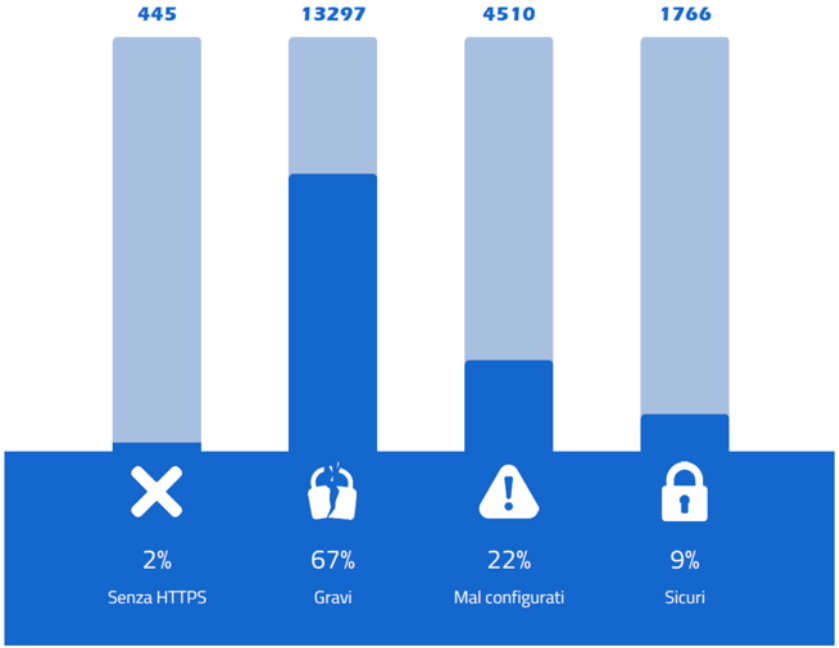
# Raccomandazioni AGID su TLS e CIPHER suites

Michele Petito, Agid

25/06/2021

# Risultati del monitoraggio sull'utilizzo del protocollo HTTPS

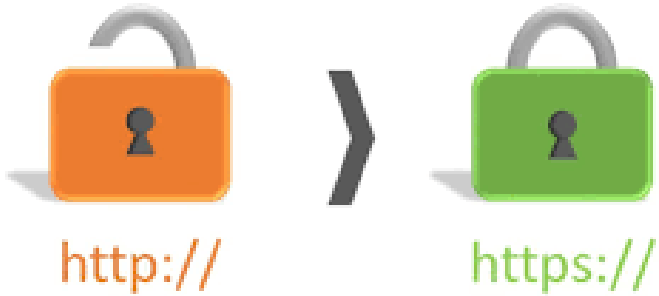
- **20.018 domini sottoposti a controllo**. Di cui:
  - **445 (2%)** portali istituzionali risultano senza HTTPS abilitato;
  - **13.297 (67%)** hanno gravi problemi di sicurezza;
  - **4.510 (22%)** hanno un canale HTTPS mal configurato;
  - **1.766 (9%)** utilizzano un canale HTTPS sicuro.




https://cert-agid.gov.it/news/monitoraggio-sul-corretto-utilizzo-del-protocollo-https-e-dei-livelli-di-aggiornamento-delle-versioni-dei-cms-nei-portali-istituzionali-della-pa/

# Cos’è Https

- È un **protocollo per la comunicazione sicura** con i siti web.
- **Garantisce:**
  - **riservatezza** e **integrità** dei dati scambiati tra i portali e i suoi utilizzatori.
  - **l’autenticazione del portale visitato**
- **L’attivazione è semplice** e richiede solitamente **1-2 giorni per un nuovo dominio**
  - In base al tipo di certificato richiesto alla Certification Authority, vengono effettuate differenti verifiche sull’identità dei richiedenti che influiscono su costo finale del certificato.
- **Navigare su un sito https non significa che questo sia sicuro**: potrebbe trattarsi pure di sito legittimo compromesso che contiene malware o un sito di phishing.



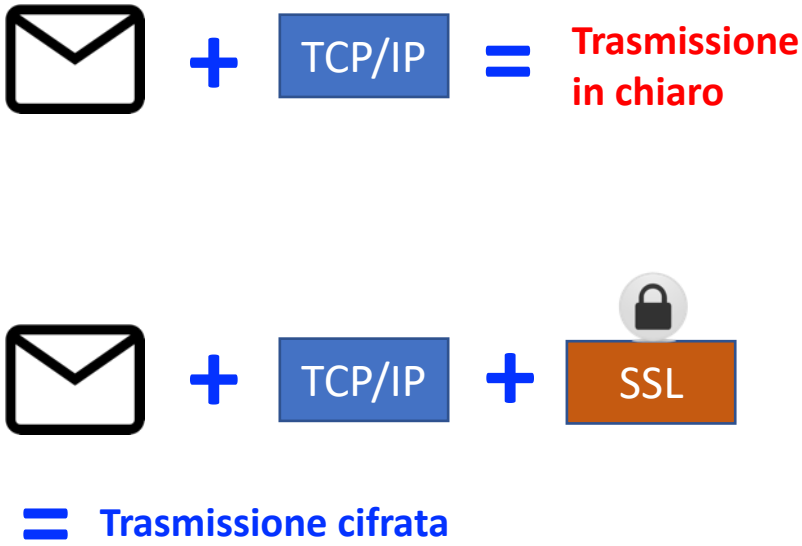
# Vediamo cosa c’è sotto il cofano…

- **Il lucchetto** è una semplice indicazione visiva, ma **non dice niente sul livello di sicurezza della trasmissione**
- **Dobbiamo quindi entrare più in profondità**. No panic, non farò un corso crittografia!
- La **conoscenza dei di meccanismi di base della crittografia, ci consente di navigare con più sicurezza e maggiore consapevolezza degli eventuali rischi** quando utilizziamo protocolli o algoritmi deprecati.

# Trasmissione dati cifrata (con SSL)

- **Per inviare un messaggio da un punto A e un punto B su Internet viene utilizzato il protocollo TCP/IP.** Prima del ‘94 i dati venivano trasmessi «in chiaro», quindi chiunque aveva l’accesso al canale poteva intercettare il traffico e leggere i messaggi scambiati.
- Nel **1994** viene introdotto dalla **Netscape** l’**SSL (Secure Sockets Layers)** una tecnologia standard che garantisce la sicurezza di una connessione a Internet.
- **SSL** oggi è ritenuto **insicuro** ed è stato aggiornato con una versione più recente detta **TLS (Transport Layer Security**)

- **Nota**:
  - i termini "SSL", "SSL/TLS" e "TLS" sono spesso usati in modo intercambiabile e in molti casi "SSL" viene utilizzato quando si fa riferimento al protocollo TLS più moderno.



