Introduction – Cryptography and Secured Communications –

Lionel Morel

Telecommunications - INSA Lyon

Fall-Winter 2022-23

Introduction

Lecturer - Lionel Morel (lionel.morel@insa-lyon.fr)

- MSc in Computer Science - Grenoble 2001
- PhD in CS at INPGrenoble - Programming of Critical Reactive Systems
- Associate Professor at INSA Lyon since 2007.
- (past) Research topics:
  - at Grenoble, Turku (Finland), Rennes, and Lyon: Models of concurrency and computations, programming languages, performance analysis for parallel multi-core architectures.
  - at CEA-Grenoble (2017-2020): **Counter-measures against physical attacks (side-channel, fault-injection, etc)**
- Current Research: **operating systems** and programming languages **for** addressing so-called **frugality**, Phenix [a]
- Teaching at the IF department: Computer Architecture, Operating Systems, Compiler Construction

[a] `https://phenix.citi-lab.fr/`, `lionel.morel.ouvaton.org`

Un petit détour

Course Objectives

Course Objectives

Give you some “necessary and sufficient” background on:

- Cryptography
- Cryptographic protocols
- Public-key infrastructures
- Associated ethical issues

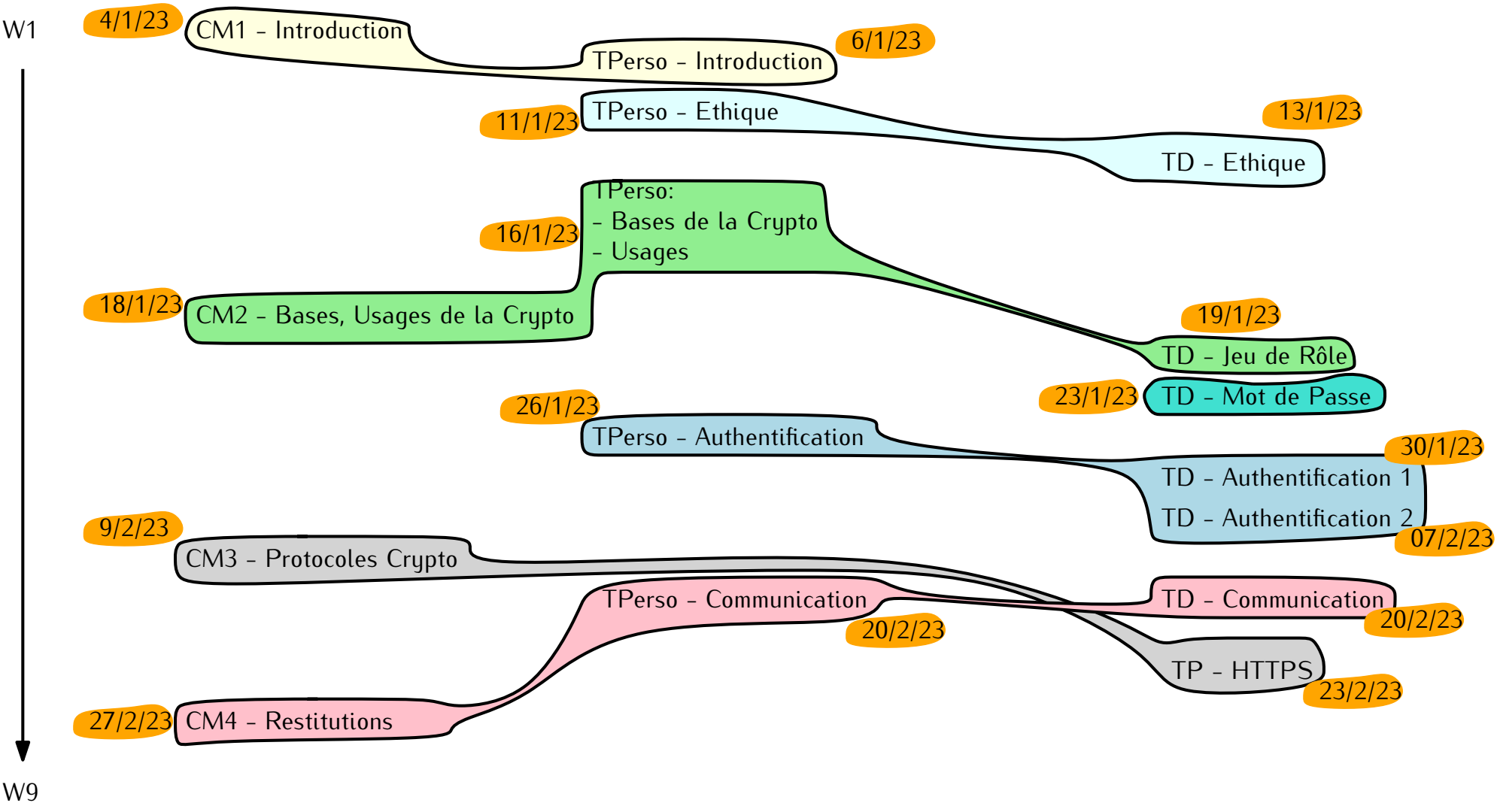
But:

- Security is a vast topic, covered by **several years** of studies if you want to specialize
- You will not be a specialist, but a **enlightened neophytes**.

