

IDATT2503 The Cryptography Part

Lecture 1: Introduction and practical matters

Some important ideas and concepts introduced through some classical ciphers

13. Oktober 2024

Dag Olav Kjellemo

Practicalities

- Lectures in weeks 41 – 46, repetition week 47
- 6 assignments, 4 must be approved
- Learning resources: Most of syllabus will be covered by lecture notes and assignments, with some additional material

Exam

4 hours, 50% cryptography

Not allowed to bring written material, but a cheat sheet will be provided as attachment in Inspira. It will cover things that I consider not so important to memorize, and want to focus on testing understanding. It will contain

- Number theory,
- Algorithms
- Some definitions, e.g. modes of operation.

Open to “negotiation” what to include there.

Assignments

Hand in files in Blackboard.
Please do your own work!

- CTF?

Teaching material and resources

The curriculum will be defined by lecture notes, with some supplements. Some sources, all freely available:

- <https://www.crypto101.io>
- <https://joyofcryptography.com/>
- Introduction to modern Cryptography
<https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf>

Software:

- For digital exploration of cryptosystems, one can start with
<https://www.cryptool.org/en/cto/>
- [Practical Cryptography](#)

Book:

- Cryptography and network security, by W. Stallings

Other references may be provided along the way

Today's lecture

- Introduce some classical ciphers
- See examples of how easily they are broken
- “Extract” features and concepts from these that we will develop further.
- Define modern block ciphers, more in depth next week
- Some number theory we will need, mostly modular arithmetic.
- A little statistics/probability theory for cryptanalysis.

Provide notes to number theory, these are available on exam.

Todo: What is available on exam? Get formula sheet.

Læringsmål og pensum

Vi skal se på noen historiske chifre, men hovedfokus er å lære prinsipper for moderne kryptografi

- Matematisk grunnlag:
 - tallteori og diskret matematikk
 - Sannsynlighetsregning og statistikk for å forstå sikkerheten til et system
- konsepter brukt i moderne kryptografi
- Noen historiske og moderne kryptosystemer
 - Symmetriske systemer
 - Asymmetriske systemer (offentlig nøkkel kryptografi)
- Definere og forstå hva som menes med «sikkerhet» i kryptosammenheng.
- Kryptanalyse
 - Kjenne til forskjellige angrepsmodeller og konkrete eksempler på angrep kryptosystemer

Pensum er forelesningsnotater, øvinger og annet materiell som blir lagt ut og ***definert som pensum.***

Some necessities

Modern crypto is mathematical, and one of the learning goals of this course is to know about some of this mathematics. We should be familiar with

- Number representations, especially binary/hexadecimal numbers (assumed as known)
- Number theory: modulo calculus and prime numbers, some key results (which we'll cover)
- Some basic intuition about probabilities and statistics.
- It is used to define security, and as tools in cryptanalysis.

Some programming. It's a good way to better understand certain algorithms involved. This is only relevant in the assignments, not exams. Free choice of programming language, but nothing too exoteric I guess...

What is Cryptography

Modern Cryptography (from Greek: hidden writing) is the science of ways to store and transmit information in ways that prevent unauthorized access or interference. This includes:

- Keep information secret (secrecy)
- Avoiding tampering (integrity)
- Providing authenticity (proof of sender)
- Avoiding denial of sending (non-repudiation)

The «reverse» of this is *cryptanalysis*: The «breaking» of ciphers, how an adversary may get some or all information, changing in malicious ways, or maskering as someone else.

Cryptology is used to cover both cryptography and cryptanalysis, but sometimes one uses cryptography to cover both aspects.

The shortened **crypto** today often refers to cryptocurrency.

Information security

Cryptography is part of *Information security*: Minimizing vulnerabilities of information assets.

- Vulnerability: Any weakness that could be exploited to violate a system or the information it contains.
- Information assets: Can include data, software and hardware, people and even buildings.
- Threat: A threat is a potential violation of security.

What threats may an adversary pose?

Recommended video: First part of

https://youtu.be/o1x_Oa0XiDI?si=lcky38jflYIXdCpf

Historical ciphers to modern

Historically cryptography was based on ingenuity and beliefs, and not a science.

- The Vigenère cipher was first described in 1553, and for a long time believed to be unbreakable, until 1863!

Modern cryptography defines precisely what security of a cryptographic scheme means. It involves

- Giving precise assumptions, and mathematicall proving properties from these assumptions.
- This includes assumptions about our adversaries: what are their capabilities, what information do they have, and what computational facilities.
- Much is also based on our belief that $P \neq NP$, and other mathematical conjenctures.

