Lecture 1

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

January 27, 2016

Introduction and Administration

Information About the Course

- Oral information given and agreements made during lectures.
- Read at: `https://www.kth.se/social/course/DD2448`
- Read your KTH email: `<username>@kth.se`

If this fails, then email `dog@kth.se`.
Use `Krypto16` in the subject line.

What is cryptography?

Cryptography is concerned with the conceptualization, definition, and construction of computing systems that address security concerns.

– Oded Goldreich, Foundations of Cryptography, 1997

Applications of Cryptography

Historically.

- Military and diplomatic secret communication.
- Communication between banks, e.g., credit card transactions.

Modern Time.

- Protecting satellite TV from leaching.
- Secrecy and authenticity on the Internet, mobile phones, etc.
- Credit cards.

Applications of Cryptography

Today.

- Distributed file systems, authenticity of blocks in bit torrents, anonymous remailers, Tor-network, etc.
- RFID tags, Internet banking, Försäkringskassan, Skatteverket, "e-legitimation".

Future.

- Secure distributed computing (multiparty computation): election schemes, auctions, secure cloud computing, etc.
- Variations of signatures, cryptosystem, and other primitives with special properties, e.g., group signatures, identity based encryption, etc.

Goal

The goal of the course is to

- give an overview of modern cryptography

in order that students should

- know how to evaluate and, to some extent, create cryptographic constructions, and
- to be able to read and to extract useful information from research papers in cryptography.

Prerequisites

- *DD1352 Algorithms, data structures and complexity*, or *DD2354 Algorithms and complexity*.
- Knowledge of mathematics and theory of algorithms corresponding to the required courses of the D or F-programmes at KTH.

Tentative Plan of Content (1/2)

- Administration, introduction, classical cryptography.
- Symmetric ciphers, substitution-permutation networks, linear cryptanalysis, differential cryptanalysis.
- AES, Feistel networks, DES, modes of operations, DES-variants.
- Entropy and perfect secrecy.
- Repetition of elementary number theory,
- Public-key cryptography, RSA, primality testing, textbook RSA, semantic security.

Tentative Plan of Content (2/2)

- RSA in ROM, Rabin, discrete logarithms, Diffie-Hellman, El Gamal.
- Security notions of hash functions, random oracles, iterated constructions, SHA, universal hash functions.
- Message authentication codes, identification schemes, signature schemes, PKI.
- Elliptic curve cryptography.
- Pseudorandom generators.
- Guest lecture.
- Make-up time and/or special topic.

Course Requirements

Presentations. **a)** Choose a research topic, and **b)** summarize the topic in a 12-min oral presentation.

Gives P-points ($P = 0$ or $30 \leq P \leq 80$), which is the sum of:

- (20P) Choice of content.
- (20P) Understanding of the content
- (20P) Quality of slides (or whiteboard)
- (20P) Presentation skills.

Up to 4 talks in 1 hour-sessions. Listen to the talks in your session.

Detailed rules and advice are found on the course homepage.

Course Requirements

Homework 1-4. Each homework is a set of problems giving I-points and T-points ($I \geq 10$ and $I + T \geq 50$).

- Solved in groups of up to three students, which may differ for each homework.
- Only informal discussions are allowed.
- Each student writes and submits his own solution.

Detailed rules and advice are found on the course homepage.

Only complete homeworks can be replaced following years.

Course Requirements

Oral Exam. Purpose is to give a fair grade.

Discussion starts from submitted solutions and presentation to ensure that the grading corresponds to the skills of the student.

- For each problem I-points or T-points may be added or removed from the original grading depending on the understanding shown by the student.
- The updated number of points of a problem is never negative and never more than the nominal maximum number of points of the problem stated in the homework.
- A single O-point is awarded after passing the exam.

Deadlines

The deadlines in this course are given on the homepage and are strict. Late solutions are awarded zero points.

However, if practically possible, then we negotiate the deadlines to not conflict unnecessarily with other courses.

Grading

To earn a given grade the requirements of all lower grades must be satisfied as well, with $A = I + T + P + O$.

Grade	Requirements
E	$I \geq 30$, $T \geq 40$, $P \geq 30$, and $O \geq 1$.
D	$A \geq 120$.
C	$A \geq 140$ and $P \geq 50$.
B	$A \geq 170$.
A	$A \geq 210$ and $P \geq 60$.

Kattis

Kattis is a judging server for programming competitions and for grading programming assignments. We use it for all exercises where code is submitted as a solution.

We assume that your Kattis id is the same as your KTH user name. If this is not the case, then email us your Kattis user name and use the subject `Krypto16 Kattis`.

