Criptografía y Seguridad Computacional **2017-01**

Ejercicios de estudio Prueba 2

1. Hemos estudiado en clases que la forma correcta de asegurar privacidad e integridad en el mundo simétrico es primero cifrar y luego autenticar ($\langle c = \mathsf{Enc}_k(m), \mathsf{Mac}_{k'}(c)\rangle$). Argumente por qué el análogo asimétrico no es seguro ($\langle c = \mathsf{Enc}_e(m), \mathsf{Sign}_s(c)\rangle$). Proponga una solución.

2. Considere el esquema de firmas digitales de *Schnorr*, definido a continuación. Sea $\mathbb{G}$ un grupo generado por g de orden q.

   La llave secreta de firma x uniforme en $\mathbb{Z}_q \setminus \{0\}$ y la llave pública verificadora es el elemento $y = g^x$.

   Para firmar un mensaje m, el firmante elige k aleatorio en $\mathbb{Z}_q \setminus \{0\}$ y computa:

   a) $r = g^k$.

   b) $e = H(m, r)$ en donde $H : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q \setminus \{0\}$

   c) $s = (k - xe) \mod q$.

   La firma es el par (r, s).

   Para verificar una firma (r, s) para un mensaje m, the receptor computa $e = H(m, r)$, y verifica que $g^s = r \cdot (y^e)^{-1}$.

   El esquema anterior es infalsificable si modelamos H como un oráculo aleatorio. Demuestre que si eliminamos r del hash (es decir, definimos $e = H(m)$), entonces existe un adversario que falsifica firmas sin saber la llave secreta.

3. Sea $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ un esquema de firmas digitales infalsificable. Considere el esquema de firmas digitales $\Pi_y = (\mathsf{Gen}, \mathsf{Sign}_y, \mathsf{Vrfy})$ tal que:

   - Si $H(0) \neq y$ entonces $\mathsf{Sign}_{y_{sk}}(m) = \mathsf{Sign}_{sk}(m)$.
   - Si $H(0) = y$, entonces $\mathsf{Sign}_{y_{sk}}(m) = sk||\mathsf{Sign}_{sk}(m)$.

   a) Demuestre que si H es un oráculo aleatorio, entonces Π_y es infalsificable para *cualquier* y.

b) Demuestre que existe un y tal que Π_y no es seguro si H es instanciado con una función de hash fija (sin llave), como por ejemplo SHA-1.

4. Sea N, e, d, generados por GenRSA. Imagine que utilizamos el siguiente padding para el cifrador RSA: con input m, se retorna $\texttt{0x00}||r||\texttt{0x00}||m$, en donde r tiene largo $||N||/2 - 16$ y el largo de m es exactamente $||N||/2$.

   Demuestre que RSA usando este padding no es CCA seguro (es decir, el adversario puede distinguir entre el cifrado 2 mensajes m_0 y m_1 si tiene acceso al oráculo de descifrado). Hint: Dado el texto cifrado $c = \mathsf{Enc}_{pk}(m_b)$, multiplique c por una constante elegida cuidadosamente.

5. Sea $H = (\mathsf{Gen}, \mathsf{Hash})$ una función de hash definida como:

   - $\mathsf{Gen}(1^\lambda)$: $\langle \mathbb{G}, q, g \rangle \leftarrow \mathcal{G}(1^\lambda)$, con $q = |\mathbb{G}|$ primos, $a, b \xleftarrow{\$} \mathbb{Z}_q$ y retorna $s = \langle q, g, a, b \rangle$.
   - $\mathsf{Hash}^s(x_1, x_2, x_3)$: retorna $g^{x_1} \cdot g^{a \cdot x_2} \cdot g^{b \cdot x_3}$

   Demuestre que si existe un algoritmo PPT $\mathcal{A}$ que encuentra colisiones en H, entonces existe un algoritmo PPT que distingue si T es uniforme en $\mathbb{G}$ o es g^{ab} dado el siguiente input:

   $$q, g, g^a, g^b, g^{a^2}, g^{b^2}, T$$

   con $q = |\mathbb{G}|$ primo, g un generador para $\mathbb{G}$, y a, b uniformes en $\mathbb{Z}_q$. (Similar a DDH, pero además el distinguidor tiene acceseo a g^{a^2} y g^{b^2}).