



DESDE LAS HORAS EXTRAS

CRIPTOGRAFÍA, CERTIFICADOS Y PRIVACIDAD EN INTERNET

Apuntes del curso

Desde las horas extras

Realizado por
José Luis Bautista Martín

Junio de 2018



Esta hoja está en blanco para facilitar la impresión de este documento a doble página.



ACERCA DEL INSTRUCTOR

José Luis Bautista Martín, Ingeniero de Sistemas, con maestría en “investigación en ingeniería de Software”.

Mi experiencia laboral en cuanto a desarrollo de software abarca desde tecnologías “legacy”, hasta tecnologías de vanguardia, poniendo siempre una especial atención en la construcción de software escalable, modular y sostenible.

Igualmente estoy especializado en la interconexión de diversos sistemas y plataformas para conseguir una solución coherente entre la tecnología actual en producción y nuevas tecnologías del mercado.

Una de mis inquietudes actuales es simplificar el desarrollo de software, permitiendo mediante herramientas generadoras de código, patrones de software, programación orientada a aspectos o simplemente interfaces sencillas y claras que el programador se concentre en resolver los problemas propios de la solución a implementar (esto es, los requisitos de negocio a representar en forma de software) y no se tenga que preocupar de tareas repetitivas, generalidades de los sistemas, o problemas técnicos, que no hacen más que distraerle de sus verdaderos objetivos.



ACERCA DE ESTE DOCUMENTO

La misión de este documento es exponer los objetivos, mecánica y temerarios planteados, así como la documentación para el curso de CreSer “Criptografía, Certificados y privacidad en internet”.

OBJETIVOS DEL CURSO

Este curso tiene como objetivo, ser una introducción acerca de la realidad de la privacidad y seguridad en Internet, revisando los motivos y circunstancia (actuales) en la que esta es violada, para posteriormente presentar las herramientas de criptografía, y a los certificados como los medios actuales para garantizarla.



CONTENIDO

Acerca del instructor	3
Acerca de este documento	4
Objetivos del curso	4
Contenido	5
Metodología	6
Requisitos	6
Introducción	7
Breve historia de las criptografía	8
Breve historia de Internet	11
Seguridad en internet.....	14
Privacidad en internet.....	15
Caso espionaje NSA	16
Caso del teléfono el terrorista de San Bernardino.....	16
Caso Cambridge Analítica	17
Tipos de criptografía.....	18
Criptografía simétrica	18
Criptografía asimétrica	19
Certificados	22
Entidades certificadoras.....	22
Entidades intermediarias.....	23
Certificados personales	24
Configuración de certificados con GnuPG	25
Anexo I: Ejemplos de certificados	33
Anexo II: Extensiones de archivos criptograficos	34



METODOLOGÍA

Sera una conferencia de dos horas, en la que la primera hora se comentara sobre las circunstancias actuales de la seguridad en Internet, y la segunda tendrá una escenario más práctico al usar las herramientas de criptografía y generación de certificados.

REQUISITOS

No requiere ningún requisito en especial, si los asistentes quieren acompañar al instructor en la realización de los ejemplos de criptografía, deberán llevar una laptop, pero no es imprescindible.



INTRODUCCIÓN

- ¿Qué es una información secreta?
- ¿Qué es una información privada?

¿Qué es una información secreta?

Un secreto es una información que no debe ser compartida en prácticamente ningún ámbito. Pueden considerarse secretos elementos tales como NIP o password.

¿Qué es una información Privada?

Una información privada, es aquella que aunque no se es publica, se puede proporcionar en algunos ámbitos restringidos. Algunos ejemplos significativos son:

- Datos económica personales (como el salario o el patrimonio personal)
- Datos familiares (información de los hijos tal como escuelas, y horarios).
- Datos de índole medica.

Mucha de nuestra información (secreta y/o privada) viaja a través de internet, a diario y frecuentemente sin que seamos constantes de ellos.

Este curso tiene como objetivo ser una introducción acerca de la realidad de la privacidad y seguridad en Internet, revisando los motivos y circunstancia (actuales) en la que esta es violada, para posteriormente presentar herramientas de criptografía, y manejo de certificados.



BREVE HISTORIA DE LAS CRIPTOGRAFÍA

La palabra "criptografía" se compone de dos partes "cripto", que quiere decir oculto y "grafía", que quiere decir escritura.

La criptografía garantiza que un mensaje solo puede ser entendido por su destinatario, aunque otras personas puedan ver o conseguir dicho mensaje.

El origen de la criptografía se asocia frecuentemente con la política y con la guerra.

Uno de los algoritmos clásicos más populares es el de Julio Cesar, que consiste en desplazar tres posiciones en el abecedario cada letra.

En la historia moderna, la criptografía se ha usando ampliamente en los dos escenarios bélicos más representativos del siglo XX, pero sobretudo en la segunda guerra mundial.

La representación más importante de la criptografía en la segunda guerra mundial, es la maquina enigma:

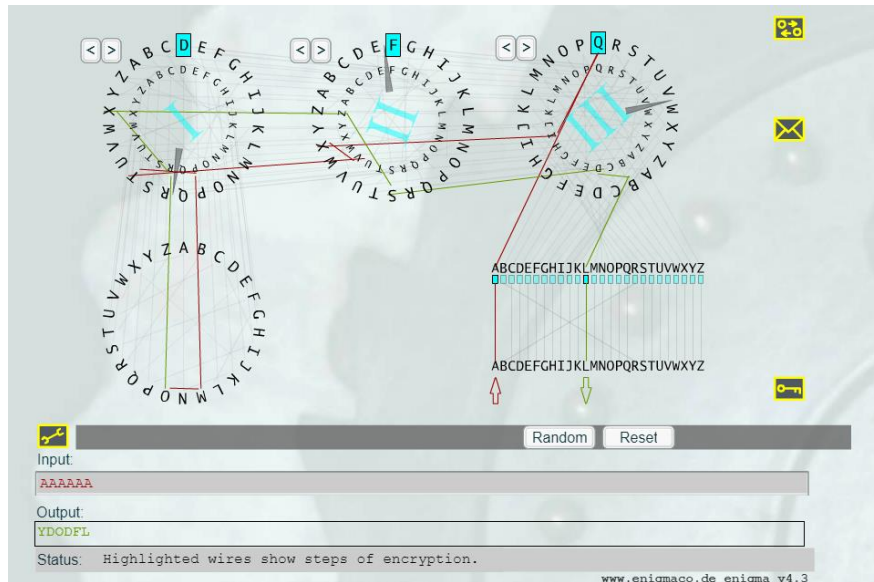


Es una maquina electromecánica de cifrado.

