UNIONE EUROPEA
Fondo Sociale Europeo
Fondo Europeo di Sviluppo Regionale

Ministero dell'Istruzione, dell'Università e della Ricerca

PON RICERCA E INNOVAZIONE 2014 - 2020

Avviso 1735 del 13.07.2017 MIUR

Progetti di Ricerca Industriale e Sviluppo Sperimentale nelle 12 Aree di Specializzazione individuate dal PNR 2015-2020

Progettazione di soluzioni di cyber security e privacy in ambito Smart Grid

Rapporto Tecnico di Ricerca Industriale D5.6

Avviso	Avviso 1735 del 13.07.2017 MIUR
Codice progetto	ARS01_01259
Nome del progetto	Community Energy Storage Gestione Aggregata di Sistemi di Accumulo dell'Energia in Power Cloud
Acronimo	ComESto
Documento	D5.6
Tipologia	Rapporto Tecnico
Data di Rilascio	26/04/2021
Obiettivo Realizzativo	OR5
Attività Realizzativa	A5.6
Soggetti Beneficiari Proponenti	UNICAL, TIM, SPINTEL
Elaborato (Nome, Cognome – Soggetto Beneficiario)	Floriano De Rango, Nicola Sorrentino – UNICAL Caterina Cippone, Giacomo Morello – TIM Francesco Noto – SPINTEL
Verificato (Nome, Cognome – Soggetto Beneficiario)	Floriano De Rango, Nicola Sorrentino - UNICAL
Approvato (Nome, Cognome – Soggetto Beneficiario)	Membri del PEB

Indice

Indice delle figure

Indice delle tabelle

Abbreviazioni ed acronimi

Abbreviazione/Acronimo	Testo Esteso
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AgID	Agenzia per l'Italia Digitale
AGM	Absorbent Glass Mat
AMI	Advanced Metering Infrastructure
ANIE	Federazione Nazionale Imprese Elettrotecniche
APAC	Asia Pacific
API	Interfaccia di Programmazione
ARERA	Autorità di Regolazione per Energia Reti e Ambiente
ASE	Altri Sistemi Esistenti
BEV	Battery Electric Vehicle
BYOD	Bring Your Own Device
CAM	Content Addressable Memory
CBC	Cipher-Block-Chaining
CCMP	CTR mode con CBC-MAC Protocol
CoAP	Constrained Application Protocol
CRC	Cyclic Redundancy Check
DDoS	Distributed Denial of Service
DGM	Distribution Grid Management
DMS	Distributed Management System
DNS	Domain Name System
DoS	Denial of Service
DTLS	DNS over TLS
ECIES	ECIES Hybrid Encryption Scheme
ECDSA	Elliptic Curve Digital Signature Algorithm
eIDAS	electronic IDentification Authentication and Signature
ENISA	European Union Agency for Cybersecurity
EOL	End Of Life
FAC	Fake Access Point
GCM	Galois Counter Mode

GDPR	General Data Protection Regulation
GMAC	Galois Message Authentication Code
GOOSE	Generic Object Oriented Substation Event
HMAC	keyed-Hash Message Authentication Code
HSTS	HTTP StrictTransport Security
HTTPS	HTTP Secure
ICS	Industrial Control System
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Devices
IIoT	Industrial Internet of Things
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
ISO	International Organization for Standardization
ISP	Internet Service Provider
M2M	Machine-to-Machine
MAC	Message Authentication Code
MD5	Message Digest 5
MDM	Mobile Device Management
MiTM	Man-in-the-Middle
MMS	Manufacturing Message Specification
MQTT	Message Queue Telemetry Transport
NESCOR	National Electric Sector Cybersecurity Organization Resource
NIS	Network and Information Security
NIST	National Institute of Standards and Technology
NSC	Nucleo sicurezza cibernetica
NSM	Network Security Management
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
Piercer	Persistent Information ExposuRe by the CorEnetwoRk
PIM	Protocol Independent Multicast
PLC	Programmable Logic Controller
PMK	Pairwise Master Key
PRNG	Pseudo-Random Number Generator
PSK	Pre Shared Key
PTK	Pairwise Transient Key
PTW	Pyshkin, Tews and Weinmann
RC4	Rivest Cipher 4
SCADA	Supervisory Control and Data Acquisition
SCL	SubstationConfiguration Language
SHA	Secure Hash Algorithm

SNMP	Simple Network Management Protocol
SSDP	Simple Services Discovery Protocol
SSID	Service Set IDentifier
SSL	Secure Sockets Layer
SV	Sampled Value
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
ToRPEDO	TRacking via Paging mEssage DistributiOn
VPN	Virtual Private Network
WAMPAC	Wide Area Monitoring, Protection and Control
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2

EXECUTIVE SUMMARY

La necessità di una gestione ottimizzata dei dispositivi connessi alla rete elettrica ha portato all'introduzione di funzionalità ICT (Information and Communication Technology) nelle reti di telecontrollo e all'interno dei dispositivi di rete stessi. La natura di queste reti, le cosiddette Smart Grid, è molto eterogenea: vengono impiegati dispositivi di vendor diversi e sono necessarie interconnessioni tra le reti di operatori differenti. Dal punto di vista informatico, nelle comunicazioni occorre impiegare un linguaggio comune definito da uno standard condiviso, mentre un altro aspetto molto importante è la sicurezza delle comunicazioni stesse.
Come ogni rete di comunicazione informatica, le Smart Grid non sono esenti dalle minacce di natura ICT, che si concretizzano in attacchi di vario genere, con molteplici meccanismi di azione e obiettivi diversi. Nel contesto specifico, la necessità di misure di difesa è più che mai critica data la natura dei flussi informativi in esame: occorrono soluzioni che possano fornire un livello di protezione adeguato a questi sistemi.

Negli ultimi anni si sta assistendo ad una forte evoluzione delle reti elettriche di distribuzione, caratterizzata dal progressivo abbandono del modello di rete passiva grazie all'introduzione di nuove funzionalità ICT, in cui tutti i nodi della rete possono contribuire al bilancio energetico del sistema elettrico globale. Queste funzionalità sono necessarie per conseguire una gestione della rete "intelligente", che integri e ottimizzi il funzionamento di tutti gli elementi ed essa connessi, che siano generatori, consumatori o utenti finali in grado di effettuare entrambe le operazioni. La gestione dei flussi multidirezionali di energia che transitano sulla rete (atta a garantire l'equilibrio tra la domanda e l'offerta di energia) e le operazioni di telecontrollo ICT, consentono di integrare la gestione distribuita e cambiare il ruolo dell'utente finale, non più solo consumatore ma anche produttore che immette energia nella rete (utente attivo). L'architettura di queste reti, le Smart Grid, risulta quindi essere molto complessa. Basti pensare all'eterogeneità dei dispositivi e delle tecnologie impiegate dai diversi operatori, che potrebbero implementare protocolli di comunicazione differenti, anche non standard. Connettendo le reti di più operatori si rafforza la necessità di garantire l'interoperabilità tra i vari sistemi coinvolti, in modo che porzioni di rete non rimangano isolate.
Lo IEC 61850 è uno standard [2] per la progettazione dei sistemi di automazioni per le sottostazioni elettriche che, per far fronte alle necessità legate alla sicurezza è coadiuvato da un ulteriore standard, lo IEC 62351 [3], che si pone l'obiettivo di rendere sicuro sotto vari aspetti lo standard per l'automazione delle sottostazioni.

Sicurezza e attacchi informatici
I tre obiettivi principali della sicurezza informatica sono generalmente disponibilità, integrità e riservatezza.
Oltre a questi, lo standard IEC 62351 considera anche il non ripudio, l'autenticazione e l'autorizzazione tra i suoi obiettivi [4]. Di seguito viene fornita una definizione di ciascuno di questi termini e dei relativi attacchi informatici.

Autenticazione
L'autenticazione può essere suddivisa in due aspetti. L'autenticazione dell'entità [5] serve a verificare che le entità siano veramente chi o cosa dichiarano di essere. Un modo comune per ottenere ciò è tramite l'uso di chiavi, password, certificati o dati biometrici univoci per ciascuna entità. Questi assicurano l'autenticazione poiché solo l'entità legittima dovrebbe conoscere ed essere in grado di presentare le credenziali corrette. L'autenticazione dei messaggi [5], invece, serve a verificare la fonte di alcune informazioni. Per autenticare i dati, vengono utilizzati i codici MAC (Message Authentication Codes) e le firme digitali, come Rivest-Shamir-Adleman (RSA). Queste tecniche consentono di produrre un piccolo valore per il messaggio che può essere calcolato solo dal legittimo detentore della chiave. L'autenticazione non può essere ottenuta se l'autenticatore non è affidabile, come nel caso dei sensori di impronte digitali che vengono ingannati da impronte digitali false. Inoltre, fa molto affidamento sulla sicurezza delle credenziali, poiché un utente malintenzionato che le ottiene può quindi impersonare un'altra entità.

Autorizzazione
Definizione: Limitare l'accesso alle sole entità legittime [5].

L'autorizzazione consente solo azioni da parte di entità che dispongono delle autorizzazioni richieste. In altre parole, il sistema deve consentire che le azioni vengano eseguite da entità che hanno il ruolo oi diritti necessari per eseguirle, nonché impedire azioni da parte di entità che non dispongono di tali diritti. L'autorizzazione è un requisito fondamentale per gli altri obiettivi di sicurezza, poiché distingue le entità reali (autorizzate) dalle altre. Tieni presente che l'autenticazione è un prerequisito per l'autorizzazione.

Riservatezza
Definizione: Garantire che le informazioni non siano accessibili da entità non autorizzate [4].
Riservatezza significa nascondere le informazioni ad eccezione di coloro che sono destinati ad accedervi. La protezione dei dati riservati può essere ottenuta utilizzando la crittografia e distribuendo la chiave di de-crittografia solo a entità autorizzate. Gli algoritmi di crittografia sono progettati in modo che le informazioni crittografate appaiano come bit casuali inutili a chiunque non abbia la chiave di decrittazione, rendendo allo stesso tempo molto semplice il recupero delle informazioni per i possessori di chiavi. Un algoritmo di crittografia comunemente utilizzato è Advanced Encryption Standard (AES). La riservatezza è essenziale per proteggere non solo i dati, ma anche le credenziali che devono essere tenute segrete per raggiungere gli altri obiettivi di sicurezza. Gli attacchi alla riservatezza includono lo sniffing di pacchetti che viaggiano su una rete, il furto di dati sensibili da un file o un database su un dispositivo finale, l'indovinare chiavi di crittografia e la raccolta di informazioni su un sistema tramite scansione o intercettazioni.

Integrità
Definizione: Garantire che le informazioni non possano essere modificate da entità non autorizzate [4].
Integrità significa che ci si può fidare di un sistema o di un'informazione. Mentre la riservatezza mira a impedire la lettura non autorizzata di informazioni, l'integrità mira a impedire la scrittura non autorizzata di informazioni. Ciò si ottiene spesso facendo affidamento sulle funzioni hash. Una funzione hash prende dati di qualsiasi lunghezza e li converte in un unico, piccolo valore "hash" di una lunghezza specifica. Ricalcolare l'hash di un input specifico dovrebbe sempre dare come risultato lo stesso hash: se non lo fa, questo indica che l'input è stato alterato e non può essere considerato attendibile. Le funzioni hash rilevanti per questa tesi sono Message Digest 5 (MD5) e Secure Hash Algorithm (SHA)-1. L'MD5 è ora considerato insicuro. SHA-1 è leggermente migliore dell'MD5, ma il National Institute of Standards and Technology (NIST) consiglia di sostituirlo con il superiore SHA-256 [6]. Una firma MAC o digitale (come RSA) unisce integrità e autenticazione, poiché sono essenzialmente hash che richiedono una chiave segreta per il calcolo. Gli attacchi all'integrità comportano la falsificazione dei dati in qualche modo, come la modifica del contenuto di file o database. Nelle reti, gli attacchi all'integrità includono il reinvio di pacchetti inviati in precedenza (replay), l'iniezione di pacchetti contraffatti nella rete o la modifica di pacchetti esistenti dopo aver impostato un attacco Man-in-the-Middle (MitM).

Disponibilità
Definizione: Garantire un accesso tempestivo e autorizzato alle informazioni.
Disponibilità significa garantire l'accesso al sistema ogniqualvolta sia richiesto dalle parti autorizzate [4]. Un sistema con alta disponibilità è pronto per l'accesso ogni volta che è necessario da parte di un utente legittimo. Il sistema deve inoltre completare i propri compiti in un lasso di tempo accettabile, come dettato dai requisiti di prestazione. Gli attacchi alla disponibilità includono attacchi Denial-of-Service (DoS) e attacchi per degradazione del servizio. Il primo mira a rendere un sistema inutilizzabile da parte di soggetti legittimi. Quest'ultimo riduce solo parzialmente la disponibilità, ad esempio rallentando notevolmente un sistema. Perché la degradation-of-service è spesso considerata un sottotipo o un sinonimo di denial-of-service, spesso si usa l'acronimo DoS per riferirci a entrambi.

Non ripudio
Definizione: Impedire alle parti di negare le loro azioni passate o di rivendicare azioni che non si sono effettivamente verificate [4].
Il non ripudio garantisce che le parti non possano nascondere o falsificare le registrazioni della loro attività. Questa proprietà è importante per il controllo dopo un evento di sicurezza per garantire che gli archivi di log siano affidabili. Le firme digitali

possono fornire un grado di non ripudio, nonché integrità e autenticazione per messaggi specifici, poiché le firme digitali possono essere calcolate solo dal legittimo titolare di una chiave privata che non è condivisa con altre entità.

Le smartgrid

Una rete elettrica è l'infrastruttura utilizzata per fornire elettricità ai clienti. Ha tre principali sottosistemi: generazione, trasmissione e distribuzione [7]. I generatori sono la fonte di elettricità. Il sistema di trasmissione aumenta la tensione di potenza proveniente dai generatori, in quanto ciò riduce la perdita di energia in transito e trasporta l'elettricità al sistema di distribuzione. Lì, la tensione viene ridotta a livelli utilizzabili per i clienti prima della consegna [7]. Nel 2009, il Dipartimento dell'Energia degli Stati Uniti (US DOE) ha dichiarato che le reti elettriche in uso all'epoca si avvicinavano alla fine del ciclo di vita (EOL) [8]. Le utility sono quindi motivate a passare alle tecnologie delle reti intelligenti. La rete intelligente è un aggiornamento delle reti elettriche esistenti che mira a fornire efficienza e affidabilità integrando le tecnologie ICT [9]. Fornisce molti vantaggi, come il rilevamento automatico e la risposta agli eventi, l'integrazione di dispositivi di nuova generazione e di archiviazione per consentire la generazione distribuita e il miglioramento della qualità dell'energia [8].
Il monitoraggio viene effettuato utilizzando un sistema SCADA (Supervisory Control and Data Acquisition) che consente agli operatori di controllare a distanza la rete da un centro di controllo [10]. Le comunicazioni nella smartgrid avvengono tramite una varietà di protocolli, vale a dire Distributed Network Protocol (DNP3), IEC 60870-5 e IEC 61850 [10]. Nell'ambito dello sviluppo della rete intelligente, le sottostazioni di nuova generazione sostituiscono i vecchi dispositivi elettromeccanici con Intelligent Electronic Device (IED) in grado di svolgere funzioni di protezione, monitoraggio e controllo [11]. Questa modifica ha portato alla pubblicazione della norma IEC 61850 che definisce come questi IED dovrebbero comunicare in modo interoperabile anche quando sono progettati da produttori diversi [11].

Minacce alle smartgrid

Le reti elettriche sono potenzialmente vulnerabili a minacce quali disastri naturali, errori da parte dei dipendenti delle utenze e sabotaggio deliberato delle apparecchiature elettriche. Oltre a queste minacce già esistenti, l'integrazione delle tecnologie IT nelle reti elettriche porta necessariamente con sé il potenziale di attacchi informatici contro questi nuovi dispositivi, soprattutto se la rete di comunicazione della rete è esposta a Internet [9]. Tali attacchi possono essere eseguiti anche da remoto. I titoli delle notizie evidenziano già diversi esempi di attacchi informatici contro i sistemi di controllo industriale (ICS). Stuxnet è un malware sofisticato che prende di mira i controllori logici programmabili (PLC) all'interno di ICS. Li costringe ad aumentare e diminuire la velocità dei loro motori per raggiungere valori estremi al di fuori della loro gamma normale, causando danni alle apparecchiature [12]. Stuxnet nasconde anche la sua attività per eludere il rilevamento. Il malware ha infettato circa 100.000 dispositivi in tutto il mondo (la maggior parte dei quali in Iran) tramite unità rimovibili, secondo un rapporto del 2011 di Symantec [12]. Più di recente, la rete elettrica dell'Ucraina è stata presa di mira in due diverse occasioni nel 2015 [13] e nel 2016 [14]. Il primo incidente è stato un attacco informatico coordinato che ha causato un'interruzione di corrente per oltre 200.000 clienti. Gli hacker hanno assunto da remoto il controllo del software degli operatori (possibilmente utilizzando il trojan BlackEnergy), hanno aperto i breaker tramite comandi non autorizzati, quindi hanno eseguito il malware KillDisk per cancellare i sistemi e bloccare i tentativi di ripristino [13]. Al contrario, il secondo attacco è stato effettuato utilizzando il malware CrashOverride. Questo malware ha molte funzionalità, tra cui la scansione delle reti per raccogliere informazioni su di esse, lo spoofing dei comandi ICS, l'esecuzione di attacchi DoS per bloccare le comunicazioni o spegnere i dispositivi e cancellare completamente i dispositivi [14].
CrashOverride è tra i primi malware che possono prendere di mira il protocollo IEC 61850, tra gli altri protocolli ICS [10]. Questi esempi sottolineano l'importanza di proteggere la rete intelligente e la sottostazione IEC 61850, dato che esistono già minacce sofisticate progettate specificamente per questi ambienti [15].

INTRODUZIONE E OBIETTIVO DI PROGETTO

Negli ultimi anni si sta assistendo ad una forte evoluzione delle reti elettriche di distribuzione, caratterizzata dal progressivo abbandono del modello di rete passiva grazie all'introduzione di nuove funzionalità ICT, in cui tutti i nodi della rete possono contribuire al bilancio energetico del sistema elettrico globale. Queste funzionalità sono necessarie per conseguire una gestione della rete "intelligente", che integri e ottimizzi il funzionamento di tutti gli elementi ed essa connessi, che siano generatori, consumatori o utenti finali in grado di effettuare entrambe le operazioni.

Si sta afferarmando una nuova figura nota con il nome di prosumer il quale è un utente finale capace non solo di consumare ma anche di produrre energia per immetterla nel sistema.
In tale constesto risulta fondamentale lo studio degli standard e dei protocolli di comunicazione che permettorno lo scambio di informazione tra tutti gli attori presenti nel sistema complessivo. Conseguentemente, risulta di forndamentale importanza analizzare e progettare soluzioni ad hoc che possano garantire un'adeguata sicurezza all'interno del sistema per preservare queste informazioni che vengono scambiate e che potrebbero essere oggetto di attacchi fraudolenti.

L'obiettivo principale delle attività di ricerca condotte nell'OR5.6 riguarda lo studio e l'analisi dei principali meccanismi di sicurezza utilizzabili all'interno del sistema energetico a salvaguardia della grossa mole di dati che continuamente viene scambiata dai dispositivi che ne fanno parte.
Dopo un breve excursus sulle principali definizioni attinenti al contesto della sicurezza informatica, in questo documento viene fornita un'attenta analisi dei principali attacchi che possono essere perpetrati a danno delle smartgrid, dando opportuna rilevanza anche a tutte le vulnerabilità tipiche dei sistemi IoT adottate nello specifico contesto.

Si sono analizzate inizialmente le normative di riferimento in ambito sicurezza ed il GDPR (General Data Protection Regulation) – relativo alla protezione delle persone fisiche con riguardo al trattamento e alla libera circolazione dei dati personali, nell'ottica di dare una panoramica generale su tutto ciò che regolamenta la sicurezza in ambito informatico.
Ci si è poi concentrati sul fornire delle informazioni di base per quel che riguarda la sicurezza in ambito smartgrid e quindi su quei sistemi di produzione intelligente dell'energia che prevedono una intelligenza nella gestione del sistema elettrico distribuito secondo i nuovi paradigmi in tale ambito. Infatti, l'inovazione più importante delle smart grid è la gestione bidirezionale dell'energia in quanto ora i nuovi utenti del sistema sono sia consumatori che produttori potendo immettere nel sistema tutta l'energia generata in eccesso che non viene da loro consumata e che può essere fornita al sistema redistribuendo così il flusso energetico a seconda dei reali bisogni in tempo reale. Tali sistemi, dovendo garantire questo flusso di informazione bidirezionale, sono soggetti ad attacchi da parte dei cosidetti cyber criminali che possono sfruttare qualsiasi falla presente nei protocolli di comunicazione per inserirsi nel sistema e provacare danni all'intero processo di scambio. A tal proposito il rapporto del NISTIR 7628 ha identificato alcuni esempi di potenziali attacchi e rischi associati alle smartgrid. Tali sistemi devono quindi sviluppare e progettare veri e propri meccanismi volti a garantire sicurezza a livello dispositivo con gestione di certificazioni e crittografia e mettere così in sicurezza l'intera rete e l'intero sistema energetico.
I nuovi sistemi energetici sono basati sull'utilizzo di una serie di dispositivi/sensori per cui risulta importante definire e studiare le vulnerabilità in ambito IoT al fine di trovare le giuste contromisure per la mitigazione dei potenziali rischi dovuti alla fitta rete di connessioni e di dispositivi che fanno parte dell'intero sistema energetico.

Una attenta analisi è stata condotta anche per ciò che riguarda la sicurezza in ambito tecnologico legata alle possibili tecnologie di comunicazione da adoperare per la comunicazione nelle smartgrid, affrontando la questione sicurezza in ambito 4G e 5G evidenziando sempre attacchi e contromisure. Un cenno è stato anche dato alla ricerca in ambito energetico per quello che riguarda l'utilizzo di meccanismi innovativi come quelli rappresentati dalle blockchain.
La sicurezza nello standard IEC 61850 è stata anche oggetto di studio nell'ottica di evidenziare vantaggi e debolezze nell'utilizzo di tale standard per le comunicazioni delle sottostazioni energetiche.

Infine, ampio spazio è stato dato allo studio della sicurezza per quanto riguarda i protocolli machine-to-machine. In tale ambito si sono studiate e analizzate le principali tipologie di minacce con particolare attenzione ai protocolli CoAP e MQTT, individuati come potenziali protocolli adatti ad essere utilizzati nel contesto di riferimento, dando evidenza delle problematiche legate alle proprietà tipiche della sicurezza: confidenzialità, integrità, disponibilità e autenticazione.
A seguito di questa ampia analisi, ci si è focalizzata sulle tecnologie IoT e le comunicazioni M2M adottate come soluzione di comunicazione tra dispositivi di campo e nano-grid e tra nano-grid e sistema cloud.
Infine, si è proceduto alla progettazione di alcune feature di sicurezza nella comunicazione M2M per le smartgrid evidenziando attraverso un prototipo realizzativo le specifiche soluzioni di sicurezza da implementare in ambito energetico per garantire le proprietà di confidenzilità, integrità, disponibilità e autenticazione.
Lo studio prototipale ha individuato in MQTT e nella soluzione MQTT security con l'uso del TLS come meccanismo di comunicazione da utilizzare all'interno della piattaforma ComESTO per garantire all'intero sistema quei requisiti di sicurezza richiesti per lo svolgimento sicuro delle operazioni di distribuzione, consumo e produzione dell'energia da fonti rinnovabili.

SICUREZZA INFORMATICA IN EUROPA E IN ITALIA: TUTTE LE NORME DI RIFERIMENTO/GDPR

Il 2018 è stato l'anno della sicurezza informatica: da maggio le norme dell'Unione europea sono diventate operative in tutti gli stati membri. Una cornice di riferimento sulla cybersecurity costituita da due regolamenti e due direttive, entro cui si collocano tutte le leggi e le applicazioni nazionali [16]. In particolare:

- La direttiva Nis: il primo passo della strategia europea per la sicurezza informatica
- GDPR, la sicurezza informatica dei dati personali
- La direttiva 680, la sicurezza informatica dei dati trattati dalle autorità
- EIdas, la sicurezza informatica applicata alla firma e alle transazioni elettroniche
- La via italiana alla sicurezza informatica e il decreto Gentiloni

1.1 La direttiva Nis: il primo passo della strategia europea per la sicurezza

Possiamo considerare la direttiva Nis, Network and Information Security, il primo passo della strategia europea per la cybersecurity. Approvata dal Parlamento Ue il 6 Luglio 2016, la direttiva ha l'obiettivo di rafforzare la sicurezza e la resilienza informatica all'interno del Vecchio Continente creando, di fatto, un'unica linea strategica **c**ontro il rischio di incidenti ai danni delle reti informatiche e dei sistemi informativi**.** In Italia è stata recepita dal Decreto Legislativo 18 maggio 2018 n. 65. [17].

Un'esigenza che parte da una considerazione di base: le reti, i sistemi e i servizi informativi svolgono oggi un ruolo d'importanza vitale nella società. Senza di loro il mercato interno non potrebbe funzionare, per questo è essenziale che siano affidabili e sicure per le attività economiche e sociali.

Il Nis si applica sostanzialmente a due categorie di soggetti: gli operatori di servizi essenziali, cioè soggetti considerati necessari al mantenimento di attività sociali e/o economiche considerate fondamentali (imprese che operano nel settore trasporti, energia, sanità, bancario, fornitura e distribuzione acque e infrastrutture digitali); fornitori di servizi digitali, persone giuridiche che forniscono servizi di e-commerce, cloud computing e motori di ricerca con sede sociale (o rappresentante designato) sul territorio nazionale. Non sono soggetti alla Direttiva NIS i Fornitori di Servizi Digitali con meno di cinquanta dipendenti o con fatturato inferiore ai 10 milioni di euro l'anno.

Entrambe le categorie interessate dovranno adottare misure tecniche e organizzative adeguate alla gestione dei rischi e alla prevenzione degli incidenti informatici, tenendo conto di alcuni elementi specificati nella norma a cui, probabilmente, seguiranno linee guida o altri provvedimenti. In caso di inadempienza sono stabilitepesanti sanzioni che vanno da un minimo di 12mila fino a 150mila euro.

A questo proposito è prevista, a livello nazionale, la designazione di un gruppo di intervento per la sicurezza informatica in caso di incidente e la costituzione di un'autorità nazionale competente in materia di sicurezza delle reti e dei sistemi informativi.
Il Nis fissa, inoltre, l'istituzione di un gruppo di cooperazione composto dai rappresentanti degli Stati membri, dalla Commissione e dall'Agenzia dell'Unione europea per la sicurezza delle reti e dell'informazione (ENISA). Un team che avrà l'obiettivo di promuovere la collaborazione tra i paesi dell'Unione in relazione alla sicurezza delle reti e dei sistemi informativi e facilitare lo scambio di informazioni.

1.2 GDPR, la sicurezza informatica dei dati personali

Il Gdpr, General data protection regulation, è il nuovo regolamento europeo sulla privacy e i dati personali che è diventato operativo in Italia a partire dal 25 maggio 2018 (Regolamento generale per la protezione dei dati personali n. 2016/679).Un testo che si compone di 99 articoli che rivolto a tutte le aziende che gestiscono qualsiasi tipo di dato personale. Non va confuso con la direttiva Nis, in quanto ha diverso oggetto e diverso ambito di applicazione. Tuttavia, le due normative possono sovrapporsi quando un incidente nella sicurezza informatica implica anche una violazione di dati personali.

Tra le prescrizioni, sono da menzionare: una richiesta di consenso in forma chiara, "comprensibile e facilmente accessibile"; l'istituzione di un registro delle attività dove si elencano le finalità dell'elaborazione dei dati, i destinatari, l'eventuale scadenza per la loro cancellazione; e la notifica delle violazioni entro 72 ore dall'avvenuta conoscenza. Inoltre, è richiesta la designazione di un "responsabile protezione dati" che avrà il ruolo di vigilare sull'applicazione effettiva del Gdpr nelle aziende.Il responsabile può essere scelto tra i dipendenti o anche nominato esternamente [18].

1.3 La direttiva 680, la sicurezza informatica dei dati trattati dalle autorità

Speculare al Gdpr è la direttiva 680 del 2016. Molte delle norme contenute sono simili a quelle presenti nel testo del nuovo regolamento europeo sulla privacy e i dati personali, se non addirittura uguali. Ma Quest'ultima direttiva ha un ambito di applicazione specifico e riguarda i trattamenti effettuati dalle autorità competenti a fini di: prevenzione, indagine, accertamento e perseguimento di reati; esecuzione di sanzioni penali; salvaguardia e prevenzione di minacce alla sicurezza pubblica.

In particolare, il testo prescrive che i dati siano conservati per il tempo necessario al conseguimento delle finalità per le quali sono trattati, sottoposti a esame periodico per verificarne la persistente necessità di conservazione e cancellati o anonimizzati una volta decorso tale termine. Introduce, inoltre, una nuova disciplina riguardo alla differenziazione tra categorie di dati (fondati su fatti ovvero su valutazioni) e di interessati, in ragione della loro specifica posizione processuale.

Inoltre, riguardo ai diritti dell'interessato (ricezione di informazioni, accesso, rettifica, cancellazione, limitazione del trattamento), il testo prevede che rispetto ai dati personali contenuti in una decisione giudiziaria, in atti o documenti oggetto di trattamento nel corso di accertamenti o indagini, nel casellario giudiziale o in un fascicolo oggetto di trattamento nel corso di un procedimento penale o in fase di esecuzione penale, l'esercizio di tali diritti è regolato dalle disposizioni normative che disciplinano tali atti e procedimenti.

In ambito giudiziario, la tutela degli interessati è quindi assicurata, per le parti, dalle garanzie che riconoscono i diritti di difesa all'interno del procedimento penale, anche con riguardo ai dati personali necessariamente oggetto di trattamento, assicurando quindi la possibilità di limitare l'esercizio dei diritti dell'interessato, conformemente alle esigenze di prevenzione, di indagine e processuali.

Per garantire i diritti in ambito giudiziario anche con riferimento ai terzi, è previsto uno speciale procedimento attraverso il quale qualsiasi interessato, durante il procedimento penale o dopo la sua definizione, può chiedere la rettifica, la cancellazione o la limitazione dei dati personali che lo riguardano.

In materia di sicurezza del trattamento, si prevede come obbligatoria anche per l'autorità giudiziaria la nomina del responsabile della protezione dati, in ragione dell'ausilio che tale figura può fornire nella gestione di trattamenti complessi e spesso inerenti dati sensibili, quali appunto quelli svolti in sede giurisdizionale.

Per quanto riguarda i trasferimenti di dati personali verso Paesi terzi o organizzazioni internazionali, si stabilisce che esso sia consentito solo nei confronti delle autorità competenti e per le finalità di pubblica sicurezza e in presenza di specifiche condizioni, tra cui l'adozione, da parte della Commissione dell'Unione europea, di una decisione di adeguatezza.

Il decreto individua nel Garante nazionale l'autorità deputata a vigilare sul rispetto delle norme attuative della direttiva in funzione della tutela dei diritti e delle libertà fondamentali delle persone fisiche, coinvolte dalle attività di trattamento di dati personali, escludendo il potere di controllo del Garante in ordine al trattamento svolto dall'autorità giudiziaria nell'esercizio delle funzioni giurisdizionali, comprese quelle del pubblico ministero. Infine, per quanto riguarda la violazione delle nuove norme, il testo prevede sanzioni amministrative che nei casi più gravi possono estendersi da 50mila a 150mila euro per le violazioni inerenti alle modalità del trattamento. Introduce anche sanzioni penali per il trattamento operato con finalità illegittime.

1.4 eIdas, la sicurezza informatica applicata alla firma e alle transazioni elettroniche

L'eIDAS, Electronic IDentification, Authentication and trust Services, è il regolamento europeo (n. 910/2014 in vigore dal luglio 2016) che disciplina la firma elettronica, i trasferimenti di denaro e altri tipi di transazioni elettroniche nel mercato unico europeo. Ha permesso di creare standard unici per la firma elettronica, certificati digitali, marche temporali, e altre forme di autenticazione elettronica, consentendo di sostituire documenti cartacei con equivalenti digitali che hanno lo stesso valore legale e riconoscimento ufficiale in tutti i paesi dell'Unione europea.

In particolare:

- fissa le condizioni a cui gli Stati membri riconoscono i mezzi di identificazione elettronica delle persone fisiche e giuridiche che rientrano in un regime notificato di identificazione elettronica di un altro Stato membro;
- stabilisce le norme relative ai servizi fiduciari, in particolare per le transazioni elettroniche;
- istituisce un quadro giuridico per le firme elettroniche, i sigilli elettronici, le validazioni temporali elettroniche, i documenti elettronici, i servizi elettronici di recapito certificato e i servizi relativi ai certificati di autenticazione di siti web

I paesi membri dell'Unione europea sono tenuti a riconoscere le firme elettroniche che rispettano gli standard fissati dall'eIDAS. In particolare, distingue tre tipologie di firme elettronica: firma elettronica definita come "un insieme di dati in forma elettronica, acclusi oppure connessi tramite associazione logica ad altri dati elet-tronici e utilizzati dal firmatario per firmare"; la firma elettronica avanzata, che deve soddisfare determinati requisiti; e la firma elettronica qualificata che, invece, si basa su un certificato qualificato ed è realizzata mediante un dispositivo sicuro per la creazione della firma.

1.5 La via italiana alla sicurezza informatica e il decreto Gentiloni

I due regolamenti e le due direttive costituiscono la cornice di riferimento entro le quali si collocano tutte le leggi e le applicazioni italiane. Un passo in autonomia sul fronte della sicurezza informatica interna è stato fatto dal governo Gentiloni nel febbraio del 2017. Ha approvato un programma nazionale per la cybersecurity in più fasi e un nuovo decreto, sostituendo il provvedimento dell'ex presidente del Consiglio Mario Monti che ha regolato l'architettura nazionale per la sicurezza cibernetica dal gennaio 2013 al 2017.

Il provvedimento ha rafforzato il ruolo del Cisr (Comitato interministeriale per la sicurezza della Repubblica) che ha il compito di emanare direttive con l'obiettivo di innalzare il livello della sicurezza informatica del Paese, avvalendosi del supporto del coordinamento interministeriale della parte tecnica del Cisr e del Dis (Dipartimento delle informazioni per la sicurezza).

Tra le novità, il Nucleo sicurezza cibernetica (NSC) viene ricondotto all'interno del Disper assicurare la risposta coordinata agli eventi cibernetici significativi per la sicurezza nazionale in raccordo con tutte le strutture dei ministeri competenti in materia.

