LibreIM

Seguridad y Protección de Sistemas Informáticos
Grado en Ingeniería Informática
Universidad de Granada

31 de diciembre de 2019

Sugerencias para el examen

Fijamos a continuación la notación que seguiremos en este documento.

- Denotamos por ω al primer ordinal infinito, cuya cardinalidad coincide con la de $\mathbb{N}$. Podemos entenderlo como una forma de «ordenar» los números naturales, es decir, $\omega = \{0, 1, 2, \dots\}$. Dado $n \in \omega^* := \omega - \{0\}$, se tiene que $n = \{0, 1, \dots, n-1\}$, que identificamos cuando sea preciso con el número natural n.

- $\mathscr{A}$ será un alfabeto: un conjunto finito no vacío de símbolos. El conjunto de expresiones o palabras que podemos formar por yuxtaposición de símbolos de $\mathscr{A}$ lo denotaremos como $\exp(\mathscr{A})$, es decir,

$$\exp(\mathscr{A}) = \bigcup_{k \in \omega} \mathscr{A}^k,$$

  donde $\mathscr{A}^0 = \{\varepsilon\}$ (la palabra vacía) y $\mathscr{A}^k = \mathscr{A}^{k-1}\mathscr{A}$ para $k > 0$. Reservaremos la notación $\exp(\mathscr{A})^*$ para indicar que no se admite la palabra vacía.

- Dado $s \in \exp(\mathscr{A})$, existirá $k \in \omega$ tal que $s \in \mathscr{A}^k$, es decir, $s = s_1 s_2 \cdots s_k$ con $s_i \in \mathscr{A}$ para todo i. Denotamos $s = \langle s_j \rangle_{j \in k}$, y entonces k será la *longitud* de s, denotada por $\mathit{len}(s)$. Cuando la longitud de la palabra no sea relevante, escribiremos simplemente $s = \langle s_j \rangle_j$.

- En ocasiones será necesario poner en correspondencia los símbolos del alfabeto con números naturales. Si $\mathscr{A}$ es un alfabeto de $n \in \omega^*$ símbolos, consideraremos fijada una función biyectiva $f : \mathscr{A} \longrightarrow n$. De esta forma, si $s \in \exp(\mathscr{A})$ nos referiremos a $\langle f(s_j) \rangle_j$ como $f(s)$.

- Un *criptosistema* será una terna $\langle \mathscr{A}, E, D \rangle$, donde $\mathscr{A}$ es un alfabeto y E (resp. D) es una aplicación de $\exp(\mathscr{A})$ en $\exp(\mathscr{A})$, de forma que $D \circ E$ sea la identidad en $\exp(\mathscr{A})$. Nos referiremos a los elementos de $\exp(\mathscr{A})$ como *texto plano* cuando los veamos como argumentos de E (función de cifrado) y como *texto cifrado* cuando los veamos como argumentos de D (función de descifrado). Es usual que las funciones de cifrado y descifrado dependan de un parámetro llamado *clave*, que en general no tiene que coincidir para las dos. Esta situación la denotamos como E_{k_e} y D_{k_d}.

Cuestión 1. *Cifrado de Vigenère.*

Respuesta. Dado $n \in \omega^*$ y $\mathscr{A}$ un alfabeto de n símbolos, el cifrado de Vigenère es un criptosistema polialfabético que consiste en una clave $\alpha \in \exp(\mathscr{A})^*$ y sendas funciones E_α y D_α para cifrar y descifrar respectivamente. Podemos definir la función $E_\alpha : \exp(\mathscr{A})^* \longrightarrow \exp(\mathscr{A})^*$ como

$$E_\alpha(s) = \langle f^{-1}((f(s_j) + f((\alpha^{len(s)})_j)) \bmod n)\rangle_j,$$

donde $s = \langle s_j\rangle_j$ y la expresión $\alpha^{len(s)}$ es la yuxtaposición de la expresión α consigo misma y eventual truncamiento final hasta conseguir una palabra con la misma longitud que s. Hacemos esto para asegurar la existencia de una letra en cada posición $j \in len(s)$ de α. Por ejemplo, si $\alpha = HOLA$ entonces α^{10} = HOLAHOLAHO. De modo análogo se define $D_\alpha : \exp(\mathscr{A})^* \longrightarrow \exp(\mathscr{A})^*$ como:

$$D_\alpha(s) = \langle f^{-1}((f(s_j) - f((\alpha^{len(s)})_j)) \bmod n)\rangle_j.$$

