

• • • • •
• • • • •

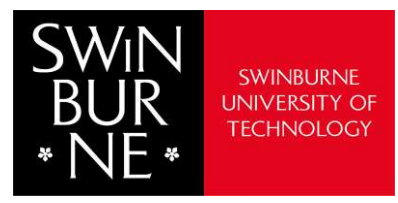
COS80013

Internet Security

Week 9

Presented by Yasas

5 May 2025



• • • • •
• • • • •

Acknowledgement of Country

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne’s Australian campuses are located in Melbourne’s east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne’s Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures, and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.

• •
• •

• • • • • • • • • • • •
• • • • • • • • • • • •





Cryptography

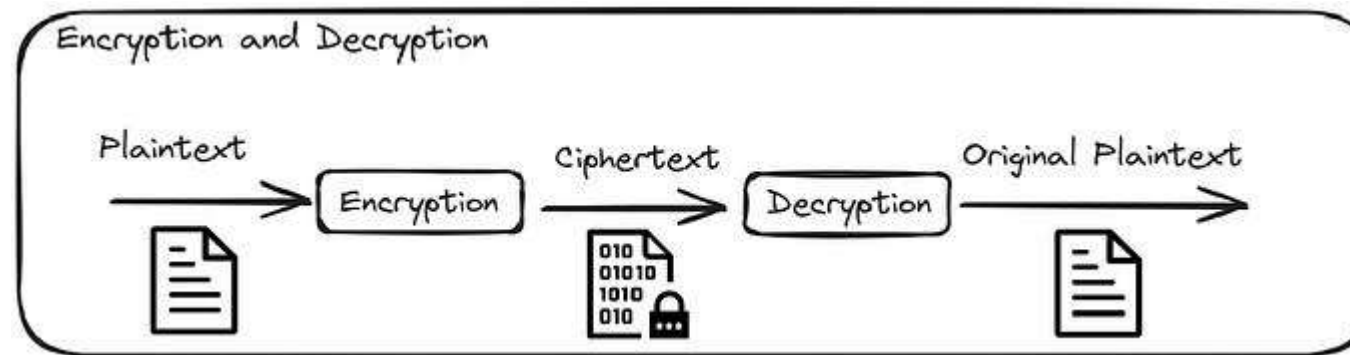
"brutus"
"euxwxv".

Terminology

Suppose a sender wants to send a message to a receiver. She wants to make sure an eavesdropper cannot read the message.

- Message or Plaintext: it can be a stream of bits, a text file, a bitmap, audio, video, etc. and can be intended for either transmission or storage.
- Encryption: process of hiding a message.
- Ciphertext: encrypted message.
- Decryption: process of turning ciphertext back into plaintext

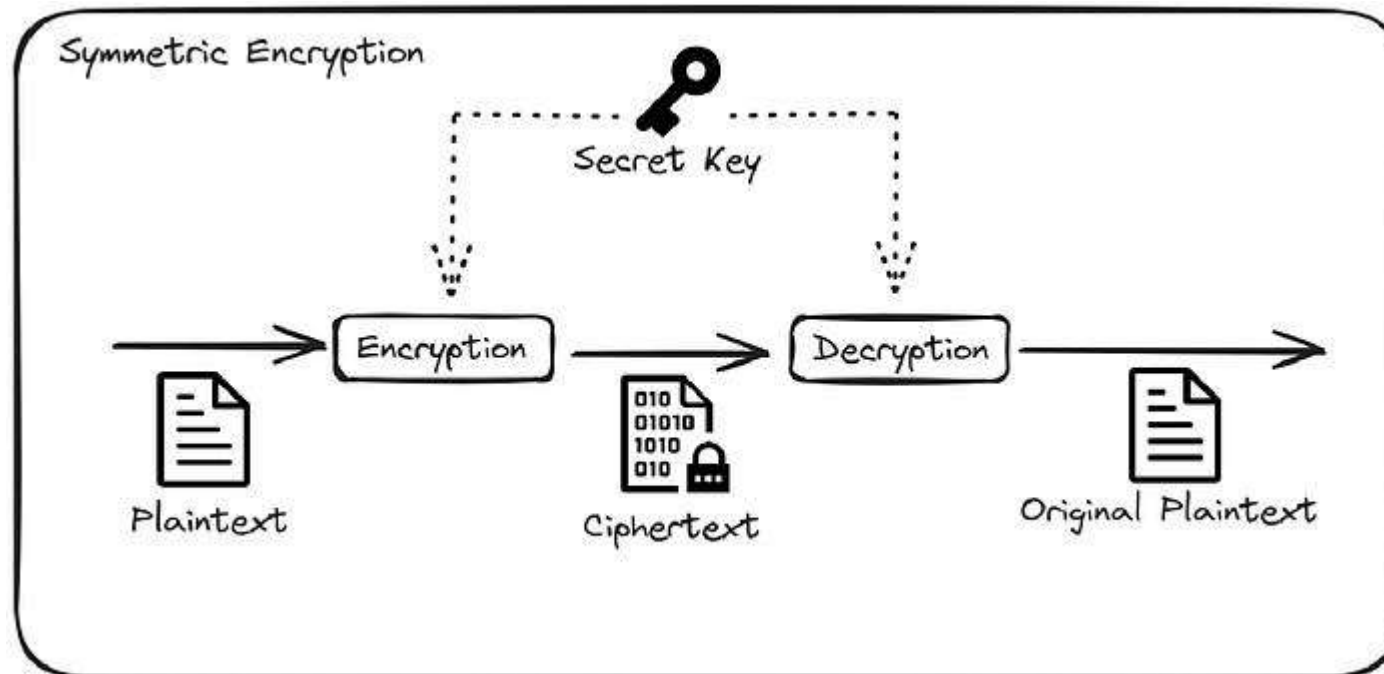
- Unconditional Security
- one-time pad
- Brute-Force Attack
- Computational Security



Type - Symmetric Cryptography

Symmetric Cryptography:

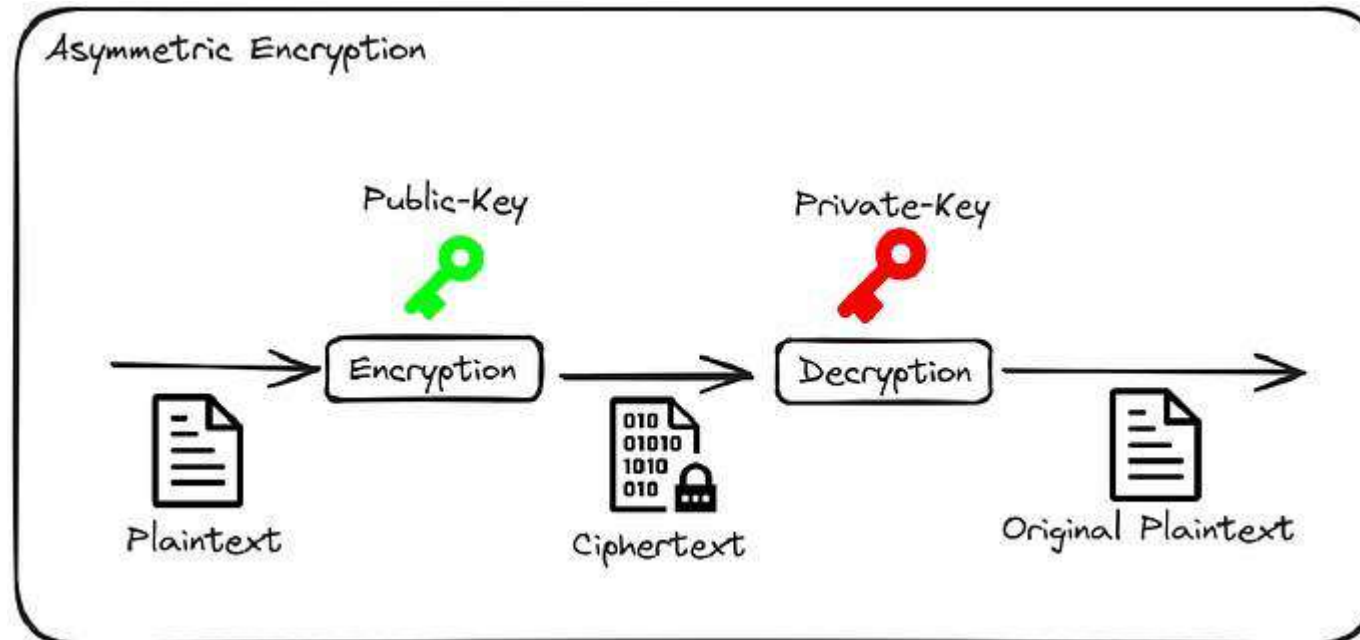
- **Single Key**: Symmetric cryptography uses a **single secret key** for both encryption and decryption. This key must be kept confidential and known only to the parties involved.
- **Efficiency**: Symmetric encryption algorithms are generally **faster** and more computationally efficient than asymmetric algorithms. They are suitable for encrypting **large amounts of data**.
- **Key Management**: Key distribution and management can be challenging, especially in large-scale systems, as each pair of communicating parties needs to share a common secret key securely.



Type - Asymmetric Cryptography

Asymmetric Cryptography (Public-Key Cryptography):

- **Key Pair:** Asymmetric cryptography uses a pair of keys: a public key and a private key. The public key is widely distributed and used for encryption, while the private key is kept secret and used for decryption.
- **Security and Key Exchange:** Asymmetric cryptography provides a solution to the key distribution problem in symmetric cryptography. Public keys can be freely shared, allowing anyone to encrypt data for the owner of the corresponding private key.
- **Digital Signatures:** Asymmetric cryptography is often used for digital signatures, where the private key is used to sign data, and the public key is used to verify the signature.

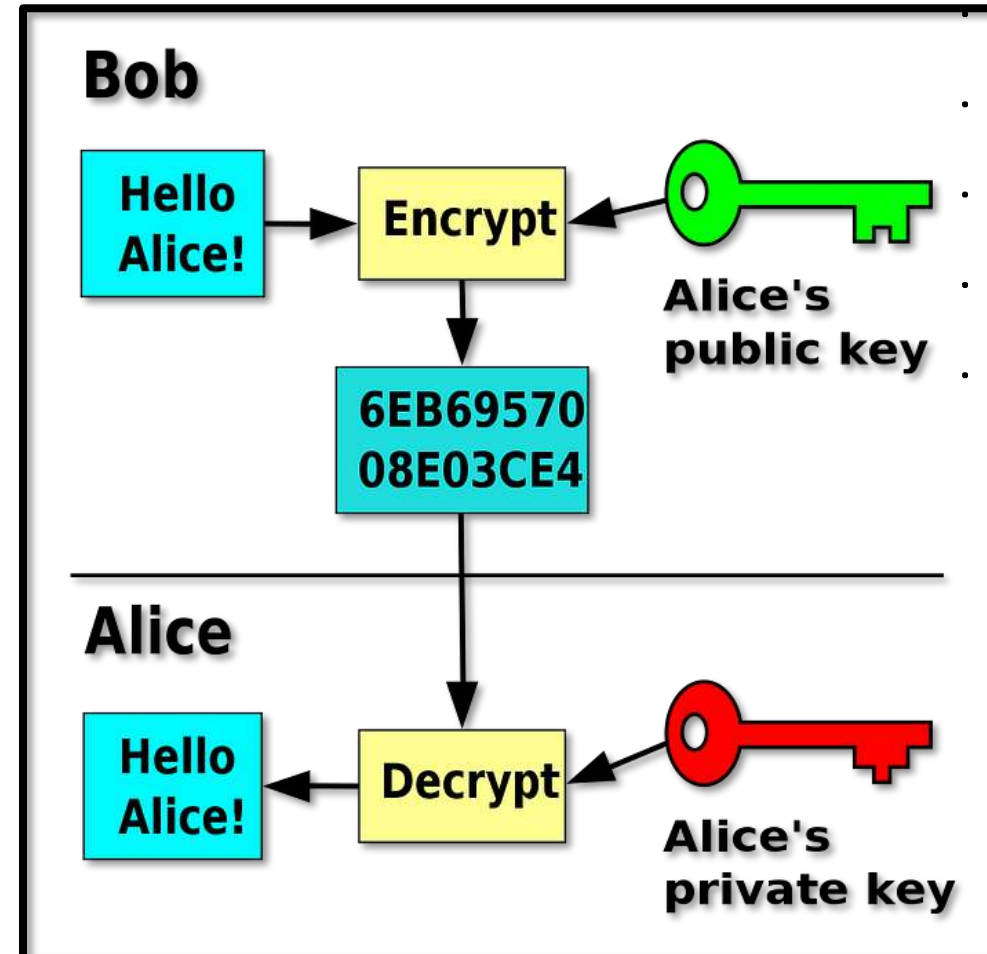


Asymmetric Cryptography - Example

Public-Key Real World Example

Imagine you want to send a secret message to your friend, but you're worried about someone intercepting it. Public-key encryption can help.