# Applicazioni che usano TLS

## HTTPS

- è il protocollo che usa di più TLS

## SMTPS

- Simple Mail Transfer Protocol, è usato per l'invio di posta elettronica

## POP3S

- Post Office Protocol, usato per la ricezione della posta elettronica

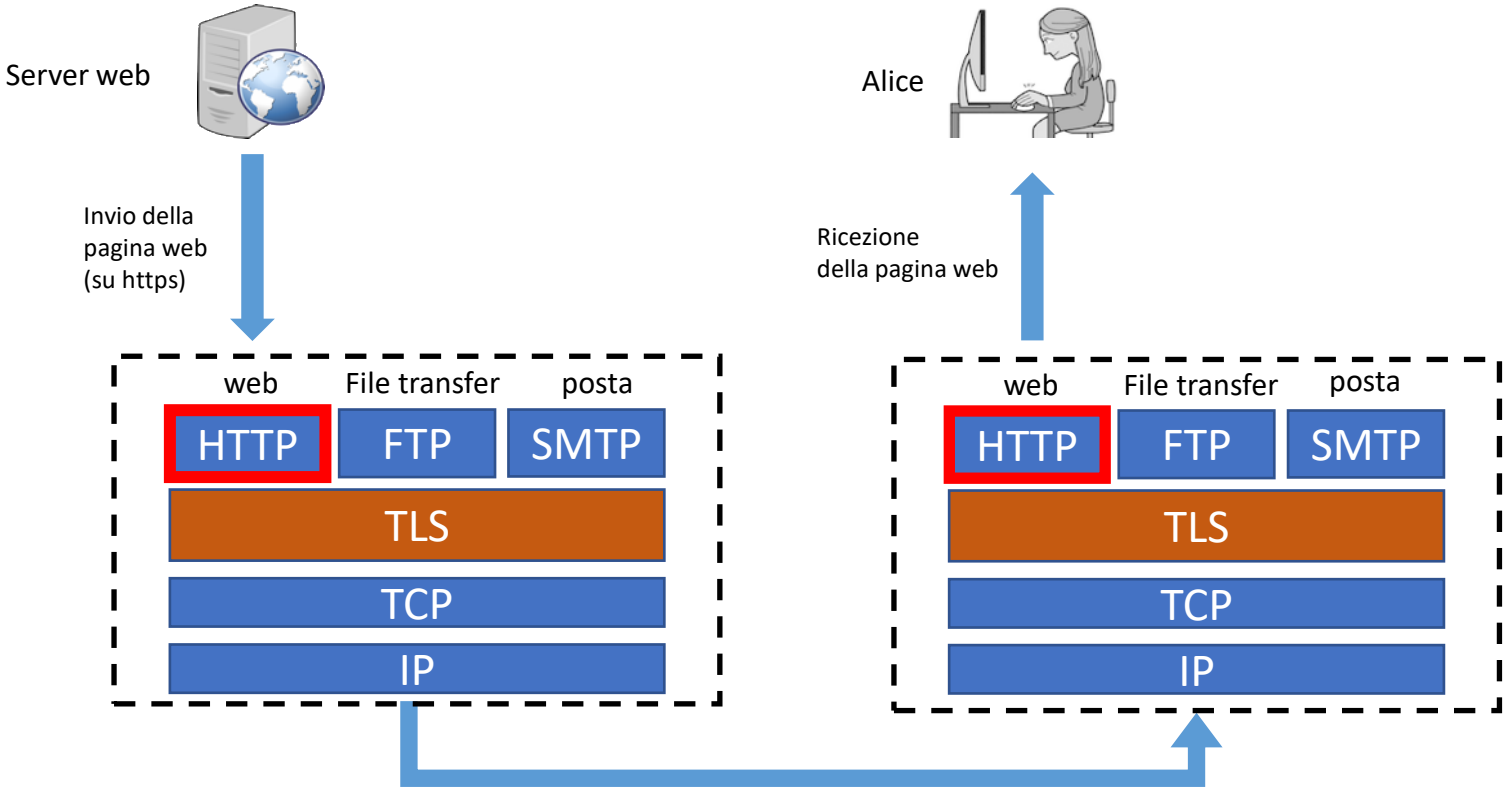
## IMAPS

- Internet Message Access Protocol consente l'accesso alla posta da remoto

## FTPS

- File Transfer Protocoll è utilizzato per il trasferimento di file da un computer e un server

# Trasmissione web cifrata (https)



# Storia dei protocolli SSL/TLS

- 1994: SSL v.1 (sviluppato da Netscape)
- 1995: SSL v.2 (sviluppato da Netscape)
- 1996: RFC 6101, SSL v.3.0 (vulnerabile a Poodle dal 2014)
- 1999: RFC 2246, TLS v.1.0 (sviluppato dalla IETF - *Internet Engineering Task Force* - come standard aperto)
- 2006: RFC 4346, TLS v.1.1
- 2008: RFC 5246, TLS v.1.2
- 2018: RFC 8446, **TLS V.1.3**

**Solo TLS 1.2 e 1.3 sono considerati sicuri.** I precedenti protocolli non supportano i moderni algoritmi crittografici e sono vulnerabili ad attacchi.

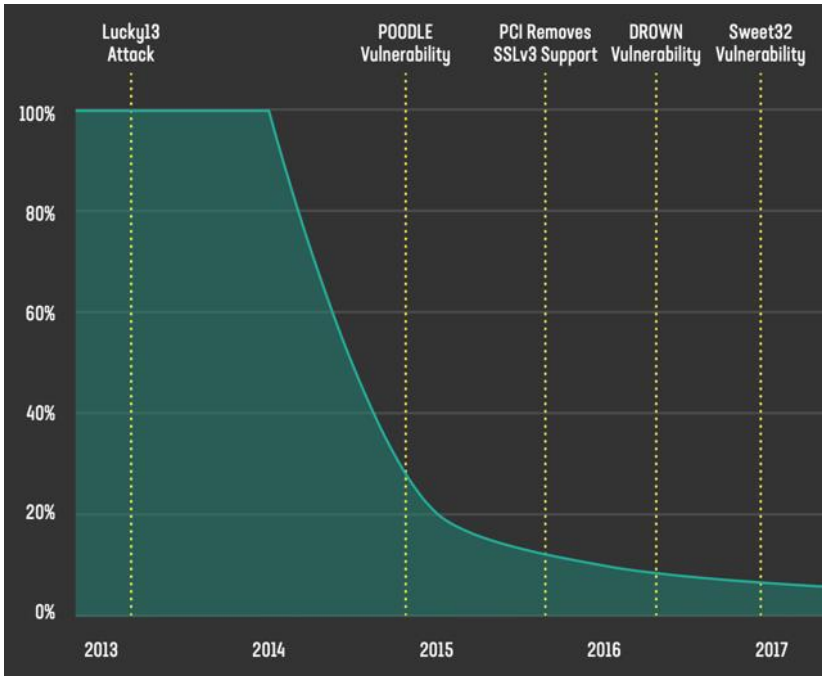
# TLS 1.0 e TLS 1.1 non più supportati dal 2018

- Lo standard **PCI DSS (Payment Card Industry Data Security Standard)** vie**ta l'utilizzo di TLS 1.0** e a partire dal **30 giugno 2018**. **Inoltre suggerisce di disabilitare TLS 1.1**. Questi protocolli possono essere interessati da vulnerabilità come **FREAK**, **POODLE**, **BEAST** e **CRIME** . Vengono inoltre vietati l'uso di cifrari deboli come il DES (che può essere violato in poche ore) e il RC4.
- Il 15 ottobre 2018 **Apple** decide di non supportare **TLS 1.0 e 1.1** nel suo browser **Safari** ma a marzo 2020, il supporto viene rimosso completamente grazie agli aggiornamenti di Apple iOS e macOS
- Marzo 2020 **Mozilla** ha disabilitato **TLS 1.0 e 1.1** in **Firefox**
- Il 15 gennaio 2020 **Microsoft** ha disabilitato **TLS 1.0 e 1.1** in **Microsoft 365** negli ambienti GCC High e DoD
- Il 31 marzo 2020 **Cisco** non supporta più **TLS 1.0 e 1.1** dai server **Umbrella** e dai loro servizi..

