Le Chiavi ed i Certificati digitali

Crittografia e Chiave pubblica

Uno dei grossi problemi che affligge tutti noi nell'inviare un documento attraverso Internet è la certezza che nessuno, al di fuori del mittente e del destinatario, ne possa leggere il contenuto, questo perché durante l'invio del documento cifrato, in un canale non sicuro, si doveva **inviare anche la chiave di codifica per permetterne la decodifica**. In questo modo, quindi, se è possibile accedere al documento, è **possibile accedere anche alla chiave utilizzata** per la sua codifica.

La chiave di codifica/decodifica del messaggio è composta da una serie di caratteri, che consente a chi ne è in possesso di leggere un determinato documento codificato con quella chiave.
Per capire meglio cos'è una chiave possiamo pensare la cifratura di un messaggio come alla porta di casa nostra, la porta può essere aperta solo grazie ad una determinata chiave che posso dare ad un'altra persona. Per accedervi dobbiamo avere la disponibilità di quella medesima chiave e non di un'altra.
Questo tipo di codifica viene detta **simmetrica**, ovvero **la chiave di codifica è identica a quella di decodifica**.
Dal 1975 circa è possibile codificare un documento utilizzando un sistema di chiavi detto **asimmetrico** (la chiave di codifica è diversa da quella di decodifica). Questo sistema si basa su una coppia di chiavi:

- Una **pubblica** che può essere fornita molte persone in quanto utilizzata per la codifica del messaggio da inviare.
- Una **privata** che deve essere mantenuta segreta in quanto dovrà essere utilizzata per la decodifica dei messaggi.

Le due chiavi, sono legate tra loro tramite un algoritmo matematico complesso e da una chiave non è possibile risalire all'altra, in quanto sono formate da un elevatissimo numero di bit e anche il computer più potente impiegherebbe anni per ricavarne la corretta cifratura.
Con questo sistema possiamo codificare un messaggio attraverso la chiave pubblica del destinatario e solamente la chiave privata associata alla pubblica che ha codificato il messaggio potrà rendere leggibile il documento.
Un programma diffuso che opera questo tipo di codifica è **PGP *(Pretty Good Privacy)***. **PGP** associa la codifica tradizionale a quella a chiave pubblica, in quanto la codifica e la decodifica dei dati è molto più veloce con il sistema di codifica tradizionale. Per fare ciò **PGP** ad ogni sessione di codifica crea una chiave che utilizzerà per crittografare il documento, a seguito cripta la chiave appena utilizzata con la chiave pubblica del destinatario e invia sia il documento che la chiave. **In questo modo, però, anche se documento e chiave vengono intercettate il messaggio non potrà essere decifrato**, salvando la sicurezza dei propri dati.

A questo punto rimane solo una domanda….. *"dove si possono trovare le chiavi pubbliche?"*

Se si utilizza PGP questo non è un problema, infatti procedure automatiche consentiranno il reperimento delle chiavi. Nel caso non si abbia a disposizione PGP esistono dei grossi server detti **Keyserver** i quali, attraverso database dedicati, permettono la ricerca delle chiavi.

Esempio di Criptatura/Decriptatura di un File con chiave simmetrica

```
openssl aes-256-cbc -a -salt -in secrets.txt -out secrets.txt.enc
```

Come funziona?

OpenSSL è il comando per il toolkit.

- **-aes-256-cbc** è il tipo di crittografia da utilizzare (256bit AES)
- **-a** (è opzionale) dice che l'output crittografato potrà essere visualizzato in un editor di testo
- **-salt** obbligatorio nell'utilizzo della crittografia.
- **-in secrets.txt** specifica il file di input .
- **-out secrets.txt.enc** specifica il file di output .
- Verrà richiesta una **password**

```
openssl aes-256-cbc -d -a -in secrets.txt.enc -out secrets.txt.new
```

- **-d** decodifica dei dati .
- **-a** dice OpenSSL che i dati cifrati sono in base64 .
- **-in secrets.txt.enc** file dei dati da decifrare .
- **-out secrets.txt.new** file di destinazione in chiaro (con dati decriptati)

