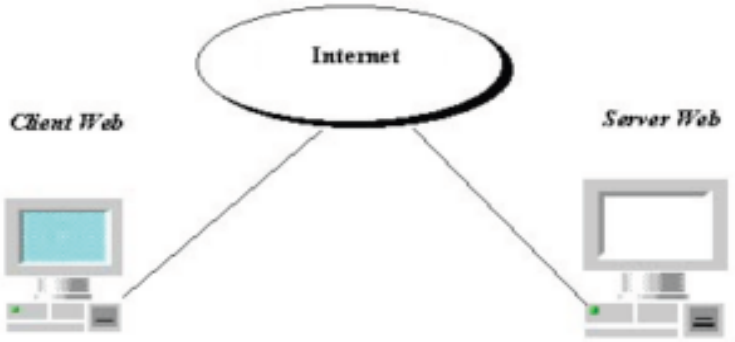
Comunicazioni sicure su Internet: HTTPS e SSL/TLS

HTTP - 1

- Come funziona nel dettaglio il Web?



- Uniform Resource Locator (**URL**) (Indirizzo simbolico pagina)
- Hyper Text Transfer Protocol (**HTTP**)
- Hyper Text Markup Language (**HTML**)

HTTPS (HTTP Secure) - 2

Hyper Text Transfer Protocol (HTTP) -------> HTTPS

HTTPS è invece *Hyper Text Transfer Protocol over Secure Socket Layer /Transport Layer Security*.

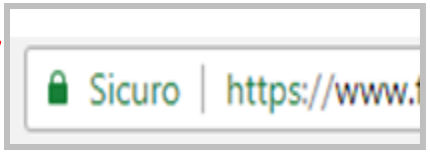
Utilizza il **protocollo SSL/TLS** (*Secure Socket Layer /Transport Layer Security* - letteralmente *livello di socket sicuro/ sicurezza del livello di trasporto*) tra il livello di Trasporto e di Sessione.

L' **HTTPS** è:

- Sintatticamente identico allo schema **http://** ma con la **differenza** che gli accessi vengono effettuati sulla porta **443** (e non sulla **80**) - **https://**
- Tra il protocollo TCP e HTTP si interpone un livello di **crittografia/autenticazione**.

HTTPS - 3

- **HTTPS verifica l'identità di un sito web e crittografa** le informazioni inviate tra il sito web e il client e viceversa. Queste includono i cookie, i percorsi degli URL e i dati inviati tramite i moduli (form). **HTTPS** è progettato per impedire che queste informazioni vengano lette o modificate mentre sono in transito.

- Le **URL** di **HTTPS** iniziano con **https://** 

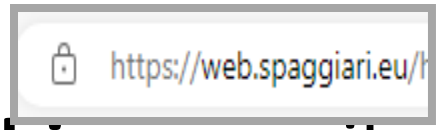
e utilizzano la porta **443** di default, a differenza di **HTTP** le cui URL cominciano con **http://** e utilizzano la porta **80**.

HTTPS - 4

Come **riconoscere** una connessione **HTTPS**.

Per **incentivare** la **migrazione** verso **siti sicuri**, tutti i browser visualizzano oramai lo **status** dei website:

- **Un'icona a forma di lucchetto chiuso** di colore **grigio** o **verde** significa che il **sito è sicuro** ed **ha un certificato attendibile** (in Microsoft Edge o Crhome il **lucchetto grigio** indica: sito web crittografato e verificato; il **lucchetto verde** indica un certificato di convalida più esteso (**EV**, **Extended Validation**) che richiede un processo di verifica dell'identità molto rigoroso);



- **Un'icona a forma di cerchio** con il simbolo di **Informazioni** o **triangolo con !** significa **sito non sicuro**;

SSL/TLS *Secure Socket Layer /Transport Layer Security* - 5

Si tratta di un **insieme di protocolli** crittografici che aggiungono funzionalità di **cifratura e autenticazione a protocolli preesistenti**.

SSL: versione iniziale sviluppata da **Netscape** nel **1995** , poi evoluta nel protocollo standardizzato **TLS** nel **1999**.

Protocolli SSL e TLS

Protocollo	Pubblicato	Status
SSL 1.0	Non pubblicato	Non pubblicato
SSL 2.0	1995	Deprecato nel 2011 (RFC 6176)
SSL 3.0	1996	Deprecato nel 2015 (RFC 7568)
TLS 1.0	1999	Deprecazione pianificata per il 2020[1]
TLS 1.1	2006	Deprecazione pianificata per il 2020[1]
TLS 1.2	2008	
TLS 1.3	2018	

Il protocollo **SSL/TLS** provvede alla sicurezza del collegamento garantendo:

- **Autenticazione**: sicurezza dell'identità dei soggetti che comunicano (**crittografia asimmetrica**, ovvero a **chiave pubblica e privata** (ex. RSA).
- **Riservatezza-Confidenzialità**: protezione dei dati da osservatori non autorizzati (la crittografia è usata dopo un **handshake** (**accordo**) iniziale per definire una **chiave simmetrica segreta di sessione** (**Diffie-Helmann**). In seguito, per **crittografare i dati** è usata la **crittografia simmetrica** (**3DES**, **AES**, **IDEA** ecc.)).
- **Integrità**: sicurezza che il dato ricevuto è uguale al dato inviato (il livello di Trasporto include un **controllo dell'integrità del messaggio** basato su un apposito **MAC** (**Message Authentication Code**) che utilizza funzioni **hash** sicure (**MD5** Message Digest, **SHA** Secure Hash Algorithm). In tal modo si verifica che i dati spediti tra client e server non siano stati **alterati** durante la trasmissione.

