Advanced Network Security

Anonymity

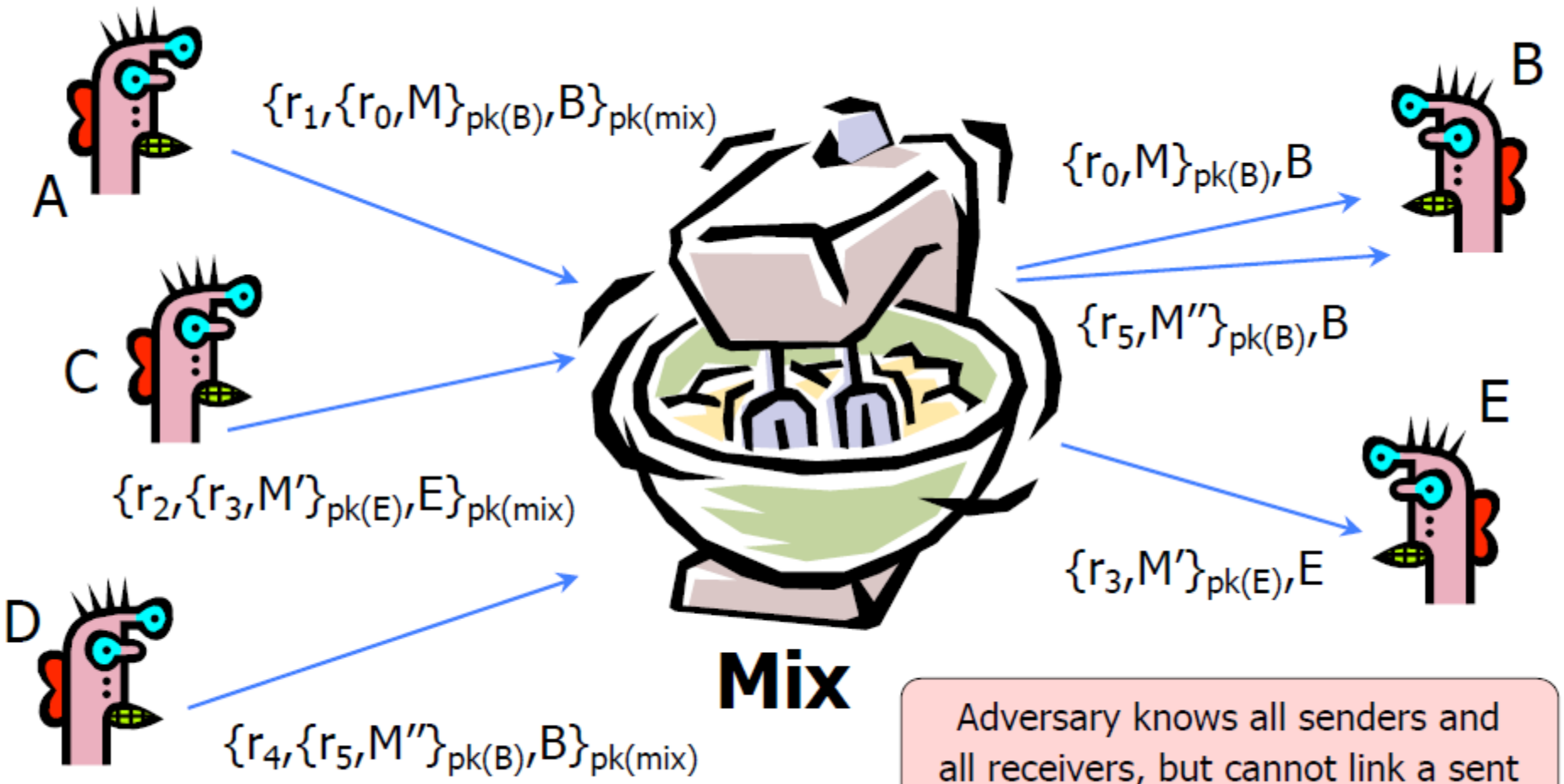
Amir Mahdi Sadeghzadeh, Ph.D.

Most slides have been adapted from Mehdi Kharrazi, CE40-817, Sharif University of Technology, 2015

Mix-net

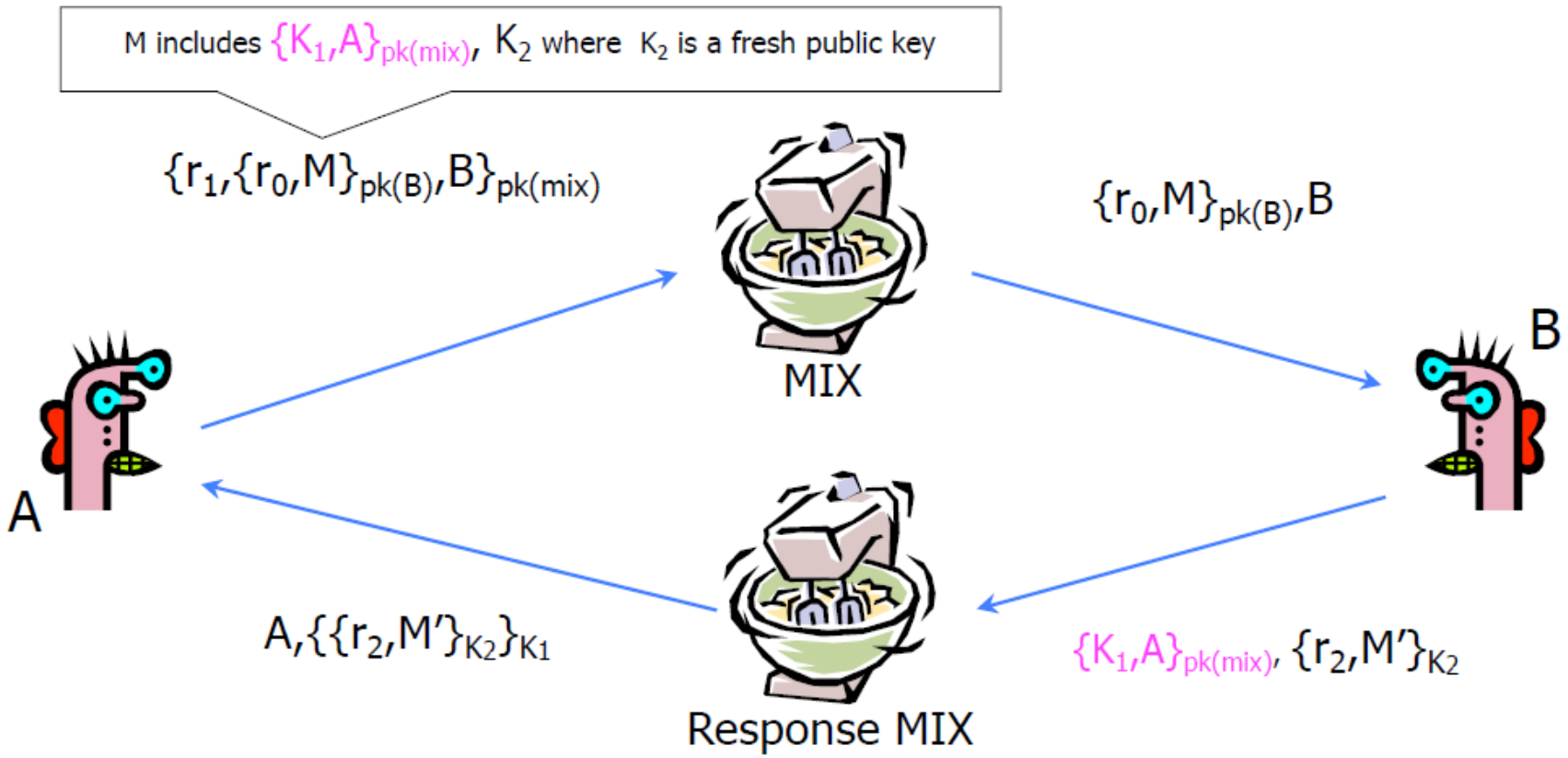
- Untraceable electronic mail, return addresses, and digital pseudonyms
- Chaum 1981
- Two Assumptions:
  - No correlation between a set of sealed and unsealed items
  - Anyone may learn the origin, destination and representation of all messages and may inject, remove, or modify messages

