Introduction to Cybersecurity

Chapter 3

Cryptography

Chapter Objectives

By the end of this chapter, you should be able to:

- Describe cryptography
- Describe hash, symmetric, and asymmetric cryptographic algorithms
- Describe Digital Signature and Certificates
- Explain different cryptographic attacks
- Learn to use password-cracking tools
- Use Password recovery tool to crack password hashes: Hashcat/JtR

Cryptography

- “Hidden writing”
- “the art of writing or solving codes” – Concise Oxford English Dictionary

Cryptography

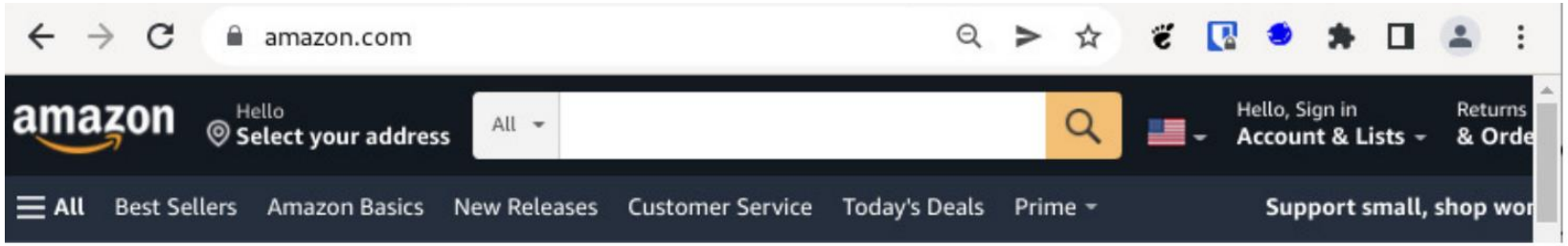
- “Hidden writing”
- “the art of writing or solving codes” – Concise Oxford English Dictionary

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior- "Cryptography". In J. Van Leeuwen (ed.). Handbook of Theoretical Computer Science. Vol. 1. Elsevier.

Cryptography that consists of: (i) an approved algorithm; (ii) an implementation that has been approved for the protection of classified information in a particular environment; and (iii) a supporting key management infrastructure- **NIST**

Did you use any cryptography today?

If you visited a website with HTTPS (the lock icon in the address bar), encryption protected your data in transit



- https invokes the TLS protocol
- TLS uses cryptography
- TLS is in ubiquitous use for secure communication: shopping, banking, Netflix, gmail, Facebook, ...

Many messaging apps use end-to-end encryption.

- WhatsApp, Signal, iMessage/FaceTime, Viber, Telegram, LINE, ...

Which ones do you use?

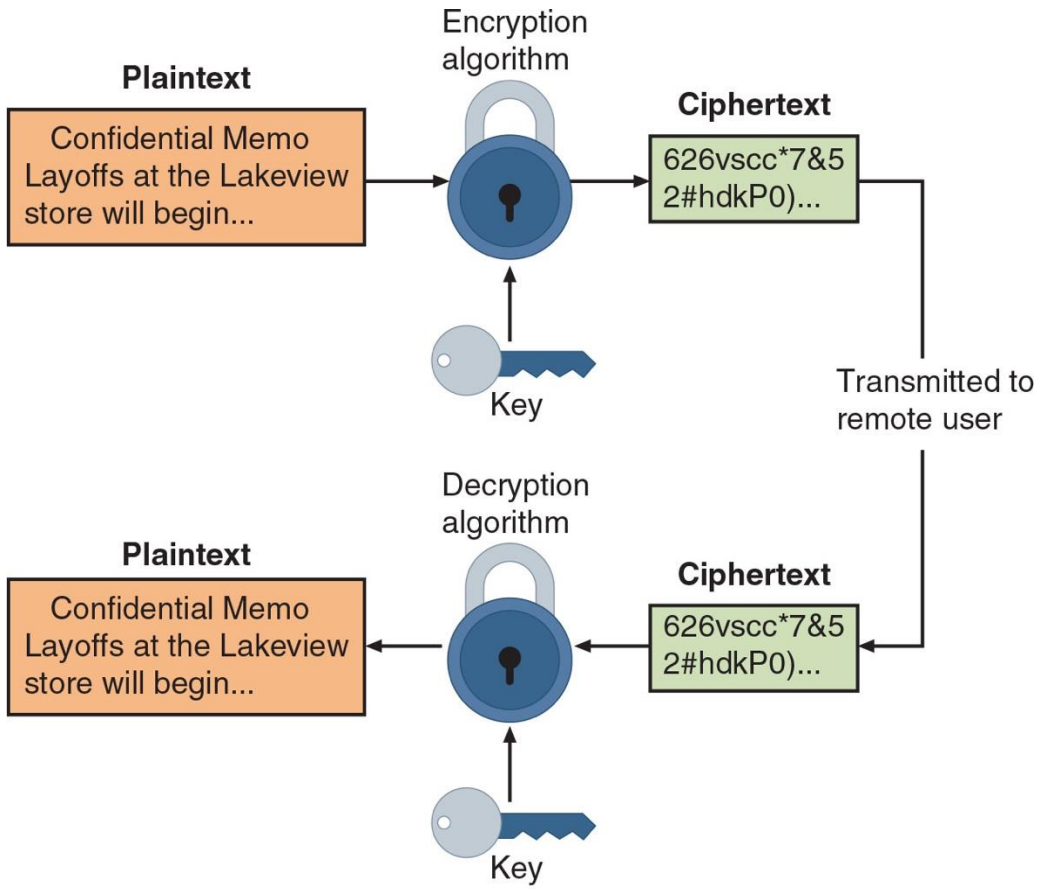
Key Terms

- **Cryptology**: Cryptology is the study of cryptography and cryptanalysis
- **Cryptography** is the process of hiding or coding information so that only the person a message was intended for can read it
- **Cryptanalysis**: study of breaking or analyzing cryptographic systems to find weaknesses

- **Cleartext/plaintext** - What you are reading now
- **Encrypt** - convert information or data into code to prevent unauthorized access
- **Decrypt** - convert an encoded or unclear message into something intelligible, to plaintext
- **Cipher** - An algorithm to perform encryption and/or decryption
- **Ciphertext** - encrypted text transformed from plaintext using an encryption algorithm
- **Key** - A secret value or parameter used by a cryptographic algorithm to control the transformation of data
- **Hash Function** - A one-way function that converts input data into a fixed-length string of characters, used for data integrity and password storage.

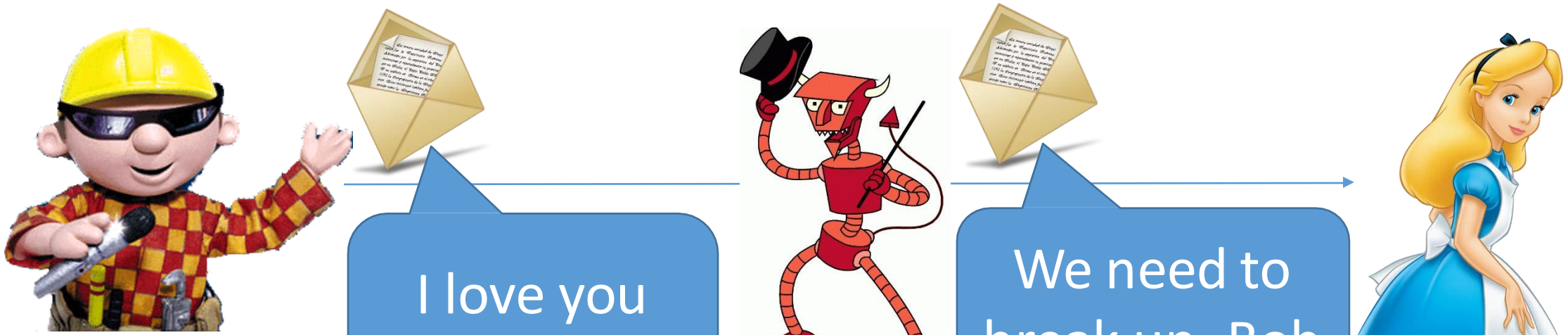
Read more

Cryptography



Why do we need cryptography?

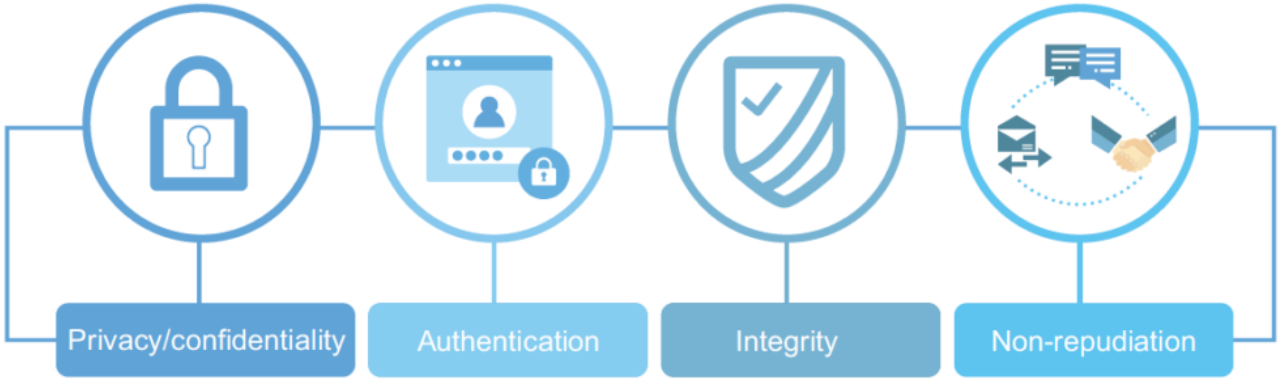
- Confidentiality (Security/Privacy)
  - Only intended recipient can see the communication
- Integrity
  - Ensures data cannot be modified in an unauthorized manner since the data was created, stored or transmitted by an authorized entity



Cryptography Security Services

Cryptography provides four main security services:

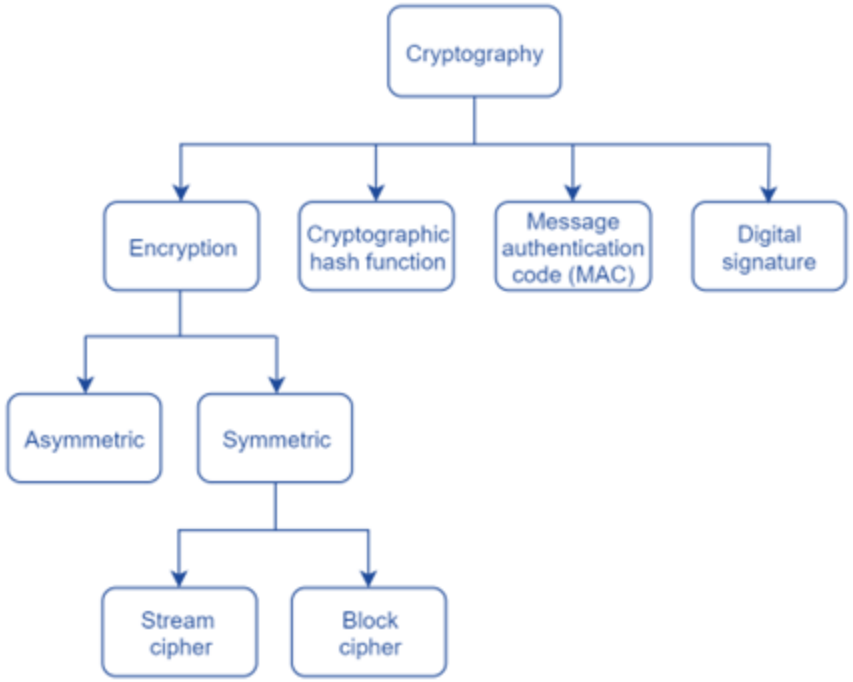
1. **Confidentiality** ensures only authorized parties can view it
2. **Integrity** ensures data cannot be modified in an unauthorized manner since the data was created, stored or transmitted by an authorized entity
3. **Authentication** ensures sender can be verified through cryptography
4. **Nonrepudiation** proves that a user performed an action, a message sender cannot deny sending a message



Cryptographic primitives

Basic building blocks of cryptography

- **Encryption**: uses an algorithm and a key to hide the meaning of a message
  - **symmetric encryption**: same keys
  - **asymmetric encryption**: two different, but mathematically related keys are used for encryption and decryption
- **Cryptographic hash function**: outputs a fixed-length string for a variable-length input string, provides the data **integrity** security service
- **Message authentication code (MAC)**: uses a cryptographic hash function and **symmetric** encryption to provide the data **integrity** and **authenticity** security services
- **Digital signature**: uses a cryptographic hash function and **asymmetric** encryption to provide the data **integrity**, **authenticity**, and **non-repudiation** security services



Familiar names for the protagonists in security protocols

- **Alice-** First participant
- **Bob** - Second participant
- **Carol** - Participant in three-and four-party protocols
- **Dave** - Participant in four-party protocols
- **Eve** - Eavesdropper
- **Mallory** - Malicious attacker
- **Sara** - A server
- **Trent** – A trusted third party

Security Notations

K_A	Alice's secret key
K_B	Bob's secret key
K_{AB}	Secret key shared between Alice and Bob
K_{Apriv}	Alice's private key (known only to Alice)
K_{Apub}	Alice's public key (published by Alice for all to read)
$\{M\}_K$	Message M encrypted with key K
$[M]_K$	Message M signed with key K

The Golden Rule

- ***"Don't roll your own crypto" (*Why shouldn't we create our own security schemes?*)***

- You can roll your own, but you probably will make a major security mistake if you are not an expert in security/cryptography or have had your scheme analyzed by multiple experts

- *The reason: https://security.stackexchange.com/questions/18197/whyshouldnt-we-roll-our-own*

- *"Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break. It's not even hard. What is hard is creating an algorithm that no one else can break, even after years of analysis. And the only way to prove that is to subject the algorithm to years of analysis by the best cryptographers around." –Bruce Schneier*

- *If you are not convinced of "Don't Roll Your Own [Cryptography/Security]", then you probably are not an expert and there are many mistakes you likely will make- dr jimbob*

Snake Oil Ad

Crypto-Gram
February 15, 1999

by Bruce Schneier

"Encryptor 4.0 uses a unique in-house developed incremental base shift algorithm. Decryption is practically impossible, even if someone manages to reverse engineer our program to obtain the algorithm, the decryption of a file depends on the exact password (encryption key). Even if someone is guessing the encryption key the file will only be decrypted correctly if the encryption key is 100 percent correct. See the IMPORTANT WARNING on our Web site http://ten4.com/encryptor."

Read what are common snake-oil warning signs, and how you can pre-judge products from their advertising claims

Snake oil: https://www.schneier.com/crypto-gram/archives/1999/0215.html

Security of a Cryptosystem

Is any crypto algorithm perfectly secure?

- Security is based on the assumption that it is computationally infeasible to break them within a reasonable amount of time using available resources, such as computational power and time
- As technology advances and computing capabilities improve, what was once considered secure may become vulnerable

- **Tradeoff 1**: cost of breaking a cipher exceeds the value of the encrypted information
- **Tradeoff 2**: time required to break the cipher exceeds the useful lifetime of the information
- Very difficult to estimate cost and time required to break a cipher
    - There is always brute force
    - ...and then there is plain-old stealing or just asking for it

Historical Cryptosystems

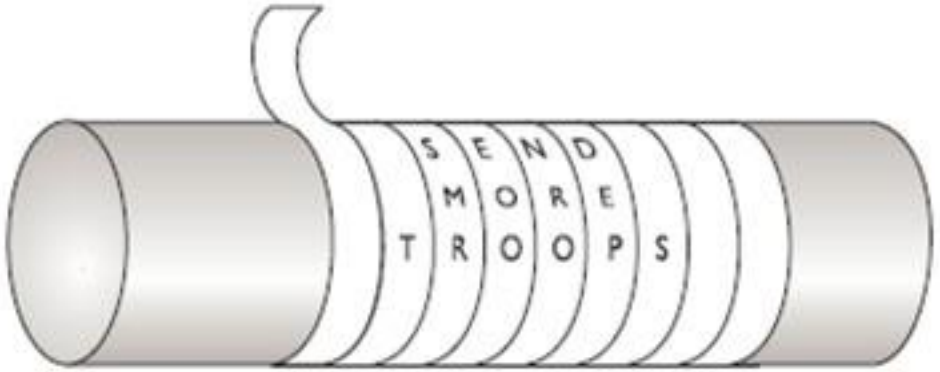
Cryptography History

- 2500+ years
- Ongoing battle between Codemakers and codebreakers

- Steganography comes from the Greek steganos , meaning covered or secret, and graphy , meaning writing or drawin
- Steganography simply takes one piece of information and hides it within another

Scytale cipher 400 B.C.

- Write a message on a sheet of papyrus that was wrapped around a staff;
- The papyrus was delivered and wrapped around a different staff by the recipient;
- The message was only readable if it was wrapped around the correct size staff, which would make the letters properly match up



Steganography
Early Examples

In his history of the Persian Wars, Herodotus tells of a messenger who shaved his head and allowed a secret message to be tattooed on his scalp. He waited until his hair grew back. Then he journeyed to where the recipient awaited him and shaved his head again. The message was revealed. It was history's first use of steganography.

Steganography
Invisible Ink

Ancient Romans used to write between lines using invisible ink based on various natural substances such as fruit juices, urine, and milk. Their experience was not forgotten: even nowadays children play spies and write secret messages that appear only when heated.

