Elementi essenziali e gestione della sicurezza informatica per il settore energetico

CSP001_C_E

PRESENTAZIONE DI:
DAVIDE FERRARIS
UNIVERSITÀ DI MALAGA, SPAGNA

Ringraziamenti

- *Cofinanziato dall'Unione Europea. Le opinioni e i pareri espressi sono tuttavia esclusivamente quelli dell'autore/degli autori e non riflettono necessariamente quelli dell'Unione Europea o dell'HADEA. Né l'Unione Europea né l'autorità che ha concesso il finanziamento possono essere ritenute responsabili per essi.*
- *Accordo di progetto n. 101083594*

Argomento 6: Selezione e implementazione dei controlli di sicurezza per gli ambienti energetici

Panoramica

- Selezionare e implementare controlli di sicurezza adeguati in base alle alle esigenze specifiche dei sistemi energetici
- Implementare politiche di password complesse e autenticazione a più fattori (MFA) per proteggere gli account degli utenti
- Crittografare i dati sensibili inattivi e in transito per impedire accessi non autorizzati e violazioni dei dati
- Applicare regolarmente aggiornamenti di sicurezza e patch ai sistemi software per risolvere le vulnerabilità

Davide Ferraris, Università di Malaga, Spagna

Argomento 6: Selezione e implementazione dei controlli di sicurezza per gli ambienti energetici

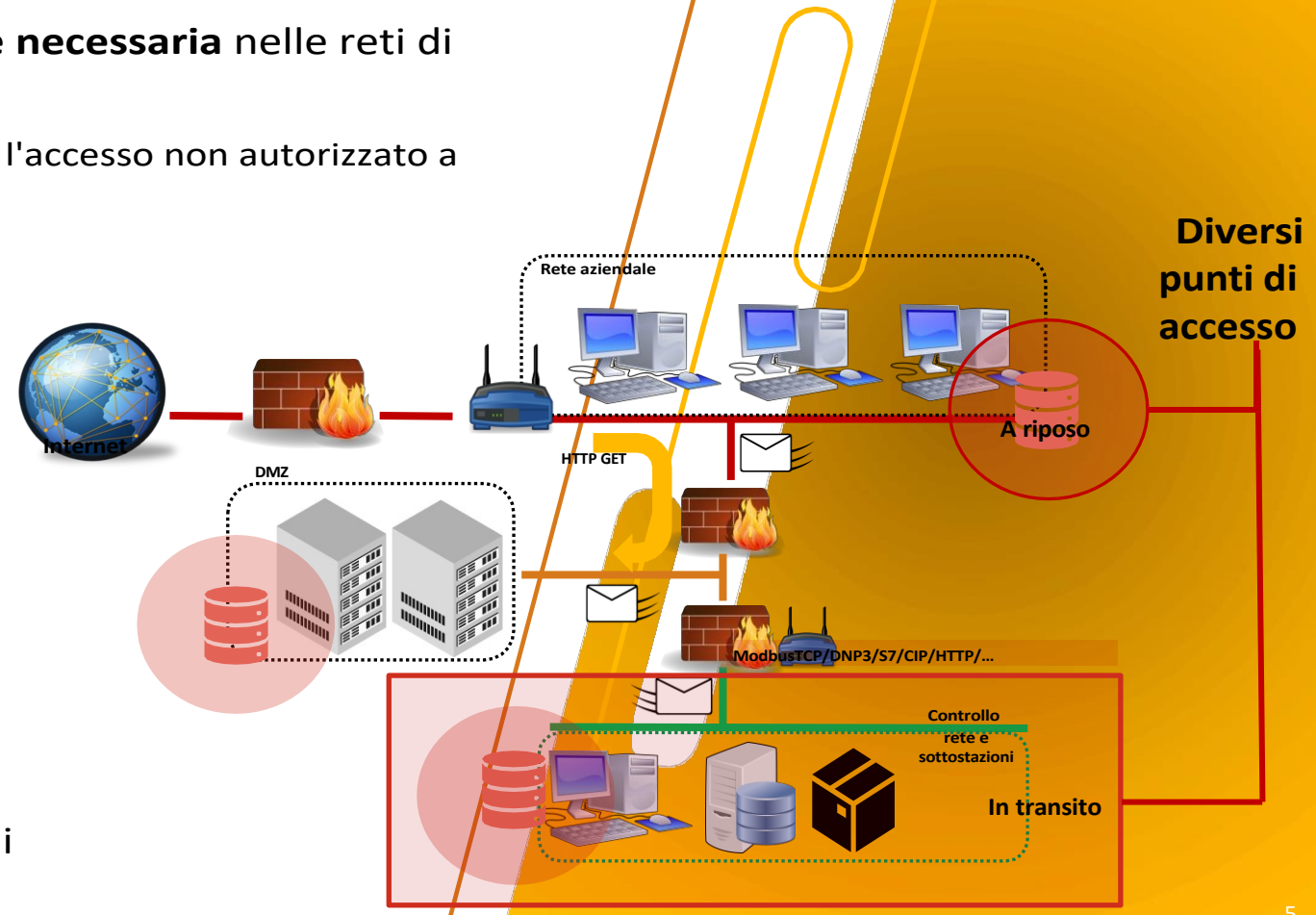
Panoramica

- Selezionare e implementare controlli di sicurezza adeguati in base alle alle esigenze specifiche dei sistemi energetici
- Implementare politiche di password complesse e autenticazione a più fattori (MFA) per proteggere gli account utente
- **Crittografare i dati sensibili inattivi e in transito per impedire accessi non autorizzati e violazioni dei dati**
- Applicare regolarmente aggiornamenti di sicurezza e patch ai sistemi software per risolvere le vulnerabilità

Davide Ferraris, Università di Malaga, Spagna

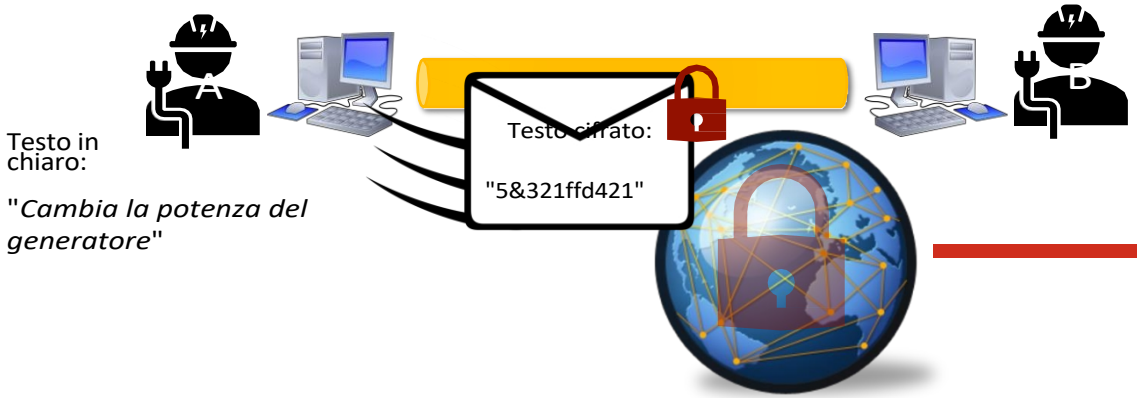
Crittografia nei sistemi di controllo dell'alimentazione

- **La crittografia è una misura di protezione necessaria** nelle reti di controllo dell'energia
  - Attraverso la crittografia è possibile impedire l'accesso non autorizzato a dati sensibili, garantendone **la riservatezza**
- Questi dati possono essere:
  - Istruzioni di comando e controllo
  - Misurazioni dei sensori
  - Dati di rete
  - Richieste degli utenti
  - Dati di autenticazione
  - Dati di autorizzazione
  - Scambio di chiavi
  - Ecc.
- Inoltre, tale protezione DEVE applicarsi sia ai
  - **Dati in transito e dati inattivi**



Crittografia applicata ai dati in transito

- **I canali di comunicazione** delle reti di controllo dell'energia DEVONO essere protetti mediante l'applicazione di algoritmi crittografici
  - Responsabile della protezione dall'accesso non autorizzato ai dati in transito



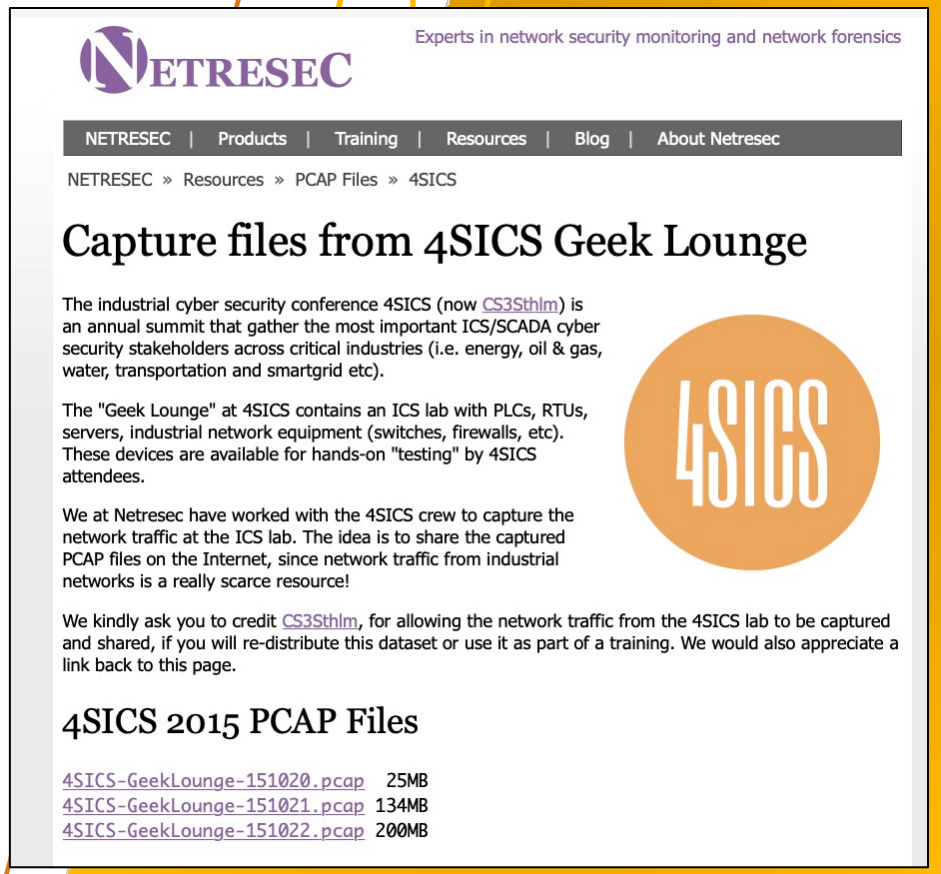
Si noti che nell'argomento 5 abbiamo già visto che tutti i

I protocolli VPN si basano su algoritmi crittografici per garantire la riservatezza dei canali di comunicazione, *che possono essere basati su protocolli industriali non sicuri come ModbusTCP*

  - Altrimenti, le comunicazioni verrebbero inviate in "testo semplice", consentendo agli aggressori con le risorse necessarie di intercettare, catturare e leggere le comunicazioni (intercettazioni)
  - Purtroppo, **l'intercettazione** può verificarsi nelle reti di controllo industriale, poiché la maggior parte dei dispositivi operativi legacy si basa ancora su protocolli non sicuri come **telnet** per il controllo remoto