Moderne kryptografi

Er en *vitenskap*, med

- Rigorøse analyser, med et solid og velutviklet teoretisk fundament
- Veldefinert angrepsmodellering og beviselig sikkerhet under spesifiserte betingelser
- I tillegg til *hemmelighold*, så omhandler moderne kryptografi også
 - **Integritet** av data
 - **Signering**
 - **Ikke-fornektbarhet**
 - Med mer...
- Moderne kryptografi handler om design, analyse og implementasjon av matematiske og andre teknikker for å sikre informasjon, systemer og distribuerte beregninger fra angrep.

Cryptography is pervasive

- Secure transactions over open networks
- Encryption of stored information (e.g. disk encryption)
- Digitally signed software updates
- Password handling
- Cryptocurrency (not curriculum)

Skole-versjoner vs. «ekte vare»

- Det kan være forskjell på «skole-bok-versjoner» som vi ser på, og implementasjoner av faktisk brukte krypteringsprotokoller
- Vi skal fokusere mest på noen sentrale byggestener, såkalte **kryptografiske primitiver**,
- Disse inngår i **kryptografiske protokoller**, som er ment å ivareta sikkerheten til hele prosesser. (for eksempel SSL)
- Et systems sikkerhet er basert på at visse premisser holder, og feil bruk av ellers sikre mekanismer kan gjøre det usikkert.

A general classification of ciphers

- Symmetrical with private shared keys
 - Secrecy: Private key encryption/decryption
 - Integrity: Message authentication codes (MACs)
- Asymmetrical with public key (and associated private key)
 - Secrecy: Public key ciphers like RSA and ElGamal
 - Integrity: Digitale signatures

Important ingredients to these are:

- (Pseudo-) random numbers
- (Pseudo-) random functions
- Cryptographic hash functions

Some character roles

Popular characters that represent roles in the cryptographic play:

- **Alice and Bob:** (A and B) The parties that want to communicate securely
- **Eve:** (Eavesdropper) An adversary, often just listening to the communication between Alice and Bob, not tampering with it (but not always restricted to this)
- **Mallory:** An active adversary that may tamper with the communication: Changing it, sending own messages masquerading as coming from Eve, etc. A more dangerous type of adversary!

(see https://en.wikipedia.org/wiki/Alice_and_Bob)



Some historical ciphers

Why historical ciphers

They are not secure! So why?

- They are perhaps a little easier to use to introduce certain topics that are relevant to modern ciphers.
- Illustrating perhaps that secure ciphers are hard to design?
- And hopefully a little fun to play with 😊

Cæsar cipher: Encryption

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
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



The alphabet has an ordering.

Encrypt by *replacing* each letter with letter 3 places after, at end, continue to count from start.

Encrypt «Encrypt»

Cæsar cipher: Encryption

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
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



The alphabet has an ordering.

Encrypt by *replacing* each letter with letter 3 places after, at end, continue to count from start.

Encrypt «encrypt this»:

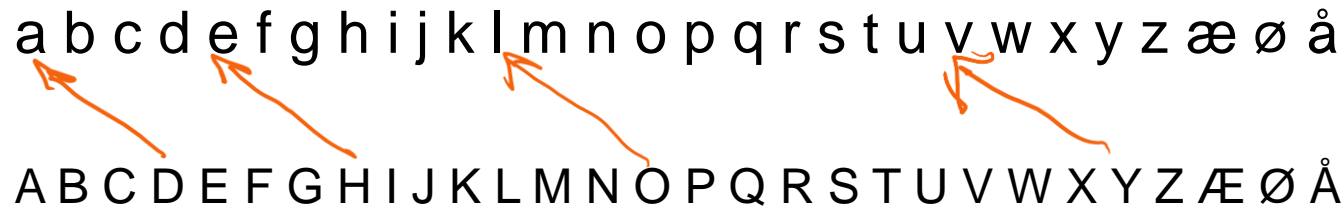
HQFUB SWWKV

Its tradition to ignore case, and use lower case for plain text, and UPPER case for encrypted text. Also, spaces and punctuation is removed, and the encrypted text is written in blocks of five to make it more «readable».

THIS MAKES CRYPTANALYSIS MUCH HARDER, as we shall see later.

Cæsar cipher, decryption

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 æ ø å
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 Æ Ø Å



Substitute with letter shifted back by three places

Decrypt the following:

YHONR PPHQW LONUB SWRJU DILGH OHQ

Plain text:

Shift or rotation cipher

- The Cæsar-cipher *has no variable key*. Security is only provided by the adversary not knowing the algorithm.
- We generalize: Shift by k instead of 3.
- k is then the *key of the cipher*.
- The *algorithm* is to rotate the alphabet by k .

