Topics in Algebra: Cryptography

Univ.-Prof. Dr. Goulnara ARZHANTSEVA

WS 2019

Cryptography: Overview

Cryptography

- I Past: Diffie–Hellman (1976) and Rivest-Shamir-Adleman (1977)
- II Nowadays: Blockchain ([1991], 2008)
- III Future: Quantum ([1927, 1982], 1983) and Post-quantum cryptography (1994,1996)

Cryptography: Overview

Cryptography

- I Past: Diffie–Hellman (1976) and Rivest-Shamir-Adleman (1977)
- II Nowadays: Blockchain ([1991], 2008)
- III Future: Quantum ([1927, 1982], 1983) and Post-quantum cryptography (1994,1996)

1. Martin, Keith M. Everyday cryptography. Fundamental principles and applications. Second edition. Oxford, 2017.

2. Stinson, Douglas R. Cryptography. Theory and practice. Third edition. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

3. Daniel J. Bernstein & Tanja Lange, Post-quantum cryptography, Nature, 2017, Vol.549, 188–194.

Cryptography: Overview

Cryptography

- I Past: Diffie–Hellman (1976) and Rivest-Shamir-Adleman (1977)
- II Nowadays: Blockchain ([1991], 2008)
- III Future: Quantum ([1927, 1982], 1983) and Post-quantum cryptography (1994,1996)

Cryptography principles

1. Confidentiality: limits access to information
2. Data Integrity: accuracy of data
3. Authentication : confirms the truth of data / entity
4. Non-Repudiation: a technical/legal proof of authorship

Cryptography principles = security services

Confidentiality / secrecy

- limits access to information
- not always required / not alone

Data Integrity

- data was not altered (intentionally or accidentally)
- detection of alteration (not prevention)

Cryptography principles = security services

Data origin authentication / message authentication

- confirms the origin of data with no temporal aspect
- not necessarily an immediate source / not when

Entity authentication

- a given entity is involved and currently active

Cryptography principles = security services

Non-Repudiation

- a source of data cannot deny to a third party being at the origin

Cryptography principles = security services

Non-Repudiation

- a source of data cannot deny to a third party being at the origin

Data origin authentication $\Rightarrow$ Data integrity

Non-Repudiation $\Rightarrow$ Data origin authentication

Data origin authentication $\neq$ Entity authentication

Secrecy $\not\Rightarrow$ Data origin authentication

Cryptography system as a part of a security service

- Cryptography = toolkit
- Cryptographic primitive = a basic tool in this toolkit
  Examples:
  Encryption, hash function, MAC (message authentication code), digital signature, etc.
- Cryptographic algorithm = Cipher = a specification of a primitive
- Cryptographic protocol = a way to choose primitives and use them for a security goal
- Cryptosystem = implementation of primitives and the infrastructure

Cryptosystem: basic model for secrecy

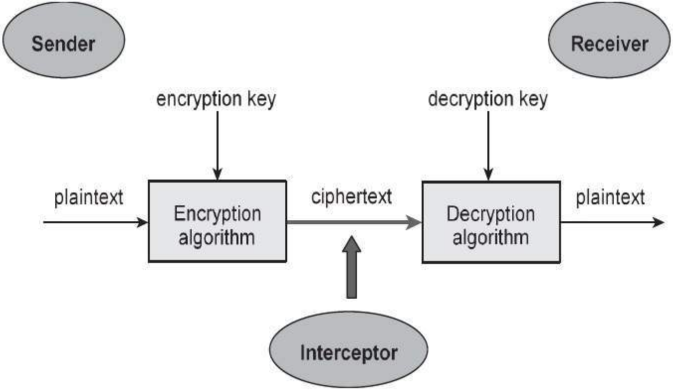

Figure: Basic model of a cryptosystem [image: K. Martin's book]