$\Rightarrow$

- Please don’t change cryptography yourself
- **Go and ask** a specialist

Course Plan


W1
4/1/23
CM1 - Introduction
6/1/23
TPerso - Introduction
11/1/23
TPerso - Ethique
13/1/23
TD - Ethique
16/1/23
TPerso:
- Bases de la Crypto
- Usages
18/1/23
CM2 - Bases, Usages de la Crypto
19/1/23
TD - Jeu de Rôle
23/1/23
TD - Mot de Passe
26/1/23
TPerso - Authentification
30/1/23
TD - Authentification 1
TD - Authentification 2
07/2/23
9/2/23
CM3 - Protocoles Crypto
TPerso - Communication
TD - Communication
20/2/23
20/2/23
TP - HTTPS
23/2/23
27/2/23
CM4 - Restitutions
W9

General Considerations

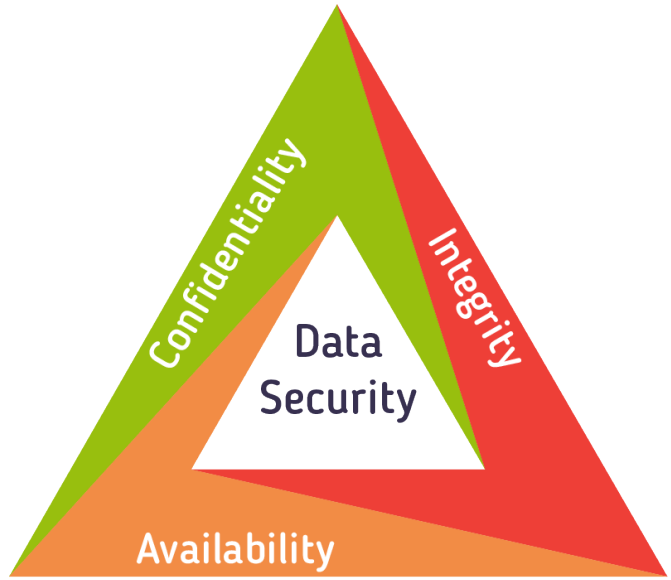
Information Security

- **Information security** $\triangleq$ practice of protecting information by mitigating information risks[1]
- Need to protect all elements dealing with information: computers, networks, people
- Security covers a lot of different aspects: physical security, social engineering, communication security, etc.
- $\triangleq$ practice that allows to maintain the CIA triad (see next)

[1] https://en.wikipedia.org/wiki/Information_security

The CIA Triad

- **Confidentiality**: Information is not made available or disclosed to unauthorized individuals, entities, or processes.[2]
- **Integrity**: Information is not modified in an unauthorized or undetected manner. Also called **anti-tampering**.
- **Availability**: Information is available when it is needed.



[2] Beckers, K. (2015). Pattern and Security Requirements: Engineering-Based Establishment of Security Standards.

Threats

- A **threat** is a potential negative action or event that can result in unwanted impact to a computer system, application or user information.
- A **threat model** is a set of properties that characterize threats associated to a particular environment. Often implies **security requirements** on a system.

Vulnerabilities

- A **vulnerability** is a weakness which can be exploited by an attacker to access unauthorized information or to compromise the attacked system's behavior.
- The **attack surface** of a system/application is the set of (known) vulnerabilities exposed by it to a potential attacker.

Attacks

- **Attack** = Attempt to exploit a vulnerability
- Attack can be:
  - Passive (eavesdropping, side-channel, etc)
  - Active (worm, faults, etc)
  - Denial-of-service, ie render the service unusable.
- When the attack is successful, we say the system is **compromised**

Trust

- **Trust** = Degree to which an entity (person, system, hardware, software) is going to behave as expected
- A **Trust model** describes which entity(ies) is/are trusted and at which level.

The Attacker's Toolbox

Attack Examples

- **Trojan**: a malevolent binary that pretends to be something else
- **Worm**: self-replicates to propagate to other computer hosts
- **Virus**: replicates itself by modifying other programs to insert its own code
- **Buffer Overflow**: use adjacent placement of data in memory to modify some private data by writing to public data:

variable name	A								B	
value	[null string]								1979	
hex value	00	00	00	00	00	00	00	00	07	BB

```
char           A[8] = "";
unsigned short B    = 1979;
...
strcpy(A, "excessive");
```

variable name	A								B	
value	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	25856	
hex	65	78	63	65	73	73	69	76	65	00

Buffer Overflow

$\triangleq$ "Anomaly whereby a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations."[3]

- Different types: stack-based, heap-based, format-string attack
- Different consequences: private data corruption, arbitrary code execution, etc

[3] https://en.wikipedia.org/wiki/Buffer_overflow

Buffer Overflow Example

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  // I have a "secret" variable, statically allocated
  char secretBuf[9]= {'p','r','o','t','e','c','t','e','d'};
  // This is an input Buffer (5 chars including "end of line")
  char inputBuffer[5];

  // a prompt how to execute the program...
  if (argc < 2)
        {
          printf("strcpy() NOT executed....\n");
          printf("Syntax: %s <characters>\n", argv[0]);
          exit(0);
        }

  // copy the user input to my input buffer, without any
  // bound checking
  strcpy(inputBuffer, argv[1]);
  printf("buffer content= %s\n", inputBuffer);

  printf("secret Buf = %s\n", secretBuf);

  return 0;
}
```

To test:

```
gcc bufover.c -o buf
./buf spraythis
```

outputs:

```
buffer content= spraythis
secret Buf = this
```

Format-String Attack

≙ “[...] occurs when the submitted data of an input string is evaluated as a command by the application.”[4]

