COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Communications Privacy I: Introduction

Information Privacy with Applications

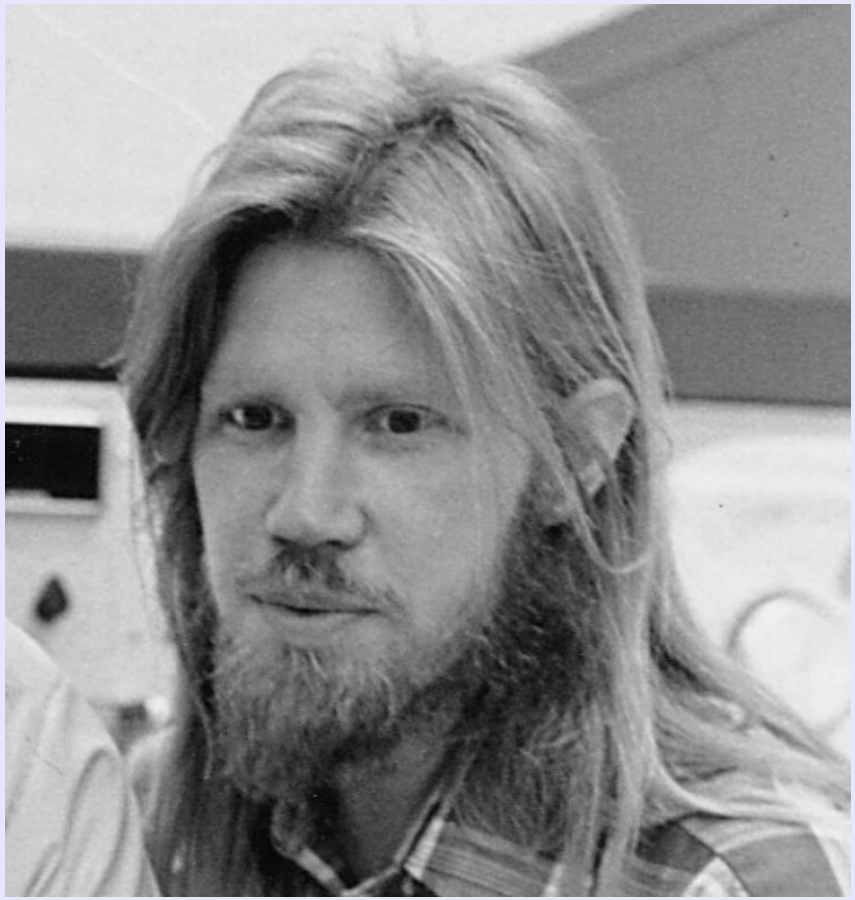
David Sidi (dsidi@email.arizona.edu)

Warm-up

- Give one example of a layer 8+ issue described in **POTL**.

Small mention of interesting things

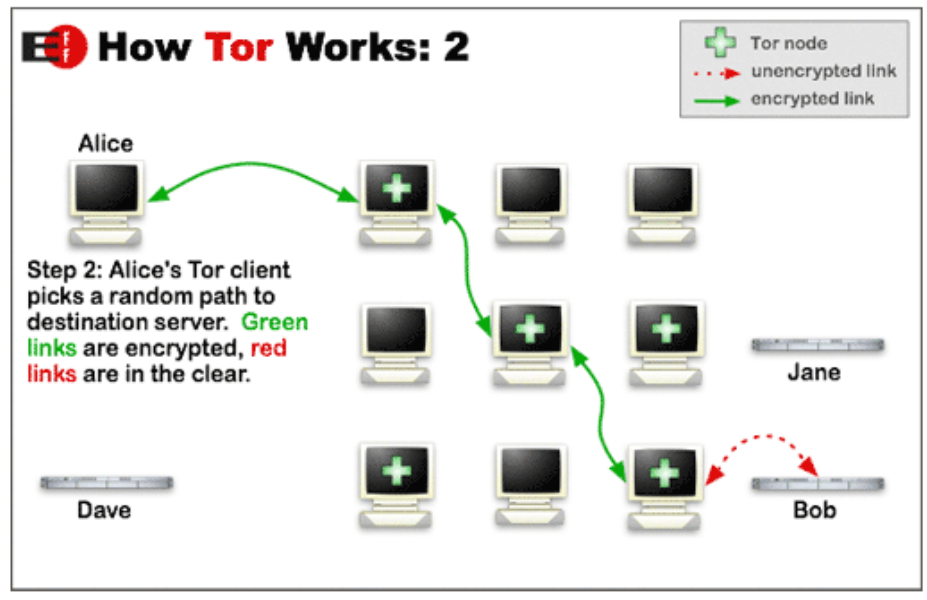
- Whitfield Diffie (from today's reading)
  - Father of PKC (with Martin Hellman and Ralph Merkle)
  - Turing Award Winner
  - recently: CISAC, ICANN, Cryptomathic , Blackridge, ASECO Lab



