Cybersecurity Essentials and Management for Energy Sector

CSP001_C_E

PRESENTATION BY:
DAVIDE FERRARIS
UNIVERSITY OF MALAGA, SPAIN

Acknowledgement

- *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

Topic-6: Security Controls Selection and Implementation for Energy Environments

Overview

- Select and implement appropriate security controls based on the specific needs of energy systems
- Implement strong password policies and multi-factor authentication (MFA) to protect user accounts
- Encrypt sensitive data at rest and in transit to prevent unauthorised access and data breaches
- Regularly apply security updates and patches to software systems to address vulnerabilities

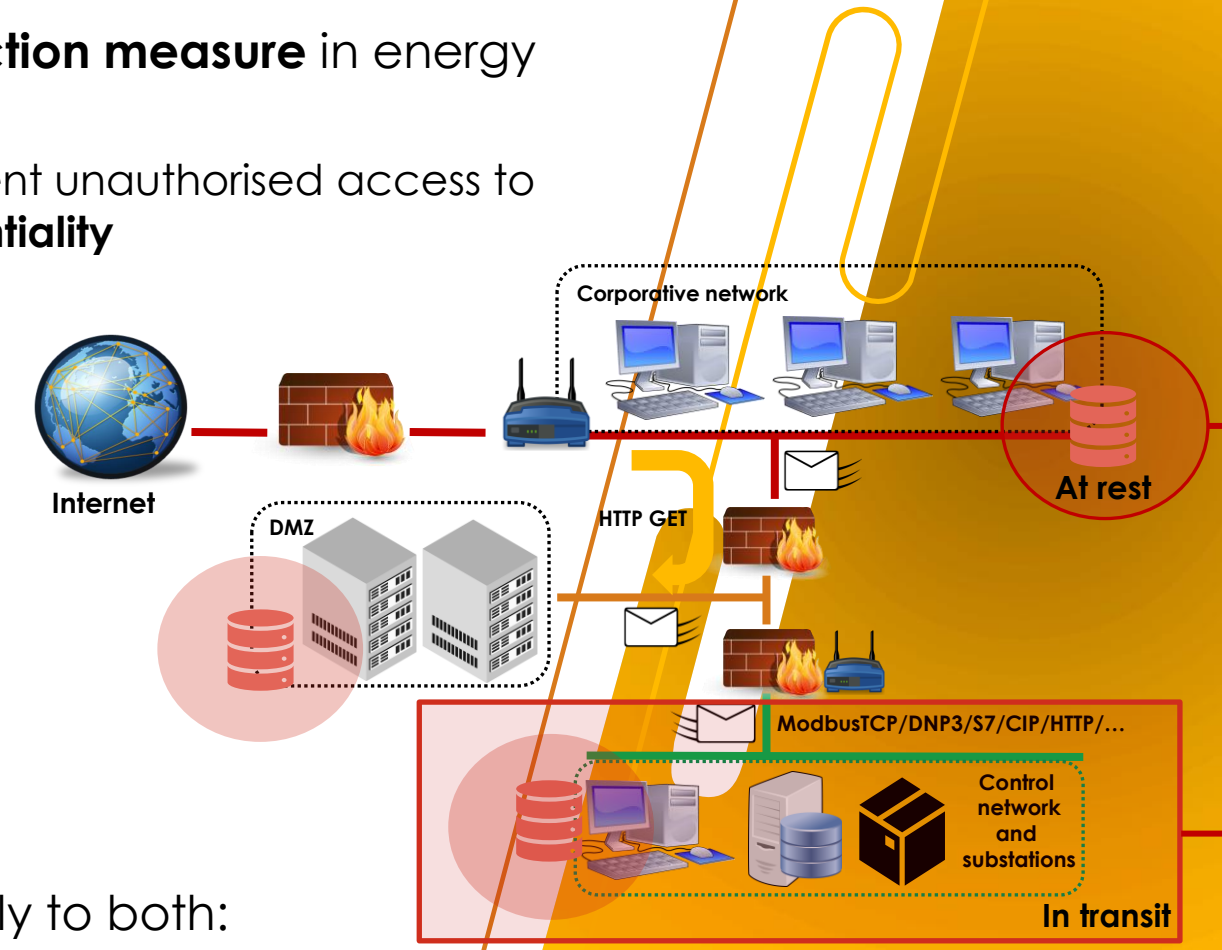
Topic-6: Security Controls Selection and Implementation for Energy Environments

Overview

- Select and implement appropriate security controls based on the specific needs of energy systems
- Implement strong password policies and multi-factor authentication (MFA) to protect user accounts
- **Encrypt sensitive data at rest and in transit to prevent unauthorised access and data breaches**
- Regularly apply security updates and patches to software systems to address vulnerabilities

Cryptography in power control systems

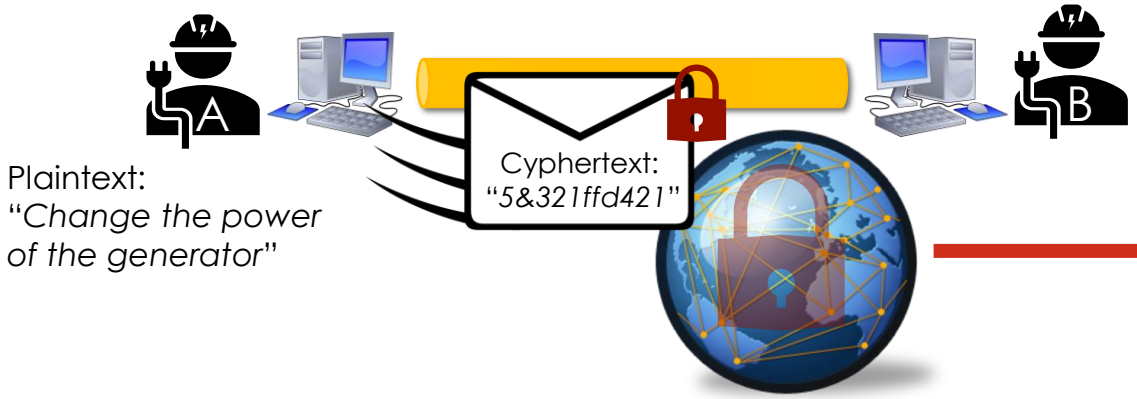
- **Cryptography is a necessary protection measure** in energy control networks
  - Through cryptography, we may prevent unauthorised access to sensitive data, guaranteeing **confidentiality**
- These data can be:
  - Command and control instructions
  - Sensor measurements
  - Network data
  - User requests
  - Authentication data
  - Authorization data
  - Key exchange
  - Etc.
- Moreover, this protection MUST apply to both:
  - **Data in transit and data at rest**



SENSITIVE DATA PROTECTION

Cryptography applied to data in transit

- **Communication channels** of energy control networks MUST be protected by applying cryptographic algorithms
  - In charge of protecting unauthorised access to data in transit



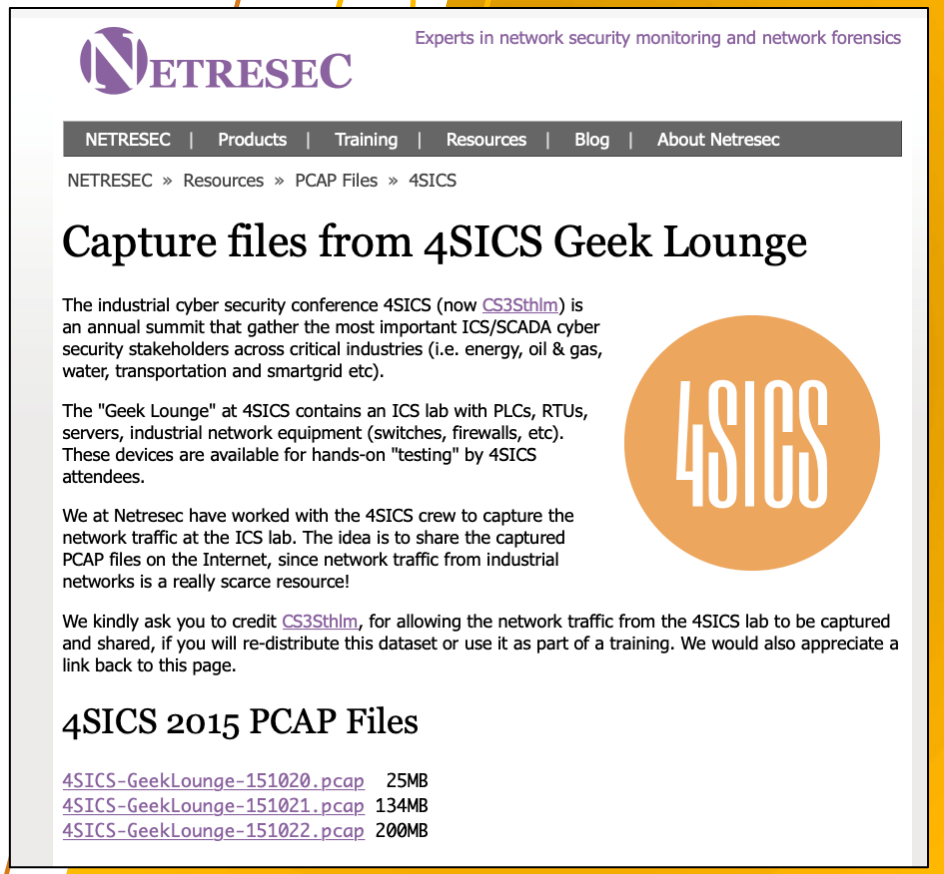
Note that in Topic 5 we have already seen that all VPN protocols rely on cryptographic algorithms to guarantee the confidentiality of the communication channels -- *which may be based on insecure industrial protocols such as ModbusTCP*

  - Otherwise, communications would be sent in 'plaintext', allowing attackers with the necessary resources to intercept, capture and read communications (eavesdrop)
  - Unfortunately, **eavesdropping** can arise in industrial control networks, as most legacy operating devices still rely on insecure protocols such as **telnet** for the remote control

SENSITIVE DATA PROTECTION

Cryptography applied to data in transit

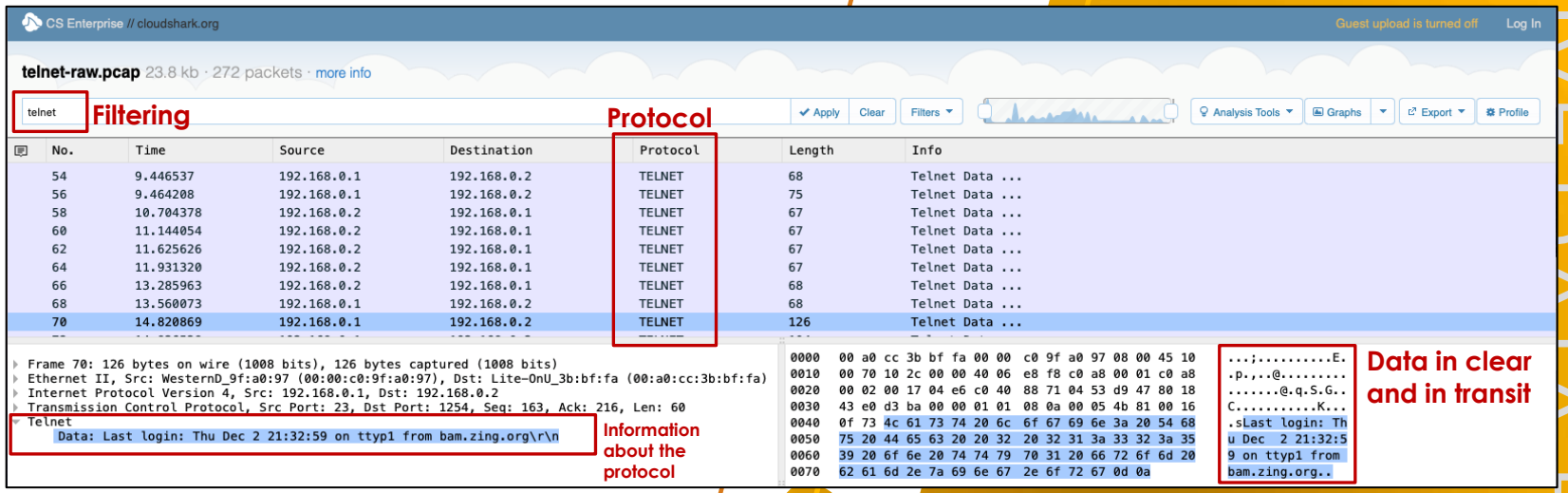
- Netresec offers **4SICS Geek Lounge**, offering some files with captures of industrial traffic
  - URL: https://www.netresec.com/?page=PCAP4SICS
  - The captures are based on an Industrial Control System (ICS) Lab with PLCs, RTUs, servers and industrial network equipment - *all the information about the ICS Lab is found at the same website*
  - As requested, special acknowledgement goes to CS3Sthlm for allowing the community access to these captures
  - More SCADA/ICS network captures can also be found at: https://www.netresec.com/?page=PcapFiles
- With **Wireshark**, it is possible to analyse telnet traffic filtering
  - So an interesting task would be to download the screenshots and filter them by "telnet"