Inoltre, è prevista una forte interazione con l'Agenzia per l'Italia Digitale (AgID) del Dipartimento della Funzione Pubblica, con il ministero dello Sviluppo Economico, con il ministero dell'Interno, con il ministero della Difesa e, infine, con il ministero dell'Economia e Finanze.

Il compito di definire linee di azione che devono assicurare i necessari livelli di sicurezza dei sistemi e delle reti di interesse strategico, sia pubblici che privati, è affidato al direttore generale del Dis che assume così un ruolo centrale nella sicurezza informatica italiana.

1.6 Il Cybersecurity Act

In vigore dal 27 giugno 2019, il Cybersecurity Act, nuovo strumento normativo europeo che mira a una sicurezza informatica più coesa e comunitaria. Si tratta di un Regolamento che ha lo scopo di creare un quadro europeo ben definito sulla certificazione della sicurezza informatica di prodotti ICT e servizi digitali [19].

Com'è fatto il Cybersecurity Act

Questo nuovo strumento normativo è composto da due parti. La prima specifica quello che è il ruolo dell'Agenzia dell'Unione europea per la sicurezza delle reti e dell'informazione. L'Enisa, istituita nel 2004, in questi anni ha fatto un po' da guardiano senza un vero ruolo operativo. L'agenzia, infatti, ha dato assistenza agli stati membri nella fase di elaborazione delle strategie sulla cybersecurity, lasciando di fatto, la gestione della sicurezza in mano al singolo Paese.
Con il Cybersecurity Act, il ruolo di Enisa diventa molto più operativo, entrando concretamente nella gestione di un cyberattacco. Inoltre, avrà compiti ben specifici in chiave di certificazione di servizi, processi e prodotti.

La certificazione, infatti, diventa un pilastro portante della seconda parte del Cybersecurity Act.
Ad oggi, la maggior parte delle certificazioni di cyber sicurezza esistenti nei vari stati membri, hanno esclusiva valenza nazionale e, pertanto, non vengono riconosciuti all'estero.
Col muovo regolamento europeo, invece, si tende ad unificare i processi, non tanto imponendo un certificato unico, quanto dettando le linee guida che rendano le certificazioni omogenee e riconosciute a livello comunitario.

Concretamente, l'agenzia Enisa predisporrà nuovi schemi europei di certificazione conformi al Cybersecurity Act. Questi schemi saranno successivamente adottati dalla Commissione UE, e diventeranno esecutivi e disponibili alle aziende europee. Chiaramente questi schemi andranno a sostituire i regolamenti nazionali. Per i prodotti certificati a livello nazionale, la validità rimarrà invariata fino alla loro scadenza.

SICUREZZA INFORMATICA E SMART GRID CYBER SECURITY

La rete intelligente, a differenza della rete tradizionale che segue un modello di generazione centralizzata di energia che dalle grandi centrali viene veicolata nelle reti di trasmissione, prevede la presenza di sistemi di generazione distribuita. Essi sono sistemi di produzione di elettricità da fonti rinnovabili, sotto forma di unità di piccola produzione, come possono essere gli impianti fotovoltaici residenziali o aziendali o piccole centrali a biomassa, allacciati direttamente alla rete elettrica di distribuzione.

Dato che le fonti rinnovabili non sono programmabili, gestire sistemi di generazione distribuita di energia richiede anche una “intelligenza” che si manifesta nella gestione del sistema elettrico complessivo così da consentirgli di gestire a livello locale eventuali surplus di energia redistribuendoli in aree vicine, prevenendo o riducendo al minimo un’interruzione potenziale.

Altra innovazione importante delle smartgrid è la gestione bidirezionale dell’energia, potendo riceverla, ma anche immetterla nel sistema quando è in eccesso, redistribuendo il flusso in tempo reale e a seconda degli effettivi bisogni.

Per riuscire in questo le smartgrid contano su dispositivi intelligenti, tali da permettere uno scambio continuo di informazioni tra tutti i nodi. In tal modo, oltre a ovviare a “buchi”, permette di ridurre gli sprechi. Da qui il suo valore di rete efficiente. Gli smart device che fanno parte della rete intelligente sono sensori, smartmeter, computer e altri apparati tecnologici.

Le smartgrid rappresentano quindi l’evoluzione della rete elettrica attuale con l’integrazione intelligente della gestione del flusso di energia elettrica dei produttori, degli utilizzatori e di entrambi.

La rete intelligente, quindi, è un'infrastruttura digitale che si trova sulla parte superiore della rete elettrica già esistente. Serve a monitorare le condizioni della rete, il consumo e la generazione di energia, nonché ad automatizzare molte delle sue operazioni. La sovrapposizione di una rete di dati non è solo un piccolo aggiornamento della rete elettrica, ma sarà una rivoluzione nel modo in cui le utility generano e distribuiscono energia e i consumatori consumano elettricità. I principali scopi e obiettivi della rete intelligente si possono riassumere in:

1. Migliorare l'affidabilità della rete elettrica
2. Migliorare la sua efficienza complessiva
3. Ridurre i costi di distribuzione e generazione
4. Consentire il monitoraggio in tempo reale della rete elettrica

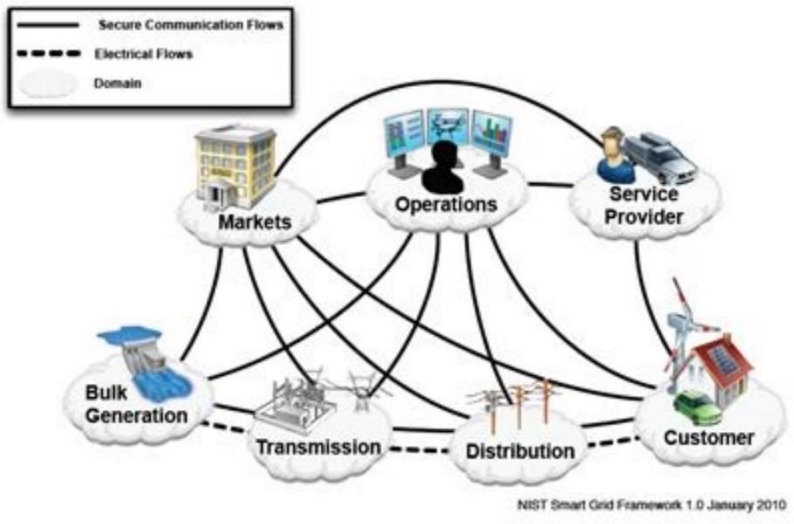

Figura 1. Flusso di dati ed elettricità attraverso un dominio di rete intelligente sicuro (Linee guida NISTIR 7628)

Per raggiungere questi obiettivi e attività, la rete intelligente impiegherà dispositivi e strumenti intelligenti sia lato cliente che da quello dell'utilità. Alcune di queste tecnologie includono l'uso di contatori intelligenti; ad esempio il microprocessore ha consentito ai contatori elettrici di comunicare al consumatore condizioni di rete e prezzi dell'elettricità in tempo reale [20].

1.7 Smart Grid Cyber-Threat

Con l'avvento del cyber-crimine è nata una sempre maggiore preoccupazione per la sicurezza, specialmente per quanto riguarda la comunicazione. Sebbene la cyber-threat sia associata a tutti gli aspetti dei domini della smartgrid, inclusi i dispositivi smartgrid, la principale preoccupazione riguarda le tecnologie di comunicazione che sono il cuore della smartgrid.

Progettato per il contatto in tempo reale, ciascuno di questi dispositivi intelligenti offrirà un nuovo vettore di attacco che potrebbe essere sfruttato se non gestito con cautela.

Il rapporto NISTIR 7628 ha identificato alcuni esempi di potenziali rischi associati alla Smart Grid, che sono:

- una maggiore complessità aumenta l'esposizione a potenziali aggressori e errori involontari;
- le reti che si collegano più frequentemente ad altre reti introducono vulnerabilità comuni che ora possono estendersi a più domini Smart Grid (vedere Figura 1) e aumentare il potenziale di guasti a cascata;
- più interconnessioni presentano maggiori opportunità di attacchi “denial of service”, introduzione di codici dannosi (nel software / firmware);
- all'aumentare del numero di nodi di rete, aumenta anche il numero di punti di ingresso e percorsi che i potenziali avversari potrebbero sfruttare;
- l'ampia raccolta di dati e i flussi di informazioni bidirezionali possono ampliare il potenziale per compromessi della riservatezza dei dati personali e violazioni della privacy dei clienti.

Inoltre, il rapporto afferma che "Il rischio è il potenziale di risultati indesiderati conseguenti da fattori interni o esterni, determinati dalla probabilità di accadimenti e dalle conseguenze associate. "

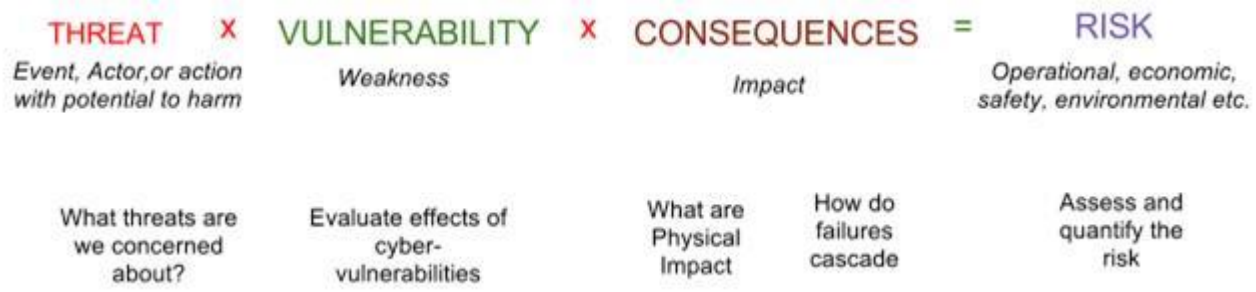

Figura 2. Modello generico di rischio [21]

Sulla base della metodologia di valutazione del rischio esistente, dovrebbero essere derivati approcci di valutazione del rischio della rete intelligente che identifichino minacce, risorse e vulnerabilità e il potenziale impatto che può causare all'infrastruttura della rete intelligente. La Smart Grid è l'infrastruttura nazionale vitale, la sicurezza informatica della rete intelligente non dovrebbe solo affrontare le potenziali minacce di dipendenti scontenti, terroristi e operazioni di spionaggio, ma dovrebbe anche prendersi cura delle vulnerabilità derivanti da errori degli utenti, guasti alle apparecchiature e catastrofi naturali.

1.8 Smart Grid Cyber Strategy

La smartgrid è un ecosistema complesso che non è solo una fusione di vari sistemi, reti e processi, ma anche la convergenza di varie tecnologie come l'IT e la comunicazione con la rete elettrica. Per un sistema tecnologico così complesso, il paese dovrebbe consultare tutte le parti interessate per sviluppare un quadro globale di sicurezza informatica che sia tutto compreso, interoperabile e di natura solida. Inoltre, la sicurezza informatica non dovrebbe essere considerata come retrofit, ma dovrebbe far parte dello sviluppo della rete intelligente stessa. Organizzazioni come l'Istituto nazionale di standard e tecnologia (NIST), l'Agenzia europea per la sicurezza delle reti e dell'informazione (ENISA) hanno sviluppato linee guida per la sicurezza informatica delle reti intelligenti, che dovrebbero essere prese in considerazione durante lo sviluppo di una strategia informatica coerente. Secondo il rapporto NISTIR 7628, la strategia di cyber grid intelligente dovrebbe essere progettata in modo tale da affrontare i processi di prevenzione, rilevazione, risposta e recupero per contrastare eventuali minacce esistenti e potenziali.

Di seguito sono riportate alcune delle linee guida chiave.

- Gli organismi competenti dovrebbero sviluppare un quadro politico e regolamentare che fornisca un ambiente di supporto per gli obiettivi di sicurezza informatica
- Sviluppare metodologie di valutazione del rischio che valutino minacce, vulnerabilità e impatto.
- La privacy è di fondamentale importanza e dovrebbero essere prese misure per proteggere quattro aspetti chiave della privacy degli utenti:
  1) informazioni personali
  2) privacy personale
  3) privacy comportamentale

4) privacy delle comunicazioni personali.

- Sviluppare un'architettura di sicurezza collegata al modello di riferimento concettuale della griglia intelligente.
- Sviluppare schemi di certificazione per dispositivi, reti, sistemi e processi di smartgrid e / o creare un meccanismo di governance della sicurezza che consenta agli stakeholder di confrontare le proprie infrastrutture.
  Promuovere il programma di ricerca per la sicurezza informatica della rete intelligente sfruttando il programma di ricerca esistente. Al riguardo il NIST ha identificato quattro sfide chiave per la ricerca e lo sviluppo della sicurezza informatica:
  1) Sicurezza a livello di dispositivo
  2) Gestione crittografica e delle chiavi
  3) Sicurezza relativa ai problemi di rete
  4) Sicurezza a livello di sistema
- Progettare programmi di sensibilizzazione e formazione sulla sicurezza conformi alle politiche e alle normative locali, statali e nazionali dell'organizzazione che supportano la sicurezza della rete intelligente globale.

VULNERABILITÀ DELL'IOT: LE BEST PRACTICE PER LA MITIGAZIONE DEL RISCHIO

Le connessioni eterogenee tipiche della Internet of Things determinano un "aumento della superficie esposta" con una estensione esponenziale delle vulnerabilità dell'IoT, soprattutto in ambito industriale.
Quello dell'Internet of Things e in particolare delle vulnerabilità dell'IoT, complice anche la recente evoluzione della tecnologia e delle reti 5G, è indubbiamente uno dei temi maggiormente dibattuti. Per due motivi, il primo riguarda gli aspetti relativi alla sicurezza dei protocolli Internet per lo scambio dei dati. Il secondo riguarda, invece, la visione strategica e la governance sulla condivisione e sostenibilità dei Big Data originati dalle interconnessioni eterogenee tra device, ma anche sulla loro strutturazione effettuata attraverso strumenti capaci di enderli sempre più qualitativi ed anche per favorire l'automazione di interi processi [22].

1.9 Vulnerabilità dell'IoT: i possibili rischi

L'internet delle cose (IoT) è la rappresentazione sempre più diffusa e consolidata di un ecosistema digitale caratterizzato da una fitta rete di connessioni (sempre) più eterogenee di dispositivi.
In tale ambiente digitale, tutti i dispositivi "smart" interagiscono tra loro con varie tipologie di connessione per finalità di controllo, monitoraggio, attivazione di funzioni da remoto, trasferimento di informazioni e tanto altro ancora.
Ed infatti, l'integrazione dei dispositivi si caratterizza per l'uso di applicazioni connesse a Internet, in grado di elaborare informazioni attraverso la componente tecnologica del device che, per questo, diventa intelligente.
Si pensi ad esempio alla gestione della domotica, oppure ai dispositivi medicali, oppure ancora agli orologi smart che monitorano le performances di chi li indossa.
In tale scenario, le connessioni eterogenee determinano quello che in information security si definisce tecnicamente un "aumento della superficie esposta", con una estensione esponenziale delle vulnerabilità hardware e software, collegate a potenziali rischi di exploitation da parte dei criminali informatici.

Uno degli attacchi più significativi e purtroppo in continua espansione dell'ecosistema IoT è il DDoS (Distributed Denial of Service) che sfrutta le vulnerabilità del protocollo correlato all'IoT per perpetrare, più spesso, attacchi sistemici.
Ciò, attraverso una sempre maggiore proliferazione delle botnet "infettate" da malware e server vulnerabili che generano automaticamente ulteriori attacchi contro obiettivi a loro volta vulnerabili.
In molti casi di attacchi DDoS, i criminali informatici tendono a falsificare l'indirizzo IP di un target connesso all'IoT al fine di inviare ad un server vulnerabile una serie di richieste di informazioni che generano una quantità amplificata di risposte (packet amplification) verso l'indirizzo IP della vittima, vanificando di fatto le capacità di difesa del server colpito.

Molti devices eterogeneamente connessi all'IoT, infatti, pur disponendo di una buona connettività, accedono a segmenti di rete non monitorati, rendendosi così vettori di un volume significativo del traffico degli attacchi DDoS, anche per la loro ridotta capacità di elaborazione.
In tali casi, non è infrequente che i dispositivi dell'IoT vengano anche utilizzati come proxy e, per questo, la compromissione di un device connesso ad una rete rende inevitabilmente vulnerabile tutte le altre risorse internamente ed esternamente connesse.
In genere, gli attacchi DDoS sfruttano i protocolli Internet più diffusi e consolidati come Network Time Protocol (NTP), risolutori DNS (Domain Name System) e SSDP (Simple Services Discovery Protocol) e si caratterizzano per la falsificazione dell'indirizzo IP ed il cosiddetto "packet amplification".

Vi sono tuttavia anche nuove rilevate vulnerabilità che risultano correlate all'uso del protocollo Constrained Application Protocol (CoAP).

Si tratta, in particolare, di un protocollo Machine-to-Machine (M2M) che può essere eseguito su smart device con memoria e capacità di elaborazione limitate, parimenti vulnerabile tanto alla falsificazione "reflective" dell'IP quanto alla packet amplification, tipici del dell'attacco DDoS.

1.10 Vulnerabilità dell'IoT: il pericolo dei mobile malware

I criminali informatici sono sempre più abili nell'exploitation di vulnerabilità hardware e software di dispositivi connessi all'IoT al fine di penetrare in modo pervasivo reti aziendali per perpetrare attacchi più sistemici, con impatti ben più gravi anche in termini di continuità operativa dei processi primari.

I mobile malware consentono ai cyber criminali di assumere il controllo degli smart device forzandone le password di protezione dei dispositivi (spesso di default) o anche con algoritmi di risoluzione (negli attacchi cosiddetti brute force), al fine di utilizzarli sia negli attacchi DDoS sia anche nel mining malevolo di cryptovalute ovvero, ancora, per l'inserimento di dispositivi in reti botnet per ulteriori attacchi sistemici.
Diversi sono i malware rilevati negli attacchi ai dispositivi connessi all'IoT.
Secondo il report IoT di Kaspersky Lab, nella prima metà del 2018, i dispositivi IoT sono stati attaccati da più di 120.000 varianti di malware (più del triplo della quantità di malware IoT registrati in tutto il 2017).
Viene, tra l'altro, rilevata una crescita vertiginosa dei mobile malware con un trend che già nel 2017 era aumentato di ben dieci volte rispetto al 2016.
Nel rapporto SonicWall 2019 viene, addirittura, rilevato un aumento del 55% degli attacchi malware IoT rispetto ai primi due trimestri del 2018, con evidenza di un aumento delle connessioni ad internet di dispositivi senza adeguate misure di sicurezza, nonché di una loro exploitation per dispensare payload di malware.
Molti mobile malware sfruttano la tecnologia di comunicazione peer-to-peer (P2P) che per la connessione dei dispositivi all'IoT non richiede una procedura di autenticazione e, soprattutto, non utilizza alcun algoritmo di crittografia per proteggere il traffico di rete.

In tale contesto, il server di back-end collegato ai device dispone di un range di numeri seriali noti dei devices connessi e al criminale informatico basterà solo scansionarli per individuarli e, dunque, accedervi senza controllo.
Altri malware, come ad esempio Mirai, utilizzano i dispositivi come nodi di una botnet poi utilizzata per attacchi sistemici su larga scala.
Anche IoTroop/Reaper è una botnet che attacca i dispositivi vulnerabili dell'IoT non aggiornati da patch di sicurezza.
VPNFilter, infine, è un malware che colpisce sia i router sia alcuni dispositivi di storage di rete.
Vi sono poi tipologie di attacco cryptojacking che sfruttano, invece, la potenza di calcolo dei dispositivi colpiti e, soprattutto, la loro connessione permanente per minare criptovalute aziendali.

1.11 Uno scenario di rischio nella Industrial IoT

Queste tipologie di attacco si rilevano particolarmente pervasive ed incidenti, soprattutto nell'Industrial Internet of Things (IIoT), laddove vengono presi di mira i sistemi di controllo di infrastrutture critiche ed è altamente strategico investire anche nella tecnologia operativa (OT) per mitigare il rischio malware.
Ed infatti, come ampiamente desumibile dagli ultimi report sulla sicurezza informatica, i dispositivi connessi all'IoT o all'IIoT rappresentano per i criminali informatici un importante elemento di vulnerabilità da exploitare per colpire con attacchi sistemici.
E così anche un circuito di videosorveglianza, ove non monitorato ai fini del risk assessment, può agevolmente costituire una minaccia per una rete interna, sia sotto il profilo dello spionaggio sia, nei casi più gravi, del sabotaggio.
È anche vero che in molte aziende, soprattutto industriali, l'introduzione di forme di lavoro smart e sempre più flessibili ha determinato le condizioni per una maggiore connettività all'IoT, attraverso un maggiore uso di dispositivi con accesso da remoto ai sistemi informativi aziendali al fine di favorire l'incremento della produttività aziendale.
In tale scenario, l'aumento della superficie esposta richiede la governance ed il presidio dei dispositivi mobili, in molti casi di proprietà dei dipendenti, al fine di monitorarne ogni rischio valutabile onde mitigarne gli impatti, nell'ambito di una rigida gestione dei privilegi degli accessi.

1.12 Vulnerabilità IoT: le best practice di sicurezza

Per questo, in information security, è molto importante elaborare una rigida policy BYOD (Bring Your Own Device) che disciplini l'uso e le best practice dei device mobili per assicurare la sicurezza dell'infrastruttura IT e mitigare il rischio di vulnerabilità della rete aziendale, proteggendo il patrimonio informativo aziendale e le informazioni afferenti ai dati personali, anche in compliance al GDPR.

È dunque essenziale che il dispositivo mobile venga preliminarmente registrato servendosi di un software ad hoc per il Mobile Device Management (MDM) che ne assicuri anche la previa ed imposta installazione di un software di sicurezza antimalware ed antivirus supportato che preveda, tra l'altro, la protezione del device con password forte (sequenze alfanumeriche, caratteri speciali, autenticazione a due fattori, expiration time ed aggiornamento) e blocco schermo (PIN, password, impronta digitale o riconoscimento facciale) nonché aggiornamenti costanti del firmware del device e l'installazione della patch di sicurezza.

È poi parimenti importante l'adozione di metodi di connessione sicura e virtualizzata (VPN), soprattutto per gli accessi da remoto tramite Wi-Fi pubblico non protetto, adottando tecnologie di crittografia efficaci, con autenticazione periodica dell'utente del device, ovvero attraverso l'uso di uno stack di sicurezza basato su cloud per fornire, da qualunque luogo, un accesso sicuro a Internet su ogni dispositivo.

Inoltre, l'opportuno coordinamento tra la compliance al GDPR e la protezione dei sistemi informativi aziendale e del patrimonio informativo aziendale richiedono anche una rigida separazione tra dati aziendali e personali attraverso la best practice della segmentazione dei dati, per consentire di eliminare agevolmente ed in sicurezza i dati aziendali nel caso di termination of duties della risorsa umana aziendale che utilizza un dispositivo intelligente connesso.

1.13 Mettere al sicuro il patrimonio informativo aziendale

L'uso dei dispositivi connessi all'IIoT pone sempre il problema della vulnerabilità del patrimonio informativo aziendale esposto a data leak, in considerazione della circostanza che molti dati vengono ormai archiviati su dispositivi mobili.
In questi casi diventa essenziale imporre una policy ad hoc per assicurare che tutti i device mobili connessi alla rete soddisfino gli standard ed i protocolli di crittografia condivisi.
Inoltre, nel mobile management è fondamentale assicurarsi anche l'identificazione e la cancellazione da remoto di device cosiddetti "jailbreak" ovvero di quei dispositivi che consentano l'installazione di app, software e pacchetti di terze parti non sicuri che possono potenzialmente compromettere il sistema informativo aziendale.
Non vi è dubbio che l'azione dei criminali informatici è prevalentemente orientata verso lo sfruttamento delle vulnerabilità presenti nei dispositivi IoT e si aggiunge a quella tipica di accesso abusivo con credenziali di default.
La presenza di vulnerabilità basilare nella sicurezza dei dispositivi intelligenti dell'IoT è oggi la principale minaccia collegata al più elevato rischio di diffusione di malware.
Dal recente report 2019 di Honeypot è emerso che i nuovi dispositivi IoT subiscono in meno di un giorno il tentativo di fare leva sulle vulnerabilità note e sono soggetti in meno di 5 minuti a tentativi di accesso mediante le credenziali IoT predefinite.
A questo proposito, è utile ricordare che il Cybersecurity Act, favorirà la realizzazione di un mercato unico digitale anche attraverso l'introduzione di standard europei per la certificazione di prodotti e processi afferenti alla protezione del dominio cibernetico, secondo il paradigma, invero mutuato dal Regolamento (UE) 2016/679, della security by design.
In punto di progettazione dell'architettura della sicurezza, entrambe le normative regolamentari presentano un denominatore comune che trae origine dalla esigenza di proteggere i dati "sensibili" che possono costituire anche patrimonio informativo delle aziende produttrici, nonché informazioni classificate degli operatori dei servizi essenziali o di fornitori di servizi digitali, intesi questi ultimi come infrastrutture critiche per la loro importanza strategica e, talvolta, per la sicurezza nazionale.

1.14 Conclusioni

In questo scenario, anche il tema della privacy by design disciplinato dal GDPR, dovrà essere sempre più integrato nella nuova accezione di security by design, ed in quanto principio fondamentale di tutela, costituirà un requisito essenziale per i titolari e i responsabili del trattamento, soprattutto, in materia di appalti pubblici.

Inoltre, la creazione di un mercato unico digitale, nella cornice di un sistema europeo di certificazione (entro il 2021) di prodotti e servizi con l'attribuzione di un mandato permanente con poteri di intervento a supporto degli Stati membri delle crisi cibernetiche attribuito all'Enisa, Agenzia Europea per la sicurezza informatica, renderà ancora più concreto l'impegno dell'Europa nella protezione sempre più efficace del dominio cibernetico e nella proattiva sicurezza e resilienza dei sistemi.
L'uso dei dispositivi connessi.

RETI 4G/5G: PROFILI DI VULNERABILITA' E POSSIBILI CONTROMISURE

Le reti mobili stanno evolvendo velocemente nell'ultimo periodo. Dall'entrata in funzione del 4G, si sono potuti toccare con mano i benefici della banda larga anche nelle connessioni in mobilità. A distanza di qualche anno, il 4G Plus ha permesso un incremento di prestazioni sostanziale rispetto alla tecnologia di base. Con l'arrivo del 5G, previsto nel 2020, saremo di fronte ad una vera e propria rivoluzione, che prevede la proliferazione dei dispositivi IoT e un incremento di velocità di banda nell'ordine dei Gbit/s. Come appreso dal recente passato, ogni miglioria tecnologica porta con sé l'introduzione di nuove

vulnerabilità di progettazione, potenzialmente sfruttabili da utenti malevoli. È quindi facilmente ipotizzabile che, come per le connessioni Bluetooth e WiFi, anche le connessioni mobili soffrano di potenziali errori di progettazione [23].

Difatti, nell'ultimo periodo sono emersi numerosi studi nei quali alcuni ricercatori evidenziano le vulnerabilità dello standard 4G. La cosa ben più grave è che alcune di esse impattano anche sul nuovo standard 5G di imminente introduzione.

Analizziamo, quindi, di seguito le falle più gravi e verifichiamo se vi sono contromisure applicabili.

1.15 AUTHENTICATION RELAY ATTACK (4G)

Questo tipo di attacco potrebbe non solo consentire a un attaccante di compromettere la rete cellulare per leggere i messaggi – in entrata e in uscita – delle vittime, ma anche di sostituirsi a qualcun altro per l'esecuzione di un qualsiasi atto delinquenziale.

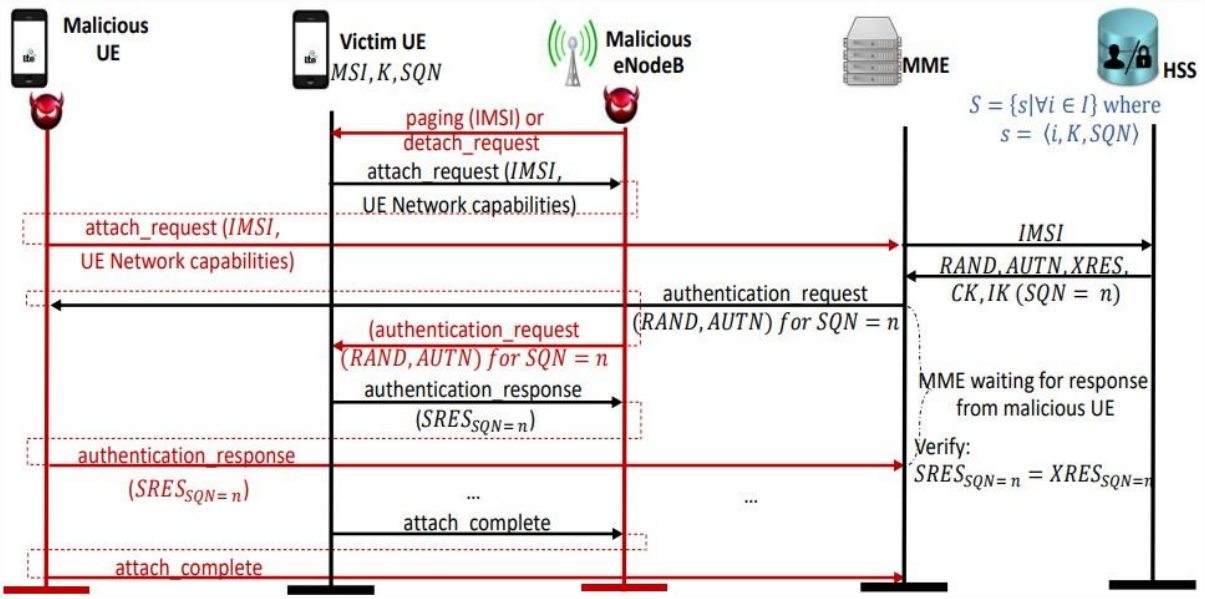

Figura 3 diagramma di flusso dell'attacco di un malintenzionato

Come è banalmente intuibile analizzando la Figura 3, disponendo di una cella e di un dispositivo effimeri, è possibile:

1. forzare la disconnessione del dispositivo vittima da una cella valida (Stealthy kicking-off Attack);
2. forzare la connessione di quest'ultimo alla cella effimera;
3. reindirizzare la richiesta di autenticazione fornita dal dispositivo vittima al fornitore del servizio;
4. tramite la cella effimera, far decodificare i messaggi di autenticazione al dispositivo vittima;
5. completare l'autenticazione, al posto del dispositivo vittima, con dispositivo effimero.

A seguito dell'autenticazione, l'attaccante possiede una copia speculare (agli occhi della rete) del dispositivo della vittima.

1.16 aLTEr (4G/5G)

Gli standard 4G/5G prevedono diversi meccanismi di sicurezza. Quando un dispositivo si collega alla rete, stabilisce un'autenticazione reciproca (cioè la rete e il telefono possono verificare rispettivamente le loro identità) con il fornitore del servizio, ottenendo una chiave di crittografia condivisa per la trasmissione sicura dei dati.
Tale chiave è utilizzata per crittografare anche il traffico di "controllo". Quest'ultimo è protetto da meccanismi di controllo dell'integrità, il che significa che l'utente malintenzionato non è in grado di manipolare il traffico durante la trasmissione.

Scendendo nel dettaglio, aLTEr interviene sfruttando un errore di progettazione relativo alla gestione del "livello 2" dello stack di rete. Nello stack del protocollo TCP/IP, il layer 2 è quello che corrisponde al *Data Link*, posto gerarchicamente al di sotto dei livelli TCP ed IP, nei quali è implementato il meccanismo di sicurezza precedentemente illustrato.
Brevemente: i dati utente, a livello 2– *Data Link*, vengono crittografati in modalità contatore (AES-CTR), in cui l'algoritmo genera un keystream (caratteri pseudorandomici). Il testo cifrato viene quindi calcolato dallo XORing del keystream con il testo in chiaro.
Non essendo stati previsti negli standard dei controlli di integrità a questo livello, e conoscendo il testo in chiaro originale della richiesta (ipotizzando i siti più visitati dalla vittima, tramite Website Fingerprinting), è facile modificare il contenuto di un pacchetto DNS anche se crittografato, reindirizzando la richiesta a un server DNS sotto il controllo dell'avversario.

Riassumendo: un utente malintenzionato può creare una cella LTE effimera, alla quale il dispositivo della vittima si collegherà in automatico, poiché il segnale di quest'ultima sarà più potente per prossimità rispetto a una qualsiasi cella reale limitrofa. A questo punto il traffico e le relative richieste effettuate dal dispositivo mobile della vittima verranno tutte filtrate dalla cella effimera, per essere poi inoltrate alla cella reale, ignara di eventuali manipolazioni avvenute precedentemente.

La dimostrazione dell'attacco è visionabile tramite il video https://www.youtube.com/watch?v=3d6lzSRBHU8.

1.17 ToRPEDO (4G/5G)

Acronimo di *TRacking via Paging mEssage DistributiOn*, sfrutta una debolezza nel protocollo di *paging* utilizzato dagli ISP per segnalare ad un dispositivo in stato di inattività una notifica, prima dell'effettivo arrivo di una comunicazione in entrata. Se un dispositivo non stabilisce, per un certo periodo, una comunicazione attiva con la rete cellulare, entra in una sorta di modalità di risparmio della batteria. In questo contesto la rete cellulare, prima che una chiamata o un messaggio di testo raggiungano il dispositivo, invia a quest'ultimo un messaggio di paging per anticipare, tramite una notifica, una chiamata o un messaggio in arrivo. Il messaggio di paging incorpora in sé l'identificativo temporaneo del dispositivo – TMSI (*Temporary Mobile Subscriber Identity*) (un numero univoco che lo smartphone utilizza per interfacciarsi con la rete GSM. Serve a garantire la riservatezza degli utenti, evitando di mandare il codice IMSI: https://it.wikipedia.org/wiki/IMSI) – che viene aggiornato, a seguito dell'aggancio ad una cella, solo in rari casi.
I ricercatori hanno scoperto che, effettuando e annullando un gran numero di chiamate in un breve lasso di tempo, avviene, da parte del fornitore del servizio, l'aggiornamento del valore TMSI nel proprio database più frequentemente del solito. Quanto detto si compie unicamente sulla stazione mobile, senza generare un tipo di avviso nel dispositivo vittima. A seguito di quanto descritto, utilizzando un IMSI Catcher (ad esempio Stingrays and DRTBox), un malintenzionato può verificare se la vittima si trova in un raggio di distanza entro il quale è possibile intercettare eventuali comunicazioni.

Quindi, conoscendo il *Paging Occasion* (PO) [24], l'attaccante potrebbe:

- tracciare il dispositivo vittima;
- dirottare il canale di paging iniettando o negando i messaggi di paging di controllo;
- effettuare lo spoofing delle comunicazioni;

- lanciare attacchi DoS, saturando le risorse del dispositivo vittima, fino al punto da mandarlo in blocco.