Tiene unos rotores en la maquina superior con el abecedario y un cableado que los conectaba.

Cada vez que pulso una tecla, gira un rotor (de forma parecida a un tacómetro), cambiando la configuración.

Pudiera generar unos 159 trillones combinaciones de encriptación para un texto.



- **Simulador enigma**

<http://enigmaco.de/enigma/enigma.html>

- **Explicación de la maquina enigma**

https://www.youtube.com/watch?v=XK_1gUo8YDE



BREVE HISTORIA DE INTERNET

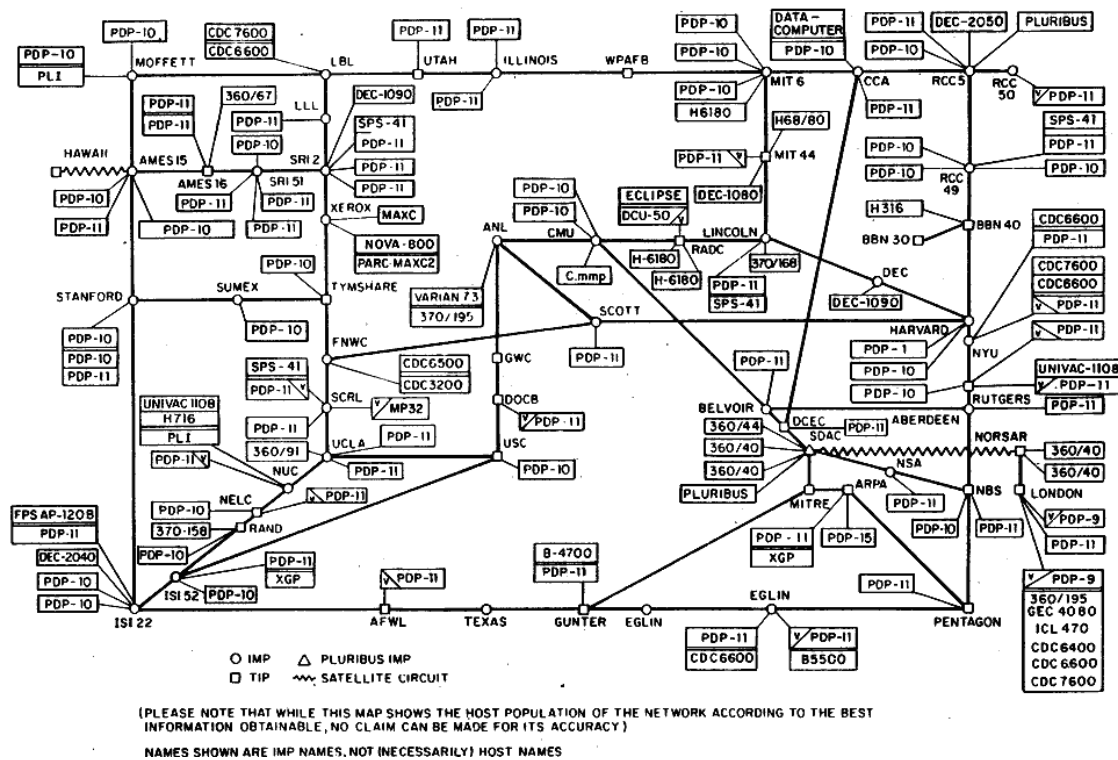
La semilla de lo hoy conocemos por internet, tiene su origen en plena guerra fría (conflicto entre, principalmente estados Unidos y la Unión Soviética, en distintos niveles, social, político, económico, militar o científico, con distintos periodos de intensidad se alargo desde finales de la segunda guerra mundial hasta el año 91).

Algunos hechos cronológicos que se dieron en este periodo son:

- **1962:** Crisis de los misiles.
- **1966:** Planeación de ARPANET (Advanced Research Projects Agency Network, es decir, la Red de la Agencia de Proyectos de Investigación Avanzada), los objetivos son:
 - El uso de una red descentralizada con múltiples caminos entre dos puntos.
 - La división de mensajes completos en fragmentos que seguirían caminos distintos.
- **1969:** ARPANET transporta sus primeros paquetes
- **1977:** Ejemplo de mapa (completo de Internet):



ARPANET LOGICAL MAP, MARCH 1977



- **1981:** Implementación del protocolo TCP/IP
- **1989:** Caída del muro de Berlín.
- **1990:** Desaparece ARPANET y comienza el giro a lo que conocemos actualmente como Internet (comercialmente hablando), en la misma época se considera el fin de la guerra fría.
- **1991:** Fecha que se considera el fin guerra fría.

Es muy posible que el miedo inicial a un ataque nuclear definiera él como debiera ser la arquitectura de una red de comunicaciones descentralizada que siempre estuviera disponible aun incluso cuando se atacaran nodos de la red.

- **1992 (aproximadamente):** Se crea la WWW.
- **1995:** Difusión pública de internet a través de proveedores de internet.



- **1998:** Se crea Google.
- **2001:** Explosión de la burbuja .com
- **2001:** Wikipedia.
- **2004:** Se crea Facebook.
- **2005:** Creación del (concepto) de la web 2.0, el internet enfocado en servicios y comienza la época de la banca electrónica
- **2007:** Se presenta el iPhone y con el comienza la era de los smartphones entre los usuarios comunes (no empresariales).

En la actualidad es común compartir en internet información personal (incluyendo la familiar) en Internet (Facebook, Instagram, Twitter), y usar la banca móvil (o digital). Es un desafío que esta información siga siendo privada y que no se use de forma perjudicial para nosotros.



SEGURIDAD EN INTERNET

El diseño de internet y los protocolos que lo sustentan de origen, no están pensando para garantizar la privacidad.

Los paquetes TCP/IP (el protocolo base de internet), eran recibidos por todos los nodos de una red. Cada equipo decidía si el paquete era para él o lo desechaba. Esto implica que cualquier nodo pudiera analizar los envíos de una red (a esto se llama “modo promiscuo”).

De origen, protocolos como la recepción de emails, ftp, incluso páginas web, no contemplaban seguridad exponiendo todo el flujo de información a cualquier persona que estuviera conectada a la red.

