Práctica: confidencialidad con clave pública en OpenSSL

Seguridad de Sistemas Informáticos

Tercera Práctica

Resumen

El objetivo de está práctica es trabajar con esquemas de cifrado asimétrico disponibles en OpenSSL.

1. Confidencialidad en OpenSSL con clave pública

1.1. Generación de claves y cifrado con los comandos *"genpkey"* y *"pkey"* y *"pkeyutl"*

En este apartado nos centraremos en ejemplos basados en el RSA[1] puesto que es el algoritmo de clave pública más extendido actualmente. En OpenSSL hay tres comandos básicos para generar, examinar, manipular y usar las claves de criptosistemas de clave pública: `genpkey, pkeya` y `pkeyutl`.

- `genpkey` se usa para generar una nueva clave privada . La longitud apropiada para una clave RSA deber ser igual o superior a 2048 bits, no se recomienda utilizar claves de longitud menor. Por defecto dicha clave se almacena sin protección, pero el comando permite especificar una opción para cifrarla usando un cifrado de clave secreta.

  ```
  openssl genpkey [-help] [-out filename] [-outform DER|PEM] [-quiet] [-pass arg] [-cipher]
  [-paramfile file] [-algorithm alg] [-pkeyopt opt:value] [-genparam] [-text] [-engine id]
  [-provider name] [-provider-path path] [-propquery propq] [-config configfile]
  ```

  El significado de mucho de los posibles parámetros se puede deducir a partir de su nombre. A continuación se describen algunos de ellos.

  - -out filename: Fichero en el que se almacena la clave privada generada. La salida estándar es usada por defecto.
  - -outform DER—PEM Formato en el que se codifica la clave generada, por defecto PEM.
  - -quiet: No da información durante el proceso de generación de la clave.

[1] En las versiones de OpenSSL anteriores a 3.0, existen otros comandos que permiten usar el criptosistma RSA directamente: genrsa, rsa y rsautl.

- -pass arg: Permite dar una contraseña de acceso al fichero que contiene la clave generada que se solicitará cada vez que se vaya a acceder a él. Se puede facilitar de varias maneras, la más directa pero no la más segura es `pass:password`, siendo ''password" la contraseña seleccionada.
- -cipher: Esta opción permite cifrar la clave privada con el cifrado especificado.
- -algorithm alg: Especifica el algoritmo de clave pública a utilizar. Se puede seleccionar: RSA, RSA-PSS, EC, X25519, X448, ED25519 y ED448. Si se utiliza esta opción debe preceder a cualquier opción especificada con -pkeyopt. Las opciones -paramfile y -algorithm son mutuamente excluyentes. Los motores pueden añadir algoritmos además de los estándar incorporados.
- -pkeyopt opt:value: Establece las opciones del algoritmo de clave pública opt a value. El conjunto preciso de opciones admitidas depende del algoritmo de clave pública utilizado y de su implementación. Para el caso del RSA las opciones son las siguientes:
  - rsa_keygen_bits:numbits: Número de bits de la clave generada, por defecto es 2048.
  - rsa_keygen_primes:numprimes: Número de primos usados en la clave, por defecto son 2.
  - rsa_keygen_pubexp:value: Exponente público usado en RSA, puede ser un número entero de gran tamaño expresadpo en decimal o hexadecimal si se precede con 0x. El valor por defecto es 65537.
- -genparam: Generar un conjunto de parámetros en lugar de una clave privada. Si se utiliza esta opción debe preceder a cualquier opción -algorithm, -paramfile o -pkeyopt.
- -paramfile file: Algunos algoritmos de clave pública generan una clave privada basada en un conjunto de parámetros. Estos pueden ser suministrados utilizando esta opción. Si se utiliza esta opción, el algoritmo de clave pública utilizado viene determinado por los parámetros. Si se utiliza esta opción debe preceder a cualquier opción -pkeyopt. Las opciones -paramfile y -algorithm son mutuamente excluyentes.
- -text: Imprime una representación de texto (sin cifrar) de las claves privadas y públicas y los parámetros junto con la estructura PEM o DER.