- You and your friend each have a pair of keys: a public key and a private key.
- Your public key is like a padlock that everyone can see, but only your private key can unlock it.
- You lock your message with your friend's public key and send it.
- Only your friend, who has the corresponding private key, can unlock and read the message



Substitution Cipher (Caesar Cipher):

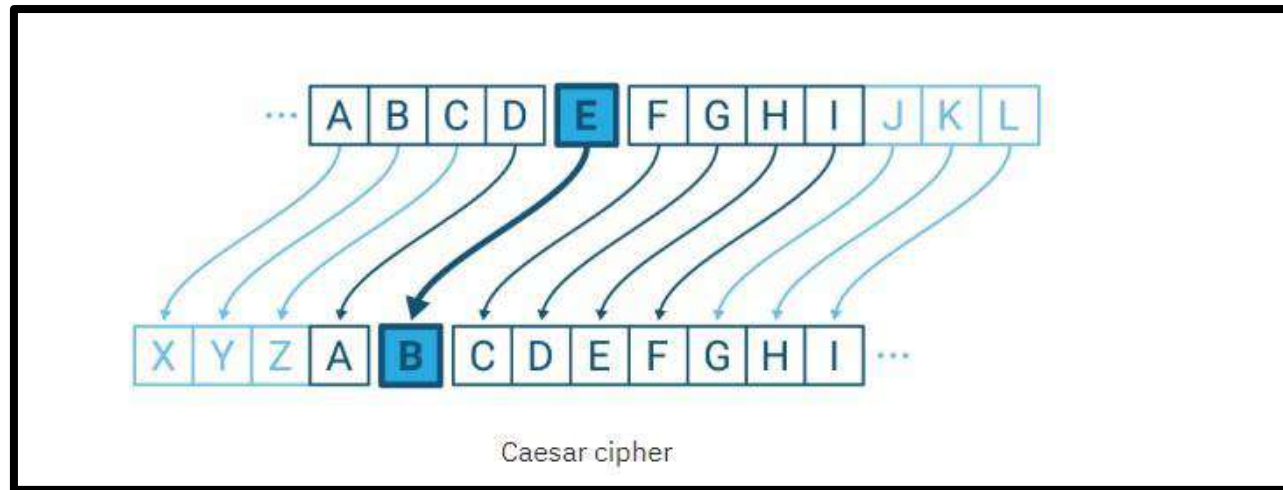
In a Caesar cipher, each letter in the plaintext is shifted a fixed number of positions down or up the alphabet.

Plaintext: "HELLO"

Shift: 3

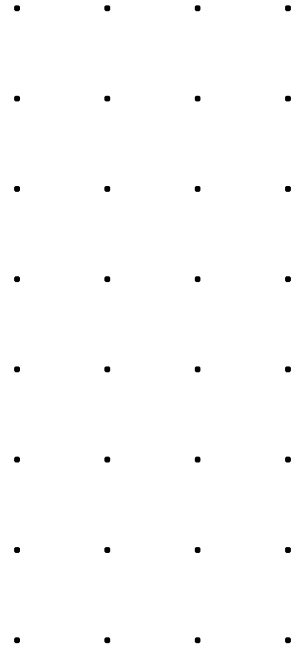
Ciphertext: "EBIIL"

Here, each letter in the plaintext has been shifted three positions up the alphabet to produce the ciphertext. . . .



Security - Caesar Cipher

- Not considered secure for modern encryption purposes
- Reason:
 - 1) With only 26 possible shift values in the English alphabet (assuming a standard Caesar Cipher), an attacker can easily try all possible shifts to decrypt the message using a brute-force attack
 - 2) Does not obscure letter frequencies, making it vulnerable to frequency analysis.



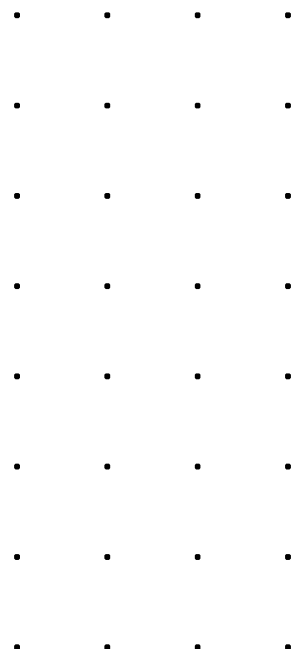
Polyalphabetic Cipher (Vigenère Cipher)

Key Components:

- **Plaintext** - original message
- **Keyword** - a secret word or phrase

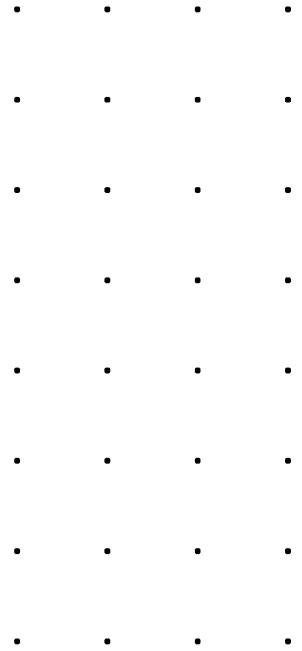
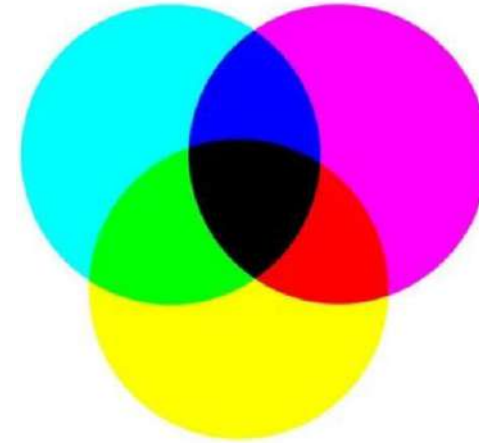
Encryption Process:

- **Choose a Keyword:** Select a secret keyword that is as long as or longer than the plaintext message.
- **Repeat the Keyword:** If the keyword is shorter than the plaintext, repeat it to match the length of the plaintext.
- **Encrypt the Message** - For each letter in the plaintext:
 - Find the corresponding letter in the keyword.
 - Use the Caesar Cipher principle to shift the letter in the plaintext by the value of the corresponding keyword letter.
 - Record the resulting letter in the ciphertext.



Public Key Cryptography

- A.k.a. asymmetric cryptography
- Two keys – public and private
- Public key is shared
- Private key is kept secret
- Well suited for organizations



Public Key Cryptography

Diffie-Hellman



Client

1

Shared

Mod = $P = 13$

Base = $g = 5$



Server

2

Secret = $a = 5$

Secret = $b = 3$

3

$A = g^a \bmod P$

A

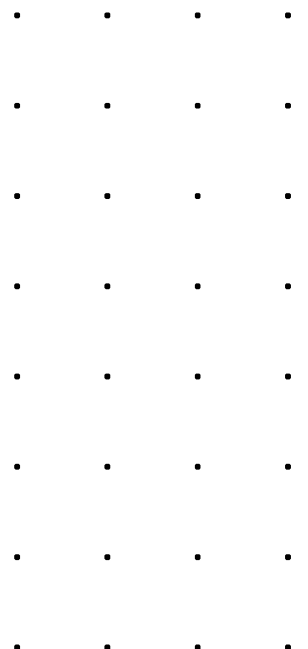
$B = g^b \bmod P$

B

4

$Shared = B^a \bmod P$

$Shared = A^b \bmod P$



Public Key Cryptography

Diffie-Hellman



Client

1

Shared

Mod = $P = 13$

Base = $g = 5$



Server

2

Secret = $a = 5$

3

$$\begin{aligned} A &= g^a \bmod P \\ &= 5^5 \bmod 13 \\ &= 5 \end{aligned}$$

$A = 5$

$B = 8$

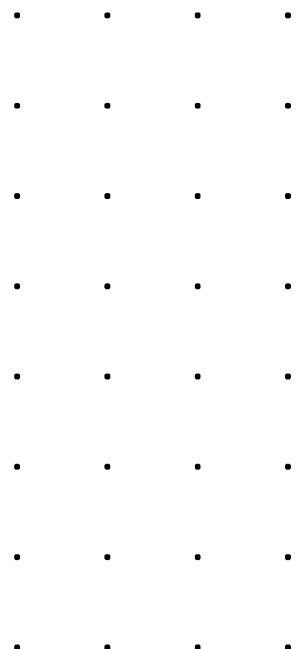
4

$$\begin{aligned} \text{Shared} &= B^a \bmod P \\ &= 8^5 \bmod 13 \\ &= 8 \end{aligned}$$

Secret = $b = 3$

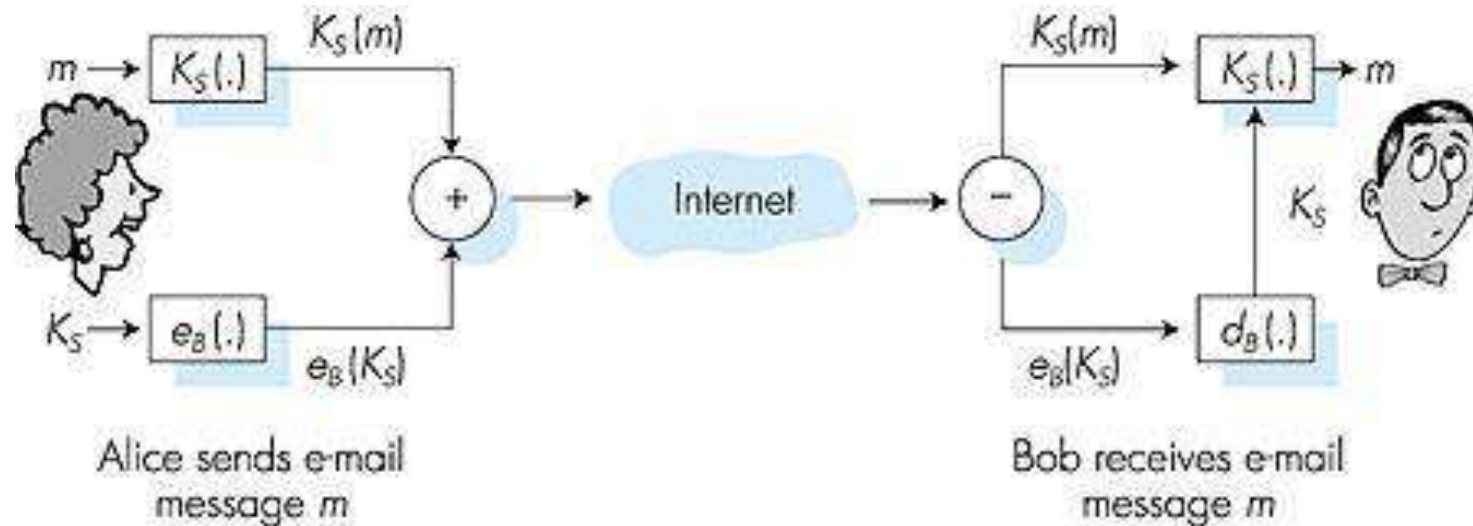
$$\begin{aligned} B &= g^b \bmod P \\ &= 5^3 \bmod 13 \\ &= 8 \end{aligned}$$

$$\begin{aligned} \text{Shared} &= A^b \bmod P \\ &= 5^3 \bmod 13 \\ &= 8 \end{aligned}$$



