LASSONDE
SCHOOL OF ENGINEERING | YORK U

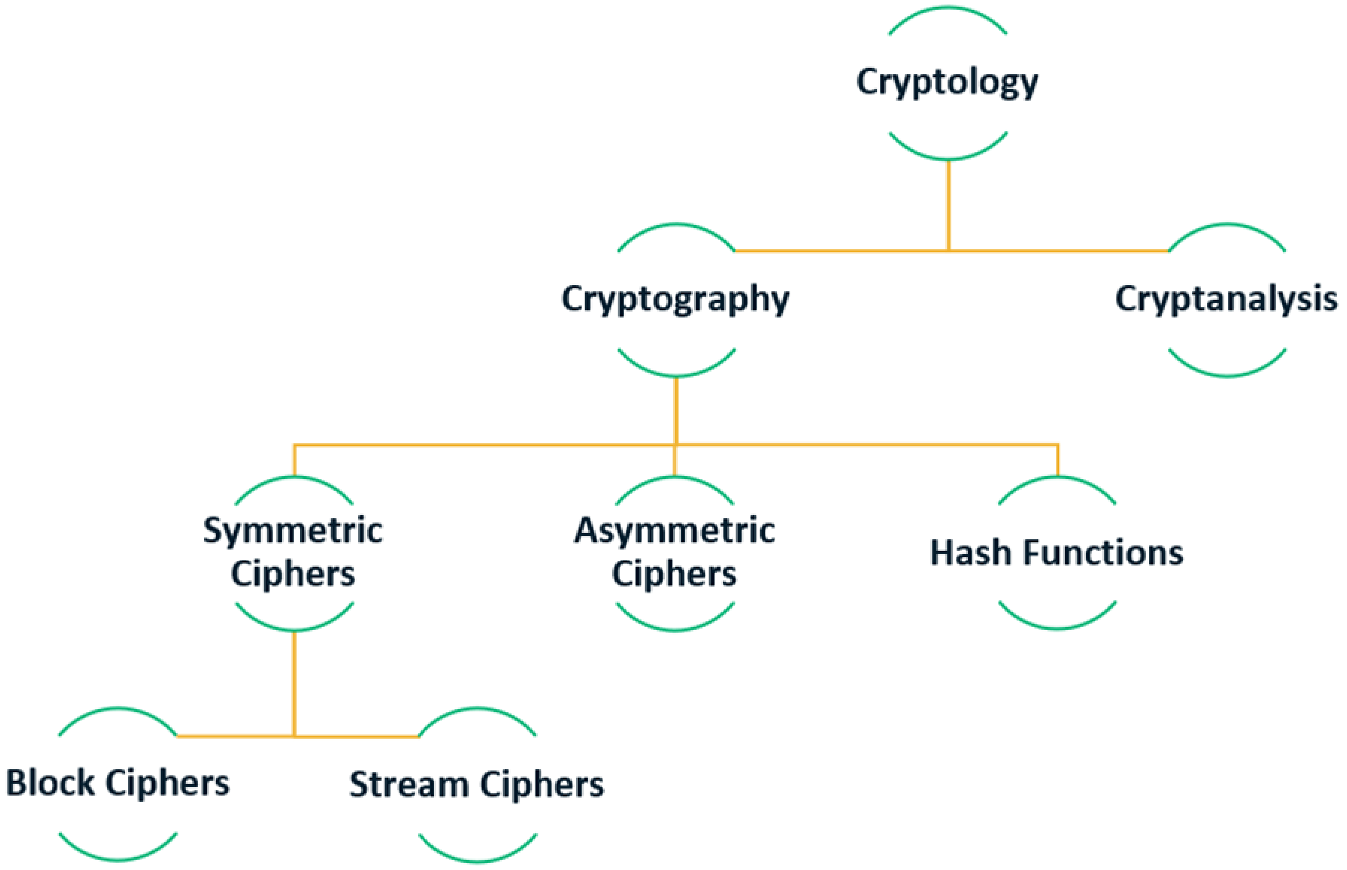
Cryptology

Dr. Ruba Al Omari

EECS 3401 – Applied Cryptography

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security



Source: https://sectigostore.com/blog/cryptology-vs-cryptography-whats-the-difference/

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security

Etymology

- Kryptos (Greek) = hidden
  - Cryptology.
  - Ology = Study.
- Sifr (Arabic) = zero
  - cipher, encipher, decipher.
- Steganos (Greek) = covered
  - Steganography.
  - Graphy – Written or Process.

https://en.wikipedia.org/wiki/Kryptos#

Kryptos: By Jim Sanborn - Jim Sanborn, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=8253447

Definitions 1/2

- **Plaintext**: An original message.
- **Ciphertext**: The coded message.
- **Enciphering/Encryption**: The process of converting from plaintext to ciphertext.
- **Deciphering/Decryption**: Restoring the plaintext from the ciphertext.

Definitions 2/2

- **Cryptography**: The process of making and using codes to secure information.
- **Cryptanalysis**: The process of obtaining the plaintext message from a ciphertext message without knowing the keys used to perform the encryption.
- **Cryptology**: The field of science that encompasses cryptography and cryptanalysis.

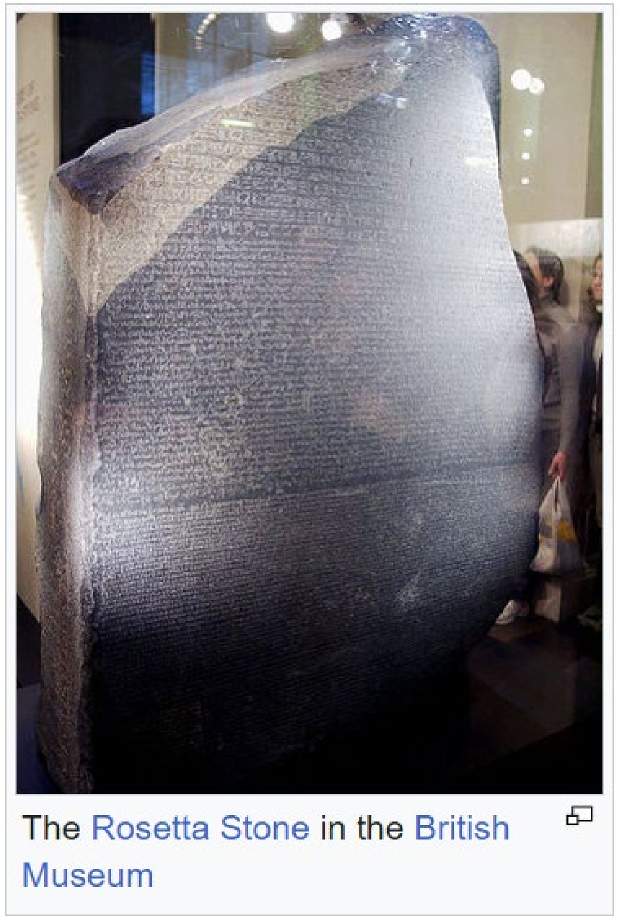
CRYPTOLOGY = CRYPTOGRAPHY + CRYPTANALYSIS

History of Cryptography

- 1900 BC: Egypt - first documented use of written cryptography.

Egyptian hieroglyphs

Hieroglyphs from the tomb of Seti I (KV17), 13th century BC



The Rosetta Stone in the British Museum

Hieroglyphs typical of the Graeco-Roman period

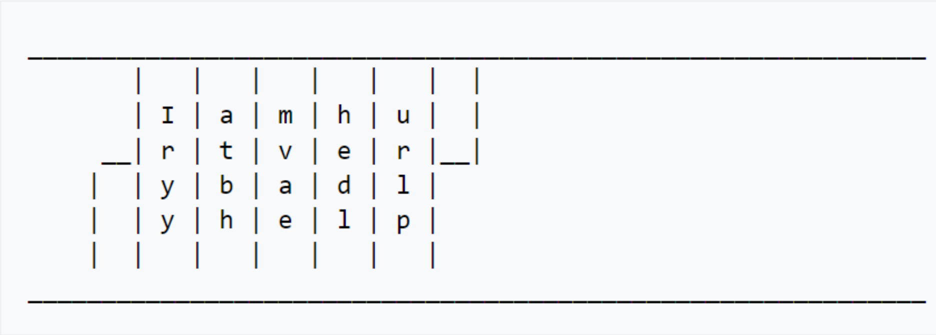
Ibn Wahshiyya's attempt at a translation of a hieroglyphic text

https://en.wikipedia.org/wiki/Egyptian_hieroglyphs

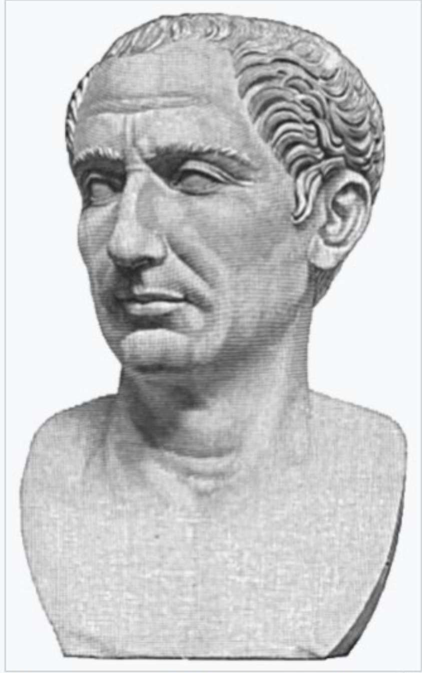
History of Cryptography

- 487-50 BC: Spartans & Julius Caesar

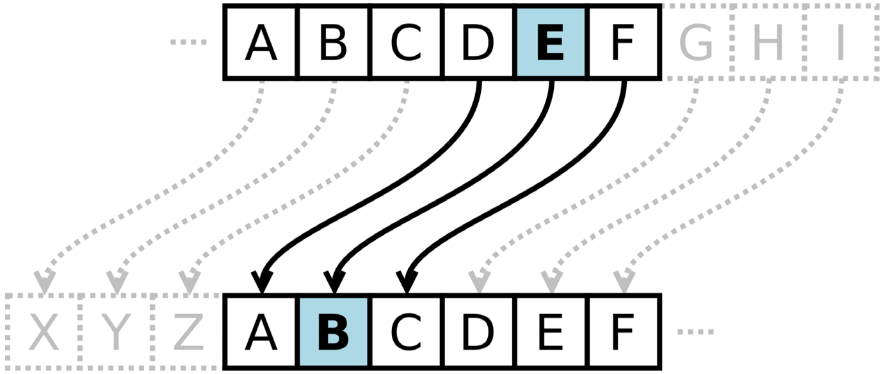
A scytale



```
_____________________________________________________________
       |   |   |   |   |   |  |
       | I | a | m | h | u |  |
     __| r | t | v | e | r |__|
    |  | y | b | a | d | l |
    |  | y | h | e | l | p |
    |  |   |   |   |   |   |
_____________________________________________________________
```



The Caesar cipher is named for Julius Caesar, who used an alphabet where decrypting would shift three letters to the left.