Kerckhoff's principle: The security of a cipher should depend on the secrecy of the key, assuming that the algorithm is known to an adversary.

Importance of Randomness

- For the rotation cipher to be any better than Cæsars, we obviously need to keep the key secret, and that an adversary cannot guess it!
- This means that it has to be chosen randomly!
- We will put this into the definition of a cipher now:

Definition of a cipher

Lets formalized what we have so far

- We need an **alphabet** (set of symbols) Σ .
- The set of legal plain texts (messages) \mathcal{M} that we can encrypt (often the strings Σ^* over Σ)
- The set of legal cipher texts \mathcal{C}
- The set of \mathcal{K}
- A (pseudo-) random key generator $K \leftarrow \text{Gen}$
- Two algorithms, one for encryption and one for decryption

$$\mathcal{E} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$$

$$\mathcal{D} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$$

- Decryption must be inverse of encryption for each key.

We will often write

$$\mathcal{E}(k, m) = \mathcal{E}_k(m)$$

$$\mathcal{D}(k, c) = \mathcal{D}_k(c)$$

So that the inverse property can be written as

$$\mathcal{D}_k(\mathcal{E}_k(m)) = m$$

(And often we'll not use fancy-fonts like \mathcal{E} ... *this is not standardized*)

Writing the rotation cipher with math

Exercise: Define the rotation cipher mathematically.

\mathbb{Z}/\mathbb{Z}_n Numbers modulo n , and modulær arithmetic

1 is congruent to all the numbers
1, 6, 11, $1 + 5k$, k heltall,
modulo 5.

This is the same as $a - b$ being
divisible by 5. We can write this as

$$a \equiv b \pmod{n} \Leftrightarrow n \mid (a - b)$$

We'll also write $a \pmod{n}$ or $a \% n$ for
the **unique** number in the range
0 to $n-1$ that is congruent to a .

Quotient-remainder theorem

For a positive integer n , and any integer a , there is a unique integer r with the properties

$$a = kn + r, \quad 0 \leq r < n$$

This r is the remainder we get when we divide a by n

- $a \% n = r$.

More modular arithmetic

- Most of the rules of arithmetic are valid «modulo n ».
- We can always reduce mod n during calculations, or at the end, if we wish
- $3 * 7 \pmod{5} = 21 \% 5 = 1$
- $33 * 124 \% 5 = 3 * 4 \% 5 = 12 \% 5 = 2$
- $2^{10} \% 7 = 2^3 * 2^7 = 1 * 2^3 * 2^3 * 2 \% 7 = 2$
- Associative, commutative, distributive laws.
- $11^{100} \% 10 = 1^{100} \% 10 = 1$
- $10^{100} \% 10 = 0$
- NB! $2^{**8} \% 7$ is not equal to $2^{**1} \% 7$, cannot reduce the exponents towards 7. Will come back to this.

Formalizing Rotation cipher

For one character/letter, we have

- $\Sigma = \mathcal{K} = \mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$
- $\mathcal{E}: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}, (K, M) \mapsto C = (M + K) \% n$
- $\mathcal{D}: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}, (K, C) \mapsto M = (C - K) \% n$

We extend this to apply to arbitrary long sequences/strings $\mathcal{M} = \mathcal{C} = \Sigma^*$ by repeating the algorithm above for each character/number and then concatenating these:

If $m = m_1 m_2 \dots m_l$ is a sequence of l integers in \mathbb{Z}_{26} , then
 $\mathcal{E}_k(m_1 m_2 \dots m_l) = \mathcal{E}_k(m_1) \mathcal{E}_k(m_2) \dots \mathcal{E}_k(m_l)$

Cryptanalysis of Rot

- **Attack model:** Adversary only has access to encrypted messages. Forkortes C(ipher)O(nly)A(ttack).
- Since there are so few keys, Eve can try each key in turn. This is what we call a *brute force attack*.

We must make the following assumption:

An attacker will know when he has found the correct plaintext.

This is a reasonable assumption for natural languages, at least for messages that are not very short.

When the messages are other data, this may be a little less obvious, but unencrypted text will normally have recognizable features.

Rot provides perfect secrecy if...

If we are to encrypt a single character, choosing the key randomly, we in fact get *perfect secrecy*.

What does that mean? Under what assumptions?

Hint: Do the adversary get any extra knowledge about the plaintext knowing the single ciphertext?

The problem is that we use the same key to encrypt more than one character.