Secure e-mail

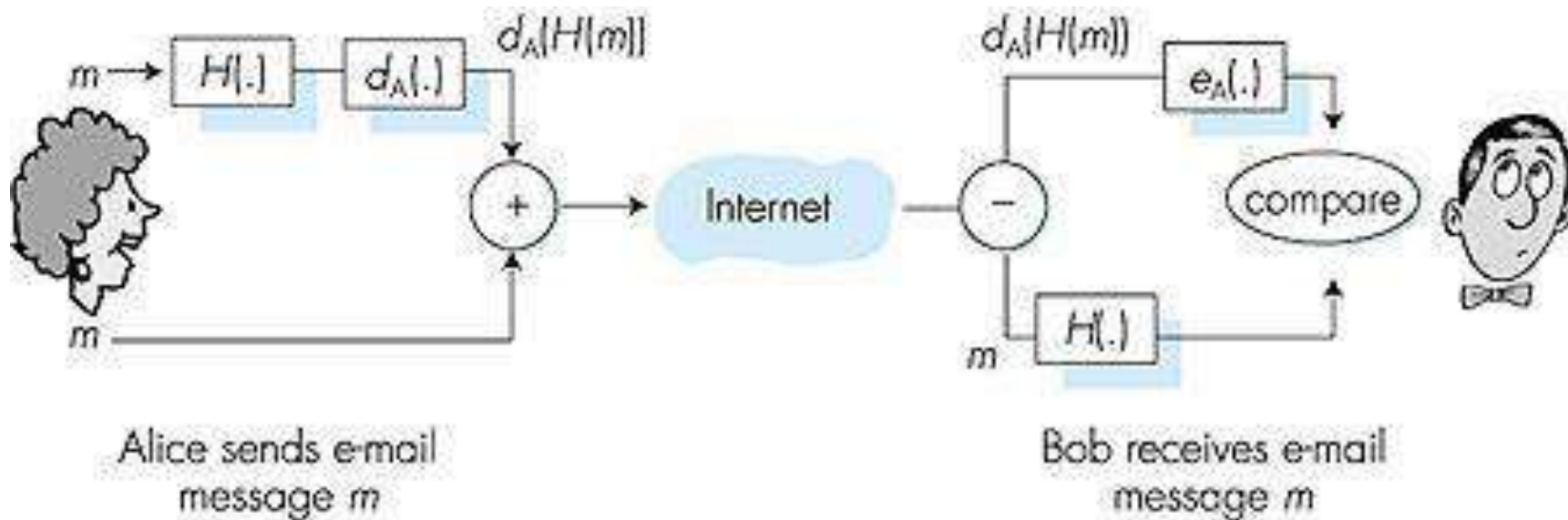
- Alice wants to send secret e-mail message, m , to Bob.



- generates random symmetric private key, K_S .
- encrypts message m with K_S
- also encrypts K_S with Bob's public key.
- sends both $K_S(m)$ and $e_B(K_S)$ to Bob.

Secure e-mail (continued)

- Alice wants to provide sender authentication message integrity.

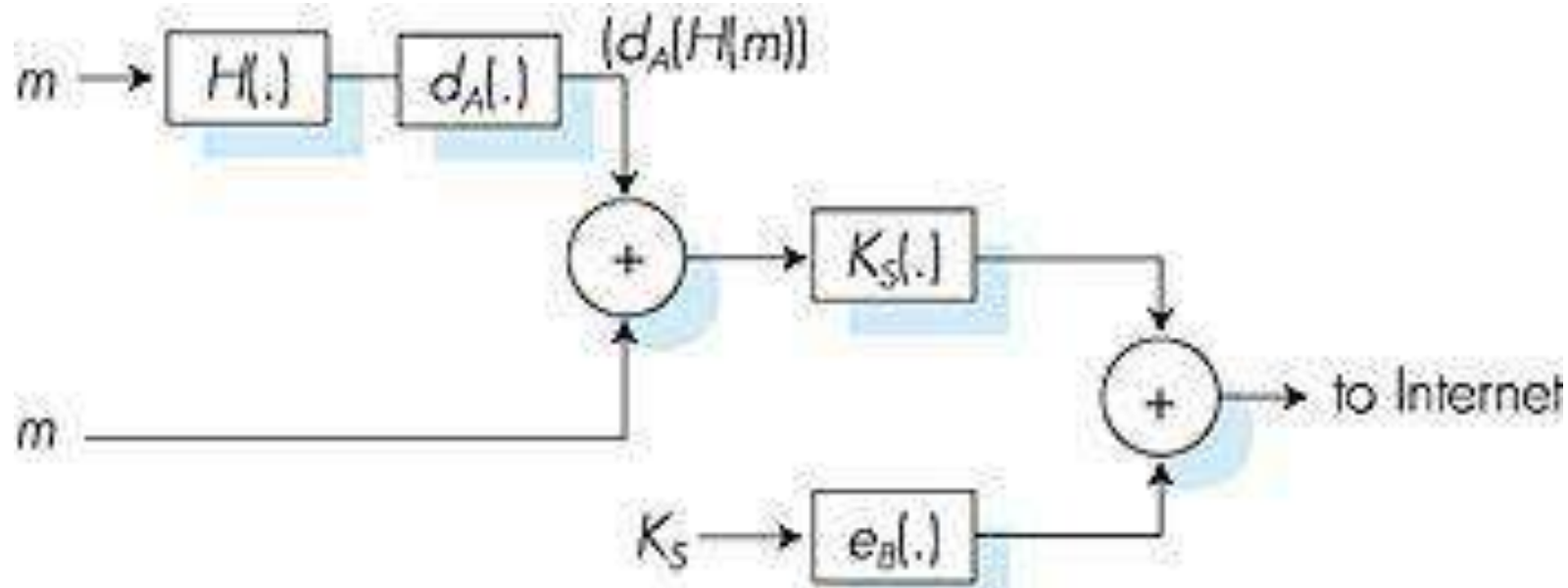


- Alice digitally signs message.
- sends both message (in the clear) and digital signature.

Secure e-mail (continued)

Alice wants to provide secrecy, sender authentication, message integrity.

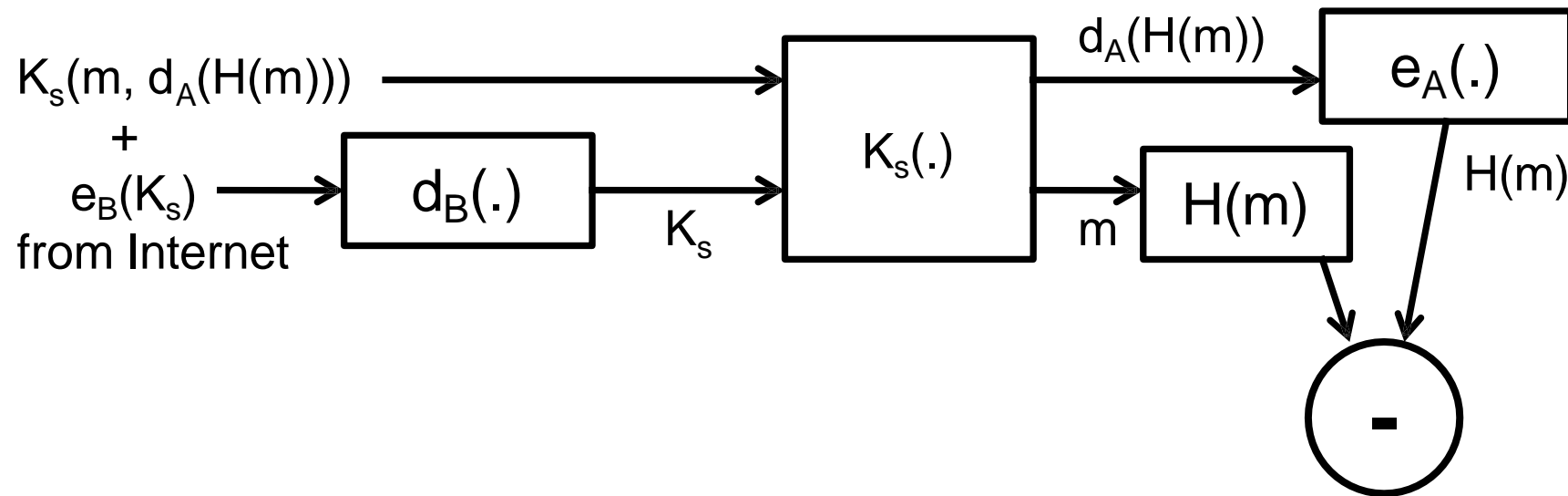
Alice hashes and encrypts the hash with her private key (dig. signature), adds the message and encrypts with a session key. Encrypts the session key with Bob's public key and sends both.



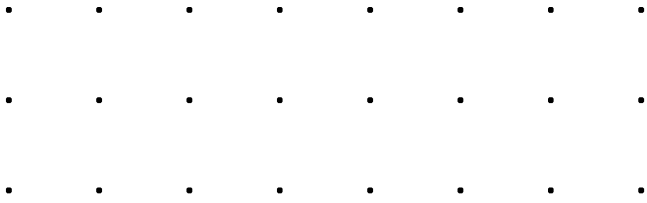
Note: Alice uses both her private key d_A , Bob's public key e_B .

Secure e-mail (continued)

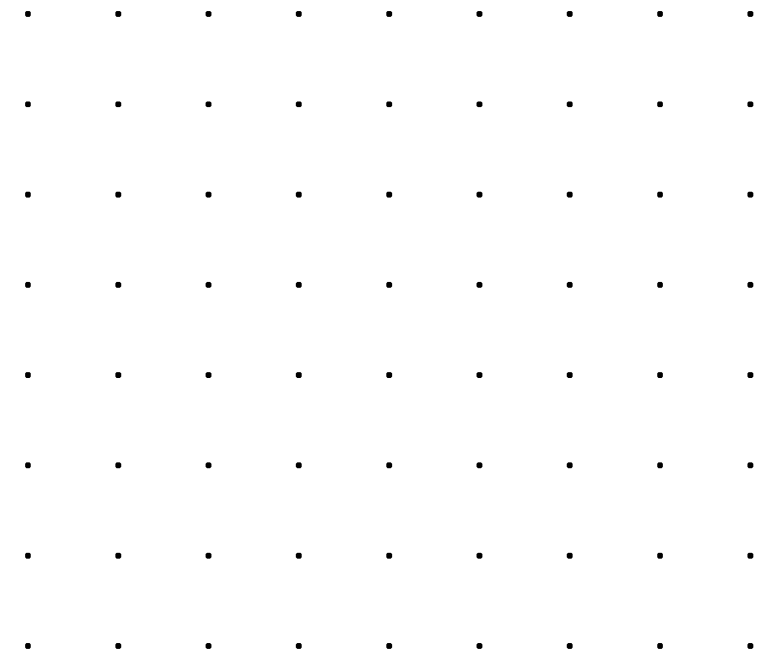
Bob extracts session key with Bob's private key, extracts dig. sig. and message with session key and uses Alice's public key to extract hash from dig. sig. Hashes message and compares hashes.



Note: Bob uses Alice's public key e_A and Bob's private key d_B .



Hash Function

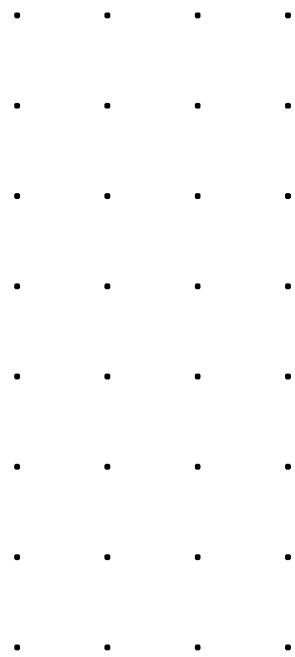


Hash Function

A hash function is a mathematical algorithm that takes input data of arbitrary size and produces a fixed-size output known as a hash value or message digest.