Cryptosystem: basic model for secrecy

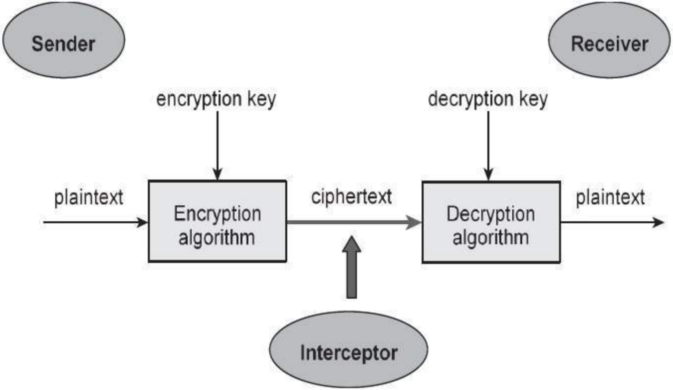

Figure: Basic model of a cryptosystem [image: K. Martin's book]

An interceptor may or may not know the encryption / decryption algorithm and the encryption key. The encryption key is known by the receiver. The decryption key may or may not be known by the sender.

Cryptosystem: basic model for secrecy

Encryption does not prevent communication interception.
For example, it is used over open networks.

Encryption of the communication channel does not guarantee ‘end-to-end’ confidentiality.
For example, the plaintext may be vulnerable.

Cryptosystem: basic model for secrecy

Encryption does not prevent communication interception.
For example, it is used over open networks.

Encryption of the communication channel does not guarantee 'end-to-end' confidentiality.
For example, the plaintext may be vulnerable.

Secrecy can be provided by (combination of):

(1) Cryptography (via encryption)

(2) Steganography (via information hiding)

(3) Access control (via software or hardware)

Cryptography systems for secrecy

$$\text{Encryption key} \overset{?}{\longleftrightarrow} \text{Decryption key}$$

- Symmetric = Secret-key cryptosystem: same keys
- Asymmetric = Public-key cryptosystem: Public vs Private keys

- Theoretical security: mathematics
- Practical security: implementation

Cryptosystem: basic model for secrecy

Definition: Cryptosystem is a 5 -tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ satisfying:

- $\mathcal{P}$ is a finite set of possible plaintexts;
- $\mathcal{C}$ is a finite set of possible ciphertexts;
- $\mathcal{K}$, the keyspace, is a finite set of possible keys;
- $\mathcal{E} = \{E_k : k \in \mathcal{K}\}$ consists of encryption functions $E_k : \mathcal{P} \to \mathcal{C}$;
- $\mathcal{D} = \{D_k : k \in \mathcal{K}\}$ consists of decryption functions $D_k : \mathcal{C} \to \mathcal{P}$;
- For all $e \in \mathcal{K}$ there exists $d \in \mathcal{K}$ such that for all plaintexts $p \in \mathcal{P}$ we have:

$$D_d(E_e(p)) = p$$

- Symmetric cryptosystem: $d = e$
- Public-key cryptosystem: d cannot be derived from e in a computationally feasible way

Cryptography applications

- Securing Internet
- WLAN = Wireless Local Area Network
- Mobile communications (GSM, etc.)
- Payment card transactions
- Video broadcasting
- Identity Cards
- Online Anonimity (Tor, etc.)
- Digital currency
- File protection
- Email security
- Messaging security (WhatsApp, Telegram, etc.)
- Platform security (iOS, etc.)

Breaking encryption algorithms

- A practical method of determining the decryption key is found.

- A weakness in the encryption algorithm leads to a plaintext.

Key lengths and sizes

Length of the key = number of bites it takes to represent the key

Size of the keyspace = number of possible different decryption keys

$$\text{Length} \overset{?}{\longleftrightarrow} \text{Size}$$

- Symmetric: Size $\leqslant 2^{\text{Length}}$

  Example: Size of a 256-bit keyspace is 2^{128} times as big as Size of a 128-bit key.