Crittografia applicata ai dati in transito

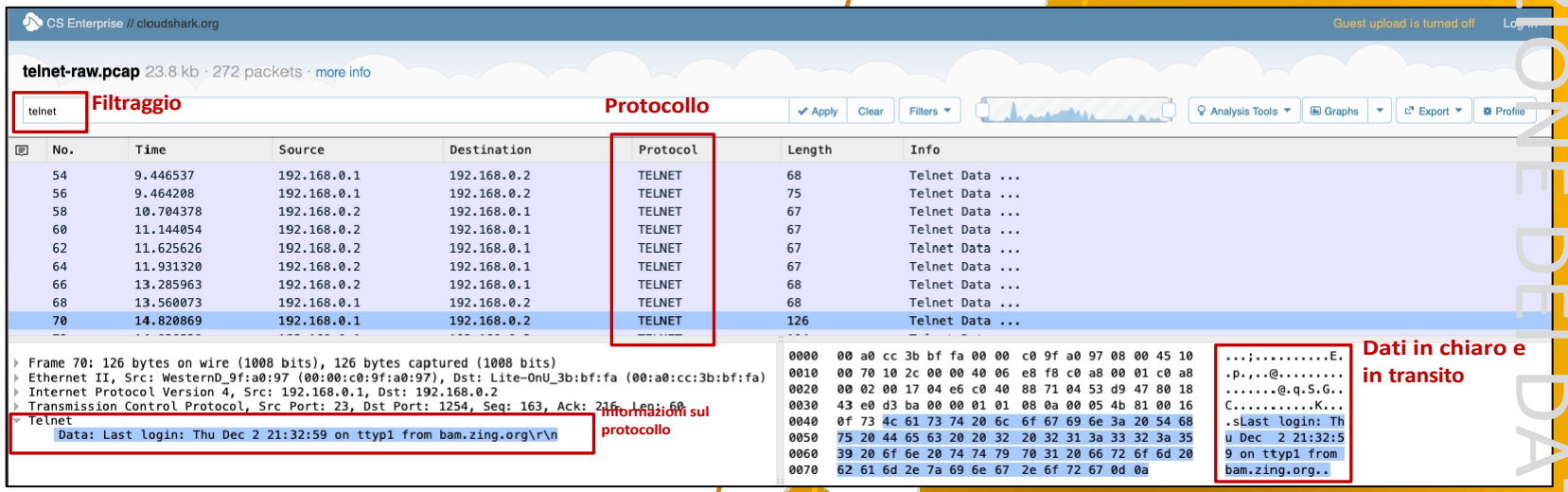
- Netresec offre **4SICS Geek Lounge**, che mette a disposizione alcuni file con acquisizioni di traffico industriale
  - URL: https://www.netresec.com/?page=PCAP4SICS
  - Le acquisizioni si basano su un laboratorio di sistemi di controllo industriale (ICS) con PLC, RTU, server e apparecchiature di rete industriale: *tutte le informazioni sul laboratorio ICS sono disponibili sullo stesso sito web*
  - Come richiesto, un ringraziamento speciale va a CS3Shtlm per aver consentito alla comunità di accedere a queste acquisizioni
  - Altre acquisizioni di rete SCADA/ICS sono disponibili anche all'indirizzo: https://www.netresec.com/?page=PcapFiles
- Con **Wireshark** è possibile analizzare il filtraggio del traffico telnet
  - Un'attività interessante potrebbe quindi essere quella di scaricare gli screenshot e filtrarli per "telnet"



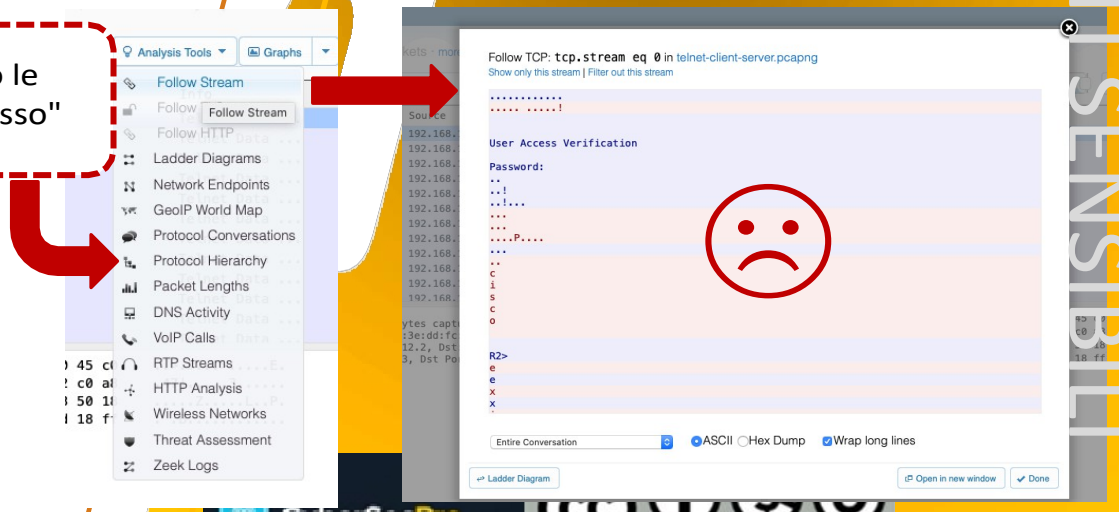
Fonte e fonte dell'immagine: Netresec, "File di acquisizione da 4SICS Geek Lounge", 2024 URL: https://www.netresec.com/?page=PCAP4SICS
Fonte: CS3Sthtlm, 2014-2020, consultato nel 2024. URL: https://cs3sthlm.se

Crittografia applicata ai dati in transito

- Per semplicità, consideriamo la seguente acquisizione telnet fornita da **CloudShark**:
  - Cattura del traffico: https://www.cloudshark.org/captures/818ceaef07b8
- CloudShark:
  - Si tratta di un dissector online in grado di visualizzare le acquisizioni del traffico di rete, presentando un pannello di controllo simile a Wireshark
  - URL: https://www.qacafe.com/analysis-tools/cloudshark/



È inoltre possibile semplificare ulteriormente l'analisi utilizzando le opzioni esistenti, come "Segui flusso"



Fonte e fonte della figura: CloudShark "telnet-client-server.pcapng", consultato nel 2024
URL: https://www.cloudshark.org/captures/818ceaef07b8?filter=telnet

Crittografia applicata ai dati in transito

- Esiste quindi una **chiara necessità di proteggere** i canali **di comunicazione** nei sistemi di controllo dell'energia, dove devono coesistere più apparecchiature industriali per il controllo

- Questo tipo di protezione prevede:
  - Una fase **di crittografia**, in cui i dati vengono protetti combinando il "testo in chiaro" con un "segreto", ottenendo un "testo cifrato"
  - Una fase **di decrittografia**, in cui il "testo cifrato" viene trasformato in il "testo in chiaro" originale tramite un segreto
- Questo **segreto** = la "CHIAVE"

Crittografia: tipi

- Nell'ambito della crittografia moderna sono emerse tre aree crittografiche rilevanti

Simmetrica

- Gli algoritmi crittografici simmetrici utilizzano la **stessa chiave** sia per la crittografia che per la decrittografia

Asimmetrica / chiave pubblica

- Gli algoritmi crittografici asimmetrici utilizzano **due chiavi diverse** per eseguire il processo di crittografia e decrittografia

Ibrido

- La tecnica ibrida mira a **combinare gli algoritmi asimmetrici e simmetrici**

Crittografia: Tipi

- Nell'ambito della crittografia moderna sono emerse tre aree crittografiche rilevanti

Simmetrica

- Gli algoritmi crittografici simmetrici utilizzano la **stessa chiave** sia per la crittografia che per la decrittografia

Asimmetrica / chiave pubblica

- Gli algoritmi crittografici asimmetrici utilizzano **due chiavi diverse** per eseguire il processo di crittografia e decrittografia

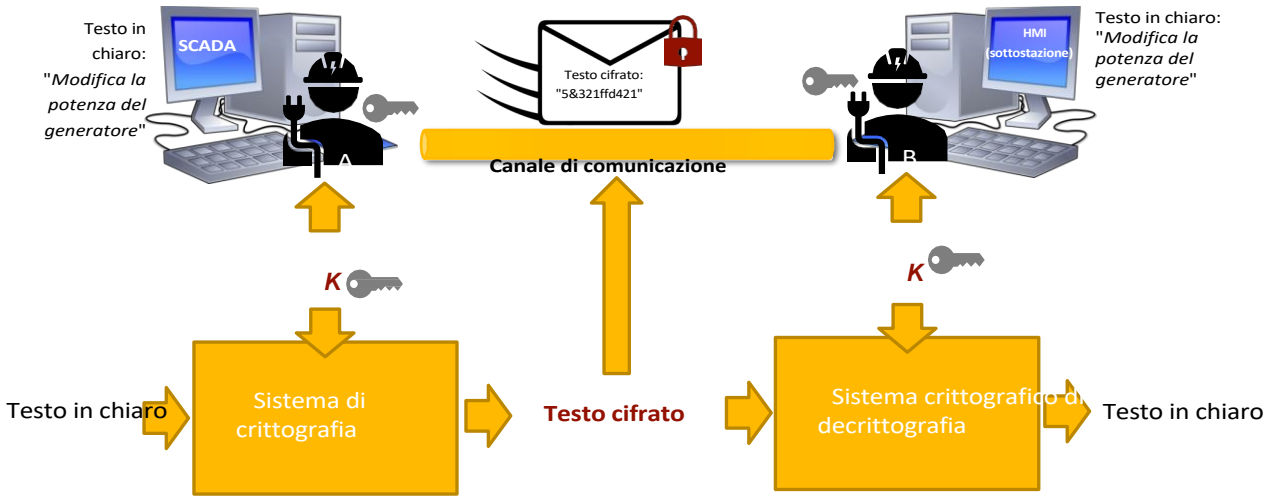
Ibrida

- La tecnica ibrida mira a combinare l'asimmetrico e il algoritmi simmetrici

PROTEZIONE DEI DATI SENSIBILI

Crittografia simmetrica: procedura

- Fondamentalmente, la tecnica mira a:
  1. Gli amministratori IT/OT, le apparecchiature di rete industriali o i processi DEVONO prima concordare le condizioni di crittografia, quali:
     - Tipo di algoritmo di crittografia/decrittografia
     - La chiave di "sessione", in modo tale che una delle due entità generi la chiave utilizzando un generatore (pseudo-casuale)
  2. Una delle entità coinvolte crittografa il messaggio in base alle condizioni prestabilite e invia il messaggio crittografato



Fonte e fonte dell'immagine: Buchanan, William J., Chiavi di crittografia leggibili dall'uomo. Asecuritysite.com, 2024.
URL: https://asecuritysite.com/encryption/plain

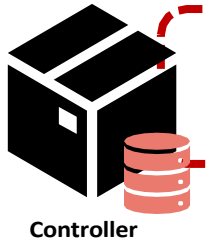
Crittografia simmetrica: sistemi crittografici

Caratteristiche	DES	3DES	AES	Camellia
Nome	Standard di crittografia dei dati	Triplo DES	*Standard di crittografia avanzato*	Camellia
Autore	IBM	IBM	*Vincent Rijmen, Joan Daemen*	Mitsubishi Electric e NTT
Lunghezza chiave	56 bit	168 (3 chiavi) o 112 (2 chiavi)	*128, 192, 256 bit*	128, 192, 256 bit
Dimensione blocco di cifratura	64 bit	64 bit	*128 bit*	128 bit
Velocità di elaborazione	Lenta	Molto lenta	*Veloce*	Lenta
Sicurezza	Bassa – Non consigliata	Media	*Alta*	Alta

Entrambi sono i sistemi crittografici più diffusi e consigliati in transito e a riposo - entrambi fanno parte di TLS-v1.2/1.3

