DD2448 Foundations of Cryptography
Lecture 1

Douglas Wikström
KTH Royal Institute of Technology
`dd2448@kth.se`

March 18, 2024

Introduction and Administration

Information About the Course

- **Information given and agreements made during lectures.**
- **Your group members and friends if you miss a lecture.**
- `https://www.kth.se/social/course/DD2448`
- `https://canvas.kth.se/courses/46237`
- Read your KTH email: `<username>@kth.se`
- You cannot use "Discussions" at the course page at Canvas and there is no official course group in any other social media.

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".
- Crypto currencies

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.

Intended Learning Outcomes

After a completed course, the student should be able to discuss the following basic concepts in cryptography:

- symmetric and asymmetric encryption, digital signatures, cryptographic hash functions and strong pseudorandom generators and to give examples of instantiations of each concept
- conduct simple analyses of cryptographic constructions such as cryptosystems and cryptographic protocols
- read analyses performed by others of cryptographic constructions such as cryptosystems and cryptographic protocols and decide if the given analysis can be trusted
- read and understand technical articles in cryptography.

Recommended Prerequisites

Knowledge equivalent to either one of the courses DD1352 Algorithms, Data Structures and Complexity or DD2354 Algorithms and Complexity and knowledge of probability theory, mathematics and algorithm theory acquired in the mandatory courses of the D or F program.

Tentative Plan of Content (1/2)

- Administration, introduction, classical cryptography.
- Discussion about survey project.
- Symmetric ciphers, substitution-permutation networks, linear cryptanalysis, differential cryptanalysis.
- AES, Feistel networks, DES, modes of operations, DES-variants.
- Entropy and perfect secrecy.
- Security notions of hash functions, random oracles, iterated constructions, SHA, universal hash functions.
- Public-key cryptography, RSA, primality testing, textbook RSA, CPA security.

Tentative Plan of Content (2/2)

- RSA in ROM, Rabin, discrete logarithms, Diffie-Hellman, El Gamal.
- Message authentication codes, identification schemes, signature schemes, PKI.
- Elliptic curve cryptography.
- Pseudorandom generators.
- Post-quantum cryptography
- Cryptographic protocols
- Guest lecture?
- Make-up time and/or special topic.

Course Requirements (1/3)

Survey Project. Gives S points. In groups of three students:

1. Read about the motivation for post-quantum cryptography and the standardization process and describe it.
2. Describe your favorite candidate and its pros and cons.
3. Review 3 other groups' survey projects.

Detailed rules and advice will be published at Canvas.

Course Requirements (2/3)

Homework. Gives T points. In groups of three students:

- Solve a set of problems at the end of the course.
- Only **informal** discussions are allowed within the group.
- Each student writes and submits their own solution.

Similar problems are available as exercises on Canvas.

Detailed rules and advice will be published on Canvas.

Course Requirements (3/3)

Oral Exam. The purpose is to give a fair grade.

- Each student takes the oral exam individually.
- S points or T points may be added or removed from the tentative grading depending on the understanding shown.
- The number of points for a homework problem or the survey project is always non-negative.
- A single E point is awarded after passing the exam.
- Every student can give feedback about the course.

Grading

S_0 and T_0 are the nominal number of survey project, and theory, points, respectively. The grades G_S and G_T for the GRU1 and INDA, respectively, are defined as follows[1].

S/S_0	$8/10$	$7/10$	$6/10$	$5/10$	$4/10$
G_S	A	B	C	D	E

T/T_0	$8/10$	$7/10$	$6/10$	$5/10$	$4/10$
G_T	A	B	C	D	E

We identify the grades from F to A with the integers from 0 to 5. If $E = 0$, $G_T = 0$, or $G_S = 0$, then the course grade is F, and otherwise it is $\lfloor (G_S + G_T)/2 \rfloor$ or $\lceil (G_S + G_T)/2 \rceil$ depending on if $G_T < G_S$ or not.

[1] GRU1 gives a P/F grade in LADOK, but influences the course grade.