- Asymmetric: Length is an indication on Size

Exhaustive key search = brute-force attack

1. Select a decryption key from the keyspace

2. Decrypt the ciphertext

3. Check if the plaintext makes sense

4. If ‘yes’ then label the decryption key as a candidate;
otherwise, select a new decryption key

Exhaustive key search = brute-force attack

Assumptions:

– All keys from the keyspace are equally likely to be selected

– The correct decryption key is identified as soon as it is tested

Exhaustive key search = brute-force attack

Assumptions:

– All keys from the keyspace are equally likely to be selected

– The correct decryption key is identified as soon as it is tested

If Size = $n = 2^k$, then, on average, one needs $\sim 2^{k-1}$ attempts to find the correct decryption key:

$$\mathbb{E}[X] = \sum_{i=1}^{n} i \cdot \frac{1}{n} = \frac{n(n+1)}{2} \cdot \frac{1}{n} = \frac{2^k + 1}{2} \sim 2^{k-1}$$

Exhaustive key search = brute-force attack

If Size = $n = 2^k$, then, on average, one needs $\sim 2^{k-1}$ attempts to find the correct decryption key.

1 year = 31556926 seconds $\sim 3 \cdot 10^7$ seconds $\sim 2^{25}$ = 33554432 sec.

$1000 \sim 2^{10} = 1024$ and $1000000 \sim 2^{20} = 1048576$

In 1 year, 1000 processors testing 1000000 keys per second will test in total:

$$\sim 2^{25} \cdot 2^{10} \cdot 2^{20} = 2^{55} \text{ keys}$$

Therefore, a 56-bit key will be enough if the cover time is 1 year.

Cover time = the time for which a plaintext must be kept secret.

Exhaustive key search = brute-force attack

Key lengths needed to protect against a brute-force attack if the cover time is 1 year:

Strength of attack	Key length
Human: one key per second	26 bits
1 processor: 1000000 keys per second	46 bits
1000 processors: each 1000000 keys per second	56 bits
1000000 processors: each 1000000 keys per second	66 bits

Types of attack

Passive attack = unauthorized access to data (remains unnoticed)

- Traffic analysis (location / hosts / frequency / length of messages)
- Release of message contents
- Monitoring processor computations (timing / power analysis)

Active attack = changing the information in an unauthorized way

- Initiating unintended or unauthorized transmission of information.
- Unauthorized deletion of data
- Denial of access to information for legitimate users (denial of service).

Examples of symmetric cryptosystems: Caesar

Caesar Cipher = Shift Cipher Vienna $\xrightarrow{\text{Caesar}}$ Ylhqqd

Replace each alphabet by another alphabet which is 'shifted' by some fixed number between 0 and 25. Key = 'secret shift number'. Length=1

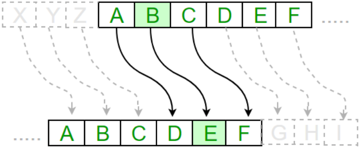

Figure: Caesar Cipher with a shift of 3 [image: geeksforgeeks.org]

Examples of symmetric cryptosystems: Caesar

Caesar Cipher = Shift Cipher Vienna $\xrightarrow{\text{Caesar}}$ Ylhqqd

Replace each alphabet by another alphabet which is 'shifted' by some fixed number between 0 and 25. Key = 'secret shift number'. Length=1

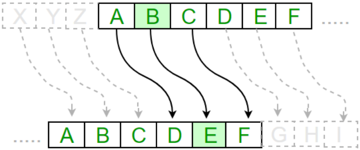

Figure: Caesar Cipher with a shift of 3 [image: geeksforgeeks.org]

Plaintext / Ciphertext: strings of letters (or numbers between 0 and 25)
Encryption / Decryption key: a number between 0 and 25, Size = 26

Ciphertext letter = Plaintext letter + Key mod 26

Examples of symmetric cryptosystems: Substitution

Simple Substitution Cipher Vienna $\xrightarrow{\text{Substitution}}$ Saiffp

Replace each alphabet by another alphabet which is its random permutation. Key = a permutation of 26 letters. Length = 26

Plain alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher alphabet: PHQGIUMEAYLNOFDXJKRCVSTZWB

Plaintext / Ciphertext: strings of letters (or numbers between 0 and 25)
Encryption / Decryption key: a permutation $\sigma \in Sym(26)$, Size = 26!