Crittografia simmetrica: sistemi crittografici

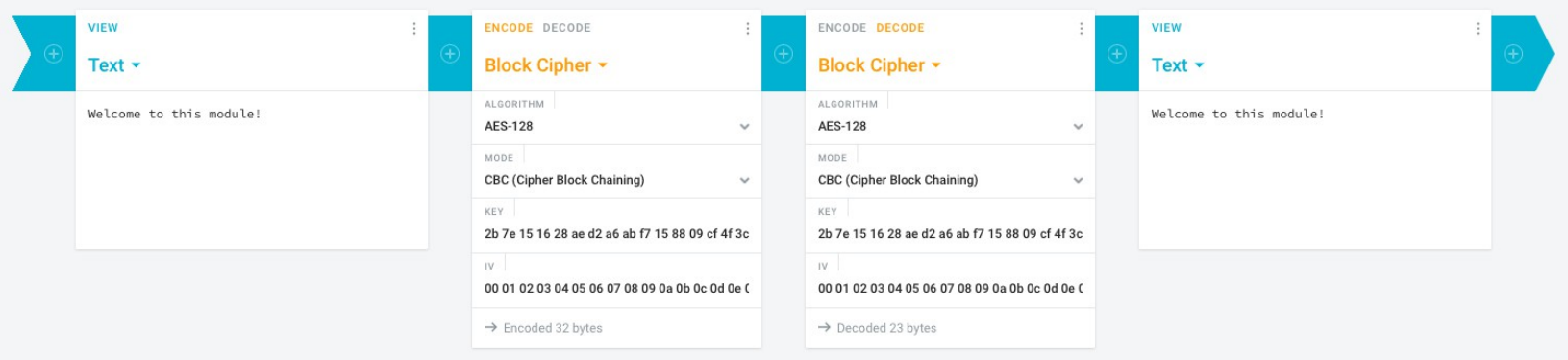
Caratteristiche	DES	3DES	AES	Camellia
Nome	Standard di crittografia dei dati	Triplo DES	*Standard di crittografia avanzato*	Camellia
Autore	IBM	IBM	*Vincent Rijmen, Joan Daemen*	Mitsubishi Electric e NTT
Lunghezza chiave	56 bit	168 (3 clave) o 112 (2 chiavi)	*128, 192, 256 bit*	128, 192, 256 bit
Dimensione blocco di cifratura	64 bit	64 bit	*128 bit*	128 bit
Velocità di elaborazione	Lenta	Molto lenta	*Veloce*	Lenta
Sicurezza	Bassa – Non consigliata	Media	*Alta*	Alta



Tuttavia, non possiamo dimenticare che esistono ancora dispositivi operativi legacy che applicano questi sistemi crittografici

Crittografia simmetrica: una visione pratica

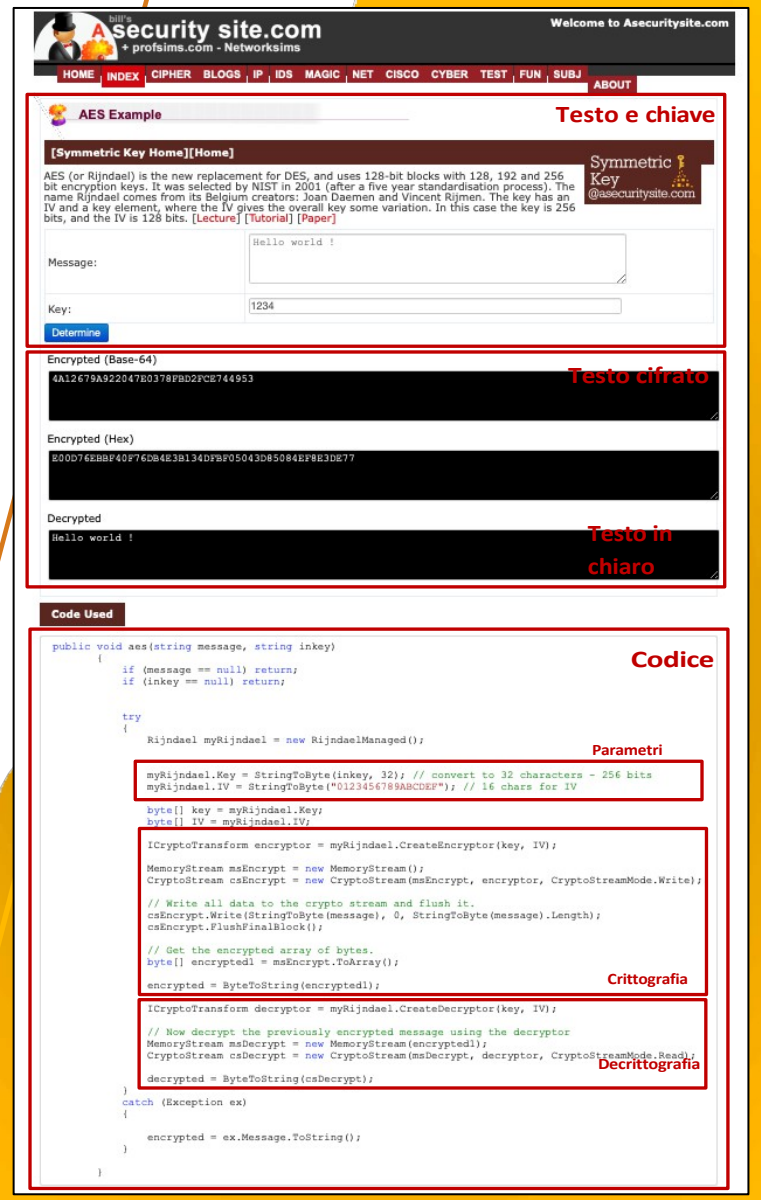
- Attraverso lo strumento online **cryptii** (https://cryptii.com), possiamo mettere in pratica le lezioni apprese
  - *Vai a "Crittografia moderna" > Crittografia a blocchi*
- Esercizi:
  1. Crittografa il seguente messaggio "*Benvenuto in questo modulo!*", utilizzando la seguente chiave: *2b7e151628aed2a6abf7158809cf4f3c*
  2. Decrittografa il testo cifrato utilizzando la stessa chiave e le stesse condizioni di crittografia
- Cosa sta succedendo?
  - Wuala! ☺
  - Possiamo crittografare i messaggi e recuperare il messaggio originale



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

Crittografia simmetrica: una visione pratica

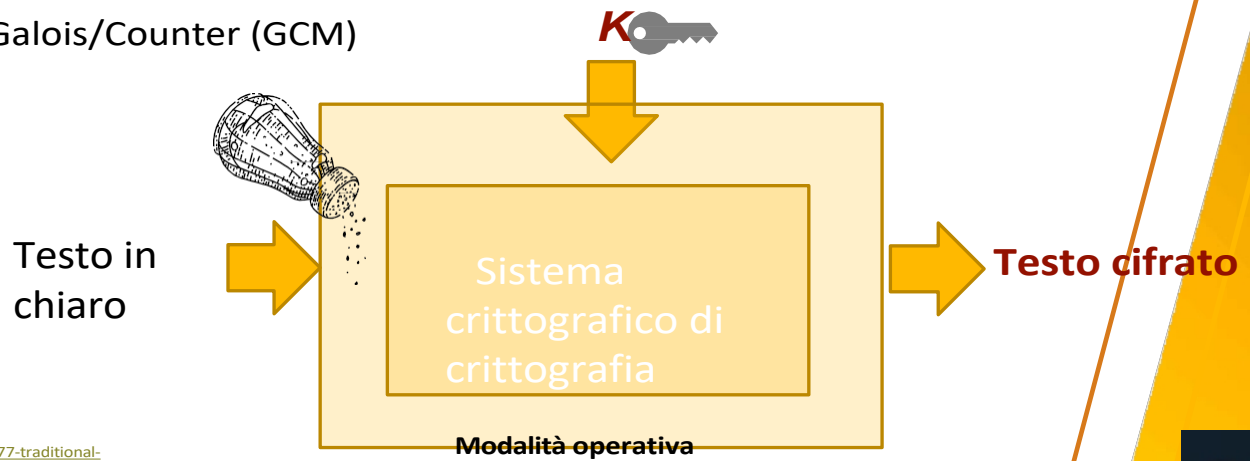
- Possiamo migliorare le nostre conoscenze tecniche ed estendere le nostre "COMPETENZE" prendendo in considerazione lo strumento online di **Asecuritysite**
  - AES: https://asecuritysite.com/symmetric/aes
  - DES: https://asecuritysite.com/symmetric/des
  - 3DES: https://asecuritysite.com/symmetric/threedes
  - Camelia: https://asecuritysite.com/symmetric/camellia
- Esercizi:
  1. Ottenere la chiave di sessione, dove il seme (una chiave preliminare speciale) per la generazione della chiave potrebbe essere "1234".
  2. Crittografa e decrittografa un messaggio, ad esempio "Hello World!".
  3. Se hai tempo, dai un'occhiata al codice allegato nella stessa pagina web



Fonte e fonte della figura: Buchanan, William J., *AES*. Asecuritysite.com, 2024. URL: https://asecuritysite.com/symmetric/aes
Fonte: Buchanan, William J., *DES Cipher*. Asecuritysite.com, 2024. URL: https://asecuritysite.com/symmetric/des
Fonte: Buchanan, William J., 3DES Cipher. Asecuritysite.com, 2024. URL: https://asecuritysite.com/symmetric/threedes
Fonte: Buchanan, William J., Camellia cipher. Asecuritysite.com. 2024. URL: https://asecuritysite.com/symmetric/camellia

Crittografia simmetrica: "Modalità operative"

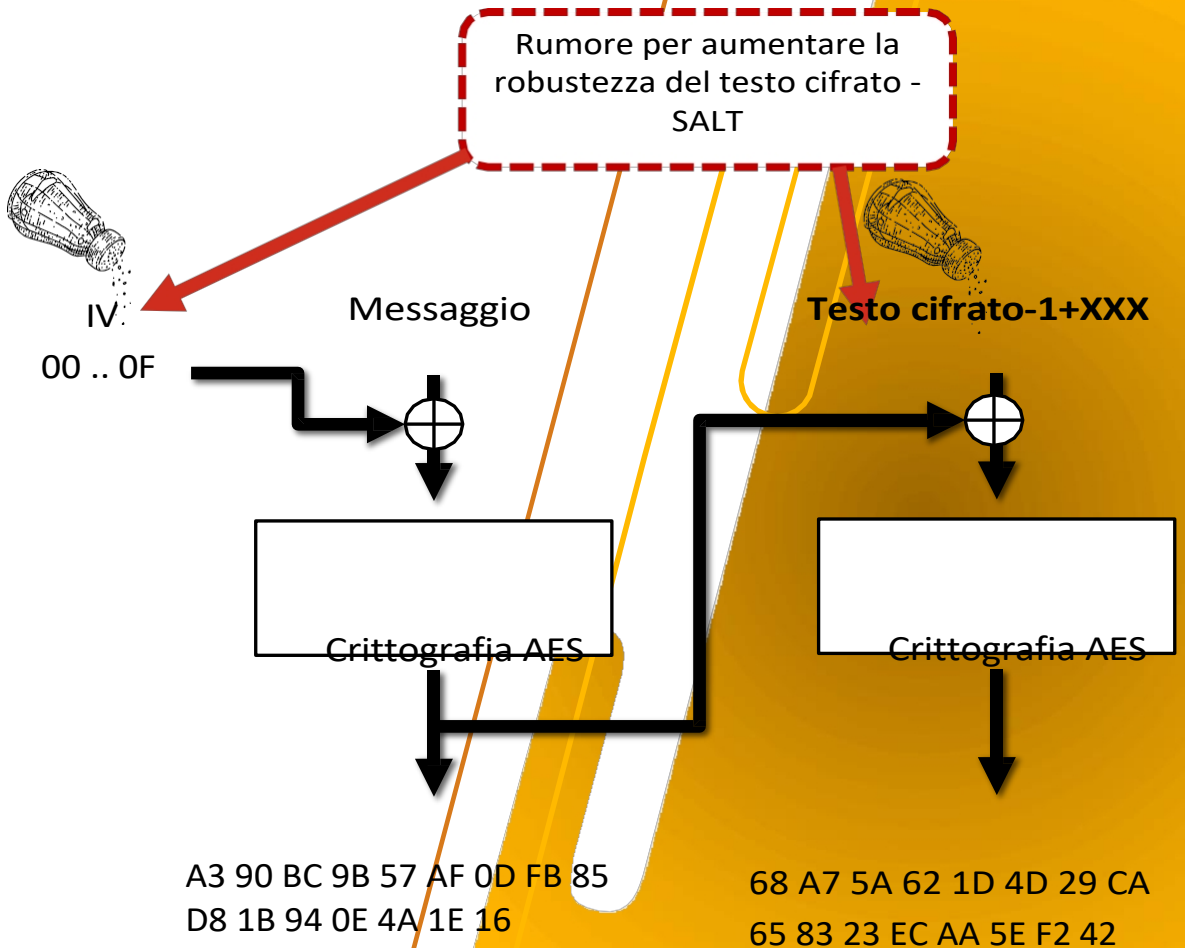
- Si noti che se gli algoritmi crittografici simmetrici fossero utilizzati così come sono, sarebbe possibile ottenere ogni volta lo stesso testo cifrato
  - Cioè, per lo stesso testo in chiaro come input, si otterrebbe sempre lo stesso testo cifrato come output!
- Per evitare questa situazione, vengono applicate **modalità di funzionamento** a cifratura a blocchi per **inserire rumore** (SALT) nel processo di crittografia:
  - Cipher Block Chaining (CBC)
  - Contatore (CTR)
  - Modalità Galois/Counter (GCM)