Basic Mix Design



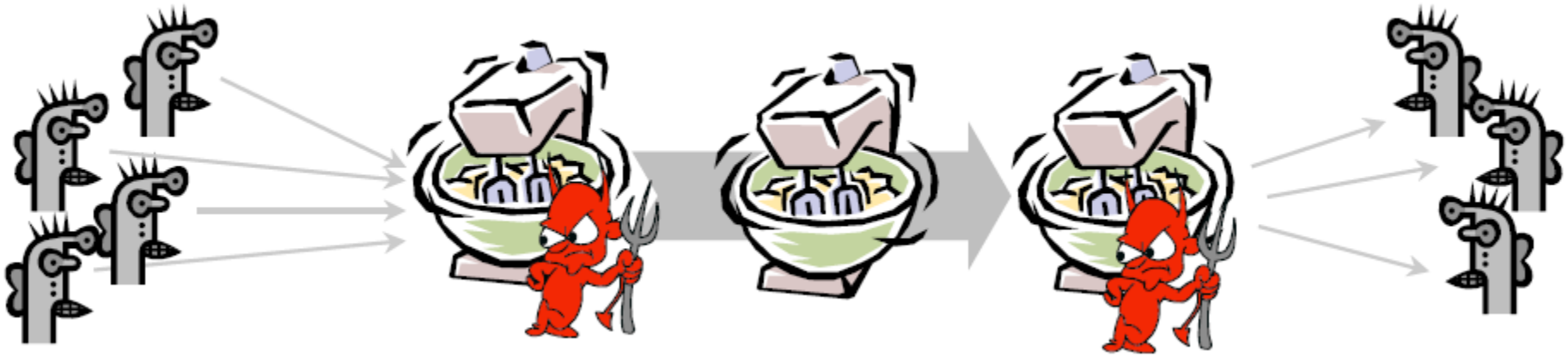
Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Anonymous Return Addresses



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



Mix-net

- No item which is repeated in the input should be allowed to be repeated in the output
  - Solution, make sure the nonce has some correspondence to time
- Depends on security of public key system
- Good for Periodic deliveries
- Has high Latency

Onion Routing

Onion Routing

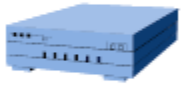
- Anonymous Connections and Onion Routing
- Reed, Syverson, and Goldschlag 1997
- http://www.onion-router.net/
- Real-time Chaum mixes
  - Called onion network

What makes up an onion network?

Application Proxy

Onion Proxy

Entry or Exit funnel



Onion Router

Onion

ACI (Anonymous Connection Identifier)

Onion Routing

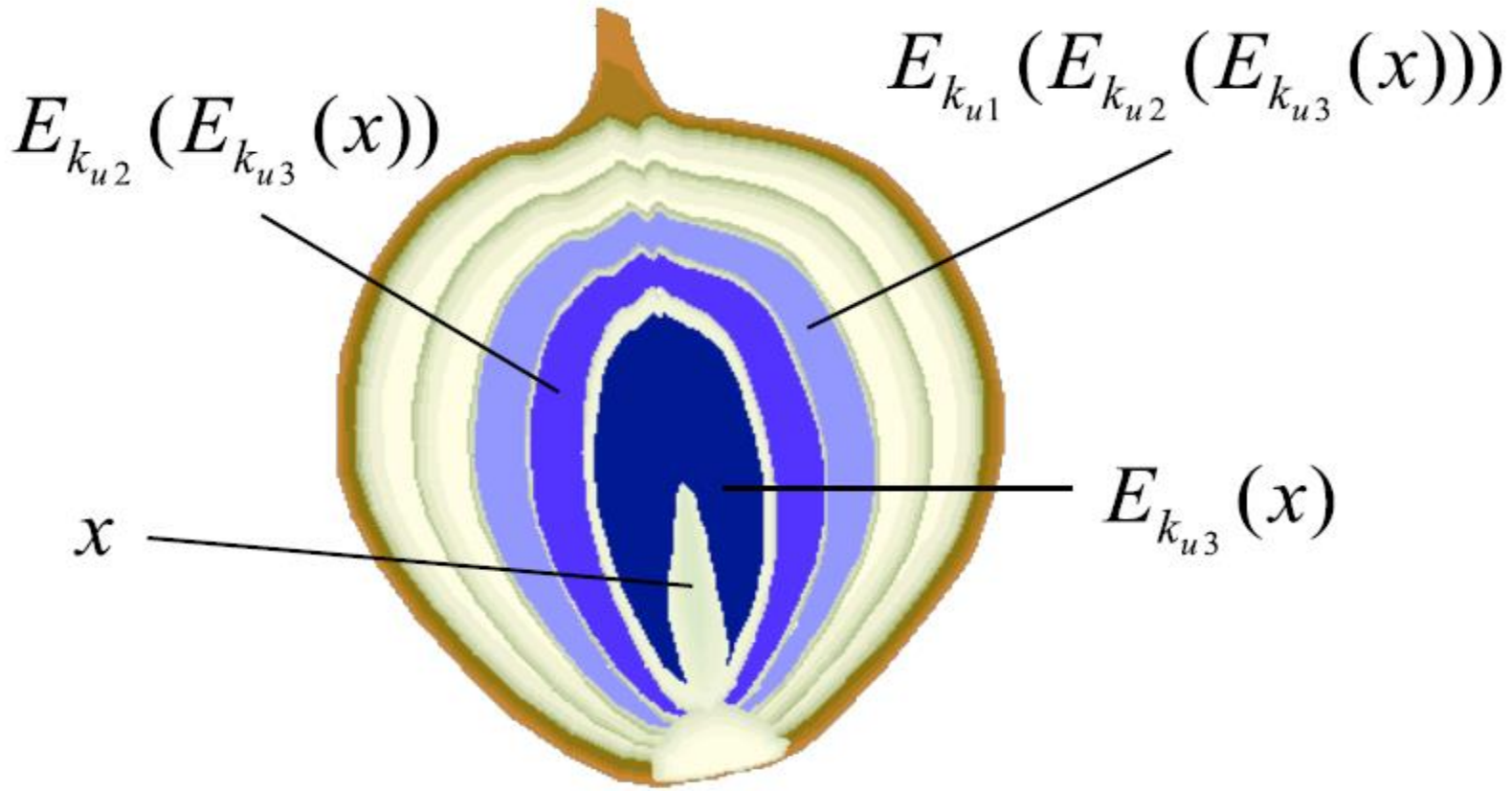
- 3 phases
  - Connection setup
    - Public key system is used
    - ACIs are made
  - Data delivery
    - Symmetric encryption used for speed
    - Received packets in a fixed time interval by a mix are Randomly reordered
  - Connection termination