Ciphertext letter = σ (Plaintext letter)

Examples of symmetric cryptosystems: Substitution

Caesar Cipher is a specific example of Simple Substitution cipher.

$26! = 4.0329146e + 26 \sim 4 \cdot 10^{26} \gg 10^{22}$ = number of stars in universe

Exhaustive key search is currently not feasible.

Simple Substitution Ciphers are examples of monoalphabetic ciphers (each given letter is encrypted into a unique letter).

Simple Substitution Cipher is breakable by Letter frequency analysis. (A long enough plaintext is required.)

A large keyspace is necessary but not sufficient for security.

Example: Letter frequency analysis

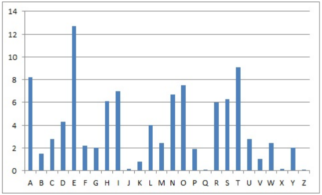

Figure: English Letter Frequencies. [image: Crypto Corner]

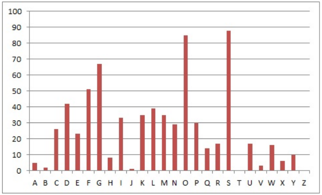

Figure: Ciphertext letter frequencies [image: Crypto Corner]

Examples of symmetric cryptosystems: Vigenère

Vigenère Cipher Vienna $\xrightarrow{\text{Vigenère}}$ Bwyyaa

Generate a key by repeating a given key until it matches the length of the plaintext. Replace each plaintext letter by another letter using a Caesar Cipher, whose key is the number associated to the corresponding letter of the generated key. Key = a string of letters.

Plaintext: U N I V E R S I T Y Key: GOULNARA
Generated key: G O U L N A R A G O
Ciphertext: A B C G R R J I Z M

Plaintext / Ciphertext: strings of letters (or numbers between 0 and 25)
Encryption / Decryption generated key length = length of the plaintext

$$\text{Ciphertext letter}_i = \text{Plaintext letter}_i + \text{Key}_i \mod 26$$

Examples of symmetric cryptosystems: Vigenère

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

Figure: Vigenère Cipher table [image: geeksforgeeks.org]

Examples of symmetric cryptosystems: Vigenère

Vigenère Cipher is an example of polyalphabetic ciphers (each given letter can be encrypted into 'length of the key' different letters).

Same letter is encrypted differently depending on its position in the plaintext. Hence, a natural letter frequency analysis is not feasible.

For large enough plaintexts the exhaustive key search is currently not feasible.

Vigenère Cipher is breakable by breaking a sequence of Caesar Ciphers in a strict rotation. (A length of the given key is required.)

Enigma machine: a sequence of component substitution encryption processes in rotation, using a long key.

Test questions

Question 1

Give an example of an application where
(i) entity authentication and data origin authentication are both required;
(ii) data origin authentication is required but not data integrity.

Question 2

If the given key of a Vigenère Cipher has repeated letters, does it make it any easier to break?

Question 3

Invent and analyze (length, size, attacks?) an Affine Cipher.

Computational complexity

Operation	Complexity
Addition of two n-bit numbers	n
Multiplication of two n-bit numbers	n^2
Raising a number to an n-bit power	n^3
Exhaustive key search for an n-bit key	2^n

Complexity of multiplication

$$\sum_{0\leqslant k\leqslant n-1} a_k \cdot 2^k \times \sum_{0\leqslant \ell\leqslant n-1} b_\ell \cdot 2^\ell = \sum_{0\leqslant m\leqslant 2(n-1)} c_m \cdot 2^m, c_m = \sum_{k+l=m} a_k b_\ell$$

Calculation of each c_m requires $\leqslant 2n-1$ elementary multiplications and $\leqslant 2n-2$ additions and corresponding carries, thus the algorithm requires less than $2n \cdot 4n$ steps, hence, at most quadratic complexity.