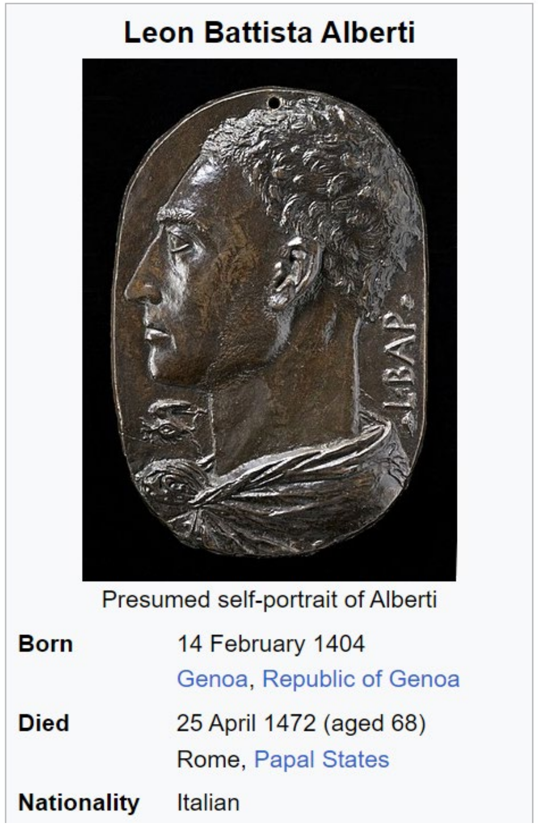


https://en.wikipedia.org/wiki/Scytale

https://en.wikipedia.org/wiki/Caesar_cipher

History of Cryptography

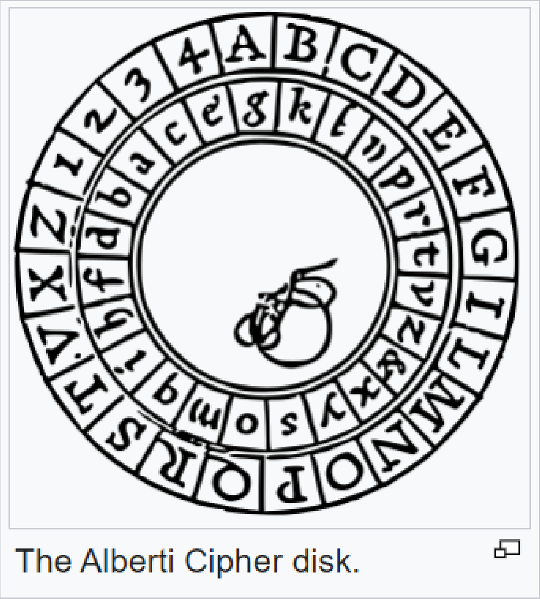
- 1466: Alberti Polyalphabetic Cipher



Leon Battista Alberti

Presumed self-portrait of Alberti

Born	14 February 1404 Genoa, Republic of Genoa
Died	25 April 1472 (aged 68) Rome, Papal States
Nationality	Italian



The Alberti Cipher disk.

https://en.wikipedia.org/wiki/Leon_Battista_Alberti
https://en.wikipedia.org/wiki/Alberti_cipher

History of Cryptography

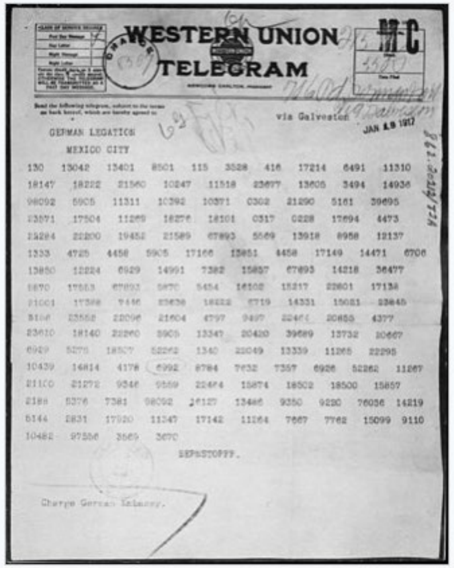
- World War I: **Substitution** and **Transposition** Ciphers

Hello	**Hello**
asddf	**Holel**

History of Cryptography

Vernam, an AT&T employee, invented a polyalphabetic cipher machine that used a nonrepeating random key.

- World War I: Zimmerman Telegram & OTP (One-Time Pad) – 1914-18



WESTERN UNION TELEGRAM

via Galveston

JAN 19 1917

GERMAN LEGATION

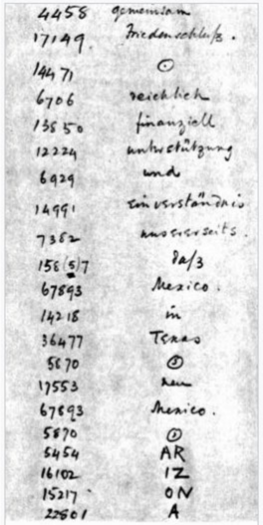
MEXICO CITY

```
130    13042  13401  8501   115    3528   416    17214  6491   11310
18147  18222  21560  10247  11518  23677  13605  3494   14936
98092  5905   11311  10392  10371  0302   21290  5161   39695
23571  17504  11269  18276  18101  0317   0228   17694  4473
22284  22200  19452  21589  67893  5569   13918  8958   12137
1333   4725   4458   5905   17166  13851  4458   17149  14471  6706
13850  12224  6929   14991  7382   15857  67893  14218  36477
5870   17553  67893  5870   5454   16102  15217  22801  17138
21001  17388  7446   23638  18222  6719   14331  15021  23845
3156   23552  22096  21604  4797   9497   22464  20855  4377
23610  18140  22260  5905   13347  20420  39689  13732  20667
6929   5275   18507  52262  1340   22049  13339  11265  22295
10439  14814  4178   6992   8784   7632   7357   6926   52262  11267
21100  21272  9346   9559   22464  15874  18502  18500  15857
2188   5376   7381   98092  16127  13486  9350   9220   76036  14219
5144   2831   17920  11347  17142  11264  7667   7762   15099  9110
10482  97556  3569   3670
```

BERNSTORFF.

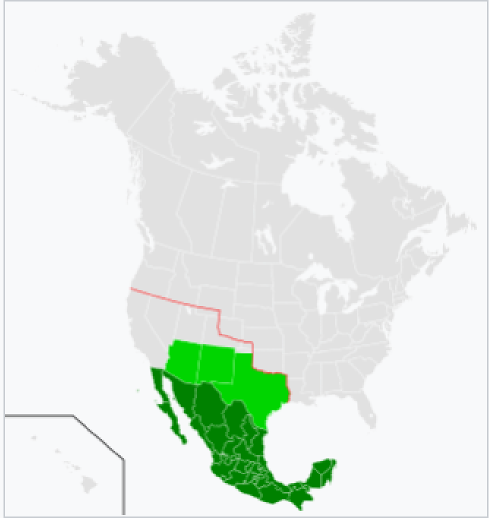
Charge German Embassy.

The Zimmermann telegram as it was sent from Washington, DC, to Ambassador Heinrich von Eckardt, the German ambassador to Mexico.

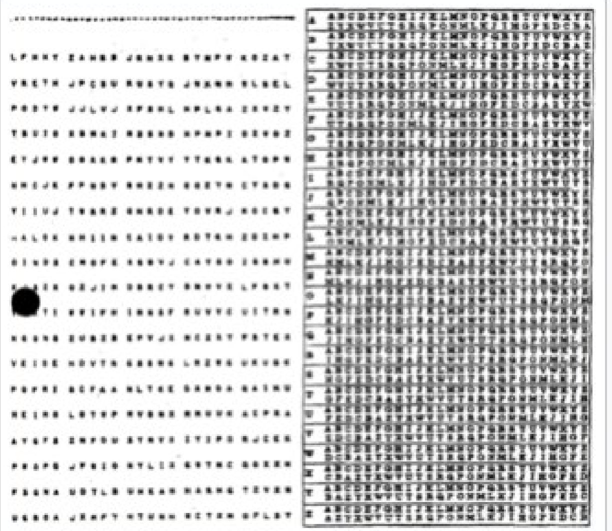


4458	gemeinsam
17149.	Friedensschluß.
14471	⊙
6706	reichlich
13850	finanziell
12224	unterstützung
6929	und
14991	Einverständnis
7382	unsererseits.
158(5)7	daß
67893	Mexico.
14218	in
36477	Texas
5870	⊙
17553	New
67893	Mexico.
5870	⊙
5454	AR
16102	IZ
15217	ON
22801	A

A portion of the telegram as decrypted by British Naval Intelligence codebreakers. Since the word *Arizona* was not in the German codebook, it had to be split into phonetic syllables.



Mexico in 1916 (in dark green); territory promised to Mexico in the Zimmermann telegram (in light green); and the pre-1836 original Mexican territory (red line)



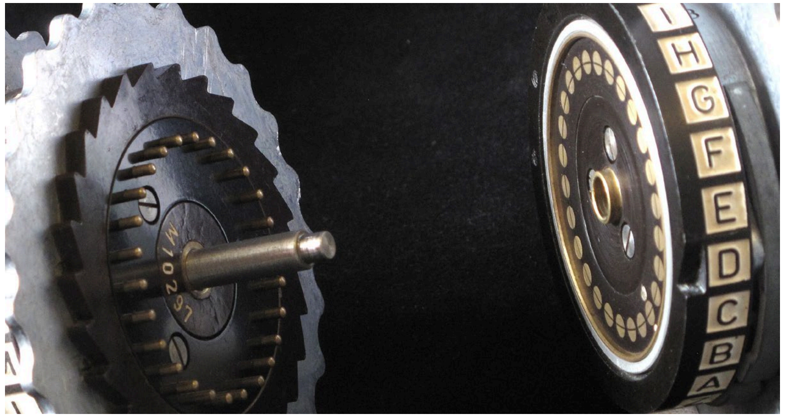
A format of one-time pad used by the U.S. National Security Agency, code named DIANA. The table on the right is an aid for converting between plaintext and ciphertext using the characters at left as the key.

https://en.wikipedia.org/wiki/Zimmermann_telegram
https://en.wikipedia.org/wiki/One-time_pad

History of Cryptography

- World War II: Enigma Machine (1939-45)
  - Typing illuminates an alternate letter
    - This is your encoded letter.
  - Which letter lights up depends on the arrangement of the rotors.
    - Each rotor has a possible 26 settings.
  - Adding extra security is a plugboard
    - This swipes letters of any letter of the operator’s choosing.

Military Model Enigma I, in use from 1930

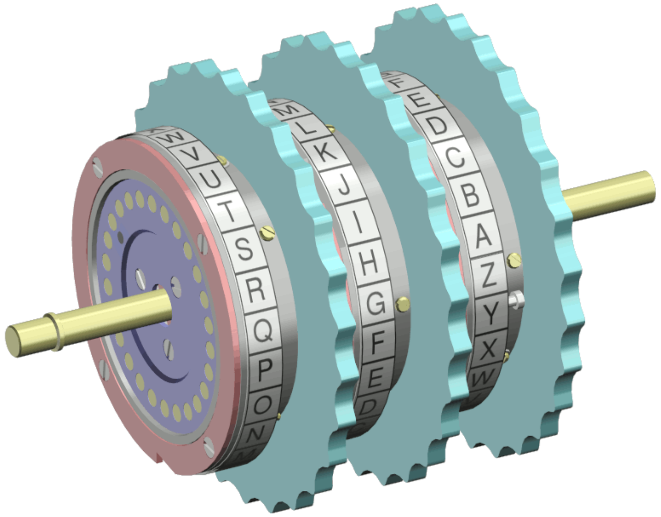


Two Enigma rotors showing electrical contacts, stepping ratchet (on the left) and notch (on the right-hand rotor opposite **D**).

https://en.wikipedia.org/wiki/Enigma_machine

History of Cryptography

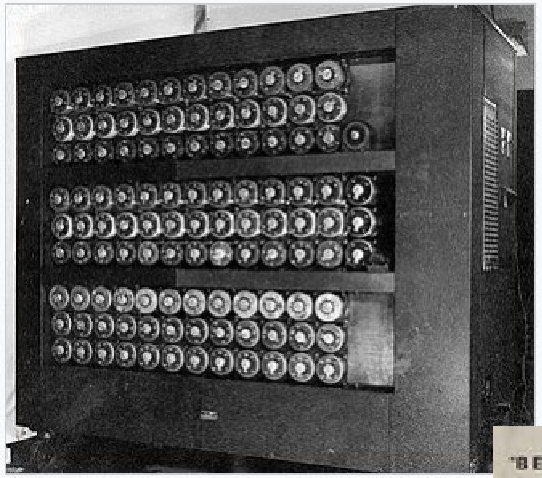
- World War II: Enigma Machine
  - To read a message you need to enter the right initial rotor and plugboard settings on your Enigma machine.
  - Without these settings, decoding is almost impossible with over **150 million, million, million** possible combinations.



https://en.wikipedia.org/wiki/Enigma_machine

History of Cryptography

- World War II: Enigma Machine
  - The Allies broke the Enigma machine by exploiting operator errors, captured codebooks, and developing advanced decryption techniques, notably using the **Bombe** machine designed by **Alan Turing** and others at Bletchley Park.
  - This allowed them to reverse-engineer Enigma settings and decode messages.