- Purpose

- Message Authentication: verify that a message has not been tampered with during transmission by generating a hash value of the message at the sender's end and comparing it to the hash value generated by the receiver. If the hash values match, it indicates that the message has not been altered.
 - Password Protection: Passwords are often stored as hash values to prevent unauthorized access to user accounts. When a user enters their password, the hash function is applied to the entered password and compared with the stored hash value. If the two hash values match, the user is authenticated and granted access.
 - Digital Signatures: Hash functions are used in digital signature schemes to ensure that the signed message has not been tampered with. The hash value of the message is encrypted using the signer's private key to generate the digital signature.
- Popular hash functions include SHA-256, SHA-3, and BLAKE2.

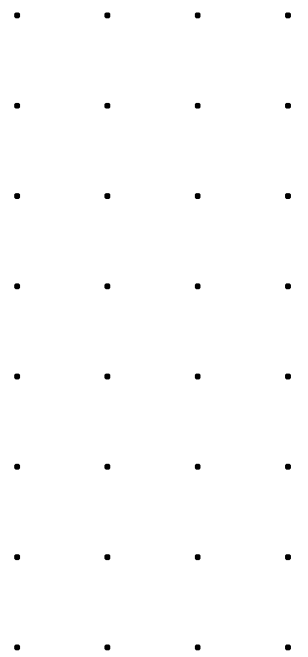


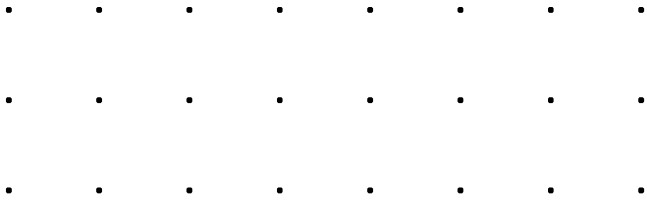
Hash Function – Properties

This hash value is a unique representation of the input data, meaning that any change made to the input data will result in a completely different hash value.

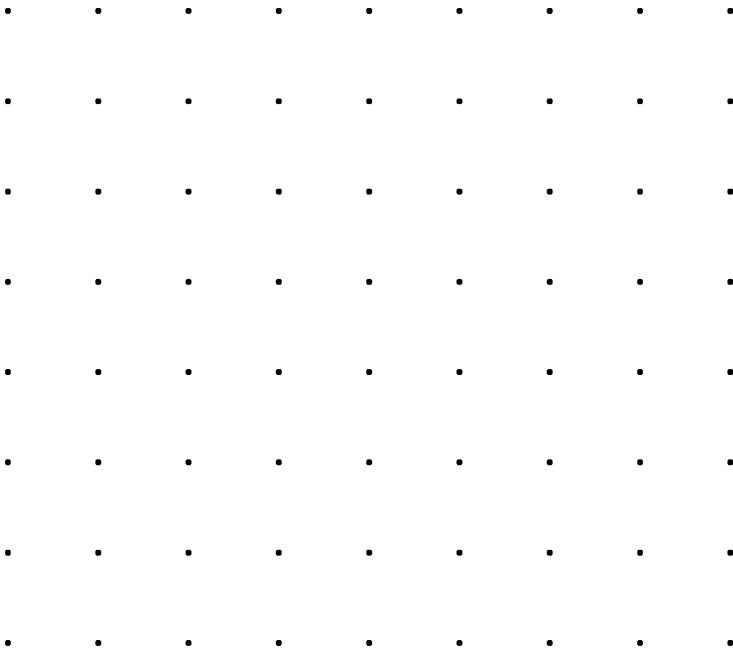
Some common properties of a secure hash function include:

- Deterministic: given the same input, the hash function will always produce the same output.
- Pre-image resistant: it should be computationally infeasible to find an input that generates a specific hash value.
- Collision resistant: it should be computationally infeasible to find two different input values that produce the same hash value.
- Avalanche effect: any small change to the input should result in a significant change in the output.





Thank you



.
.

COS80013

Internet Security

Week 9

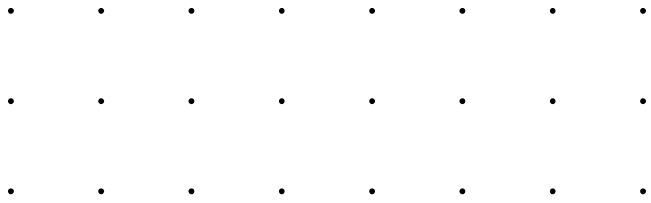
Presented by Dr Rory Coulter

05 May 2025

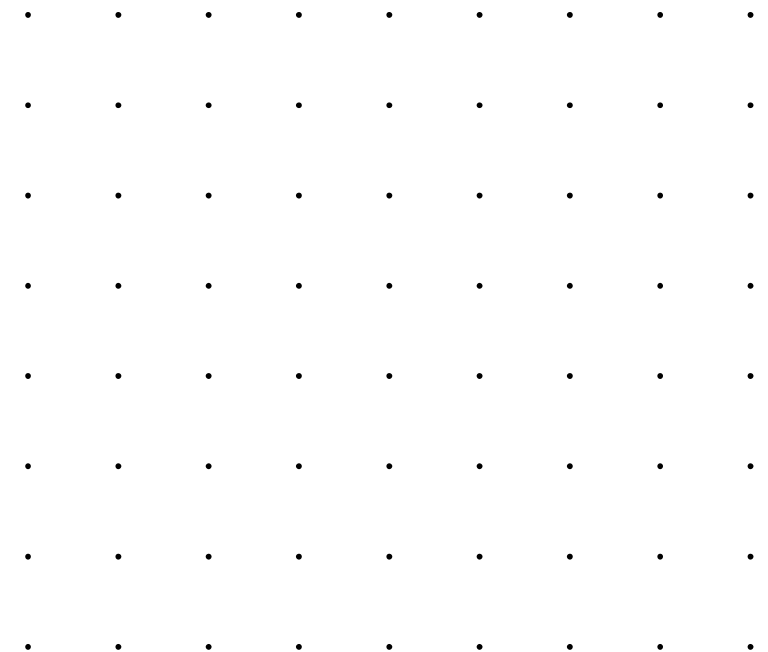


. . .
. . .

.
.



Week 9 Class

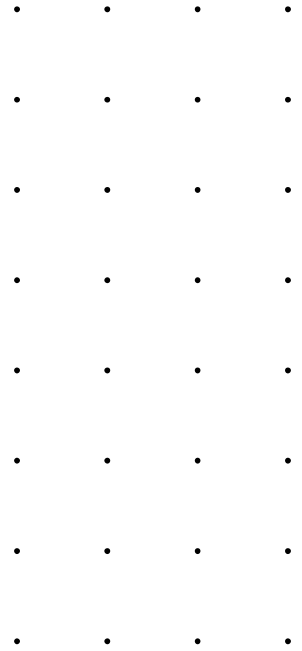


Assignment 2 Preparation

OSINT, Prefetch

Overview and basic usage of each element provided

- What
- Resources
- Demonstration

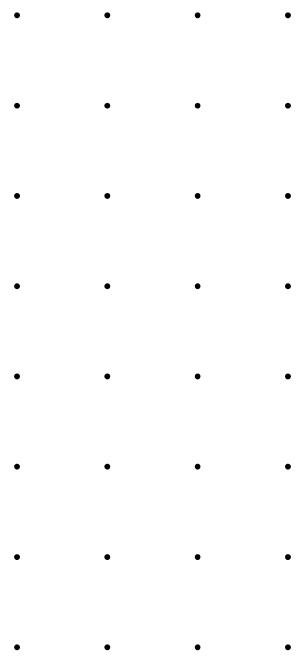


OSINT

Open Source Intelligence

Gather and analyse publicly available information

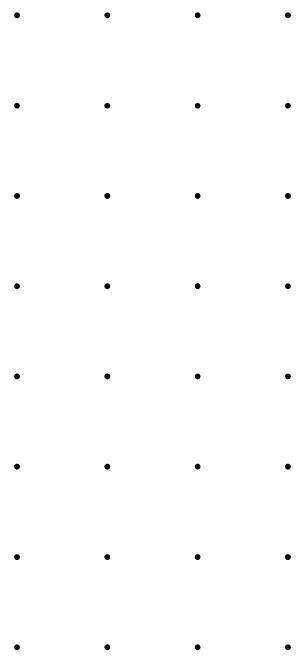
- In piecing together an incident, OSINT plays a key role
 - Likely you are not the first organisation targeted or to see these indicators/tradecraft
 - If you are, then trouble
 - A range of open exchanges exist online, coupled with automated malware analysis
- tools
 - VirusTotal, AlienVault OTX
 - Advisories (ASD ACSC, CISA, etc.)
 - Advisories (Vendor)



Recap: Indicators of Compromise (IoC)

Indicate an incident has taken place

- Help understand the type of incident and its source
- Threat intelligence solutions leverage IoCs to quickly connect cybersecurity incidents to known threat profiles
- For example, if a company has outbound traffic to an IP address known to be used for malicious activity, cyber threat intelligence can connect that IP address to a threat actor, and provide information about malware distributed by that attacker.
- File hash
- IP, Domain
- Registry key types
- File extensions
- Directory path
- Etc.

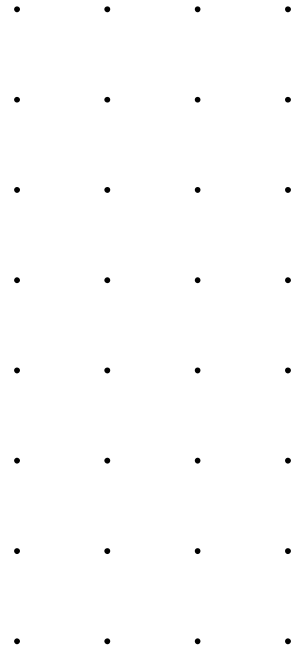


Operational Security

Keeping our communication and actions hidden from the adversary

OPSEC

- What are the ramifications of sharing (inputting) information into public systems?
- What if we upload files?

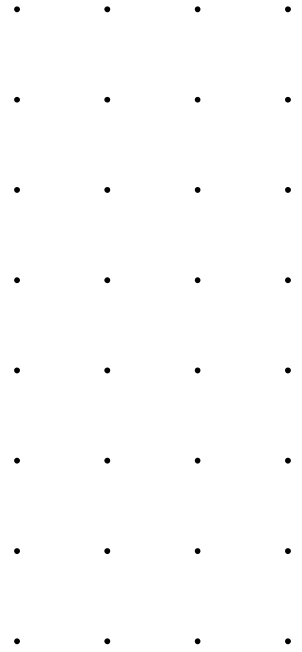


Operational Security

Keeping our communication and actions hidden from the adversary

OPSEC

- What are the ramifications of sharing (inputting) information into public systems?
- What if we upload files?