Se comprueba que efectivamente la terna $\langle \mathscr{A}, E_\alpha, D_\alpha\rangle$ es un criptosistema, es decir, $D_\alpha \circ E_\alpha = 1_{\exp(\mathscr{A})^*}$. En efecto, si $s = \langle s_j\rangle_j \in \exp(\mathscr{A})^*$, se tiene que:

$$\begin{aligned}
D_\alpha(E_\alpha(s)) &= D_\alpha(\langle f^{-1}((f(s_j) + f((\alpha^{len(s)})_j)) \bmod n)\rangle_j) \\
&= \langle f^{-1}((f(f^{-1}((f(s_j) + f((\alpha^{len(s)})_j)) \bmod n)) - f((\alpha^{len(s)})_j)) \bmod n)\rangle_j \\
&= \langle f^{-1}(((f(s_j) + f((\alpha^{len(s)})_j)) \bmod n) - f((\alpha^{len(s)})_j)) \bmod n)\rangle_j \\
&= \langle f^{-1}(f(s_j) \bmod n)\rangle_j \\
&= \langle f^{-1}(f(s_j))\rangle_j \\
&= \langle s_j\rangle_j \\
&= s.
\end{aligned}$$

Por último, enunciamos un resultado que nos dice que en realidad ambas funciones son la misma salvo un cambio de clave.

Teorema. *Si $\alpha = \langle \alpha_j\rangle_j$ es una clave de Vigenère, definiendo $\alpha' = \langle(-\alpha_j) \bmod n\rangle_j$ tenemos que $D_\alpha = E_{\alpha'}$.*

La principal debilidad del cifrado Vigenère es la naturaleza repetitiva de su clave. Si un criptoanalista adivina correctamente la longitud de la clave, el texto cifrado puede tratarse como cifrados entrelazados de César, que pueden romperse fácilmente de forma individual.

Cuestión 2. *Explicar la transformación `SubBytes()` que es parte del algoritmo simétrico de cifrado AES.*

Respuesta. El algoritmo de cifrado de AES realiza en cada ronda una serie de transformaciones a nivel de bytes. La primera de estas transformaciones es `SubBytes()`, que consiste en sustituir los bytes del estado según una tabla de sustitución, llamada S-box. Esta sustitución se realiza de forma no lineal e independiente para cada byte del estado.

La tabla de sustitución es una matriz cuadrada de orden 16 (no simétrica), que contiene en cada posición (i, j) un byte en hexadecimal que representa la transformación del byte s_{ij} del estado. La construcción de la tabla se realiza de la siguiente forma, dada una entrada xy:

1. Si $xy = 00$, se mantiene igual. Si $xy \neq 00$, se calcula el inverso de xy en $\mathrm{GF}(2^8)$, que será otro byte $b = b_7 \cdots b_0$.

2. El bit i-ésimo del byte b, b_i, se transformará en b'_i de la siguiente forma:

$$b'_i = b_i + b_{(i+4) \bmod 8} + b_{(i+5) \bmod 8} + b_{(i+6) \bmod 8} + b_{(i+7) \bmod 8} + c_i,$$

   donde c_i es el i-ésimo bit del byte {63} ó {01100011} para todo $0 \leq i < 8$. El byte $b' = b'_7 \cdots b'_0$ resultante ocupará la posición (x, y) de la tabla.

Cuestión 3. *Limitaciones de los sistemas simétricos de cifrado en la comunicación y cómo la criptografía de clave pública los ha resuelto.*

Respuesta. En la criptosistemas de clave simétrica, emisor y receptor acuerdan una clave k que marca como funcionarán las funciones E_k y D_k, y cualquier persona que tenga esta clave puede entrar en la comunicación.

Esto conlleva diferentes problemas:

- El intercambio inicial de la clave se tiene que hacer de algún modo seguro antes de poder establecer el canal encriptado. Esto es uno de los mayores problemas de la criptografía clásica. Si bien, se pudieron vadear en un inicio por medios militares y personas de confianza, esto se hace imposible cuando, por ejemplo, estamos hablando de encriptar transacciones comerciales donde los involucrados son completos desconocidos. Además el coste de este intercambio es demasiado alto considerando la posibilidad de que no se vuelva a necesitar nunca más el canal.