Transport Layer Security - 6

Funzionalità di TLS

- **Cifrare la connessione** al fine di garantire la **riservatezza** dei dati.
- **Autenticare** il **server** a cui si è connessi.

Funzionamento di TLS

- Il **server** invia al **client** un **certificato** attestante la **propria identità** e contenente la **propria chiave pubblica** (ad es. una chiave **RSA**).
- Il **client verifica l'identità** del **server** tramite **Certification Authority**. Poi genera una **chiave di sessione casuale** (o **Diffie-Hellman**) e la invia al server, **cifrandola** con la **chiave pubblica** del server.
- Il server **decifra** il messaggio con la propria **chiave privata** ed ottiene la **chiave di sessione**. Questa viene utilizzata per cifrare il successivo traffico dati con un **algoritmo simmetrico** (ad es. **3DES** o **AES** o **IDEA**).
- La **riservatezza** è quindi garantita dall'utilizzo **ibrido** di **cifratura simmetrica** e **asimmetrica**.
- L'**autenticazione** del server è **garantita** dal **certificato** rilasciato dalla **CA**.

Autorità di Certificazione -7

Lo **scambio** di **chiavi pubbliche** introduce un nuovo problema da risolvere.

Se ad esempio vogliamo comunicare in modo sicuro con Google, abbiamo bisogno della sua chiave pubblica e lo stesso vale per altri siti che vogliamo visitare (Facebook, Twitter, etc...).

Dato che esistono miliardi di siti web su Internet, come possiamo ottenere una **chiave pubblica** per ogni sito web che vogliamo visitare?

Ed è qui che entrano in scena le **Autorità di Certificazione** o **CA** – **Certification Authorities** come **DigiCert**, **Comodo, Symantec**, **GlobalSign**, **Google Trust Services**, **VeriSign**, **GeoTrust, Actalis**.

Lo scopo del **certificato digitale** è quello di garantire che una **chiave pubblica** sia **associata** alla vera **identità del soggetto** che la rivendica come propria.

Una **CA** è un'organizzazione di terze parti con 3 obiettivi principali:

- Rilasciare certificati digitali.
- Confermare l'identità del proprietario del certificato.
- Fornire la prova che il certificato è valido.

Nello scenario di **HTTPS** la **chiave pubblica** è rappresentata da un **certificato digitale**, più noto come **certificato SSL/TLS**. Un **certificato digitale** è un **certificato di identità elettronico,** cioè un **documento elettronico** **che associa l'identità di una persona ad una chiave pubblica.**

Certificato SSL/TLS - 8

Ora vediamo in che modo è possibile **ottenere un certificato SSL/TLS firmato da una Certification Authority**:

1. Il proprietario del sito web genera una **chiave pubblica** e una **chiave privata** (ex. **RSA**) ed invia un file di **richiesta di firma** del certificato (**CSR** Certificate Signing Request) e la sua **chiave pubblica** alla **CA**.
2. La **CA** crea quindi un **certificato personale basato sulla CSR**, dove sono indicati nome di dominio, nome del proprietario, data di scadenza e altre informazioni, appone la **firma digitale**, infine **crittografa** l'intero certificato con **la chiave pubblica** del sito e lo rimanda al proprietario del sito web.
3. Il **certificato** viene quindi **decifrato con la chiave privata** del proprietario del **sito web** ed, infine, **viene installato sul server**.

Il certificato è firmato digitalmente dall'autorità certificante (certification authority, CA) che quindi si fa garante dell'autenticità delle informazioni contenute nel certificato.

- **N.B.** La **firma digitale della CA** è **crittografata** dalla **chiave privata della CA** e può essere decifrata solamente con la **chiave pubblica** della **CA**, chiamiamo questa **chiave Certificato Radice** (**Root Certificate**).
- Ogni dispositivo (pc, smartphone) ha, installato nel browser, un elenco di **certificati radice** delle **CA attendibili** e delle **CA intermedie**.