Historical cryptosystems

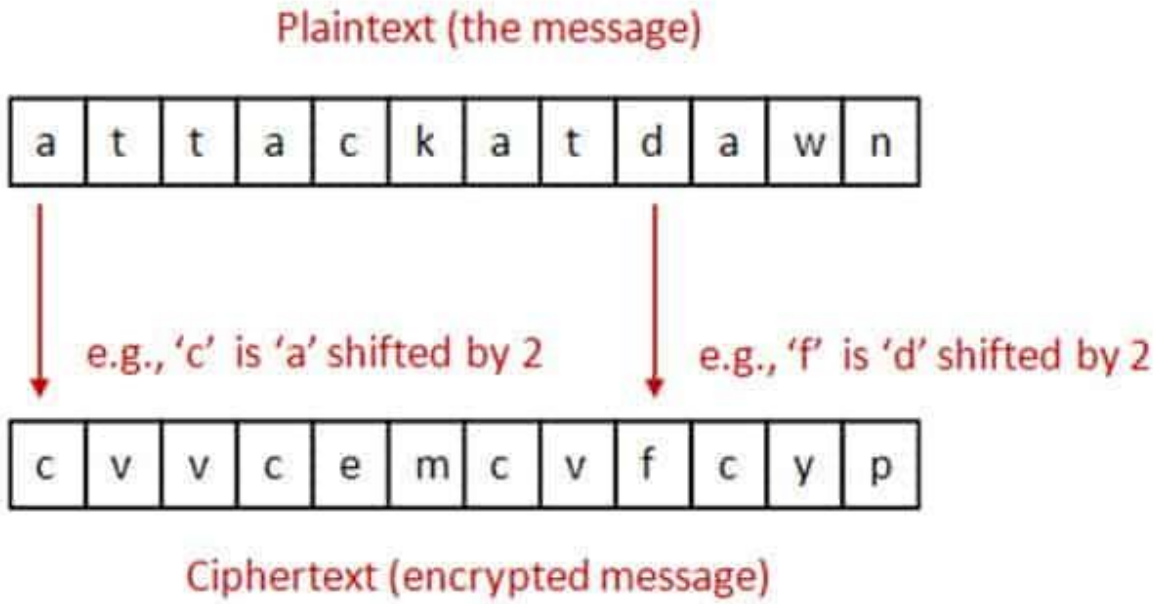
Substitution cipher: substitutes or replaces one element of plaintext with a different element to generate a ciphertext

- **Monoalphabetic substitution** cipher, a letter in a plaintext is replaced by a fixed letter to generate a ciphertext. Ex: The letter 'C' in a plaintext is replaced by the letter ‘F’
  - **Caesar cipher** is a well-known monoalphabetic cipher

- **Polyalphabetic substitution** cipher, a specific letter in a plaintext is replaced by different letters in a ciphertext. Ex: The letter 'C' in a plaintext may be replaced by the letters 'P', 'T', 'X', or any other letter depending on rules
  - **Vigenère cipher** is a classic example of a polyalphabetic cipher

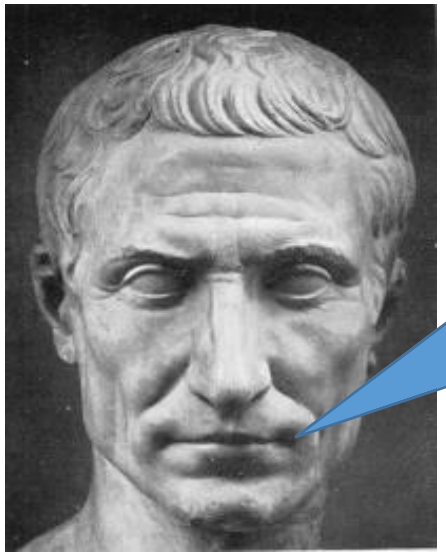
Caesar cipher (Caesar shift cipher)

- Caesar cipher is one of the oldest and simplest encryption techniques in history
- Julius Caesar (100-44 B.C)
- It's a substitution cipher
- Each letter in the plaintext is shifted a fixed number of positions in the alphabet
- This shift is known as the "key" or "shift value"



positive shift (e.g., +2) moves letters **forward** in the alphabet (A → C, B → D)
negative shift (e.g., -11) moves letters **backward** (M → B, N → C)

Caesar Cipher



Three shall be the number of thy shifting and the number of thy shifting shall be three. Four shalt thou not shift, neither shift thou two, excepting that thou then proceed to three. Five is right out…..

Caesar adopted the shift cipher with secret key k=3

Caesar Cipher (Example)

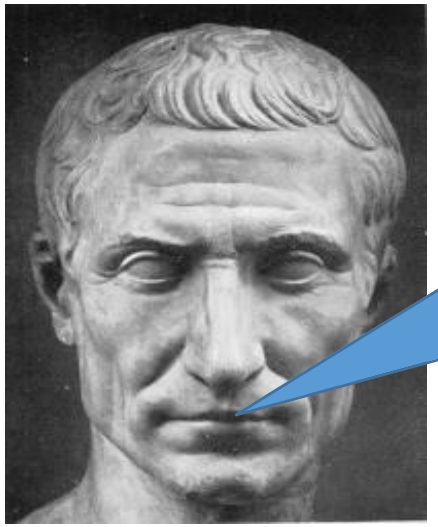
BEGINTHEATTACKNOW

→

EHJLQWKHDWWDFNQRZ

Caesar adopted the shift cipher with secret key k=3

Caesar Cipher (Example)



BEGINTHEATTACKNOW
→
EHJLQWKHDWWDFNQRZ

Immediate Issue: anyone who knows method can decrypt (since k=3 is fixed)

Shift Cipher: Brute Force Attack

- Ciphertext: “lwxrw ztn sd ndj iwxcz xh gxvwi?”
  - k=1 → m = “mxysx auo te oek jxyda yi hywxj?”
  - k=2 → m=“nyzty bvp uf pfl kyzeb zj izxyk?”
  - k=3 → m=“ozauz cwq vg qgm lzafc ak jayzl?”
  - k=4 → m = “pabva dxr wh rhn mabgd bl kbzam?”
  - k=5 → m=“qbcwb eys xi sio nbche cm lcabn?”
  - k=6 → m=“rcdxc fzt yj tjp ocdif dn mdbco?”

Did you find the key here?

Shift Cipher: Brute Force Attack

- Ciphertext: “lwxrw ztn sd ndj iwxcz xh gxvwi?”
  - …
  - k=7 → m=“sdeyd gau zk ukq pdejg eo necdp?”
  - k=8 → m=“tefze hbv al vlr qefkh fp ofdeq?”
  - k=9 → m = “ufgaf icw bm wms rfgli gq pgefr?”
  - k=10 → m=“vghbg jdx cn xnt sghmj hr qhfgs?”
  - k=11→ m= “which key do you think is right?”
  - k=12→ m= “xijdi lfz ep zpv uijol jt sjhiu?”

"Zkdw lv wkh sdvvzrug?"

What is the original plaintext message?

You suspect it was encrypted using a simple shift. Try brute-forcing different shift values to **decrypt** it.

Permutation Cipher

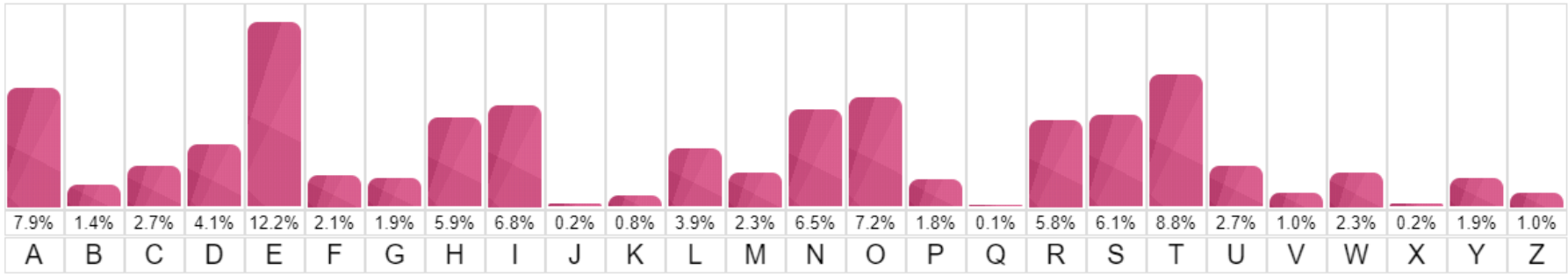
- Monoalphabetic Substitution
- Unlike a Caesar cipher (which shifts letters by a fixed amount), a permutation cipher shuffles the alphabet using a fixed key
- **Key is a randomly shuffled alphabet** (not just a shift)
- Each letter is substituted using a **fixed mapping** from a **scrambled alphabet**
- A → X, B → M, C → J, D → O, E → K, F → L, G → W, H → P, I → Q, etc.

- Plaintext: **HELLO**

- Ciphertext: **PKKKW** (Each letter replaced using the key)

Permutation Cipher

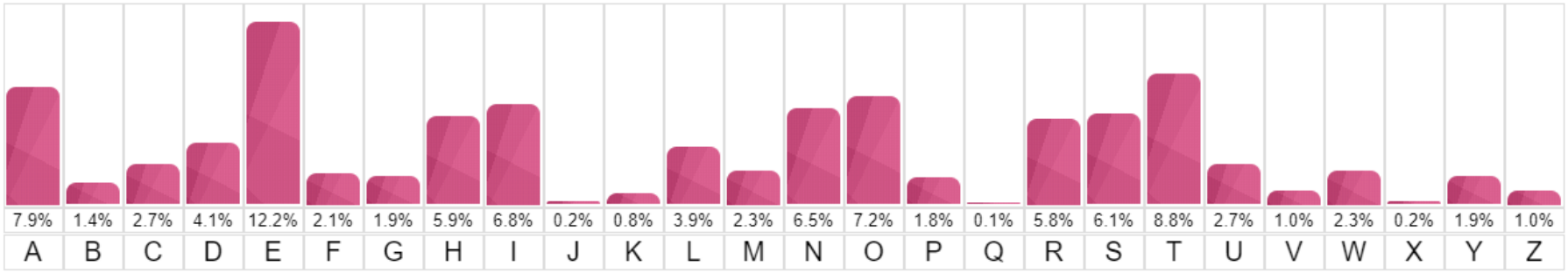
- If multiple ciphertexts are encrypted using the same permutation, frequency analysis can be used to **break the cipher** (i.e., identifying common letters like 'E', 'T', or 'A' based on their frequencies in English, Z, Q and X are not as frequently used.)
- Look for **common bigrams (TH, HE, IN, ER)**
- Consider **one-letter words**—they are likely **A** or **I**
- We can then recognise patterns/words in the partly decoded text to identify more substitutions
- A major **weakness** of a fixed permutation is that once the mapping is discovered, **all messages using the same key can be decrypted**



What is the Plain Text?

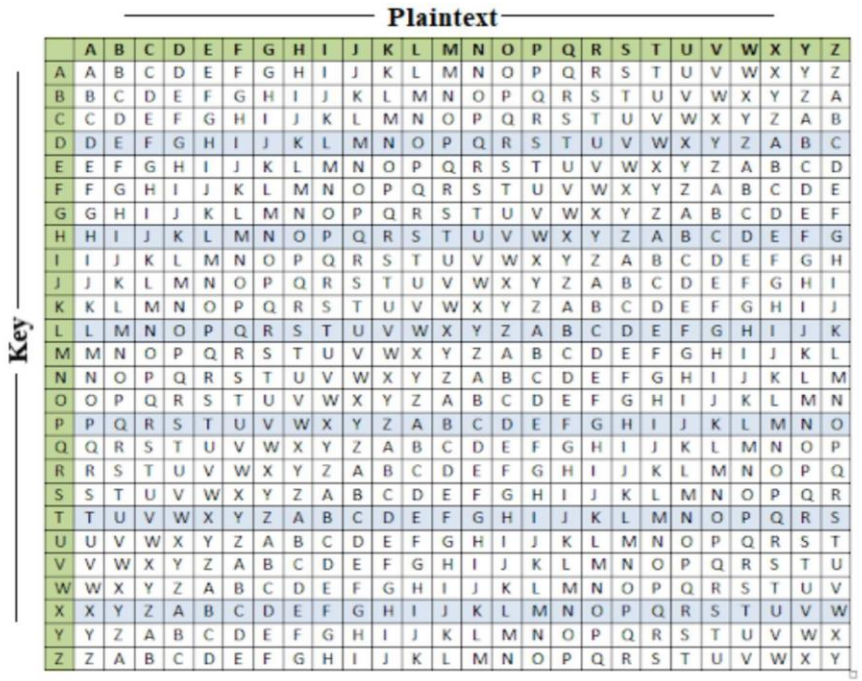
"WKLW LV D VHFUHW PHVVDJH" - harder to fully decode without more samples

- Letter that appears the most often in English is **E** (~12.7%)
- Look for **common bigrams (TH, HE, IN, IS ER)**
- Consider **one-letter words**—they are likely **A** or **I**
- Try substituting different letters and look for recognizable words.



Polyalphabetic substitution

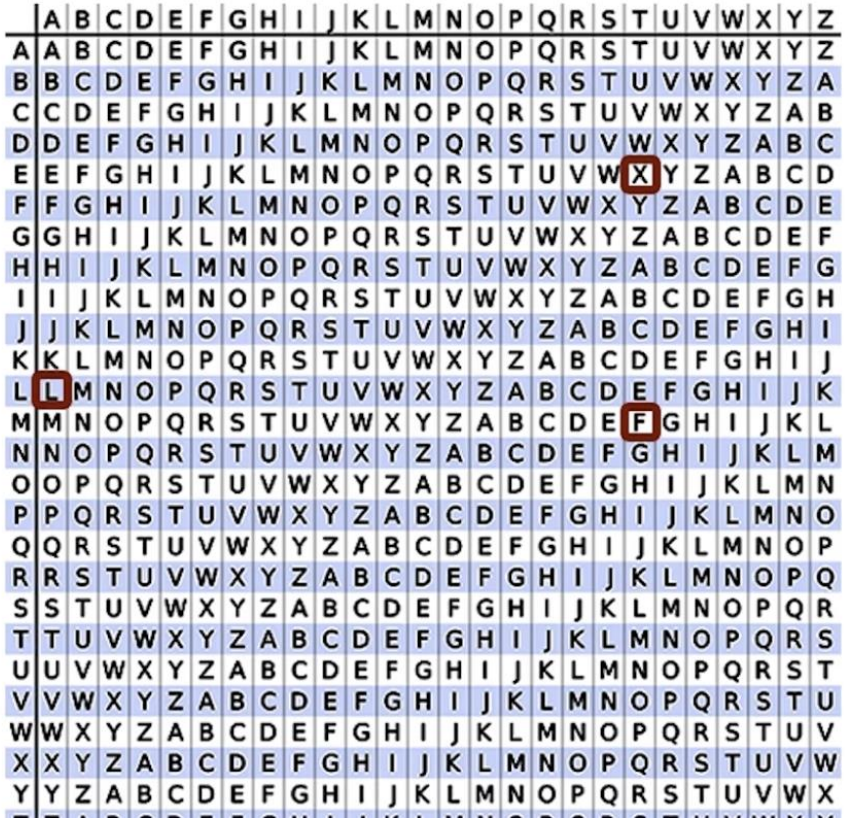
- A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets
- Ex: Vigenère Cipher
- Encryption of the original text is done using the Vigenère square or Vigenère table
- Table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Ciphers



Polyalphabetic substitution

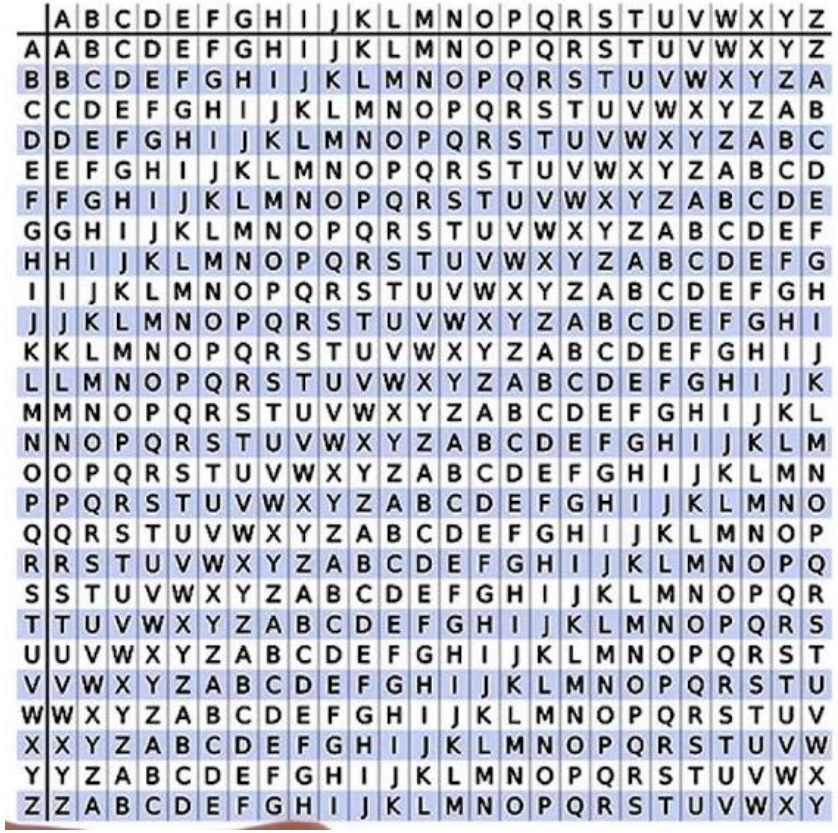
Vigenere Cipher

- Plaintext:
  ATTACKATDAWN
- Key:
  LEMON
- Keystream:
  LEMONLEMONLE
- Ciphertext:
  LXFOPVEFRNHR



Exercise: Find Message

Key is DECEPTIVE and the Cipher is *ZI CVT WQNGRZGVTW*

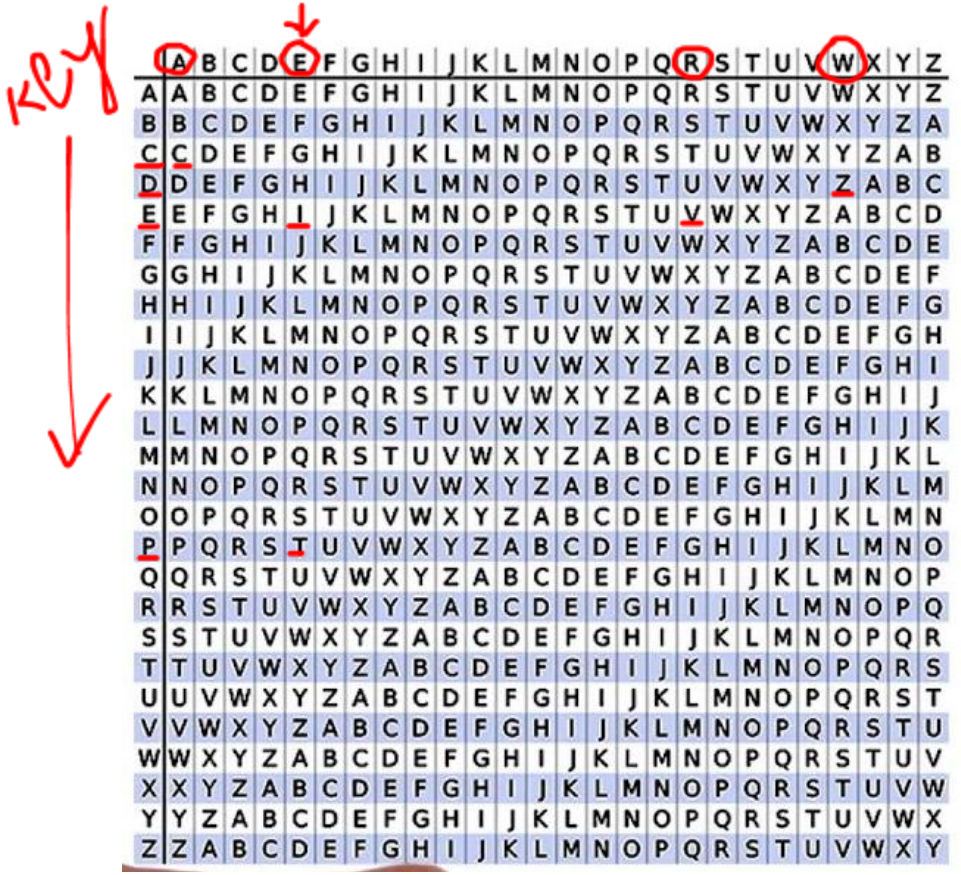


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

Exercise: Find Message

Key is DECEPTIVE and the Cipher is *ZI CVT WQNGRZGVTW*

```
Message:  WE  ARE  DISCOVERED
Key:       DE  CEP  TIVEDECEPT
Cipher:    ZI  CVT  WQNGRZGVTW
```



Transposition cipher

- **A transposition cipher**: reorders elements of a plaintext in the ciphertext without adding or removing elements
  - Ex: word 'elephant' would then become 'tnahpele'
- **Rail fence cipher**: plaintext's letters are written diagonally downwards on successive rails of an imaginary fence
  - Ciphertext is read rail-by-rail from the top rail to the bottom rail
  - Key is the number of rails

Plaintext T H I S I S A S E C R E T M E S S A G E

Rail Fence Encoding ***key = 4***

T						A						T						G	
	H				S		S				E		M				A		E
		I		I				E		R				E		S			
			S						C						S				

Ciphertext T A T G H S S E M A E I I E R E S S C S

https://crypto.interactive-maths.com/rail-fence-cipher.html

Plaintext T H I S I S A S E C R E T M E S S A G E

Rail Fence Encoding

key = 4

T						A						T						G	
	H				S		S				E		M				A		E
		I		I				E		R				E		S			
			S						C						S				

Ciphertext T A T G H S S E M A E I I E R E S S C S

4 rows 20 columns

T						A						T						G	
	-				-		-				-		-				-		-
		-		-				-		-				-		-			
			-						-						-				

Decryption process for the Rail Fence Cipher involves reconstructing the diagonal grid used to encrypt the message. We start writing the message, but leaving a dash in place of the spaces yet to be occupied

Exercise: Decryption Rail Fence

- Decryption process for the Rail Fence Cipher involves reconstructing the diagonal grid used to encrypt the message. We start writing the message, but leaving a dash in place of the spaces yet to be occupied

- Example: ciphertext "TEKOOHRACIRMNREATANFTETYTGHH", encrypted with a key of 4
- We have a table with 4 rows because the key is 4, and 28 columns as the ciphertext has length 28
- Start by placing the "T" in the first square. You then dash the diagonal down spaces until you get back to the top row, and place the "E" here. Continuing to fill the top row you get the pattern below

T						E						K						O						O			
	-				-		-				-		-				-		-				-		-		
		-		-				-		-				-		-				-		-				-	
			-						-						-						-						-

Decryption Rail Fence

- Example: ciphertext "TEKOOHRACIRMNREATANFTETYTGHH", encrypted with a key of 4

- Continuing this row-by-row, we get the successive stages shown below.

