

CSE 484 (Winter 2010)

Usability + Anonymity

Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

- ◆ User authentication
- ◆ Anonymity
- ◆ Ethics

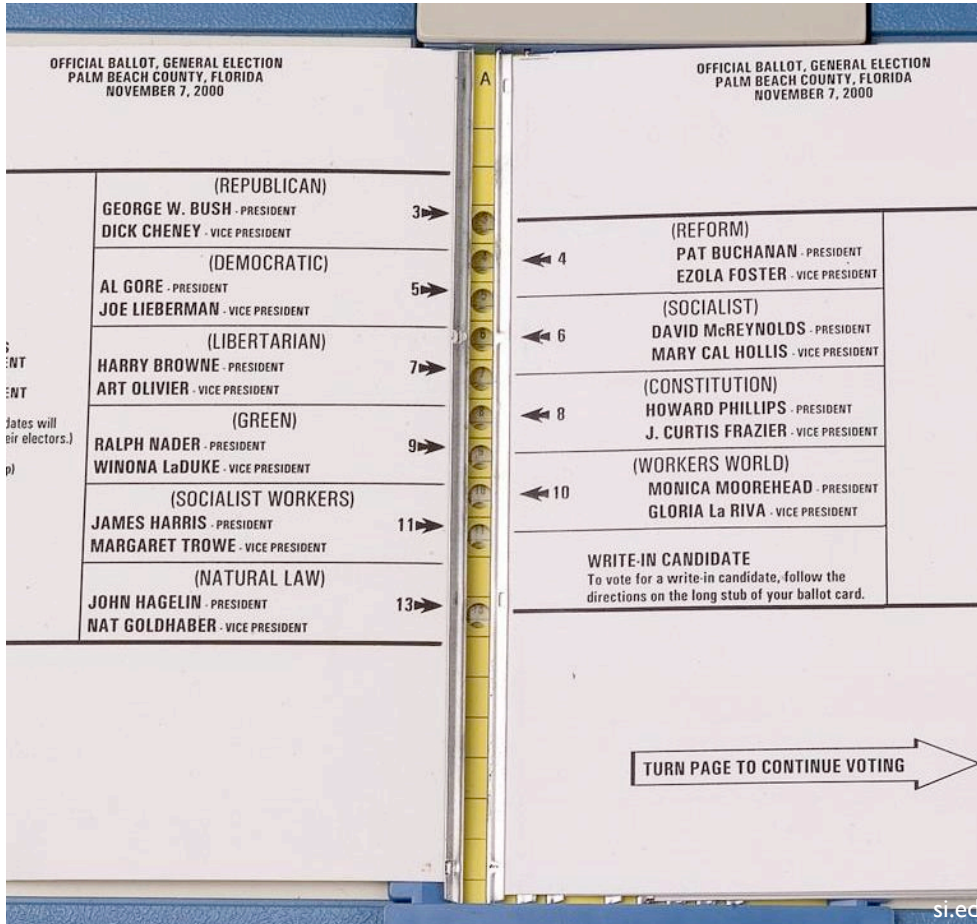
Final

- ◆ Closed book
- ◆ Closed computers
- ◆ No calculators (unless we send an email out by Monday saying otherwise)
- ◆ We'll provide scrap paper if necessary

- ◆ Comparable to last year's final that I emailed out earlier this week.
- ◆ Comparable to homeworks

- ◆ Good luck!

Poor Usability Causes Problems



si.ed

AP

Importance

◆ Why is usability important?

- People are the critical element of any computer system
 - People are the real reason computers exist in the first place
- Even if it is **possible** for a system to protect against an adversary, people may use the system in other, **less secure** ways

◆ Next

- Challenges with security and usability
- Key design principles
- New trends and directions

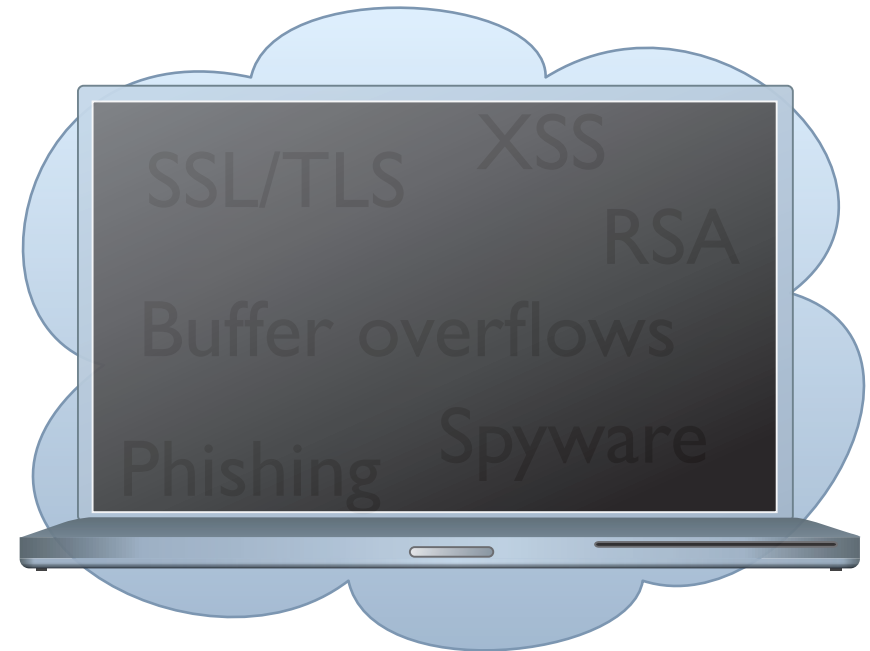
Issue #1: Complexities, Lack of Intuition

Real World



We can see, understand, relate to.

Electronic World



Too complex, hidden, no intuition.

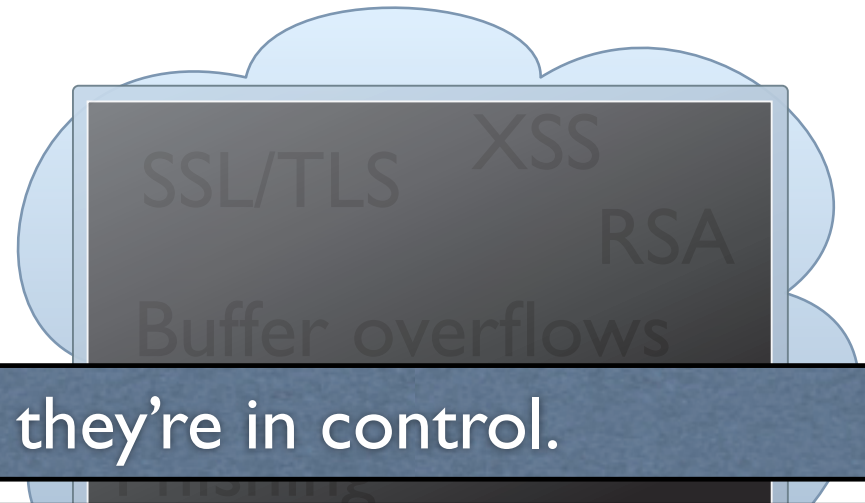
Issue #1: Complexities, Lack of Intuition

- ◆ Mismatch between perception of technology and what really happens
 - Public keys?
 - Signatures?
 - Encryption?
 - Message integrity?
 - Chosen-plaintext attacks?
 - Chosen-ciphertext attacks?
 - Password management?
 - ...

