{ CSCI 6331 · 4331 | Lecture 1 }

Cryptography

Hoeteck Wee · hoeteck@gwu.edu

http://tinyurl.com/cryptogw/

Q. How often do you use cryptography?

A. Whenever you use the Internet, cell-phone, type `https://`, ...

- WEP/WPA algorithms for encrypting wifi traffic

  (between mobile station and access point)
- SSL/TLS algorithms to prevent eavesdropping and tampering

  (e.g. between web/email server and client)
- A3/A8 algorithms used in GSM cell phones

Course Objectives

1. reasoning about security of cryptographic constructions
   - "Is *blah* secure?"
   - "What does it mean to be *secure*?"
2. applying this knowledge to real-world applications
   - why every engineer should know some crypto

Topics.

- basic symmetric-key encryption (one-time pads, block ciphers, ...)
- public-key cryptography (encryption, digital signatures, authentication)
- real world crypto (SSL/TLS, IPsec, ...)

Administration

- Assessment.
  - Homeworks, In-class exercises, Programming assignments, Final projects, Examinations, Presentations, Participation [details to be announced]
  - Homework 1 due Tues Jan 31 (no late submissions)
- Course Website.
  - Main site: `http://tinyurl.com/cryptogw/`
  - Homeworks, discussions, etc: Piazza
- Prerequisites.
  - discrete math, probability, algorithms, proofs
- Lecture Time.
  - 5.55 - 8.25 pm? (instead of 6.10 - 8.40 pm) [take a vote]

Private-Key Encryption

classic problem. secret communication between two parties

- attack model: eavesdropping adversary over a public channel (no tampering)
- goal: exchange message while hiding from adversary

private-key. share secret information in advance (a.k.a. symmetric-key)

- both parties know secret key k
- adversary does not know k
- question: where does k come from?

Private-Key Encryption

classic problem. secret communication between two parties

- attack model: eavesdropping adversary over a public channel
- goal: exchange message while hiding from adversary

syntax. private-key encryption = three algorithms $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$

  - key generation Gen – probabilistic algorithm outputs a key k
  - encryption Enc – input key k and a message m; output ciphertext $\mathrm{c} = \mathrm{Enc}_{\mathrm{k}}(\mathrm{m})$
  - deryption Dec – input key k and a ciphertext c; output plaintext $\mathrm{m} = \mathrm{Dec}_{\mathrm{k}}(\mathrm{c})$

- correctness: for all keys k and all messages m, $\mathrm{Dec}_{\mathrm{k}}(\mathrm{Enc}_{\mathrm{k}}(\mathrm{m})) = \mathrm{m}$
- notation: key space $\mathcal{K}$, message/plaintext space $\mathcal{M}$ and ciphertext space $\mathcal{C}$

observation. must hide k from adversary (why?)

- question: do we need to hide the algorithm Dec?

Adversarial Model

Kerckhoffs' Principle (1883). encryption scheme is public, only the key k is secret

- i.e. security should rely solely on the secrecy of the key
- reason #1: easier to maintain secrecy of keys than of algorithms
  (keys are shorter and easier to store; cannot reverse-engineer)
- reason #2: easier to change/refresh keys than algorithms/software
  (in case of key exposure; regular updates good security practice)
- reason #3: easier for many people to share same algorithm/software (than keys)

Modern Interpretation. advocating "open cryptographic design"

- public scrutiny, flaws are detected by "ethical hackers"/academics
- enables establishment of standards e.g. DES, AES, SSL
- failure of "security via obscurity", e.g. Intel's HDCP, GSM A5/1.

Adversarial Model

Question. What does the adversary know?

- knows all algorithms $\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}$ and message space $\mathcal{M}$
- e.g. $\mathcal{M}$ = "attack on J$\times$n $1\times$ at $\times\times$: $\times\times$".

Question. What are the capabilities of the adversary?

- cipertext-only attack: adversary observes a single ciphertext
- known-plaintext attack: adversary learns multiple plaintext-ciphertext pairs
- chosen-plaintext attack: adversary obtains encryption of plaintexts of its choice
- chosen-ciphertext attack: adversary obtains decryptions of ciphertexts of its choice

Next. Review historical ciphers, then formalize security.