T						E						K						O						O			
	H				R		A				C		I				R		M				N		R		
		-		-				-		-				-		-				-		-				-	
			-						-						-						-						-

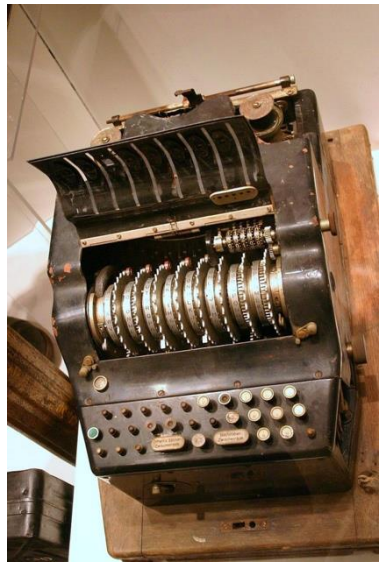
Decryption Rail Fence

- The fourth and final stage in the decryption process.
- From this we can now read the plaintext off following the diagonals to get "they are attacking from the north".

T						E						K						O						O			
	H				R		A				C		I				R		M				N		R		
		E		A				T		A				N		F				T		E				T	
			Y						T						G						H						H

The War Machines: Enigma

- Enigma is developed by Arthur Scherbius
  - Used by the Germans during World War II
  - Enigma substituted each letter typed by an operator
  - Substitutions were computed using a key and a set of switches or rotors
  - Code was broken first by a group of Polish cryptographers
    - Machine for breaking the code was called the "Bombe"
    - Bombe machine was primarily designed by Alan Turing and improved by Gordon Welchman at Bletchley Park



https://www.cia.gov/legacy/museum/artifact/enigma-machine/

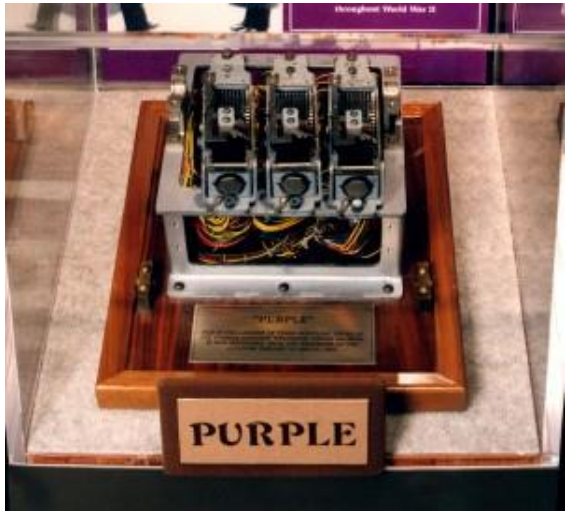
32 cm x 26 cm x 15 cm

https://www.cia.gov/legacy/museum/artifact/enigma-machine/

Official Name: Type B Cipher Machine

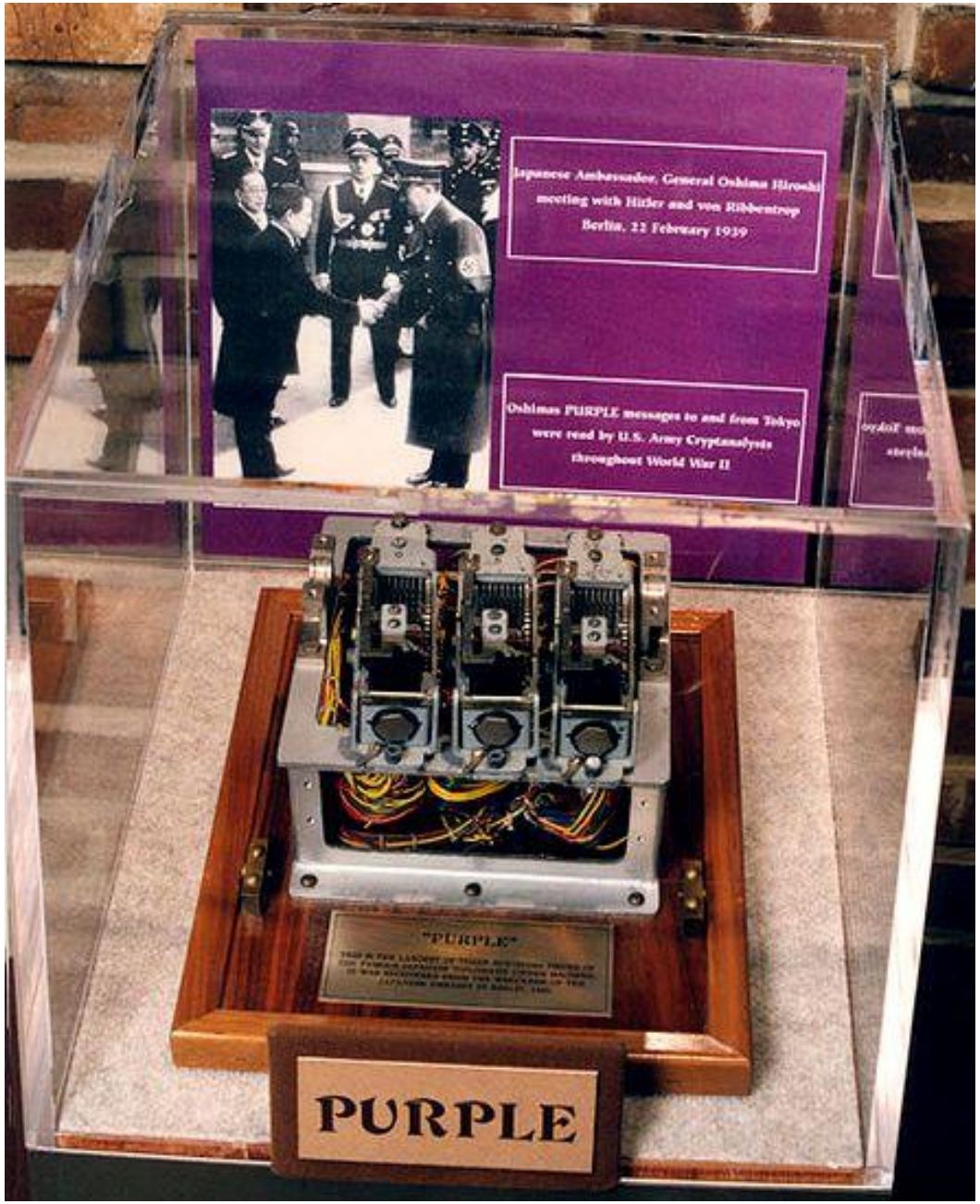
The War Machines: The Purple Machine

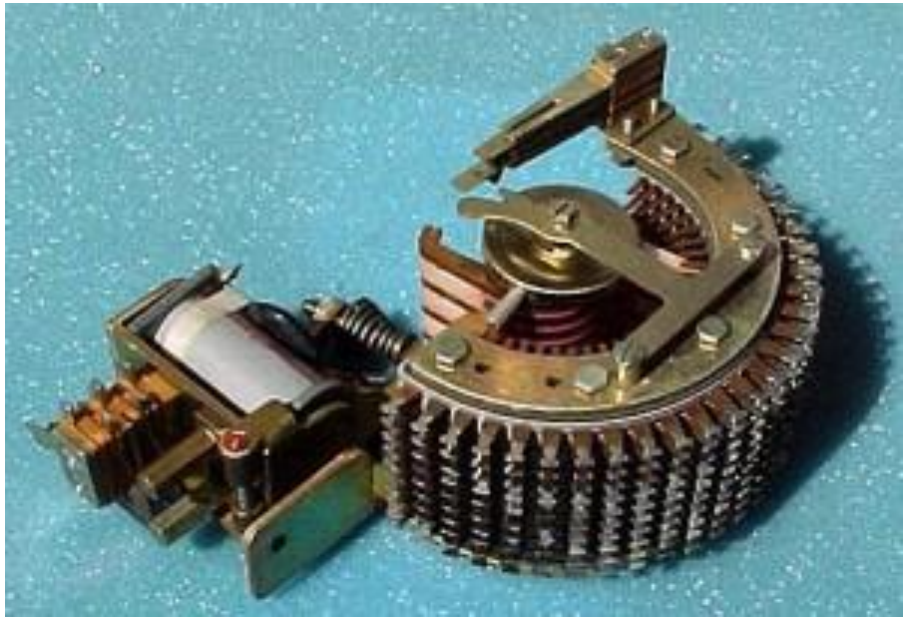
- Purple Machine developed and used by the Japanese during World War II
  - ➢Used a series of cylinders, each giving one substitution, which rotated and changed after each letter was encrypted
  - ➢Used rotor-based encryption, similar to the German Enigma machine, but with a different mechanical and electrical structure
  - ➢Code was broken by William Frederick Friedman
    - Known as the "Father of U.S. Cryptanalysis"

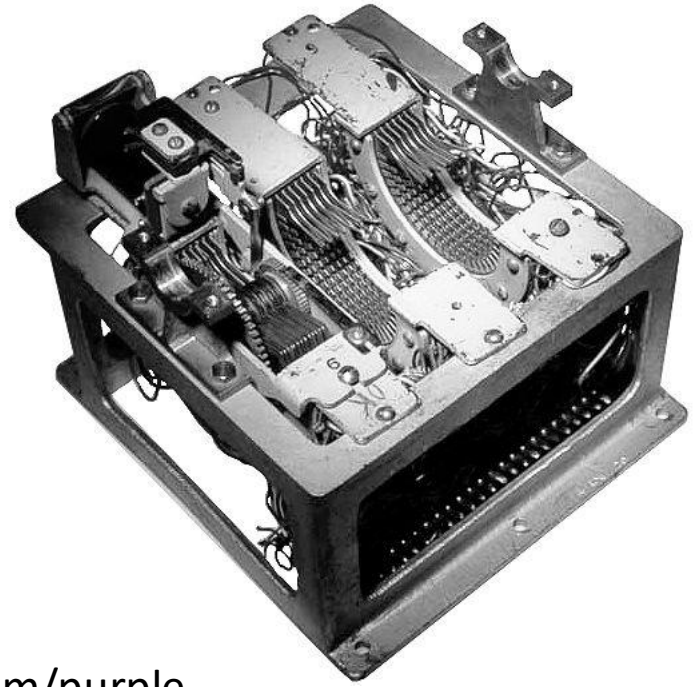


Remnant of the Japanese Purple cipher recovered from their bombed-out embassy in Berlin

https://www.wondersandmarvels.com/2013/02/secrets-abroad-a-history-of-the-japanese-purple-machine.html
https://ciphermachines.com/purple







https://ciphermachines.com/purple

Trivia: Which movie is this?

The Imitation Game (2014)

Who Uses Cryptography

- Traditionally: Military
- Modern Times: Everyone!

Modern Cryptography

- Modern cryptography focuses on securing digital communication, data integrity, and authentication using mathematical algorithms and computational technique

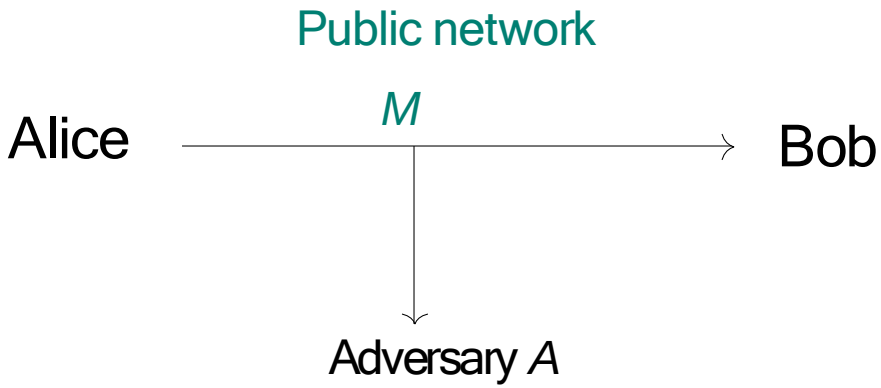
  - *Security of a "practical" system must rely not on the impossibility but on the computational difficulty of breaking the system.*

("Practical" = more message bits than key bits)

- Rather than: *"It is impossible to break the scheme"*

  - Attacks can exist as long as cost to mount them is prohibitive, where Cost = computing time/memory, $$$

What is cryptography about?



Adversary: clever person with powerful computer

Security goals:

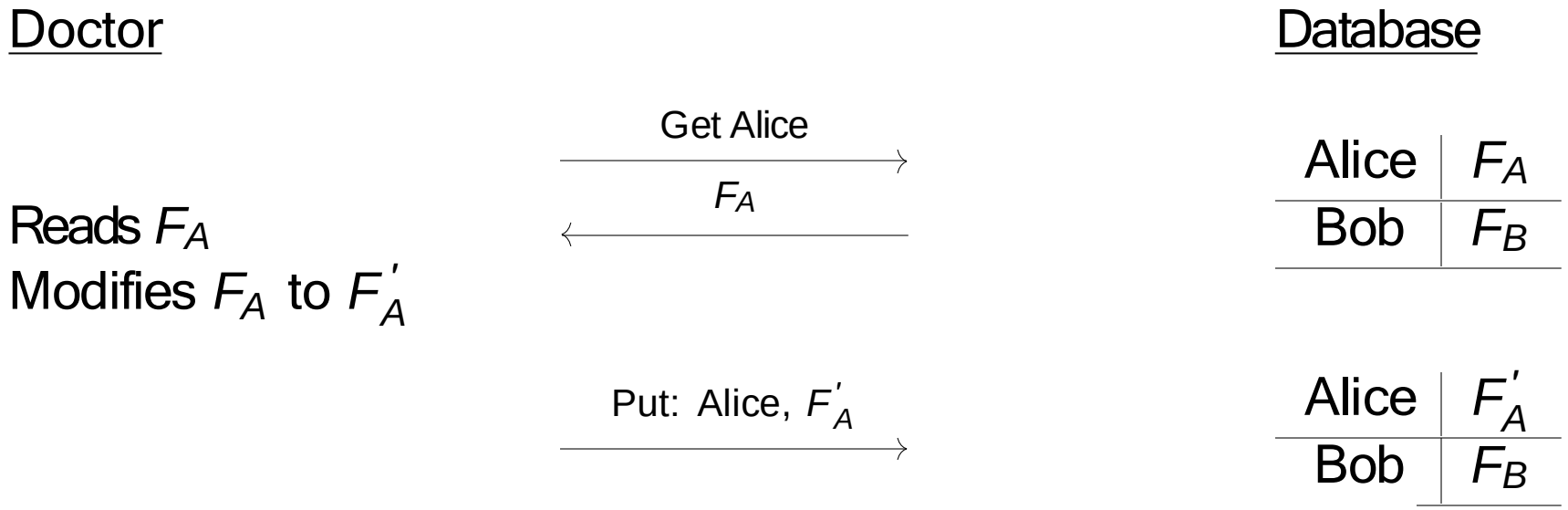
Data privacy: Ensure adversary does not see or obtain the data(message) M.

Data integrity and authenticity: Ensure M really originates with Alice and has not been modified in transit

Key Principles of Modern Cryptography

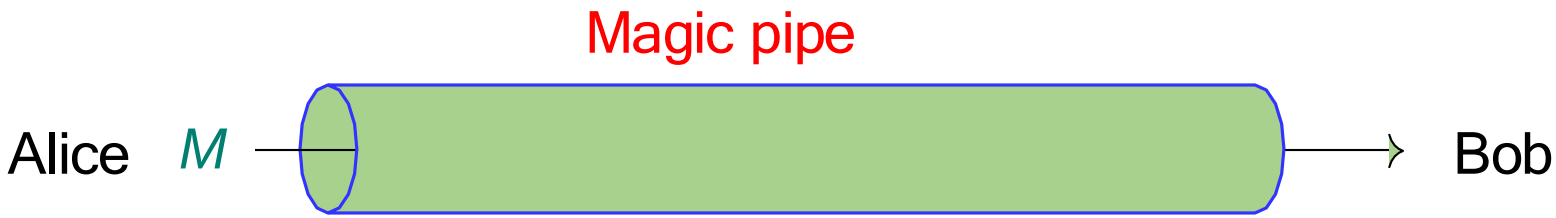
1. **Confidentiality** – Ensures that information is accessible only to authorized individuals
2. **Integrity** – Guarantees that data has not been altered in transit
3. **Authentication** – Verifies the identities of communicating parties
4. **Non-repudiation** – Prevents entities from denying their actions (e.g., digital signatures)

Example: Medical databases



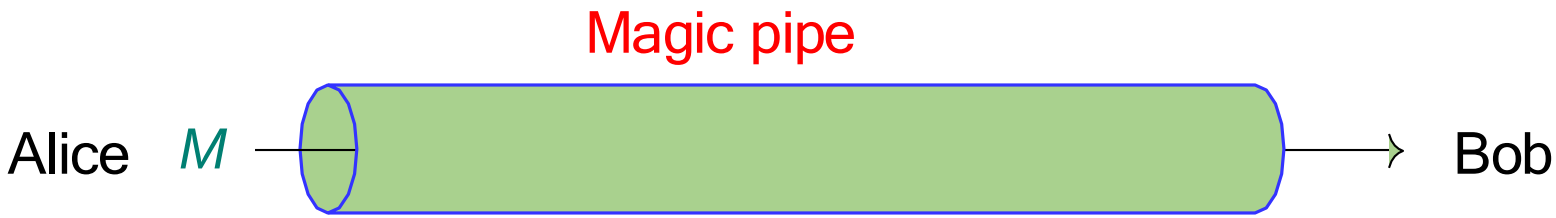
- Privacy: F_A, F_A' contain confidential information and we want to ensure the adversary does not obtain them
- Integrity and authenticity: Need to ensure
  - doctor is authorized to get Alice's file
  - F_A, F_A' are not modified in transit
  - F_A is really sent by database
  - F_A' is really sent by (authorized) doctor

Ideal World



Magic pipe: Cannot see inside or alter content. All our goals would be achieved!