Issue #2: Who's in Charge?

Real World

Electronic World



Users want to feel like they're in control.

Adversaries in the electronic world can be *intelligent, sneaky, and malicious.*

Complex, hidden, but
doctors manage

Complex, hidden, and *users manage*

Issue #2: Who's in Charge?

- ◆ Systems developers should help protect users
 - Usable authentication systems
 - Red/green lights
- ◆ Software applications help users manage their applications
 - P3P for privacy control
 - PwdHash, Keychain for password management
 - Some say: Can we trust software for these tasks?

Issue #3: Hard to Gage Risks

“It won’t happen to me!” (Sometimes a reasonable assumption, sometimes not.)

Schneier on Security

A weblog covering security and security technology.

« [The Emergence of a Global Infrastructure for Mass Registration and Surveillance](#) | [Main](#) | [PDF Redacting Failure](#) »

May 02, 2005

Users Disabling Security

It's an old story: users disable a security measure because it's annoying, allowing an attacker to bypass the measure.

A [REDACTED] accused in a deadly courthouse rampage was able to enter the chambers of the judge slain in the attack and hold the occupants hostage because the door was unlocked and a buzzer entry system was not activated, a sheriff's report says.

Security doesn't work unless the users want it to work. This is true on the personal and national scale, with or without technology.

Street Journal, Jan 29, 2007)

Issue #4: No Accountability

- ◆ Issue #3 is amplified when users are not held accountable for their actions
 - E.g., from employers, service providers, etc.
 - (Not all parties will perceive risks the same way)

Issue #5: Awkward, Annoying, or Difficult

◆ Difficult

- Remembering 50 different, “random” passwords

◆ Awkward

- Lock computer screen every time leave the room

◆ Annoying

- Browser warnings, virus alerts, forgotten passwords, firewalls

◆ Consequence:

- Changing user’s knowledge may **not** affect their behavior

Issue #6: Social Issues

- ◆ Public opinion, self-image
 - Only “nerds” or the “super paranoid” follow security guidelines
- ◆ Unfriendly
 - Locking computers suggests distrust of co-workers
- ◆ Annoying
 - Sending encrypted emails that say, “what would you like for lunch?”

Issue #7: Usability Promotes Trust

- ◆ Well known by con artists, medicine men
- ◆ Phishing
 - More likely to trust professional-looking websites than non-professional-looking ones

Response #1: Education and Training

◆ Education:

- Teaching technical concepts, risks

◆ Training

- Change behavior through
 - Drill
 - Monitoring
 - Feedback
 - Reinforcement
 - Punishment

◆ May be part of the solution - but not the solution

Response #2: Security Should Be Invisible

- ◆ Security should happen
 - Naturally
 - By Default
 - Without user input or understanding
- ◆ Recognize and stop bad actions
- ◆ Starting to see some invisibility
 - SSL/TLS
 - VPNs
 - Automatic Security Updates

Response #2: Security Should Be Invisible

- ◆ “Easy” at extremes, or for simple examples
 - Don’t give everyone access to everything
- ◆ But hard to generalize
- ◆ Leads to things not working for reasons user doesn’t understand
- ◆ Users will then try to get the system to work, possibly further reducing security
 - E.g., “dangerous successes” for password managers

Response #3: “Three-word UI:” “Are You Sure?”

- ◆ Security should be invisible
 - Except when the user tries something dangerous
 - In which case a warning is given
- ◆ But how do users evaluate the warning? Two realistic cases:
 - Always heed warning. But see problems / commonality with Response #2
 - Always ignore the warning. If so, then how can it be effective?

Response #4: Use Metaphors, Focus on Users

- ◆ Clear, understandable metaphors:
 - Physical analogs; e.g., red-green lights
- ◆ User-centered design: **Start with user model**
- ◆ Unified security model across applications
 - User doesn't need to learn many models, one for each application
- ◆ Meaningful, intuitive user input
 - Don't assume things on user's behalf
 - Figure out how to ask so that user can answer intelligently

Response #5: Least Resistance

- ◆ “Match the most comfortable way to do tasks with the least granting of authority”
 - Ka-Ping Yee, [Security and Usability](#)
- ◆ Should be “easy” to comply with security policy
- ◆ “Users value and want security and privacy, but they regard them only as secondary to completing the primary tasks”
 - Karat et al, [Security and Usability](#)

Anonymity

Privacy on Public Networks

- ◆ Internet is designed as a public network
 - Machines on your LAN may see your traffic, network routers see all traffic that passes through them
- ◆ Routing information is public
 - IP packet headers identify source and destination
 - Even a passive observer can easily figure out **who is talking to whom**
- ◆ Encryption does not hide identities
 - Encryption hides payload, but not routing information
 - Even IP-level encryption (tunnel-mode IPSec/ESP) reveals IP addresses of IPSec gateways

Applications of Anonymity

◆ Privacy

- Hide online transactions, Web browsing, etc. from intrusive governments, marketers and archivists

◆ Untraceable electronic mail

- Corporate whistle-blowers
- Political dissidents
- Socially sensitive communications (online AA meeting)
- Confidential business negotiations

◆ Law enforcement and intelligence

- Sting operations and honeypots
- Secret communications on a public network

What is Anonymity?

- ◆ Anonymity is the state of being not identifiable within a **set of subjects**
 - You cannot be anonymous by yourself!
 - Big difference between anonymity and confidentiality
 - Hide your activities among others' similar activities
- ◆ Unlinkability of action and identity
 - For example, sender and the email he or she sends are no more related after observing communication than they were before
- ◆ Unobservability (hard to achieve)