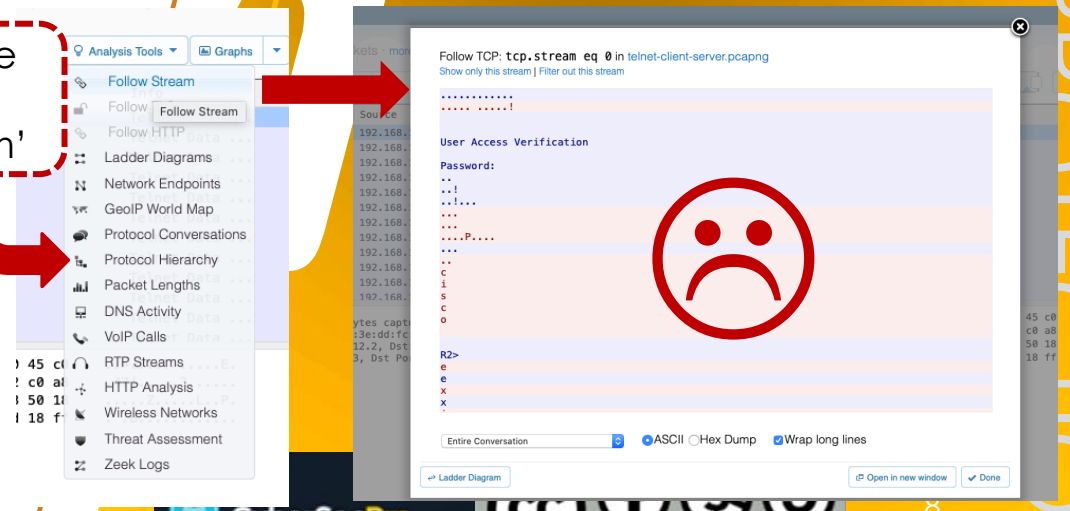
Source and figure source: Netresec, "Captures files from 4SICS Geek Lounge", 2024
URL: https://www.netresec.com/?page=PCAP4SICS
Source: CS3Sthtlm, 2014-2020, accessed in 2024.
URL: https://cs3sthlm.se

Cryptography applied to data in transit

- For the sake of simplicity, we consider the following telnet capture provided by **CloudShark**:
  - Traffic capture: https://www.cloudshark.org/captures/818ceaef07b8
- CloudShark:
  - It is an online dissector with the ability to display network traffic captures, presenting a control panel similar to Wireshark
  - URL: https://www.qacafe.com/analysis-tools/cloudshark/



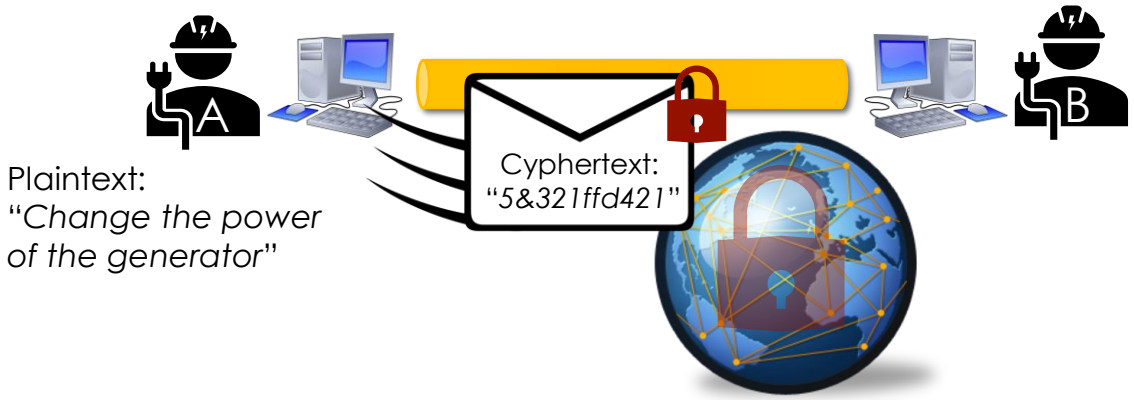
We can also further simplify the analysis by using the existing options, such as 'Follow Stream'



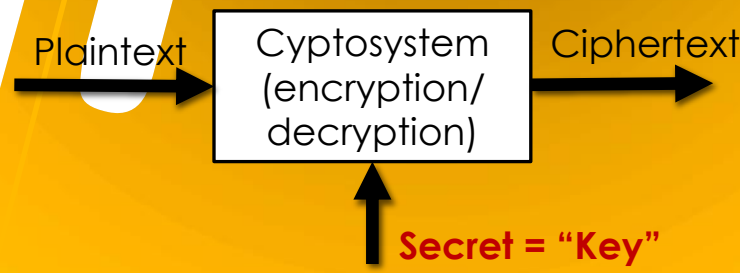
Source and figure source: CloudShark "telnet-client-server.pcapng", accessed in 2024
URL: https://www.cloudshark.org/captures/818ceaef07b8?filter=telnet

Cryptography applied to data in transit

- There is therefore a **clear need to protect communication** channels in energy control systems, where multiple industrial equipment must coexist for control



- This type of protection involves:
  - An **encryption** phase, in which the data is protected by combining the 'plaintext' with a 'secret', resulting in a 'ciphertext'
  - A **decryption** phase, where the 'ciphertext' is transformed into the original 'plaintext' by means of a secret
- This **secret** = the "KEY"



SENSITIVE DATA PROTECTION

Cryptography: Types

- Within modern cryptography, three relevant cryptographic areas have emerged

Symmetric 

- Symmetric cryptographic algorithms use the **same key** for both encryption and decryption

Asymmetric / public key 

- Asymmetric cryptographic algorithms use **two different keys** to carry out the encryption and decryption process

Hybrid 

- The hybrid technique aims to **combine the asymmetric and symmetric algorithms**

SENSITIVE DATA PROTECTION

Cryptography: Types

- Within modern cryptography, three relevant cryptographic areas have emerged

Symmetric 

- Symmetric cryptographic algorithms use the **same key** for both encryption and decryption

Asymmetric / public key 

- Asymmetric cryptographic algorithms use **two different keys** to carry out the encryption and decryption process

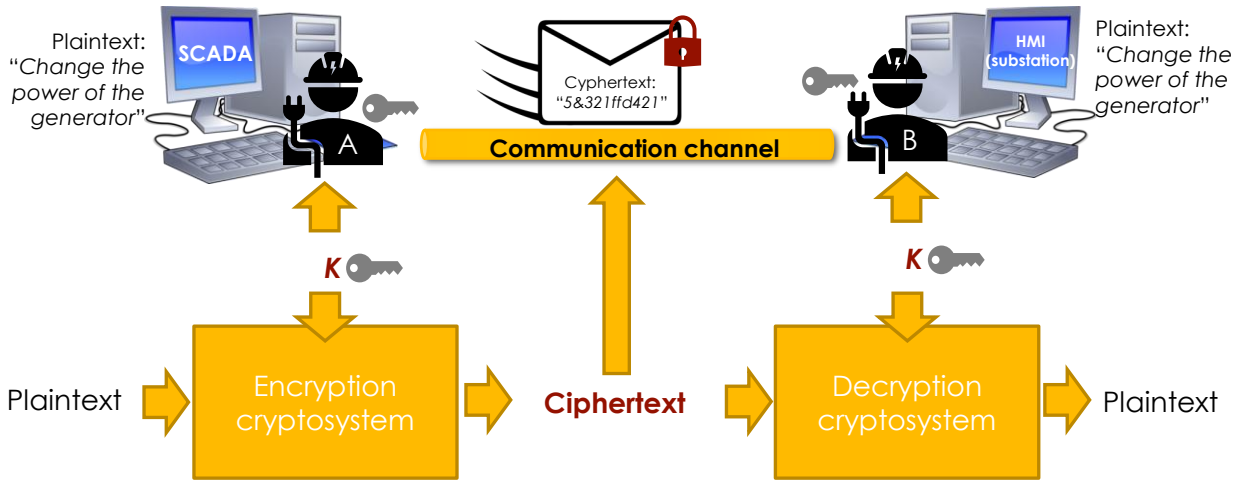
Hybrid 

- The hybrid technique aims to **combine the asymmetric and symmetric algorithms**

SENSITIVE DATA PROTECTION

Symmetric cryptography: Procedure

- Basically, the technique aims to:
  1. IT/OT administrators, industrial network equipment or processes MUST first agree the encryption conditions such as:
     - Type of encryption/decryption algorithm
     - The 'session' key, such that one of the two entities generates the key using some a (pseudo-random) generator
  2. One of the entities involved, encrypt the message under the pre-set conditions and send the encrypted message



Source and figure source: Buchanan, William J., Human-Readable Encryption Keys. Asecuritysite.com, 2024. URL: https://asecuritysite.com/encryption/plain

Symmetric cryptography: Cryptosystems

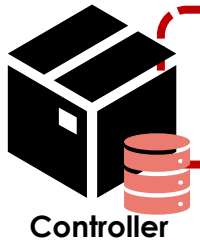
Features	DES	3DES	AES	Camellia
Name	Data Encryption Standard	Triple DES	*Advanced Encryption Standard*	Camellia
Author	IBM	IBM	*Vincent Rijmen, Joan Daemen*	Mitsubishi Electric y NTT
Key length	56 bits	168 (3 claves) o 112 (2 claves)	*128, 192, 256 bits*	128, 192, 256 bits
Cipher block size	64 bits	64 bits	*128 bits*	128 bits
Processing speed	Slow	Very slow	*Fast*	Slow
Security	Low – It is not recommended	Medium	*High*	High

Both are the most widespread and recommended cryptosystems in transit and at rest - both are part of TLS-v1.2/1.3

SENSITIVE DATA PROTECTION

Symmetric cryptography: Cryptosystems

Features	DES	3DES	AES	Camellia
Name	Data Encryption Standard	Triple DES	*Advanced Encryption Standard*	Camellia
Author	IBM	IBM	*Vincent Rijmen, Joan Daemen*	Mitsubishi Electric y NTT
Key length	56 bits	168 (3 claves) o 112 (2 claves)	*128, 192, 256 bits*	128, 192, 256 bits
Cipher block size	64 bits	64 bits	*128 bits*	128 bits
Processing speed	Slow	Very slow	*Fast*	Slow
Security	Low – It is not recommended	Medium	*High*	High

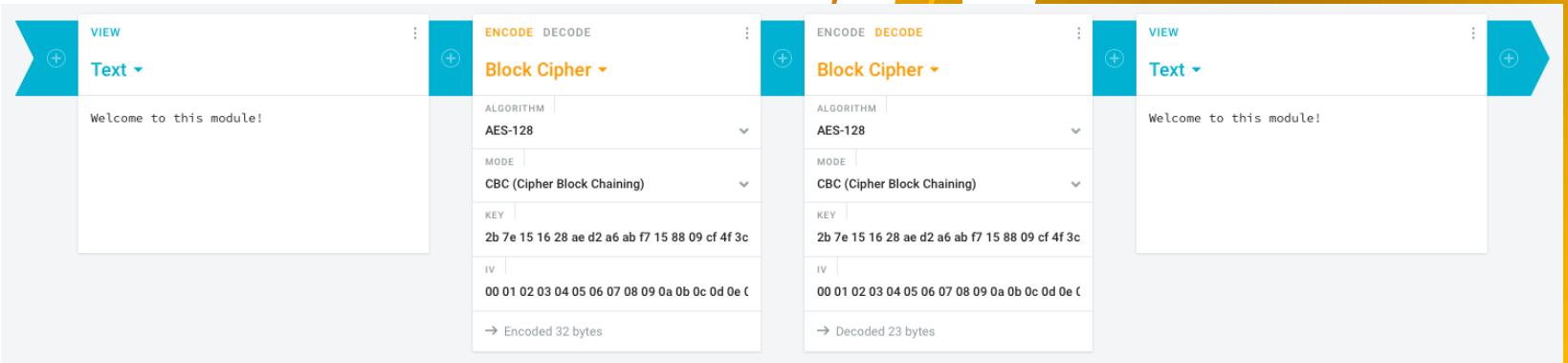


However, we cannot forget that there are still legacy operational devices that apply these cryptosystems