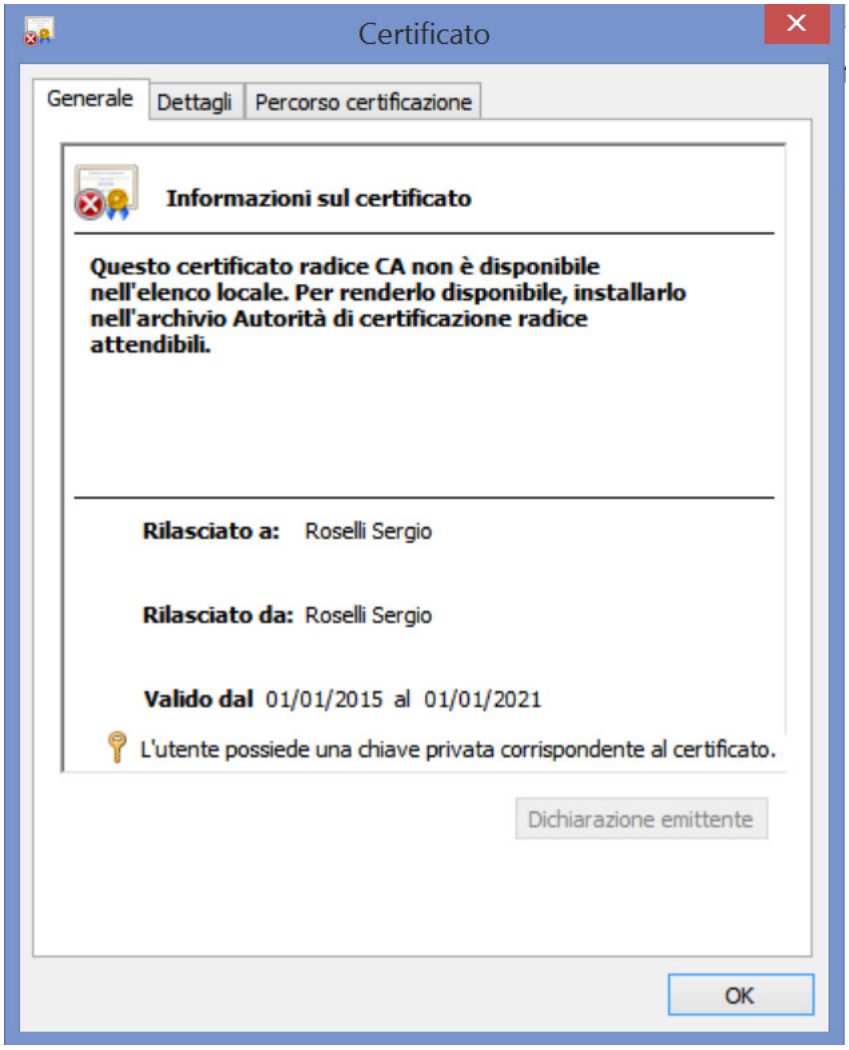
Esempio di Criptatura/Decriptatura di un File con chiave asimmetrica

Un esempio esaustivo di creazione di Chiave privata asimmetrica lo trovate qui

Certificato Digitale - Generare un certificato digitale con OpenSSL

Se volete implementare la connessione sicura via HTTPS al vostro server web oppure firmare digitalmente la corrispondenza, avete bisogno di un certificato digitale. Prima di mettere mano al portafogli, sappiate che potete generarvene uno in casa, con il quale studiare e svolgere tutte le prove del caso.

Volendo semplificare al massimo un argomento piuttosto complesso, possiamo riferirci ad un certificato digitale come ad un documento elettronico che lega una chiave crittografica all'identità di un determinato e specifico soggetto



Questo, per fare un esempio, impedisce che **Furfante** possa mandare un messaggio crittografato alla **Banca** spacciandosi per **Voistessi** senza che la **Banca** non se ne accorga.
Il modo "giusto" per entrare in possesso di questo oggetto è quello di rivolgersi ad una **Certification Authority (CA)** accreditata: si tratta di un'azienda che, a fronte del pagamento di un corrispettivo, verifica la vostra identità (tramite i vostri documenti d'identità) e vi rilascia a seguito il vostro certificato.