How to increase number of keys

There are simply too few keys to choose from in Rot

We can increase the number of keys in different ways:

1. A bigger class of encryption functions

- Affine cipher: $E_k(x) = (ax + b) \bmod n$, where key is the pair $k = (a, b)$
- ...other formulas with more parameters
- Ultimately: An arbitrary, random permutation of the alphabet, not defined by a fixed type of (simple) formula.

2. Block ciphers: Encrypting more characters at a time

- Defined by some type of formula or algorithm
- Ultimately: A random permutation of the whole set of blocks

3. Stream ciphers: Encryption key varies with the position in the message stream

- The encryption is usually simple e.g XOR for each character

Simple (mono-alphabetical) substitution ciphers

We can significantly increase the number of keys if we allow arbitrary permutations of the alphabet.

- Σ alphabet
- $\mathcal{M} = \mathcal{C} = \Sigma^*$ all sequences of strings (tuples, sequences)
- \mathcal{K} = all permutations av Σ (substituting one letter for another)
- $\pi \leftarrow \text{Gen}(p) \in \mathcal{K}$ a random permutation of Σ

Example:

- $\pi(a) = f, \pi(b) = z, \pi(c) = a$
- Encrypt «abc»: $\pi(a)\pi(b)\pi(c) = fza$

The permutation can be given as an reordering of the alphabet:

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

Meaning that 'a' in plaintext is substituted by 'H' in cipher 'b' by 'L' etc.

Cryptanalysis of simple substitution cipher

How many keys are there with a 26 letter alphabet?

$26 \cdot 25 \cdot 24 \cdot \dots \cdot 3 \cdot 2 \cdot 1 = 26! = 403291461126605635584000000$

Or about $4 \cdot 10^{26} = 2^{88.4}$

Key length about 89 bits.

Is this secure against **brute force (ciphertext only)** attacks?

Yes, because it is practically impossible to try all of them, and one must expect that one has to try about half of them.

But is it safe?

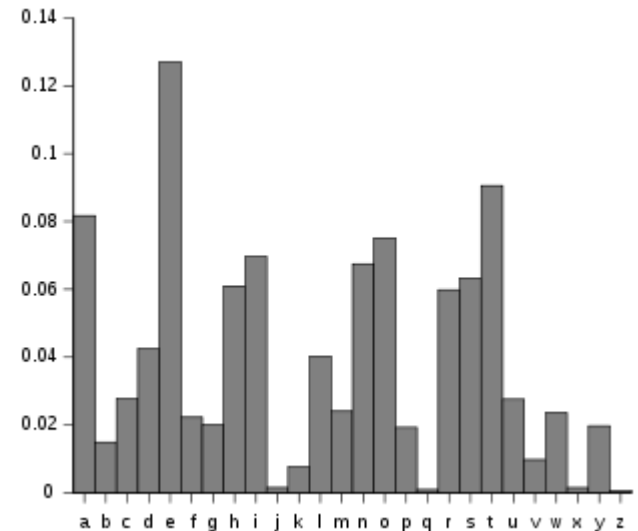
Cryptanalysis: Using frequency analysis

Languages have statistical regularities. The substitution cipher keeps the relative frequencies intact.

This can be extended to use frequencies of combinations of letters to help us reverse the permutation.

If we also encrypted spaces, it would normally be the most frequent character, and we can use whole words in our cryptanalysis.

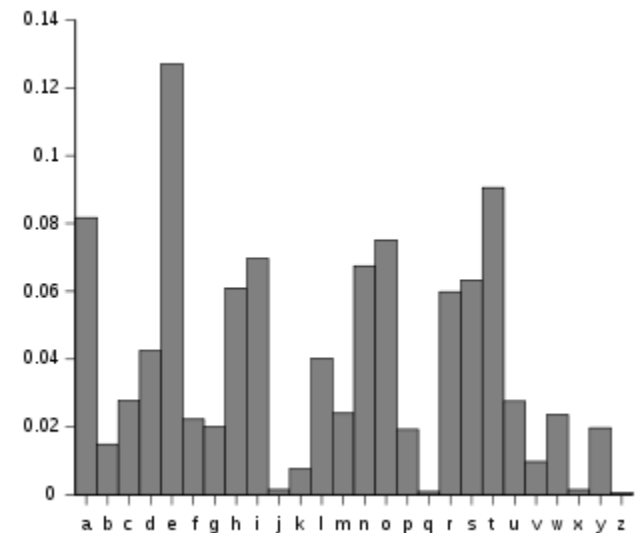
The effectiveness relies on the length and type of text.



With substitution cipher, only the ordering change, we can try to reverse the substitution by comparing frequencies.