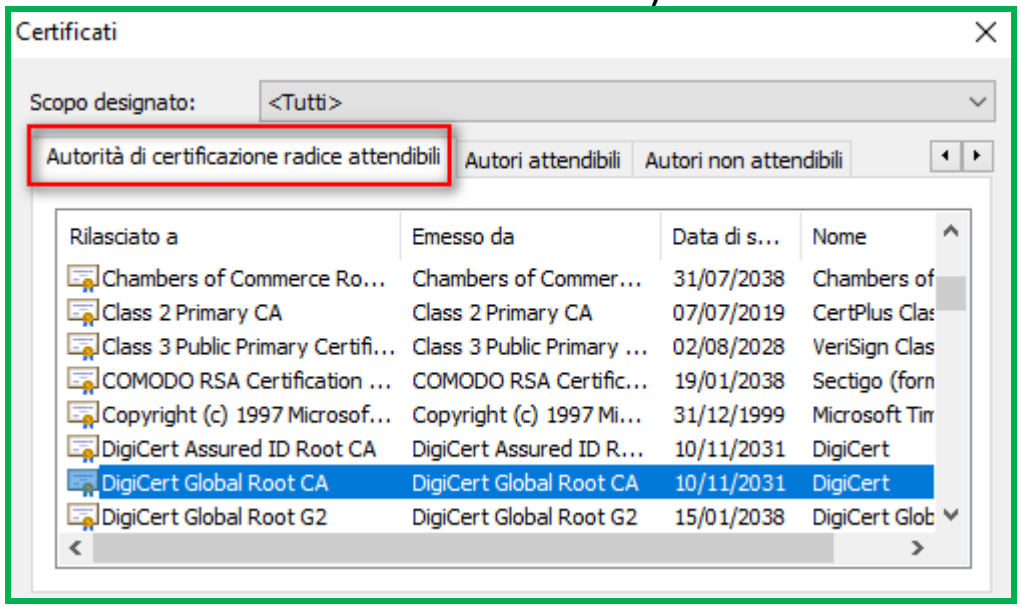
I Certificati SSL/TLS - 9

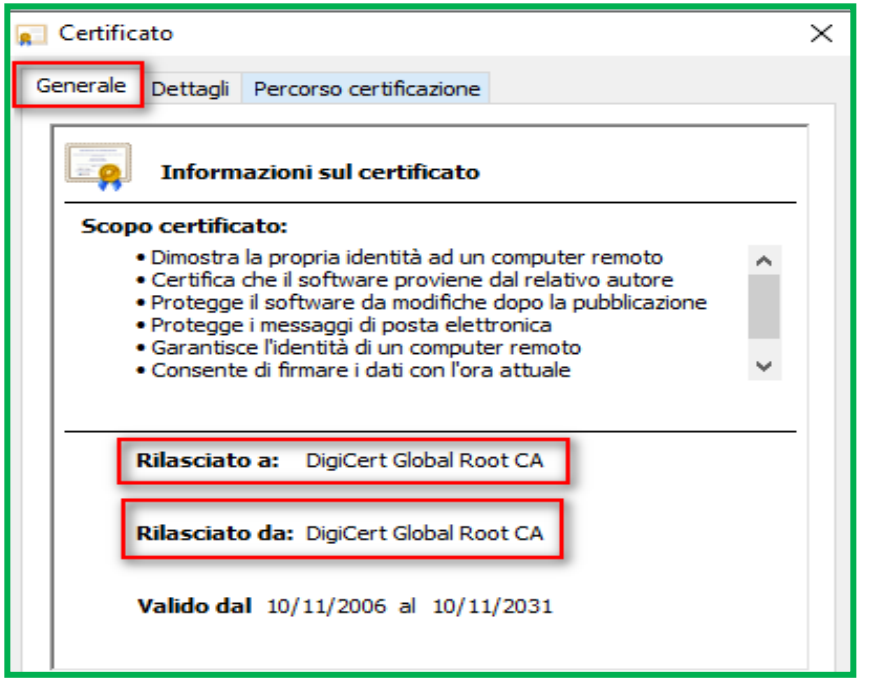
Il **Certificato Digitale** viene tipicamente distribuito mediante file **.DER** e contiene le seguenti **informazioni**:

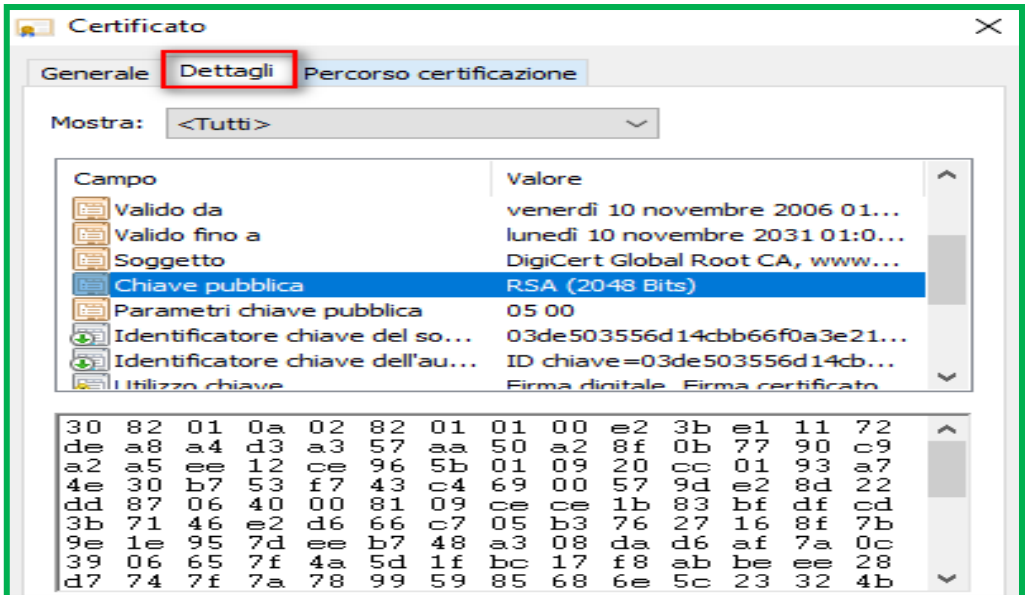
- **nome del proprietario** a cui è stato rilasciato il Certificato.
- **il nome del dominio** associato al Certificato.
- **chiave pubblica** (2048 bit) del Certificato (visualizzabile in forma esadecimale).
- **nome della CA** che ha firmato il Certificato Digitale
- **numero di serie** del Certificato.
- **validità** del Certificato (data di inizio e data di scadenza)
- la **firma digitale** della **CA** (**CERTIFICATE SIGNATURE**) e l'algoritmo utilizzato per la firma (**MD5** o **SHA**): la **CA** si fa garante dell'**autenticità** delle informazioni contenute nel certificato.