- Cuando consideramos una red de usuarios, hemos de tener una clave para cada par de usuarios para tener máxima seguridad. Esto hace que tengamos un ratio de crecimiento del número de claves cuadrático en relación al tamaño de la red. Claramente no es un sistema escalable. Por ejemplo, con 10 usuarios necesitamos 45 claves diferentes y con 100 usuarios necesitaremos 4950 claves.

  Por este motivo no es fácil añadir un usuario a la red y actualizar la base de datos. Este gran número de claves es también un problema de seguridad al ser necesario que cada usuario guarde sus claves.

- Estos sistemas también adolecen de falta de crédito, ya que ambos usuarios tienen la misma clave en cada pareja emisor receptor.

  Si Alice y Bob comparten una clave secreta y se da que Carol la descubre, esta puede interceptar los mensajes de Alice a Bob y modificarlos sin que

Bob lo descubra. También es posible que Bob envíe mensajes falsos y después se encubra en esta posibilidad para eximirse de las culpas.

A raíz de esto, necesitamos un nuevo criptosistema que satisfaga: autenticación, integridad y no repudiabilidad. El criptosistema de clave pública soluciona estos problemas:

- Al tener una función E pública y una D privada deja de haber problema con el intercambio de claves en un canal seguro, dado que este intercambio inicial no se produce.
- Solo se mantienen dos claves por usuario del círculo, lo que es mucho más clave. Más aún, en la base de datos solo se guarda la clave pública de cada usuario.
- Se implementa un sistema de autenticación y no repudio del siguiente modo: Como E y D son biyecciones e inversas. Sean $A = \{E_a, D_a\}$ las funciones de Alice y $B = \{E_b, D_b\}$ las funciones de Bob. Para que Alice envíe un mensaje a Bob encripta el mensaje m con $E_b(D_a(m))$. Cuando Bob lo recibe, primeramente lo desencripta con su función privada, y posteriormente para comprobar que efectivamente es de Alice lo desencripta con su función pública. Así se sabe que es de Alice y solo puede ser suyo pues es la única que posee la función inversa de E_a.En la criptosistemas de clave simétrica, emisor y receptor acuerdan una clave k que marca como funcionaran las funciones E_k y D_k, y cualquier persona que tenga esta clave puede entrar en la comunicación.

Cuestión 4. *Explicar los fundamentos de la criptografía de clave pública y las líneas fundamentales de la firma a través de la misma.*

Cuestión 5. *Enumerar resumidamente las precauciones más destacables a tomar al generar un cículo de comunicación basado en RSA.*

Respuesta. Para adherirse a un círculo RSA cada usuario elige dos números primos p y q y dos exponentes e y d de forma que $(e, \Phi(n)) = 1$ y $ed \equiv 1 \pmod{\Phi(n)}$, donde $n = pq$. Se ha conjeturado que la seguridad de este criptosistema reside en la dificultad de descomponer el número n en sus factores primos. Los usuarios incluirán en un archivo público la pareja $\langle n, e\rangle$ y guardarán celosamente cualquiera de las entradas de la 4-tupla $\langle p, q, \Phi(n), d\rangle$. Además, deben tomar las siguientes precauciones:

1. El número n debe superar (a comienzos del siglo XXI) los 308 dígitos para que no sea factible su factorización. Algunas implementaciones actuales de RSA emplean módulos de 617 dígitos (2048 bits).
2. Los primos p y q deben ser ambos elevados para que sea difícil aplicar algoritmos de factorización como la criba general del cuerpo de números. Además, jamás deben ser elegidos de entre una lista conocida ni ser próximos el uno

al otro; de lo contrario podría emplearse eficientemente el método de factorización de Fermat.

3. El valor de $(p-1, q-1)$ no debe ser elevado en exceso, y nunca debe ocurrir que $p-1 \mid q-1$. Si $(p-1, q-1)$ es muy elevado, $[p-1, q-1]$ sería pequeño en comparación con $\Phi(n)$ y factible de ser encontrado por fuerza bruta. Es indeseable también que los factores primos de $\Phi(n)$ sean pequeños.

