Week 1

What is cryptography?

- The art and science of keeping data secure. It provides principles, means and methods to secure data from being accessed or modified by unauthorised parties either during its transmission or storage.

Prevent Unauthorised Access	Detect Unauthorised Modification
For example, an email from a sender to a receiver being intercepted by an eavesdropper through a communication channel.	The eavesdropper could intercept the email and alter its contents, sending the tampered data to the receiver, blocking the direct message path

- Cryptography scrambles the **cleartext/plaintext** via **encryption**, using a **cipher** with the help of a **key**, into **ciphertext**. During **decryption**, the receiver uses the key to decipher the ciphertext back into cleartext.

Terminology:

- A **cipher** is a **cryptographic algorithm** for performing **encryption** or **decryption**.
- The detailed operation of a cipher is controlled by both the algorithm and, *in each instance*, by a **key**.
- The message to be communicated in its' original form is **clear text / plain text**.
- **Encryption** is the act of scrambling the clear text into **ciphertext / encrypted text**, generally using a cipher and a key **Decryption** is the reversing of the encryption process.
- **Cryptographers** practice cryptography.
- **Cryptanalysis** is the art and science of breaking ciphertext, practiced by **cryptanalysts**.
- The branch of mathematics encompassing both cryptography and cryptanalysis is **cryptology**, practiced by **cryptologists.**
- Data and system intrusion and disruption events are called **cyber-incidents**.

Application of Cryptography

- In digital security; comprised of cybersecurity and Data / Information Security.
- Its potential for use as a tool for espionage and sedition has led many governments to classify it as a weapon and to limit or even prohibit its use and export.

Three Pillars of Cybersecurity (CIA Triad):

1. Confidentiality: Prevent unauthorised access to data / information and systems.
2. Integrity: Prevent unauthorised alteration to data / information and systems.
3. Availability: Ensure data and systems are consistently available when needed.

Cyber and Data Security Relationship:

- Data Security maintains the 3 pillars for data, whilst Cybersecurity does so for systems.
- Both overlap under the domain of digital data. Non-secure ICT systems will increase the risk of data security breaches.

Importance of Data Security:

- Data is the lifeblood of the digital economy, society and government. It is used and shared among various entities, and stakeholders for improved revenue, products and services.
- Data/Information Security is about protecting the organisation and personal data and preserving the privacy of individuals.
- Data must be protected from abuses, misuse and destruction by malicious parties. Cyber-criminals steal, destroy, misuse or abuse victim organisations' important data and sitrupt functioning systems.
- Leak of personal data can lead to undesired consequences to the privacy of individuals.
- Cyber-incidents incur monetary and reputational losses to affected organisations.

Application of Cryptography for the Three Pillars and Others:				
Confidentiality	Integrity	Availability	Authentication	Nonrepudiation
Encryption & Decryption	Message digest, Digital signature & PKI	No direct application of cryptography	It should be possible for the receiver of a message to ascertain its origin; intruder should not be able to disguise as someone else.	A sender should not be able to falsely deny later that they sent a message.

Passive Communication Attack:

- Targets message confidentiality. → Attacker does not attempt to change the message content. → Passive Attacks are hard to detect.
- Intercepts and captures the message, partial or full, via eavesdropping or monitoring of communication.
- Attacks can be of two types: *Release of Message Content* to unauthorised third parties; or if the message is encrypted, *Traffic Analysis on the Message* to try to figure out the clear text via cryptanalysis.
- Example: Spyware. A surveillance software that targets devices. Once installed without authorisation, can syphon data stored on the device to third parties.

Active Communication Attack:

- Unlike a passive attack, an active attack involves interception and modification to the contents of the original message.
- It has 3 sub-categories:
    - **Denial-of-Service (DOS) Attack**: Tries to prevent legitimate users from accessing services, which they are eligible for. E.g.: The attackers send large network traffic to overwhelm victim servers.
    - **Masquerading / Impersonation Attack**: An entity poses as another entity. E.g. Attacker impersonating a contact to send messages to you.
    - **Modification Attack**: E.g.: Attacker changing the recipient of a bill transfer to themselves.
        - **Replay Attacks:** Occur when a cybercriminal eavesdrops on a secure network communication, intercepts it, and then fraudulently delays or resends it to misdirect the receiver into doing what they want. E.g. An attacker eavesdrops and keeps an intercepted encrypted bank password, later using it to steal the account in the next session.
        - **Message Alternation Attack:** An attacker intercepts the network communication, captures and modifies the message, and replays it to their advantage. E.g. An attacker eavesdrops on bank communications, captures a message for a bill transfer and modifies its recipient to themselves and the amount transferred. Bank assumes the message is authorised and executes the transfer.

Week 2

Types of Cryptography:

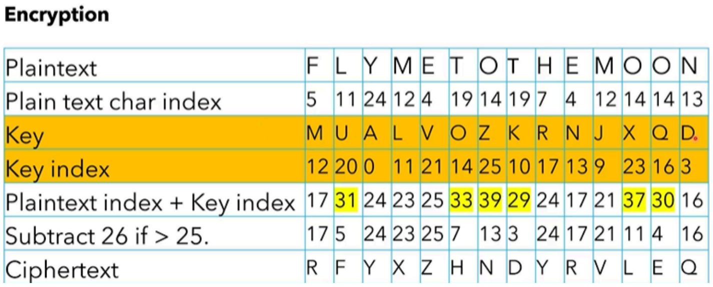
- Symmetric Key Cryptography: Uses the same key and cipher for both encryption and decryption.
- Asymmetric Key Cryptography Uses different keys for encryption and decryption.

Three Classical Symmetric Ciphers: Substitution Ciphers + Transposition Ciphers = Product Ciphers

Substitution Cipher (Confusion Technique):

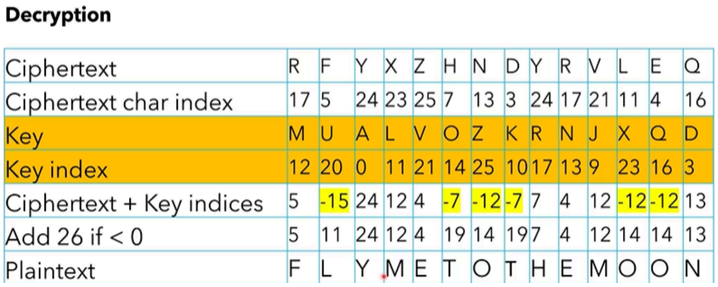
- Where each character in the plaintext is substituted for another character in the ciphertext.
- Receiver inverts the substitution on the ciphertext to recover the plaintext.

1) Caesar Cipher: Where each plaintext character is replaced by an alphabet 3 places down. Modified versions pick any places down the order → up to 25 tries to break it.
    a) *Advantage:* Easy to create a ciphertext. *Disadvantage:* Easy to break.
    b) Broken by: *Brute-force Method*, where you cycle though all 25 possible keys until the data makes sense. *Frequency Analysis Method*, where you count the occurrence of each letter in the ciphertext. 'E" is the most commonly written letter in english. The letter which appears the most in the ciphertext will probably be 'E', afterwards you can count the offset places and decrypt.
2) ROT13: Where every letter is rotated 13 places → Only works for even-numbered alphabets. For non-english alphabets, rotate by ½ the total letters..
3) Mono-alphabetic Substitution Cipher: Employs random substitutions.
    a) It uses a secret, fixed key which consists of the 26 shuffled letters and which plaintext letter they correspond to. The key can have any permutation or combination of the 26 letters (26!).
    b) Significantly harder to break than the Caesar Cipher
4) Vigenere Cipher: Encryption using a key table. If all characters of the keyword have been used, then the next keyword character cycles back to the start of the keyword
    a) E.g.: Keyword: CHIFFRE → Encrypting: VIGNERE becomes XPOJSVVG (Keyword 'C' and Plaintext 'V' intersect at ciphertext 'X').
5) Vernam Cipher (One-time Pad): A type of Vignere Cipher. The key is a random set of non-repeating characters. It's length is at least equal to the length of the original plaintext, and once a key is used, it is **never** used again.
    a) Highly secure.
    b) Suitable for small plaintext message, but impractical for large messages.



