Práctica 1

Criptosistemas simétricos

Introducción

Vamos a introducir el uso de OpenSSL, un proyecto de código abierto que proporciona un conjunto de herramientas robustas, completas y de nivel comercial para los protocolos TLS y SSL. Incluye una biblioteca criptográfica de propósito general. La mejor fuente de información sobre OpenSSL podéis encontrarla en `https://www.openssl.org/`
Vamos a utilizarlo en la línea de comandos, aunque su mayor potencia se obtiene al llamarlo desde otros protocolos o software. Los ejemplos que voy a poner se han realizado con plataformas UNIX® y similares. También hay disponibles implementaciones para Microsoft Windows.
Para saber qué versión tenemos podemos escribir

```
$> openssl version
```

Podemos invocar ayuda de dos maneras, mediante su página manual

```
$> man openssl
```

o escribiendo cualquier comando incorrecto, lo que nos mostrará la lista de comandos

```
$> openssl ayuda
```

En esta práctica nos centraremos en los criptosistemas simétricos que implementa, todos ellos de bloque. No todas las implementaciones de openssl contienen los mismos criptosistemas. El comando enc permite acceder a las funciones de cifrado simétrico. Por ejemplo, una opción incorrecta nos permitirá saber qué criptosistemas contiene nuestra implementación,

```
$> openssl enc ayuda
```

aunque todas suelen incluir `AES`.
Las claves y vectores de inicialización pueden introducirse directamente mediante las opciones `-K` y `-iv` o mediante una contraseña, que puede introducirse con la opción `-pass`. Cuando se emplea una contraseña, la clave y el vector de inicialización se obtienen a partir de la contraseña mediante funciones hash que serán explicadas en temas posteriores.
Para evitar ataques de diccionario eficientes, `OpenSSL` incluye por defecto la opción `-salt`, que agrega una cadena aleatoria antes de generar la clave y el vector de inicialización. Si la clave es introducida directamente, no es necesaria esta opción. En cualquier caso puede eliminarse con la opción `-nosalt`.
Otra acción necesaria que requiere el uso de cifrados por bloque es el padding. Esta técnica completa, según un criterio preestablecido, el tamaño del archivo para que coincida

con un múltiplo del tamaño del bloque. Las opción -nopad elimina el padding, pero sólo debe usarse si estamos seguros de que nuestro archivo tiene un tamaño múltiplo exacto del tamaño del bloque.
Para esta práctica necesitaréis un editor o visor hexadecimal. En UNIX® una opción es xxd.

Tareas a realizar

1. Partiremos de un archivo binario de 1024 bits, todos ellos con valor 0. Para hacer referencia al mismo voy a suponer que se llama `input.bin`, pero podéis dar el nombre que os convenga.
2. Creamos otro archivo binario del mismo tamaño, que contenga un único bit con valor 1 entre los bits 130 y 150, y todos los demás con valor 0. Me referiré a este archivo como `input1.bin`
3. (0,75) Cifrad `input.bin` e `input1.bin` con `AES-256` en modos `ECB`, `CBC` y `OFB` usando una clave (no una contraseña) a elegir del tamaño adecuado, y con vector de inicialización `0123456789abcdef`, cuando sea necesario. Explicad los diferentes resultados.
4. (0,75) Cifrad `input.bin` e `input1.bin` con `AES-128` en modos `ECB`, `CBC` y `OFB` usando una contraseña a elegir. Explicad los diferentes resultados.
5. (0,75) Repetid el punto anterior con la opción `-nosalt`.

6. (0,75) Cifrad `input.bin` con `AES-192` en modo `OFB`, clave y vector de inicialización a elegir (no contraseña). Supongamos que la salida es `output.bin`.
7. (0,75) Descifrad `output.bin` utilizando la misma clave y vector de inicialización que en 6.
8. (0,75) Vuelve a cifrar `output.bin` con `AES-192` en modo `OFB`, clave y vector de inicialización del punto 6. Compara el resultado obtenido con el punto 7, explicando el resultado.
9. (2,25) Repite los puntos 6 al 8 pero empleando contraseña en lugar de clave y vector de inicialización.
10. (1) Presentad la descripción de otro algoritmo de cifrado simétrico que aparezca en vuestra implementación de OpenSSL.
11. (2,25) Repetid los puntos 3 a 5 con el cifrado presentado en el punto 10.

NOTA: En cada punto se indica el valor en puntos del mismo.
La entrega consistirá en un archivo en formato PDF que contenga, además de las explicaciones requeridas, los comandos empleados y capturas de los archivos generados. La entrega se subirá a la plataforma.