

CONTINUIAMO MQTT:

Ci Sono 3 livelli di Qualità del Servizio (AFFIDABILITÀ) diversi:

tutto Vado Bene

da NON Confondere con la SICUREZZA

Tutto ciò che può Andare Storto per Causa
di Operazioni Malevoli Umane

MQTT Basato Su livi Tresporto (che è Già Affidabile)

AFFidabilità HTTP, Mai Sentita? No Perché Mi Basavo Su TCP e Quindi
non Avevo Nessun Problema.

MQTT Parla di Affidabilità per un Motivo Ben Specifico, Ovvero:
che Qui Abbiamo 3 Processi e non più Solo 2 e Quindi va
ad Aggiungere Complessità.

Potrebbe Essere l'END TO END il Problema, Non le Singole Trasmissioni

All'UTENTE Importa il Passaggio di INFORMAZIONI Corretto e non
il BROKER (interessa Solemente ai TECNICI)

↳ Oggetto Intermedio a Livello Applicativo (come RANGE EXTENDER A)
LVL FISICO

Se Va giù il BROKER (che è lui che crea problemi di Affidabilità)

Dopo PUBLISH connessione viene Chiusa. Rispetto al SUBSCRIBER che Rimane Aperta in Attesa in una NOTIFY da Parte del BROKER

→ Heart/Keep Alive per non Essere Rimossa dal NAT

Se Provo a fare PUBLISH e Broker è già giù allora me ne Accorgo, ma Se ha fatto publish, chiudo Connessione e poi BROKER Va giù, allora Nessuno Se ne Accorge

QoS Lvl 0 (at Most One) ⇒ Significa che non c'è Controllo e c'è Affidabilità Nulla.

A Lvl. END TO END il Dato Arriva al Massimo una Volta... Quindi può non Arrivare. Non Arriva /può Essere DUPLICATO, Quindi non Viene RITRASMESSO.

→ devo Poi Capire Quando Viene Ritrasmesso Per Distinguere i PACHETTI DUPLICATI da Quelli "NORMALI" (Anche Perchè io Potrei Valer Inviarti 2 Volte lo Stesso Dato)

SUBACK/CONNACK

In MQTT Sono Stati Inseriti degli ACK/Scambi di Messaggi di Servizio che Viaggiano Su TCP per Garantire Certi livelli di Affid.

QoS Lvl 1 (At Least One) ha Bisogno di più Scambi, Sforzo Maggiore Per La RETE (Ogni Messaggio MQTT Genererà degli

ACK TCP, Quindi ha un EFFETTO VALANGA)

Qui DEVE Arrivare e può Arrivare più di una Volta, devo Quindi Aggiungere un Qualcosa Per distinguere i Messaggi duplicati come ad Esempio può Essere un TIMESTAMP.

■ QoS Lvl 2 (Esettamente 1 Volta) \Rightarrow proprio come TCP che Essetta Riconsegnazione fino a che NON Viene Scambiato Messaggio Con Successo \leq Volta e Basta.

Così facendo ARRIVERÀ SEMPRE e SOLAMENTE 1 VOLTA.

Si fa Riferimento a Messaggi END TO END perché tra Singoli Scambi c'è Già TCP.

Nella Successiva Si Applica Stessa Cosa, Posso Mettere TLS/SSL sulla MQTT e così diventa SICURO