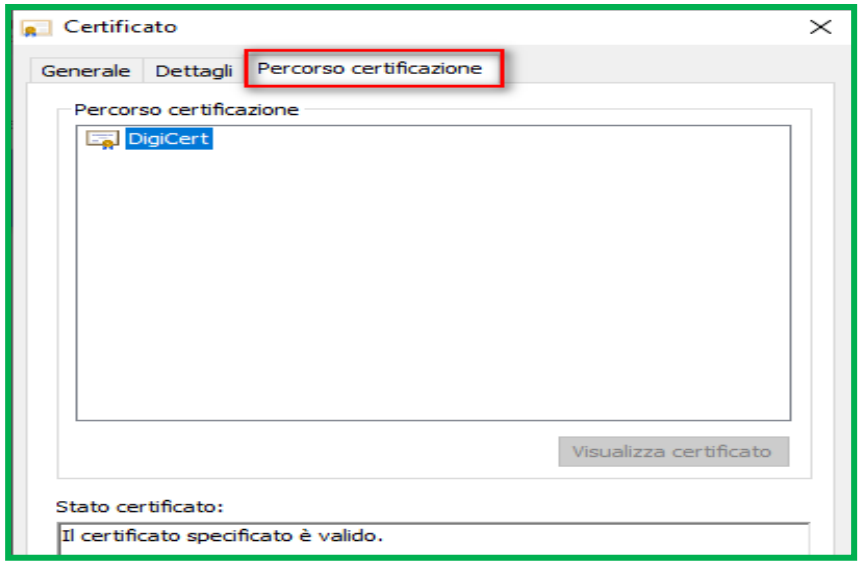
Elenco Certificati Radice - 10

Elenco di **certificati radice** sul browser **google chrome** (Impostazioni -> Privacy e sicurezza -> Sicurezza -> Gestisci certificati):



