Course Information

CSE 207 — Modern Cryptography

Instructor: Mihir Bellare

Website: `http://cseweb.ucsd.edu/~mihir/cse207`

Cryptography usage

Did you use any cryptography today?

Cryptography usage

- https invokes the TLS protocol
- TLS uses cryptography
- TLS is in ubiquitous use for secure communication: shopping, banking, Netflix, gmail, Facebook, ...

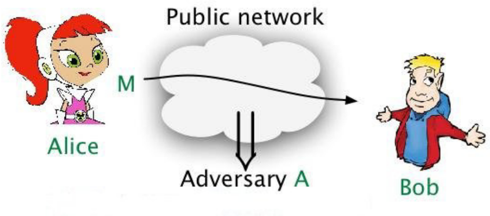
Cryptography usage

Other uses of cryptography:

- ATM machines
- Bitcoin
- Messaging apps: whatsapp, viber, line, telegraph, goldbug, chatsecure, ...
- Google authenticator
- ...

11,748 android apps use cryptography (encryption), and 10,327 get it wrong [EBFK13]

What is cryptography about?



Adversary: clever person with powerful computer

Security goals:

- **Data privacy:** Ensure adversary does not see or obtain the data (message) M.
- **Data integrity and authenticity:** Ensure M really originates with Alice and has not been modified in transit.

Example: Medical databases

Doctor | Database

$\xrightarrow{\text{Get Alice}}$

Alice	F_A
Bob	F_B

$\xleftarrow{F_A}$

Reads F_A

Modifies F_A to F'_A

$\xrightarrow{\text{Put: Alice, } F'_A}$

Alice	F'_A
Bob	F_B

Example: Medical databases

Doctor

Database

Get Alice $\longrightarrow$

F_A $\longleftarrow$

Alice	F_A
Bob	F_B

Reads F_A

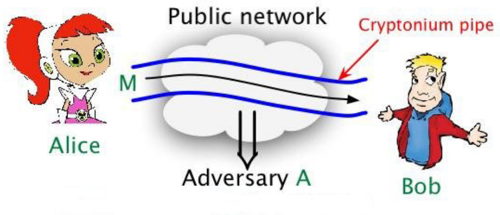
Modifies F_A to F'_A

Put: Alice, F'_A $\longrightarrow$

Alice	F'_A
Bob	F_B

- Privacy: F_A, F'_A contain confidential information and we want to ensure the adversary does not obtain them
- Integrity and authenticity: Need to ensure
  - doctor is authorized to get Alice's file
  - F_A, F'_A are not modified in transit
  - F_A is really sent by database
  - F'_A is really sent by (authorized) doctor

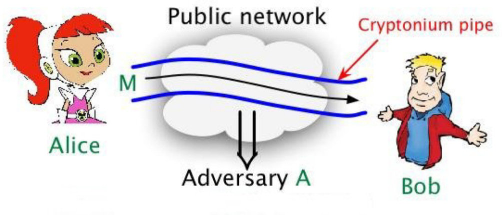
Ideal World



Cryptonium pipe: Cannot see inside or alter content.

All our goals would be achieved!

Ideal World

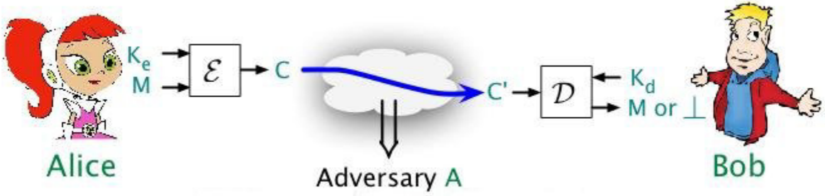


Cryptonium pipe: Cannot see inside or alter content.

All our goals would be achieved!