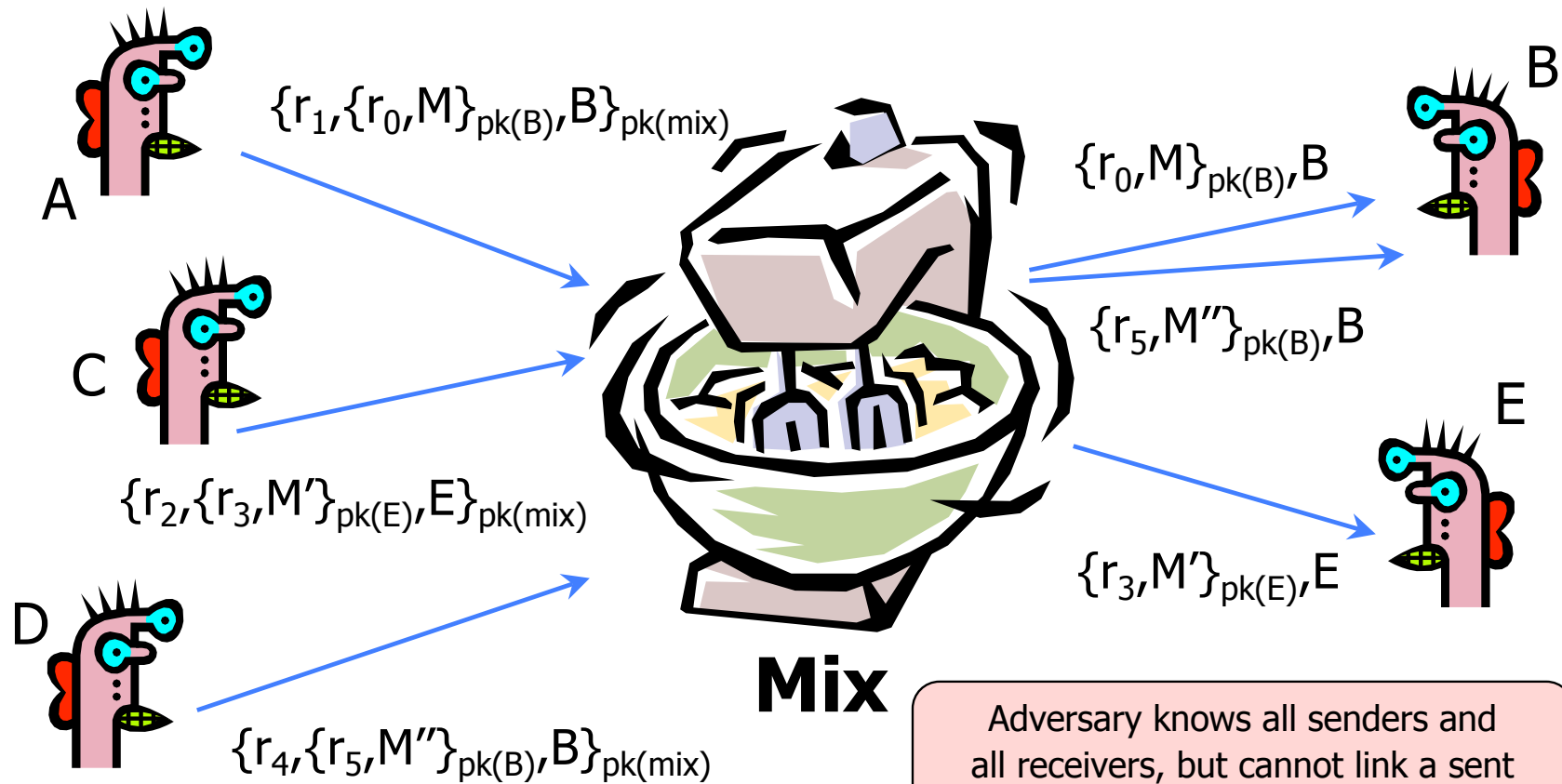
Chaum's Mix

- ◆ 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

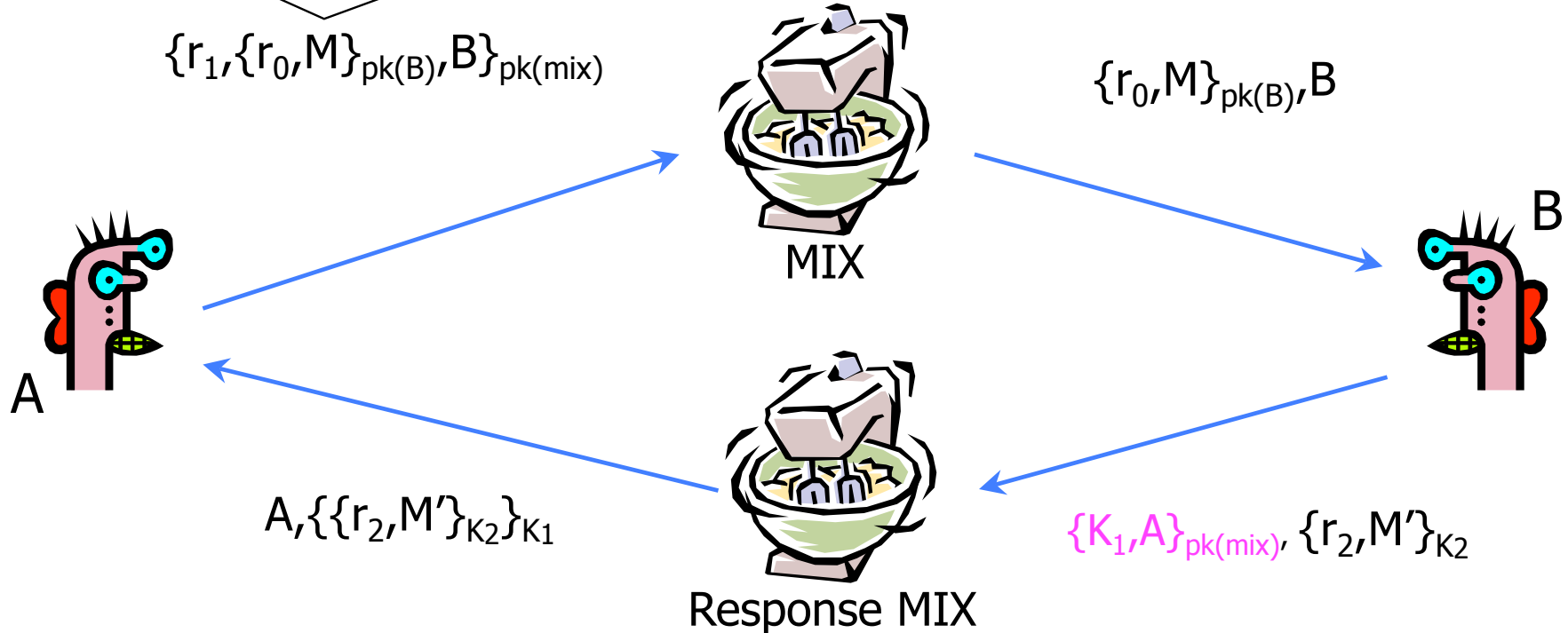
Basic Mix Design



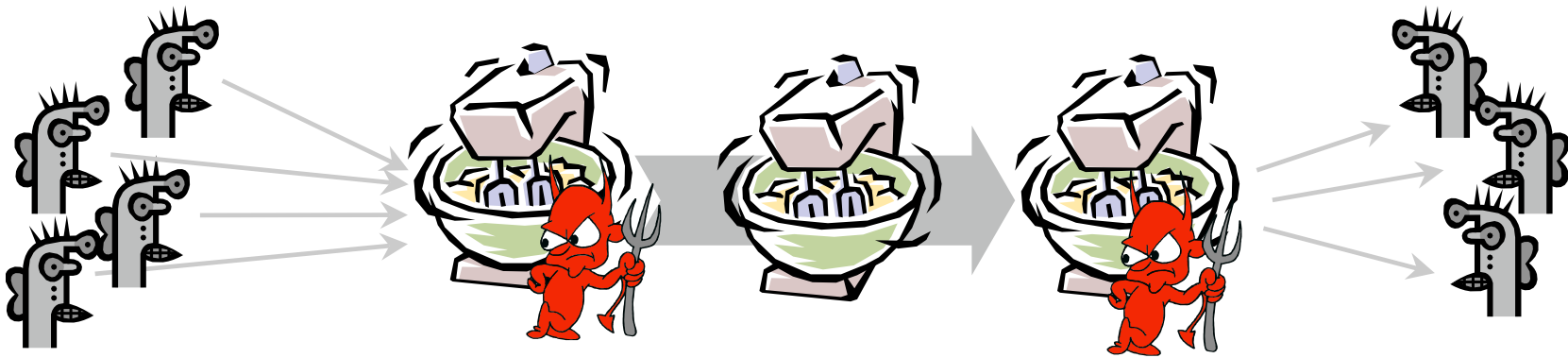
Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Anonymous Return Addresses

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



Mix Cascade

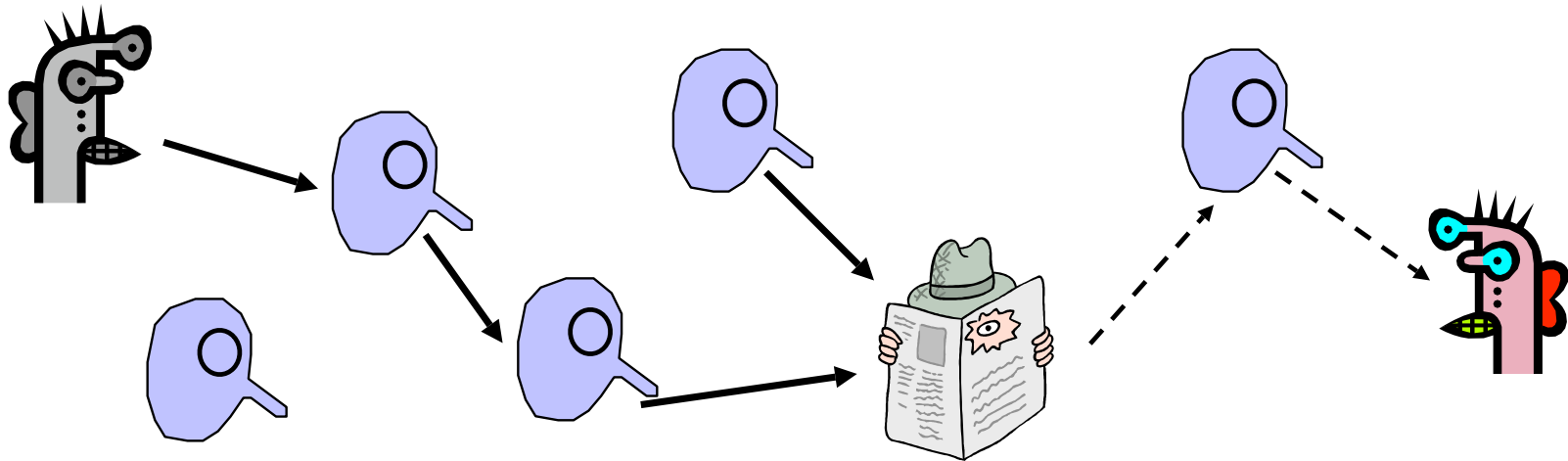


- ◆ Messages are sent through a **sequence of mixes**
