Security II: Security Strikes Back

15-441/641 Spring 2019

Profs Peter Steenkiste & **Justine Sherry**

Carnegie Mellon University

Cryptography Overview

	Symmetric	**Asymmetric**
Confidentiality	One-Time Pad Stream Ciphers Block Ciphers	Encrypt w/ Public Key
Integrity	Message Authentication Code (e.g., HMAC, CBC-MAC)	Digital Signature
Authentication	MAC + Nonce	Digital Signature + Nonce

What is confidentiality?

What is integrity?

What is authentication?

Why does authentication require a nonce?

How many keys are needed for two folks to talk using symmetric cryptography?

How many keys are needed for two folks to talk using asymmetric cryptography?

Where we left off on Tuesday…

How do I get these keys in the first place??
Remember:

- Symmetric key primitives assumed Alice and Bob had already shared a key.
- Asymmetric key primitives assumed Alice knew Bob's public key.

This may work with friends, but when was the last time you saw Amazon.com walking down the street?

What do we use in practice?

Let’s put it all together!

Transport Layer Security (TLS)
aka Secure Socket Layer (SSL)

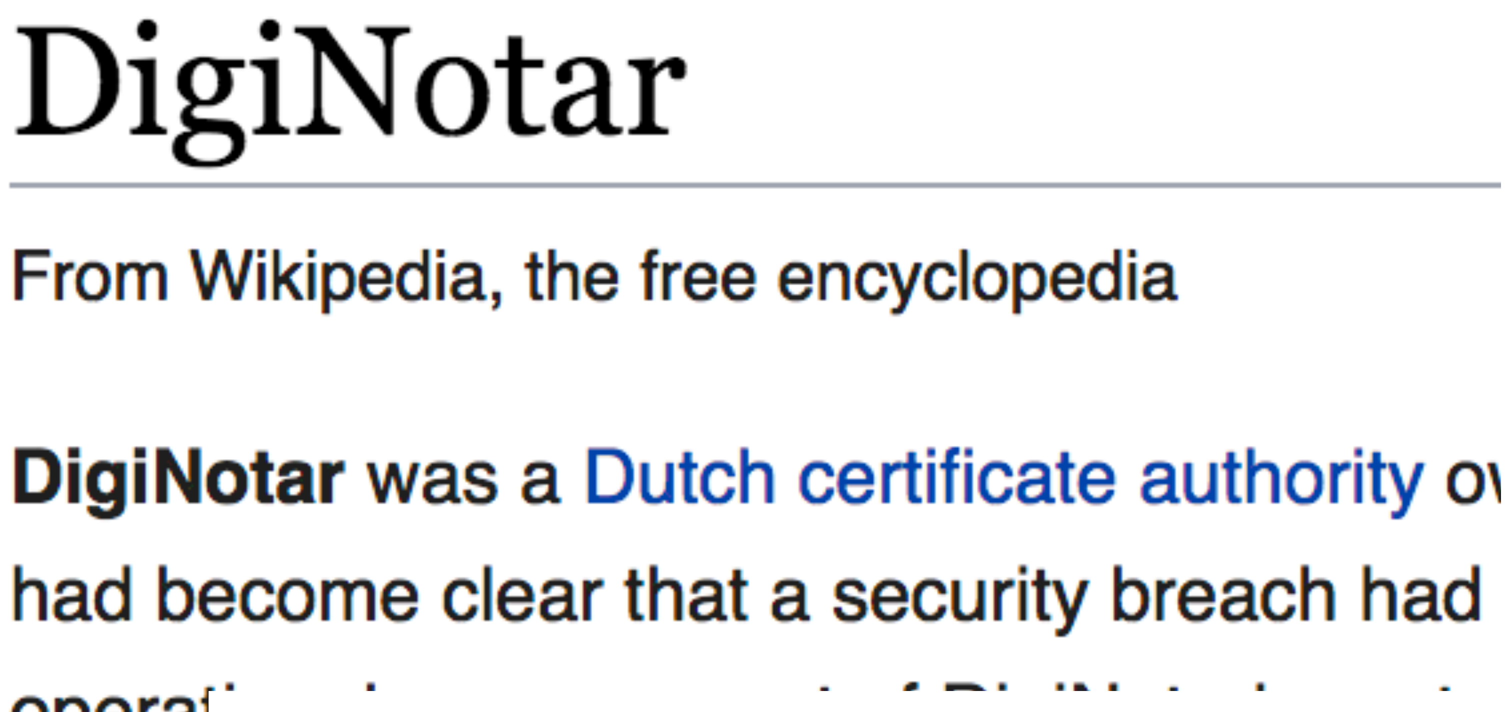
Uses **certificate authority** to provide public key

Uses **asymmetric crypto** to establish symmetric key

Uses **symmetric crypto** for data encryption

Which Authority Should You Trust?