Es un tanto curioso que en algún momento, se separen la necesidad de disponibilidad, de la necesidad de privacidad en la red, incluso más si consideramos el temprano uso de la criptografía de forma militar.

En la actualidad la criptografía nos ayuda en los siguientes aspectos:

- Asegurar la identidad del emisor de la información y del receptor (se identifican mutuamente, mediante un proceso de autenticación).
- Proteger la privacidad de los mensajes que se transfieren por una red (solo el envía la información y el destinatario de esta son capaces de comprenderla apropiadamente).
- Asegurar la integridad de la información, es decir poder asegurar que la información que estamos compartiendo no ha sido manipulada de alguna forma.
- Poder garantizar irrefutablemente el responsable de un mensaje, mediante firmas electrónicas.



PRIVACIDAD EN INTERNET.

En la actualidad es prioritario garantizar nuestra privacidad en internet. El mal uso de nuestra información personal nos pone en situaciones de riesgo económico y de otra índole, como físico.

Mucha gente comparte en Facebook su perfil de forma pública, sin siquiera enterarse, proporcionando información de sus horarios, hábitos e incluso hijos y familiares que no pueden defenderse por sí mismo, aceptamos “amigos” que son prácticamente desconocidos para nosotros, y con los que compartimos información sin tener una plena confianza real en ellos.

En cuanto a nuestras operaciones bancarias, es posible que seamos víctimas de fraude, que nos conectemos a páginas en las que se hagan pasar por nuestras páginas y extraiga información de nuestras cuentas o de nuestra identidad (y hagan operaciones en nuestro nombre), también es posible que nos estemos conectando a la página real pero nuestra información sea interceptada, recolectada y manipulada.

Otro posible uso de nuestra información, es integrándola en un sistema de Big Data. Se recolectan nuestros datos, la forma de usar servicios, e incluso nuestra ubicación. Mediante estos sistemas como Amazon, Netflix, o Facebook, nos da sugerencias basadas no solo en nuestros gustos, sino en los gustos de personas que tengan un perfil parecido al nuestro. Esto es realmente muy práctico, pero en cierta forma, cuando el sistema privilegia cierta información en beneficio de un vendedor (o un candidato político, por ejemplo), es muy posible que nos demos cuentas, y que seamos manipulados para tomar acciones que de otra forma no tomaríamos.

A continuación una serie de ejemplos de uso de nuestros datos de sin nuestra autorización



Caso espionaje NSA

En junio de 2013, se difundió que el FBI y la NSA, recolectaba datos directamente de los servidores de Microsoft, Yahoo, Google, Facebook, PalTalk, AOL, Skype, YouTube y Apple.

Más información:

<https://www.20minutos.es/noticia/1850380/0/caso-snowden/cronologia/espionaje-ee-uu/>

En diciembre del 2013, se reveló que la NSA, pagó 10 millones para que la librería de encriptación de RSA BSAFE (en el 2004).

Caso del teléfono el terrorista de San Bernardino

El tiroteo de San Bernardino ocurrió el 2 de diciembre de 2015, a las 10:59 de la mañana (UTC -8) en el Inland Regional Center en San Bernardino, California, que resultó 14 muertos y 21 heridos al menos.

En Febrero de 2016, una jueza pidió a Apple que ayuda desbloquear el teléfono móvil del terrorista implicado.

Apple se negó alegando que lo que estaba pidiendo era una herramienta que pudiera desbloquear cualquier virtualmente cualquier iPhone,



Caso Cambridge Analítica

Cambridge Analítica era empresa dedica a la segmentación publicitaria, esto es dar mensajes extremadamente personalizados a un sector en particular de la población, a veces influenciando su opinión gracias a la recolección previa de datos del individuo.

En el 2014, Cambridge hizo un test de personalidad (aplicación de Facebook), la cual solicita permiso para acceder a todos los datos del perfil, y la gente inocentemente se los proporcionaba.

Facebook permitía acceder a la información de los usuarios (y de su amigos), para uso académico (por sus políticas), Cambridge Analítica recolecto ilícitamente estos datos, y lo uso para influir en la opinión política de la población.

La mala praxis de Facebook, era que una vez que conoció la fuga de datos, no la notifico, ni la prohibió.

Más información:

<https://www.xataka.com/privacidad/el-escandalo-de-cambridge-analytica-resume-todo-lo-que-esta-terriblemente-mal-con-facebook>



TIPOS DE CRIPTOGRAFÍA

Criptografía simétrica

Tanto el receptor del mensaje encriptado, como la persona que encripta, usan la misma clave para realizar sus operaciones.



Algoritmos que lo usa son por ejemplo AES (Rijndael).

Las **ventajas** de este método son:

- Es la practica tiene una velocidad aceptable
- Es muy seguro

Las **desventajas**:

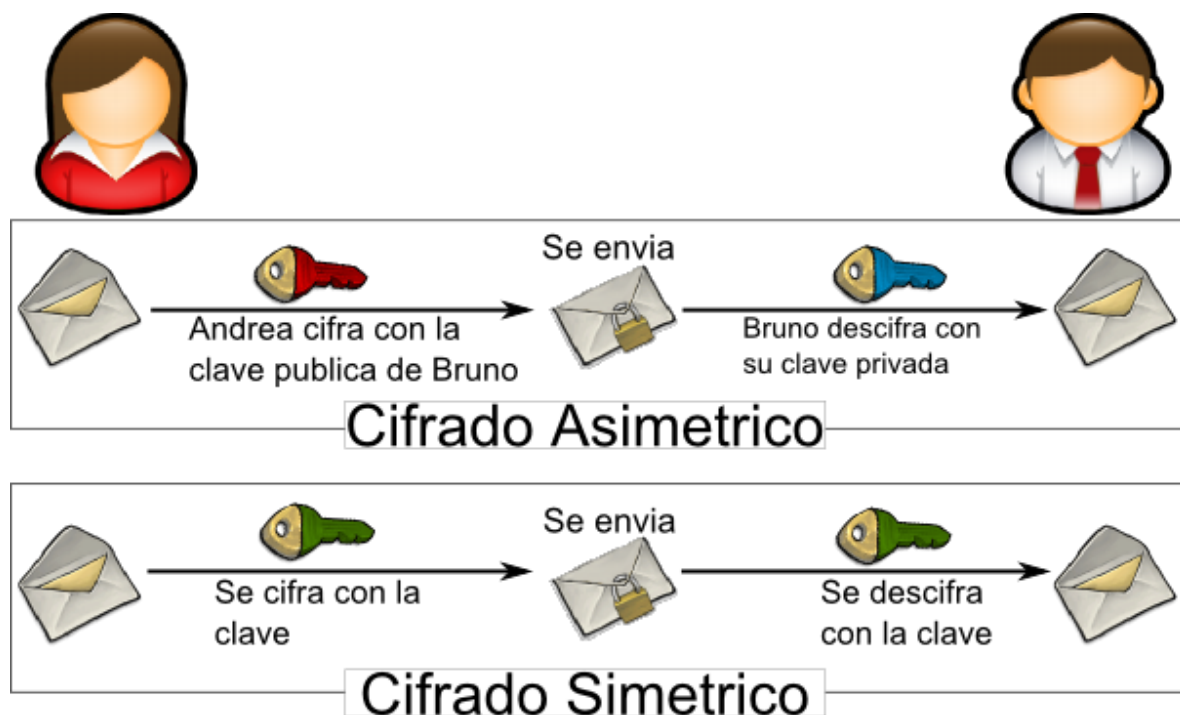
- La principal desventaja es que un usuario genera la clave y tiene que pasársela a otro usuario, lo cual podría comprometer la conexión desde un inicio