Continuing last time: Onion Routing

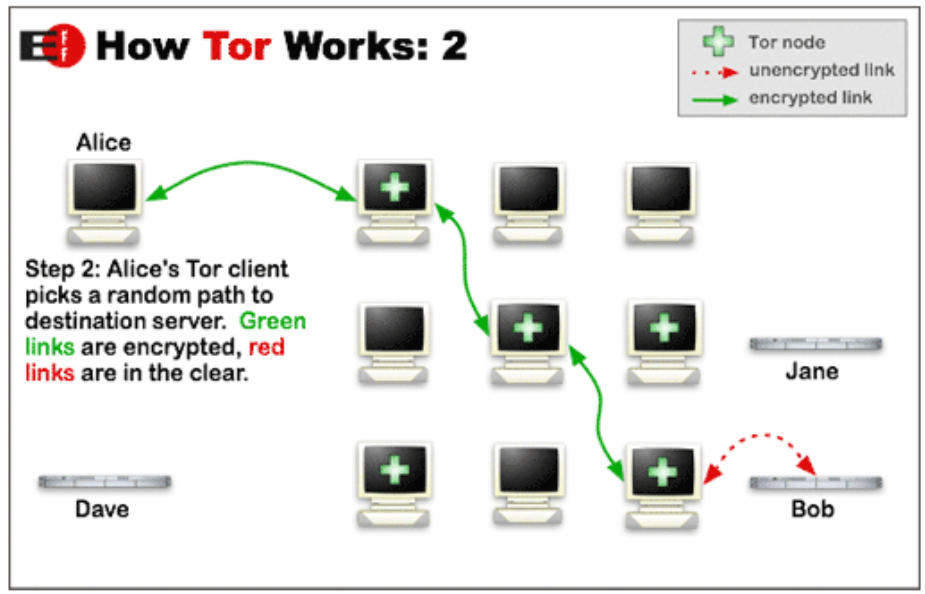
Onion Routing

- First was from the US Naval Laboratory, 1996
  - pure peering at this stage, loafers!
- Freedom Network was an independent onion routing network from Zero Knowledge Systems
- Tor is a third-gen. onion routing network



Tor is an onion routing system

- Example: simplified, slightly out-of-date Tor (*link*)
- Distributed overlay network for TCP-based applications
- Sets up a “virtual circuit” as a cascade of three onion relays (OR) from the initial client onion proxy (OP)
- guard (from “helper nodes”), relay, and exit nodes
  - each node only knows its immediate predecessor and successor



Tor is an onion routing system

- originally, onion routing systems sent an initial onion message that was “just layers” to set up the circuit; Tor does it in stages (“telescoping”)
- Next hop in the circuit is determined by unwrapping an “extend” relay cell with a symmetric key, which causes the OR to send its own “create” control cell