```c
#include  <stdio.h>

void main(int argc, char **argv)
{
        // This line is safe
        printf("%s\n", argv[1]);
        printf(argv[1]);
}
```

To test:

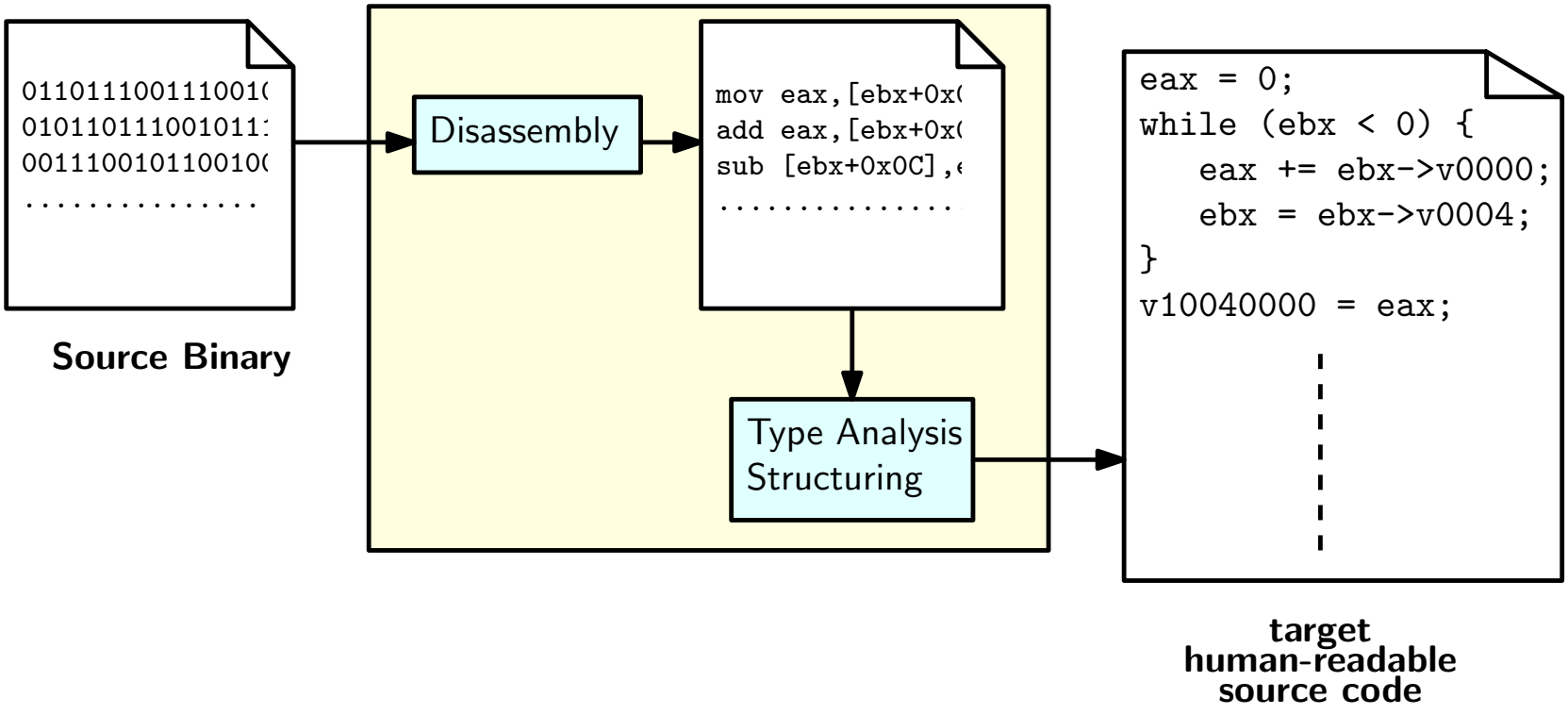
```
gcc formatstring.c -o formats
./formats "Hello World %p %p %p %p %p %p %p
```

outputs:

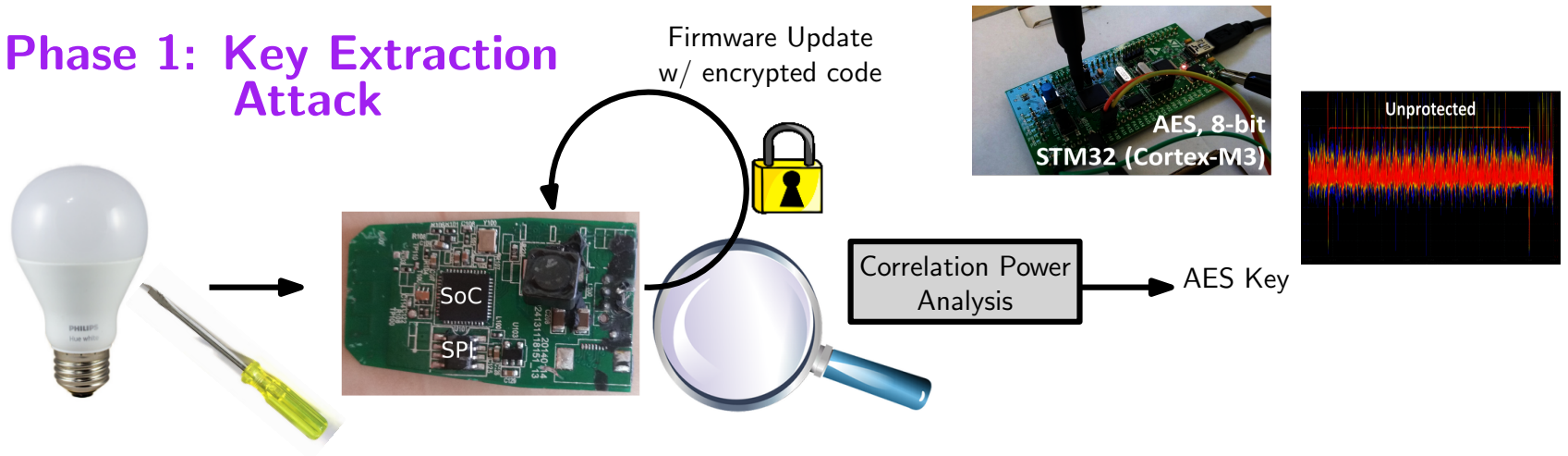
```
Hello World %p %p %p %p %p %p %p %p
Hello World 0x1 0x1 0x7fcbf1fdfb23 0x3 0x77
```

[4] https://owasp.org/www-community/attacks/Format_string_attack

Reverse Engineering

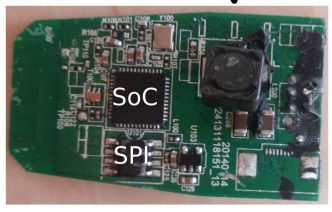


Physical Attack Examples



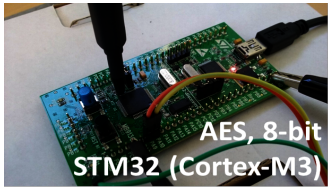
Physical Attack Examples

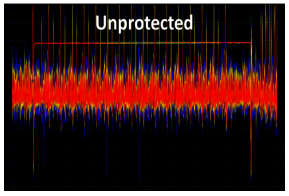
Phase 1: Key Extraction Attack



Firmware Update
w/ encrypted code

Correlation Power Analysis → AES Key





Phase 2: Firmware Update Takeover

ZigBee range = 400m
Take Over ONE light bulb
Propagate worm through lightbulbs

The Defender's Toolbox

Defenses - a quick panorama

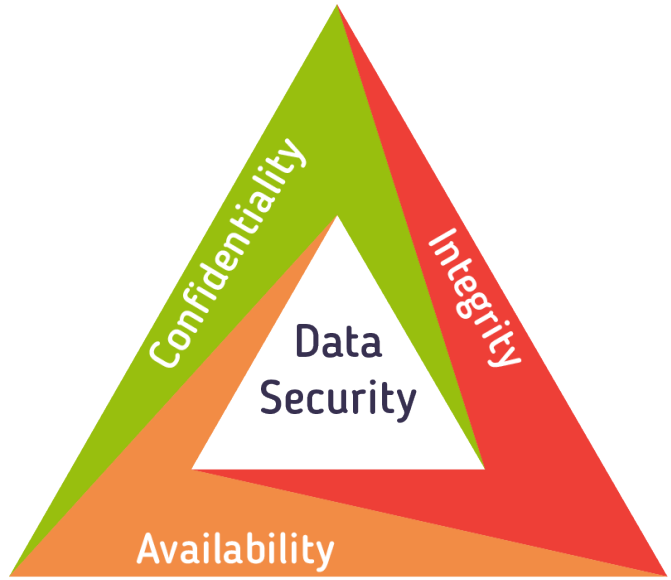
- Cryptography: how to encrypt data
- Secured communication protocols: how to encrypt data + share keys
- Physical shielding: how to protect device from physical alteration
- **ICI il y a du travail**

Definition - Communication Security [5]

Confidentiality

Communication Security $\triangleq$ discipline of preventing unauthorized interceptors from accessing telecommunications in an intelligible form, while still delivering Availability un-altered content to the intended recipients.

Integrity



[5] https://en.wikipedia.org/wiki/Communications_security

Cryptology

Cryptology, is the science of practice and study of techniques for secure communication in the presence of adversarial behavior.

- **Cryptography**: Practice and study of techniques for secure communication in the presence of adversarial behavior.