1.18 PIERCER IMSI-Cracking attack (4G/5G)

I ricercatori hanno ulteriormente precisato che Torpedo può fungere da apripista per altri due tipi di attacco.Il primo, denominato Piercer (*Persistent Information ExposuRe by the CorEnetwoRk*), consente all'attaccante di determinare l'IMSI (*International Mobile Subscriber Identity*) della vittima sulla rete 4G/5G, ed associarlo al proprio dispositivo.I numeri IMSI sono crittografati nelle reti 4G e 5G, in modo da poterli proteggere da eventuali attacchi; tuttavia i ricercatori hanno nuovamente riscontrato che tali protezioni sono inadeguate.Il problema sta nell'implementazione del vettore, denominato per l'appunto Piercer, che espone i numeri IMSI sull'attuale rete 4G e sulla futura rete 5G. Alcuni fornitori di servizi utilizzano l'identificativo IMSI al posto del TMSI nei messaggi di paging per identificare i dispositivi con servizi "eccezionali", o per verificare servizi rimasti in stato "pending". I test effettuati hanno rivelato che è possibile dare l'impressione, al fornitore di servizi, che si stia verificando uno di questi casi eccezionali, in modo da costringere il dispositivo a rivelare il proprio IMSI.

Il secondo attacco, detto IMSI-Cracking attack, utilizza un attacco a forza bruta per individuare il codice IMSI su rete 4G o 5G.

Con il numero IMSI a disposizione, gli aggressori possono inscenare attacchi di vario genere: potrebbero spiare la lista delle chiamate della vittima, registrare le conversazioni audio, leggere gli SMS in arrivo, individuare l'esatta posizione del dispositivo, quindi gli spostamenti della vittima durante l'arco della giornata. E molto altro ancora…

1.19 POSSIBILI CONTROMISURE

Per l'**Autentication Relay Attack** non vi sono contromisure. Fonti attendibili confermano che tale metodologia è spesso sfruttata dagli organismi investigativi statunitensi, per perseguire eventuali azioni delinquenziali.

Per quanto riguarda **aLTEr**, l'utente non ha modo di difendersi, se non quello di consultare esclusivamente siti web che utilizzano HSTS (*HTTP StrictTransport Security*), cosa ben diversa dal protocollo HTTPS (*HTTP secure*).

Sta agli *Internet Service Provider (ISP)*, quindi ai gestori telefonici, utilizzare protocolli di autenticazione del tipo AES-GCM o ChaChat20-Poly1305. Come evidenziato anche dai ricercatori, però, una modifica dell'attuale infrastruttura comporterebbe un elevato sforzo finanziario e organizzativo. Perciò, la maggior parte dei gestori potrebbe decidere di non fixare la vulnerabilità, dal momento che non è obbligatorio farlo.

Altra contromisura potrebbe consistere nell'adozione del protocollo *DNS over TLS or DTLS*, che permetterebbe di proteggere i dispositivi provvedendo alla cifratura del traffico DNS, garantendo la di fatto l'integrità dello stesso.Ad ogni modo, allo stato dell'arte, ne deriva che anche il 5G potrebbe essere potenzialmente insicuro a questo tipo di attacco. Essendo la crittografia autenticata una funzione ad implementazione facoltativa, molti ISP potrebbero decidere di non adottarla nelle nuove reti 5G che stanno implementando.

Per quanto riguarda ToRPEDO, PIERCER ed IMSI-Cracking, vengono sfruttate debolezze intrinseche nei protocolli LTE 4G e 5G, che riguardano le prime fasi dell'interazione con i terminali mobili ed al mutuo riconoscimento con la rete in previsione di trasferimento di messaggi e chiamate.

1.20 CONCLUSIONI

Le dinamiche di autenticazione ed identificazione reciproca dovrebbero essere rese più sicure sin dalla fase di progettazione. Sarebbe opportuno evitare l'applicazione di metodi di autenticazione antiquati volti a garantire vecchie esigenze, dipendenti da reti e terminali meno performanti, o a metodi di comunicazione di base (ad esempio la retro-compatibilità) in cui non si è tenuto conto del concetto di "security by design".

Come si comporteranno i vari fornitori? Probabilmente non prenderanno alcun provvedimento. Secondo gli stessi ricercatori è difatti molto probabile che queste vulnerabilità siano troppo onerose da correggere (sia tecnicamente che economicamente), e che sia necessario aspettare una radicale modifica del protocollo per una reale risoluzione del problema.

Nel fra tempo sarà meglio tener presente che gli input digitali che ci giungono, potrebbero non provenire dal reale mittente della comunicazione e/o potrebbero essere intercettati da chiunque sia a portata di tiro.

RAPPORTO ENISA SU STANDARD DI SICUREZZA IOT

1.21 ENISA

L'Agenzia dell'Unione europea per la sicurezza delle reti e dell'informazione (ENISA) è una rete in materia di sicurezza delle informazioni per l'UE, i suoi Stati membri, il settore privato e i cittadini. ENISA collabora con questi gruppi per sviluppare consigli e raccomandazioni sulle buone pratiche in materia di sicurezza delle informazioni. Assiste gli Stati membri nell'attuazione della pertinente legislazione dell'UE e lavora per migliorare la resilienza di infrastrutture e reti di informazione. ENISA cerca di migliorare le competenze esistenti negli Stati membri sostenendo lo sviluppo di comunità transfrontaliere impegnate a migliorare sicurezza delle reti e delle informazioni in tutta l'UE. Maggiori informazioni su ENISA e il suo lavoro possono essere disponibile su www.enisa.europa.eu.

1.22 Analisi Divario degli standard di sicurezza IoT

Nel 2017, l'ENISA ha studiato e definito una serie di raccomandazioni di sicurezza di base per l'IoT. Lo scopo del lavoro era di fornire informazioni dettagliate sui requisiti di sicurezza dell'IoT, mappando risorse critiche e minacce, valutare possibili attacchi e identificare potenziali buone pratiche e misure di sicurezza da applicare per proteggere i sistemi IoT.

Lo studio in [25] analizza le lacune e fornisce le linee guida per, in particolare, lo sviluppo o il riposizionamento di norme, facilitando l'adozione di norme e laloro standardizzazione in UE nel settore della sicurezza delle reti e dell'informazione. L'ENISA porta in questo rapporto il suo know-how tecnico e organizzativo che può essere ulteriormente sfruttato per gli standard in termini di estensione o valutazione per renderli più appropriati parti e più conformi al quadro normativo prevalente.

Particolare attenzione è data alle esigenze dell'UE relative ai sistemi emergenti di certificazione della cibersicurezza che opererà nell'ambito del quadro europeo di certificazione della sicurezza informatica eagli strumenti di valutazione della sicurezza adottati nel mondo dell'IoT, urgentemente necessari per integrare le iniziative esistenti, le buone pratiche e le linee guida del settore sulla sicurezza dell'IoT.

Dall'analisi è emerso che non esiste un divario significativo tra gli standard. Esistono però Standards diversi, adottati dai vari produttori, per rendere sicuro un dispositivo o un servizio. Tuttavia, quando si fa riferimento a IoT, ci si riferisce a un ecosistema di non solo dispositivi e servizi. Inoltre, il contesto di utilizzo dell'IoT, la scalabilità e altre particolarità complicano ulteriormente il campo e richiedono una maggiore flessibilità negli approcci. Il divario negli standard per la sicurezza dei dispositivi IoT (standard non trattati in modo olistico) ha come conseguenza la possibilità diimmettere sul mercato un dispositivo in grado di autenticare l'utente, di crittografare i dati che trasmette, di decrittografare i dati che riceve, che può consegnare o verificare la prova di integrità, ma che sarà ancora insicuro.
Allo stesso modo, l'organizzazione che produce il prodotto o servizio IoT, può avere processi di sviluppo definiti nelle linee guida di gestione come quelli della ISO-27000 ma offrire ancora un prodotto insicuro.
La sfida per regolatori e fornitori è quella di portare sul mercato solo dispositivi IoT sicuri e questo richiede un approccio diverso, che dovrà essere abbastanza flessibile da adattarsi alla natura dell'ecosistema IoT dinamico.

Una ulteriore sfida è riuscire ad immaginare come sarà la società tra qualche anno e considerare le minacce per la società in quel momento. l'ipotesi generale è che le TIC raggiungeranno in modo molto più capillare la società con più connettività e ciò richiederà una risposta adeguata alla sicurezza informatica. Le preoccupazioni dei prossimi anni, tuttavia, vanno ben oltre l'ambito delle tecnologie di sicurezza e delle raccomandazioni contenute nel documento ENISA, si estendono per acquisire una migliore comprensione della società e del modo in cui le TIC, e in particolare le TIC, che incorporano la sicurezza informatica, incideranno sulla vita quotidiana.

L'elenco dei requisiti di sicurezza per la sicurezza IoT e la sua mappatura a standard specifici può fungere da trampolino verso la sicurezza olistica ed efficace dell'IoT. Non sono solo le sfide tecnologiche alla base chiedendo soluzioni adattive, basate sul contesto e sul rischio, ma anche i vincoli del mercato dell'IoT devono essere presi in considerazione, per non ostacolare la competitività e l'innovazione.

1.23 L'opportunita' di certificazione

Lo scopo generale degli standard dal punto di vista del mercato è duplice definire quale standard è destinato a raggiungere: (1) interoperabilità e (2) fiducia. Il ruolo degli standard nel dominio della fiducia è meno ben definito e in un contesto di sicurezza è difficile da stabilire in parole povere. Quando si fa riferimento all'IoT, non si dovrebbero considerare solo i singoli dispositivi ma più in generalela connettività e le interdipendenze di dispositivi, servizi, persone, processi e dati che richiedono quindiuna visione molto più olistica del ruolodel dispositivo rispetto a visione relativamente chiusa a quale norma è conforme per esempio alla crittografia.
Gli standard possono essere utilizzati per lo sviluppo di specifiche tecniche in un contesto specifico di un tipo di prodotto e fornire un quadro per la valutazione della sicurezza dei prodotti. Tale concetto generale è presentato nella Figura 4.

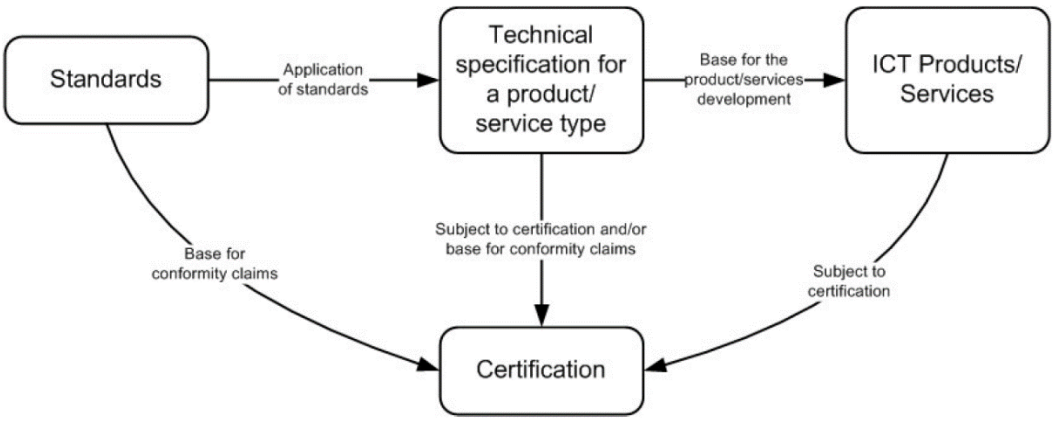

Figura 4 Relazione tra standard, certificazioni e servizi

1.24 Ruolo della standardizzazione

Nel contesto dei dispositivi IoT, un'ampia condivisione degli standard è che la loro funzione è quella di fornire **interoperabilità** delle "cose". È anche ampiamente condiviso che gli standard prevedono requisiti da soddisfare e non forniscono istruzioni su come implementare un requisito. Per gli standard di sicurezza queste dichiarazioni si applicano con la leggera modifica che molti standard di sicurezza, o più probabilmente le funzioni di sicurezza definite negli standard, assicurano l'interoperabilità delle "cose" quando soggette ad attacchi di parti ostili.Pertanto, gli standard possono riguardare la funzionalità (ad es. una crittografia algoritmo), applicazione di tale funzionalità (ad es. utilizzo di modalità di crittografia specifice) e uso contestuale di tale funzionalità (ad es. applicazione della crittografia per fornire protezione della riservatezza Servizi).
Anche per le entità coinvolte nella sicurezza crittografica, che devono interagire, dovranno essere condivise conoscenza e funzionalità che includeranno l'identificazione di chiavi e algoritmi. Quindi sicurezza le norme devono affrontare semplici interconnessioni meccaniche, significato semantico e sintattico condiviso, e gestione di attributi e organizzazioni per reagire alle trasgressioni di sicurezza in modo appropriato.

1.24.1 Interoperabilità organizzativa

Esiste una classe di standard di gestione organizzativa in materia di sicurezza che definisce i ruoli all'interno delle organizzazioni che cercano di imporre un "bisogno di sapere". Dal punto di vista della sicurezza quando due organizzazioni condividono i dati possono trasferire gli stessi in modo sicuro disponendo di un framework comune per la sicurezza delle comunicazioni (ComSec), ma lo scambio ComSec non può fare alcuna deduzione sul modo in cui i dati vengono trattati prima o dopo il trasferimento. Così si ritiene che la politica di sicurezza IT locale delle organizzazioni di invio e ricezione sia equivalente. In questo contesto questo la fiducia può essere rafforzata da misure esterne.

1.24.2 Interoperabilità sintattica

La sintassi deriva dalla parola greca che significa ordinamento e disposizione. La frase in lingua inglese la struttura dell'oggetto soggetto-verbo-oggetto è un semplice esempio di sintassi, e generalmente nella sintassi del linguaggio formale lo è l'insieme di regole che consente di formare un'espressione ben formulata da un insieme fondamentale di simboli. La sintassi della scienza informatica si riferisce alla struttura normativa dei dati. Per raggiungere l'interoperabilità sintattica ci deve essere una comprensione condivisa dell'insieme di simboli e dell'ordinamento dei simboli. In qualsiasi lingua il dizionario dei simboli è limitato, quindi in generale un verbo non deve essere frainteso come un nome per esempio (anche se ci sono esempi particolarmente eclatanti di uso improprio che sono diventati uso normale, ad es. l'uso di "medaglia" come verbo in cui il testo convenzionale "Ha vinto una medaglia" è stato abusato in "Ha medagliato").

1.24.3 Interoperabilità semantica

La sintassi non può trasmettere significato ed è qui che viene introdotta la semantica. La semantica deriva il significato da dichiarazioni sintatticamente corrette. La stessa comprensione semantica dipende sia dalla pragmatica che dal contesto. Esistono diversi modi per scambiare informazioni semantiche, sebbene il successo sia dipende dalla strutturazione dei dati per ottimizzare la disponibilità di contenuti semantici e il trasferimento di conoscenza contestuale (sebbene il trasferimento della pragmatica sia meno chiaro). Gli esempi più ovvi di contenitori semantici per informazioni sintatticamente corrette sono protocolli in base ai quali il protocollo (ad es. protocollo di autenticazione) fornisce contesto ai set di messaggi.
Questo può essere ulteriormente esteso usando il concetto di stato condiviso come mezzo per identificare il contesto e questo è spesso incorporato nel protocollo (ad es. il protocollo di autenticazione può passare attraverso stati che includono "Identificato", "Sfida emessa", "Risposta in sospeso "prima della finalizzazione sullo stato" Autenticato ").

1.24.4 Interoperabilità elettrica e meccanica

Molto semplicemente un dispositivo con un connettore di alimentazione che utilizza, ad esempio, una connessione di tipo IEC 60906-2 non può accettare l'alimentazione da qualsiasi altro che non sia un connettore di tipo IEC 60906-2. Allo stesso modo, ad esempio, una porta seriale la conformità a USB-Type-A non sarà in grado di connettersi con un cavo USB-Type-C. Oltre a semplice compatibilità meccanica è necessario garantire l'interoperabilità elettrica che copre tra altri il livello di tensione, livello di amperaggio, CC o CA, frequenza se CA, livelli di variazione e così via.

1.24.5 Interoperabilità delle comunicazioni radio

La comunicazione radio (wireless) richiede una conoscenza condivisa della banda di frequenza, della tecnica di modulazione, percentuale di simboli, potenza e così via. In generale, la comunicazione radio può essere definita come broadcast e inaffidabili. La natura dei media fisici richiede che i protocolli radio prevedano di massimizzare il collegamento affidabilità, il più delle volte ottenuto utilizzando varie forme di ForwardErrorCorrection nel Link Layer (livello 2 di lo stack OSI).

1.25 CONCLUSIONI

L’analisi preliminare condotta da ENISA sul panorama degli standard relativi all'IoT, indica che non esiste un divario significativo negli standard per introdurre l'IoT sicuro sul mercato. Questo però non significa chel'ecosistema IoT nel suo insieme al momento sia sicuro nonostante la presenza di un numero elevato di standard. In generale, esiste un gap identificabile nei processi attuativi. E’ necessario un approccio olistico verso la sicurezza dell'IoT ed elementi di tale approccio si possono trovare in una serie di standard, tuttavia per raggiungere un approccio globale che protegge l'intero ecosistema IoT è necessario lavorare ancora. La principale conclusione del documento ENISA è che gli standard sono essenziali ma non sufficienti per garantire libero accesso ai mercati. Nel caso particolare della sicurezza un gran numero di processi oltre chegli standard/tecnici devono essere messi in atto per garantire che qualsiasi dispositivo immesso sul mercato sia sicuro.

VALUTAZIONE DELLA TECNOLOGIA BLOCK-CHAIN NEL CONTESTO DI RIFERIMENTO

Una **blockchain** è un registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati **transazioni**) in modo sicuro, verificabile e permanente. Una volta scritti, i dati in un blocco non possono essere retroattivamente alterati senza che vengano modificati tutti i blocchi successivi ad esso e ciò, per la natura del protocollo e dello schema di validazione, necessiterebbe del consenso della maggioranza della rete. La blockchain, quindi, è rappresentabile come una lista, in continua crescita, di "blocchi" collegati tra loro e resi sicuri mediante l'uso della crittografia. Ad un blocco possono essere associate una o più transazioni e ogni blocco, inoltre, contiene un puntatore hash al blocco precedente e una marca temporale.
La natura distribuita e il modello cooperativo rendono robusto e sicuro il processo di validazione, ma presentano tempi non trascurabili, dovuti in gran parte al processo di validazione dei blocchi e alla sincronizzazione della rete. L'autenticazione avviene tramite collaborazione di massa ed è alimentata da interessi collettivi. L'utilizzo di questa tecnologia consente anche di superare il problema dell'infinita riproducibilità di un bene digitale e dalla doppia spesa, senza l'utilizzo di un server centrale o di una autorità.
La blockchain decentrata sfrutta il passaggio di messaggi ad hoc e un networking distribuito per fare in modo di memorizzare i dati su tutta la sua rete ed evitare di avere un single point of failure in modo che non esista una centralizzazione che i cracker potrebbero sfruttare per abbattere l'intero sistema. Tra i metodi di sicurezza della blockchain c'è anche la **crittografia a chiave pubblica**. La **chiave pubblica** è un indirizzo sulla blockchain. I token di valore inviati nella rete vengono registrati come appartenenti a questo indirizzo. Invece la chiave privata è come una password che permette al suo proprietario di accedere alle sue risorse digitali oppure di interagire con le varie funzionalità della blockchain. I dati salvati sulla blockchain sono considerati incorruttibili. Ogni nodo o **miner** nel sistema decentrato ha una copia della blockchain: difatti la qualità dei dati è mantenuta grazie a una massiva replicazione del database. Non esiste nessuna copia ufficiale centralizzata e nessun utente è più credibile di altri, tutti sono allo stesso livello di credenziali. I nodi miner, ovvero gli utenti, validano le nuove transazioni e le aggiungono al blocco che stanno costruendo dopo aver verificato l'intera blockchain. Una volta completato il blocco, lo trasmettono agli altri nodi della rete.
La blockchain usa differenti schemi di **timestamp** per serializzare le modifiche. La crescita della **blockchain** decentrata va di pari passo con il rischio della centralizzazione dei nodi, perché le risorse informatiche richieste per operare e gestire dati sempre più grandi diventano sempre più costose. Infatti molti nodi miner si aggregano in pool per riuscire a completare i blocchi e ottenere criptovalute.
Le transazioni sono raggruppate nei blocchi della blockchain e il numero di transazioni all'interno di ognuno di questi blocchi varia in base alla dimensione della transazione stessa. Invece, la dimensione della transazione varia in base al numero di input e di output della stessa. Un blocco è composto da due parti principali: **header e body**. Le transazioni sono racchiuse nel bodi del blocco e nell'header sono presenti **sette campi di gestione** del blocco stesso.
I campi nell'header del blocco sono questi:

Versione	02000000
Hash del blocco precedente (PrevHash)	E87C17C45768w7e1643fsd5481sd3f4131df681
Merkle root	697we168t4v1a4rv3v1e3r43c4er14ca8c4168a
Timestamp	358b0553
Bits	535f0119
Nonce	48750933
Numero di transazione	64

Il campo **Versione** dipende dalla versione del software utilizzato, il campo **PrevHash** è un hash di 256 bit che serve per fare riferimento al precedente blocco della catena. Il **MerkeRoot** è l'hash di tutti gli hash di tutte le transazioni nel blocco, il campo **timestamp** rappresenta il **timestamp** dell'ultima transazione con un algoritmo conosciuto come **Unix Hex Timestamp** da non confondere con il **timestamp Unix** che esprime bit numerici in secondi dal 1970-01-01T 00:00 UTC. Il campo **bits** rappresenta il corrente valore **target**: l'hash sha-256 dell'header di un blocco deve essere minore o al massimo uguale al corrente valore di target per essere accettato dalla rete. Il campo **nonce** è un valore di 8 byte che viene aggiunto al blocco in modo che l'output della funzione hash vari facendo in modo che risulti inferiore al valore **target**, il valore viene ricalcolato finché l'hash del blocco non contiene il numero richiesto di zeri principali e infine il campo **Numero di Transazione** che identifica il numero della transazione.
La combinazione della blockchain e dell'IoT può portare a tanti vantaggi [26]:

1. ***Una gestione decentralizzata della rete***: questa può essere ottenuta attraverso la registrazione di tutti gli eventi della rete come transazioni della blockchain. In questo modo, tutti gli eventi saranno verificati senza la necessità di una terza parte fidata.
2. ***Interoperabilità sui dispositivi IoT, sistemi IoT, settori industriali e SP:*** attraverso la blockchain si può costruire una rete con accesso uniforme per diversi sistemi fornendo una autenticazione unificata, meccanismi di autorizzazione e sistemi di fatturazione.
3. ***Tracciabilità e affidabilità dei dati IoT:*** un timestamp (historic timestamp) in ciascun blocco salvato nella blockchain assicurerà come conseguenza la tracciabilità dei dati. Con algoritmi di crittografia asimmetrici intrinseci, firme digitali e funzioni hash nelle blockchain, viene rafforzata l'integrità e la confidenzialità dei dati IoT. In questo modo garantire la sicurezza e la privacy e la sicurezza dei dati del massive IoT potrà essere realtà.

La blockchain ha i seguenti vantaggi tecnologici:

1. **Immutabilità**: i dati transazionali nella blockchain sono immutabili una volta che vengono registrati dal momento che ogni blocco è collegato con gli altri attraverso un valore hash. La possibilità di rompere l'intera catena e modificare il contenuto dei blocchi è molto molto molto limitata.
2. **Decentralizzazione**: la blockchain applica i meccanismi del consenso per gestire e mantenere il ledger distribuito senza la necessità di una entità centrale o una terza parte. I blocchi sono replicati e condivisi per tutta l'intera rete blockchain, in tal modo si evita il rischio del singolo punto di fallimento, rafforzando la persistenza dei dati e la sicurezza, fornendo inoltre flessibilità.
3. **Trasparenza**: tutti i partecipanti della blockchain hanno uguali diritti e possono accedere alle informazioni di tutte le transazioni della blockchain.
4. **Security and Privacy**: l'adozione della crittografia asimmetrica, l'intrinseca caratteristica dell'immutabilità dei dati, il meccanismo del consenso e l'anonimato rende la blockchain sicura, affidabile e con la proprietà di privacy.

Tutti i vantaggi offerti dalla blockchain possono essere impiegati nel campo delle smart grid per poter rafforzare e migliorare il processo del "two-way flow of power" ossia il flusso di energia a due vie: dal consumatore al fornitore e viceversa. La natura decentralizzata della blockchain si adatta perfettamente all'altrettanta natura decentralizzata della distribuzione dell'energia propria delle smart grid.
Per esempio, in [27] si propone la blockchain come tool per gestire le transazioni energetiche all'interno della smart grid. Le transazioni vengono eseguite con gli smart contract, e la rete agisce come un verificatore di transazioni. La blockchain, come visto prima, fornisce immutabilità delle transazioni, il che assicura che ciascuna transazione tra generatori e consumatori sarà sempre eseguita correttamente. Inoltre, fornisce immutabilità costruendo una vera e propria storia delle transazioni, che può essere utilizzata per effettuare operazioni di monitoraggio oppure risolvere dispute riguardanti una specifica transazione.

In [28] viene proposto l'uso della blockchain per poter rendere sicuro l'intero sistema decentralizzato "smart grid" per proteggerlo da eventuali minacce e vulnerabilità cibernetiche. La struttura a blocchi sopra descritta, la lista dei blocchi, il

puntatore al blocco precedente, il timestamp del blocco e la totale assenza dell'intervento umano rendono l'intero sistema più robusto ed è impossibile alterare le transazioni energetiche coinvolte nel sistema. Le applicazioni possono eseguire in una modalità unificata e trasparente con la blockchain, senza un controller centralizzato, quindi riuscendo ad ottenere gli stessi risultati che si otterrebbero aggiungendo, però, una terza autorità fidata. Inoltre, grazie alla natura centralizzata della blockchain le transazioni possono essere naturalmente eseguite in una modalità peer-to-peer tra consumatori e produttori e viceversa.

In [29] si progetta un meccanismo più specifico che fa sempre leva sulle proprietà cardine della blockchain. In particolare, viene proposto un sistema per rendere sicura l'infrastruttura della smart grid. L'idea riguarda una soluzione blockchain-based che rende immune da cyber attacchi di injection di fake data (FDI). Nel cyber attacco FDI, gli smart meter potrebbero essere controllati dagli attaccanti che potrebbero inviare letture fasulle per sovvertire completamente il funzionamento dell'intera smart grid.

SICUREZZA NELLO STANDARD IEC-61850

IEC 62351

Lo standard IEC 62351, dal titolo: "Gestione dei sistemi di alimentazione e scambio di informazioni associato - Sicurezza dei dati e delle comunicazioni", è uno standard pubblicato dalla IEC che mira a migliorare la sicurezza informatica nel contesto delle operazioni di controllo dei sistemi di alimentazione [4]. Gli standard precedenti, incluso IEC 61850, non includono le funzionalità di sicurezza informatica nel loro ambito, motivando la necessità di questo nuovo standard. È diviso in più parti. Le parti da 3 a 6 si concentrano sull'estensione dei protocolli di comunicazione IEC TC 57 esistenti, come IEC 61850 e IEC 60870-5, per migliorare la sicurezza delle comunicazioni. La parte 7 dello standard introduce Network Security Management (NSM) per fornire sicurezza end-to-end. Le parti rimanenti coprono altri aspetti della sicurezza informatica nel contesto dei sistemi di alimentazione, come la gestione delle chiavi, l'architettura della sicurezza, ecc.

Poiché il nostro focus è su IEC 62351-7 nella sottostazione IEC 61850, discutiamo brevemente ogni parte che è rilevante per essa:
Parte 1, in quanto tratta degli attacchi informatici e di quelli che lo standard dovrebbe affrontare;
Parte 3, in quanto mira a proteggere TCP, utilizzato da Manufacturing Message Specification (MMS) e altri protocolli applicativi;
Parte 4, in quanto protegge MMS;
Parte 6, in quanto protegge i protocolli di IEC 61850, ovvero GOOSE, SV e IEC 61850 MMS.
Le parti che non discutiamo includono la parte 2 (il glossario), la parte 5 (relativa ai protocolli IEC 60870-5e non IEC 61850) e parti 8 e successive che discutono altre misure di sicurezza.
Di seguito viene fornita una breve descrizione di ogni sua differente parte [15].
IEC 62351-1: la prima parte contiene una panoramica generale dello standard IEC 62351, delineando l'obiettivo dello standard, oltre a introdurre brevemente i diversi capitoli. Fornisce inoltre informazioni generali sulla sicurezza, un elenco delle minacce alla sicurezza (sia involontarie che intenzionali, ad esempio guasti alle apparecchiature, hacker informatici, ecc.), Nonché una panoramica generale delle possibili contromisure di sicurezza. La parte descrive anche brevemente concetti quali valutazione del rischio, gestione delle chiavi e processi di sicurezza, tra le altre cose.
IEC 62351-2: la seconda parte dello standard IEC 62351 è un glossario di termini, che spiega termini come controllo degli accessi, sicurezza dei dati, ecc.
IEC 62351-3: la terza parte della IEC 62351 riguarda la sicurezza dei protocolli basati su TCP / IP utilizzati per i sistemi di automazione nel dominio della distribuzione elettrica. In particolare, prescrive l'uso di TransportLayer Security (TLS) con certificati X.509 per i protocolli basati su TCP / IP. Lo scopo è garantire l'autenticità e l'integrità dei dati sul livello di trasporto e, facoltativamente, anche la riservatezza utilizzando i meccanismi di crittografia di TLS. L'utilizzo di TLS contrasta anche minacce come attacchi man-in-the-middle e attacchi replay. Questa parte dello standard richiede anche l'autenticazione reciproca tramite certificati (cioè, client e server presentano ciascuno un certificato) e prescrive gli algoritmi e alcune lunghezze minime delle chiavi da utilizzare, nonché come gestire la revoca dei certificati.
IEC 62351-4: questa parte dello standard IEC 62351 riguarda la sicurezza per profili come la Manufacturing Message Specification (MMS) (International Organization for Standardization (ISO) (2014)), che viene utilizzata in altri standard IEC (ad esempio, IEC 61850- 8-1 e IEC 60870-6). In particolare, la parte fornisce raccomandazioni per il profilo A e per il profilo T basato su TCP / IP. Per il profilo A, IEC 62351-4 descrive l'uso dei certificati X.509 per autenticare le applicazioni, mentre per il profilo T TCP lo standard descrive come utilizzare TLS come livello tra TCP e il servizio di trasporto ISO (Rose e Cass (1987)) utilizzando una porta TCP diversa per connessioni sicure. Ulteriori definite sono le suite di crittografia TLS che devono (o dovrebbero) essere supportate.
IEC 62351-5: la quinta parte dello standard IEC 62351 descrive la sicurezza per i protocolli relativi a IEC 60870-5 e come DNP-3 (IEEE StandardsAssociation (2012)). Questi protocolli sono basati sui messaggi e pertanto l'autenticazione deve essere eseguita in base al messaggio. Inoltre, qualsiasi meccanismo di sicurezza deve tenere conto della potenza di elaborazione

spesso limitata disponibile nei dispositivi interessati. Poiché le chiavi utilizzate per l'autenticazione e/o la crittografia dovrebbero essere cambiate regolarmente, questa parte propone anche meccanismi che consentono di aggiornare le chiavi in un dispositivo da remoto.

IEC 62351-6: la parte 6 dello standard IEC 62351 riguarda la sicurezza per i protocolli descritti nel relativo standard IEC 61850 (International ElectrotechnicalCommission (IEC). Per i protocolli in IEC 61850 che utilizzano TCP/IP e MMS, devono essere applicate le disposizioni descritte in IEC 62351-4. Inoltre, questa parte propone un'estensione alle PDU IEC 61850 GOOSE e SMV (unità di dati del protocollo), aggiungendo un campo alla PDU contenente informazioni rilevanti per la sicurezza. L'estensione ha lo scopo di autenticare una PDU contenendo un hash firmato della PDU. Questa parte dello standard aggiunge anche estensioni al SubstationConfiguration Language (SCL) (International ElectrotechnicalCommission (IEC)) che consentono di includere le definizioni dei certificati nella configurazione.

IEC 62351-7: l'infrastruttura dei sistemi di alimentazione fa un uso massiccio di sistemi di informazione interconnessi per gestire le operazioni. Anche questa infrastruttura dei sistemi informativi deve essere gestita in modo sicuro, che viene eseguita utilizzando il protocollo SNMP (Simple Network Management Protocol). La parte 7 dello standard IEC 62351 descrive i modelli di oggetti dati da utilizzare specifici per i sistemi di alimentazione.

IEC 62351-8: la parte 8 dello standard IEC 62351 definisce il controllo degli accessi basato sui ruoli a livello di sistema per l'infrastruttura dei sistemi di alimentazione. Affronta diverse modalità di accesso, come l'accesso diretto e remoto, nonché l'accesso da parte di utenti umani e l'accesso automatizzato da parte di agenti informatici. Per trasportare i ruoli, questa parte propone tre diversi formati per i token di accesso, vale a dire, certificati ID X.509 con estensioni, certificati di attributo X.509 e token software. Inoltre, lo standard definisce alcuni diritti e ruoli obbligatori.

IEC 62351-9: questa parte dello standard non è stata ancora rilasciata, ma intende indirizzare la gestione dei certificati e/o delle chiavi.

IEC 62351-10: la parte 10 dello standard IEC 62351 fornisce linee guida generali per l'architettura di sicurezza dei sistemi di alimentazione. Ciò include una panoramica dei controlli di sicurezza che possono essere applicati nei sistemi di alimentazione, nonché consigli sull'architettura del sistema su come strutturare l'infrastruttura di comunicazione dei sistemi di alimentazione.

IEC 62351-1: Introduzione allo standard di sicurezza informatica

IEC 62351-1 fornisce una panoramica del resto dello standard e una discussione sugli attacchi inclusi nel suo ambito di applicazione.

Gli obiettivi della norma IEC 62351 possono essere sintetizzati come segue [4]:

- Autenticazione per le entità e le loro azioni;
- Riservatezza di messaggi e chiavi;
- Integrità rilevando manomissioni e impedendo la riproduzione o lo spoofing;
- Monitorare la disponibilità di dispositivi e reti e un "grado di rilevamento delle intrusioni" [4];
- Consenti dispositivi protetti e non protetti sulla stessa rete per compatibilità con le versioni precedenti;
- Politiche di gestione delle identità (descritte nella norma IEC 62351-8).

