Introduction to Cryptography

Lecture 1

Benny Pinkas

Administrative Details

- Grade
  - Exam 75%, homework 25%

- Email: benny@pinkas.net

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

Bibliography

- Textbooks:
  - *Introduction to Modern Cryptography,* by J. Katz and Y. Lindell.
  - *Cryptography Theory and Practice, Second (or third) edition* by D. Stinson. (Also, מדריך למידה בעברית של האוניברסיטה הפתוחה!)

Bibliography

- Optional reading:
  - *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.

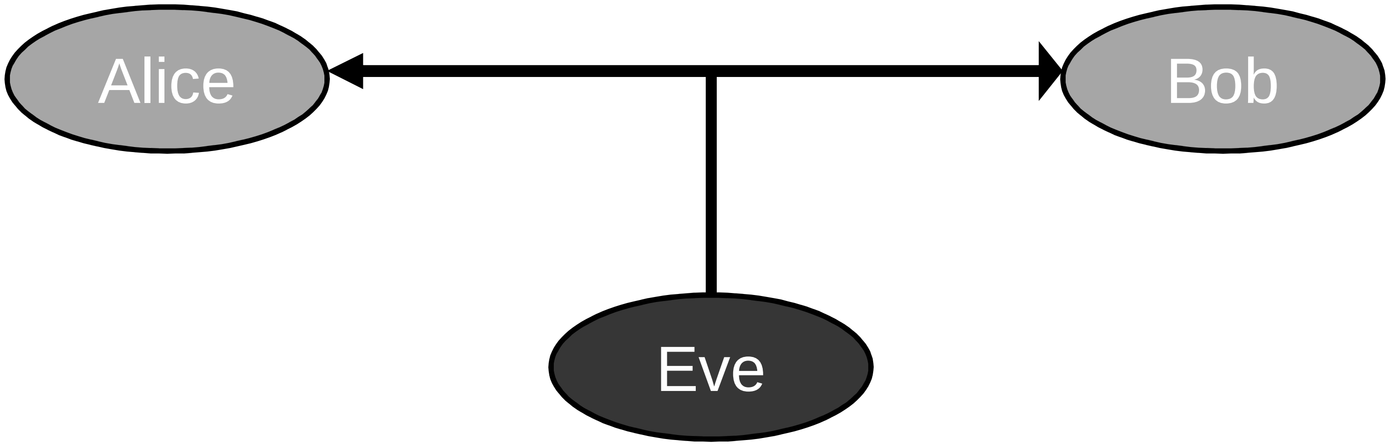
Probability Theory

- One of the perquisites of this course is the course "Introduction to probability"
  - If you haven't taken that course, it is your responsibility to learn the relevant material.
  - You can read Luca Trevisan's notes on discrete probability, available at http://www.cs.berkeley.edu/~luca/crypto-class-99/handouts/notesprob.ps
  - Afterwards, you can also read the part on probability in Chapter 2 of the Handbook of Applied Cryptography, which is available at http://www.cacr.math.uwaterloo.ca/hac/about/chap2.pdf

Course Outline

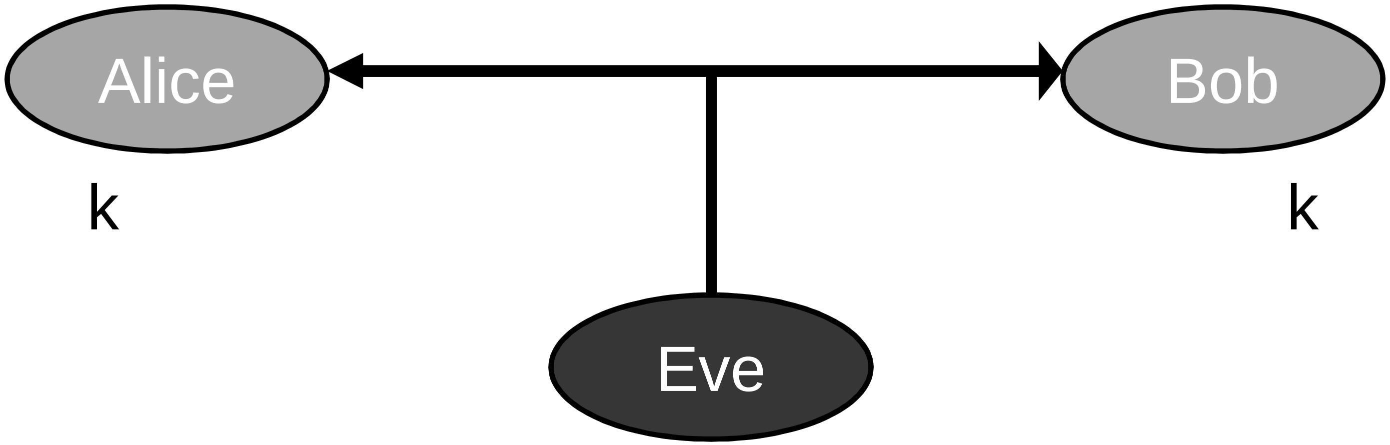
- Course Outline
  - Data secrecy: encryption
    - Symmetric encryption
    - Asymmetric (public key) encryption
  - Data Integrity: authentication, digital signatures.
  - Required background in number theory
  - Cryptographic protocols