Encryption

Plaintext	F	L	Y	M	E	T	O	T	H	E	M	O	O	N
Plain text char index	5	11	24	12	4	19	14	19	7	4	12	14	14	13
Key	M	U	A	L	V	O	Z	K	R	N	J	X	Q	D
Key index	12	20	0	11	21	14	25	10	17	13	9	23	16	3
Plaintext index + Key index	17	31	24	23	25	33	39	29	24	17	21	37	30	16
Subtract 26 if > 25.	17	5	24	23	25	7	13	3	24	17	21	11	4	16
Ciphertext	R	F	Y	X	Z	H	N	D	Y	R	V	L	E	Q



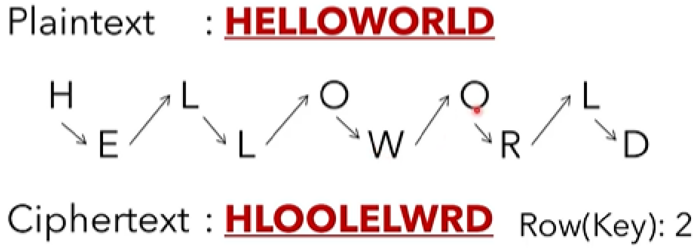
Decryption

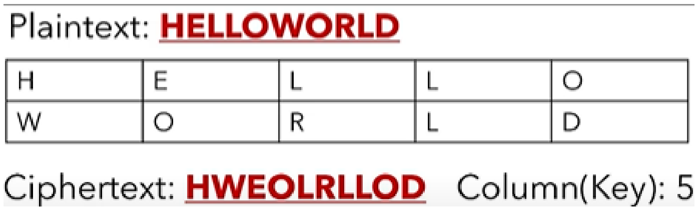
Ciphertext	R	F	Y	X	Z	H	N	D	Y	R	V	L	E	Q
Ciphertext char index	17	5	24	23	25	7	13	3	24	17	21	11	4	16
Key	M	U	A	L	V	O	Z	K	R	N	J	X	Q	D
Key index	12	20	0	11	21	14	25	10	17	13	9	23	16	3
Ciphertext + Key indices	5	-15	24	12	4	-7	-12	-7	7	4	12	-12	-12	13
Add 26 if < 0	5	11	24	12	4	19	14	19	7	4	12	14	14	13
Plaintext	F	L	Y	M	E	T	O	T	H	E	M	O	O	N

6) Homophonic Substitution Cipher: Single plaintext letters can be replaced by any amount of different ciphertext letters. E.g.: A key can replace plaintext character 'E' with ciphertext '7', 'Q', or '5' → Aims to defeat the Frequency Analysis Method → Generally harder to break than standard substitution ciphers.

Transposition Ciphers (Diffusion Technique):

- Transposition Ciphers rearrange the plaintext characters to produce the ciphertext.
- Spreads the influence of individual plaintext or key characters over as much of the ciphertext as possible.
    - Diffusion increases the redundancy of the plaintext by spreading it across rows and columns → an additional capacity to protect the plaintext.
    - Diffusion hides statistical relationships between the plaintext and ciphertext which makes cryptanalysis more difficult.

<table>
<tr><th>Rail Fence Technique</th><th>Columnar Transposition Technique</th></tr>
<tr>
<td>Write down the plaintext message as a sequence of diagonals. Then read the plaintext written instep 1 as a sequence of rows.

Plaintext : <b>HELLOWORLD</b>

Ciphertext : <b>HLOOLELWRD</b> Row(Key): 2</td>
<td>Write the plaintext message row-by-row in a rectangle of predefined size. Then read the message column-by-column. However, column roders may be random.

Plaintext: <b>HELLOWORLD</b>
<table><tr><td>H</td><td>E</td><td>L</td><td>L</td><td>O</td></tr><tr><td>W</td><td>O</td><td>R</td><td>L</td><td>D</td></tr></table>
Ciphertext: <b>HWEOLRLLOD</b> Column(Key): 5</td>
</tr>
</table>

Substitution (a) VS Transposition (b):

1a) Aims to protect the relationship between the statistics of *ciphertext and the key*.
1b) Aims to protect the statistical relationship between the *ciphertext and plaintext*.
2a) If one character in a key is changed, the calculation of most or all the ciphertext characters will be affected.
2b) If one bit of the plaintext is changed, then about half the bits in the ciphertext should change & vice-versa.
3a) Aims to increase the ambiguity of the ciphertext.
3b) Aims to increase 'redundancy', that is any patterns in the plaintext are not apparent in the ciphertext.

Binary Numbers: A binary number is a number expressed in the base-2 numeral system: 0 or 1.

Random Numbers: A random number is a number chosen by chance, randomly, from a set of numbers. They are used in cryptography as keys or basis to generate new keys.

Two types of random numbers:

- Truly Random - numbers exhibiting 'true' randomness, e.g.: radioactive decay.
- Pseudo-Random - numbers used in computer programs. They are generated in a predictable fashion using a mathematical formula.

Two important properties of a sequence of random numbers are:

- Unpredictability → Every number is equally probable every where within the range.
- Independence → The current value of a random variable has no relation with the previous values (no pattern).

Prime Numbers: Divisible by itself and 1. Very large prime numbers play a key role in cryptography, especially in asymmeetric cryptography, key distribution protocol, and others.

Key Management in Symmetric Cryptography and its Issues:

Key management is the hardest part of cryptography.

The security of a symmetric cryptosystem rests on the following two properties:

1. The strength of the cipher/algorithm
2. The secrecy and length of the key.

Issue: An attacker is usually armed with the knowledge of the ciphertext and the cipher. Only the actual key value remains the challenge for the attacker.

Brute-force crypanalysis or attack:

- An attacker tries all possible keys one by one in an assumed key-range (the total number of keys from smallest to larget available key)
- The bigger the key range (i.e. the longer the key), the more difficult brute-forcing it becomes.
  - 128 bit key = 2^128 possible keys. It is practically impossible to lauch a brute-force attack to obtain plaintext from a ciphertext encrypted with a modern cipher suite with a suifficiently long key.

Issue: All attacks in symmetric cryptography focus on obtaining the key first. Cryptanalysts often attack both symmetric and public-key cryptosystems through their key management. If cryptographically weak processes are used to generate keys, then the whole system is weak.

Key Generation:

- Password → Prone to various attacks (e.g.: dictionary attack)
- Solution: Random Keys → Generate the key bits from either a reliably random source or a cryptographically secure pseudo-random generator.

Key Distribution:

- Keys can be shared by in person meetings.
- Share partial keys through different medium (e.g.: E-mail ½ the key, text the other ½.)
- Above methods will not work for a large network of users, For n people to use symmetric key cryptography, n*(n-1)/2 keys are required. Public Key Cryptography is invented for this issue.

Key Verification: How can one ascertain that the key is from the right party?

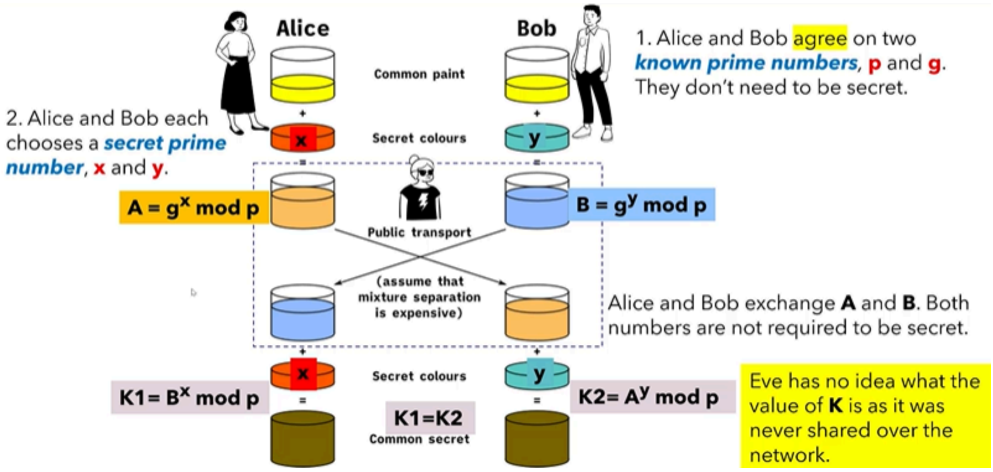
Key usage: The keys might only be authorised for use on a certain machine or at a certain time.

Key Storage: Strength of the storage from theft and tampering.

Updating Keys: To update keys once they were found compromised or broken into.

Key Exchange using Diffie-Hellman Protocol

- Aims to automate the key exchange securely. Sender and receiver can decide upon a key without meeting, splitting key and share in different mediums, etc. → these are common keys / shared secrets.
- No shared secrets are exchanged over the communication which can be eavesdropped on, but a secret key can be agreed upon.
- Applicable for key exchange; Not applicable for encryption/decryption.



1. Alice and Bob agree on two known prime numbers, p and g. They don't need to be secret.

2. Alice and Bob each chooses a secret prime number, x and y.

Alice and Bob exchange A and B. Both numbers are not required to be secret.