Se per motivi di studio o per meglio comprendere il funzionamento di questo meccanismo, desiderate crearvi "in casa" uno di questi certificati, allora alla fine di questa dispensa sarete in grado di raggiungere l'obbiettivo (sia in Linux che in Windows).

Esistono diversi modi per poter creare un file contenente dei dati casuali e noi utilizzeremo uno di questi : il comando **rand** di **OpenSSL**.

Tramite il programma **OpenSSL** impareremo a creare un certificato digitale. Si tratta di uno strumento gratuito, (anche a scopo commerciale), rilasciato con licenza open source "GNU/GPL" che include al suo interno alcune funzionalità crittografiche.

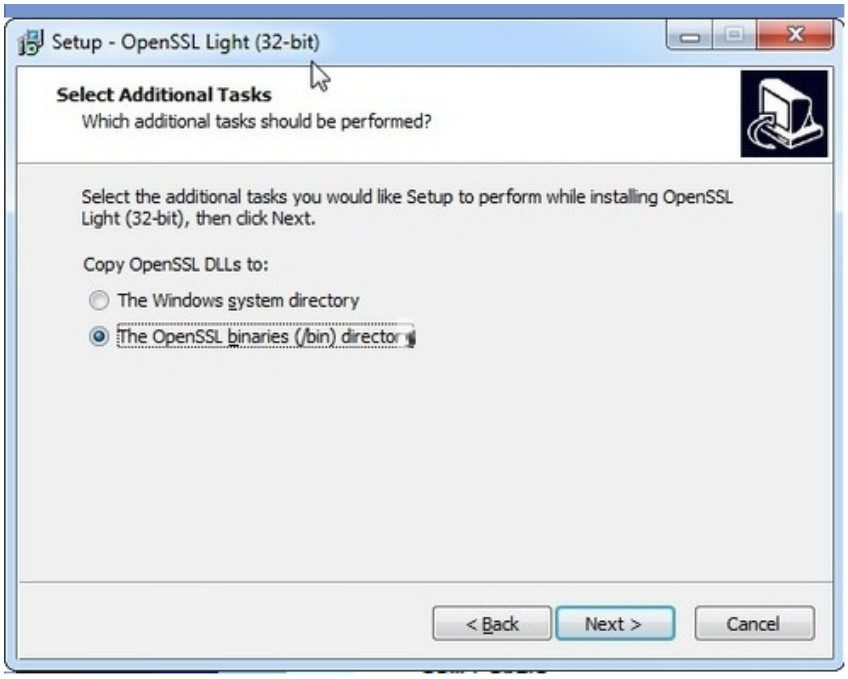
OpenSSL è un programma a linea di comando che consente di generare certificati digitali e chiavi crittografiche (simmetriche e asimmetriche). Esso ha la possibilità di implementare molte altre funzioni crittografiche legate al mondo privacy e della riservatezza dei dati nella comunicazione.

Gli utenti Linux e Mac OS X trovano il componente già installato o eventualmente possono recuperarlo mediante le funzioni preposte ed integrate nell'ambiente di lavoro.

Installazione Windows

Prima ancora di iniziare a predisporre OpenSSL su Windows, assicuratevi di aver installato "**Microsoft Visual C++ Redistributable Package**" in versione "2008" o successiva.
Procedete quindi all'installazione del software ***OpenSSL*.** Una volta aperto, accettate la licenza d'uso e specificate il percorso d'installazione ed il nome del gruppo di collegamenti da crearsi nel menu Start (potete accettare i predefiniti).

Alla domanda **Copy OneSSL DDLs to:** scegliere **The OpenSSL binaries (/bin) directory**.



Clicchiamo su neXt ancora un paio di volte quindi aprire la Shell dei comandi ed andate nella sottocartella **bin** della directory in cui avete installato **OpenSSL**. (**cd C:\openssl\bin**.)