Encryption



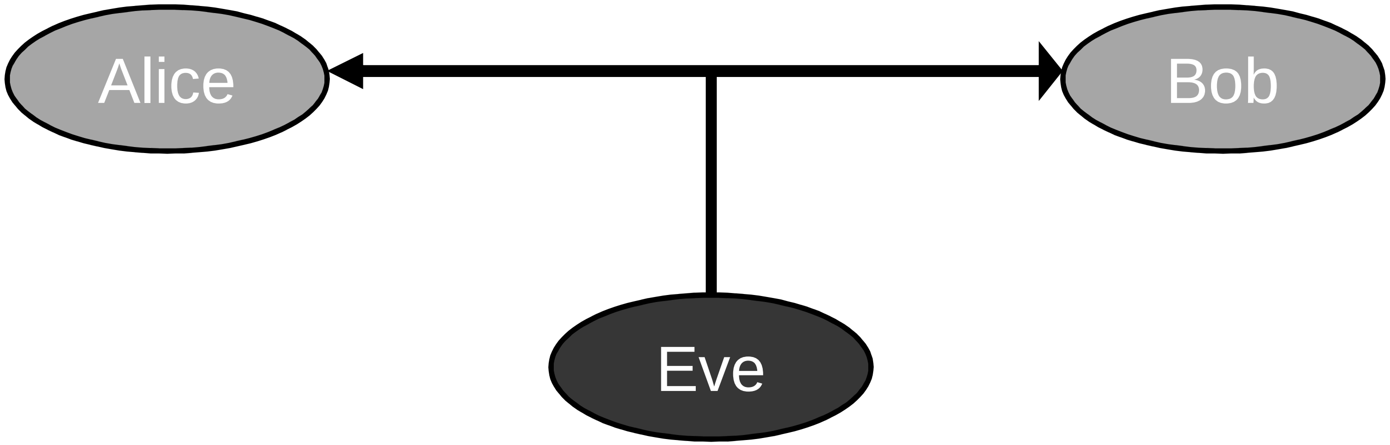
- Two parties: Alice and Bob
- Reliable communication link
- Goal: send a message m while hiding it from Eve (as if they were both in the same room)
- Examples: military communication, Internet transactions, HD encryption.

Secret key



- Alice must have some secret information that Eve does not know. Otherwise…
- In symmetric encryption, Alice and Bob share a secret key k, which they use for encrypting and decrypting the message.

Authentication / Signatures



- Goal:
  - Enable Bob to verify that Eve did not change messages sent by Alice
  - Enable Bob to prove to others the origin of messages sent by Alice
- (We'll discuss these issues in later classes)

Encryption

- Message space *{m}* (*e.g.* $\{0,1\}^n$)
- Key generation algorithm
- Encryption key k_1, *decryption key* k_2
- 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 $k = k_1 = k_2$

Defining an Encryption Scheme

- Must specify the following three algorithms
  - GEN
    - key generation
  - ENC
    - Input: encryption key, plaintext
    - Output: ciphertext
  - DEC
    - Input: decryption key, ciphertext
    - Output: plaintext

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, at 17:15”.*
- The adversary can at most learn that
  - m = “attack on S**day, a* 17:**”
  - m = “******   ** *u****** ** *****”

- Here, goal (1) is satisfied, but not goal (2)
- We will discuss this in more detail…

Adversarial Model

- To be on the safe side, assume that adversary knows the encryption and decryption algorithms *E* and *D,* and the *message space*.
- Kerckhoff’s Principle (1883)

Adversarial Model

- To be on the safe side, assume that adversary knows the encryption and decryption algorithms *E* and *D,* and the *message space*.
- Kerckhoff's Principle (1883)
  - The only thing Eve does not know is the secret key *k*
  - The design of the cryptosystem is public
  - This is convenient
    - Only a short key must be kept secret.
    - If the key is revealed, replacing it is easier than replacing the entire cryptosystem.
    - Supports standards: the standard describes the cryptosystem and any vendor can write its own implementation (e.g., SSL)

Adversarial Model

- Keeping the design public is also crucial for security
  - Allows public scrutiny of the design (Linus' law: "given enough eyeballs, all bugs are shallow")
  - The cryptosystem can be examined by "ethical hackers"
  - Being able to reuse the same cryptosystem in different applications enables to spend more time on investigating its security
  - No need to take extra measures to prevent reverse engineering
  - Focus on securing the key

- Examples
  - Security through obscurity, Intel's HDCP, GSM A5/1. ☹
  - DES, AES, SSL ☺

Adversarial Power

- What does the adversary know or seen before?