Come si può vedere da quanto sopra, IEC 62351 mira a soddisfare i requisiti di sicurezza menzionati nella Sezione 2.1. Sebbene sia auspicabile soddisfare tutti i requisiti, vengono presi in considerazione alcuni dei requisiti
più critico di altri nel contesto di una sottostazione IEC 61850. In generale, la disponibilità e l'integrità hanno la massima priorità, mentre la riservatezza ha la più bassa [30]. Questo perché l'obiettivo della sottostazione è garantire che la fornitura di energia sia continua e controllata. La disponibilità è essenziale per garantire che le utility possano sempre eseguire le operazioni di controllo richieste entro i limiti di tempo necessari per stabilizzare o ripristinare la griglia [30]. L'integrità, insieme all'autenticazione, autorizzazione e non ripudio, è necessaria per evitare che queste stesse operazioni di controllo vengano abusate da parti non autorizzate per interferire con l'erogazione di energia. Questo è esattamente ciò che è accaduto durante l'attacco informatico contro l'Ucraina nel dicembre 2015, in cui gli aggressori hanno operato in remoto CB per causare deliberatamente un'interruzione di corrente [13]. La riservatezza è considerata il meno importante dei requisiti. Ciò è esemplificato dalle specifiche della norma IEC 62351-6, in cui la crittografia è resa facoltativa anziché

obbligatoria a causa del suo possibile impatto negativo sulle prestazioni (e quindi sulla disponibilità) [31]. IEC 62351-1 afferma che la crittografia "non è considerata così importante" nel contesto dei protocolli GOOSE e SV. Afferma inoltre che, in generale, l'autenticazione dei comandi ha una priorità più alta rispetto a nascondere i dati [4]. Ciononostante, la riservatezza è ancora essenziale per impedire agli aggressori di raccogliere conoscenze sul layout della sottostazione e per proteggere le credenziali delle informazioni utilizzate dagli operatori, poiché questo tipo di conoscenza può essere utilizzato in ulteriori attacchi una volta ottenuti.

IEC 62351-3: Sicurezza per TCP mediante la sicurezza del livello di trasporto

Lo standard IEC 62351-3 è denominato "Profili incluso TCP / IP" e si applica a qualsiasi protocollo che si basi sul protocollo TCP per il suo livello di trasporto [32]. È destinato a essere indicato da altri standard (con un buon esempio che è IEC 62351-4, relativo agli MMS). Questa parte della IEC 62351 mira a fornire integrità, autenticazione a livello di messaggio e riservatezza utilizzando il protocollo TransportLayer Security (TLS), che già vede un uso diffuso nelle reti IT. Inoltre, tiene conto del fatto che TCP viene utilizzato in modo diverso negli ambienti di telecontrollo. Le connessioni tendono ad essere molto più lunghe (o addirittura permanenti) rispetto ad altri contesti, questo influisce sulla scadenza e la revoca del certificato [32]. Poiché TLS funziona solo quando entrambe le parti che si scambiano messaggi concordano sulla stessa suite di crittografia, lo standard specifica alcune suite di crittografia TLS che dovrebbero essere supportate da tutti i dispositivi conformi per garantire l'interoperabilità tra di loro.

IEC 62351-4: estensioni di sicurezza per profilo T e profilo A MMS

IEC 62351-4 ha il titolo "Profili inclusi MMS" e mira a fornire sicurezza per il protocollo MMS [33]. Ciò include qualsiasi protocollo che utilizzi l'MMS come modello. MMS viene utilizzato per derivare il protocollo MMS IEC 61850, rendendo lo IEC 62351-4 applicabile al contesto della sottostazione. Lo standard IEC 62351-4 fornisce la sicurezza a due livelli di rete: il profilo T per il livello di trasporto e il profilo A per il livello dell'applicazione.

T-Profile

Per il T-Profile su TCP, si fa riferimento a IEC 62351-3. Lo standard IEC 62351-4 specifica i parametri da utilizzare con TLS, vale a dire le porte di rete, i pacchetti di crittografia, come gestire la revoca dei certificati e così via [33].

A-Profile

Per l'A-Profile, lo standard IEC 62351-4 specifica come autenticare le entità durante la creazione di una associazione iniziale. Nello specifico, afferma di aggiungere campi nelle PDU utilizzate per l'autenticazione ACSE (Association Control Service Element): AARQ (richiesta) e AARE (risposta). Quando si crea un'associazione, il client MMS deve includere tre campi in AARQ e AARE [33]:

(1) SignatureCertificate per portare il certificato X.509 utilizzato per verificare il valore SignedValue;
(2) SignedValue per portare una firma digitale del campo ora;
(3) tempo per rappresentare l'ora di creazione della richiesta.

Il destinatario dell'AARQ o dell'AARE deve quindi verificare che SignedValue corrisponda a una firma valida utilizzando i campi SignatureCertificate e time. Il campo dell'ora deve anche avere una differenza inferiore a 10 minuti rispetto all'ora locale del destinatario. La firma deve essere una che l'abbonato non ha mai visto prima. La PDU viene accettata solo se tutte queste condizioni sono vere. Altrimenti, viene emesso un P-ABORT [33]. Ciò non consente agli intrusi di creare una nuova associazione se non possono produrre il SignedValue necessario.

Al di fuori della creazione di un'associazione, non ci sono altre PDU che utilizzano questo meccanismo di autenticazione [33]. Questo è presumibilmente perché il T-Profile dovrebbe già essere protetto da TLS, fornendo una difesa contro un gran numero di attacchi informatici.

IEC 62351-6: estensioni di sicurezza per GOOSE e SV

Lo standard IEC 62351-6, intitolata "Sicurezza per IEC 61850", si concentra sui protocolli presenti in tale standard, vale a dire i protocolli IEC 61850 MMS, GOOSE e SV [31]. L'unica nuova aggiunta di sicurezza all'MMS IEC 61850 fornita da IEC 62351-6 è una suite di crittografia aggiuntiva per TLS. Per il resto, fa riferimento alla IEC 62351-4, che fornisce la sicurezza per l'MMS in generale [31]. Per questo motivo, non includiamo esplicitamente MMS IEC 61850 nell'ambito di IEC 62351-6. Ci concentriamo invece sulle sue aggiunte di sicurezza per GOOSE e SV. Il contributo principale della IEC 62351-6 alla sicurezza

di GOOSE e SV è duplice. Il primo è l'aggiunta di un nuovo campo denominato AuthenticationValue, utilizzato per verificare l'integrità e la crittografia AES-128 opzionale alle PDU GOOSE e SV. La seconda è la modifica dei protocolli GOOSE e SV nel tentativo di contrastare gli attacchi di replay [31]. Da notare che lo standard non impone l'uso della crittografia in tutti i pacchetti a causa di problemi di prestazioni che si verificano in determinati contesti, ma consiglia di utilizzare la crittografia ogni volta che non causa problemi. Usandolo, il contenuto dei messaggi viene nascosto agli aggressori passivi che ascoltano il traffico, il che previene il furto di dati e rende più difficili da eseguire alcuni altri attacchi [31].
IEC 62351-6 introduce un totale di due nuovi campi per GOOSE e SV PDU:
(1) Valore di autenticazione per trasportare la firma digitale RSA: l'input è un hash SHA-256 dei contenuti della PDU;
(2) timestamp per rappresentare l'ora di creazione della PDU (utilizzato solo per SV).
I publisher devono aggiungere un valore AuthenticationValue valido a ogni PDU. I subscriberpossono quindi verificare AuthenticationValue per confermare la legittimità della PDU convalidando la firma, poiché solo il vero publisher conosce la chiave privata richiesta per produrla. Con questa modifica, gli aggressori non possono più falsificare o modificare i pacchetti poiché entrambi richiedono la produzione di una firma valida. Il nuovo campo timestamp per SV PDU fa parte delle misure introdotte per proteggere dagli attacchi di replay.

1.26 Limitazioni alla sicurezza dello standard

Le preoccupazioni per le vulnerabilità della sicurezza continuano ad evolversi con la crescente complessità e inter-connettività delle reti intelligenti come sistemi cyber-fisici. Pertanto, un'analisi approfondita delle potenziali violazioni della sicurezza nei protocolli di comunicazione all'interno dei componenti della smartgrid è necessaria per pianificare una nuova infrastruttura cyberphysicalsmartgrid per garantire produzione, trasmissione e distribuzione di energia sicura e affidabile [34]. Le fonti di attacchi alle sottostazioni possono essere correlate a:

- Una persona interna con accesso alla rete IEC 61850 che può infettare il sistema con malware intenzionalmente o involontariamente utilizzando dispositivi infetti.
- Una catena di fornitura in cui i dispositivi IEC 61850 possono essere infettati da malware durante i processi di produzione o installazione.

Simile alla maggior parte degli standard comuni di tecnologia di comunicazione attualmente adottati nelle reti elettriche, IEC 61850 è suscettibile a molti diversi tipi di attacchi, inclusi attacchi denial of service, password cracking e intercettazione.

1) **Attacchi Denial of Service**: un attacco Denial of Service si verifica quando l'autore dell'attacco tenta di impedire agli utenti o alle macchine autorizzati di accedere a un servizio. Un modo per farlo è interrompere o sfruttare i servizi di uno IED. Per interrompere il funzionamento di uno IED, l'autore dell'attacco trasmette un codice dannoso allo IED mirato. Questo codice scrive in modo casuale dati di dimensioni eccessive per causare un overflow del buffer. Se questo codice fosse progettato per eseguire un'azione specifica, potrebbe anche causare una modifica non autorizzata dei dati. Questo tipo di attacco è possibile a causa di un controllo del limite insufficiente. Se l'aggressore sceglie di sfruttare i servizi comuni su un IED, può farlo aprendo più sessioni sui servizi FTP (File Transfer Protocol) o Telnet e mantenendoli inattivi tutto il tempo.
Il GOOSE poisoning è un'altra forma di attacchi di negazione del servizio. Nella messaggistica GOOSE, il mittente dei messaggi GOOSE è chiamato publisher e il destinatario dei messaggi GOOSE è chiamato subscriber. Il subscriberè solitamente un IED. Ogni messaggio GOOSE ha un campo di stato e numero di sequenza (stNum, sqNum). Quando si verifica un evento, l'IED inizia a trasmettere un messaggio con un nuovo stNum. Il messaggio viene ripetuto con un ritardo di tempo variabile. Ogni messaggio ripetuto ha un sqNum incrementato. Al fine di prevenire attacchi replay, il subscriber scarta qualsiasi messaggio avente uno stNum minore o uguale al messaggio precedente fino a quando non si verifica un rollover. In [35] GOOSE poisoning è analizzato in cui l'attaccante tenta di convincere il subscriber ad accettare messaggi con uno stNum più alto di quelli inviati dal publisher. Ciò fa sì che tutti i messaggi GOOSE dal publisher autentico siano considerati obsoleti dai subscriber. I subscriber ora accetteranno ed elaboreranno solo i messaggi GOOSE inviati dall'aggressore. Esistono tre forme di attacchi di GOOSE poisoning: attacchi con numero di stato elevato, attacchi di inondazione ad alta frequenza e attacchi semantici. Negli attacchi con numero di stato elevato, l'attaccante invia in multicast un frame GOOSE contraffatto con un numero di stato molto alto [35]. Questo modello di attacco è rappresentato in Figura 5.

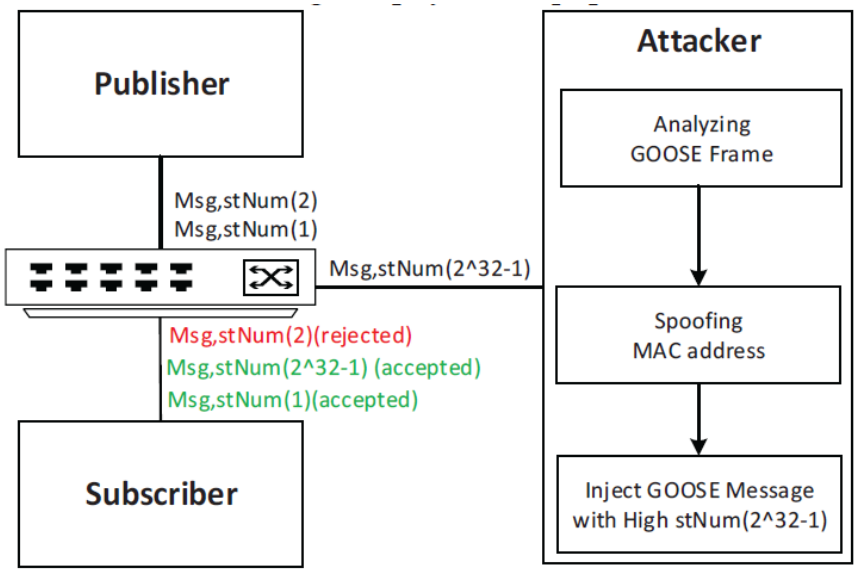

Figura 5. Attacco di GOOSE Poisoning

Vale la pena notare che un tale attacco mira solo a impedire a qualsiasi comando di controllo di raggiungere l'IED compromesso. Ciò impedirà all'IED di rispondere a eventi di rete critici come la protezione.
Oltre alla negazione del servizio, un utente malintenzionato potrebbe bloccare l'elaborazione di messaggi legittimi e iniettare messaggi con contenuto alterato. Come spiegato in [36] e illustrato in Figura 6, per ottenere ciò, un utente malintenzionato deve prima monitorare la rete per i messaggi GOOSE. Dopo aver ricevuto il messaggio, l'aggressore decodificherà il messaggio, ne modificherà il contenuto, codificherà di nuovo il messaggio e infine lo rimanderà indietro con un indirizzo MAC contraffatto e uno stNum più alto allo IED compromesso. È di fondamentale importanza rendersi conto dei rischi di questo attacco di rete. Un malintenzionato, ad esempio, può modificare lo stato di un determinato interruttore di circuito collegando così l'alimentazione a circuiti isolati. Ciò potrebbe comportare la perdita di vite umane e danni alle apparecchiature fisiche.

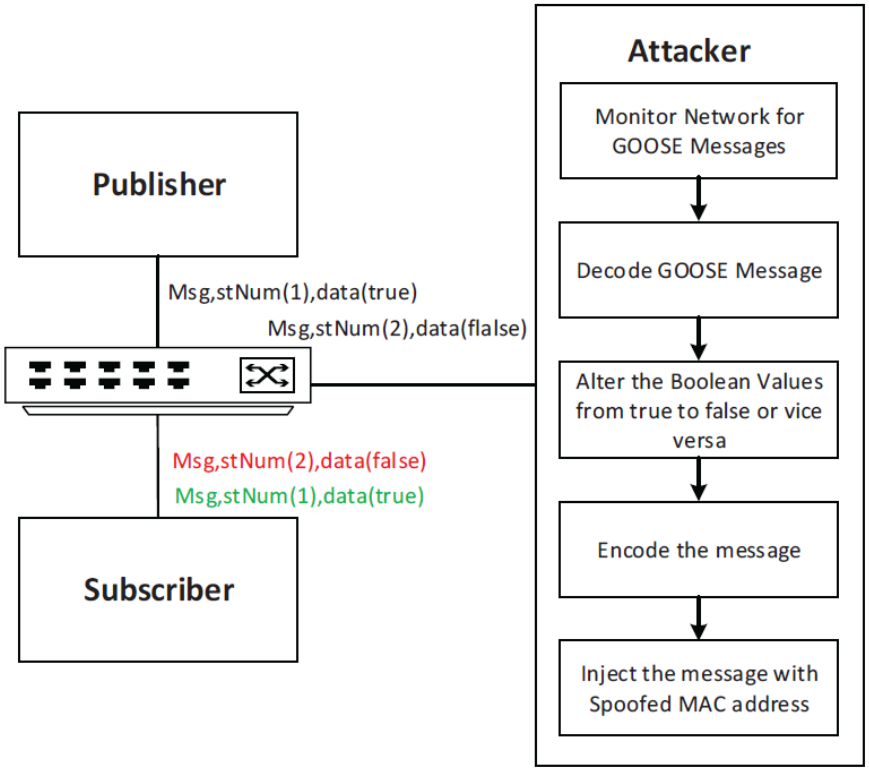

Figura 6. Attacco manipolazione messaggio GOOSE

Il Denial of Service potrebbe anche essere il risultato di un attacco flooding. Gli attacchi Flooding tentano di ritardare la consegna dei messaggi oltre il tasso di Flooding critico congestionando il canale ed esaurendo le risorse di calcolo dei nodi comunicanti. Il ritardo aumenta con l'aumentare del tasso di flooding. Secondo [37], il tasso di flooding critico è quando il ritardo supera il limite di 3 ms. Gli attacchi di flooding possono verificarsi sulla rete o sul livello dell'applicazione.
Il flooding a livello di rete può essere eseguito tramite flooding a un terzo nodo, broadcast flooding o flooding "unsolicited" al mittente o al destinatario. In caso di flooding al terzo nodo, l'attaccante inonda i pacchetti UDP a un nodo che non è il mittente o il destinatario. In questo caso, i messaggi inondati sono traffico in background, ma continuano a competere con messaggi legittimi nel canale wireless condiviso [37]. Questo tipo di flooding, tuttavia, non influisce tanto sul ritardo nell'Ethernet.
Nel broadcast flooding, l'attaccante trasmette pacchetti UDP all'intera rete. Ciò si traduce in un ritardo molto più elevato rispetto al caso di allagamento al terzo nodo. Questo perché i messaggi broadcast consumano più larghezza di banda di rete e risorse di canale rispetto ai messaggi unicast. Nel broadcast flooding, il tasso critico di flooding è stato rilevato da [37] per essere raggiunto a soli 4 pacchetti / secondo.
Il flooding "unsolicited" al mittente o al destinatario si verifica quando l'autore dell'attacco invia i pacchetti UDP direttamente a loro. Poiché non hanno alcun servizio per accettare questi tipi di pacchetti, questi pacchetti non raggiungeranno il livello dell'applicazione. In una rete Ethernet commutata, il ritardo aumenta all'aumentare della velocità di inondazione di questi attacchi. Questo perché i pacchetti inondati nel mittente o nel destinatario competono con pacchetti legittimi per il collegamento tra lo switch e il mittente o il destinatario, provocando rispettivamente ritardi aggiuntivi.
Il flooding a livello di applicazione determina ritardi di consegna dei messaggi più lunghi rispetto al flooding a livello di rete, soprattutto all'aumentare del tasso di flooding. Il flooding a livello di applicazione può essere eseguito tramite inondazione al mittente o al destinatario. In questo tipo di attacco, i pacchetti UDP vengono inviati al mittente o al destinatario. Questa

volta, il mittente o il destinatario può eseguire un servizio sulla porta in arrivo in modo che i pacchetti possano essere elaborati in qualche modo. Nel flooding "solicited", il ritardo aumenta con l'aumento del flooding rate a causa del fatto che i pacchetti inondati si contendono canali di rete con pacchetti legittimi. Il mittente e il destinatario impiegano tempo per elaborare i pacchetti inondati man mano che li ricevono, consumando cicli della CPU e provocando il ritardo. Secondo [37], la velocità di inondazione critica di questi attacchi si verifica a 128 pacchetti / secondo. Si è anche scoperto che il ritardo aumenta in modo significativo all'aumentare della velocità di inondazione in Ethernet commutata [37].

2) Attacchi di cracking delle password: gli attacchi di cracking delle password sono un tentativo da parte degli aggressori di ottenere l'accesso non autorizzato a un sistema o dispositivo, ad esempio un IED, individuando una password di utente autorizzato. Questo può essere fatto in due modi: attacco forza bruta o attacco dizionario. L'attacco di forza bruta si verifica quando tutte le possibili combinazioni di una password vengono tentate fino a quando non viene trovata quella desiderata. Questo potrebbe richiedere molto tempo. Gli attacchi al dizionario utilizzano solo parole costruite correttamente che potrebbero essere trovate in un dizionario per indovinare la password. Questi tipi di password sono più probabili e richiedono meno tempo per decifrarli.

3) Attacchi di packet sniffing (Eavesdropping): gli attacchi di packet sniffing, noti anche come intercettazioni, sono un tentativo di leggere e rubare i pacchetti che vengono trasmessi sulla rete. Questi attacchi devono essere lanciati dall'interno della LAN. Di conseguenza, l'aggressore deve compromettere una macchina all'interno della rete IEC 61850 o avere accesso fisico al dispositivo. Questi attacchi hanno successo perché i servizi a cui prendono di mira, inclusi FTP, HTTP e Telnet, non crittografano i messaggi e possono quindi essere letti facilmente una volta acquisiti. Ciò consente inoltre all'attaccante di eseguire attacchi man-in-the-middle. Un modo in cui un utente malintenzionato può intercettare è eseguire un avvelenamento della cache ARP (Address Resolution Protocol) (ARP Poinsoning) come mostrato nella Figura 7. ARP è un protocollo di comunicazione essenziale che converte gli indirizzi IP in indirizzi MAC. In questo attacco, un indirizzo IP falso e la mappatura dell'indirizzo MAC vengono installati su altri host nella LAN. Un indirizzo IP legittimo viene quindi associato a un indirizzo MAC errato, il che fa sì che lo switch inoltri tutti i pacchetti indirizzati a quell'IP alla macchina dell'aggressore.
Un altro modo è causare il Flooding della tabella CAM (Content Addressable Memory). Questo viene fatto riempiendo la tabella CAM dello switch con voci false. Quando essa è piena, i pacchetti indirizzati a un indirizzo MAC non presente verranno trasmessi a tutta la rete. Ciò consente all'attaccante di catturarli.
L'intercettazione può essere eseguita anche tramite Switch Port Stealing. L'attaccante prende di mira lo switch con gli indirizzi MAC dell'host di destinazione principale nell'intestazione. Quindi invia falsi frame allo switch facendo sì che lo switch modifichi la tabella CAM in modo che consenta il collegamento dell'indirizzo MAC all'interfaccia che conduce all'attaccante. Di conseguenza, tutti i pacchetti trasmessi all'indirizzo MAC di destinazione verranno inoltrati alla macchina dell'aggressore anziché al destinatario previsto.

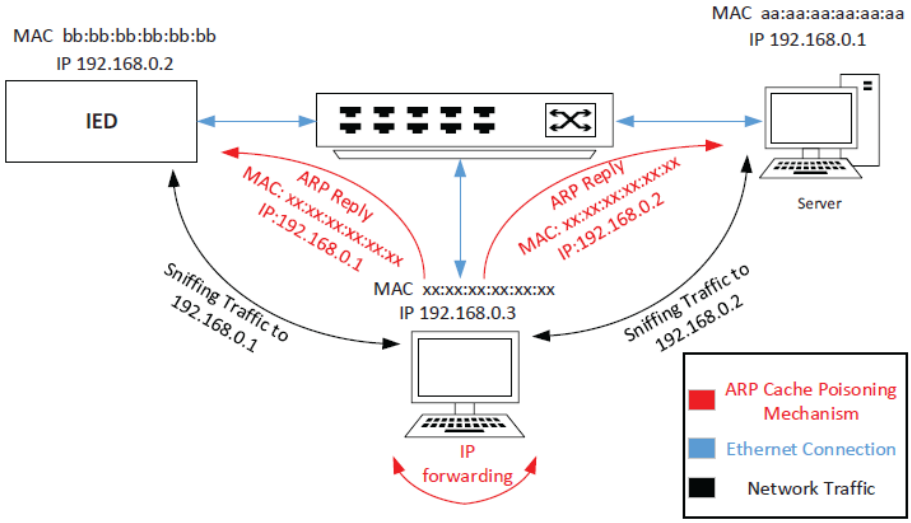

Figura 7. Attacco di ARP Cache Poisoning

1.27 Possibili attacchi al protocollo

Si studieranno i punti di forza e di debolezza del protocollo IEC-61850 dal punto di vista della sicurezza informatica. Ci occupiamo dell'elaborazione di attacchi informatici contro i protocolli IEC 61850. Esiste uno standard di interesse al riguardo: IEC 62351 [38][39][40].
Come introduzione alle minacce esistenti nel contesto della rete intelligente, la **National Electric Sector Cybersecurity Organization Resource** (**NESCOR**) documenta un gran numero di scenari di failure, alcuni dei quali possono essere causati da attacchi informatici. Il documento intende aiutare i servizi di pubblica utilità nella valutazione dei rischi [41]. Copre molti domini tra cui **Advanced Metering Infrastructure** (**AMI**), **Wide Area Monitoring, Protection and Control** (**WAMPAC**) e **Distribution Grid Management** (**DGM**).
Sebbene gli scenari **NESCOR** non siano specifici della sottostazione IEC 61850, molti di essi possono essere adattati per adattarsi al contesto della sottostazione. Tra questi scenari critici c'è lo scenario DGM.11, in cui un utente malintenzionato ottiene l'accesso al **sistema di gestione della distribuzione** (**DMS**) e invia comandi di sgancio sequenziali agli alimentatori critici e provoca un blackout.

Tabella 1 - Vulnerabilità e problematiche in IEC 61850 e IEC 62351

Vulnerability or issue	Addressed by IEC 62351	Proposed solution
Modification and replay on GOOSE/SV	✓(RSA)	Use IEC 62351-6, but then performance is a problem
MMS credential hijacking	✓(TLS)	None suggested
Force GOOSE subscriptions (PIM)		None suggested
MMS MitM attacks	✓(TLS)	Use IEC 62351-3 and -4, if performance is not too affected
Spoofing SV PDUs	✓(RSA)	Algorithm pinpoints faulty IED based on SV measurements
Spoofing GOOSE PDUs	✓(RSA)	Use cryptography, inspect packets, apply measures to prevent attacks on Ethernet
GOOSE poisoning attack	✓(RSA)	None suggested
Weak cipher suites for TLS in IEC 62351-3		Use secure cipher suites
Performance issues if using TLS in IEC 62351-3		Reconsider use of TLS , use cipher suite with no encryption
MMS MitM attacks (multi-hop scenario)		Use application-level secure session , use non-repudiation tokens
Performance issues if using RSA in IEC 62351-6		Use symmetric cryptography (HMACs)
GOOSE delay and replay attack		Track the last $stNum$ reset
Redirecting SV PDUs		Include sender MAC address to calculate $AuthenticationValue$
Weak SNTP security		Use same signature mechanisms as ones used for GOOSE/SV

Questo scenario potrebbe essere facilmente ricreato inviando comandi remoti alle sottostazioni IEC 61850 e vale la pena considerare. Inoltre, i passaggi eseguiti da un utente malintenzionato comuni a più scenari di errore sono documentati anche sotto forma di alberi di attacco. Questi sono molto istruttivi in quanto possono anche essere applicati agli attacchi contro la sottostazione IEC 61850. Mentre gli scenari di fallimento di NESCOR e gli alberi di attacco associati tendono a fornire

descrizioni di alto livello di possibili attacchi, essi costituiscono una risorsa molto utile per progettare ulteriori attacchi informatici.
Osservando le vulnerabilità all'interno della sottostazione digitale, esistono molti lavori. Molti dei contributi non prendono in considerazione le misure di sicurezza offerte da IEC 62351, e quindi gli attacchi proposti sono spesso già affrontati da tale standard. Altri si concentrano su vulnerabilità o problemi in IEC 61850 utilizzati in modo esplicito con IEC 62351. In questo modo, evidenziano alcune possibili lacune in IEC 62351 e offrono soluzioni per migliorare lo standard. Riassumiamo tutti questi precedenti contributi nella Tabella 1 e li discutiamo in dettaglio in questa sezione. Possiamo vedere che la maggior parte degli attacchi scoperti sulla sola IEC 61850 sono indirizzati dalla IEC 62351. Tuttavia, si sostiene spesso che l'applicazione di questi ultimi può portare a problemi di prestazioni.
Ci sono molti lavori in letteratura che trattano i tipi di attacchi informatici contro le sottostazioni dello standard IEC 61850. Molti dei lavori trattano sia gli attacchi contro le reti IT sia gli attacchi esclusivi delle sottostazioni IEC 61850. I primi sono classificati in tre categorie.

- Gli attacchi DoS possono essere eseguiti utilizzando richieste TCP SYN, FTP o causando overflow del buffer.
- Gli attacchi di password cracking possono essere eseguiti su servizi attivi in IED come FTP, HTTP (Hypertext Transfer Protocol) e Telnet.
- Gli attacchi di packet sniffing includono l'intossicazione della cache ARP (Address Resolution Protocol), il flooding della tabella CAM (Content Addressable Memory) o il furto di porte dello switch per consentire a un utente malintenzionato di visualizzare il traffico Ethernet destinato ad altri host.

Gli attacchi esclusivi della IEC 61850 prendono di mira i protocolli GOOSE e SV. In particolare, questi protocolli sono vulnerabili in molte situazioni che saranno descritte in seguito.

Attacchi conosciuti contro sottostazioni IEC 61850 senza l'uso di IEC 62351

In questa sezione viene fornita una buona panoramica degli attacchi noti contro le sottostazioni [42].
Wright et al. descrivono nel loro lavoro due possibili attacchi contro i protocolli IEC 61850 [43]. Questa ricerca esclude deliberatamente l'uso delle raccomandazioni della IEC 62351, in quanto ciò riflette il caso in cui i problemi di prestazione impongono che misure come la crittografia non possano essere utilizzate. Il primo attacco descritto è un attacco di dirottamento delle credenziali contro MMS. È dimostrato usando un automa. Per impostazione predefinita, MMS fornisce un meccanismo di controllo dell'accesso per stabilire sessioni per gli utenti autorizzati, ma non fornisce alcuna protezione per le sue PDU, lasciandole aperte allo sniffing e all'alterazione. L'attacco consiste nell'invio di una richiesta di accesso con credenziali errate contemporaneamente a una richiesta di accesso legittima da parte di un utente. Il server che riceve queste richieste risponde con un errore per la PDU dell'attaccante e un riconoscimento per l'utente. Tuttavia, l'attaccante può reindirizzare l'errore PDU destinato a lei e inoltrarlo all'utente. A quest'ultima è rimasta l'impressione che le sue credenziali siano sbagliate, e la sessione viene dirottata. Il secondo attacco è un attacco di amplificazione basato sul multicast Protocol Independent Multicast (PIM). GOOSE utilizza un modello editore-abbonato con Ethernet multicast [44].L'idea di questo attacco è di forzare LN extra a iscriversi inutilmente agli editori GOOSE. L'attaccante lo fa inviando un pacchetto PIM-flood o PIM-graft, a seconda del tipo di rete. Queste richieste indicano ai router di aggiungere un'interfaccia specifica (LN di destinazione) come destinatario di messaggi multicast, anche se non si prevede che li ricevano. Forzando ulteriori abbonamenti, l'attacco proposto inonda gli IED con traffico inutile che potrebbero non essere in grado di gestire. Dato che gli IED hanno risorse limitate e requisiti di prestazione rigorosi, questo può essere sufficiente per un attacco DoS riuscito.
Kang et al. investigare potenziali attacchi contro IEC 61850 MMS nel contesto di Distributed Energy Resource (DER) [45]. Questa ricerca si concentra sugli attacchi MitM contro i veri inverter fotovoltaici (PV) resi possibili dall'avvelenamento da ARP. In questo scenario, l'attaccante ha la capacità di annusare, falsificare, modificare e rilasciare PDU MMS. Gli attacchi eseguiti in questo lavoro sono esplicitamente associati a scenari di errore NESCOR [55] specifici specifici di DER. Utilizzando le funzionalità MitM, l'attaccante modifica le comunicazioni tra un client MMS e un server MMS falsificando una richiesta di scrittura MMS per modificare l'impostazione di limitazione della potenza dell'inverter FV su un valore falso. Per impedire alle PDU inviate in risposta alla PDU dannosa di raggiungere il client MMS legittimo, l'attaccante le rilascia. La modifica della limitazione di potenza fa sì che l'inverter FV passi in modalità standby anziché funzionare come previsto. Questo lavoro evidenzia i difetti di sicurezza degli MMS e fornisce un esempio di come sfruttare gli scenari NESCOR per elaborare attacchi informatici più dettagliati. Tuttavia, mentre si parla di IEC 62351-4 e di come migliora la sicurezza degli MMS, gli esperimenti

descritti in questo lavoro non applicano questa specifica. Sappiamo dallo standard [33] che farlo teoricamente previene gli attacchi MitM descritti.
La ricerca di Valdes et al. documenta due attacchi di iniezione di dati falsi contro il protocollo SV [46]. Il primo attacco ha lo scopo di inserire false PDU SV con l'obiettivo di indicare un errore quando non ce n'è. Ciò ha il potenziale per causare tempi di inattività non necessari. Il secondo funziona allo stesso modo, ma le false PDU nascondono invece i guasti effettivi. Nascondere i guasti ha la capacità di causare molti danni a causa del suo potenziale di causare un altro relè a monte per far scattare un CB e di conseguenza causare un'interruzione più grande del previsto. Entrambi gli attacchi sono buoni esempi di come un utente malintenzionato possa abusare delle PDU SV per causare danni fisici a una sottostazione. Il lavoro propone un approccio per rilevare questi attacchi usando IED collaborativi. Utilizzando le misurazioni reciproche e le leggi di Kirchhoff, possono individuare l'origine delle misurazioni non valide. L'approccio è testato in MATLAB e mostra che può individuare misurazioni e IED difettosi. Tale approccio è molto diverso dall'uso di NSM per il monitoraggio, come fatto in questa tesi, poiché si basa sull'utilizzo delle misurazioni stesse per rilevare attacchi informatici.
Hoyos et al. descrivere un possibile attacco di spoofing GOOSE nel loro contributo [47]. Questo attacco funziona annusando il traffico GOOSE esistente per individuare il valore corrente di stNum, sqNum e allData. Quindi, l'iniezione di una PDU GOOSE forgiata con un valore alterato per allData provoca un cambiamento di stato nel sistema. In particolare, l'attaccante modifica tutti i dati capovolgendo tutti i bit che rappresentano valori booleani. Oltre a modificare allData, lo stNum deve essere incrementato dal valore stNum visualizzato nel traffico sniffato e sqNum deve essere impostato su 0. Queste modifiche sono necessarie per garantire che la PDU dannosa sia trattata come una modifica di stato valida dal sottoscrittore. L'autore dell'attacco potrebbe non conoscere il significato dei bit in allData, poiché queste informazioni non si trovano nelle stesse PDU. Nonostante ciò, è probabile che l'attacco causi effetti indesiderati modificando lo stato di alcuni LN. Ad esempio, questo stato potrebbe essere se un CB è aperto o chiuso. Un'idea simile è descritta da Kush et al. nella loro ricerca presentando l'attacco avvelenamento GOOSE [48]. Viene eseguito allo stesso modo dell'attacco di spoofing, con una piccola modifica. In questo attacco e nelle sue varianti, l'attaccante aumenta lo stnum della PDU dannosa per renderlo superiore a quello attualmente utilizzato dall'editore legittimo e imposta sqNum su 0. Si noti che la PDU dannosa contiene lo stesso allData delle PDU legittime. In questo modo l'aggressore può dirottare completamente le comunicazioni GOOSE. L'attacco avvelenamento GOOSE sfrutta un comportamento unico di GOOSE in cui l'abbonato rilascia PDU con un campo stNum inferiore a quello che ha ricevuto per ultimo, come descritto in IEC 62351-6 [31] e discusso in precedenza nella Sezione 2.5.4. Ciò si traduce in un attacco DoS efficace quando vengono rilasciate le PDU del legittimo editore (e quindi i comandi critici). Questi attacchi mostrano in che modo una singola PDU GOOSE dannosa può avere un impatto significativo sul sistema. Tuttavia, entrambi si basano sulla mancanza di controllo dell'integrità nelle PDU GOOSE, rendendo probabile che entrambi possano essere affrontati con un metodo simile.