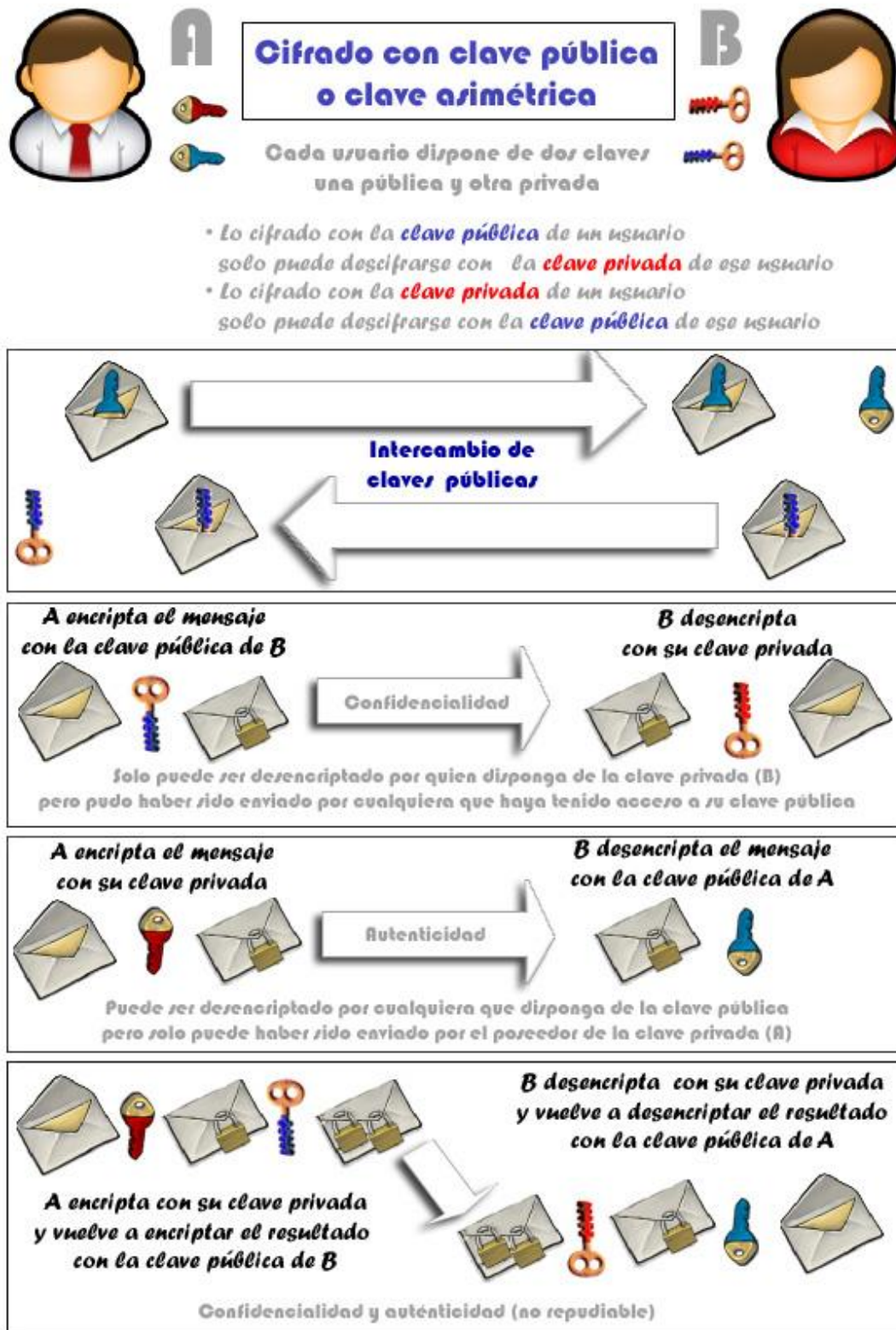


Criptografía asimétrica

Costa de dos claves, en lugar de una, llamadas clave privada y clave pública, la clave pública sirve para encriptar, y es de libre acceso, la clave privada para desencriptar y solo la tiene una persona. Además la privada sirve para firmar mensajes y garantizar que la persona que la usa es el dueño de la clave.

En la práctica funciona así, una persona genera (a través de un algoritmo) una clave privada y pública (que se complementan entre sí), se queda para sí la clave privada, y distribuye la clave pública (a quien desee), a partir de ese momento, quien desee mandarle un mensaje encriptado, puede hacerlo usando la clave pública, y solo él podrá desencriptarlo usando la clave privada. Igualmente si él quiere mandar un comunicado y garantizar su autenticidad, podrá firmarlo con su llave privada, y todo aquel que tenga la llave pública, podrá validar su origen.







Algoritmos que implementan esta tecnología son por ejemplo RSA o DSA.

Las **ventajas** de estos algoritmos son:

- Seguridad en cuanto al custodio de las claves (nunca se difunde la clave privada)
- Seguro en cuanto los mecanismo de inscripción

Las **desventajas**:

- Increíblemente lento.