# Vulnerabilità note in SSL/TLS

- 23 ottobre 2011**:** **BEAST attack** → (cifrari CBC) → aggiornamento a **TLS 1.1**
- 15 settembre 2012 : **CRIME attack** → (compressione HTTP) → disattivazione della compressione
- 01 ottobre 2013 : **BREACH attack** (compressione HTTP) → disattivazione della compressione
- 7 aprile 2014 : **HEARTBLEED bug** (OpenSSL)
- 14 ottobre 2014 : **POODLE attack** (padding) → debolezza dei cifrari CBC che consente MITM (Man In The Middle) → **SSL muore**
- 8 dicembre 2014 : **POOPLE attack** (padding) → aggiornamento **TLS 1.2**
- 20 febbraio 2015 : **FREAK attack** → disattivazione dei cifrati di tipo «EXPORT» che consentivano MITM
- 20 maggio 2015 : **Logjam attack** (DH) → DH > 2048, ECDHE
- 20 maggio 2016 : **RACOON attack** → disabilitazione cifrari DHE (Diffie-Hellman Key Exchange)

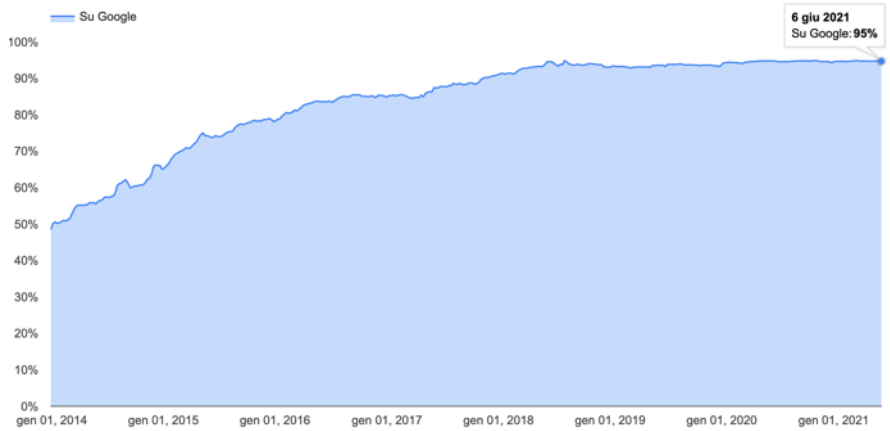
**% di server che accettano connessioni SSL v.3**



# Trend https

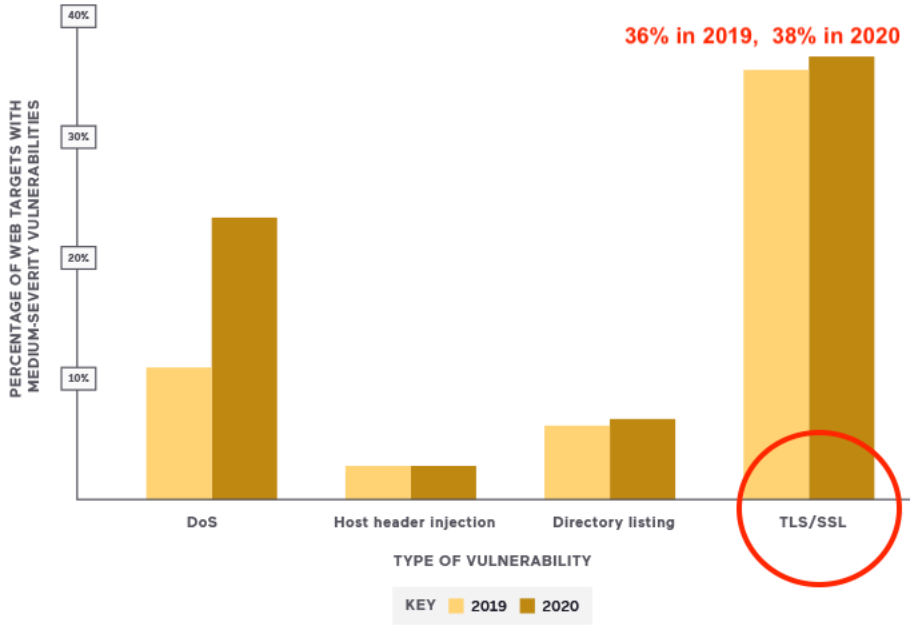
## Trend di crescita traffico https da dati Google

- **Al 6/6/2021** il **95% del traffico era su https**
- **I dati si basano solo sul traffico effettuato dai client Chrome**, quindi rappresenta una fotografia limitata dell'adozione di TLS dei server web.