- 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

Adversarial Power

- What is the computational power of the adversary?
  - Polynomial time?
  - Unbounded computational power?

- We might assume restrictions on the adversary's capabilities, but we cannot assume 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: $k \in_R \{0\ldots25\}$, chosen at random.
  - Message space: English text (i.e., $\{0...25\}^{|m|}$)
  - Algorithm: ciphertext letter = plaintext letter + k mod 26
- Follows Kerckhoff’s principle
  - But not a good cipher
- A similar “cipher”: ROT-13

Brute Force Attacks

- Brute force attack: adversary tests all possible keys and checks which key decrypts the message
  - *Note that this assumes we can identify the correct plaintext among all plaintexts generated by the attack*

- Caesar cipher: |key space| = 26
- We need a larger 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 for a key length
  - $|k| = 80$ starts to be reasonable
  - Why? (what can be done by 1000 computers in a year?)
    - $2^{55} = 2^{20}$ (ops per second)
    - $\times\ 2^{20}$ (seconds in two weeks)
    - $\times\ 2^{5}$ (≈ fortnights in a year) (might invest more than a year..)
    - $\times\ 2^{10}$ (computers in parallel)
- All this, assuming that the adversary cannot do better than a brute force attack

How much computation is feasible?

- Many speculations and extrapolations on available computing power.
- NIST SP 800-57 allows 80 bit keys until 2010.
- ECRYPT document (2006-2009):

תוקף	budget	hardware	minimal key length secure against attack
hacker	$400	PC/FPGA	~60
small org	$10K	PC/FPGA	64
medium org	$300K	FPGA/ASIC	68
large org	$10M	FPGA/ASIC	78
government	$300M	ASIC	84

Monoalphabetic Substitution cipher

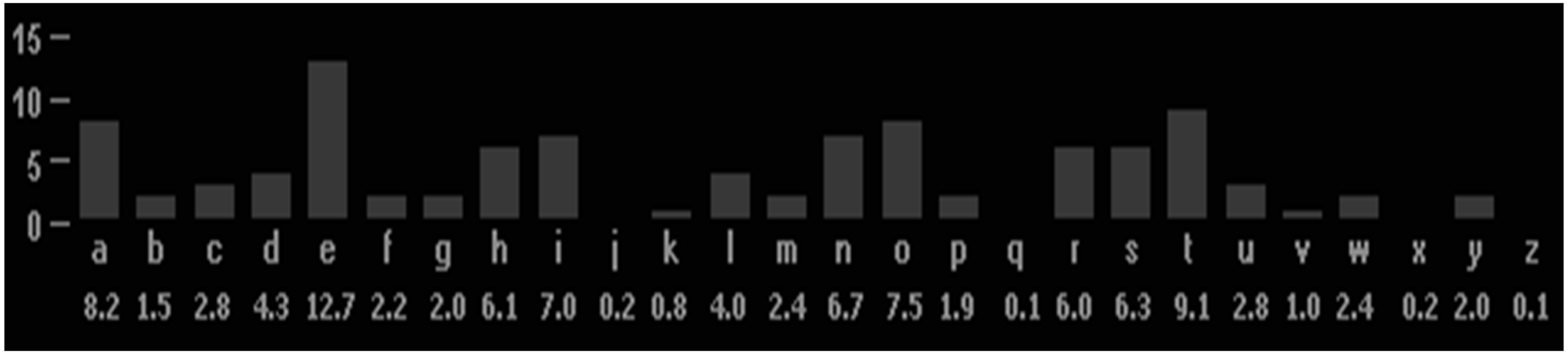
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L	R	C	V	M	U	E	K	J	D	I	X	N

- 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 x 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”)



  - We can also use the fact that the mapping of plaintext letters to ciphertext letters is fixed

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
    - T̲HIS IS T̲HE PLAI̲NTEXT T̲O BE ENC̲RYPTED̲
    - S̲ECR ET S̲EC RETS̲ECRET S̲E CR ETS̲ECRETS̲
  - Attack time: (|*k-1*| + |*k*|) x *time of attacking a shift cipher*[(1)]

- Chosen plaintext attack:
  - Use the plaintext "aaaaaaa…"

(1) How?
- |k-1| failed tests for key lengths 1,…,|k-1|. |k| tests covering all |k| letters of the key.
- Attacking the shift cipher: Assume known letter frequency (no known plaintext). Can check the difference of resulting histogram from the English letters histogram.

Perfect Cipher

- What type of security would we like to achieve?