Computational complexity of attacks

We can estimate real attack times.

Assumption: computer makes 1 000 000 operations per second

Exhaustive key search real attack time for a 30-bit key

$$\frac{2^{30}}{10^6} \text{ sec.} = 1073.741824 \text{ seconds} = 17.8956970667 \text{ minutes}$$

Computational complexity is an indication on a real attack time, on a computational security.

Test questions

Question 4

How long (in years, days, hours, seconds) it will take 1000000 computers, each processing 1000000 operations per second, to

(1) multiply two 1000-bit numbers together;

(2) perform an exhaustive search for a 128-bit key;

(3) find the correct key (on average) while performing a brute-force attack on a 128-bit key.

Evaluating security

Computational security: computational complexity is high.

Provable security: breaking the cryptosystem would solve a problem known to be hard.

Unconditional security: breaking is not possible even if computational resources are unlimited.

Perfect secrecy

A cryptosystem has perfect secrecy if seeing the ciphertext gives not extra information about the plaintext.

A cryptosystem with perfect secrecy is unconditionally secure against a ciphertext only attack.

Probability distributions on plaintexts and keyspace

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem and probability distributions are given on $\mathcal{P}$ and $\mathcal{K}$:

$\Pr[\mathbf{p} = p]$ denotes the probability that a plaintext $p \in \mathcal{P}$ occurs,

$\Pr[\mathbf{k} = k]$ denotes the probability that a key $k \in \mathcal{K}$ is chosen.

Analogously, $\Pr[\mathbf{c} = c]$ denotes the probability that a ciphertext $c \in \mathcal{C}$ transmitted.

Assumptions:
– the key and the plaintext are independent random variables;
– each key is used for only one encryption.

Probability distribution on ciphertexts

For $k \in \mathcal{K}$, let $C(k) := \{E_k(p) : p \in \mathcal{P}\}$ be the set of possible ciphertexts if k is the key. Then $\forall c \in \mathcal{C}$ we have:

$$\Pr[\mathbf{c} = c] = \sum_{\{k : c \in C(k)\}} \Pr[\mathbf{k} = k] \Pr[\mathbf{p} = D_k(c)]$$

Then: $$\Pr[\mathbf{c} = c \mid \mathbf{p} = p] = \sum_{\{k : p = D_k(c)\}} \Pr[\mathbf{k} = k]$$

Probability distribution on ciphertexts

For $k \in \mathcal{K}$, let $C(k) := \{E_k(p) : p \in \mathcal{P}\}$ be the set of possible ciphertexts if k is the key. Then $\forall c \in \mathcal{C}$ we have:

$$\Pr[\mathbf{c} = c] = \sum_{\{k \,:\, c \in C(k)\}} \Pr[\mathbf{k} = k] \Pr[\mathbf{p} = D_k(c)]$$

Then: $$\Pr[\mathbf{c} = c \mid \mathbf{p} = p] = \sum_{\{k \,:\, p = D_k(c)\}} \Pr[\mathbf{k} = k]$$

Using Bayes' theorem $\left(\Pr[X \mid Y] = \dfrac{\Pr[X]\Pr[Y \mid X]}{\Pr[Y]} \text{ if } \Pr[Y] > 0\right)$:

$$\Pr[\mathbf{p} = p \mid \mathbf{c} = c] = \frac{\Pr[\mathbf{p} = p] \displaystyle\sum_{\{k \,:\, p = D_k(c)\}} \Pr[\mathbf{k} = k]}{\displaystyle\sum_{\{k \,:\, c \in C(k)\}} \Pr[\mathbf{k} = k] \Pr[\mathbf{p} = D_k(c)]}$$

Perfect secrecy

Definition: Perfect secrecy — Shannon'49

A cryptosystem has perfect secrecy if $\Pr[\mathbf{p} = p \mid \mathbf{c} = c] = \Pr[\mathbf{p} = p]$ for all $p \in \mathcal{P}, c \in \mathcal{C}$.