Fonte: https://transparencyreport.google.com/https/overview

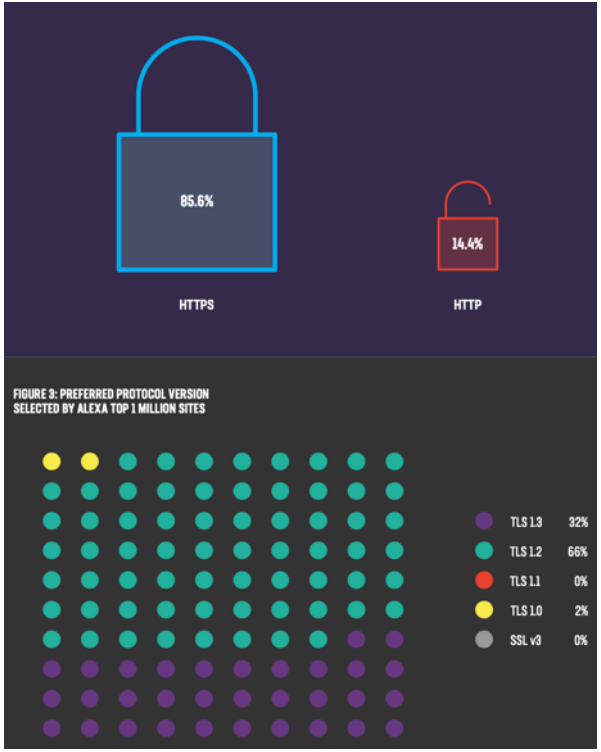
## % vulnerabilità su TLS/SSL



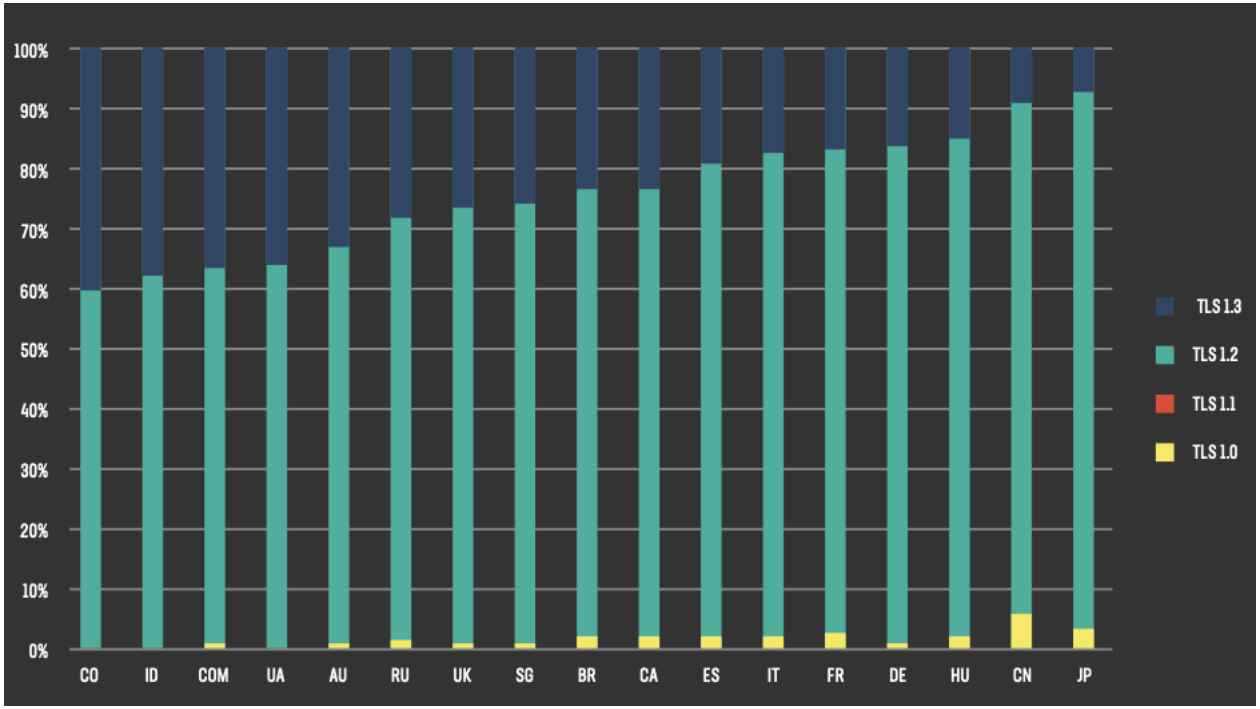

Fonte: *https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2021/*

# Adozione di TLS 1.1 e 1.2

## Adozione TLS




## Adozione TLS per the top-level domain (TLD)




Fonte: https://www.f5.com/content/dam/f5-labs-v2/article/pdfs/F5Labs-2019-TLS-Telemetry-Report-Summary.pdf

# Firefox

| Version | Platforms | SSL 2.0 (insecure) | SSL 3.0 (insecure) | TLS 1.0 (deprecated) | TLS 1.1 (deprecated) | TLS 1.2 | TLS 1.3 | EV certificate | SHA-2 certificate | ECDSA certificate |
|---|---|---|---|---|---|---|---|---|---|---|
| 24, 25.0.0<br>ESR 24.0–24.1.0 | Windows (7+)<br>macOS (10.12+)<br>Linux | No | Enabled by default | Yes | Disabled by default | Disabled by default [105] | No | Yes | Yes | Yes |
| 63–77<br>ESR 68 | | No | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| 78–88<br>ESR 78.0–78.10 | | No | No | Disabled by default[121] | Disabled by default[121] | Yes | Yes | Yes | Yes | Yes |
| ESR 78.11 89 | | | | | | | | | | |

Protocolli supportati

| BEAST | CRIME | POODLE (SSLv3) | RC4 | FREAK | Logjam | Protocol selection by user |
|---|---|---|---|---|---|---|
| Not affected | Mitigated | Vulnerable | Vulnerable | Not affected | Vulnerable | Yes[n 18] |
| Not affected | Mitigated | Not affected | Disabled by default [n 16] | Not affected | Mitigated | Yes[n 18] |
| Not affected | Mitigated | Not affected | Disabled by default [n 16] | Not affected | Mitigated | Yes[n 18] |

Vulnerabilità

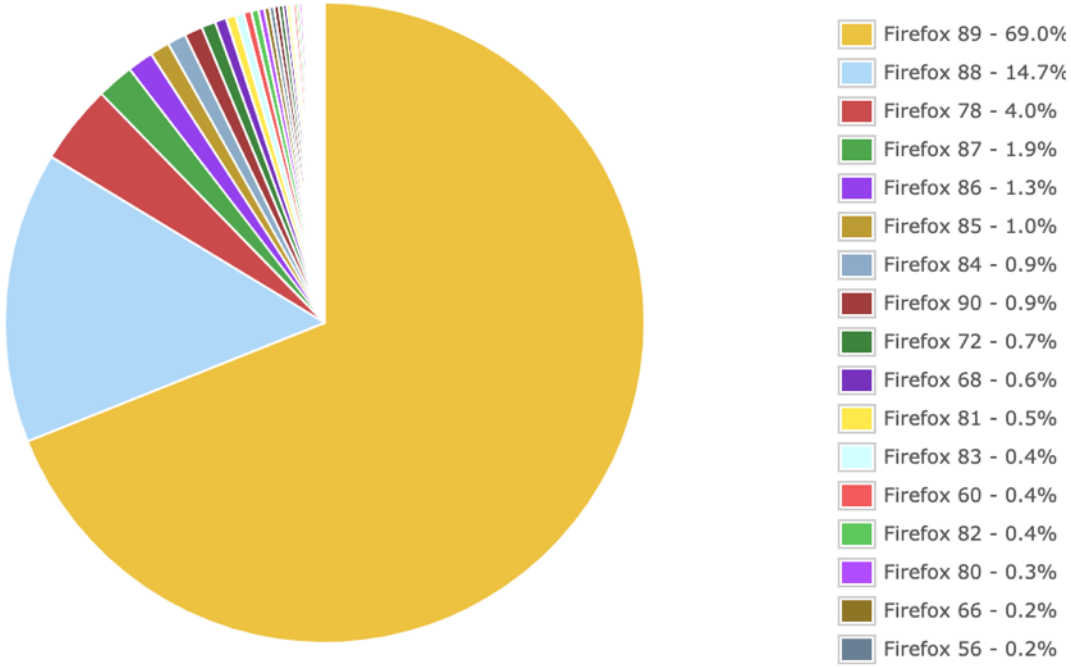
**Firefox v.25 del 2013**

**Firefox v.89 del 2021**

Rif. URL: https://en.wikipedia.org/wiki/Transport_Layer_Security#Attacks_against_TLS/SSL