- **Cryptanalysis**: Process of analyzing information systems in order to understand hidden aspects of the systems.
- **Cryptology = Cryptography + Cryptanalysis**

In this course, we mainly focus on **Cryptography**.

History

A brief history of cryptography

- Keeping message secret has always been a (powerful) men's concern ...
- ... but (at least today) it's also of every person's interest.
- ... because there is no "I got nothing to hide"

History (1) The Skytale

- Oldest cryprtographic device known (-404 BC)
- Write a message on a leather strap
- Wrap the strap around a rod with correct diameter
- Key = Shape of the rod (diameter, number of sides)

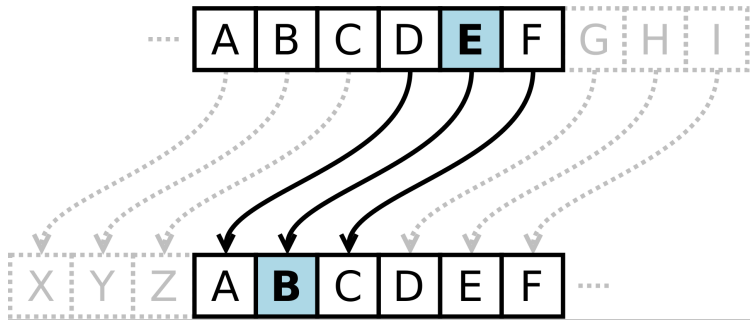
History (1) Caesar cipher

- Substitution cipher
- Each letter is encoded with its order in the alphabet: A$\rightarrow$0, B$\rightarrow$1, ..., Z$\rightarrow$26
- We choose a **fixed *shift value*** *sh*
- To **encrypt**, each letter P_i in Plaintext is replaced by the corresponding shifted letter:

$$\boldsymbol{E}(\boldsymbol{P_i}) = (\boldsymbol{P_i} + \boldsymbol{sh}) \textbf{ mod } \boldsymbol{26}$$

- To **decrypt**, each letter C_i in the Ciphertext is converted back with :

$$\boldsymbol{D}(\boldsymbol{C_i}) = (\boldsymbol{C_i} - \boldsymbol{sh}) \textbf{ mod } \boldsymbol{26}$$



History (1) Caesar cipher

- ⊕ Encryption and decryption are cheap
- ⊖ Easy to crack with frequency analysis
- ⊕/⊖ Sufficient when no-one around can read :) (in particular, what's the difference between a foreign language and an encrypted language, if you can't read the first).

General case: Substitution cipher

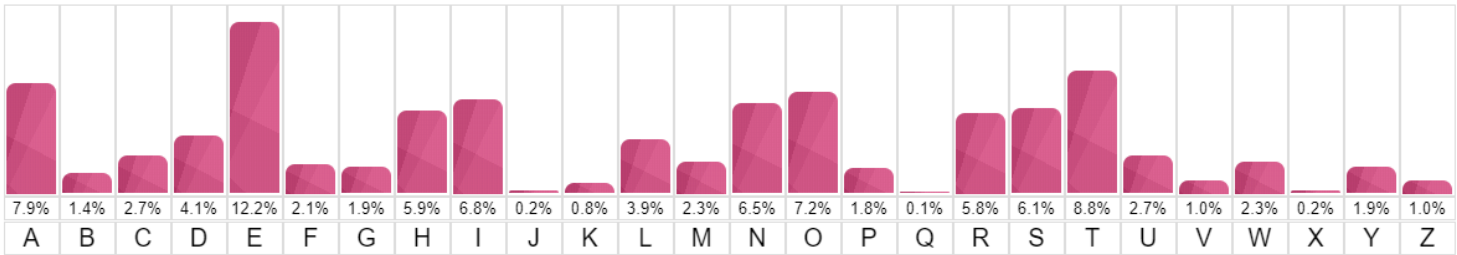
Principle

Replace a letter by another

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
AZERTYUIOPQSDFGHJKLMWXCVBN
```