  La generación de la clave privada RSA implica esencialmente la generación de dos o más números primos. Cuando se genera una clave privada, aparecen varios símbolos que indican el progreso de la generación. Un `.` representa cada número que ha pasado una prueba de criba inicial, `+` significa que un número ha pasado una sola ronda de la prueba de primalidad de Miller-Rabin, `*` significa que el primo actual comienza un progreso de regeneración debido a algunas pruebas fallidas. Una nueva línea significa que el número ha superado todas las pruebas de primalidad (el número real depende del tamaño de la clave).

- `pkey`: El comando pkey se utiliza para manipular y examinar claves generadas para criptosistemas de clave pública.

  Es capaz agregar, modificar y eliminar el cifrado de protección de una clave privada. También es capaz de producir una clave pública a partir de una clave privada. El comando también se puede utilizar para

mostrar información acerca de una clave pública o privada y para cambiar el formato de odificación de las mismas.

`openssl pkey [-help] [-engine id] [-provider name] [-provider-path path] [-propquery propq] [-check] [-pubcheck] [-in filename|uri] [-inform DER|PEM|P12|ENGINE] [-passin arg] [-pubin] [-out filename] [-outform DER|PEM] [-cipher] [-passout arg] [-traditional] [-pubout] [-noout] [-text] [-text_pub] [-ec_conv_form arg] [-ec_param_enc arg]`

- -check: comprueba la consistencia de un par de claves para las componentes pública y privada.
- -pubcheck: comprueba la corrección de una clave pública o de la componente pública de un par de claves.
- -in filename—uri: Especifica la entrada para leer una clave o la entrada estándar si no se especifica esta opción. Si la entrada de la clave está encriptada y no se indica -passin, se solicitará una frase de paso.
- -inform DER—PEM—P12—ENGINE: El formato de entrada de la clave, no es obligatorio especificarlo;
- -passin arg: La fuente de la contraseña para la entrada de la clave.
- -pubin: Por defecto se lee una clave privada de la entrada. Con esta opción sólo se leen los componentes públicos.
- -out nombre_de_archivo Especifica el nombre de archivo de salida para guardar la salida codificada y/o de texto de la clave o la salida estándar si no se especifica esta opción. Si se establece cualquier opción de cifrado pero no se indica -passout, se solicitará una frase de paso. El nombre del archivo de salida no debe ser el mismo que el de entrada.
- -outform DER—PEM El formato de salida de la clave; el valor predeterminado es PEM.
- -cipher Cifra la clave privada codificada en PEM con el cifrado suministrado **??**. El cifrado no es compatible con la salida DER.
- -traditional Normalmente una clave privada se escribe utilizando el formato estándar: esto es la forma PKCS#8 con el algoritmo de cifrado apropiado (si lo hay). Si se especifica la opción -traditional, se utiliza el formato "tradicional"más antiguo.
- -pubout Por defecto se emiten las claves privada y pública; esta opción restringe la salida a las componentes públicas. Esta opción se establece automáticamente si la entrada es una clave pública. Cuando se combina con -text, equivale a -text_pub.
- -noout No mostrar la clave en forma codificada.
- -text: Muestra los distintos componentes de una clave privada o pública en formato de texto plano.

- `pkeyutl`: implementa operaciones de clave pública de bajo nivel utilizando cualquier algoritmo soportado.

