# Firme digitali

Alessandro Pioggia

27 luglio 2022

# Indice

# Capitolo 1

# Introduzione

## 1.1 Che cos'è la firma digitale?

### 1.1.1 Dal punto di vista generale

La firma digitale rappresenta una tecnica, fondata su precisi principi matematici, che ha lo scopo di dimostrare l'autenticità di un documento digitale, garantendo:

- l'integrità dei dati in esso contenuti;
- l'autenticità delle informazioni relative al sottoscrittore;
- la non alterabilità del documento;
- le non ripudiabilità della firma.

Le caratteristiche sovraelencate implicano che, il sottoscrittore una volta apposta la firma:

- non potrà disconoscere il documento, il quale non potrà essere assolutamente modificato (una eventuale modifica ne annulla la validità);
- diventa l'unico titolare del certificato, dal momento che è in possesso delle credenziali per accedervi.

### 1.1.2 Dal punto di vista tecnico

La firma può essere realizzata attraverso protocolli crittografici simmetrici (a chiave privata) o asimmetrici (a chiave pubblica). La maggior parte delle applicazioni utilizzano procolli asimmetrici, perché permettono di creare soluzioni più semplici e computazionalmente efficienti rispetto agli altri.

## 1.2 Firma digitale vs analogica

Facendo una attenta ricerca, è possibile stabilire che, le proprietà godute dalle firme digitali sono presenti anche in quelle analogiche, eppure è risaputo che le prime sono decisamente più sicure ed affidabili rispetto alle altre, questo perché:

- la firma è falsificabile solo attraverso la conoscenza della chiave privata del firmatario, inoltre, modificando anche solo un bit del documento, esso perde di validità (per il motivo analogo, non è nemmeno riutilizzabile), di conseguenza è una tutela molto più forte rispetto alla firma analogica;
- l'autore non può negare la paternità della dichiarazione presente nel documento siccome, al momento della firma era l'unico in possesso della chiave privata.
- la firma dipende fortemente dal documento sul quale viene posta.

# Capitolo 2

# Certificati digitali

I protocolli a chiave pubblica portano tanti vantaggi, non è un caso il loro utilizzo pervasivo, in ambito comunicativo, a livello globale... detto ciò, c'è purtroppo un side-effect, derivato dalla natura del protocollo, ovvero il fatto che sia semplice ed immediato il recupero della chiave pubblica. L'esposizione della public key rende il sistema potenzialmente vulnerabile agli attacchi informatici del tipo key-only (il più noto è l'attacco man in the middle), il problema viene parzialmente risolto attraverso il rilascio di certificati digitali.

### 2.0.1 Attacco man in the middle

Data una comunicazione con protocollo a chiave asimmetrica fra due critto-analisti, l'attacco informatico consiste nell'intrusione di un terzo soggetto, che si interpone fra mittente e destinatario, a loro insaputa.

#### Scenario base

Supponiamo che Marco voglia comunicare con Riccardo, inconsapevoli della presenza di un intruso:

- Marco richiede la chiave pubblica di Riccardo attraverso l'invio di una mail;
- l'intruso, in attesa del momento giusto, intercetta la richiesta di Marco e risponde inviando la propria chiave pubblica, ingannandolo;
- l'attacco è quasi ultimato, a questo punto l'intruso rimane in ascolto e ciascun crittogramma inviato da Marco, viene cifrato con la chiave dell'intruso e ritrasmesso poi a Riccardo.

### 2.0.2 Metodo risolutivo

Come già citato, l'attacco può essere contenuto attraverso l'utilizzo di certificati digitali, ovvero dei documenti elettronici che attestano la relazione 1:1 tra la chiave pubblica e l'identità di un soggetto. Il certificato in questione contiene una vasta gamma di informazioni, le più significative sono:

- firma digitale dell'emittente;
- identità del proprietario;

- periodo di validità.

Ciò che rende i certificati digitali validi ed affidabili è il rilascio da parte di un ente terzo e fidato come autorità di certificazione, in sigla CA. Dal momento che si tratta di un trusted third party, si scongiura l'eventualità che il documento venga falsificato.

### 2.0.3 Certificate authority

Da wikipedia:

In crittografia, una Certificate Authority, o Certification Authority (CA; in italiano: "Autorità Certificativa"[1]), è un soggetto terzo di fiducia (trusted third part), pubblico o privato, abilitato ad emettere un certificato digitale tramite una procedura di certificazione che segue standard internazionali e in conformità alla normativa europea e nazionale in materia. Il sistema in oggetto utilizza la crittografia a doppia chiave, o asimmetrica, in cui una delle due chiavi viene resa pubblica all'interno del certificato (chiave pubblica), mentre la seconda, univocamente correlata con la prima, rimane segreta e associata al titolare (chiave privata). Una coppia di chiavi può essere attribuita ad un solo titolare. L'autorità dispone di un certificato con il quale sono firmati tutti i certificati emessi agli utenti, e ovviamente deve essere installato su una macchina sicura.