SENSITIVE DATA PROTECTION

Symmetric cryptography: A practical view

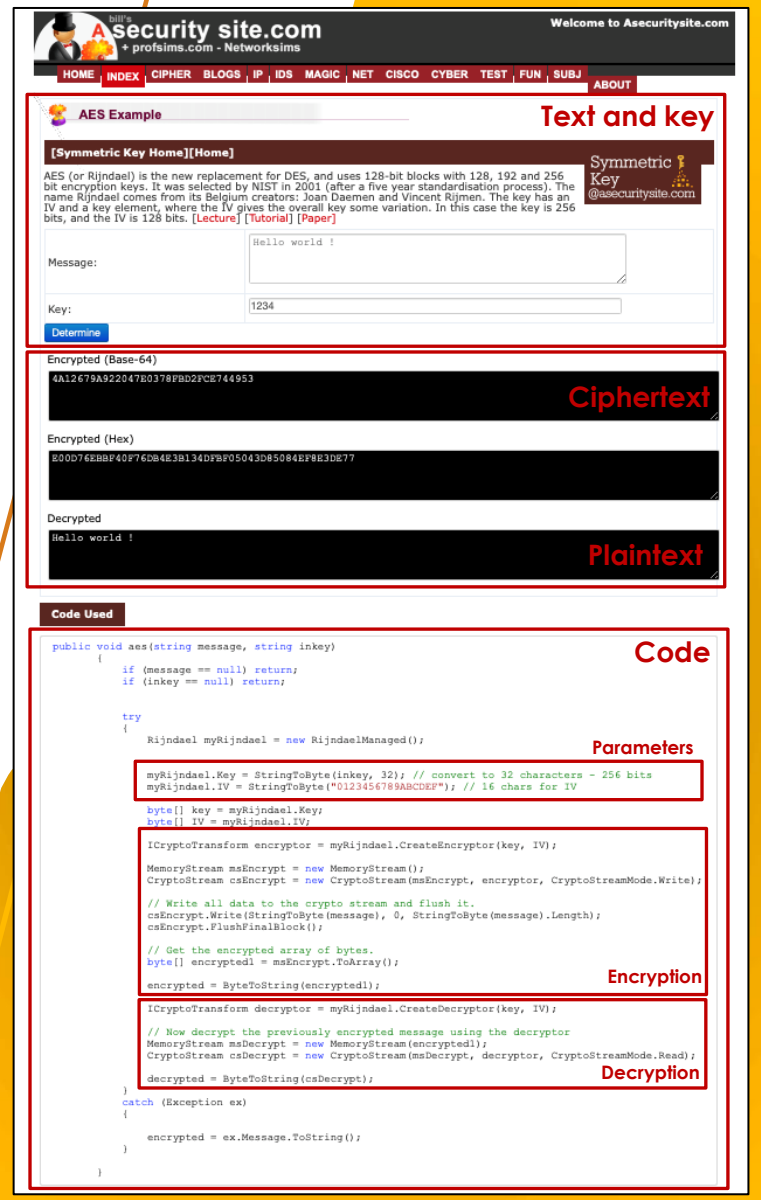
- Through the online tool **cryptii** (https://cryptii.com), we can put the lessons learned into practice
  - *Got to 'Modern Cryptography' > Block Cipher*
- Exercises:
  1. Encrypt the following message '*Welcome to this module!*', using the following Key: *2b7e151628aed2a6abf7158809cf4f3c*
  2. Decrypt the ciphertext using the same key and encryption conditions
- What is happening?
  - Wuala ! ☺
  - We can encrypt messages and recover the original message



Source: Wierk, Cryptii, 2024.
URL: https://cryptii.com

Symmetric cryptography: A practical view

- We can enhance our technical knowledge and extend our 'SKILLs' by considering the online tool of **Asecuritysite**
  - AES: https://asecuritysite.com/symmetric/aes
  - DES: https://asecuritysite.com/symmetric/des
  - 3DES: https://asecuritysite.com/symmetric/threedes
  - Camelia: https://asecuritysite.com/symmetric/camellia
- Exercises:
  1. Obtain the session key, where the seed (a special pre-key) for the generation of the key could be "1234"
  2. Encrypt and decrypt a message, e.g., "Hello World !"
  3. If you have time, take a look at the attached code on the same web page



Source and figure source: Buchanan, William J., *AES*. Asecuritysite.com, 2024.
URL: https://asecuritysite.com/symmetric/aes
Source: Buchanan, William J., *DES Cipher*. Asecuritysite.com, 2024.
URL: https://asecuritysite.com/symmetric/des
Source: Buchanan, William J., 3DES Cipher. Asecuritysite.com, 2024.
URL: https://asecuritysite.com/symmetric/threedes
Source: Buchanan, William J., Camellia cipher. Asecuritysite.com. 2024.
URL: https://asecuritysite.com/symmetric/camellia

Symmetric cryptography: 'Modes of operation'

- Note that if the symmetric cryptographic algorithms were used as they are, it would be possible to obtain the same ciphertext every time
  - That is, for the same plaintext as input, the same ciphertext will always be obtained as output !
- To avoid this situation, block cipher **modes of operation** are applied to **inject noise** (SALT) into the encryption process:
  - Cipher Block Chaining (CBC)
  - Counter (CTR)
  - Galois/Counter Mode (GCM)

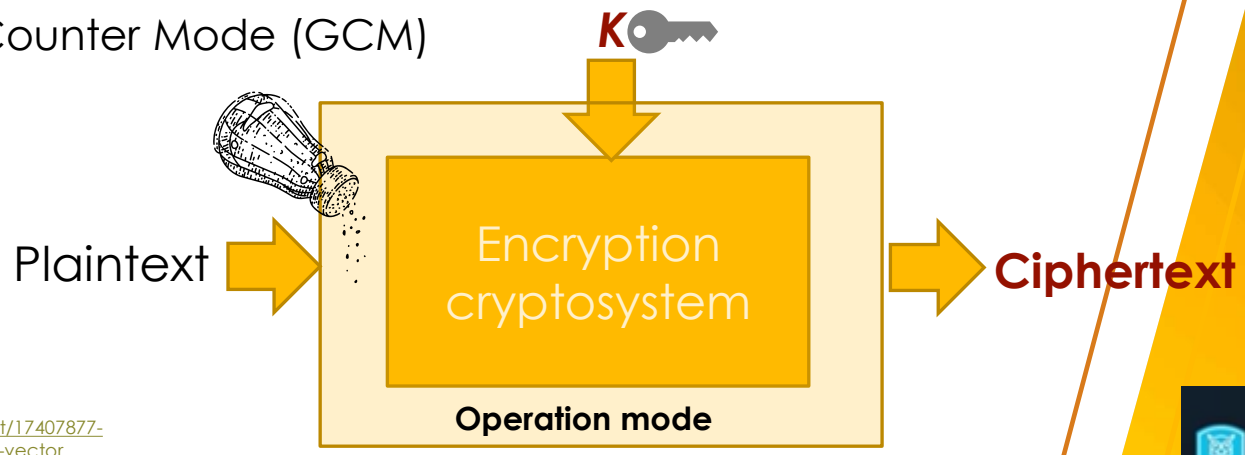
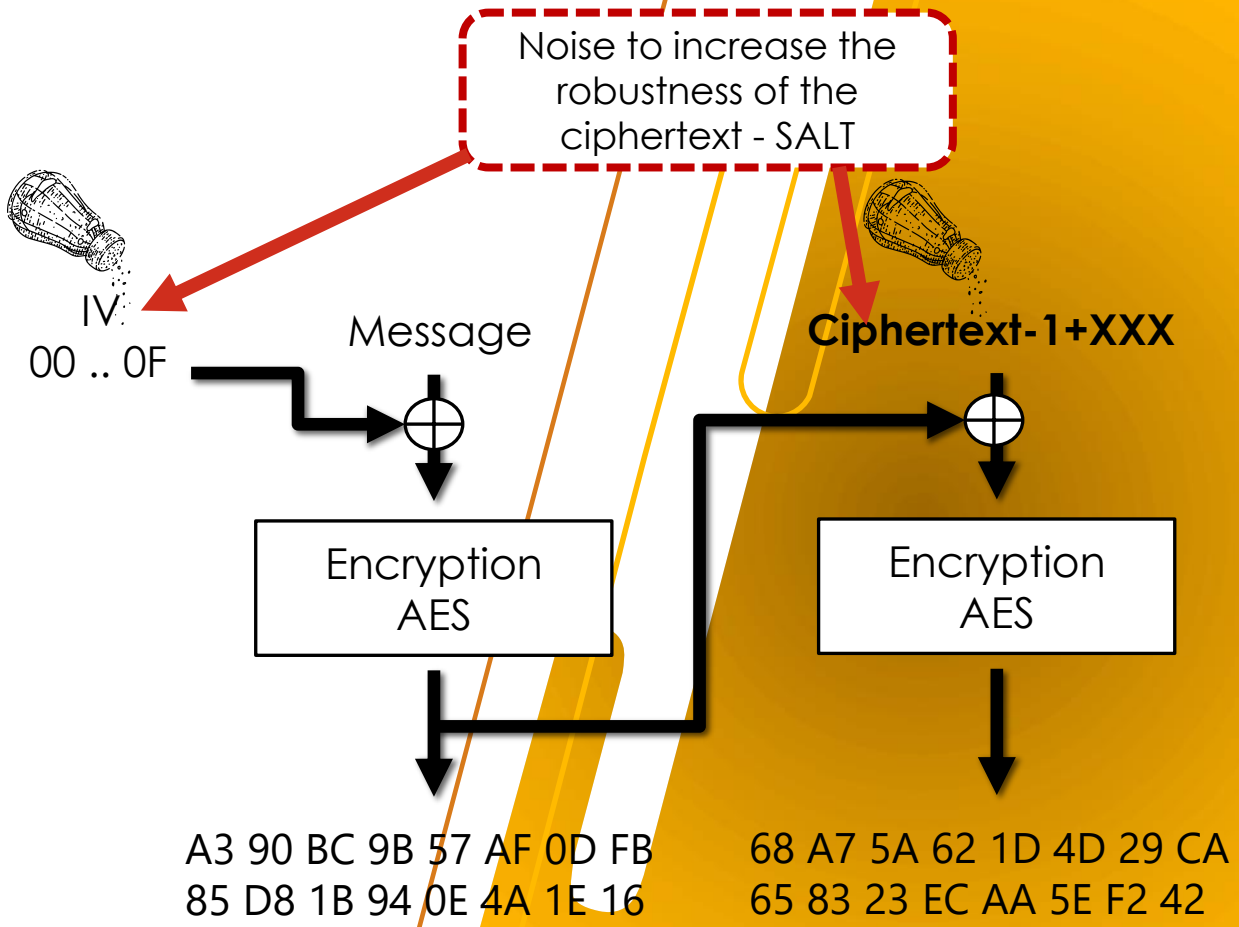


Figure source: Vecteezy
URL:https://www.vecteezy.com/vector-art/17407877-traditional-salt-shaker-sketch-hand-drawn-vector

Symmetric cryptography: 'Modes of operation'