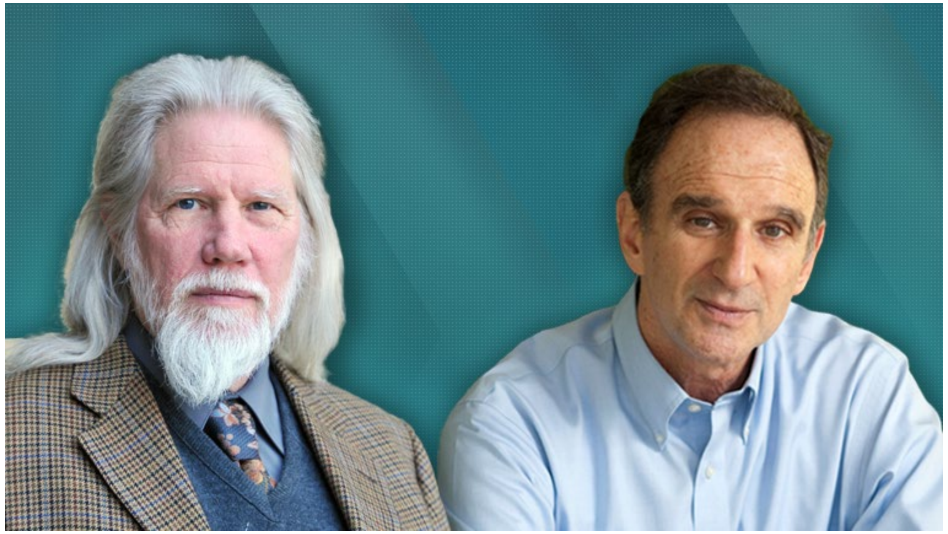


A wartime picture of a Bletchley Park Bombe

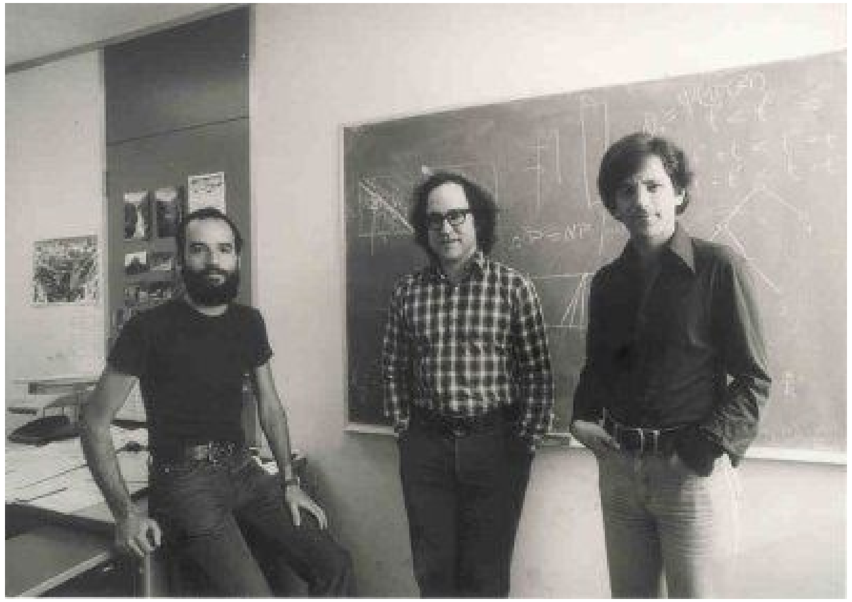
https://en.wikipedia.org/wiki/Enigma_machine

History of Cryptography

- 1976- : Public Key Cryptography

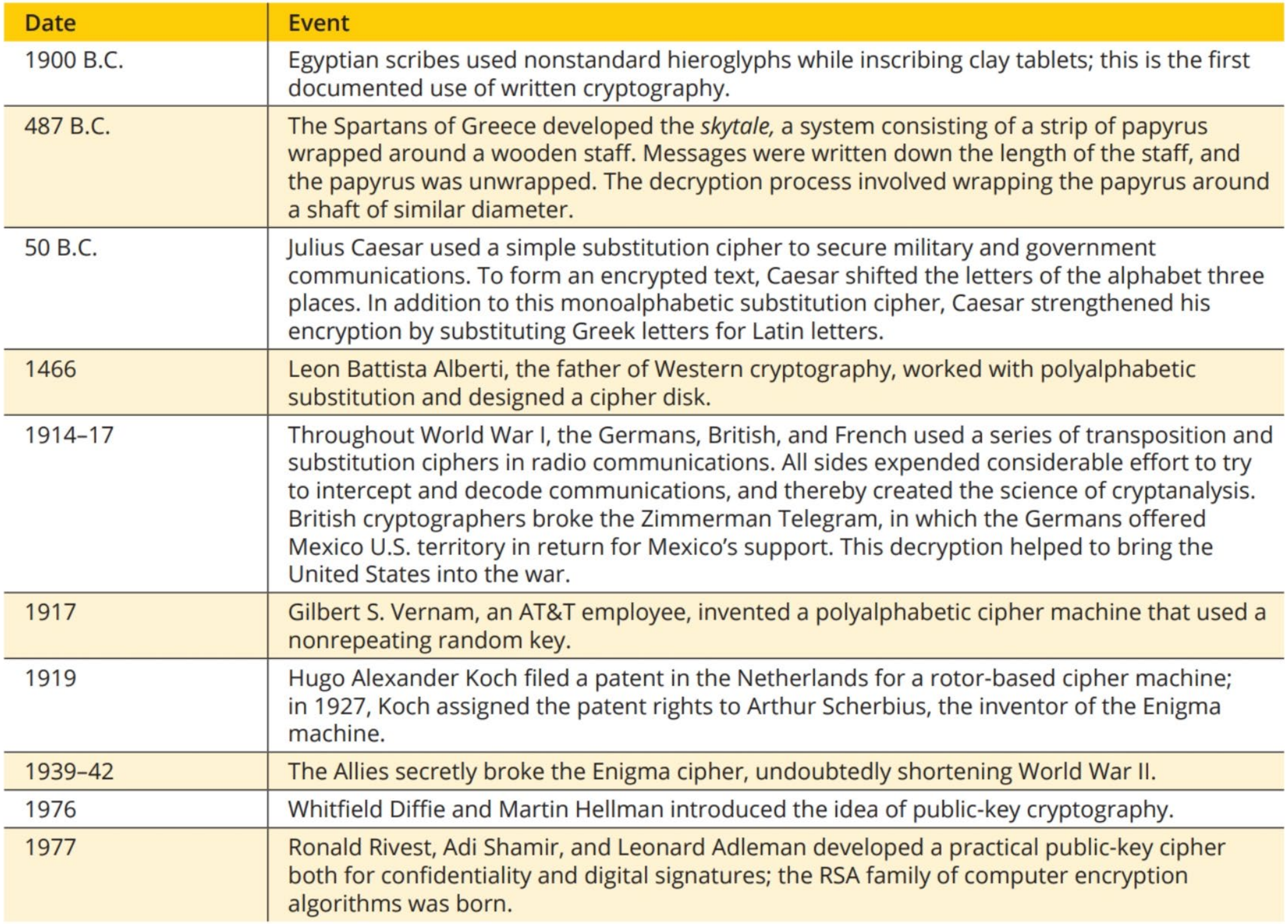


Whitfield Diffie and Martin Hellman
https://www.acm.org/articles/bulletins/2016/march/turing-2015-diffie-hellman



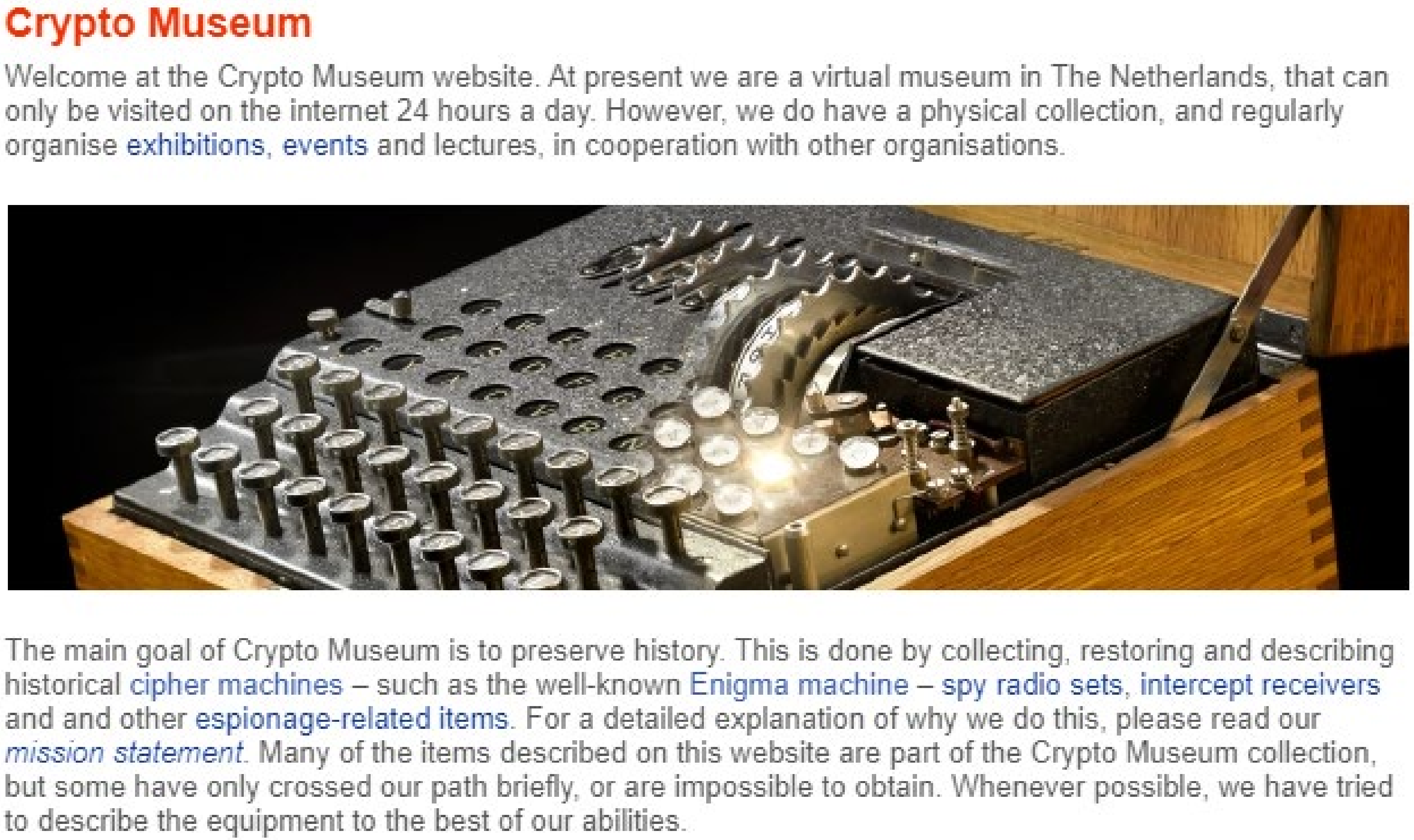
December 197X, left to right: Adi Shamir, Ron Rivest, and Len Adleman Image https://people.csail.mit.edu/rivest/photos/photos.html

History of Cryptography



Date	Event
1900 B.C.	Egyptian scribes used nonstandard hieroglyphs while inscribing clay tablets; this is the first documented use of written cryptography.
487 B.C.	The Spartans of Greece developed the *skytale,* a system consisting of a strip of papyrus wrapped around a wooden staff. Messages were written down the length of the staff, and the papyrus was unwrapped. The decryption process involved wrapping the papyrus around a shaft of similar diameter.
50 B.C.	Julius Caesar used a simple substitution cipher to secure military and government communications. To form an encrypted text, Caesar shifted the letters of the alphabet three places. In addition to this monoalphabetic substitution cipher, Caesar strengthened his encryption by substituting Greek letters for Latin letters.
1466	Leon Battista Alberti, the father of Western cryptography, worked with polyalphabetic substitution and designed a cipher disk.
1914–17	Throughout World War I, the Germans, British, and French used a series of transposition and substitution ciphers in radio communications. All sides expended considerable effort to try to intercept and decode communications, and thereby created the science of cryptanalysis. British cryptographers broke the Zimmerman Telegram, in which the Germans offered Mexico U.S. territory in return for Mexico's support. This decryption helped to bring the United States into the war.
1917	Gilbert S. Vernam, an AT&T employee, invented a polyalphabetic cipher machine that used a nonrepeating random key.
1919	Hugo Alexander Koch filed a patent in the Netherlands for a rotor-based cipher machine; in 1927, Koch assigned the patent rights to Arthur Scherbius, the inventor of the Enigma machine.
1939–42	The Allies secretly broke the Enigma cipher, undoubtedly shortening World War II.
1976	Whitfield Diffie and Martin Hellman introduced the idea of public-key cryptography.
1977	Ronald Rivest, Adi Shamir, and Leonard Adleman developed a practical public-key cipher both for confidentiality and digital signatures; the RSA family of computer encryption algorithms was born.

Table Source: Principles of Information Security, by Whitman, 7th Edition



Crypto Museum

Welcome at the Crypto Museum website. At present we are a virtual museum in The Netherlands, that can only be visited on the internet 24 hours a day. However, we do have a physical collection, and regularly organise exhibitions, events and lectures, in cooperation with other organisations.