### 2.0.4 CRL

I certificati digitali hanno una scadenza, però possono andare incontro ad una revoca anticipata, l'insieme delle revoche sono catalogate in un certificate revocation list (CRL), che viene pubblicato e di conseguenza aggiornato periodicamente dalla autorità di certificazione. Lo stato di revoca si concretizza nel caso in cui si verifichi almeno uno dei seguenti casi:

- il CA rilascia in modo improprio un certificato;
- si sospetta che una chiave privata sia stata compromessa;
- inadempimento.

Il CRL dà dunque la possibilità di verificare la validità di un certificato, non è però l'unico, ad esempio è possibile interrogare direttamente la CA attraverso il protocollo OCSP.

# Capitolo 3

# Funzionamento

## 3.1 Fasi

### 3.1.1 Generazione impronta digitale

La prima fase consiste nella generazione dell'impronta digitale (o message digest), attraverso una funzione di hashing, che permette di ottenere una stringa di grandezza costante, è unica e non invertibile.

### 3.1.2 Generazione della firma digitale

Il risultato ottenuto nel passaggio precedente viene cifrato, attraverso la chiave privata, il risultato di questo procedimento è la firma. Nel passaggio successivo la firma viene allegata, insieme alla chiave pubblica al documento. Importante considerare che chiunque può verificare la firma, il caso più comune è quello in cui un giudice si impegna per risolvere un dissidio fra 2 soggetti.

### 3.1.3 Invio al destinatario

Il mittente invia il documento firmato attraverso il metodo indicato e il certificato (CA) al destinatario.

### 3.1.4 Verifica del certificato

Il destinatario verifica l'autenticità attraverso la propria copia della chiave pubblica di CA. Se la verifica va a buon fine, si procede con la decifrazione e consecutiva verifica della firma. In particolare, il ricevitore decifra il crittogramma ricevuto attraverso la sua chiave privata. Nel caso in cui il messaggio hashato, sia uguale alla tupla (funzione hash, chiave pubblica del mittente), la verifica va a buon fine.

#### Hash function

Il ruolo giocato dalla funzione hash è fondamentale in questo processo, le sue caratteristiche fanno in modo che, la probabilità che da documenti diversi, si possa ottenere la stessa impronta, è infinitesimale.

# Capitolo 4

# Implementazione della firma digitale con RSA

In questo capitolo verrà mostrata l'implementazione di firma digitale, nella pratica, attraverso la generazione di chiavi RSA. Si tratta di uno script in python, che sfrutta la tecnologia pycryptodome per la generazione della chiave. L'esempio comprende creazione, istanziamento e verifica della firma.

## 4.1 Generazione delle chiavi

Attraverso il metodo generate, fornito dalla libreria pycryptodome, si è in grado di generare agilmente la coppia di chiavi, pubblica e privata, sotto-forma di stringhe. Il metodo prende come argomento un intero, che rappresenta il numero di bit della chiave, per questo esempio è stata generata una coppia di chiavi a 128 byte, quindi 1024 bit.

```
from Crypto.PublicKey import RSA
from Crypto.Signature.pkcs1_15 import PKCS115_SigScheme
from Crypto.Hash import SHA256
import binascii

keyPair = RSA.generate(bits=1024)
privateKey, publicKey = keys.privateKey(), keys.publicKey()
```

## 4.2 Firma del messaggio (mittente)

```
#funzione che esegue lo hash del messaggio
def hash(message):
   return SHA256.new(msg)

#funzione che effettua la firma
def sign(hash, keys):
   signer = PKCS115_SigScheme(keyPair)
   return signer.sign(hash)

message = input("Insert the message that you'd like to sign")
signature = sign(hash(message))
```

In seguito alla generazione delle chiavi RSA, si procede con la firma di un messaggio arbitrario sfruttando la chiave privata. Il codice sovrastante esegue i seguenti passaggi:

- hashing del messaggio, sfruttando l'algoritmo SHA-512;
- generazione della firma, attraverso la chiave privata ottenuta precedentemente;
- creazione del crittogramma, che servirà poi al destinatario per effettuare la verifica.

## 4.3 Verifica della firma

```
#Se ritorna true -> firma valida, false -> firma non valida

def verify(message):
   hash = SHA256.new(msg)
   verifier = PKCS115_SigScheme(publicKey)
   return verifier.verify(hash, signature)

message = input("Insert the message that you'd like to verify")
verify(message)
```

In questa fase si effettua la verifica della firma, dunque viene effettuata la decrittazione attraverso la chiave pubblica e comparando l'hash della firma con l'hash del messaggio originale.