CERTIFICADOS

Un certificado digital es un documento electrónico, expedido por una entidad competente para tal efecto, que básicamente garantiza la identidad de una persona o un servidor, ligándolo a una clave pública.

Se basa en la confianza que se tenga en la entidad certificadora, si confiamos en la entidad, también confiamos en la identidad de los certificados que expida.

Se realiza un ejemplo completo de creación de certificados, se toma como manual, el instructivo de la siguiente página:

<https://jamielinux.com/docs/openssl-certificate-authority/>

Se incluye una carpeta de ejemplo llama "Certificadora Inicial" y "Certificadora Final", **los certificados aquí incluidos son para efectos de capacitación y nunca deben usarse productivamente.**

Las descripción de todos los certificados a crear se incluyen en **el anexo uno.**

Entidades certificadoras

Las entidades certificadoras son las encargadas de asociar una llave pública, con una identidad, además de garantizar mediante su confiabilidad dicha identidad.

Los certificados de las entidades certificadores están firmados a sí mismos, se llaman certificados autofirmados.



1. Genero las llaves privada

```
openssl genrsa -aes256 -out private/ca.key 4096
```

2. Genero el certificado autofirmado

```
openssl req -config openssl.cnf -key private/ca.key -new -x509 -days 7300 -sha256 -extensions v3_ca -out certs/ca.cer
```

3. Verifico el certificado

```
openssl x509 -noout -text -in certs/ca.cer
```

Entidades intermediarias

Las entidades intermediarias son certificadores en las que la certificadora raíz confía para realizar su trabajo.

1. Creo las llaves

```
openssl genrsa -aes256 -out intermediate/private/intermediate.key 4096
```

2. Creo la petición de generación del certificados

```
openssl req -config intermediate/openssl.cnf -new -sha256 -key intermediate/private/intermediate.key -out intermediate/csr/intermediate.csr
```

3. Firmo el certificado

```
openssl ca -config openssl.cnf -extensions v3_intermediate_ca -days 3650 -notext -md sha256 -in intermediate/csr/intermediate.csr -out intermediate/certs/intermediate.cer
```

4. Verifico el resultado

```
openssl x509 -noout -text -in intermediate/certs/intermediate.cer  
openssl verify -CAfile certs/ca.cer intermediate/certs/intermediate.cer
```



5. Creo la cadena de certificación

```
cat intermediate/certs/intermediate.cer certs/ca.cer > intermediate/certs/ca-chain.cer
```

Certificados personales

Son los que encargan de identificar de una maquina o una persona.

1. Establezco el nombre del host

```
set hostcrt=ejemplohost
```

2. Copio y edito la plantilla

```
copy intermediate\template.cnf intermediate\csr\%hostcrt%.cnf
```

En el archivo csr\ejemplohost.cnf, tengo que agregar una configuración alternativa para el nombre del DSN, con poner el mismo nombre que el host, es suficiente

```
[ alt_names ]
DNS.1 = ejemplohost
```

3. Genero las llaves para el servidor

```
openssl genrsa -aes256 -out intermediate/private/%hostcrt%.key 2048
```

4. Creo la petición del certificado

```
openssl req -config intermediate/csr/%hostcrt%.cnf -key
intermediate/private/%hostcrt%.key -new -sha256 -out intermediate/csr/%hostcrt%.csr
```

5. Firmo el certificado



```
openssl ca -config intermediate/csr/%hostcrt%.cnf -extensions server_cert -days 375  
-notext -md sha256 -in intermediate/csr/%hostcrt%.csr -out  
intermediate/certs/%hostcrt%.cer
```

6. Verifico que se hayan creado adecuadamente

```
openssl x509 -noout -text -in intermediate/certs/%hostcrt%.cer  
openssl verify -CAfile intermediate/certs/ca-chain.cer  
intermediate/certs/%hostcrt%.cer
```

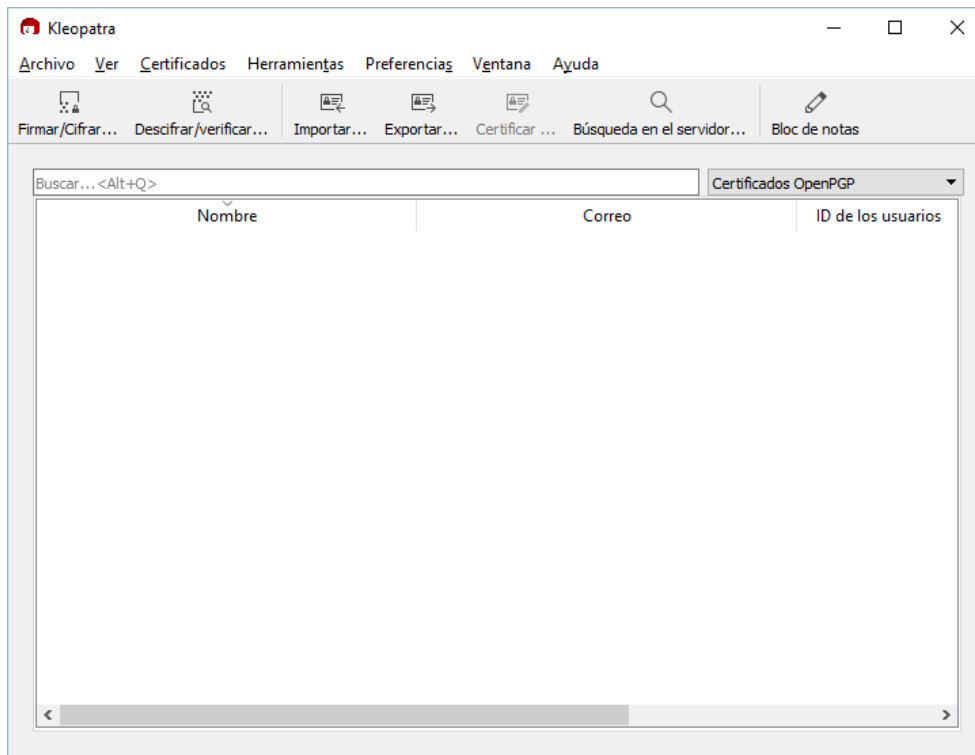
7. Creo el certificado que incluya la llave privada

```
openssl pkcs12 -export -out intermediate/private/%hostcrt%.pfx -inkey  
intermediate/private/%hostcrt%.key -in intermediate\certs\%hostcrt%.cer
```

Configuración de certificados con GnuPG

GnuPG, es una herramienta de software libre para la encriptación y firmado de documentos, tiene varias interfaces graficas, y se integran directamente con algunos gestores de correo electrónico.