Defanging Indicators

Reduce OPSEC Errors

Prepare indicators for sharing or reporting

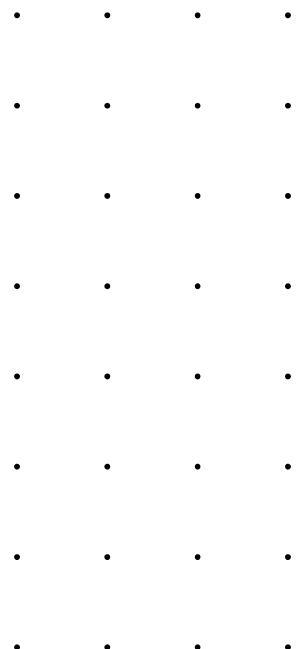
- Aim: remove the ability for links or IPs to resolve
- Add brackets around the last dot
- 10.1.1[.]1, example[.]com
- Replace tt with xx
- Hxxps://example[.]com
- Remove the @
- infoATexample[.]com
-

OSINT Resources

VirusTotal, AlienVault, Advisories, Web

Not limited to these sources

- Search indicators: VirusTotal & AlienVault, Joesandbox MalwareBazaar also
- Search but also obtain reports, Advisories
- General searching across the Internet

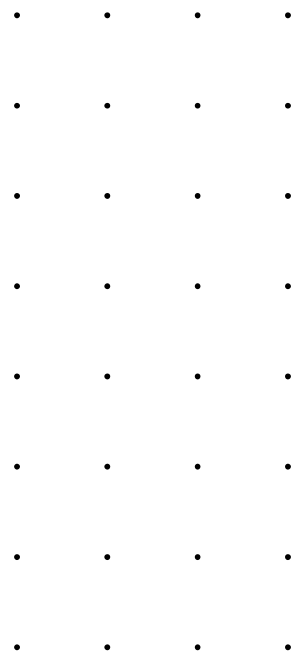


OSINT Demonstration

499d440f84e0d1cd662575356b4398865063bef6cfc1078668a4cec6eacb9e22

What can we discover using automated tools

- VirusTotal
- AlienVault
- Joesandbox
- MalwareBazaar

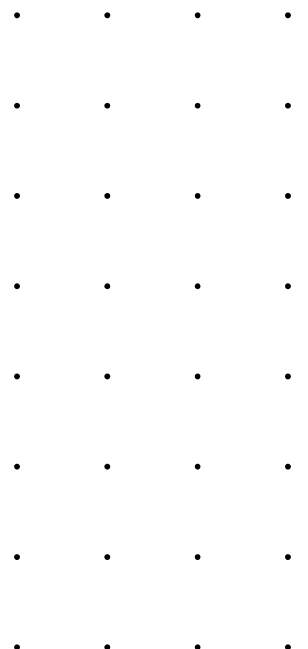


OSINT Demonstration

944153fb9692634d6c70899b83676575

Advisories

- Search and pivot on information
- Use indicators in advisories

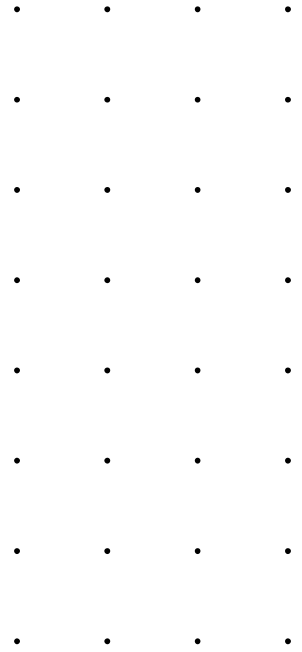


OSINT Demonstration

Wpzlbji file extension

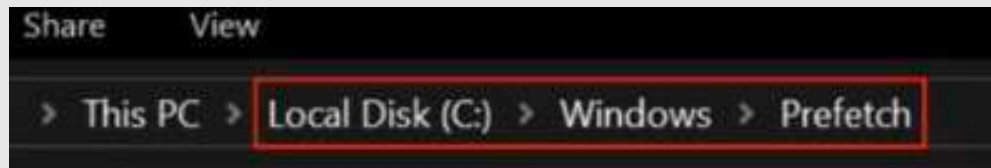
Internet search












- Can we attribute
- Recall ransomware is overt



What are Prefetch Files?

- **Definition:** Prefetch files are used by Windows to speed up the startup process of applications. They store information about the program's startup and commonly accessed files
- **Location:** Found in the **C:\Windows\Prefetch** directory



PC > Local Disk (C:) > Windows > Prefetch			
Name	Date modified	Type	Size
 BACKGROUNDTASKHOST.EXE-62B44B82....	22/12/2021 3:58 PM	PF File	11 KB
 SVCHOST.EXE-D5ACC972.pf	22/12/2021 3:58 PM	PF File	5 KB
 CHROME.EXE-CCF9F3F5.pf	22/12/2021 3:57 PM	PF File	34 KB
 SEARCHFILTERHOST.EXE-10E4267C.pf	22/12/2021 3:56 PM	PF File	4 KB
 SEARCHPROTOCOLHOST.EXE-C6CFE2A8....	22/12/2021 3:56 PM	PF File	4 KB
 SVCHOST.EXE-8A29D439.pf	22/12/2021 3:56 PM	PF File	4 KB
 TIWORKER.EXE-0D12692A.pf	22/12/2021 3:56 PM	PF File	18 KB
 TRUSTEDINSTALLER.EXE-B018CCBF.pf	22/12/2021 3:56 PM	PF File	5 KB
 DLLHOST.EXE-8E84E9F3.pf	22/12/2021 3:56 PM	PF File	6 KB
 TASKHOSTW.EXE-1EAF2222.pf	22/12/2021 3:56 PM	PF File	17 KB
 SVCHOST.EXE-824BF13F.pf	22/12/2021 3:55 PM	PF File	5 KB

What are Prefetch Files?

- **Format:** Prefetch files have a .pf extension (e.g., APPNAME.EXE-XXXXXXXXX.pf)
- **Range:** include GUI applications , command line, exes, .com. binary

Purpose of Prefetch Files

- **Faster Boot and Application Load Times:** They optimize the loading of applications by pre-loading necessary data into memory
- **Tracking Usage:** Helps track how often and when a program is used
- **Increase or enhance or improve the user experience**

How Prefetch Files Work

- **Process:**
 - **1. When a program is executed, Windows checks if a prefetch file exists**
 - **2. If it does, Windows uses the prefetch data to load the program faster**
 - **3. If not, Windows creates a new prefetch file for that application**

Important of Prefetch Files in Digital Forensics

- **Evidence Collection:** Prefetch files can provide timestamps of program execution
- **Identifying Malicious Activity:** They help forensic analysts identify unusual or unauthorized program usage
- **User Behaviour Analysis:** Understanding which applications were used and when

Example

- Run Count
- Last Run and Other Run Times
- Directories Referenced
- Files Referenced

```
Executable name: DLLHOST.EXE
Hash: 6F82F1F3
File size (bytes): 40,250
Version: Windows 10

Run count: 4
Last run: 2021-12-22 06:47:11
Other run times: 2021-12-22 06:33:29, 2021-12-22 06:15:23, 2021-12-22 06:14:11

Volume information:
M0: Name: \VOLUME{01d50914a6ef8437-9ca78b19} Serial: 9CA78B19 Created: 2019-05-12 22:47:18 Directories: 31 File references: 101

Directories referenced: 31
00: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAM FILES
01: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAM FILES\WINDOWSAPPS
02: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAM FILES\WINDOWSAPPS\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C
03: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAM FILES\WINDOWSAPPS\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
04: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA
05: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT
06: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS
07: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY
08: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES
09: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C
10: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
11: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\
12: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\
13: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\
14: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\
15: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\
16: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE\

Files referenced: 66
00: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\NTDLL.DLL
01: \VOLUME{01d50914a6ef8437-9ca78b19}\$MFT
02: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64.DLL
03: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64WIN.DLL
04: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\KERNEL32.DLL
05: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\USER32.DLL
06: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\GDI32.DLL
07: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64CPU.DLL
08: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64\NTDLL.DLL
09: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64\DLLHOST.EXE
10: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\WOW64\KERNELBASE.DLL
11: \VOLUME{01d50914a6ef8437-9ca78b19}\WINDOWS\SYSTEM32\LOCALIZATION\NLS
12: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
13: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
14: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
15: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
16: \VOLUME{01d50914a6ef8437-9ca78b19}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\MICROSOFT.SKYPEAPP_15.79.95.0_XB6_KZF8QXF38ZG5C\SKYPE
```

Conclusion

- **Key Takeaways:**

- Prefetch files are crucial for performance optimisation in Windows
- They provide valuable data for forensic investigations
- Understanding prefetch files can help in both system optimization and digital forensic analysis

• • • • • • • •
• • • • • • • •
• • • • • • • •



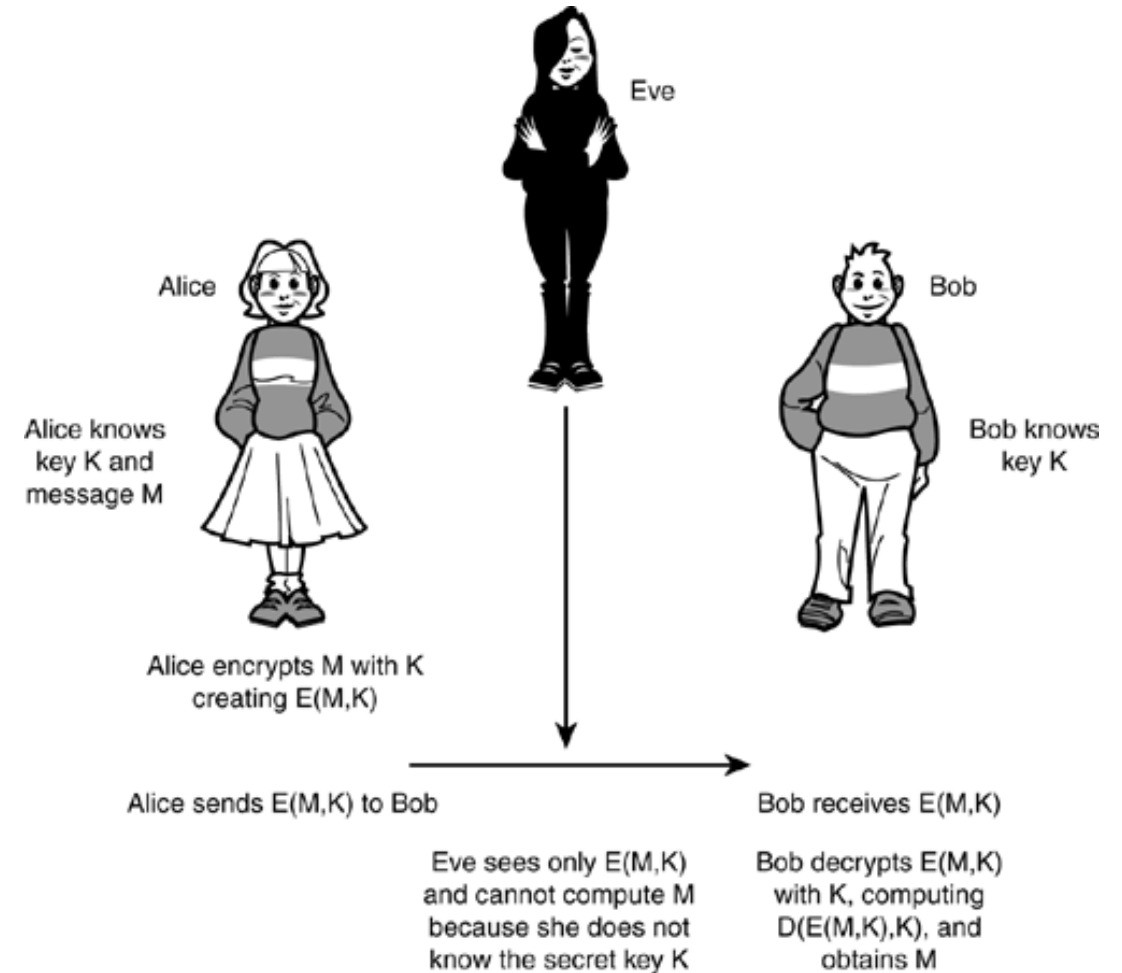
Basic Scenario of Cryptography

• • • • • • • •
• • • • • • • •
• • • • • • • •
• • • • • • • •
• • • • • • • •
• • • • • • • •
• • • • • • • •
• • • • • • • •



Basic Scenario of Cryptography

- Alice, who wants to say something privately to Bob
- Bob, who wants to hear from Alice
- Eve, the person who is trying to eavesdrop on their conversation.
Eve's goal:
 - Read M
 - Get the Key Alice is using, and read all messages encrypted using that key
 - Modify the content of the message in such a way that Bob will think Alice sent the altered message.
 - Impersonate Alice and communicate with Bob who thinks he is communicating with Alice.



(<https://flylib.com/books/en/1.581.1.188/1/>)

Passive

Active

Terminologies of Cryptography

- **Cryptography: the art of secret writing**

- The art of mangling information into apparent unintelligibility in a manner that allows a secret method of unmangling.

