# Today in Cryptography (5830)

Summary of crypto we've seen
Crypto backdoors

# Some basic primitives

- Symmetric cryptography (shared key K)
  - encryption & decryption using K
  - message authentication using K
  - pseudorandom functions (PRF)
- Public-key cryptography (public key pk, secret key sk)
  - encrypt with pk and decrypt with sk
  - digitally sign using sk and verify with pk
- Hash functions (no keys)
  - used to “compress” messages in a secure way

# TLS handshake for RSA transport

Client

Server

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods →

Pick random Ns

← ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Check CERT using CA public verification key

← CERT = (pk of bank, signature over it)

Pick random PMS
C <- E(pk,PMS)

C →

PMS <- D(sk,C)

ChangeCipherSpec,
{ Finished, PRF(MS, "Client finished" || H(transcript)) } →

ChangeCipherSpec,
{ Finished, PRF(MS, "Server finished" || H(transcript')) } ←

Bracket notation means contents encrypted

MS <- PRF(PMS, "master secret" || Nc || Ns )

| Primitive | Use cases / examples | Security goals | Good schemes | Bad schemes |
|---|---|---|---|---|
| **Block ciphers** | Building block for symmetric encryption | Indistinguishable from random permutation | AES, 3DES | DES, Skipjack |
| **Pseudorandom Functions (PRFs) / Message authentication codes (MACs)** | Authenticating data with shared secret key, key derivation | Indistinguishable from random function | HMAC w/ good hash function<br>OMAC, CMAC, PMAC | CBC-MAC without prefix-free encoding |
| **Symmetric encryption (length-extending, authenticated encryption)** | Main mechanism for encrypting data; TLS record layer, encrypting data at rest | Message confidentiality and associated data + ciphertext authenticity | Encrypt-then-MAC<br>GCM<br>OCB | Encryption only modes:<br>CTR mode, CBC mode, ECB mode, RC4 |
| **Hash functions** | Key derivation, PW hashing, digital signatures, HMAC | Behave like a public random function (implies coll resist, one-wayness, etc.) | SHA-256<br>SHA-3 | MD4, MD5, SHA-1 |
| **Password-based key derivation** | Password hashing, PW-based encryption | No shortcut attacks | PBKDF2, scrypt, bcrypt, argon2 | Plain hash function |

| Primitive | Use cases / examples | Security goals | Good schemes | Bad schemes |
|---|---|---|---|---|
| **RSA PKE** | Encrypt symmetric key | No partial info on messages leaked to active attacker | RSA-OAEP w/ 2048 bit moduli | RSA-PKCS#1 v1.5, “raw” RSA < 2048 bit N |
| **ECC PKE** | Encrypt symmetric key | No partial info on messages leaked to active attacker | | ElGamal by itself |
| **Hybrid encryption** | Encrypt data efficiently using recipient public key | No partial information on messages leaked; attacker can’t maul ciphertext | ECIES<br>RSA-OAEP w/ one-time Encrypt-then-MAC scheme | Raw RSA kem, bad sym encryption (e.g., CBC mode) |
| **Digital signatures** | Authenticated key exchange, code signing | Unforgeability under chosen message attacks | ECDSA, RSA PSS | RSA-PKCS#1 v1.5 |
| **Diffie-Hellman key exchange** | Establishing secure channel | Attacker can’t recover derived session key | ECC DH,<br>Finite field DH | << 256 bit ECC groups, << 2048 bit FF groups |

# Cryptographic backdoors

- Long debate over whether average citizens should have access to strong crypto
  - "Crypto wars" of 1990s: export restrictions that treat crypto software as munitions