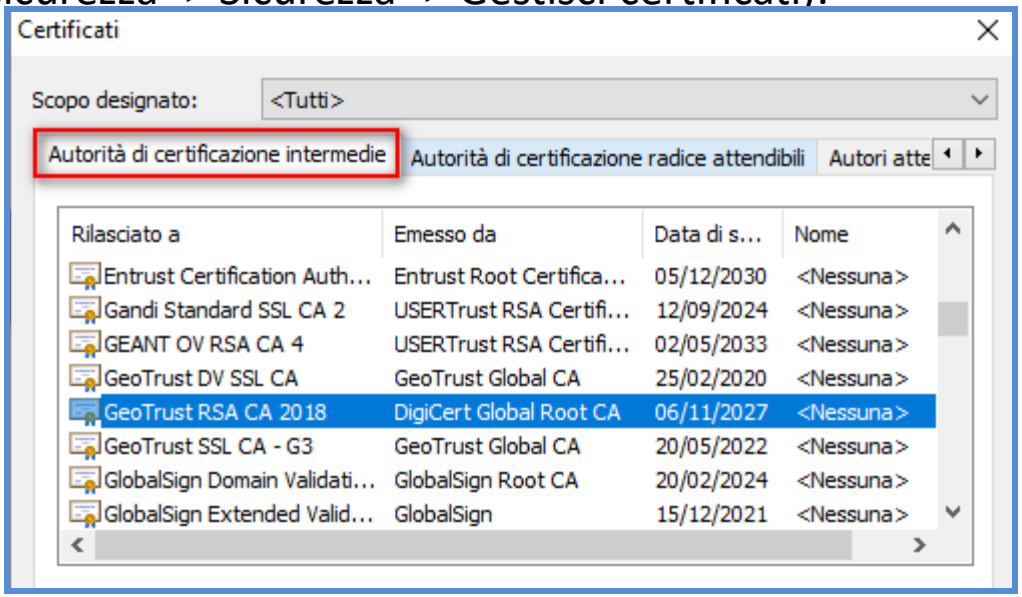


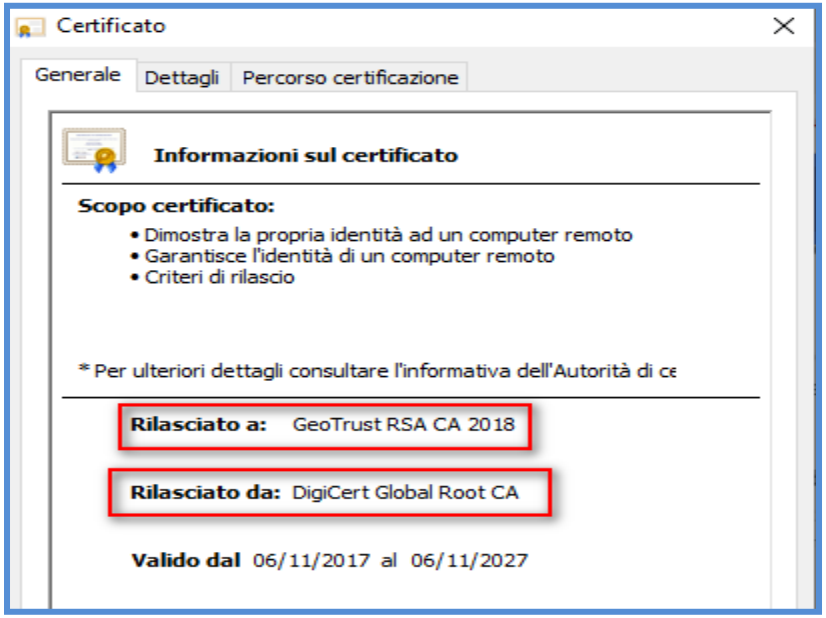


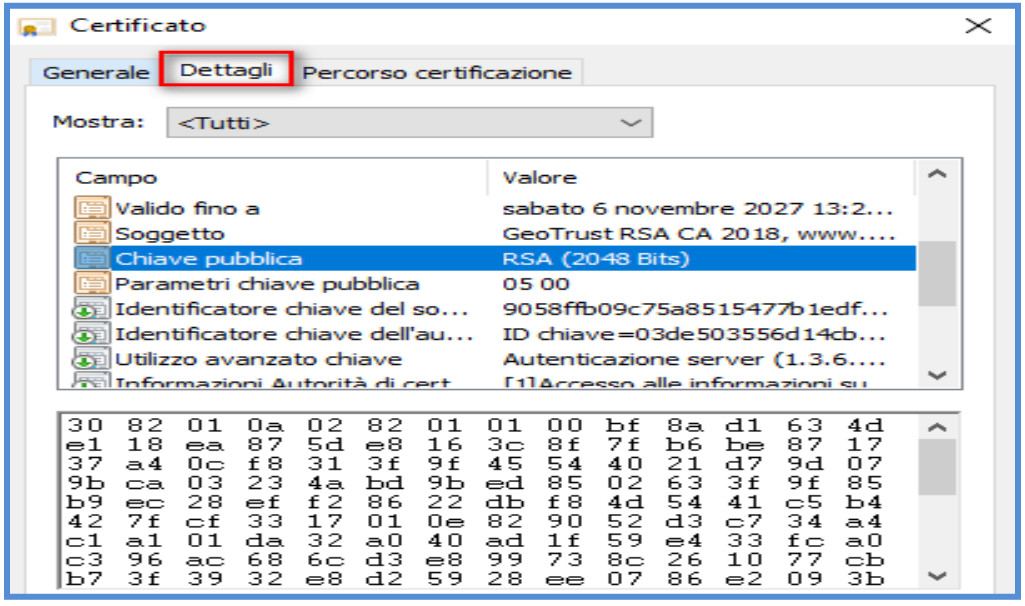


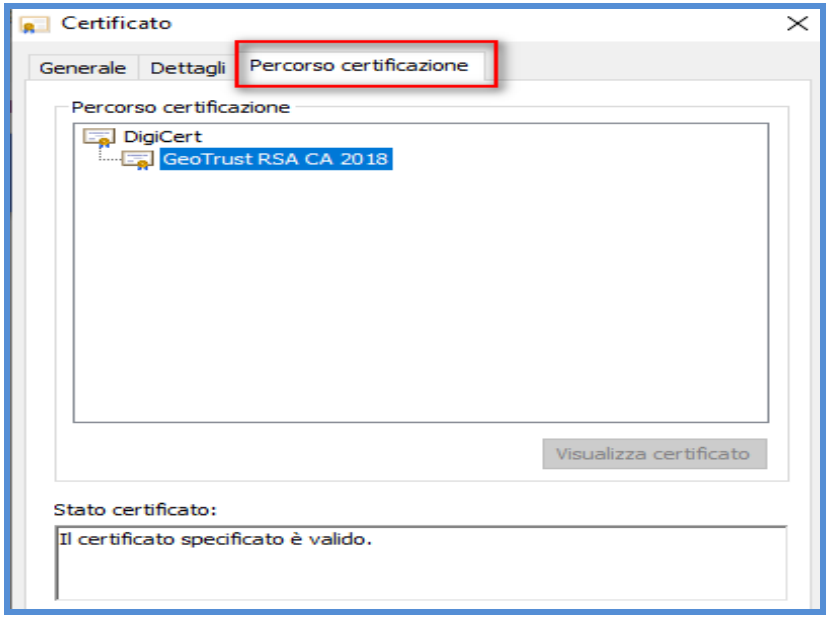
Elenco Certificazioni Intermedie - 11

Elenco di **certificati di Autorità Intermedie** sul browser **google chrome** (Impostazioni -> Privacy e sicurezza -> Sicurezza -> Gestisci certificati):









Gerarchia di Certificati - 12

Ma come fidarsi della **Certification Authority**?

- L'identità di ogni **CA** è **garantita** a sua volta da un certificato rilasciato da una **CA** "**superiore**"…
- Fino ad arrivare alla **root authority**, un'autorità certificante **non certificata da nessuno**, ovvero **certificata** da **se stessa**.
- Se la **root authority** è nota a livello mondiale e sottoposta a verifiche annuali da appositi organismi di controllo, possiamo ragionevolmente **fidarci** dei suoi **certificati**.

Root Authorities - 13

• Per questo motivo le **root authorities** fidate sono poche (meno di 40 in tutto il mondo), con **VeriSign** che detiene attualmente più del **50%** del mercato.