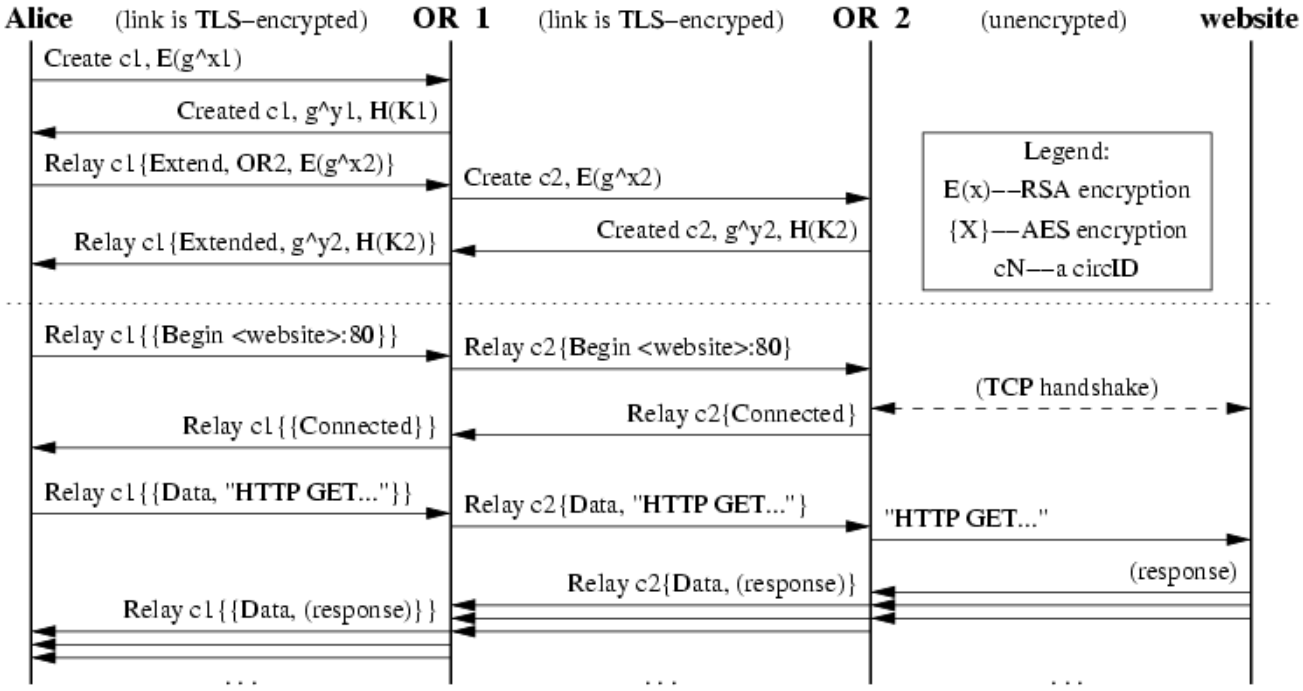


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

OP picks the route

- First picks the exit node E such that E's exit policy includes at least one pending stream that needs a circuit
- Choose N-1 distinct nodes (default is three total nodes), with some order
- Open a connection to the first (guard) node, negotiate session keys
- extend the circuit incrementally over the remaining N-1 nodes

Tor uses PKC to protect negotiation of a session key

- One hop at a time over an encrypted and authenticated channel
  - TLS: use *identity keys* to sign certs
- Use public-key cryptography (PKC) over this channel to set up an ephemeral session key
  - PKC is RSA (legacy) or Curve25519: use short-term *onion keys*
  - symmetric is AES, set up with DHE (legacy) or ECDHE
- Once ephemeral keys are set up OP layers them, and ORs unwrap them