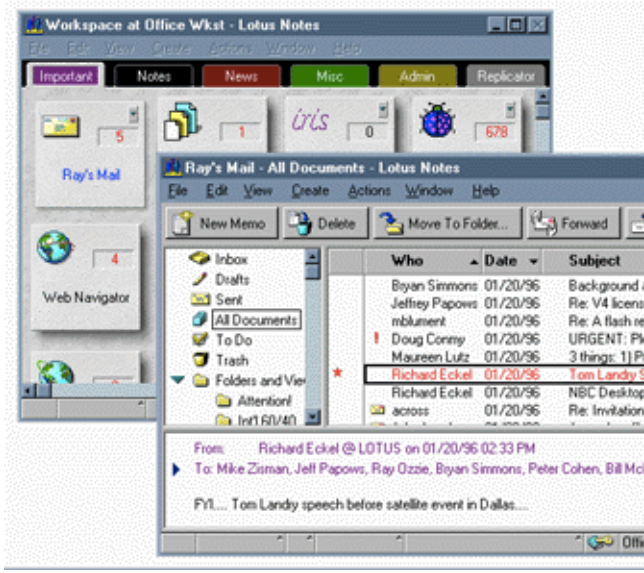
- Overt and surreptitious backdoors seen as backup plan by governments

# Overt backdoors

- Clipper chip
  - NSA hardware for encrypting telecommunications
  - Each chip had secret key, this was given to (escrowed with) NSA at manufacture time
- Significant backlash
- "The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption" by Abelson et al.

# Overt backdoors

- Export controls required only 40-bit keys for international software
- Lotus Notes "Differential Workfactor Cryptography"
  - 64 bit symmetric key K
  - C1 = RSA-Enc($pk_{NSA}$ , top24(K) )
  - C2 = Enc(K, data)



# Surreptitious backdoors

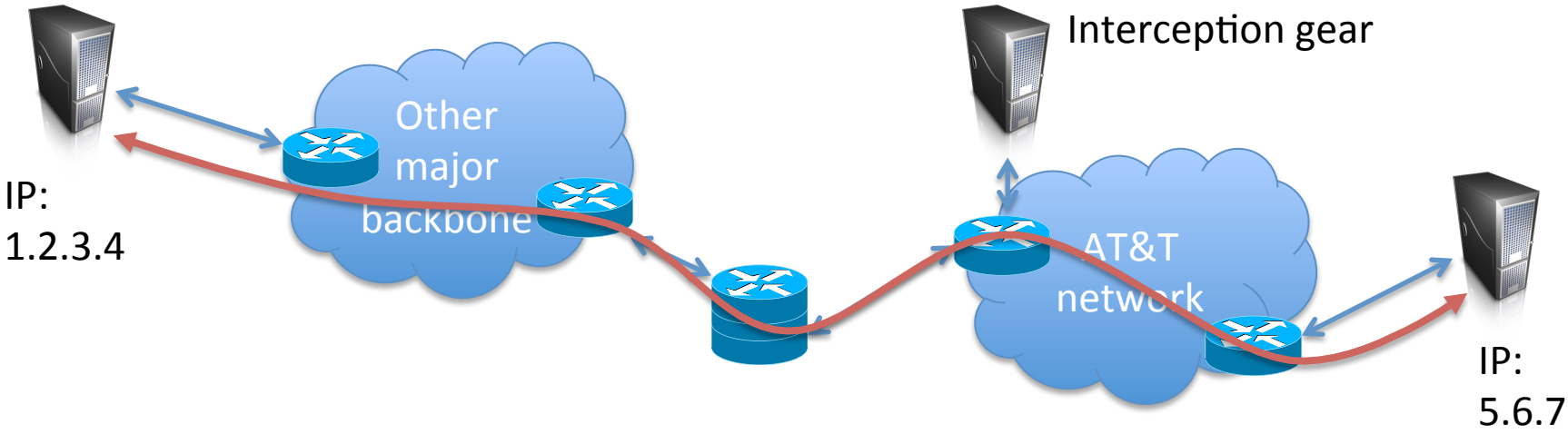
- Secretly weaken / sabotage cryptographic systems
- Usually done to dovetail with interception capabilities