 - Can also form an arbitrary network of mixes ("mixnet")
- ◆ Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity
- ◆ Pad and buffer traffic to foil correlation attacks

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

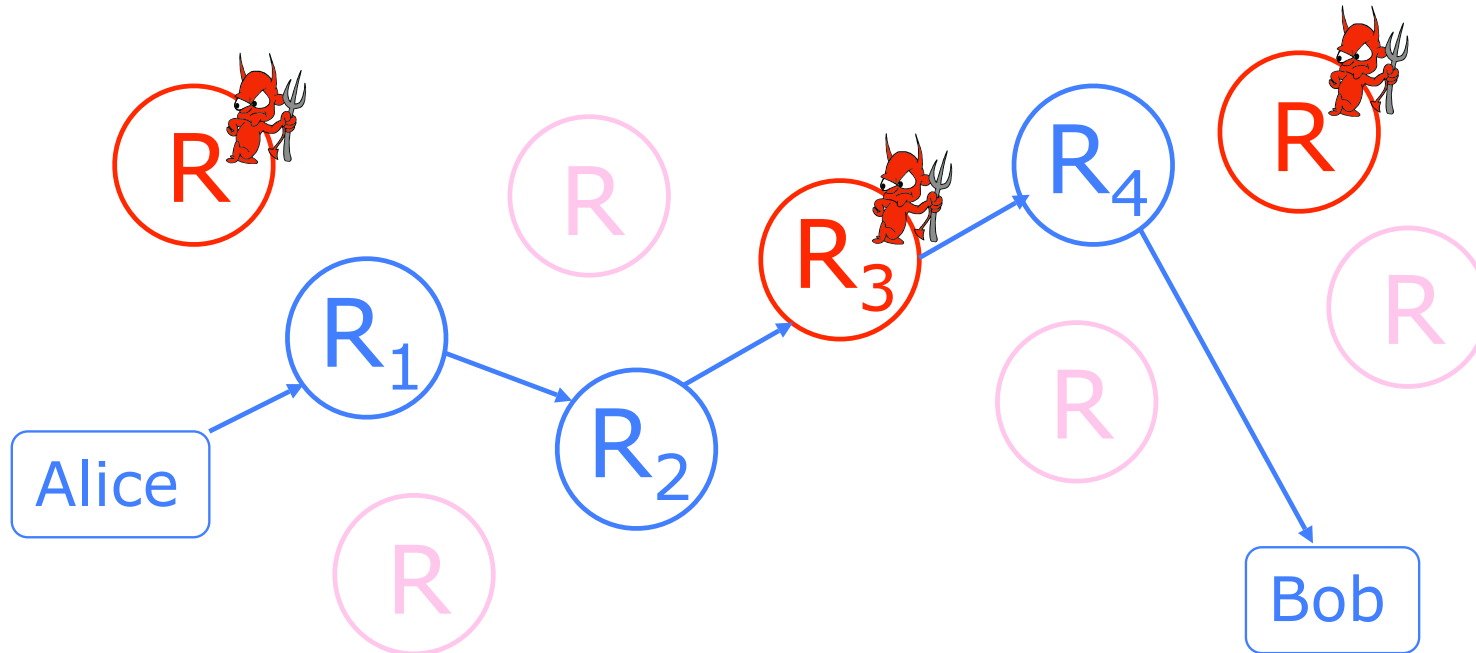
Another Idea: Randomized Routing



- ◆ Hide message source by routing it randomly
 - Popular technique: Crowds, Freenet, Onion routing
- ◆ Routers don't know for sure if the apparent source of a message is the true sender or another router