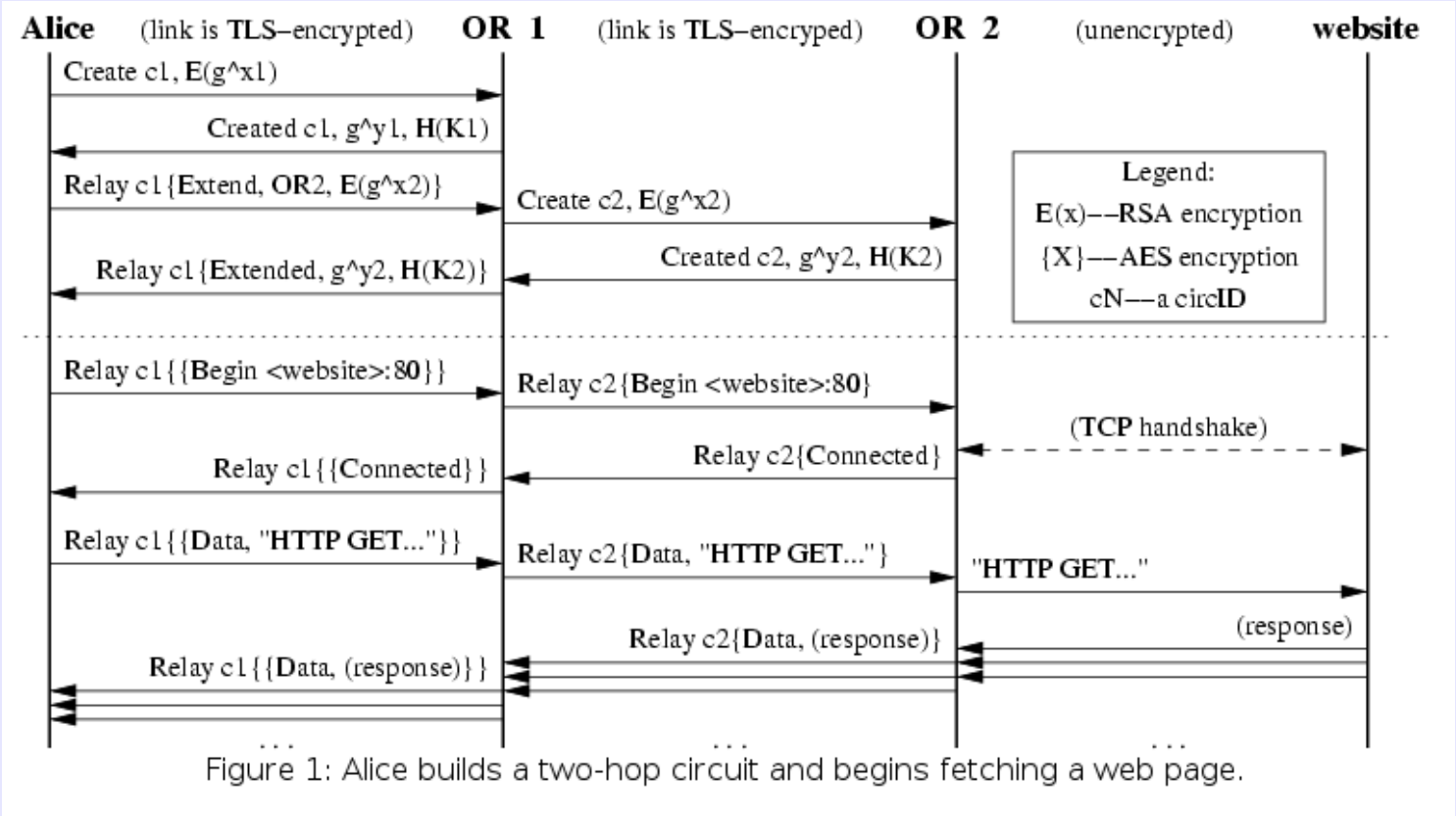


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

Discussion

- We have a public key for the guard. What reason did I give to not just use PKC to encrypt communications?

Session keys are negotiated using Diffie-Hellman Key Exchange

- First published in 1976; still around
- Alice and Bob want to share a secret key for use in a symmetric cipher. Every piece of information that they exchange is observed by their adversary Eve. How is it possible for Alice and Bob to agree on a key without making it available to Eve?

Diffie-Hellman

Publicly choose:

- a safe large prime *p* (e.g. Tor docs use rfc2409 section 6.2. But see Logjam)
- g, a primitive root mod p, with *2 <= g <= p-2*

Secretly generate:

- Alice and Bob randomly choose secret integers 1 <= *x, y <= p-2* respectively

$A := g^x \pmod p$

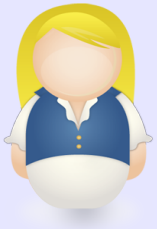
$K_B := A^y \pmod p$

$K_A := B^x \pmod p$

$B := g^y \pmod p$

$$K_A = (g^y)^x = (g^x)^y = K_B \text{ is the key}$$

Diffie-Hellman



$x = \log_g A \pmod p$

$y = \log_g B \pmod p$

Diffie-Hellman

$x = \log_g A \pmod p$

$y = \log_g B \pmod p$

Discrete Log Problem is in NP

Diffie Hellman Problem is no harder than DL problem; there is no proof of the converse