Ideal World

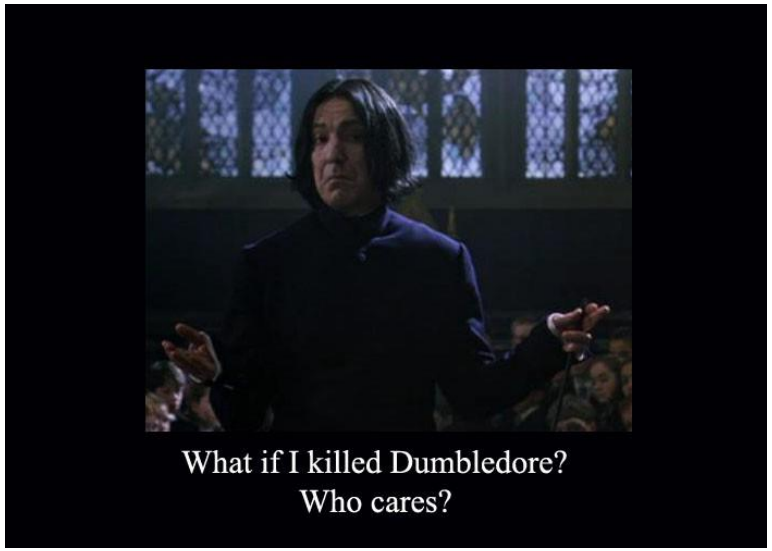


Adversary *A*

Magic pipe: Cannot see inside or alter content. All our goals would be achieved!

But magic pipe is only available on magic world and is in short supply.

Modern Application: Avoid Spoilers (ROT13)



Modern Application: Avoid Spoilers (ROT13)

Harry Potter

[ROT13 to avoid spoilers] V jnf fubpxrq naq ubeevsvrq jura Fancr xvyyrq Qhzoyrqber.

Like · Comment · 32 minutes ago ·

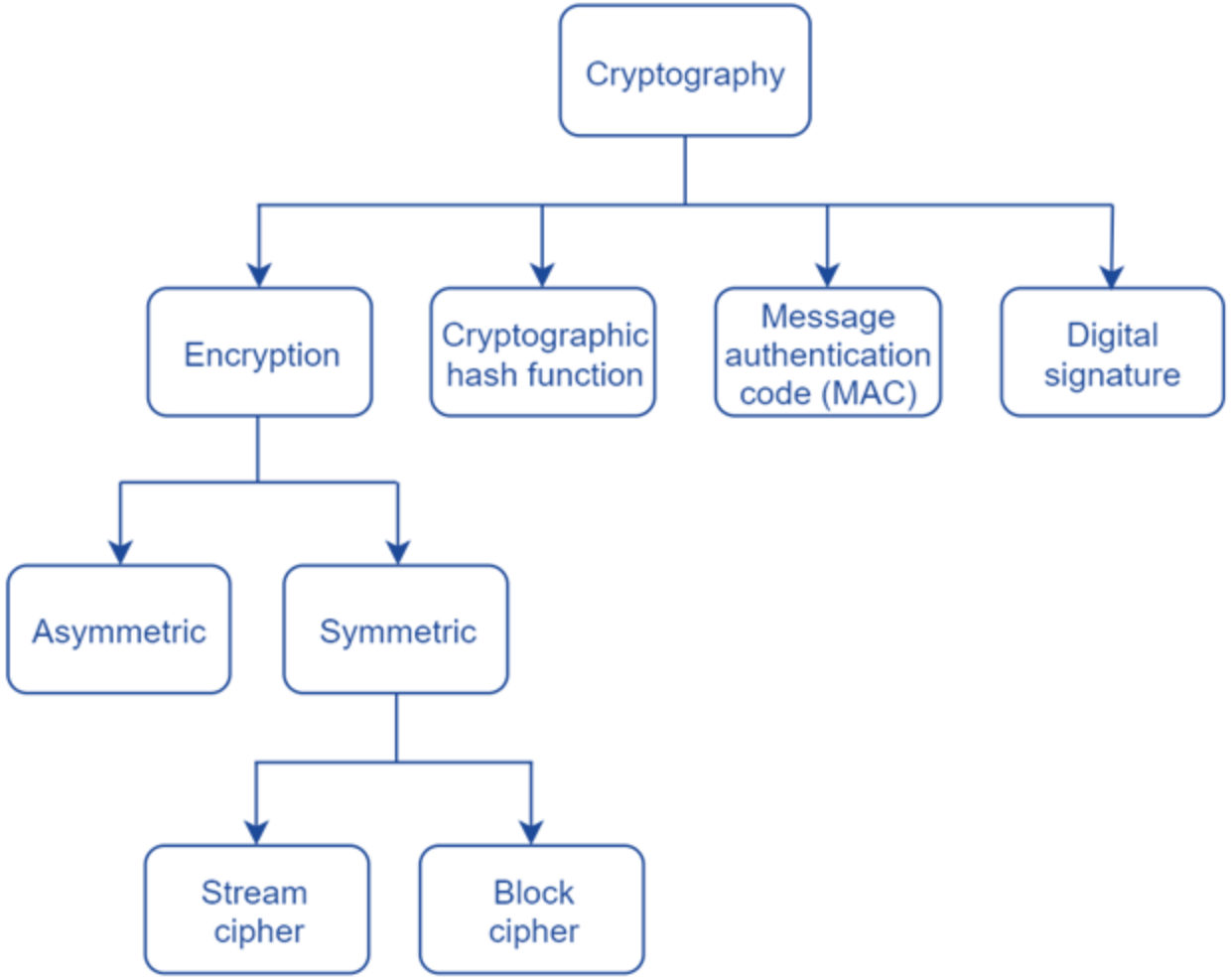
20 people like this.

Dumbledore I am dying to find out what will happen, but I will wait to decrypt until after I read the book.

15 minutes ago · Like · 23

Write a comment ...

Modern Cryptography

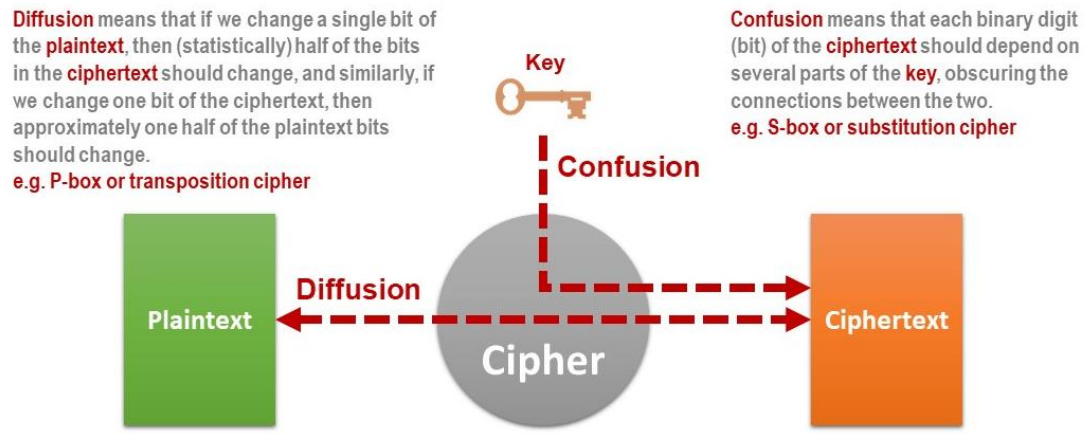


Encryption

- An encryption algorithm transforms a plaintext into a ciphertext using a key
- A secure encryption algorithm should have two **properties**:

- **Confusion** ensures that changing a single bit of an encryption key impacts most of the ciphertext bits
- Confusion property hides the relationship between a *ciphertext* and the *encryption* key

- **Diffusion** ensures that changing a single plaintext bit changes about half of the ciphertext bits and changing a single ciphertext bit changes about half of the plaintext bits
- Diffusion property hides the relationship between a *plaintext* and a *ciphertext*



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

1) Alice can easily find the relationship between a ciphertext and the key used to generate the ciphertext.

a. Diffusion
b. Confusion
c. Confusion and diffusion

2)Alice can easily find the relationship between a ciphertext and the key used to generate the ciphertext, and the relationship between a ciphertext and the plaintext.

a. Diffusion
b. Confusion
c. Confusion and diffusion

3)Alice can easily find the relationship between a ciphertext and the plaintext.

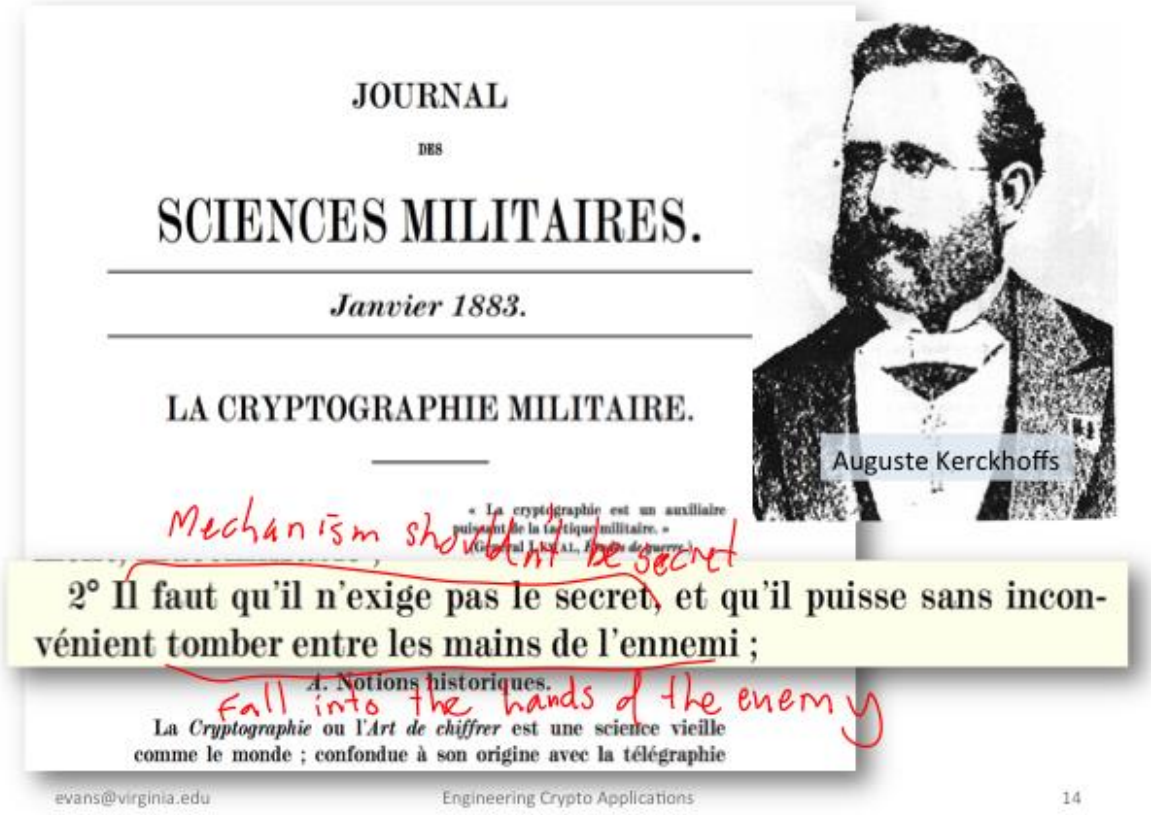
a. Diffusion
b. Confusion
c. Confusion and diffusion

Kerckhoffs's principle

- ***Kerckhoffs's principle*** states that the security of a cryptographic algorithm should depend on the secrecy of the key, not on the secrecy of the algorithm

"A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."

- Kerckhoffs's principle is applied to all contemporary encryption algorithms. The details of the most widely used encryption algorithms are publicly known and have been subject to extensive cryptanalysis.



Cryptographic Algorithms

- Cryptographic algorithms are mathematical procedures and processes used in the field of cryptography to secure and protect information

- A fundamental difference in cryptographic algorithms is the amount of data processed at a time
  - **Stream cipher** - takes one character and replaces it with another
  - **Block cipher** - manipulates an entire block of plaintext at one time
  - **Sponge function** - takes as input a string of any length and returns a string of any requested variable length

- Three categories of cryptographic algorithms
  - Symmetric cryptographic algorithms
  - Asymmetric cryptographic algorithms
  - Hash algorithms

Symmetric Cryptographic Algorithms

- **Symmetric cryptographic algorithms** use the same single key to encrypt and decrypt a document
  - Original cryptographic algorithms were symmetric
  - Also called *private key cryptography* (the key is kept private between sender and receiver)

- Common algorithms include:
  - *Data Encryption Standard (DES)*
  - *Triple Data Encryption Standard (3DES)*
  - *Advanced Encryption Standard (AES)*

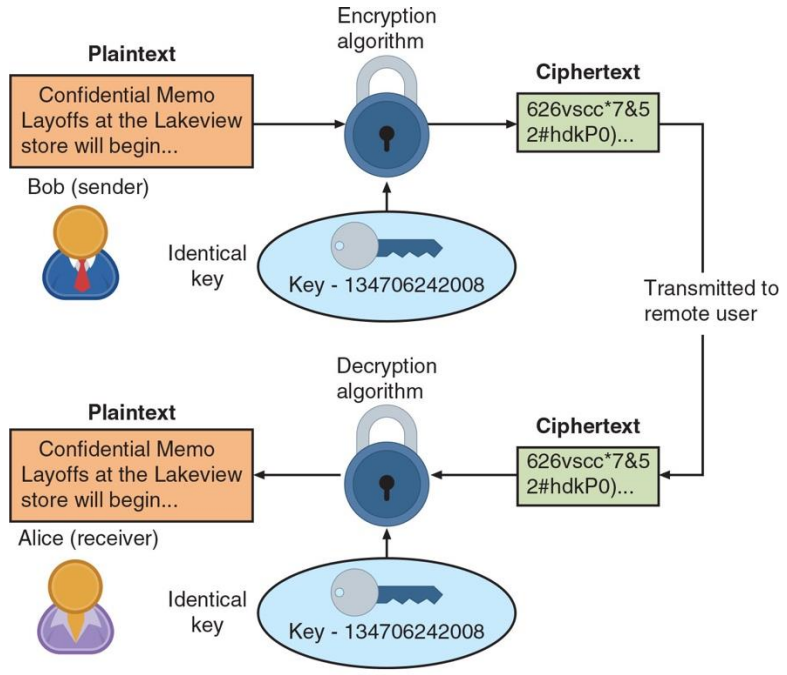
Symmetric Cryptographic Algorithms

One key for encryption and decryption
• Let K = secret key (think password), C = Ciphertext, P = Plaintext, E = Encrypt function, D = Decrypt function
• $C = E_k(P)$
• $P = D_k(C)$

Applications: Password protect a ZIP file –which uses AES

Strengths:
- Modified key K will result in garbage plaintext in decryption
- Fast!

Weaknesses:
- Those who know K can participate in communications (eavesdropping)
- Impersonation attack if attacker knows K
- Not good for authenticity



How large is a good key size?

- These days a good key size is: 256 binary bits (i.e., in the form of 1s and 0s)
- Why 256 bits?
- In binary arithmetic, with 256 bits we can have a total of 2^{256} possible choices for a key.
- Assume an attacker can test a billion keys per second (= 10^9 choices/second)
- 2^{256} keys will then : $2^{256}/10^9 \sim 10^{77}/10^9 = 10^{68}$ seconds ~ 10^{16} years!
- 10^{16} years is a lot of years!!
- So with a 256 bit key brute forcing through all the keys clearly takes a lot of time for now!

Asymmetric Cryptographic Algorithms

- Primary weakness of symmetric algorithms: distributing and maintaining a secure single key among multiple users distributed geographically poses challenges

➢ Asymmetric cryptographic algorithms use two mathematically related keys
  - ➢ Also known as ***public key cryptography***
  - ➢ Public key is available to everyone and freely distributed
  - ➢ Private key known only to the individual to whom it belongs

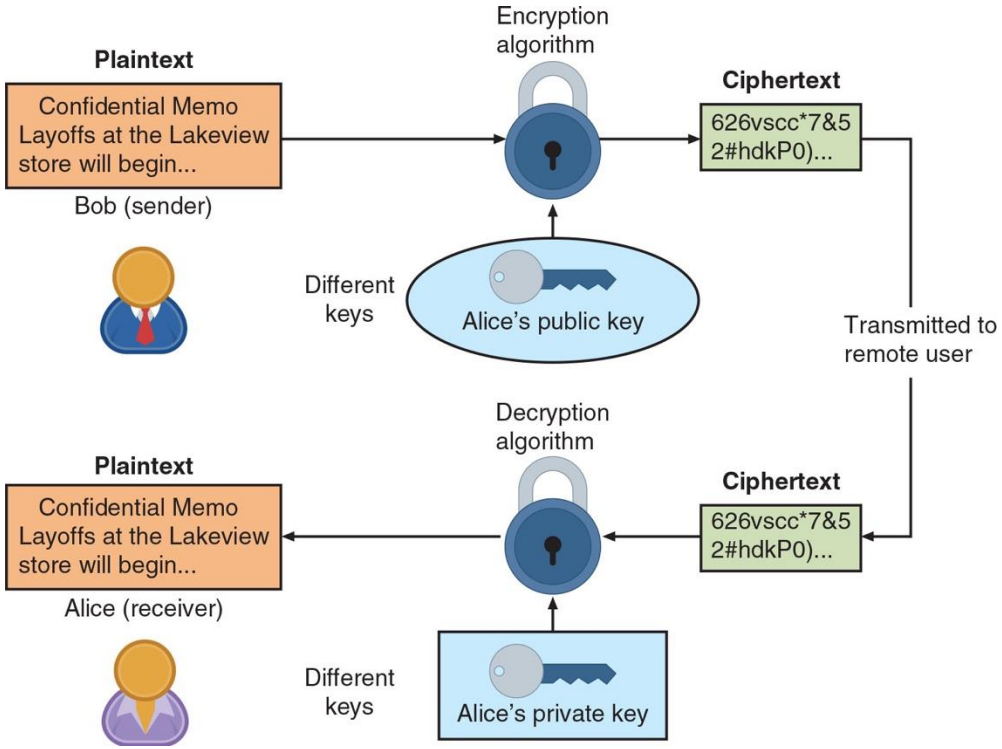
- Important principles
  - ➢ *Key pairs*
  - ➢ *Public key*
  - ➢ *Private key*
  - ➢ *Both directions* - keys can work in both directions