Se basa en la combinación de criptográfica asimétrica y simétrica para cifrar y firmar los mensajes.



Puede manejar sus propios certificados o certificados mas estándares como X509

Generar un certificado con el asistente



← Asistente de creación de par de claves

Elegir formato

Por favor, elija que tipo quiere usted crear.

→ Crear un par de claves personales OpenPGP
Los pares de claves OpenPGP están certificados por la confirmación de la huella digital de la clave pública.

→ Crear un par de claves personales X.509 y una solicitud de certificación
Los pares de claves X.509 se certifican por una autoridad de certificación (CA). La petición generada necesita enviarse a la CA para finalizar la creación.

Next Cancel

← Asistente de creación de par de claves

Introduzca detalles

Por favor, introduzca sus detalles personales debajo. Si desea más control sobre los parámetros, pulse el botón «Configuración avanzada».

Nombre de pila (CN): (requerido)

Dirección de correo (EMAIL): (requerido)

Ubicación (L): (opcional)

Unidad organizativa (OU): (opcional)

Organización (O): (requerido)

Código de país (C): (requerido)

CN=Certificado Personal gpg 001,O=Capacitacion Criptografia Certificados Privacidad,C=MX

☐ Añadir dirección de correo al DN (solo para CA defectuosas)

Configuración avanzada...

Next Cancel

Firmado de los solicitudes de certificados con OpenSSL

Los pasos para firmar una solicitud de certificado con OpenSSL son:



1. Establezco el nombre del host

```
set hostcrt=certificadogpg001
```

2. Convierto el p10 a csr

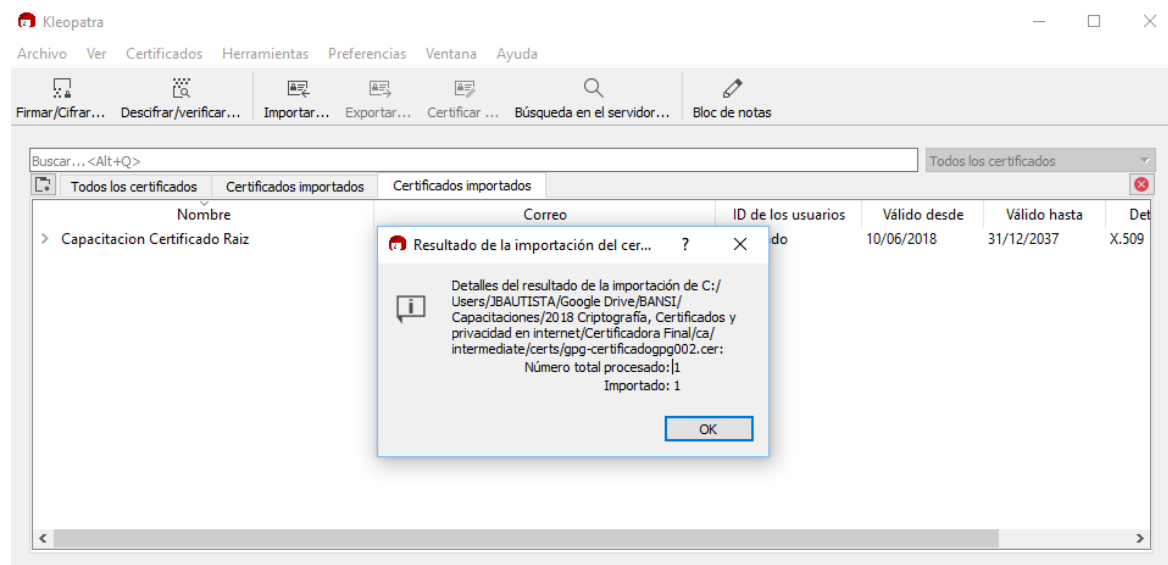
```
openssl req -in intermediate\csr\%hostcrt%.p10 -inform der -out  
intermediate\csr\%hostcrt%.csr
```

3. Firmo el certificado

```
openssl ca -config intermediate/openssl.cnf -extensions usr_cert -days 375 -notext  
-md sha256 -in intermediate/csr/%hostcrt%.csr -out intermediate/certs/gpg-  
%hostcrt%.cer
```