But cryptonium is only available on planet Crypton and is in short supply. ☹

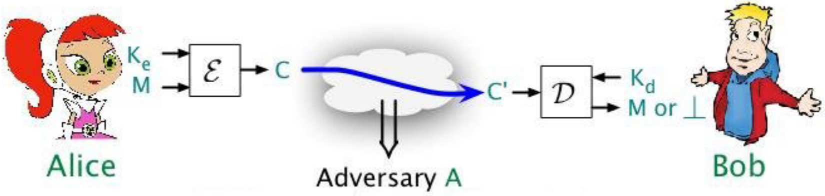
Cryptographic schemes



$\mathcal{E}$: encryption algorithm $\quad$ K_e: encryption key

$\mathcal{D}$: decryption algorithm $\quad$ K_d: decryption key

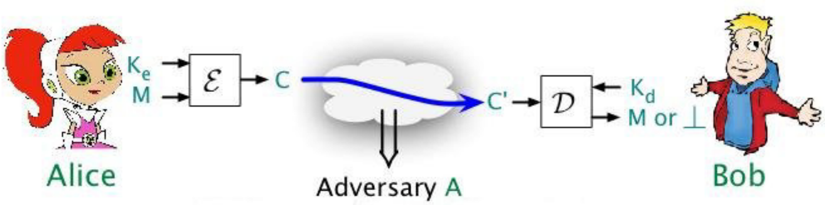
Cryptographic schemes



$\mathcal{E}$: encryption algorithm K_e: encryption key
$\mathcal{D}$: decryption algorithm K_d: decryption key

Algorithms: standardized, implemented, public!

Cryptographic schemes



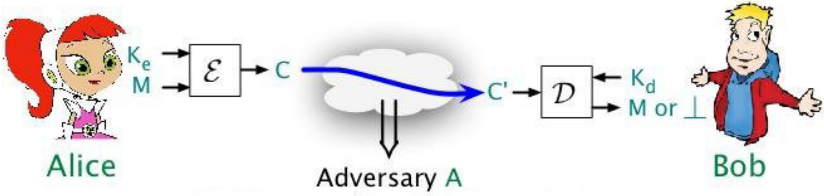
$\mathcal{E}$: encryption algorithm $\quad$ K_e: encryption key

$\mathcal{D}$: decryption algorithm $\quad$ K_d: decryption key

Settings:

- public-key (assymmetric): K_e public, K_d secret
- private-key (symmetric): $K_e = K_d$ secret

Cryptographic schemes

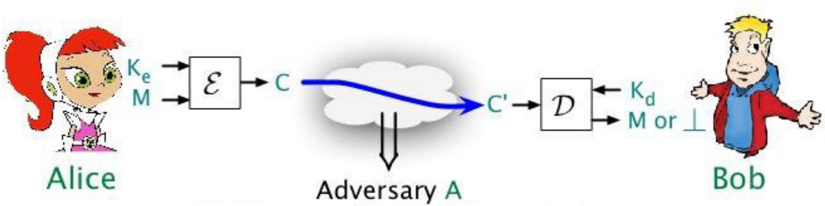


$\mathcal{E}$: encryption algorithm $\quad$ K_e: encryption key

$\mathcal{D}$: decryption algorithm $\quad$ K_d: decryption key

How do keys get distributed? Magic, for now!

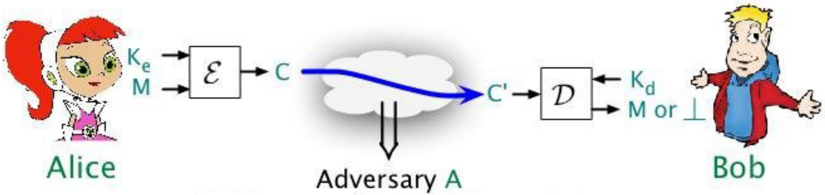
Cryptographic schemes



Our concerns:

- How to define security goals?
- How to design $\mathcal{E}$, $\mathcal{D}$?
- How to gain confidence that $\mathcal{E}$, $\mathcal{D}$ achieve our goals?

Cryptographic schemes



Computer Security: How does the computer/system protect K_e/K_d from break-in (viruses, worms, OS holes, ...)? (CSE 127,227)

Cryptography: How do we use K_e, K_d to ensure security of communication over an insecure network? (CSE 107,207)

Why is cryptography hard?