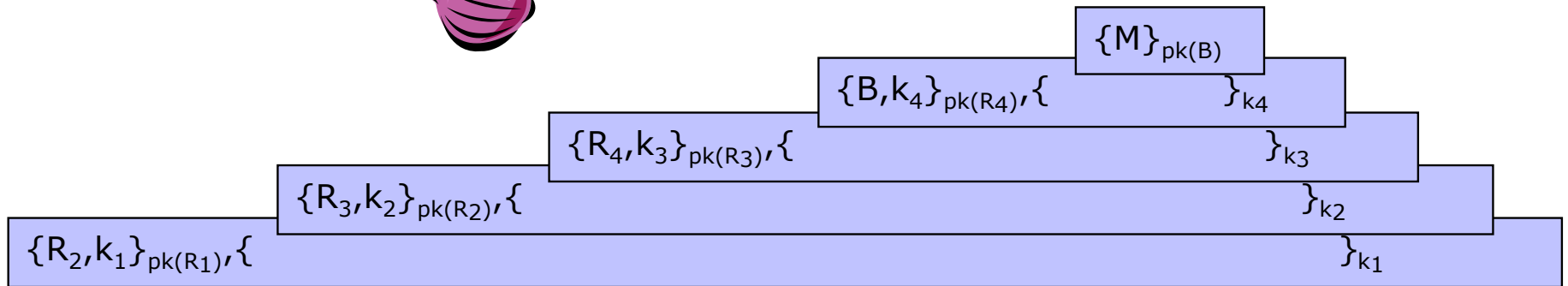
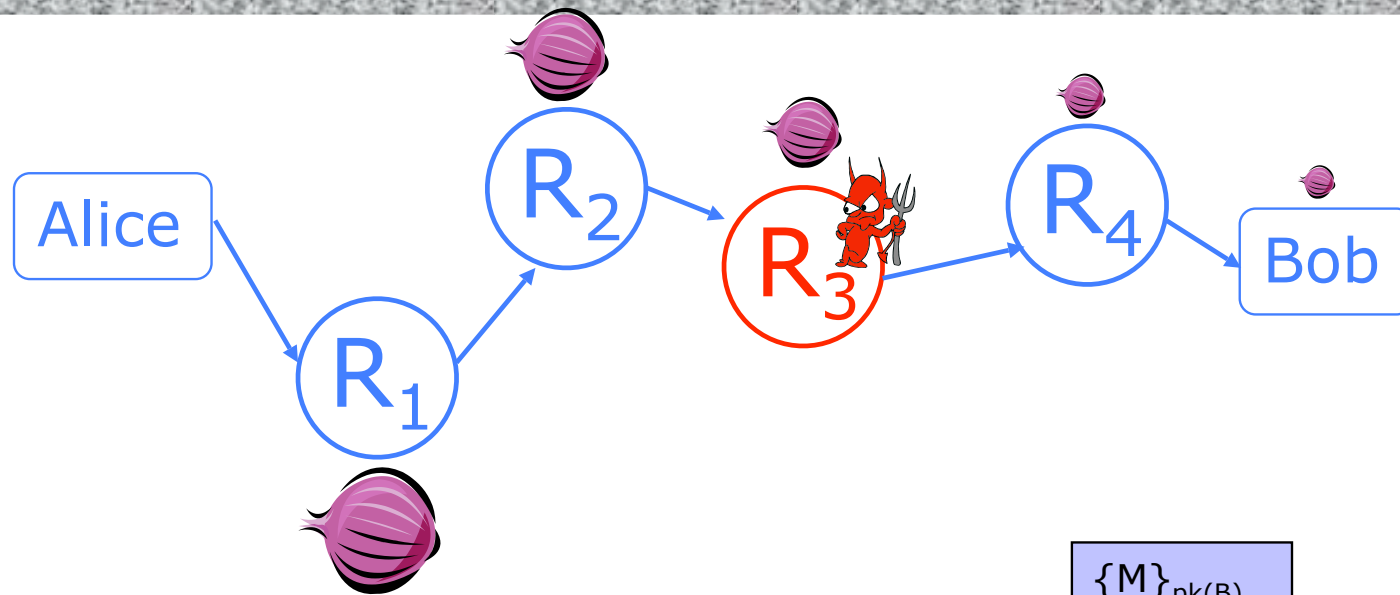
Onion Routing

[Reed, Syverson, Goldschlag '97]



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

Route Establishment



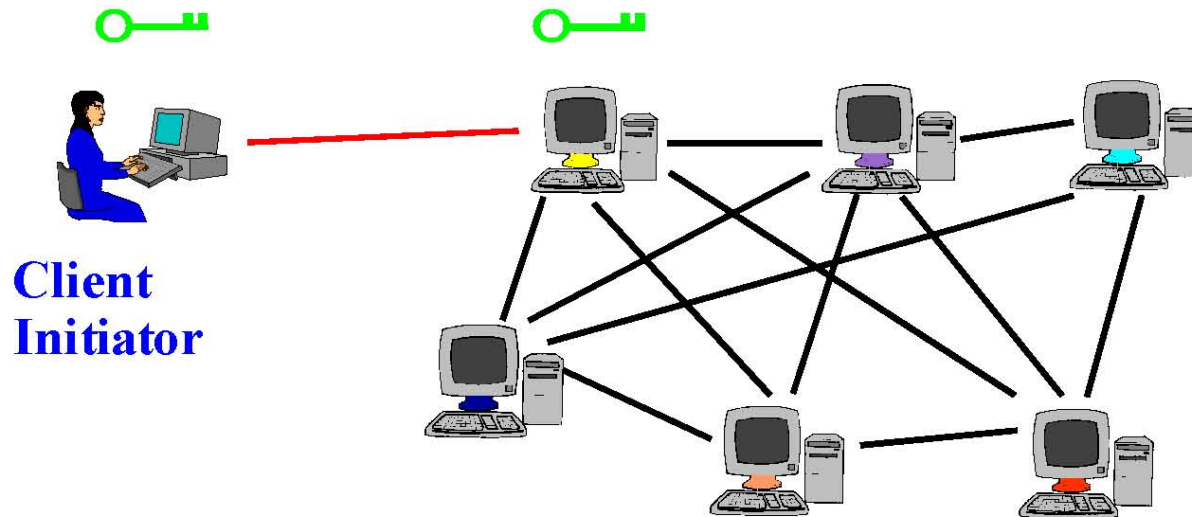
- Routing info for each link encrypted with router's public key
- Each router learns only the identity of the next router

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
- ◆ “Easy-to-use” client proxy
 - Freely available, can use it for anonymous browsing