# Capitolo 5

# Falsificazione della firma

La firma digitale garantisce un ottimo grado di sicurezza, detto ciò, nonostante l'utilizzo di certificati rilasciati da enti di terze parti, in determinati casi è stato possibile falsificare la firma. Importante ricordare però che gli attacchi, nella maggior parte dei casi, sono efficaci perché nella creazione della firma digitale sono stati commessi degli errori. Le vulnerabilità più comuni:

- l'utente rende nota, probabilmente per inesperienza, la chiave privata;
- la manipolazione della procedura di firma;
- introduzione di un attore malintenzionato che si assicura che il firmatario veda qualcosa di diverso da ciò che intende firmare.

## 5.1 Universal signature forgery (USF)

L'USF è un attacco informatico, che sfrutta la proprietà di non falsificazione della firma digitale. In particolare, l'attacco consiste nel rendere appositamente invalida la firma durante la fase di verifica, aggiungendo al documento ulteriori dati. Si dice che l'USF "confonda" la logica di validazione, ovvero l'insieme di operazioni crittografiche che portano alla certificazione della firma. Se l'hacker riesce con il suo attacco USF, la logica di convalida online o l'applicazione di visualizzazione mostrerà che la firma elettronica è valida e appartiene ad un individuo o entità specifica sul suo pannello di visualizzazione.

## 5.2 Incremental saving attack (ISA)

Nel caso di un Incremental Saving Attack (ISA), l'obiettivo è quello di effettuare un salvataggio incrementale su un documento ridefinendone la struttura. Pertanto, l'obiettivo di questo attacco è il salvataggio incrementale o la funzione di aggiornamento incrementale di un documento PDF, che se utilizzata legittimamente consente a un utente di aggiungere annotazioni al proprio PDF. Queste annotazioni vengono salvate in modo incrementale come un nuovo corpo PDF dopo il contenuto originale del PDF. La funzione di salvataggio incrementale viene utilizzata anche per la firma del PDF e consente di aggiungere l'oggetto firma al contenuto del file originale. Normalmente, qualsiasi modifica dopo la firma di un documento attiverebbe un avviso che il documento è stato manomesso. Tuttavia, quando esegue un attacco ISA, l'autore dell'attacco potrebbe aggiungere contenuto aggiuntivo, come nuove pagine o annotazioni a un PDF già

firmato. Tecnicamente, questa violazione non è un attacco. Invece, è un exploit della funzione di salvataggio incrementale del PDF. Tuttavia, la vulnerabilità si verifica quando la logica di convalida della firma non rileva che il contenuto del file PDF è stato manomesso. Il contenuto non firmato che è stato aggiunto dopo la firma del documento è semplicemente visto come un aggiornamento dalla persona fisica o giuridica che ha originariamente creato la firma elettronica del documento. Un attacco ISA riuscito comporterà la visualizzazione di nuovi aggiornamenti di contenuto/corpo, mentre i processi di verifica della firma rimarranno ignari che sono state apportate modifiche o aggiornamenti al documento PDF.

## 5.3 Signature Wrapping (SWA)

Un attacco Signature Wrapping (SWA) utilizza un approccio unico per aggirare la protezione della firma di un PDF senza accedere alla sua funzione di salvataggio incrementale. Lo fa spostando la seconda parte del /ByteRange firmato alla fine del documento violato. Nel frattempo, l'attaccante riutilizza quindi il puntatore xrif all'interno del trailer firmato del documento per fare riferimento al suo xrif manipolato. In alcuni casi, l'attaccante può anche avvolgere la seconda parte riposizionata con un oggetto stream o un dizionario per impedirne l'elaborazione da parte del PDF o della funzione di protezione della firma online. In un attacco SWA riuscito, un utente malintenzionato può aggiungere oggetti dannosi non firmati nel documento. Se ha scelto di avvolgere la seconda parte spostata, questi oggetti possono essere posizionati prima o dopo l'xrif manipolato. Se non viene aggiunto alcun wrapping, gli oggetti dannosi verranno posizionati dopo l'xrif manipolato. A seconda del visualizzatore PDF, l'autore dell'attacco può copiare l'ultimo trailer del file e inserirlo dopo il suo xrif manipolato per consentire l'apertura del file PDF e per ignorare la verifica della firma senza che le manipolazioni vengano rilevate.

