Seguridad y Protección de Sistemas Informáticos

Materia: Prácticas
Módulo: Tecnología de la Información

Grado en Ingeniería Informática
UGR 2019/2020

13 de octubre de 2019

Tabla de Contenidos

Generalidades

Clave: Podemos utilizar las herramientas

`pkeyutl` o bien `rsautl`, correspondiente al viejo RSA. Para conocer las posibilidades de cada una de ellas podemos ejecutar:

- Para `pkeyutl`:

  ```
  openssl pkeyutl --help
  ```

- Para `rsautl`:

  ```
  openssl rsautl --help
  ```

Generalidades

Clave: Podemos utilizar las herramientas

`pkeyutl` o bien `rsautl`, correspondiente al viejo RSA. Para conocer las posibilidades de cada una de ellas podemos ejecutar:

- Para `pkeyutl`:

  ```
  openssl pkeyutl --help
  ```

- Para `rsautl`:

  ```
  openssl rsautl --help
  ```

Generalidades

Clave: Podemos utilizar las herramientas

`pkeyutl` o bien `rsautl`, correspondiente al viejo RSA. Para conocer las posibilidades de cada una de ellas podemos ejecutar:

- Para `pkeyutl`:

  ```
  openssl pkeyutl --help
  ```

- Para `rsautl`:

  ```
  openssl rsautl --help
  ```

Ejemplo de Uso de `pkeyutl`

Clave: Una de las aplicaciones típicas de `pkeyutl`

consiste en la preparación de mensajería cifrada y firmada mediante parejas de claves privada/pública.

- Es útil para transmisión de mensajes breves, como puede ser una clave de cifrado para sistema simétrico.
- Para el cifrado de mensajes pesados se usa `smime`.
- Generación de llaves pública y privada:

```
openssl genpkey -algorithm RSA -pkeyopt
    rsa_keygen_bits:2048 -pkeyopt
    rsa_keygen_pubexp:3 -out privkey-ID.pem
```

- La llave privada, contenida en `privkey-ID.pem` está expresada en base 64.

Ejemplo de Uso de `pkeyutl`

Clave: Una de las aplicaciones típicas de `pkeyutl`

consiste en la preparación de mensajería cifrada y firmada mediante parejas de claves privada/pública.

- Es útil para transmisión de mensajes breves, como puede ser una clave de cifrado para sistema simétrico.
- Para el cifrado de mensajes pesados se usa `smime`.
- Generación de llaves pública y privada:

```
openssl genpkey -algorithm RSA -pkeyopt
        rsa_keygen_bits:2048 -pkeyopt
        rsa_keygen_pubexp:3 -out privkey-ID.pem
```

- La llave privada, contenida en `privkey-ID.pem` está expresada en base 64.

Ejemplo de Uso de `pkeyutl`

Clave: Una de las aplicaciones típicas de `pkeyutl`

consiste en la preparación de mensajería cifrada y firmada mediante parejas de claves privada/pública.

- Es útil para transmisión de mensajes breves, como puede ser una clave de cifrado para sistema simétrico.
- Para el cifrado de mensajes pesados se usa `smime`.
- Generación de llaves pública y privada:

```
openssl genpkey -algorithm RSA -pkeyopt
        rsa_keygen_bits:2048 -pkeyopt
        rsa_keygen_pubexp:3 -out privkey-ID.pem
```

- La llave privada, contenida en `privkey-ID.pem` está expresada en base 64.

Ejemplo de Uso de pkeyutl

Clave: Una de las aplicaciones típicas de pkeyutl

consiste en la preparación de mensajería cifrada y firmada mediante parejas de claves privada/pública.

- Es útil para transmisión de mensajes breves, como puede ser una clave de cifrado para sistema simétrico.
- Para el cifrado de mensajes pesados se usa smime.
- Generación de llaves pública y privada:

```
openssl genpkey -algorithm RSA -pkeyopt
        rsa_keygen_bits:2048 -pkeyopt
        rsa_keygen_pubexp:3 -out privkey-ID.pem
```

- La llave privada, contenida en privkey-ID.pem está expresada en base 64.

Ejemplo de Uso de `pkeyutl`

Clave: Una de las aplicaciones típicas de `pkeyutl`

consiste en la preparación de mensajería cifrada y firmada mediante parejas de claves privada/pública.