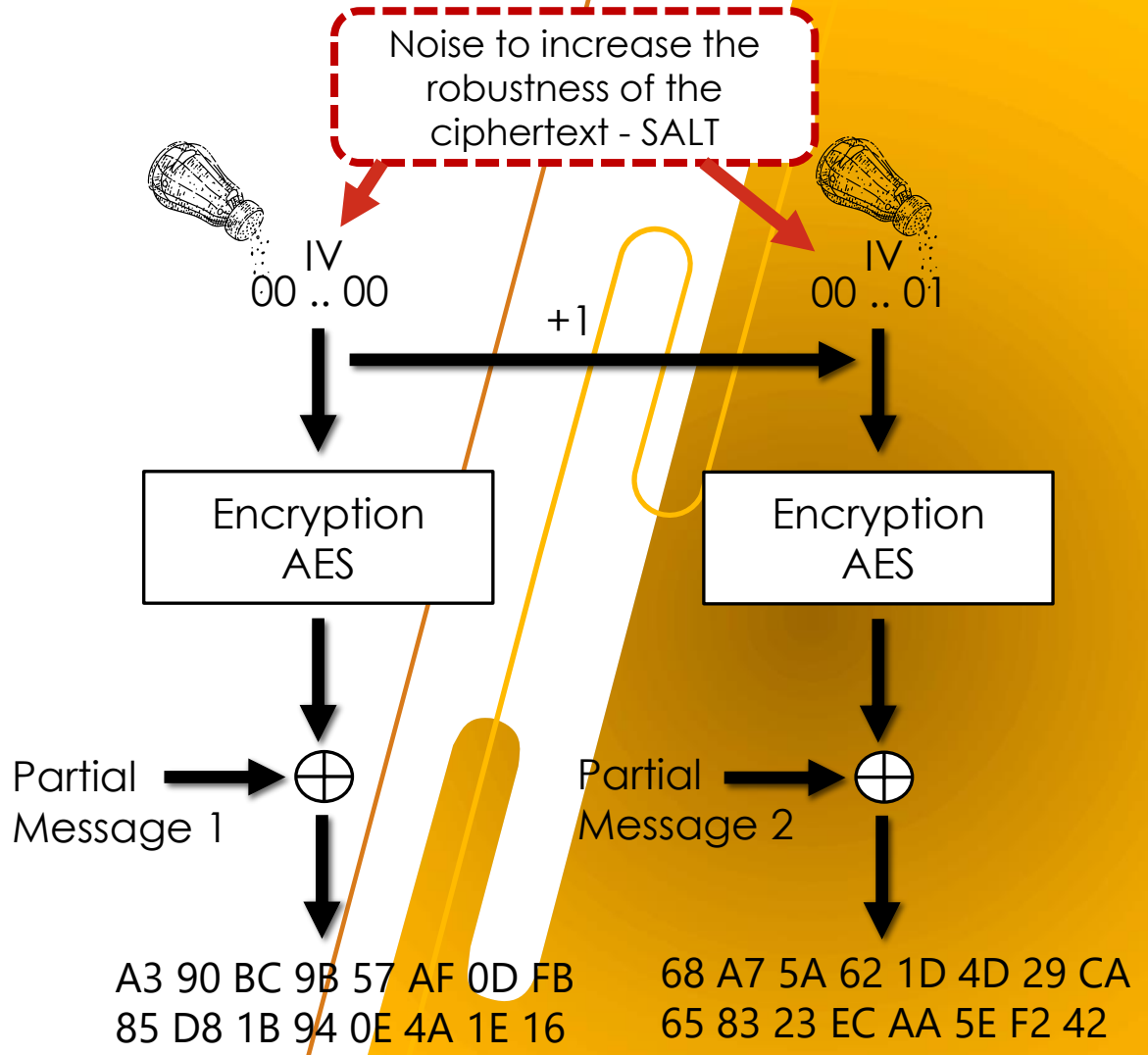
- **CBC mode**
  - The output of each cipher block is used to encrypt the next block
  - Uses a random '**Initialisation Vector**' (IV) as input to the first cipher - *provides variability (= SALT)*
    - Note that this IV value is public (e.g. 00..0F) and must be sent the destination node together with the ciphertext
  - A padding operation is normally performed at the end of the process to facilitate the computation
    - 'Pad the input (XXX) to be a multiple of the block size



SENSITIVE DATA PROTECTION

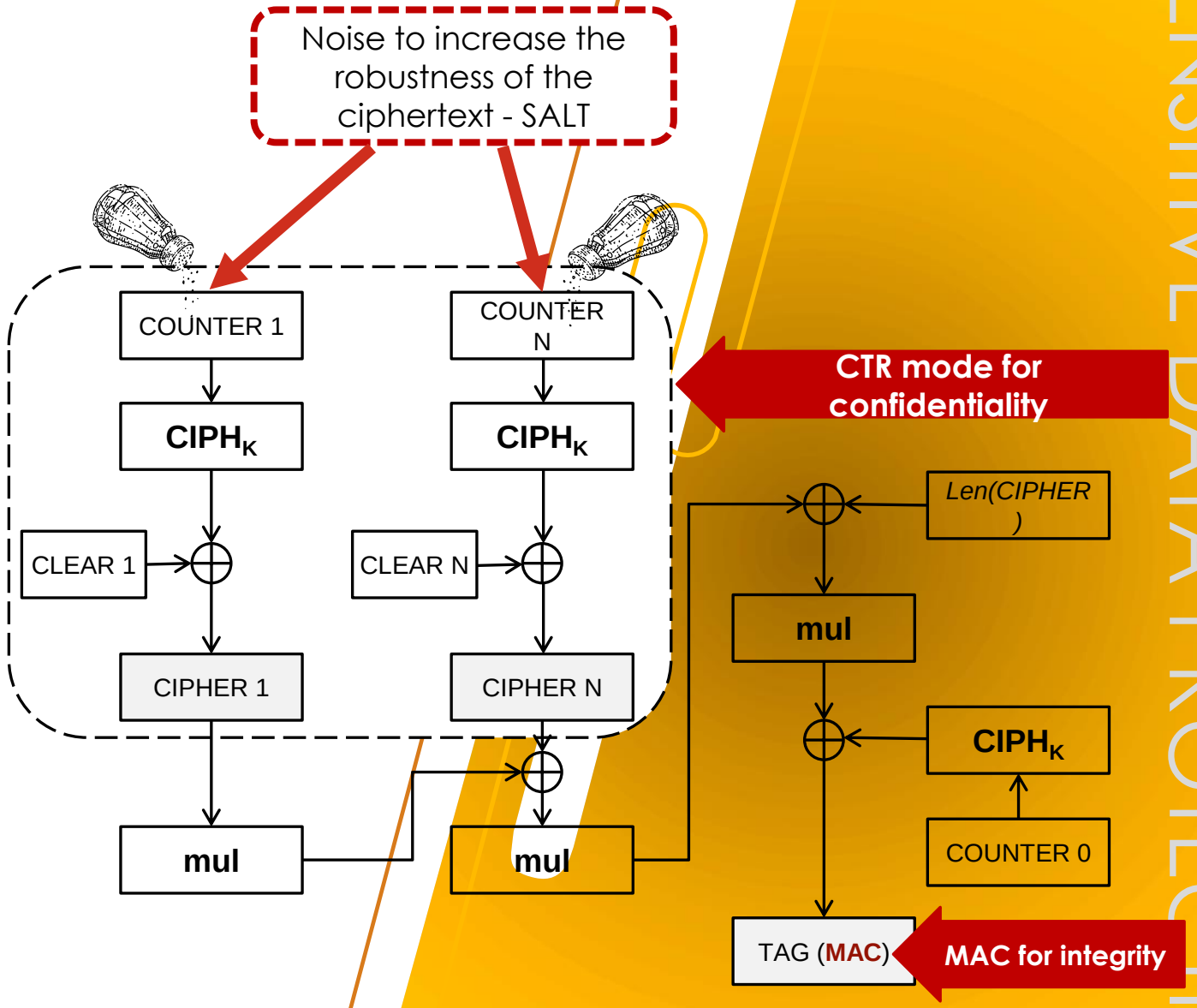
Symmetric cryptography: 'Modes of operation'

- **CTR mode**
  - The result of each cipher block is part of the ciphertext
  - When all operations are completed, each ciphertext piece is concatenated to provide the final ciphertext piece
  - This operation model also uses a random '**Initialisation Vector**' (IV) as input for the first cipher
    - Its value is increased for subsequent operations (= SALT)



Symmetric cryptography: 'Modes of operation'

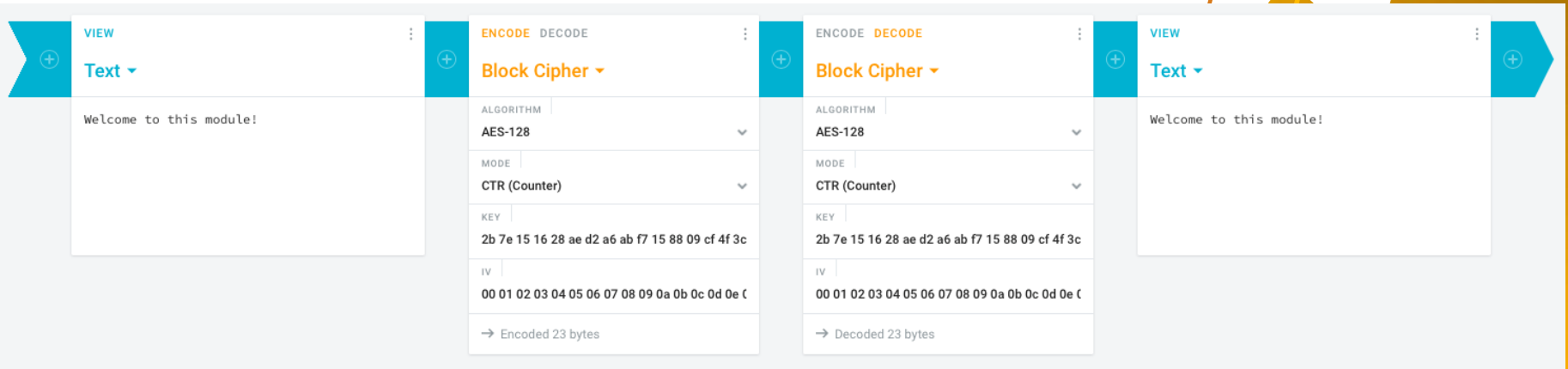
- **GCM mode**
  - Uses a random value ("nonce" / "counter" / IV) as input to the first cipher - *provides variability (= SALT)*
    - This value is public and must be sent along with the ciphertext
  - No padding operation to be done
  - Provides a cipher integrity code ("Tag" / "MAC")
    - A change in the cipher changes the integrity code
    - This features comply with the **Authenticated Encryption with Additional Data (AEAD)** property – *confidentiality + integrity*



SENSITIVE DATA PROTECTION

Modes of operation: A practical view

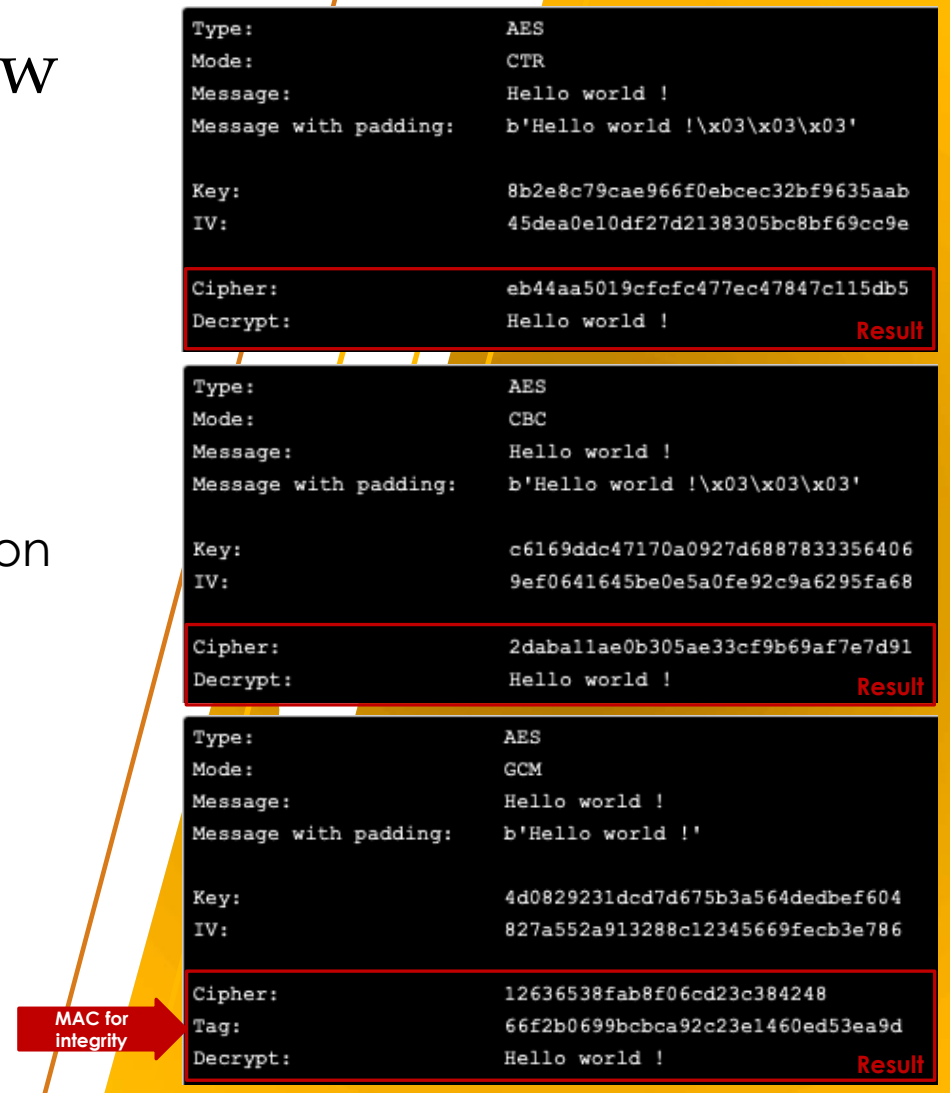
- Go again to **cryptii** to '*Modern Cryptography*' > *Block Cipher*
- Exercises:
  1. Encrypt the following message '*Welcome to this module!*', using:
     - *Key: 2b7e151628aed2a6abf7158809cf4f3c*
     - *Operation mode: CTR*
     - *IV: 000102030405060708090a0b0c0d0e0f*
  2. Decrypt the ciphertext but using the same encryption conditions