## 5.4 Il paradosso della chiave privata

L'utilizzo di una chiave privata, nei protocolli asimmetrici, ne garantisce il funzionamento e dunque è una componente fondamentale nel garantire una comunicazione sicura. Detto ciò però, è necessario memorizzarla in un dispositivo fisico, ad esempio il computer dell'utente a cui è stata rilasciata e questo comporta una vulnerabilità, ovvero il fatto che la sicurezza della chiave privata dipenda fortemente dal dispositivo in cui è memorizzata. A questo proposito, negli anni si è cercato di ovviare al problema, proponendo diverse soluzioni, le più note:

- la smart card;
- l'otp (la one time password);
- token usb;
- token wireless.

### 5.4.1 Smart card

Una smart card è una carta, generalmente dalle dimensioni di un comune bancomat, che possiede al suo interno, un insieme di dispositivi hardware che permettono di gestire ed elaborare dati, seguendo precisi standard di sicurezza. Ho deciso di citarla perché è possibile memorizzare al suo interno la chiave privata. In particolare la carta, interfacciandosi con un lettore esterno, riceve l'hash calcolato, lo firma con la chiave privata e lo restituisce. Un ulteriore layer (o strato) di sicurezza è dato dal fatto che prima di ogni utilizzo, è necessario inserire un pin, in modo da

pregiudicarne un utilizzo malevolo da parte di un malintenzionato (è inoltre possibile revocare la validità del certificato della carta in caso di furto).

#### Problema principale

Il noto Ronald Rivest, inventore del crittosistema RSA, ha rimarcato una contraddizione presente nel meccanismo di firma digitale attraverso questi dispositivi. Lo scienziato definisce il meccanismo "intrinsecamente insicuro", in quanto, nonostante la chiave privata sia memorizzata in un dispositivo sicuro, durante il procedimento di firma diventa vulnerabile, in quanto deve dipendere dalla sicurezza del dispositivo con cui si interfaccia. Per ovviare al problema, si potrebbe pensare ad un'applicazione perfettamente affidabile per la firma digitale che sia eseguibile solo ed esclusivamente su una tipologia di dispositivi stand-alone portatili, che non permettano l'esecuzione di altri programmi.

### 5.4.2 Dispositivi OTP (one-time-password)

Mobile OTP non è altro che una password usa-e-getta, con breve scadenza (pochi secondi dopo la creazione) utile per l'autenticazione online e l'accesso sicuro ai servizi di firma digitale online, anche detta firma digitale "remota". Questo servizio aggira la necessità di utilizzo di token o smartcard fisici per l'accesso a servizi di firma digitale, rendendolo particolarmente efficace per i dispositivi mobili come gli smartphone o i tablet. La OTP Mobile è un sistema basato sull'autenticazione a due fattori (Two-factor Authentication), ossia il sistema utilizzerà due evidenze per verificare, con un ampio grado di sicurezza, l'identità dell'utente. Una volta verificata l'autenticità dell'identità attraverso il sistema OTP Mobile, l'utente potrà accedere ai servizi di firma digitale remota messi a disposizione dai certificatori accreditati DigitPA.

#### Principali vantaggi

- semplicità, è sufficiente possedere uno smartphone;
- rapidità, l'attivazione è molto rapida rispetto alla firma digitale classica;
- sicurezza, per ovvi motivi, una password che scade dopo pochi secondi elimina il rischio della staticità e violabilità di una password tradizionale;
- accessibilità, il servizio è utilizzabile sempre, comunque ed ovunque.

#### Considerazioni

I sistemi protetti da OTP, se implementati in maniera impropria, possono risultare vulnerabili ad un attacco informatico, chiamato snooping. Lo snooping rappresenta una tecnica di hacking in cui, in una comunicazione di rete, fra stazioni remote, l'hacker è in grado di intercettare il traffico trasmesso. Il rimedio principale consiste nell'utilizzo dello standard https (secure http), il quale oscura i dati trasmessi, crittografandoli. Al giorno d'oggi è pressoché impossibile trovare siti di aziende rinomate, pubbliche o private, che non siano protette da questo standard, di conseguenza non si tratta di una minaccia.

### 5.4.3 Token USB

L'usb token è un dispositivo hardware, che permette di effettuare la firma, funziona in modo autonomo ,è sufficiente una uscita usb e comprende il software per la firma e per la gestione del dispositivo oltre ad una memoria di piccole ma sufficienti dimensioni. La caratteristica principale del token USB, è che viene consegnato direttamente dall'ente certificatore, che svincola il possessore del dispositivo dalla necessità di introdurre password temporanee o codici pin di accesso, in quanto è gestito tutto autonomamente dal dispositivo.

# Capitolo 6

# Bibliografia


- Wikipedia
- https://www.geeksforgeeks.org/types-of-digital-signature-attacks/
- https://www.pd.camcom.it/camera-commercio/notizie/avvisi-e-comunicazioni/Attacco-Malware-Spam-possessori-Firma-Digitale-Infocert
- https://shakeylead.com/en/digital-signature-vulnerabilities-and-protection-methods/