Frequency analysis

- A more detailed explanation can be found at
- <https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html> or here:
- <http://practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-simple-substitution-cipher/>
- The frequencies of the given ciphertext is compared to the frequencies of the language in general, to give an initial guess of the substitution.
- Looking for combinations of letters can help improve the guess
- One modifies the substitution to give better matches, until one has a meaningful text.
- For this to be effective, the ciphertext must be of a certain length, typically at least 50 characters long.



Frequency analysis contd.

Attack models – what is available to an adversary

- *Ciphertext only attack (COA)*: Eve know only ciphertext.
- *Known-plaintext attack (KPA)*: Eve has a plain text m and associated cipher text c
- *Chosen plaintext attack (CPA)*: As above, but Eve has chosen plain text herself, (not the specific message she wants to reveal) (**CPA2**: adaptive CPA)
- *Chosen ciphertext attack (CCA)*: Eve has a chosen ciphertext that she has had decrypted.

In addition, it will also matter how much text Eve has available.

Other attack models

There are other types of attacks as well, for example side channel attacks. We will not cover this, as they do not depend so much on the strength of the strength of the cipher, but more on implementation details.

Attack models – what does adversary want to achieve

What constitutes a successful attack will also vary

- Find the full or partial contents of an encrypted message
- Find a full or partial key to encrypt other messages
- Change a message without the sender/receiver detecting it
 - This can be done even without the attacker being able to decrypt the message.
 - An attacker can have more or less control over how the message is modified.
 - We will also look at other attack models/targets for applications other than secrecy.

Cryptanalysis of simple substitution ciphers: Other attack models than plaintext only

What can we say about the security of substitution cipher for

- Known plaintext?
- Chosen plaintext?
- Chosen cipher text?

Cryptanalysis of simple substitution ciphers: Other attack models than plaintext only

What can we say about the security of substitution cipher for

- Known plaintext?
 - Will reveal the key for the signs that occur. It will then be much easier to decrypt the rest, and also find the rest of the key.
- Chosen plaintext?
 - Plain text: The entire alphabet. The key is revealed in full.
- Chosen cipher text?
 - Select the full (cipher) alphabet. You find the whole key.

Varieties of substitution ciphers

Simple substitution ciphers are monoalphabetic, there is a fixed substitution for each letter.

Polyalphabetic ciphers, such as Vigenère, use a sequence of different substitutions, depending on the position in the text (more later).

Polygraphical ciphers involves substituting group of letters.

For example, for two characters: aa, ab, ac, ...,ba, bb, ... are considered as different characters. This corresponds to having an alphabet with n^2 characters.

This will increase the number of possible keys considerably.

It's still possible to use frequency analysis, but one needs more ciphertext to see statistical patterns.

Block ciphers

- We can use the same definition as for substitution cipher, except that we break a message into blocks of a fixed length.
- We may need to pad the last block to get a full block length.

Good! How many keys do we get depending on block size?
Let's go binary, ie. Measure the block length in bits.

Number of possible keys in a block cipher

Block length in bits	n	5	8	16	256	
Number of possible permutations = number of substitutions of blocks	$(2^n)!$	$2,6 \cdot 10^{35}$	$8,5 \cdot 10^{506}$			
Number of bits to represent an arbitrary substitution	$\log_2((2^n)!) \approx n \log_2 n - n$	113	1678	$9,5 \cdot 10^5$	$3 \cdot 10^{79}$	

Longer blocks means better security, e.g. making statistical analysis more difficult

BUT: Describing arbitrary permutations is impractical for blocks larger than 8 bits.

We need ways to *generate* randomly looking permutations/substitutions from a smaller set of keys.

This will be a major topic for next week!

Some other things that needs mention

Practical calculability of \mathcal{E}, \mathcal{D} , and the need for random key generator Gen

- The encryption and decryption functions should be practically calculable
- Gen must give randomly selected keys k , otherwise security is weakened since an attacker can exploit properties of the key to perform more effective attacks/exploits.
- Cf video-link at start (find time)

Injectivity of encryption function

In order for decryption to be well-defined, the encryption function must be injective. Most often, it will also be surjective, making it bijective.

There are also variants of substitution ciphers that use a larger cipher alphabets, where one can choose between several cipher characters for a given plaintext character. Such systems are called homophonic. Then the encryption is not deterministic, but the decryption is.

It can be used to even out letter frequencies in the cipher text.

We will not cover this in this course.

Kerchoff's principle

To assess a systems security, we shall assume that the only unknown to an adversary, is the key.

Reasons:

- Hard to develop new secure algorithms, than to change keys if they are exposed
- Easier to keep the key secret
- Algorithm can be thoroughly tested and optimized
- Makes standardization possible: Require a given algorithm with known characteristics