Source: Wierk, Cryptii, 2024.
URL: https://cryptii.com

Modes of Operation: A practical view

- Go again to **Asecuritysite** and particularly to:
  - URL: https://asecuritysite.com/symmetric/sym
- Exercises:
  1. Encrypt with AES+CBC, AES+CTR and AES+GCM
  2. Decrypt the ciphertext but using the same encryption conditions



Source and figure source: Buchanan, William J., *Symmetric key modes: GCM, ChaCha20, CBC, CFB8, CFB, OFB, GCM, CTR, and XTS*, Asecuritysite.com, 2024.
URL:https://asecuritysite.com/symmetric/sym

Symmetric cryptography: Advantages and disadvantages

Advantages	Disadvantages
• Symmetric encryption algorithms typically offer **high performance at low** (or relatively low) **computational cost**	• The secret key ***K*** has to be agreed a priori 
• The keys used are **relatively short** • It is recommended to use keys of **128-bits size or larger** 	• The **number of keys may grow** significantly if all users of a community need to communicate (individually) with the other members of that community → **(n * (n-1)) / 2**

SENSITIVE DATA PROTECTION

Cryptography: Types

- Within modern cryptography, three relevant cryptographic areas have emerged

Symmetric 

- Symmetric cryptographic algorithms use the **same key** for both encryption and decryption

Asymmetric / public key 

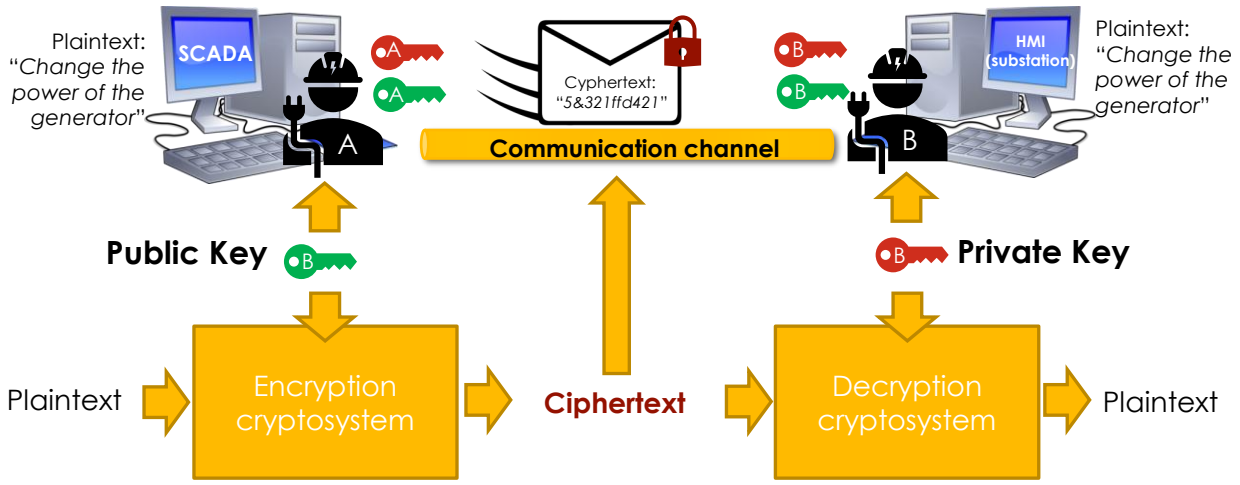
- Asymmetric cryptographic algorithms use **two different keys** to carry out the encryption and decryption process

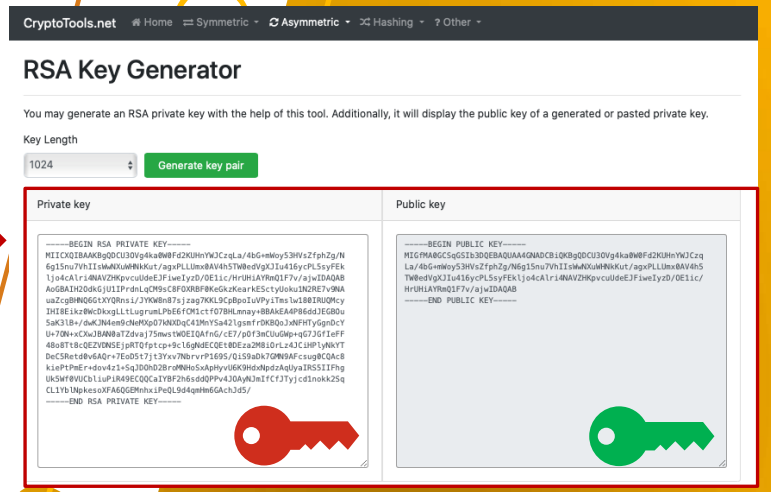
Hybrid 

- The hybrid technique aims to **combine the asymmetric and symmetric algorithms**

Asymmetric cryptography: Procedure

- Basically, the technique aims to:
  1. Each IT/OT administrator, industrial network equipment or process MUST create two relevant keys:
     - **1 Private Key**: known by only the owner of the key
     - **1 Public key**: known by all entities
  2. One of the entities involved **encrypts** the message using the 'public key' of the destination as it is the only entity with the private key



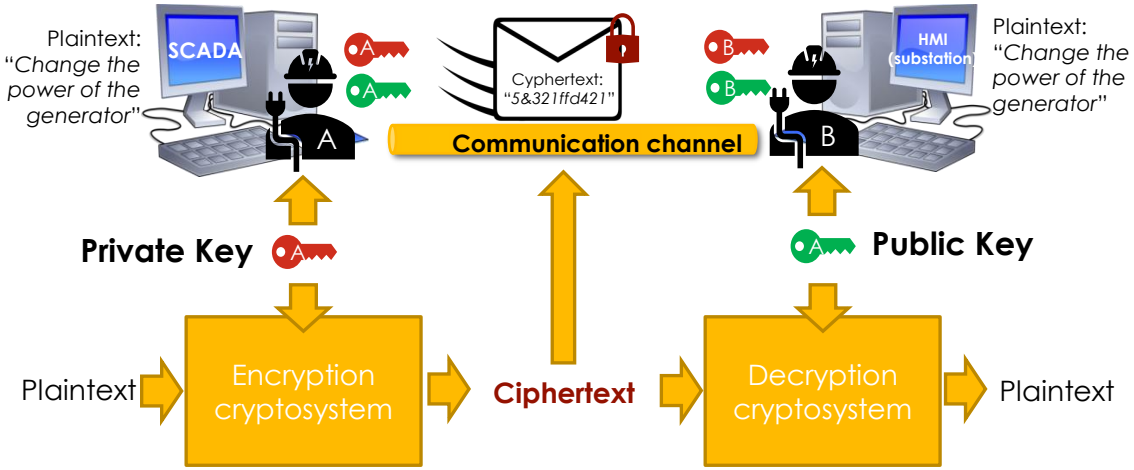


These two keys are different but closely related to each other, allowing the encryption and decryption of information

Source and figure source: CryptoTool.net, RSA Key Generator, 2024.
URL: https://cryptotools.net/rsagen

Asymmetric cryptography: Procedure

- However, if the private key is applied to encrypt (sign) the message, what is achieved is a **Digital Signature**



- Thus, depending on how the two keys are applied, this is achieved:

Encryption for confidentiality	Digital Signature for authentication
• Encrypt with the receiver's public key • Decrypt with the receiver's private key	• Encrypt with the sender's private key • Decrypt with the sender's public key

SENSITIVE DATA PROTECTION

Asymmetric cryptography: Cryptosystems

Features	RSA	ECC
Name	RSA	Elliptic-curve cryptography
Author	Ron Rivest, Adi Shamir and Leonard Adleman	Neal Koblitz and Victor S. Miller
Key length	1024, 2048 y 3072 bits	160, 224, 256, 384, 521 bits
Processing speed	Slow	Fast
Security	High	High
Functionality	Encryption, signature and key exchange	Encryption, signature and key exchange

RSA is the most widespread, but ECC is ideal for computationally constrained operational devices, such as RTUs/PLCs, sensors and actuators

Asymmetric cryptography: Advantages and disadvantages

Advantages	Disadvantages
• All entities involved in the energy control system **can generate the two keys without having to agree on them a priori**	• Asymmetric encryption algorithms require **computational capabilities** to process their encryption algorithms
• El number of keys is 2*n • **Reduces the cost of storage** at end points	• The keys they use are relatively large • It is **recommended to use keys of size 1024 bits or larger**

Cryptography: Types

- Within modern cryptography, three relevant cryptographic areas have emerged

Symmetric 

- Symmetric cryptographic algorithms use the **same key** for both encryption and decryption

Asymmetric / public key 

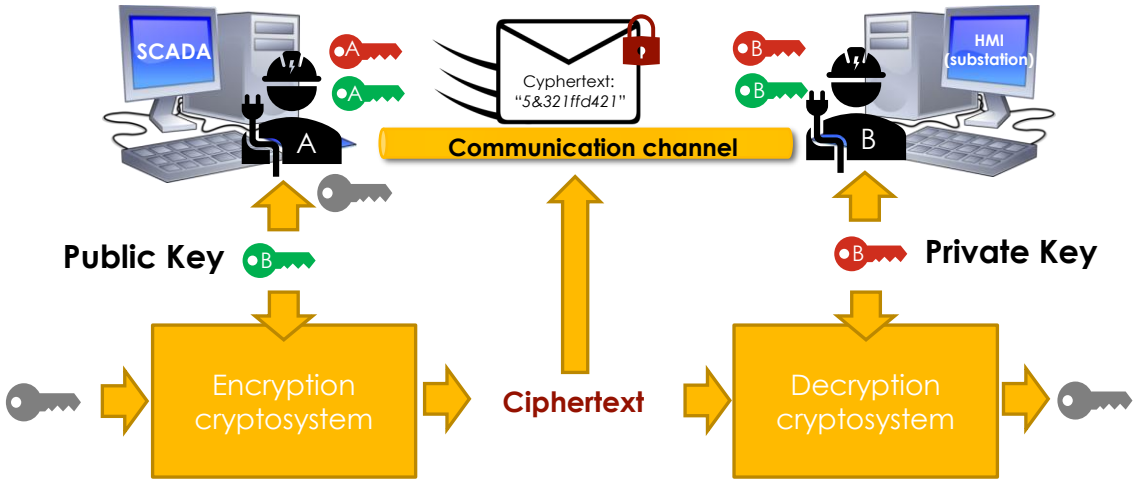
- Asymmetric cryptographic algorithms use **two different keys** to carry out the encryption and decryption process

Hybrid 

- The hybrid technique aims to **combine the asymmetric and symmetric algorithms**

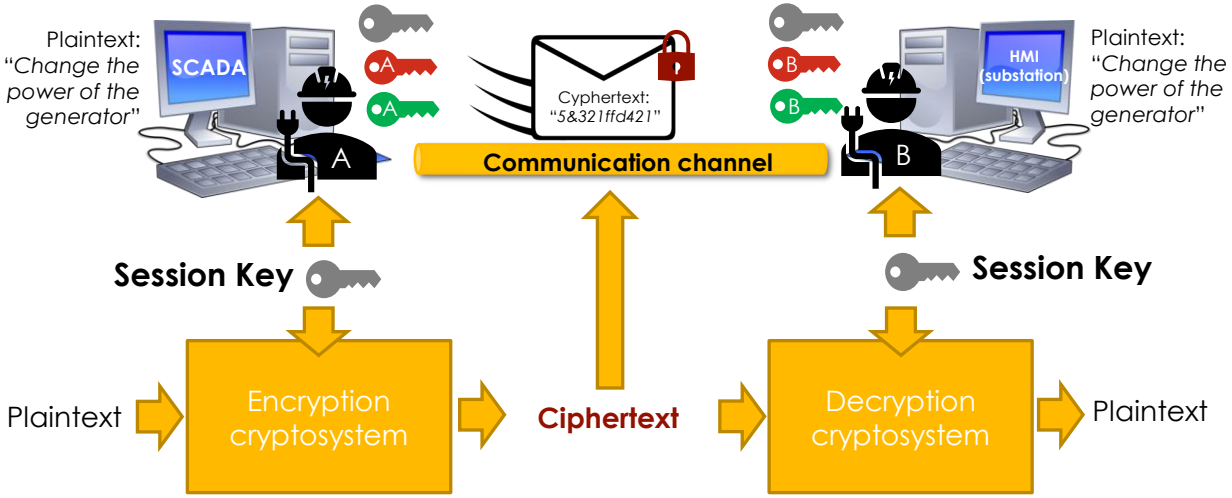
Hybrid cryptography: Procedure

- Hybrid cryptography aims to obtain the advantages offered by the two approaches:
  - **Asymmetric cryptography** to negotiate the session key
  - **Symmetric cryptography** to encrypt/decrypt messages
- Basically, the technique aims to:
  1. One of the entities generates the session key, and sends it to the receiver using asymmetric cryptography