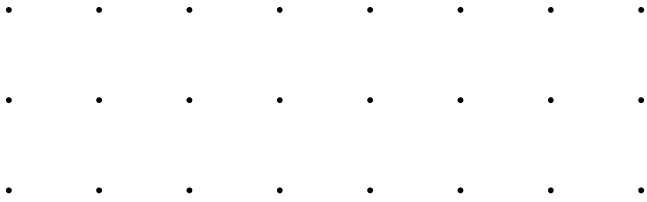
- **Related terminologies**

- **Cryptology:** The study of communication over non-secure channels, and related problems
- **Cryptography:** The process of designing systems that achieve secure communications.
- **Cryptanalysis:** Breaking such systems. (The techniques used to recover the secret information hidden in cryptographic systems)
- **Plaintext:** message to be sent, in readable form
- **Ciphertext:** message in coded form, unreadable without special information such as a key
- **Encrypt:** turn plaintext into ciphertext
- **Decrypt:** turn ciphertext back into plaintext

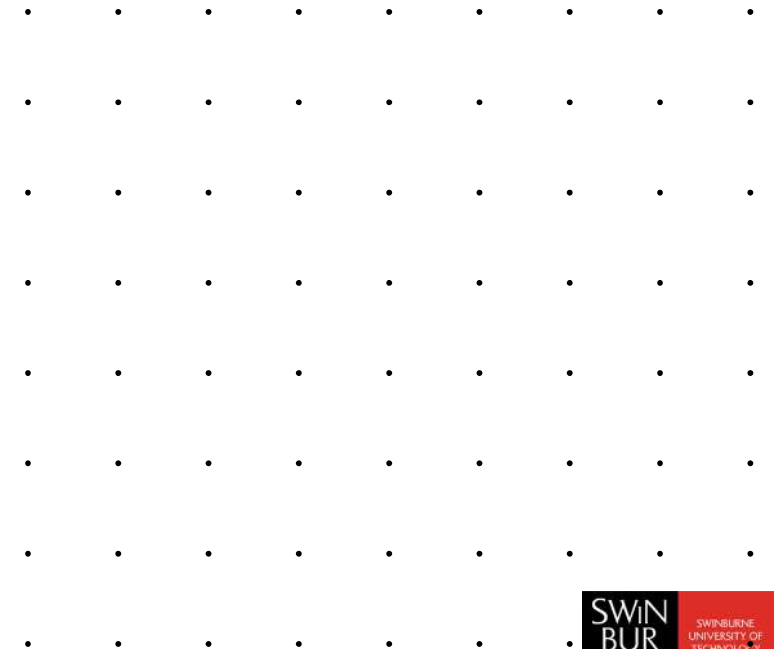
Cryptosystem attacks

- **Ciphertext-only attack**
- **Known-plaintext attack**
- **Chosen-plaintext attack**
- **Chosen-ciphertext attack**





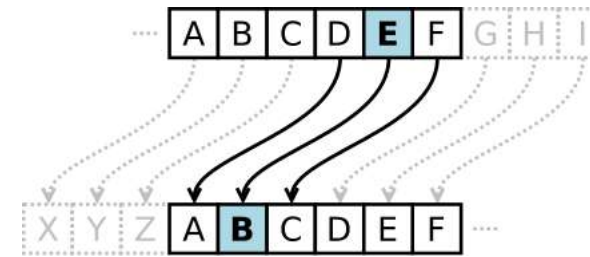
Symmetric Cryptography



Substitution Cipher

Substitution ciphers: swap one letter to another one.

- Simple substitution cipher
 - Simplest one is Caesar Cipher
 - Easy to break
- Monoalphabetic cipher
- Polyalphabetic cipher
- Code book cipher



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

Simple Substitution

Writing out the alphabet in some order to represent the substitution

- Write out a keyword
- Remove repeated letters
- Write all remaining letters alphabetically
- a.k.a monoalphabetic

Keyword: **zebras**

Plaintext alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ciphertext alphabet: ZEBRASCD EFGHIJKLMNOPQTUVWXY

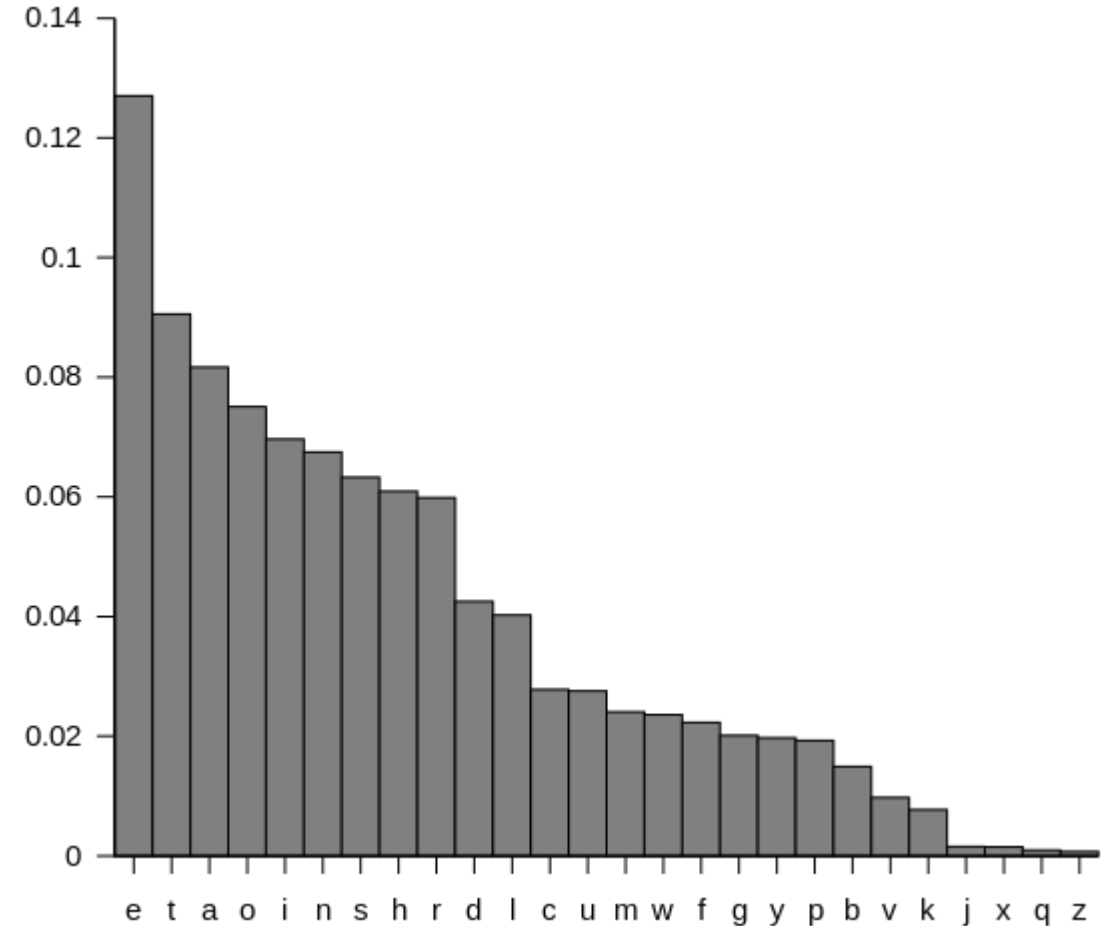
Message: **welcome to it security**

Encrypted: **VAIBLJA QL PABTOFQX**

FIVE LETTER: **VAIBL JAQLP ABTOF QXXXX**

Polyalphabetic

- Each character "rotated" by a different amount (1-25). The key is a look-up table (shared).
- mapping of each crypto-letter to plain-letter is repeated.
- Easy to crack using statistical methods (no shuffling) and knowledge of commonly used words.



Codebook cipher

- Each character "rotated" by a different amount (1-25). The key different for every instance of a letter. **Constantly-changing**
- mapping of each cipher-letter to plain-letter is rarely repeated.
- Very hard to crack if word groupings are preserved.
- Impossible to crack if punctuation removed, key totally random, no repetition.

	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

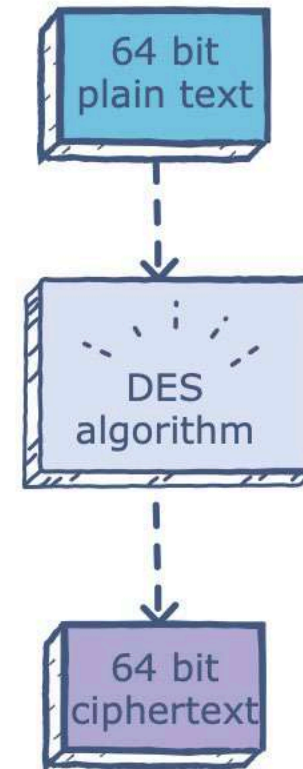
Data Encryption Standard (DES)

Definition

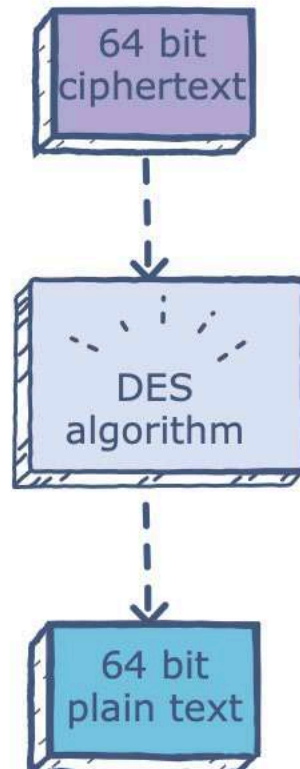
- Block Cipher
- symmetric key
- Out-dated now

History

- 1972: National Bureau of Standards begins search
- 1975: DES: Lucifer by IBM, modified by NSA Approved by NBS '76, ANSI '81
- renewed every 5 years by NIST
- now considered obsolete



