Laboratorio:
Criptografía y seguridad
Práctica 3

Andrea Itzel González Vargas
Carlos Gerardo Acosta Hernández

Facultad de Ciencias UNAM

Preguntas

1. Describe los tipos de ataques pasivos y activos y las soluciones que aplicarías para evitar cada tipo de ataque.

- **Ataque pasivo**
  Un ataque pasivo en un sistema criptográfico es aquel en el que el criptoanalista o atacante no interviene con los mecanismos involucrados en el sistema, es decir, no modifica ni altera la información que se busca proteger. Este tipo de ataque tiene por objetivo perpetrar en el sistema utilizando únicamente la observación y monitoreo de los datos transmitidos por el canal de comunicación.

- **Ataque activo**
  Son aquellos que afectan la operación de un sistema criptográfico mediante la modificación del flujo de datos transmitido o la suplantación de éste por un flujo falso.

Para evitarlos, me iría al cerro de Daiana.

2. ¿Cuál es la diferencia más importante entre un algoritmo de cifrado y un algoritmo de autenticación?

Que un algoritmo de autenticación, por definición, no conserva la información completa de la entrada en la salida que devuelve. Es decir, que a partir de un elemento de la imagen de la función *hash*, no es posible recuperar el elemento del dominio del que proviene contando únicamente con la salida; mientras que en un algoritmo de cifrado, debe ser posible recuperar el texto claro a partir de su criptotexto correspondiente.

3. ¿Cuáles son las diferencias entre las funciones de hash y las funciones de hash seguras?

Las funciones de hash en general buscan minimizar el número de colisiones posibles, sin embargo las funciones de hash seguras tienen requerimientos más estrictos que las otras funciones menos seguras no necesariamente cumplen:
Sea $H(m)$ una función de hash segura.

- Debe de ser muy difícil [1] encontrar cualesquiera dos mensajes m, m' tales que $H(m) = H(m')$.
- Dado un hash h, debe de ser muy difícil encontrar cualquier mensaje m tal que $h = H(m)$.
- Dado un mensaje m debe de ser muy difícil encontrar otro mensaje m' tal que $H(m) = H(m')$.

4. Supongamos que H(m) es una función hash resistente a colisiones que mapea un mensaje de una longitud de bits arbitraria en un valor hash de n bits. ¿Es cierto que para todos los mensajes x, x' con x $\neq$ de x', tenemos ¿H(x) $\neq$ H(x')? Argumenta tu respuesta.

Sea M el conjunto de mensajes y H el conjunto de todos los hashes posibles con $H(m)$ tal que $m \in M$, supongamos que la longitud l_m de los mensajes m es a lo más n, i.e. $0 \leq l_m \leq n$. Entonces sabemos que $|H| = 2^n$ y $|M| = \sum_{i=0}^{n} 2^i$, por lo tanto $|M| > |H|$. Sin embargo, sabemos que l_m puede ser mayor a n, por lo que $|M|$ es mucho mayor a $|H|$. Podemos ver entonces que como el dominio de $H(m)$ es mayor a la imagen, necesariamente sucede que $\exists x, x' \in M$ tales que $H(x) = H(x')$, por el *principio del palomar*, o sea que el que una función sea resistente a colisiones no significa que no existan mensajes que causen colisiones, es más, estamos seguros de que existen dichas colisiones, sin embargo la función debe de asegurar que el encontrarlas sea muy difícil.

5. ¿Qué es "birthday attack"? ¿Cómo se puede evitar? ¿Qué propiedad de las funciones hash seguras protege contra este tipo de ataques?

Es un ataque basado en la *Paradoja del cumpleaños* de la Teoría de probabilidad. Este problema describe un escenario en el que dado un grupo de personas, encontrar a dos individuos que comparten cumpleaños no es tan poco probable, contrario a la intuición inmediata. De hecho, con tan sólo 23 personas, se tiene una probabilidad del 50 % y con 70 personas del 99 %. Este problema trasladado a criptología, refiere a la probabilidad de que dos conjuntos de datos distintos tengan una misma imagen dada una función de *hash*. Sea H el número de posibles salidas de una función de hash f. El número de entradas necesarias para encontrar la primer colisión en f, es aproximado por

$$Q(H) \approx \sqrt{\frac{\pi}{2} H} \tag{1}$$

Por ejemplo, en una función de *hash* de 64 bits, $H = 1{,}8 \times 10^{19}$, entonces $Q(H) \approx 5{,}38 \times 10^9$ (5 billones de entradas) serían necesarios para generar una colisión por fuerza bruta. A esto se le

[1] Por "muy difícil" se entiende que no sea factible encontrar la respuesta con cualquier algoritmo que corra en tiempo polinomial.

conoce como la *cota del cumpleaños*. Dicho esto, es posible evitar la vulnerabilidad a este ataque, con un *hash* de salida lo suficientemente grande para que sea inviable computacionalmente encontrar una colisión. Precisamente es el objetivo de las funciones de *hash* criptográficas definir un tamaño de bloque suficientemente largo para que sea muy difícil encontrar una colisión lo que los protege de este tipo de ataques, sin embargo, por el principio del palomar, es seguro que existan colisiones; además, para algunas funciones como *MD* y *SHA*, cuyas imágenes no están distribuidas uniformemente, una colisión puede encontrarse más prontamente por un ataque de este estilo.