Seguridad y Protección de Sistemas Informáticos

Materia: Prácticas
Módulo: Tecnología de la Información

Grado en Ingeniería Informática

UGR 2019/2020

ugr | Universidad de Granada

13 de octubre de 2019

Tabla de Contenidos

Generalidades

Clave: Para intercambio de mensajería en internet:

podemos usar `smime` de `openssl`:

- `smime` es el acrónimo de *Secure Multipurpose Internet Mail Exchange*
- Es capaz de cifrar, descifrar, firmar y verificar mensajes S/MIME.
- Para conocer sus posibilidades ejecutamos:

```
openssl smime -help
```

Generalidades

Clave: Para intercambio de mensajería en internet:

podemos usar `smime` de `openssl`:

- `smime` es el acrónimo de *Secure Multipurpose Internet Mail Exchange*
- Es capaz de cifrar, descifrar, firmar y verificar mensajes S/MIME.
- Para conocer sus posibilidades ejecutamos:

```
openssl smime -help
```

Generalidades

Clave: Para intercambio de mensajería en internet:

podemos usar `smime` de `openssl`:

- `smime` es el acrónimo de *Secure Multipurpose Internet Mail Exchange*
- Es capaz de cifrar, descifrar, firmar y verificar mensajes S/MIME.
- Para conocer sus posibilidades ejecutamos:

```
openssl smime -help
```

Generalidades

Clave: Para intercambio de mensajería en internet:

podemos usar `smime` de `openssl`:

- `smime` es el acrónimo de *Secure Multipurpose Internet Mail Exchange*
- Es capaz de cifrar, descifrar, firmar y verificar mensajes S/MIME.
- Para conocer sus posibilidades ejecutamos:

```
openssl smime -help
```

Ejemplos

Clave: Los siguientes son unos ejemplos

de cómo usar `smime` de `openssl`:

- Crear un mensaje firmado llano:

```
openssl smime -sign -in message.txt -text \
        -out mail.msg \
        -signer mycert.pem
```

- Crear un mensaje firmado opaco:

```
openssl smime -sign -in message.txt -text \
        -out mail.msg \
        -nodetach -signer mycert.pem
```

Ejemplos

Clave: Los siguientes son unos ejemplos

de cómo usar `smime` de `openssl`:

- Crear un mensaje firmado llano:

```
openssl smime -sign -in message.txt -text \
        -out mail.msg \
        -signer mycert.pem
```

- Crear un mensaje firmado opaco:

```
openssl smime -sign -in message.txt -text \
    -out mail.msg \
    -nodetach -signer mycert.pem
```

Ejemplos

Clave: Los siguientes son unos ejemplos

de cómo usar `smime` de `openssl`:

- Crear un mensaje firmado llano:

```
openssl smime -sign -in message.txt -text \
        -out mail.msg \
        -signer mycert.pem
```

- Crear un mensaje firmado opaco:

```
openssl smime -sign -in message.txt -text \
    -out mail.msg \
    -nodetach -signer mycert.pem
```

Ejemplos

- Crear un mensaje firmado, incluir varios certificados adicionales y leer la clave privada de otro fichero:

```
openssl smime -sign -in in.txt -text -out mail.msg \
    -signer mycert.pem -inkey mykey.pem \
    -certfile mycerts.pem
```

- Crear un mensaje firmado con dos firmantes:

```
openssl smime -sign -in message.txt -text \
    -out mail.msg -signer mycert.pem  \
    -signer othercert.pem
```

Ejemplos

- Crear un mensaje firmado, incluir varios certificados adicionales y leer la clave privada de otro fichero:

```
openssl smime -sign -in in.txt -text -out mail.msg \
    -signer mycert.pem -inkey mykey.pem \
    -certfile mycerts.pem
```

- Crear un mensaje firmado con dos firmantes:

```
openssl smime -sign -in message.txt -text \
    -out mail.msg -signer mycert.pem  \
    -signer othercert.pem
```

Ejemplos

- Enviar un mensaje firmado bajo Unix directamente a sendmail, incluyendo cabeceras:

```
openssl smime -sign -in in.txt -text \
 -signer mycert.pem \
 -from steve@openssl.org -to someone@somewhere \
 -subject "Signed message" \
 | sendmail someone@somewhere
```

- Verificar un mensaje y extraer el certificado del firmante si se tiene éxito:

```
openssl smime -verify -in mail.msg \
      -signer user.pem -out signedtext.txt
```