Fonte dell'immagine: Vecteezy
URL:https://www.vecteezy.com/vector-art/17407877-traditional-salt-shaker-sketch-hand-drawn-vector

Crittografia simmetrica: "Modalità operative"

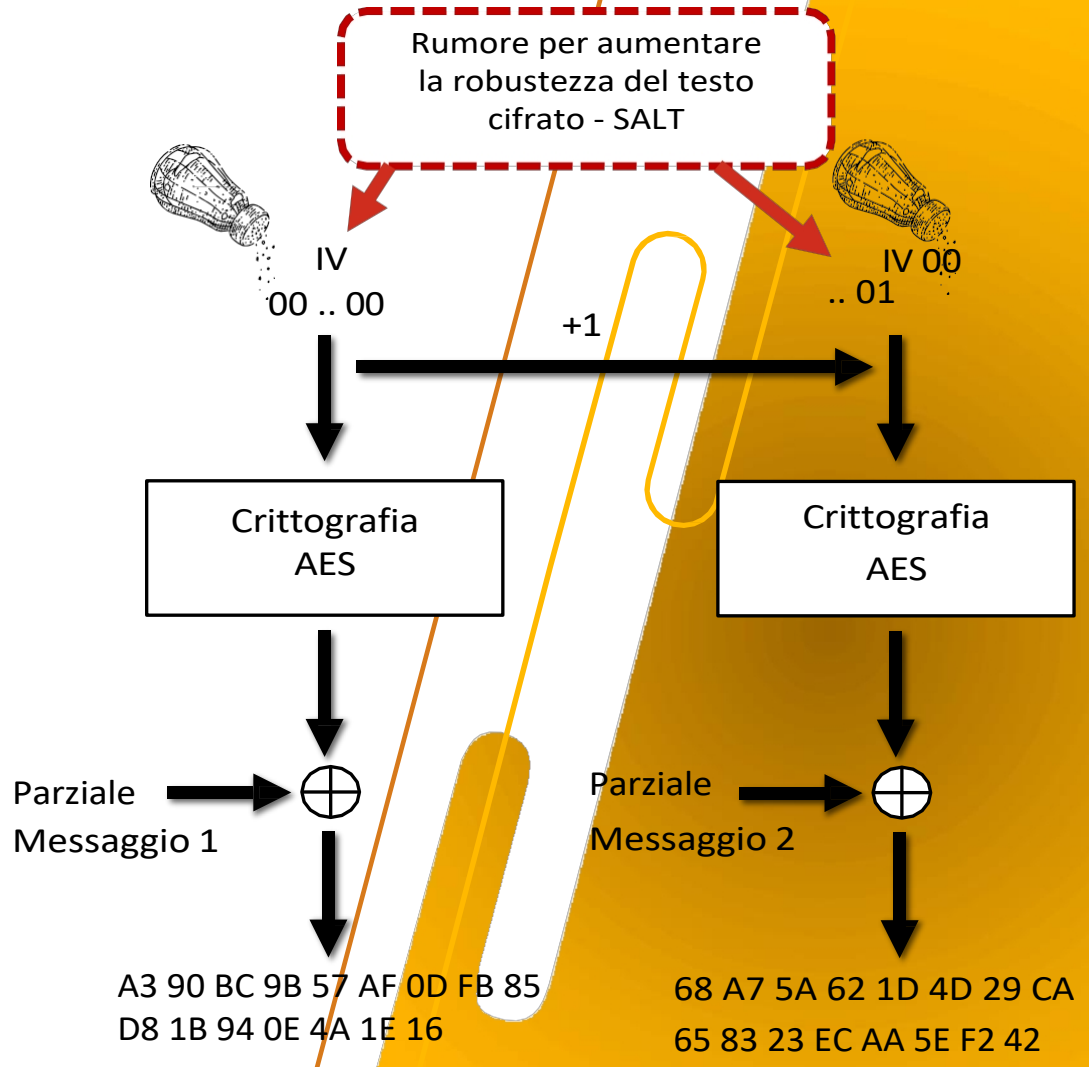
- **Modalità CBC**
  - L'output di ciascun blocco cifrato viene utilizzato per crittografare il blocco successivo
  - Utilizza un "vettore **di inizializzazione**" casuale
    (IV) come input per il primo cifrario - *fornisce variabilità (= SALT)*
    - Si noti che questo valore IV è pubblico (ad es. 00..0F) e deve essere inviato al nodo di destinazione insieme al testo cifrato
  - Normalmente, al termine del processo viene eseguita un'operazione di riempimento per facilitare il calcolo
    - "Padding dell'input (XXX) per ottenere un multiplo della dimensione del blocco



PROTEZIONE DEI DATI SENSIBILI

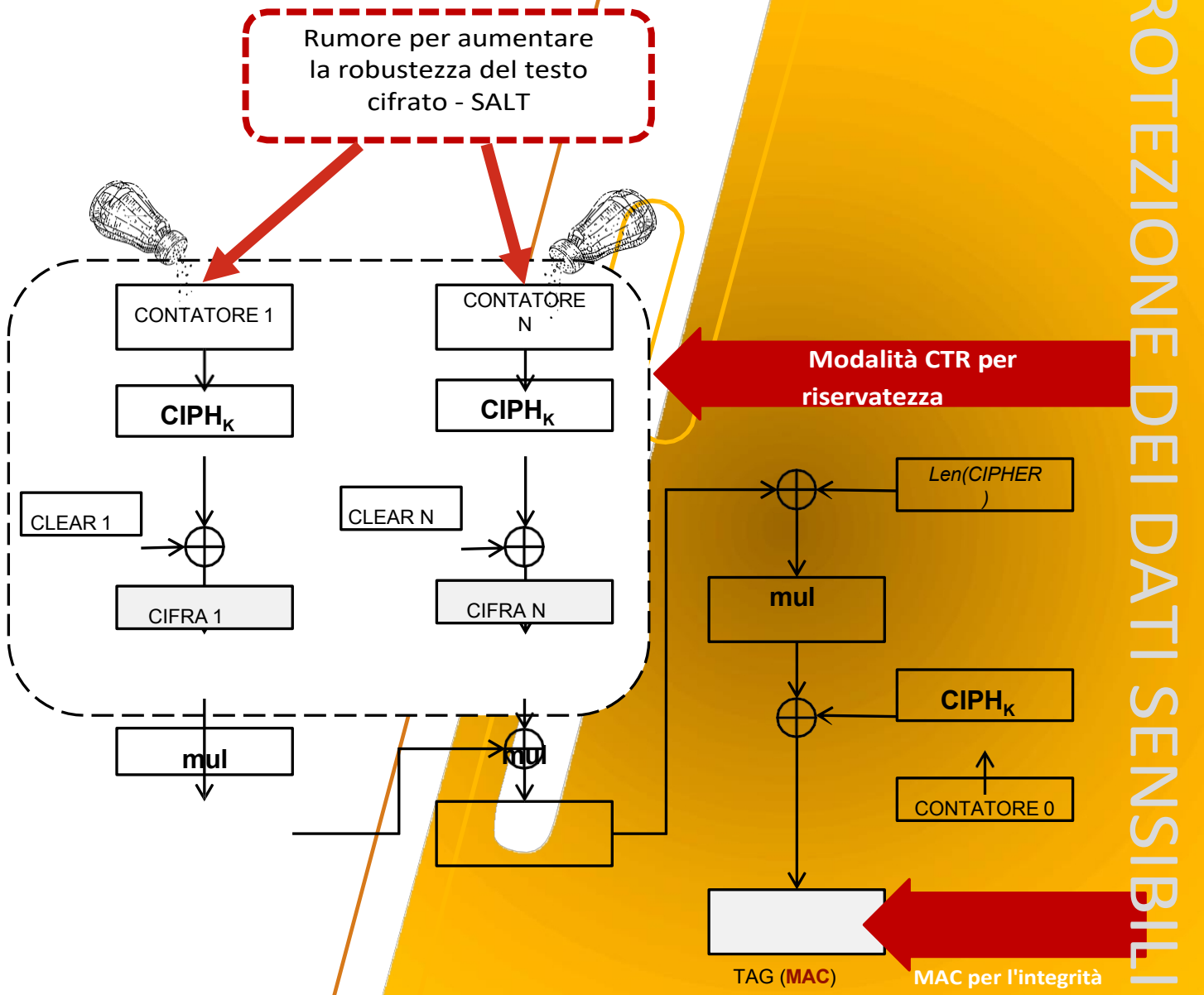
Crittografia simmetrica: "Modalità operative"

- **Modalità CTR**
  - Il risultato di ciascun blocco di cifratura fa parte del testo cifrato
  - Una volta completate tutte le operazioni, ogni parte del testo cifrato viene concatenata per fornire la parte finale del testo cifrato
  - Questo modello operativo utilizza anche un "**vettore di inizializzazione**" (IV) casuale come input per il primo cifrario
    - Il suo valore viene aumentato per le operazioni successive (= SALT)



Crittografia simmetrica: "Modalità operative"

- **Modalità GCM**
  - Utilizza un valore casuale ("nonce" / "contatore" / IV) come input per il primo cifrario - *fornisce variabilità (= SALT)*
    - Questo valore è pubblico e deve essere inviato insieme al testo cifrato
  - Non è necessaria alcuna operazione di riempimento
  - Fornisce un codice di integrità della cifratura ("Tag" / "MAC")
    - Una modifica nella crittografia cambia il codice di integrità
    - Queste caratteristiche sono conformi alla **Crittografia autenticata con dati aggiuntivi (AEAD)** – *riservatezza + integrità*

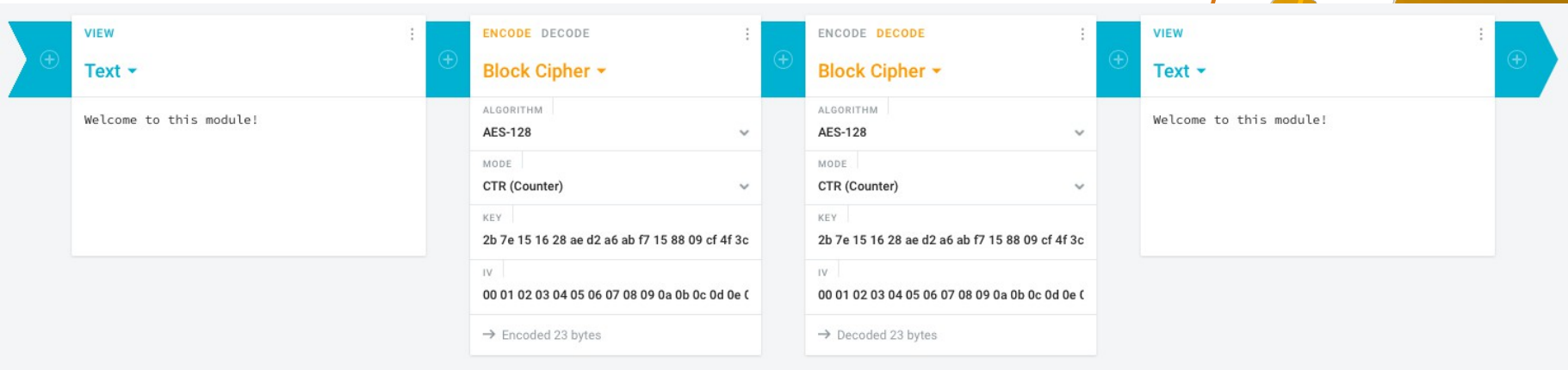


PROTEZIONE DEI DATI SENSIBILI

CSP001_C_E – ARGOMENTO 6: Davide Ferraris, Università di Malaga, Spagna

Modalità operative: una visione pratica

- Torna su **cryptii** alla sezione "*Crittografia moderna" > Crittografia a blocchi*
- Esercizi:
  1. Crittografa il seguente messaggio "*Benvenuto in questo modulo!*", utilizzando:
     - *Chiave: 2b7e151628aed2a6abf7158809cf4f3c*
     - *Modalità operativa: CTR*
     - *IV: 000102030405060708090a0b0c0d0e0f*
  2. Decrittografare il testo cifrato utilizzando le stesse condizioni di crittografia