# Distribuzione delle versioni Firefox nel mondo

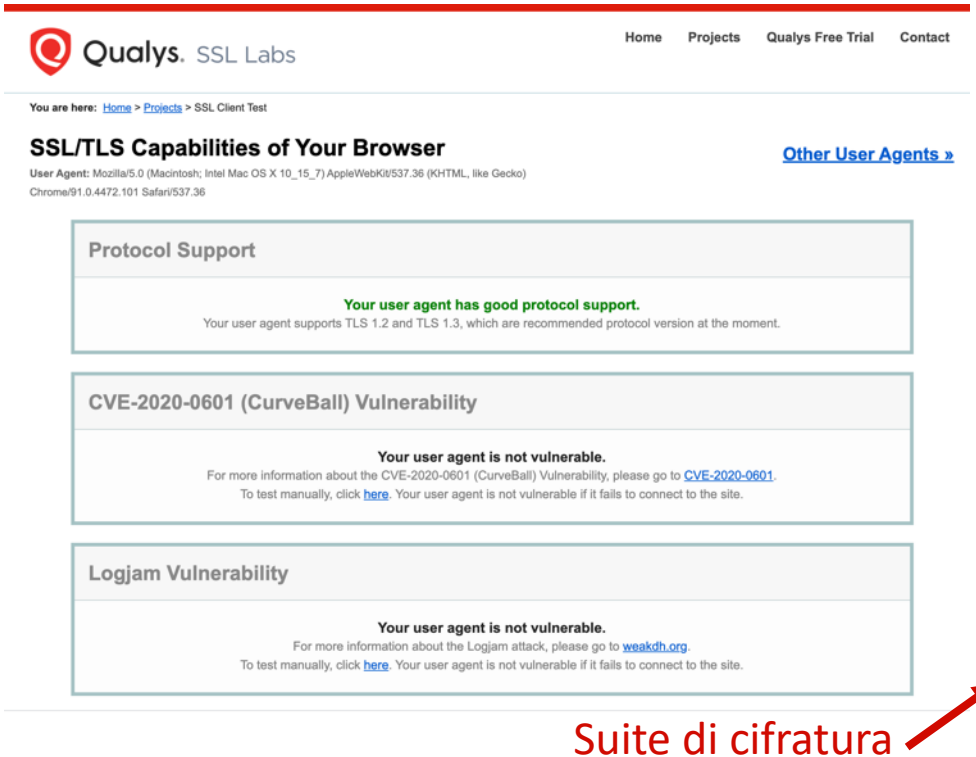
## al 20/6/2021

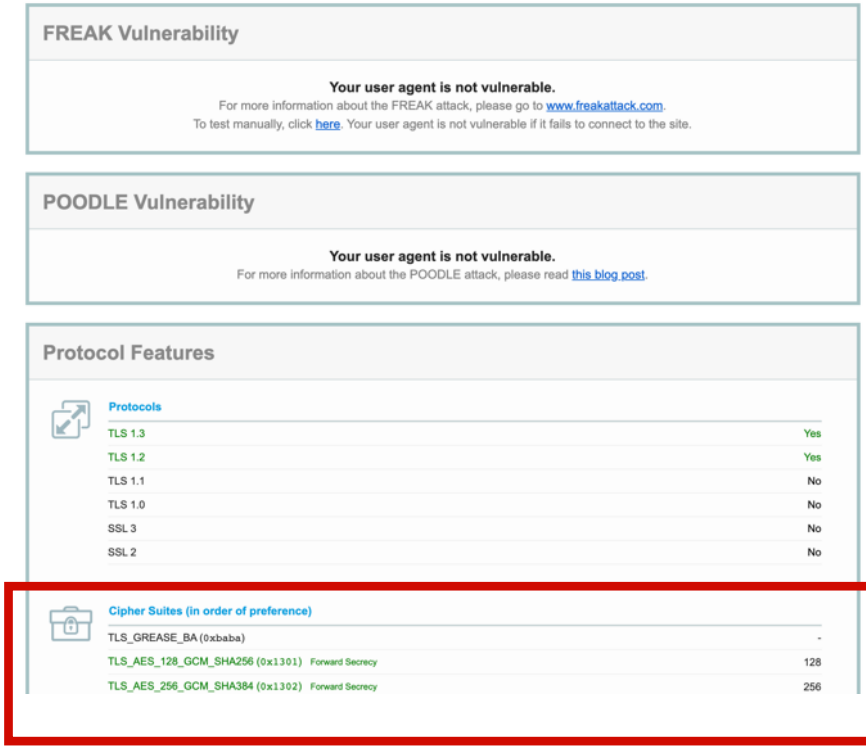



Fonte: https://firefoxgraphics.github.io/telemetry/

# Il mio browser è sicuro?

**https://clienttest.ssllabs.com:8443/ssltest/viewMyClient.html**




Suite di cifratura

# Cipher suite

- **Una suite di cifratura è un insieme di algoritmi che consentono di proteggere una connessione di rete**.
  - al momento della negoziazione di una sessione TLS il client presenta una serie di cipher suite supportate che propone al server il quale ne seleziona una
- Esistono molte **cipher suite per TLS 1.2** (alcune insicure, altre sicure), mentre in **TLS 1.3** sono solo 5 (attualmente ritenute sicure).
- In TLS 1.3 la struttura della cipher suite si è semplificata molto grazie all'utilizzo degli algoritmi di tipo **authenticated encryption with associated data (AEAD)** che forniscono contemporaneamente riservatezza e autenticità.

**Struttura di una cipher suite di TLS 1.2:**

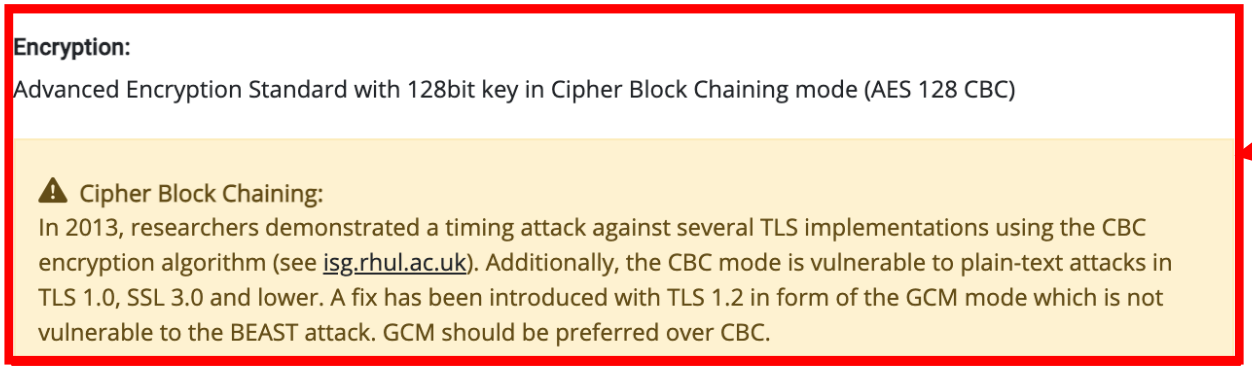

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

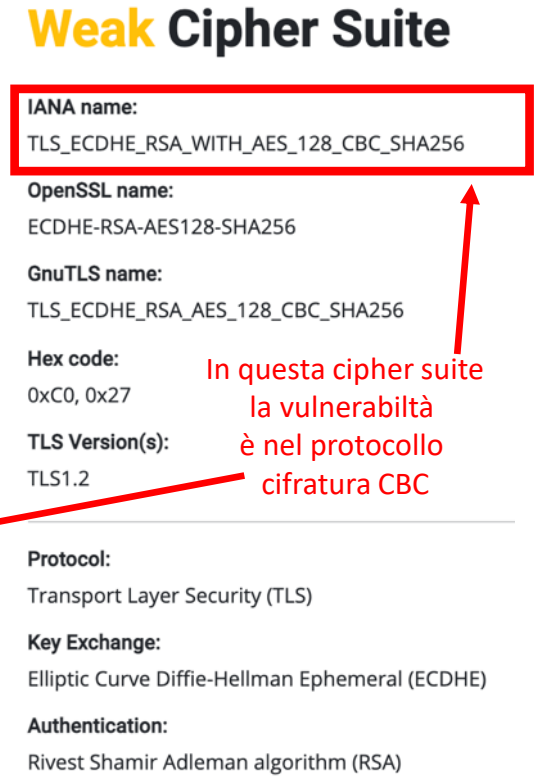
**Struttura di una cipher suite di TLS 1.3:**


TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_GCM_SHA256
TLS_AES_128_CCM_8_SHA256
TLS_AES_128_CCM_SHA256

# Ciphersuite.info

- **Tool on-line open source** sviluppato dal ricercatore di sicurezza tedesco Hans Christian Rudolph
- Fornisce **informazioni aggregate su una specifica cipher suite**, ad esempio:
  - I **nomi alternativi della suite** (IANA name, OpenSSL name ecc.)
  - La **versione TLS** a cui la suite appartiene
  - I **nomi degli algoritmi utilizzati** per lo scambio chiave, l'autenticazione, cifratura ed hash
  - **RFC di riferimento**
  - **Le eventuali vulnerabilità** presenti nella suite
  - Interrogabile anche tramite API




