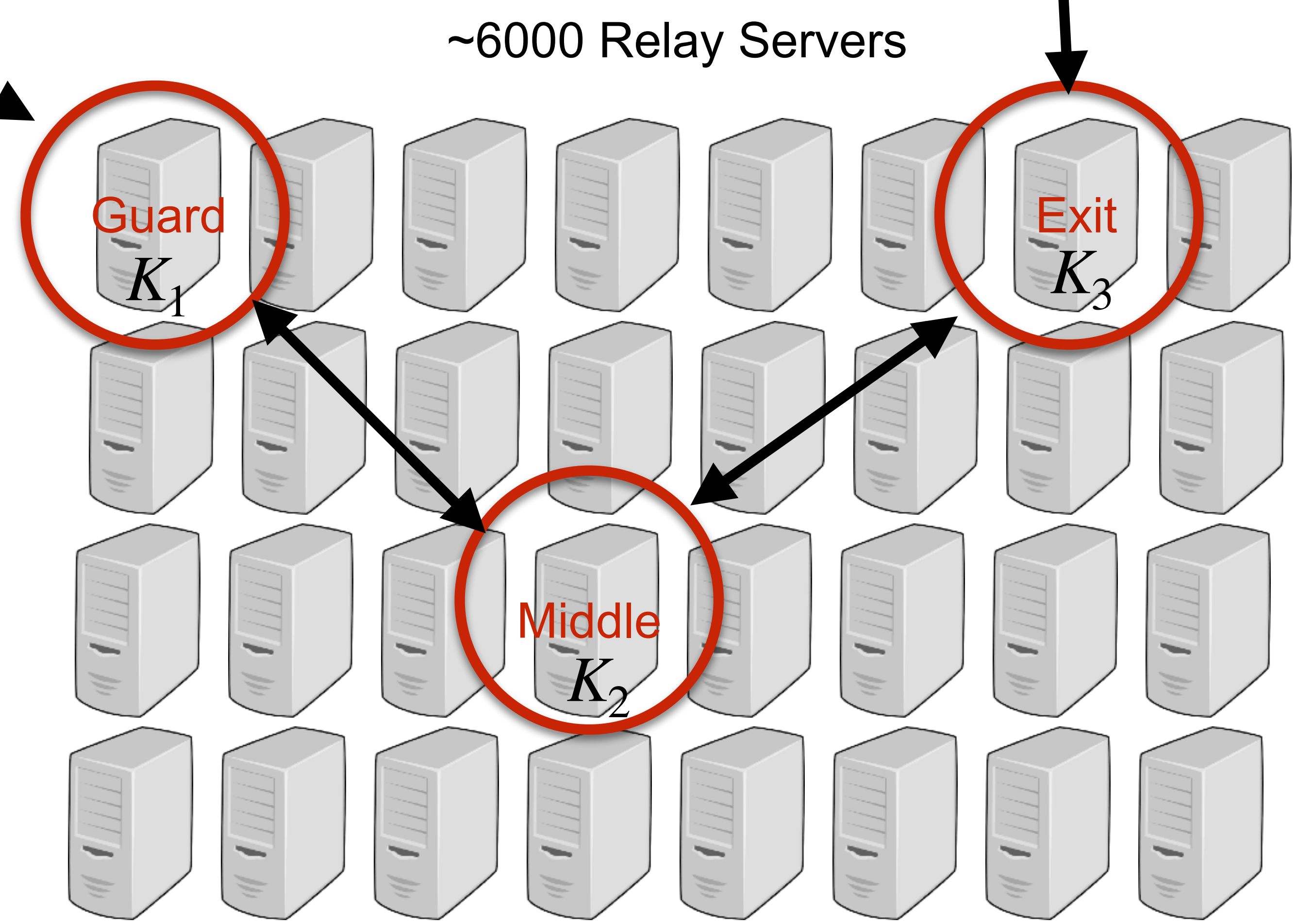
- 1st is “guard relay”
- 2nd is “middle relay”
- 3rd is “exit relay”