- Es útil para transmisión de mensajes breves, como puede ser una clave de cifrado para sistema simétrico.
- Para el cifrado de mensajes pesados se usa `smime`.
- Generación de llaves pública y privada:

```
openssl genpkey -algorithm RSA -pkeyopt
        rsa_keygen_bits:2048 -pkeyopt
        rsa_keygen_pubexp:3 -out privkey-ID.pem
```

- La llave privada, contenida en `privkey-ID.pem` está expresada en base 64.

Ejemplo de Uso de pkeyutl

- Para ver los valores:

```
openssl pkey -in privkey-ID.pem -text
```

- Para obtener la llave pública en un fichero:

```
openssl pkey -in privkey-ID.pem -out
        pubkey-ID.pem -pubout
```

- Para ver los valores individuales:

```
openssl pkey -in pubkey-ID.pem -pubin -text
```

- Nuestro mensaje estaría contenido en el fichero message-ID.txt:

```
$ echo "Deseo vender todas mis acciones de Microsoft
      antes de las elecciones" >> message-ID.txt
$ cat message-ID.txt
```

Ejemplo de Uso de pkeyutl

- Para ver los valores:

```
openssl pkey -in privkey-ID.pem -text
```

- Para obtener la llave pública en un fichero:

```
openssl pkey -in privkey-ID.pem -out
        pubkey-ID.pem -pubout
```

- Para ver los valores individuales:

```
openssl pkey -in pubkey-ID.pem -pubin -text
```

- Nuestro mensaje estaría contenido en el fichero message-ID.txt:

```
$ echo "Deseo vender todas mis acciones de Microsoft
      antes de las elecciones" >> message-ID.txt
$ cat message-ID.txt
```

Ejemplo de Uso de pkeyutl

- Para ver los valores:

```
openssl pkey -in privkey-ID.pem -text
```

- Para obtener la llave pública en un fichero:

```
openssl pkey -in privkey-ID.pem -out
        pubkey-ID.pem -pubout
```

- Para ver los valores individuales:

```
openssl pkey -in pubkey-ID.pem -pubin -text
```

- Nuestro mensaje estaría contenido en el fichero message-ID.txt:

```
$ echo "Deseo vender todas mis acciones de Microsoft
      antes de las elecciones" >> message-ID.txt
$ cat message-ID.txt
```

Ejemplo de Uso de pkeyutl

- Para ver los valores:

```
openssl pkey -in privkey-ID.pem -text
```

- Para obtener la llave pública en un fichero:

```
openssl pkey -in privkey-ID.pem -out
        pubkey-ID.pem -pubout
```

- Para ver los valores individuales:

```
openssl pkey -in pubkey-ID.pem -pubin -text
```

- Nuestro mensaje estaría contenido en el fichero `message-ID.txt`:

```
$ echo "Deseo vender todas mis acciones de Microsoft
      antes de las elecciones" >> message-ID.txt
$ cat message-ID.txt
```

Ejemplo de Uso de pkeyutl

- Para firmar el mensaje necesitamos calcular su huella hash y luego cifrar esa huella por medio de la llave privada.
- Para crear la huella hash de un mensaje (no cifrado):

```
$ openssl dgst -sha1 message-ID.txt
SHA1(message-ID.txt)=
        33b3d182bcbd97d2250cf695bd09090e46ebbad7
```

Ejemplo de Uso de pkeyutl

- Para firmar el mensaje necesitamos calcular su huella hash y luego cifrar esa huella por medio de la llave privada.
- Para crear la huella hash de un mensaje (no cifrado):

```
$ openssl dgst -sha1 message-ID.txt
SHA1(message-ID.txt)=
        33b3d182bcbd97d2250cf695bd09090e46ebbad7
```

Emisión del Mensaje

Clave: Para emitir un mensaje breve firmado y cifrado

debemos conocer la clave pública del receptor, digamos `pubkey-Luis.pem`

- Previamente ha sido creada la pareja de claves pública y privada del receptor `privkey-Luis.pem` y `pubkey-Luis.pem`.
- OpenSSL ofrece la posibilidad de calcular la huella hash del mensaje y luego firmarlo produciendo de `message-ID.txt` el fichero `sign-ID.bin`:

```
$ openssl dgst -sha1 -sign privkey-ID.pem
      -out sign-ID.bin message-ID.txt
$ ls -l
```