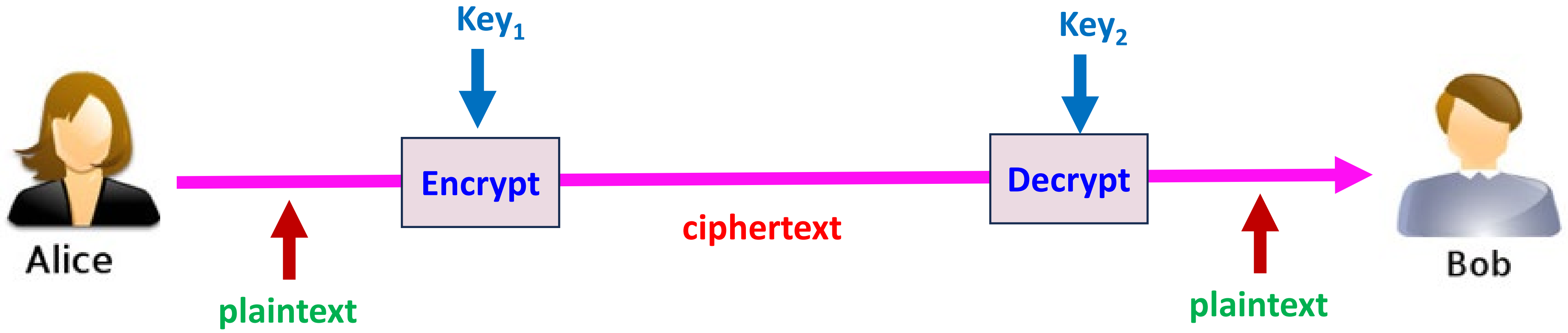
The main goal of Crypto Museum is to preserve history. This is done by collecting, restoring and describing historical cipher machines – such as the well-known Enigma machine – spy radio sets, intercept receivers and and other espionage-related items. For a detailed explanation of why we do this, please read our *mission statement*. Many of the items described on this website are part of the Crypto Museum collection, but some have only crossed our path briefly, or are impossible to obtain. Whenever possible, we have tried to describe the equipment to the best of our abilities.

Source: https://www.cryptomuseum.com/index.htm

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security

The Model



- **The P and C sets**

Strings $\sum *$ over some alphabet Σ, e.g., $\Sigma = \{0, 1\}$ or $\{A - Z\}$

- **The E and D transformations**

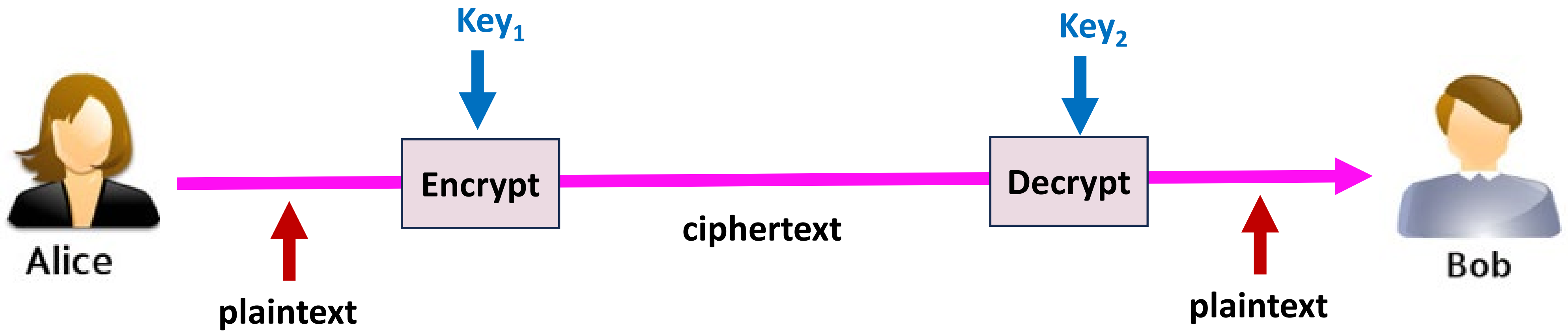
Keyed functions transofming $P \leftrightarrow C$.

- **The Cryptosystem**

The tuple $(P, C, K1, K2, E, D)$

Alice and Bob Image Source: Didia, CC BY-SA 4.0, via Wikimedia Commons

Secret vs. Public vs. Private



- **Symmetric** (aka Secret-Key) Cryptography.

  Key$_1$=Key$_2$ and known only to the legitimate parties.

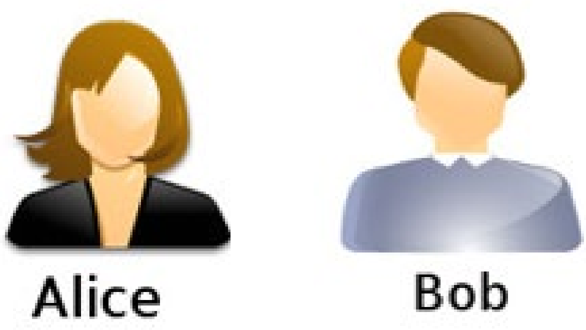
- **Asymmetric** (aka Public-Key) Cryptography.

  Key$_1$ is Bob's public, Key$_2$ is his private key.

- The **Kerckhoff** Principle.
  - Reject Security by Obscurity: E and D are public but not the key.

Alice and Bob

- Do you know how old are Alice and Bob?



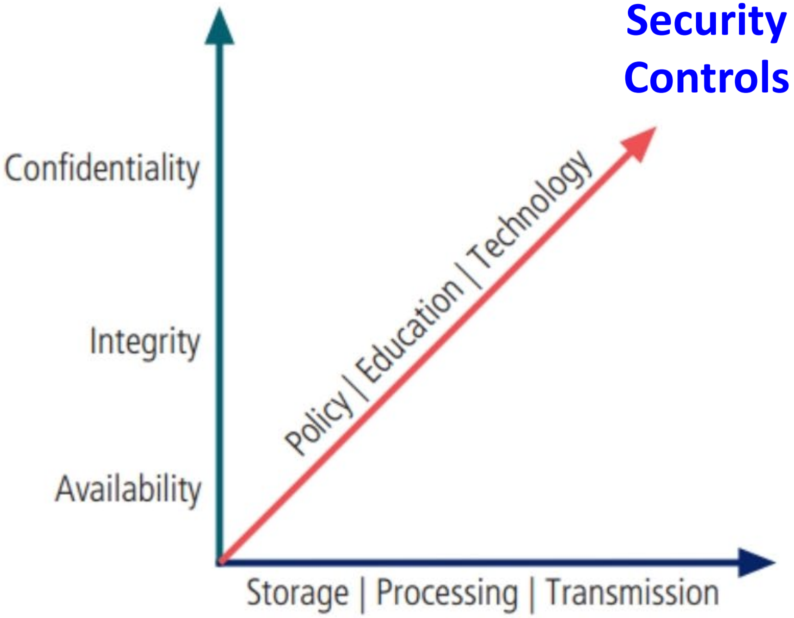
Alice and Bob Image Source: Didia, CC BY-SA 4.0, via Wikimedia Commons

The CIA Triad

- **C**onfidentiality, **I**ntegrity, and **A**vailability.
- Which one do you think is the most important?

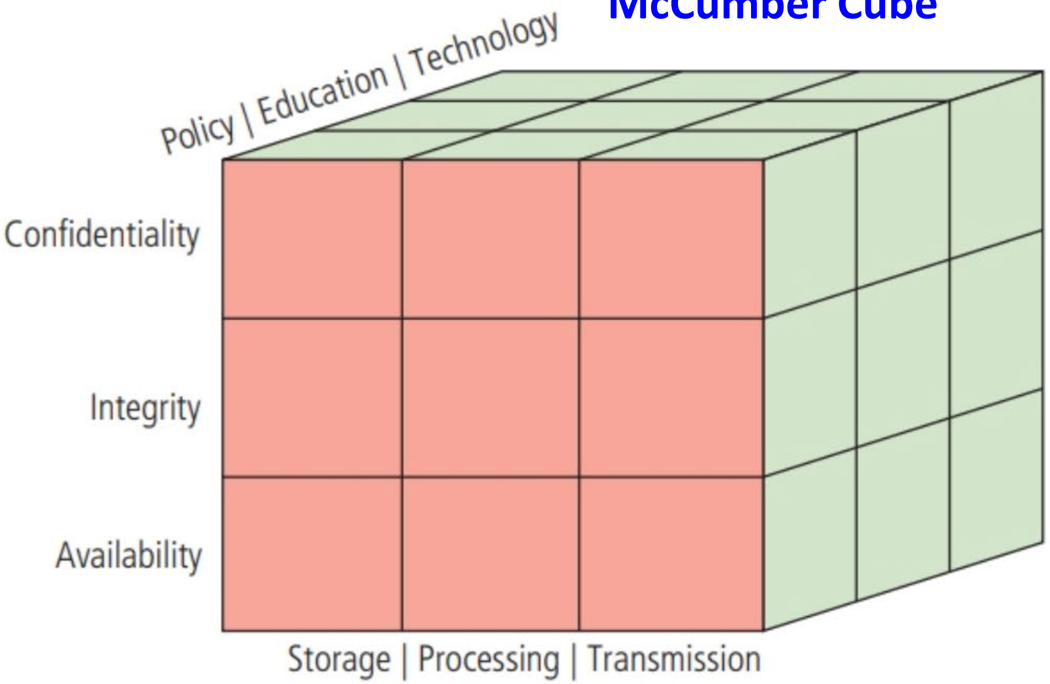
Information Characteristics

Security Controls


Confidentiality
Integrity
Availability
Policy | Education | Technology
Storage | Processing | Transmission

Information Location

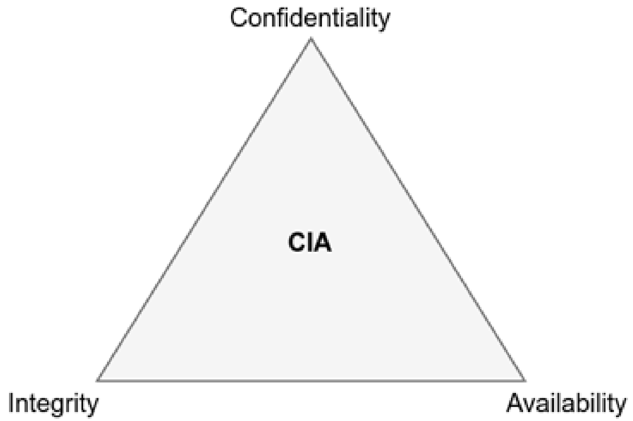
McCumber Cube

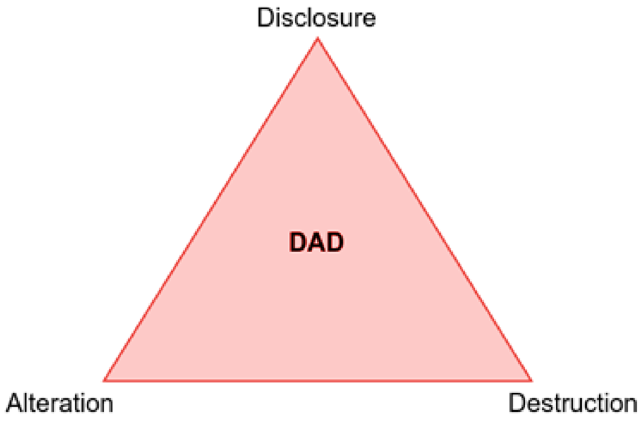

Policy | Education | Technology
Confidentiality
Integrity
Availability
Storage | Processing | Transmission

Source: Principles of Information Security, by Whitman, 7th Edition, modified by Alomari

The CIA Triad

- **Confidentiality**
  - Based on need-to-know
- **Integrity**
  - Of Content .... integrity
  - Of Parties .... authentication
  - Of Time of sending ...... freshness
  - Of Act of sending .... non-repudiation
- **Availability**
  - Anytime from anywhere