Latex

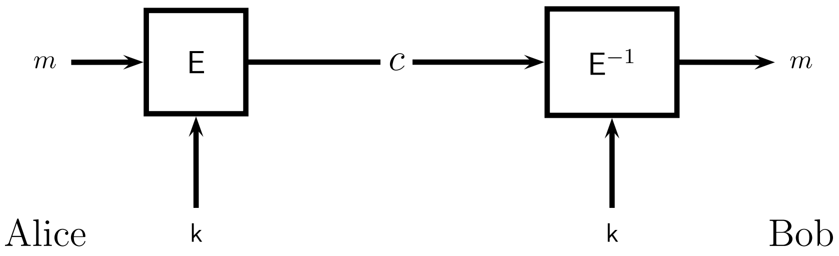
- LaTeX is the standard typesetting tool for mathematics.
- It is the fastest way to typeset mathematical writing.
- **You must use the templates found under Files on Canvas to typeset your solutions.**
- The best way to learn LaTeX is to read: `http://tobi.oetiker.ch/lshort/lshort.pdf`

Introduction to Ciphers

Cipher (Symmetric Cryptosystem)

$c = \mathsf{E}_{\mathsf{k}}(m)$

$m = \mathsf{E}_{\mathsf{k}}^{-1}(c)$



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 encryption followed by decryption recovers the message.

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 $\mathsf{E}^{-1}_{\mathsf{k}}(\mathsf{E}_{\mathsf{k}}(m)) = m$ for every $m \in \mathcal{P}$ and every $\mathsf{k} \in \mathcal{K}$.

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

Caesar Cipher (Shift Cipher)

Consider English, with alphabet A-Z_, where _ denotes space, thought of as integers modulo 27, 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
$$c_i = m_i + k \bmod 27 \enspace .$$
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where
$$m_i = c_i - k \bmod 27 \enspace .$$

Caesar Cipher (Example)

Encoding the alphabet as $\mathbb{Z}_{27}$.

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

Example

Key: $G = 6$
Substitution table:

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

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

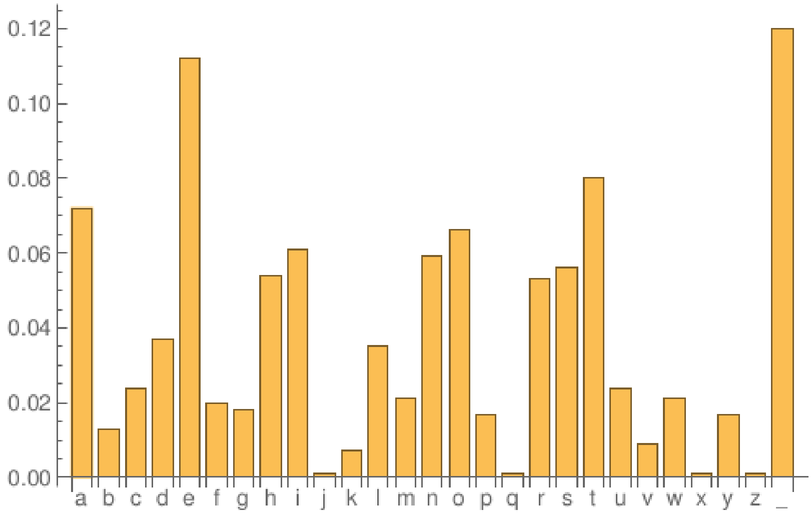
Letter Frequencies (1/2)

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!