4. Para solucionar los problemas del apartado anterior, es deseable que p y q sean *primos seguros*. Además, tanto $p-1$ como $q-1$ deben tener factores primos elevados para evitar ataques de cifrados iterados.

5. El exponente d debe ser elevado, y ésta es la razón de comenzar eligiéndolo para luego determinar e. Así se evita que pueda ser encontrado mediante procedimientos de prueba y error.

6. Dos usuarios nunca deben elegir el mismo módulo como parte de su clave pública. Si un tercer usuario cifra un mismo mensaje para ellos, este puede ser leído si los exponentes públicos de ambos usuarios son primos relativos.

7. No es recomendable tener valores pequeños de e y, en caso de tenerlo, jamás debe ser elegido por varios integrantes de un mismo círculo RSA. De lo contrario, si se enviara un mismo mensaje cifrado a estos integrantes se podría obtener el texto plano aplicando convenientemente el *teorema chino del resto*.

8. Cuidarse de mensajes inocultables. Existen mensajes para determinadas claves cuyo cifrado es igual al mensaje original.

Cuestión 6. *Protocolo de intercambio de llaves según el esquema de Diffie-Hellman y explicación de su supuesta fortaleza.*

Cuestión 7. *El criptosistema de ElGamal.*

Respuesta. El criptosistema de ElGamal es un criptosistema de clave pública propuesto por Taher ElGamal en 1985. Está basado en el protocolo de intercambio de claves de Diffie-Hellman.

Preliminares

En primer lugar, se debe elegir un primo p elevado y un elemento primitivo $g \in GF(p)^*$. Suponemos que las unidades de texto llano estarán expresadas en números de $GF(p)$, y llamamos $q = |GF(p)^*| = p-1$. Ahora, cada usuario elige aleatoriamente un valor x con $0 < x < q$, que será su clave privada. La clave pública será $y = g^x \bmod p$.

Cifrando el mensaje

Supongamos que Alice quiere compartir con Bob un mensaje $0 \leq m \leq q$. Para ello, recoge la clave pública de Bob (y_B), y calcula $K = y_B^{x_A} \bmod p$, donde x_A es su clave privada. Finalmente envía a Bob la pareja $c = \langle g^{x_A} \bmod p, mK \bmod p \rangle$.

Descifrando el mensaje

Cuando Bob recibe el mensaje de Alice obtiene una pareja de números $\langle c_1, c_2 \rangle$ menores que p. Para descifrarlo hace lo siguiente:

- Calcula el valor de K a partir de su clave privada, sin más que observar que $c_1^{x_B} = g^{x_A x_B} \bmod p = y_B^{x_A} \bmod p = K$.
- Ahora calcula el inverso de K módulo p y computa el valor de m haciendo $c_2 K^{-1} \bmod p = mKK^{-1} \bmod p = m$, lo que le permite recuperar el texto plano.

Vulnerabilidades

No es recomendable elegir la misma clave x_A para cifrar más de un bloque del mensaje. Esto se debe a que si un atacante supiera el contenido de un primer mensaje m_1, podría calcular el contenido de las porciones sucesivas. Sean

$$c_{11} \equiv g^{x_A} \bmod p, \quad c_{21} \equiv m_1 K \bmod p,$$
$$c_{12} \equiv g^{x_A} \bmod p, \quad c_{22} \equiv m_2 K \bmod p.$$

Entonces, es fácil calcular $m_2 m_1^{-1} \bmod p = c_{22} c_{21}^{-1} \bmod p$, de donde se deriva inmediatamente m_2 (multiplicando por m_1).

La seguridad de este criptosistema reside en la dificultad de cálculo del logaritmo discreto, por lo que es imprescindible tomar las precauciones necesarias para que no sea fácil calcular dicho logaritmo. A saber, necesitamos que $p - 1$ contenga al menos un factor primo elevado. Para ello podemos por ejemplo elegir p como un *primo seguro*, es decir, de la forma $p = 2r + 1$ con r primo.

Cuestión 8. *Explicación del algoritmo de firma estándar (DSA).*

Cuestión 9. *Rasgos esenciales de SSH: cifrado, funcionamiento, negociación de cifrado para la sesión y autenticación del acceso del usuario al servidor*