Onion Routing

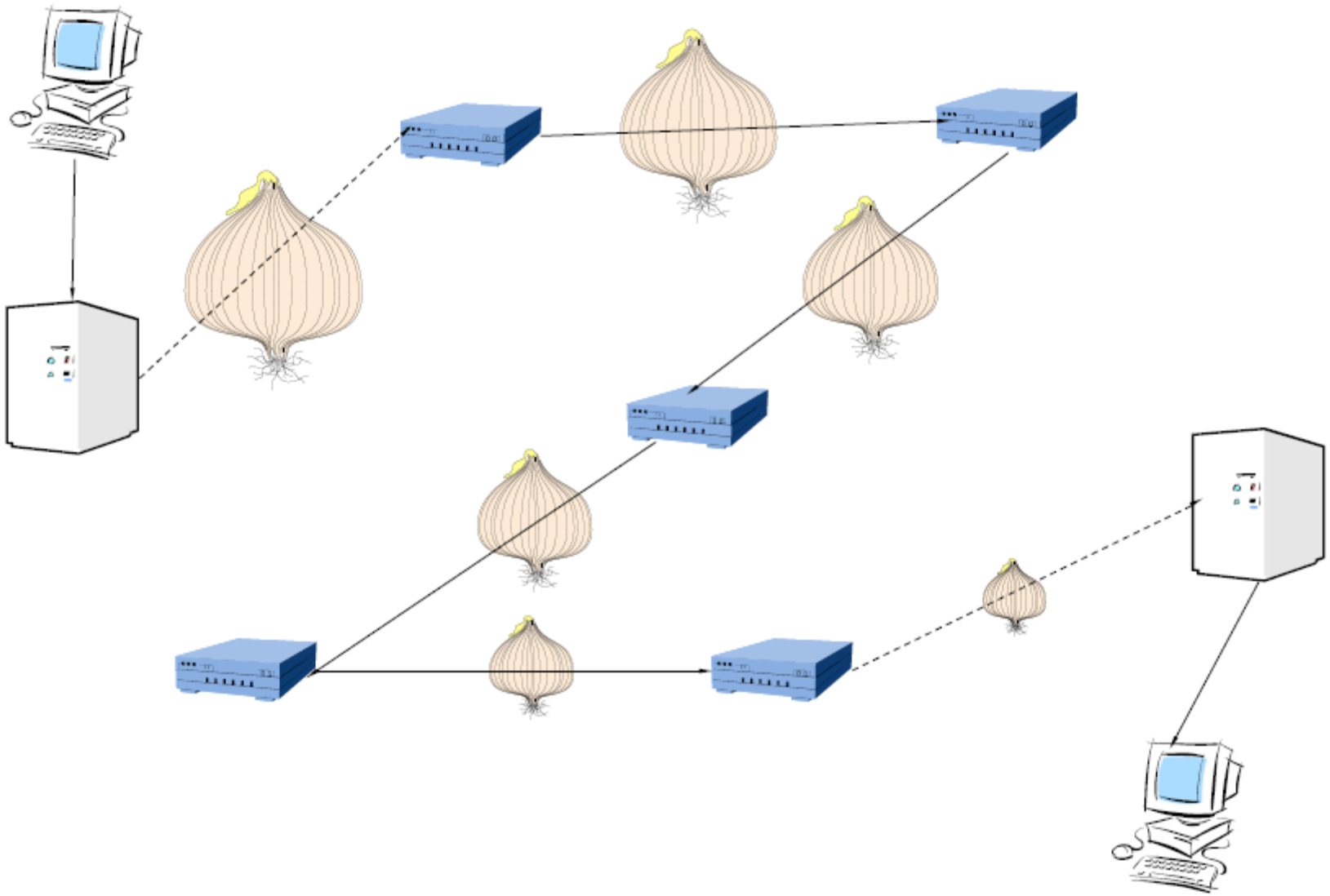
- Uniform sized cells of 128 bytes
  - Use of padding to maintain fix size
  - Because of cell size, route limited to maximum of 11 nodes
- Lower Limit on traffic flow
  - To avoid data burst attacks
- The network is intended for real time traffic

Onion details

- Were k_u is the public key of the onion routers
- The first encryption is done using the key of that last router and so on.



Onion Routing Example



Onion Routing

- Based on TCP/IP
- 2 separate functions:
  - Anonymity of connection
  - Anonymity of data stream
- Real-time capability
- Link padding

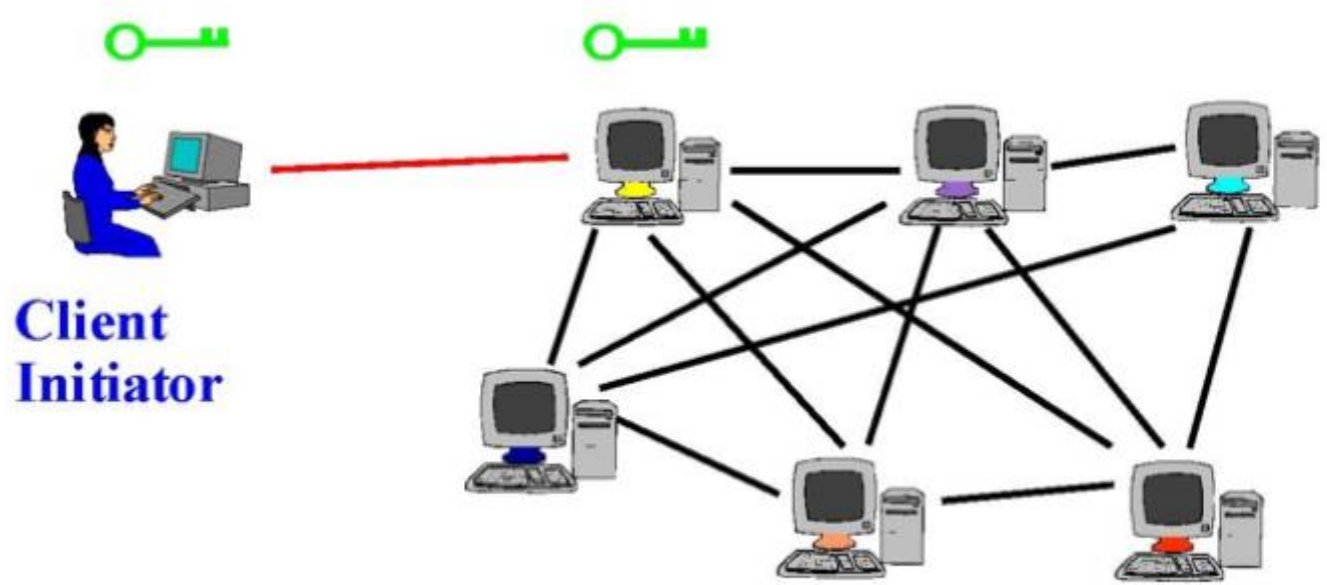
Tor (The Onion Router)

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
- About 420 routers in 2006 [Bauer]
- About 3000 in 2012
- “Easy-to-use” client proxy
  - Freely available, can use it for anonymous browsing

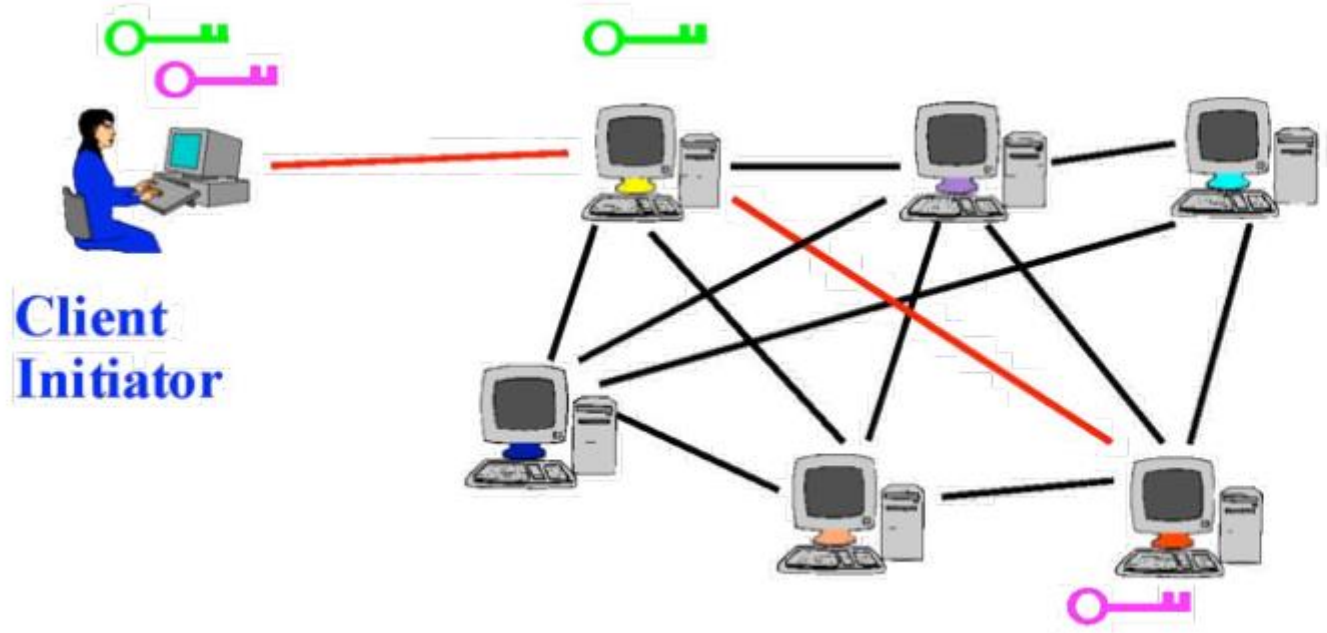
Tor Circuit Setup (1)