Valutazione della sicurezza dello standard IEC 62351

Schlegel et al. fornire una panoramica della IEC 62351 e delle parti di essa rilasciate fino al 2015, nonché una valutazione di ciascuna parte quando si considerano le conclusioni del lavoro esistente [49]. È una risorsa molto utile per conoscere inizialmente lo standard IEC 62351 e una lettura consigliata per chiunque sia interessato a questo argomento. Nel complesso, la valutazione conclude che IEC 62351 fa molto per fornire sicurezza ai sistemi di alimentazione. Evidenzia alcune lacune, molte delle quali vengono discusse in maggiore dettaglio più avanti in questa sezione. In generale, la valutazione rileva che i potenziali vettori di attacco derivano principalmente dal consentire configurazioni che consentono la compatibilità con le versioni precedenti ma riducono la sicurezza nel processo. Poiché è stata pubblicata nel 2015, questa valutazione non fornisce informazioni aggiornate su IEC 62351-7: 2017 o IEC 62351-9, in quanto non erano state pubblicate all'epoca

IEC 62351-3 and IEC 62351-4: Security for TCP and MMS

Poiché IEC 62351-4 mira a proteggere l'MMS e fa riferimento a IEC 62351-3 per proteggere il TCP, il livello di trasporto dell'MMS, queste due parti specifiche dello standard tendono a essere valutate simultaneamente in precedenti lavori sull'MMS sicuro.
Khaled et al. studiare gli effetti sulle prestazioni quando si applicano le specifiche dello IEC 62351-3 e dello IEC 62351-4 [50]. Questo lavoro fornisce un'analisi delle 10 suite di crittografia specificate dallo standard. Si consiglia di sostituirne alcuni per

includere l'uso di Diffie-HellmanEphemeral (DHE), RSA e Cipher Block Chaining (CBC), poiché forniscono ulteriori vantaggi in termini di sicurezza. Per valutare le prestazioni di queste suite di crittografia, gli esperimenti vengono condotti utilizzando l'implementazione open source OpenIEC61850 e i socket Secure Sockets Layer (SSL) di Java. Questi possono essere utilizzati per supportare le comunicazioni MMS su TLS secondo gli standard IEC. Vengono raccolte diverse metriche, tra cui latenza di handshake, latenza delle richieste, overhead della CPU e overhead della memoria che si verificano durante una richiesta Get MMS. La conclusione di questo lavoro è che TLS stesso aumenta la latenza del 75%, ma che le prestazioni complessive soddisfano comunque i requisiti per i tipi di messaggio IEC 61850 2, 3 e 5, che sono i tipi trasportati dagli MMS. Diverse suite di cifratura influenzano le prestazioni a vari livelli, ma soddisfano tutte i requisiti; quindi, si consiglia di utilizzare le suite proposte nello studio in quanto forniscono una sicurezza più affidabile. Questo è uno studio utile in quanto fornisce una valutazione delle suite di cifratura raccomandate in IEC 62351-4 e mostra che esiste il potenziale per consentire l'uso di suite di cifratura migliori. Tuttavia, non è chiaro se i risultati si applicano alle sottostazioni effettive. Gli autori includono una menzione che il modello utilizzato non rappresenta accuratamente un tale ambiente. Si presume anche che gli IED forniscano supporto hardware per AES come le CPU utilizzate nello studio.

Wright et al. forniscono un'analisi della IEC 62351-3 e sottolineano alcuni dei suoi difetti, principalmente riguardanti la gestione delle chiavi e l'uso di una crittografia più debole [51]. Lo standard specifica l'uso di TLS per i protocolli che utilizzano TCP. L'utilizzo di TLS necessario implica l'uso di certificati e tutto ciò che comportano, come il controllo dei certificati scaduti o revocati e le autorità di certificazione. Non si parla di come questi requisiti debbano essere implementati nelle reti reali, in quanto la gestione delle chiavi è invece lasciata alla IEC 62351-9, che all'epoca non era stata pubblicata. Questo lavoro spiega che, indipendentemente dal fatto che l'architettura di fiducia utilizzata si basi su elenchi di revoche di certificati (CRL) o sul protocollo OCSP (Online Certificate Status Protocol), esistono potenziali attacchi contro queste architetture di cui è necessario tenere conto. Inoltre, la IEC 62351-3 consente l'uso di suite di cifratura deboli per consentire la compatibilità con le versioni precedenti [32], come quelle che utilizzano Rivest Cipher 4 (RC4) o MD5. Secondo questa ricerca, utilizzando un attacco di downgrade, un utente malintenzionato potrebbe indurre un dispositivo a utilizzare una di quelle suite di crittografia deboli per eseguire attacchi noti contro di loro [51]. Questo lavoro di Wright et al. fa luce sui potenziali miglioramenti alla IEC 62351-3.

Chowdhury et al. tentano di implementare l'uso di TLS negli MMS come raccomandato dalla IEC 62351-4 in ambienti legacy [52]. Poiché quella parte dello standard fa riferimento alla IEC 62351-3 per proteggere il livello di trasporto [33], il loro lavoro valuta implicitamente anche la IEC 62351-3. Uno degli obiettivi di questa ricerca è misurare le prestazioni dell'MMS con TLS in sistemi integrati a poche risorse, come ci si aspetterebbe in un sistema di alimentazione. La preoccupazione è che TLS comporti operazioni costose per gestire certificati, sessioni, firme digitali e crittografia che potrebbero non essere fattibili per tali sistemi. L'implementazione di TLS in MMS in questo lavoro si basa su OpenSSL e sul sistema operativo in tempo reale (OS) VxWorks. I risultati indicano che l'utilizzo della crittografia in TLS aumenta l'utilizzo della memoria dell'85% e aumenta notevolmente la durata delle operazioni di lettura e scrittura. Le prestazioni migliorano se si utilizza una suite di crittografia che include solo MAC e rinuncia alla crittografia, considerata meno critica nel contesto del sistema di alimentazione. L'handshake SSL potrebbe richiedere fino a 3 secondi, il che potrebbe influire su alcune applicazioni. Questo lavoro fornisce informazioni sull'uso di TLS in un contesto di sistema di alimentazione più realistico e le potenziali ramificazioni sulle prestazioni quando si utilizza la crittografia. Non copre la sicurezza del **profilo A** come specificato da IEC 62351-4 [33] poiché è fuori dal suo campo di applicazione.

Fries et al. [53] e Ruland et al. [54] discute una debolezza nella sicurezza del **profilo A** MMS come specificato dalla IEC 62351-4. In situazioni tipiche, i profili di sicurezza **T-Profile** e **A-Profile** sono entrambi utilizzati per le comunicazioni MMS tra due dispositivi finali, fornendo autenticazione, riservatezza e integrità delle PDU [33]. Tuttavia, entrambi gli studi sottolineano che esistono diversi casi d'uso che coinvolgono una connessione multi-hop. In altre parole, le parti comunicanti non comunicano direttamente tra loro, ma richiedono invece un proxy intermedio per inoltrare il loro traffico [55].

SICUREZZA NEI PROTOCOLLI MACHINE-TO-MACHINE

Si effettuerà uno studio sulle potenziali minacce alla sicurezza dei protocolli M2M [56] con particolare riferimento al protocollo COAP [57] e all'MQTT [58].
I dispositivi IoT connessi tra loro sono caratterizzati da un numero significativo di potenziali superfici di attacco e di modelli di interazione, tutti da prendere in considerazione nel fornire protezione agli stessi.Il termine "accesso digitale" viene usato per distinguerlo da qualsiasi altra operazione che preveda l'interazione diretta con i dispositivi laddove è presente la sicurezza dell'accesso attraverso il controllo di accesso fisico (ad esempio, collocando il dispositivo in una stanza chiusa a chiave). Nonostante non sia possibile negare l'accesso fisico usando software e hardware, è però possibile adottare misure preventive per far sì che l'accesso fisico non conduca a intromissioni nel sistema. Esplorando i modelli di interazione vengono presi in esame il controllo dei dispositivi e i dati del dispositivo con lo stesso livello di attenzione. Per controllo dei dispositivi si intende qualsiasi informazione fornita a un dispositivo da qualsiasi parte, con l'obiettivo di cambiarne o influenzarne il comportamento nei riguardi del suo stesso stato o dello stato del relativo ambiente. Per dati del dispositivo si intende qualsiasi informazione emessa da un dispositivo a qualsiasi altra parte riguardo il suo stato e lo stato osservato del relativo ambiente. Allo scopo di ottimizzare le procedure consigliate di sicurezza, è consigliabile suddividere una tipica architettura IoT in svariati componenti/zone come parte dell'esercizio di modellazione delle minacce:

- Dispositivo
- Gateway sul campo
- Gateway cloud
- Servizi

Le zone rappresentano una segmentazione generica di una soluzione. Ogni zona spesso contiene dati e requisiti di autenticazione e autorizzazione specifici. Le zone possono anche essere usate per isolare i danni e limitare l'impatto delle zone di bassa attendibilità sulle zone di attendibilità superiore. Ciascuna zona, separata da un limite di attendibilità, rappresenta una transizione di dati/informazioni da un'origine a un'altra. Durante questa transizione, i dati potrebbero essere soggetti a spoofing, manomissione, ripudio, divulgazione di informazioni, rifiuto del servizio ed elevazione dei privilegi, semplificati dall'acronimo STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege).

Zona del dispositivo

L'ambiente del dispositivo è lo spazio fisico immediato che circonda il dispositivo, in cui l'accesso fisico e/o l'accesso digitale peer-to-peer via "rete locale" al dispositivo è fattibile. Si presuppone che una "rete locale" sia una rete distinta e isolata dalla rete Internet pubblica (ma potenzialmente connessa con bridging alla stessa) che includa tutte le tecnologie wireless a corto raggio per consentire la comunicazione peer-to-peer dei dispositivi. La stessa rete non include tecnologie di virtualizzazione delle reti che creino l'illusione di una rete locale, né reti di operatore pubblico che richiedano la comunicazione di una coppia qualsiasi di dispositivi in uno spazio di rete pubblica, qualora dovessero entrare in una relazione di comunicazione peer-to-peer.

Zona del gateway sul campo

Un gateway sul campo è un dispositivo o un software per server di uso generico che agisce come componente che abilita la comunicazione e, potenzialmente, come sistema di controllo dei dispositivi e hub di elaborazione dei dati dei dispositivi. La zona di un gateway sul campo include il gateway stesso e tutti i dispositivi collegati. Come suggerisce il nome, i gateway sul campo operano all'esterno della struttura di elaborazione dei dati, sono solitamente associati alla posizione, sono potenzialmente soggetti a intrusioni e hanno una limitata ridondanza operativa. Tutto ciò per affermare che un gateway sul campo è in genere un sistema che è possibile toccare e sabotare ben conoscendone la funzione.

Zona Gateway cloud

Un gateway cloud è un sistema che consente la comunicazione remota da e verso dispositivi o gateway sul campo da più siti diversi attraverso lo spazio di rete pubblico, in genere verso un controllo basato sul cloud e un sistema di analisi dei dati. In alcuni casi, un gateway cloud può immediatamente facilitare l'accesso ai dispositivi per scopi specifici dai terminali, ad esempio tablet o telefoni.

Zona Servizi
Un "servizio" è definito per questo contesto come qualsiasi componente software o modulo che si interfaccia con i dispositivi attraverso un gateway sul campo o cloud per la raccolta e l'analisi dei dati, nonché per il controllo e il comando.

1.28 SCENARIO DI RIFERIMENTO

Per quanto riguarda l'analisi della sicurezza si è proceduto alla realizzazione di uno scenario di riferimento sul quale basarci per effettuare uno studio sulle possibili minacce verso un sistema basato su CoAP o basato su MQTT. Dopo aver analizzato le possibili minacce si è provveduto ad un'attenta analisi, coadiuvata da esperimenti, sulle possibili contromisure da realizzare per far rispettare i requisiti di sicurezza su di un sistema come quello analizzato per questo studio.
Inizialmente si è proceduto a progettare una rete di comunicazione fisica composta principalmente da dispositivi IoT capaci sia di interagire con l'ambiente esterno e circostante tramite l'utilizzo di appositi sensori, sia di comunicare tra loro e con i gateway di riferimento, attraverso l'utilizzo di nuovi protocolli legati al contesto dell'Internet Of Things, ovvero CoAP (Constrained Application Protocol) e MQTT (Message Queue Telemetry Transport). Si è poi proceduto ad identificare le possibili fonti e tipologie di attacco, principalmente incentrate sullo sfruttamento di possibili vulnerabilità presenti all'interno dei protocolli e dei meccanismi di rete e di comunicazione utilizzati, che possano vertere e minacciare l'aspetto di confidenzialità, integrità e autenticazione dei dati che viaggiano e vengono elaborati dal sistema distribuito stesso. Successivamente si è voluto estendere ed evolvere la struttura stessa della rete IoT ottenuta tramite l'integrazione e l'utilizzo di meccanismi e metodologie di sicurezza, incentrate principalmente su elementi crittografici, sia pubblici che privati, capaci di aggiungere agli aspetti protocollari e di comunicazione del sistema stesso tutte quelle features di protezione e sicurezza non presenti nelle versioni iniziali con lo scopo di bloccare tutte le principali tipologie di attacco precedentemente introdotte.

Si è inizialmente messo sotto attacco il sistema stesso, nella sua versione basilare, attraverso l'utilizzo di tecniche e metodologie legate al contesto del "Network Penetration Testing". Tali tecniche sono state soprattutto basate sull'ausilio e lo sfruttamento delle vulnerabilità conosciute e appositamente presenti all'interno dei diversi protocolli utilizzati e dei meccanismi di comunicazione su cui la rete stessa si basa: più precisamente, in tutto ciò, si è cercato principalmente di mettere sottopressione e sotto minaccia, gli aspetti di confidenzialità, autenticazione e integrità, dei tre protocolli maggiormente utilizzati, ovvero CoAP, MQTT e HTTP, oltre a focalizzarsi anche sull'esecuzione di attacchi miranti ai dispositivi fisici interni della stessa "rete privata" di riferimento, come i router e i diversi servizi messi a disposizione.
Infine, si è posta maggiormente l'attenzione sull'estensione e l'evoluzione delle funzionalità e caratteristiche interne del nostro sistema di sensori, al fine di migliorare l'aspetto qualitativo e di sicurezza stesso della rete: per raggiungere tale scopo abbiamo aggiunto una serie di features, soprattutto a livello protocollare, miranti specialmente, a raggiungere quei margini di confidenzialità, integrità, autenticazione e anche disponibilità dei servizi non presenti nella versione base.
L'architettura utilizzata per lo studio delleì features di sicurezza è rappresentata nelle due figure sottostanti, Figura 8 e Figura 9. Esse mostrano lo scenario utilizzando i 2 principali protocolli M2M: l'MQTT (Message Queue Telemetry Transport) e il CoAP (Constrained Application Protocol).
L'analisi e le considerazioni sugli aspetti di sicurezza sono stati effettuati sia nel collegamento dispositivo di campo (sensore) verso il gateway, analizzando separatamente i due protocolli IoT, sia sul collegamento gateway verso internet e quindi verso un potenziale cloud dove si è supposto l'utilizzo di un classico protocollo HTTP.

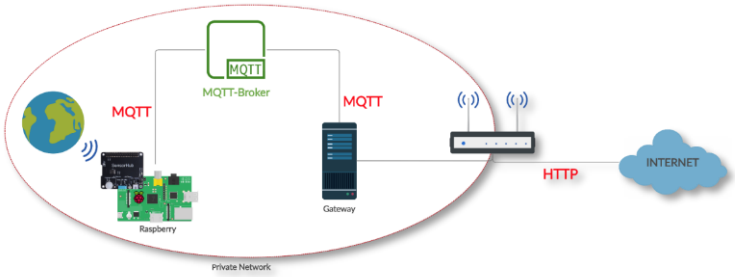

Figura 8. Rete IoT che usa il protocollo MQTT.

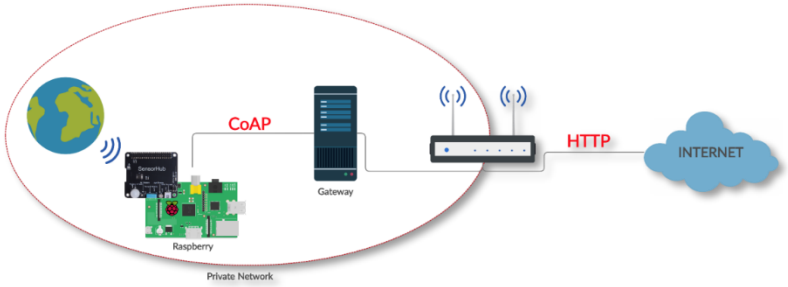

Figura 9. Rete IoT che usa il protocollo CoAP.

Prima però di analizzare nel dettaglio gli aspetti implementativi e di sviluppo di tali meccanismi comunicativi, è opportuno fare una piccola descrizione tecnica e precisa delle componenti hardware e fisiche, che definiscono la struttura interna del nostro tipo di dispositivo IoT utilizzato; come infatti già accennato in precedenza, il nostro componente IoT progettato risulta essere composto da due principali elementi di interesse: da una parte emerge la base funzionale del dispositivo, risultante rappresentata da un semplice Raspberry Pi 3 Model B+, mentre dall'altra parte, per quanto riguarda l'utilizzo del sensore esterno da integrare, si è optato per un DockerPi Sensor Hub Development Board.

1.29 TIPOLOGIE DI MINACCE E POSSIBILI VETTORI DI ATTACCO

Verranno di seguito analizzate quelle che sono le principali vulnerabilità e problematiche che un sistema come quello descritto può presentare evindenziandone i possibili vettori di attacco e le possibili minacce che possono essere eseguite sfruttate con lo scopo di intaccare la sicurezza per quanto concerne **confidenzialità**, **integrità**, **autenticazione** e **disponibilità** dei possibili servizi che possono essere messi a disposizione da parte dei dispositivi della rete. Tali forme di attacco sono strettamente e principalmente correlate al contesto della Network Security, quindi rivolte a sfruttare le principali vulnerabilità e falle di sicurezza presenti nella struttura architetturale ed esecutiva dei diversi protocolli comunicativi utilizzati, sia per ciò che riguarda i protocolli definiti e funzionali a livello applicativo, come CoAP, MQTT e HTTP sia per quelli legati ai livelli più bassi della pila protocollare [59],[60].
Prima di passare ad analizzare le principali forme e tipologie di attacchi eseguibili nei confronti del sistema, emerge il necessario bisogno nel definire, tramite assunzione, quello che è **il Threat Model** che va a descrivere la fisionomia generale del possibile attaccante della nostra rete:si assume in questo caso che l'attaccante riesca in qualche modo, in primis, ad ottenere accesso alla rete locale e privata (LAN) all'interno della quale risultano essere presenti sia il generico **Local Gateway**

che il **dispositivo IoT** per poi successivamente riuscire ad ottenere persino la possibilità, sfruttando i diversi protocolli sopra citati, di interagire con i diversi componenti presenti al suo interno, senza essere direttamente riconosciuto da essi come una possibile minaccia proveniente dall'esterno.

1.29.1 WiFi Hacking: Cracking WEP e WPA/WPA2 Network

Ricordando lo schema descritto nelle Figura 8 e Figura 9, si può notare come al suo interno, risulta essere presente un dispositivo di rete intermedio che funge da Access Point (punto di accesso) per la stessa rete privata (nel nostro caso di interesse, è rappresentato da un semplice router casalingo); ciò significa che, affinché l'attaccante riesca ad ottenere un accesso valido e verificato all'interno della LAN di riferimento, deve innanzitutto riuscire a farsi autenticare ed identificare dall'Access Point presente al suo interno: per fare ciò, si assume che l'attaccante abbia la possibilità di individuare il posizionamento fisico e reale del dispositivo, e abbia la capacità tecnica di raggiungerlo ed interagire con esso [61].
Ovviamente, come già accennato in precedenza, l'Access Point utilizzato all'interno delle nostre reti locali ed interne, è un semplice Router domestico che quindi può essere banalmente configurato con solo tre differenti meccanismi e standard crittografici intenti a garantire una corretta autenticazione ed un corretto e sicuro accesso alla rete, ovviamente in un contesto wireless (es. WiFi):

Nessuna protezione o misura di autenticazione scelta
Questa sicuramente è la situazione ideale e ottimale per l'attaccante, in quanto può tranquillamente interconnettersi ed accedere alla rete, senza bisogno di dover passare oltre alcun meccanismo di verifica e autenticazione, potendo quindi interagire con ogni dispositivo presente al suo interno, senza alcuna limitazione; ovviamente tale scenario, dall'altra parte, risulta essere il più pericoloso e meno sicuro per il sistema stesso.

WEP
Il WEP (Wired Equivalent Privacy) fu il primo standard crittografico che nacque nel lontano 1999 con lo scopo di proteggere e rendere maggiormente sicure ed affidabili le cosiddette "reti senza fili" (wireless) (es. reti che sfruttano come mezzo trasmissivo il WiFi), cercando di far si che il loro livello di protezione e sicurezza risulti essere il più equivalente possibile a quello praticamente assoluto che emerge all'interno delle reti wired (cablate): quest'oggi, anche rientra ancora a far parte dello standard IEEE 802.11, è altamente sconsigliato il suo utilizzo in ambiti reali e domestici, in quanto anche un utente con un minimo di competenze tecniche ed informatiche, può riuscire a bucare i suoi meccanismi di sicurezza interni.

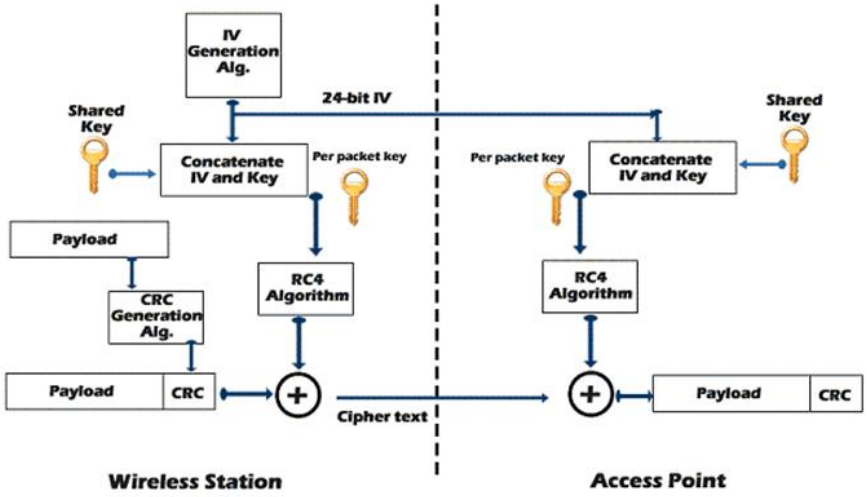

Figura 10. Attacco WEP

Come si evince dalla Figura 10, il funzionamento di base di tale standard crittografico, dal punto di vista protocollare, è molto semplice ed intuitivo.
Innanzitutto, la Wireless Station e l'Access Point devono avere in comune una chiave segreta, spesso a 40 bit, precondivisa (si parla di una PSK, ovvero di una Pre Shared Key); supposto che entrambi siano a conoscenza di tale chiave (che in genere viene associata alla "password del WiFi"), nel momento in cui la Wireless Station deve inviare un messaggio (payload) verso l'Access Point, affinché poi quel messaggio sia distribuito da quest'ultimo all'intero della rete scelta, deve seguire i seguenti passi: inizialmente si deve occupare di generare un IV (Initialitation Vector) a 3 byte, affinchè tale valore venga concatenato alla PSK e il valore risultante venga passato come seed ad un PRNG (Pseudo-Random Number Generator), basato sull'utilizzo dell'algoritmo RC4, che si occuperà successivamente di generare un keystream pseudorandomico; fatto ciò, verrà preso poi il payload da inoltrare e ad esso sarà concatenato con il suo CRC (Cyclic Redundancy Check) al fine di eseguire lo XOR di tale valore risultante, con il keystream precedentemente ottenuto; da qui si otterrà il testo cifrato, che concatenato col valore in chiaro dell'IV corrente, sarà inviato all'Access Point.
Una volta che l'Access Point avrà ricevuto il messaggio cifrato, estrarrà da esso l'IV associato, lo allegherà alla PSK di cui è in possesso, rigenererà per tramite di tale valore ottenuto lo stesso keystream (usando sempre RC4 con lo stesso seed) della Wireless Station, e facendo lo XOR col testo cifrato, otterrà la versione in chiaro di quest'ultimo.
Detto ciò possiamo ora andare ad analizzare uno dei più utilizzati e rapidi meccanismi di attacco che vengono adoperati per riuscire ad ottenere la PSK che risulta essere tenuta nascosta e segreta all'interno del dispositivo che funge da Access Point; tale tipologia di attacco si basa su un particolare metodo statistico chiamato col termine di **PTW (Pyshkin ,Tews and Weinmann**): l'idea che sta alla base del metodo PTW è quella di ottenere una lista associata di keystream e IV differenti, con il successivo scopo di riuscire, sfruttando la relazione che sussiste tra essi, a risalire alla PSK utilizzata per la stessa cifratura.

WPA/WPA2

Come analizzato in precedenza, il protocollo crittografico WEP risulta essere diventato ormai obsoleto e totalmente vulnerabile a diverse tipologie di attacco di tipo statistico che si basavano ad esempio sull'ottenimento di nuovi e differenti IV e keystream, come visto nel caso del metodo di PTW: proprio per questo motivo, intorno agli anni 2003/2004, la Wi-Fi Alliance decise di proporre due nuovi possibili standard crittografici legati principalmente all'aspetto di sicurezza e di autenticazione delle reti wireless, associabili specialmente al contesto WiFi e denominandoli appositamente col termine di WPA (Wi-Fi Protected Access) e WPA2 (Wi-Fi Protected Access II). La sostanziale differenza che occorre tra le due differenti versioni di WPA, è semplicemente data dal tipo di protocollo di cifratura utilizzato per garantire la trasmissione sicura e autenticata dei dati, dalle client-station verso i corrispettivi Access Point (come può essere un router domestico): difatti la versione WPA iniziale, utilizza come protocollo crittografico il **TKIP (Temporal Key Integrity Protocol)**, mentre la versione successiva (WPA2) utilizza alcune estensioni del protocollo **CCMP (CTR mode con CBC-MAC Protocol)** come ad esempio **AES-CCMP** [62].
Sia il protocollo WPA che WPA2 offrono due differenti e possibili scelte sul meccanismo e sul processo di autenticazione da poter utilizzare: in generale, per l'utilizzo domestico, viene data la possibilità di sfruttare un meccanismo "autenticativo" basato sul concetto di PSK (Pre-Shared-Key), mentre se ci si sposta in un contesto aziendale ed "enterprise", viene data la possibilità di utilizzare un processo di autenticazione EAP-Based, con l'ausilio di SERVER AAA esterni, come può essere un server Radius.
Detto ciò, sicuramente la parte più importante e critica, anche dal punto di vista di un attaccante, di tale protocollo crittografico e di autenticazione, è sicuramente la fase iniziale del four-way-handshake:

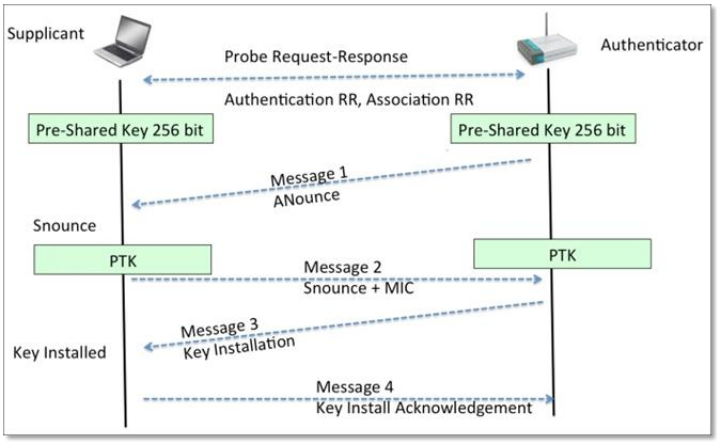

Figura 11. Attacco WPA/WPA2 alla rete IoT.

Innanzitutto, prima dell'esecuzione effettiva del four-way-handshake, sia il Supplicant che l'Authenticator, si generano quella che viene chiamata col termine di **PMK (Pairwise Master Key)**, la quale viene generata in maniera differente, dipendentemente dal tipo di protocollo di autenticazione.
Una volta che entrambi le entità coinvolte sono riuscite a generare correttamente la **PMK** grazie alla conoscenza del passphrase segreto e condiviso, può iniziare effettivamente la fase di four-way-handshake, necessaria a garantire la corretta generazione della chiave **PTK (Pairwise Transient Key)**, che risulterà essere la chiave crittografica simmetrica di sessione, attraverso la quale, successivamente, la client-station e l'AP, potranno scambiarsi in maniera sicura ed autentica i dati di rete.
Una delle principali tipologie di attacco nei confronti della quale, tale protocollo risulta essere direttamente vulnerabile, ovvero gli attacchi basati su dizionario. L'idea difatti alla base di quest'ultimi, è molto semplice ed intuitiva: l'obbiettivo che l'attaccante si prefigge di ottenere, è il recupero della PSK (passphrase) segreta e condivisa che la client-station e l'AP hanno in comune, così poi da poterla utilizzare per farsi successivamente autenticare dall'AP stesso, ottenendo completo accesso alla rete interna scelta; per raggiungere tale scopo, ciò che inizialmente l'attaccante deve compiere, è mettersi in ascolto lungo il canale di trasmissione dell'AP, al fine di sniffare tutti i differenti pacchetti in transito da esso verso i client e viceversa, sperando di intercettare tutti e quattro i pacchetti di un generico four-way-handshake che viene indetto, durante l'autenticazione di un nuovo client.

Evil Twin Attack

Per terminare questa fase prettamente descrittiva per ciò che concerne le principali tipologie di attacco che possono essere montate ed eseguite con lo scopo di recuperare i passphrase (PSK) associate ai relativi AP che utilizzano protocolli standard crittografici come WEP e WPA2, è opportuno analizzare una forma di attacco e di minaccia, che si discosta dalle tipologie precedentemente analizzate, ma che permette, non lavorando in maniera diretta sulle vulnerabilità presenti nei due protocolli crittografici precedentemente introdotti, di riuscire comunque a bucarli con uno sforzo ridotto e minimo: tale nuovo metodo di attacco è indicato col nome di Evil Twin attack [63].
Lo scopo applicativo di questa forma di attacco è molto semplice e diretta: difatti l'obiettivo dell'attaccante è quello di ottenere il valore della **PSK** (che sia relativa a WPA/WPA2 o WEP) in chiaro, inoltratagli direttamente dalla vittima stessa, facendo sì che quest'ultima si colleghi e si autentichi nei confronti di un **Fake Access Point (FAC)**, il quale maschera e nasconde la presenza di un AP legittimo, rubandogli completamente la propria identità.
Detto ciò, analizziamo quelli che sono i diversi e i principali step che un attaccante deve seguire, per montare correttamente questa tipologia di attacco:
1. L'attaccante fa una rapida scansione per individuare ed ottenere più informazioni possibili sul target AP scelto, informazioni personali come il nome dell'SSID, il suo Channel number e il suo MAC Address (esse serviranno per ricreare un nuovo e fake Access Point con le stesse e identiche caratteristiche di quello reale).

2. Una volta creato il fake Access Point con le info ottenute al passo precedente, ciò che fa l'attaccante è quello di indurre, tramite deauthentication attack, una disconnessione continua dei clients nei confronti del legittimo AP, in maniera tale da indurli e forzarli a connettersi all'AP fraudolento.
3. Una volta che il client si sarà connesso al FAP di riferimento, con ottima probabilità, aprirà il browser per navigare; ovviamente tale richiesta verrà intercettata e arriverà direttamente al fake Access Point di riferimento.
4. Tale FAP, risponderà alla richiesta del client, inviandogli come risposta alla sua precedente richiesta, una pagina Web contenente un web administrator warning con la seguente frase associata: "Enter WEP/WPA2 password to download and upgrade the router firmware"
5. Il client preso di mira, non sapendo la pericolosità di tale attacco, inserirà le credenziali richieste (quindi la PSK), facendo sì che esse vengano inoltrate al fake AP, il quale le andrà a memorizzare su un apposito database MySQL, permettendo successivamente all'attaccante, accedendo allo stesso DB, di poterle leggere in chiaro.

Qui di seguito, viene mostrata una immagine riassuntiva del tutto, Figura 12:

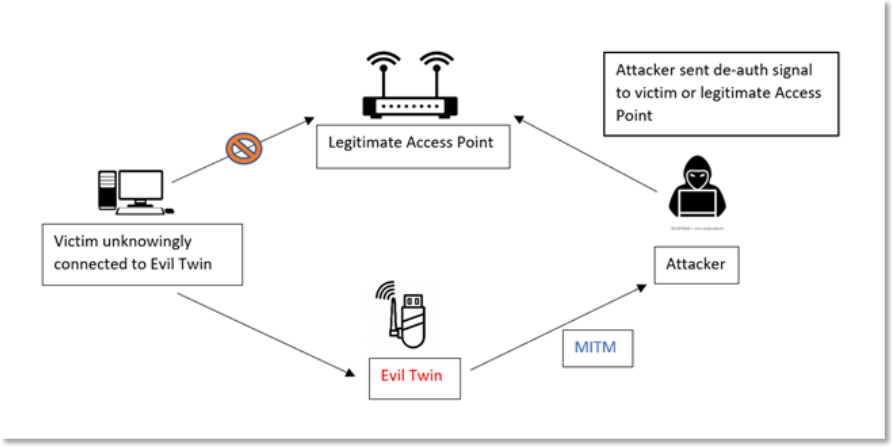

Figura 12. Attacco Evil Twin in una rete IoT.

