Criptografía y seguridad
Tarea 3

Andrea Itzel González Vargas
Carlos Gerardo Acosta Hernández

Entrega: 27/03/17
Facultad de Ciencias UNAM

1. Con tus propias palabras explica por qué el siguiente esquema de cifrado para mensajes de tamaño $\ell(n)$ es seguro:
Sea G un generador seudoaleatorio con factor de expansión $\ell(n)$. El algoritmo Gen, con entrada n, devuelve una clave k escogida uniformemente del conjunto $\{0,1\}^n$. El algoritmo de cifrado Enc recibe como entradas una clave k de tamaño n y un mensaje m de tamaño $\ell(n)$ y devuelve el mensaje cifrado $c = G(k) \oplus m$. Para descifrar se calcula $m = G(k) \oplus c$.

R: Como k es escogida aleatoriamente, $G(k)$ da entonces una salida aleatoria tomando a k como semilla. Como $G(k)$ sólo da una salida, es decir sólo tiene que hacer una iteración a partir de la semilla ya que nuestro mensaje es de tamaño $\ell(n)$ y el factor de expansión de G es también $\ell(n)$, no podemos nunca inferir qué salida nos va a dar G si se desconoce a k, entonces mientras k sea siempre renovada uniformemente al cifrar un nuevo mensaje, tendremos un esquema que simula al cifrado de Vernam, y por lo tanto es seguro.

2. Supongamos que Alicia y Bartolo se quieren mandar mensajes cifrados y solo tienen una clave k de 128 bits que ambos conocen. Para mandar un mensaje m hacen lo siguiente:

- Se elige una cadena aleatoria s de 80 bits.
- Se obtiene el mensaje cifrado $c = RC4(s \parallel k) \oplus m$.
- Se manda la pareja (s, c).

a) Si Alicia manda un mensaje (s, c), ¿qué tiene que hacer Bartolo para recuperar el mensaje claro?

R: Para recuperar el mensaje debe calcular $m = RC4(s \parallel k) \oplus c$.

b) Si un adversario puede ver una lista de mensajes (s_1, c_1), (s_2, c_2), . . . que fueron enviados entre Alicia y Bartolo, ¿cómo puede comprobar que se usó el mismo flujo de claves para cifrar

dos mensajes? (Flujo de claves es la salida de RC4.)

R: Ya que Alicia y Bartolo siempre usan la misma clave k, basta con que el adversario encuentre dos mensajes (s_i, c_i) y (s_j, c_j) tales que s_i sea igual a s_j, de tal suerte que:

$$RC4(s_i \parallel k) = RC4(s_j \parallel k).$$

c) Usando la paradoja del cumpleaños, ¿aproximadamente cuántos mensajes tendría que enviar Alicia para que se repita un flujo de claves?

R: Ya que la cadena aleatoria s es la única que cambia y mide 80 bits, cuando se hayan mandado aproximadamente $\sqrt{2^{80}} = 2^{40}$ mensajes habrá una probabilidad del 50 % de que se repita un flujo de claves, para tener una probabilidad de 90 % se necesita aproximadamente de 2^{41} mensajes.

3. Explica qué es una función seudoaleatoria y qué relación tiene con los algoritmos de cifrado por bloques.

R: Primero vemos que una función aleatoria f se define como aquella que mapea uniformemente una entrada $\{0,1\}^n$ a una salida $\{0,1\}^n$, por lo tanto f mapea 2^n posibles entradas a 2^n posibles resultados y como cada uno de estos es de longitud n, si concatenamos todas las salidas de f resultará una cadena de longitud $n \cdot 2^n$. La familia de funciones aleatorias llamada Func_n está compuesta por funciones f y tiene cardinalidad finita de $2^{n \cdot 2^n}$, es decir todas las posibles combinaciones de funciones f vistas con la representación ya mencionada. Seleccionar una $f \in \mathsf{Func}_n$ uniformemente y mapear una entrada $x \in \{0,1\}^n$ usando f tal que $f(x) = y$, debe ser equivalente a seleccionar y uniformemente en $\{0,1\}^n$.

Una función pseudoaleatoria es de la forma $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. La primer entrada es una llave k y la segunda es la entrada x a mapear. Podemos notar que $|k| = |x| = |F(k,x)|$. Cada k define en F una función F_k que tal como f mapea uniformemente una entrada $\{0,1\}^n$ a una salida $\{0,1\}^n$ y $F_k(x) = F(k,x)$. Cabe notar que sólo hay 2^n F_k posibles en F (el número de llaves posibles de longitud n), a diferencia de las $n \cdot 2^n$ f posibles en Func_n, a pesar de esto, para que F sea una función pseudoaleatoria debe de suceder que el seleccionar una F_k para una k escogida uniformemente, debe ser indistinguible de seleccionar uniformemente una $f \in \mathsf{Func}_n$.

Una permutación pseudoaleatoria es una función pseudoaleatoria tal que para toda $x, y \in \{0,1\}^n$ y dada una llave k, sucede que si $F_k(x) = F_k(y)$, entonces $x = y$, o sea que toda F_k es biyectiva. A demás debe de suceder que toda F_k sea invertible, esto es que dada $F_k(x) = y$, debe de existir $F_k^{-1}(y) = x$. Un algoritmo de cifrado por bloques es entonces una permutación pseudoaleatoria.

4. En el cifrado por bloques los errores se propagan de manera distinta dependiendo del modo de operación que se utilice. Explica qué ocurre en el mensaje descifrado cuando se usan los modos ECB, CBC, OFB y CTR, si el texto cifrado contiene un error de un solo bit.

- ECB: Como en este modo cada bloque se cifra y descifra con la misma llave, cada uno independientemente del otro, sólo el bloque que contiene al bit con el error difiere del bloque en el mensaje original. Los otros bloques no se ven afectados.
- CBC: En este modo el cifrado y descifrado de cada bloque depende del anterior, por lo que el bloque que contiene el bit con el error y todos los bloques que le siguen serán distintos al mensaje original una vez descifrados.
- OFB: Como en este modo ningún bit del mensaje depende del otro ya que sólo se hace un xor con una cadena para cifrar y descifrar, el mensaje original y el descifrado sólo difieren en el bit con el error.
- CTR: Análogamente al caso anterior, el mensaje original y el cifrado, difieren sólo en el bit con error.

5. Para usar un cifrado por bloques de tamaño n es necesario que el mensaje tenga longitud múltiplo de n, así que se agrega un padding cuando hace falta. Si se establece que se usará alguna forma de padding (por ejemplo, agregar un uno seguido de ceros), ¿por qué es necesario agregar padding aun en los mensajes de tamaño múltiplo de n?

R: Es necesario porque a partir del criptograma no se puede distinguir si el mensaje original tenía longitud múltiplo de n. Si el *padding* no se incluyera siempre, no sería posible decidir cuándo éste debe ser removido después de descifrar y cuando no. Es por ello que lo más conveniente es siempre poner el padding antes de cifrar y tener confianza de quitarlo después de descifrar sin corromper la información.