Installazione Linux

In questo caso **OpenSSL è già presente** in tutte le distribuzioni.
Verifichiamo la presenza **OpenSSL** che dovrebbe essere già installato, digitando direttamente il comando **Openssl** da Terminale, se viene restituito il messaggio "***comando sconosciuto***" allora lanciate il comando ***sudo apt-get install openssl -y*** .
Una volta insallato **OpenSSL**, aprite un'istanza del Terminale ed assicuratevi di far precedere da **sudo** i vostri comandi se non state usando l'account **root**.

Generare un certificato

Per generare un certificato digitale ci si muove in modo pressoché identico indipendentemente che si stia lavorando sotto Linux o con Windows.

Eseguiamo un istruzione simile a questa :

```
openssl > req -x509 -nodes -days 3650 -newkey rsa:1024 -keyout C:\mycert.pem -out C:\mycert .pem
```

dove:

- -days 3650: indica che il certificato sarà valido per dieci anni (in altre parole, potrete utilizzarlo per giocare a vostro piacimento senza preoccuparvi che scada). Siete, naturalmente, liberi di scegliere un numero maggiore o minore;
- C:\mycert.pem: indica il percorso nel quale salvare il certificato che si andrà a generare.

Notate che lo stesso percorso dovrà essere indicato due volte: la prima dopo il parametro -keyout, la seconda dopo il parametro -out;

Una volta impartito il comando, vi verà richiesto di indicare alcuni dati anagrafici relativi all'identità dell'utente che state certificando.

Notate che l'unico realmente indispensabile è quello chiamato **Common Name** tutti gli altri possono essere saltati premendo il tasto Invio, in tal caso, verrà usato il valore di default mostratovi tra parentesi quadre ([]).

In particolare:

- Country Name (2 letter code) [AU]: specificare il codice di due lettere del Paese. Digitate **IT**;
- State or Province Name [Some-State]: il nome esteso del Paese. Nel nostro caso, **Italia**;

- Locality Name (eg, city) []: il nome della località. Usato **Bergamo**;
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: il nome della vostra organizzazione. Inserire ***miosito.it***;
- Organizational Unit Name (eg, section) []: l'unità operativa della quale fate parte all'interno dell'azienda. Usare **Amministrazione**;
- Common Name (eg, YOUR name) []: questo campo non può essere lasciato vuoto. Se state generando un certificato digitale da utilizzarsi per creare connessioni HTTPS, indicate qui l'hostname completo per raggiungere il servizio (ad esempio ***www.miosito.it***). In tutti gli altri casi, potete specificare il vostro nome e cognome;
- Email Address []: è il vostro indirizzo di posta elettronica;

Arrivati a questo punto, dovreste ritrovare il vostro certificato digitale pronto all'uso nel percorso indicato nel comando.
Sotto Windows, potete anche aprirlo con blocco note, salvare su un nuovo file con estensione **.cer** tutto il blocco compreso fra -----BEGIN CERTIFICATE----- e -----END CERTIFICATE----- (estremi inclusi) e quindi farvi doppio click per aprirlo direttamente ed osservarne i vari valori.

Mi segnala un errore, Come mai ?

Notate che il certificato digitale ottenuto con questa procedura fa scattare almeno un allarme di sicurezza quando utilizzato.
Il meccanismo di protezione rileva infatti che lo stesso non è stato rilasciato ad una Certification Authority riconosciuta (come descritto in apertura): di conseguenza, tale certificato non può essere considerato realmente attendibile.

È, ovviamente, tutto normale.

Per saperne di più

OpenSSL è un software davvero ampio e potente: per conoscere tutti i parametri utilizzabili e le opzioni disponibili, si può fare riferimento alla **documentazione ufficiale**.
A **questa pagina**, nello specifico, sono mostrati con dovizia di particolari tutti i parametri utilizzabili per la creazione di un certificato digitale come quello da noi generato.

Potremmo inoltre crearci un istruzione ad hoc per un nostro strettamente personale certificato a questo indirizzo

Il servizio **certmgr** (Gestire i certificati del computer) di windows vi permetterà di tenere sott'occhio e gestire i vostri ed anche gli altri certificati digitali.

Download

PGP (Pretty Good Privacy)
OpenSSL