Asymmetric Cryptographic Algorithms

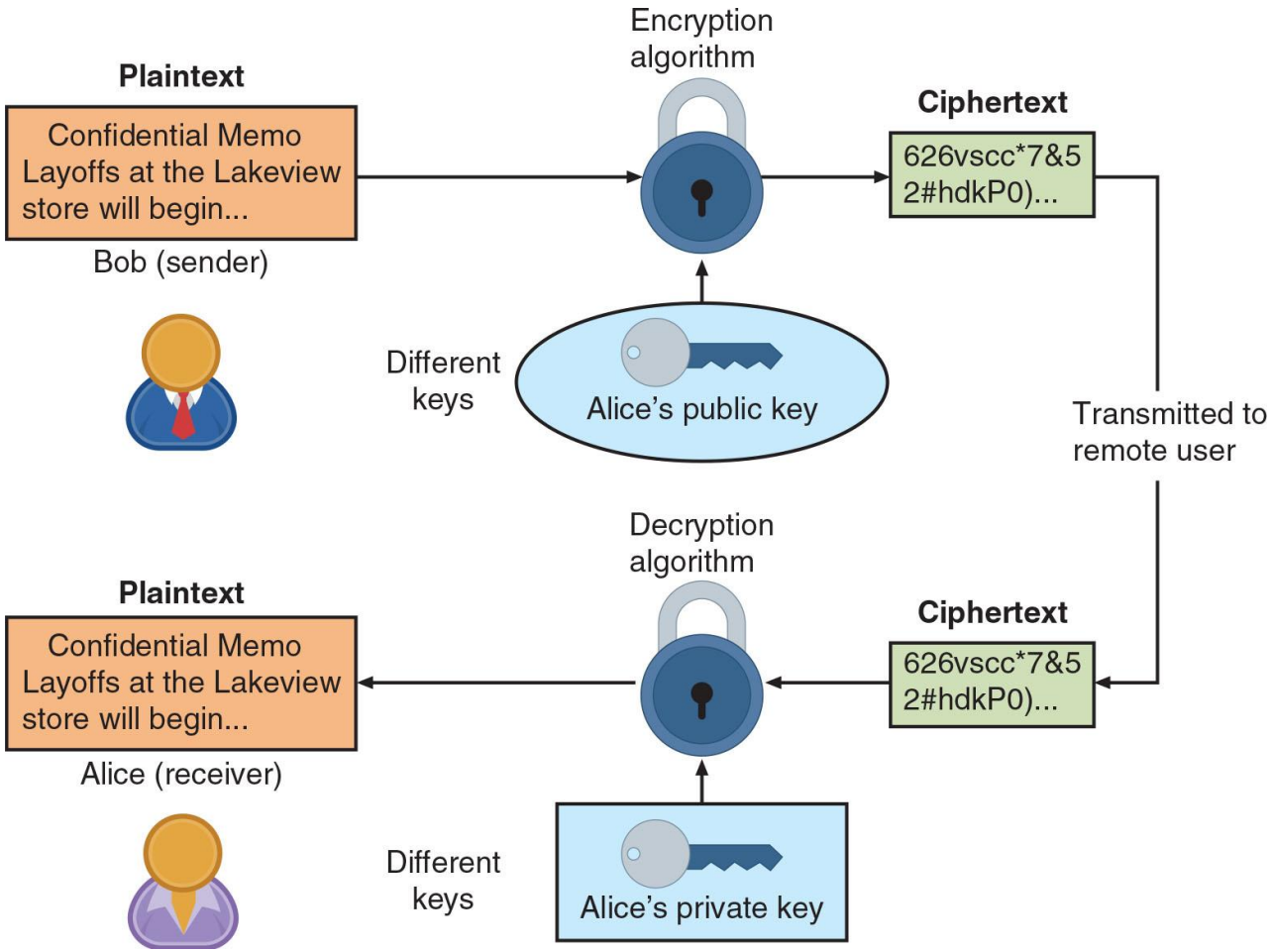
How it works:

- Alice and Bob agree on a public-key cryptosystem
- Alice and Bob have their own public and private keys
- Alice gives Bob her public key
- Bob encrypts the message with Alice's public key
  Alice decrypts the message with her private key

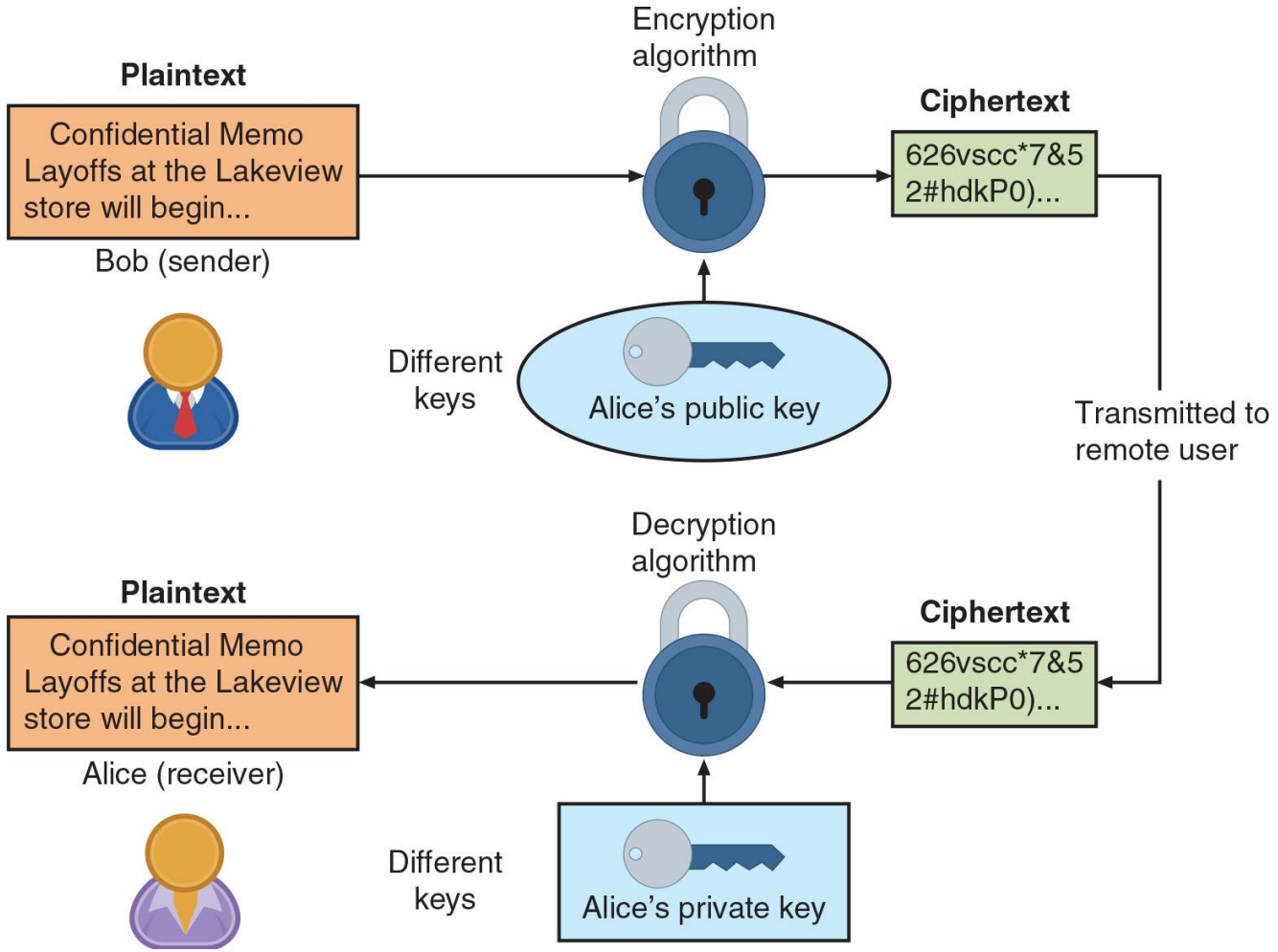
• Arguably the most popular algorithm: RSA
Walkthrough: http://www.di-mgt.com.au/rsa_alg.html



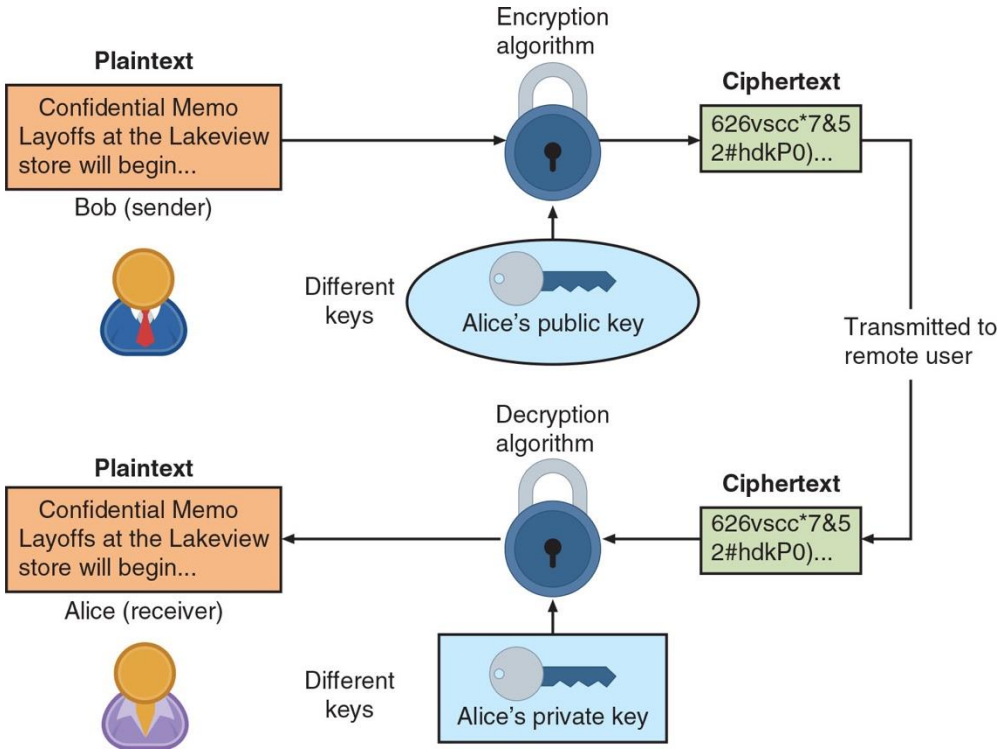
Is this providing confidentiality or authenticity?



Can you think how it can be used for getting **authenticity**?



Can you think how it can be used for getting authenticity?



Step 1 – Bob Encrypts with His Private Key:

Bob wants to prove to Alice that the message is genuinely from him. So, instead of using Alice's public key to encrypt the message, he encrypts it with **his private key**.

Step 2 – Alice Decrypts with Bob's Public Key:

When Alice receives the message, she can use **Bob's public key** (which is publicly available) to decrypt it. If the message decrypts correctly, Alice knows for certain that the message came from Bob, because only Bob's private key could have created it.

Some Asymmetric Cryptographic Algorithms

- DSA (Digital Signature Algorithm): Designed for digital signatures, often used in conjunction with other algorithms like SHA for hashing.

- RSA (Rivest-Shamir-Adleman): A widely used asymmetric algorithm for encryption and digital signatures. RSA is widely used in secure data transmission, such as in SSL/TLS for securing web traffic.

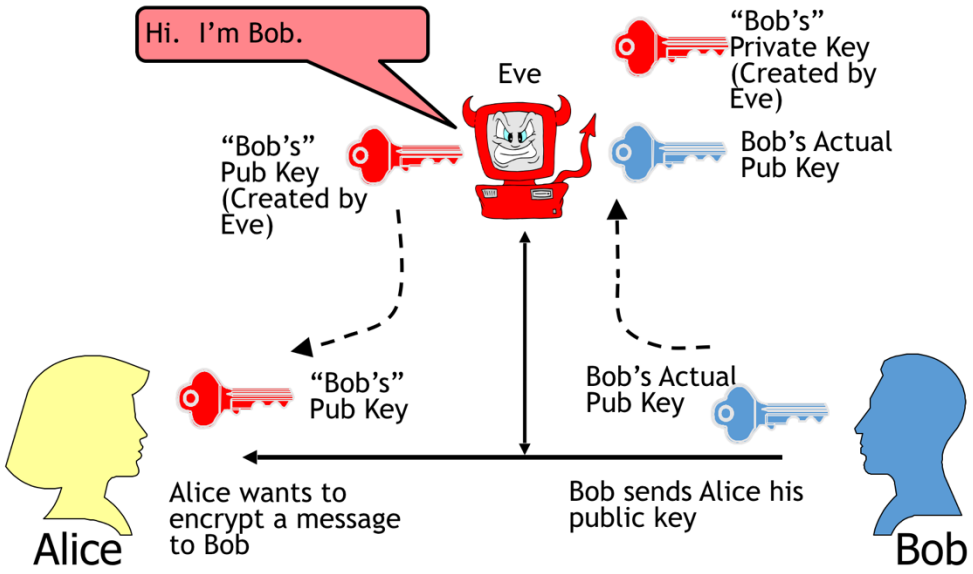
- Elliptic Curve Cryptography (ECC): A family of asymmetric algorithms based on the algebraic structure of elliptic curves, known for providing strong security with smaller key sizes compared to RSA.

Asymmetric Cryptographic Algorithms

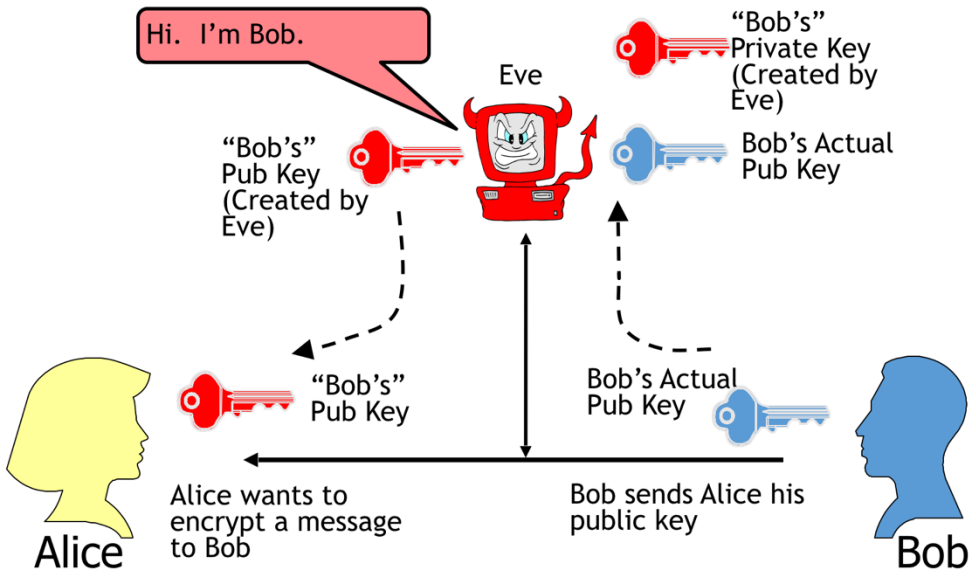
Strengths:

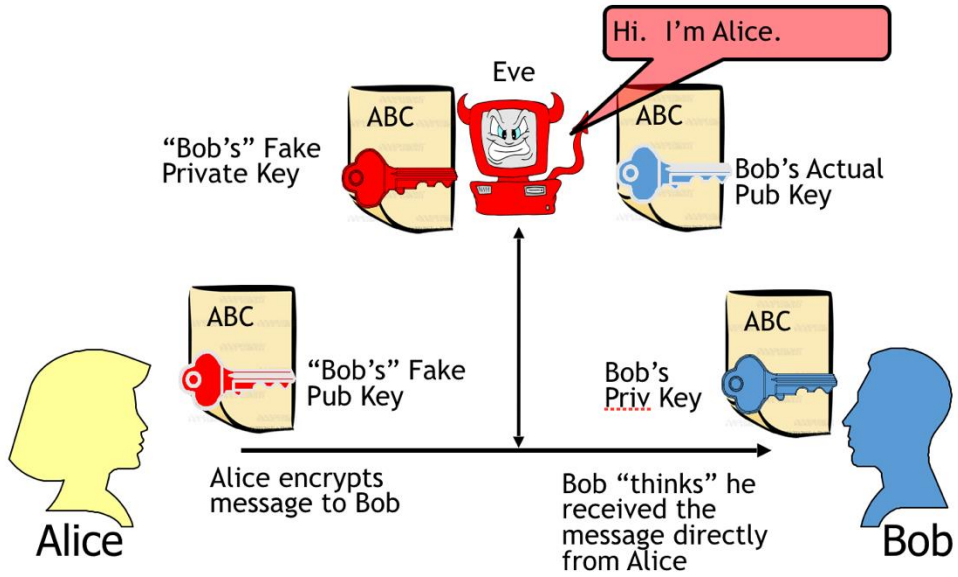
- **Key distribution:** Public key can be distributed in any way possible
- **Confidentiality**: only the holder of the private key can decrypt the message
- **Non-Repudiation**: sender cannot deny sending a message, as the digital signature can prove the origin of the message

- Weakness:
  - Key management is challenging in large-scale system
  - Man-in-the-Middle (MitM) attack
  - Depends on cryptographic algorithm; if algorithm is cracked, it can jeopardize security



Asymmetric Cryptographic Algorithms





Source: techgirls.ece.vt

Problem

- How do we know we are really talking to the right party?
- Anyone can send you a public key to use.
- A digital signature only ties a message to a private key, not to a person.
- We need a way to bind a public/private key pair to a specific individual.

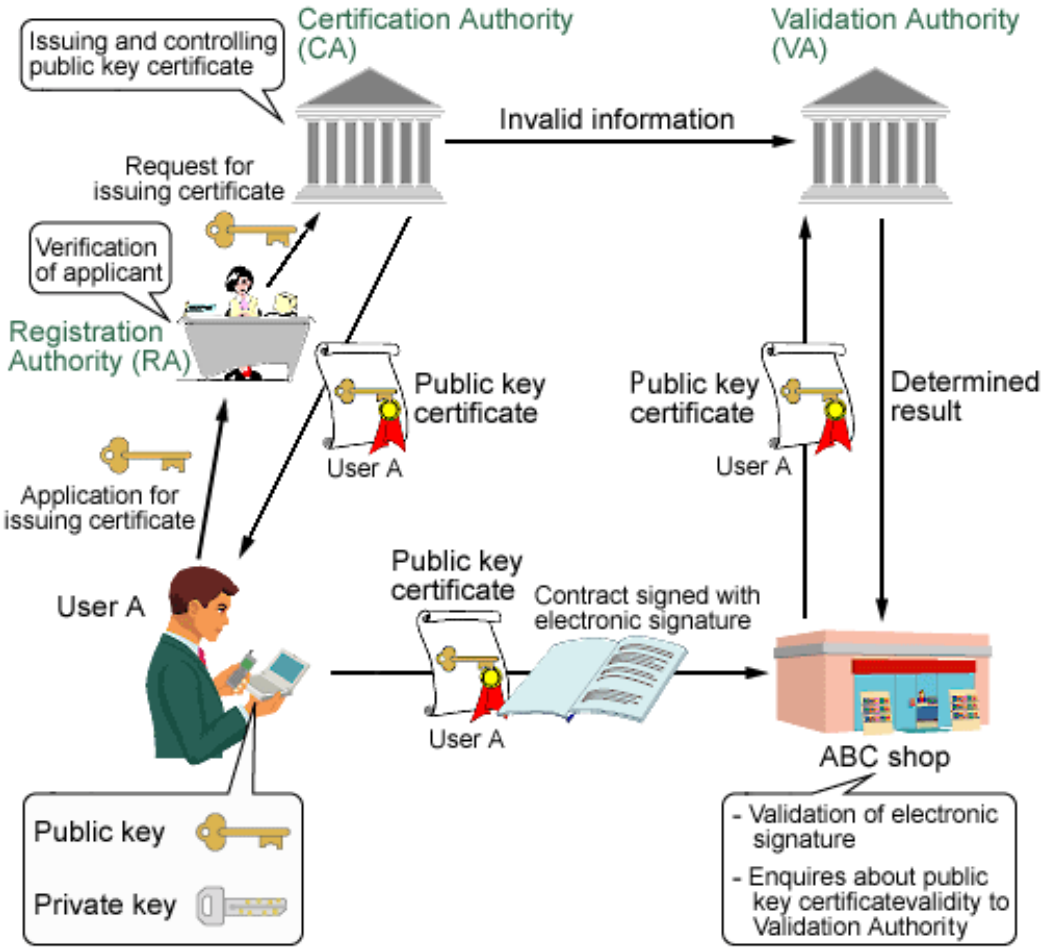
Solution: PKI

- Public Key Infrastructure (PKI) is a set of tools and procedures that help secure electronic communications and transactions. PKI uses encryption to protect data, verify identities, and ensure the integrity of digital message