1.29.2 Attacker: Information Gathering

Una volta che l'attaccante è riuscito, ad esempio sfruttando uno qualsiasi dei vettori di attacco precedentemente analizzati, ad ottenere in maniera "illegittima" l'accesso all'interno della rete locale e interna di riferimento, bypassando le contromisure in termini crittografici e di sicurezza imposte dall'Access Point presente, deve necessariamente cercare di estrapolare ed ottenere più informazioni possibili su quelli che sono i differenti dispositivi interni e presenti in tale rete, al fine poi da riuscir ad individuare nuovi possibili target vittima da attaccare e minacciare, come potrebbero risultare esserlo il *dispositivo IoT* e il *Local Gateway*. In termini prettamente formali, tale fase esecutiva di attacco in cui si cerca di estrarre e ottenere più informazioni e dati possibili in rifermento ad un possibile target scelto, che può essere come in questo caso una wireless network, prende il nome di information gathering.
Detto ciò, andiamo ad analizzare rapidamente, quelli che sono i principali tools che un attaccante può sfruttare ed utilizzare, al fine di eseguire correttamente la fase appena citata, prendendo come target di riferimento, la stessa rete all'interno della quale si suppone che esso sia riuscito ad ottenere accesso.
Il più famoso e conosciuto strumento che viene ad essere utilizzato in contesti di questo tipo, è sicuramente **nmap**. Come permette facilmente di intuire il nome dato a questo tool, esso risulta essere uno strumento free e open source, che offre intrinsecamente una serie di servizi necessari a garantire una rapida ed esaustiva ricerca basata sul **Mapping della rete (Network Mapping)**, sul **Network Discovery** e persino sul **security auditing**: difatti quest'ultimo, sfruttando la realizzazione di raw socket e la generazione e ricezione di paccheti raw, può riuscire ad esempio a determinare quali hosts sono attivi e disponibili all'interno della rete di analisi, il loro indirizzi di riferimento, i servizi a livello applicativo che essi hanno esposto

sulla stessa rete, la tipologia di sistema operativo e la sua versione utilizzata, la presenza di filters o firewall e molte altre caratteristiche tecniche di questo tipo.

1.29.3 Attacker: Network Protocol Attacks

Una volta che l'attaccante è riuscito, in primis ad ottenere un accesso illegittimo all'interno della rete targhettata e poi tramite tecniche di information gathering, come visto in precedenza, ad ottenere molte info sensibili e di interesse per quanto riguarda i dispositivi interni da attaccare e prendere di mira, può cercare di concentrare la propria attenzione, sull'andare ad imbastire una serie di tipologie di attacchi, sia passivi che attivi, che mirano a corrompere e compromettere il normale funzionamento interno e applicativo dei differenti protocolli comunicativi che permettono ai dispositivi convolti di scambiarsi dati e informazioni di contesto.
Analizzando lo stesso **threat model** citato e descritto in precedenza, i "*poteri*" di attacco che si assume che l'attaccante stesso abbia a disposizione sono molteplici ma i principali e sicuramente i più critici da tenere in considerazione sono quelli riguardanti il fatto che esso ha la possibilità di interagire e comunicare, senza alcun problema di stampo autenticativo, coi differenti dispositivi interni alla rete coinvolta, potendo sfruttare come punto di attacco, le vulnerabilità presenti in uno qualsiasi tra i protocolli applicativi presenti e utilizzabili al suo interno.
Nel nostro caso di interesse, l'attaccante mira principalmente ad ottenere la possibilità di intercettare il flusso di messaggi che risulta essere scambiato tra il dispositivo IoT e lo stesso Local Gateway presente nell'infrastruttura, col fine di, in primis, cercare di leggerne il contenuto per poi successivamente cercare di alterarlo, falsificarlo o addirittura scartare il messaggio affinché non arrivi mai a destinazione: da ciò quindi si capisce, come l'obbiettivo principale dell'attaccante stesso sarà quello di rompere le contromisure di sicurezza che vengono imposte all'interno dei messaggi a livello protocollare, in termini di **confidenzialità**, **integrità**, **autenticazione** e **disponibilità**.
Ovviamente, affinché l'attaccante riesca a svolgere correttamente le tipologie di attacco descritte ed introdotte in precedenza, deve riuscire in primis ad intercettare l'intero traffico comunicativo che viene ad essere scambiato tra le due entità coinvolte all'interno del sistema: per raggiungere tale obbiettivo, l'attaccante stesso può, sfruttando il fatto di trovarsi all'interno della stessa rete locale ed interna dei dispositivi vittima, imbastire uno dei più tipici e pericolosi network attack di base, che prende il nome di **Man In The Middle (MITM) attack**.

Man In The Middle (MITM) attack

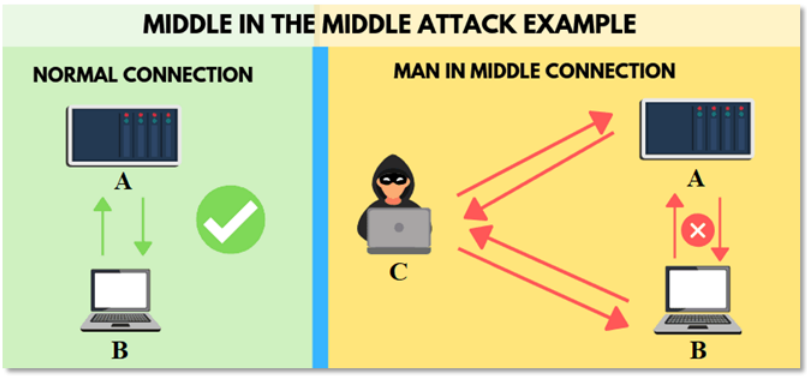

Figura 13. Man in the middle (MITM)

In Figura 13 viene mostrato in forma pratica e reale, in cosa consiste effettivamente tale tipologia di attacco. Una volta concepito e capito quello che è il concetto che sta alla base di questo attacco, si può cercare di intuire in quale modo l'attaccante possa trarre effettivamente vantaggio sfruttandolo all'interno dell'architettura di rete interna del nostro sistema IoT; difatti l'attaccante, una volta penetrato e autenticato all'interno della rete locale in cui sono presenti sia il

dispositivo IoT sia il Local Gateway, può usufruire di questa pericolosissima forma di attacco, realizzando ed ottenendo le seguenti tipologie di contesti altamente pericolosi:

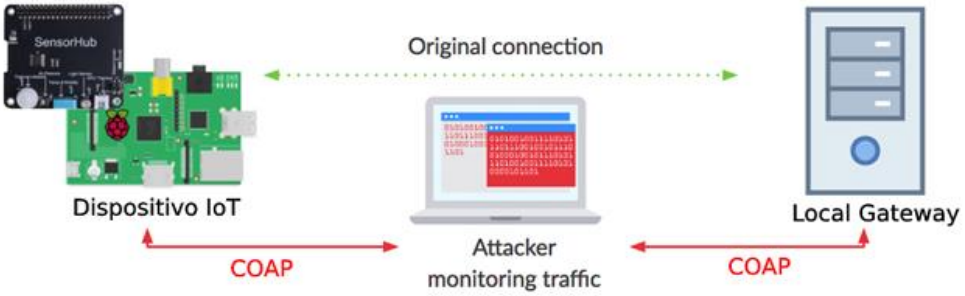

Figura 14. Attacco MITM con protocollo CoAP

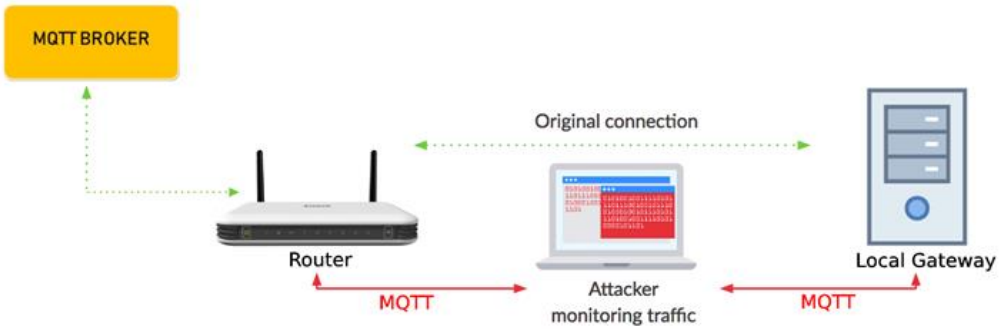

Figura 15. Attacco MITM con protocollo MQTT

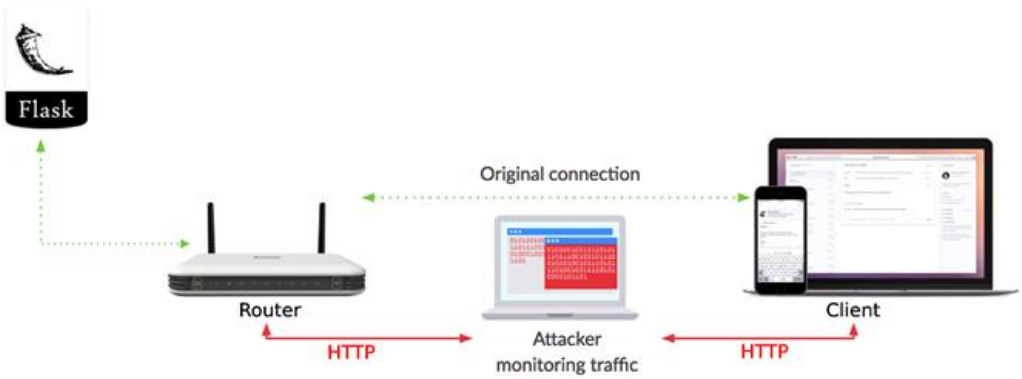

Figura 16. Attacco MITM con protocollo HTTP

NOTA: **Flask** è un Web Framework di Python (una third-party Python Library) quest'oggi molto conosciuto ed utilizzato, in quanto permette la realizzazione della logica di back-end di una Web Application, in maniera molto semplice, diretta ed efficiente per lo sviluppatore, permettendo a quest'ultimo di poter sfruttare a pieno i vantaggi che un linguaggio ad "altissimo" livello come Python mette lui a disposizione

Come quindi si può osservare dalle tre figure sopra, Figura 14, Figura 15 e Figura 16, l'obiettivo ultimo dell'attaccante è semplicemente quello di riuscire ad intercettare il contenuto dei messaggi di tipo CoAP, MQTT e HTTP che vengono scambiati tra le differenti entità presenti nella stessa rete, al fine poi di poterne leggere il contenuto, modificarlo, alterarlo, falsificarlo o addirittura eliminarlo totalmente.

ARP Poisoning (MITM)

Capita l'importanza e la pericolosità esecutiva di un MiTM Attack, è opportuno cercare di capire come a livello pratico possa essere realizzato un attacco di questo tipo. In realtà esistono diverse tipologie di meccanismi e di processi applicativi che un attaccante può seguire e portare avanti con lo scopo di instaurare un man in the middle attack: tra questi, sicuramente la tipologia che interessa maggiormente noi, visto il threat model che è stato assunto ed associato all'attaccante, riguarda il modo con cui tale tipologia di minaccia possa essere instaurata all'interno di una rete locale e privata, come può risultare esserlo una generica LAN; al fine di raggiungere lo scopo precedentemente citato, ciò che serve unicamente compiere all'attaccante, in tale contesto, è rappresentato da una forma di network attack denominato col termine di ARP Poisoning, ovvero "avvelenamento delle tabelle/cache ARP" [64].

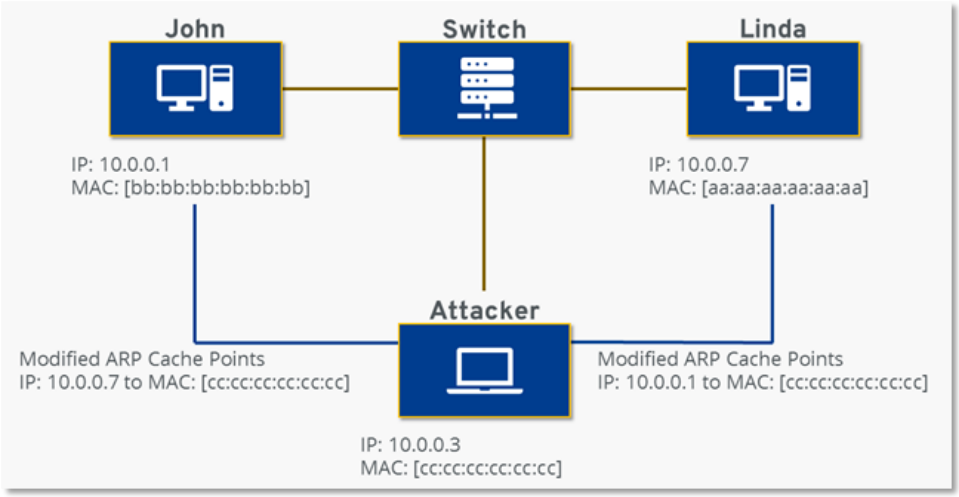

Figura 17. Attacco ARP Poisoning nella rete

Due fondamentali tools che possono essere utilizzati per eseguire praticamente e realmente l'ARP Poisoning e quindi indirettamente l'attacco locale del MiTM sono: **Ettercap** e **arpspoof**.

1.30 Gestione della sicurezza e relative minacce nel protocollo HTTP

Una volta aver capito l'importanza e soprattutto la pericolosità del MiTM attack, attraverso il quale l'attaccante riesce banalmente a mettersi nel mezzo del flusso di comunicazione e del traffico di messaggi scambiato tra due entità, è opportuno ora cercare di capire ed approfondire ciò che effettivamente esso può compiere, una volta riuscito ad intercettare ed ottenere tali messaggi di riferimento: ovviamente questa tipologia di attacchi successivi, sono principalmente incentrati o sulla lettura in chiaro del contenuto dei messaggi (attacchi passivi) o addirittura sulla loro modifica e alterazione, la loro falsificazione e persino la possibilità di scartarli evitando che quest'ultimi arrivino al destinatario (attacchi attivi); è importante sottolineare però come, la possibilità che l'attaccante ha nell'eseguire questa tipologia di attacchi successivi, dipende fortemente dal tipo di vulnerabilità che risultano essere presenti nei protocolli applicativi che vengono utilizzati per scambiare i messaggi stessi. In questo paragrafo verranno messi in evidenza le principali

tipologie di minacce e di attacchi che possono essere eseguiti nei confronti dei parametri di **confidenzialità**, **integrità**, **autenticazione** e **disponibilità** presenti e definiti all'interno del protocollo HTTP.
In questo caso si suppone che, per semplicità, lo scenario descritto sia quello in cui l'attaccante è riuscito a penetrare all'interno della stessa rete locale e privata (utilizzando uno degli attacchi descritti in precedenza) dove è presente il dispositivo Client che vuole connettersi in remoto per visionare il contenuto dei dati ambientali messi a disposizione dal Server Flask, e che sia riuscito persino successivamente, tramite ARP Poisoning, a mettersi nel mezzo della comunicazione tra il router che gestisce il traffico locale della rete e lo stesso dispositivo client (MiTM Attack).

1.30.1 Confidenzialità

Senza scendere troppo nei dettagli architetturali, una delle principali pecche presenti all'interno del protocollo HTTP (HyperText Transfer Protocol) sin dalle sue origini, è legata alla mancanza di misure preventive per garantire la corretta confidenzialità e segretezza delle informazioni applicative che viaggiano all'interno dei propri messaggi di richiesta e di risposta: difatti, tale contenuto informativo viene inserito nel body della richiesta/risposta HTTP di riferimento completamente in chiaro e leggibile da chiunque riceve legittimamente o illegittimamente il messaggio scelto. Questa tipologia di vulnerabilità può ovviamente essere sfruttata dall'attaccante sopradescritto, in quanto quando un client richiede l'accesso e il contenuto informativo relativo alle condizioni ambientali di una certa locazione al server remoto, sia la richiesta che la risposta saranno gestite e inoltrate tramite protocollo HTTP: ciò implica che l'attaccante, essendosi messo in mezzo a tale flusso comunicativo può intercettare tali messaggi HTTP e leggere interamente il contenuto informativo presente al loro interno, che teoricamente dovrebbe rimanere confidenziale, visto l'importanza personale dei dati presenti. In parole povere l'attaccante, nel momento in cui riceve pacchetti tramessi attraverso la rete, li memorizza all'interno di una NetworkQueue, ottenuta sfruttando un'apposita libreria di Python *netfilterqueue*; una volta fatto ciò, individua i pacchetti che presentano avere a livello applicativo il protocollo HTTP e molto semplicemente ne scrive il contenuto su un apposito file .pcap chiamato sniffing_passivo_http.pcap. Infine l'attaccante può liberamente aprire usando wireshark il file .pcap e leggere il contenuto in chiaro dei pacchetti HTTP intercettati.

1.30.2 Integrità

Una successiva e ulteriore mancanza, in termini di features orientate alla sicurezza, all'interno del protocollo HTTP, è sicuramente l'assenza effettiva di contromisure orientate a garantire l'integrità degli stessi messaggi: ciò implica necessariamente che l'attaccante che riesce ad intercettare il messaggio HTTP in transito, può compiere qualsiasi modifica di interesse al suo interno avendo la certezza che reinoltrandolo al destinatario effettivo, quest'ultimo non si potrà mai accorgere delle effettive modifiche e alterazioni illecite in atto presenti al suo interno. Riadattando questa problematica al nostro caso di interesse, l'attaccante può quindi, senza problemi, dopo aver intercettato i messaggi HTTP in transito da parte del client, modificarne a suo piacimento il loro contenuto, andando per esempio ad alterare il contenuto in chiaro dei valori ambientali richiesti e ricevuti dal client stesso. Questa volta l'attaccante, intercettando i messaggi HTTP e memorizzandoli nella NetworkQueue, va a verificare se essi contengono a livello di payload, i frammenti dissezionati relativi alla pagina HTML richiesta e inoltrata al client dove risultano essere presenti i differenti dati atmosferici richiesti; una volta individuati tali frammenti, molto semplicemente altera il loro contenuto, inserendo per esempio un valore "irreale", come il -1, nei differenti campi presenti.

1.30.3 Disponibilità

Un'altra tipologia di attacco nei confronti del protocollo HTTP che un MiTM attacker può compiere, prende il nome di **Black Hole Attack**: questa tipologia di attacco può essere ricondotta ad una sottospecie di attacco di tipo *DOS (Denial of Service)* in quanto l'attaccante impedisce al destinatario di ricevere determinati messaggi e pacchetti inoltrati verso di lui scartandoli a priori (da qui appunto deriva il concetto di Black Hole in quanto l'attaccante diviene un "buco nero" nella rete, che assorbe i messaggi in transito, scartandoli e non facendoli arrivare alla destinazione corretta). Ovviamente, anche questa tipologia di attacco, può essere facilmente ricondotta e applicata nel nostro contesto e scenario di riferimento, rispetto a quelli che

sono i pacchetti HTTP in transito verso il client richiedente. In questo caso, una volta che l'attaccante riceve, intercettando e sniffando, i messaggi HTTP, li salva prima nell'apposita NetworkQueue costruitasi, per poi successivamente prenderli uno ad uno e "dropparli", scartandoli a priori senza reinoltro verso il destinatario.

1.30.4 Autenticazione

Come ultima problematica in termini di sicurezza che può essere associata al protocollo HTTP, la si può sicuramente riscontrare nell'assoluta mancanza di meccanismi che possano permettere di firmare ed autenticare l'autore effettivo del messaggio: difatti non viene eseguito alcuno controllo applicativo che permetta di avere la certezza in riferimento alla provenienza stessa del messaggio; ciò significa che chiunque potrebbe, tramite tecnica di injection, iniettare all'interno del canale TCP creatosi tra client e server remoto, messaggi HTTP con qualsiasi tipologia di contenuti presenti al suo interno, senza ovviamente che il client possa accorgersi che quei messaggi HTTP in realtà non provengono dal mittente reale con cui ha istaurato precedentemente il canale comunicativo.

1.31 Gestione della sicurezza e relative minacce nel COAP

Come già accennato in precedenza, il protocollo **Constrained Application Protocol (CoAP)**, può essere visto come una versione più leggera e riadattata del protocollo HTTP, utilizzabile e sfruttabile in contesti in cui risultano essere presenti dispositivi "*constrained*", che presentano avere un quantitativo di risorse computazionali, di memorizzazione ed energetiche, nettamente limitate e ristrette. Questo necessariamente implica che tutte le diverse pecche e mancanze in termini di features di sicurezza presenti all'interno dell'*HyperText Transfer Protocol*, vengono anche riscontrate e derivate all'interno della sua "*versione più leggera*" di riferimento: ciò quindi porta a comprendere come anche il protocollo CoAP risulta essere fortemente vulnerabile a tutte quelle tipologie di attacchi passivi e attivi che un attaccante malevolo può compiere come ad esempio la possibilità di mettere in piedi un MiTM attack e quindi effettuare attacchi di sniffing e eavesdropping per poter percepire e analizzare il traffico scambiato dalle entità legacy. Quindi nel momento in cui risultano essere presenti all'interno di una stessa rete locale, due entità intente a scambiarsi dati e informazioni lavorando a livello applicativo con il protocollo CoAP, il loro traffico informativo, se correttamente intercettato da un dispositivo malevolo, può sia essere letto in chiaro, senza alcuna limitazione, sia essere facilmente modificato, alterato, falsificato e scartato, senza che il destinatario possa accorgersi di nulla [65].
Detto ciò, adesso verranno proprio messe in evidenza le principali tipologie di minacce e di attacchi che possono essere eseguiti nei confronti delle proprietà di **confidenzialità**, **integrità**, **autenticazione** e **disponibilità** presenti e definiti all'interno dei messaggi scambiati per tramite del protocollo CoAP.
Anche in questo caso si suppone che, per semplicità, lo scenario descritto sia quello in cui l'attaccante è riuscito a penetrare all'interno della stessa rete locale e privata (utilizzando uno degli attacchi descritti in precedenza) dove è presente sia il Local Gateway, intento a ricevere i messaggi CoAP provenienti dal dispositivo IoT e sia lo stesso dispositivo IoT intento a ricevere le informazioni dai sensori per trasformarli in messaggi nel corrispettivo formato CoAP ed inviarli al Local Gateway di riferimento. Ovviamente si suppone anche che l'attaccante sia riuscito persino, tramite *ARP Poisoning*, a mettersi nel mezzo della comunicazione tra i due dispositivi appena introdotti (MiTM Attack).
È opportuno inoltre notare come, per semplicità, gli attacchi verranno mostrati all'interno della rete locale strutturata con l'architettura CoAP che sfrutta l'invio del messaggio PUT dal dispositivo IoT verso il Local Gateway, ma il tutto sarebbe risultato completamente identico, anche eseguendo i test nel caso in cui fosse stata sfruttata l'architettura opposta con l'observer e il subject.

1.31.1 Confidenzialità

Senza anche qui scendere troppo nei dettagli architetturali, una delle principali pecche presenti all'interno del protocollo CoAP sin dalle sue origini, è legata alla stessa mancanza di misure preventive per garantire la corretta confidenzialità e segretezza delle informazioni applicative che viaggiano all'interno dei propri messaggi di richiesta e di risposta, che è stata

riscontrata in precedenza all'interno del protocollo applicativo HTTP: difatti, tale contenuto informativo viene inserito nel campo payload del corrispettivo messaggio di richiesta/risposta CoAP di riferimento completamente in chiaro e leggibile da chiunque riceve legittimamente o illegittimamente il messaggio scelto (da notare come anche i parametri configurativi e opzionali del messaggio CoAP sono completamente in chiaro).
Questa tipologia di vulnerabilità può ovviamente essere sfruttata dall'attaccante sopradescritto, in quanto quando il dispositivo IoT inoltra il contenuto informativo relativo alle condizioni ambientali della sua locazione di riferimento al Local Gateway, tale messaggio sarà gestito e inoltrato tramite protocollo CoAP: ciò implica che l'attaccante, essendosi messo in mezzo a tale flusso comunicativo, attraverso lo script mostrato seguentemente, può intercettare tali messaggi CoAP e leggere interamente il contenuto informativo presente al loro interno, che teoricamente dovrebbe rimanere confidenziale, visto l'importanza personale dei dati presenti.
Esattamente in maniera analoga a ciò che è stato visto ed analizzato in precedenza col protocollo HTTP quello che molto semplicemente l'attaccante va a compiere è: nel momento in cui riceve pacchetti tramessi attraverso la rete, li memorizza all'interno di una NetworkQueue, ottenuta sfruttando un'apposita libreria di Python netfilterqueue; una volta fatto ciò, individua i pacchetti che presentano avere a livello applicativo il protocollo CoAP e molto semplicemente ne scrive il contenuto su un apposito file .pcap chiamato sniffing_passivo_coap.pcap. Infine, l'attaccante può liberamente aprire usando wireshark il file .pcap e leggere il contenuto in chiaro dei pacchetti CoAP intercettati:

1.31.2 Integrità

Anche con CoAP, in maniera analoga a ciò che accade col protocollo HTTP, una successiva e ulteriore mancanza, in termini di features orientate alla sicurezza, è sicuramente l'assenza effettiva di contromisure orientate a garantire l'integrità degli stessi messaggi: ciò implica necessariamente che l'attaccante che riesce ad intercettare il messaggio CoAP in transito, può compiere qualsiasi modifica di interesse al suo interno avendo la certezza che reinoltrandolo al destinatario effettivo, quest'ultimo non si potrà mai accorgere delle effettive modifiche e alterazioni illecite in atto presenti al suo interno. Riadattando questa problematica al nostro caso di interesse, l'attaccante può quindi, senza problemi, dopo aver intercettato il messaggio CoAP della PUT in transito dal dispositivo IoT verso il Local Gateway, modificarne a suo piacimento il contenuto, andando per esempio ad alterare il contenuto in chiaro dei valori ambientali richiesti e presenti
Anche qui, in maniera analoga a ciò che è stato già analizzato nel segmento relativo al protocollo HTTP l'attaccante può intercettare i messaggi CoAP memorizzandoli nella NetworkQueue, andando successivamente a verificare se essi contengono a livello di payload, sotto forma di JSON, i differenti valori relativi ai dati informativi energetici richiesti; una volta individuati tali frammenti, molto semplicemente altera il loro contenuto, inserendo un valore "irreale", falsando il corretto scambio e la lecita condivisione delle risorse energetiche tra nodi della smart grid.

1.31.3 Disponibilità

Esattamente come già analizzato per il protocollo HTTP, esiste un'altra tipologia di attacco nei confronti del protocollo CoAP che un MiTM attacker può compiere, che va sotto il nome di Black Hole Attack: questa tipologia di attacco può essere ricondotta ad una sottospecie di attacco di tipo DOS (Denial of Service) in quanto l'attaccante impedisce al destinatario di ricevere determinati messaggi e pacchetti inoltrati verso di lui scartandoli a priori (da qui appunto deriva il concetto di Black Hole in quanto l'attaccante diviene un "buco nero" nella rete, che assorbe i messaggi in transito, scartandoli e non facendoli arrivare alla destinazione corretta). Ovviamente anche questa tipologia di attacco può essere facilmente ricondotta e applicata nel nostro contesto e scenario di riferimento, rispetto a quelli che sono i messaggi PUT CoAP in transito dal dispositivo IoT verso il Local Gateway richiedente.
Anche qui, una volta che l'attaccante riceve i messaggi CoAP, intercettandoli e sniffandoli, li salva prima nell'apposita NetworkQueue costruitasi, per poi successivamente prenderli uno ad uno e "dropparli", scartandoli a priori senza alcun reinoltro verso il destinatario (da notare come in questo caso, il dispositivo IoT inoltra tramite PUT il messaggio contenente le info richieste, ma tale messaggio non arriverà mai al Local Gateway in quanto verrà prima scartato dall'attaccante; ciò implica quindi che il dispositivo stesso non riceverà mai l'ack di riferimento e dopo un determinato intervallo di attesa,

crasherà). Ne consegue, dunque, l'eventuale indisponibilità dei nodi presi di mira nell'attacco che non potranno partecipare correttamente al processo di "**two-way flow of power**" rendendo indisponibile il normale transito del flusso energetico.

1.31.4 Autenticazione

Infine, anche CoAP come anche già visto col protocollo HTTP, come ultima problematica in termini di sicurezza, si può sicuramente riscontrare l'assoluta mancanza di meccanismi che possano permettere di firmare ed autenticare l'autore effettivo del messaggio inoltrato: di fatti non viene eseguito alcuno controllo applicativo che permetta di avere la certezza in riferimento alla provenienza e l'origine stessa del messaggio; ciò significa che chiunque potrebbe, tramite tecnica di **injection**, iniettare ed inviare verso la socket UDP creata e messa a disposizione dal Local Gateway, messaggi CoAP con qualsiasi tipologia di contenuto presenti al suo interno, senza ovviamente che il gateway stesso possa accorgersi che quei messaggi CoAP in realtà non provengono dal possibile mittente reale, ovvero il dispositivo IoT, bensì sono stati generati e falsificati da parte di un dispositivo malevolo.
Ovviamente, il tutto può essere facilmente riportato al nostro caso di interesse: di fatti all'attaccante basta semplicemente, tramite attacco passivo di sniffing (visto in precedenza) intercettare e memorizzarsi determinati messaggi CoAP, in maniera tale da individuare al suo interno, l'IP destinatario del messaggio e il Path della risorsa di riferimento presente (es. 192.168.1.10/grid_node). Una volta fatto ciò, anche lui potrà inviare senza problemi, messaggi PUT in stile CoAP con dentro il payload contenente i dati da lui scelti e fissati, aggiornando in maniera illegittima e falsificata i dati memorizzati e presenti della risorsa sensori, senza che il Local Gateway possa accorgersi della sua falsa identità:
Una volta che l'attaccante ha ottenuto le due info necessarie per eseguire l'injection, ovvero l'IP destinatario e il path della risorsa, molto semplicemente gli basta creare un JSON dello stesso formato richiesto dal Local Gateway (anche questa info la ottiene tramite sniffing passivo) e poi generando un nuovo messaggio di PUT CoAP e inserendo il contenuto dello stesso JSON come payload al suo interno, lo andrà infine ad inviare al destinatario ignaro di tutto, che ovviamente andrà a supporre, senza alcun controllo di autenticazione, che il messaggio sia stato inviato legittimamente dal dispositivo IoT. Chiaramente, tutto ciò, riportato al contesto delle smart grid di riferimento può facilmente portare alla situazione in cui un attaccante può fingersi di essere un nodo legacy nella smart grid e partecipare in maniera illecita al processo di generazione e condivisione dell'energia.

1.32 Gestione della sicurezza nel protocollo MQTT con relative minacce

Come già accennato in precedenza, anche il protocollo **MQTT (Message Queue Telemetry Transport)** esattamente come il protocollo CoAP, viene ad essere utilizzato e sfruttato principalmente in contesti in cui sono presenti dispositivi "*constrained*", che risultano avere un quantitativo di risorse computazionali e di memorizzazione, prettamente limitate e ristrette, come accade all'interno di reti composte da dispositivi IoT. Esattamente come accade per il protocollo CoAP, anche il protocollo MQTT, a causa del fatto che deve risultare essere il *"più leggero e il meno dispendioso*" possibile per ridurre i consumi energetici e computazionali dei dispositivi coinvolti, necessariamente presenta avere a livello progettuale e implementativo una serie di differenti pecche e mancanze in termini di features orientate alla sicurezza presenti al suo interno: ciò quindi porta a comprendere come anche tale protocollo, esattamente come i due precedenti analizzati, risulta essere fortemente vulnerabile a tutte quelle tipologie di attacchi passivi e attivi che un attaccante malevolo può compiere, una volta aver approntato, anche qui, un MiTM attack. Quindi nel momento in cui risultano essere presenti all'interno di una stessa rete locale, due entità intente a scambiarsi dati e informazioni lavorando a livello applicativo con il protocollo MQTT, come possono esserlo un generico dispositivo IoT e il Local Gateway ad esso associato, il loro traffico informativo, se correttamente intercettato da un dispositivo malevolo, può sia essere letto in chiaro, senza alcuna limitazione, sia essere facilmente modificato, alterato, falsificato e scartato, senza che il destinatario possa accorgersi di nulla [65].
Detto ciò, in questo paragrafo verranno proprio messe in evidenza le principali tipologie di minacce e di attacchi che possono essere eseguiti nei confronti dei parametri di confidenzialità, integrità, autenticazione e disponibilità presenti e definiti all'interno dei messaggi scambiati per tramite del protocollo MQTT.

Anche in questo caso si suppone che, per semplicità, lo scenario descritto sia quello in cui l'attaccante è riuscito a penetrare all'interno della stessa rete locale e privata (utilizzando uno degli attacchi descritti in precedenza) dove è presente sia il Local Gateway, intento a ricevere i messaggi MQTT provenienti dal dispositivo IoT e sia lo stesso dispositivo IoT intento a ricevere le informazioni dai sensori per trasformarli in messaggi nel corrispettivo formato MQTT ed inviarli al Local Gateway di riferimento. Ovviamente si suppone anche che l'attaccante sia riuscito persino, tramite ARP Poisoning, a mettersi nel mezzo della comunicazione tra i due dispositivi appena introdotti (MiTM Attack).
È opportuno inoltre notare come, per semplicità, nel nostro caso di interesse, in quanto il Broker è esterno alla nostra architettura di rete, l'attaccante si può porre in maniera analoga, o nel mezzo della comunicazione che intercorre tra il dispositivo IoT e il router o tra il router stesso e il Local Gateway di riferimento (dal punto di vista interno della rete locale, il router è come se divenisse una sottospecie di proxy per il Broker remoto).

1.32.1 Confidenzialità

Esattamente in maniera analoga ai due casi precedentemente descritti, una delle principali pecche presenti all'interno del protocollo MQTT sin dalle sue origini, è legata alla mancanza di misure preventive per garantire la corretta confidenzialità e segretezza delle informazioni applicative che viaggiano all'interno dei propri messaggi, specialmente per quanto riguarda quelli di publishing: difatti, tale contenuto informativo viene inserito nel campo payload del corrispettivo messaggio di pubblicazione completamente in chiaro e leggibile da chiunque riceve legittimamente o illegittimamente il messaggio scelto (da notare come anche i parametri configurativi e opzionali del messaggio MQTT sono completamente in chiaro e quindi leggibili) (ovviamente lo stesso identico discorso può essere fatto considerando i messaggi di subscribing).
Questa tipologia di vulnerabilità può ovviamente essere sfruttata dall'attaccante sopra descritto, in quanto quando il dispositivo IoT inoltra il contenuto informativo relativo alle condizioni ambientali della sua locazione di riferimento tramite una apposita publish associata ad un determinato topic al Broker, affinché poi quest'ultimo possa rigirarla al Local Gateway che precedentemente aveva eseguito una subscribe su quello stesso topic di riferimento, tale messaggio sarà gestito e inoltrato senza apporre alcuna contromisura crittografica: ciò implica che l'attaccante, essendosi messo in mezzo a tale flusso comunicativo può intercettare tali messaggi MQTT di publish/subscribe e leggere interamente il contenuto informativo presente al loro interno, che teoricamente dovrebbe rimanere confidenziale, visto l'importanza personale dei dati presenti.
Esattamente in maniera analoga a ciò che è stato visto ed analizzato in precedenza col protocollo HTTP e CoAP, quello che molto semplicemente l'attaccante va a compiere è: nel momento in cui riceve pacchetti tramessi attraverso la rete, li memorizza all'interno di una NetworkQueue, ottenuta sfruttando un'apposita libreria di Python *netfilterqueue*; una volta fatto ciò, individua i pacchetti che presentano avere a livello applicativo il protocollo MQTT e molto semplicemente ne scrive il contenuto su un apposito file .pcap chiamato sniffing_passivo_mqtt.pcap. Infine, l'attaccante può liberamente aprire usando wireshark il file .pcap e leggere il contenuto in chiaro dei pacchetti MQTT intercettati. Questo significa che le informazioni degli utenti coinvolti nel processo di generazione e condivisione energetica saranno totalmente comprensibili ad una eventuale entità esterna attaccante. Inoltre, il broker accoda i messaggi all'interno di code presenti al suo interno, di conseguenza se il broker viene compromesso automaticamente vengono compromesse le informazioni sensibili in chiaro degli utenti memorizzate all'interno delle sue code e in attesa di essere evasi.