• I certificati delle root authorities sono "**self-signed**" in quanto non possono essere firmati da nessun'altra **CA**.

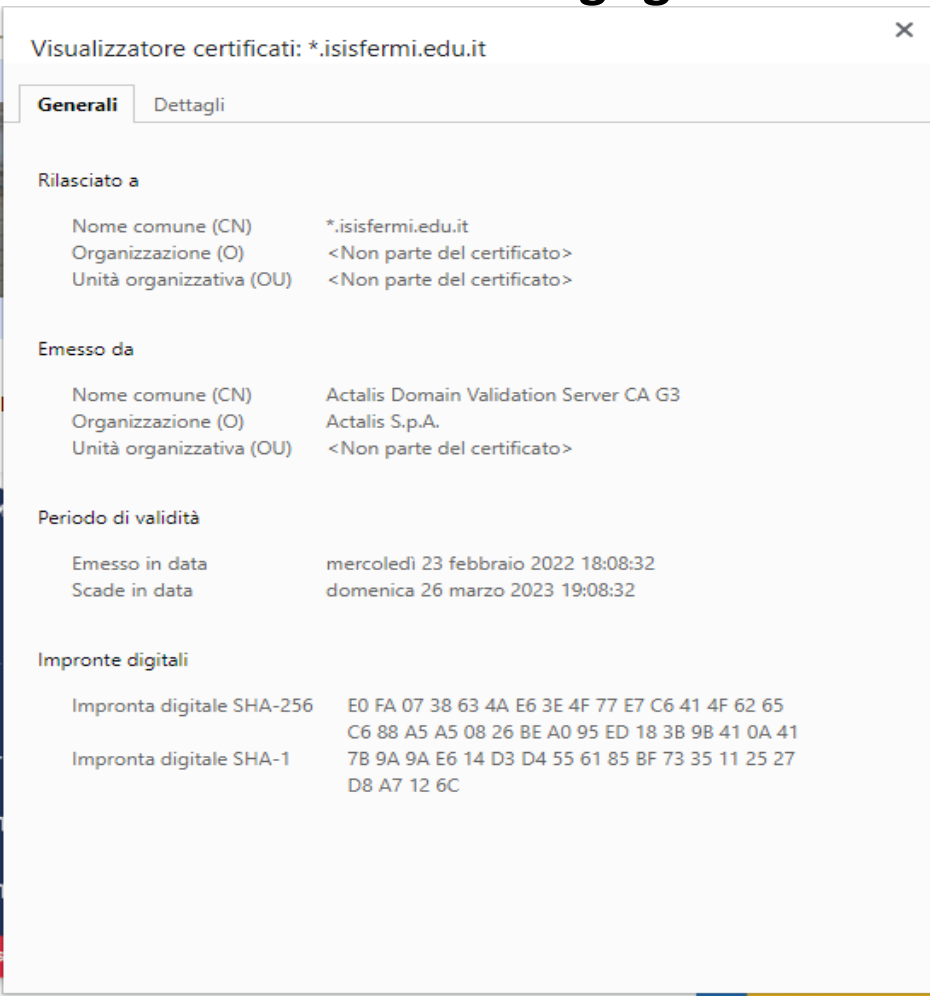
• Per garantire il processo di **verifica dei certificati**, **tutti i browser**, **client di posta elettronica**, e in generale ogni software che faccia uso di connessioni **TLS**, **sono distribuiti assieme** ai **certificati delle root authorities** (**Root Certificate - Certificati Radice**).

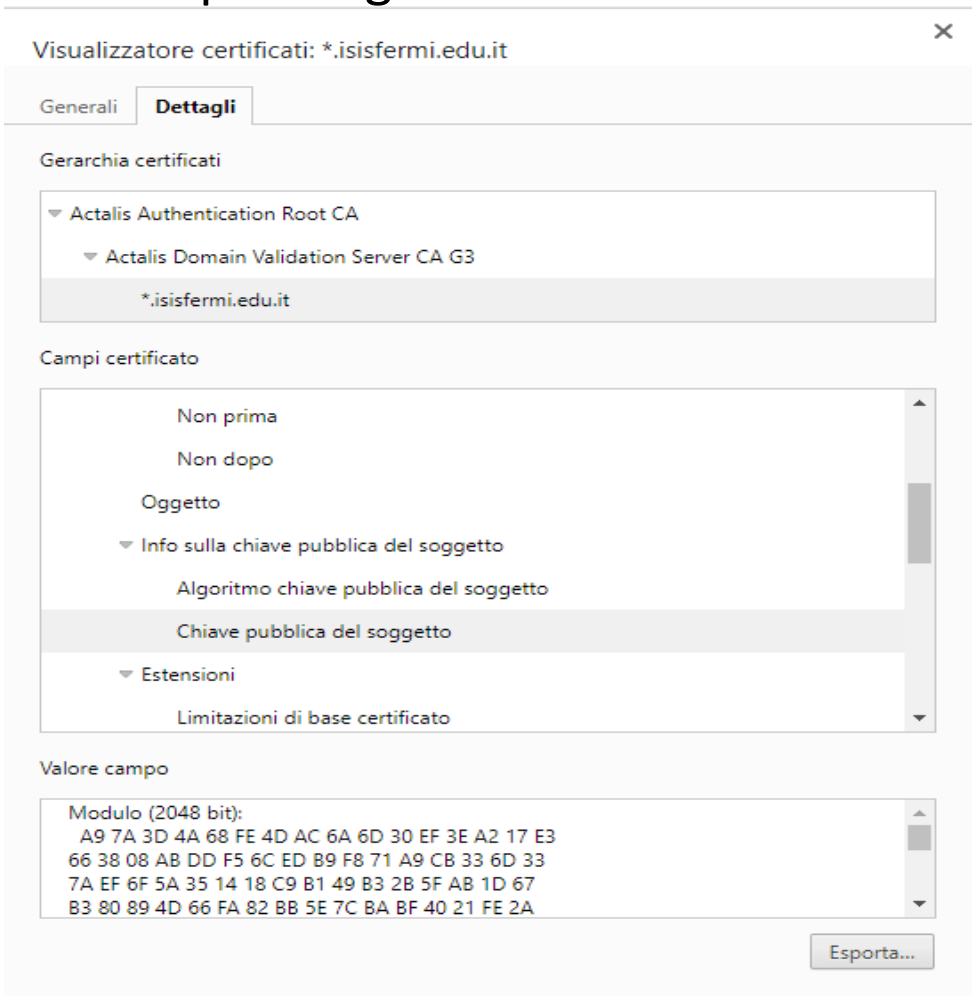
Connessione HTTPS tra Browser e Server - 14