Components :

- **Certificate authority (CA):** CA is a trusted entity that issues, stores, and signs the digital certificate. The CA signs the digital certificate with their own private key and then publishes the public key that can be accessed upon request

- **Registration authority (RA):** RA verifies the identity of the user or device requesting the digital certificate. This can be a third party, or the CA can also act as the RA



Source: https://software-engineer-tips-and-tricks.blogspot.com/2012/09/what-is-pki.html

PKI

- **Validation Autority (VA): C**onfirms whether a digital certificate is valid at a given point in time
  - Revocation Notices (e.g., a certificate has been revoked for security reasons)
  - Certificate Status Updates (e.g., a certificate's status changed from valid to suspended)
  - Inaccurate Data discovered post-issuance (e.g., incorrect user details invalidating the certificate)

- **Certificate database:** This database stores the digital certificate and its metadata, which includes how long the certificate is valid

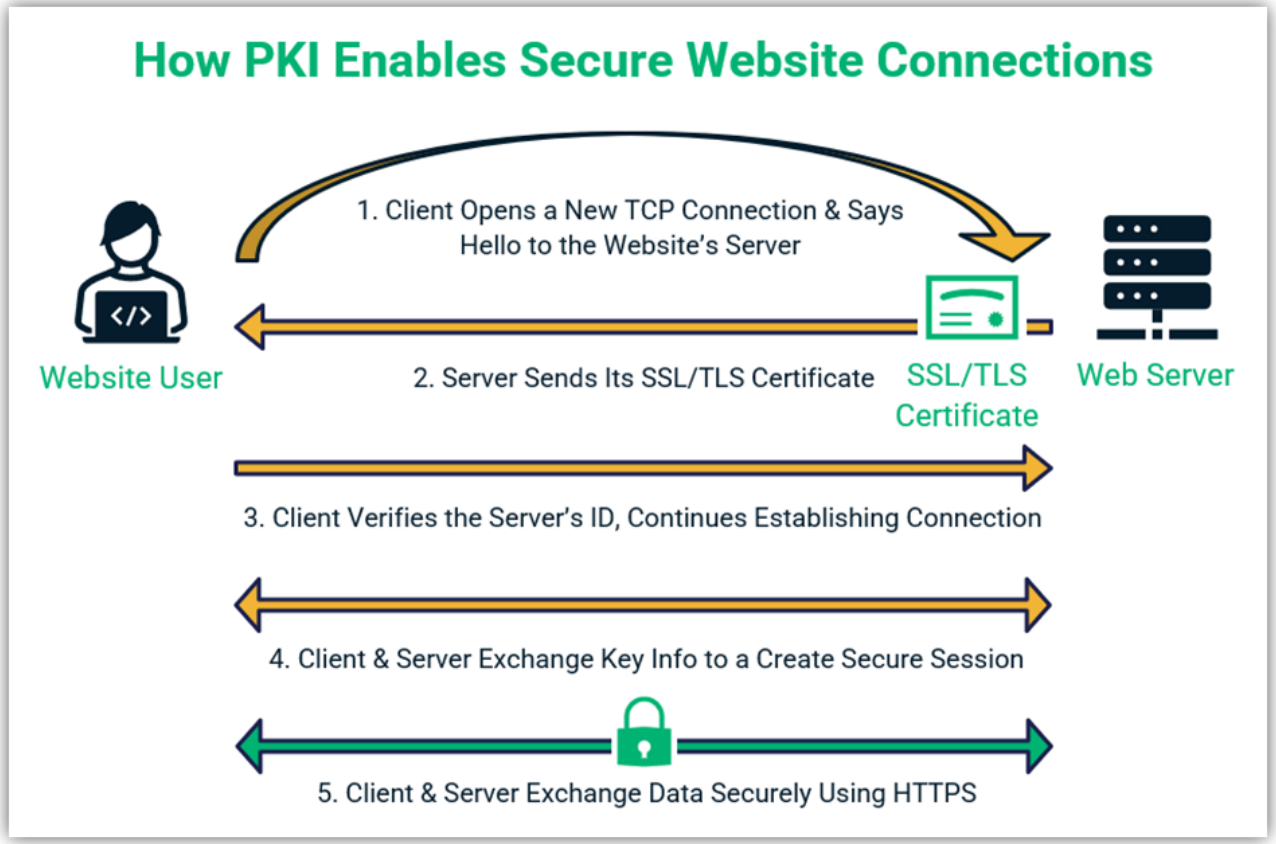
- **Central directory:** This is the secure location where the cryptographic keys are indexed and stored

- **Certificate management system:** This is the system for managing the delivery of certificates as well as access to them

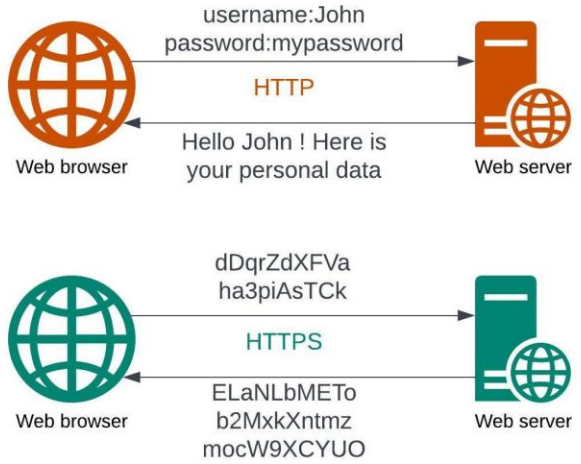
- **Certificate policy:** This policy outlines the procedures of the PKI. It can be used by outsiders to determine the PKI's trustworthiness

Why is PKI used?

- One of the most common uses of PKI is the TLS/SSL (transport layer security/secure socket layer), which secures encrypted HTTP (hypertext transfer protocol) communications

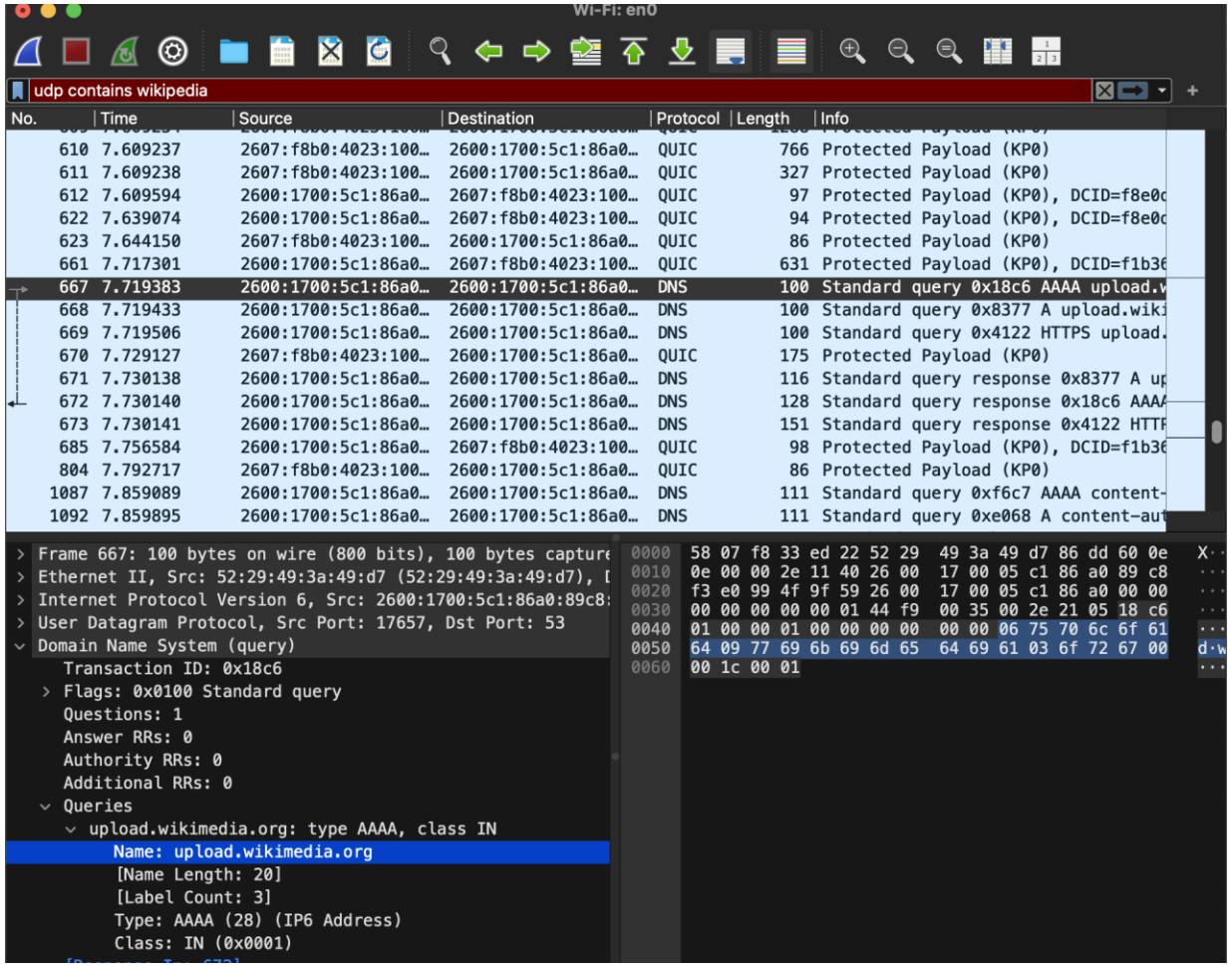


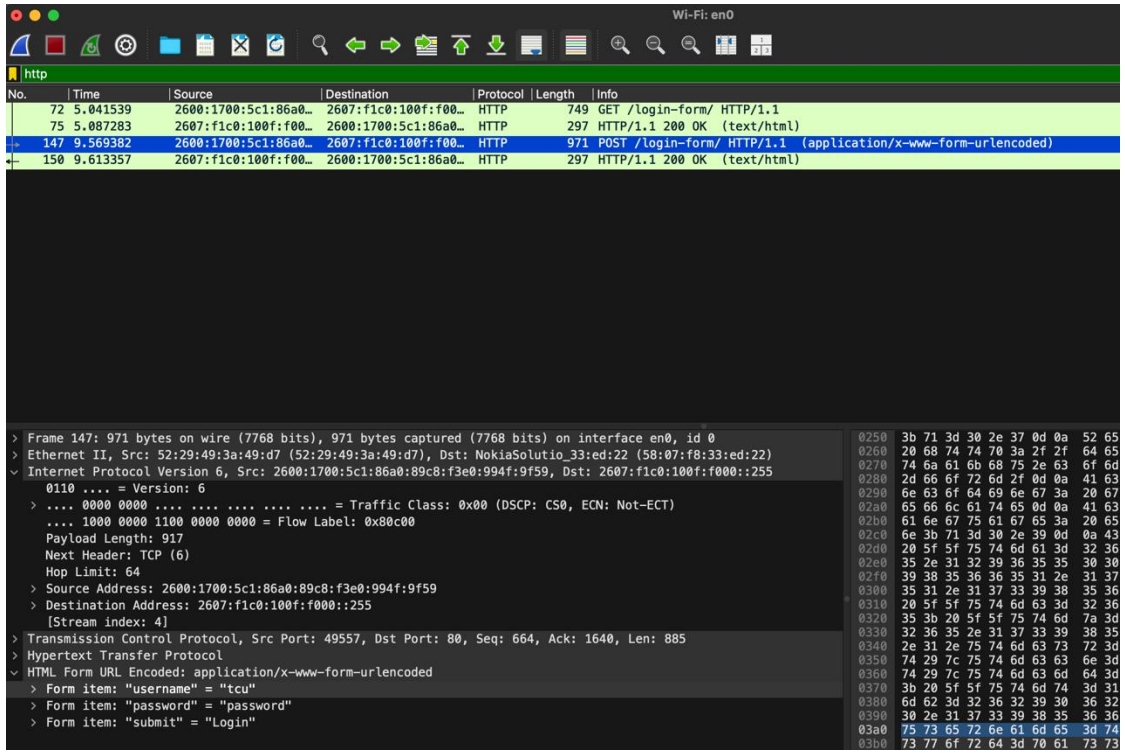
Source: https://sectigostore.com/blog/pki-101-all-the-pki-basics-you-need-to-know-in-180-seconds/



Source: https://medium.com/@jayampathiadhikari/ssl-tls-simplified-c3c1f08051b2

Read more





http://demo.amitjakhu.com/login-form/

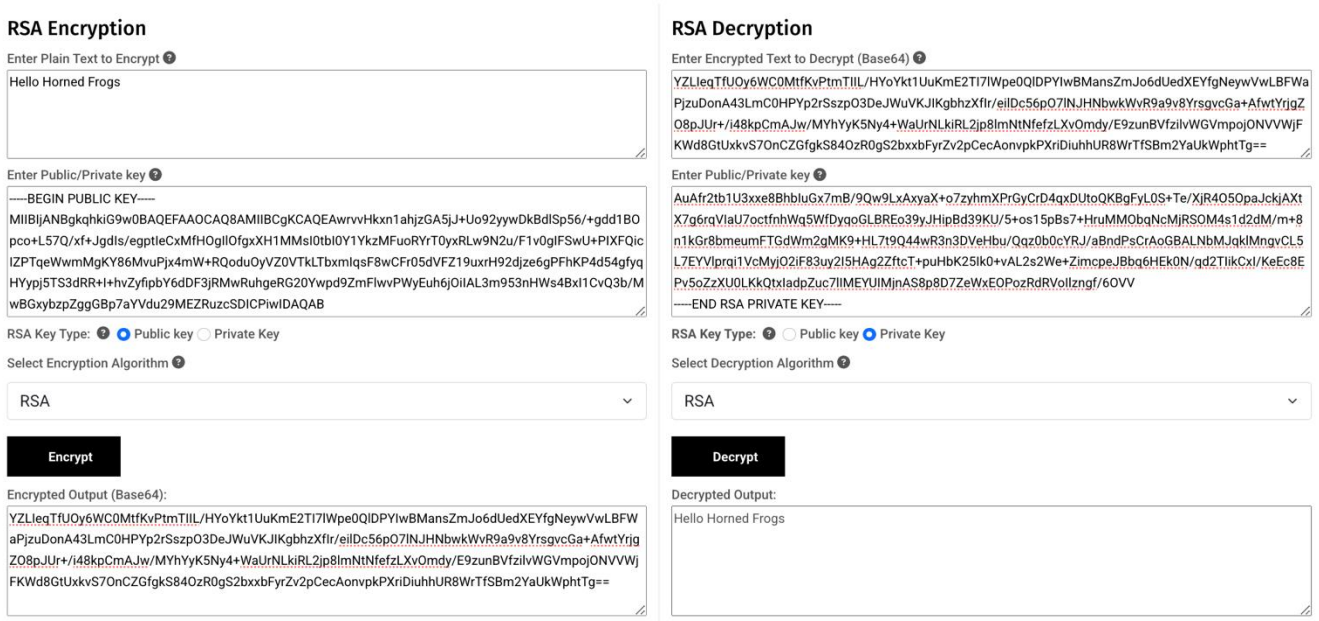
Why is PKI used?

- Email encryption and authentication of the sender
- Signing documents and software
- Using database servers to secure internal communications
- Securing web communications, such as e-commerce
- Authentication and encryption of documents
- Securing local networks and smart card authentication
- Encrypting and decrypting files

Try: RSA Encryption and Decryption Online

Create your keys: https://cryptotools.net/rsagen

Test : https://www.devglan.com/online-tools/rsa-encryption-decryption#google_vignette



create your own private key and a certificate

https://www.ibm.com/docs/en/license-metric-tool?topic=certificate-step-1-creating-private-keys-certificates

Imagine you receive an email from your bank saying that your account details need to be updated. The email includes a PDF attachment that looks like an official form.

"How can you verify that this document truly came from your bank and wasn’t altered by someone else?"

Digital Signature

- A digital signature is an electronic, encrypted, stamp of authentication on digital information such as email messages or electronic documents
- A signature confirms that the information originated from the signer and has not been altered
- Uses cryptography to confirm the origin, authenticity, and integrity of a digital asset (e.g., a document, an email, or a piece of software) to show that the signer created it

Source: https://codesigningstore.com/what-is-a-digital-signature-in-cryptography

Are electronic and digital signature same?

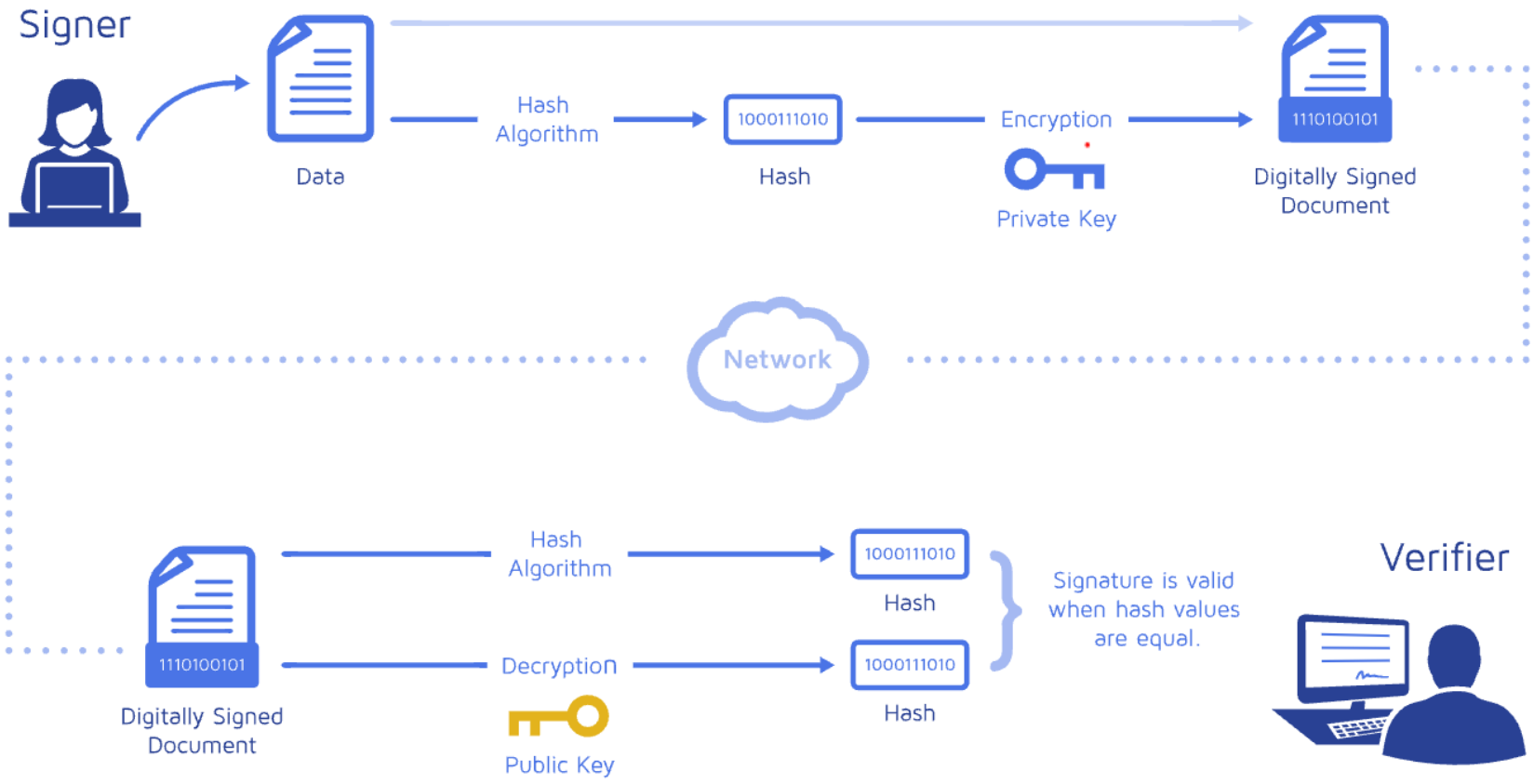
	Electronic Signature	Digital Signature
Main Purpose	Verify a document and identify the source and author(s).	Secure a document so that it cannot be modified by people without authorization.
Regulation	Not regulated.	Authorized and regulated by certification authorities.
Security	External security measures can be utilized (i.e. encrypted transmission) however tampering with the document is feasible.	Embedded security features within the digital certificate prevents tampering.
Types	Ranges from approval clicks to inserting digitized signatures within documents.	Digital certificates mainly based on document processing platforms (i.e. Microsoft, Adobe).
Verification	Owner of the document or possible tampering cannot directly be verified.	Document can be verified for tampering and digital certificate can be used to track the document author.