Eve has no idea what the value of K is as it was never shared over the network.

Let the agreed non-secret numbers be: p=11, g=7.
Alice's Number: Let x=3. So, A=7^3 modulo 11 =2
Bob-s Number: Let y=6. So, B=7^6 modulo 11 =4

Alice/K1 = 4^3 modulo 11 = 9
Bob/K2 = 2^6 modulo 11 = 9

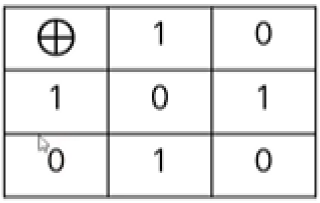
Therefore agreed key K = K1 = K2 = 9

Week 3

Two Types of Symmetric Ciphers:

Block Ciphers	Stream Ciphers
- Operates on the plaintext in groups of bits, (data) called blocks. - Data must be available before encryption starts. - E.g.: Encrypting a file saved on a storage medium.	- Operate on the plaintext a single bit or byte at a time. - Encrypt as data becomes available. - E.g.: Encrypting a character typed on a keyboard, one at a time.

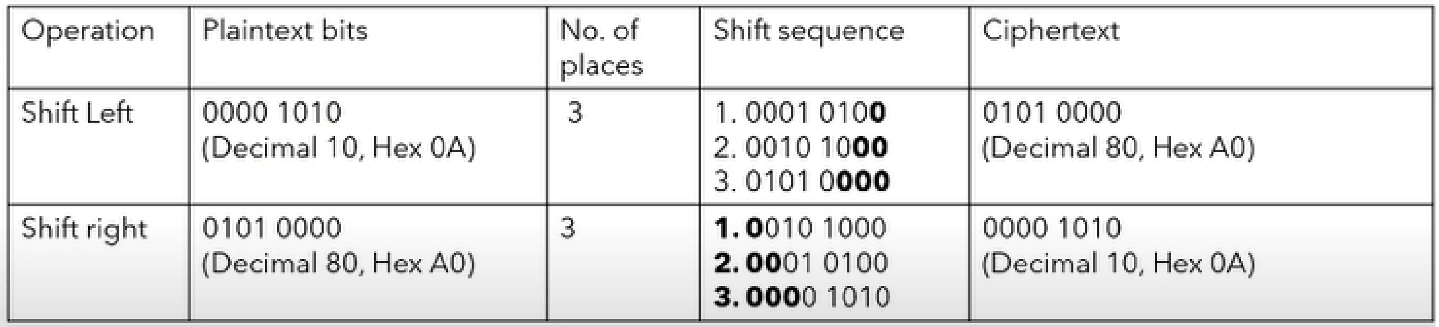
Before we delve into symmetric stream ciphers, we need to know: **Exclusive OR (XOR)** of bits (binary numbers)



⊕	1	0
1	0	1
0	1	0

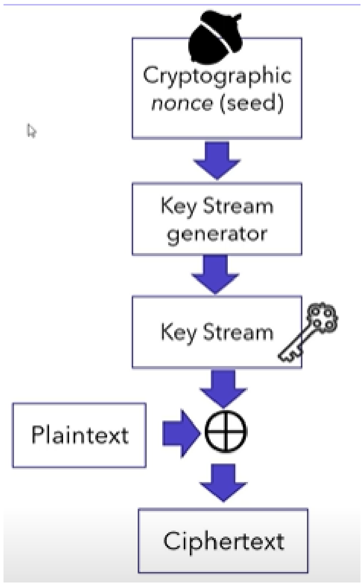
Essentially, if there is an intersect of two 1s or 0s, it is 0. Otherwise, for an intersection of mixed values, it is 1.

Not only this, but we need to know the workings simple **shift operations** such as Caesar or ROT13 ciphers. This time, instead of shifting alphabets to scramble plaintext, we will shift its bits to form the bit value of the ciphertext..



Operation	Plaintext bits	No. of places	Shift sequence	Ciphertext
Shift Left	0000 1010 (Decimal 10, Hex 0A)	3	1. 0001 010**0** 2. 0010 10**00** 3. 0101 0**000**	0101 0000 (Decimal 80, Hex A0)
Shift right	0101 0000 (Decimal 80, Hex A0)	3	**1. 0**010 1000 **2. 00**01 0100 **3. 000**0 1010	0000 1010 (Decimal 10, Hex 0A)

Stream Ciphers' Mode of Operation:



- Stream Ciphers use a pseudo-random keystream which is generated serially from a random seed value (Key/Nonce). The key stream generator is a random number generator.
- The pseudo-random keystream is XORed with the plaintext in a similar fashion to the one-time pad cipher.
- The original seed value serves as the cryptographic key to regenerate the keystream to decrypt the ciphertext.
- It is most effective in hardware implementation.

Key Size: A stream cipher generally makes use of smaller key sizes such as 128 bit keys. Like the one-time pad, if the key is exhausted before full encryption, the encryption continues restarting at the first value of the key.

This differs from a one-time pad, which is more secure. One of the extensively used stream ciphers is **RC4.** In Cryptool: Encrypt > Symmetric(Modern) > RC4 > Insert hexadecimal value of the key.

Example, Encrypting plaintext 'H' with key 'A': Binary value of H = 0100 1000, A = 0100 0001. XOR is applied to the two, making ciphertext = 0000 1001. For decryption, perform XOR using the key on the ciphertext. Ciphertext = 0000 1001, Key = A = 0100 0001, plaintext = 0100 1000 = 'H'.

Block Ciphers:

- Encrypt one block of plaintext after another. Each block has a typical size of 64 or 128 bits, as they must be large enough to preclude analysis, yet small enough to be workable.
- Data to be encrypted must be beforehand (already pre-available), such as files stored on a disk.
- Modern block ciphers are product ciphers that use both substitution and transposition methods.
- Are widely used to encrypt large amount of data.

These Ciphers have **Cipher / Algorithm Modes**, which dictate how the encryption of a plaintext is carried out.

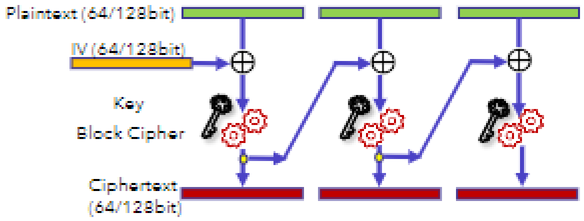
For Block Ciphers Only		For both Block and Stream Ciphers	
Electronic Codebook (ECB)	Cipher Block Chaining (CBC)	Cipher Feedback (CFB)	Output Feedback (OFB)

ECB Mode:

- The plaintext is divided into 64/128 bit blocks, then each block is independently encrypted with the same key.
- If plaintext blocks repeat (are the same), so do the corresponding cipher text blocks.
- Patterns in ciphertext may not be well hidden. ECB exhibits weak diffusion.
- ECB was originally designed to encrypt messages that never span more than a single block, such as to encrypt keys to distribute for other operations.

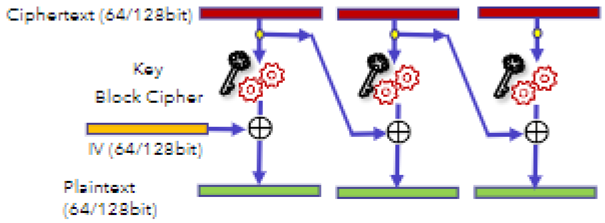
CBC Mode: A attempt to improve on ECB

- The encryption of each block is dependent not just on the key but also on the ciphertext of the previous block (except the first block)
- When the first block is encrypted, it's ciphertext is sent to the next plaintext block and XOR'ed with it before encryption. The result is then encrypted with the same key and sent to the next block to repeat.
- Because this CBC process starts after the first block, the Initialisation Vector(IV) is introduced to XOR with the first plaintext block, to make each message unique.



CBC Mode Encryption

- IV is a 64-bit block of random bits, a.k.a. Cryptographic nonce.
- IV is never resued under the same key.
- Distinct IV produces distinct ciphertextx even if the same plaintext is encrypted multiple times.
- IV usually does not need to be secret.



