MA-53G Intr. a la Criptografía	**Semestre Otoño, 1998**

Tarea No. 2 : 9 de Junio

Profesor: Marcos Kiwi *Auxiliar: Alejandro Hevia*

Fecha de entrega: 23 de Junio

Las mismas reglas que se aplicaron en el caso de la tarea No. 1 son válidas para esta tarea.

De haber clarificaciones se comunicarán en clases o se enviarán por correo electrónico.

PROBLEMA 1: En la literatura se han propuesto varias funciones hash basadas en criptosistemas de chorro (también denominados de flujo) pero sin utilizar la clave secreta. Las funciones hash basadas en este tipo de métodos son susceptibles del mismo tipo de ataques que cualquier función hash, pero además adolecen de otras debilidades. En este problema se ilustra una de estas debilidades.

Suponga que se divide un mensaje $M \in \{0,1\}^*$ en bloques[1] de m bits $M_1, M_2, \ldots, M_N \in \{0,1\}^m$ y se tiene un criptosistema convencional (por ejemplo DES) con función de encripción E_k. Asumiendo que E_k es una función bien definida para toda clave $k \in \{0,1\}^m$ cuyo dominio y recorrido son $\{0,1\}^n$, $n < 2m$, se sugiere calcular el valor hash G del mensaje M de la siguiente forma:

1. $H_0 \in \{0,1\}^n$ valor inicial fijo y conocido.
2. $H_i = E_{M_i}[H_{i-1}]$, $i = 1, \ldots, N$.
3. $G = H_N$.

Pruebe que dada una constante $\epsilon > 0$ y bloques $Q_1, Q_2, \ldots, Q_{N-2} \in \{0,1\}^m$ se pueden encontrar, con probabilidad al menos $1 - \epsilon$ y realizando a lo más $O(2^{n/2})$ evaluaciones de la función de encripción, dos bloques $X, Y \in \{0,1\}^m$ tales que la concatenación de los bloques $Q_1, Q_2, \ldots, Q_{N-2}, X, Y$ forman un mensaje con valor hash G.

En caso de necesitarlo, especifique que supuestos razonables debe satisfacer el criptosistema utilizado para que su argumento sea válido.

<u>Indicación</u>: Recuerde que $(1+x)^n \leq e^{nx}$, para todo $x \geq 0$.

PROBLEMA 2: Estudiaremos el concepto de *sistemas públicos de firmas digitales de grupos*.

Suponga que una corporación requiere que para que un documento sea firmado en nombre de la corporación sus n vice-presidentes deben firmar el documento. Una posible forma de lograr lo ante-

[1] Agregando 0's a M si fuese necesario.

rior es que cada vice-presidente cree su propia firma digital utilizando un sistema de firmas digitales público. Sólo los documentos que contengan las firmas de los n vice-presidentes se considerarán firmados por la corporación. Este sistema es ineficiente pues requiere agregar n firmas a un documento para que este quede validado.

Los sistemas de firmas digitales de grupos solucionan el problema mencionado en el parrafo anterior. Esto se logra compartiendo una clave secreta entre los vice-presidentes. Sea k una clave secreta, y k_i la parte de la clave de propiedad del i-ésimo vice-presidente. En el momento de la firma el i-ésimo vice-presidente produce una firma parcial $sig_{k_i}(M)$ las cuales luego son combinadas para formar la firma $sig_k(M)$ del documento M.

Por ejemplo, consideremos RSA con clave pública (N, e) y clave secreta d. Una forma de implementar un sistema de firmas digitales de grupo como el discutido más arriba es que un cierto árbitro de confianza de todos genere una clave (p, q, e, d) (donde p y q son primos tales que $N = pq$, y $ed = 1 \bmod \varphi(N)$) haga público (N, e), le entregue al i-ésimo vice-presidente el valor d_i donde

$$d = d_1 + d_2 + \cdots + d_n \bmod \varphi(n)\,,$$

y luego el árbitro se auto-destruya.

Más tarde cuando los vice-presidentes quieran firmar un mensaje M el i-ésimo vice-presidente puede producir $\sigma_i = M^{d_i} \bmod N$ y luego en forma conjunta pueden calcular

$$\sigma = \prod_{i=1}^{n} \sigma_i \bmod N = \prod_{i=1}^{n} M^{d_i} \bmod N = M^d \bmod N\,,$$

que es la firma correcta de la corporación para el documento M.

Considere ahora el siguiente sistema de firmas: sea p primo y g un generador de $\mathcal{Z}_p^*$, $a \in \mathcal{Z}_{p-1}$ la clave secreta y $\beta = g^a \bmod p$ la clave pública. Para firmar un mensaje x se elije un número $k \in \mathcal{Z}_{p-1}$ y se calcula $\gamma = g^k \bmod p$ y $\delta = a\gamma + k - x \bmod (p-1)$. El par (γ, δ) es la firma. Para verificar la firma uno debe comprobar que $g^{x+\delta} \equiv \gamma\beta^\gamma \pmod{p}$.

1. Describa un sistema de firmas digitales de grupo similar al basado en RSA pero basado en el sistema de firmas digitales recien descrito.
2. Basandose en la parte anterior, y asumiendo que todos los vice–presidentes tienen acceso a una misma fuente de aleatoreidad, proponga un sistema de firmas digitales de grupo pero que no requiera un árbitro de confianza de todas las partes involucradas.

PROBLEMA 3:

El objetivo de esta pregunta es implementar RSA.

Para esto, se dispone de una biblioteca de procedimientos para el manejo de números grandes y de números pseudoaleatorios. Las funciones requeridas para esta tarea estan explicadas en

`http://www.dcc.uchile.cl/~ahevia/ma53g/tarea2.html`

Se le pide a Ud.:

(a) Implementar el procedimiento `GeneraClaves`(p, q, n, d, e) que genere números[2] primos p y q, un número compuesto $n = pq$ y números e y d tales que $ed = 1 \bmod \varphi(n)$, que sirvan como parámetros para una instancia especifíca del criptosistema RSA.

(b) Implementar la función `Encripta`(n, e, b) que encripta la cadena de caracteres b, de tamaño arbitrario, usando RSA con los parámetros definidos en (a) y devuelve como resultado el texto encriptado.

(c) Implementar la función `Decripta`(n, d, c) que decripta el texto c encriptado con la función de la parte anterior usando el módulo n y el exponente secreto d.

(d) Crear un programa de ejemplo que utilice las anteriores funciones (es decir, primero genere claves públicas y privadas, y luego encripte y decripte un texto de tamaño arbitrario).

Cualquier consulta, dirigirla al profesor auxiliar (`ahevia@dcc.uchile.cl`) o postearla en el newsgroup del curso.

[2]Donde *número* se entiende por la representación usada en los procedimientos de la biblioteca mencionada.