Introduction to Cryptography

Lecture 1

Benny Pinkas

Administrative Details

- Grade
  - Exam 75%
  - Homework 25% (might include programming)
- Office hours: Tuesday, 11-12.
- Email: benny@cs.haifa.ac.il
- Web page: http://www.pinkas.net/courses/itc/2005/index.html

- Goal: Learn the basics of modern cryptography
- Method: introductory, applied, precise.

Bibliography

- Textbook:
  - *Cryptography Theory and Practice, Second edition* by D. Stinson. (Also, a version in עברית by the Open University!).

- Optional:
  - *Handbook of Applied Cryptography,* by A. Menezes, P. Van Oorschot, S. Vanstone. (Free!)
  - *Introduction to Cryptography Applied to Secure Communication and Commerce*, by Amir Herzberg. (Free!)
  - *Applied Cryptography,* by B. Schneier.

In the Library

- In the “reserved books” section:

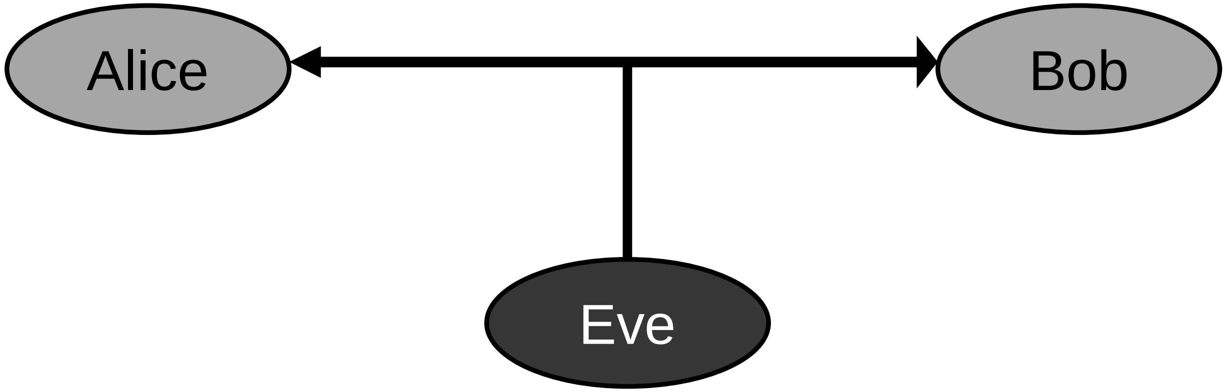
- Four copies of
  - Cryptography :theory and practice / Douglas R. Stinson
  - Introduction to cryptography :principles and applications /Hans Delfs, Helmut Knebl
  - Foundations of cryptography / Oded Goldreich

- One copy of
  - Handbook of applied cryptography / Alfred J. Menezes et al. *(also available online)*
  - Applied cryptography / Bruce Schneier

Course Outline

- Course Outline
  - Data secrecy: encryption
    - Symmetric encryption
    - Asymmetric (public key) encryption
  - Data Integrity: authentication, digital signatures.
  - Required background in number theory
  - Cryptographic protocols