Tool open source: https://github.com/hcrudolph/ciphersuite.info

# Come viene scelta la suite di cifratura

**Il browser (client) invia un «hello» al server**. Nel messaggio vengono incluse:

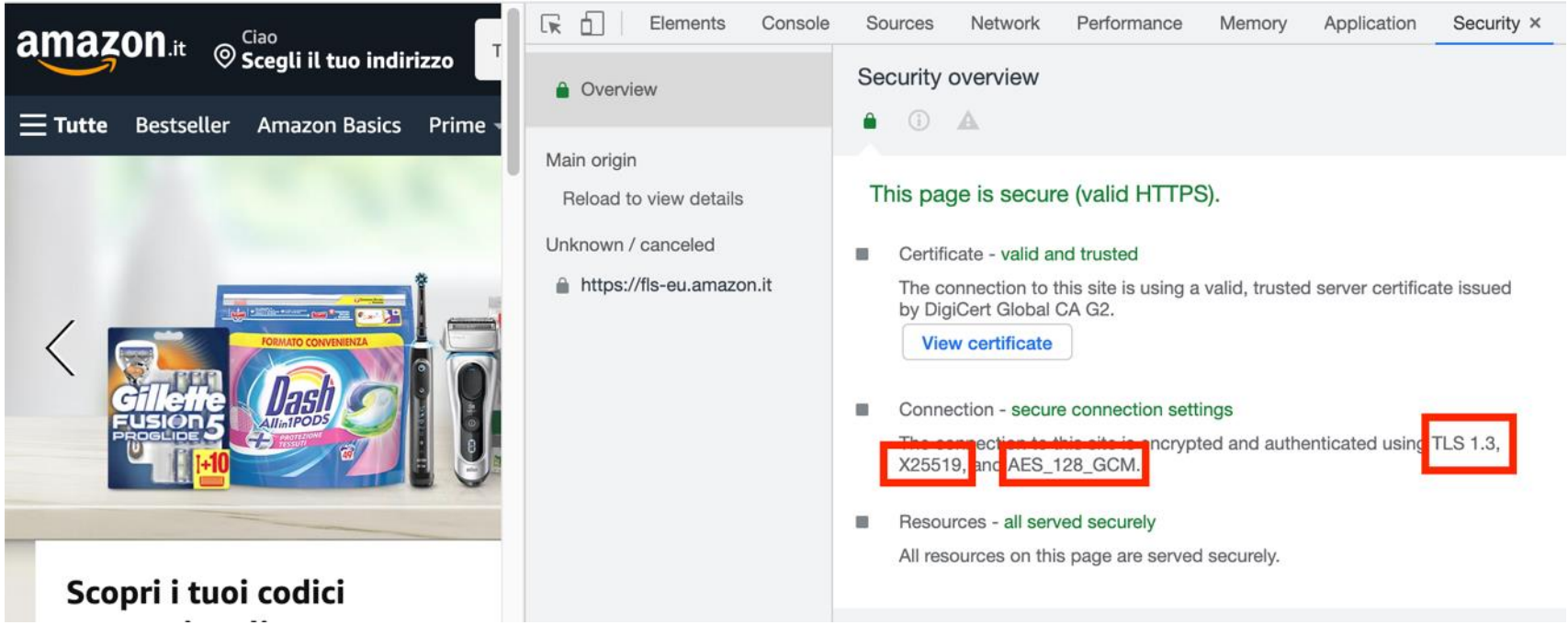
- **La versione TLS**
- **Le suite di cifratura** supportate dal client
- Altro

**Il server web invia al client un «server hello»** insieme a:

- il certificato SSL del server
- **la versione del protocollo TLS**
- **la suite di crittografia** scelta dal server

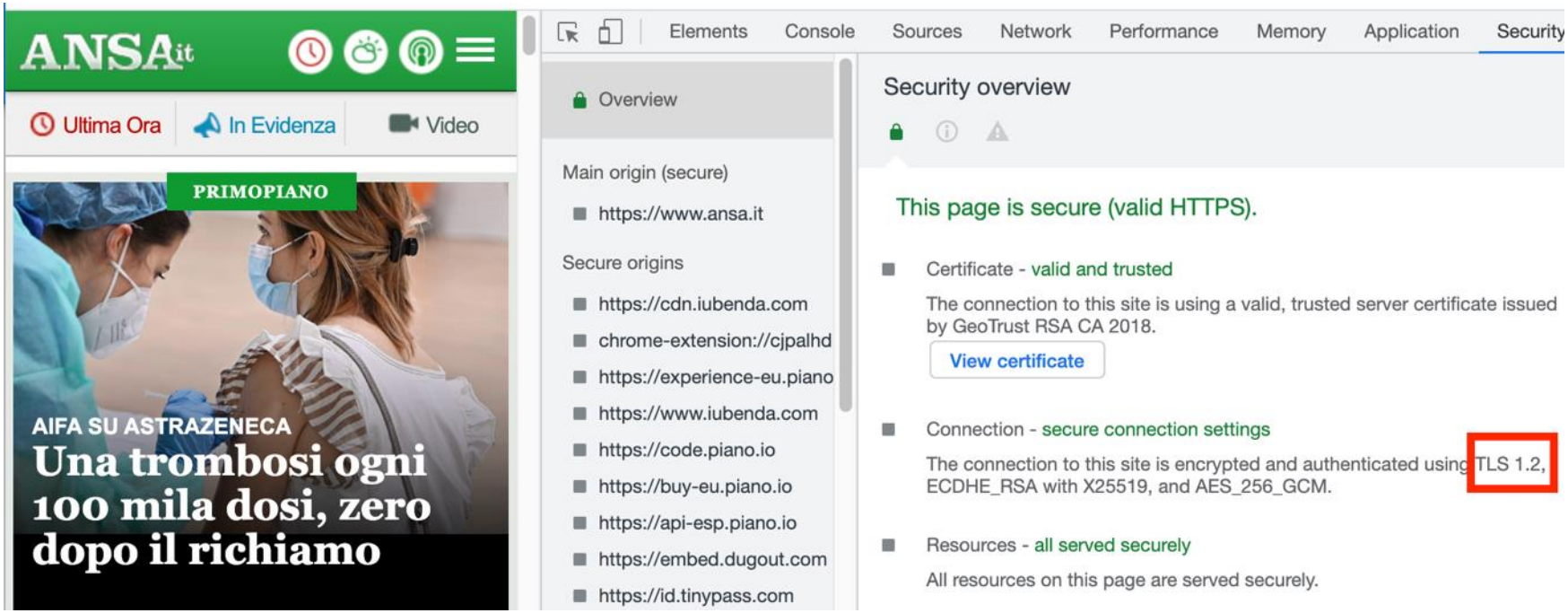
# Verifica della cipher suite utilizzata

## Esempio 1



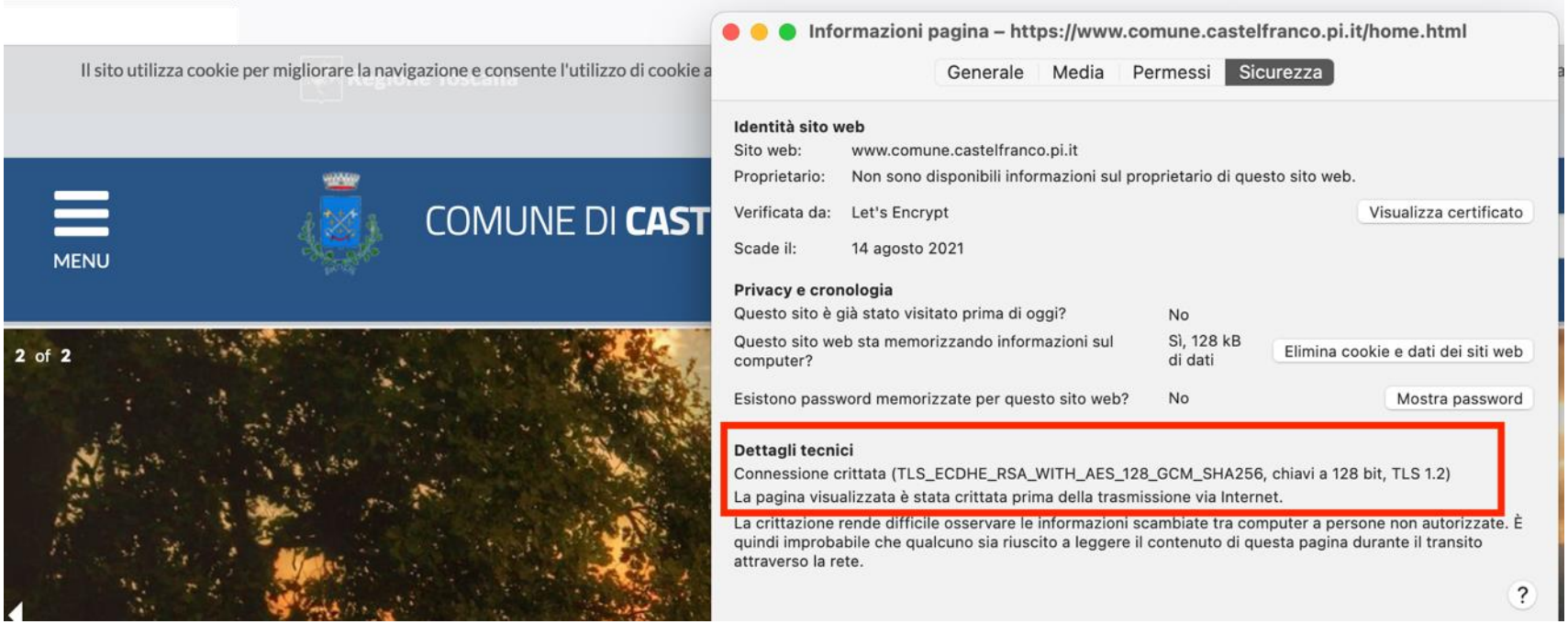
# Verifica della cipher suite utilizzata

