6.857 - Network & Computer Security

2/15/12
L3.1

Reminders: Mon = 66-110 Wed = 56-114
http://courses.csail.mit.edu/6.857

Administrivia: Sign up on line if you haven't yet.
Pset #1 now posted.
No recitation this week.

Today:
- ☐ Finish material from lecture 1 (notes L1.5 - L1.8)
- ☐ Encryption
- ☐ Perfect secrecy
- ☐ One-time pad (OTP)

News:
- "Traveling Light in a Time of Digital Thievery" NYT 2/11/12
- "Ron was wrong, Whit is right" (Lenstra et al.) IACR eprint 2012/064, 2/14/12
- "Freedom to Tinker: There's no need to panic over factorable keys" 2/15/12 by N. Heninger

Reading: (highly recommended)
Katz/Lindell chapters 1, 2, 3

Who is adversary? (Know your enemy!)

- may be insider/outsider, vendor, ...

  Examples: Voter may wish to sell her vote.
  Election official may be corrupt.
  Vendor may install "backdoor" in system.
  Eavesdropper may manipulate communications.

- what does adversary know?

  Examples: system design & implementation details
  passwords
  facebook profiles of all personnel

- what resources does adversary have?

  Examples:
  - large computers
  - ability to intercept & modify all communications
  - ability to corrupt some participants
    (e.g. pay TV subscriber, voter, server, ...)

We typically make generous assumptions about adversary's abilities.

Vocab:

"vulnerability" = weakness that might be exploited by an adversary
(e.g. poor password, buffer overflow possibility)

"threat" = potential violation of security policy
(e.g. by exploiting a vulnerability)

"risk" = likelihood that threat will materialize

"risk management" = balancing one risk against another, or other factors, such as cost, ease-of-use, understandability, availability, ...

No mechanism is perfect — we build fences, not impenetrable walls
(how high is fence?)

Security mechanisms may involve:

- identification of principals (e.g. "user name")
- authentication of principals (e.g. password, biometric)
- authorization: checking to see if principal is authorized for requested action
- physical protection: locks, enclosures
- cryptography: math in service of security (hard computational problems)
- economics: (note model change here: parties are self-interested, e.g. spammer...)
- deception: to get adversary to reveal himself or waste his efforts (e.g. honeypot)
- randomness, unpredictability: e.g. for passwords & crypto keys

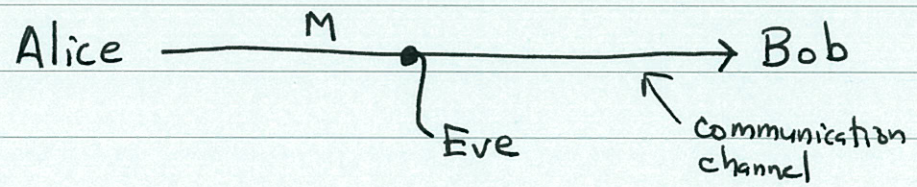
Some principles:

- be sceptical & paranoid
- don't aim for perfection
  ("there are no secure systems, only degrees of insecurity..."
- tradeoff cost / security
  ("to halve the risk, double the cost..." – Adi Shamir)
- be prepared for loss
- "KISS" ("keep it simple, stupid!")
- ease of use is important
- separation of privilege – require 2 people to perform action
- defense in depth (layered defense)
- complete mediation (all requests checked for authorization)
- least privelege (don't give some more permissions than they need)
- education
- transparency (no security through obscurity)

Encryption

Goal: confidentiality of transmitted (or stored) message

Parties: Alice, Bob — "good guys"
Eve — "eavesdropper", "adversary"



M = transmitted message

In basic picture above, there is nothing to distinguish Bob from Eve; they both receive message.

Could have dedicated circuits (e.g. helium-filled pipes containing fiber optic cable... ?) or steganography.

Crypto approach:
- Bob knows a key K that Eve doesn't
- Alice can encrypt message so that knowledge of K allows decryption.
- Eve hears ciphertext, but learns "nothing" about M.

With classical (non public key) crypto, Alice & Bob both know key K.

