Proyecto(s) de seguridad informática,

SoSe 2023

Proyecto 1

(14,99 puntos)

En lo que sigue, tú ≃ tu grupo. Cada uno de vosotros va a representar los papeles de Harry/MJ y Bully. Como Harry/MJ, vais a poner en práctica un cifrado que se parece a un one-time-pad (OTP). Como Bully, intentarás recuperar o corromper mensajes secretos con un ataque de sólo texto cifrado (CT).

(i) Escribirás un programa para generar una lista de claves de 1024 bits (128 Bytes) que mantendrás **en secreto**. Cada clave se generará de la siguiente manera: primero calcule una clave (aleatoria) de 16 bits (2 Bytes) k_0 . Sea k_1 el desplazamiento cíclico de 1 bit de k_0 a la derecha. Sean h_0 y h_1 los hashes SHA-512 de k_0 y k_1 , respectivamente. La concatenación $h_0 \,||\, h_1$ es su clave.1 Puede utilizar cualquier biblioteca criptográfica de su elección (por ejemplo, OpenSSL).

Redacta 10 mensajes de texto sin formato (PT) que Harry y MJ podrían querer comunicarse y guárdalos en los archivos Plaintext-i.txt (para $i = 1 \ldots 10$). Cada mensaje debe tener un máximo de 128 caracteres. Sólo se permiten caracteres en inglés (se admiten signos de puntuación).

Ahora aumenta tu programa para llevar a cabo el cifrado y descifrado basado en OTP. En el modo de encriptación, tu programa debería tomar como entrada un PT (≤ 128 caracteres) y una clave de 1024 bits, y calcular el CT. Lo mismo ocurre con el descifrado.

Puede convertir cada mensaje/clave a binario o hexadecimal utilizando ASCII/UTF para facilitar la operación XOR.

- Cifrar siete mensajes PT (digamos $m_1 , \ldots , m_7$) utilizando OTP, cada uno con una clave única (digamos $k_1 , \ldots k_7$).
- Elige tres mensajes cualesquiera del conjunto anterior $\{m_1 , \ldots m_7\}$ y vuelva a cifrarlos, cada vez con una nueva clave única (k_8 , k_9 , k_{10}).
- Para los tres mensajes restantes m_8 , m_9 , m_{10} encripta reutilizando una clave del conjunto $\{k_1 , \ldots k_7\}$ para cada mensaje.

Barajar los CT. Recodifica el cifrado binario en un formato legible. Almacena los resultados en 13 archivos, nombrándolos como Ciphertext-i.txt (para $i = 1 \ldots 13$). Reúne el código fuente, las claves, los textos planos y los textos cifrados en un archivo $^{\text{tar/zip2}}$. Como comprobación de cordura, ejecuta el descifrado en tu opus de TC y asegúrate de que recuperas los TP.

(ii) Se te dará el zip de un grupo aleatorio que contiene CTs y ahora tienes que jugar a Bully. Intenta desencriptar el conjunto de CTs con tu propio programa, de cualquier manera que puedas. El método de fuerza bruta pasaría por cada una de las ^{216} claves. ¿Puedes hacerlo mejor que la fuerza bruta? Recoge tus resultados en Decrypt-Ciphertext-i.txt (...).

(iii) Como Bully, quieres poner tierra de por medio en los mensajes. Con suerte, habrás recuperado algunos de los PTs y/o claves. En cualquier caso, manipula algunos de los TC de forma significativa y recoge tus resultados como Ciphertext-i-dirty.txt (...). Más tarde, el grupo de cifrado descifrará tus modificaciones.

Debes respetar los formatos de entrada/salida.

[1] SHA-512 produce una salida de 512 bits. Concatenando dos de estas salidas, se obtiene una clave de 1024 bits. Por supuesto, estas no son claves ideales para OTP (¿piensas por qué?) pero serán suficientes para nosotros. Nadie puede forzar 2^{1024} claves en la parte (ii).

[2] Las instrucciones sobre el uso de lenguajes de programación, los detalles de presentación, etc. vienen más adelante.