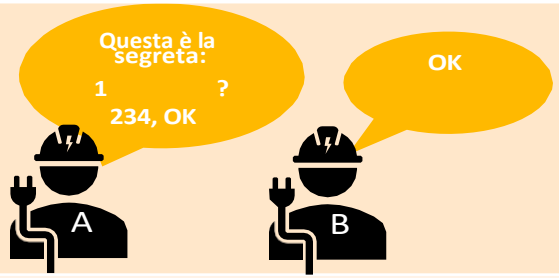
Fonte: Wierk, Cryptii, 2024.
URL: https://cryptii.com

Modalità operative: una visione pratica

- Torna nuovamente al sito **Asecuritysite** e in particolare a:
  - URL: https://asecuritysite.com/symmetric/sym
- Esercizi:
  1. Crittografa con AES+CBC, AES+CTR e AES+GCM
  2. Decrittografare il testo cifrato utilizzando le stesse condizioni di crittografia

Fonte e fonte della figura: Buchanan, William J., *Modalità a chiave simmetrica: GCM, ChaCha20, CBC, CFB8, CFB, OFB, GCM, CTR e XTS*, Asecuritysite.com, 2024. URL:https://asecuritysite.com/symmetric/sym

Crittografia simmetrica: vantaggi e svantaggi

Vantaggi	Svantaggi
• Gli algoritmi di crittografia simmetrica offrono in genere **prestazioni elevate a costi computazionali bassi** (o relativamente bassi)	• La chiave segreta ***K*** deve essere concordata a priori 
• Le chiavi utilizzate sono **relativamente brevi** • Si consiglia di utilizzare chiavi di **dimensioni pari o superiori a 128 bit** 	• Il **numero di chiavi può aumentare** in modo significativo se tutti gli utenti di una comunità devono comunicare (individualmente) con gli altri membri di quella comunità→ **(n * (n-1)) / 2**

Crittografia: Tipi

- All'interno della crittografia moderna sono emerse tre aree crittografiche rilevanti

Simmetrica

- Gli algoritmi crittografici simmetrici utilizzano la **stessa chiave** sia per la crittografia che per la decrittografia

Asimmetrica / chiave pubblica

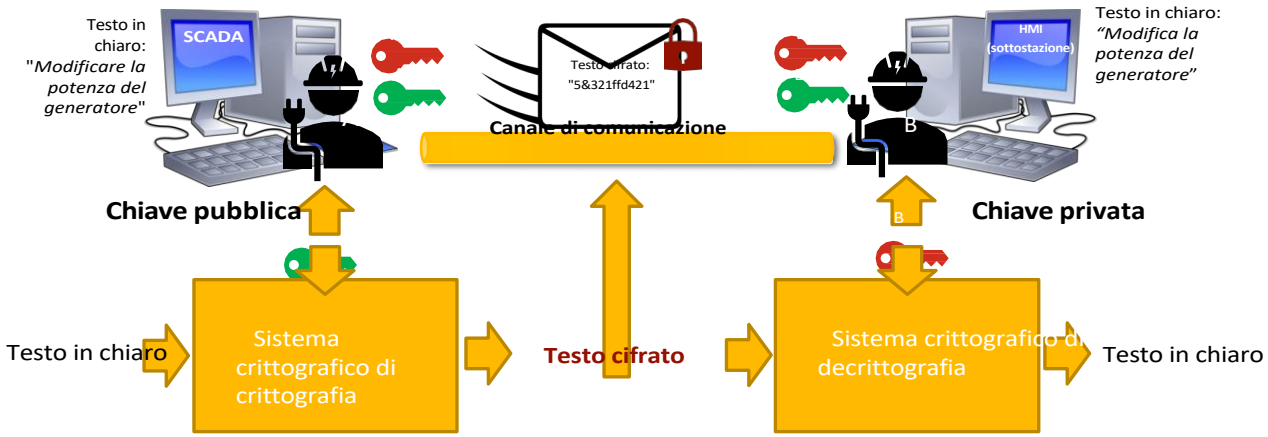
- Gli algoritmi crittografici asimmetrici utilizzano **due chiavi diverse** per eseguire il processo di crittografia e decrittografia

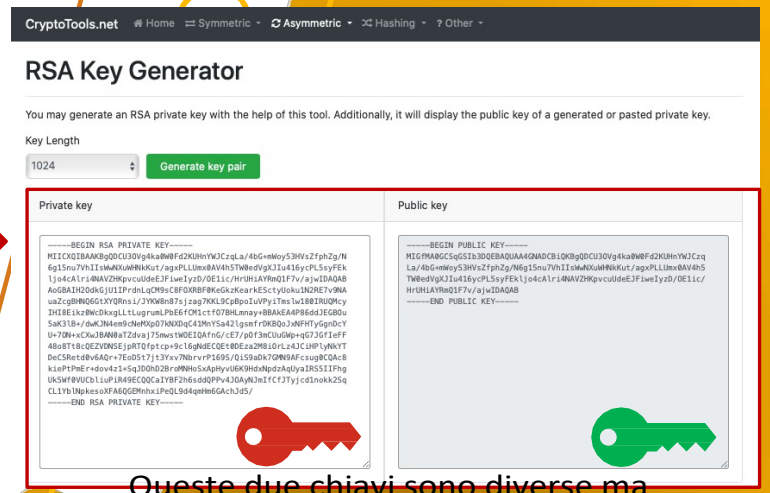
Ibrida

- La tecnica ibrida mira a combinare gli algoritmi asimmetrici e simmetrici

Crittografia asimmetrica: procedura

- Fondamentalmente, la tecnica mira a:
  1. Ogni amministratore IT/OT, apparecchiatura di rete industriale o processo DEVE creare due chiavi rilevanti:
     - **1 Chiave privata**: conosciuta solo dal proprietario della chiave
     - **1 Chiave pubblica**: nota a tutte le entità
  2. Una delle entità coinvolte **crittografa** il messaggio utilizzando la "chiave pubblica" della destinazione, poiché è l'unica entità in possesso della chiave privata



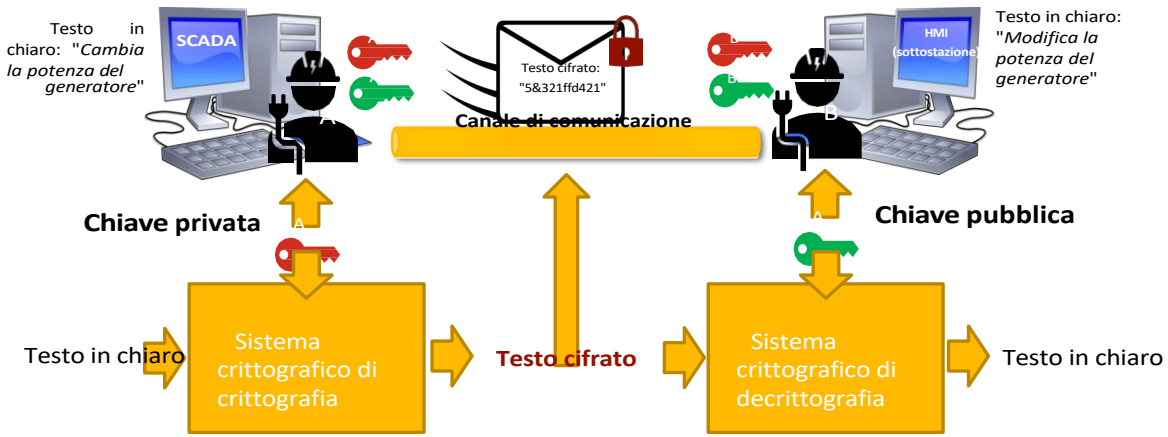


Queste due chiavi sono diverse ma strettamente correlate tra loro, consentendo la crittografia e la decrittografia delle informazioni

Fonte e fonte della figura: CryptoTool.net, RSA Key Generator, 2024.
URL: https://cryptotools.net/rsagen

Crittografia asimmetrica: procedura

- Tuttavia, se la chiave privata viene applicata per crittografare (firmare) il messaggio, ciò che si ottiene è una **firma digitale**



- Pertanto, a seconda di come vengono applicate le due chiavi, si ottiene:

Crittografia per la riservatezza	Firma digitale per l'autenticazione
• Crittografia con la chiave pubblica del destinatario • Decrittografare con la chiave privata del destinatario	• Crittografare con la chiave privata del mittente • Decrittografare con la chiave pubblica del mittente

Crittografia asimmetrica: sistemi crittografici

Caratteristiche	RSA	ECC
Nome	RSA	Crittografia a curva ellittica
Autore	Ron Rivest, Adi Shamir e Leonard Adleman	Neal Koblitz e Victor S. Miller
Lunghezza della chiave	1024, 2048 y 3072 bit	160, 224, 256, 384, 521 bit
Velocità di elaborazione	Lenta	Veloce
Sicurezza	Elevata	Elevata
Funzionalità	Crittografia, firma e scambio di chiavi	Crittografia, firma e scambio di chiavi

RSA è il più diffuso, ma ECC è ideale per operazioni con limitazioni computazionali dispositivi quali RTU/PLC, sensori e attuatori

Crittografia asimmetrica: vantaggi e svantaggi

Vantaggi	Svantaggi
• Tutte le entità coinvolte nel sistema di controllo energetico **possono generare le due chiavi senza doversi accordare preventivamente**	• Gli algoritmi di crittografia asimmetrica richiedono **capacità di calcolo** per elaborare i loro algoritmi di crittografia
• Il numero di chiavi è 2*n • **Riduce il costo di archiviazione** nei punti finali	• Le chiavi utilizzate sono relativamente grandi • Si **consiglia di utilizzare chiavi di dimensioni pari o superiori a 1024 bit**

Crittografia: tipi

- Nell'ambito della crittografia moderna sono emerse tre aree crittografiche rilevanti

Simmetrica

- Gli algoritmi crittografici simmetrici utilizzano la **stessa chiave** sia per la crittografia che per la decrittografia

Asimmetrica / chiave pubblica

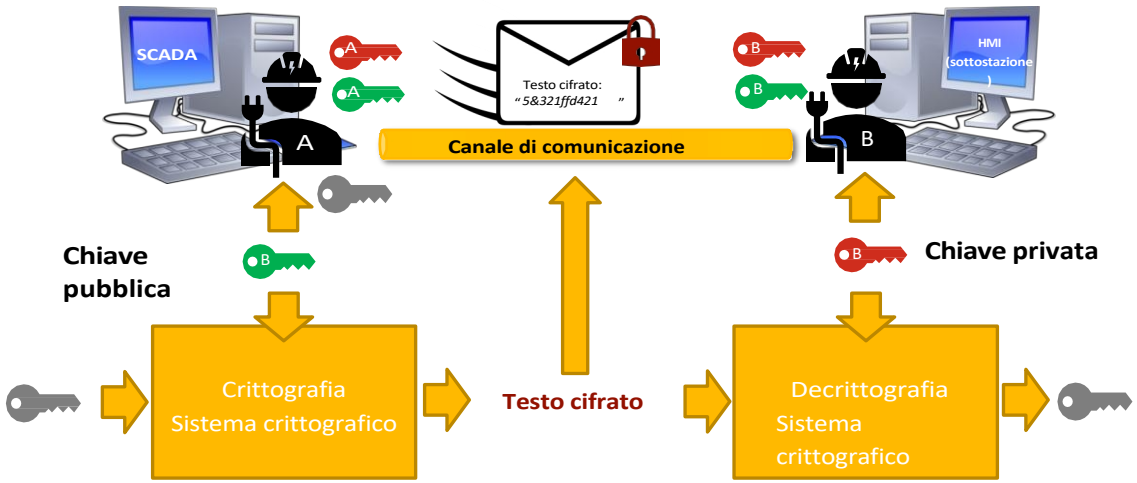
- Gli algoritmi crittografici asimmetrici utilizzano **due chiavi diverse** per eseguire il processo di crittografia e decrittografia