Caesar Cipher

Caesar Cipher. a shift cipher

- Gen outputs $\mathrm{k} \in \{0, 1, 2, \ldots, 24, 25\}$ (chosen at random)
- Example: $\mathrm{k} = 3$. To encrypt, replace `A` by `D`, `B` by `E`, `C` by `F`, ...
- Plaintext: `ATTACK AT DAWN`
- Ciphertext: `DWWDFN DW GDZQ`

Observation. Follows Kerckhoffs' Principle, but not a good cipher

- susceptible to brute force attack: test all possible keys to decrypt
- Caesar Cipher: $|\mathcal{K}| = 26$— need a larger key space
- How large? $|\mathcal{K}| \geq 2^{80}$

Substitution Cipher

Substitution Cipher.

- k is a look-up table: `A` $\to$ `Y`, `B` $\to$ `A`, `C` $\to$ `H`, `D` $\to$ `P`, ...
  (permutation on the alphabet)
- Plaintext: `ATTACK AT DAWN`
- Ciphertext: `YEEYHT YE PYDL`

Observation. Follows Kerckhoffs' Principle, but still not a good cipher

- What is $|\mathcal{K}|$? $|\mathcal{K}| = 26 \approx 2^{95}$
- Susceptible to frequency analysis
- Known letter distribution in English e.g. $\Pr[\mathtt{E}] = 0.13$; pairs of letters `th` vs `jj`
- Exploit the fact that mapping of plaintext letters to ciphertext letters is fixed.
- Challenge: decrypt `QEFP FP QEB CFOPQ QBUQ`

Perfect Cipher: Definition

Question. What type of security would we like in a perfect cipher?

Attempt. "Given the ciphertext, the adversary has no idea what the plaintext is"

- Impossible since the adversary might have a-prior information.
- e.g. $\mathcal{M}$ = "attack on J$\times$n $1\times$ at $\times\times$: $\times\times$".

Principle. "What happens in the ideal world?"

- ideal world = adversary sees nothing (i.e., no ciphertext)
- already knows some information about plaintext as revealed by $\mathcal{M}$, e.g. $|\mathcal{M}| = 1$.

Definition. a perfect cipher / perfect secrecy

- "ciphertext does not add information about the plaintext"
- $\Pr[\text{plaintext} = \mathrm{P} \mid \text{ciphertext} = \mathrm{C}] = \Pr[\text{plaintext} = \mathrm{P}]$

Notes.

- Probability is taken over choices of the key, the plaintext and the ciphertext
- Check: does this hold for substitution cipher?
  suppose $\mathcal{M}$ = all possible 2-letter combinations, equally likely

Perfect Cipher: Construction

One-Time Pad. one-bit messages

- $\mathcal{M} = \{0,1\}, \mathcal{K} = \{0,1\}$
- Gen outputs a random bit k, i.e. $\Pr[\mathrm{k}=0] = \Pr[\mathrm{k}=1] = 1/2$.
- $\mathrm{Enc}_{\mathrm{k}}(\mathrm{m}) = \mathrm{k} \oplus \mathrm{m}$
- What is $\mathrm{Dec}_{\mathrm{k}}(\mathrm{c})$? $\mathrm{Dec}_{\mathrm{k}}(\mathrm{c}) = \mathrm{k} \oplus \mathrm{m}$

Question. Is this secure?

- Suppose $\Pr[\mathrm{m}=1] = 0.8$. What is $\Pr[\mathrm{m}=1 \mid \text{ciphertext} = 1]$?

Perfect Cipher: Construction

One-Time Pad. ℓ-bit messages

- $\mathcal{M} = \{0,1\}^\ell, \mathcal{K} = \{0,1\}^\ell$
- Gen outputs a random ℓ-bit string k
- $\mathrm{Enc}_\mathrm{k}(\mathrm{m}) = \mathrm{k} \oplus \mathrm{m}$ (bit-wise XOR)

Shannon.

- One-Time Pad is a perfect cipher
- Disadvantage #1: needs a long key (in fact, necessary)
- Proof: ciphertext c can be an encryption of any plaintext m
  thus different key for each m
- Disadvantage #2: cannot reuse key