Proposition:

TFAE:

1. $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ has perfect secrecy;
2. random variables $\mathbf{p}$ and $\mathbf{c}$ are independent;
3. $\Pr[\mathbf{c} = c \mid \mathbf{p} = p] = \Pr[\mathbf{c} = c]$;
4. $\forall p_1, p_2 \in \mathcal{P} \quad \Pr[\mathbf{c} = c \mid \mathbf{p} = p_1] = \Pr[\mathbf{c} = c \mid \mathbf{p} = p_2]$

In particular, a cryptosystem has perfect secrecy independently of the language used in the plaintext (prob. distribution on $\mathcal{P}$ is irrelevant).

Perfect secrecy: Example

$\mathcal{P} = \{a, b\}$ with $\Pr[a] = 1/4$, $\Pr[b] = 3/4$ and $\mathcal{C} = \{1, 2, 3, 4\}$

$\mathcal{K} = \{k_1, k_2, k_3\}$ with $\Pr[k_1] = 1/2$, $\Pr[k_2] = \Pr[k_3] = 1/4$.

Let the encryption be defined by:

E_k	a	b
k_1	1	2
k_2	2	3
k_3	3	4

Perfect secrecy: Example

$\mathcal{P} = \{a, b\}$ with $\Pr[a] = 1/4, \quad \Pr[b] = 3/4$ and $\mathcal{C} = \{1, 2, 3, 4\}$

$\mathcal{K} = \{k_1, k_2, k_3\}$ with $\Pr[k_1] = 1/2, \quad \Pr[k_2] = \Pr[k_3] = 1/4$.

Let the encryption be defined by:

E_k	a	b
k_1	1	2
k_2	2	3
k_3	3	4

Then the induced probability distribution on $\mathcal{C}$ is defined, e.g.
$\Pr[2] = 7/16, \Pr[3] = 1/4$, etc.

Perfect secrecy: Example

$\mathcal{P} = \{a, b\}$ with $\Pr[a] = 1/4$, $\Pr[b] = 3/4$ and $\mathcal{C} = \{1, 2, 3, 4\}$

$\mathcal{K} = \{k_1, k_2, k_3\}$ with $\Pr[k_1] = 1/2$, $\Pr[k_2] = \Pr[k_3] = 1/4$.

Let the encryption be defined by:

E_k	a	b
k_1	1	2
k_2	2	3
k_3	3	4

Then the induced probability distribution on $\mathcal{C}$ is defined, e.g. $\Pr[2] = 7/16, \Pr[3] = 1/4$, etc.
Then the conditional probability distributions on the plaintext, given a certain ciphertext can be computed, e.g. $\Pr[b \mid 2] = 6/7$, etc.

Perfect secrecy: Example

$\mathcal{P} = \{a, b\}$ with $\Pr[a] = 1/4$, $\Pr[b] = 3/4$ and $\mathcal{C} = \{1, 2, 3, 4\}$

$\mathcal{K} = \{k_1, k_2, k_3\}$ with $\Pr[k_1] = 1/2$, $\Pr[k_2] = \Pr[k_3] = 1/4$.

Let the encryption be defined by:

E_k	a	b
k_1	1	2
k_2	2	3
k_3	3	4

Then the induced probability distribution on $\mathcal{C}$ is defined, e.g. $\Pr[2] = 7/16, \Pr[3] = 1/4$, etc.
Then the conditional probability distributions on the plaintext, given a certain ciphertext can be computed, e.g. $\Pr[b \mid 2] = 6/7$, etc.
Hence, this cryptosystem has no perfect secrecy (although, it has it on a specific ciphertext $c = 3$).