Encryption

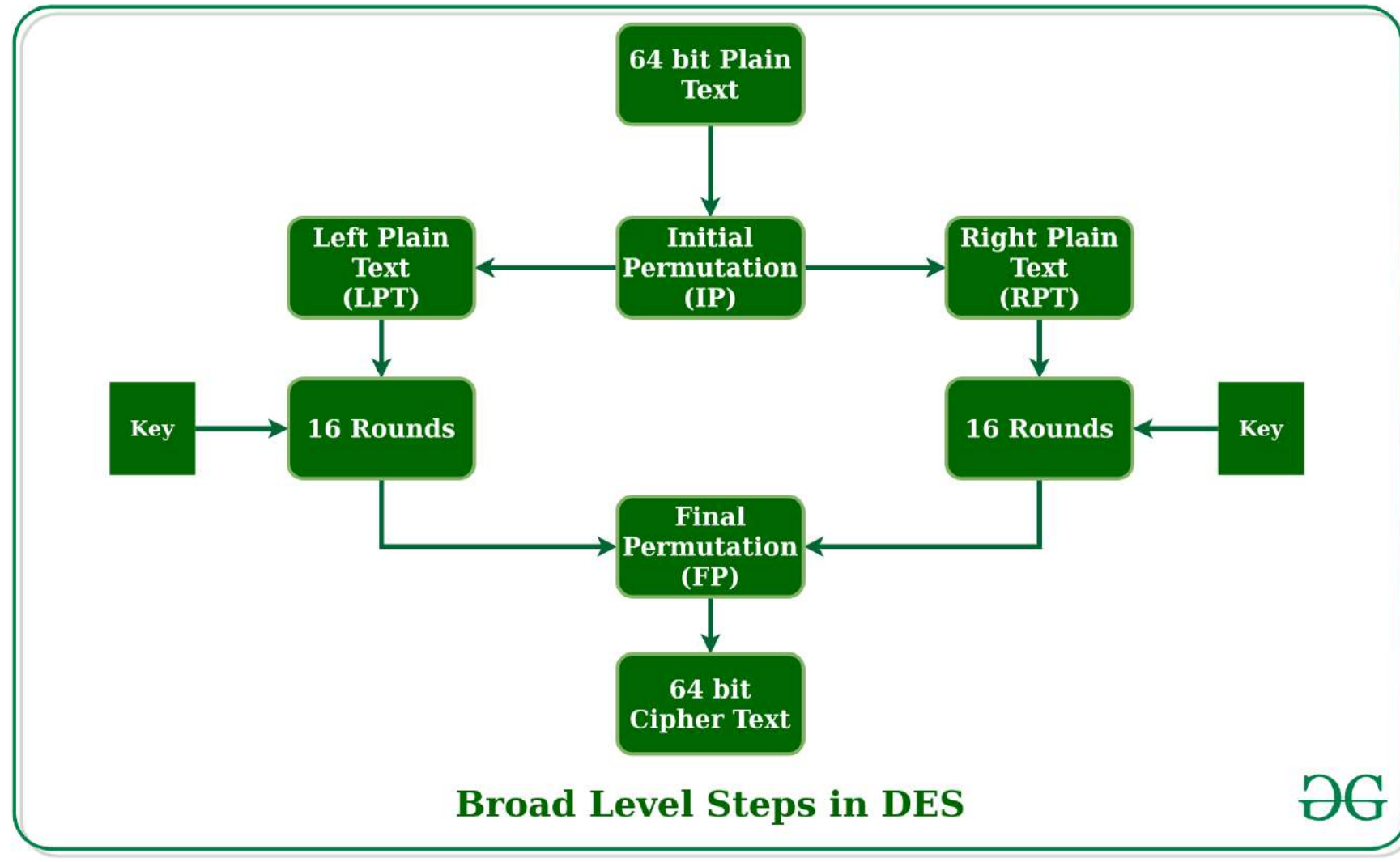


Decryption

DES

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64 bit plaintext input
- How secure is DES?
 - DES Challenge: 56-bit-key-encrypted phrase (“Strong cryptography makes the world a safer place”) decrypted (brute force) in 4 months
 - no known “backdoor” decryption approach
- making DES more secure
 - use three keys sequentially (3-DES) on each datum (triple DES)
 - use cipher-block chaining

How does DES work



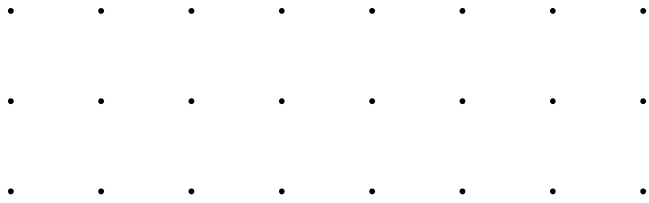
<https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/>

Advantages

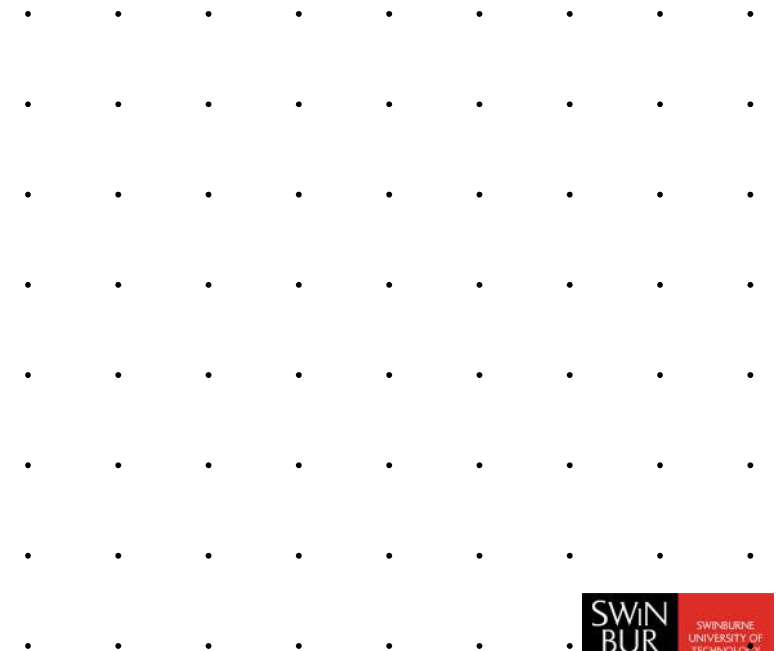
1. DES has been around a long time (since 1977), even now no real weaknesses have been found: the most efficient attack is still brute force.
2. DES is an official United States Government standard; the Government is required to re-certify, DES every five years and ask it be replaced if necessary.
3. DES is also an ANSI and ISO standard - anybody can learn the details and implement it.
4. Since DES was designed to run on 1977 hardware, it is fast in hardware and *relatively* fast in software.

Disadvantages

1. The 56-bit key size is the biggest defect of DES.
2. DES was not designed for software and hence runs relatively slowly.
3. As the technology is improving lot more day by day so there is a possibility to break the encrypted code, so AES is preferred than DES.
4. Only one private key is used for encryption as well as for decryption.



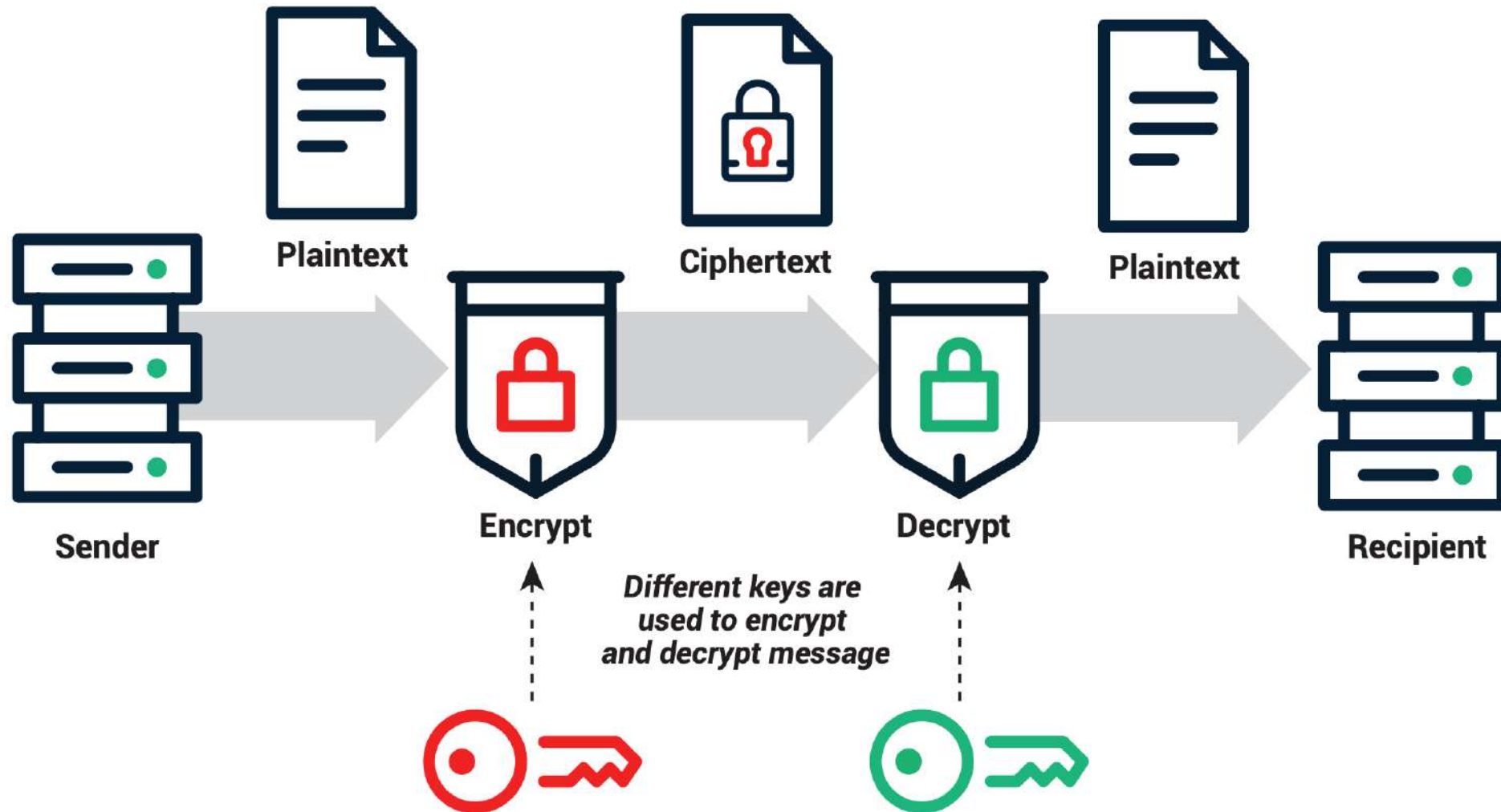
Public Key Cryptography



Public Key Cryptography

- **A.k.a. asymmetric cryptography**
- **Two keys – public and private**
- **Public key is shared**
- **Private key is kept secret**
- **Well suited for organizations**

How does it work

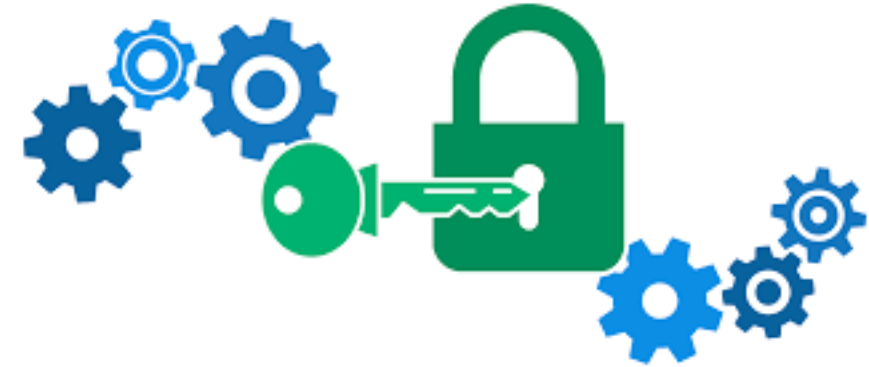


Public VS Private Key Cryptography

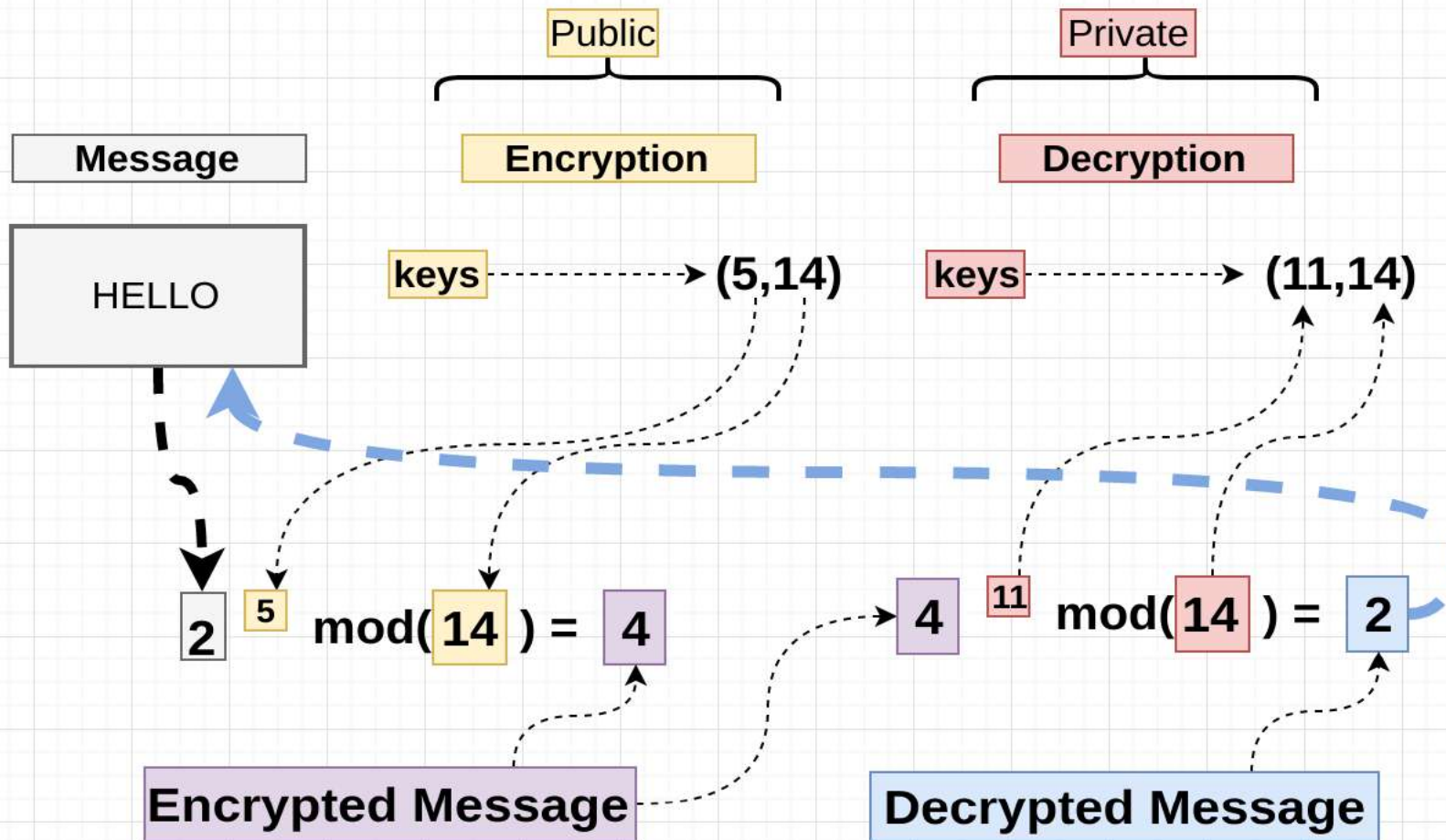
Public	Private
Slower	Faster
Two key	One keys
One is is public	Key is kept secret
Asymmetrical	Symmetrical
Sender and Receiver don't share key	Share the same key