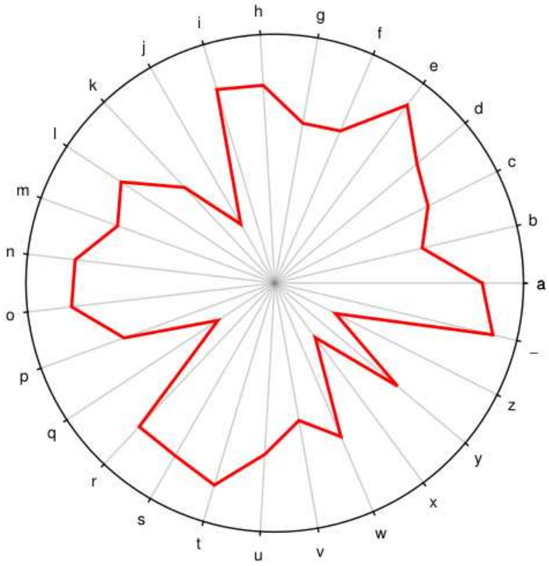
Letter Frequencies (2/2)



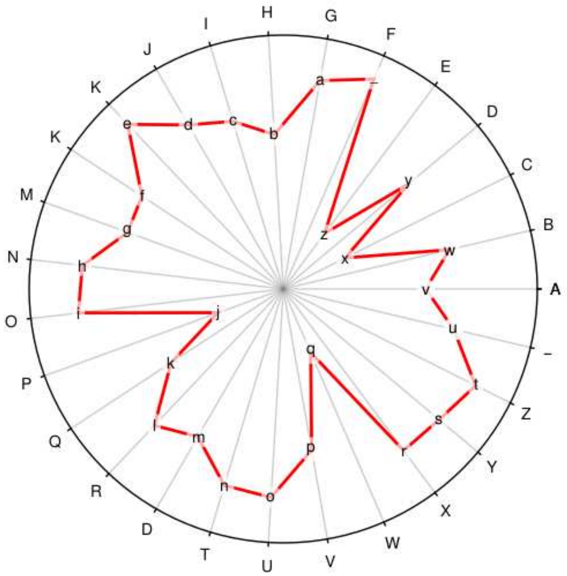
Cryptanalysis of the Caesar Cipher

- **Exploit structure.** If we know that a **single** letter α is encrypted into a ciphertext letter β, then $k = \beta - \alpha \bmod 27$.
- **Brute force and redundancy.** Try all keys and check if the result is English, e.g., by counting Google hits. This works since the keyspace is very small.
- **Statistics and structure.** "Rotate" until the letter frequences of the ciphertext and English match. Works for any "language" with non-uniform letter frequencies!

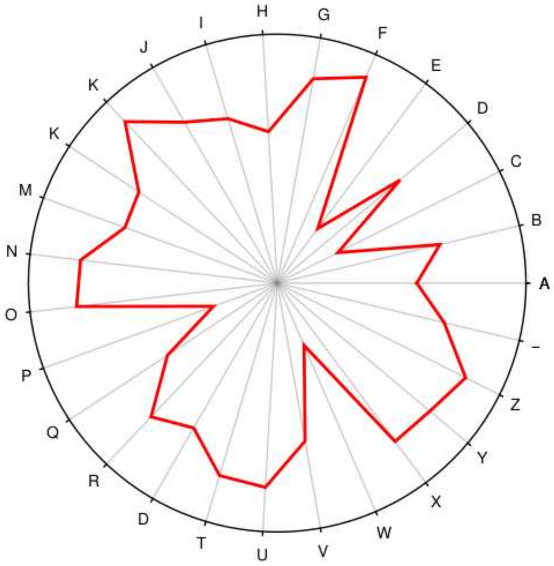
Polar Plot of Letter Frequencies (log-scaled for clarity)



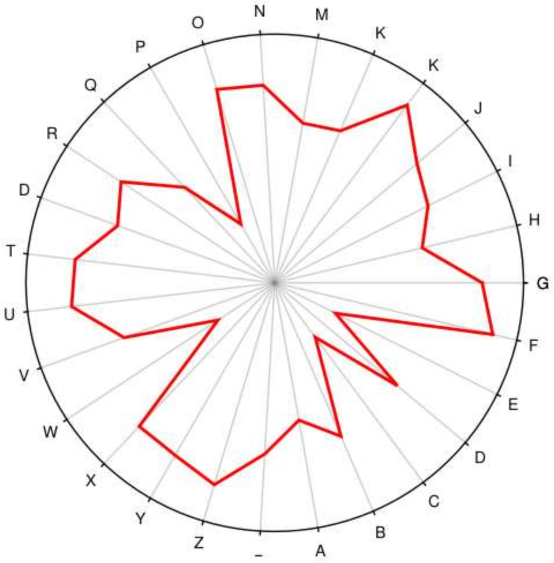
Substitution Table (Key: G=6)