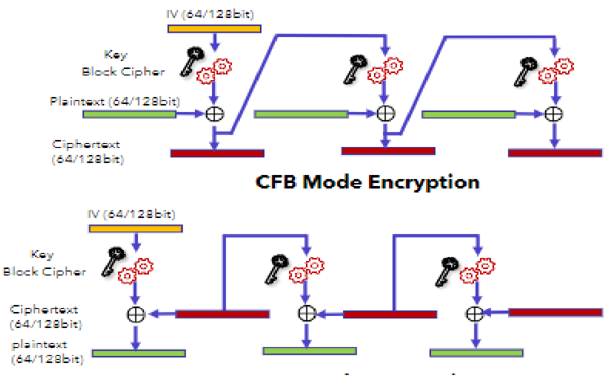
CBC Mode Decryption

Disadvantages of the CBC Mode:

- The encryption process is sequential and cannot be parallelised (meaning each block cannot be encrypted concurrently at the same time)
- Each message includes data from the previous block and hence needs to wait for encyption/decryption of the previous block to be completed first.
- But, the decryption process can be mostly parallel.
- Due to the "chaining" (including previous block data in current block encryption), any error in one block can propagate to the subsequent block.

CFB Mode: Similar to CBC

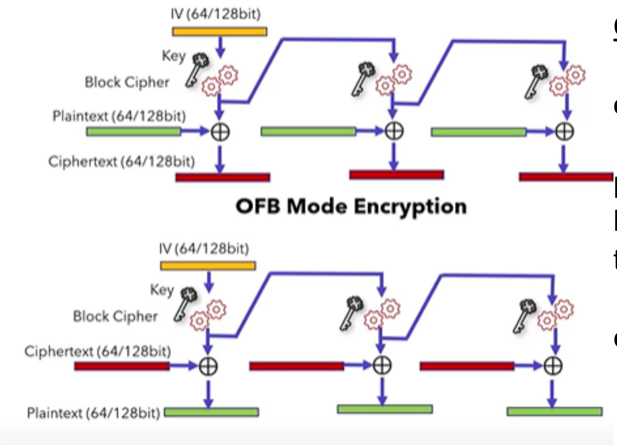
- The cipher text bits of the current block is fed to the next stage of encryption, and erorrs may propagate as such.
- Encryption process cannot be parallelised, but the decryption process can be.



CFB Mode Encryption

OFB Mode:

- Unlike CFB, only the output bits are fed to the next stage of encryption.
- On the first block, the IV is encrypted with the key and passed on before it is XOR'ed with the plaintext. Subsequent blocks then further encrypt the output bits and skip XOR'ing with the plaintext as well.
- OFB does not use the previous blocks of plaintext or ciphertext, so error in one block does not carry over to the next.



OFB Mode Encryption

Padding in Block Ciphers:

- Blocks are fixed in a block cipher, but messages/dagta come in a variety of lengths.
- ECB and CBC modes require that the final block be padded bore encryption.
- Example: A file with 3541 bytes = 5 64-bit blocks + 21 bytes. Padding needed = 64-21=35 bytes.

Padding Schemes:

- Add null (or 0) bytes to the plaintext to bring its length up to a multiple of the block size, but care must be taken that the original length og the plaintext can be recovered.
- Padding bits are removed during the decryption process.
- CBC-specific padding schemes are also used:
  - Append a byte with value 128 (hex 80), followed by as many zero bytes as needed to fill the last block, or
  - Pad the last block with n bytes all with value n.

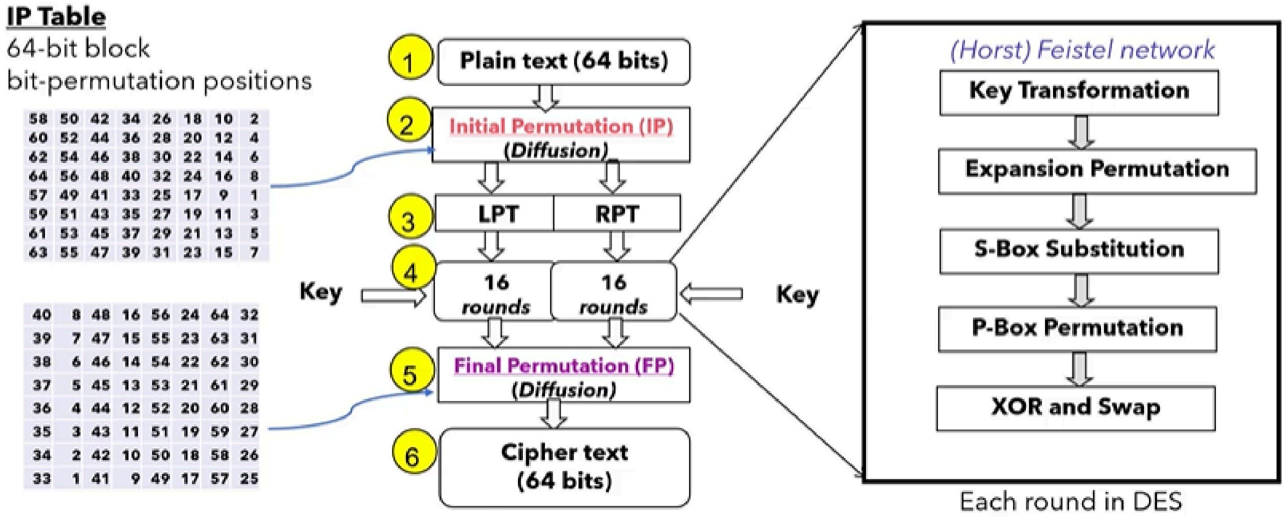
Block Ciphers	**Stream Ciphers**
The modes used are ECB and CBC.	The modes used are CFB and OFB.
Uses confusion as well as diffusion. Works on transposition techniques like rail-fence, columnar transposition, etc.	Only uses confusion. Works on substitution techniques like Caesar Cipher, etc.
Converts the plain text by taking a block at a time. The usual size of the block can be 64 or 128 bits.	Converts the text bt taking one byte of plaintext at a time. 1 bye (8 bits) at a time is more common.
Simple but slow as compared to the latter.	Fast but more complex as compared to the former.
Used by nearly all block ciphers. ● DES, 3DES, AES, IDEA, Blowfish, RC5	Used for some data-in-transit encryption. Insome TLS suites, RC4 for wireless networks, A5 for cellular networks, etc.

<u>Data Encryption Standard:</u> DES is a symmetric block cipher; it encrypts data in 64-bit blocks. It uses only standard arithmetic and logical operations on the numbers of blocks, and was first brute-forced in June 1998.

- Key length is 56 bits. It works with 64-bit blocks, but every least significant bit of 8 bytes is used for error checking and ignored.
- DES is a product cipher of confusion and diffusion. It encrypts through 16 rounds of substitution followed by a permutation based on the key.
- All security rests in the key. The cipher is public domain.

<u>Padding of DES:</u>

- If the message does not end on a block boundary, add a 1 bit, followed by enough 0 bits to fill out the block.
- If the message ends on a block boundary, a whole block of padding will be added so that the last message block is not mistakened as padding.



IP Table
64-bit block
bit-permutation positions

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Each round in DES

<u>Summary of DES Encryption:</u> Plaintext 64-bit block is undergoes Initial Permutation with a fixed 64-bit table. Then, it is divided into 2 32-bit halves and sent to the left and right parts. Each part undergoes 16 rounds of additional encryptions using the key. The final results will undergo Final Permutation using a fixed 64-bit table.

- The IP Table (which is fixed) reads from left ro right, up to down and its values are the positions of a 64-bit block. Take the first bit in a block of plaintext, and read it against the first value in the IP Table, 58. This means the first bit of plaintext will be transferred from the 1st position to the 58th position.
- Cryptool: Individual Procedure > Visualisation of Algorithm > DES