Encryption

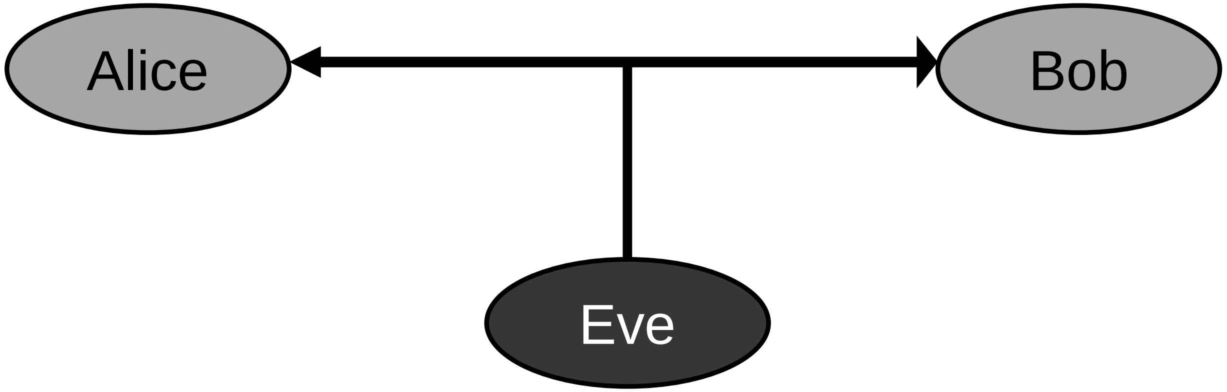


[illegible]

[illegible]

[illegible]

Authentication / Signatures



[illegible]

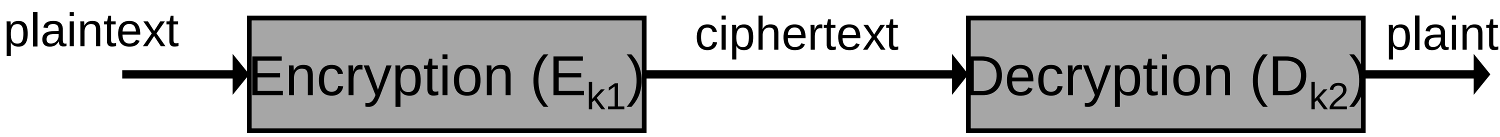
- [illegible]
- [illegible]

[illegible]

Encryption

- Message space *{m}*
- Encryption key k_1, *decryption key* k_2
- Key generation algorithm
- Encryption function *E*
- Decryption function *D*

Define the encryption system



- For every message *m*
  - $D_{k2}(E_{k1}(m)) = m$
  - I.e., the decryption of the encryption of *m* is *m*
- Symmetric encryption $k = k_1 = k_2$

Security Goals

(1) No adversary can determine *m*

or, even better,

(2) No adversary can determine any information about *m*

- Suppose *m* = *“attack on Sunday, October 17, 2004”.*
- The adversary can at most learn that
  - m = “attack on S**day, Oct**er 17, 2004”
  - m = “******   ** *u*****   *******   *** ****”

- Here, goal (1) is satisfied, but not goal (2)

Adversarial Model

- Adversary Knows encryption and decryption algorithms *E* and *D,* and *message space*.
- Kerckhoff’s Principle (1883):
  - The only thing Eve does not know is the secret key *k*
  - The design is public
    - Allows public scrutiny of the design
    - No need to replace the system if the design is exposed -> no need to keep the design secret
    - Same design can be used for multiple applications
    - Focus on securing the key
  - Examples
    - Security by obscurity, Intel’s HDCP ☹
    - DES, AES, SSL ☺

Adversarial Power

- Types of attacks:
  - Ciphertext only attack – ciphertext known to the adversary (eavesdropping)
  - Known plaintext attack – plaintext and ciphertext are known to the adversary
  - Chosen plaintext attack – the adversary can choose the plaintext and obtain its encryption (e.g. he has access to the encryption system)
  - Chosen ciphertext attack – the adversary can choose the ciphertext and obtain its decryption

- Assume restrictions on the adversary's capabilities, but not that it is using specific attacks or strategies.