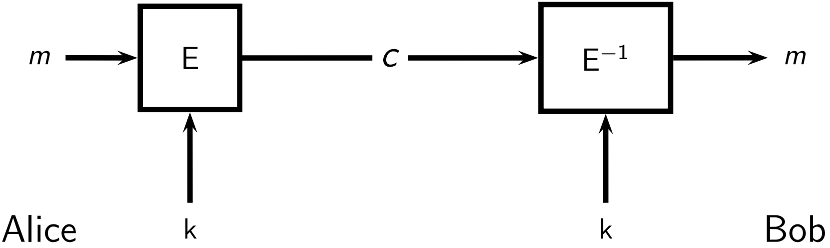
Latex

- Latex is the standard typesetting tool for mathematics.
- It is the fastest way to produce mathematical writing. **You must use it to typeset your solutions.**
- The best way to learn it is to read: `http://tobi.oetiker.ch/lshort/lshort.pdf`

Introduction to Ciphers

Cipher (Symmetric Cryptosystem)

$c = \mathrm{E_k}(m)$ $\qquad$ $m = \mathrm{E_k^{-1}}(c)$



Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

- Gen is a probabilistic **key generation algorithm** outputting keys from a key space $\mathcal{K}$,

Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

- Gen is a probabilistic **key generation algorithm** outputting keys from a key space $\mathcal{K}$,
- $\mathcal{P}$ is a **set of plaintexts**,

Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

- Gen is a probabilistic **key generation algorithm** outputting keys from a key space $\mathcal{K}$,
- $\mathcal{P}$ is a **set of plaintexts**,
- E is a deterministic **encryption algorithm**, and

Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

- Gen is a probabilistic **key generation algorithm** outputting keys from a key space $\mathcal{K}$,
- $\mathcal{P}$ is a **set of plaintexts**,
- E is a deterministic **encryption algorithm**, and
- E^{-1} is a deterministic **decryption algorithm**,

Cipher (Symmetric Cryptosystem)

Definition. A cipher (symmetric cryptosystem) is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathsf{E}, \mathsf{E}^{-1})$, where

- Gen is a probabilistic **key generation algorithm** outputting keys from a key space $\mathcal{K}$,
- $\mathcal{P}$ is a **set of plaintexts**,
- E is a deterministic **encryption algorithm**, and
- E^{-1} is a deterministic **decryption algorithm**,

such that $\mathsf{E}_\mathsf{k}^{-1}(\mathsf{E}_\mathsf{k}(m)) = m$ for every message $m \in \mathcal{P}$ and $\mathsf{k} \in \mathcal{K}$. The set $\mathcal{C} = \{\mathsf{E}_\mathsf{k}(m) \mid m \in \mathcal{P} \wedge \mathsf{k} \in \mathcal{K}\}$ called the **set of ciphertexts**.

Attacks

Throughout the course we consider various attacks on cryptosystems. With small changes, these attacks make sense both for symmetric and asymmetric cryptosystems.

- Ciphertext-only attack.
- Known-plaintext attack
- Chosen-plaintext attack
- Chosen-ciphertext attack

Ceasar Cipher (Shift Cipher)

Consider English, with alphabet A-Z_, where _ denotes space, thought of as integers 0-26, i.e., $\mathbb{Z}_{27}$

- **Key.** Random letter $k \in \mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + k \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - k \bmod 27$.

Ceasar Cipher Example

Encoding.

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	_
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Key: $G = 6$

Plaintext.	B	R	I	B	E	_	L	U	L	A	_	T	O	_	B	U	Y	_	J	A	S
Plaintext.	01	17	08	01	04	26	11	20	11	00	26	19	14	26	01	20	24	26	09	00	18
Ciphertext.	07	23	14	07	10	05	17	26	17	06	05	25	20	05	07	26	03	05	15	06	24
Ciphertext.	H	X	O	H	K	F	R	_	R	G	F	Z	U	F	H	_	D	F	P	G	Y

Decrypt with all possible keys and see if some English shows up, or more precisely...

Written English Letter Frequency Table $F[\cdot]$.

A	0.072	J	0.001	S	0.056
B	0.013	K	0.007	T	0.080
C	0.024	L	0.035	U	0.024
D	0.037	M	0.021	V	0.009
E	**0.112**	N	0.059	W	0.021
F	0.020	O	0.066	X	0.001
G	0.018	P	0.017	Y	0.017
H	0.054	Q	0.001	Z	0.001
I	0.061	R	0.053	**␣**	**0.120**

Note that the same frequencies appear in a ciphertext of written English, but in shifted order!

Statistical Attack Against Ceasar (3/3)

- Check that the plaintext of our ciphertext has similar frequencies as written English.
- Find the key k that maximizes the inner product $T(\mathsf{E}_\mathsf{k}^{-1}(C)) \cdot F$, where $T(s)$ and F denotes the frequency tables of the string s and English.

This usually gives the correct key k.

Affine Cipher.

- **Key.** Random pair $\mathrm{k} = (a, b)$, where $a \in \mathbb{Z}_{27}$ is relatively prime to 27, and $b \in \mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = am_i + b \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = (c_i - b)a^{-1} \bmod 27$.