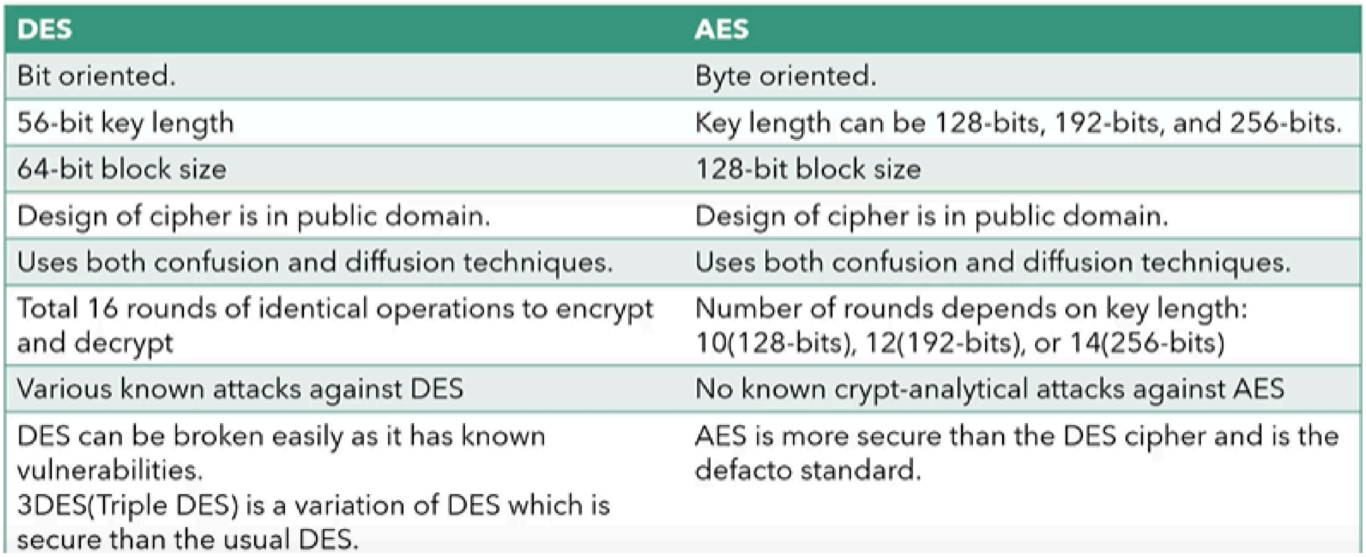
<u>DES Updates:</u>

- Double DES: Perform DES twice with two different keys. Not widely used.
- Triple DES / 3DES: Uses a 'key bundle' of 3 unique DES keys. K1 encrypts plaintext → K2 decrypts ciphertext, but as its not the same as K1, the decrypted text is scrambled → K3 then further encrypts the scrambled decrypted text.
- In two-key versions, K1 and K3 are the same. They're less secure and less common.

<u>Advanced Encryption Standard:</u> AES, a.k.a. Rijndael, is a symmetric block cipher that encrypts data in 128 bit blocks.

- Key length can be either 128 (most common), 192 or 256 bits.
- AES is a product cipher of confusion and diffusion. The key size used specifies the number of transformation rounds that encrypt the plaintext. 128 = 10 rounds, 192 = 12, and 256 = 14. Hence, 128-bit is most common as it saves resources.
- All security rests in the key. The cipher is public domain.
- Unlike DES, AES manipulates bytes (8bits) instead of bits. It views a 16-byte plainntext block on a 4x4 2d array of bytes. AES transforms the bytes, columns, and rows of this array to produce the ciphertext.

- AES is the most used cipher today, it is yet to be broken.
- AES is secure because all output bits depend on all input bits in some complex, pseudorandom way.
    - To achieve this, AES designers chose each component for a reason:
        - Mix Columns: Diffusion: Perform matrix multiplication. Each column is multipled by a specific matrix.
        - Shift Rows: Diffusion: Each row is shifted a particular number of times.
        - Substitute Bytes: Consumption: Each byte is substituted by another byte.
    - This composition is proven to protect AES against whole classes of cryptanalytic attacks.



DES	AES
Bit oriented.	Byte oriented.
56-bit key length	Key length can be 128-bits, 192-bits, and 256-bits.
64-bit block size	128-bit block size
Design of cipher is in public domain.	Design of cipher is in public domain.
Uses both confusion and diffusion techniques.	Uses both confusion and diffusion techniques.
Total 16 rounds of identical operations to encrypt and decrypt	Number of rounds depends on key length: 10(128-bits), 12(192-bits), or 14(256-bits)
Various known attacks against DES	No known crypt-analytical attacks against AES
DES can be broken easily as it has known vulnerabilities. 3DES(Triple DES) is a variation of DES which is secure than the usual DES.	AES is more secure than the DES cipher and is the defacto standard.

Cryptanalysis Concepts and Types: An attempted cryptanalysis is known as an attack. It is assumed that the attacker has complete details of the cryptographic algorithm and implementation. Six possible attacks will be covered. **(Adaptive-Chosen-Plaintext Attack is included, but not talked about)**

Ciphertext-only Attack:

- The attacker has the ciphertext of several messages, all of which have been encrypted using the same cipher.
- The attacker's job is to recover the plaintext of as many messages as possible, or better yet to deduce the encryption key(s) used to decrypt other messages.

Known-plaintext Attack:

- The attacker not only has access to the ciphertext of several messages, but also to the plaintext of those messages.
- The attacker is to deduce the encryption key(s), or an algorithm to decrypt any new messages to decrypt new messages encrypted using the same key(s)

Chosen-plaintext Attack:

- The attacker not only has several ciphertexts and their plaintexts, but can also choose a plaintext and request the cryptographic algorithm to encrypt it for them.
- This is more powerful than a known-plaintext attack, because the attacker can send an inherently vulnerable plaintext block to be encrypted, such that it reveals more information about the key.
- The attacker is to deduce the encryption key(s), or an algorithm to decrypt any new messages to decrypt new messages encrypted using the same key(s)

Chosen-ciphertext Attack:

- The attacker can choose different ciphertexts to be decrypted and has access to the their plaintexts.
- The attacker’s job is to deduce the encryption key(s).

Rubber-hose Attack:

- The attacker threatens, blackmails, or tortures someone until they give them the key.
- When bribery is involved, it is sometimes referred to as a purchase-key attack.
- These are all very powerful attacks and often the best way to break an algorithm

Week 4

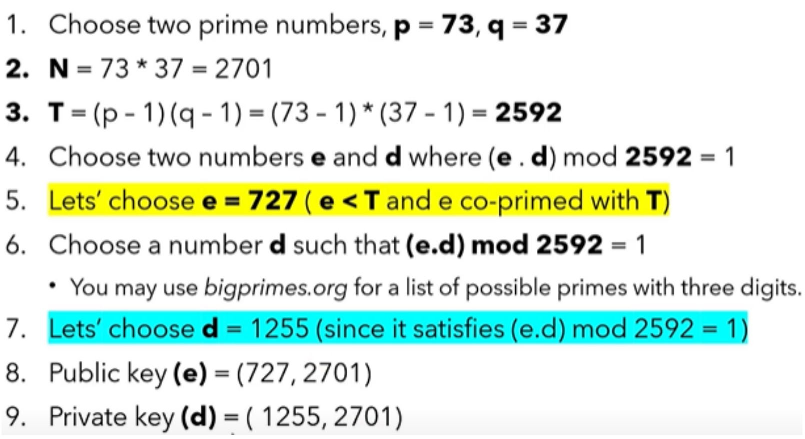
Asymmetric Cryptography, a.k.a Public-key Cryptography: Used to exchange symmetric keys securely.

- A process that uses a pair of related keys: 1 Public and 1 Private key encrypt and decrypt a message and protect it from unauthorised access or use.
- Public Keys are published and available upon request.

Message Origin Authentication (Non-repudiation)	**Message Confidentiality**
Not for confidentiality; ensures the message comes from a legitimate sender. : Message undergoes encryption with the sender's private key. Recipient retrieves the sender's public key to decrypt. If the public key can decrypt the message, it means it was encrypted with the sender's private key only they know, confirming its them.	Sender retrieves Recipient's public key to encrypt his message. Recipient then decrypts the ciphertext with their private key. Anyone can send a message to the recipient, the sender cannot be confirmed. Thus, only keeping the message confidential.

ASCII (American Standard Code for Information Interchange): A 7-bit character code where every byte represents a unique character.

Modulo Arithmetic Operation: Returns a division's remainder or signed remainder after dividing a number.



1. Choose two prime numbers, **p** = **73**, **q** = **37**
2. **N** = 73 * 37 = 2701
3. **T** = (p - 1)(q - 1) = (73 - 1) * (37 - 1) = **2592**
4. Choose two numbers **e** and **d** where (**e** . **d**) mod **2592** = 1
5. Lets' choose **e = 727** (**e < T** and e co-primed with **T**)
6. Choose a number **d** such that **(e.d) mod 2592** = 1
   - You may use *bigprimes.org* for a list of possible primes with three digits.
7. Lets' choose **d** = 1255 (since it satisfies (e.d) mod 2592 = 1)
8. Public key **(e)** = (727, 2701)
9. Private key **(d)** = (1255, 2701)

RSA Algorithm: Is based on the fact that it is easy to find and multiply large prime numbers together to make a product, but extremely difficult to find the factors from the results.. If the factors can be found, it can be used for key-pair. Most widely accepted public-key solution.

RSA Algorithm Steps:

1) Choose two very large primes, p & q.
2) Compute N = p * q
3) Compute Euler Totient T ; **T = (p-1)(q-1)**
4) Choose two numbers e and d, where **(e*d) mod T = 1**, such that: 1 < e < T and e co-primed with T.
5) Publish public key (e, N).