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



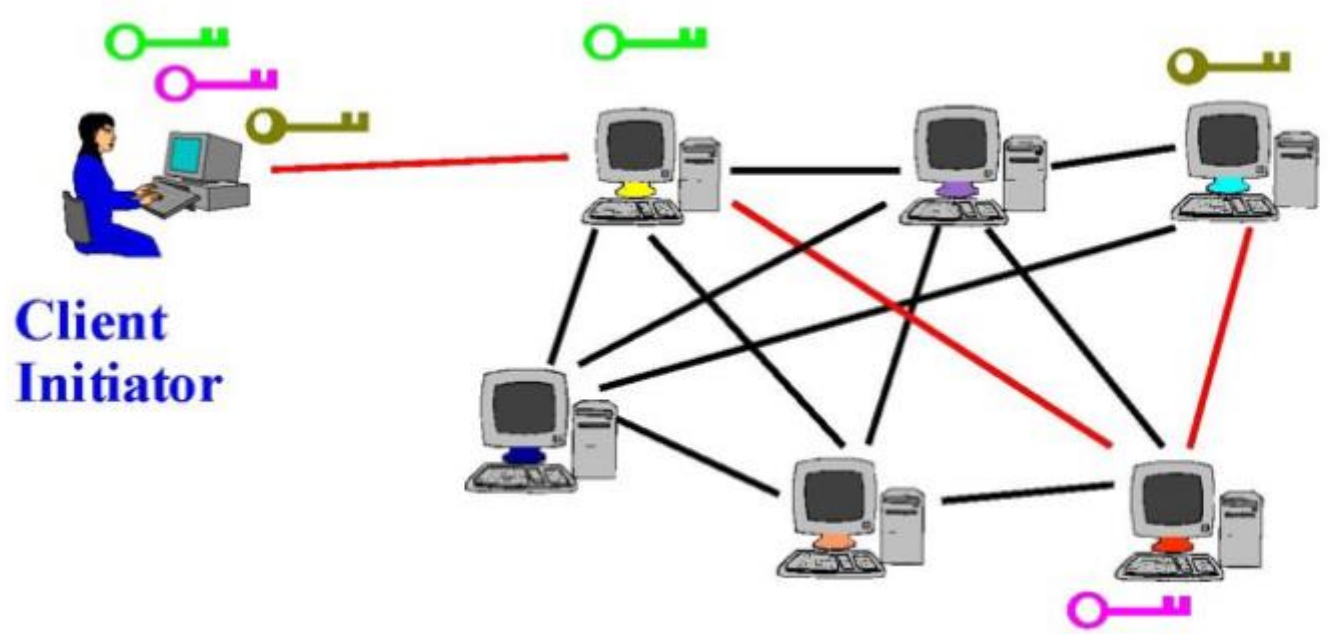
Tor Circuit Setup (2)

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



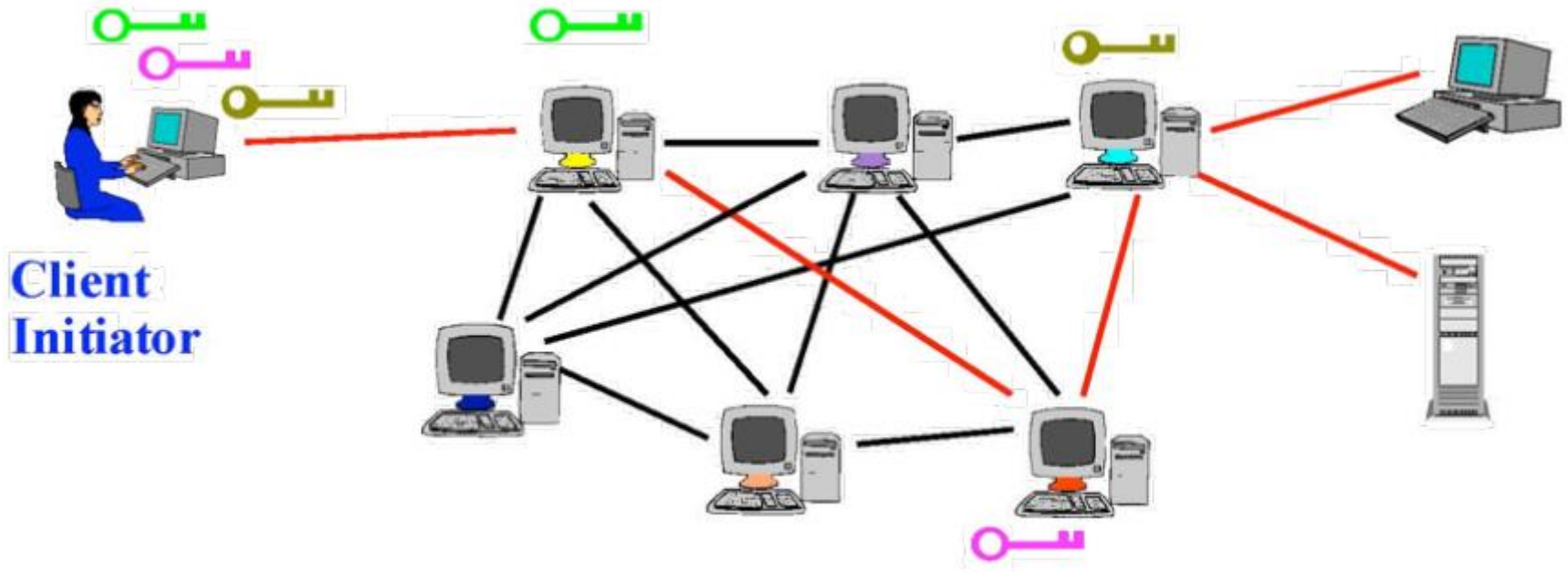
Tor Circuit Setup (3)