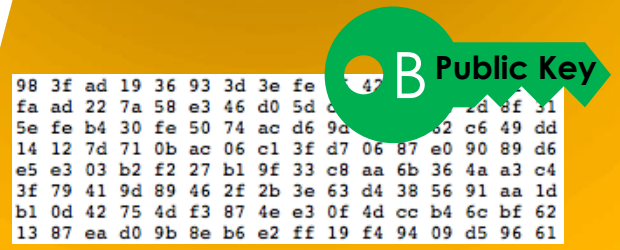


Hybrid cryptography: Procedure

2. Once the session key has been obtained by the receiver, one of the entities generates a message, which is protected using the session key



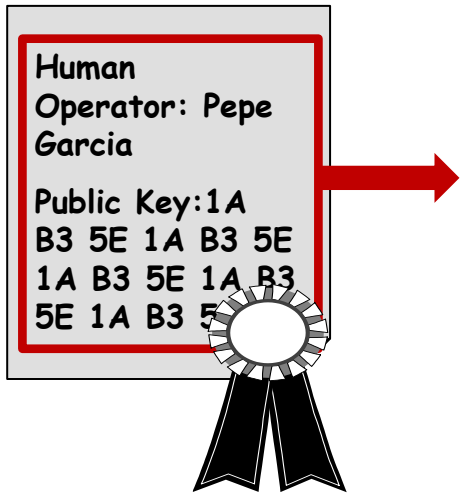
- However, the exchange of public keys without additional information about the entity, only the key (hexadecimal values), brings with it the need to manage **digital certificates** in order to trust the public key



Digital certificates: x.509

- A digital certificate is the digital document with the "seal of assurance" that a key belongs to a specify entity, mainly because it is certified by a **trusted entity** that validates and signs the document
- This entity in charge of validating and certifying is known as a **Certification Authority (CA)**

The CA should be the entity with legal rights to certify the document, such as the system's managers or directives, etc.



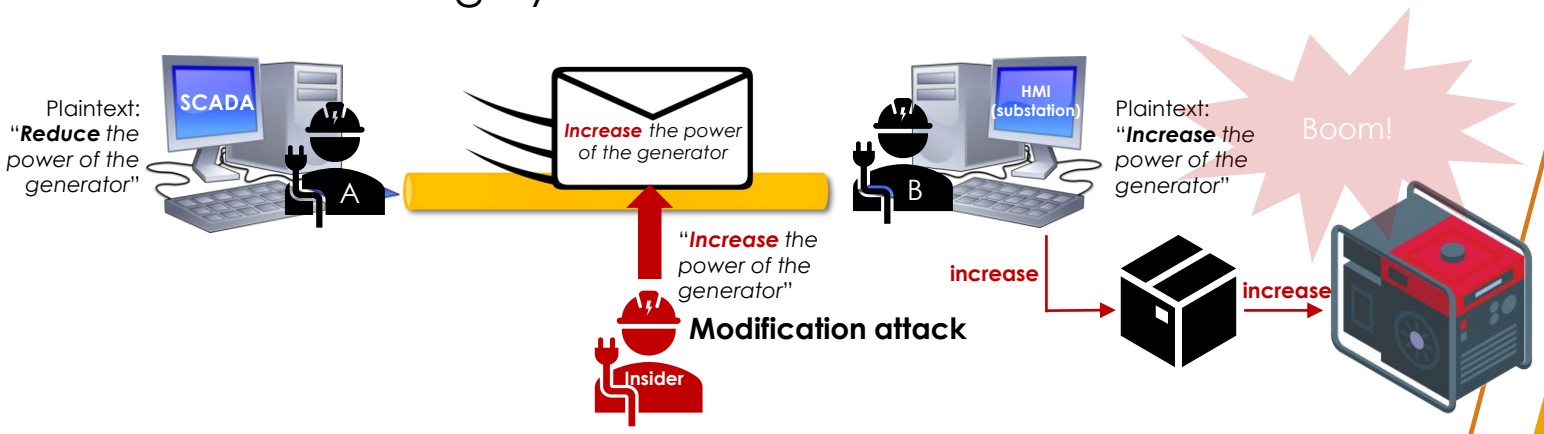
The document contains at least

- A unique serial number
- The identity of the user for whom the information is provided
- The value of the public key
- The expiry date of the certificate
- The identity of the issuer of the document
- The digital signature of the document issue

SENSITIVE DATA PROTECTION

Hash for sensitive data integrity

- One way to detect unwanted changes to the content of messages would be through specific cryptographic functions for "integrity"



- **Hash functions** are the most cryptographically efficient validation methods
  - These functions are **one-way functions** that ensure that two different messages result in different hash values (or digests)
  - With this, we also ensure "**collision-free**" (the main vulnerability of such functions)

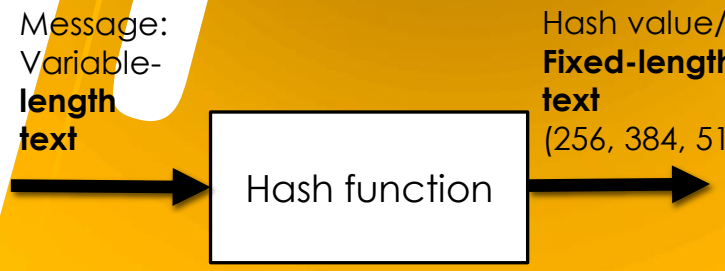
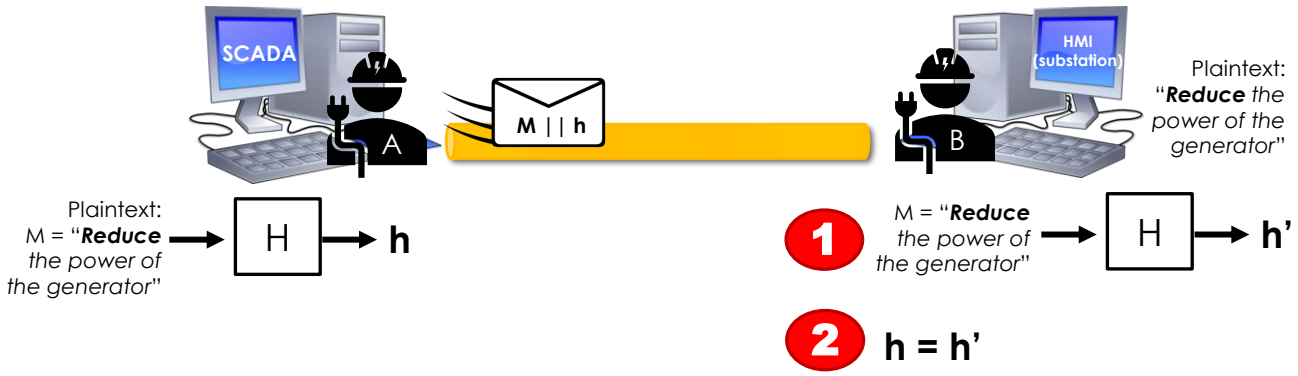


Figure source: Vecteezy
URL:https://www.vecteezy.com/vector-art/19053049-petrol-generator-icon-isometric-vector-industrial-equipment

Hash function: Procedure

- Basically, the technique aims to:
  1. The sender prepares the message and obtain its corresponding hash value
  2. Both the message and the hash value are sent to the destination
  3. The receiver calculates the hash value of the received message and compares it with the received hash value
  4. If both hashes are equal, the receiver confirms the integrity of the message

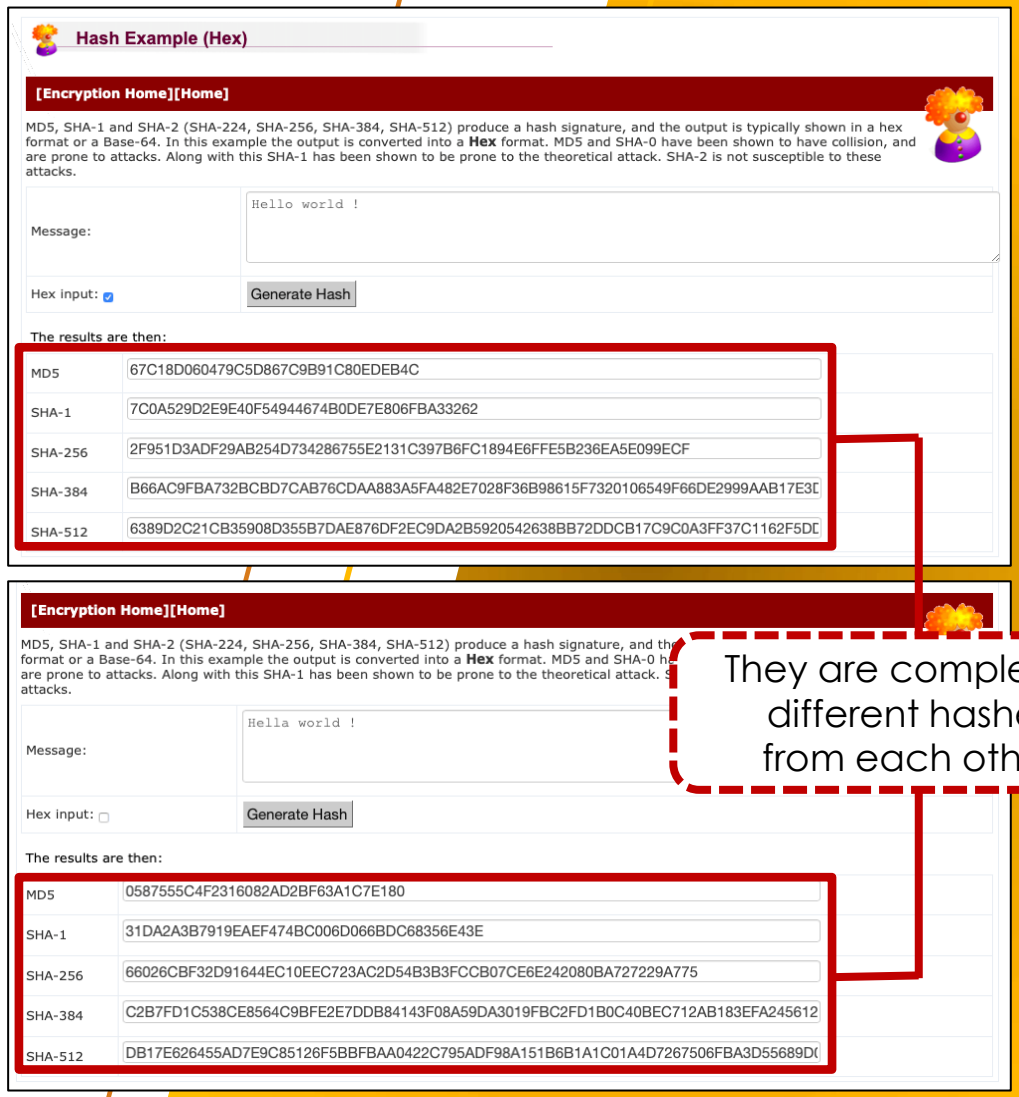


Hash function: A practical view

- Go to **Asecuritysite**, and particularly to:
  - URL: https://asecuritysite.com/encryption/md5
- Exercises:
  1. Generate the hash of a message
  2. Change a character to the message
  3. Compare the hash values
- Repeat the exercise with **criptii**

- It is important to note that some hash functions are susceptible to collisions, such as: MD5 and SHA-1
  - Unfortunately, **legacy operating devices can still support them**