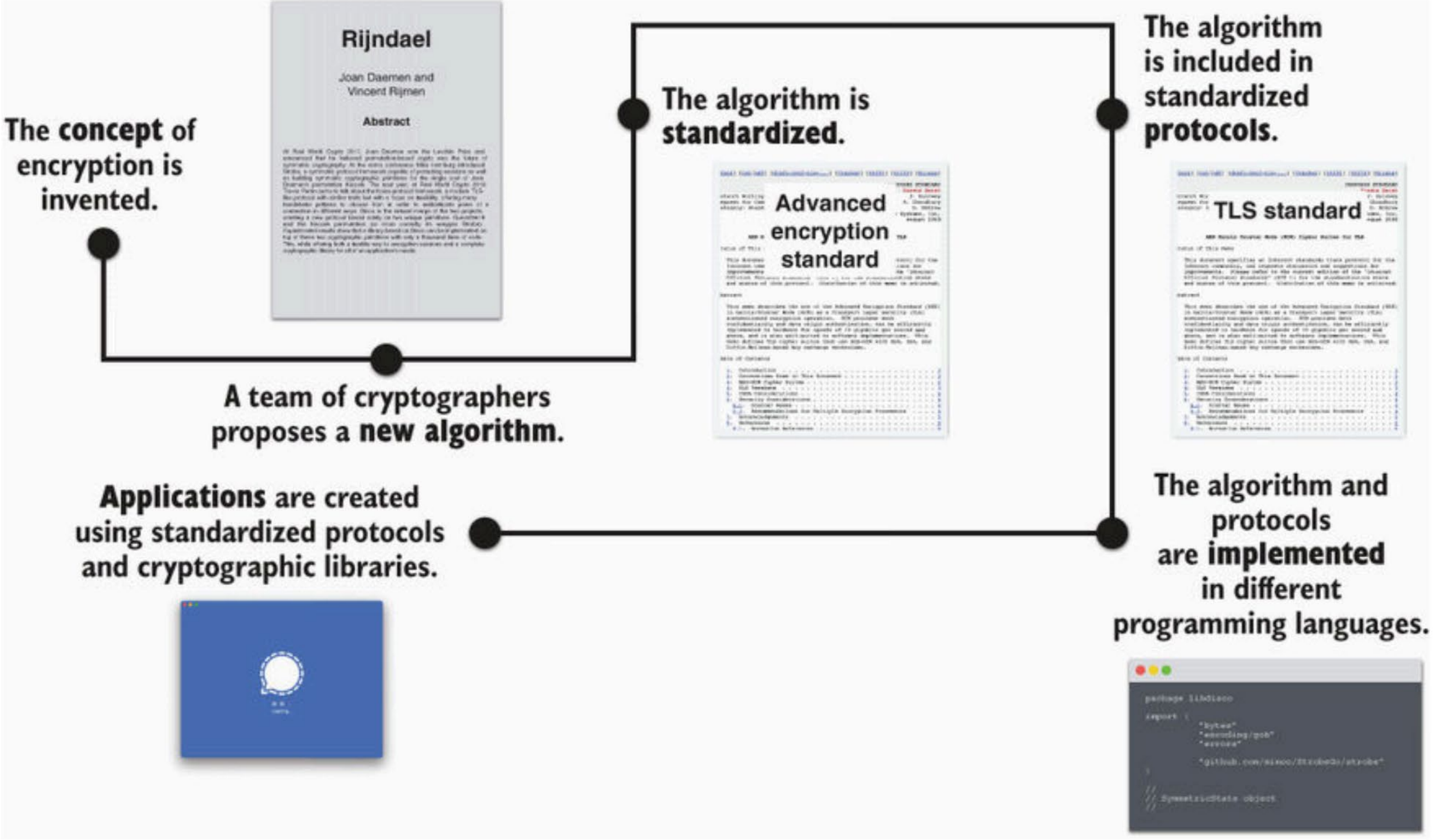
The Primitives of Cryptography

1. **Symmetric Ciphers** *(efficient, AES)* - One pre-shared **secret** key per pair of actors
2. **Asymmetric Ciphers** *(inefficient, RSA)* - One **Public/Private** key pair per actor
3. **Cryptographic Hash Functions** *(efficient, SHA)* - Maps a message of arbitrary size to a fixed-size **digest**. Any change in the message will almost certainly change the digest!

- *Using a mix of these three building blocks plus a random source, cryptography promises to meet — any — security requirement!*

Source: https://www.eecs.yorku.ca/~roumani/

Life Cycle for Cryptographic Algorithm



The **concept** of encryption is invented.

A team of cryptographers proposes a **new algorithm.**

The algorithm is **standardized.**

The algorithm is included in standardized **protocols.**

The algorithm and protocols are **implemented** in different programming languages.

```
package libDisco

import (
        "bytes"
        "encoding/gob"
        "errors"

        "github.com/mimoo/StrobeGo/strobe"
)

//
// SymmetricState object
//
```

Applications are created using standardized protocols and cryptographic libraries.

Source: Real-World Cryptography, 2021, by David Wong

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security

Attacks

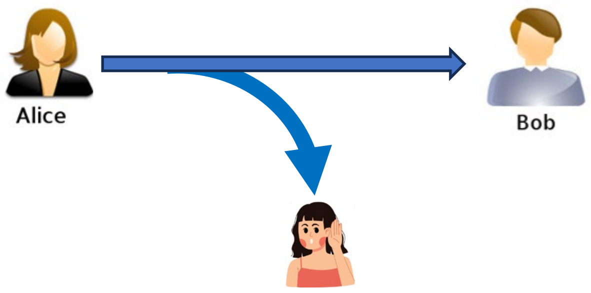
- Confidentiality

  Authentication & Authorization

  Based on need-to-know

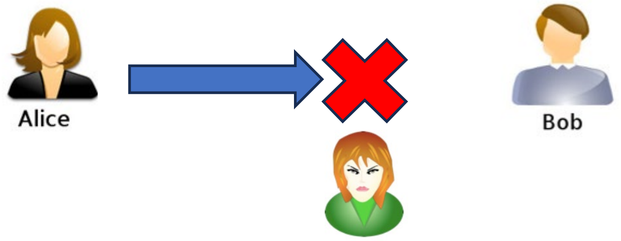
SPY / SNIFF

DISCLOSURE



- Availability
  Anytime from anywhere

DoS/DDoS



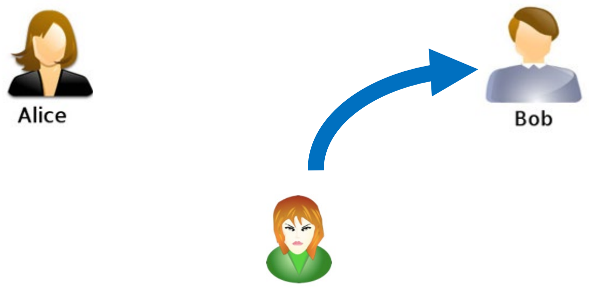
Attacks

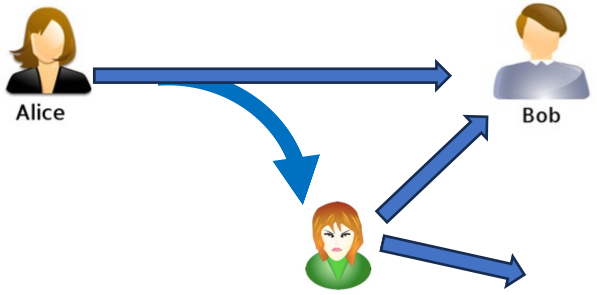
- Integrity
  Of Sender and Sending
  Of Time of sending
  Of Content

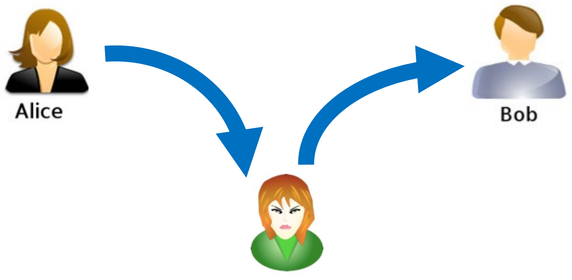
SPOOF / REPUDIATE

REPLAY

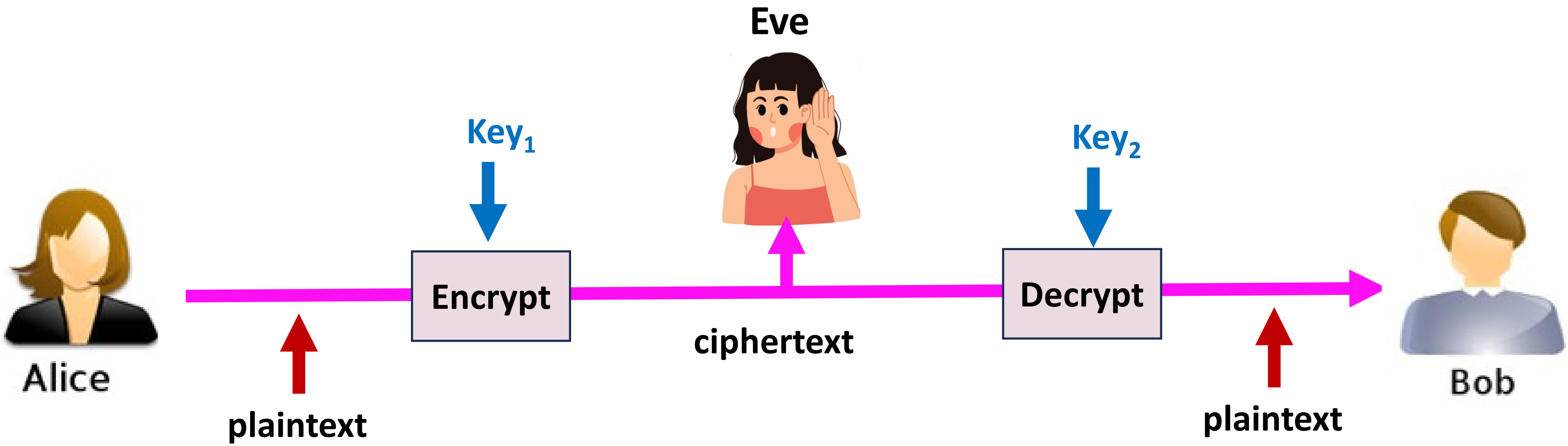
FABRICATE







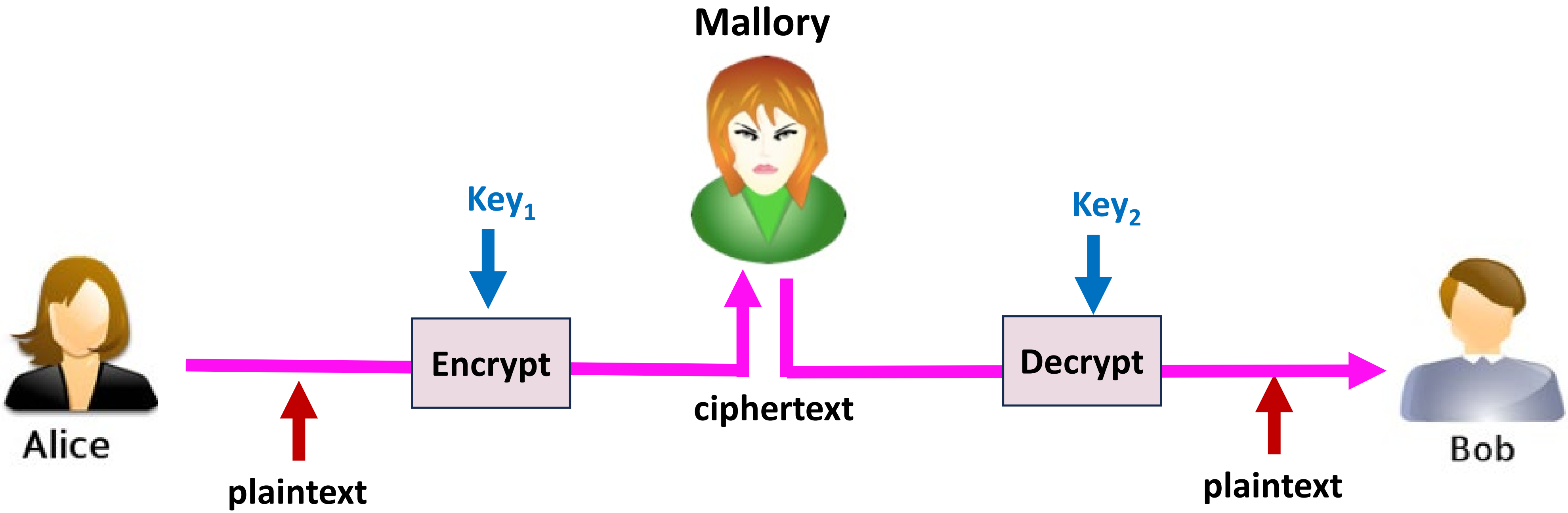
Meet Eve (Eavesdropper)



- Eve is a passive intruder: she doesn’t affect the message but can gather stats/observe patterns, e.g., metadata; create a codebook via KPA, etc. She therefore violates confidentiality.

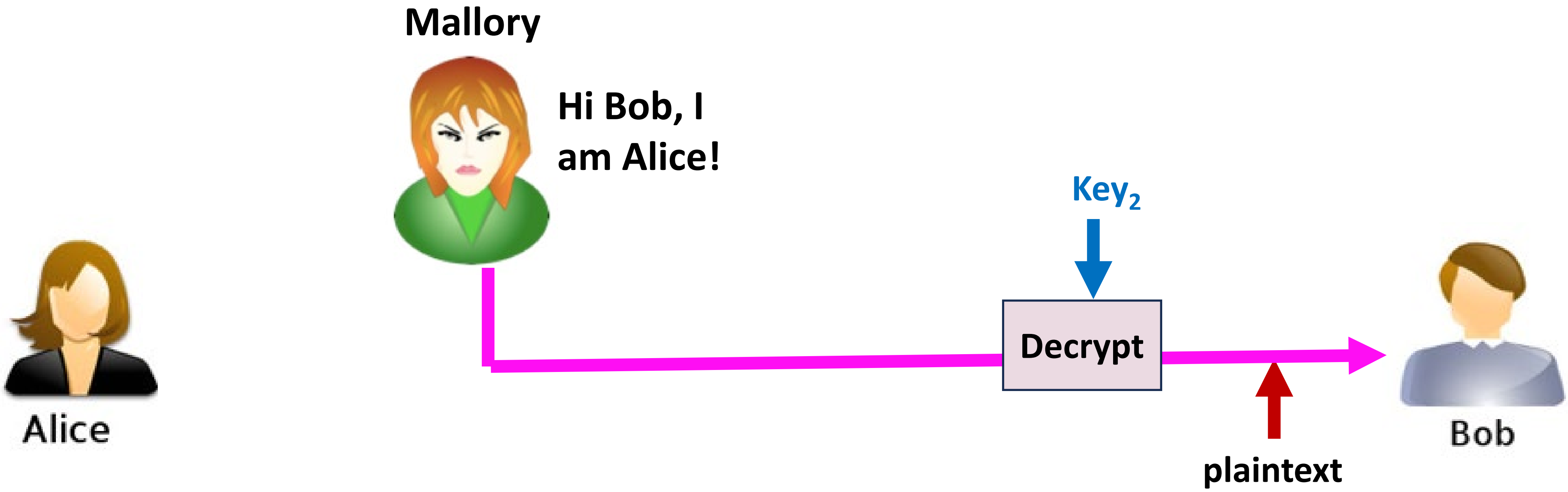
Eve Image: Vecteezy.com

Meet Mallory* (Malicious)



* AKA Trudy (Intruder). Mal is an active attacker: she can block or change the message. She can violate all three CIA goals.