## Esempio 2



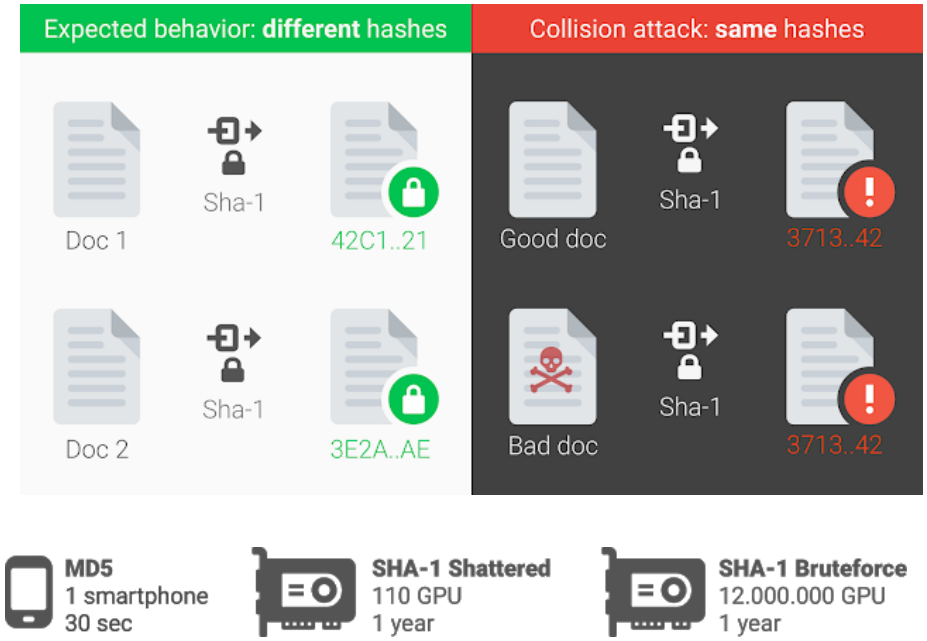
# Verifica della cipher suite utilizzata

## Esempio 3



# SHA1 collision

- **SHA1 è vulnerabile ad attacchi di collisione dal 2005**
- Ufficialmente **deprecato dal NIST nel 2011**
- Oggi gli algoritmi di hash sicuri sono lo **SHA-256 e SHA-3**
- Quali sistemi sono interessati?
  - Certificati digitali
  - Firme PGP/GPG
  - Checksum ISO
  - Sistemi di backup
  - Ecc.




Rif. URL: https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

# Algoritmi di firma e scambio chiavi

## ECC vs RSA / DH

- **Preferire se possibile algoritmi a curva ellittica ECC (Elliptic Curve Cryptography)** ritenuti più sicuri di quelli di prima generazione come **RSA e Diffie-Helllman**:
  - Il governo degli Stati Uniti la utilizza per proteggere le comunicazioni interne
  - Il progetto TOR per garantire l'anonimato
  - È il meccanismo utilizzato da Bitcoin per dimostrare la proprietà del proprio portafoglio (in particolare utilizza l'ECDSA con a curva secp256k1)
  - Consente di firmare i messaggi nell'app iMessage di Apple
- **Gli algoritmi ECC** possono essere usati sia per **cifrare la connessione** sia per **firmare i certificati al posto di RSA**.
- **I più noti algoritmi a curva ellittica sono:**
  - **ECDHE** (Elliptic Curve Diffie Hellman Ephemeral): algoritmo di scambio di chiavi diffie-hellman con curve ellittiche
  - **ECDSA** (Elliptic Curve Digital Signature Algorithm) : algoritmo di firma con curve ellittiche
    - **consente di di firmare con lo stesso livello di sicurezza di RSA ma con chiavi più piccole**. Secondo le raccomandazioni ECRYPT II sulla lunghezza della chiave, una chiave con curva ellittica a 256 bit fornisce la stessa protezione di una chiave asimmetrica a 3.248 bit.
    - **Le chiavi più** piccole **consentono di velocizzare la cifratura** e di conseguenza **migliora i tempi di caricamento delle pagine web**

# Lunghezza della chiave

- **La lunghezza della chiave è un parametro di sicurezza importante**
- Molte organizzazione accademiche pubbliche e private forniscono raccomandazioni e formule per approssimare la **lunghezza minima della chiave richiesta**
- Il sito https://www.keylength.com/ raccoglie le raccomandazioni sulle lunghezze delle chiavi di ECRYPT, NIST, NSA e altri.

| Metodo | Data | Simmetrico | Modulo di factoring | | Logaritmo discreto Chiave | Logaritmo discreto Gruppo | Curva ellittica | Hash |
|---|---|---|---|---|---|---|---|---|
| [1] Lenstra / Verheul | 2020 | 86 | 1881 | 1472 | 151 | 1881 | 161 | 171 |
| [2] Lenstra aggiornato | 2020 | 82 | 1387 | 1568 | 163 | 1387 | 163 | 163 |
| [3] ECRYPT | 2018 - 2028 | 128 | 3072 | | 256 | 3072 | 256 | 256 |
| [4] NIST | 2019-2030 | 112 | 2048 | | 224 | 2048 | 224 | 224 |
| [5] ANSSI | 2014-2020 | 100 | 2048 | | 200 | 2048 | 200 | 200 |
| [6] NSA | - | 256 | 3072 | | - | - | 384 | 384 |
| [7] RFC3766 | - | - | - | | - | - | - | - |
| [8] BSI | 2020-2022 | 128 | 2000 | | 250 | 2000 | 250 | 256 |

Tutte le dimensioni delle chiavi sono fornite in bit. Queste sono le dimensioni minime per la sicurezza.

# Best practice lato client e server

- **Lato client**
  - Utilizzare **sistema operativo supportato e aggiornato**
  - è sufficiente **utilizzare un browser aggiornato all'ultima versione**.
  - **I browser recenti utilizzano i cifrari più moderni e sicuri** e non consentono connessioni con quelli obsoleti e deprecati.
- **Lato server**
  - **Assicurarsi** di utilizzare un **sistema operativo supportato**
  - **Verificare** periodicamente la **presenza di aggiornamenti di sicurezza del sistema operativo** e di tutti i software installati, in particolare di **OpenSSL**
  - I software installati (database, cms, mail server, ecc) devono essere configurati per **impedire l'utilizzo di protocolli, algoritmi e cipher suite deboli e/o deprecati**.
  - **Tutte le connessioni verso client che utilizzano cifrari deboli e/o deprecati dovrebbero essere vietate**.