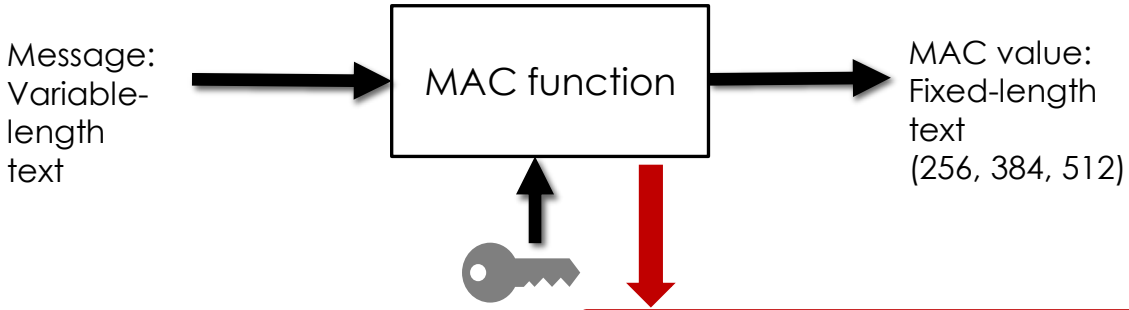
Source: Buchanan, William J, Hashing, Asecuritysite.com, 2024.
URL: https://asecuritysite.com/encryption/md5
Source: Wierk, Cryptii, 2024.
URL: https://cryptii.com



They are completely different hashes from each other

MAC for data integrity and authentication

- The **Message Authentication Code (MAC) function** does the same as the hash function, but considering as input values both:
  - The message, and
  - A symmetric KEY shared by the two entities of the communication



There are several MAC functions, such as **Hash-Based Message Authentication Codes (HMAC)**

- MAC guarantees:
  - **Integrity** through the hash function (e.g. HMAC)
  - **Authentication** through the secret key

As in the previous points, learners can practice with HMAC using the online tools:

- **Asecuritysite**, https://asecuritysite.com/mac/go_hmac
- **Cryptii**, https://cryptii.com

Cryptography applied to e-mails (data in transit)

- Energy professionals can activate security measures that allow them to protect their e-mails, using, for example:
  - **Secure/Multipurpose Internet Mail Extension (S/MIME)**
  - **Pretty Good Privacy (PGP)** under the OpenPGP standard
- Both protocols offer **confidentiality, integrity and authentication** guarantees
  - S/MIME handles digital certificates x.509 for authentication
  - PGP offers its own certificate format
- S/MIME and PGP are compatible for all the platforms and can be configured in multiple email editors, such as Thunderbird
  - Also note that PGP can imply the installation of GnuPGP, which is an implementation of the OpenPGP standard
  - And PGP can be applied to protect data at rest, such as files

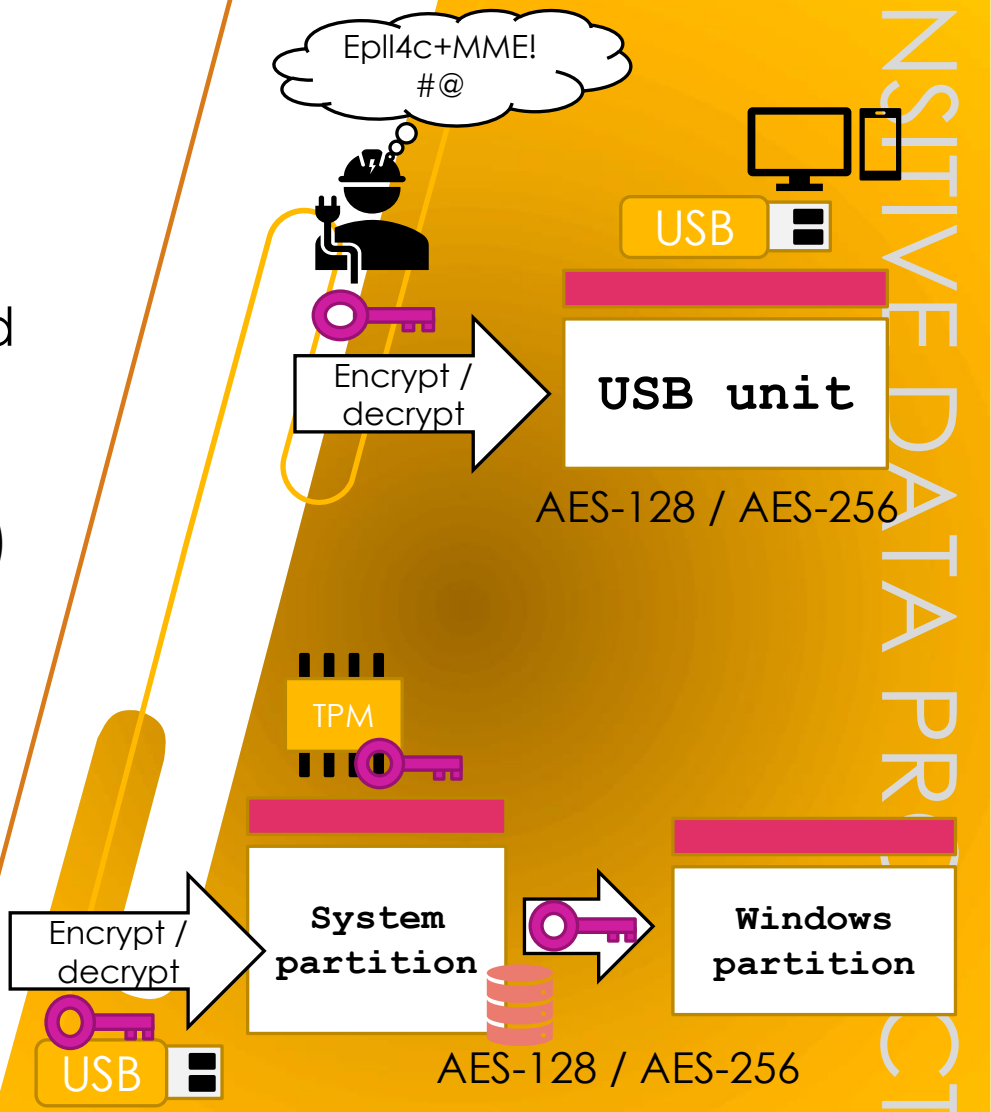
SENSITIVE DATA PROTECTION

As in the previous points, learners can also practice the potential of PGP to protect files locally, using, for example:

- **Kleopatra**
  (open-source, integrating GnuPG)
  https://www.openpgp.org/software/kleopatra/

Cryptography applied to data at rest

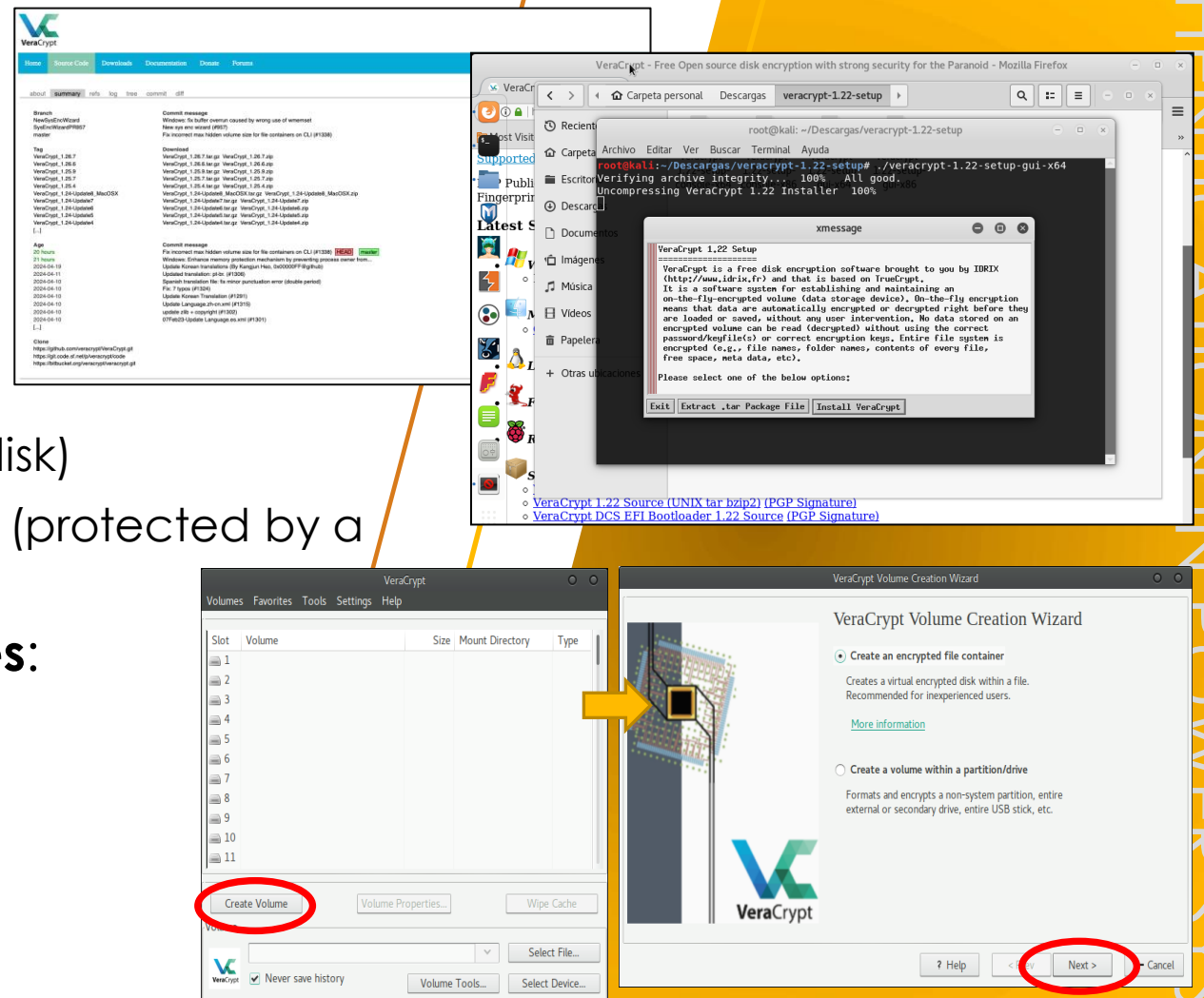
- Operating Systems normally support cryptographic applications to encrypt/decrypt sensitive data on hard disk and removable drive
  - Windows 10/11: **Bitlocker** (by default)
  - Linux and Windows: **Veracrypt, TrueCrypt** (open sources)
- The protection of these units usually follows common procedures:
  - **USB driver protection** relies on a password to obtain the 'key' needed to operate the unit by means of an encryption/decryption algorithm
  - **Hard disk protection** may require a Trusted Platform Module (TPM) 1.2 (or a USB driver) to obtain the key responsible for protecting the disk partition