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



Tor Circuit Setup

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

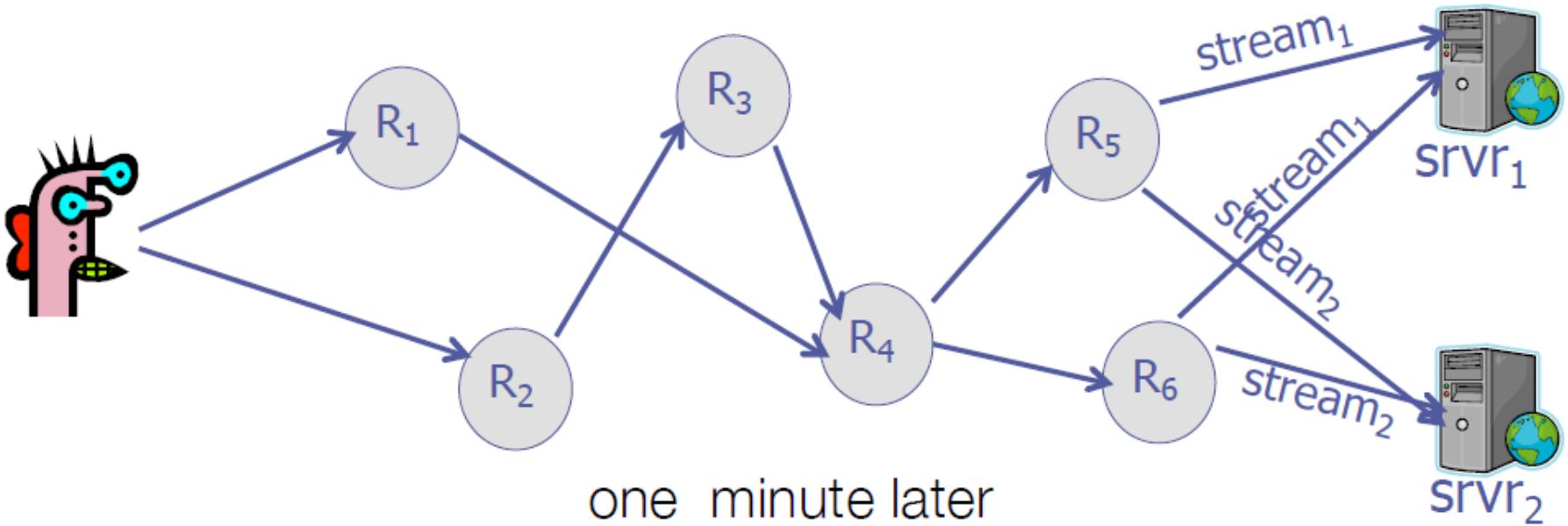


Tor Management Issues

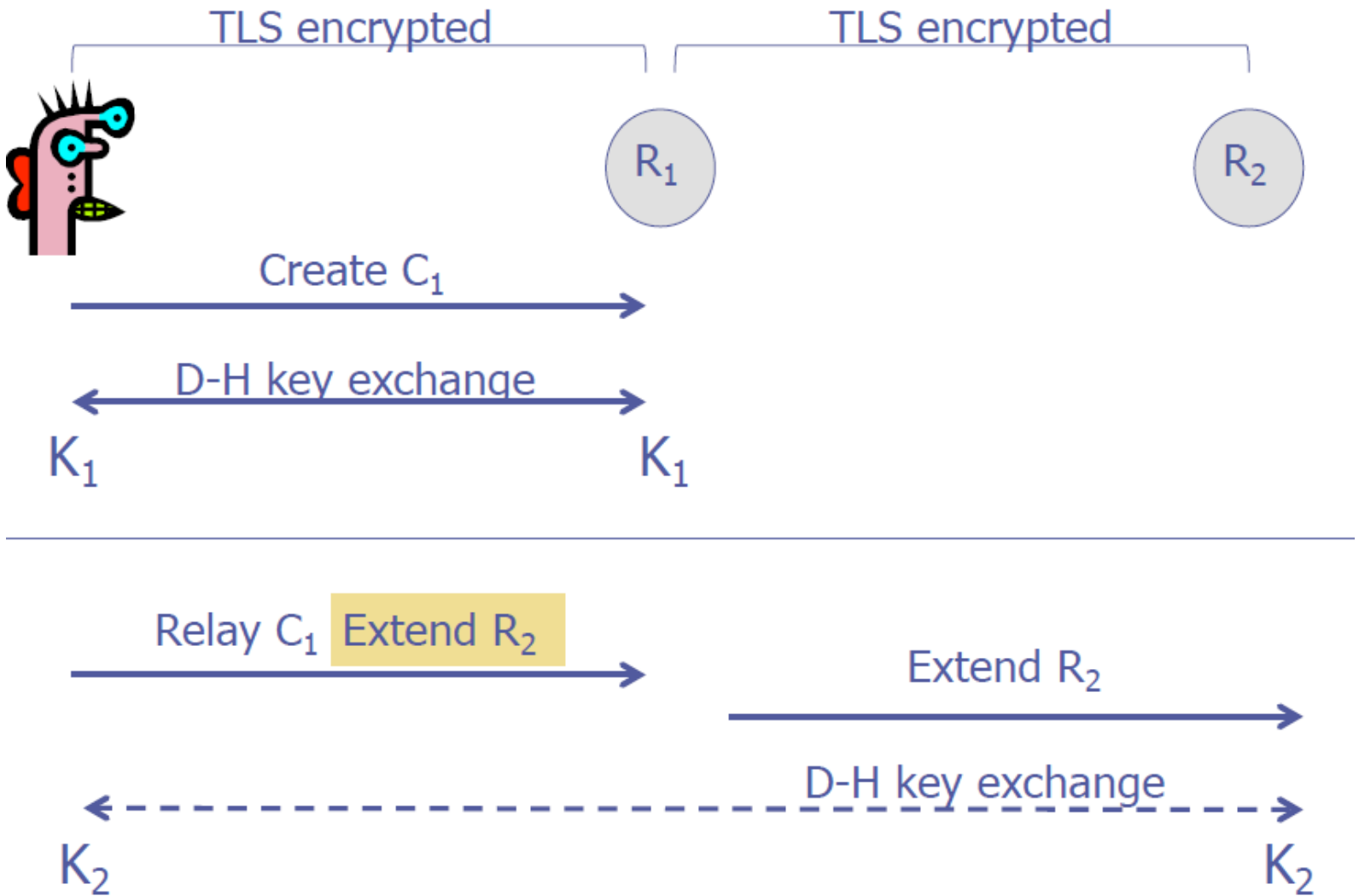
- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn't need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active onion routers,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

The Tor design

- Trusted directory contains list of Tor routers
- User's machine preemptively creates a circuit
  - Used for many TCP streams
  - New circuit is created once a minute

