| | |
|---|---|
| **MA-53G Intr. a la Criptografía** | **Semestre Primavera, 1996** |
| Tarea No. 1 : 10 de Septiembre | |
| *Profesor: Marcos Kiwi* | *Fecha de entrega: 1 de Octubre* |

## Elija 5 de los siguientes 6 problemas.

De haber clarificaciones se publicarán en el newsgroup del curso.

PROBLEMA 1: Sea $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ un criptosistema perfectamente secreto bajo la distribución de textos planos $P_{\mathcal{P}}$ y de claves $P_{\mathcal{K}}$. De una demostración o un contraejemplo para la siguiente afirmación: $S$ es un criptosistema perfectamente secreto bajo la distribución de textos planos $P'_{\mathcal{P}}$ y de claves $P_{\mathcal{K}}$, cualquiera sea la distribución $P'_{\mathcal{P}}$.

PROBLEMA 2: Sea $k$ una clave secreta compartida por Alicia y Roberto. Roberto desea una garantía que los mensajes que recibe de Alicia realmente provienen de ella. Para lograr esto Alicia envía sus mensajes junto con un MAC que depende de la clave $k$.

Se dice que un MAC es seguro con respecto a ataques de texto elegido si un adversario, Eva, una vez obtenido los MAC, $MAC_k(m_i)$, para $n$ mensajes distintos de su elección $m_1, \ldots, m_n$, no podrá producir un *nuevo* mensaje $m \notin \{m_1, \ldots, m_n\}$ y su respectivo MAC, $MAC_k(m)$ .

Sea $f_k\colon \{0,1\}^l \to \{0,1\}^l$ una función de encripción con clave $k$ (por ejemplo si $l = 64$ la función $f$ podría ser DES). Sea además

$$MAC_k(m) \quad = \quad f_k(f_k(\ldots f_k(f_k(m_1) \oplus m_2) \oplus \ldots m_{b-1}) \oplus m_b)\,,$$

donde $m = m_1 \ldots m_b$ es un mensaje de $bl$ bits, $m_i \in \{0,1\}^l$ para todo $i \in \{1, \ldots, b\}$, y $\oplus$ denota el XOR de dos secuencias de bits.

1. Muestre que si los mensajes $m$ tienen largo variable (i.e., $b$ es un entero positivo cualquiera que no esta fijo), entonces se puede diseñar un ataque de texto plano elegido contra el MAC antes definido.

   Indicación: Elija dos textos planos de $l$ bits y a partir del MAC de los mismos construya el MAC de un texto plano de $2l$ bits.

2. Suponga que el largo de los mensajes, $m$, cuyo MAC se desea calcular siempre tienen largo menor que $2^l$. Una forma de evitar el ataque del punto anterior sería aumentar $m$ en $l$ bits de forma que los últimos $l$ bits representen el largo de $m$ expresado en binario y luego calcular el MAC del mensaje así obtenido. Pruebe que todavía es posible montar un ataque de texto plano elegido contra esta modificación del MAC antes definido.

PROBLEMA 3: Sea $N$ un entero muy grande. $N$ se mantiene secreto y se usa para activar un sistema de misiles, i.e., el conocer $N$ permitiría lanzar los misiles. Suponga que se tiene un comandante y $n$

oficiales a su mando. En el evento que el comandante (que conoce $N$) este incapacitado, se quiere que los $n$ oficiales a su mando tengan suficiente información parcial sobre $N$ de manera que cualquiera tres de ellos (pero nunca dos de ellos) puedan acordar lanzar los misiles.

1. Sean $p_1, \ldots, p_n$ números primos distintos, todos mayores que $N^{\frac{1}{3}}$ pero bastante menores que $N^{\frac{1}{2}}$. Usando los $p_i$ describa la información parcial que se le debe dar a cada oficial y como podrían estos utilizarla en caso de incapacidad del comandante.
2. Generalize el sistema anterior a la situación en que se quiere que cualquier conjunto de $k$ ($k \geq 2$) oficiales trabajando conjuntamente puedan lanzar los misiles (pero cualquier conjunto de $k-1$ de ellos no puedan activar el sistema de misiles).

Problema 4:

1. Suponga que tres individuos $A, B, C$ estan usando RSA y sus claves públicas son $(n_A, 3)$, $(n_B, 3)$, $(n_C, 3)$ respectivamente. Suponga además que $D$ quiere enviar el *mismo* mensaje $m \in \{0, \ldots, n-1\}$, donde $n = \min\{n_A, n_B, n_C\}$, a los tres individuos $A, B, C$. Para ello $D$ calcula
$$y_A = m^3 \bmod n_A\,, \qquad y_B = m^3 \bmod n_B\,, \qquad y_C = m^3 \bmod n_C\,,$$
y se los envía a $A$, $B$, y $C$ según corresponda.