Step Two: Setup Circuit

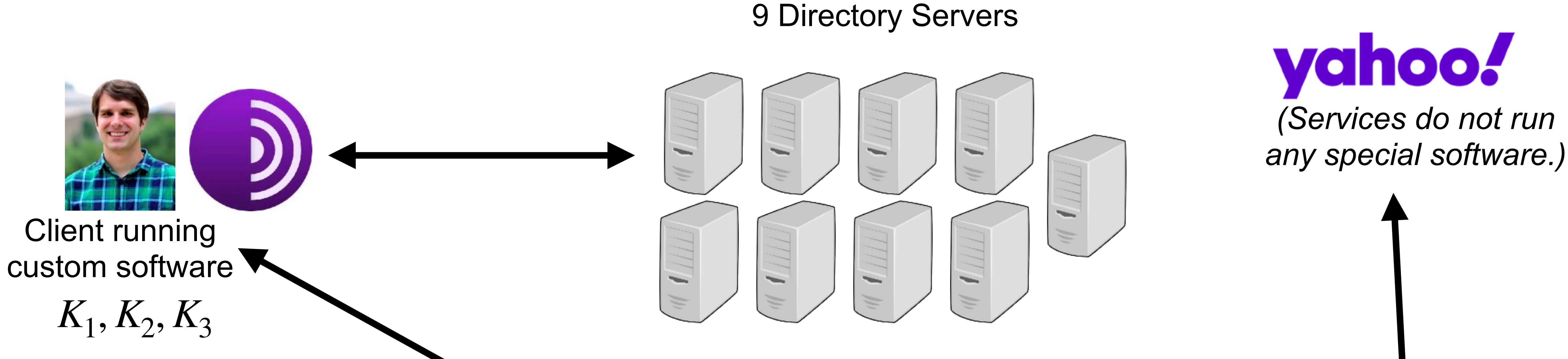


Agree on keys with each relay

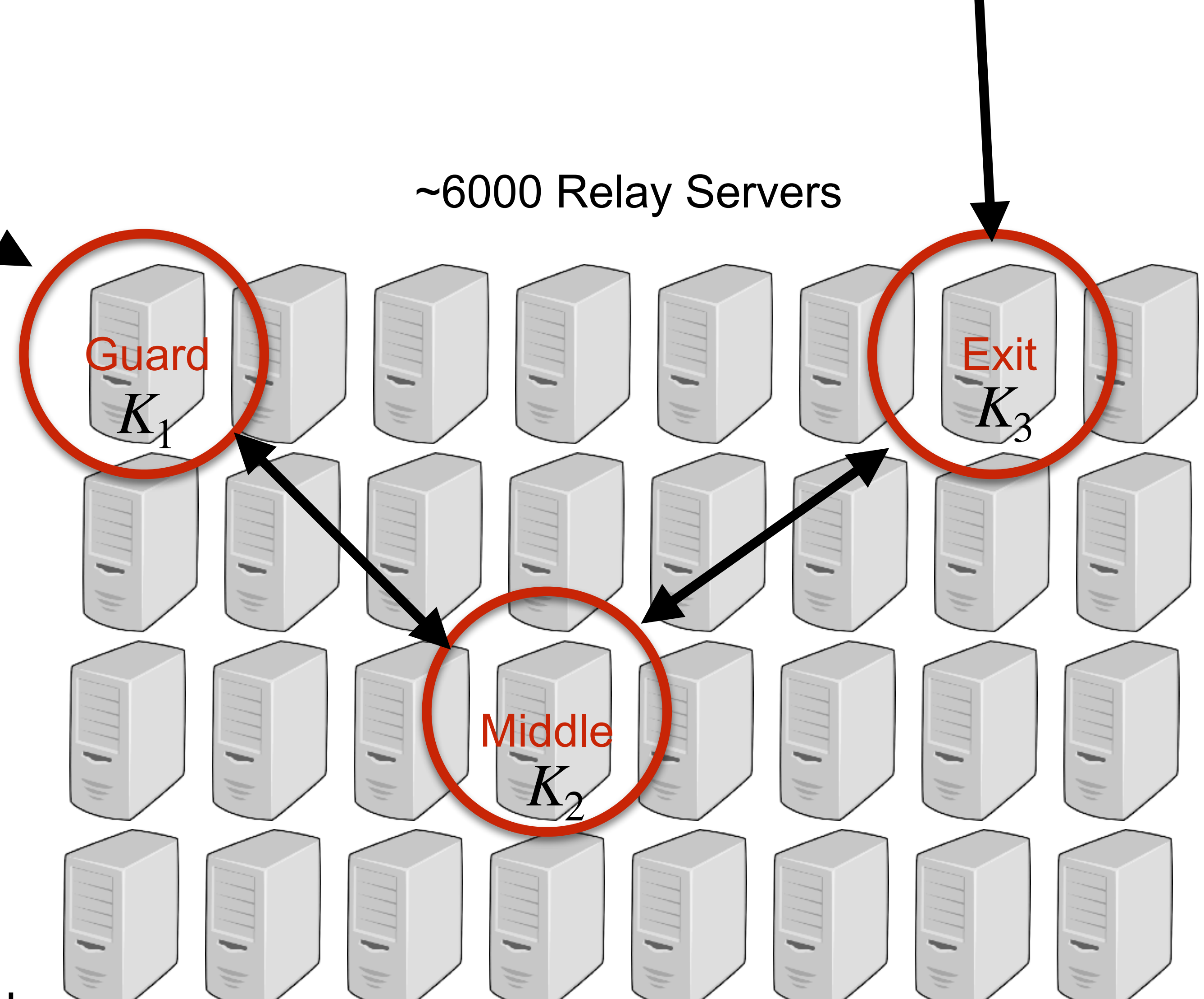
- Run key exchange with guard directly to get K_1
- Tunnel through guard to middle node and run key exchange to get K_2
- Tunnel through first two nodes to agree on key K_3 with exit node