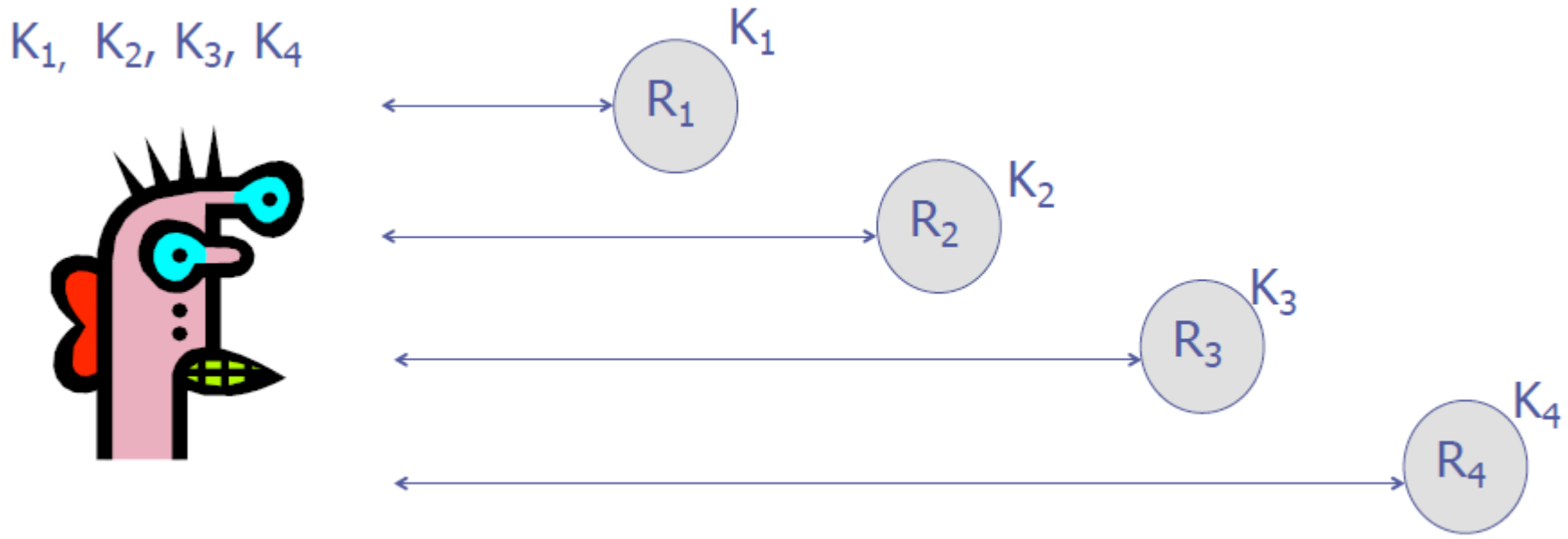


Creating circuits

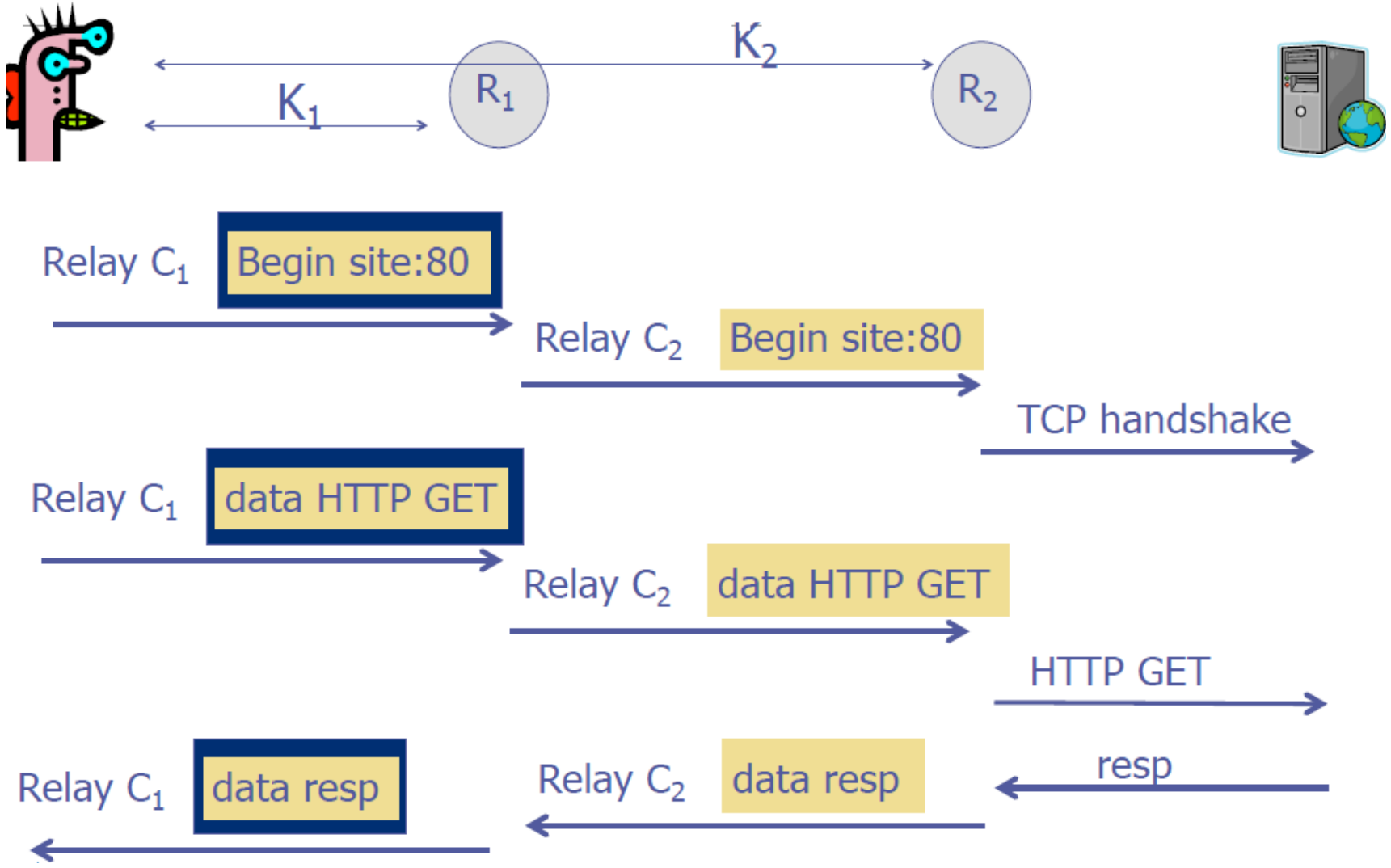


Once circuit is created

- User has shared key with each router in circuit
- Routers only know ID of successor and predecessor



Sending data



Properties

- Performance:
  - Fast connection time: circuit is pre-established
  - Traffic encrypted with AES: no pub-key on traffic
- Downside:
  - Routers must maintain state per circuit
  - Each router can link multiple streams via CircuitID
    - all streams in one minute interval share same CircuitID

Location Hidden Servers

- Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- Accessible from anywhere
- Resistant to censorship
- Resistant to physical attack
  - Can’t find the physical server!

Location Hidden Servers

- A connection to a hidden service involves five important nodes in addition to the nodes used for basic anonymous communication over Tor.
  - HS, the Hidden Server offering some kind of (hidden) service to the users of the Tor network, e.g. web pages, mail accounts, login service, etc.
  - C, the client connecting to the Hidden Server.
  - DS, a Directory Server containing information about the Tor network nodes and used as the point of contact for information on where to contact hidden services.
  - IP, the Introduction Point where the Hidden Server is listening for connections to the hidden service.
  - RP, the Rendezvous Point is the only node in the data tunnel that is known to both sides.

Overlier, Lasse, and Paul Syverson. "Locating hidden servers." 2006 IEEE Symposium on Security and Privacy (S&P'06). IEEE, 2006.

Location Hidden Servers

- All the displayed message flows are anonymized, i.e., they are routed through several anonymizing nodes on their path towards the other end