Algorithms:

$K \leftarrow Gen(1^\lambda)$ — generate key of length λ (λ given in unary)

$C \leftarrow Enc(K, M)$ — encrypt message M with key K, result is ciphertext C

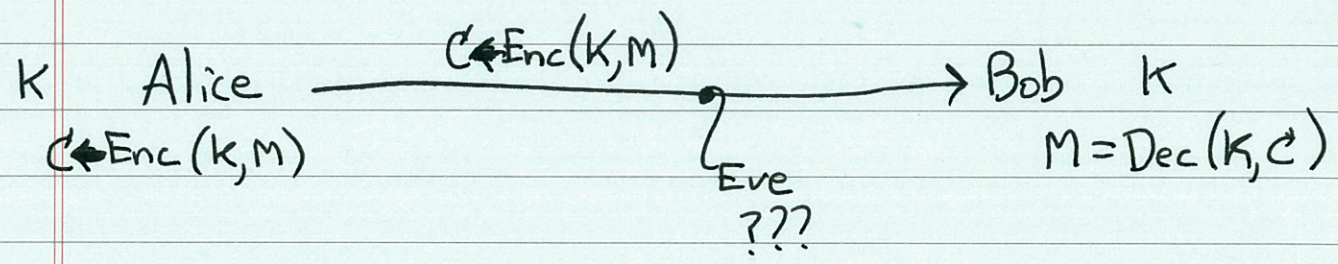
$M = Dec(K, C)$ — decrypt C using K to obtain M

(Note Katz/Lindell convention: "$\leftarrow$" for randomized operations, $=$ for deterministic ones

Often $\xleftarrow{R}$ or $\xleftarrow{\$}$ is used for randomized operation.)

Setup: Someone computes $K \leftarrow Gen(1^\lambda)$
(Someone may be Alice, or Bob)
Ensures that Alice & Bob both have K (and Eve doesn't) (how!?)

Communication:



Security objective:

Eve can't distinguish $Enc(K, M_1)$ from $Enc(K, M_2)$, even if she knows (or chooses) M_1 and M_2 $(M_1 \neq M_2)$ (of the same length).

(Encryption typically does not hide message length.)

Attacks:

- known ciphertext
- known CT/PT pairs
- chosen PT
- chosen CT
- ...

} assumes K is re-used

One-Time Pad (OTP)

- Vernam 1917 paper-tape based. Patent.

- Message, key, and ciphertext have same length (λ bits)
- Key K also called <u>pad</u>; it is random & known only to Alice & Bob.
  (Note: used by spies, key written on small pad...)

- <u>Enc:</u>

$$\begin{array}{rl} M = & 101100\cdots \quad \text{(binary string)} \\ \oplus\ K = & 011010\cdots \quad \text{(mod-2 each column)} \\ \hline C = & 110110\cdots \end{array}$$

- <u>Dec:</u> Just add K again: $(m_i \oplus k_i) \oplus k_i = m_i$

<u>Joke:</u> (Desmedt Crypto rump session)
OTP is <u>weak</u>, it only encrypts 1/2 the bits! leakage!
Better to change them <u>all</u>!

<u>Theorem:</u> OTP is unconditionally secure.
(Secure against Eve with unlimited computing power.)
a.k.a. <u>information-theoretically secure</u>.

$P(M)$ = probability of message M
(Eve's prior probability, in Bayesian sense)

$P(M|C)$ = probability of message M, having seen C
(Eve's posterior probability, in Bayesian sense)

security of OTP $\equiv$ $\left(P(M) = P(M|C)\right)$

Adversary learned nothing about M by having seen C.

so: $P(K) = 2^{-\lambda}$ for λ-bit key

$P(C)$ = probability of a given ciphertext C

$$= \sum_M \underbrace{P(C|M)}_{\text{Lemma}} \cdot P(M)$$

$$= \sum_M 2^{-\lambda} \cdot P(M)$$

$$= \left(\sum_M P(M)\right) \cdot 2^{-\lambda} = 2^{-\lambda}$$ <u>uniform.</u>

<u>Lemma:</u> $P(C|M)$ = prob of C, given M

$$= P(K = C \oplus M)$$

$$= 2^{-\lambda}$$

Now $P(m|c)$ = Prob of M, given C

$$= \frac{P(C|M) \cdot P(M)}{P(C)}$$ Bayes' Rule

$$= \frac{2^{-\lambda} \cdot P(M)}{2^{-\lambda}}$$

$$= P(M)$$ QED

This is perfect secrecy! (except for length λ of M)

Compare to computational security: assumes adversary has limited computational power (e.g. can't factor large numbers)

Users need to:
- generate large secrets
- share them securely
- keep them secret
- avoid re-using them!
  (google "Venona")

$$C_1 \oplus C_2 = (M_1 \oplus K) \oplus (M_2 \oplus K)$$
$$= M_1 \oplus M_2$$

Also: OTP is malleable!

Changing bit i of C causes decrypted bit i to change.