**MA-53G Intr. a la Criptografía** **Semestre Primavera, 1996**

# Tarea No. 2 : 19 de Noviembre

*Profesor: Marcos Kiwi* *Fecha de entrega: 3 de Diciembre*

Las mismas reglas que se aplicaron en el caso de la tarea No. 1 son válidas para esta tarea, excepto que ahora sólo se permite colaborar en grupos de a lo más dos (2) personas.

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

PROBLEMA 1: Se dice que el lenguaje $L \subseteq \{0,1\}^*$ está contenido en RP si existe un polinomio $p$ y una máquina de Turing probabilista que en entradas de largo $n$ se detiene en a lo más $p(n)$ pasos, acepta con probabilidad al menos $2/3$ aquellas entradas que están en $L$, y acepta con probabilidad 0 aquellas entradas que no están en $L$. Por otro lado, se dice que $L \subseteq \{0,1\}^*$ está en coRP si $\bar{L} = \{\omega \in \{0,1\}^* \mid \omega \notin L\}$ esta en RP.

Se dice que el lenguaje $L \subseteq \{0,1\}^*$ está en ZPP si existe un polinomio $p$ y una máquina de Turing probabilista que se detiene en un tiempo esperado de a lo más $p(n)$ en entradas de largo $n$, y que acepta si y sólo si la entrada está en $L$.

Pruebe que ZPP $=$ RP $\cap$ coRP.

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 anterior 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

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

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 $(\gamma, \delta)$ 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. Proponga un sistema de firmas digitales de grupo pero que no requiera un árbitro de confianza de todas las partes involucradas.

Problema 3: 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. $H_0$ valor inicial.
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.

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