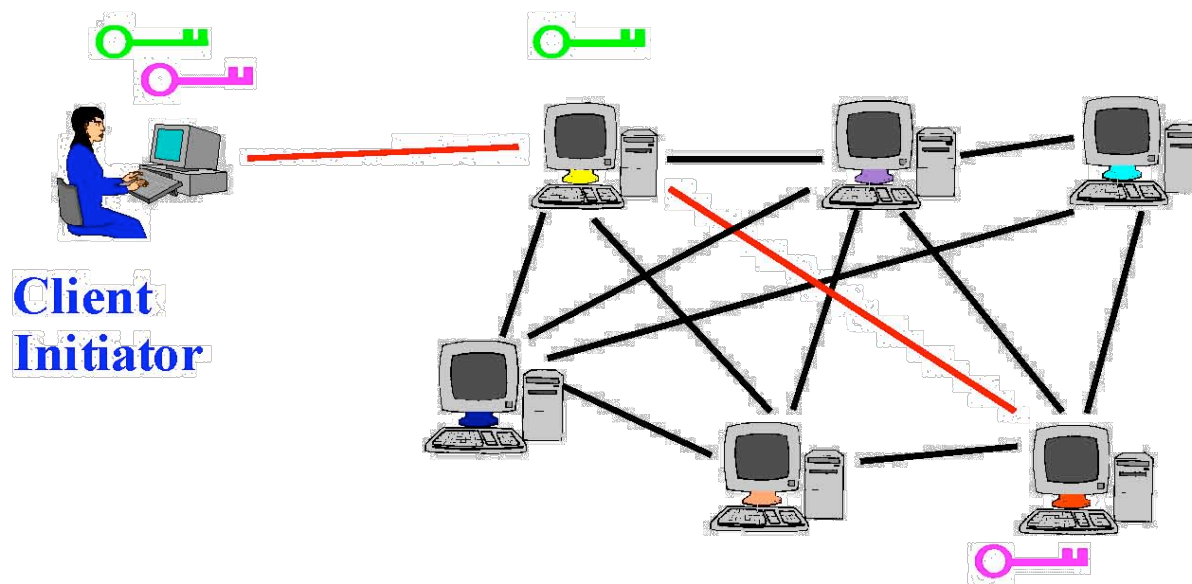
Tor Circuit Setup (1)

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



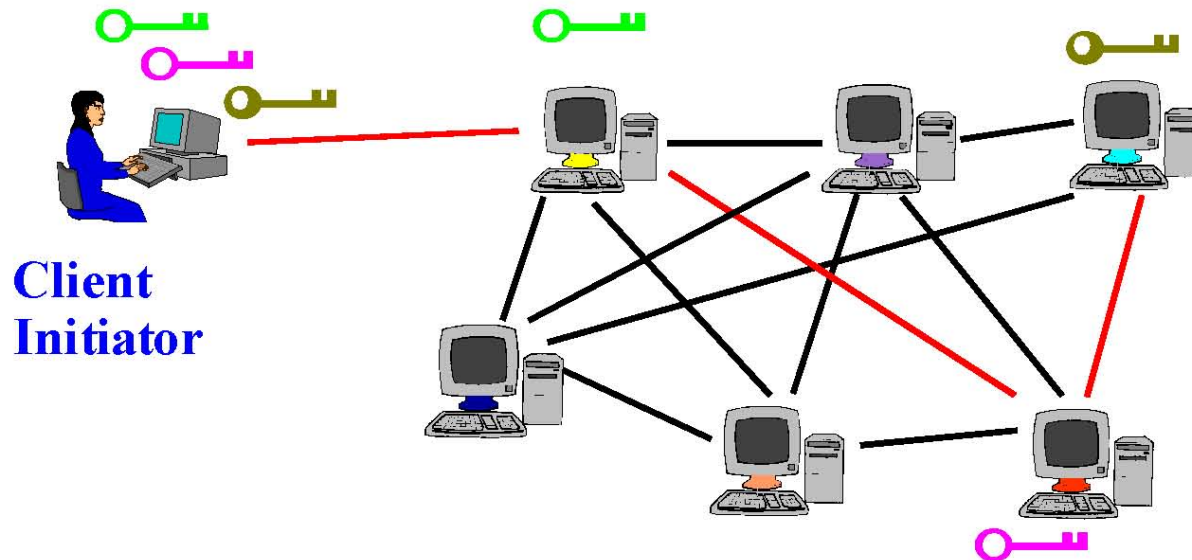
Tor Circuit Setup (2)

- ◆ Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
 - Tunnel through Onion Router #1 (don't need )



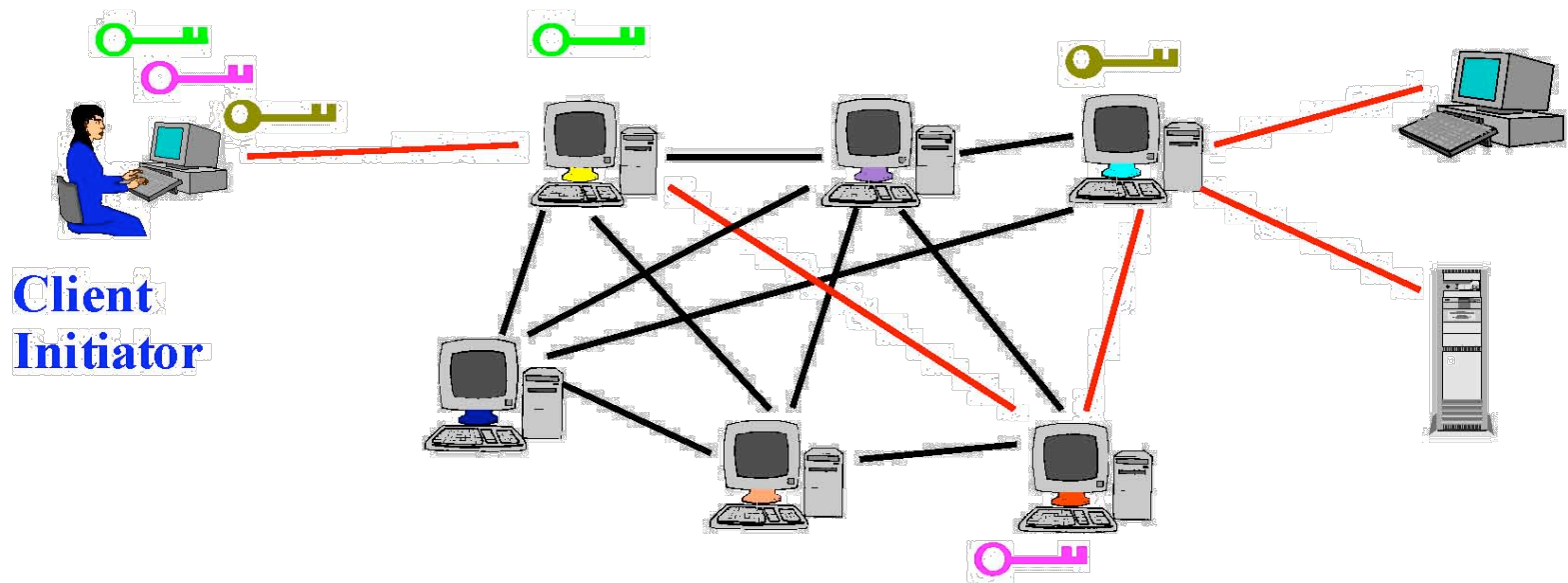
Tor Circuit Setup (3)

- ◆ 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



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

Attacks on Anonymity

◆ Passive traffic analysis

- Infer from network traffic who is talking to whom
- To hide your traffic, must carry other people's traffic!