# moz://a SSL Configuration Generator

### Server Software

- ○ Apache
- ○ AWS ALB
- ○ AWS ELB
- ○ Caddy
- ○ Dovecot
- ○ Exim
- ○ Go
- ○ HAProxy
- ○ Jetty
- ○ lighttpd
- ○ MySQL
- ◉ nginx
- ○ Oracle HTTP
- ○ Postfix
- ○ PostgreSQL
- ○ ProFTPD
- ○ Redis
- ○ Tomcat
- ○ Traefik

### Mozilla Configuration

- ○ Modern
  Services with clients that support TLS 1.3 and don't need backward compatibility
- ◉ Intermediate
  General-purpose servers with a variety of clients, recommended for almost all systems
- ○ Old
  Compatible with a number of very old clients, and should be used only as a last resort

### Environment

| | |
|---|---|
| Server Version | 1.17.7 |
| OpenSSL Version | 1.1.1d |

### Miscellaneous

- ☑ HTTP Strict Transport Security
  This also redirects to HTTPS, if possible
- ☑ OCSP Stapling

```
    # intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;
```

# Verifica della corretta configurazione dell'https (lato server)

Avere l'https abilitato sul server non significa essere sicuri. E' necessario anche verificare che la sua **configurazione** sia corretta, ciò significa assicurarsi che il server utilizzi protocolli sicuri (TLS 1.2 e/o 1.3) e relative cipher suite. In particolare i tool di verifica andranno a effettuare vari handshake tramite diverse tipologie di client (browser firefox / chrome su linux, windows, ios, android ecc) per verificare:

- **Supporto dei protocolli** TLS 1.0, 1.1, 1.2
- **Validità del Certificato** : la validità del certificato, essenziale per la riservatezza e integrità del canale HTTPS
- **Funzionalità di «Autenticazione nulla»** : tale funzionalità dovrebbe essere DISABILITATA, in quanto se il server supporta l'autenticazione nulla (ovvero nessuna autenticazione dei parametri crittografici) può compromettere il canale HTTTPS.
- **Verifica dei cifrari** : alcuni cifrari deboli (quelli obsoleti o con lunghezza delle chiavi non più sufficienti) non andrebbero utilizzati. Alcuni cifrari deboli sono quelli che supportano la modalità CBC che sono potenzialmente soggetti ad attacchi noti.
- **Verifica della compressione**: la compressione andrebbe disabilitata. Tale funzione comprotte la sicurezza del canale HTTPS.
- **Verifica del redirect HTTP –> HTTPS** : questa funzionalità dovrebbe essere sempre abilitata e il redirect dovrebbe avvenire sullo stesso dominio o sottodominio, non verso siti terzi in quanto non è possibile in questo caso stabilire se il comportamento è voluto o dovuto ad attacco).
- **Verifica HTTPS –> HTTP** Questo redirect non dovrebbe mai avvenire in quanto rompe la catena di fiducia garantita da SSL/TLS.

# Possibili risultati del test

| | |
|---|---|
| A | L'implementazione HTTPS è sicura |
| B | L'implementazione HTTPS è aggirabile |
| C | L'implementazione HTTPS non è sicura ma non esistono attacchi noti che possono sfruttarla |
| D | L'implementazione HTTPS non è sicura ed è vulnerabile ad attacchi noti |
| E | Il server non supporta TLS |

# Tool per https lato server

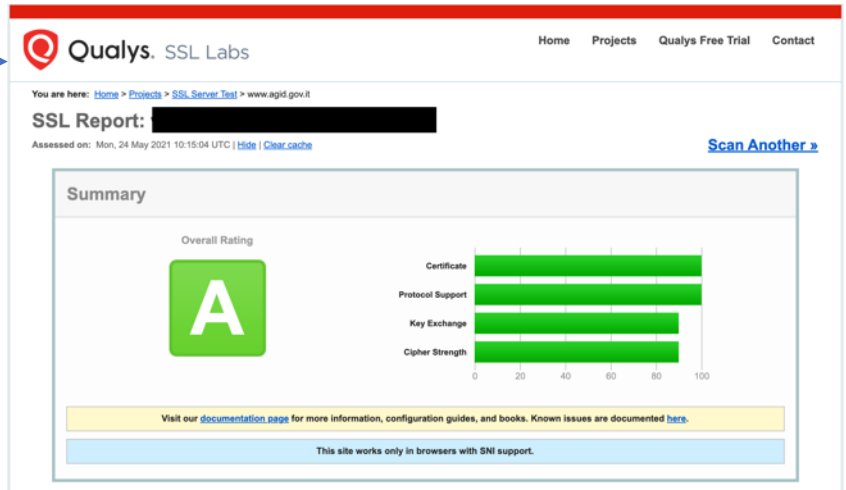
## consigliati da OWASP

### Tool off-line

- O-Saft - OWASP SSL advanced forensic tool
- CipherScan
- CryptoLyzer
- SSLScan - Fast SSL Scanner
- SSLyze
- testssl.sh - Testing any TLS/SSL encryption
- tls-scan

### Tool on-line

- **SSL Labs Server Test**
- CryptCheck
- CypherCraft
- Hardenize
- ImmuniWeb
- Observatory by Mozilla




Rif. URL: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html#test-the-server-configuration

# Tool per https lato server

## Risultati su cipher suites

**Cipher Suites**

| # TLS 1.2 (suites in server-preferred order) | | |
|---|---|---|
| TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) | ECDH x25519 (eq. 3072 bits RSA) FS | 128 |
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) | ECDH x25519 (eq. 3072 bits RSA) FS | 256 |
| TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) | ECDH x25519 (eq. 3072 bits RSA) FS | 256 |
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) | ECDH x25519 (eq. 3072 bits RSA) FS WEAK | 128 |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) | ECDH x25519 (eq. 3072 bits RSA) FS WEAK | 256 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) | WEAK | 128 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) | WEAK | 256 |
| TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) | WEAK | 128 |
| TLS_RSA_WITH_AES_256_CBC_SHA (0x35) | WEAK | 256 |
| TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) | ECDH x25519 (eq. 3072 bits RSA) FS WEAK | 112 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) | WEAK | 112 |

**Handshake Simulation**

| | | | | |
|---|---|---|---|---|
| Android 4.4.2 | RSA 4096 (SHA256) | TLS 1.2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Android 5.0.0 | RSA 4096 (SHA256) | TLS 1.2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Android 6.0 | RSA 4096 (SHA256) | TLS 1.2 > http/1.1 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Android 7.0 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Android 8.0 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Android 8.1 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Android 9.0 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| BingPreview Jan 2015 | RSA 4096 (SHA256) | TLS 1.2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Chrome 49 / XP SP3 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Chrome 69 / Win 7 R | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Chrome 70 / Win 10 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Chrome 80 / Win 10 R | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Firefox 31.3.0 ESR / Win 7 | RSA 4096 (SHA256) | TLS 1.2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Firefox 47 / Win 7 R | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Firefox 49 / XP SP3 | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH secp256r1 FS |
| Firefox 62 / Win 7 R | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |
| Firefox 73 / Win 10 R | RSA 4096 (SHA256) | TLS 1.2 > h2 | TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | ECDH x25519 FS |

UNIONE EUROPEA
Fondo sociale europeo
Fondo Europeo di Sviluppo Regionale

Agenzia per la Coesione Territoriale

Presidenza del Consiglio dei Ministri
Dipartimento della Funzione Pubblica

PON GOVERNANCE E CAPACITÀ ISTITUZIONALE 2014-2020

AGID Agenzia per l'Italia Digitale

# GRAZIE PER L'ATTENZIONE!

petito@agid.gov.it