Mallory Masquerading as Alice



- Mal is spoofing Bob by masquerading as Alice (perhaps via replay). This causes loss of integrity of message origin.

Attacks

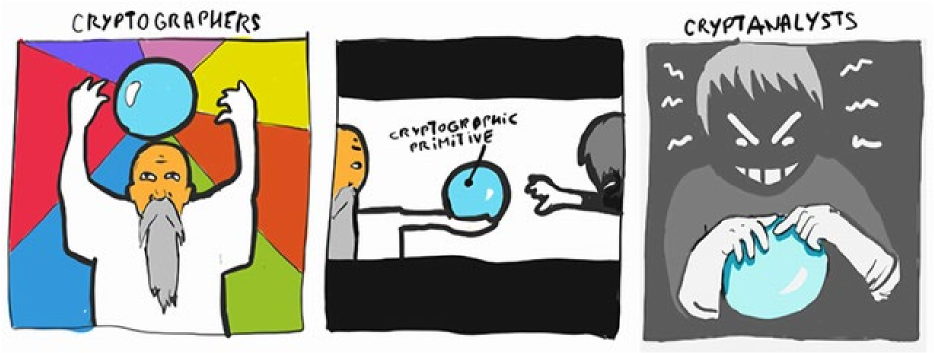
Exhaustive

Scan the Entire Key Space

Does not depend on the specific algorithm, only **depends on key length.**

Try each value until a collision occurs.

Cryptanalytic



Scan a Subset of the Key Space

Based on **weaknesses** in a particular cryptographic algorithm.

Seek to exploit some property of the algorithm to perform some attack other than an exhaustive search.

Image Source: Real-World Cryptography, 2021, by David Wong

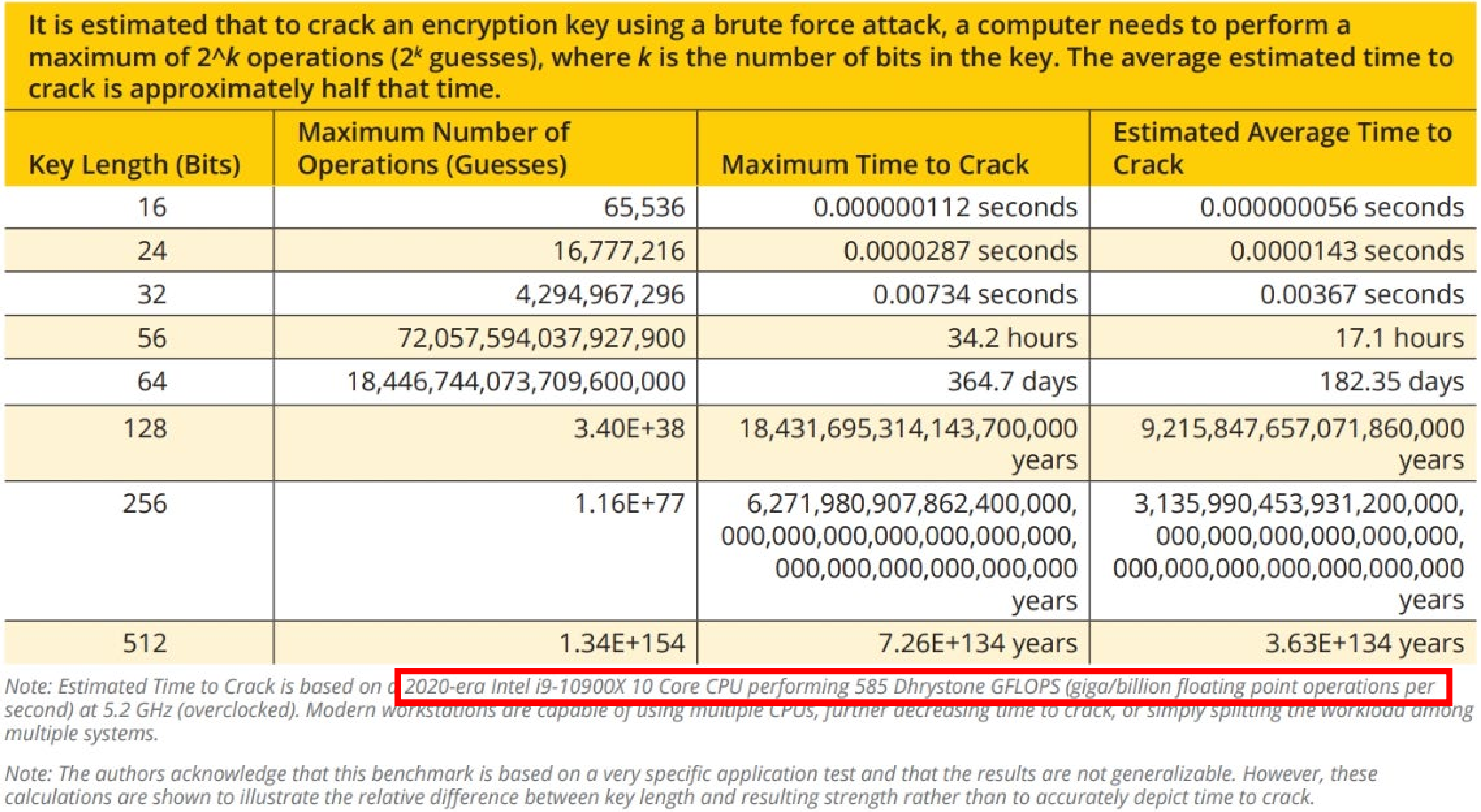
Exhaustive (Scan The Entire Key Space)

- Aka *Brute-Force* or *Trial-and-Error*
- Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained
- Lends itself naturally to parallel computations

Key size /b	Number of keys	1 Core	Cluster
32	$2^{32} = 4.3 \times 10^{9}$	1 hr	2.15 ms
56	$2^{56} = 7.2 \times 10^{16}$	10^{3} yr	10 hr
128	$2^{128} = 3.4 \times 10^{38}$	10^{24} yr	10^{18} yr
168	$2^{168} = 3.7 \times 10^{50}$	10^{36} yr	10^{30} yr

The age of the universe is only ~13.6 billion (10^{10}) years

Encryption Key Power – Specific Example



It is estimated that to crack an encryption key using a brute force attack, a computer needs to perform a maximum of 2^*k* operations (2^k guesses), where *k* is the number of bits in the key. The average estimated time to crack is approximately half that time.

Key Length (Bits)	Maximum Number of Operations (Guesses)	Maximum Time to Crack	Estimated Average Time to Crack
16	65,536	0.000000112 seconds	0.000000056 seconds
24	16,777,216	0.0000287 seconds	0.0000143 seconds
32	4,294,967,296	0.00734 seconds	0.00367 seconds
56	72,057,594,037,927,900	34.2 hours	17.1 hours
64	18,446,744,073,709,600,000	364.7 days	182.35 days
128	3.40E+38	18,431,695,314,143,700,000 years	9,215,847,657,071,860,000 years
256	1.16E+77	6,271,980,907,862,400,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 years	3,135,990,453,931,200,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 years
512	1.34E+154	7.26E+134 years	3.63E+134 years

Note: Estimated Time to Crack is based on a 2020-era Intel i9-10900X 10 Core CPU performing 585 Dhrystone GFLOPS (giga/billion floating point operations per second) at 5.2 GHz (overclocked). Modern workstations are capable of using multiple CPUs, further decreasing time to crack, or simply splitting the workload among multiple systems.

Note: The authors acknowledge that this benchmark is based on a very specific application test and that the results are not generalizable. However, these calculations are shown to illustrate the relative difference between key length and resulting strength rather than to accurately depict time to crack.

Table Source: Principles of Information Security, by Whitman, 7th Edition

Cryptanalytic (Scan a Subset of the Key Space)

- Attack Models
  - Known / Chosen Ciphertext (KCA/CCA)
  - Known / Chosen Plaintext (KPA/CPA)
- Methodologies
  - Residual Patterns in CT
  - Pass the Hash
    - Targets authentication protocol.
  - Algorithmic/Implementation Vulnerability
  - Dictionary, Rainbow Table
  - Birthday
  - Meet-in-the-Middle, Man-in-the-Middle

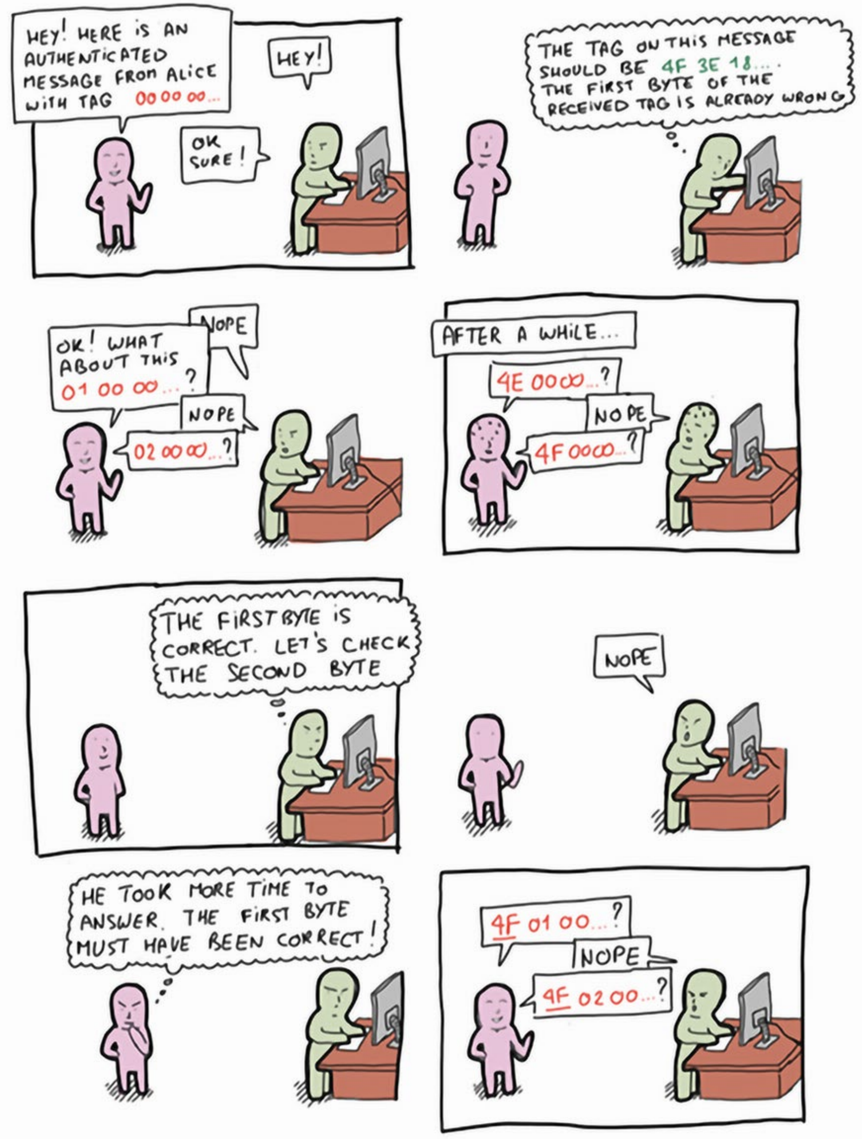

HEY! HERE IS AN AUTHENTICATED MESSAGE FROM ALICE WITH TAG 00 00 00...
HEY!
OK SURE!
THE TAG ON THIS MESSAGE SHOULD BE 4F 3E 18.... THE FIRST BYTE OF THE RECEIVED TAG IS ALREADY WRONG
OK! WHAT ABOUT THIS 01 00 00...?
NOPE
02 00 00...?
NOPE
AFTER A WHILE...
4E 00 00...?
NOPE
4F 00 00...?
THE FIRST BYTE IS CORRECT. LET'S CHECK THE SECOND BYTE
NOPE
HE TOOK MORE TIME TO ANSWER. THE FIRST BYTE MUST HAVE BEEN CORRECT!
4F 01 00...?
NOPE
4F 02 00...?