Question

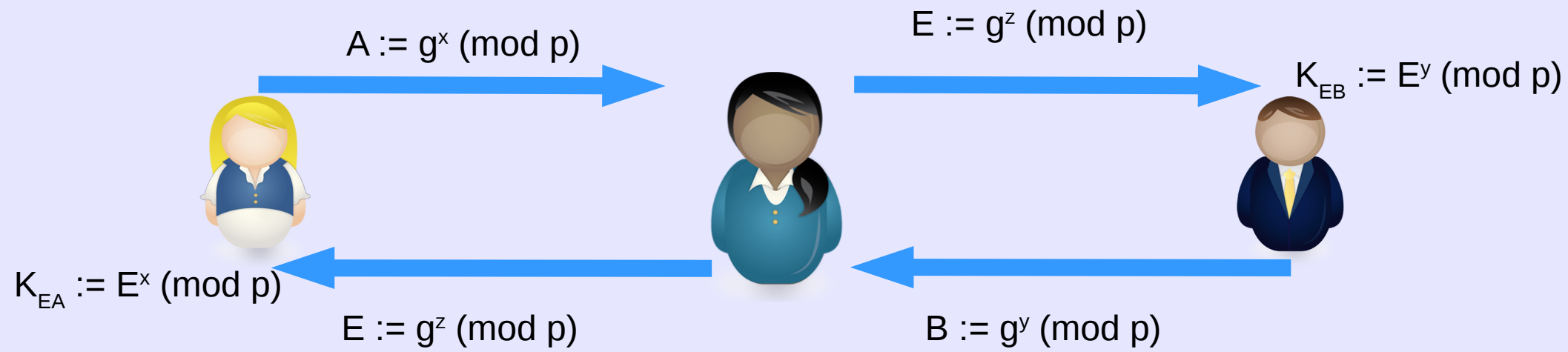
- Ian Goldberg remarked that a good way to fight mass surveillance by a global passive adversary would be to “do a quick Diffie-Hellman” by default when setting up otherwise unprotected connections. He notes that this won’t help against an active attack. Can you guess what he means by an active attack?

MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with *2 <= g <= p-2*

Secretly generate:

- Alice and Bob choose secret integers 0 <= *x, y <= p-2* respectively
- Eve picks her own secret integer, *z*



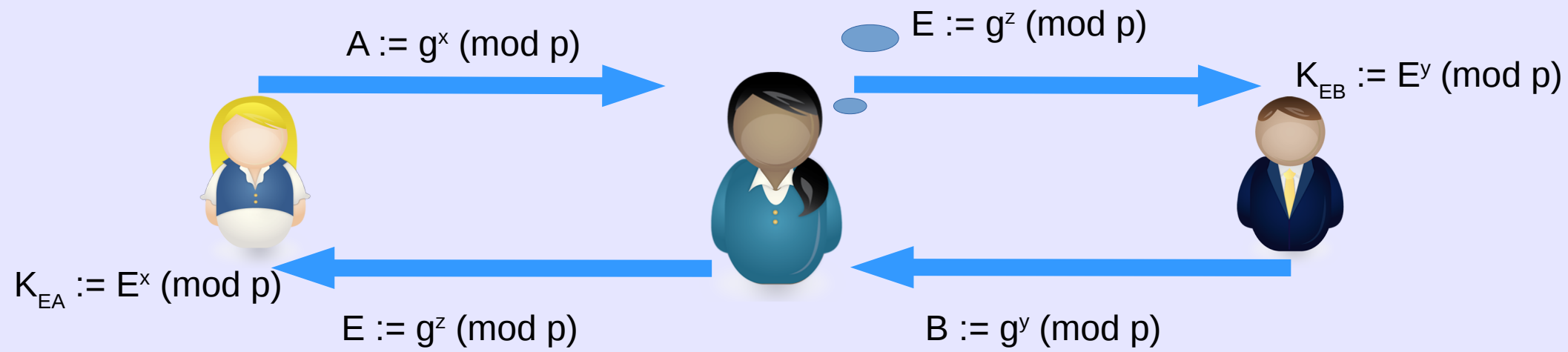
MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with *2 <= g <=*

$K_{EA} := g^{xz} \pmod p$
$K_{EB} := g^{yz} \pmod p$

Secretly generate:

- Alice and Bob choose secret integers 0 <= *x, y <= p-2* respectively
- Eve picks her own secret integer, *z*



Tor strengths and weaknesses

- Strengths
  - faster than mixnets
  - perfect forward secrecy
  - easy to run nodes, easy to use as a client: adds to security
  - bridges, pluggable transports for censorship circumvention
  - sandboxing
- Weaknesses
  - traffic analysis by a pervasive passive adversary
  - end-to-end timing attacks
  - content is revealed to exit node
  - blockable exit nodes

Who runs exit nodes?

- Universities (as of Oct 2017) (link)
  - MIT+, Michigan, CMU, UNC, Karlsruhrer IT, Stanford, Clarkson, U. Washington, Utah+, Caltech, RIT+, Bowdoin, Northeastern+, Princeton
- Bad people too! (Why might they do that?)
- Not Arizona :-(
  - yet :-)

Who runs hidden services?