`openssl pkeyutl [-help] [-in file] [-rawin] [-digest algorithm] [-out file] [-sigfile file] [-inkey filename|uri] [-keyform DER|PEM|P12|ENGINE] [-passin arg] [-peerkey file] [-peerform DER|PEM|P12|ENGINE] [-pubin] [-certin] [-rev] [-sign] [-verify] [-verifyrecover] [-encrypt] [-decrypt] [-derive] [-kdf algorithm] [-kdflen length] [-pkeyopt opt:value] [-pkeyopt_passin opt[:passarg]] [-hexdump] [-asn1parse] [-engine id] [-engine_impl] [-rand files] [-writerand file] [-provider name] [-provider-path path] [-propquery propq] [-config configfile]`

- -rawin Indica que los datos de entrada no han sido procesados por ningún algoritmo resumen (hash). Se puede especificar un algoritmo resumen utilizando la opción -digest. Esta opción sólo puede utilizarse con -sign y -verify y debe es obligatoria con los algoritmos Ed25519 y Ed448.
- -digest algorithm Especifica el algoritmo resumen que se utiliza para hacer un hash de los datos de entrada antes de firmarlos o verificarlos. Esta opción puede omitirse si el algoritmo de firma no requiere uno (por ejemplo, EdDSA). Si se omite esta opción pero el algoritmo de firma lo requiere, se utilizará un valor por defecto. Para algoritmos de firma como RSA, DSA y ECDSA, SHA-256 será el algoritmo por defecto. Para SM2, será SM3. Si esta opción está presente, también debe especificarse la opción -rawin.
- -sigfile archivo Archivo de firma, necesario sólo para las operaciones de -verificación
- -inkey nombre de archivo—uri La clave de entrada, por defecto debe ser una clave privada.
- -keyform $DER|PEM|P12|ENGINE$ El formato de la clave
- -passin arg El origen de la contraseña de la clave de entrada.
- -peerkey file El archivo de clave compartida utilizado por las operaciones de derivación (acuerdo) de claves.
- -peerform $DER|PEM|P12|ENGINE$ El formato de la clave compartida.
- -pubin El archivo de entrada es una clave pública. La entrada es un certificado que contiene una clave pública.
- -rev Invierte el orden del búfer de entrada. Esto es útil para algunas bibliotecas (como CryptoAPI) que representan el búfer en formato little endian.
- -sign Firma los datos de entrada (que deben ser un hash) y emite el resultado firmado. Esto requiere una clave privada.
- -verify Verifica los datos de entrada (que deben ser un hash) contra el archivo de firma e indica si la verificación tuvo éxito o falló. Usa la clave pública.
- -verifyrecover Verifica los datos de entrada (que deben ser un hash) e indica los datos recuperados.
- -encrypt Cifrar los datos de entrada utilizando una clave pública.
- -decrypt Descifrar los datos de entrada utilizando una clave privada.
- -derive Genera una clave compartida.

- -Kdf algoritmo Utilizar el algoritmo de la función de derivación de claves. Los algoritmos soportados actualmente son TLS1-PRF y HKDF[2].
- -kdflen longitud Establece la longitud de salida para KDF.
- -hexdump vuelca en hexadecimal los datos de salida.

En las últimas versiones de OpenSSL se da soporte a la gestión de claves para criptosistemas basados en curvas elípticas y se incluye nuevas opciones de cifrados para proteger las claves desechando el des.

2. Firma digital con OpenSSL

Un resumen cifrado con la clave pública del autor/dueño del fichero es equivalente a la firma digital de dicho fichero. Otros usuarios pueden comprobar el resumen del fichero usando la clave pública del autor. Es por ello que veremos en este apartado cómo usar el comando `dgst` para utilidades relacionadas con la firma digital.

- `openssl dgst [hash function] -sign private.key -out file.sign file.input`: cifra el resumen del fichero file.input con la clave privada almacenada en private.key.
- `openssl dgst [hash function] -verify public.key -signature file.sign file.input`: descifra el contenido del fichero file.sign (resumen) con la clave pública almacenada en public.key y compara el resultado con el resumen obtenido del fichero file.input

[2]Ambas están basada en códigos de autenticación de mensajes generados con HMAC