- One cannot anticipate an adversary strategy in advance; number of possibilities is infinite.
- "Testing" is not possible in this setting.

Early history

Substitution ciphers/Caesar ciphers:

$$K_e = K_d = \pi : \Sigma \to \Sigma, \text{a secret permutation}$$

e.g., $\Sigma = \{A, B, C, \ldots\}$ and π is as follows:

σ	A	B	C	D	$\cdots$
$\pi(\sigma)$	E	A	Z	U	$\cdots$

$$\begin{aligned} \mathcal{E}_\pi(CAB) &= \pi(C)\pi(A)\pi(B) \\ &= Z\ \ E\ \ A \\ \mathcal{D}_\pi(ZEA) &= \pi^{-1}(Z)\pi^{-1}(E)\pi^{-1}(A) \\ &= C\ \ A\ \ B \end{aligned}$$

Early history

Substitution ciphers/Caesar ciphers:

$$K_e = K_d = \pi : \Sigma \to \Sigma, \text{a secret permutation}$$

e.g., $\Sigma = \{A, B, C, \ldots\}$ and π is as follows:

σ	A	B	C	D	$\cdots$
$\pi(\sigma)$	E	A	Z	U	$\cdots$

$$\begin{aligned} \mathcal{E}_\pi(CAB) &= \pi(C)\pi(A)\pi(B) \\ &= Z \ \ E \ \ A \\ \mathcal{D}_\pi(ZEA) &= \pi^{-1}(Z)\pi^{-1}(E)\pi^{-1}(A) \\ &= C \ \ A \ \ B \end{aligned}$$

Not very secure! (Common newspaper puzzle)

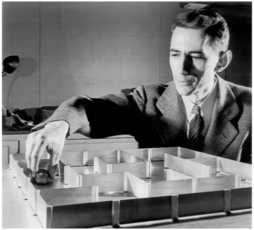
The age of machines

Enigma: German World War II machine

Broken by British in an effort led by Turing

Shannon and One-Time-Pad (OTP) Encryption

$$K_e = K_d = \underbrace{K \stackrel{\$}{\leftarrow} \{0,1\}^k}_{\substack{K \textit{ chosen at random} \\ \textit{from } \{0,1\}^k}}$$

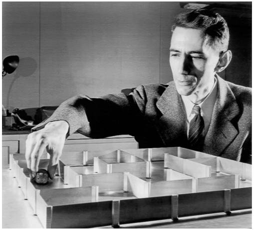


For any $M \in \{0,1\}^k$

- $\mathcal{E}_K(M) = K \oplus M$
- $\mathcal{D}_K(C) = K \oplus C$

Shannon and One-Time-Pad (OTP) Encryption

$$K_e = K_d = \underbrace{K \xleftarrow{\$} \{0,1\}^k}_{K \textit{ chosen at random from } \{0,1\}^k}$$



For any $M \in \{0,1\}^k$

- $\mathcal{E}_K(M) = K \oplus M$
- $\mathcal{D}_K(C) = K \oplus C$

Theorem (Shannon): OTP is perfectly secure as long as only one message encrypted.

"Perfect" secrecy, a notion Shannon defines, captures mathematical impossibility of breaking an encryption scheme.

Fact: if $|M| > |K|$, then no scheme is perfectly secure.

Modern Cryptography: A Computational Science

Security of a "practical" system must rely not on the impossibility but on the computational difficulty of breaking the system.

("Practical" = more message bits than key bits)

Modern Cryptography: A Computational Science

Rather than:

"It is impossible to break the scheme"

We might be able to say:

"No attack using $\leq 2^{160}$ *time succeeds with probability* $\geq 2^{-20}$*"*

I.e., Attacks can exist as long as cost to mount them is prohibitive, where Cost = computing time/memory, \$\$\$

Modern Cryptography: A Computational Science

Security of a "practical" system must rely not on the impossibility but on the computational difficulty of breaking the system.