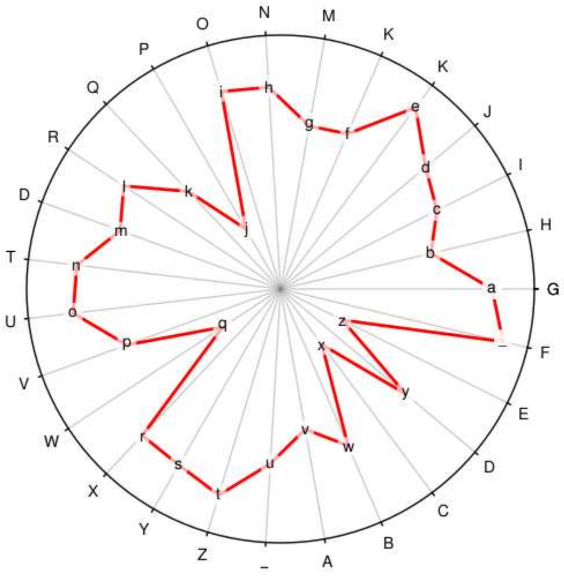
Attacker's Statistical View of a Ciphertext



Rotate Until It Matches Letter Frequencies



Interpret



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

Affine Cipher

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 $m_i = (c_i - b)a^{-1} \bmod 27$.

Relative primality of a and 27 implies that ($a^{-1} \bmod 27$) exist.

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 $m_i = \sigma^{-1}(c_i)$.

Break Substitution Cipher

Identify. Check that the plaintext of our ciphertext has similar frequencies as written English. How can we do this?

Known cipher. If we know the cipher E we find the key k that maximizes the inner product $T(\mathrm{E}_{\mathrm{k}}^{-1}(C)) \cdot F$, where $T(s)$ and F denotes the frequency tables of the string s and English.

This usually gives the correct key k.

Break Substitution Cipher

Identify. Check that the plaintext of our ciphertext has similar frequencies as written English. How can we do this?

Known cipher. If we know the cipher E we 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.

Kerckhoff's Principle and Guessing?

Digrams and Trigrams

To differentiate symbols which have similar frequencies and/or are rare during deciphering, it is useful to compute frequency tables for the most frequent digrams and trigrams.

- A **digram** is an ordered pair of symbols.
- A **trigram** is an ordered triple of symbols.

Examples: “sh”, “the”, “I␣”,...

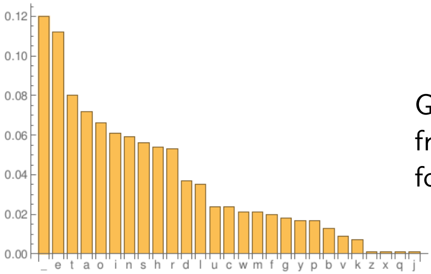
Generic Attack Against Substitution Cipher

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?

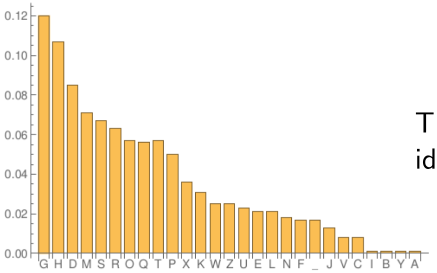
Cryptanalysis of the Substitution Cipher

Sorted English Frequencies



Group by similar frequency and brute force within each group!

Sorted Ciphertext Frequencies



This bar chart is not identical to the above!