# AT&T Wiretap case

- Mark Klein discloses potential wiretapping activities by NSA at San Francisco AT&T office
- Fiber optic splitter on major trunk line for Internet communications
  - Electronic voice and data communications copied to “secret room”
  - Narus STA 6400 device

# Preventing intercept

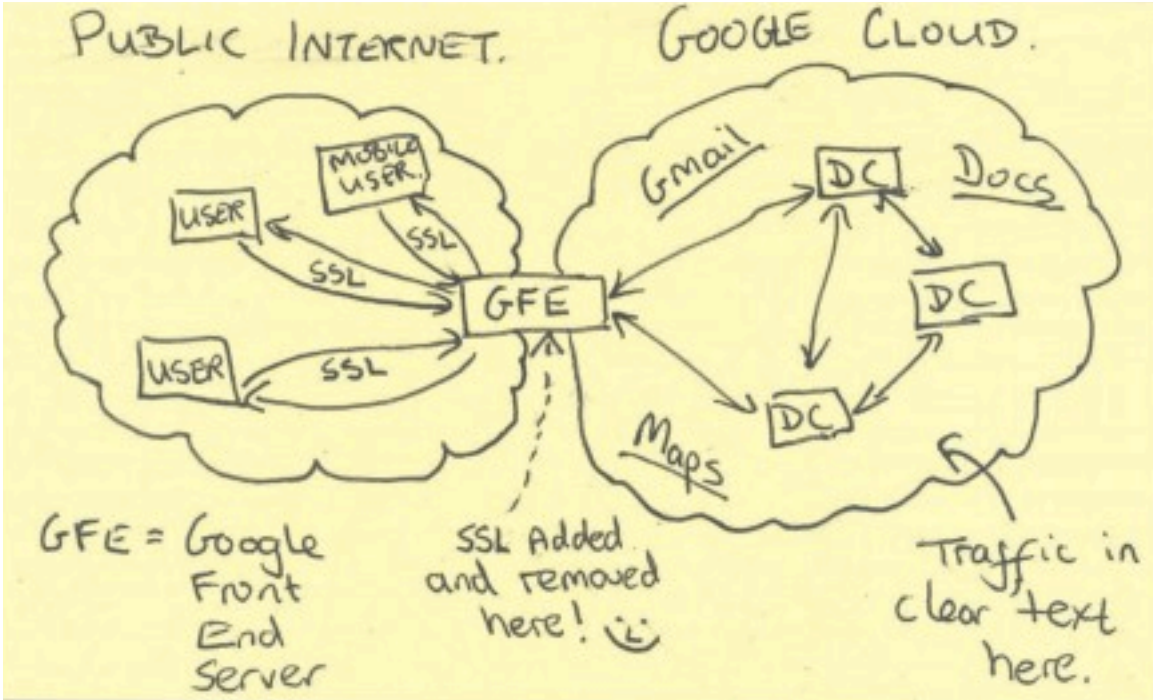
- End-to-end encryption (TLS, SSH)




- What does this protect? What does it leak?
- What can go wrong?

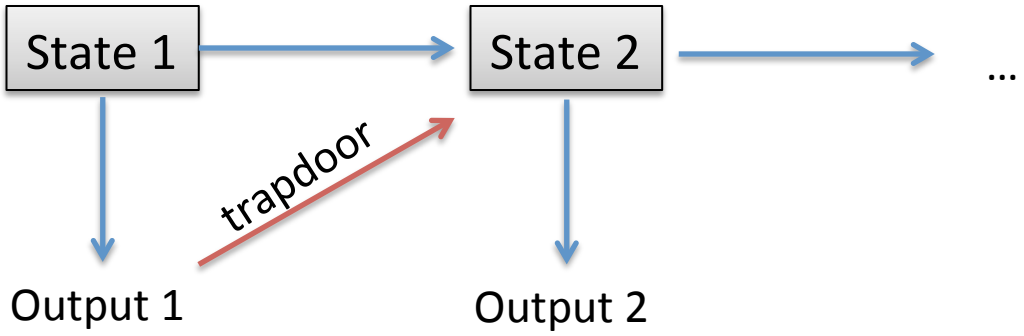
# End-run around HTTPS

- HTTPS terminated at edge of Google networks
- Internal data center-to-data center communications on privately leased lines
  - No encryption up until summer 2013