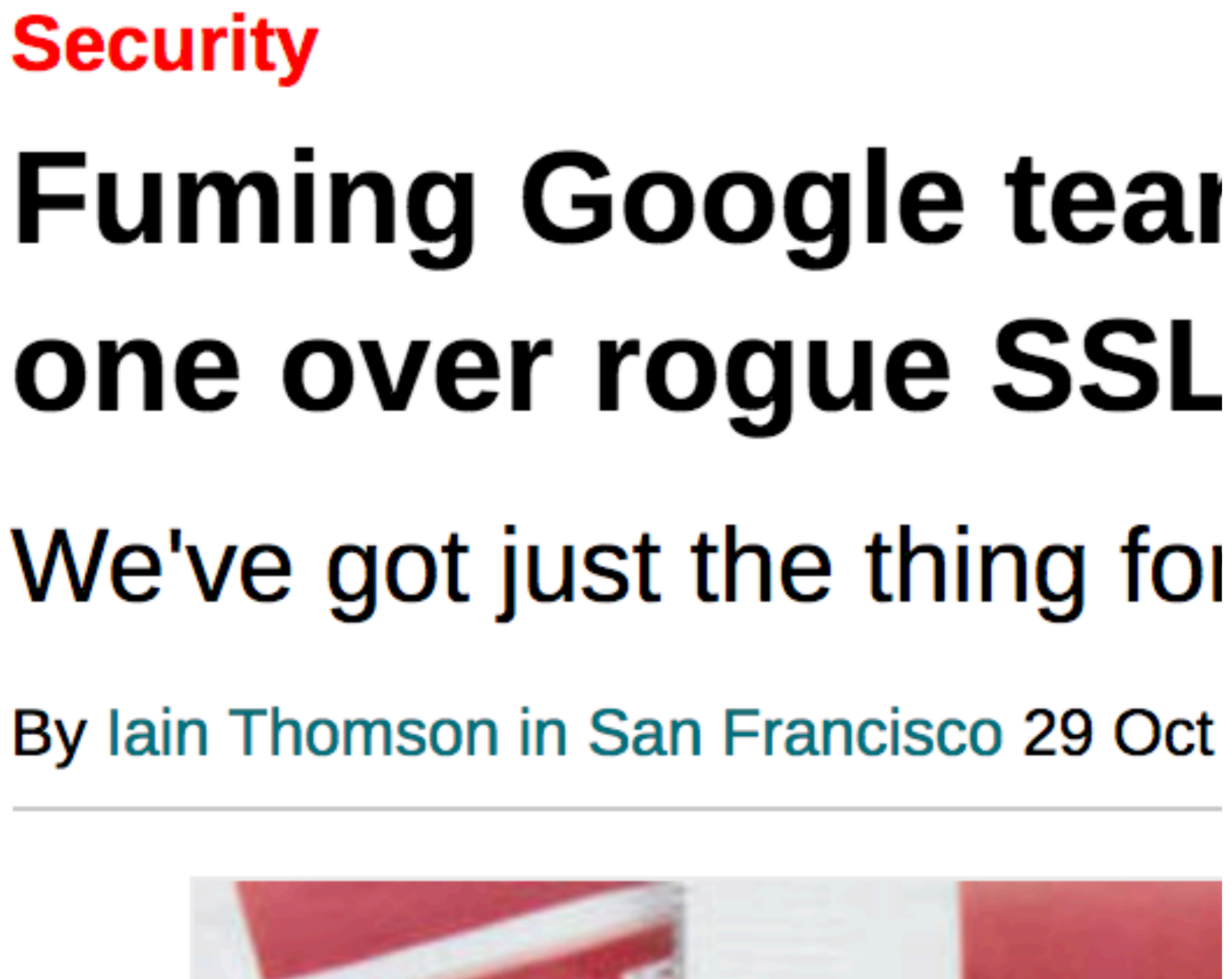
- Today: many autho



DigiNotar

From Wikipedia, the free encyclopedia

DigiNotar was a Dutch certificate authority o
had become clear that a security breach had
operat

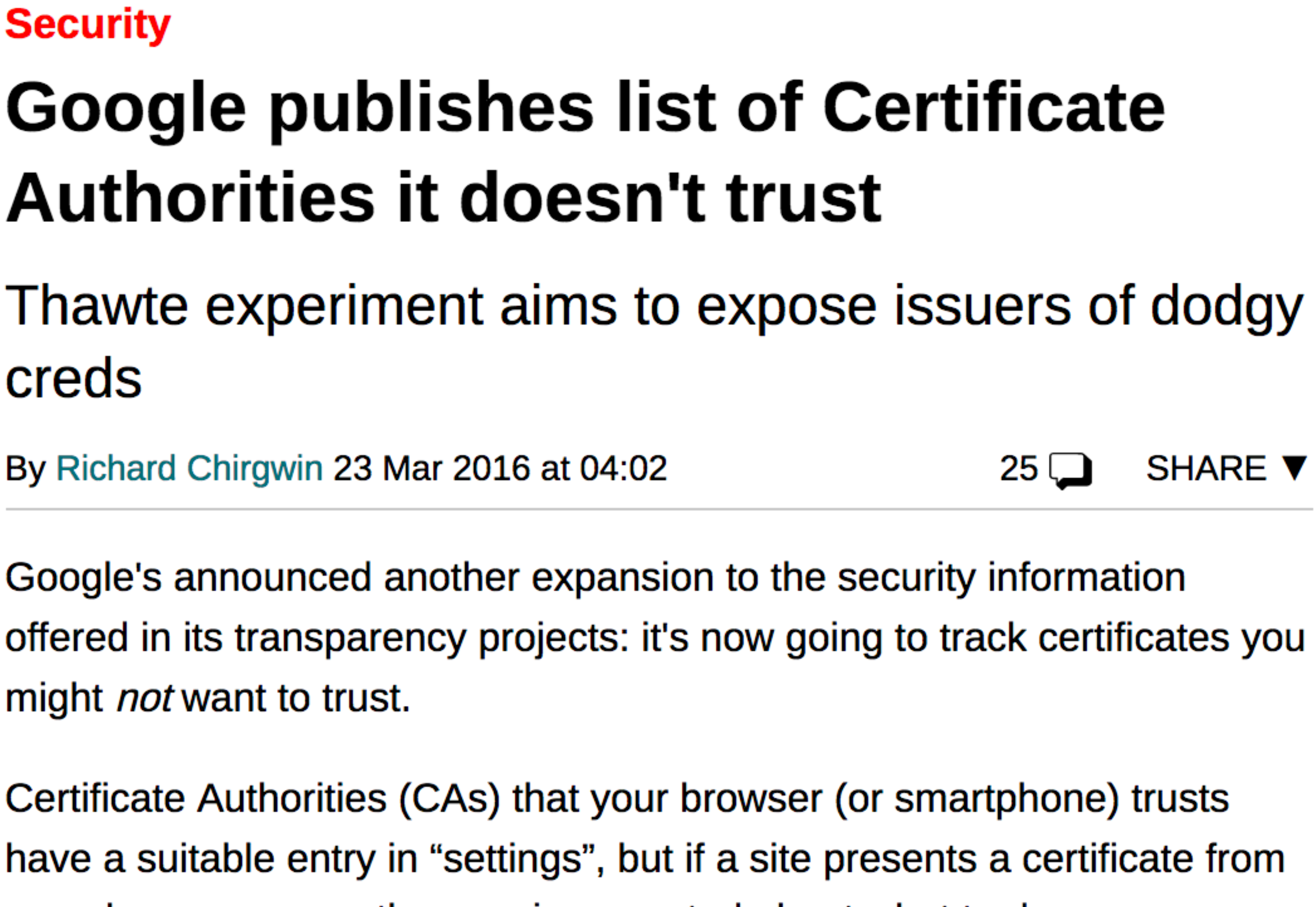


Security

Fuming Google tea
one over rogue SSL

We've got just the thing fo

By Iain Thomson in San Francisco 29 Oct



Security

Google publishes list of Certificate Authorities it doesn't trust

Thawte experiment aims to expose issuers of dodgy creds

By Richard Chirgwin 23 Mar 2016 at 04:02

Google's announced another expansion to the security information offered in its transparency projects: it's now going to track certificates you might *not* want to trust.