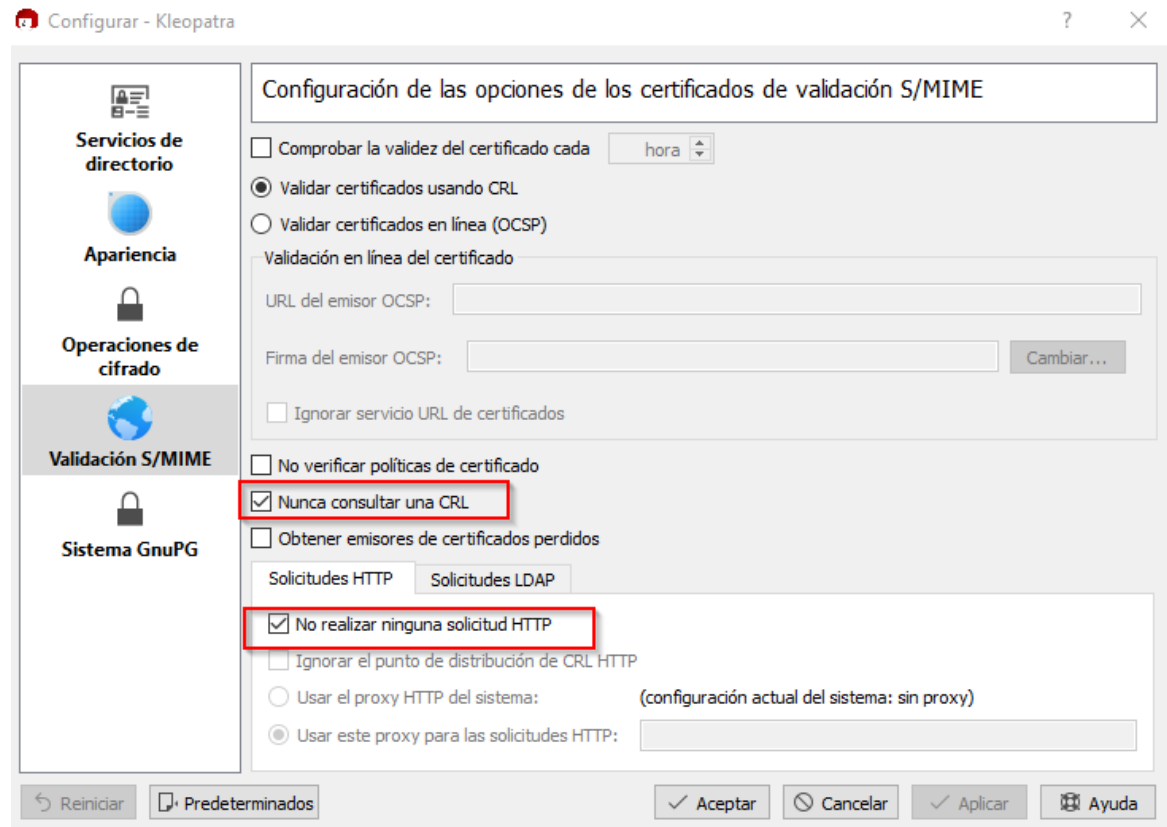
Importación de los certificados

Se importan el certificado en Kleopatra






Exclusivamente para este ejemplo vamos a deshabilitar la lista de revocación de certificados:



Uso de los certificados: Encriptación y firma



 Firmar/cifrar archivos - Kleopatra?✕

Firmar/cifrar archivos

Probar autenticidad (firmar)

☒ Firmar como:

Certificado Personal gpg 001 (certificado, S/MIME, creado: 10/06/2018) ▼

Cifrar

☐ Cifrar para mí:

Certificado Personal gpg 002 (certificado, S/MIME, creado: 10/06/2018) ▼

☒ Cifrar para otros:

Certificado Personal gpg 002 (certificado, X.509, creado: 10/06/2018) ✕


? Por favor, introduzca un nombre o dirección de correo...

☐ Cifrar con contraseña. Cualquiera con el que usted comparta la contraseña podrá ver los datos.

Salida

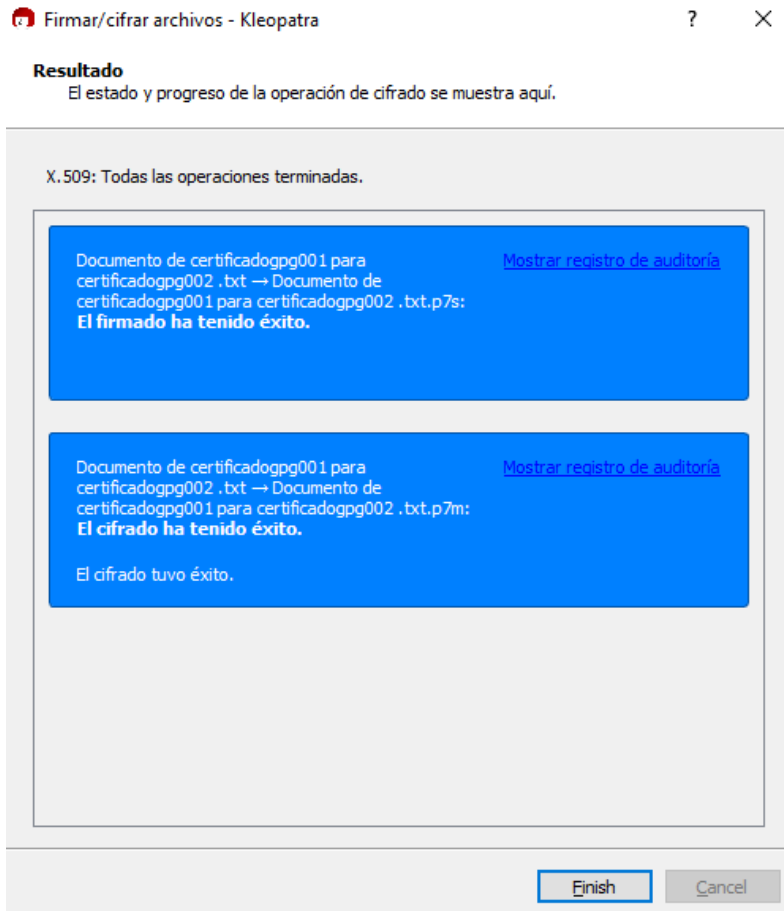
☒ Cifrar / Firmar cada archivo por separado.