Perfect secrecy: Shannon's theorem

Theorem: Perfect secrecy Shannon'49

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then it has perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Perfect secrecy: Shannon's theorem

Theorem: Perfect secrecy Shannon'49

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then it has perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Proof: $(\Rightarrow)$ We can assume that $\forall p \in \mathcal{P} \Pr[p] > 0, \forall c \in \mathcal{C} \Pr[c] > 0$.

Perfect secrecy: Shannon's theorem

Theorem: Perfect secrecy Shannon'49

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then it has perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Proof: $(\Rightarrow)$ We can assume that $\forall p \in \mathcal{P} \; \Pr[p] > 0, \forall c \in \mathcal{C} \; \Pr[c] > 0$. Fix $p \in \mathcal{P}$. For each $c \in \mathcal{C}$, we have $\Pr[c \mid p] = \Pr[c] > 0$, that is, $\forall c \in \mathcal{C}$ there is at least one $k \in \mathcal{K}$ with $E_k(p) = c$.

Perfect secrecy: Shannon's theorem

Theorem: Perfect secrecy Shannon'49

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then it has perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Proof: $(\Rightarrow)$ We can assume that $\forall p \in \mathcal{P} \Pr[p] > 0, \forall c \in \mathcal{C} \Pr[c] > 0$. Fix $p \in \mathcal{P}$. For each $c \in \mathcal{C}$, we have $\Pr[c \mid p] = \Pr[c] > 0$, that is, $\forall c \in \mathcal{C}$ there is at least one $k \in \mathcal{K}$ with $E_k(p) = c$.

Therefore, $|\mathcal{C}| = |\{E_k(p) \mid k \in \mathcal{K}\}| \leqslant |\mathcal{K}|$ and, as $|\mathcal{K}| = |\mathcal{C}|$, there is no distinct $k_1 \neq k_2$ with $E_{k_1}(p) = E_{k_2}(p) = c$. That is, $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Perfect secrecy: Shannon's theorem

Theorem: Perfect secrecy Shannon'49

Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then it has perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

Proof: $(\Rightarrow)$ We can assume that $\forall p \in \mathcal{P} \Pr[p] > 0, \forall c \in \mathcal{C} \Pr[c] > 0$. Fix $p \in \mathcal{P}$. For each $c \in \mathcal{C}$, we have $\Pr[c \mid p] = \Pr[c] > 0$, that is, $\forall c \in \mathcal{C}$ there is at least one $k \in \mathcal{K}$ with $E_k(p) = c$.

Therefore, $|\mathcal{C}| = |\{E_k(p) \mid k \in \mathcal{K}\}| \leqslant |\mathcal{K}|$ and, as $|\mathcal{K}| = |\mathcal{C}|$, there is no distinct $k_1 \neq k_2$ with $E_{k_1}(p) = E_{k_2}(p) = c$. That is, $\forall p \in \mathcal{P}$, $\forall c \in \mathcal{C}$, there is a unique key $k \in \mathcal{K}$ such that $E_k(p) = c$.

(Analogously, $|\mathcal{P}| \leqslant |\mathcal{K}|$.)

Perfect secrecy: Shannon's theorem (continued)

Let $n = |\mathcal{K}|, \mathcal{P} = \{p_1, \ldots, p_n\}$, and $c \in \mathcal{C}$ be fixed. Let $k_i \in \mathcal{K}$ be so that $E_{k_i}(p_i) = c$. Using Bayes' theorem:

$$\Pr[p_i \mid c] = \frac{\Pr[c \mid p_i]\Pr[p_i]}{\Pr[c]} = \frac{\Pr[k_i]\Pr[p_i]}{\Pr[c]}.$$

Perfect secrecy implies that $\forall i \quad \Pr[k_i] = \Pr[c]$, all keys are used with equal probability. Since there are $|\mathcal{K}|$ keys, the probability is $1/|\mathcal{K}|$.