all digital signatures are electronic signatures, but not all electronic signatures are digital ones

Digital Signature

- A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents
- Based on public key cryptography
- To provide high levels of security, digital signatures use a standard, accepted format called Public Key Infrastructure (PKI)
- A digital signature can:
  - Verify the sender
  - Prevent the sender from disowning the message
  - Proves the authenticity and integrity of a document, just like a notarized signature
  - Identity validation: Performed by a trusted certificate authority (CA) through the release of a digital certificate
  - Highly secure: **hashing and encryption**
  - Uses: Secure documents, codes, emails, and files

Digital Signature



Watch: https://www.youtube.com/watch?v=JR4_RBb8A9Q&ab_channel=Lisk

Source:docusign

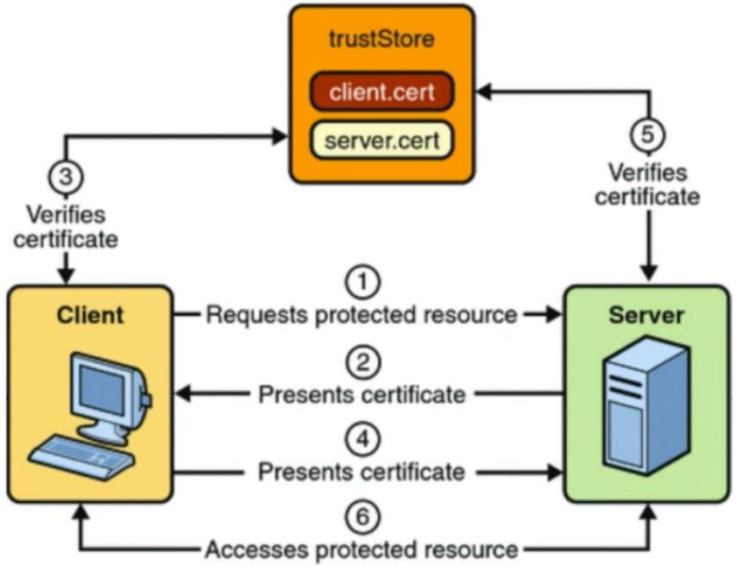
You are about to download a new software application from a website. When you start the installation, a warning appears:

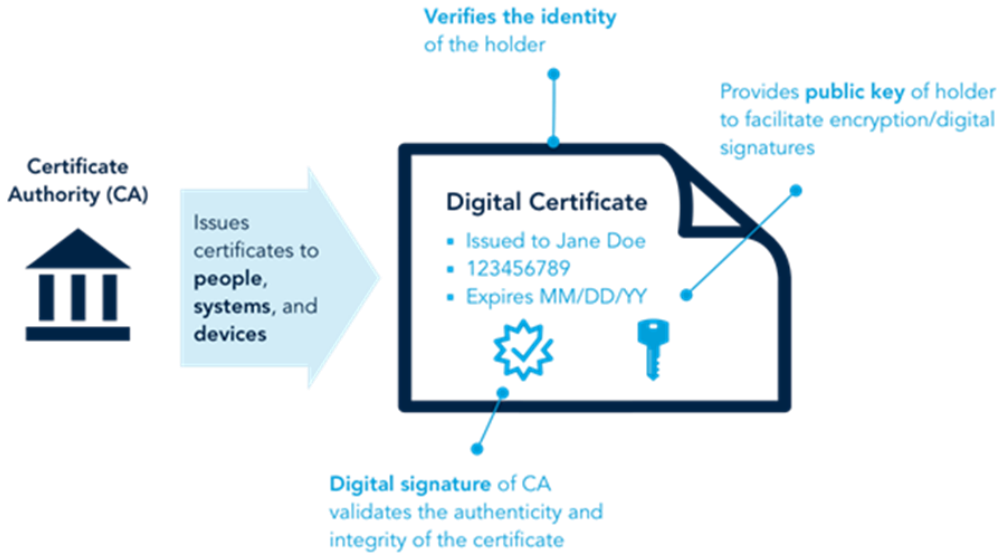
"The publisher of this software could not be verified. Do you want to continue?"

How can you ensure that the software is from a legitimate source and hasn’t been tampered with?

Digital Certificate or public key certificate

•A public key certificate, is an electronic document which proves the ownership of a public key
•A certificate is digitally signed by a certificate issuer, known as a Certificate Authority (CA), that has verified a certificate's content
•Used to provide legitimacy to websites, devices, web servers, signatures, code, software, email and more





Watch: CA certificates

Certificate Viewer: www.paypal.com

General | Details

Issued To

Common Name (CN)	www.paypal.com
Organization (O)	PayPal, Inc.
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	DigiCert SHA2 Extended Validation Server CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

Validity Period

Issued On	Thursday, July 20, 2023 at 8:00:00 PM
Expires On	Tuesday, August 20, 2024 at 7:59:59 PM

Fingerprints

SHA-256 Fingerprint	FF EE C8 F0 6C AD 2E B2 20 0C DA 81 96 26 E8 B3 3B 56 69 A1 9F 77 56 94 AB 3F B2 F8 49 32 16 97
SHA-1 Fingerprint	8F 77 81 3F E2 25 14 80 8B DA F8 F9 D6 E7 B2 4B 21 08 B1 4D

Certificate Viewer: www.robinchataut.com

General | Details

Issued To

Common Name (CN)	www.robinchataut.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	R11
Organization (O)	Let's Encrypt
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Friday, December 27, 2024 at 8:00:27 PM
Expires On	Thursday, March 27, 2025 at 9:00:26 PM

SHA-256 Fingerprints

Certificate	3c44f6d058b672d6f2d4de80c2fb5aecc4aec80cceb0dfb4c67743fc9cbff1ca
Public Key	0d073b9b73b4b0ee16e3f580e521c7b6c7aa20a48a764a758e1b31dff5043d90

If you run a website, you can obtain an SSL certificate from a CA:

1.Free SSL Certificates – Let’s Encrypt (good for most websites)., Internet Security Research Group (ISRG) runs Let's Encrypt, a free, automated, and open certificate authority (CA)

- Many web hosting companies (like Bluehost, SiteGround, and Cloudflare) automatically issue Let's Encrypt certificates for their customers
- If you enabled SSL in your hosting panel, it may have used Let's Encrypt
- Only Domain Validation (DV) – Confirms ownership of a domain, but not the identity of the business

2.Paid SSL Certificates – IdenTrust, DigiCert, Sectigo, GlobalSign (better for businesses)

3.Self-Signed Certificates – Not trusted by browsers (used for internal networks)

How to generate your own / self-signed SSL certificates for use with an On-Premise deployments

Market share trends for SSL certificate authorities

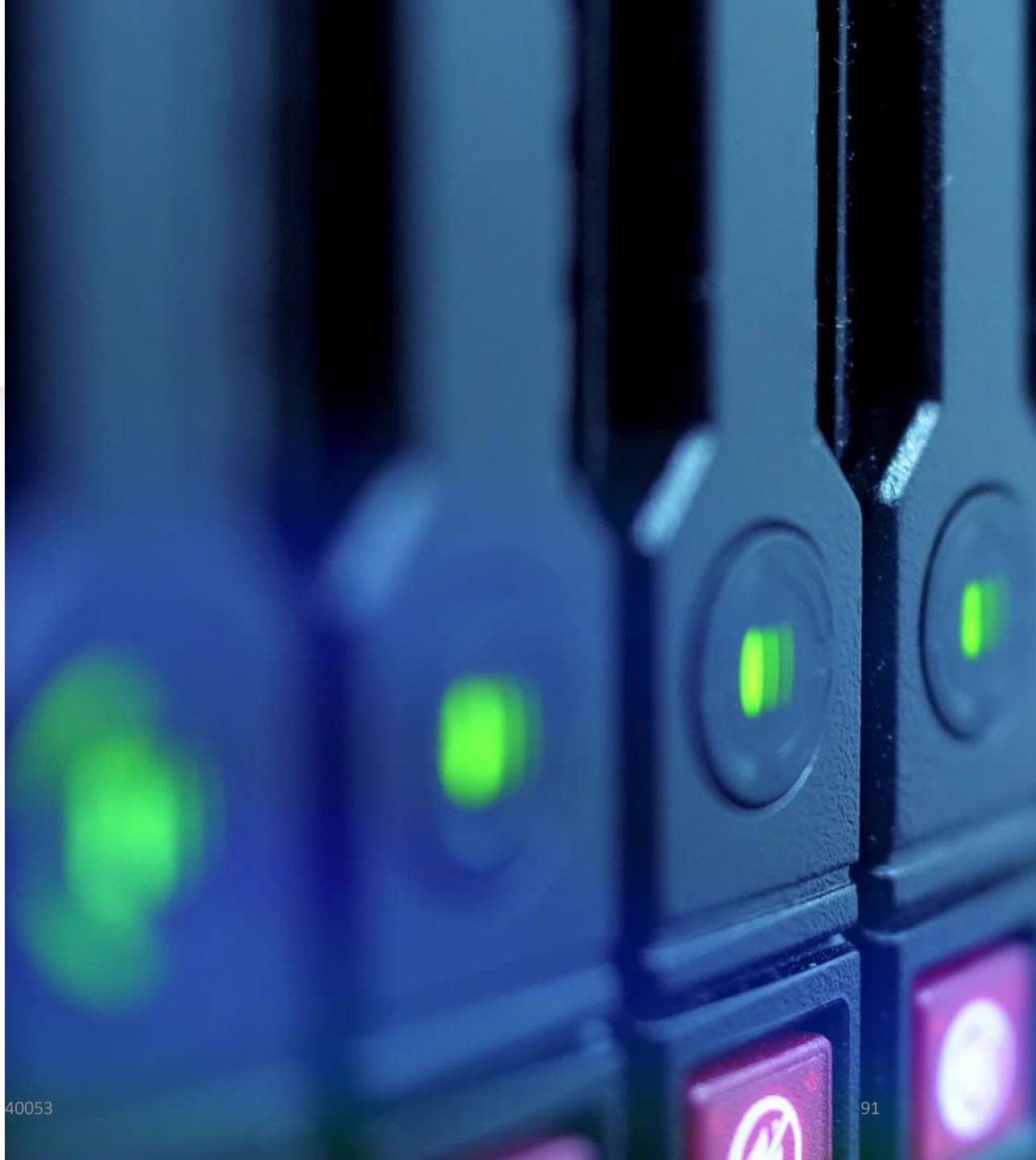
This report shows the market share trends for the top SSL certificate authorities since February 2024.

	2024 1 Feb	2024 1 Mar	2024 1 Apr	2024 1 May	2024 1 Jun	2024 1 Jul	2024 1 Aug	2024 1 Sep	2024 1 Oct	2024 1 Nov	2024 1 Dec	2025 1 Jan	2025 1 Feb	2025 18 Feb
Let’s Encrypt	7.9%	12.6%	32.7%	48.2%	52.6%	53.5%	56.3%	59.1%	63.1%	64.7%	62.9%	63.3%	63.6%	63.7%
GlobalSign	10.6%	10.9%	11.0%	11.2%	11.5%	12.2%	14.0%	15.4%	16.4%	19.0%	21.5%	22.0%	22.2%	22.4%
Sectigo	11.6%	10.9%	10.0%	9.0%	8.0%	7.6%	7.3%	6.9%	6.7%	6.6%	6.5%	6.3%	6.1%	5.9%
GoDaddy Group	4.8%	4.7%	4.7%	4.6%	4.5%	4.5%	4.4%	4.4%	4.3%	4.3%	4.3%	4.2%	4.1%	4.1%
DigiCert Group	7.7%	7.3%	6.9%	6.1%	5.7%	5.5%	5.3%	5.2%	4.9%	4.6%	4.3%	3.7%	3.5%	3.4%
Actalis	0.5%	0.5%	0.5%	0.5%	0.5%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%
Certum	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%
Secom Trust	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%
IdenTrust	57.2%	53.5%	34.9%	20.8%	17.4%	16.5%	12.5%	8.7%	4.2%	0.3%	0.2%	0.1%	0.1%	0.1%
Entrust	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
SSL.com	0.1%	0.1%	0.1%	<0.1%	0.1%	0.1%	0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	0.1%	0.1%

Source

Let’s say you sign up on a new website using your email id and trusted password combination.

How do you think it will be stored on their servers?

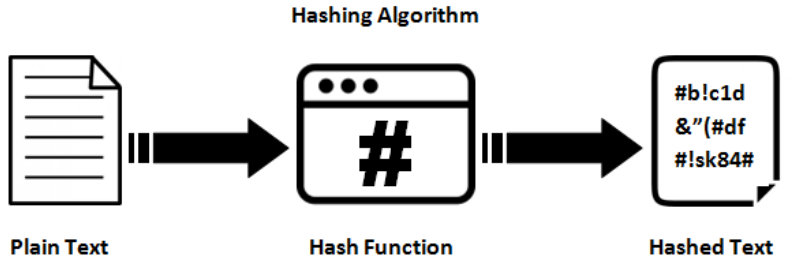


Hash Algorithms

- Hash algorithm creates a unique “digital fingerprint” of a set of data and is commonly called *hashing*
  - ➢ This fingerprint, called a digest (sometimes called a *message digest* or *hash*), represents the contents

- Hashing is intended to be one way in that its digest cannot be reversed to reveal the original set of data

- Secure hashing algorithm characteristics:
  - *Fixed size* - short and long data sets have the same size hash
  - *Unique* - two different data sets cannot produce the same hash
  - *Original* - data set cannot be created to have a predefined hash
  - *Secure* - resulting hash cannot be reversed to determine original plaintext



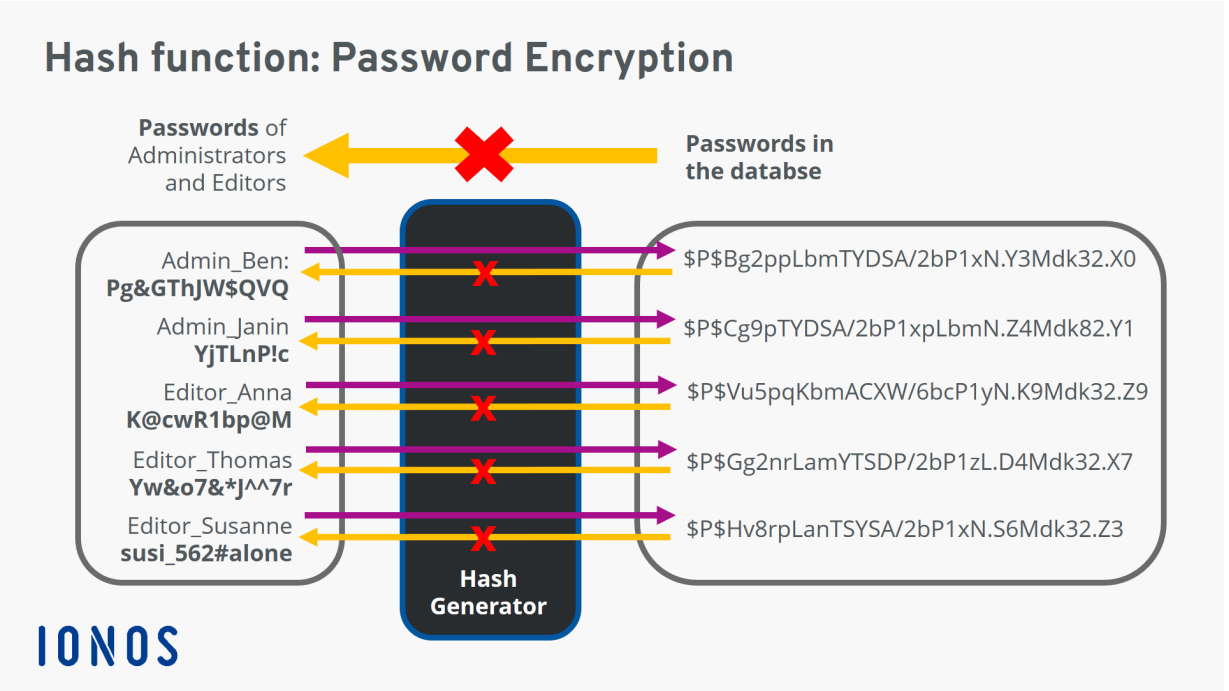
Hash Algorithms

Cryptographic hash function is a function which maps a variable length string, or ***message***, to a fixed-size value, ***message digest***

Image Name	Torrent	Version	Size	SHA256Sum
Kali Linux 64-Bit (Installer)	Torrent	2020.2	3.6G	ae9a3b6a1e016cd464ca31ef5055506cecfc55a10f61bf1acb8313eddbe12ad7
Kali Linux 64-Bit (Live)	Torrent	2020.2	2.9G	e90e0cfb4bc8fc640219dba66c9fe4308c9502164e432c47a30af50ce9cb3ba2
Kali Linux 64-Bit (NetInstaller)	Torrent	2020.2	420M	def160159e12ff52fb5f4991240bd760500d7cd5ee38601a8bf35809a20f9450

Source: Kali Linux

Verifying downloads with digests



https://www.youtube.com/watch?v=b4b8ktEV4Bg&ab_channel=Computerphile
Source: https://stytch.com/blog/what-is-password-hashing/

Properties of cryptographic hash functions

A cryptographic hash function has three properties:

- **Collision resistance**: Finding two different inputs to a cryptographic hash function that have the same digest is computationally infeasible

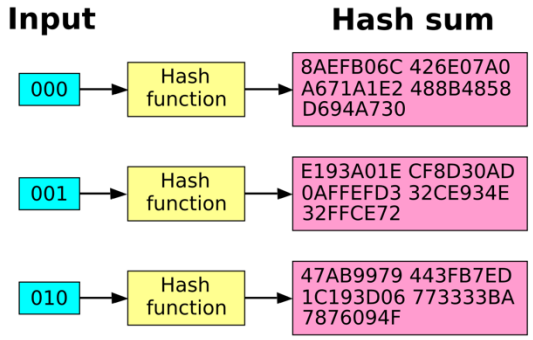
- **Preimage resistance**: Finding the original message from a message digest is computationally infeasible; this property is also called the **one-way property**

- **Second preimage resistance**: Finding a second input that has the same digest as any other specified input is computationally infeasible