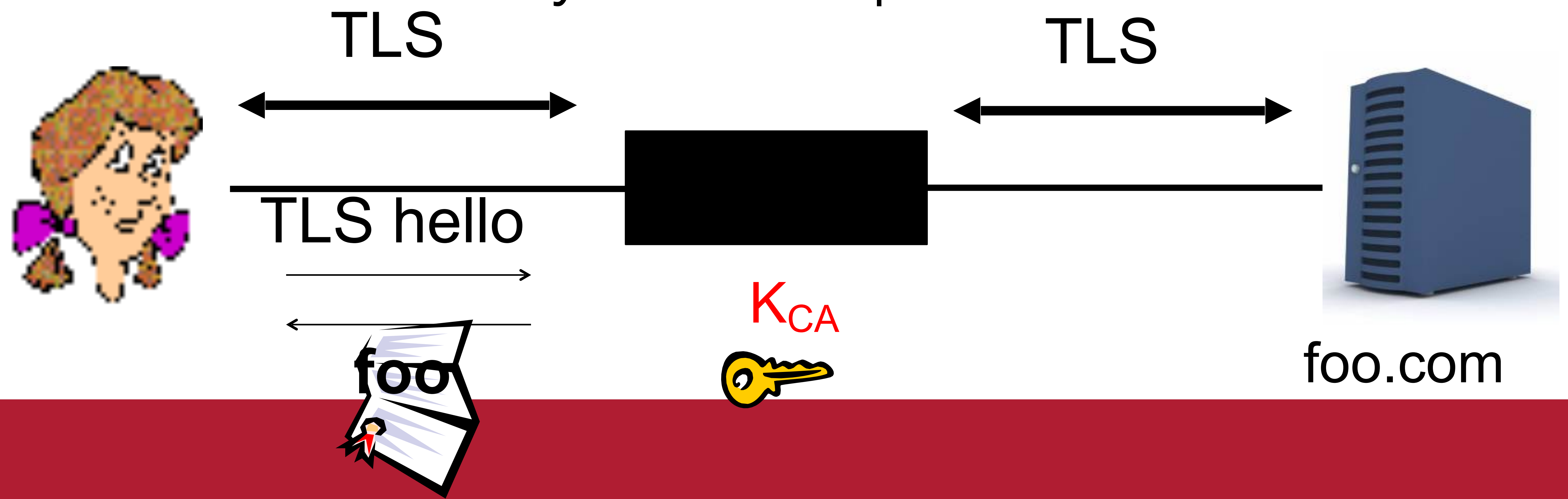
Certificate Authorities (CAs) that your browser (or smartphone) trusts have a suitable entry in “settings”, but if a site presents a certificate from

Which Authority Should You Trust?

- If the browser detects a problem with a certificate, it asks user what to do
  - Invalid, expired, self-signed, …
- Users often blindly click “yes”
  - They don’t know about certificates or TLS; don’t understand implications of a bad certificates
- Certificates are hard to read and can be misleading
  - Most information makes no sense to user
  - Names can be confusing, e.g., minor variants

Middleboxes + TLS :(

- Middleboxes are very widely used in the Internet
  - Companies have firewalls
  - Cellular operators use caches, compression, …
- But TLS makes middleboxes ineffective
- “Solution”: install fake root certificate on device
  - Common for corporate networks
  - Sometimes also done by service providers



BONUS CONFIDENTIALITY TIME

Does TLS keep who you are talking to confidential?

TLS gives confidentiality, but not anonymity.

Anonymity is confidentiality for *who is talking,* not just *what they are saying.*

What is Anonymity?

- Anonymity is the state of being not identifiable within a set of subjects
  - You cannot be anonymous by yourself!
  - Hide your activities among others' similar activities
- Unlinkability of action and identity
  - For example, sender and his email are no more related after observing communication than they were before
- Unobservability (hard to achieve)
  - Any item of interest (message, event, action) is indistinguishable from any other item of interest

Do we even want anonymity?