Substitution Cipher

Ceasar cipher and affine cipher are examples of substitution ciphers.

Substitution Cipher.

- **Key.** Random permutation $\sigma \in S$ of the symbols in the alphabet, for some subset S of all permutations.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = \sigma(m_i)$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = \sigma^{-1}(c_i)$.

Digrams and Trigrams

- A digram is an ordered pair of symbols.
- A trigram is an ordered triple of symbols.
- It is useful to compute frequency tables for the most frequent digrams and trigrams, and not only the frequencies for individual symbols.

Generic Attack Against Substitution Cipher

1. Compute symbol/digram/trigram frequency tables for the candidate language and the ciphertext.
2. Try to match symbols/digrams/trigrams with similar frequencies.
3. Try to recognize words to confirm your guesses (we would use a dictionary (or Google!) here).
4. Backtrack/repeat until the plaintext can be guessed.

This is hard when several symbols have similar frequencies. A large amount of ciphertext is needed. How can we ensure this?

Vigénère Cipher.

- **Key.** $\mathsf{k} = (\mathsf{k}_0, \ldots, \mathsf{k}_{l-1})$, where $\mathsf{k}_i \in \mathbb{Z}_{27}$ is random.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + \mathsf{k}_{i \bmod l} \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - \mathsf{k}_{i \bmod l} \bmod 27$.

Vigénère Cipher.

- **Key.** $\mathsf{k} = (\mathsf{k}_0, \ldots, \mathsf{k}_{l-1})$, where $\mathsf{k}_i \in \mathbb{Z}_{27}$ is random.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + \mathsf{k}_{i \bmod l} \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - \mathsf{k}_{i \bmod l} \bmod 27$.

We could even make a variant of Vigénère based on the affine cipher, **but is Vigénère really any better than Ceasar?**

Attack Vigénère (1/2)

Index of Coincidence.

- Each probability distribution $p_1, \ldots, p_n$ on n symbols may be viewed as a point $p = (p_1, \ldots, p_n)$ on a $n-1$ dimensional hyperplane in $\mathbb{R}^n$ orthogonal to the vector $\overline{1}$
- Such a point $p = (p_1, \ldots, p_n)$ is at distance $\sqrt{F(p)}$ from the origin, where $F(p) = \sum_{i=1}^{n} p_i^2$.
- It is clear that p is closest to the origin, when p is the uniform distribution, i.e., when $F(p)$ is minimized.
- $F(p)$ is invariant under permutation of the underlying symbols $\longrightarrow$ tool to check if a set of symbols is the result of *some* substitution cipher.

Attack Vigénère (2/2)

1. For $l = 1, 2, 3, \ldots$, we form

$$\begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_{l-1} \end{pmatrix} = \begin{pmatrix} c_0 & c_l & c_{2l} & \cdots \\ c_1 & c_{l+1} & c_{2l+1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ c_{l-1} & c_{2l-1} & c_{3l-1} & \cdots \end{pmatrix}$$

   and compute $f_l = \frac{1}{l}\sum_{i=0}^{l-1} F(C_i)$.

2. The local maximum with smallest l is probably the right length.
3. Then attack each C_i separately to recover k_i, using the attack against the Ceasar cipher.

Hill Cipher.

- **Key.** $\mathrm{k} = A$, where A is an invertible $l \times l$-matrix over $\mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where (computed modulo 27):

$$(c_{i+0}, \ldots, c_{i+l-1}) = (m_{i+0}, \ldots, m_{i+l-1})A \ .$$

- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where (computed modulo 27):

$$(m_{i+0}, \ldots, m_{i+l-1}) = (c_{i+0}, \ldots, c_{i+l-1})A^{-1} \ .$$

  for $i = 1, l+1, 2l+1, \ldots$

Hill Cipher.

- **Key.** $\mathrm{k} = A$, where A is an invertible $l \times l$-matrix over $\mathbb{Z}_{27}$.

- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where (computed modulo 27):

$$(c_{i+0}, \ldots, c_{i+l-1}) = (m_{i+0}, \ldots, m_{i+l-1})A \ .$$

- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where (computed modulo 27):

$$(m_{i+0}, \ldots, m_{i+l-1}) = (c_{i+0}, \ldots, c_{i+l-1})A^{-1} \ .$$

  for $i = 1, l+1, 2l+1, \ldots$

The Hill cipher is easy to break using a known plaintext attack.

Permutation Cipher (Transposition Cipher)

The permutation cipher is a special case of the Hill cipher.

Permutation Cipher.

- **Key.** Random permutation $\pi \in S$ for some subset S of the set of permutations of $\{0, 1, 2, \ldots, l-1\}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_{\lfloor i/l \rfloor + \pi(i \bmod l)}$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_{\lfloor i/l \rfloor + \pi^{-1}(i \bmod l)}$.