Cryptography is now not just classical mathematics; it needs to draw on computer science

- Computational complexity theory (CSE 105,200)
- Algorithm design (CSE 101,202)

Defining security

Being able to precisely (formally, mathematically) state what is the security goal of a design is challenging but important.

We will spend a lot of time developing and justifying strong, precise notions of security.

Thinking in terms of these precise goals and understanding the need for them may be the most important thing you get from this course!

Defining Security

What does it mean for an encryption scheme to provide privacy?

Does it mean that given $C = \mathcal{E}_{K_e}(M)$, adversary cannot

- recover M?
- recover the first bit of M?
- recover the XOR of the first and the last bits of M?
- ...

We will provide a formal definition for privacy, justify it, and show it implies the above (and more).

Lego Approach

We typically design high-level primitives from atomic ones

$$\boxed{\text{Atomic primitive}}$$
$$\downarrow$$
$$\boxed{\text{Transformer}}$$
$$\downarrow$$
$$\boxed{\text{High-level primitive}}$$

Atomic Primitives or Problems

Examples:

- Factoring: Given large $N = pq$, find p, q
- Block ciphers: DES, AES, ...
- Hash functions: MD5, SHA1, SHA3, ...

Features:

- Few such primitives
- Confidence by cryptanalysis.

Drawback: Don't directly solve any security problem.

Higher Level Primitives

Goal: Solve security problem of direct interest.

Examples: encryption, authentication, digital signatures, session-key distribution, ...

Provable security

Enables us to get transformers for which we can guarantee

Atomic primitive secure $\Rightarrow$ High-level primitive secure

Proven-secure schemes in use (TLS, SSH, IPSec, . . .):

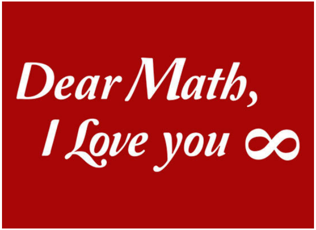
- HMAC
- OAEP
- ECIES
- . . .

This is a theory course! Largely definitions and proofs, although of applied value.

Needed: algorithms, theory of computation, probability theory, a little calculus, and

MATHEMATICAL MATURITY

Which is you?



Enrollment

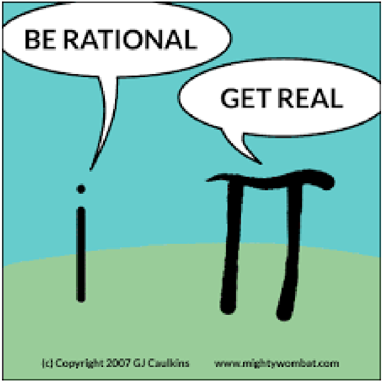
Formal pre-requisites:

((CSE 202) AND (CSE 200 OR CSE 227)) OR CSE 107

- All students start on waitlist
- If you meet above pre-requisite condition, email instructor to be cleared
- Else, Quiz 1 results will be used as assessment
- Quiz 1 goes out Tuesday 9/27, due Thursday 9/29

Make sure you enroll for backup courses in case you do not get in to CSE 207!

Math Test



Don't get it? Either your **pre-reqs** or your **sense of humor** need work.

Rules and grading

- Each student must do their Quiz on their own
- Looking at solutions from previous years of the course or finding them on the Internet is not allowed.
- Non-compliance with rules is reported to UCSD Academic Integrity Office and can result in dismissal.

- Grader expects neat, mathematically precise and well-written solutions. Quality of exposition will impact score.
- Doesn't work to come back and say "You did not understand what I meant." You have to write in proper mathematical language so that your meaning is clear. You are graded on what you write, not on what you think is in your head.

Administrative

- READ course information sheet
- Then READ, sign and return affirmation

Resources:

- Lecture slides: The ONLY source of materiel
- Course notes on webpage: IGNORE!
- Homework/Quiz solutions
- No textbook

All resources on course web page. Some handed out.