Funziona Solo su TCP

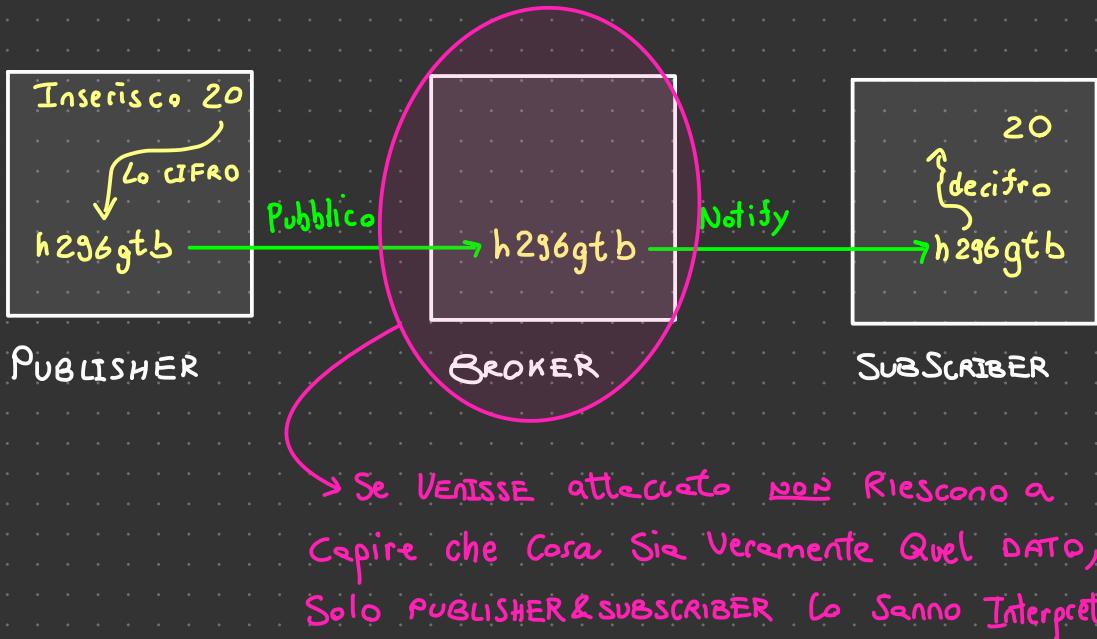
↓
Non modifichiamo nulla a Lvl Applicativo, chiamo Libreria e WRAPPA con TLS

Diventa SICURO Nei Singoli Viaggi, ma NON END TO END.

Tra PUB e BROKER è Sicuro Si Ma poi dopo Quando il dato lo ha in PANCIA il BROKER li Cade il Tutto perché Se Accessi il BROKER Non Sarei Protetto.

Una Soluzione Sarebbe Potrebbe Essere Quella di Negociare una Chiave per Cifrare il DATO per RENDERLO un DATO SENZA SENSO e Solo Quando Arriva al SUBSCRIBER (che sa la KEY)

Viene DECIFRATO.



Questo è un ESEMPIO, ma non ha Standardizzato un AFFIDAB. ma non ha Sicurezza.

Io Sono AUTENTICATO al BROKER, e Anche il BROKER Verso di noi ma non però END TO END, Quindi non So Chi Sarà Che legge i miei PUBLISH.

KAFKA ad Esempio può Impostare DIRITTI Su Alcuni TOPIC.

CONNESSIONI BREVI per PUBLISHER e LUNGHE per SUBSCRIBER

Se SUBSCRIBER è dietro al NAT Nessun Problema Perchè è lui che si Attacca al BROKER. \Rightarrow NAT COMPLIANT

STRUTTURA DEL TOPIC:

Sono Stringhe, Meglio Non Usare gli Spazi

Posso Sfruttare TOPIC Gerarchici Implementati tramite '/' che mi Permette di COSTRUIRE degli Alberi di TOPIC

Nei TOPIC Posso Usare dei caratteri '#' o '+' per:

• '#' Fare SUBSCRIBE Di tutti i Sotto-TOPIC (Messo Sulle foglie) [come * sul PATH di Linux] Lvl1/Lvl2/#

• '+' Va Messo Sulle RADICI +/Lvl1/Lvl2

ESEMPIO ESAME Mostrare la differenza Con CHAT Web SOCKET

Pub-Sub

↓
Devo Implementarmi Tutto

↓
Devo fargli Scaricare App

Ho Browser

↓
GUI è facilitata

↓
Non devo far Scaricare nessuna App

SICUREZZA: \Rightarrow Ricordarsi i 5 punti

CRITTOGRAFIA: Implementazione

Vuole rendere sicura l'informazione in modo che un utente non autorizzato non possa DECIFRARE l'informazione e comprenderla

Le mie stringhe di BYTE

CRITTOANALISI Invece è il contrario, quindi cercare di aggirare o superare le protezioni critografiche accedendo alle info protette

Ci sono 2 aspetti ben separati:

ALGORITMO \rightarrow Brutta Pratica, non posso sapere se c'è un buco
Non me ne accorgo, no segretezza ALGORITMO
La forza dell'Algoritmo deve stare nella segretezza della CHIAVE

CHIAVE \rightarrow Deve Essere Tenuta Segreta

Quando BYTE Sono cifrati prende il Nome di CIPHER TEXT
anche se non è per FORZA Testo