Image Source: Real-World Cryptography, 2021, by David Wong

KPA & CPA

- KPA: The attacker knows some or all of the plaintext of one or more messages (as well as the ciphertext).
  - “Dear bob”, or any constant header (e.g., Date: or Host: in protocols).

- CPA: The attacker is able to submit any plaintext the attacker chooses and obtain the corresponding ciphertext, and hopefully crack the key.

KCA & CCA

- KCA (most difficult): An attacker only has access to the encrypted traffic (ciphertext).
  - Some information about the plaintext can be guessed (e.g., language, character probability distribution, format of the message).

- CCA: The attacker is able to submit any ciphertext and obtain the corresponding plaintext.
  - **Example**: The attack on SSL 3.0 developed by Bleichenbacher of Bell Labs, which could obtain the RSA private key of a website after trying between 300,000 and 2 million chosen ciphertexts.

Types of Attacks on Encrypted Messages Summary

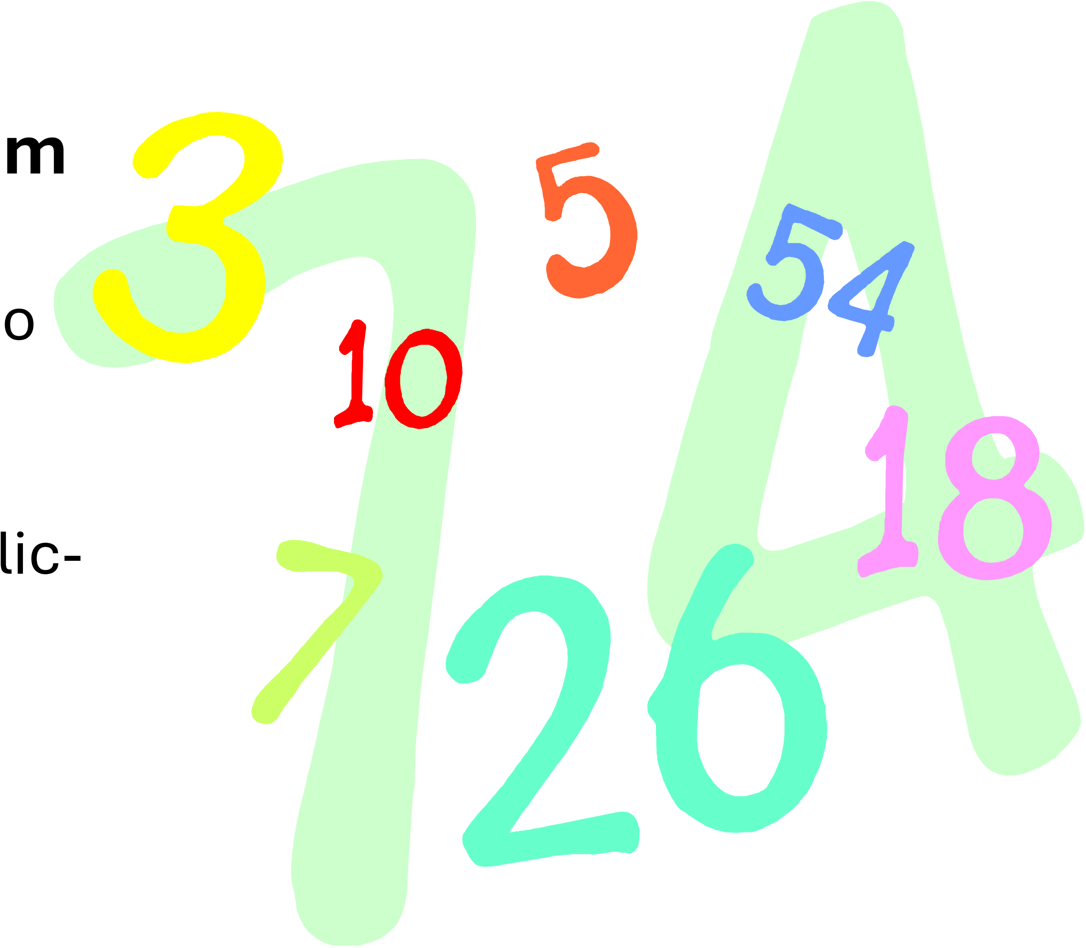
Type of Attack	Known to Cryptanalyst
Ciphertext Only **(most difficult)**	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security

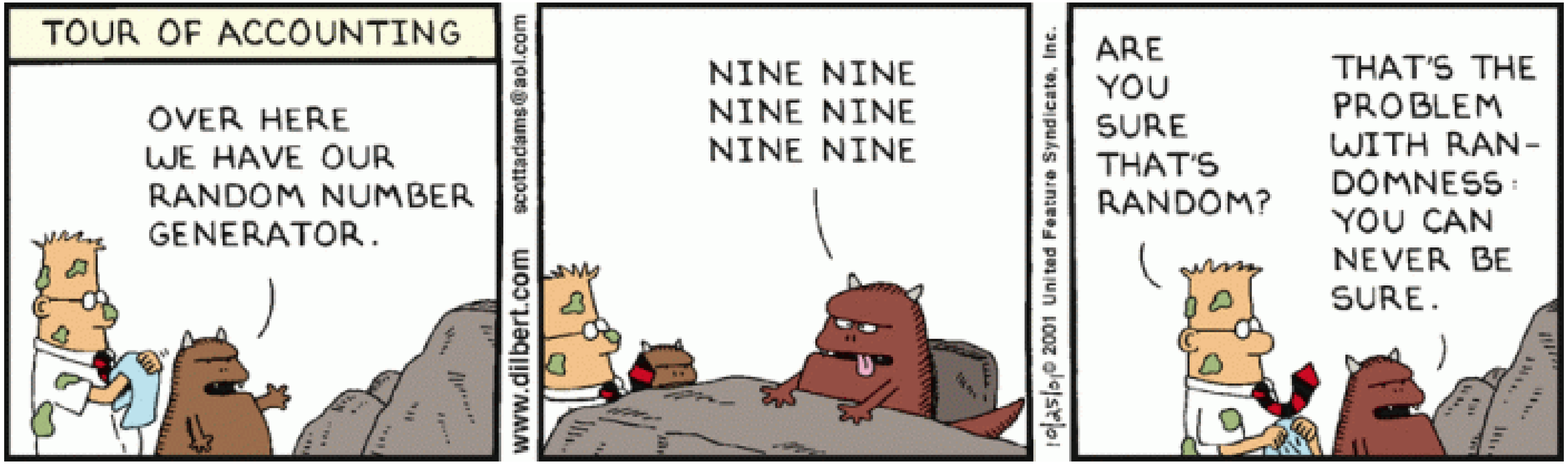
The Use of Random Numbers

- Cryptography make use of **random** binary numbers, examples:
  - Nonces are used for handshaking to prevent replay attacks.
  - Session key generation.
  - Generation of keys for the RSA public-key encryption algorithm.
  - Generation of a bit stream for symmetric stream encryption.



Randomness /1

- A sequence generator is **pseudo-random** if it has this property:
  - **1- It looks random.**
  - It passes all the statistical tests of randomness that we can find (e.g., Diehard tests https://en.wikipedia.org/wiki/Diehard_tests).


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
scottadams@aol.com
www.dilbert.com
NINE NINE NINE NINE NINE NINE
10/25/01 © 2001 United Feature Syndicate, Inc.
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

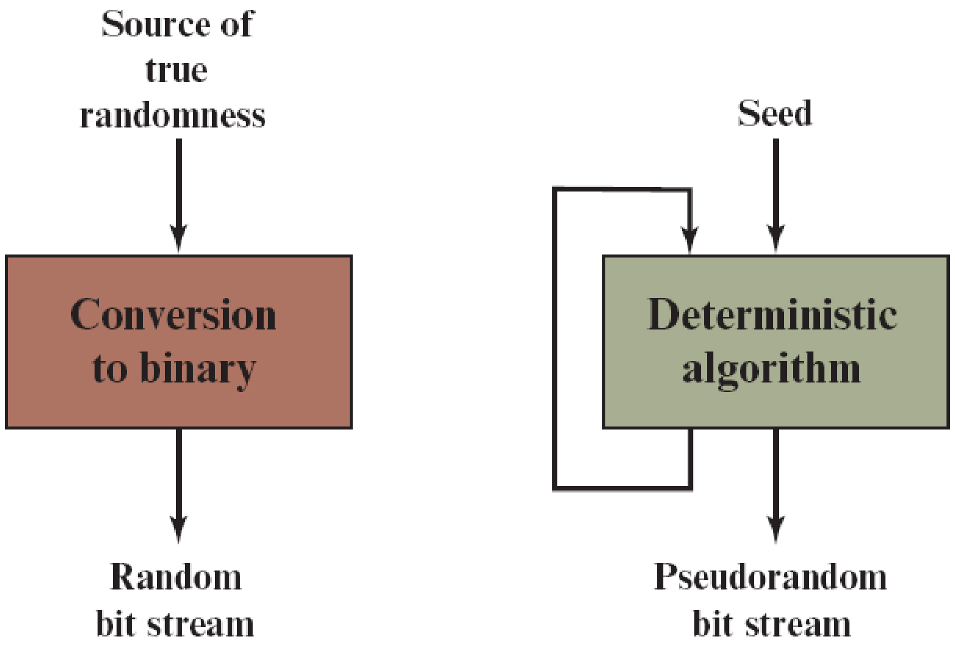
Randomness /2

- For a sequence to be **cryptographically secure pseudo-random**, it must also have this property:
    - **2. It is unpredictable.**
    - It must be computationally infeasible to predict what the next random bit will be, given complete knowledge of the algorithm or hardware generating the sequence and all of the previous bits in the stream.

Randomness /3

- A sequence generator is **true (or real) random** if it has this additional third property:
  - **3- It cannot be reliably reproduced.**
  - If you run the sequence generator twice with the exact same input (at least as exact as humanly possible), you will get two completely unrelated random sequences.

True Random and Pseudorandom Number Generators



(a) TRNG

(b) PRNG

TRNG = true random number generator
PRNG = pseudorandom number generator

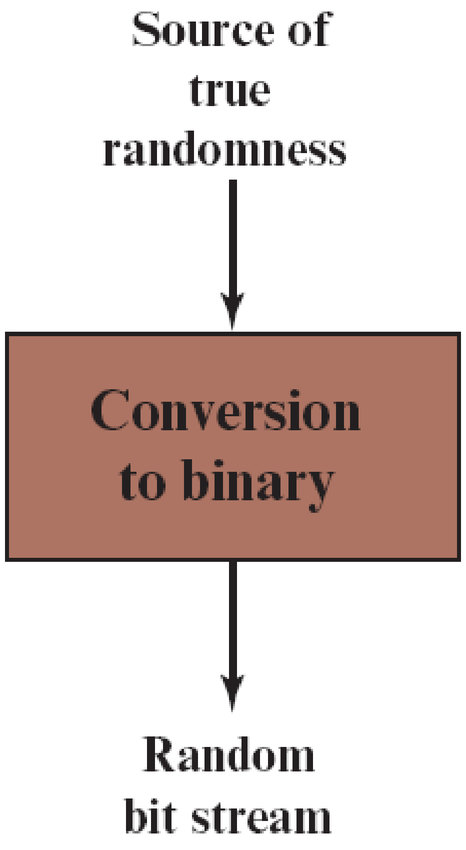
	True Random Number Generators	Pseudorandom Number Generators
Efficiency	Generally inefficient	Very efficient
Determinism	Nondeterministic	Deterministic

Source: Cryptography and Network Security: Principles and Practice – 10th Edition, by Stallings

True Random Number Generator (TRNG)

- Takes as input an entropy source that is drawn from the physical environment of the computer.
  - e.g., keystroke timing patterns, disk electrical activity, mouse movements, sound/video input.

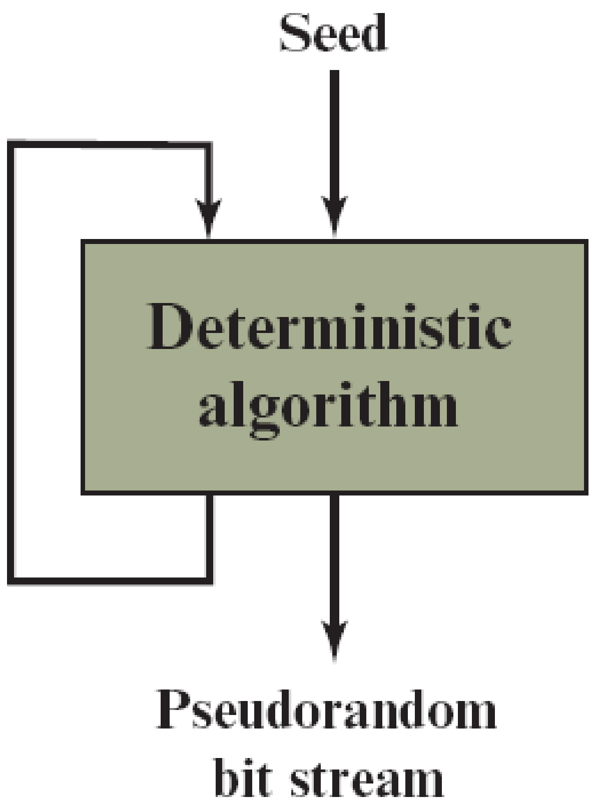
- The source, or combination of sources, serve as input to an algorithm that produces random binary output.