2018 Criptografía, Certificados y privacidad en internet/Certificadora Final/ejemplos ✕

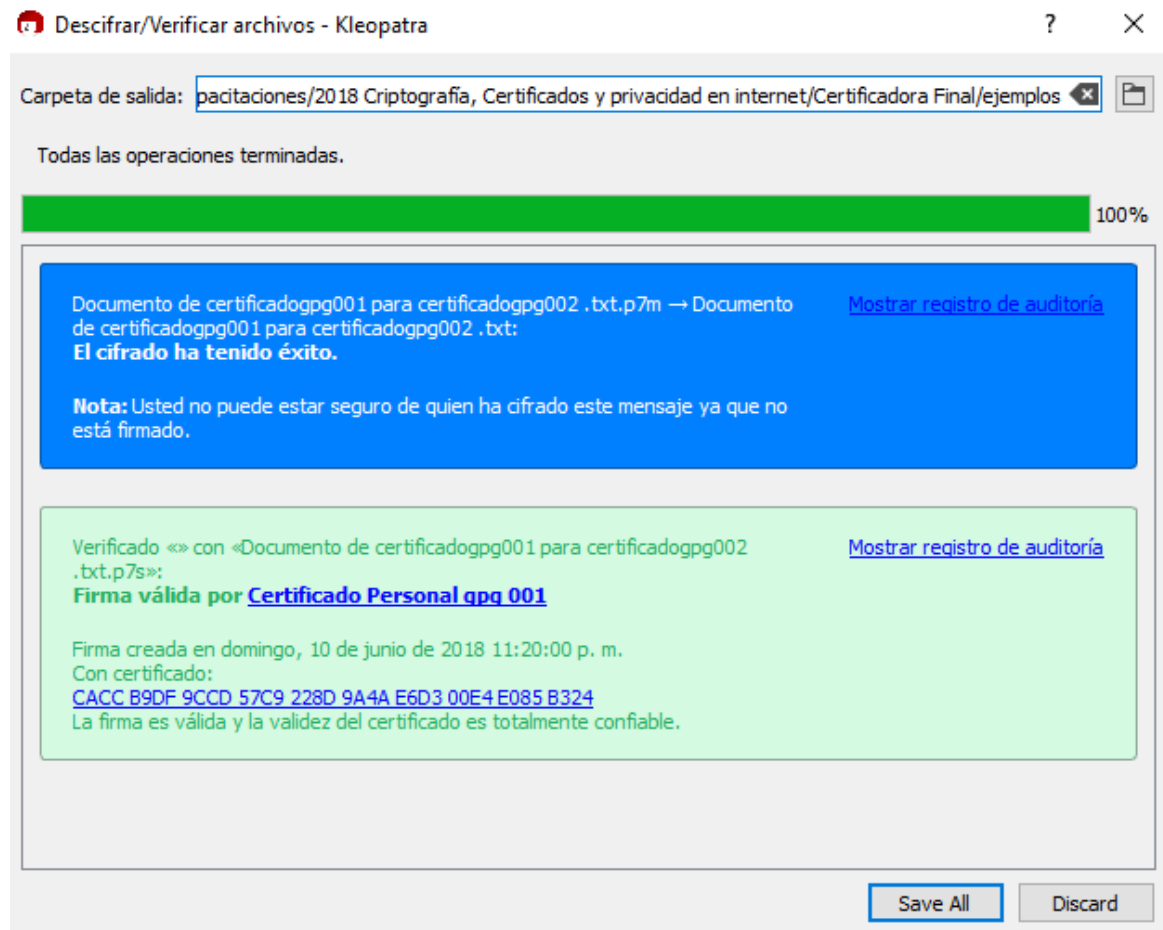


Firmar/cifrar

Cancel



Uso de los certificados: Descripción y validación





ANEXO I: EJEMPLOS DE CERTIFICADOS

Tipo	Nombre Empresarial	Nombre Común	Nombre archivo	Password	email
Certificadora Raiz	Capacitacion Criptografia Certificados Privacidad	Capacitacion Certificado Raiz	ca	ejemplo	
Certificadora Intermediaria	Capacitacion Criptografia Certificados Privacidad	Capacitacion Certificado Intermediario	intermediate	ejemplo	
Certificado Personal Servidor	Capacitacion Criptografia Certificados Privacidad	ejemplohost	ejemplohost	ejemplo	
Certificado Personal GPG	Capacitacion Criptografia Certificados Privacidad	Certificado Personal GPG 001	certificadogpg00 1	ejemplo	certificadogpg001@e jemplo.com
Certificado Personal GPG	Capacitacion Criptografia Certificados Privacidad	Certificado Personal GPG 002	certificadogpg00 2	ejemplo	certificadogpg002@e jemplo.com



ANEXO II: EXTENSIONES DE ARCHIVOS CRIPTOGRAFICOS

Esta información esta extraída de:

<https://serverfault.com/questions/9708/what-is-a-pem-file-and-how-does-it-differ-from-other-openssl-generated-key-file>

- **.csr** this is a Certificate Signing Request. Some applications can generate these for submission to certificate-authorities. The actual format is PKCS10 which is defined in [RFC 2986](#). It includes some/all of the key details of the requested certificate such as subject, organization, state, whatnot, as well as the *public key* of the certificate to get signed. These get signed by the CA and a certificate is returned. The returned certificate is the public *certificate* (which includes the public key but not the private key), which itself can be in a couple of formats.
- **.pem** Defined in RFC's [1421](#) through [1424](#), this is a container format that may include just the public certificate (such as with Apache installs, and CA certificate files `/etc/ssl/certs`), or may include an entire certificate chain including public key, private key, and root certificates. Confusingly, it may also encode a CSR (e.g. as used [here](#)) as the PKCS10 format can be translated into PEM. The name is from [Privacy Enhanced Mail \(PEM\)](#), a failed method for secure email but the container format it used lives on, and is a base64 translation of the x509 ASN.1 keys.
- **.key** This is a PEM formatted file containing just the private-key of a specific certificate and is merely a conventional name and not a standardized one. In Apache installs, this frequently resides in `/etc/ssl/private`. The rights on these files are very important, and some programs will refuse to load these certificates if they are set wrong.
- **.pkcs12 .pfx .p12** Originally defined by RSA in the [Public-Key Cryptography Standards](#) (abbreviated PKCS), the "12" variant was originally enhanced by Microsoft, and later submitted as [RFC 7292](#). This is a passworded container format that contains both public and private certificate pairs. Unlike .pem files, this container is fully encrypted. Openssl can turn this into a .pem file with both public and private keys: `openssl pkcs12 -in file-to-convert.p12 -out converted-file.pem -nodes`

A few other formats that show up from time to time:

- **.der** A way to encode ASN.1 syntax in binary, a .pem file is just a Base64 encoded .der file. OpenSSL can convert these to .pem (`openssl x509 -inform der -in to-convert.der -`



out converted.pem). Windows sees these as Certificate files. By default, Windows will export certificates as .DER formatted files with a different extension. Like...

- **.cert .cer .crt** A .pem (or rarely .der) formatted file with a different extension, one that is recognized by Windows Explorer as a certificate, which .pem is not.
- **.p7b .keystore** Defined in [RFC 2315](#) as PKCS number 7, this is a format used by Windows for certificate interchange. Java understands these natively, and often uses .keystore as an extension instead. Unlike .pem style certificates, this format has a *defined* way to include certification-path certificates.
- **.crl** A certificate revocation list. Certificate Authorities produce these as a way to de-authorize certificates before expiration. You can sometimes download them from CA websites.

In summary, there are four different ways to present certificates and their components:

- **PEM** Governed by RFCs, it's used preferentially by open-source software. It can have a variety of extensions (.pem, .key, .cer, .cert, more)
- **PKCS7** An open standard used by Java and supported by Windows. Does not contain private key material.
- **PKCS12** A Microsoft private standard that was later defined in an RFC that provides enhanced security versus the plain-text PEM format. This can contain private key material. It's used preferentially by Windows systems, and can be freely converted to PEM format through use of openssl.
- **DER** The parent format of PEM. It's useful to think of it as a binary version of the base64-encoded PEM file. Not routinely used by much outside of Windows.