- Propublica, Duckduckgo, Facebook, Scihub, Riseup, Protonmail, Debian, Whonix, The Intercept, Wikileaks, Securedrops for The Freedom of the Press Foundation , The Guardian, The Associated Press, NY TImes, USA Today, Washington Post, etc., TORCH (these are all onion links)
- A bunch of illegal stuff
- Hidden services are easy to set up (demo)
  - even inside firewalled networks

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Fun with mitmproxy (demo)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Communication Privacy

Cryptography is useful when it is difficult to secure a channel

- Confidentiality in FF voice communication requires that no unwanted third party is listening in
- However hard that is, phone communication presupposes it too. In addition, it requires that the call isn't intercepted while in transit
- Interception: Face to face < Copper wire < Radio link < Optical link (**POTL** 11)

Is privacy easier to achieve if privacy failure is easier to detect?

- Is it harder to read a letter surreptitiously over someone's shoulder than to listen to a conversation surreptitiously?
  - 2 minutes then rejoin

Is privacy is easier to achieve if violating privacy is easier to detect?

- If attackers go for the stealthiest option, the greater threat to letter communication is interception of a letter in transit
- One way to go: secure the channel: US postal service. Remember the history there in colonial America?
- Another way: encrypt the communications
  - notice you still have a part of the channel to secure (think back to FF case, and our “Layer 8+” discussion)

Tamper detection in electronic communication is hard

- Envelopes (weakly) detect tampering in written communication
- there is no analog for encrypted communications
  - can check authenticity and integrity, though
  - also, see QKD

Tamper proof key distribution + one time pads

- OTP create *key management* problems, as we'll see. QKD helps with that
- "A hub-and-spoke network has been operated by Los Alamos National Laboratory since 2011. All messages are routed via the hub. The system equips each node in the network with quantum transmitters—i.e., lasers—but not with expensive and bulky photon detectors. Only the hub receives quantum messages. To communicate, each node sends a one-time pad to the hub, which it then uses to communicate securely over a classical link. The hub can route this message to another node using another one time pad from the second node." (link)

One-time pad systems (OTP) are information-theoretically secure, subject to some conditions

- Key is as long as the message
- Key is random
- Key is secret
- Key is not reused
  - Creates a key distribution problem

OTP systems create 101 key-management problems

- Encode the time and place of an event as 8 two-digit decimal numbers
- YYYY MM DD hh mm NS EW, where Y:=year, M:= month, D:=day, h:=hour, m:=minute, NS:=north-south street number, EW:=east-west street number
- Say the message is

  19 99 12 30 15 25 01 44

OTP systems create 101 key-management problems

- Say the mesage is

  19 99 12 30 15 25 01 44
- Key is a random set of 8 two-digit numbers

  64 25 83 09 76 23 55 72
- Add the key to the message, “forgetting any carrying” (i.e. add in $\mathbb{Z}_{10}$):

  19 99 12 30 15 25 01 44

  64 25 83 09 76 23 55 72

  ---

  73 14 95 39 81 48 56 16

OTP systems create 101 key-management problems

- No one without the key can decrypt the message, there isn't enough information for a ciphertext-only attack
- Suppose the message and the key were, respectively,

  20 00 01 11 10 45 05 23, and

  53 14 94 28 71 03 51 93,

  then the ciphertext would be the same

OTP systems create 101 key-management problems

- OTP protects against ciphertext-only attacks; known plaintext attacks are another story
- Say the event you're encoding happens on 12/30/1999 at 3:25 on the corner of 1st and 44th, and Eve has the ciphertext
- She subtracts to get the key, "forgetting borrowing" (i.e. subtract in $\mathbb{Z}_{10}$):

  73 14 95 39 81 48 56 16

  19 99 12 30 15 25 01 44

  ---

  64 25 83 09 76 23 55 72

OTP systems create 101 key-management problems

- Lesson: OTP is only as secure as the key management protocols that go with it
- This can be an organizational nightmare
  - Leave it to the Soviets to use...central planning

For reasons that are still unclear, a serious mistake was made in the early months of 1942. Rather than making exactly two copies of the key sheets, they made four. These excess keys then entered the inventory and remained in use for several years. Western intelligence noted and exploited the multiple use of the keys, with disastrous results for Soviet security. Under the code name Venona, cryptanalytic study of the reused “one-time” keys went on for decades. The system was used for the most sensitive Soviet information, and the Americans and the British studied it in hopes of identifying Soviet “moles” thought to be operating at the highest levels of their intelligence establishments. (POTL 19)

OTP systems create 101 key-management problems

- Lesson: OTP is only as secure as the key management protocols that go with it
- This can be an organizational nightmare
  - Leave it to the Soviets to use...central planning
- (Enter QKP)