Breaking the Enigma

- German cipher in WW II

- Kerckhoff’s principle
- Known plaintext attack
- (somewhat) chosen plaintext attack

Caesar Cipher

- A shift cipher

- Plaintext: “`ATTACK AT DAWN`”
- Ciphertext: “`DWWDFN DW GDZQ`”
- Key: $k \in_R \{0,25\}$. (In this example k=3)

- More formally:
  - Key: $k \in_R \{0…25\}$, chosen at random.
  - Message space: English text (i.e., $\{0...25\}^{|m|}$)
  - Algorithm: ciphertext letter = plaintext letter + k mod 25
- Kerckhoff’s principle
- Not a good idea

Brute Force Attacks

- Brute force attack : adversary tests all key space and checks which key decrypts the message

- Caesar cipher: |key space| = 26
- We need a large key space
- Usually, the key is a bit string chosen uniformly at random from $\{0,1\}^{|k|}$. Implying $2^{|k|}$ equiprobable keys.
- How long should *k* be?

- The adversary should not be able to do $2^{|k|}$ decryption trials

Adversary’s computation power

- Theoretically
  - Adversary can perform poly(|*k*|) computation
  - Key space = $2^{|k|}$
- Practically
  - |k| = 64 is too short
  - |k| = 80 starts to be reasonable
  - Why? (what can be done by 1000 computers in a year?)
    - 2^{55} = 2^{20} (ops per second)
    - x 2^{20} (seconds in two weeks)
    - x 2^{5} (≈ fortnights in a year) (might invest more than a year..)
    - x 2^{10} (computers in parallel)
- All this, assuming that the adversary cannot do better than a brute force attack

Monoalphabetic Substitution cipher

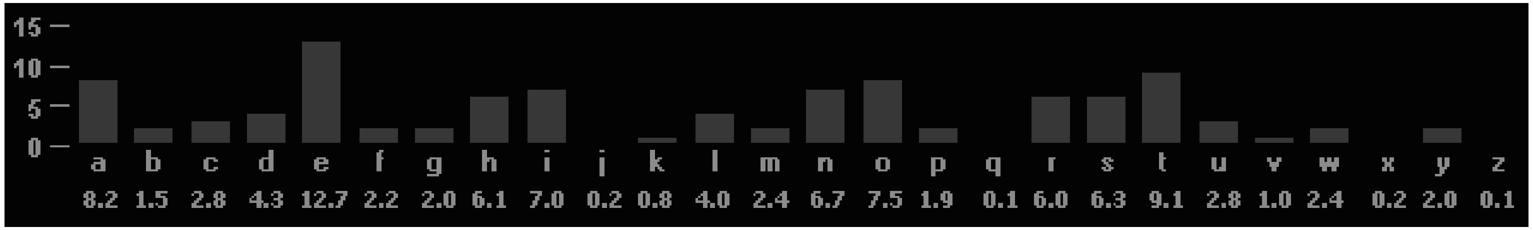
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L	R	C	V	M	U	E	K	J	D	I	X	N

- Plaintext: “`ATTACK AT DAWN`”
- Ciphertext: “`YEEYHT YE PYDL`”
- More formally:
  - Plaintext space = ciphertext space = $\{0..25\}^{|m|}$
  - Key space = 1-to-1 mappings of *{0..25} (i.e.,* permutations)
  - Encryption: map each letter according to the key

- Key space = 26! ≈ 4×10^{28} ≈ 2^{95}. (Large enough.)
- Still easy to break

Breaking the substitution cipher

- The plaintext has a lot of structure
  - Known letter distribution in English (e.g. Pr(“e”) = 13%).
  - Known distribution of pairs of letters (“th” vs. “jj”)



Cryptanalysis of a substitution cipher

- QEFP FP QEB CFOPQ QBUQ
- QEFP FP QEB CFOPQ QBUQ
- TH   TH   T T T
- THFP FP THB CFOPT TBUT
- THIS IS TH   I ST T T
- THIS IS THB CIOST TBUT
- THIS IS THE I ST TE T
- THIS IS THE FIRST TEXT

The Vigenere cipher

- Plaintext space = ciphertext space = $\{0..25\}^{|m|}$
- Key space = strings of |k| letters $\{0..25\}^{|K|}$
- Generate a pad by repeating the key until it is as long as the plaintext (e.g., “`SECRETSECRETSEC..`”)