($\Leftarrow$) $\forall p \in \mathcal{P}, \forall c \in \mathcal{C} \quad \Pr[c \mid p] = 1/|\mathcal{K}|$, hence, we conclude by the Proposition. ■

One-time pad

Definition: One-time pad Vernam'1917

Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}/2\mathbb{Z})^n$ and $E_k(p) = k + p \mod 2$.

One-time pad has perfect secrecy:

$$\forall p \in \mathcal{P}, \forall c \in \mathcal{C} \quad \Pr[c \mid p] = 1/|\mathcal{K}|,$$

hence, we conclude by the Proposition (alternatively, one can use Shannon's theorem).

Test questions

Question 5

(1) Does one-time pad remain with perfect secrecy if we reuse the same key twice?

(2) Has Vigenère Cipher perfect secrecy?

(3) Could we use one-time pads in practice?

Symmetric encryption

- DES = Data Encryption Standard'1975
- AES = Advanced Encription Standard'2000

Asymmetric encryption: Public-key encryption

- RSA = Rivest-Shamir-Adleman cryptosystem'[1970] 1977
- ECC = Elliptic curves cryptography'[1985] 2004

Asymmetric encryption: Public-key encryption

- RSA = Rivest-Shamir-Adleman cryptosystem'[1970] 1977
- ECC = Elliptic curves cryptography'[1985] 2004

Public-key cryptosystem can never provide unconditional security. Therefore, we study the computational security of public-key cryptosystems.

RSA cryptosystem

Definition: RSA cryptosystem

Let $n = pq$, where p, q are primes. Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}/n\mathbb{Z}$ and

$$\mathcal{K} = \{(n, p, q, a, b) \ : \ ab = 1 \mod \phi(n)\}$$

For $k = (n, p, q, a, b)$, we define

$$E_k(x) = x^b \mod n \ \text{ and } \ D_k(c) = c^a \mod n.$$

Public-key is (n, b) and private-key is (p, q, a).

Here, x is a plaintext.

Euler's function $\phi(n)$ = the number of positive integers less than n and relatively prime to n.

RSA cryptosystem

Encryption and decryption are inverse operations.

$n = pq \Rightarrow \phi(n) = (p-1)(q-1)$
We have that $ab = 1 \mod \phi(n)$, i.e. $ab = t\phi(n) + 1$ for some $t \in \mathbb{Z}$.

RSA cryptosystem

Encryption and decryption are inverse operations.

$n = pq \Rightarrow \phi(n) = (p-1)(q-1)$
We have that $ab = 1 \mod \phi(n)$, i.e. $ab = t\phi(n) + 1$ for some $t \in \mathbb{Z}$.
(1) Suppose that $x \in (\mathbb{Z}/n\mathbb{Z})^*$, then

$$(x^b)^a = x^{t\phi(n)+1} \mod n = (x^{\phi(n)})^t x \mod n = 1^t x \mod n = x \mod n.$$

RSA cryptosystem

Encryption and decryption are inverse operations.

$n = pq \Rightarrow \phi(n) = (p-1)(q-1)$
We have that $ab = 1 \mod \phi(n)$, i.e. $ab = t\phi(n) + 1$ for some $t \in \mathbb{Z}$.
(1) Suppose that $x \in (\mathbb{Z}/n\mathbb{Z})^*$, then

$$(x^b)^a = x^{t\phi(n)+1} \mod n = (x^{\phi(n)})^t x \mod n = 1^t x \mod n = x \mod n.$$

(2) If $x \notin (\mathbb{Z}/n\mathbb{Z})^*$, then $x = 0 \mod p$ or $x = 0 \mod q$.

If $x = 0 \mod p$, then $(x^b)^a = 0 \mod p$ as well. If the same holds for mod q we are done by the Chinese remainder theorem.

Otherwise, $x \neq 0 \mod q$. Then, by Fermat's little theorem, $(x^b)^a = x^{ba-1}x = x^{t(p-1)(q-1)}x = (x^{q-1})^{t(p-1)}x = 1^{t(p-1)}x \mod q = x \mod q$. We conclude by the Chinese remainder theorem.