RSA

- Rivest–Shamir–Adleman (RSA)
- Invented in 1977
- Asymmetric
- A bit slow
- Not commonly to directly encrypt user data
- OpenPGP, S/MIME, SSL/TLS



How does RSA work



RSA: Choosing keys

1. Choose two large prime numbers p, q .
(e.g., 1024 bits each)
2. Compute $n = p.q$, $z = (p-1)(q-1)$
3. Choose e (with $e < n$) that has no common factors with z . (e, z are "relatively prime").
4. Choose d such that $ed-1$ is exactly divisible by z .
(in other words: $e.d \bmod z = 1$).
5. Public key is (n, e) . Private key is (n, d) .
 $\underbrace{(n, e)}_{K_B^+} \quad \underbrace{(n, d)}_{K_B^-}$

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt bit pattern, m , compute

$c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

2. To decrypt received bit pattern, c , compute

$m = c^d \bmod n$ (i.e., remainder when c is divided by n)

Magic
happens!

$$m = (m^e \bmod n)^d \bmod n$$

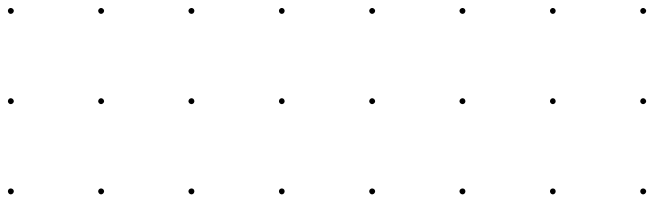
RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

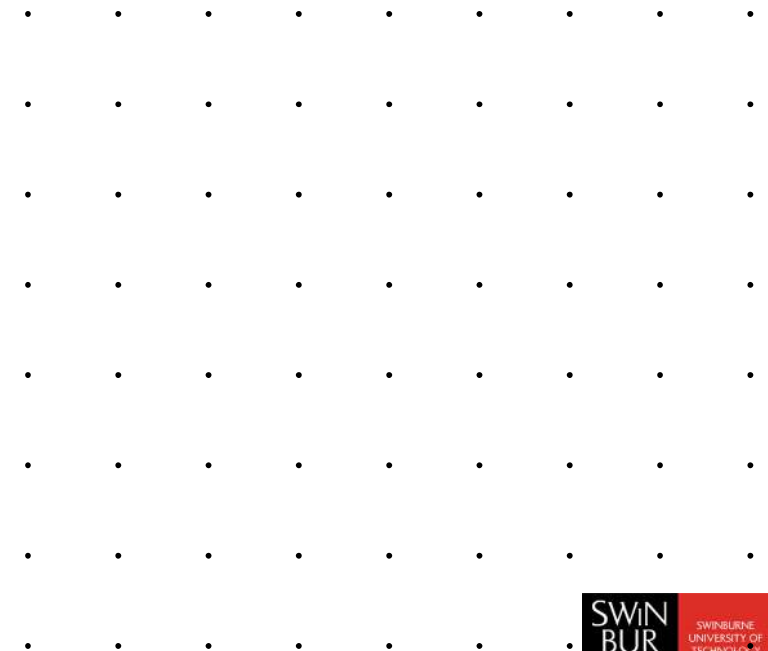
$e=5$ (so e , z relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z).

encrypt:	<u>letter</u>	<u>m</u>	<u>m^e</u>	<u>c = m^e mod n</u>
	I	12	1524832	17
decrypt:	<u>c</u>	<u>c^d</u>	<u>m = c^d mod n</u>	<u>letter</u>
	17	481968572106750915091411825223072000	12	I



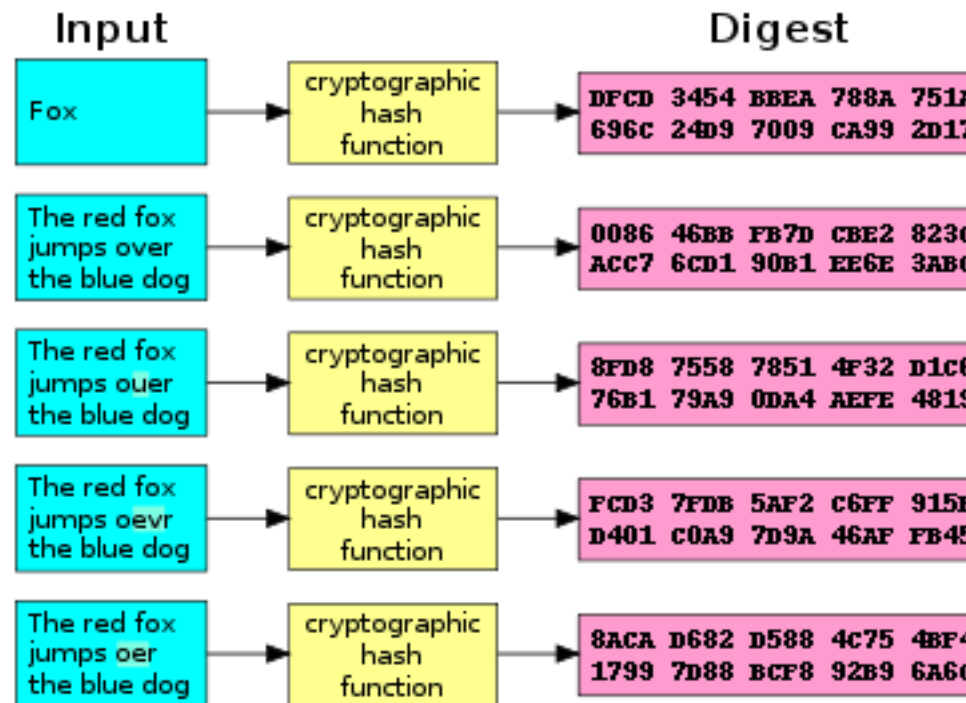
Hash Function



Hash Function

cryptographic hash function (CHF)

- a mathematical algorithm that maps data of arbitrary size to a bit array of a **fixed size**



https://en.wikipedia.org/wiki/File:Cryptographic_Hash_Function.svg

Hash Function Properties

- **Pre-Image Resistance**
 - hard to reverse a hash function
- **Second Pre-Image Resistance**
 - given an input and its hash, it should be hard to find a different input with the same hash
- **Collision Resistance**
 - it should be hard to find two different inputs of any length that result in the same hash

Design

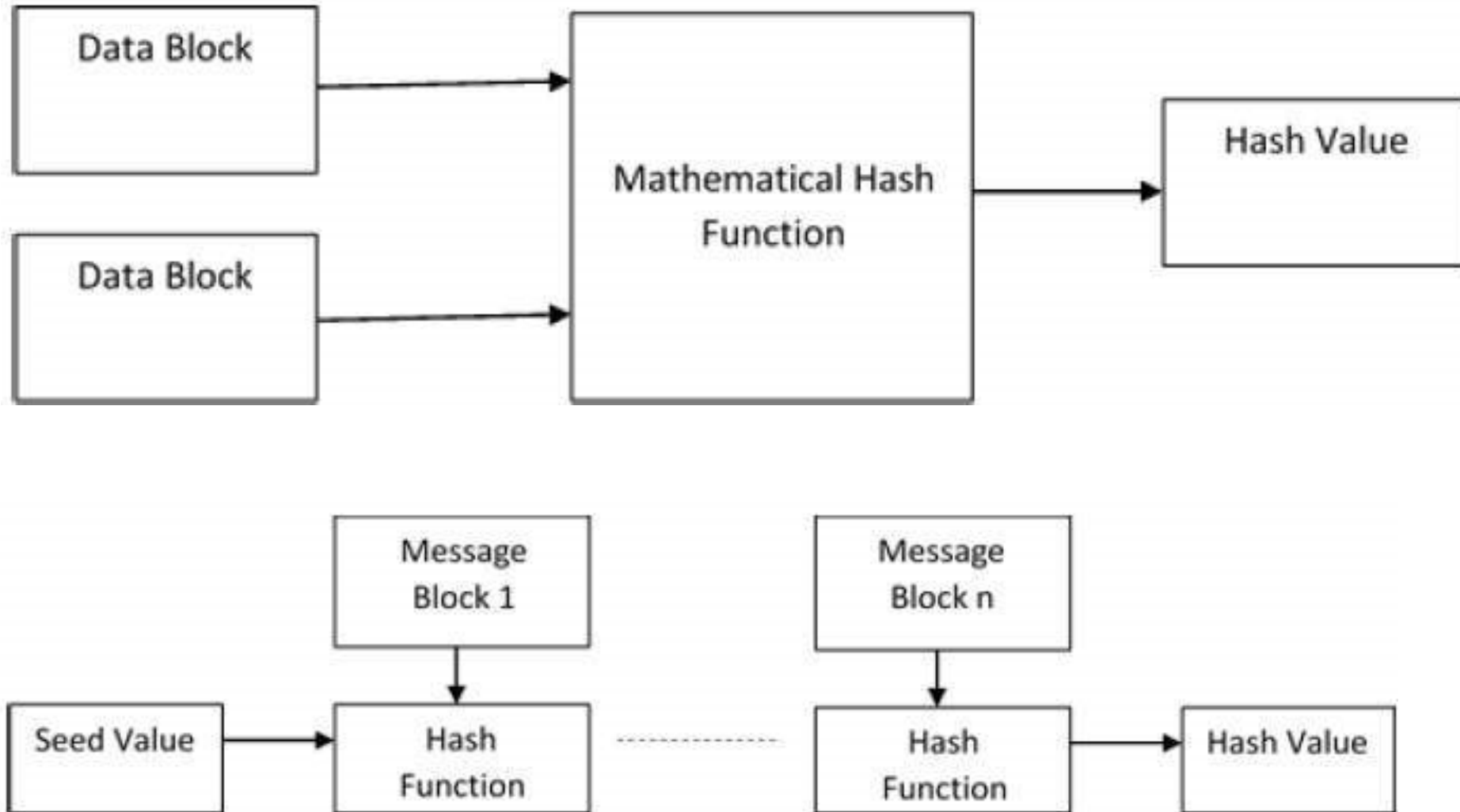


Figure: Schematic of hashing algorithm

Popular Hash Functions

- **Message Digest (MD)**
- **Secure Hash Function (SHA)**
- **RIPEMD**
- **Whirlpool**