Anonymity Activity

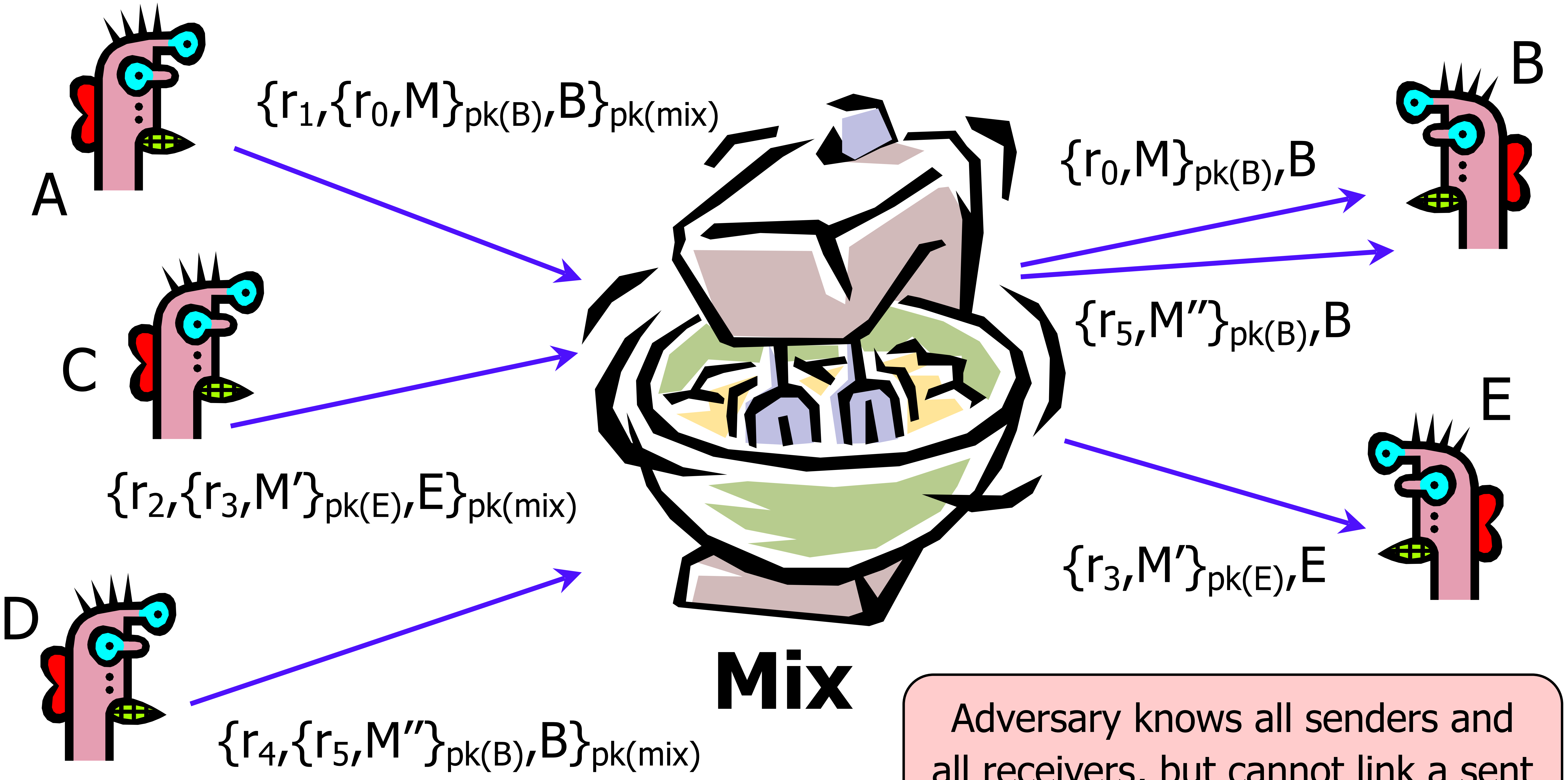
Chaum’s Mix

- Early proposal for anonymous email
  - David Chaum. “Untraceable electronic mail, return addresses, and digital pseudonyms”. Communications of the ACM, February 1981.

Before spam, people thought anonymous email was a good idea ☺️

- Public key crypto + trusted re-mailer (Mix)
  - Untrusted communication medium
  - Public keys used as persistent pseudonyms
- Modern anonymity systems use Mix as the basic building block

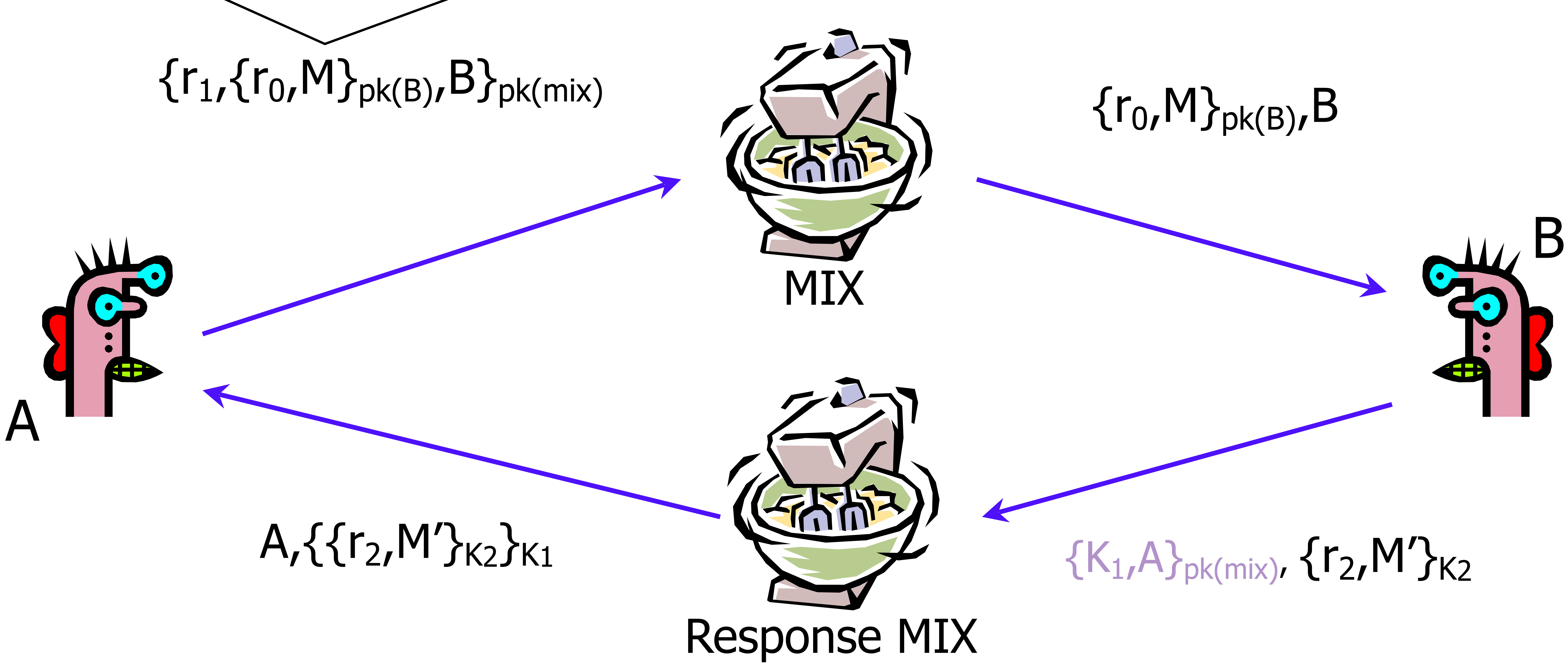
Basic Mix Design


A
$\{r_1,\{r_0,M\}_{pk(B)},B\}_{pk(mix)}$
C
$\{r_2,\{r_3,M'\}_{pk(E)},E\}_{pk(mix)}$
D
$\{r_4,\{r_5,M''\}_{pk(B)},B\}_{pk(mix)}$
Mix
$\{r_0,M\}_{pk(B)},B$
B
$\{r_5,M''\}_{pk(B)},B$
E
$\{r_3,M'\}_{pk(E)},E$

Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Anonymous Return Addresses

M includes $\{K_1,A\}_{pk(mix)}$, K_2 where K_2 is a fresh public key



$\{r_1,\{r_0,M\}_{pk(B)},B\}_{pk(mix)}$

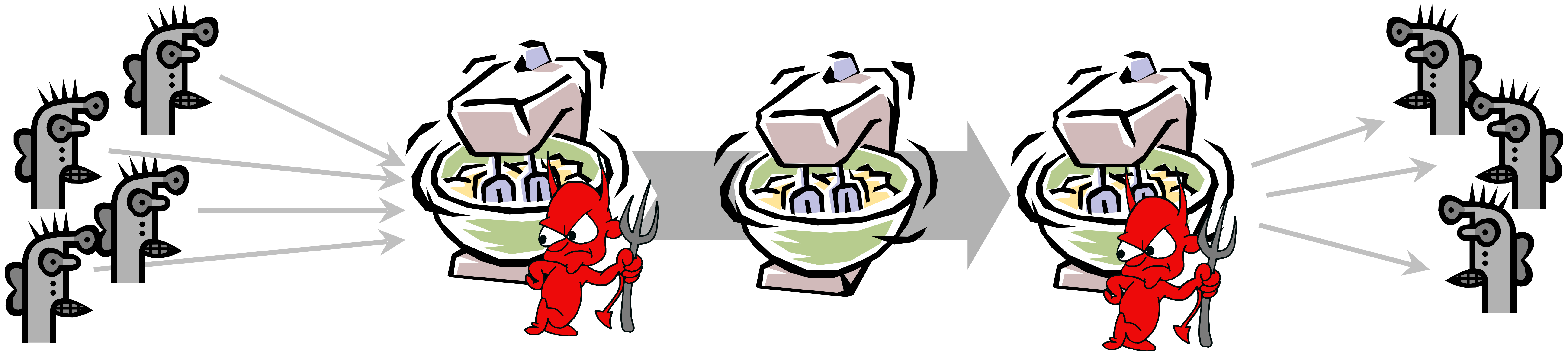
$\{r_0,M\}_{pk(B)},B$

$\{K_1,A\}_{pk(mix)}, \{r_2,M'\}_{K2}$

$A,\{\{r_2,M'\}_{K2}\}_{K1}$

Secrecy without authentication
(good for an online confession service ☺)

Mix Cascade



- Messages are sent through a sequence of mixes
  - Can also form an arbitrary network of mixes (“mixnet”)

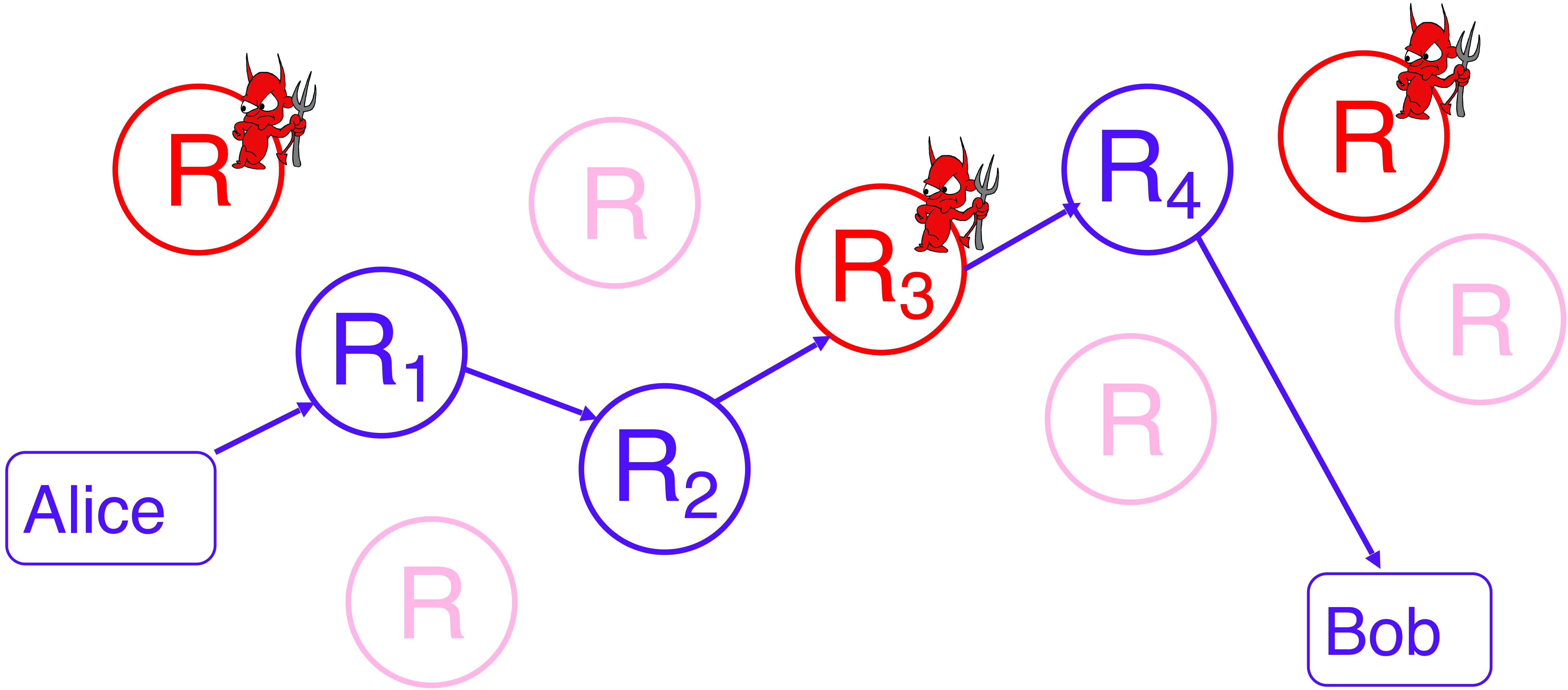
- Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity

- Pad and buffer traffic to foil correlation attacks

Disadvantages of Basic Mixnets

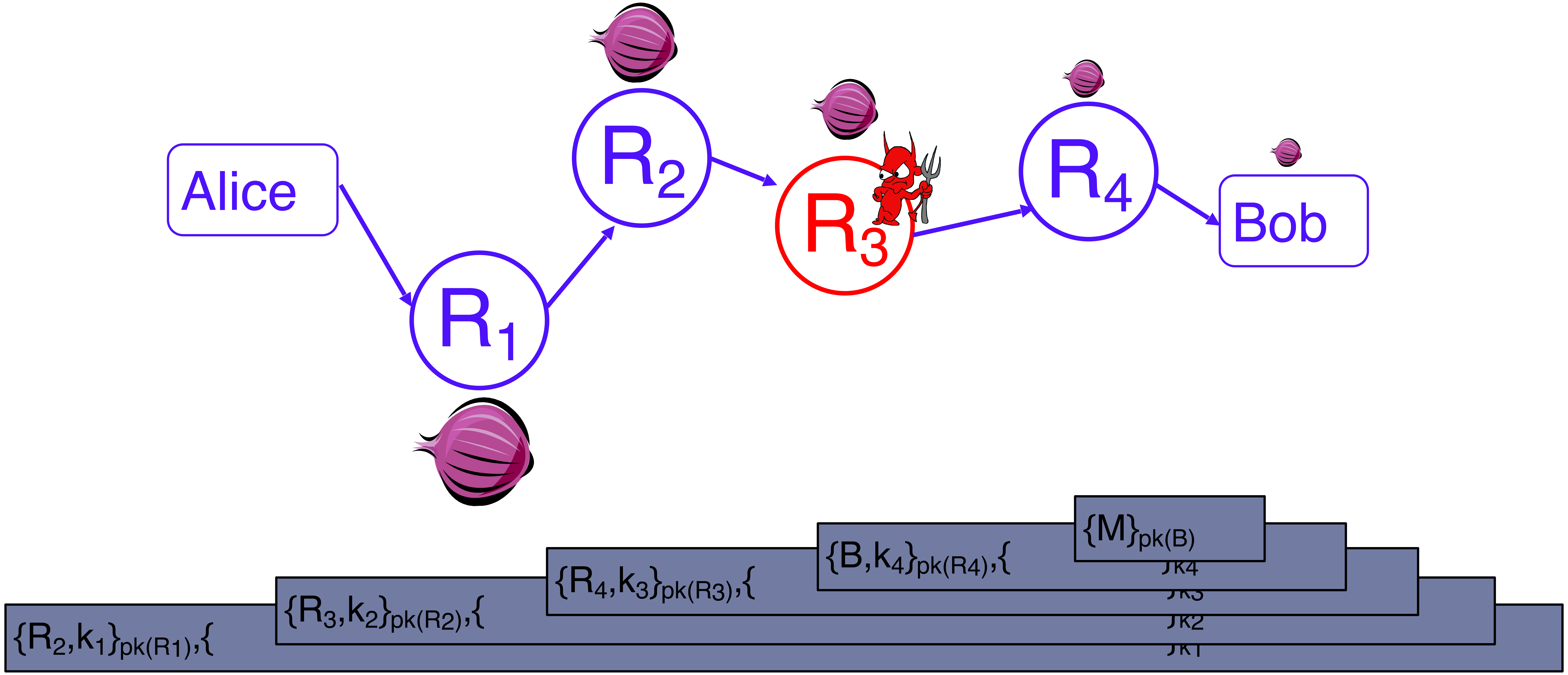
- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - Ok for email, not Ok for anonymous Web browsing
- Challenge: low-latency anonymity network
  - Use public-key cryptography to establish a “circuit” with pairwise symmetric keys between hops on the circuit
  - Then use symmetric decryption and re-encryption to move data messages along the established circuits
  - Each node behaves like a mix; anonymity is preserved even if some nodes are compromised

Onion Routing



- Sender chooses a random sequence of routers
  - Some routers are honest, some controlled by attacker
  - Sender controls the length of the path

Route Establishment



- Routing info for each link encrypted with router’s public key
- Each router learns only the identity of the next router