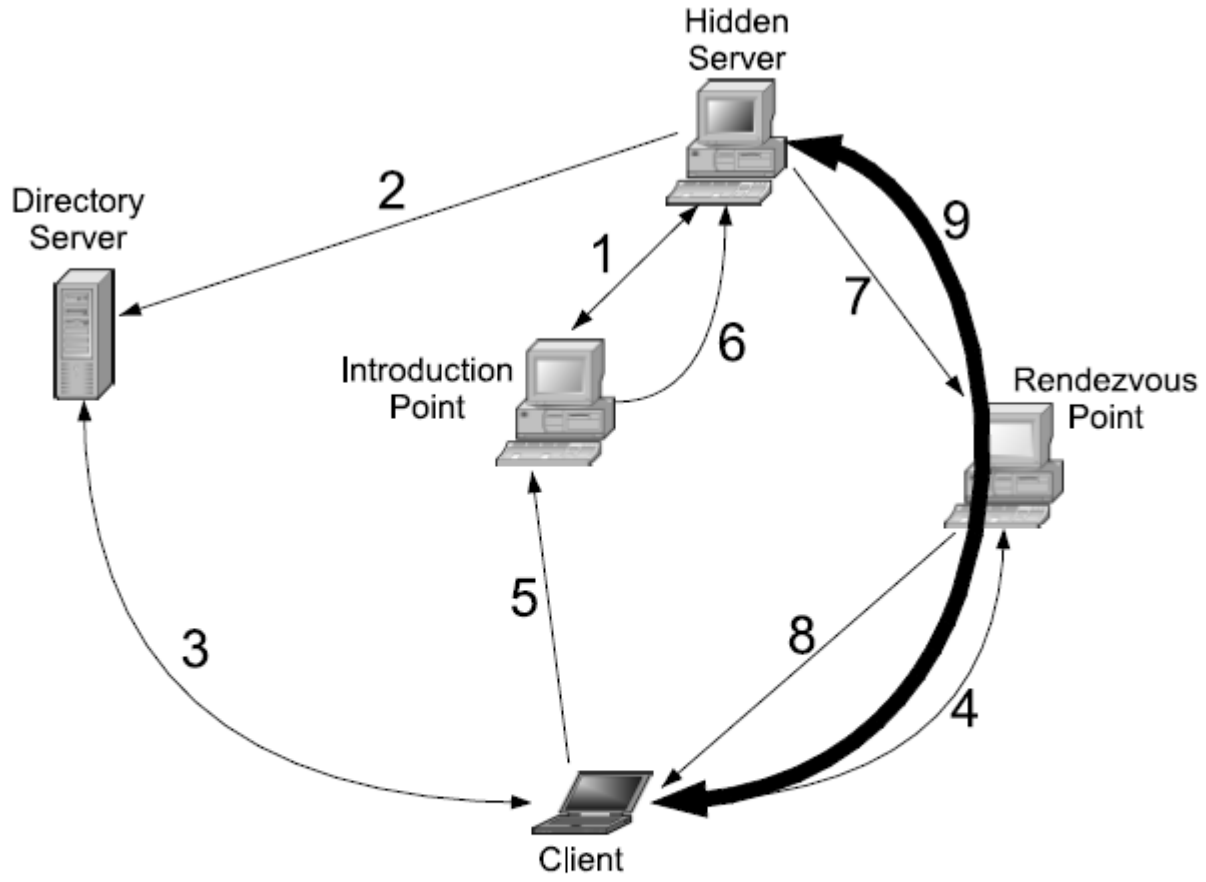


Figure 1. Normal use of hidden services and rendezvous servers

Overlier, Lasse, and Paul Syverson. "Locating hidden servers." 2006 IEEE Symposium on Security and Privacy (S&P'06). IEEE, 2006.

Location Hidden Servers

- First the Hidden Server connects (1) to a node in the Tor network and asks if it is OK for the node to act as an Introduction Point for his service. If the node accepts, we keep the circuit open and continue; otherwise HS tries another node until successful. These connections are kept open forever, i.e., until one of the nodes restarts or decides to pull it down.
- Next, the Hidden Server contacts (2) the Directory Server and asks it to publish the contact information of its hidden service. The hidden service is now ready to receive connection requests from clients.
- In order to retrieve data from the service the Client connects (3) to DS and asks for the contact information of the identified service and retrieves it if it exists (including the addresses of Introduction Points). There can be multiple Introduction Points per service.
- The Client then selects a node in the network to act as a Rendezvous Point, connects (4) to it and asks it to listen for connections from a hidden service on C's behalf.
- The Client repeats this until a Rendezvous Point has accepted, and then contacts (5) the Introduction Point and asks it to forward the information about the selected RP.
- The Introduction Point forwards (6) this message to the Hidden Server who determines whether to connect to the Rendezvous Point or not.
- If OK, the Hidden Server connects (7) to RP and asks to be connected to the waiting rendezvous circuit, and RP then forwards (8) this connection request to the Client.
- Now RP can start passing data between the two connections and the result is an anonymous data tunnel (9) from C to HS through RP.

Overlier, Lasse, and Paul Syverson. "Locating hidden servers." 2006 IEEE Symposium on Security and Privacy (S&P'06). IEEE, 2006.

Acknowledgments/References

- [Shmatikov] CS 378 - Network Security and Privacy, Vitaly Shmatikov, University of Texas at Austin, Fall 2007.
- David L. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, Communication of the ACM, Vol. 24, No. 2, Feb 1981
- Pfitzaman, A. and Waidner, M. 1987. Networks without user observability. Computer Security 2, 6, 158-166
- Marit Kohntopp, and Andreas Pfitzman, Anonymity, Unobservability, and pseudonymity-A Proposal for Terminology, Draft v.012 June 17, 2001
- David L. Chaum, The dining cryptographers problem: unconditional sender and recipient untraceability, journal of cryptology, 1:56-75, 1988
- Anonymous connection and onion routing, Syverson P., Goldshlag D., and Reed M. ,IEEE Journal on Selected Areas in Communications, VOL. 16, NO. 4, MAY 1998
- Reiter M.K. and Rubin A.D. Crowds: Anonymity for Web Transactions. ACM Transactions for on Information and System Security, 1(1):66-92, November 1998.

Advanced Network Security

Wireless Security

Amir Mahdi Sadeghzadeh, Ph.D.

Most slides have been adapted from Mehdi Kharrazi, CE40-817, Sharif University of Technology, 2015

Wireless Security

- What is Wireless Security?
- The usual: confidentiality, integrity, availability?

Confidentiality

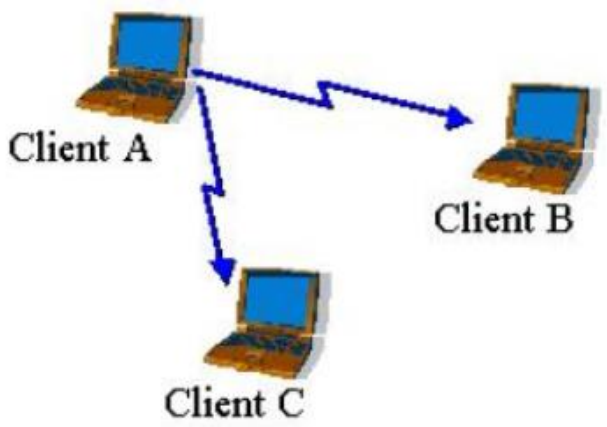
- Obvious danger — it’s easy to intercept traffic
- Obvious countermeasure — cryptography

Integrity

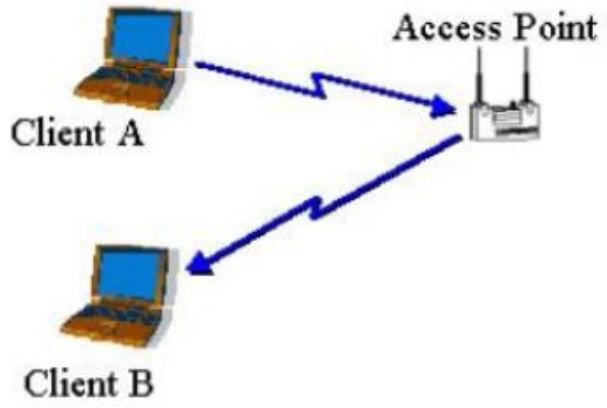
- At first glance, integrity seems ok
- This is radio — how can an attacker change messages in mid-packet?
- Solution: the “Evil Twin” (or “Sybil”) attack

Wireless Architecture

- The obvious architecture is pure peer-to-peer — each machine has a radio, and talks directly to any other machine
- In fact, 802.11 (WiFi) can work that way, but rarely does
- More common scenario: base stations (also known as access points)



IBSS (ad hoc) mode



BSS (infrastructure) mode

Access Points

- An ordinary wireless node associates with an access point (AP)
- More precisely, it associates with the AP having a matching network name (if specified) and the strongest signal
- If another AP starts sending a stronger signal (probably because the wireless node has moved), it will re-associate with the new access point
- All transmissions from the laptop go to the access point
- All transmissions to the laptop come from the access point

Access Point SSID

- Service Set Identifier (SSID) is the “name” of the access point
  - By default, access point broadcasts its SSID in plaintext “beacon frames” every few seconds
- Default SSIDs are easily guessable
  - Linksys defaults to “linksys”, Cisco to “tsunami”, etc.
  - This gives away the fact that access point is active
- Access point settings can be changed to prevent it from announcing its presence in beacon frames and from using an easily guessable SSID
  - But then every user must know SSID in advance

Example

- You could find defaults easily:

Netgear 802.11 DS products, ME102 and MA401

Default SSID: Wireless

Default Channel: 6

Default IP address: 192.168.0.5

Default WEP: Disabled

Default WEP KEY1: 11 11 11 11 11

Default WEP KEY2: 20 21 22 23 24

Default WEP KEY3: 30 31 32 33 34

Default WEP KEY4: 40 41 42 43 44

Default MAC: 00:30:ab:xx:xx:xx

Which AP?

- Which AP is your laptop associated with?
- Which network (SSID)?
- Many people know neither
- “My ISP is Linksys”
- Those who specify anything specify the SSID

The Evil Twin Attack

- Simplest way: carry an access point with you
- Simpler solution: many laptops can emulate access points
  - On Linux, use: iwconfig eth0 mode Master
- Force others to associate with your laptop, and send you all their traffic. . .