   Pruebe que un adversario que intercepta los tres mensajes $y_A, y_B, y_C$ puede determinar $m$ aunque desconozca las claves secretas de $A, B, C$.
2. Pruebe que RSA es una regla de encripción multiplicativa en el sentido que si la clave pública es $k = (n, b)$ entonces la función de encripción $f(x) = x^b \bmod n$, donde $x \in \mathcal{Z}_n$ satisface que
$$f(x)f(x') \;\equiv\; f(x \cdot x') \,(\bmod\, n)\,, \qquad \forall x, x' \in \mathcal{Z}_n\,.$$
Suponga que un adversario descubre un algoritmo eficiente (polinomial en $\log n$), para decriptar con probabilidad al menos $\epsilon$ un texto plano elegido al azar en $\mathcal{Z}_n$, donde $\epsilon \in [0, 1]$ es una constante. Pruebe que dicho adversario puede diseñar un algoritmo probabilista eficiente (polinomial en $\log n$), que decripta cada mensaje encriptado con $f$ con probabilidad al menos $1 - \frac{1}{n^c}$, donde $c$ es una constante positiva arbitraria.

Problema 5: Se dice que $x$ es un residuo cuadrático módulo $n$, donde $n$ es un entero, si existe un entero $b$ tal que $a \equiv b^2 \,(\bmod\, n)$. En lo que sigue sea $p$ un número primo impar.

1. Sea $g$ un generador de $\mathcal{Z}_p$ y $a \in \mathcal{Z}_p^*$. Pruebe que $a$ es un residuo cuadrático si y sólo si existe un entero no-negativo $k$ tal que $g^{2k} \equiv a \,(\bmod\, p)$.
2. Pruebe que si $g$ es un generador de $\mathcal{Z}_p$ entonces $g^{(p-1)/2} \equiv -1 \,(\bmod\, p)$.
3. Sea $a \in \mathcal{Z}_p^*$. Pruebe que:
   - Si $p \equiv 3 \,(\bmod\, 4)$, entonces $a$ es un residuo cuadrático y $-a$ no lo es, o $-a$ es un residuo cuadrático y $a$ no lo es.

- Si $p \equiv 1 \pmod 4$, entonces $a$ y $-a$ son residuos cuadráticos, o $a$ y $-a$ no son residuos cuadráticos.

PROBLEMA 6: El proposito de este problema es desarrollar un programa que permita encontrar una primera aproximación de la entropía de un lenguaje.

Sea $x = x_1 x_2 \ldots x_N$, una secuencia de caracteres sobre el alfabeto $\Sigma$ y $N$ un número muy grande. Sea $n$ un número entero positivo menor que $N$. Dado $n$, a $x$ se le puede asociar una distribución de probabilidad $P^{(n)}$ cuyo soporte es $\Sigma^n$ tal que

$$P^{(n)}(u) \quad = \quad \frac{1}{N-n+1} \left|\{i \mid x_i \ldots x_{i+n-1} = u\}\right| , \qquad \forall u \in \Sigma^n .$$

La entropía de $P^{(n)}$, denotada por $H(P^{(n)})$, se define como

$$H(P^{(n)}) \quad = \quad - \sum_{u \in \Sigma^n} P^{(n)}(u) \log P^{(n)}(u) ,$$

donde $x \log x = 0$ si $x = 0$, y el logaritmo es base 2. La entropía de un lenguaje $L$, denotada $H_L$, se define como $H_L = \lim_{n \to \infty} \frac{1}{n} H(P^{(n)})$, donde $P^{(n)}$ es como se definió más arriba donde $N$ tiende a infinito y $x$ es un texto típico en el lenguaje $L$.[1] La entropía de un lenguaje nos dice cual es el promedio de la información contenida en un texto típico escrito en dicho lenguaje.

Escriba un programa en C o C++ que tome tres argumentos $b \in \{0, 1\}$, un entero positivo $n$, y un archivo de caracteres `ASCII`. El programa debe entregar la siguiente salida:

- Si $b = 0$ el segundo argumento, $n$, debe ignorarse y la salida del programa debe ser una tabla de la distribución $P^{(1)}$ asociada al texto del archivo de entrada, i.e., una tabla de las frecuencias relativas con que aparece cada caracter del archivo de entrada (Nota: ignore los fines de línea y fin de archivo).

- Si $b = 1$ entonces el programa debe descartar todas los caracteres del archivo de entrada excepto las letras del alfabeto (la ñ no debe ser descartada). Las letras mayúsculas deben identificarse con las minúsculas. Para el texto así obtenido debe calcularse $H(P^{(n)})$. (Nota: en el cálculo de $P^{(n)}$ deben ignorarse los fines de línea y fin de archivo.)

Su programa debe funcionar correctamente al menos cuando $b = 0$, $b = 1$ y $n \in \{1, 2, 3\}$. El código de su programa debe enviarlo a `jaracena@dim.uchile.cl`.

<u>Nota</u>: La entropía de un lenguaje es invariante bajo ciframientos por substitución. Por lo tanto el calcular la entropía de un mensaje grande encriptado utilizando cifrados por substitución permite reconocer el idioma en que fue escrito el texto plano. Esta información es útil para decriptar el texto cifrado. Observe también que la entropía de un texto sobre el alfabeto $\Sigma$ elegido al azar es $\log |\Sigma|$. La entropía del inglés y el castellano son significativamente menores que este valor.

[1] Para más detalles acerca de entropía consultar D. R. Stinson, *Cryptography Theory and Practice, CRC, 1995.*