Ejemplos

- Enviar un mensaje firmado bajo Unix directamente a sendmail, incluyendo cabeceras:

```
openssl smime -sign -in in.txt -text \
 -signer mycert.pem \
 -from steve@openssl.org -to someone@somewhere \
 -subject "Signed message" \
 | sendmail someone@somewhere
```

- Verificar un mensaje y extraer el certificado del firmante si se tiene éxito:

```
openssl smime -verify -in mail.msg \
        -signer user.pem -out signedtext.txt
```

Ejemplos

- Enviar un mensaje cifrado usando AES:

```
openssl smime -encrypt -in in.txt \
  -from steve@openssl.org \
  -to someone@somewhere \
  -subject "Encrypted message" \
  -aes-256-cbc user.pem -out mail.msg
```

Ejemplos

- Firmar un mensaje cifrado:

```
openssl smime -sign -in ml.txt \
  -signer my.pem -text \
  | openssl smime -encrypt -out mail.msg \
     -from steve@openssl.org \
     -to someone@somewhere \
     -subject "Signed and Encrypted message" \
     -aes-256-cbc user.pem
```

Ejemplos

- Descifrar un mensaje:

```
openssl smime -decrypt -in mail.msg \
  -recip mycert.pem -inkey key.pem
```

- Añadir un firmante a un mensaje existente:

```
openssl smime -resign -in mail.msg \
  -signer newsign.pem -out mail2.msg
```

Ejemplos

- Descifrar un mensaje:

```
openssl smime -decrypt -in mail.msg \
  -recip mycert.pem -inkey key.pem
```

- Añadir un firmante a un mensaje existente:

```
openssl smime -resign -in mail.msg \
  -signer newsign.pem -out mail2.msg
```

Resumen

Clave: Como resumen escueto

de la funcionalidad esencial de `smime` podemos firmar y cifrar:

- Necesitaremos un certificado X.509 y la clave privada asociada.
- Podemos firmar el mensaje:

```
openssl smime -sign -in message.txt \
    -out message.sgn \
    -signer /path/to/your/certificate.pem \
    -inkey /path/to/your/secret-key.pem -text
```

Resumen

Clave: Como resumen escueto

de la funcionalidad esencial de `smime` podemos firmar y cifrar:

- Necesitaremos un certificado `X.509` y la clave privada asociada.
- Podemos firmar el mensaje:

```
openssl smime -sign -in message.txt \
  -out message.sgn \
  -signer /path/to/your/certificate.pem \
  -inkey /path/to/your/secret-key.pem -text
```

Resumen

Clave: Como resumen escueto

de la funcionalidad esencial de `smime` podemos firmar y cifrar:

- Necesitaremos un certificado `X.509` y la clave privada asociada.
- Podemos firmar el mensaje:

```
openssl smime -sign -in message.txt \
  -out message.sgn \
  -signer /path/to/your/certificate.pem \
  -inkey /path/to/your/secret-key.pem -text
```

Resumen

- Si además queremos cifrar el correo electrónico, podemos someterlo también a un cifrado con S/MIME:

```
openssl smime -encrypt -in message.sgn \
  -out message.sgn.enc \
  /path/to/intended-operators/certificate.pem
```

- A la llegada del mensaje, si está cifrado lo descifraremos:

```
openssl smime -decrypt -in message.sgn.enc \
  -out message.sgn \
  -recip /path/to/operators/certificate.pem \
  -inkey /path/to/operators/private-key.pem \
  -text
```

Resumen

- Si además queremos cifrar el correo electrónico, podemos someterlo también a un cifrado con S/MIME:

```
openssl smime -encrypt -in message.sgn \
  -out message.sgn.enc \
  /path/to/intended-operators/certificate.pem
```

- A la llegada del mensaje, si está cifrado lo descifraremos:

```
openssl smime -decrypt -in message.sgn.enc \
  -out message.sgn \
  -recip /path/to/operators/certificate.pem \
  -inkey /path/to/operators/private-key.pem \
  -text
```

Resumen

- Seguidamente la firma es validada y el mensaje leído:

```
openssl smime -verify -text -in message.sgn \
  -noverify -out message.txt
```

- El receptor comprueba si el firmante es ciertamente el remitente:

```
openssl smime -pk7out -in message.sgn \
  | openssl pkcs7 -print_certs -noout
```