Why This Works

- Conventionally, we worry about authenticating the client to the server
- Here, we need to authenticate the server to the client
- The infrastructure wasn't designed for that; more important, users don't expect to check for it (and have no way to do so in any event)

Integrity Attacks

- We now see how to do integrity attacks
- We don’t tinker with the packet in the air, we attract it to our attack node
- You don’t go through strong security, you go around it

Availability

- Simple version: black-hole evil twin
- Sophisticated version: battery exhaustion

Black Holes

- Emulate an access point
- Do nothing with received packets
- More subtly, drop 10-15% of them — connections will work, but very slowly

Battery Exhaustion

- Send your enemy large “ping” packets
- The reply packets will be just as big — and transmitting such packets uses a lot of power
- The more you transmit, the more power — often battery power — you use up

WEP

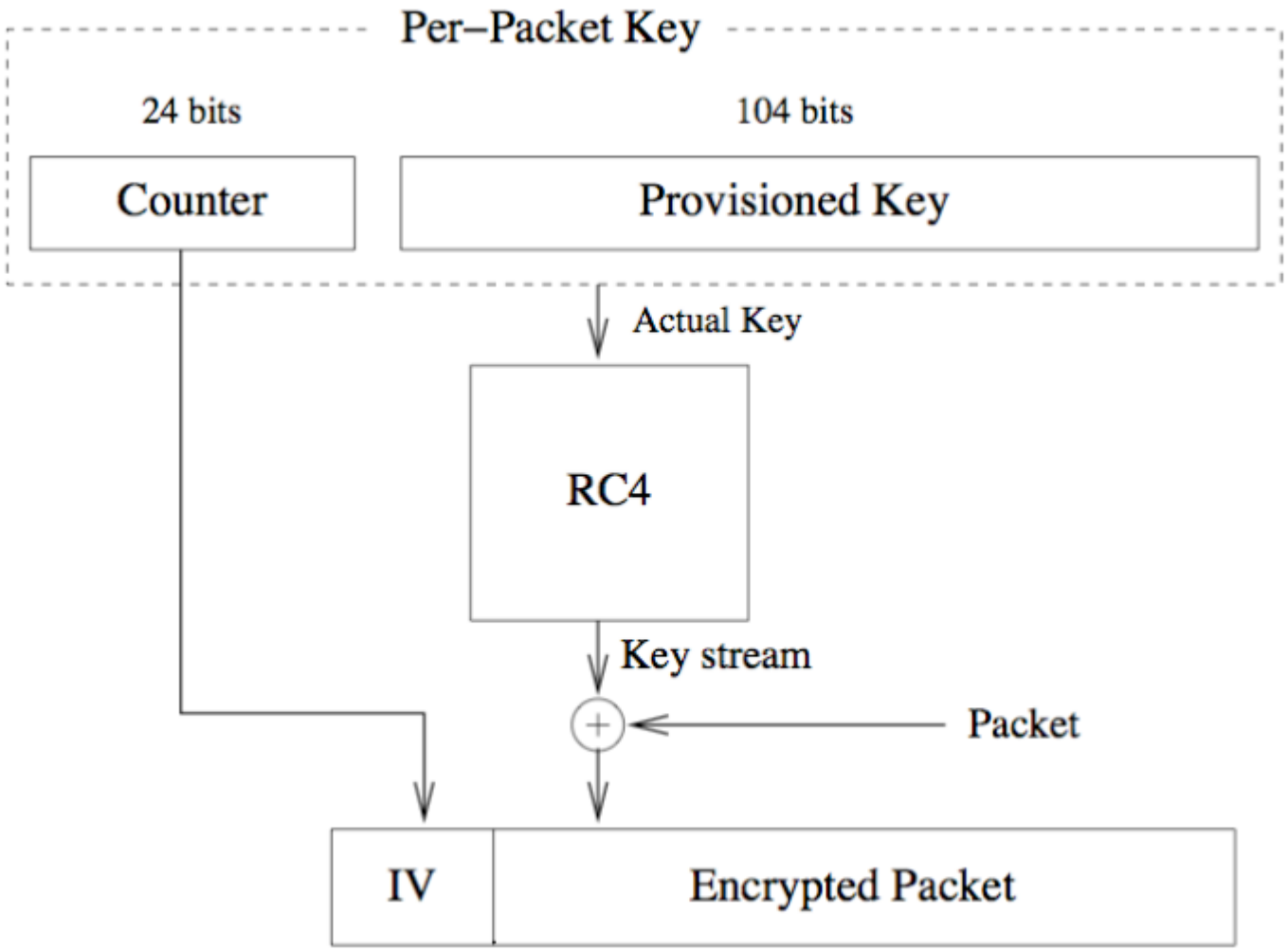
WEP — Using a Flawed Cipher in a Bad Way for the Wrong Application

- It was obvious from the start that some crypto was needed
- Choice: WEP — Wireline Equivalent Privacy for 802.11 networks
- Many different mistakes
- Case study in bad crypto design

Datagrams and Stream Ciphers

- WEP uses RC4 because RC4 is very efficient
- But 802.11 is datagram-oriented
- --> Must rekey for every packet
- But you can’t reuse a stream cipher key on different packets. . .

Key Setup


Per–Packet Key
24 bits
104 bits
Counter
Provisioned Key
Actual Key
RC4
Key stream
Packet
IV
Encrypted Packet

Key Setup for WEP

- Each WEP node keeps a 24-bit packet counter (the IV)
- Actual cipher key is configured key concatenated with counter
- Two different flaws. . .
- 2^24 packets isn't that many — you still get key reuse when the packet counter overflows
- RC4 has a cryptanalytic flaw
- But it's worse than that

Cryptanalysis of RC4

- In 2001, Fluhrer, Mantin and Shamir showed that RC4 could be cryptanalyzed if the keys were “close” to each other — a related key attack
- Because of the IV algorithm, they are close in WEP
- Key recovery attacks are feasible and have been implemented

IV Replay

- Suppose you recover the complete plaintext of a single packet
- You can generate new packets that use the same counter
- Receiving nodes don’t — and can’t — check for rapid counter reuse
- Indefinite forgery!

Packet Redirection

- Suppose you know (or can guess) the destination IP address of a packet
- Because RC4 is a stream cipher, you can make controlled changes to the plaintext by flipping ciphertext bits
- Flip the proper bits to send the packet to you instead, and reinject it

Checksums

- WEP does use a checksum
- However, it’s a CRC rather than a cryptographic hash
- It’s also unkeyed
- Result: it’s feasible to compensate for plaintext changes without disturbing the checksum

The Biggest Flaw in WEP

- There's no key management; all users at a site always share the same WEP key.
- --> You can't rekey when the counter overflows
- --> Everyone shares the same key; if it's cryptanalyzed or stolen or betrayed, everyone is at risk
- --> It's all but impossible to rekey a site of any size, since everyone has to change their keys simultaneously and you don't have a secure way to provide the new keys

What WEP Should Have Been

- Use a block cipher in CBC mode
- Use a separate key per user, plus a key identifier like the SPI
- Provide dynamic key management
- WPA — WiFi Protected Access — is better than WEP; forthcoming wireless security standards will use AES.

War-Driving

War-Driving

- Put a laptop in network (SSID) scanning mode
- Drive around a neighborhood looking for access points
- Perhaps include a GPS receiver to log locations
- Detect presence or absence of WEP

Unprotected Networks!

- Statistics show that only O(1/3) use even WEP
- The rest tend to be wide open
- Many people don’t change or hide the SSID

The Consequences

- Some incidence of theft of service
- (Is it war-driving a crime? Unclear under US law)
- Sometimes done to hide criminal activity

Acknowledgments/References

- [Bellovin06] COMS W4180 — Network Security Class Columbia University, Steven Bellovin, 2006.
- [Shmatikov] CS 378 - Network Security and Privacy, Vitaly Shmatikov, University of Texas at Austin, Fall 2007.