◆ Active traffic analysis

- Inject packets or put a timing signature on packet flow

◆ Compromise of network nodes

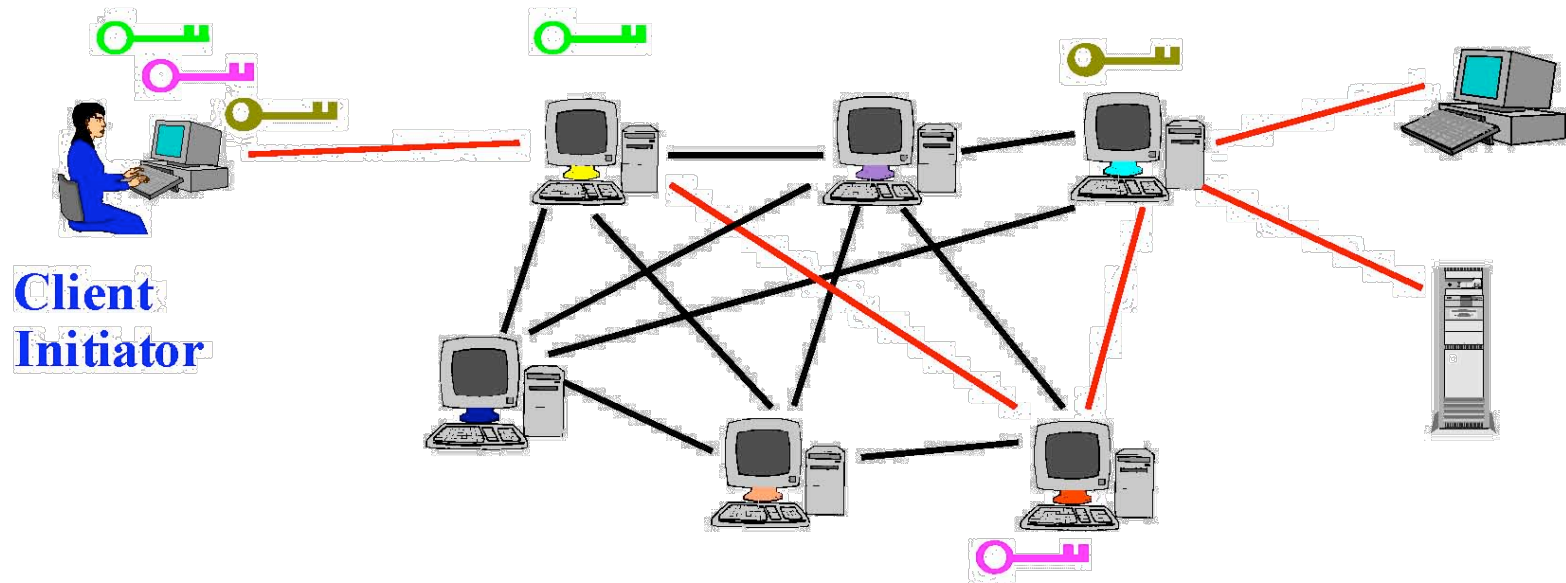
- Attacker may compromise some routers
- It is not obvious which nodes have been compromised
 - Attacker may be passively logging traffic
- Better not to trust any individual router
 - Assume that some fraction of routers is good, don't know which

Deployed Anonymity Systems

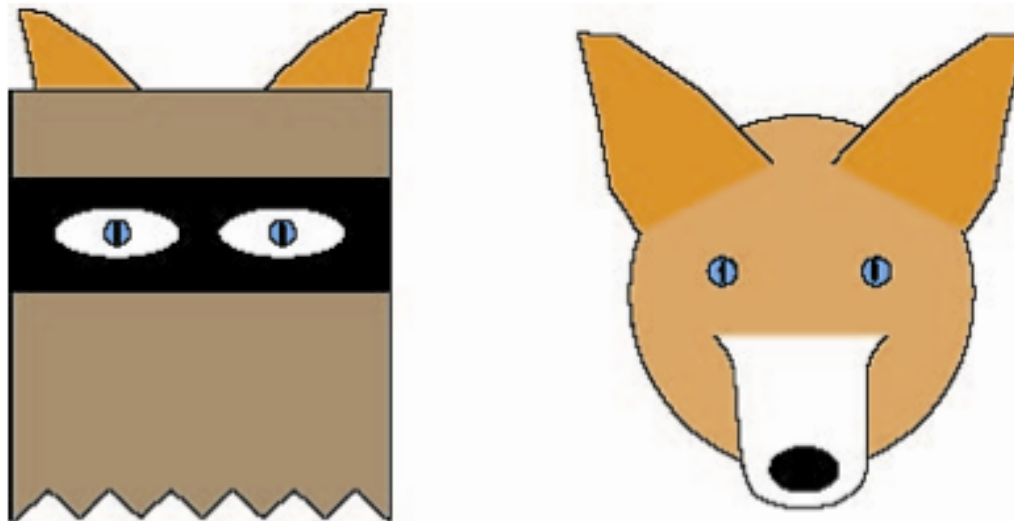
- ◆ Tor (<http://tor.eff.org>)
 - Overlay circuit-based anonymity network
 - Best for low-latency applications such as anonymous Web browsing
- ◆ Mixminion (<http://www.mixminion.net>)
 - Network of mixes
 - Best for high-latency applications such as anonymous email