Ibrida

- La tecnica ibrida mira a **combinare gli algoritmi asimmetrici e simmetrici**

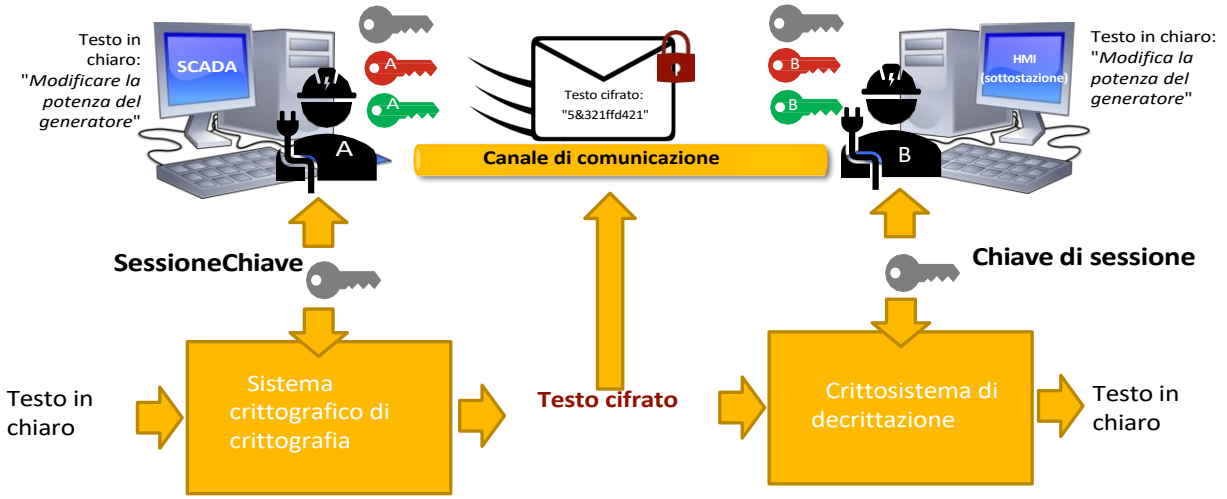
Crittografia ibrida: procedura

- La crittografia ibrida mira a ottenere i vantaggi offerti dai due approcci:
  - **Crittografia asimmetrica** per negoziare la chiave di sessione
  - **Crittografia simmetrica** per crittografare/decrittografare i messaggi
- Fondamentalmente, la tecnica mira a:
  1. Una delle entità genera la chiave di sessione e la invia al destinatario utilizzando la crittografia asimmetrica

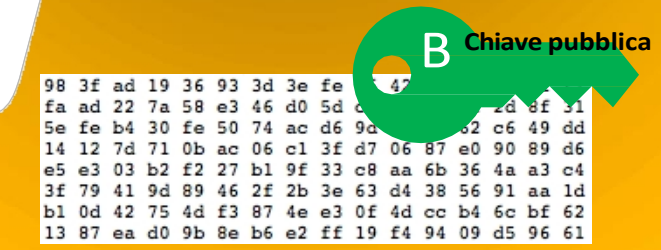


Crittografia ibrida: procedura

2. Una volta che il destinatario ha ottenuto la chiave di sessione, una delle entità genera un messaggio, che viene protetto utilizzando la chiave di sessione



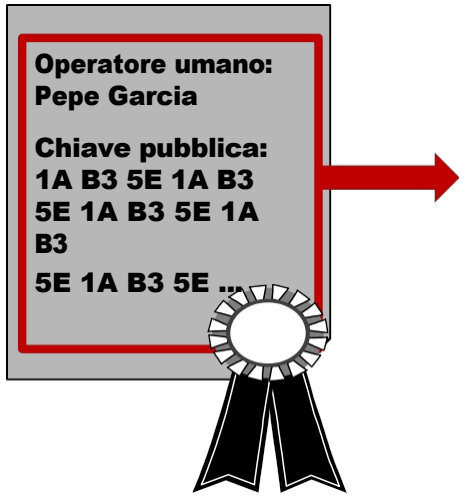
- Tuttavia, lo scambio di chiavi pubbliche senza ulteriori informazioni sull'entità, solo la chiave (valori esadecimali), comporta la necessità di gestire **certificati digitali** al fine di garantire l'affidabilità della chiave pubblica



Certificati digitali: x.509

- Un certificato digitale è un documento digitale che attesta che una chiave appartiene a un determinato soggetto, principalmente perché è certificato da un **ente affidabile** che convalida e firma il documento.
- L'ente incaricato della convalida e della certificazione è noto come **Autorità di Certificazione (CA)**

La CA dovrebbe essere l'ente con diritti legali per certificare il documento, come i gestori o i direttori del sistema, ecc.

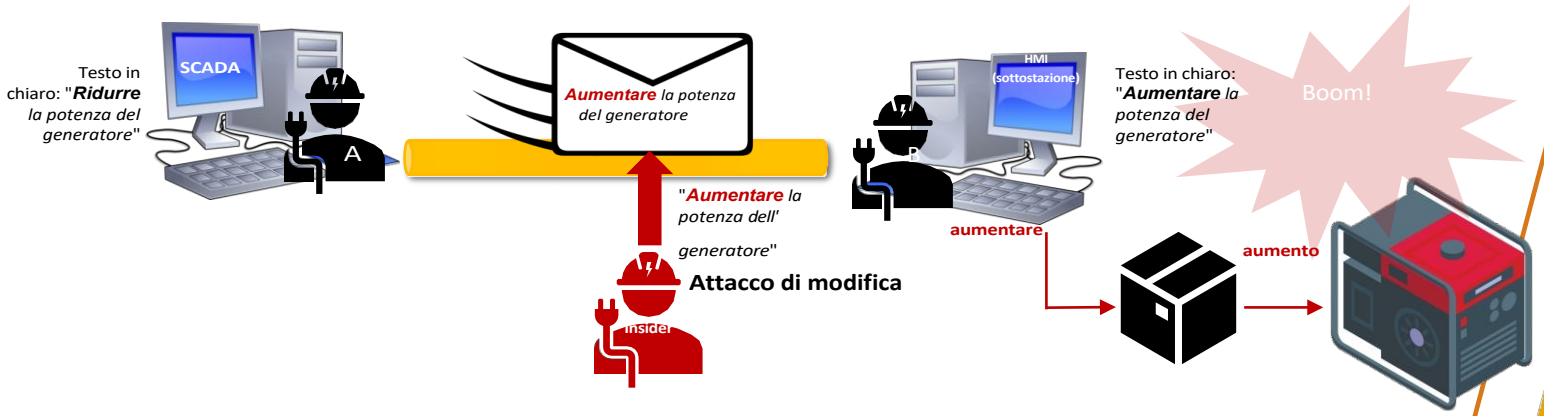


Il documento contiene almeno

- Numero di serie univoco
- L'identità dell'utente a cui si riferiscono le vengono fornite le informazioni
- Il valore della chiave pubblica
- La data di scadenza del certificato
- L'identità dell'emittente del documento
- La firma digitale del documento emesso

Hash per l'integrità dei dati sensibili

- Un modo per rilevare modifiche indesiderate al contenuto dei messaggi sarebbe attraverso specifiche funzioni crittografiche per l'"integrità".

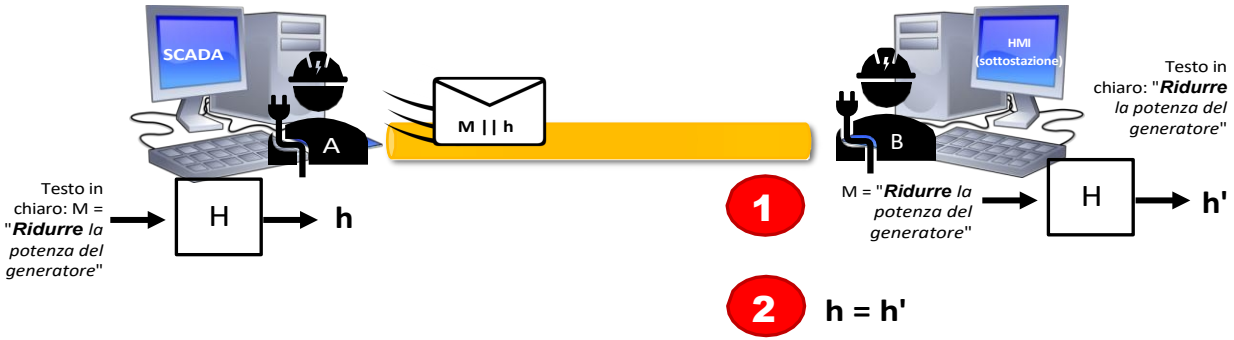


- **Le funzioni hash** sono i metodi di convalida più efficienti dal punto di vista crittografico
  - Queste funzioni sono **funzioni unidirezionali** che garantiscono che due messaggi diversi producano valori hash (o digest) diversi
  - In questo modo, garantiamo anche **l'assenza di collisioni** (la principale vulnerabilità di tali funzioni).

Fonte dell'immagine: Vecteezy
URL:https://www.vecteezy.com/vector-art/19053049-petrol-generator-icon-isometric-vector-industrial-equipment

Funzione hash: Procedura

- Fondamentalmente, la tecnica mira a:
  1. Il mittente prepara il messaggio e ottiene il valore hash corrispondente
  2. Sia il messaggio che il valore hash vengono inviati alla destinazione
  3. Il destinatario calcola il valore hash del messaggio ricevuto e lo confronta con il valore hash ricevuto
  4. Se entrambi gli hash sono uguali, il destinatario conferma l' integrità del messaggio



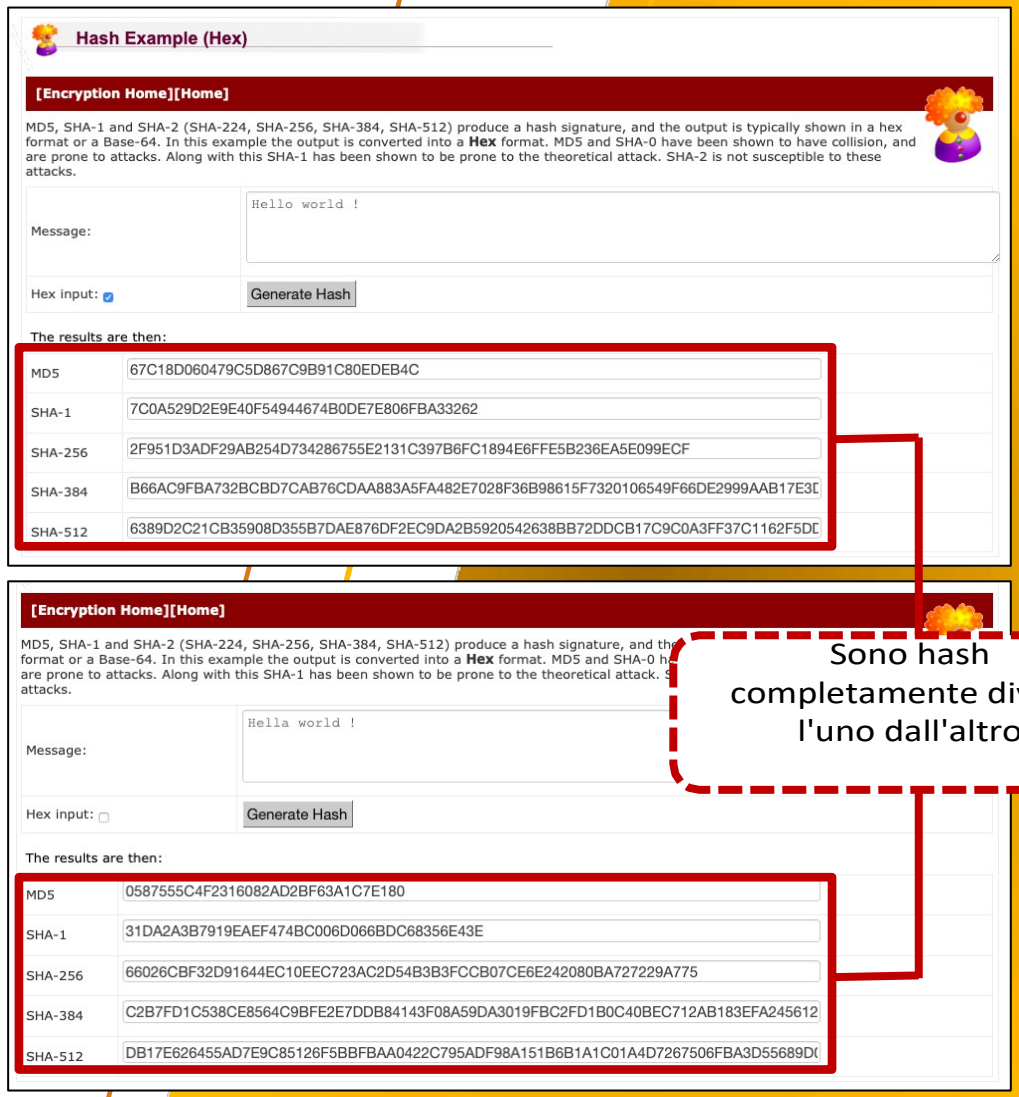
Funzione hash: una visione pratica

- Vai su **Asecuritysite**, e in particolare su:
  - URL: https://asecuritysite.com/encryption/md5