Properties of cryptographic hash functions

Avalanche effect is a desirable property of a cryptographic hash function

- a small change to a message should result in a digest that is significantly different and uncorrelated with the digest of the original message
- **Test**: enter two different input and see the hashes, change algorithms and see how it changes hash



https://emn178.github.io/online-tools/md2.html

https://www.codepunker.com/tools/string-converter

Hash Algorithms

- *Message Digest (MD)* is one of the earliest family of hash algorithms
  - Most well-known of the MD hash algorithms is MD5
  - Maps a message to a 128-bit hash
  - Some security experts recommend using a more secure hash algorithm

- *Secure Hash Algorithm (SHA)*
  - *SHA-1 is rarely used (160 bits)*
  - SHA-2 is currently considered to be a secure hash (up to 512 bits)
  - SHA-3 was announced as a new standard in 2015 and may be suitable for low-power devices (up to 512 bits)

https://www.youtube.com/watch?v=Gw_GKnalpYY&ab_channel=ProfessorMesser

Identify the hash

Install hashID/hash-identifier

- hashID is a tool written in Python 3 which supports the identification of over 220 unique hash types using regular expressions

```
$ pip install hashid
$ pip install --upgrade hashid
$ pip uninstall hashid
```

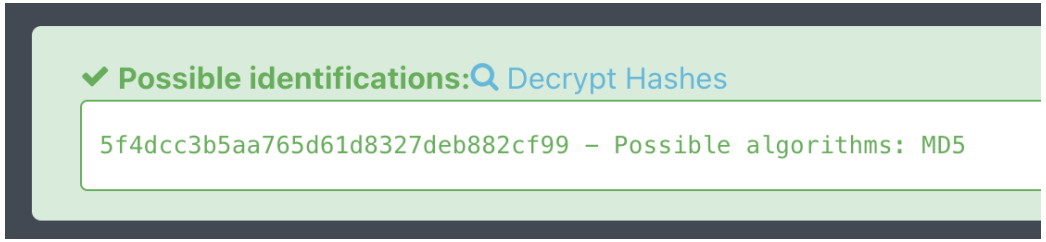
Or you can install by cloning the repository:

```
$ sudo apt-get install python3 git
$ git clone https://github.com/psypanda/hashid.git
$ cd hashid
$ sudo install -g 0 -o 0 -m 0644 doc/man/hashid.7
/usr/share/man/man7/
$ sudo gzip /usr/share/man/man7/hashid.7
```

Latest Release https://github.com/psypanda/hashID/releases

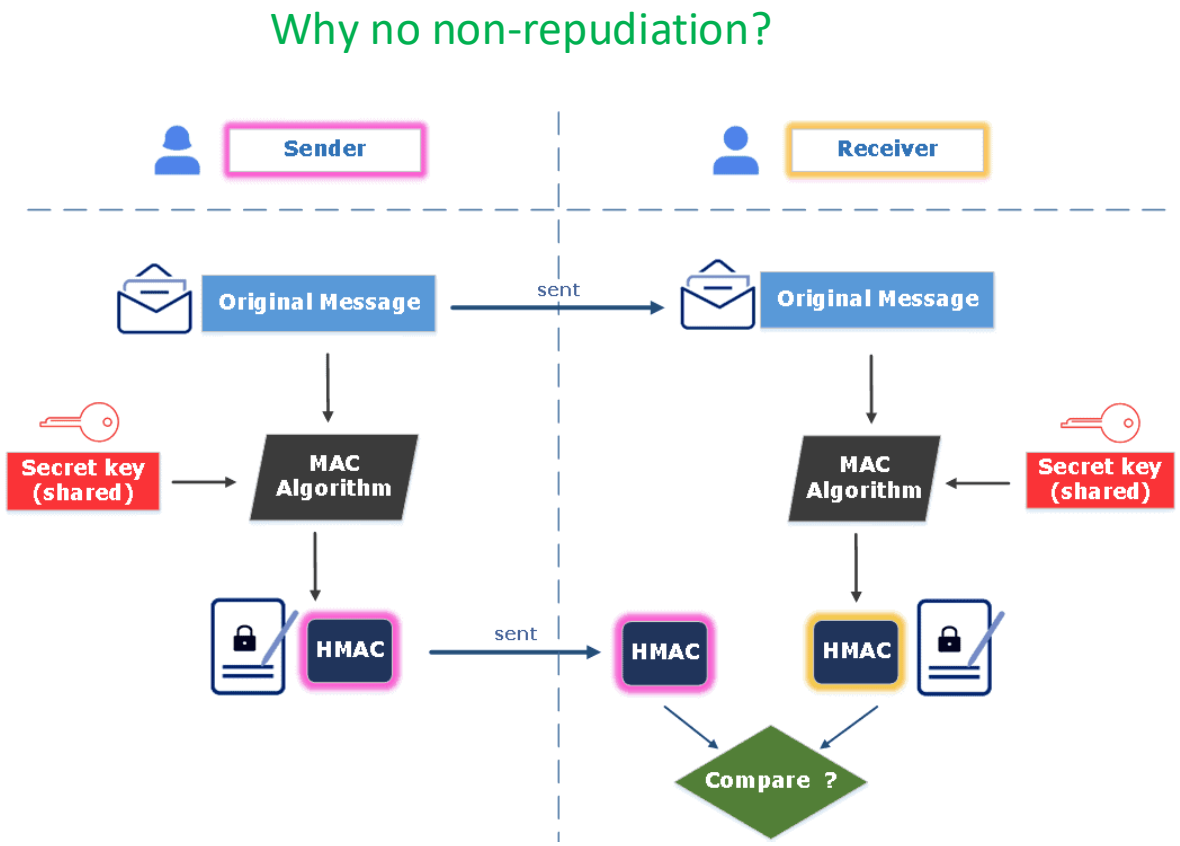
Identify the hash

- https://www.tunnelsup.com/hash-analyzer/
- https://hashes.com/en/tools/hash_identifier



MAC

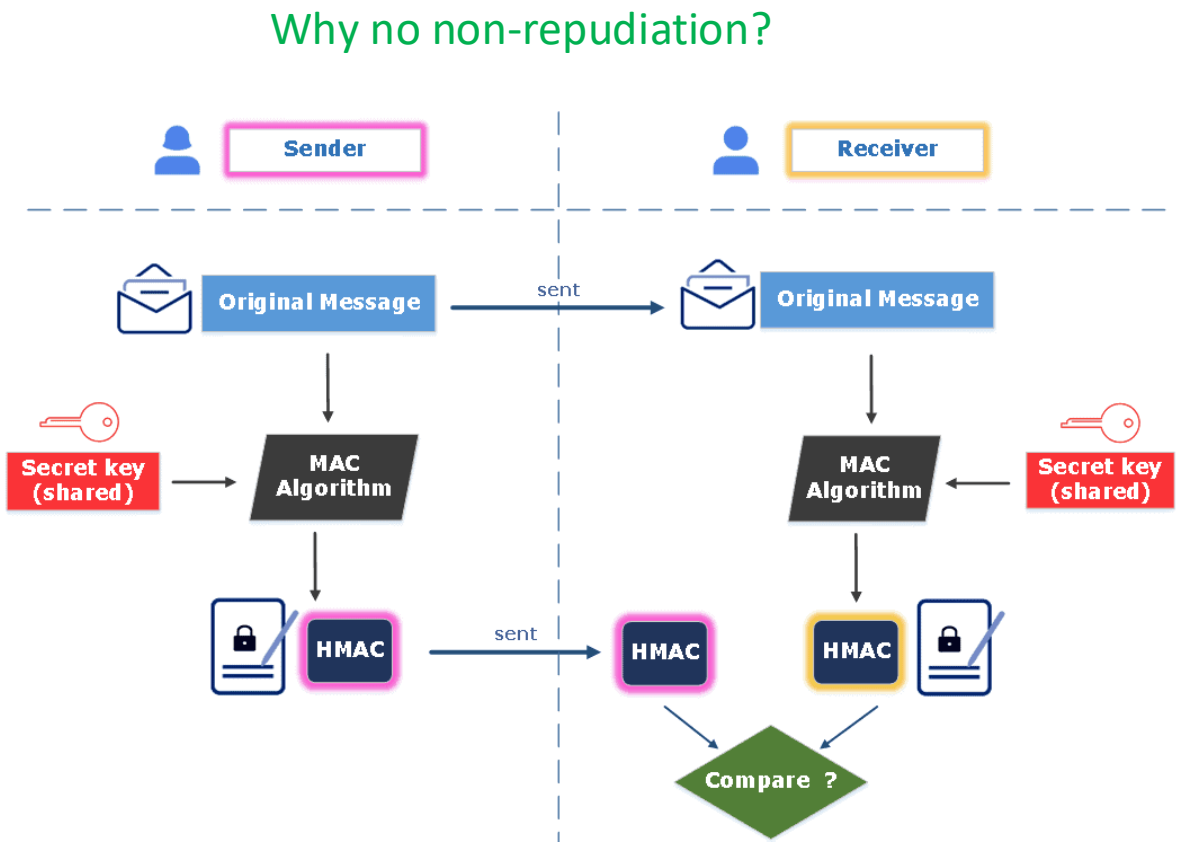
- **Message authentication code (MAC)**: uses a cryptographic hash function and symmetric encryption
- Provides the data **integrity** and **authenticity** security services
- Ensures that a message was not altered and comes from a trusted sender
- Requires establishment of shared secret prior to use of MAC
- MACs no do encrypt data, so no confedentiality



Source: https://www.baeldung.com/cs/hash-vs-mac

Can MAC provide non-repudiation?

- Since the same secret key is used both for generation and verification, MACs cannot provide **non-repudiation,** Any party with the secret key could have generated the MAC



Source: https://www.baeldung.com/cs/hash-vs-mac

Example

- **A real-world example** of where MACs are used is in financial transactions, think of your online banking transactions
- Before any transaction is completed, a MAC is used to confirm that the data hasn’t been altered during transmission between the bank’s server and your device.
- This prevents man-in-the-middle attacks, where an attacker might try to modify the message in transit.
- But, both you and the bank share the same secret key, so if there’s a dispute about who initiated the transaction, MACs alone wouldn't be able to resolve it—hence the limitation on non-repudiation.

Security Feature	Authentication	Integrity	Confidentiality	Non-Repudiation
Digital Certificate	✅ Yes	✅ Yes	✅ Yes	⚠️ Supports, but not directly
Digital Signature	✅ Yes	✅ Yes	❌ No	✅ Yes
Hashing	✅ Yes (passwords)	✅ Yes	❌ No	⚠️ Indirectly
Message Authentication Code (MAC)	✅ Yes	✅ Yes	❌ No	❌ No

Cryptography Challenges

Key Management, Distribution, and Storage

- Challenge: Securely generating, distributing, and managing encryption keys
- Consequence: Compromised or lost keys can compromise data security

Algorithm Vulnerabilities

- Challenge: Discovery of weaknesses in encryption algorithms over time
- Consequence: Vulnerabilities exploited by attackers; need for regular updates

Quantum Computing Threat

- Challenge: Quantum computers could potentially break these algorithms by efficiently factoring large numbers or solving discrete logarithm problems
- Consequence: compromised systems/keys

Cryptography Challenges

Computational Overhead

- Challenge: Strong encryption can be computationally intensive
- Consequence: Slower data transmission and processing in resource-constrained environments

Regulatory and Legal Challenges

Challenge: Cryptography is sometimes subject to regulatory restrictions and legal challenges in various countries

- ➢ government restrictions on cryptographic tools, legal requirements for encryption in various industries

Consequence: These restrictions can limit the use and distribution of strong encryption technologies

Social Engineering

- Challenge: Encryption can be compromised through manipulation
- Consequence: Phishing or social engineering can lead to key exposure

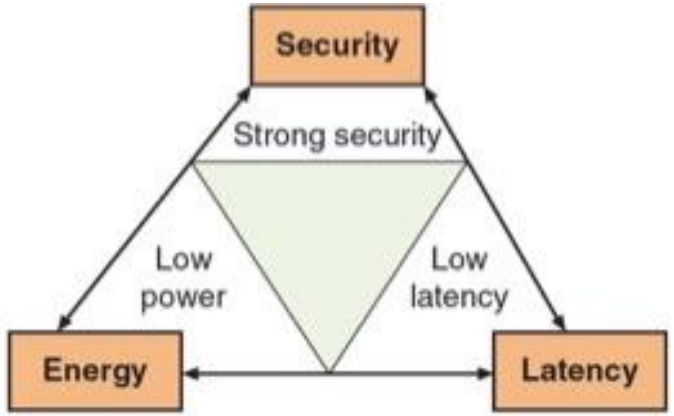
Limitations of Cryptography

- Number of small electronic devices (**low-power devices**) has grown significantly
  - ➢Limited computing power and memory
  - ➢Cannot run robust cryptographic algorithms
  - ➢These devices need to be protected from threat actors

- Applications that require extremely fast response times also face cryptography limitations

- **Resource vs. security constraint** is a limitation in providing strong cryptography due to the tug-of-war between available resources (time and energy) and the security provided by cryptography

  - ➢It is important that there be **high resiliency** in cryptography
  - ➢High resiliency is the ability to quickly recover from these resource vs. security constraints



Resource vs. security constraint

Attacks on Cryptography

- Two of the most common cryptography attacks are algorithm attacks and collision attacks

Algorithm Attacks:

- Methods attackers can use to circumvent strong algorithms:
  - Known *ciphertext attacks*
    - Statistical tools can be used to attempt to discover a pattern in the ciphertexts, which can then be used to reveal the plaintext or key
  - Downgrade attacks
    - A threat actor forces the system to abandon the current higher security mode of operation and instead "fall back" to implementing an older and less secure mode
  - Attacks based on weak algorithms or misconfigurations
    - Selecting weak algorithms should be avoided since they are no longer secure

Attacks on Cryptography

Collision Attacks

- Collision occurs when two different inputs produce the same hash digest
- **Collision attack** is an attempt to find two input strings of a hash function that produce the same hash result
- **Birthday attack**
  - Based on the *birthday paradox*
  - In a random group of 23 people, there is more than 50 percent chance that two people have the same birthday
  - Birthday attack becomes relevant when you consider the probability of two different inputs producing the same hash value
  - You stand more than 50% chance of finding an MD5 collision (sample space of 2^128 possibilities) after around 2^64 operations

Birthday attack:https://www.youtube.com/watch?v=2bEL3ok8D70&ab_channel=DrapsTV
https://auth0.com/blog/birthday-attacks-collisions-and-password-strength/

Passwords and Password Cracking

- How long is your password good for? 30 days? 60 days? 90 days?

Answer: As long as it is not broken

- The bottom line: it will only be a matter of when, not if, your password will be broken

- **Why Do I Have to Change my Password Every 30/60/90 Days?**

- The answer is simple, and there are two reasons why this policy is standard for most companies. The first is protection against stuff that hasn't happened yet. The second is protection against stuff that has happened – just not to your company.

https://www.secureauth.com/blog/security-answers-in-plain-english-why-do-i-have-to-change-my-password-every-30-60-90-days/

Cracking Passwords on a Linux System

- Two files of interest on a typical Linux box:

/etc/passwd -> contains user information like username, user id

/etc/shadow -> contains password hash

More details: http://tldp.org/LDP/lame/LAME/linuxadmin-made-easy/shadow-file-formats.html

Methods:

- Brute-force
- Dictionary attack
- Rainbow tables (type of dictionary attack)
  - ➢See http://project-rainbowcrack.com/ for example.

HashCat /John the Ripper

- Powerful and popular password cracking tools used by cybersecurity professionals, penetration testers, and ethical hackers to test the security of passwords and authentication systems

- Often employed to recover lost or forgotten passwords, assess password strength, and identify weak passwords in a system

Hashcat

- Hashcat is a popular and effective password cracker widely used by both penetration testers and sysadmins as well as black hats
- Cracking passwords has many legitimate uses besides the obvious criminal and spy
- A sysadmin may wish to pre-emptively check the security of user passwords
  - If hashcat can crack them, so can an attacker
- **How does hashcat work?**
  - At its most basic level, hashcat guesses a password, hashes it, and then compares the resulting hash to the one it's trying to crack
- rockyou.txt word list is a popular option. Containing more than 14 million passwords sorted by frequency of use

Sample hash (SHA512) from Ubuntu log:

https://hashes.com/en/tools/hash_identifier

6MOCeF0du$eCgZ9.I.hS5CDST1aQHozhLbBH6rAUj97vvW/22eaCynqLv/whZKM1freAN3n2XQiCWjDr0UFreVv0IAvI.fl0

- With a handy tool like hashcat and a trusty password list like rockyou.txt you can possibly crack the password
- **Required**: Linux machine with Hashcat installed
- **Tutorial**: How To Install Hashcat
  sudo apt install hashcat
- Step 1: Copy the hash that you want to find the password into a new file, for example, user.hash
- Step 2: Download rockyou.txt file and save it into the same folder as user.hash
- Step 3: Then execute.
  - hashcat –a 0 user.hash rockyou.txt –m 1800

-a 0	> Attack mode: use dictionary attack
user.hash	> file that contains hash to be cracked
rockyou.txt	> file containing password list
-m 1800	> Hash mode: SHA512cyrpt

Step 4: It takes time. Do some stretching or drink a coffee 🙂

```
Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$6$SaltVal2$ybuPu7Nmo9LKn0p0ozhFhFw2SS2cqkLsx8c5OEAWFkIJjtXBEJqxUQzLh900QMgFTGiw6YuFDueNAapfLKt0f1:forgot

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$SaltVal2$ybuPu7Nmo9LKn0p0ozhFhFw2SS2cqkLsx8c5OEA...LKt0f1
Time.Started.....: Mon Sep  4 03:59:46 2023 (1 sec)
Time.Estimated...: Mon Sep  4 03:59:47 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2289 H/s (5.29ms) @ Accel:256 Loops:256 Thr:1 Vec:2
Recovered........: 1/1 (100.00%) Digests
Progress.........: 3072/14344384 (0.02%)
Rejected.........: 0/3072 (0.00%)
Restore.Point....: 2816/14344384 (0.02%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4864-5000
Candidate.Engine.: Device Generator
Candidates.#1....: pirate -> dangerous
```

Tool: John the Ripper (JtR)

- Install: http://www.openwall.com/john/
  sudo apt install snapd
  sudo snap install john-the-ripper
- Can do Brute-force and Dictionary attack

- "If valid password files are specified but no options are given, John will go through the default selection of cracking modes with their default settings."
  http://www.openwall.com/john/doc/OPTIONS.shtml

- Example: **hash**: CBFDAC6008F9CAB4083784CBD1874F76618D2A97
- **Hash type**: SHA1
- Command: **john -format=raw-sha1 --wordlist=rockyou.txt user.hash**

Tool: John the Ripper (JtR)

```
robinchataut@ubunturobin:~/Desktop/hash$ john -format=raw-sha1 --wordlist=rockyou.txt user.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 ASIMD 4x])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2023-09-15 01:54) 100.0g/s 138400p/s 138400c/s 138400C/s liberty..password123
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
robinchataut@ubunturobin:~/Desktop/hash$ locate john.pot
/home/robinchataut/snap/john-the-ripper/584/.john/john.pot
robinchataut@ubunturobin:~/Desktop/hash$ cat /home/robinchataut/snap/john-the-ripper/584/.john/john.pot

$dynamic_26$cbfdac6008f9cab4083784cbd1874f76618d2a97:password123
```

Additional Passwords Crackers

- L0phtCrack (commercial; http://www.l0phtcrack.com/)
- Cain & Abel (http://www.oxid.it/cain.html)
- THC Hydra (free and open source)

References

- *Introduction to Security Cryptography*, *Ming Chow, Tufts University*
- *CompTIA, Security+*
- *Some slides were adopted from UCS CSE107: Intro to Modern Cryptography*

Reminders

- Assignment#4 available
  - Start Early

- Midterm
  - A week before spring break