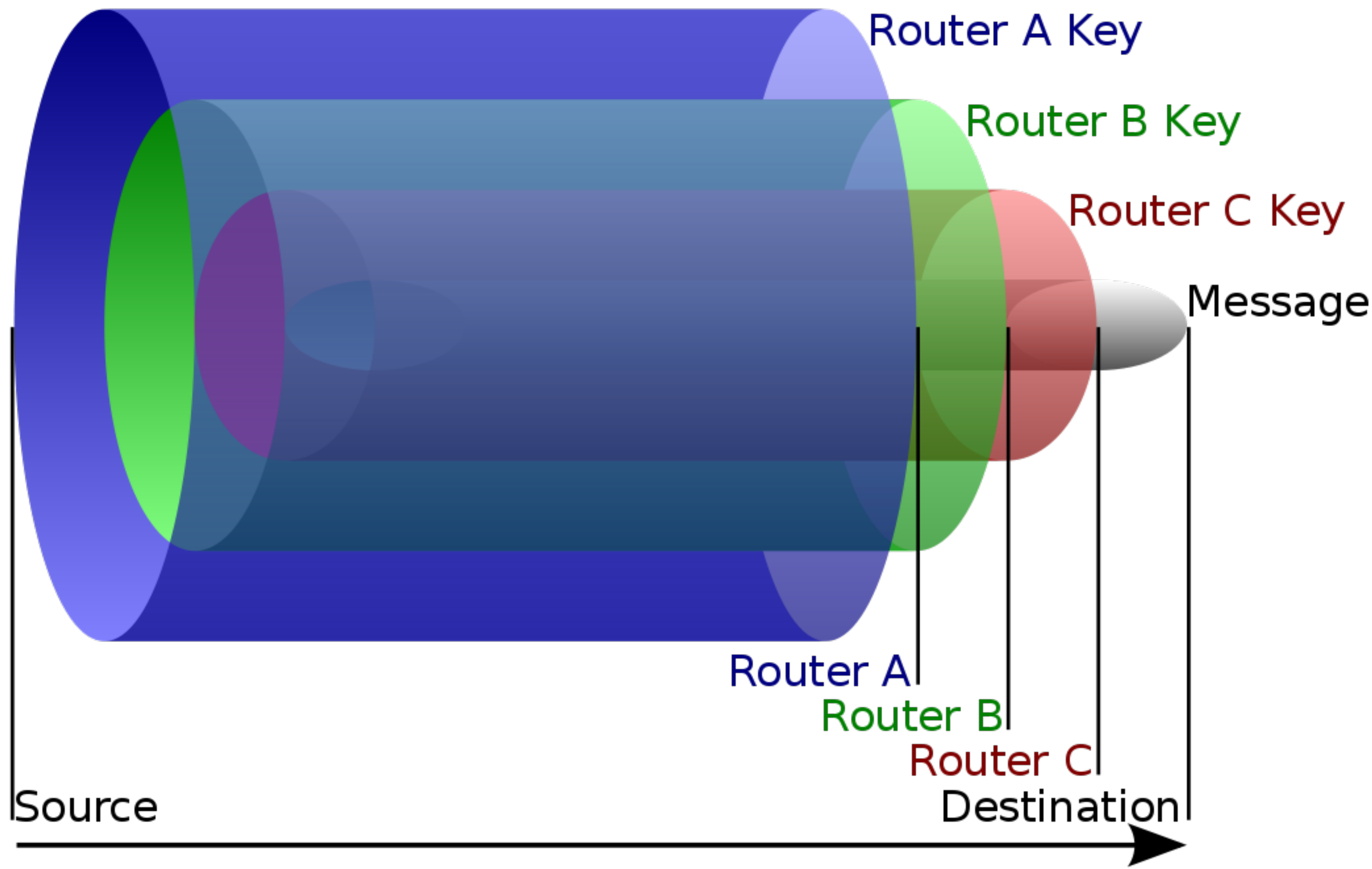


Step Three: Communicate with Onion Encryption

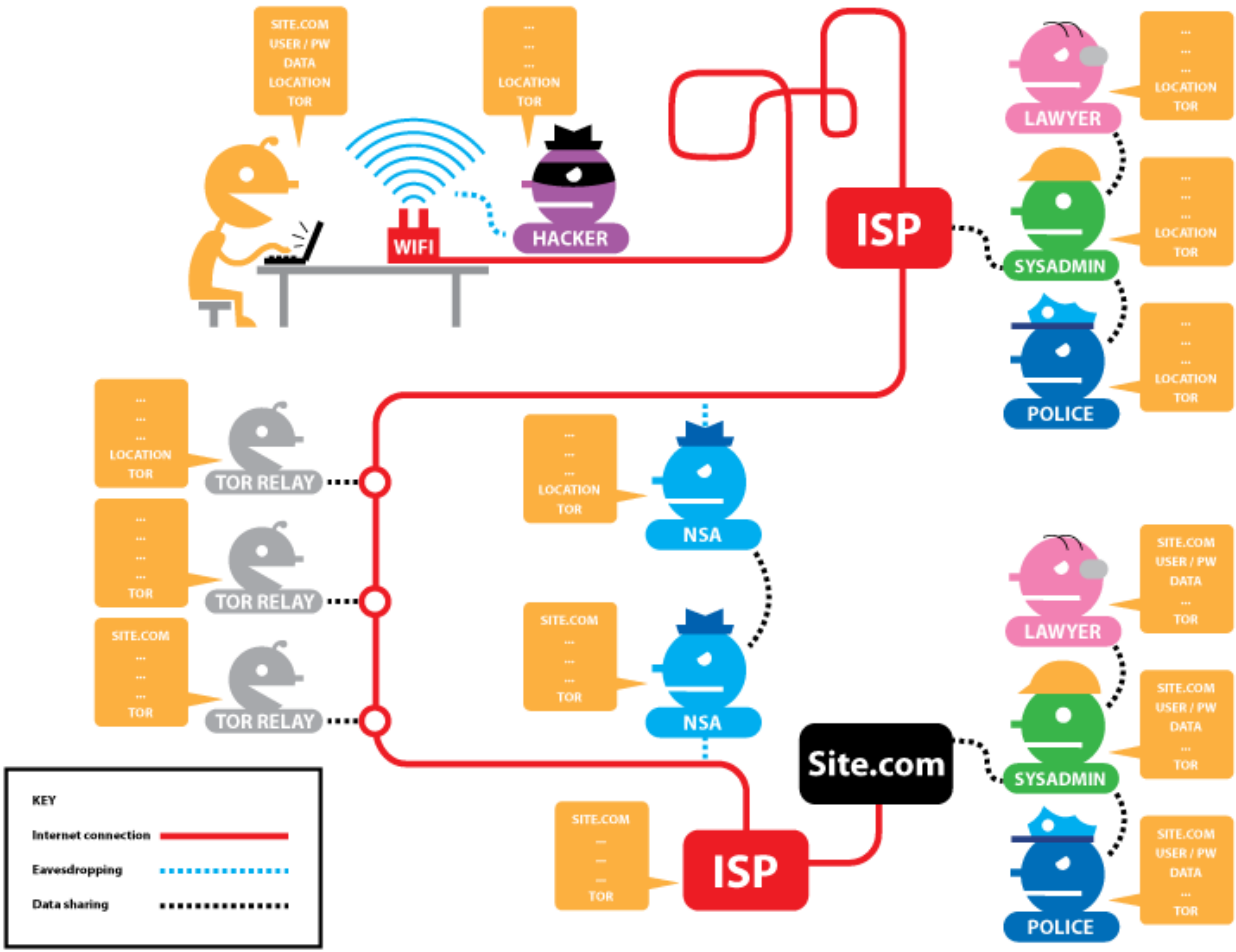


1. To send M to Yahoo!:
   a. Encrypt with K_1
   b. Encrypt *again* with K_2
   c. Encrypt *yet again* with K_3
2. Send ciphertext to guard
3. Guard decrypts a layer, forwards
4. Middle decrypts a layer, forwards
5. Exit decrypts last layer, sends to Yahoo!




Router A Key
Router B Key
Router C Key
Message
Router A
Router B
Router C
Source
Destination

Who can see what: With Tor



(Source: https://support.torproject.org/)

Attacks on Tor/Onion Routing

1. Controlling both guard and exit defeats all protection

2. If not enough users, then there is no “blending in”

3. Destination may implement usual tracking measures - use a special browser!

4. Often just detecting that you’re using Tor is enough to compromise you.

Many other attacks on availability, protocol bugs, etc https://www.torproject.org/download/download#warning

Want Tor to really work?

You need to change some of your habits, as some things won't work exactly as you are used to.

a. **Use Tor Browser**