- 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

- “Given the ciphertext, the adversary has no idea what the plaintext is”
  – Impossible since the adversary might have a-priori information

- Definition: a *perfect cipher*
  – The ciphertext does not add information about the plaintext
  – $Pr(\, plaintext = P \mid ciphertext = C\,) = Pr(\, plaintext = P)$

Probability distributions

- *Pr(plaintext = P | ciphertext = C)*

- Probability is taken over the choices of the key, the plaintext, and the ciphertext.
  - Key: Its probability distribution is usually uniform (all keys have the same probability of being chosen).
  - Plaintext: has an arbitrary distribution
    - Not necessarily uniform (*Pr("e") > Pr("j")*).
  - Ciphertext: Its distribution is determined given the cryptosystem and the distributions of key and plaintext.

- A simplifying assumption: All plaintext and ciphertext values have positive probability.

Perfect Cipher

- For a *perfect cipher*, it holds that given ciphertext *C,*
  - $Pr(\ plaintext = P \mid C\ ) = Pr(\ plaintext = P)$
  - i.e., knowledge of ciphertext does not change the a-priori distribution of the plaintext
  - Probabilities taken over key space and plaintext space
  - Does this hold for monoalphabetic substitution?

Perfect Cipher

- Perfect secrecy is a property (which we would like cryptosystems to have)
- We will now show a specific cryptosystem that has this property

- 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 = $p \oplus k$

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

The one-time-pad is a perfect cipher

ciphertext = plaintext $\oplus$ k

Lemma: $Pr(\text{ciphertext} = 0) = Pr(\text{ciphertext} = 1) = ½$

(regardless of the distribution of the plaintext)

$$
\begin{aligned}
& Pr(\text{ciphertext} = 0) \\
&= Pr(\text{plaintext} \oplus \text{key} = 0) \\
&= Pr(\text{key} = \text{plaintext}) \\
&= Pr(\text{key}=0) \cdot Pr(\text{plaintext}=0) + Pr(\text{key}=1) \cdot Pr(\text{plaintext}=1) \\
&= ½ \cdot Pr(\text{plaintext}=0) + ½ \cdot Pr(\text{plaintext}=1) \\
&= ½ \cdot (Pr(\text{plaintext}=0) + Pr(\text{plaintext}=1)) = ½
\end{aligned}
$$

The one-time-pad is a perfect cipher

ciphertext = plaintext $\oplus$ k

$$
\begin{aligned}
& \mathit{Pr}(\mathit{plaintext} = 1 \mid \mathit{ciphertext} = 1) \\
&= \mathit{Pr}(\mathit{plaintext} = 1 \ \ \& \ \ \mathit{ciphertext} = 1) / \mathit{Pr}(\mathit{ciphertext} = 1) \\
&= \mathit{Pr}(\mathit{plaintext} = 1 \ \ \& \ \ \mathit{ciphertext} = 1) / \tfrac{1}{2} \\
&= \mathit{Pr}(\mathit{ciphertext} = 1 \mid \mathit{plaintext} = 1) \cdot \mathit{Pr}(\mathit{plaintext} = 1) / \tfrac{1}{2} \\
&= \mathit{Pr}(\mathit{key} = 0) \cdot \mathit{Pr}(\mathit{plaintext} = 1) / \tfrac{1}{2} \\
&= \tfrac{1}{2} \cdot \mathit{Pr}(\mathit{plaintext} = 1) / \tfrac{1}{2} \\
&= \mathit{Pr}(\mathit{plaintext} = 1)
\end{aligned}
$$

The perfect security property holds

One-time-pad (OTP) - the general case

- Plaintext = $p_1 p_2 \ldots p_m \in \Sigma^m$ (e.g. Σ={0,1}, or Σ={A…Z})
- key = $k_1 k_2 \ldots k_m \in_R \Sigma^m$
- Ciphertext = $c_1 c_2 \ldots c_m$, $c_i = p_i + k_i \mod |\Sigma|$
- Essentially a shift cipher with a different key for every character, or a Vigenere cipher with |k|=|P|

- 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

Size of key space

- Theorem: For a perfect encryption scheme, the number of keys is at least the size of the message space (number of messages that have a non-zero probability).

- Proof:
  - Consider ciphertext C.
  - C must be a possible encryption of any plaintext m.
  - But, for this we need a different key per message m.

- Corollary: Key length of one-time pad is optimal ☹

Perfect Ciphers

- A simple criteria for perfect ciphers.
- Claim: The cipher is perfect if, and only if,

  $\forall\ m_1, m_2 \in M$, $\forall$cipher c,

  $Pr(Enc(m_1)=c) = Pr(Enc(m_2)=c)$. (homework??)

- Idea: Regardless of the plaintext, the adversary sees the same distribution of ciphertexts.

- Note that the proof cannot assume that the cipher is the one-time-pad, but rather only that *Pr(plaintext = P | ciphertext = C) = Pr(plaintext = P)*

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