Some caution

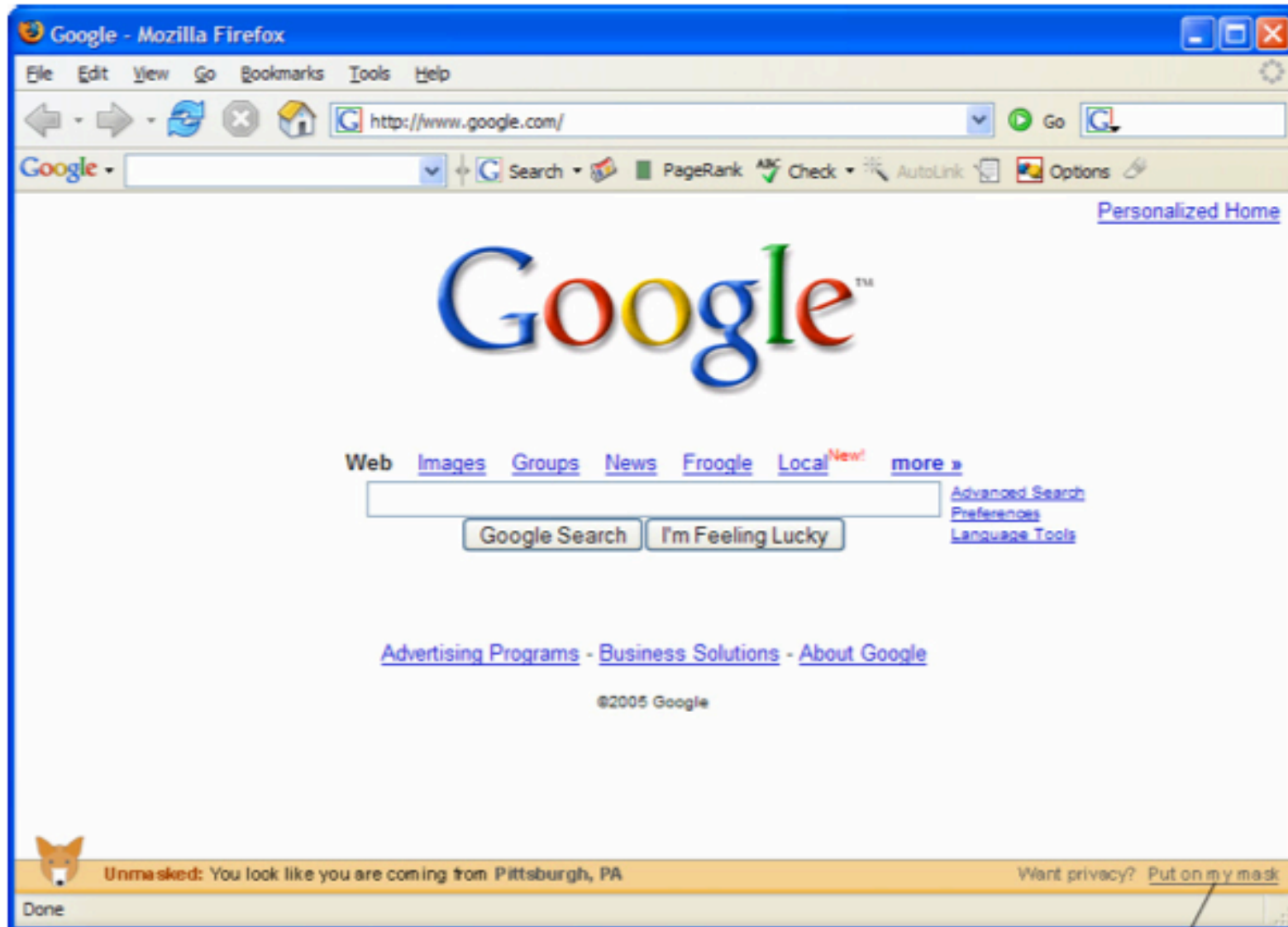
- ◆ Tor isn't completely effective by itself
 - Challenges if you have cookies turned on in your browser, are using JavaScript, etc.
 - Exit nodes can see everything!



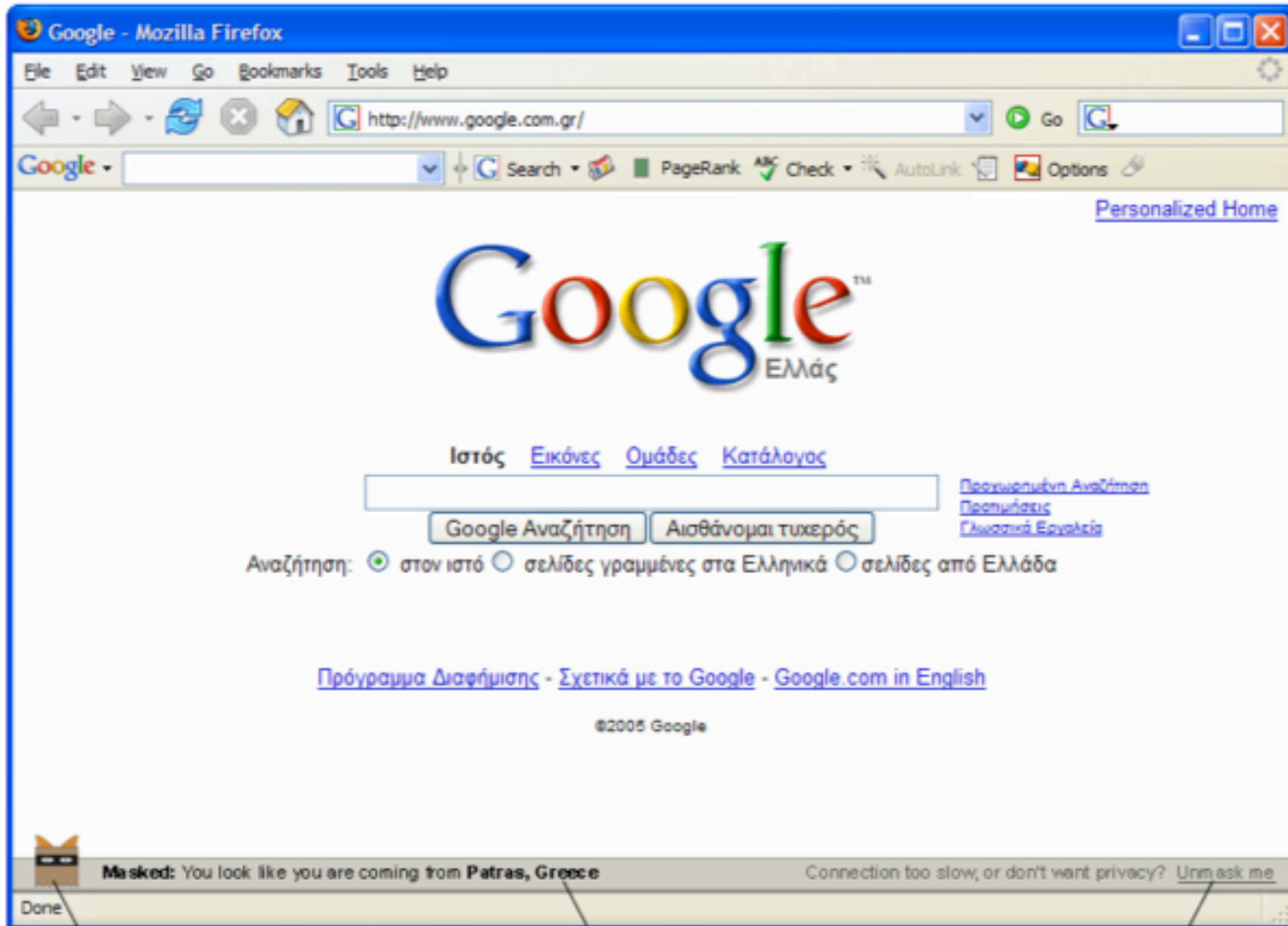
FoxTor, Images from <http://cups.cs.cmu.edu/foxtor/>



FoxTor, Images from <http://cups.cs.cmu.edu/foxtor/>



FoxTor, Images from <http://cups.cs.cmu.edu/foxtor/>



Ethics