1.32.2 Integrità

Una successiva e ulteriore mancanza, in termini di features orientate alla sicurezza che risulta purtroppo essere presente anche all'interno del protocollo MQTT, è sicuramente l'assenza effettiva di contromisure orientate a garantire l'integrità degli stessi messaggi di publish/subscribe inoltrati: ciò implica necessariamente che l'attaccante che riesce ad intercettare il generico messaggio MQTT in transito, può compiere qualsiasi modifica di interesse al suo interno avendo la certezza che reinoltrandolo al destinatario effettivo, quest'ultimo non si potrà mai accorgere delle effettive modifiche e alterazioni illecite in atto presenti al suo interno. Riadattando questa problematica al nostro caso di interesse, l'attaccante può quindi, senza problemi, dopo aver intercettato i messaggi MQTT inoltrati dal dispositivo IoT verso il Broker o dal Broker verso il

Local Gateway, modificarne a suo piacimento il loro contenuto, andando per esempio ad alterare il valore in chiaro dei dati ambientali richiesti e ricevuti dallo stesso Local Gateway tramite i messaggi di publish.
L'attaccante può intercettare i messaggi MQTT di publish memorizzandoli nella NetworkQueue, andando successivamente a verificare se essi contengono a livello di payload, sotto forma di JSON, i differenti valori relativi ai dati informativi energetici richiesti; una volta individuati tali frammenti, molto semplicemente altera il loro contenuto, inserendo un valore "irreale", come ad esempio il -1, nei differenti campi presenti. Anche qui il risultato sarà il medesimo rispetto a quello descritto, per tale caso, per il protocollo CoAP. In più si aggiunge la questione relativa al broker, che come già detto, può essere sfruttato come ulteriore punto di ancoraggio per un eventuale attaccante. L'attaccante, infatti, potrebbe forgiare dei pacchetti di publish identici a quelli generati dal publisher legacy ma modificati in alcuni campi ad hoc e farli memorizzare all'interno del broker, il quale non si accorgerà dell'accaduto e li distribuirà, come fa normalmente, ai subscriber del topic specificato nel pacchetto.

1.32.3 Disponibilità

Esattamente come già analizzato per il protocollo HTTP e CoAP, esiste un'altra tipologia di attacco nei confronti del protocollo MQTT che un MiTM attacker può compiere, prendente il nome di *Black Hole Attack*: questa tipologia di attacco, come già detto in precedenza, può essere ricondotta ad una sottospecie di attacco di tipo *DOS (Denial of Service)* in quanto l'attaccante impedisce al destinatario di ricevere determinati messaggi e pacchetti inoltrati verso di lui scartandoli a priori (da qui appunto deriva il concetto di Black Hole in quanto l'attaccante diviene un "buco nero" nella rete, che assorbe i messaggi in transito, scartandoli e non facendoli arrivare alla destinazione corretta). Ovviamente anche questa tipologia di attacco può essere facilmente ricondotta e applicata nel nostro contesto e scenario di riferimento, rispetto a quelli che sono sia i messaggi di publish che il dispositivo IoT inoltra al Broker affinché esso possa rigirarli verso il Local Gateway e sia quelli che sono i messaggi di subscribe che il Local Gateway inoltra al Broker stesso (più precisamente in questo caso, l'attaccante intercetta i messaggi di publish inoltrati dal Broker verso il Local Gateway, scartandoli e non facendoli arrivare a destinazione).
Anche in questo caso, una volta che l'attaccante riceve i messaggi di publish inoltrati verso il Local Gateway, intercettandoli e sniffandoli, li salva prima nell'apposita NetworkQueue costruitasi, per poi successivamente prenderli uno ad uno e "dropparli", scartandoli a priori senza alcun reinoltro verso il destinatario (da notare come in questo caso, il dispositivo IoT inoltra tramite publish il messaggio contenente le info richieste, esso arriverà successivamente al Broker, il quale lo reinoltrerà verso il Local Gateway ma tale messaggio non arriverà mai a quest'ultimo in quanto verrà prima intercettato e scartato dall'attaccante; ciò porterà ad un crash causa timeout dell'applicazione). È evidente, nel caso del protocollo MQTT, che il broker stesso può essere oggetto di Denial Of Service. Attraverso l'uso del tool di pentesting penIoT è infatti possibile inondare il broker di pacchetti di publish facendo si che questo non sia più in grado di gestire il traffico generato da tutti i publisher gestiti e quindi non sia in grado di consegnare ai subscriber i messaggi che si aspettano di ricevere. Sostanzialmente, rendendo indisponibile il broker, si rende indisponibile la rete publish-subscribe che esso controlla. Il che significa che eventuali nodi presenti all'interno della smart grid (che siano publisher o subscriber) non avranno modo di comunicare e scambiare le informazioni per loro necessarie.

1.32.4 Autenticazione

Infine, anche MQTT così come già visto col protocollo HTTP e CoAP, come ultima problematica in termini di sicurezza, presenta l'assoluta mancanza di meccanismi che possano permettere di firmare ed autenticare in maniera certa l'autore effettivo del messaggio pubblicato: di fatti non viene eseguito alcuno controllo applicativo che permetta di avere la certezza in riferimento alla provenienza e l'origine stessa del messaggio; ciò significa che chiunque potrebbe, tramite tecnica di injection, inviare ed iniettare al Broker, conoscendo il nome del Topic di riferimento, messaggi di tipo publish in maniera tale che poi quest'ultimi, siano inoltrati senza alcun controllo sull'autenticità, a tutti i differenti subscriber presenti e associati a quel Topic di riferimento.
Ovviamente, il tutto può essere facilmente riportato al nostro caso di interesse: di fatti all'attaccante basta semplicemente, tramite attacco passivo di sniffing (visto in precedenza) intercettare e memorizzarsi determinati messaggi MQTT, in maniera

tale da individuare al loro interno, l'IP associato al Broker e il nome del Topic di riferimento, rispetto al quale il Local Gateway ha richiesto precedentemente la sottoscrizione (es. test.mosquitto.org). Una volta fatto ciò, anche lui potrà inviare senza problemi, in quanto sia il Broker che il Local Gateway mancano di autenticazione, appositi messaggi di publish verso il Broker e il Topic precedentemente scoperto, con dentro il payload contente dati da lui scelti e fissati, aggiornando così in maniera illegittima e falsificata i dati memorizzati e presenti nella risorsa sensori, senza che il Local Gateway possa accorgersi della sua falsa identità

Una volta che l'attaccante ha ottenuto le due info necessarie per eseguire l'injection, ovvero l'IP del Broker e il nome del Topic associato alla risorsa da "avvelenare", molto semplicemente gli basta creare un JSON dello stesso formato richiesto e gestito dal Local Gateway (anche questa info la ottiene tramite sniffing passivo) e poi generando un nuovo messaggio di tipo publish e inserendo il contenuto dello stesso JSON come payload al suo interno, lo andrà infine ad inviare al Broker, che ignaro di tutto lo reinoltrerà a tutti i subscriber di quel topic, tra cui il Local Gateway, che, una volta ricevutolo, ovviamente andrà a supporre, senza alcun controllo di autenticazione, che il messaggio sia stato inviato legittimamente dal dispositivo IoT. La questione dell'autenticazione è cruciale all'interno di un ambiente applicativo quale è la smart grid. Di fatti, bisogna essere in grado di risalire sempre all'entità che ha compiuto una determinata azione e bisogna sempre essere in grado di discernere tra nodi legacy e nodi malevoli/non legacy. È, dunque, necessaria l'implementazione di un meccanismo di autenticazione basato, nel sistema da noi implementato, sull'utilizzo della crittografia a chiave pubblica per poter applicare la firma digitale e lo scambio sicuro di chiavi di cifratura. La firma digitale è un meccanismo aggiuntivo che ci permette, inoltre, di garantire autenticità e integrità al contempo. Chiaramente per tale meccanismo è necessario l'utilizzo di una opportuna e fidata Certificate Authority.

PROGETTAZIONE DI ALCUNE FEATURES DI SICUREZZA NELLA COMUNICAZIONE M2M NANOGRID– ENERGY-GATEWAY

Una volta terminata la fase descrittiva su quelle che sono le principali vulnerabilità e tipologie di attacco basate sull'errata progettazione architetturale a livello protocollare della versione di base del sistema, è opportuno andare a soffermarsi ed analizzare quelle che sono le principali tipologie di contromisure ed estensioni in termini di sicurezza che possono essere integrate all'interno della rete stessa, affinché possano essere correttamente mitigate ed eliminate tutte le possibili minacce precedentemente descritte. Ciò implica che, in questo capitolo, verranno affrontati ed analizzati differenti e variegati meccanismi estensivi che garantiranno l'aggiunta di tutte quelle features di sicurezza orientate a garantire confidenzialità, integrità, autenticazione e disponibilità nei messaggi scambiati tra le diverse entità interne alla rete, features che come visto in precedenza, non risultano essere presenti all'interno dei tre principali protocolli applicativi utilizzati, ovvero CoAP, MQTT e HTTP.

1.33 HTTP Security

Dopo aver analizzato nel dettaglio le diverse problematiche e mancanze in termini di sicurezza che vengono riscontrate all'interno del protocollo HTTP, è opportuno cercare di individuare e definire le principali contromisure che possono essere adottate al fine sopperire a tali pecche progettuali, facendo sì che il protocollo stesso e quindi le componenti del sistema che lo sfruttano per comunicare tra loro, risultino essere sicuri in termini di confidenzialità, integrità e autenticazione durante la trasmissione dei loro messaggi informativi.
Proprio per questo motivo si è deciso di integrare, a livello protocollare nello stack di riferimento, il layer applicativo relativo al protocollo HTTP con un layer di intermediazione sicuro con lo strato protocollare di livello trasporto (TCP), prendente il nome di SSL/TLS Layer:

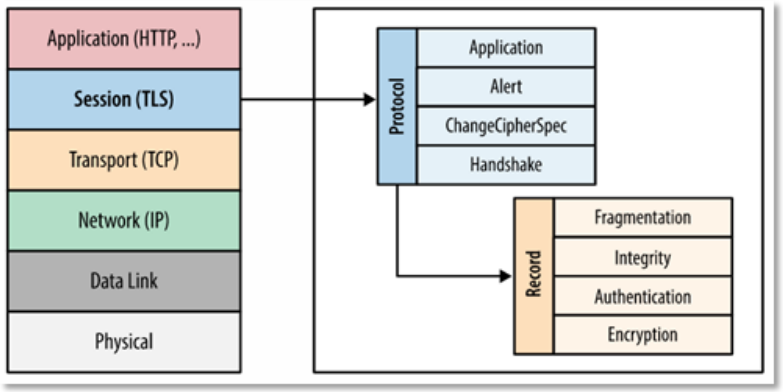

Figura 18. HTTPS protocol

Senza scendere troppo nei dettagli implementativi e progettuali, come si evince dalla Figura 18, lo scopo che si pone di compiere tale nuovo strato intermedio è semplicemente quello di prendere il pacchetto HTTP proveniente dal livello applicativo sovrastante, ed andare a manipolarlo e modificarlo al fine di aggiungere tutte quelle features di sicurezza che ad esso mancano; successivamente, il pacchetto ottenuto sarà inoltrato tramite un canale TCP precedentemente creato, che quindi trasmetterà i dati in assoluta sicurezza. Quello che in genere si ottiene da tale combinazione di strati protocollari, viene in gergo chiamato col termine di HTTPS (Hypertext Transfer Protocol Secure). Tra le principali caratteristiche di sicurezza che il protocollo TLS/SSL può offrire agli strati protocollari superiori, sicuramente troviamo aspetti legati all'integrità dei dati in transito, alla loro confidenzialità e anche all'autenticazione della loro origine di provenienza; il tutto

ottenuto sfruttando particolari meccanismi crittografici sia simmetrici che asimmetrici che vengono scelti da una suite di protocolli che client e server condividono: difatti, il protocollo TLS, può essere internamente suddiviso in due sottofasi protocollari, di cui la prima prende il nome di handshake mentre la seconda di transfer record. Ciò che accade all'interno della prima delle due fasi precedentemente introdotte, come indica il nome stesso, consiste nel mettere d'accordo client e server sul tipo di cifrari da utilizzare per comunicare tra loro nella successiva fase al fine di garantire confidenzialità e integrità (qui vengono scelti principalmente cifrari simmetrici per la confidenzialità e MAC per l'integrità), facendo sì che essi possano anche scambiarsi i certificati relativi alle loro rispettive chiavi pubbliche (crittografia asimmetrica), al fine di dimostrare la corretta identità l'uno dell'altro (questo processo di autenticazione può essere mutuo, ovvero entrambi si devono autenticare, o singolo, ovvero solo il server deve autenticarsi al client). Una volta terminata la fase di handshake, client e server possono comunicare tra loro in maniera sicura e protetta, sfruttando le scelte eseguite durante la fase precedente, prendendo i messaggi applicativi HTTP e cifrandoli più volte secondo i criteri scelti, prima di inoltrarli all'interno del canale TCP creatosi sempre durante la fase di handshake (N.B Il protocollo TLS si basa a livello trasporto sul protocollo TCP, definito in funzione della porta 443).
Detto ciò, per rendere sicura la trasmissione dei dati applicativi HTTP, sia per quanto riguarda la comunicazione che si instaura tra il server remoto e i client e sia per quanto riguarda la connessione stabilita tra il Local Gateway e lo stesso server remoto, abbiamo deciso di integrare nel nostro sistema, dal punto di vista architetturale e protocollare, lo strato SSL/TLS al fine di ottenere l'ausilio dell'intero protocollo HTTPS citato in precedenza.

1.34 COAP Security

Così come è accaduto per il protocollo HTTP, anche per il protocollo CoAP, dopo aver analizzato nel dettaglio le diverse problematiche e mancanze in termini di sicurezza che vengono riscontrate al suo interno, è opportuno cercare di individuare e definire le principali contromisure che possono essere adottate al fine sopperire a tali pecche progettuali, facendo sì che il protocollo stesso e quindi le componenti del sistema che lo sfruttano per comunicare tra loro, risultino essere sicuri in termini di confidenzialità, integrità e autenticazione durante la trasmissione dei loro messaggi informativi. Come verrà descritto ed analizzato successivamente, in questo caso le possibili tecniche di mitigazione e le potenziali contromisure da adottare che possono essere prese al fine di garantire il raggiungimento di un buon livello di sicurezza interno al protocollo CoAP sono molteplici; noi partiremo analizzando l'elemento che meglio si avvicina a ciò che è stato già visto in precedenza con l'introduzione di HTTPS, prendente in questo caso il nome di DTLS.

1.34.1 DTLS

Come già accennato in precedenza, il protocollo CoAP può essere visto dal punto di vista architetturale e progettuale come una versione rivalutata e più "light" del protocollo HTTP; ciò quindi porta immediatamente a pensare che, un primo modo per rendere sicuro il protocollo CoAP, sia seguire lo stesso sentiero affrontato precedentemente con il protocollo HTTP, integrando uno strato protocollare aggiuntivo tra il livello applicativo e quello trasporto che aggiungesse le features di sicurezza mancanti in origine al protocollo stesso; tutto ciò quindi sembra condurre a pensare di utilizzare come strato intermediario tra CoAP (livello applicativo) e il livello trasporto, la stessa suite di protocolli crittografici SSL/TLS che ci vengono messi a disposizione e che abbiamo appositamente descritto in precedenza, legandoli al protocollo HTTP. Qui però, necessariamente sorge un enorme problema: difatti la più grande differenza che intercorre tra il protocollo CoAP e il protocollo HTTP è che, a livello trasporto, utilizzano due protocolli completamente differenti, in quanto HTTP si basa sul protocollo TCP legato alla porta standard 80, mentre CoAP si basa sul protocollo UDP legato principalmente, nella sua versione originale, alla porta 5683. Ciò implica che è propriamente impossibile adattare la suite protocollare SSL/TLS all'interno del contesto protocollare del CoAP, in quanto i protocolli di intermezzo definiti e presenti al suo interno, necessitano di avere un canale TCP aperto e inerente alla porta 443 per poter sia eseguire le operazioni legate all'handshake, sia per successivamente garantire il corretto scambio dei dati cifrati tra le due entità coinvolte.
Proprio per sopperire a tale mancanza esecutiva, è stata successivamente sviluppata e progetta una nuova suite protocollare molto più adattabile al contesto definito in precedenza, prendente il nome di DTLS: il termine DTLS sta per Datagram

Transport Layer Security che come indica il nome stesso, non è altro che una suite di protocolli che si pone lo scopo di garantire l'aggiunta e il raggiungimento di un certo livello di sicurezza in termini di confidenzialità, integrità e autenticazione (quindi per fronteggiare attacchi di eavesdropping, tampering o message forgery) per tutti quei protocolli applicativi che sfruttano a livello trasporto, proprio il protocollo UDP, cercando ovviamente di raggiungere lo stesso livello di sicurezza che SSL/TLS offre ai processi applicativi basati su TCP.
Anche qui, senza scendere troppo nei dettagli implementativi di tale suite, il suo funzionamento è molto similare a quello che è stato già ampiamente descritto con la suite inerente a SSL/TLS: difatti anch'esso si basa prima su una fase di handshake, all'interno della quale le due entità coinvolte si mettono d'accordo in riferimento al tipo di cifrari e meccanismi crittografici da utilizzare per la fase successiva, per poi autenticarsi attraverso l'uso di meccanismi asimmetrici e dei corrispettivi certificati; la fase successiva invece, come accade per SSL/TLS si occupa semplicemente di gestire la corretta cifratura ed invio dei dati, usando i meccanismi crittografici e le informazioni ottenuti al passo precedente.
Ovviamente quindi da ciò, si può facilmente capire e derivare come, un primo meccanismo di sicurezza aggiuntivo che può essere associato al protocollo CoAP, in quanto utilizzante alla base il protocollo trasporto UDP, è sicuramente rappresentato dalla possibilità di integrare proprio tale nuova suite protocollare appena descritta:

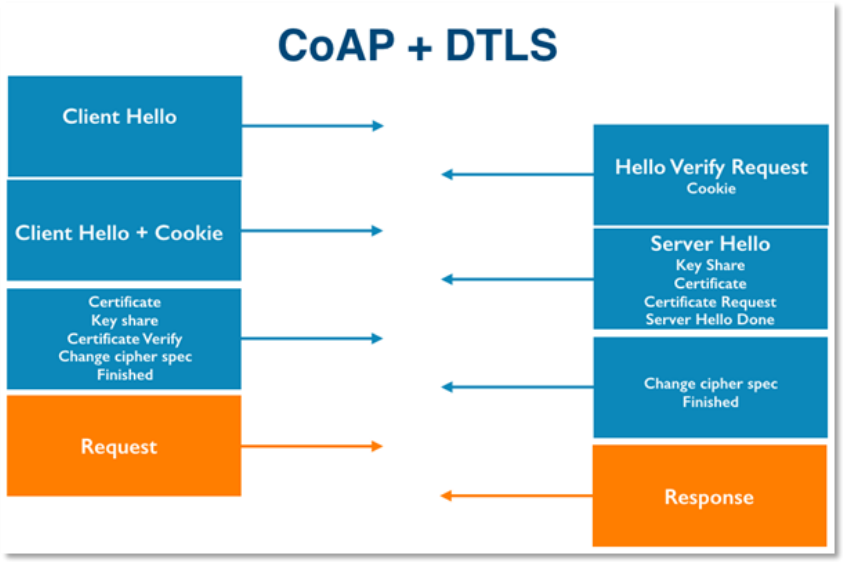

Figura 19. CoAP con DTLS

È stato utilizzato all'interno della fase di handshake del protocollo DTLS, come meccanismo di autenticazione asimmetrico, l'RSA, mentre nel secondo caso di interesse è stato utilizzato come meccanismo asimmetrico, quello basato sull'utilizzo delle curve ellittiche chiamato ECDHE/ECDSA (Elliptic Curve Diffie-Hellman Ephemeral).

1.35 MQTT Security

Così come è accaduto per il protocollo HTTP e per il protocollo CoAP, anche per il protocollo MQTT, dopo aver analizzato nel dettaglio le diverse problematiche e mancanze in termini di sicurezza che vengono riscontrate al suo interno, è opportuno cercare di individuare e definire le principali contromisure che possono essere adottate al fine sopperire a tali pecche progettuali, facendo sì che il protocollo stesso e quindi le componenti del sistema che lo sfruttano per comunicare tra loro, risultino essere sicuri in termini di confidenzialità, integrità e autenticazione durante la trasmissione dei loro messaggi informativi. Come verrà descritto ed analizzato successivamente, anche in questo caso come nei precedenti analizzati, le possibili tecniche di mitigazione e le potenziali contromisure da adottare per garantire il raggiungimento di un

buon livello di sicurezza interno al protocollo MQTT sono molteplici; noi partiremo analizzando un elemento protocollare già visto in precedenza, che è strettamente legato al protocollo HTTPS, ovvero lo strato crittografico intermedio chiamato **TLS** (è importante notare come in questo caso, a differenza del caso CoAPS, è necessario usufruire di tale componente crittografica, piuttosto che di DTLS, poiché MQTT si basa a livello trasporto, sul protocollo TCP).

1.35.1 TLS

Senza scendere troppo nei dettagli implementativi dello strato crittografico intermedio **TLS** possiamo subito passare ad analizzare a livello pratico ed implementativo, come è stato possibile integrare i servizi offerti da tale strato, al fine di garantire il raggiungimento di un ottimo livello di sicurezza in termini di integrità, confidenzialità, autenticazione e disponibilità per quanto riguarda il processo trasmissivo dei dati, attraverso l'utilizzo dello stesso protocollo MQTT; prima però di scendere nel dettaglio, è opportuno sottolineare come in questo caso, è stato necessario gestire l'implementazione di un Broker MQTT locale, al fine di riuscire ad avere un controllo in termini di configurazione maggiore così da riuscire correttamente ad impostare a mano, tutti i diversi parametri ed elementi necessari per garantire l'instaurazione di una sessione TLS tra dispositivo IoT e Broker e tra Broker e Local Gateway: più precisamente nel nostro caso, abbiamo deciso di utilizzare e configurare due differenti tipologie di Broker MQTT locali, uno Mosquitto Broker e l'altro HiveMQ Broker.

Per poter rendere la comunicazione sicura sia tra Publisher e Broker che tra Broker e Subscriber, siamo andati ad utilizzare un ulteriore livello di sicurezza, instaurando invece che una semplice e standard comunicazione TCP tra le due entità coinvolte, un canale comunicativo TLS over TCP.

Al fine di realizzare ciò, *Mosquitto*, permette di instaurare comunicazioni su una porta TCP differente da quella standard, ovvero la 8883: per poterlo fare però bisogna leggermente modificare il file di configurazione *mosquitto.conf*. Prima di passare a definire quest'ultimo, è necessario eseguire due fondamentali operazioni primordiali, attraverso l'ausilio di openssl:
1. Creare una chiave privata per quella che sarà la nostra CA Locale (CA Key) e generare attraverso di essa un certificato pubblico (CA Cert) (self-signed).
2. Creare un certificato per il broker Mosquitto MQTT (Server Cert) utilizzando una chiave privata diversa da quella precedente e firmato digitalmente dalla nostra CA.

1.36 Sicurezza e crittografia a livello applicativo

Nella prima versione sicura del nostro sistema descritta ampiamente nei paragrafi precedenti, con tutte le sue varie problematiche e la sua architettura molto ricca, emerge come le features di sicurezza introdotte col fine di garantire il raggiungimento dell'integrità, della confidenzialità, dell'autenticazione e della disponibilità dei messaggi trasmessi tramite il principale utilizzo del protocollo MQTT e/o CoAP, risultano essere troppo dispendiose e poco performanti in termini prestazionali, specialmente per quanto riguarda le ristrette e limitate componenti e caratteristiche fisiche ed esecutive che i dispositivi constrained come quelli IoT presentano avere (l'inserimento e l'utilizzo di strati protocollari intermedi come DTLS e TLS è quindi troppo dispendioso in termini computazionali in tale circostanza). Per cercare quindi di ridurre la complessità dal punto di vista delle dipendenze da altri strati software intermediari nonché tale problematica prestazionale, mirando però al contempo stesso a mantenere lo stesso ottimo livello in termini di sicurezza raggiunto in precedenza, si è deciso di abbandonare l'utilizzo di strati protocollari intermedi che mettessero a disposizione le features crittografiche richieste, che comunque definivano un contorno molto ampio e vasto di proprietà e si è invece cercato di riprodurre il tutto, in maniera più efficiente e meno dispendiosa possibile, andando a lavorare direttamente a livello applicativo. Tali migliorie ovviamente derivano da un minor consumo di banda dovuto al minor numero di messaggi necessari per rendere la comunicazione sicura e dalla garanzia di poter accedere ai dispositivi effettuando piccole preconfigurazioni a monte come ad esempio la definizione di chiavi asimmetriche o segrete condivise staticamente a priori.

Inoltre, sono state realizzate delle mitigazioni per risolvere sia il problema del MiTM, già trattato precedentemente, che prevedeva l'uso della tecnica dell'ARP Poisoning, e sia eventuali attacchi a replay che potessero essere eseguiti, utilizzando un fissato intervallo di validità. Proprio col fine di garantire il raggiungimento di tali obbiettivi in ottica sicurezza, sono stati definiti tre diversi protocolli crittografici applicativi che sfruttano le seguenti combinazioni di metodologie crittografiche, sia per CoAP che per MQTT, al fine di garantire un corretto trade-off tra un buon livello di sicurezza (confidenzialità, integrità e autenticazione) e di performance [65]:

- **Crittografia Simmetrica**: approccio puramente simmetrico sfruttando una chiave precondivisa (PSK) e segreta;
- **Crittografia Asimmetrica**: approccio puramente asimmetrico con scambio della chiave pubblica e dei dovuti certificati a monte;
- **Crittografia Asimmetrica + Crittografia Simmetrica**: approccio misto che prevede la conoscenza delle chiavi pubbliche asimmetriche con tanto di certificati, le quali vengono usate per garantire una corretta generazione e scambio di una successiva chiave simmetrica di sessione.

Si è poi scelto di utilizzare un certo insieme di algoritmi crittografici simmetrici e asimmetrici e modi operativi per raggiungere le finalità di sicurezza:

- **AEAD-AES**: meccanismo di crittografico che utilizza AES come algoritmo simmetrico;
- **RSA**: utilizzato come cifrario asimmetrico sia per lo scambio delle chiavi simmetriche che per la cifratura/decifratura di interi messaggi;
- **DSA**: utilizzato come algoritmo di firma digitale nell'ambito della crittografia asimmetrica;
- **ECIES**: utilizzato come cifrario asimmetrico basato sul meccanismo delle Curve Ellittiche, sia per lo scambio delle chiavi simmetriche che per le cifratura/decifratura di interi messaggi;
- **ECDSA**: utilizzato come algoritmo di firma digitale basato su Curve Ellittiche (analogo di DSA) nell'ambito della crittografia asimmetrica

1.36.1 Protocollo Crittografico Simmetrico

È noto come la crittografia simmetrica rispetto alla corrispettiva crittografia asimmetrica, risulti essere molto più performante in termini di complessità temporale e spaziale, soprattutto considerando la dimensione dei messaggi da cifrare/decifrare come molto elevata: proprio per tale motivo, il protocollo seguente è stato definito per delineare le modalità algoritmiche e strutturali con cui l'IoT Device e il Gateway si scambieranno i dati ambientali in maniera completamente sicura, garantendo un approccio molto leggero in termini prestazionali, alla comunicazione; ovviamente però il tutto sarà gestito e garantito tramite uno scambio statico (a mano) e prefissato di una chiave simmetrica (PSK) tra gli interlocutori, e questo potrebbe essere un problema in ottica sicurezza, specialmente per un sistema a lunga durata computazionale, visto il riutilizzo continuo della stessa chiave. Uno dei meccanismi risolutori più semplici per questa problematica è sicuramente quello di far in modo che il sistemista, periodicamente, aggiorni il valore di tale chiave crittografica simmetrica. Durante le successive pagine citeremo sempre il contesto simmetrico del progetto ed in tal caso è bene ricordare che è stato definito l'utilizzo, proprio per tale scopo, del meccanismo crittografico AEAD facente utilizzo del cifrario a blocchi simmetrico AES che per brevità indicheremo come AEAD-AES.

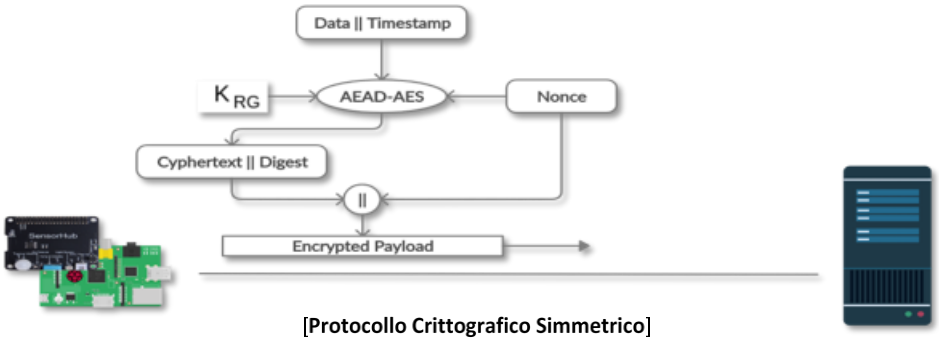

Figura 20. Protocollo Crittografico Simmetrico

La Figura 20 mostra come il messaggio da inoltrare passi una serie di fasi e di blocchi successivi al fine di riuscire a creare un payload cifrato che possa viaggiare in maniera sicura tra i due dispositivi, in termini di confidenzialità, autenticazione e integrità. Più precisamente possiamo descrivere le componenti di tale figura nella seguente legenda:

● **K_RG**: Chiave Simmetrica Segreta Condivisa tra il dispositivo IoT e il Local Gateway e utilizzata per i messaggi che vanno dal dispositivo IoT al Local Gateway (In maniera speculare esiste una chiave K_GR).

● **AEAD-AES**: Meccanismo di cifratura che sfrutta al suo interno un modo operativo a blocchi e lo stesso cifrario a blocchi AES, il tutto associato ad un particolare algoritmo MAC. A tale blocco viene dato in input in primis, un Nonce (inteso come un IV - (Initialitation Vector), (il nonce è un numero casuale o pseuda-casuale che ha un utilizzo unico) nonché successivamente ad esso viene associata la nostra chiave K_RG (o K_GR) e infine il nostro payload in chiaro. Il tutto verrà computato per generare il testo cifrato, ovvero un messaggio crittografico confidenziale e un digest per la verifica dell'integrità e dell'autenticazione dello stesso.

● **Timestamp**: utilizzato per ovviare a potenziali attacchi di replay.

Tale schema come tutte le altre varianti è utilizzato, anche per l'invio dei messaggi dal Local Gateway all'IoT Device con le opportune modifiche. Lo schema permette comunque di ottenere notevoli pregi sulle performance che dal punto di vista teorico dovrebbe essere molto più efficiente degli altri anche se ci sono da considerare alcune carenze che lo rendono vulnerabile sul lungo tempo. In particolare, lo schema garantisce la possibilità di cifrare più messaggi utilizzando lo stesso valore segreto e condiviso (chiave simmetrica *K_RG* o *K_GR*), facendo solamente variare il Nonce ad ogni invio di un nuovo Messaggio. C'è da chiarire comunque che tra le varie caratteristiche di tale protocollo sussiste la necessità di uno scambio a monte in maniera sicura e «a mano» (ovvero tramite configurazione statica) della chiave segreta condivisa tra i due interlocutori e questo, insieme all'utilizzo di un Nonce, che cambierà da messaggio a messaggio ovviamente, può portare a successive problematiche future in ottica di sicurezza: tale problematica deriva dal fatto che l'utilizzo di una chiave statica con un Nonce, il quale può assumere un valore in un intervallo di numeri finito, seppure grande in generale, e vista la necessità del cambio del Nonce ad ogni nuovo messaggio cifrato per mantenere difficile il cracking esterno della chiave, mi obbliga ad un determinato istante futuro della comunicazione, ad azzerare il Nonce stesso portandomi dunque a cifrare, da quel momento in poi, ogni nuovo messaggio, utilizzando un valore Nonce già usato in precedenza, associato con la stessa chiave simmetrica. L'attaccante quindi allora, potrebbe riuscire a risalire statisticamente al valore della chiave segreta e condivisa, intercettando messaggi diversi ma cifrati con lo stesso valore segreto (in realtà nel nostro caso, usando un cifrario a blocchi, il pericolo è molto ridotto). Una spinta in più che è stata data al protocollo, anche grazie all'utilizzo del Nonce ma specialmente del *timestamp* presente nel messaggio cifrato, è la possibilità di evitare gli attacchi di *replay*, utilizzando lato ricevente un intervallo di validità temporale dentro il quale vengono ritenuti validi i messaggi ricevuti (il resto verrà droppato).