Encryption and Decryption with RSA Algorithm: To encrypt, each plaintext character is converted to a representative integer value, usch as ASCII code.

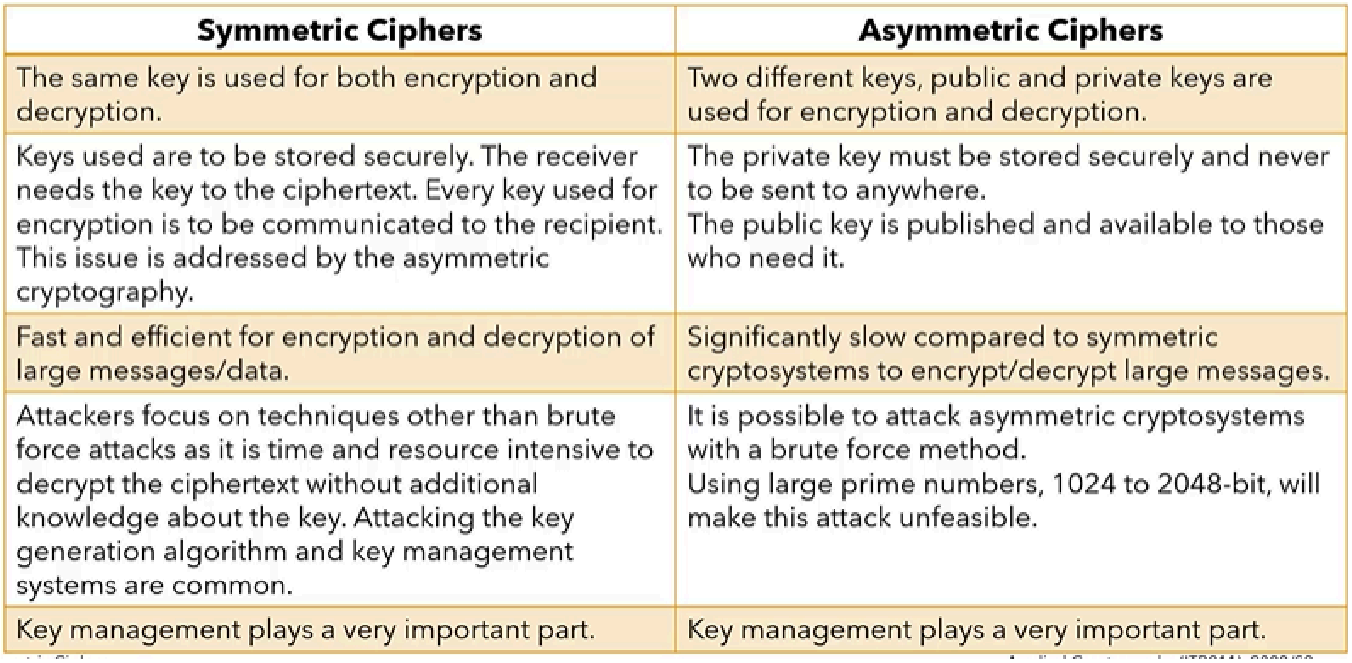
- Encryption = Plaintext^e % N
- Decryption = Ciphertext^d % N
- E.g. with previous keys: Plaintext 'H' = ASCII Code 72 → (72^727)%2701 = 1167 → 1167 will be ASCII value of the ciphertext character.

Security of the RSA Algorithm:

Attack Type	**Description**
Factorisation Attack	If the attacker can find p & q, then they can find the private key.
Key(s) Attack	Small encryption and decryption key values, e & d are prone to brute-force attack.
Revealed Decryption Exponent Attack	If the attacker can find decryption key value, d, all current and future ciphertexts are in danger of being revealed.

Other Asymmetric Algorithms:

- Elliptic curve cryptography (ECC): A newer algorithm that offers shorter keys that achieve comparable strengths compared to longer RSA keys. It is prone to key-based attacks that use quantum computing.
- El gamal Cryptography: Based on the Diffie-hellman algorithm, it also works based on key generation, encryption and decryption.
- Digital Signature Algorithm (DSA) & ECDSA: DSA implements the Digital Signature Standard (DSS) and is used for digital structures only.



Symmetric Ciphers	Asymmetric Ciphers
The same key is used for both encryption and decryption.	Two different keys, public and private keys are used for encryption and decryption.
Keys used are to be stored securely. The receiver needs the key to the ciphertext. Every key used for encryption is to be communicated to the recipient. This issue is addressed by the asymmetric cryptography.	The private key must be stored securely and never to be sent to anywhere. The public key is published and available to those who need it.
Fast and efficient for encryption and decryption of large messages/data.	Significantly slow compared to symmetric cryptosystems to encrypt/decrypt large messages.
Attackers focus on techniques other than brute force attacks as it is time and resource intensive to decrypt the ciphertext without additional knowledge about the key. Attacking the key generation algorithm and key management systems are common.	It is possible to attack asymmetric cryptosystems with a brute force method. Using large prime numbers, 1024 to 2048-bit, will make this attack unfeasible.
Key management plays a very important part.	Key management plays a very important part.

Symmetric Key Exchange using the Asymmetric System:

1) Bob encrypts the message using the AES cipher with a random 128-bit key.
2) Bob encrypts the encryption key above with Alice's public key.
3) Bob sends the encrypted message and encrypted key to Alice.
4) Alice decrypts the encrypted AES key using her private key.
5) Alice decrypts the message using the AES algorithm with the key obtained from the above step.

Week 5

Message Digest / Hash / Checksum: Is a fixed-length sequence of digits uniquely representing a message. It is like a message's fingerprint.

- The message digest is used to verify the integrity of the message against unauthorised or unintentional modifications. Any changes to the original message will result in a different message digest.
- Message digest algorithms are public domain.

Desirable Properties of a Message Digest Function:

- A hash function always hashes a message to the same hash value.
- A hash function is irreversible.
- A GOOD has function hashes two different messages to two different hash values, making it resistant to a "collision"

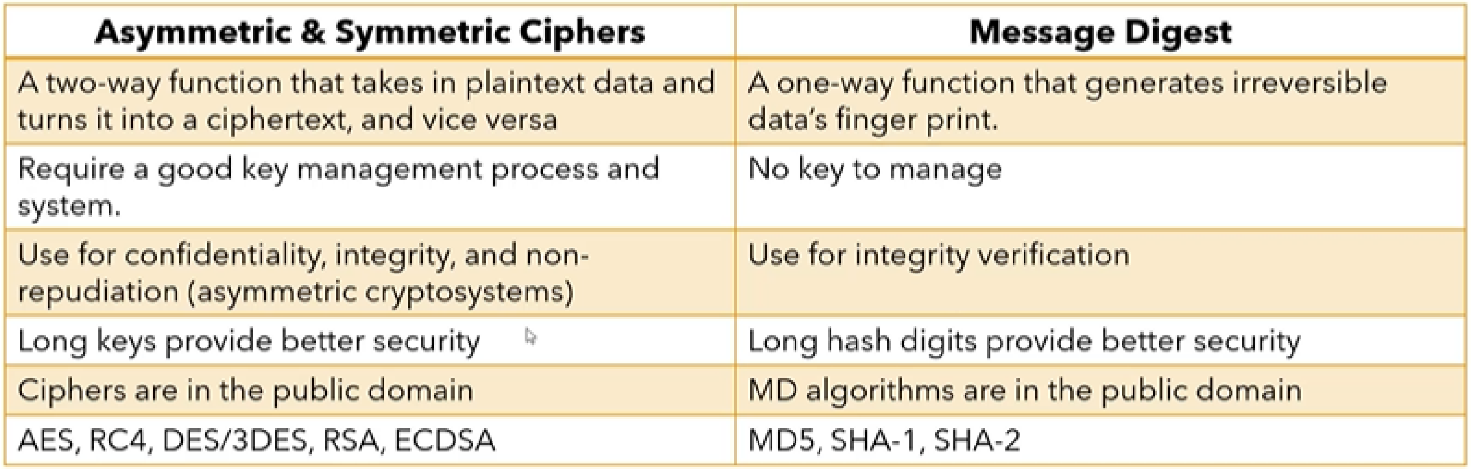
Message Digest Collision:

- A Collision occurs when two messages produce the same message digest.
- A Collision Attack on a hash function involves finding a nonsensical message that will produce the same hash value as the original message.

Collision Prevention: Use the message digest function with a longer message digest length. E.g.: The chance of collision for a 160-bit has would be 2^160.