# Sabotaging TLS

- NIST's Dual EC pseudorandom number generator (PRNG) apparently backdoored
  - Mandated public parameters are public key
  - There exists a secret key, the trapdoor
- One output of PRNG + trapdoor reveals next state of PRNG, and prediction of future outputs



# TLS handshake for RSA transport

Bank customer

Bank

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods

Pick random Ns

ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Check CERT using CA public verification key

CERT = (pk of bank, signature over it)

Pick random PMS
C <- E(pk,PMS)

C

PMS <- D(sk,C)

Say client is using Dual EC for randomness generation
What is vulnerable?

RSA BSAFE library: 2.4 seconds to recover PMS

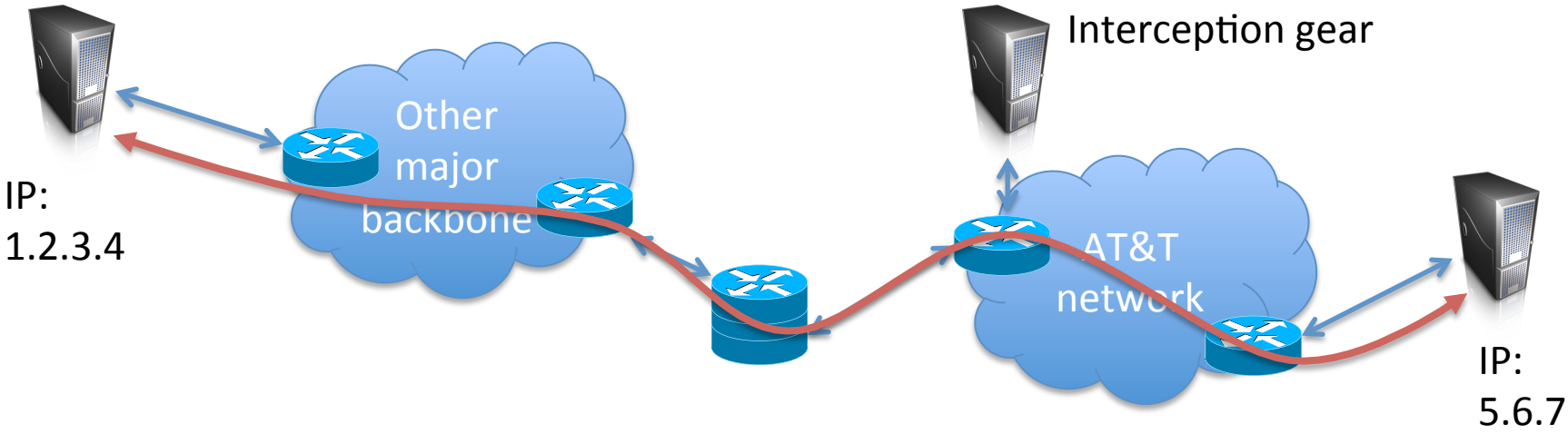
Windows: 60 minutes

OpenSSL: never (bug in code!)

See http://dualec.org/

# Preventing intercept

- End-to-end encryption (TLS, SSH)




- What does this protect? What does it leak?
- What can go wrong?

# Hiding “metadata” such as connectivity is even harder

- IP addresses are required to route communication, yet not encrypted by normal end-to-end encryption
  - 1.2.3.4 talked to 5.6.7.8 over HTTPs
- How can we hide connectivity information?