Attack

- Frequency analysis
- Each letter in a given language has a specific occurrence frequency
- Replace letter with frequency *f* in the encrypted text by letter with frequency *f* in the original alphabet



Vigenère cipher (1/3) - Principle

- Invented XVIth century
- Based on a **Vigenere table 26x26** *VigT* :
  Each line starts by a different letter $\mathcal{L}$ of the alphabet and contains the whole alphabet in the usual order, starting from $\mathcal{L}$ and looping back from A to Z ....

Clé Utilisée \ Lettre en clair	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

Lettre chiffrée

Vigenère cipher (2/3) - Principle

- Let's send a message m of length $l(m)$:
  m = killthekingtonight, $l(m) = 18$
- Choose a key k of $l(k)$ characters :
  k = HORSE, $l(k) = 5$
- Repeat the key until you reach k' of length $l(k') = l(m)$
  k' = HORSEHORSEHORSEHOS
- encoded letter m_i by replacing it by $VigT[k'_i][m_i]$:
  cipher(m,k) = RWCDXOSBARNHFFMNVL

Strength: disguise the plaintext's letter frequency to interfere with frequency analysis.

Vigenère cipher (3/3) - Kasiski's Attack (1863)

- Some repeated word may be encrypted using the same key letters:
  Ciphertext: **CSASTP**KVSIQUTGQU**CSASTP**IUAQJB
- Distance between repetitions of **CSASTP** = 16
- Assume repeated segments in the ciphertext encode the same plaintext
- Key length is 16, 8, 4, 2 or 1 ... 1 and 2 are too simple.
- We know the key starts by A, B, C, D ... Let's try all possible keys.
- Quite quickly, we find that:
  Plaintext: **CRYPTO**ISSHORTFOR**CRYPTO**GRAPHY

The longer the ciphertext, the more accurate the analysis

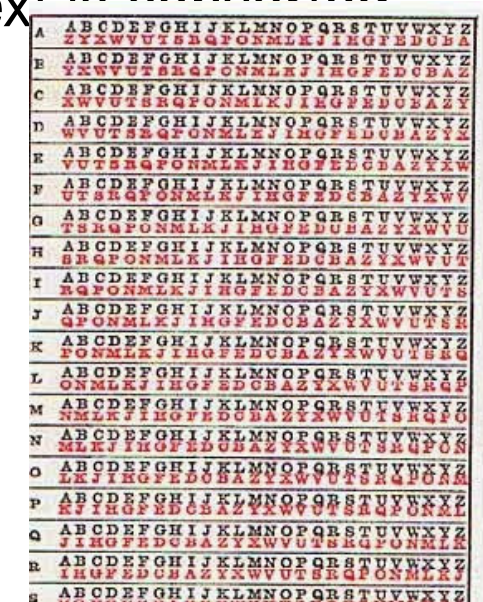
One-time pad (1/3)

- Invented in 1882
- Substitution cipher
- Choose a **random key** K at least as long as the plaintext
- To **encrypt**, each letter P_i in Plaintext is replaced by the corresponding shifted letter:

$$\boldsymbol{E(P_i) = (P_i + K_i)\ \textbf{mod 26}}$$

- To **decrypt**, each letter C_i in the Ciphertext is converted back with :

