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` y `DES`.
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` (en versiones previas se empleaban las opciones `-k` y `-kfile` para este último fin). 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.
Mirad también las opcion -nopad para ver su incidencia en las salidas.
Para esta práctica necesitaréis un editor hexadecimal.

---

***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 dentro de los primeros 40 bits y todos los demás con valor 0. Me referiré a este archivo como `input1.bin`
3. Cifrad `input.bin` con DES en modos ECB, CBC y OFB usando como claves una débil y otra semidébil, con vector de inicialización a vuestra elección, y explicad los diferentes resultados.
4. Cifrad `input.bin` e `input1.bin` con DES en modo ECB y clave a elegir, pero no débil ni semidébil. Explicad la forma de los resultados obtenidos.
5. Cifrad `input.bin` e `input1.bin` con DES en modo CBC, clave y vector de inicialización a elegir. Comparad con los resultados obtenidos en el apartado anterior.
6. Repetid los puntos 4 a 5 con AES-128 y AES-256.
7. Cifrad `input.bin` con AES-192 en modo OFB, clave y vector de inicialización a elegir. Supongamos que la salida es `output.bin`.
8. Descifra `output.bin` utilizando la misma clave y vector de inicialización que en 7.
9. Vuelve a cifrar `output.bin` con AES-192 en modo OFB, clave y vector de inicialización del punto 7.

Compara el resultado obtenido con el punto 8, explicando el resultado.

10. Presentad la descripción de otro cifrado simétrico que aparezca en vuestra implementación de OpenSSL.

11. Repetid los puntos 3 a 5 con el cifrado presentado en el punto 10 (el 3 si el cifrado elegido tuviese claves débiles o semidébiles).

NOTA: Los puntos anteriores no representan distintos items a evaluar. Son un guión de las tareas a realizar. Se puntúan por igual los 15 items que hay que entregar.
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.