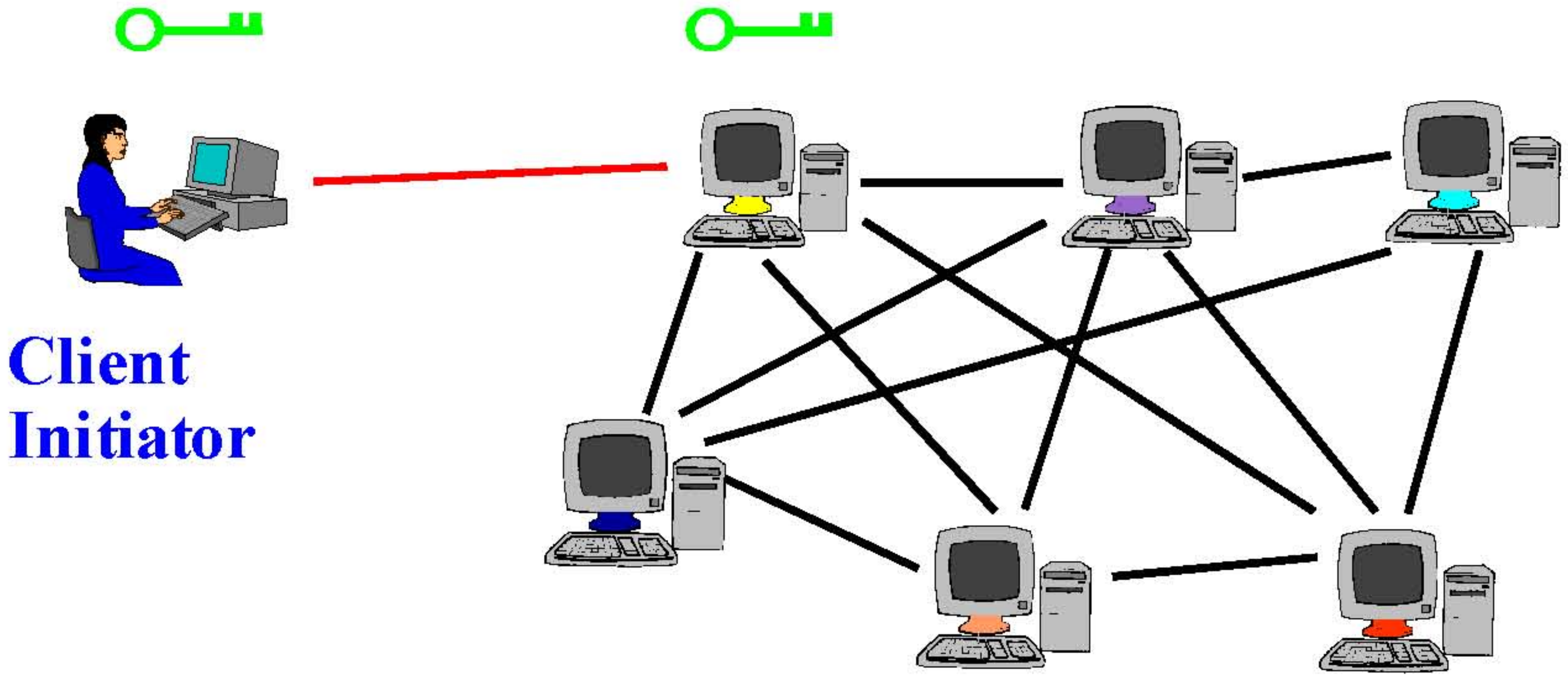
Tor

- Second-generation onion routing network
  - http://tor.eff.org
  - Developed by Roger Dingledine, Nick Mathewson and Paul Syverson
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
- 100 nodes on four continents, thousands of users
- “Easy-to-use” client proxy
  - Freely available, can use it for anonymous browsing

Have any of y’all used Tor before?

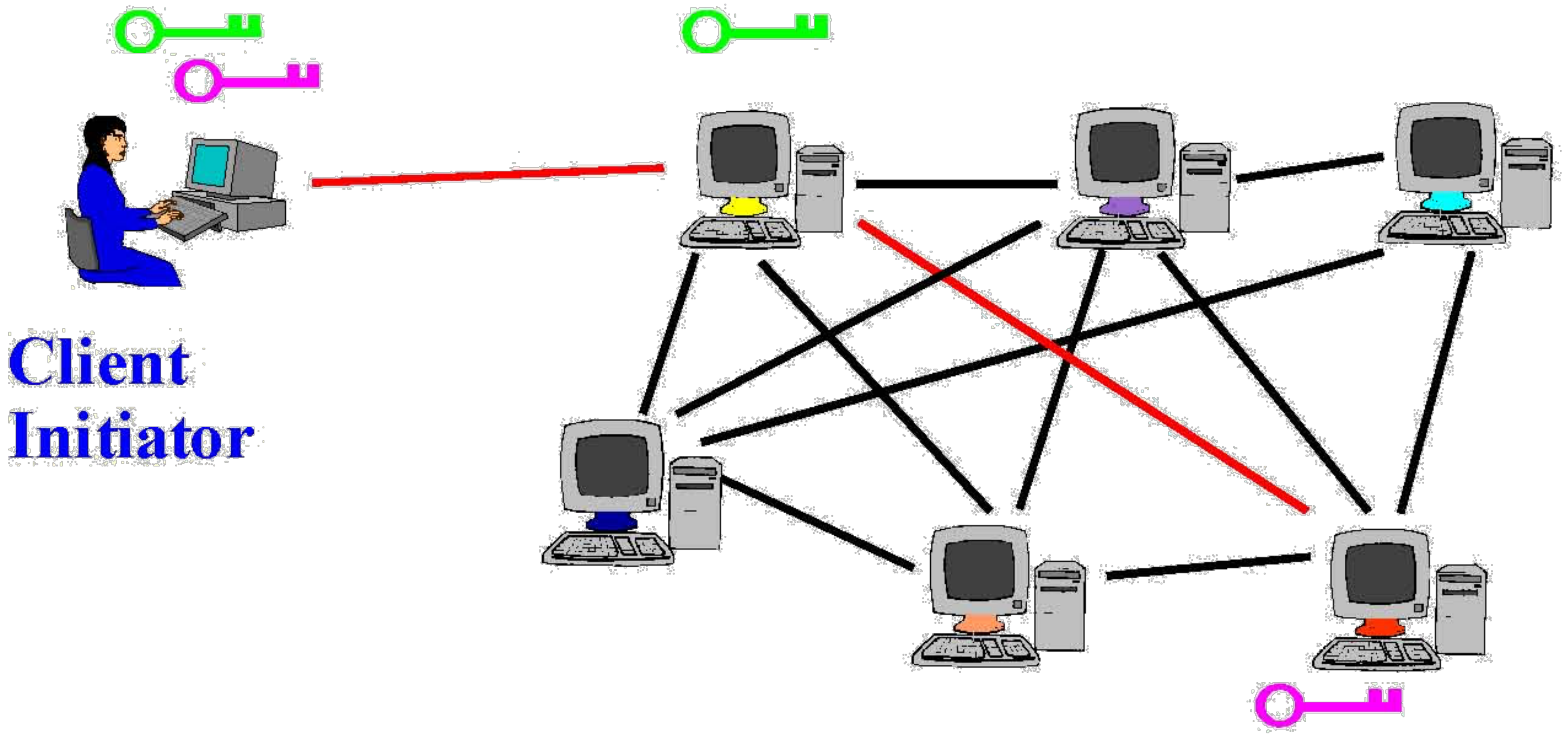
Tor Circuit Setup (1)

- Client proxy establish a symmetric session key and circuit with Onion Router #1