- Esercizi:
  1. Genera l'hash di un messaggio
  2. Modifica un carattere del messaggio
  3. Confronta i valori hash
- Ripeti l'esercizio con **criptii**

- È importante notare che alcune funzioni hash sono soggette a collisioni, come ad esempio: MD5 e SHA-1
  - Sfortunatamente, **i dispositivi operativi legacy possono ancora supportarli**

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

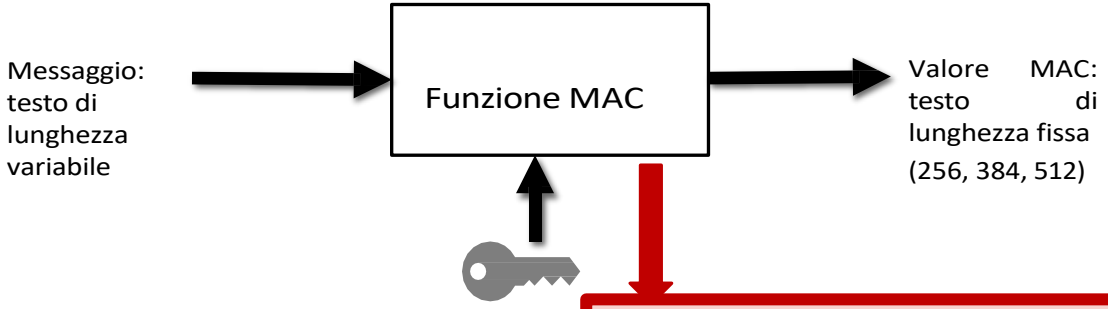


Sono hash completamente diversi l'uno dall'altro

PROTEZIONE DEI DATI SENSIBILI

MAC per l'integrità e l'autenticazione dei dati

- **La funzione Message Authentication Code (MAC)** svolge la stessa funzione della funzione hash, ma considerando come valori di input sia:
  - Il messaggio e
  - Una CHIAVE simmetrica condivisa dalle due entità della comunicazione



Esistono diverse funzioni MAC, come **i codici di autenticazione dei messaggi basati su hash (HMAC)**

- Garantite dal MAC:
  - **Integrità** tramite la funzione hash (ad es. HMAC)
  - **Autenticazione** tramite chiave segreta

Come nei punti precedenti, gli studenti possono esercitarsi con HMAC utilizzando gli strumenti online:

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

Crittografia applicata alle e-mail (dati in transito)

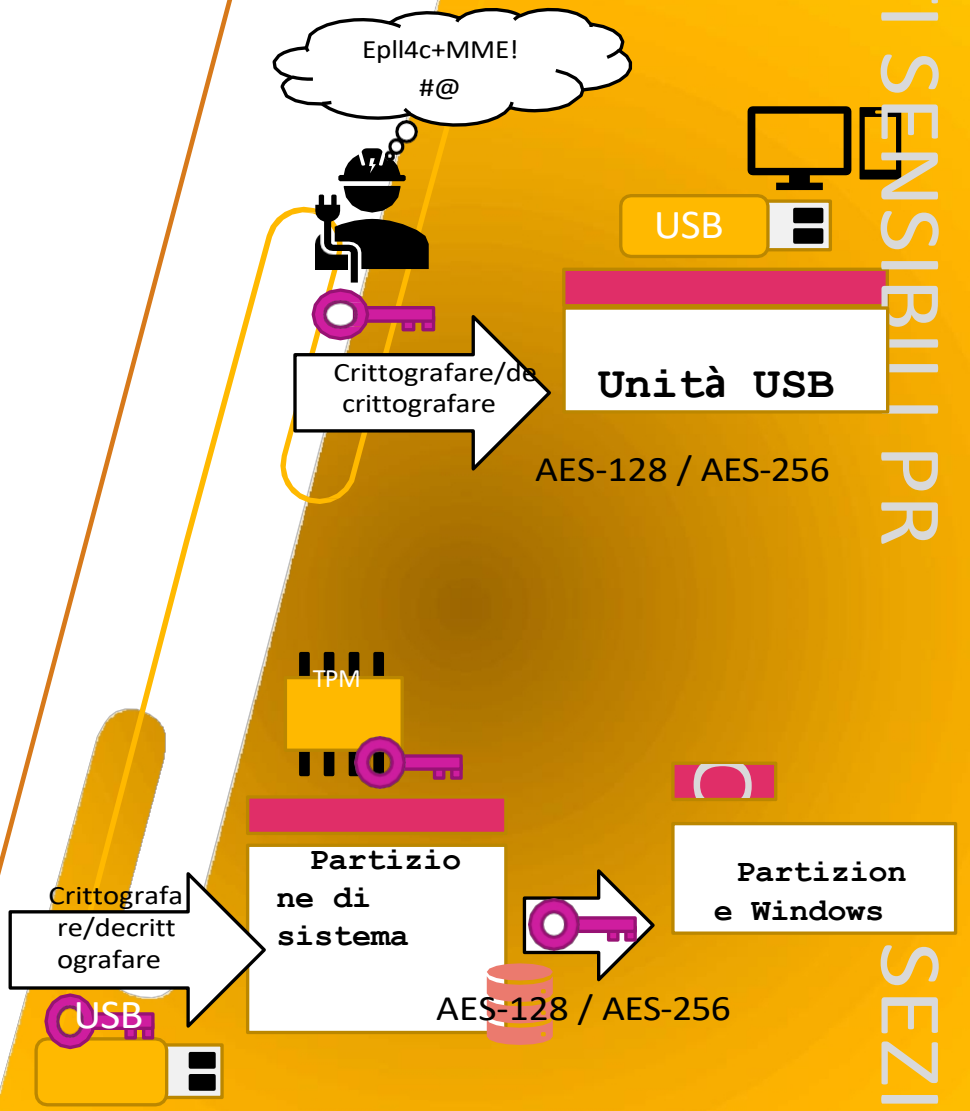
- I professionisti del settore energetico possono attivare misure di sicurezza che consentono loro di proteggere le proprie e-mail, utilizzando, ad esempio:
  - **Secure/Multipurpose Internet Mail Extension (S/MIME)**
  - **Pretty Good Privacy (PGP)** secondo lo standard OpenPGP
- Entrambi i protocolli offrono garanzie **di riservatezza, integrità e autenticazione**
  - S/MIME gestisce i certificati digitali x.509 per l'autenticazione
  - PGP offre un proprio formato di certificato
- S/MIME e PGP sono compatibili con tutte le piattaforme e possono essere configurati in diversi editor di posta elettronica, come ad esempio
  Thunderbird
  - Si noti inoltre che PGP può implicare l'installazione di GnuPGP, che è un'implementazione dello standard OpenPGP
  - PGP può essere applicato per proteggere i dati inattivi, come i file

Come nei punti precedenti, gli studenti possono anche esercitarsi con il potenziale di PGP per proteggere i file localmente, utilizzando, ad esempio:

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

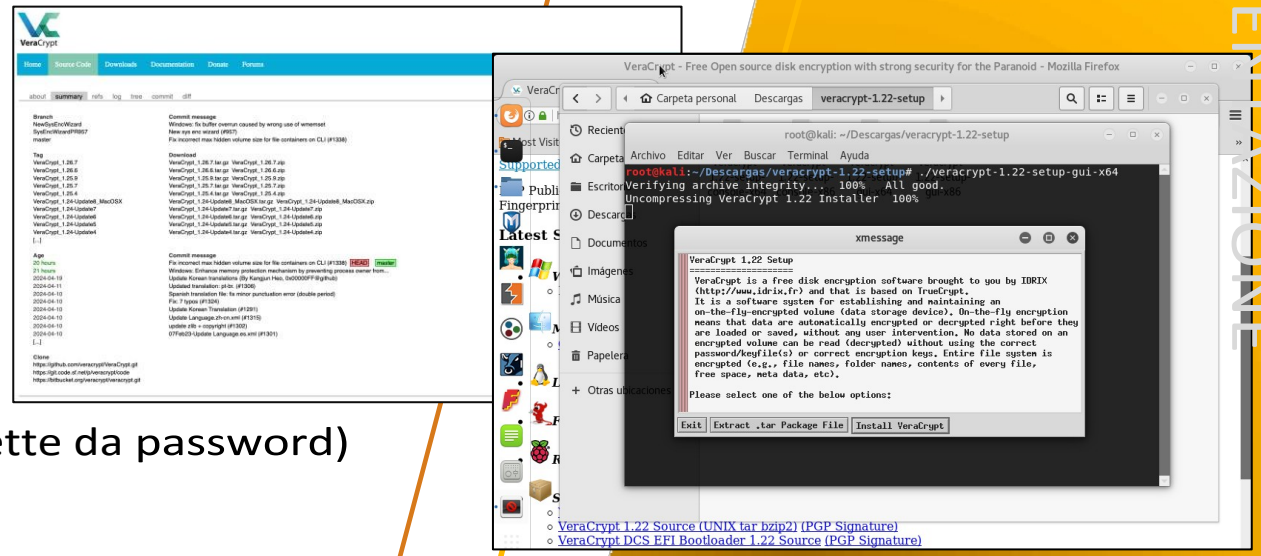
Crittografia applicata ai dati inattivi

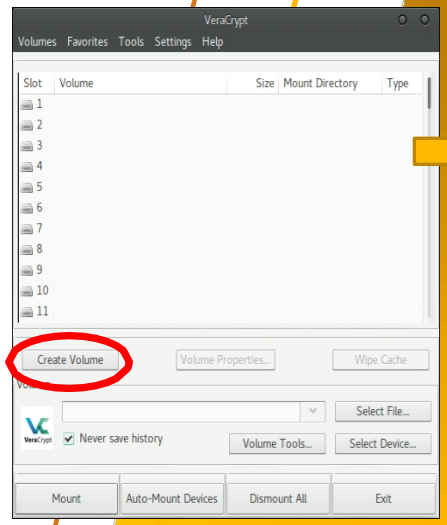
- I sistemi operativi normalmente supportano applicazioni crittografiche per crittografare/decrittografare dati sensibili su disco rigido e unità rimovibili
  - Windows 10/11: **Bitlocker** (di default)
  - Linux e Windows: **Veracrypt, TrueCrypt** (open source)
- La protezione di queste unità segue solitamente procedure comuni:
  - **La protezione dei driver USB** si basa su una password per ottenere la "chiave" necessaria per far funzionare l'unità tramite un algoritmo di crittografia/decrittografia
  - **La protezione del disco rigido** può richiedere un Trusted Platform Module (TPM) 1.2 (o un driver USB) per ottenere la chiave responsabile della protezione della partizione del disco