Common Message Digest Algorithms:

- Message Digest 5 (MD5): Already broken, not used today for cryptographic purposes. It produces a 128-bit hash value.
- Secure Hashing Algorithm 1 (SHA1): Produces 160-bit hash values.
- Secure Hashing Algorithm 2 (SHA2): There is a variety of SHA2 Algorithms:
  - SHA256: Produces a 256-bit hash value.
  - SHA512: Produces a 512-bit hash value.



Asymmetric & Symmetric Ciphers	Message Digest
A two-way function that takes in plaintext data and turns it into a ciphertext, and vice versa	A one-way function that generates irreversible data's finger print.
Require a good key management process and system.	No key to manage
Use for confidentiality, integrity, and non-repudiation (asymmetric cryptosystems)	Use for integrity verification
Long keys provide better security	Long hash digits provide better security
Ciphers are in the public domain	MD algorithms are in the public domain
AES, RC4, DES/3DES, RSA, ECDSA	MD5, SHA-1, SHA-2

Authenticity verifies the sender's identity and the source of a message. **Non-repudiation** is a procedural, legal concept that proves the legitimacy of a message transfer by *providing undeniable evidence of both authenticity and integrity.*

Message Authentication Code (MAC): Is a form of a hash, unlike it, it contains key materials inside the message to get the digest.

- Used to ensure the integrity and authenticity of the message.
- Sender and receiver must share a secret cryptographic key (a shared secret).
  - Note: The key is not for encryption/decryption. Key bits are mixed with the message to process the MAC.
- MAC is weak on non-repudiation as multiple parties may possess the same key.

- MAC Algorithms: Key Generation Algorithm (Select a random cryptographic key) → Signing Algorithm (Returns a MAC from the message and the key) → Verifying Algorithm (Verify the message's authenticity and integrity)

MAC Creation + Verification Steps:

1) Bob calculates MAC 1 using a cryptographic key and plaintext message.
2) Bob sends the message and MAC 1 to Alice.
3) Alice calculates MAC 2 using the same key and message. Then compares MAC 1 with MAC 2.
    a) If the are equal, the message is good. Otherwise, the message is rejected.

MAC Requirement with Scenario:

Requirement: The key must be known only to the sender and receiver to support authenticity.

Scenario: If an attacker changes a message but not the MAC. the receiver will calculate a different MAC from the message and conclude that message integrity has been violated. **The attacker does not have the cryptographic key to re-compute and replace the MAC.**

MAC Example: SipHash is an add-rotate-xor based pseudo-random MAC function optimised for short inputs. It computes a 64-bit MAC from a message and a secret 128-bit secret key.

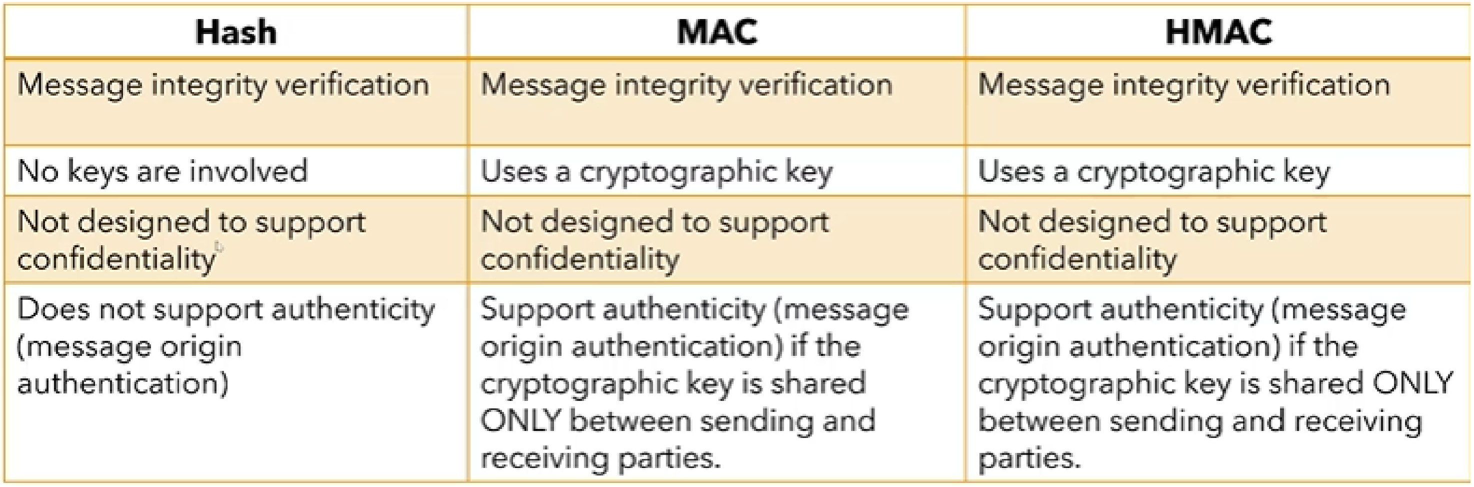
- MAC algorithms can also be constructed from hash functions (HMAC) or some of the block cipher algorithms

Hash Message Authentication Code (HMAC) Requirements + Scenarios:

- HMAC is a specific type of MAC involving a cryptographic hash function and a secret cryptographic key.
- Like MAC, it is used to verify a message's data integrity and authenticity if only the sender and recipient know the shared cryptographic key.
- Some challenges of HMAC with the cryptographic key exchange are the same as MAC.
    - Multiple recipients using multiple keys need a key management system.
    - Multiple recipients using a single shared key face authentication problems.

Using HMAC in Message Communication:

1) Cryptographic Key is created and put together with the message through a hash algorithm to generate the HMAC 1.
2) Bob sends the message and HMAC 1 to Alice.
3) Alice runs the same key and message through the same hash algorithm to generate the HMAC 2, then compares them.
    a) If they are equal, the message is authentic and integrity is preserved. Otherwise, reject.



Hash	MAC	HMAC
Message integrity verification	Message integrity verification	Message integrity verification
No keys are involved	Uses a cryptographic key	Uses a cryptographic key
Not designed to support confidentiality	Not designed to support confidentiality	Not designed to support confidentiality
Does not support authenticity (message origin authentication)	Support authenticity (message origin authentication) if the cryptographic key is shared ONLY between sending and receiving parties.	Support authenticity (message origin authentication) if the cryptographic key is shared ONLY between sending and receiving parties.

Physical Signatures:

1) Represent Contract Agreements that codify, specify and clarify transactions and relationships.
    a) It conveys the identities of parties in a contract.
    b) The definite acceptance of the contract by the parties,
    c) And the applicability of the terms of the contract with the parties.
2) Are Anchors of Trust and Evidence: That identify the signatory and confirmation of the contents of a document.
3) Act as a verification that whatever document is signed is true, real, and valid.

Digital Signature: Is a technique that binds an entity to the digital data or digital document.

- Is used to sign the Digital Certificate.
- Digital Signatures are a mathematical way of verifying the authenticity of digital messages to prevent forgery and tampering in the communication process.
    - The recipient knows who created the digital document and that it has not been altered from the time the sender created it.
- Digital Signature assures: Authenticity, Integrity, and Non-repudiation

Digital Signature Usage in RSA:

Creating a Digital Signature:

1) The sender uses a message digest algorithm to calculate the message digest of the plaintext document.
2) The sender encrypts Message Digest 1 using his private key to get the encrypted message digest, which is the digital signature.
3) The document and digital signature is sent to the receiver.

Verifying a Digital Signature:

4) The receiver uses the same message digest algorithm to calculate Message Digest 2 of the plaintext document.
5) The receiver uses the sender’s public key to decrypt the digital signature to get MD1.
6) Receiver compares MD1 with MD2. If both are the same, receiver can trust and accept the plaintext. If not, it is rejected.

Physical Signature	**Digital Signature**
Both provide the security services of authentication, data integrity and non-repudiation.	
Physical part of the document	Not physically attached to message, algorithm needs to bind signature to message.
Copy of signed paper document can be distinguished from an original.	Copy of signed message is identical to the original.
Verified by comparing it to other authentic signatures.	Verified using publicly known verification algorithm.
Slow verification process	Fast verification process
Simple and easy to understand.	Involves complex cryptographic algorithms and mathematical computation

Adding Confidentiality: Message Encryption and Signing

1) Bob encrypts the plaintext message using a strong random symmetric key and a symmetric cipher.
2) Bob encrypts the symmetric key from step 1 using Alice's public key.
3) Bob generates the message digest of his plaintext message using a strong message digest algorithm (SHA-1 or SHA-2).
4) Bob encrypts the message digest using his private key. This becomes the digital signature of the message.
5) Bob sends ciphertexts of the message, symmetric encryption key (step 2), and the digital signature (step 4) to Alice.
6) Alice decrypts the ciphertext of the symmetric key using her private key.
7) Alice decrypts the message ciphertext using the symmetric key from step 6.
8) Alice regenerates the message digest using the same hash algorithm Bob used.
9) Alice decrypts the digital signature sent by Bob (step 4) using his public key to obtain the message digest.
10) Alice compares digests from steps 8 and 9 to confirm that the message is indeed from Bob, and that message integrity is intact.