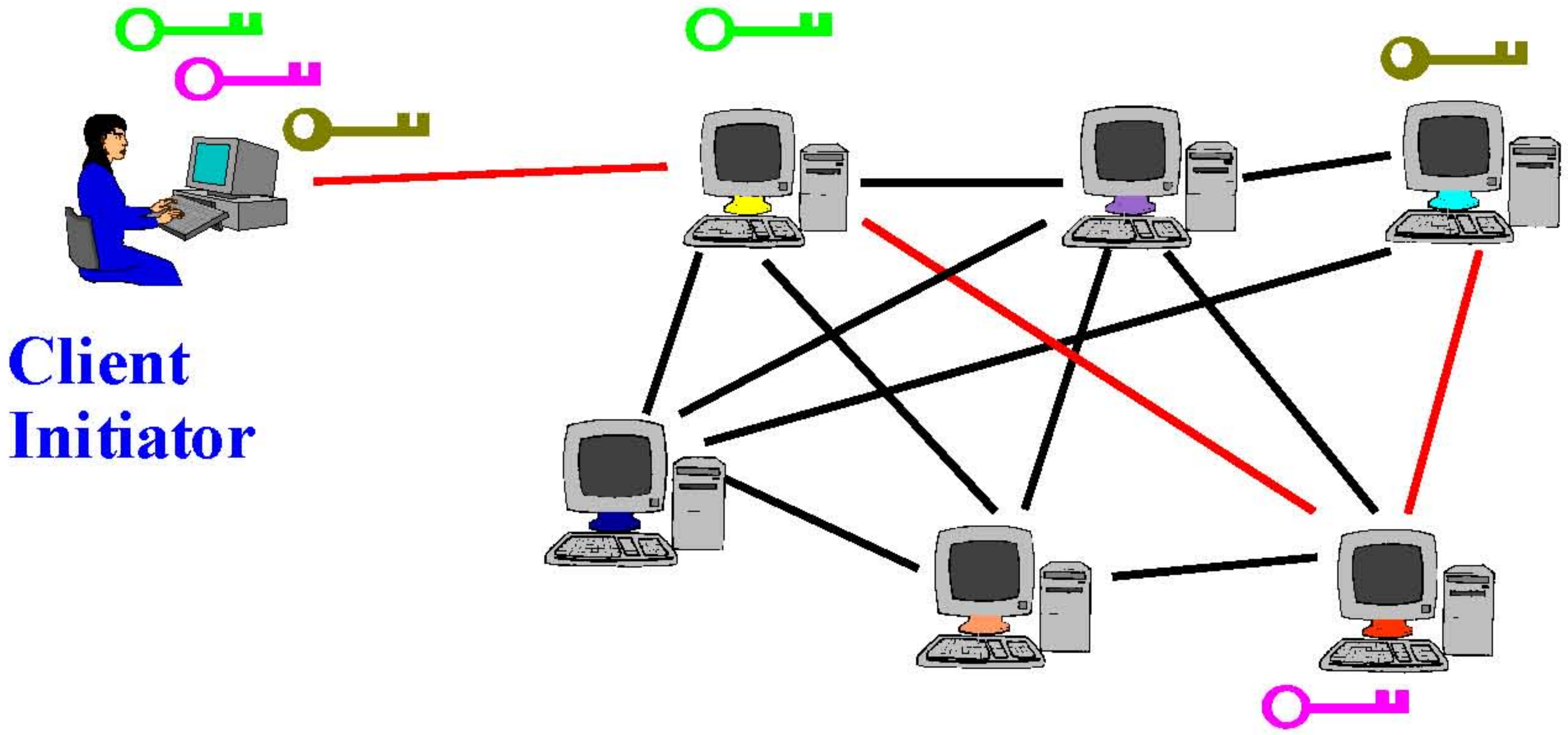
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
  - Tunnel through Onion Router #1



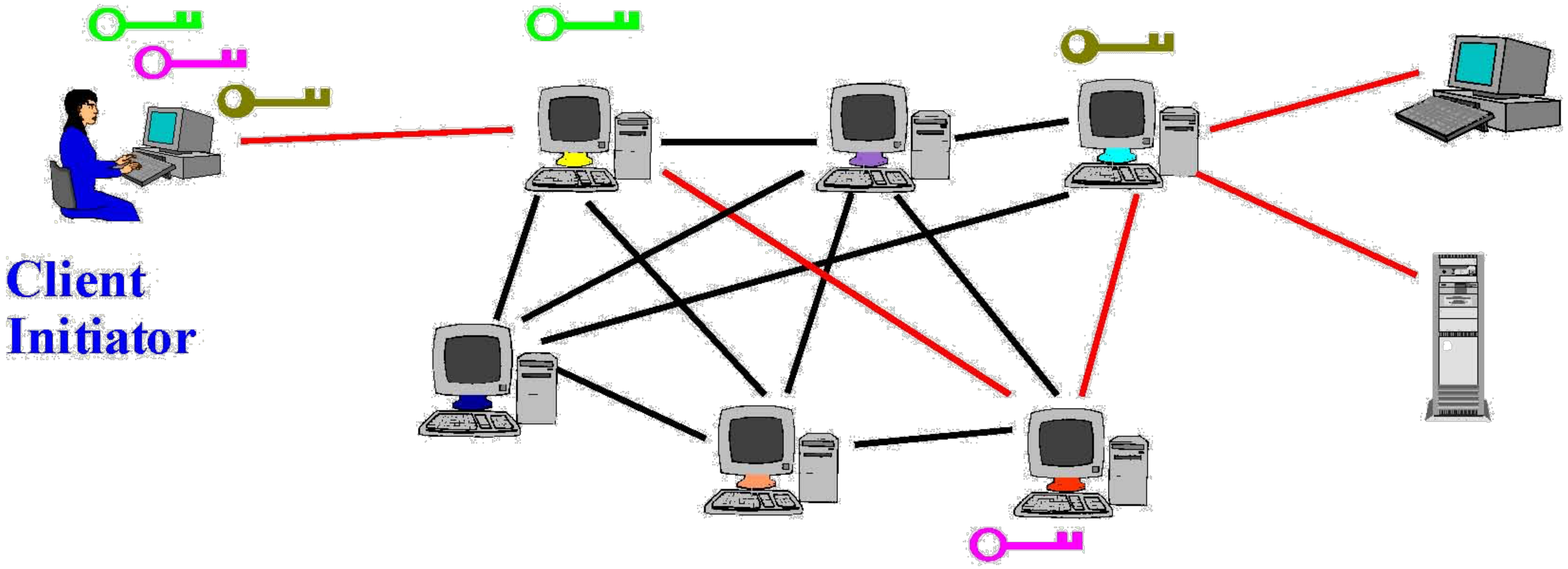
Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
  - Tunnel through Onion Routers #1 and #2



Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit
  - Datagrams are decrypted and re-encrypted at each link



Tor Management Issues

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn’t need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active onion routers, their locations, current public keys, etc.
  - Control how new routers join the network
    - “Sybil attack”: attacker creates a large number of routers
  - Directory servers’ keys ship with Tor code

Summary

- Internet design and growth => security challenges
- Symmetric (pre-shared key, fast) and asymmetric (key pairs, slow) primitives provide:
  - Confidentiality
  - Integrity
  - Authentication
- “Hybrid Encryption” leverages strengths of both.
- Great complexity exists in securely acquiring keys.
- Anonymity remains a great challenge in networking.