$$\boldsymbol{D(C_i) = (C_i - K_i)\ \textbf{mod 26}}$$



One-time pad (2/3) - Pros and Cons

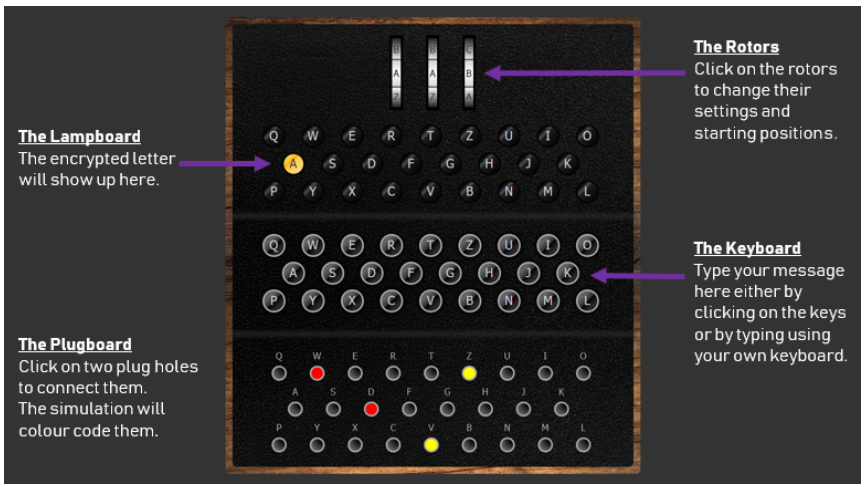
- ➕ Proven secure
- ➕ Even to frequency analysis
- ➕ Encryption and decryption are cheap
- ⛔ Fresh key is needed for every plaintext
- ⛔ Key must be as long as the plaintext
- ⛔ Key must be kept secret
- ⛔ Key must not be lost (not by one character)
- ⛔ Key must be truly random

One-time pad (3/3) - a long lasting history

- **1920**: Weimar Republic Diplomatic Service
- around **1930**: Soviet Union (after breaking of own cryptography by the British)
  KGB spies used them until the 1950s and 1960s
- during **WWII**: used in the SIGSALY secure speech system for high-level Allied communications
- from **1963**: (after the Cuban Missile Crisis):
  Moscow-Washington-DC hotline used teleprinters
- During the **1970s**: the NSA used them extensively
- from **1988**: The African National Congress to communicate between ANC leaders outside South Africa and in-country operatives

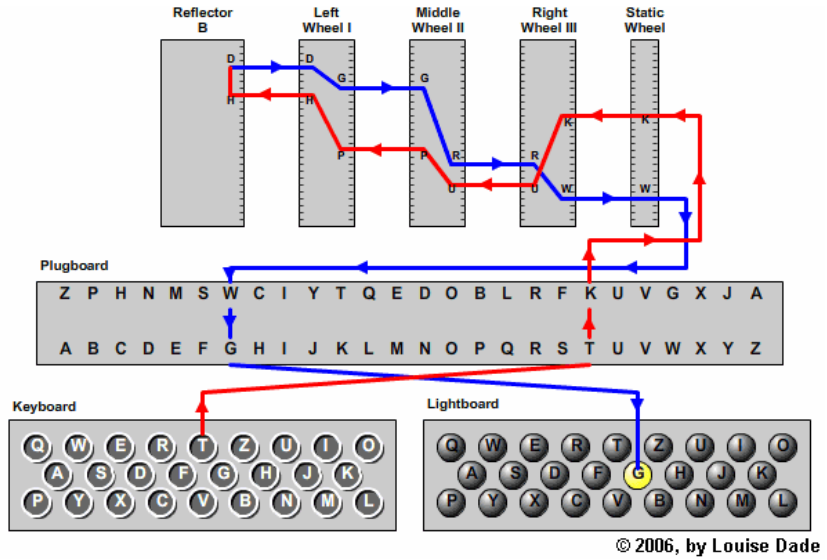
Enigma

- Invented at the end of WWI
- Used extensively by Nazi Germany during WWII
- First cracked by Polish services during the early 30s ...
- ... then by British-led effort at Bletchley Park, including Alan Turing.

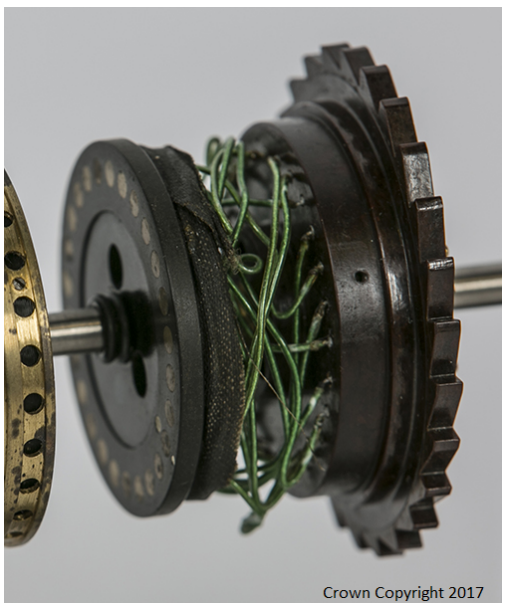


Enigma - How does it work?

- Substitution cipher
- Originally patented in 1918



- All three wheels are wired differently
- Wheels are initialized to any position
- The reflector never wires a letter to itself
- Army-grade version :
  - Choose 3 wheels amongst a set of 6
  - Plugboard
  - Change the initial rotor setting



Enigma - Combinatorial

Every day, the machine is reset to a pre-established configuration:

- 60 Rotors choice (3 or 4 or 5 among 6 possible).
- 26^3 Rotors initial letter combination
- Plug-board settings:

$$\frac{26!}{6!10!2^{10}}$$

- The initial setting for a specific day, use a pre-printed paper codebook (gives initial configuration)