# Anonymization systems

- Single-hop proxy services

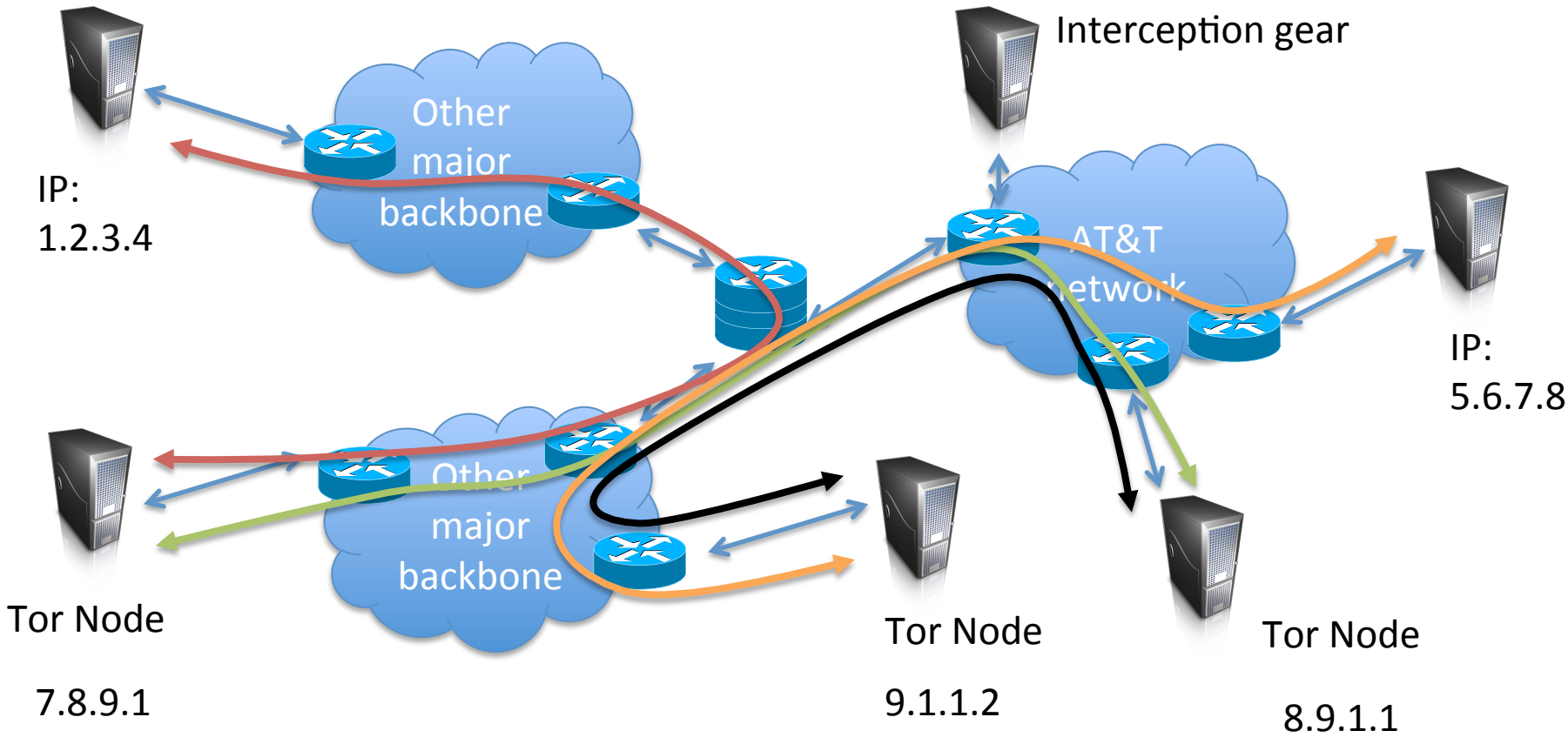

- JonDonym, anonymous remailers (MixMaster, MixMinion), many more…

Thursday, April 26, 2012

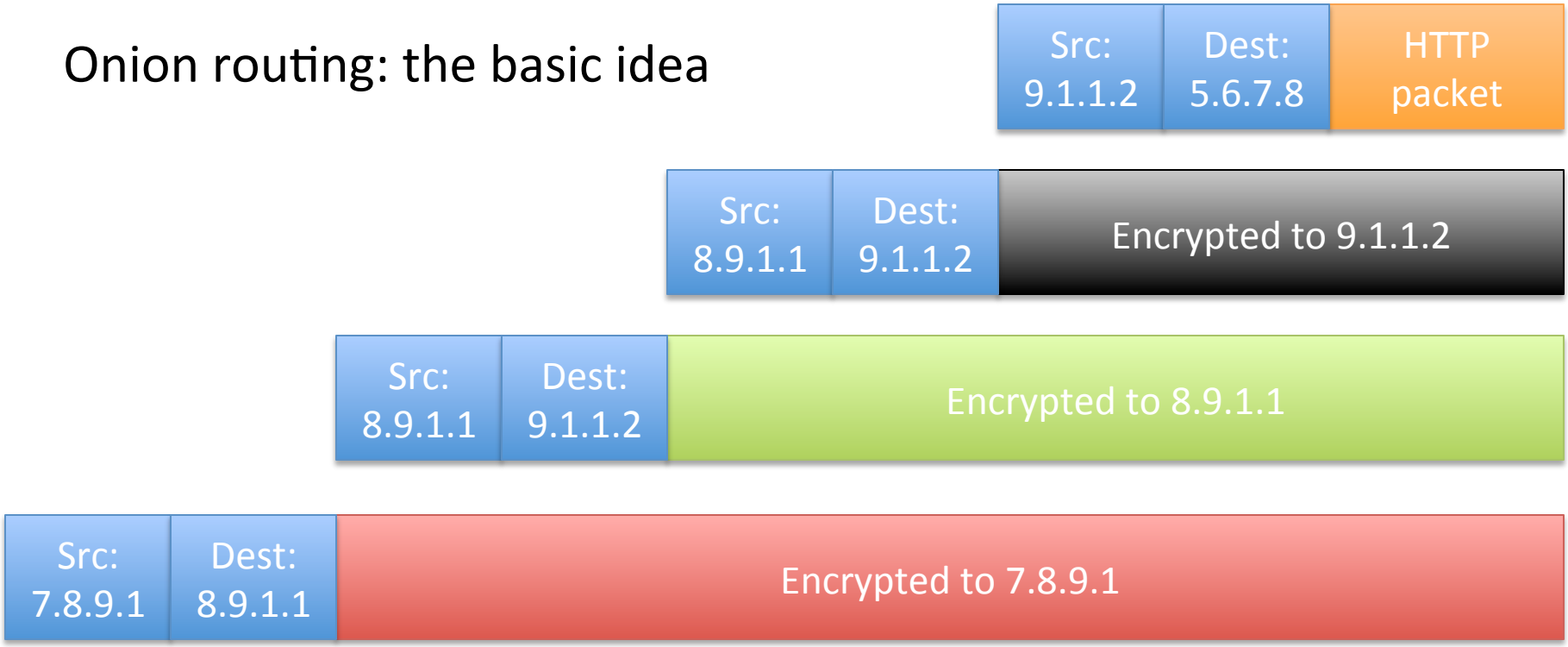
**FBI seizes server used to anonymize e-mail**