Emisión del Mensaje

Clave: Para emitir un mensaje breve firmado y cifrado

debemos conocer la clave pública del receptor, digamos `pubkey-Luis.pem`

- Previamente ha sido creada la pareja de claves pública y privada del receptor `privkey-Luis.pem` y `pubkey-Luis.pem`.
- OpenSSL ofrece la posibilidad de calcular la huella hash del mensaje y luego firmarlo produciendo de `message-ID.txt` el fichero `sign-ID.bin`:

```
$ openssl dgst -sha1 -sign privkey-ID.pem
      -out sign-ID.bin message-ID.txt
$ ls -l
```

Emisión del Mensaje

Clave: Para emitir un mensaje breve firmado y cifrado

debemos conocer la clave pública del receptor, digamos `pubkey-Luis.pem`

- Previamente ha sido creada la pareja de claves pública y privada del receptor `privkey-Luis.pem` y `pubkey-Luis.pem`.
- `OpenSSL` ofrece la posibilidad de calcular la huella hash del mensaje y luego firmarlo produciendo de `message-ID.txt` el fichero `sign-ID.bin`:

```
$ openssl dgst -sha1 -sign privkey-ID.pem
      -out sign-ID.bin message-ID.txt
$ ls -l
```

- Para cifrar el mensaje mediante `RSA`, utilizamos la clave pública del destinatario, digamos `pubkey-Luis.pem`, generando el texto cifrado `ciphertext-ID.bin`:

```
openssl pkeyutl -encrypt -in message.txt
        -pubin -inkey pubkey-Luis.pem
        -out ciphertext-ID.bin
```

- Enviamos a Luis la pareja de ficheros `ciphertext-ID.bin` y `sign-ID.bin`.

- Para cifrar el mensaje mediante `RSA`, utilizamos la clave pública del destinatario, digamos `pubkey-Luis.pem`, generando el texto cifrado `ciphertext-ID.bin`:

```
openssl pkeyutl -encrypt -in message.txt
        -pubin -inkey pubkey-Luis.pem
        -out ciphertext-ID.bin
```

- Enviamos a Luis la pareja de ficheros `ciphertext-ID.bin` y `sign-ID.bin`.

Recepción y Descifrado

Clave: Al recibir la pareja de mensajes

el receptor se dispone a descifrarlos y verificar la autenticidad del envío.

- El primer paso es descifrarlo y leerlo:

```
$ openssl pkeyutl -decrypt -in ciphertext-ID.bin
        -inkey privkey-Luis.pem -out received-ID.txt
$ cat received-ID.txt
```

- La comprobación de la autenticidad de la firma del emisor sería mediante:

```
$ openssl dgst -sha1 -verify pubkey-ID.pem -signature
      sign-ID.bin received-ID.txt
Verified OK
```

Recepción y Descifrado

Clave: Al recibir la pareja de mensajes

el receptor se dispone a descifrarlos y verificar la autenticidad del envío.

- El primer paso es descifrarlo y leerlo:

```
$ openssl pkeyutl -decrypt -in ciphertext-ID.bin
        -inkey privkey-Luis.pem -out received-ID.txt
$ cat received-ID.txt
```

- La comprobación de la autenticidad de la firma del emisor sería mediante:

```
$ openssl dgst -sha1 -verify pubkey-ID.pem -signature
        sign-ID.bin received-ID.txt
Verified OK
```

Recepción y Descifrado

Clave: Al recibir la pareja de mensajes

el receptor se dispone a descifrarlos y verificar la autenticidad del envío.

- El primer paso es descifrarlo y leerlo:

```
$ openssl pkeyutl -decrypt -in ciphertext-ID.bin
        -inkey privkey-Luis.pem -out received-ID.txt
$ cat received-ID.txt
```

- La comprobación de la autenticidad de la firma del emisor sería mediante:

```
$ openssl dgst -sha1 -verify pubkey-ID.pem -signature
        sign-ID.bin received-ID.txt
Verified OK
```