Geheime Kommandosache!
Nicht ins Flugzeug mitnehmen

Armee-Stabs-Maschinenschlüssel Nr. 28
für Oktober 1944

Nr. 00008

	Datum	Walzenlage			Ringstellung			Steckerverbindungen										Kenngruppen			
St	31.	IV	V	I	21	15	16	KL	IT	FQ	HY	XC	NP	VZ	JB	SE	OG	jkm	ogi	ncj	glp
St	30.	IV	II	III	26	14	11	ZN	YO	QB	ER	DK	XU	GP	TV	SJ	LM	ino	udl	nam	lax
St	29.	II	V	IV	19	09	24	ZU	HL	CQ	WM	OA	PY	EB	TR	DN	VI	nci	oid	yhp	nip
St	28.	IV	III	I	03	04	22	YT	BX	CV	ZN	UD	IR	SJ	HW	GA	KQ	zqj	hlg	xky	ebt
St	27.	V	I	IV	20	06	18	KX	GJ	EP	AC	TB	HL	MW	QS	DV	OZ	bvo	sur	ccc	lqe
St	26.	IV	I	V	10	17	01	YV	GT	OQ	WN	FI	SK	LD	RP	MZ	BU	jhx	uuh	giw	ugw
St	25.	V	IV	III	13	04	17	QR	GB	HA	NM	VS	WD	YZ	OF	XK	PE	tba	pnc	ukd	nld
St	24.	III	II	IV	09	20	18	RS	NC	WK	GO	YQ	AX	EH	VJ	ZL	PF	nfi	mew	xbk	yes
St	23.	V	II	III	11	21	08	EY	DT	KF	MO	XP	HN	WG	ZL	IV	JA	lsd	nuo	vcr	vcx
St	22.	I	II	IV	01	25	02	PZ	SE	OJ	XF	HA	GB	VQ	UY	KW	LR	yji	rwy	rdk	nso
St	21.	IV	I	III	06	22	03	GH	JR	TQ	KF	NZ	IL	WM	BD	UO	EC	ema	mlv	jjy	iqh
St	20.	V	I	II	12	25	08	TF	RQ	XV	DZ	PY	NL	WI	SJ	ME	GB	xjl	pgs	ggh	znd
St	19.	IV	III	II	07	05	23	ZX	EU	AC	GD	KP	VO	QS	NW	HL	RM	vpj	zqe	jrs	cgm
St	18.	II	III	V	19	14	22	WG	OM	RL	DB	ST	AQ	PZ	XH	YN	IJ	oxd	lnb	icu	ytt
St	17.	IV	I	II	12	08	21	ME	HX	BP	WY	ZD	TR	FJ	AG	IL	KQ	tak	pjs	kdh	jvh
St	16.	I	II	III	07	11	15	WZ	AB	MO	TF	RX	SG	QU	VI	YN	EL	pzg	evw	wyt	iye
St	15.	III	II	V	06	16	02	GT	YC	EJ	UA	RX	PN	IS	WB	MH	ZV	bhe	xzm	yzk	evp
St	14.	II	I	V	23	05	24	AZ	CJ	WF	UY	SO	QV	MI	NH	DP	GX	fdx	tyj	bmq	typ
St	13.	IV	II	V	03	25	10	CX	KN	JR	DQ	IU	TL	HZ	MF	EP	WB	zfo	bjr	zwx	gvn
St	12.	I	III	II	26	01	18	QB	YE	WN	AI	GJ	TO	HR	FK	PS	CM	upc	anf	tkr	pwz
St	11.	V	I	III	17	13	04	SV	GO	PA	ZR	FN	HI	YM	WT	DE	BJ	vdh	ego	wmy	uti
St	10.	I	V	IV	26	07	16	SW	AQ	NF	PO	VY	UX	MK	CL	HT	ZJ	rpl	anw	vpr	mhn
St	9.	I	III	IV	17	10	18	EH	IR	GK	NZ	SP	UA	LD	CQ	JM	YV	knq	ysq	rhj	tlj
St	8.	V	II	I	23	11	25	QY	OG	ST	HA	CB	WD	KL	JN	VX	IU	lro	avw	axh	gws
St	7.	II	III	I	06	12	03	RC	FS	TH	JE	VK	PL	CU	QA	OD	NM	atv	mbb	myo	jmz

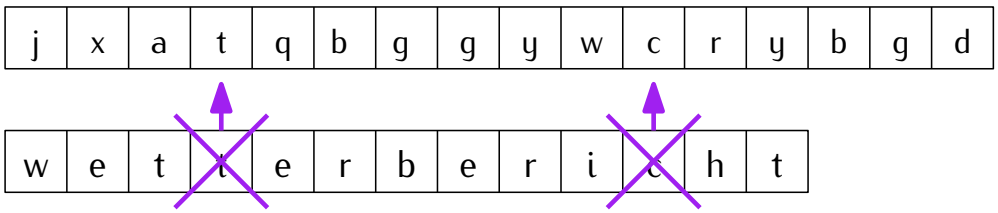
Enigma - Breaking the machine

- To brute-force Enigma is unpractical: > 150 millions millions combinations
- A letter is encrypted into a different letter every time ....
- ... but never to itself !! **Main flaw**
- Try to guess a word or phrase in a message (and Germans military did use recurring messages, like weather reports)
- ...