Jeffrey Brown 1 comment

# Tor (The Onion Router)

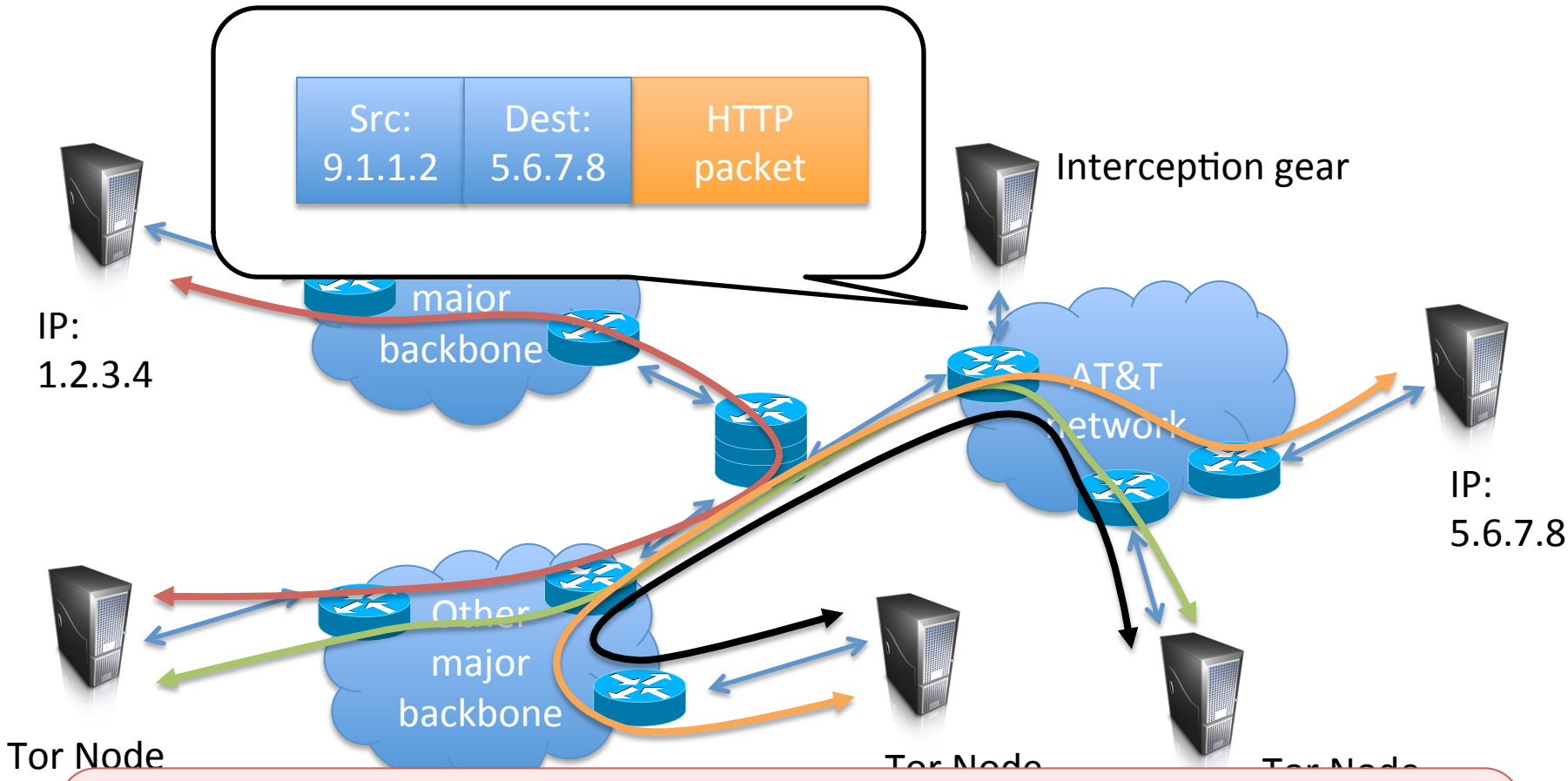



## Onion routing: the basic idea




Tor implements more complex version of this basic idea

# What does adversary see?




Tor obfuscates who talked to who, need end-to-end encryption (e.g., HTTPS) to protect payload