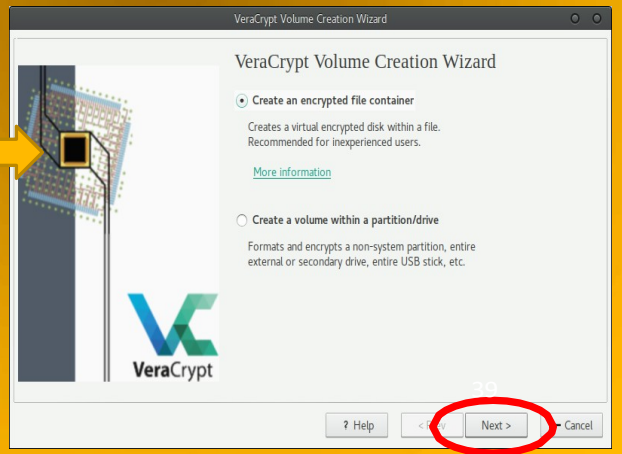


Veracrypt: un esempio della sua utilità

- **Open source:**
  - https://www.veracrypt.fr/code/VeraCrypt/
- **Funzioni principali**:
  - Crittografia di unità (chiavetta USB o disco rigido)
  - Creazione di volumi di dati e unità nascoste (protette da password)
- Esempio di **creazione di volumi di dati sicuri**:
  1. Selezionare "contenitore *file crittografato*" per archiviare i dati all'interno di un volume virtuale crittografato in un unico file





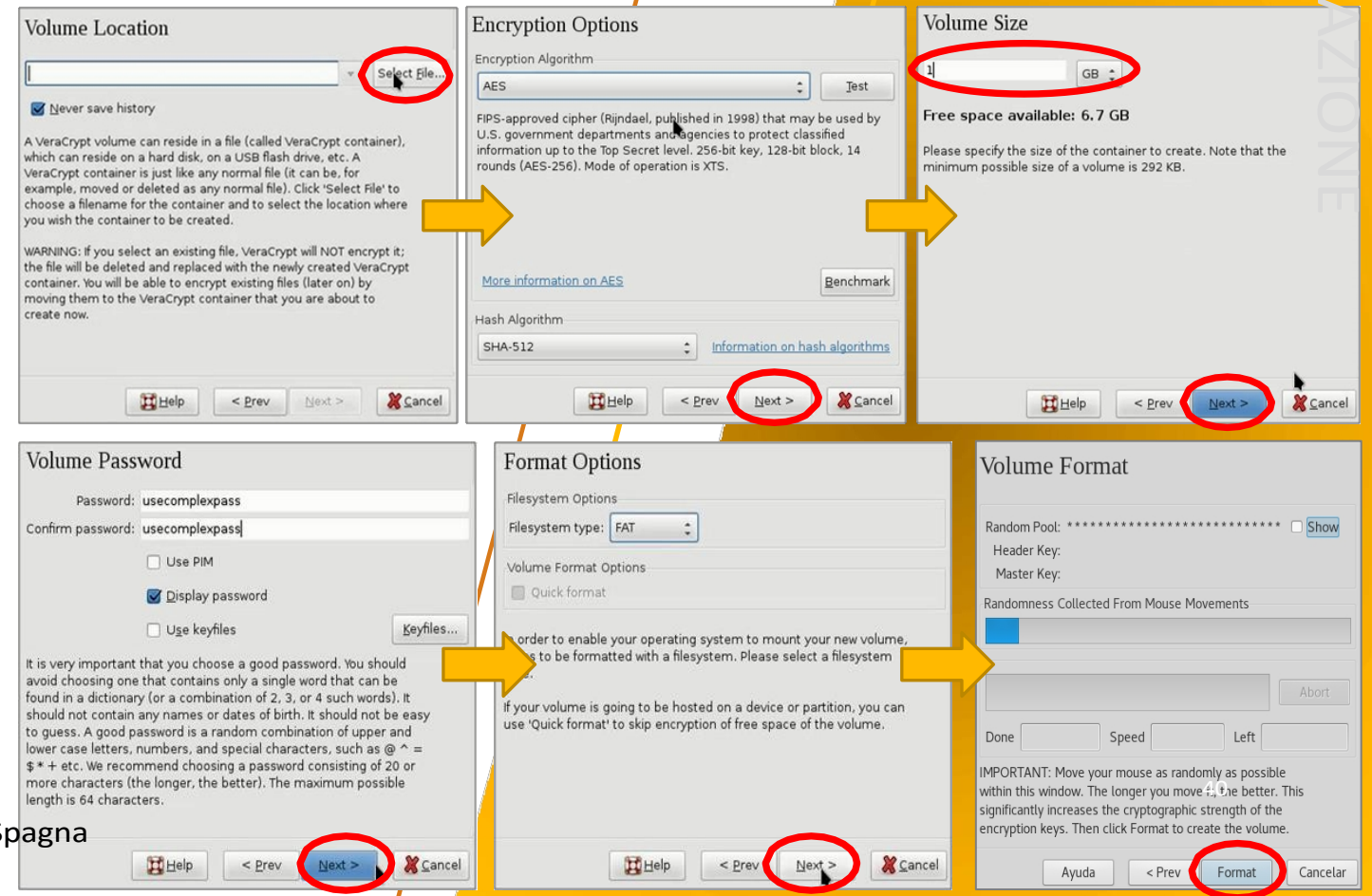


Fonte: Veracrypt, 2024

URL: https://www.veracrypt.fr/code/VeraCrypt/

Veracrypt: un esempio della sua utilità

2. Scegliere la posizione del file, gli algoritmi di crittografia (ad es. AES) e hash (ad es. SHA-512), nonché la dimensione del file
3. Scegliere la password e il formato del volume contenuto nel file (ad es. FAT, ExFAT, NTFS...)
   - A questo punto, è necessario muovere il mouse in direzioni casuali per creare l'"entropia", necessaria per la generazione della chiave.

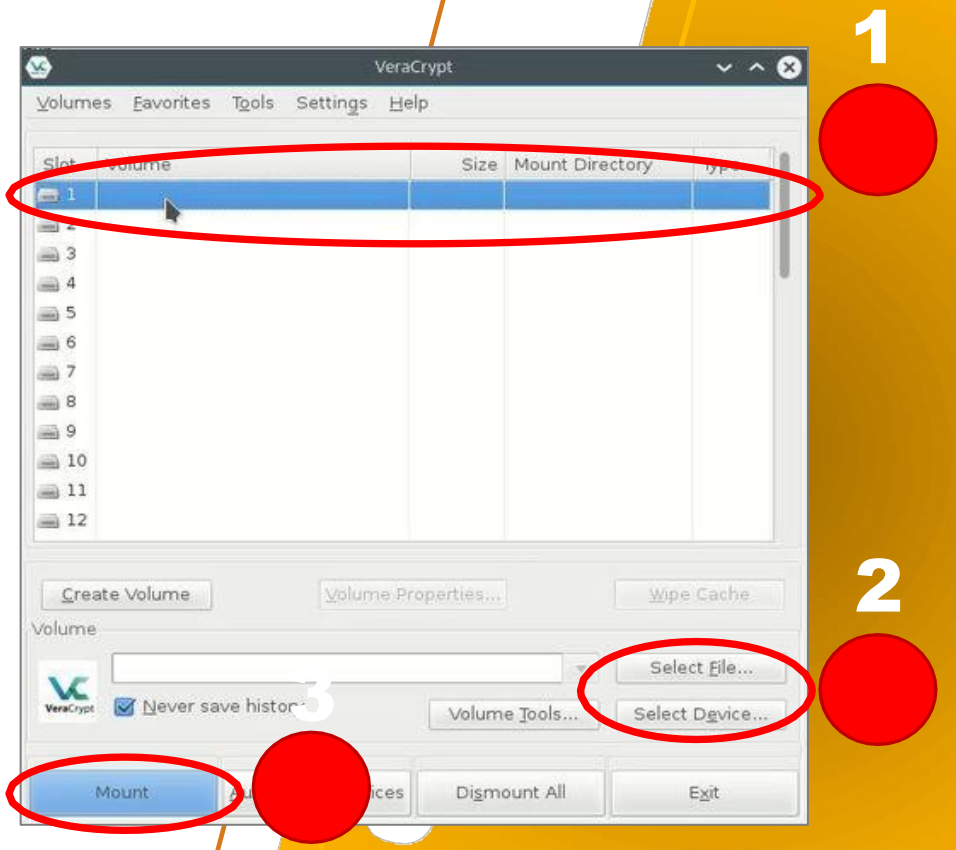


Fonte: Veracrypt, 2024

URL: https://www.veracrypt.fr/code/VeraCrypt/

Veracrypt: un esempio della sua utilità

- Esempio di utilizzo
  1. Selezionare uno slot
  2. Scegliere il file che contiene il volume di dati crittografato
  3. Monta il file, indicando la password



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

Potenziare le competenze con OpenSSL

- **OpenSSL** è un'alternativa per la protezione dei file
  - Strumento open source che include una libreria crittografica generica e meccanismi da applicare nella riga di comando
  - I comandi OpenSSL sono generici e validi per qualsiasi piattaforma
- Comandi:
  - Generici:
    - **$ openssl help**
    - **$ openssl list-standard-commands**
    - **$ openssl list-message-digest-commands**
    - **$ openssl list-cipher-commands**
  - Crittografia simmetrica:
    - **$ openssl enc –cipher-operation mode -in file.txt**
    - **$ openssl enc –cipher-operation mode –iv IV –K key –in file.txt**
  - Decrittografia simmetrica:
    - **$ 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
```

Potenziare le competenze con OpenSSL

- Generare chiavi:
  - **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**
  - **…**
- Crittografia asimmetrica:
  - **$ openssl rsautl -encrypt -in input.txt -pubin -inkey public.pubkey -out ciphertext.txt**
- Decrittografia asimmetrica:
  - **openssl rsautl -decrypt -in ciphertext.txt -out plaintext.txt -inkey private.key**
- Firma digitale:
  - **$ openssl rsautl -sign -in plaintext.txt -out signature.txt -inkey private.key**
- Verifica della firma digitale:
  - **openssl rsautl -verify -in firma.txt -out verifica.txt -pubin -inkey pubblica.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 nomefile] [-hmac chiave] [-non-fips-allow] [-fips-fingerprint] [file...]

CSP001_C_E – ARGOMENTO 6: Davide Ferraris, Università di Malaga, Spagna

Considerazioni finali

- Abbiamo esaminato i vantaggi che la crittografia può apportare alla protezione dei dati nei sistemi di controllo energetico, quali:
  - Sottostazioni, reti di controllo, reti aziendali
- Questo livello di protezione può essere applicato a:
  - **Dati in transito**: tra apparecchiature industriali, dispositivi IT e utenti, ecc.
  - **Dati inattivi**: server, archivi, tablet, HMI, ecc.
- Per la protezione, possiamo applicare:
  - Crittografia simmetrica e asimmetrica, o una combinazione delle due, per garantire la riservatezza
  - Certificati digitali per la fiducia
  - Funzioni hash per l'integrità
  - Funzioni MAC per l'autenticazione e l'integrità

... incredibile, vero? ☺

Riferimenti e fonti

1. Netresec, "Captures files from 4SICS Geek Lounge", 2024 URL: https://www.netresec.com/?page=PCAP4SICS
2. CS3Sthtlm, 2014-2020, consultato nel 2024. URL: https://cs3sthlm.se
3. CloudShark "telnet-client-server.pcapng", consultato nel 2024 URL: https://www.cloudshark.org/captures/818ceaef07b8?filter=telnet
4. Buchanan, William J., Chiavi di crittografia leggibili dall'uomo. 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., Cifratura DES. Asecuritysite.com, 2024. URL: https://asecuritysite.com/symmetric/des
8. Buchanan, William J., 3DES Cipher. Asecuritysite.com, 2024. URL: https://asecuritysite.com/symmetric/threedes

Riferimenti e fonti

9. Buchanan, William J., Camellia cipher. Asecuritysite.com. 2024. URL: https://asecuritysite.com/symmetric/camellia
10. Buchanan, William J., Modalità a chiave simmetrica: GCM, ChaCha20, CBC, CFB8, CFB, OFB, GCM, CTR e XTS, Asecuritysite.com, 2024. URL:https://asecuritysite.com/symmetric/sym
11. CryptoTool.net, Generatore di chiavi RSA, 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 per la correzione di bozze: URL: https://www.deepl.com/translator
15. Alcune immagini sono state prese da Vecteezy, URL: https://www.vecteezy.com/ - grazie!

Connettiti con CyberSecPro: come registrarsi e altre informazioni pratiche

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

Co-funded by the European Union

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

Grazie

Per qualsiasi domanda, non esitate a esitare a contattare:

- Davide Ferraris
  Professore supplente
  Università di Malaga
  ferraris@uma.es