- Encryption algorithm: add the corresponding characters of the pad and the plaintext
  - `THIS IS THE PLAINTEXT TO BE ENCRYPTED`
  - `SECR ET SEC RETSECRET SE CR ETSECRETSE`

- |Key space| = $26^{|k|}$. (k=17 implies |key space| ≈ 2^{80})
- Each plaintext letter is mapped to |*k*| different letters

Attacking the Vigenere cipher

- Known plaintext attack (or rather, known plaintext distribution)
  - Guess the key length |*k*|
  - Examine every |*k*|’th letter, this is a shift cipher
    - THIS IS THE PLAINTEXT TO BE ENCRYPTED
    - SECR ET SEC RETSECRET SE CR ETSECRETS
  - Attack time: |*k*| x |*k*| x time of attacking a shift cipher(1)

- Chosen plaintext attack:
  - Use the plaintext “aaaaaaa…”

(1) Can’t assume English plaintext. Can assume known letter frequency

Perfect Cipher

- What type of security would we like to achieve?
- "Given C, the adversary has no idea what M is"
  - Impossible since the adversary might have a-priori information

- In an "ideal" world, the message will be delivered in a magical way, out of the reach of the adversary
  - We would like to achieve similar security

- Definition: a *perfect cipher*
  - *Pr(plaintext = P | ciphertext = C) = Pr(plaintext = P)*

Perfect Cipher

- For a *perfect cipher*, it holds that given ciphertext *C,*
  - *Pr(plaintext = P | C) = Pr(plaintext = P)*
  - i.e., knowledge of ciphertext does not change the a-priori distribution of the plaintext
  - Probabilities taken over key space and plaintext space

  - One Time Pad (Vernam cipher): (for a one bit plaintext)
    - Plaintext $p \in \{0,1\}$
    - Key $k \in_R \{0,1\}$ (i.e. *Pr(k=0) = Pr(k=1) = ½*)
    - Ciphertext = $p \oplus k$

    - What happens if we know a-priori that *Pr(plaintext=1)=0.8* ?

The one-time-pad is a perfect cipher

- $\text{ciphertext} = \text{plaintext} \oplus k$

- $Pr\ (\ ciphertext = 1)$
- $= Pr\ (plaintext \oplus key = 1)$
- $= \ Pr\ (key = plaintext \oplus \ 1) = ½$

- $Pr(plaintext = 1 \mid ciphertext = 1)$
- $= Pr(plaintext = 1 \ \ \& \ \ ciphertext = 1) / Pr(ciphertext = 1)$
- $= Pr(plaintext = 1 \ \ \& \ \ ciphertext = 1) / ½$
- $= Pr(ciphertext = 1 \mid plaintext = 1) \cdot Pr(plaintext = 1) / ½$
- $= Pr(key = 0) \cdot Pr(plaintext = 1) / ½$
- $= ½ \cdot Pr(plaintext = 1) / ½$
- $= Pr(plaintext = 1)$

The one-time-pad

- Plaintext = $p_1p_2\ldots p_m \in \Sigma^m$ (e.g. $\Sigma=\{0,1\}$, or $\Sigma=\{A\ldots Z\}$)
- key = $k_1k_2\ldots k_m \in_R \Sigma^m$
- Ciphertext = $c_1c_2\ldots c_m$, $c_i = p_i \oplus k_i$
- Essentially a shift cipher with a different key for every character

- Shannon [47,49]:
  - An OTP is a perfect cipher, unconditionally secure. ☺
  - As long as the key is a random string, of the same length as the plaintext. ☹
  - Cannot use
    - Shorter key (e.g., Vigenere cipher)
    - A key which is not chosen uniformly at random

What we’ve learned today

- Introduction
- Kerckhoff’s Principle
- Some classic ciphers
  - Brute force attacks
  - Required key length
  - A large key does no guarantee security
- Perfect ciphers