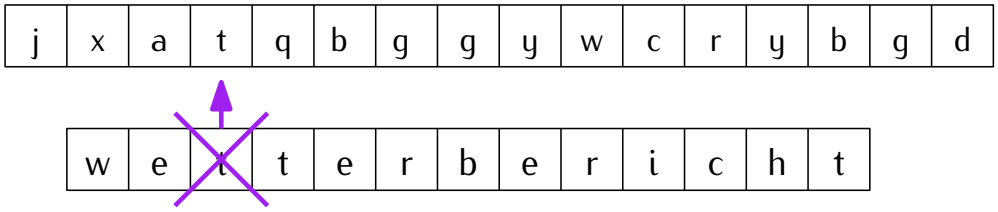
Enigma - breaking the machine

j	x	a	t	q	b	g	g	y	w	c	r	y	b	g	d

Enigma - breaking the machine



Enigma - breaking the machine



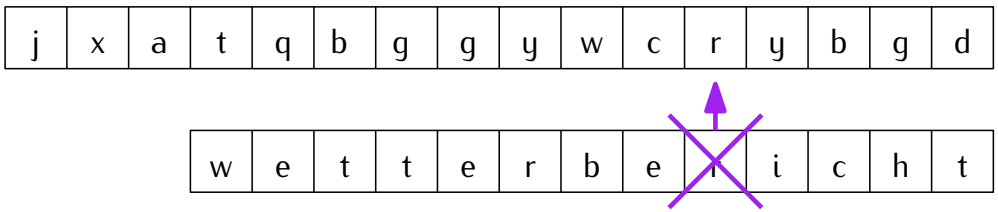
Enigma - breaking the machine

j	x	a	t	q	b	g	g	y	w	c	r	y	b	g	d

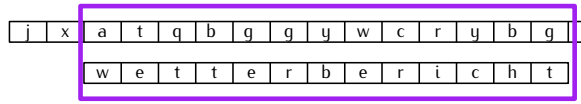
w	e	t	t	e	r	b	e	r	i	c	h	t

OK !

Enigma - breaking the machine



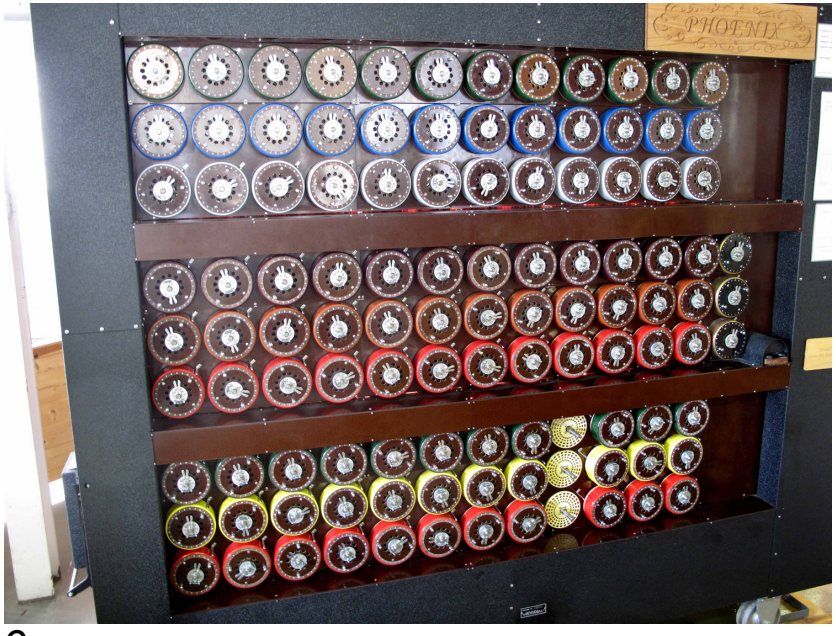
Possible Solutions

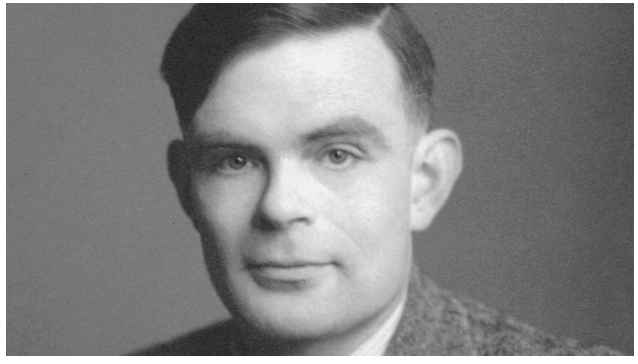


Enigma - Breaking the machine

- Adding a couple more properties, evict impossible configurations.
- Scan through the remaining combinations using **the Bombe** : electro-mechanical machine able to “play” 36 Enigma equivalent “in parallel”.
- In the end .... guess the key (wheel starting positions + plug-board) in less than 20minutes per day.

Enigma





[6]watch

`https://en.wikipedia.org/wiki/The_Imitation_Game`

6*

More Recent history

Symmetric (private-key) cryptography:

- **1975** IBM proposes the Data-Encryption Standard
- **1977** DES Adopted as a FIPS standard
- **1994** Differential-linear cryptanalysis of DES is proposed
- **1996** call for DES replacement by NIST
- **1998** Brute-force attack on DES demonstrated feasible
- **2001** AES announced as replacement for DES
- **2023** *At present, there is no known practical attack that would allow someone without knowledge of the key to read data encrypted by AES when correctly implemented.* [a]

Asymmetric (public-key) cryptography:

- **1976** Diffie-Hellman key exchange protocol proposed
- **1977** RSA (Rivest-Shavir-Adleman)
- **1985** El-Gamal encryption
- **1985-...** Elliptic-Curve Cryptography

Course Plan

Lectures

- Symmetric cryptography, Asymmetric cryptography, Key sharing, compromises
- Security protocols: Public-Key Infrastructures, TLS, SSH, HTTPS, Kerberos, VPNs,

Lab / Paper Sessions

- Ethical considerations
- Applying asymmetric encryption principles
- Password storage
- Certification and Public-Key Infrastructures
- Cryptographic Protocols
- Reading survey project

All details on `https://lmorel-insa.github.io/csc/`

References

- On exploiting buffer overflow: https://youtu.be/1S0aBV-Waeo
- Turing's Enigma Problem Part 1: https://youtu.be/d2NWPG2gB_A
- Turing's Enigma Problem Part 2: https://youtu.be/kj_7Jc1mS9k
- Some details on the working of the Enigma (with a real machine presented): https://youtu.be/G2_Q9FoD-oQ
- How easy is to crack Enigma today: https://youtu.be/RzWB5jL5RX0