Resumen

- Seguidamente la firma es validada y el mensaje leído:

```
openssl smime -verify -text -in message.sgn \
  -noverify -out message.txt
```

- El receptor comprueba si el firmante es ciertamente el remitente:

```
openssl smime -pk7out -in message.sgn \
  | openssl pkcs7 -print_certs -noout
```

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

Clave: En un ejemplo real

Eve desea enviar un mensaje firmado y cifrado a Bob. Para ello, ambos necesitan:

- La clave privada de Eve: `eve.key.pem`
- El correspondiente certificado de Eve: `eve.cert.pem`
- La clave privada de Bob: `bob.key.pem`
- El correspondiente certificado de Bob: `bob.cert.pem`
- La clave privada de cada uno es secreta y el certificado, público.

Ejemplo real

- Eve escribirá su mensaje en el fichero `message.txt`.
- Seguidamente firma `message.txt` para crear `message.txt.sgn`:

```
openssl smime -sign -in message.txt \
  -out message.txt.sgn \
  -signer eve.cert.pem \
  -inkey eve.key.pem -text
```

Ejemplo real

- Eve escribirá su mensaje en el fichero `message.txt`.
- Seguidamente firma `message.txt` para crear `message.txt.sgn`:

```
openssl smime -sign -in message.txt \
  -out message.txt.sgn \
  -signer eve.cert.pem \
  -inkey eve.key.pem -text
```

Ejemplo real

- Después cifrará `message.txt.sgn` para crear `message.txt.sgn.enc`:

```
openssl smime -encrypt -in message.txt.sgn \
  -out message.txt.sgn.enc \
  bob.cert.pem
```

- Eve enviará por cualquier medio el mensaje `message.txt.sgn.enc` a Bob, quien efectivamente lo recibe.

Ejemplo real

- Después cifrará `message.txt.sgn` para crear `message.txt.sgn.enc`:

```
openssl smime -encrypt -in message.txt.sgn \
  -out message.txt.sgn.enc \
  bob.cert.pem
```

- Eve enviará por cualquier medio el mensaje `message.txt.sgn.enc` a Bob, quien efectivamente lo recibe.

Ejemplo real

- Con `message.txt.sgn.enc` en su mano, Bob procede a descifrarlo:

```
openssl smime -decrypt -in message.txt.sgn.enc \
  -out message.txt.sgn \
  -recip bob.cert.pem \
  -inkey bob.key.pem
```

- Bob verificará el mensaje:

```
openssl smime -verify -text -in message.txt.sgn \
  -noverify -out message.txt
```

- Bob obtendrá la respuesta `Verification successful`

Ejemplo real

- Con `message.txt.sgn.enc` en su mano, Bob procede a descifrarlo:

```
openssl smime -decrypt -in message.txt.sgn.enc \
  -out message.txt.sgn \
  -recip bob.cert.pem \
  -inkey bob.key.pem
```

- Bob verificará el mensaje:

```
openssl smime -verify -text -in message.txt.sgn \
  -noverify -out message.txt
```

- Bob obtendrá la respuesta `Verification successful`

Ejemplo real

- Con `message.txt.sgn.enc` en su mano, Bob procede a descifrarlo:

```
openssl smime -decrypt -in message.txt.sgn.enc \
  -out message.txt.sgn \
  -recip bob.cert.pem \
  -inkey bob.key.pem
```

- Bob verificará el mensaje:

```
openssl smime -verify -text -in message.txt.sgn \
  -noverify -out message.txt
```

- Bob obtendrá la respuesta `Verification successful`

Ejemplo real

- Finalmente Bob comprueba que el firmante es ciertamente Eve, el remitente del correo electrónico en el que venía el mensaje cifrado:

```
openssl smime -pk7out -in message.txt.sgn | \
openssl pkcs7 -print_certs -noout
```

- obteniendo el siguiente diálogo:

```
subject=/CN=eve@example.com
issuer=/C=GB/ST=England/O=Alice
        Ltd/CN=emisor_common_name
```

Ejemplo real

- Finalmente Bob comprueba que el firmante es ciertamente Eve, el remitente del correo electrónico en el que venía el mensaje cifrado:

```
openssl smime -pk7out -in message.txt.sgn | \
openssl pkcs7 -print_certs -noout
```

- obteniendo el siguiente diálogo:

```
subject=/CN=eve@example.com
issuer=/C=GB/ST=England/O=Alice
        Ltd/CN=emisor_common_name
```