(a) TRNG

Pseudorandom Number Generator (PRNG)

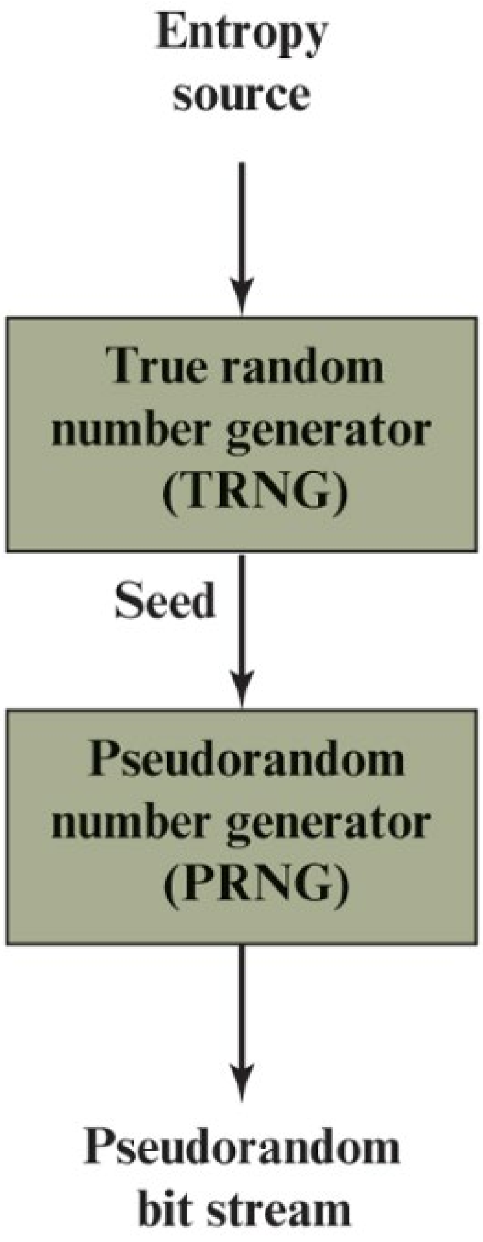
- Takes as input a fixed value, called the seed, and produces a sequence of output bits using a deterministic algorithm.
- The output bit stream is determined solely by the input value or values, so **an adversary who knows the algorithm and the seed can reproduce the entire bit stream.**



(b) PRNG

Seed Requirements

- For cryptographic applications, the seed that serves as input to the PRNG must be secure, and is generated by a TRNG.


Entropy source
True random number generator (TRNG)
Seed
Pseudorandom number generator (PRNG)
Pseudorandom bit stream

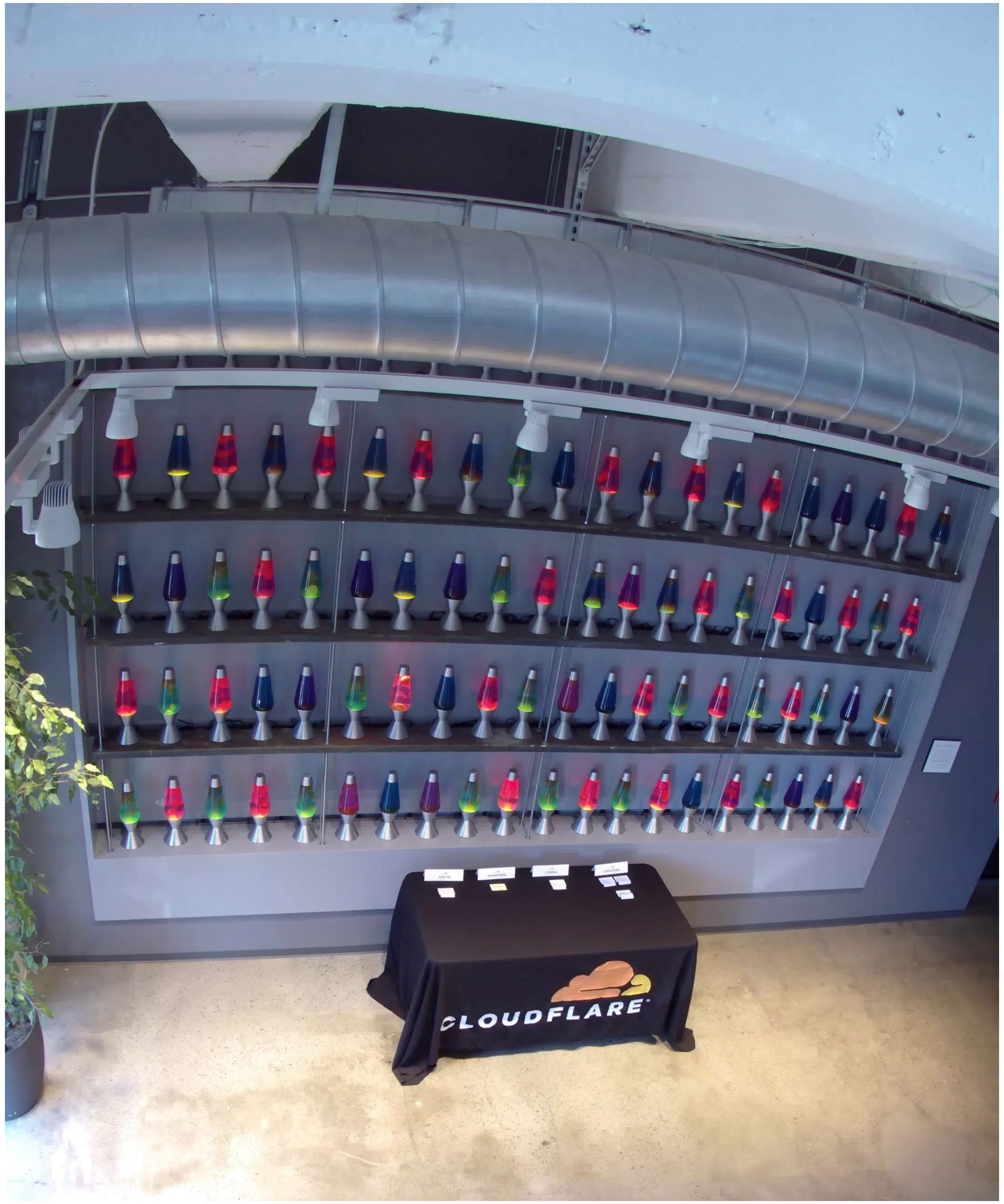
Source: Cryptography and Network Security: Principles and Practice – 10th Edition, by Stallings

"A wall of lava lamps in the lobby of our San Francisco office provides an unpredictable input to a camera aimed at the wall.

A video feed from the camera is fed into a CSPRNG, and that CSPRNG provides a stream of random values that can be used as an extra source of randomness by our production servers.

Since the flow of the "lava" in a lava lamp is very unpredictable, "measuring" the lamps by taking footage of them is a good way to obtain unpredictable randomness."

The view from the camera

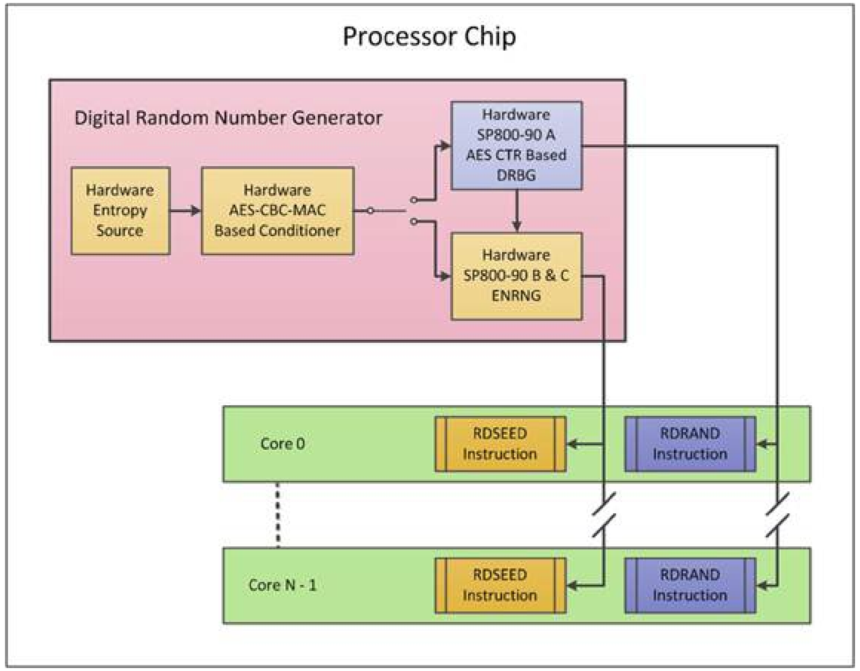


Source: https://blog.cloudflare.com/randomness-101-lavarand-in-production/

Intel Digital Random Number Generator

- The Entropy Source runs on a self-timed circuit and uses **thermal noise** within the silicon to output a random stream of bits at the rate of 3 GHz.

- Thermal noise generated by random motion of free electrons in a conductor resulting from thermal agitation.


Processor Chip
Digital Random Number Generator
Hardware Entropy Source
Hardware AES-CBC-MAC Based Conditioner
Hardware SP800-90 A AES CTR Based DRBG
Hardware SP800-90 B & C ENRNG
Core 0
RDSEED Instruction
RDRAND Instruction
Core N - 1
RDSEED Instruction
RDRAND Instruction

Intel Processor Chip with Random Number Generator

Source: https://www.intel.com/content/www/us/en/developer/articles/guide/intel-digital-random-number-generator-drng-software-implementation-guide.html

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security

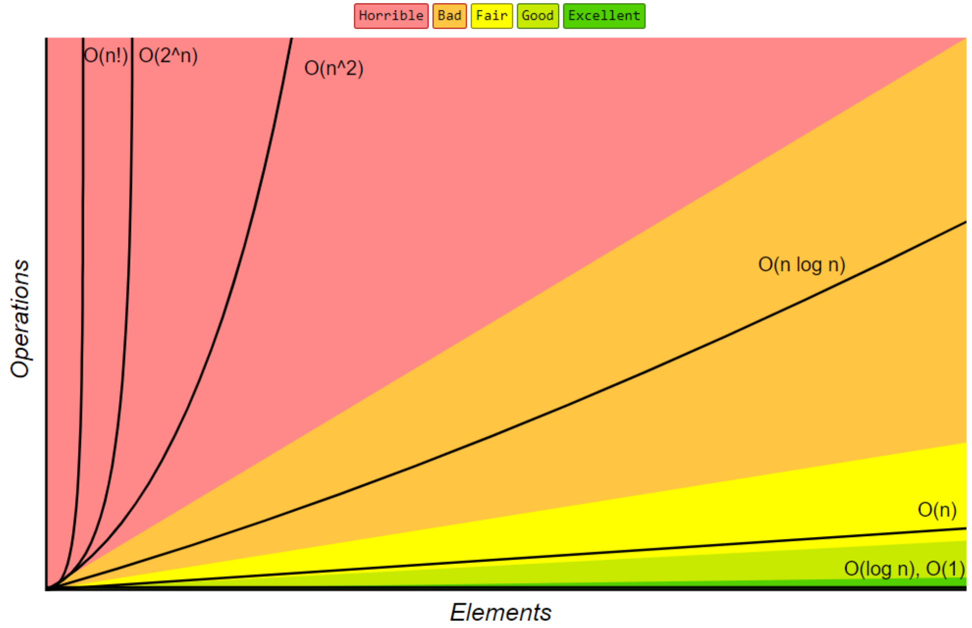
Encryption Scheme Security

- **Unconditionally secure**
  - No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there.

- **Computationally secure**
  - The **cost** of breaking the cipher **exceeds the value** of the encrypted information.
  - The **time** required to break the cipher **exceeds the useful lifetime** of the information.

Cost of Computation

- Big O notation is used to describe the complexity of a function or algorithm.
- It models the number of operations required as the input size increases.
- In a polynomial, the variable is raised to a power: n^a, where *a* is a fixed constant.
- In an exponential function, the variable is the exponent: 2^n



Big O Complexity

Source: https://www.bigocheatsheet.com/

Topics – Cryptology

- Introduction
- Cryptography
- Cryptanalysis
- The Use of Random Numbers
- Encryption Scheme Security