C'è un ALGORITMO che decifra (Sz. Inverse della Cifratura) anche
detto INVERSO, Parametrizzata Sulla CHIAVE
Che Specifica l'Algoritmo
e non è NOTA la chiave

ESEMPI STUPIDI:

- Aggiungi x ad Ogni BYTE \Rightarrow Chiave è x e non So di Quanto INCREMENTO.
- Sposto di x lettere \Rightarrow Non So di Quanto Sposto

CIFRATURA SIMMETRICA: Chiavi Uguali per CIFRATURA e DECIFRATURA (come la PORTA di Casa) [Esempio della Somma]

CIFRATURA ASIMMETRICA: Una è PUBBLICA e l'Altra è privata (Nascono Assieme e Sono legati) da una Sz. Matematica
Sono INTERSCAMBIABILI, non hanno Ruolo Specifico ma So Solo che se una viene usata per CIFRATURA non va BENE per DECIFRARE e Viceversa.

CHIAVE PUBBLICA la do a TUTTI mentre Quella PRIVATA la Tengo io Ma Salgo io a Casa Quale Voglio CONDIVIDERE Perché

- a lui di Caratteristiche Sono Uguali (DIPENDE L'uso che ne faccio)
- ↓
non devo Scambiarle Poi
- Devo RENDERE il più difficile possibile, ma Si può Sempre fare il BRUTEFORCING:
- M ho 8 Bit? faccio 2^8 prove e Risolvo.
- SICUREZZA PRATICA \Rightarrow Sforzo per fare la violazione deve ESSERE disEconomico, Quindi deve ESSERE Svantaggioso.
- Quasi IMPOSSIBILE Se ho il canale di "SESSIONE" e Quindi ho un PROTOCOLLO che RINNOVA la chiave con Tempo < Che ci mette la Tecnologia a Fare forza Bruta.
- Protocollo Usa la Chiave, ma ciò che è IMPOSSIBILE Oggi, Domani Sarà Possibile ed ESSENDO BRUTE-FORCING e Quindi:
- M Si inventa nuovo Protocollo
 - M Oppure si Allunga dimensione chiave
- SIMMETRICA ha cifratura più leggera per chi ha le chiavi e le posso Pagare Con lo Scacchiarsi delle BATTERIA o Energia.
- Ma il problema è COME MI SCAMBIO la chiave in Modo Sicuro Senza Però usare la chiave
- Usa la CIFRATURA ASIMMETRICA (che non Necessita Scambio di Chiave)

Per Scambiarmi La Chiave SIMMETRICA Per
Poi Abbandonare L'ASIMMETRICA.

CIFRARIO DI CESARE:

Chiave è #Shift, Si Applica a TUTTO, Non Solo al TESTO ma anche a JPEG Ad Esempio.

L'Alfabeto per la Macchina è 255 visto che viene Considerato il SINGOLO BYTE

1 Byte = 8 Bit



$2^8 - 1$ Possibili Chiavi

ALFABETO In INFORMATICA Rappresenta tutti i Simboli Che riesco a Creare Partendo Dall' UNITÀ BASE

Per BRUTEFORCING devo Provare TUTTE le POSSIBILI Permutazioni di una STRINGA \Rightarrow Nel NOSTRO CASO $\Rightarrow 255!$

UPPER BOUND è il BRUTEFORCING

Cifratura monoALFABETICA si Presta ad un Attacco sulle frequenze della COMPARSA dei Simboli.

Questo Capita anche Nel CODIFICARE un PDF oppure Anche Nelle IMG .Jpg.

CIFRATURA A BLOCCHI:

Quello che Viene Usato Oggi

Dati K Bit ; Possibili: 2^K ingressi Vengono Permutati

CIFRATURA ASIMMETRICA:

Sono 2 Chiavi FORMALMENTE UGUALI

Quello che cifra Con una Chiave Posso decifrarla con l'Altra e Viceversa.

Conosco una Chiave, ma Quella non dà Nessuna INFO per la Privata

Non c'è Entità Terza per lo SCAMBIO delle Chiave

Ai fini della CONFIDENZIALITÀ Basta che Si SCAMBINO le chiavi

Ma Sono Sicuro che Sto Parlando Con la PERSONA giusta?

Ruolo della BLOCKCHAIN Cruciale per la FIDUCIA Rispetto a Sidersi Nella Catena