SENSITIVE DATA PROTECTION

Veracrypt – an example of its usefulness

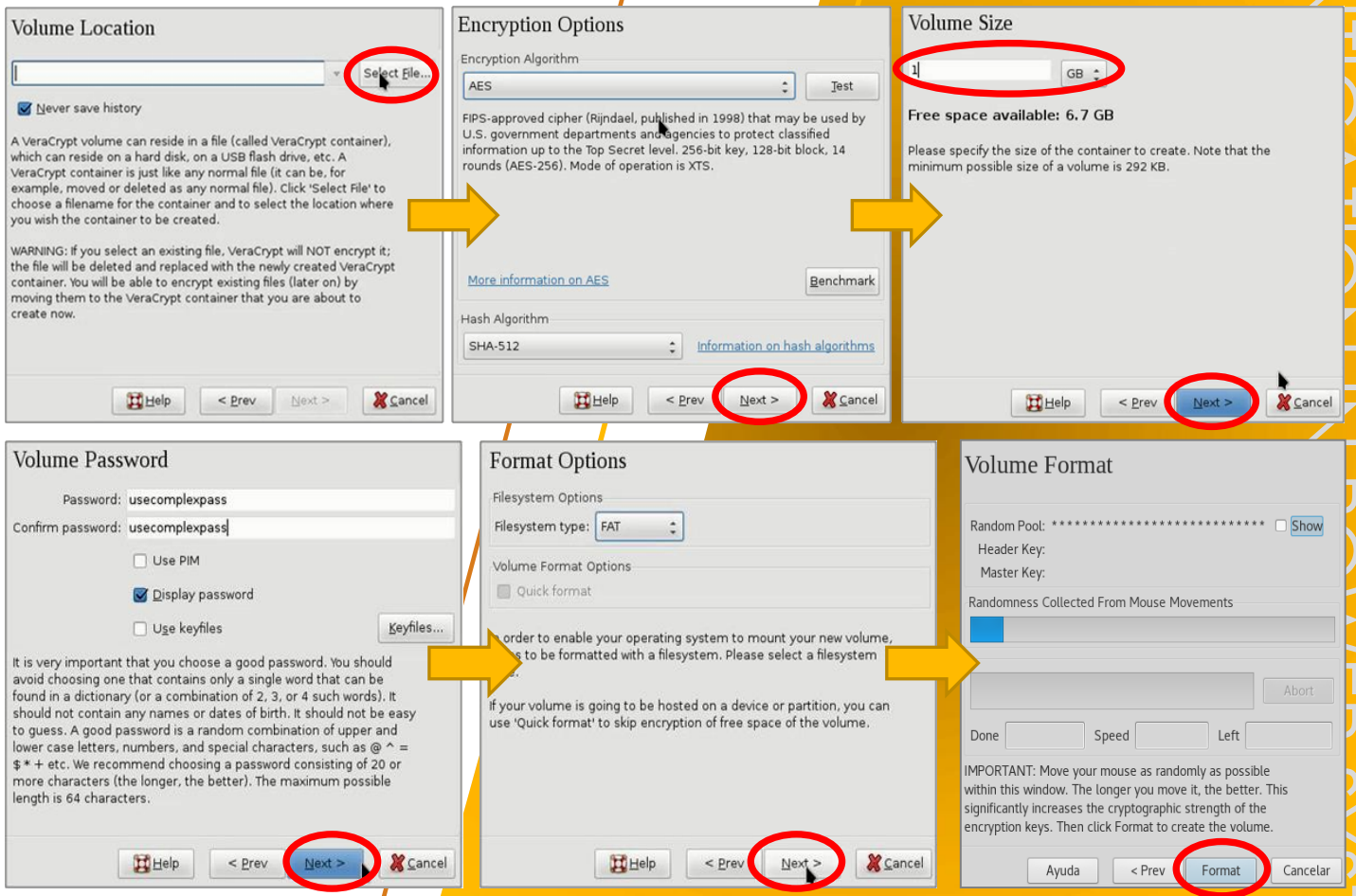
- **Open source:**
  - https://www.veracrypt.fr/code/VeraCrypt/
- **Main functions**:
  - Encryption of units (USB flash drive or hard disk)
  - Creation of data volumes and hidden units (protected by a password)
- Example of **creating secure data volumes**:
  1. Select '*encrypted file container*', to store the data inside an encrypted virtual volume in a single file



Source: Veracrypt, 2024
URL: https://www.veracrypt.fr/code/VeraCrypt/

Veracrypt – an example of its usefulness

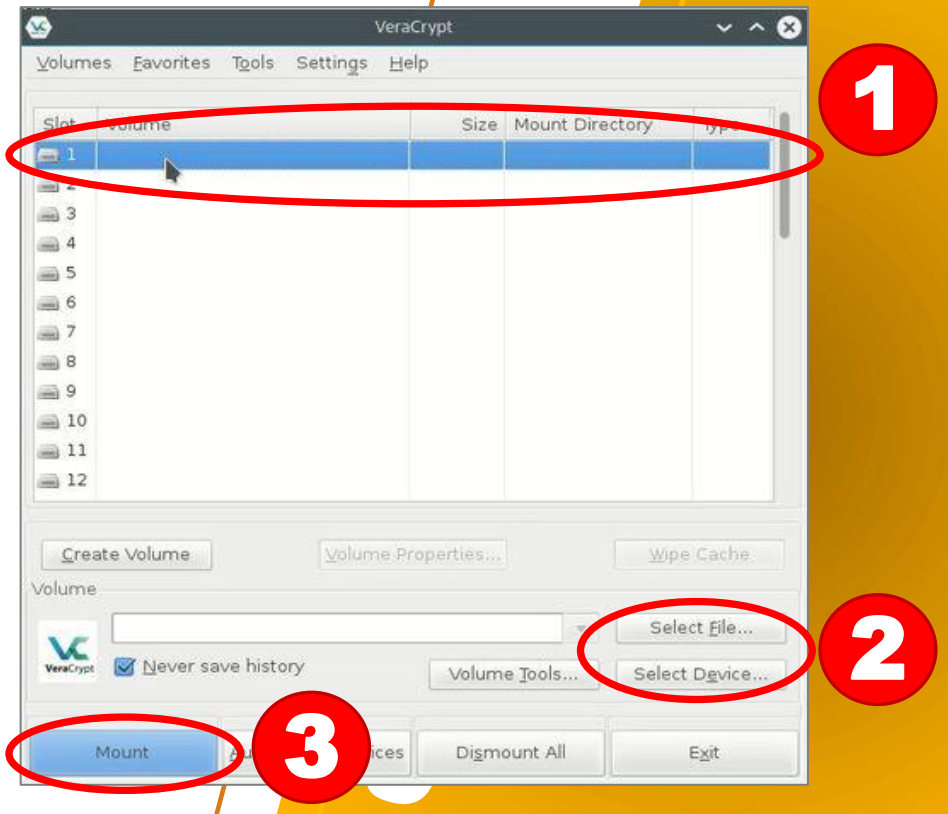
2. Choose the location of the file, the encryption (e.g. AES) and hash (e.g. SHA-512) algorithms, as well as the size of the file
3. Choose the password and the format of the volume contained in the file (e.g. FAT, ExFAT, NTFS...)
   - At this point, it is required to move the mouse in random directions to create "entropy", which is necessary for the key generation



Source: Veracrypt, 2024
URL: https://www.veracrypt.fr/code/VeraCrypt/

Veracrypt – an example of its usefulness

- Example of its use
  1. Select a slot
  2. Choose the file that contains the encrypted data volume
  3. Mount the file, indicating the password



Source: Veracrypt, 2024
URL: https://www.veracrypt.fr/code/VeraCrypt/

Empowering skills with OpenSSL

- **OpenSSL** is a file protection alternative
  - Open-Source tool that includes a general-purpose cryptographic library and mechanisms to be applied in command line
  - OpenSSL commands are generic and valid for any platform
- Commands:
  - Generic:
    - **$ openssl help**
    - **$ openssl list-standard-commands**
    - **$ openssl list-message-digest-commands**
    - **$ openssl list-cipher-commands**
  - Symmetric encryption:
    - **$ openssl enc –cipher-operation mode -in file.txt**
    - **$ openssl enc –cipher-operation mode –iv IV –K key –in file.txt**
  - Symmetric decryption:
    - **$ openssl enc –cipher-operation mode -in file.txt -d**
    - **$ openssl enc –cipher-operation mode –iv IV –K key –in file.txt -d**

```
Standard commands
asn1parse         ca                ciphers           cmp
cms               crl               crl2pkcs7         dgst
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
fipsinstall       gendsa            genpkey           genrsa
help              info              kdf               list
mac               nseq              ocsp              passwd
pkcs12            pkcs7             pkcs8             pkey
pkeyparam         pkeyutl           prime             rand
rehash            req               rsa               rsautl
s_client          s_server          s_time            sess_id
smime             speed             spkac             srp
storeutl          ts                verify            version
x509

Message Digest commands (see the `dgst' command for more details)
blake2b512        blake2s256        md4               md5
mdc2              rmd160            sha1              sha224
sha256            sha3-224          sha3-256          sha3-384
sha3-512          sha384            sha512            sha512-224
sha512-256        shake128          shake256          sm3

Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
bf                bf-cbc            bf-cfb            bf-ecb
bf-ofb            camellia-128-cbc  camellia-128-ecb  camellia-192-cbc
camellia-192-ecb  camellia-256-cbc  camellia-256-ecb  cast
cast-cbc          cast5-cbc         cast5-cfb         cast5-ecb
cast5-ofb         des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
idea              idea-cbc          idea-cfb          idea-ecb
idea-ofb          rc2               rc2-40-cbc        rc2-64-cbc
rc2-cbc           rc2-cfb           rc2-ecb           rc2-ofb
rc4               rc4-40            seed              seed-cbc
seed-cfb          seed-ecb          seed-ofb          sm4-cbc
sm4-cfb           sm4-ctr           sm4-ecb           sm4-ofb
```

Empowering skills with OpenSSL

- Generate keys:
  - **openssl enc -aes-256-cbc –P –k key**
  - **$ openssl genrsa n-bits > private.key**
  - **$ openssl rsa -in private.key -pubout -out public.pubkey**
  - **$ openssl ecparam -out key.pem -name prime256v1 –genkey**
  - …
- Asymmetric encryption:
  - **$ openssl rsautl -encrypt -in input.txt -pubin -inkey public.pubkey -out ciphertext.txt**
- Asymmetric decryption:
  - **openssl rsautl -decrypt -in ciphertext.txt -out plaintext.txt -inkey private.key**
- Digital signature:
  - **$ openssl rsautl -sign -in plaintext.txt -out signature.txt -inkey private.key**
- Digital signature verification:
  - **openssl rsautl -verify -in signature.txt -out verification.txt -pubin -inkey public.pubkey**
- Hash
  - **$openssl dgst** [**-sha** | **-sha1** | **-mdc2** | **-ripemd160** | **-sha224** | **-sha256** | **-sha384** | **-sha512** | **-md2** | **-md4** | **-md5** | **-dss1**] [**-c**] [**-d**] [**-hex**] [**-binary**] [**-r**] [**-hmac arg**] [**-non-fips-allow**] [**-out filename**] [**-sign filename**] [**-keyform arg**] [**-passin arg**] [**-verify filename**] [**-prverify filename**] [**-signature filename**] [**-hmac key**] [**-non-fips-allow**] [**-fips-fingerprint**] [**file...**]

Final remarks

- We have explored the advantages that cryptography can bring to the protection of data in energy control systems, such as:
  - Substations, control networks, corporative networks
- This level of protection can be applied to:
  - **Data in transit**: between industrial equipment, IT devices and users, etc.
  - **Data at rest**: servers, repositories, tablets, HMIs, etc.
- For protection, we can apply:
  - Symmetric and asymmetric cryptography, or its combined, for confidentiality
  - Digital certificates for trust
  - Hash functions for integrity
  - MAC functions for authentication and integrity

… amazing, isn't it? ☺

References and sources

1. Netresec, "Captures files from 4SICS Geek Lounge", 2024
   URL: https://www.netresec.com/?page=PCAP4SICS
2. CS3Sthtlm, 2014-2020, accessed in 2024.
   URL: https://cs3sthlm.se
3. CloudShark "telnet-client-server.pcapng", accessed in 2024
   URL: https://www.cloudshark.org/captures/818ceaef07b8?filter=telnet
4. Buchanan, William J., Human-Readable Encryption Keys. Asecuritysite.com, 2024.
   URL: https://asecuritysite.com/encryption/plain
5. Wierk, Cryptii, 2024.
   URL: https://cryptii.com
6. Buchanan, William J., AES. Asecuritysite.com, 2024.
   URL: https://asecuritysite.com/symmetric/aes
7. Buchanan, William J., DES Cipher. Asecuritysite.com, 2024.
   URL: https://asecuritysite.com/symmetric/des
8. Buchanan, William J., 3DES Cipher. Asecuritysite.com, 2024.
   URL: https://asecuritysite.com/symmetric/threedes

References and sources

9. Buchanan, William J., Camellia cipher. Asecuritysite.com. 2024.
   URL: https://asecuritysite.com/symmetric/camellia
10. Buchanan, William J., Symmetric key modes: GCM, ChaCha20, CBC, CFB8, CFB, OFB, GCM, CTR, and XTS, Asecuritysite.com, 2024.
    URL:https://asecuritysite.com/symmetric/sym
11. CryptoTool.net, RSA Key Generator, 2024.
    URL: https://cryptotools.net/rsagen
12. Buchanan, William J, Hashing, Asecuritysite.com, 2024.
    URL: https://asecuritysite.com/encryption/md5
13. Veracrypt, 2024
    URL: https://www.veracrypt.fr/code/VeraCrypt/
14. DeepL Translator for Proofreading:
    URL: https://www.deepl.com/translator
15. Some figures are attributed from Vecteezy,
    URL: https://www.vecteezy.com/ - thanks !

Connect with CyberSecPro:
How to register and other practical information

1. Website:
   www.cybersecpro-project.eu
2. X (Twitter):
   https://twitter.com/CyberSecPro_eu
3. LinkedIn:
   https://www.linkedin.com/company/cybersecpro-euproject/

Thank you

If you have any questions, please do not hesitate to contact:

- Davide Ferraris
  Substitute Professor
  University of Malaga
  ferraris@uma.es