Ora vediamo come avviene la **connessione HTTPS** tra client browser e sito web, prendendo come esempio il sito https://www.isisfermi.edu.it/

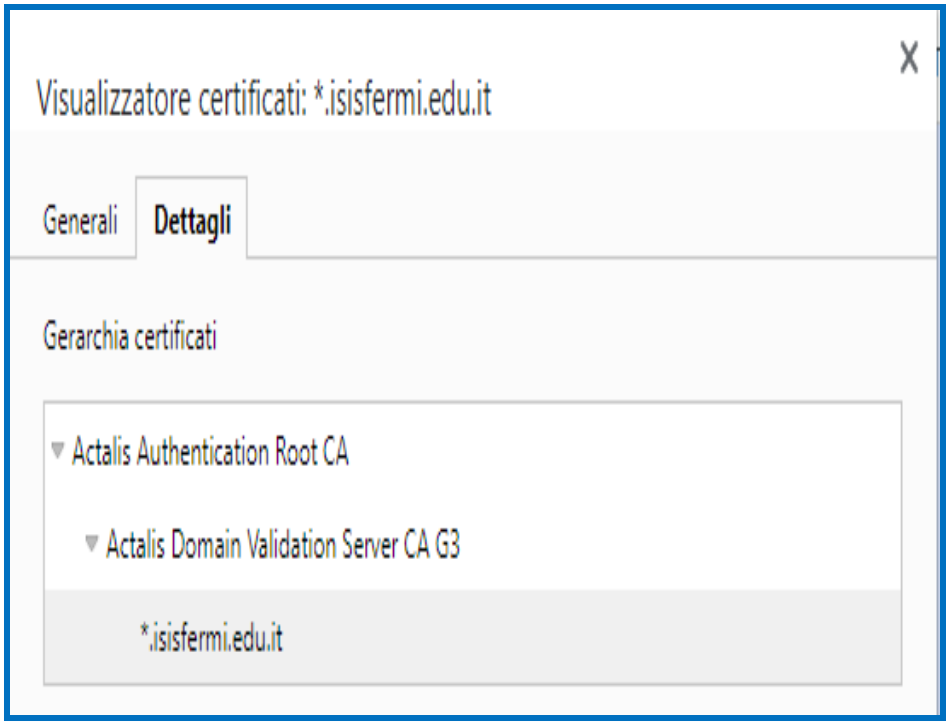
e **Actalis Domain Validation server CA G3** come **CA - Certification Authority**.

Cliccando sul **lucchetto grigio** davanti alla url e poi scegliendo **Certificato**:





Connessione HTTPS tra Browser e Server - 15



Actalis è una **Certification Authority** qualificata per l'erogazione di servizi certificati e riconosciuta in tutto il mondo per l'emissione di **certificati SSL Server**.
Dal 2009 parte del **Gruppo Aruba**, la cui capogruppo è **Aruba S.p.A.**, società italiana leader nei servizi di data center, cloud, web hosting, email e registrazione di nomi di dominio, **Actalis** oggi serve importanti funzioni della Pubblica Amministrazione Centrale italiana, oltre ad aziende private di ogni settore e dimensione a livello internazionale.
Sito: https://www.actalis.com/

Durante una connessione **HTTPS** avvengono **due passaggi**:

- L'**handshake** (letteralmente **stretta di mano**) per **convalidare** il **certificato** del **sito web**.
- La **creazione di una connessione sicura** tra **client** e **sito web**.