Three Types of Digital Signature Standards:

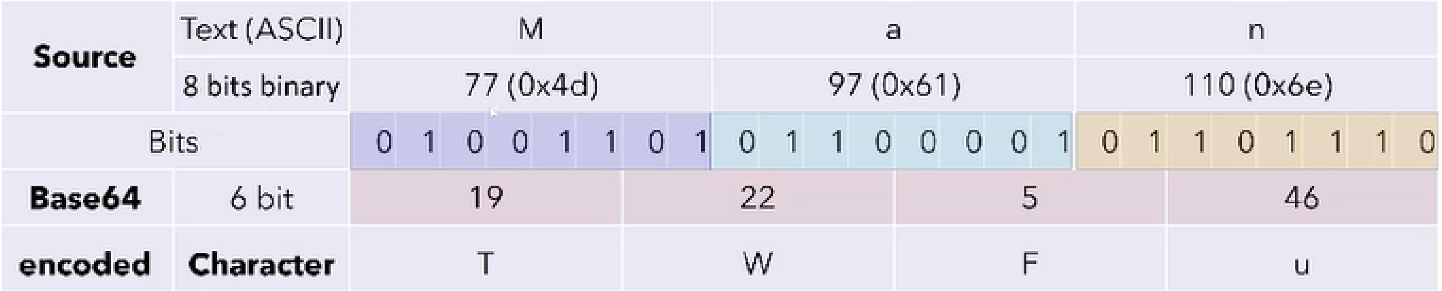
- Simple Electronic Signature (SES): The most basic form of electronic signature. SESs are quick and easy to add to documents, but they lack protection provided by cryptographic encryption methods and are less secure. E.g.: Email signature.
- Advanced Electronic Signature (AES): AESs track changes to the document made after signing. This increases security but is still not suitable to use on important contracts or documents as it is not legally binding.
- Qualified Advanced Electronic Signature (QES)L QES is the safest way to sign electronically. QUEs, a.k.a. Digital Signatures, use public key infrastructure, asymmetric cryptography, and two-factor authentication to ensure the highest level of security. They can also validate the signer's identity, making them as safe and legal as a physical signature.

Week 6 (They didn’t test this on us, maybe for you?)

What is the Base64 Format?

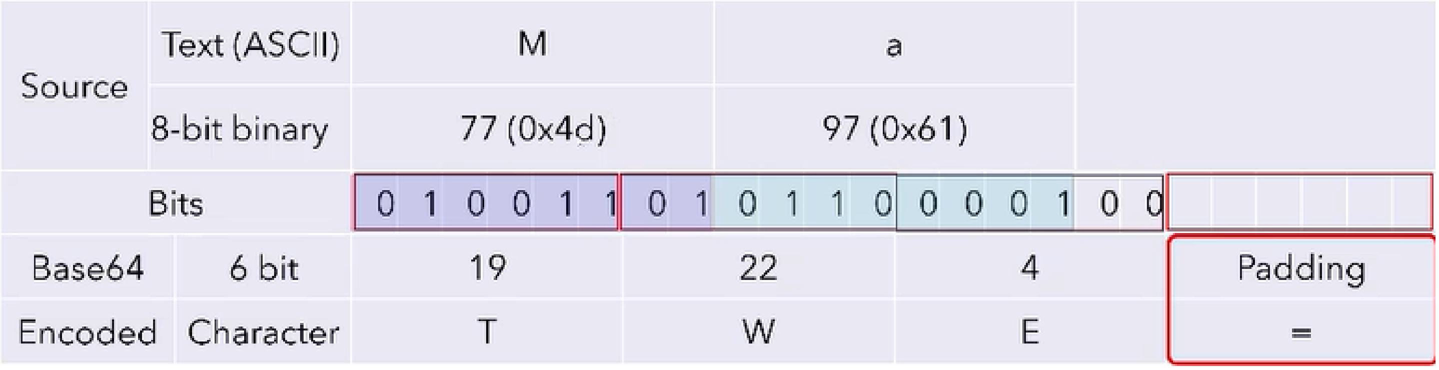
- Base64 is a binary-to-text encoding scheme.
- It represents 8-bit bytes of binary data (ASCII or UTF8) in sequences of 24 bits with four 6-bit Base64 digits, all representing printable character.
- Base64 is also widely used for sending e-mail attachments. This is required because SMTP in its original form was designed to transport 7-bit ASCII characters only.

Base64 Encoding Example: Encoding ASCII characters “Man” to Base64 (Reading 6-bit: 32, 16, 8, 4, 2, 1):



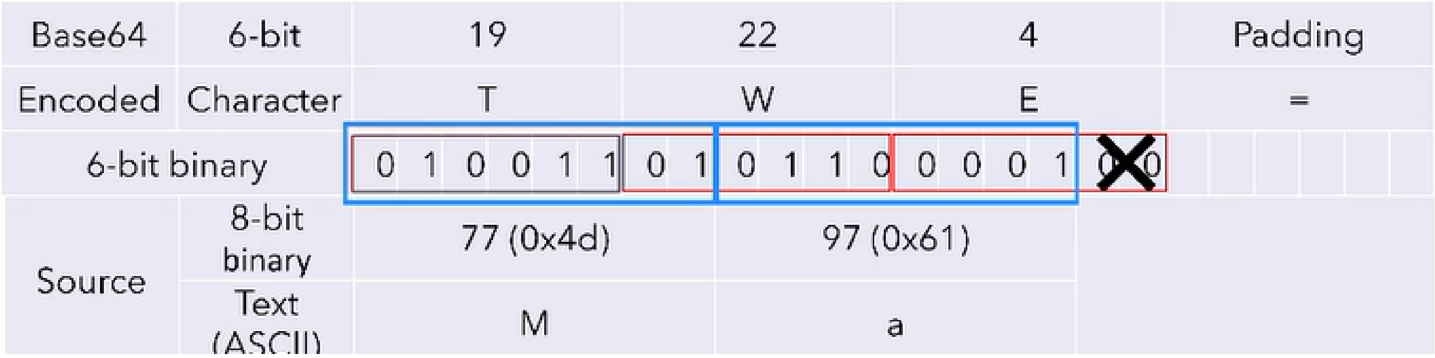
Source	Text (ASCII)	M								a								n							
	8 bits binary	77 (0x4d)								97 (0x61)								110 (0x6e)							
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Base64	6 bit	19						22						5						46					
encoded	**Character**	T						W						F						u					

Base64 Encoding Example: Padding character “=” is used to pad the text which falls short of 24 bits in the last 4 character pair



Source	Text (ASCII)	M								a															
	8-bit binary	77 (0x4d)								97 (0x61)															
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	0						
Base64	6 bit	19						22						4						Padding					
Encoded	Character	T						W						E						=					

Base46 Decoding Example: Base64 “TWE” to “Ma”:



Base64	6-bit	19						22						4						Padding					
Encoded	Character	T						W						E						=					
6-bit binary		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	~~0~~	0						
Source	8-bit binary	77 (0x4d)								97 (0x61)															
	Text (ASCII)	M								a															

Why use Base64 Format:

- Data Transmission:
  - Email Attachments: Some email systems do not handle binary data well, so using Base64 encoding allows binary files to be transmitted as plaintext in emails.
  - HTTP Data: Base64 encoding is often used in HTTP headers or URLs to transmit binary data, such as authentication credentials or image data.
- Data Storage:
  - Database Storage: Base64 encoding is used to store binary data in databases that may not handle binary data efficiently or have resetrictions on certain data types.

    - JSON and XML: Base64 encoding sometimes embeds binary data within JSON or XML documents.
- Data Representation in Text-based Protocols:
    - JSON and XML: Binary data can be included in XML or JSON documents using Base64 encoding.
- Encoding Binary Files:
    - Images and Multimedia: Binary files like images or audio files can be encoded in Base64 format for inclusion in HTML, CSS, or other text-based documents.
- Printing and Visualisation of Binary Data

Base64 Python APIs:

1. from base64 import 64encode
2. base64.b64encode(enter binary data here in bytes)
3. from base64 import b64decode
4. base64.standard_b64decode(enter Base64 bits)