1.36.2 Protocollo Crittografico Asimmetrico

A differenza dello schema simmetrico che, come già accennato in precedenza, è in generale molto più veloce dal punto di vista crittografico rispetto al corrispettivo asimmetrico, d'altra parte quest'ultimo, garantendo sempre tutte le proprietà di confidenzialità, integrità e autenticazione del caso, risulta essere comunque molto più robusto e resistente dal punto di vista del riutilizzo della stessa chiave crittografica più volte (in quanto entra in gioco la componente chiave pubblica e privata). In effetti uno schema simmetrico necessità del cambio della chiave molto più spesso, sia quando sussistono problemi relativi ad un qualsiasi scenario di sicurezza in ambito crittografico, ovvero viene ad esempio rubata la chiave segreta (che nel caso della crittografia asimmetrica è nota anche come chiave privata) e sia nel caso in cui la stessa chiave debba essere utilizzata più volte a causa dell'azzeramento del nonce. Quest'ultimo caso ovviamente non è un problema per un cifrario asimmetrico, in quanto non sussiste la necessità di combinare una chiave con un valore cangiante, e né tanto meno risulta essere necessario non poter utilizzare la stessa chiave per cifrare messaggi differenti (altrimenti ad ogni comunicazione sarebbe necessario cambiare le chiavi asimmetriche e ciò risulterebbe essere molto oneroso). Di seguito viene definito il protocollo asimmetrico generalizzato utilizzato:

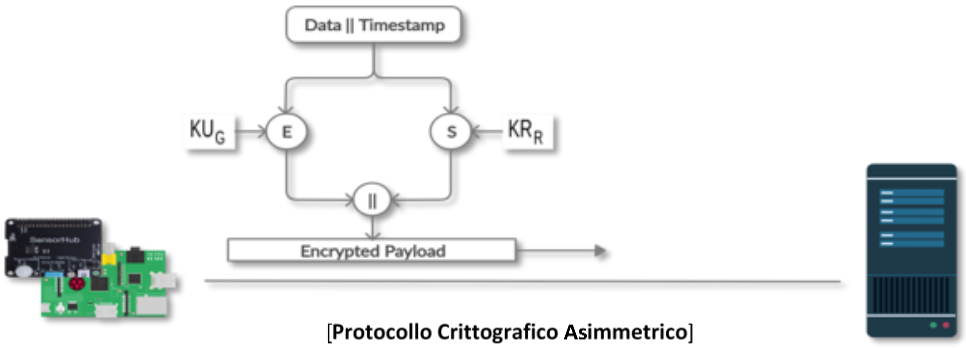

Figura 21. Protocollo Crittografico Asimmetrico

L'algoritmo risulta molto semplice da analizzare dal punto di vista delle componenti ed in particolare ritroviamo:

- **KU_G**: Chiave Pubblica del Gateway utilizzata per garantire che solo il Local Gateway utilizzando la propria chiave privata possa riuscire a decifrare l'informazione.
- **E**: Algoritmo Asimmetrico (RSA o ECIES) utilizzato per ottenere Confidenzialità cifrando l'informazione utilizzando la chiave KU_G.
- **KR_R**: Chiave Privata del dispositivo IoT (raspberry) utilizzata per garantire che sia stato lui il firmatario del messaggio.
- **S**: Algoritmo di Firma Digitale (DSA o ECDSA) per garantire integrità e autenticazione del messaggio. Ovviamente il Local Gateway utilizzerà il corrispettivo algoritmo di Verifica per verificare tramite la Chiave Pubblica dell'IoT Device che sia stato lui a firmare quel messaggio (con i dovuti certificati X.509 per attestarne l'autenticità).
- **Timestamp**: utilizzato per ovviare ad attacchi di replay definendo un intervallo di validità temporale.

Lo schema asimmetrico, come già detto in precedenza, non soffre del problema legato al riuso del Nonce presente invece nello Schema Simmetrico, non vincolando quindi il sistemista ad un cambio di chiavi periodico necessario per ovviare classici problemi di sicurezza e dunque perciò può essere ritenuto molto più robusto e stabile dello schema precedente. Ovviamente in tale schema non si considera come modalità di scambio delle chiavi asimmetriche, quello basato sulla eventuale presenza di *Pagine Gialle* o *Repository* dove pubblicare le proprie credenziali pubbliche remotamente, in quanto in quest'ultimo, come anche nei rimanenti schemi, essendo un contesto applicativo molto ristretto porterebbe solamente un aumento inutile di complessità il protocollo stesso, senza ottenere vantaggi congrui ad una rete di sensori che già di per sé è limitata nelle risorse. Dunque, tale schema presuppone lo scambio a monte in maniera sicura e "a mano" delle chiavi pubbliche coi

rispettivi certificati di validità tra i due interlocutori, che garantirà di ottenere maggiore confidenzialità, integrità e autenticazione, rispetto all'approccio simmetrico, a scapito ovviamente di un degrado prestazionale, dipendente dalla dimensione del messaggio da cifrare. Come anche descritto nel precedente schema, l'utilizzo del Timestamp e la definizione di intervalli di validità dei messaggi, consente di evitare possibili attacchi di *replay*.

1.36.3 Protocollo Crittografico Asimmetrico + Simmetrico

Tale protocollo cerca di cogliere tutti i vantaggi dei due approcci precedenti, eliminando i difetti che entrambi gli schemi presi singolarmente presentano. Ovviamente la necessità del legame dei due approcci deriva da una combinazione ben nota in letteratura e utilizzata oramai ovunque in ambito crittografico: in effetti, la loro combinazione porta da un lato ad avere performance migliori rispetto al puro schema asimmetrico, tendendo ad una complessità similare a quella dell'approccio puramente simmetrico a meno di un delta piccolo, e dall'altro otteniamo delle garanzie di robustezza che sul lungo tempo non presentando avere gli stessi problemi legati allo schema puramente simmetrico. In particolare, entrambi gli approcci hanno uno scopo nel loro uso, ed in particolare la parte asimmetrica viene utilizzata per lo scambio di una chiave segreta, simmetrica e condivisa (chiave di sessione) tra i due interlocutori mentre l'approccio simmetrico verrà successivamente utilizzato per aumentare le prestazioni cifrando il testo in chiaro in maniera molto veloce con il meccanismo dell'AEAD utilizzante AES. Vediamo di seguito la struttura di tale protocollo utilizzato:

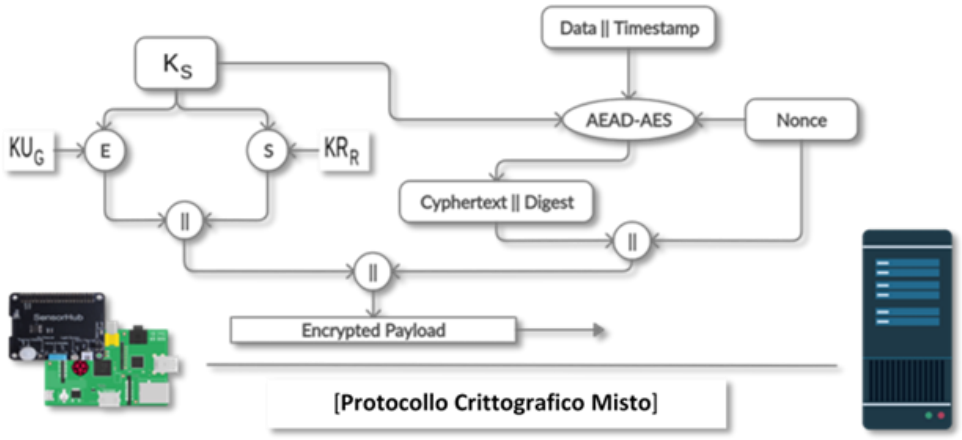

Figura 22. Protocollo Crittografico Misto

Si può evincere come lo schema della Figura 22, rispetto allo schema precedente, risulta essere molto più complesso nella sua interezza. In ogni caso non tutto lo schema è utilizzato durante l'intera comunicazione, ma alcune parti solo in alcuni istanti temporali col fine di ottenere determinate finalità.
In particolare, prima di tutto definiamo la semantica delle componenti dello schema:

- **KU_G**: Chiave Pubblica del Gateway
- **E**: Algoritmo Asimmetrico (RSA o ECIES) utilizzato per ottenere Confidenzialità.
- **KR_R**: Chiave Privata del dispositivo IoT (raspberry).
- **S**: Algoritmo di Firma Digitale (DSA o ECDSA) per garantire Integrità e Autenticazione.
- Timestamp: utilizzato per ovviare ad attacchi di replay definendo un intervallo di validità temporale.
- **K_RG**: Chiave Simmetrica Segreta Condivisa tra il dispositivo IoT e il Local Gateway e utilizzata per i messaggi che vanno dal dispositivo IoT al Local Gateway (In maniera speculare esiste una chiave K_GR).
- **AEAD-AES**: Meccanismo di cifratura che sfrutta al suo interno un modo operativo a blocchi e lo stesso cifrario a blocchi AES, il tutto associato ad un particolare algoritmo MAC. A tale blocco viene dato in input in primis, un Nonce

(inteso come un IV), nonché successivamente ad esso viene associata la nostra chiave K_RG (o K_GR) e infine il nostro payload in chiaro. Il tutto verrà computato per generare il testo cifrato, ovvero un messaggio crittografico confidenziale e un digest per la verifica dell'integrità e dell'autenticazione dello stesso.

- **Timestamp**: utilizzato per ovviare a potenziali attacchi di replay

Come già accennato, tale approccio Misto non sfrutta sempre e comunque la sua intera complessità, ma ad esempio, viene utilizzata la parte asimmetrica solo durante lo scambio della chiave segreta (simmetrica) e di sessione tra i due interlocutori. Dunque, a parte durante il primo messaggio della sessione e/o dopo tutti quelli necessari allo scambio di una nuova chiave simmetrica dovuto all'azzeramento del Nonce, verrà utilizzato solamente il pezzo puramente simmetrico. Questo produce dunque tutti gli effetti desiderati che stavamo cercando. Ovviamente tale beneficio risolve il problema legato all'azzeramento del Nonce dello Schema Simmetrico, ed inoltre non presuppone lo scambio a monte di chiavi segrete condivise tra i due interlocutori a meno delle relative chiavi pubbliche asimmetriche necessarie all'instaurazione della comunicazione. Infine, possiamo affermare che tale schema risulta essere molto più robusto dal punto di vista della confidenzialità, integrità e autenticazione rispetto ad entrambi gli schemi precedenti, anche se in certe circostanze può perdere leggermente in termini prestazionali. Inoltre, come accaduto con gli altri due schemi precedenti, osserviamo sempre la presenza del *Timestamp* che garantisce la mitigazione agli attacchi di *replay*.
Dal punto di vista applicativo sono state utilizzate due principali librerie per Python sia per l'ambito simmetrico, che per quello asimmetrico senza curve ellittiche, ovvero *pycrypto* e una sua estensione *pycrytpodome*, che introducono modi operativi ed algoritmi moderni con delle performance e proprietà di sicurezza più robusti rispetto a quelli classici. In particolare, tali migliorie si hanno soprattutto dal punto di vista degli approcci simmetrici, dove come è ben conosciuto, i modi operativi classici come **CBC (Cipher-Block-Chaining)** forniscono garanzie solo dal punto di vista della riservatezza del messaggio ma non sulla sua integrità. In altre parole, non consentono al destinatario di stabilire se il testo cifrato è stato modificato durante la comunicazione o se proviene realmente da una determinata fonte mittente. In genere infatti tali meccanismi dovevano essere contornati dall'utilizzo di un algoritmo **MAC (message authentication code)** quale **HMAC (keyed-hash message authentication code)** (HMAC è una modalità per l'autenticazione di messaggi basata su una funzione di hash, utilizzata in diverse applicazioni legate alla sicurezza informatica.) per fornire anche integrità e autenticazione ai messaggi in transito, abbisognando tuttavia di un ulteriore chiave per ambedue le direzioni in cui si svolge la comunicazione. Proprio per questo motivo, il modo operativo scelto da noi per quanto riguarda la realizzazione del progetto, è noto come **AEAD (Authenticated Encryption with Associated Data)** ed è utilizzato per combinare meccanismi crittografici che garantiscono confidenzialità a meccanismi interni garantenti dall'altra parte l'integrità e l'autenticazione degli stessi.

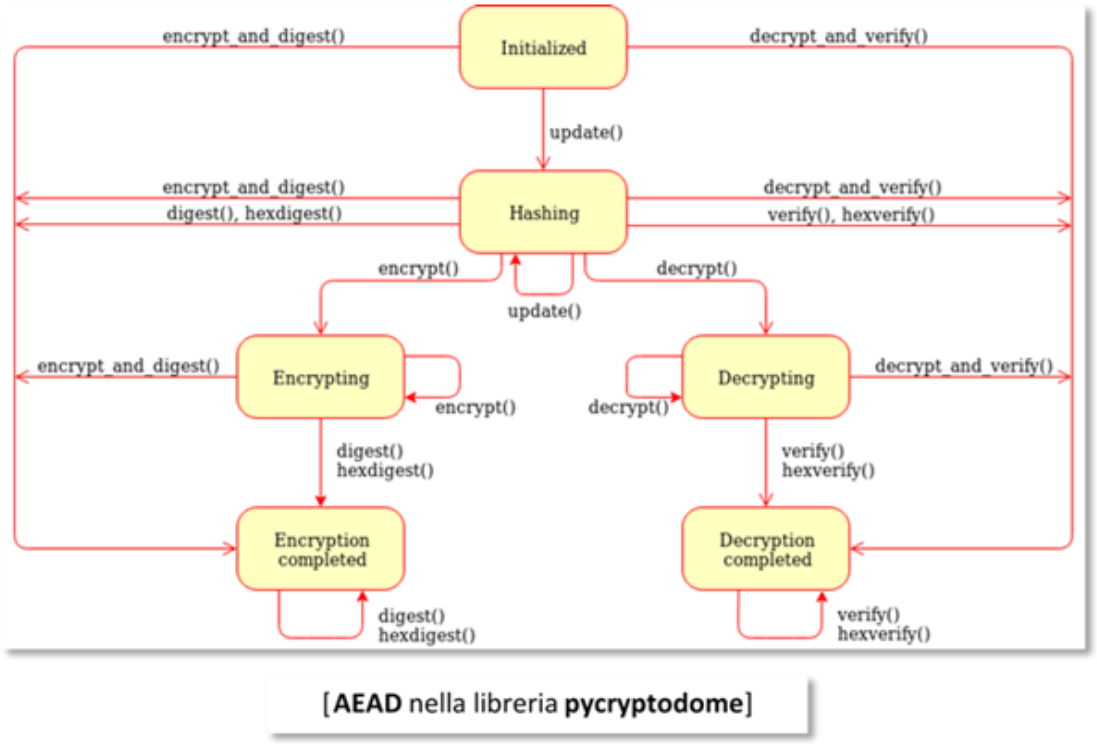

Figura 23. AEAD (Authenticated Encryption with Associated Data)

Questo schema (Figura 23) mostra come internamente funziona questo nuovo modo operativo scelto: osservando attentamente si riesce a comprendere come tramite un singola primitiva ***encrypt_and_digest()***, si riesce ad ottenere sia il testo cifrato che il relativo *Digest* per garantire le nostre proprietà ricercate. Questo modo operativo, inoltre, permette di definire anche un algoritmo crittografico interno facente parte di un particolare cifrario associandolo con un algoritmo MAC interno: in tale caso abbiamo scelto **GCM (Galois Counter Mode)** come modo operativo per l'encryption e AES (Andvanced Encryption Standard) a 256 bit. Inoltre, come algoritmo MAC, GCM fornisce internamente quello che è conosciuto come **GMAC (Galois Message Authentication Code)**.
Questo ovviamente, dal punto di vista puramente simmetrico, mentre dal punto di vista asimmetrico dobbiamo distinguere tra algoritmi scelti per la crittografia standard e quella basata su curve ellittiche. In particolare, sulla crittografia standard abbiamo utilizzato RSA a 2048 bit e DSA a 2048 bit (con ovviamente la dimensione della chiave puramente modificabile).
Più precisamente, per quanto concerne la crittografia asimmetrica basata su curve ellittiche abbiamo invece dovuto utilizzare due librerie separate da *pycrypto* e *pycryptodome*, in quanto quest'ultime non fornisco un grosso supporto per la crittografia ellittica ma forniscono solo delle primitive basilari per definire una curva ed effettuare delle operazioni basilari su tali punti quali somma tra due punti o moltiplicazione di uno scalare per un punto della curva, dunque non definiscono nessun algoritmo standardizzato che abbia adeguate metriche di sicurezza. Per sopperire allora a tale mancanza, si è fatto utilizzo della libreria **ECIES** ed **ECDSA** di Python per l'implementazione delle suddette curve ellittiche. In particolare, **ECIES (ECIES Hybrid Encryption Scheme)** permette l'utilizzo delle curve ellittiche per cifrare l'informazione utilizzando un approccio ibrido tra curve ellittiche e un approccio puramente simmetrico per aumentare le prestazioni e la sicurezza del sistema dal punto di vista della confidenzialità e integrità, mentre **ECDSA (Elliptic Curve Digital Signature Algorithm)** viene utilizzato come algoritmo su curve ellittiche per ottenere l'autenticazione.

1.36.4 Analisi e studio delle prestazioni delle implementazioni di sicurezza

In conclusione, mettiamo in evidenza quello che è l'aspetto prestazionale delle diverse metodologie implementate per quanto concerne l'aspetto della sicurezza. Essendo "constrained" nelle smart grid "constrained" è opportuno analizzare gli aspetti relativi al loro effettivo consumo energetico e delle differenti risorse fisiche presenti al loro interno, come quelle di rete: proprio al fine di raggiungere tale obbiettivo, per ognuna delle versioni implementate, sono stati eseguiti differenti *test pratici*, i quali hanno cercato principalmente di mettere in evidenza, in termini di consumo, le seguenti *risorse* mostrate nella seguente Tabella 2. All'interno della tabella, sono contenute per ogni versione, i rispettivi valori misurati: **AVG SIZE Frame** rappresenta la media campionata della dimensione in *bytes* del *frame* che viene immesso sulla rete, contenente il *payload* con i *dati* atmosferici *generato* dal *Sensor Hub* ed inoltrato verso il *Local Gateway;* **AVG SIZE Data** rappresenta la media campionata della dimensione in *bytes* del *messaggio a livello applicativo* che viene immesso sulla rete, contenente il *payload* con i *dati* atmosferici *generato* dal *Sensor Hub* ed inoltrato verso il *Local Gateway;* **AVG SIZE Certificato** rappresenta la media campionata della dimensione in *bytes* del *frame* che viene immesso sulla rete, contenente il *certificato* sia del *Dispositivo IoT* che del *Local Gateway*, trasmessi reciprocamente durante la fase di *hand-shake di TLS/DTLS;* **Key SIZE** rappresenta la dimensione in *bits* della *chiave simmetrica o asimmetrica* che è stata necessaria utilizzare all'interno del *Dispositivo IoT* per quella versione del sistema di riferimento*;* **AVG Latency** rappresenta la latenza media calcolata, dove la singola latenza per campione è stata misura come *somma tra l'intervallo di tempo necessario ad inviare il messaggio contenente il payload dal Dispositivo IoT verso il Local Gateway e l'intervallo di tempo necessario affinchè la risposta di quest'ultimo arrivi allo stesso dispositivo.* Per quanto concerne il primo parametro prestazionale, risulta esser chiaro come all'aumentare della dimensione della chiave la dimensione e variando la tecnica crittografica utilizzata per la generazione della firma digitale aumenti o diminuisca proporzionalmente la dimensione media del pacchetto. Il tutto ovviamente ha una diretta ripercussione sulla dimensione media dei dati, ossia il campo **SIZE DATA**. Per quanto riguarda la dimensione media del certificato essa risulta essere direttamente proporzionale alla dimensione della chiave pubblica per cui si genera e richiede il relativo certificato. È stato riscontrato, inoltre, come all'aumentare della dimensione delle chiavi di cifratura aumenti ulteriormente anche l'overhead prestazionale su CPU e RAM dei dispositivi coinvolti. È stato condotto anche un test sui cifrari a flusso i quali mostrano un tempo di cifratura ridotto a discapito, però, del consumo energetico che risultava essere troppo eccessivo e per tale ragione non sono stati più considerati nell'implementazione. La scelta alla fine è ricaduta sul cifrario AES, scelta con alla base un opportuno e ragionato **trade off** tra prestazioni, standardizzazione della tecnica crittografica e sicurezza. Per quanto riguarda la latenza, essa rispecchia perfettamente i risultati prestazionali noti dalla teoria. Il protocollo MQTT, di fatti, è ritenuto in letteratura più **lightweight** rispetto al protocollo CoAP. Questo si traduce anche quando si aggiungono le primitive crittografiche. Di fatti, il protocollo MQTT risulta mostrare una latenza di comunicazione ridotta rispetto al CoAP proporzionalmente, però, alle tecniche crittografiche applicate. Come si può facilmente intuire dalla tabella, infatti, le tecniche crittografiche che fanno uso delle curve ellittiche determinano, essendo notoriamente le tecniche più leggere nell'ambito della crittografia a chiave pubblica, una latenza minore. Sottolineiamo che le curve ellittiche scelte per le nostre valutazioni sono state le seguenti:

- brainpoolP160r1
- brainpoolP192r1
- brainpoolP256r1
- brainpoolP384r1

Rispettivamente, tali curve, verranno utilizzate per la generazione di chiavi a 160, 192, 256 e 384 bit.
Facciamo notare che le **X** sono state inserite nei campi, relativi alle singole versioni, non necessari e non utilizzati per il corretto funzionamento di tali versioni considerate. Come ultima considerazione, si fa presente che le soluzioni di sicurezza standard per MQTT e CoAP, rispettivamente TLS e DTLS, risultano essere quelle più dispendiose dal punto di vista computazionale e non solo. Ma questo è comunemente noto, in quanto tali soluzioni non sono pensate per il mondo IoT e per i dispositivi resource constrained ma sono delle soluzioni già esistenti applicate come soluzioni di tipo "add-on" a protocolli che sono stati pensati senza un'occhio di riguardo alla sicurezza (basti pensare che il TLS è il protocollo usato per HTTPS, un protocollo completamente general purpose).

Tabella 2 – Test effettuati

PROTOCOLLO-VERSIONE	AVG SIZE FRAME (bytes)	AVG SIZE DATA (bytes)	AVG SIZE CERT. (bytes)	KEY SIZE (bits)	AVG LATENCY (ms)
COAP PUT	265	231	X	X	78
COAP OBSERVER	289	255	X	X	12
MQTT PUBLISH	284	218	X	X	19
COAPS DTLS-RSA	281	247	1039	X	598
COAPS DTLS-ECDHE	300	266	642	X	602
COAPS PUT DTLS-RSA	286	252	1053	X	617
COAPS PUT DTLS-ECDHE	289	255	657	X	627
COAPS PUT AEAD-AES	317	283	X	256	645
COAPS PUT RSA-DSA	493	459	X	1024	608
COAPS PUT RSA-DSA	495	461	X	1536	617
COAPS PUT RSA-DSA	621	587	X	2048	627
COAPS PUT RSA-DSA	693	659	X	3072	666
COAPS PUT RSA-DSA	1069	1036	X	7680	1160
COAPS PUT ECIES-ECDSA	482	448	X	160	881
COAPS PUT ECIES-ECDSA	495	461	X	192	932
COAPS PUT ECIES-ECDSA	498	463	X	224	1023
COAPS PUT ECIES-ECDSA	506	472	X	256	1143
COAPS PUT ECIES-ECDSA	537	503	X	384	1770
COAPS PUT ECC-AEAD-AES	322	288	X	224 + 256	516
COAPS PUT RSA-AEAD-AES	343	305	X	2048 + 256	582
MQTTS TLS-ECDHE	333	267	1260	X	411
MQTTS TLS-RSA	310	244	2053	X	412
MQTTS RSA-DSA	544	478	X	1024	413
MQTTS RSA-DSA	544	478	X	1536	408
MQTTS RSA-DSA	507	440	X	2048	410
MQTTS RSA-DSA	544	478	X	3072	408
MQTTS RSA-DSA	1120	1054	X	7680	410
MQTTS ECIES-ECDSA	514	448	X	160	411
MQTTS ECIES-ECDSA	521	455	X	192	409
MQTTS ECIES-ECDSA	528	462	X	224	411
MQTTS ECIES-ECDSA	537	471	X	256	408
MQTTS ECIES-ECDSA	569	503	X	384	409
MQTTS AEAD-AES	354	288	X	256	410
MQTTS RSA AEAD-AES	360	294	X	2048 + 256	412
MQTTS ECC-AEAD-AES	357	291	X	224 + 256	411

BIBLIOGRAFIA

[1] Wylach, P., Dondossola, G., & Terruggia, R. Testing della Sicurezza nelle comunicazioni standard delle Smart Grid.
[2] Mackiewicz, R. E. (2006, June). Overview of IEC 61850 and Benefits. In 2006 IEEE Power Engineering Society General Meeting (pp. 8-pp). IEEE.
[3] Cleveland, F. (2012). Iec tc57 wg15: Iec 62351 security standards for the power system information infrastructure. White Paper.
[4] IEC 62351-1, "Communication network and system security - introduction to security issues", 2007.
[5] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography.1996.
[6] E. Barker and Q. Dang, "NIST special publication 800-57 part 3 revision 1", NIST
[7] S.W. Blume, Electric Power Systems Basics for the Nonelectrical Professional. Piscataway: IEEE Press, 2007.
[8] S. E. Collier, "Ten steps to a smartergrid", in 2009 IEEE Rural Electric Power Conference, 2009, B2–B2–7.
[9] A. R. Metke and R. L. Ekl, "Smart grid security technology", in 2010 Innovative Smart Grid Technologies (ISGT), 2010, pp. 1–7.
[10] (Jul. 13, 2020). CrashOverride - analysis of the threat to electricgrid operations, [Online]. Available: https://www.dragos.com/blog/crashoverride/CrashOverride-01.pdf.
[11] IEC 61850-1, "Communication networks and systems for power utility automation - part 1: Introduction and overview", 2013.
[12] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier", White paper, Symantec Corp., Security Response, vol. 5, no. 6, 2011.
[13] (Jul. 25, 2020). Cyber-attackagainstUkrainiancriticalinfrastructure, [Online]. Available: https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01.
[14] (Jul. 12, 2020). CrashOverride malware, [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA17-163A.
[15] Schlegel, R., Obermeier, S., & Schneider, J. (2015, September). Assessing the security of IEC 62351. In 3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3 (pp. 11-19).
[16] *https://www.cybersecurity360.it/cybersecurity-nazionale/sicurezza-informatica-in-europa-e-in-italia-tutte-le-norme-di-riferimento/*
[17] https://www.gazzettaufficiale.it/eli/id/2018/06/09/18G00092/sg
[18] *https://www.cybersecurity360.it/cultura-cyber/leggi-di-sicurezza-informatica-guida-ragionata-a-vecchie-e-nuove-regole/*
[19] *https://www.ilsole24ore.com/art/cybersecurity-act-oggi-vigore-nuovo-regolamento-ue-sicurezza-informatica-ACDp1zU*
[20] PANEL, S. G. I. (2014). A White Paper developed by the Smart Grid Interoperability Panel Smart Grid Cybersecurity Committee February 2014.
[21] http://www.nist.gov
[22] *https://www.cybersecurity360.it/soluzioni-aziendali/vulnerabilita-delliot-le-best-practice-per-la-mitigazione-del-rischio/*
[23] https://www.ictsecuritymagazine.com/articoli/reti-4g-5g-profili-di-vulnerabilita-e-possibili-contromisure/
[24] http://www.techtrained.com/paging-procedure-lte/.
[25] https://www.enisa.europa.eu/publications/iot-security-standards-gap-analysis
[26] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji and V. C. M. Leung, "Enabling Massive IoT Toward 6G: A Comprehensive Survey," in IEEE Internet of Things Journal, vol. 8, no. 15, pp. 11891-11915, 1 Aug.1, 2021, doi: 10.1109/JIOT.2021.3063686.
[27] Anak Agung Gde Agung, Rini Handayani, Blockchain for smart grid, Journal of King Saud University - Computer and Information Sciences, 2020, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2020.01.002.
[28] G. Chen, M. He, J. Gao, C. Liu, Y. Yin and Q. Li, "Blockchain-Based Cyber Security and Advanced Distribution in Smart Grid," *2021 IEEE 4th International Conference on Electronics Technology (ICET)*, 2021, pp. 1077-1080, doi: 10.1109/ICET51757.2021.9451130.

[29] S. Samy, M. Azab and M. Rizk, "Towards a Secured Blockchain-based Smart Grid," *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021, pp. 1066-1069, doi: 10.1109/CCWC51732.2021.9376089.
[30] T. Jansen, "Technical considerations for building securesubstationautomation systems", Electra, D2/B3/C2.01 Joint Working Group Report, Nov. 29, 2006.
[31] IEC/TS 62351-6, "Power systems management and associated information exchange – data and communications security - part 6: Security for IEC 61850", 2007.
[32] IEC 62351-3, "Power systems management and associated information exchange - data andcommunications security - part 3: Communication network and system security –profilesincluding TCP/IP", 2014.
[33] IEC/TS 62351-4, "Power systems management and associated information exchange– dataand communications security - part 4: Profiles including MMS", 2007.
[34] Youssef, T. A., El Hariri, M., Bugay, N., & Mohammed, O. A. (2016, June). IEC 61850: Technology standards and cyber-threats. In 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC) (pp. 1-6). IEEE.
[35] N Kush, E. Ahmed, M. Branagan, Ernest. F, "Poisoned GOOSE: Exploiting the GOOSE Protocol," Proceedings of the TwelfthAustralasian Information Security Conference (AISC 2014), Auckland, New Zealand, pp. 17-22.
[36] J. Hoyos, M. Dehus, and T. X. Brown, "Exploiting the gooseprotocol: A practicalattack on cyber-infrastructure," in Globecom Workshops (GC Wkshps), 2012 IEEE. IEEE, 2012, pp. 1508–1513.
[37] Q. Li, C. Ross, J. Yang, J. Di, J. C. Balda, and H. A. Mantooth, "The effects of floodingattacks on time-critical communications in the smartgrid."
[38] M. Strobel, N. Wiedermann, and C. Eckert, "Novelweaknesses in IEC 62351 protectedsmartgrid control systems", in 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 266–270.
[39] S. Fuloria, R. Anderson, et al., "The protection of substation communications", Cambridge University, 2010.
[40] F. Hohlbaum, M. Braendle, and F. Alvarez, "Cyber security: Practicalconsiderations for implementing IEC 62351", ABB, 2009.
[41] NESCOR, "Electric sector failure scenarios and impact analyses - version 3.0", EPRI, 2015
[42] M. T. A. Rashid, S. Yussof, et al., "A review of security attacks on IEC61850 substationautomation system network", in 2014 International Conference on Information Technologyand Multimedia (ICIMU), 2014.
[43] J. Wright and S. Wolthusen, "Access control and availabilityvulnerabilities in the ISO/IEC61850 substationautomationprotocol", in Critical Information Infrastructures Security.Nov. 2017, p. 239.
[44] IEC/TR 61850-90-4, "Communication networks and systems for power utility automation -part 90-4: Network engineering guidelines", 2013.
[45] B. Kang, P. Maynard, et al., "Investigating cyber-physical attacksagainst IEC 61850 photovoltaicinverter installations", in 2015 IEEE 20th Conference on Emerging TechnologiesFactory Automation (ETFA), 2015, pp. 1–8.
[46] A. Valdes, C. Hang, et al., "Design and simulation of fast substation protection in IEC 61850environments", in 2015 Workshop on Modeling and Simulation of Cyber-Physical EnergySystems (MSCPES), 2015, pp. 1–6.
[47] J. Hoyos, M. Dehus, and T. X. Brown, "Exploiting the GOOSE protocol: A practicalattackon cyber-infrastructure", in 2012 IEEE Globecom Workshops, 2012, pp. 1508–1513.
[48] N. Kush, E. Ahmed, et al., "Poisoned GOOSE: Exploiting the GOOSE protocol", in Proceedingsof the TwelfthAustralasian Information Security Conference (AISC 2014), Auckland,New Zealand, 2014.
[49] R. Schlegel, S. Obermeier, and J. Schneider, "Assessing the security of IEC 62351", in Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research 2015, 2015.
[50] O. Khaled, A. Marin, et al., "Analysis of secure TCP/IP profile in 61850 basedsubstationautomation system for smartgrids", Int. J. Distrib. Sen. Netw., vol. 2016, Apr. 2016
[51] J. G. Wright and S. D. Wolthusen, "Limitations of IEC62351-3's public key management",in2016 IEEE 24th International Conference on Network Protocols (ICNP), 2016.
[52] M. M. R. Chowdhury, H. Raddatz, and J. E. Y. Rosseb, "Challengeswhensecuringmanufacturingmessage service in legacy industrial control systems", in Proceedings of the2014 IEEE Emerging Technology and Factory Automation (ETFA), 2014, pp. 1–6.

[53] S. Fries, H. J. Hof, and M. Seewald, “Enhancing IEC 62351 to improve security for energyautomation in smartgridenvironments”, in 2010 Fifth International Conference on Internet and Web Applications and Services (ICIW), 2010, pp. 135–142.
[54] K. C. Ruland and J. Sassmannshausen, “Non-repudiationservices for the MMS protocol of IEC 61850”, in Security Standardisation Research: Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings, L. Chen and S. Matsuo, Eds. Cham: Springer International Publishing, 2015, pp. 70–85.
[55] P. Weerathunga, “Security aspects of smartgridcommunication”, Western University, 2012.
[56] Thota, P., & Kim, Y. (2016, December). Implementation and comparison of M2M protocols for Internet of Things. In 2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD) (pp. 43-48). IEEE.
[57] Bormann, C., Castellani, A. P., & Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. IEEE Internet Computing, 16(2), 62-67.
[58] https://www.hivemq.com/mqtt/mqtt-protocol/
[59] Xiao, Y., Chen, H., Yang, S., Lin, Y. B., & Du, D. Z. (2009). Wireless network security. EURASIP Journal on Wireless Communications and Networking, 2009(1), 1-3.
[60] Gurunath, R., Agarwal, M., Nandi, A., & Samanta, D. (2018, August). An overview: security issue in IoT network. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on* (pp. 104-107). IEEE.
[61] Bradbury, D. (2011). Hacking wifi the easy way. Network Security, 2011(2), 9-12.
[62] Sari, A., & Karay, M. (2015). Comparative analysis of wireless security protocols: WEP vs WPA. International Journal of Communications, Network and System Sciences, 8(12), 483.
[63] Gonzales, H., Bauer, K., Lindqvist, J., McCoy, D., & Sicker, D. (2010, December). Practical defenses for evil twin attacks in 802.11. In 2010 IEEE Global Telecommunications Conference GLOBECOM 2010 (pp. 1-6). IEEE.
[64] Kumar, S., & Tapaswi, S. (2012, June). A centralized detection and prevention technique against ARP poisoning. In Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec) (pp. 259-264). IEEE.
[65] Laaroussi, Z., & Novo, O. (2021, January). A Performance Analysis of the Security Communication in CoAP and MQTT. In 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) (pp. 1-6). IEEE.
[66] Nguyen, N. D., Bui, D. H., & Tran, X. T. (2020, December). A Lightweight AEAD encryption core to secure IoT applications. In 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) (pp. 35-38). IEEE.