Handshake - 16

Ecco cosa succede durante la **stretta di mano** tra **browser** e **sito web**:

- Il **Client** invia al **Server HTTPs** un **ClientHello message** contenente la versione di **TLS/SSL** impostata sul client più la cosiddetta **cipher suite**, (insieme di codici), cioè un elenco dei protocolli di **Autenticazione** (tipo **RSA**), **Crittografia** (tipo **3DES**, **AES**, **IDEA**) con relativa **profondità di chiave** e **Funzioni HASH** (tipo **MD5** e **SHA**) **supportati** dal **browser** utilizzato per la richiesta.
- Il **Server** risponde inviando in chiaro una **lista di Protocolli Scelti** (fra quelli supportati dal browser), la sua **chiave pubblica** + il **certificato digitale di autenticazione** (firmato da **Actalis Domain** con la sua chiave privata).
- Per **verificare l'autenticità del certificato**, il **browser** preleva dall'elenco dei **certificati radice o intermedi** la **chiave pubblica** di **Actalis Domain** e tenta di **decodificare la firma digitale del certificato** che è stata **crittografata tramite la chiave privata** di **Actalis Domain**.
- Se è in grado di **decifrare** la **firma digitale** del **certificato** passa allo step successivo di creazione di una **connessione sicura col server**, altrimenti mostra all'utente un avviso di **Certificato non Valido** come si può vedere dalla figura seguente.

Avviso Fallimento Handshake - 17

La connessione non è privata

Gli utenti malintenzionati potrebbero provare a carpire le tue informazioni da **untrusted-root.badssl.com** (ad esempio, password, messaggi o carte di credito). Ulteriori informazioni

NET::ERR_CERT_AUTHORITY_INVALID

☐ Invia automaticamente a Google alcune informazioni sul sistema e alcuni contenuti delle pagine per contribuire a rilevare app e siti pericolosi. Norme sulla privacy

AVANZATE

Torna nell'area protetta

Connessione Sicura a Chiave Simmetrica - 18

Come già detto, con la richiesta della pagina di **www.isisfermi.edu.it** , il server di **Isisfermi** invia la sua **chiave pubblica** al browser.

Tutti i dati cifrati con questa **chiave pubblica** possono essere decifrati solo dalla **chiave privata** di **Isisfermi**.

- Dopo aver **convalidato** il **certificato**, il browser crea una nuova **chiave simmetrica di sessione** (o una **pre-shared key Diffie-Hellman**) facendone una copia. Queste chiavi saranno usate per crittografare e decrittografare i dati.
- Il browser quindi **crittografa** la **chiave simmetrica di sessione** con la **chiave pubblica di Isisfermi** ed invia tutto al server di **Isisfermi** .
- Il server di **Isisfermi** decodifica la **chiave simmetrica di sessione** con la sua **chiave privata**.
- Ora server e browser hanno entrambi una copia della **chiave simmetrica di sessione** creata dal browser. Nessun altro ha questa chiave, quindi solo server e browser **possono cifrare** e **decifrare** i dati in **maniera sicura**.
- Quando **Isisfermi** invia dati al browser, prima li cifra con la chiave di **simmetrica di sessione** e il browser decodifica i dati con la sua copia di questa chiave. Lo stesso avviene se è il browser ad inviare i dati.
- **Client** e **Server** trasmettono i dati codificandoli con la **chiave simmetrica** concordata. In coda ad ogni messaggio l'**End Point autenticato** (il **server**, oppure **entrambi** in caso di **autenticazione bilaterale**), aggiunge la propria **Firma Digitale** che consiste in una **impronta SHA256 univoca** cifrata con la **chiave privata**. Il ricevente, utilizzando il **Certificato Digitale** del mittente, **verifica** la **Firma Digitale** che garantisce sia **l'integrità** del **messaggio** sia **l'autenticità** del **mittente**.

Funzionamento HTTPS - 19

Riepilogo passo dopo passo.

1. **Isisfermi** richiede un **certificato** a **Actalis Domain**.
2. **Actalis Domain** verifica che è davvero **Isisfermi** che sta effettuando la richiesta.
3. **Isisfermi** invia a **Actalis Domain** la propria **chiave pubblica**.
4. **Actalis Domain** usa la propria **chiave privata** per firmare digitalmente la **chiave pubblica** di **Isisfermi**.
5. **Actalis Domain** dà a **Isisfermi** la **chiave pubblica** firmata, questa rappresenta ora il **Certificato SSL/TLS** di **Isisfermi**.
6. **Lo studente** si collega col suo browser al sito web di **Isisfermi** ed inizia la fase di **handshake** (**protocolli** e **hash supportati**).
7. **Isisfermi** invia al browser il **Certificato SSL/TLS**.
8. **Utilizzando** la **chiave pubblica** di **Actalis Domain**, che si trova nell'archivio dei **certificati radice** del browser, viene verificata la firma di **Actalis Domain** sul **Certificato SSL/TLS** di **Isisfermi** .

Funzionamento HTTPS - 20

9. **Se tutto ok**, viene generata una **chiave segreta simmetrica di sessione** e si utilizza la **chiave pubblica** di **Isisfermi**, ovvero il **Certificato SSL/TLS**, per **crittografarla**.
10. **Viene** inviata a **Isisfermi** la **chiave segreta simmetrica di sessione cifrata** con la **chiave pubblica** di **Isisfermi** (in **alcuni** casi si utilizza **pre-shared key Diffie-Hellman**).
11. **Isisfermi decifra** la **chiave segreta di sessione** con la propria **chiave privata** e la trattiene.
12. **Il browser** dello studente e **Isisfermi** utilizzano la **chiave segreta simmetrica di sessione** condivisa per **cifrare** le **comunicazioni successive**.
13. **Così** viene raggiunta l'**autenticazione** e la **segretezza** e finché **Actalis Domain** è ritenuta una **CA attendibile**, lo studente può essere certo che i **certificati di Isisfermi** saranno **sicuri** e **affidabili**.