Tor does not protect all of your computer's Internet traffic when you run it. Tor only protects your applications that are properly configured to send their Internet traffic through Tor. To avoid problems with Tor configuration, we strongly recommend you use the Tor Browser. It is pre-configured to protect your privacy and anonymity on the web as long as you're browsing with Tor Browser itself. Almost any other web browser configuration is likely to be unsafe to use with Tor.

b. **Don't torrent over Tor**

Torrent file-sharing applications have been observed to ignore proxy settings and make direct connections even when they are told to use Tor. Even if your torrent application connects only through Tor, you will often send out your real IP address in the tracker GET request, because that's how torrents work. Not only do you deanonymize your torrent traffic and your other simultaneous Tor web traffic this way, you also slow down the entire Tor network for everyone else.

c. **Don't enable or install browser plugins**

Tor Browser will block browser plugins such as Flash, RealPlayer, Quicktime, and others: they can be manipulated into revealing your IP address. Similarly, we do not recommend installing additional addons or plugins into Tor Browser, as these may bypass Tor or otherwise harm your anonymity and privacy.

d. **Use HTTPS versions of websites**

Tor will encrypt your traffic to and within the Tor network, but the encryption of your traffic to the final destination website depends upon on that website. To help ensure private encryption to websites, Tor Browser includes HTTPS Everywhere to force the use of HTTPS encryption with major websites that support it. However, you should still watch the browser URL bar to ensure that websites you provide sensitive information to display a blue or green URL bar button, include **https://** in the URL, and display the proper expected name for the website. Also see EFF's interactive page explaining how Tor and HTTPS relate.

e. **Don't open documents downloaded through Tor while online**

Tor Browser will warn you before automatically opening documents that are handled by external applications. **DO NOT IGNORE THIS WARNING**. You should be very careful when downloading

The End