



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

**Criptografía y Seguridad Computacional – IIC3253**  
**Examen**  
**4 de Julio, 2022**

## Instrucciones

Este examen consta de 6 preguntas conceptuales sobre la materia vista durante el curso. Cada respuesta correcta suma dos puntos y cada respuesta incorrecta resta un punto. La respuesta a cada pregunta no puede tener más de 10 líneas. Este examen se aprueba con 4 puntos o más.

## Preguntas

Utilizaremos la notación vista en clases, en particular  $\parallel$  representa la concatenación.

1. Sea  $H$  una función de hash criptográfica (resistente a colisiones y pre-imágenes). Una de las siguientes aseveraciones es falsa. Indique cuál es y explique por qué es falsa.
  - (a) La función  $f(x) := H(x \parallel x)$  no es resistente a colisiones.
  - (b) La función  $H(x \parallel H(x))$  es también una función de hash criptográfica.

**Solución:** La afirmación falsa es la (a), puesto que una colisión para  $f$  implica una colisión para  $H$ . Luego un algoritmo eficiente para encontrar colisiones en  $f$  es también un algoritmo eficiente para encontrar colisiones en  $H$ .

**Corrección:** Se considerará buena una respuesta que de a entender que encontrar colisiones para  $f$  implica encontrar colisiones para  $H$ .

2. Suponga que encuentra en la calle un papel que en un lado tiene un par ordenado de números y en el otro dice literalmente “El par ordenado al reverso representa un número que ha sido encriptado con ElGamal sobre  $\mathbb{Z}_p^*$ , donde  $p$  vale 55340232221128654847”.  
¿Es la información disponible suficiente para decriptar el número original? De no ser así:  
¿Qué información faltaría?

**Solución:** La información disponible no es suficiente, dado que para decriptar un mensaje utilizando ElGamal se necesita conocer el valor secreto  $x$  y el generador  $g$ .

**Corrección:** Se considerará correcta una respuesta que mencione que para decriptar el número hacen falta los elementos mencionados arriba, o que mencione que hace falta la llave secreta (dado que la llave secreta, formalmente, contiene el secreto  $x$  y el generador  $g$ ).

3. Suponga que necesita tener una base de datos con información que permita autenticar usuarios en base a un password que los mismos usuarios proveen (como suele ocurrir en la Web). Explique por qué **no** sería una buena idea almacenar pares  $(e, H(p))$ , donde  $e$  es el correo del usuario,  $H$  es una función de hash criptográfica y  $p$  es el password del usuario. *Hint: Comience su respuesta con “Si se filtra la base de datos [...]”*

**Solución:** Si se filtra la base de datos, un atacante podría comparar los valores de hash en la base de datos con un conjunto de valores de hash pre-calculados (es decir, utilizar una *rainbow table*).

**Corrección:** Se considerará correcta una respuesta que de a entender que el sistema se puede atacar teniendo valores de hash pre-calculados para ciertas contraseñas.

4. Explique la estructura de un par de llaves RSA y la relación entre ellas.

**Solución:** Un par de llaves RSA se ve como  $(e, N)$  y  $(d, N)$ , y la relación entre estas llaves es que el valor  $N$  es la multiplicación de dos números primos  $P, Q$  y  $e$  es inverso de  $d$  en módulo  $(P - 1)(Q - 1)$ .

**Corrección:** Se considerará correcta una respuesta que mencione todos los elementos anteriores y que indique que  $e$  es inverso de  $d$  en módulo  $\phi(N)$ .

5. Un Message Authentication Code es una función para autenticar mensajes en base a una llave simétrica. Esta función toma una llave  $k$  y un mensaje  $m$  para generar un tag  $t$ . Intuitivamente, esperamos que una persona con la llave  $k$  pueda verificar que el tag  $t$  es válido para el mensaje  $m$ , y que alguien sin acceso a dicha llave no pueda autenticar mensajes que no se han autenticado antes. Para formalizar esta noción, en clases definimos un *juego* que constaba de cinco pasos.

- (a) El verificador genera una llave  $k$  al azar.
- (b) El adversario envía un mensaje  $m$  al verificador.
- (c) ...
- (d) Los pasos 2 y 3 se repiten tantas veces como quiera el adversario.
- (e) ...

Escriba los dos pasos faltantes, y luego explique cuándo decimos que el adversario gana el juego.

**Solución:** Los pasos faltantes son:

- (c) El verificador responde con el tag  $t$  generado con la llave  $k$  y el mensaje  $m$ .
- (e) El adversario envía un mensaje  $m_0$  junto a un tag  $t_0$ , tal que  $m_0$  es distinto de todos los mensajes enviados en el paso (b).

El adversario gana cuando  $t_0$  es un tag válido para el mensaje  $m_0$  dada la llave  $k$ .

**Corrección:** Se considerará correcta una respuesta que sea equivalente a lo anterior. En particular, la condición de que  $m_0$  sea distinto a todos los mensajes enviados anteriormente podría considerarse como parte de lo necesario para que el adversario gane el juego.

6. Ciertos documentos son aceptados por instituciones cuando están verificados por un notario y el notario asegura que el documento se puede utilizar en dicha entidad. A modo de ejemplo, la Universidad Católica aceptará una fotocopia de un título profesional siempre que un notario firme un mensaje que diga “Esta fotocopia de su título es válida, y se puede utilizar en la Universidad Católica”.

Suponga que se quiere digitalizar lo explicado arriba bajo los siguientes supuestos:

- Existe un solo notario que tiene una llave secreta y una llave pública.
- Todo el mundo conoce la llave pública del notario.
- Cada entidad  $E$  tiene un string identificador  $ID_E$  conocido por todos. Todos estos identificadores tienen el mismo largo.

Ahora, cuando una entidad  $E$  pida a un cliente un documento firmado y autorizado por el notario, se usará el siguiente protocolo.

- (a) El cliente genera el documento  $d$  y encripta  $ID_E||d$  con la llave pública del notario (suponemos que el documento es también un string). El texto cifrado resultante, que llamaremos  $c$ , es enviado al notario.
- (b) El notario verifica que el documento es válido y se puede utilizar en la entidad correspondiente. De no ser así se termina el protocolo.
- (c) El notario genera una firma  $f$  para  $c$  con su llave privada y le envía  $f$  al cliente.
- (d) Finalmente, el cliente envía  $(c, f)$  a la entidad  $E$ .

Para tener esta pregunta correcta deberá responder correctamente a las siguientes tres preguntas: (1) ¿Por qué el protocolo no satisface lo esperado? (2) ¿Cómo podemos modificar el último paso para arreglarlo? (3) ¿Qué complejidad innecesaria tiene el protocolo?

**Solución:**

- (1) Porque la entidad  $E$  al recibir  $(c, f)$  no puede verificar que esta firma corresponde al documento que se necesita, puesto que  $c$  sólo se puede decriptar con la llave secreta del notario.
- (2) Basta con enviar  $(d, f)$  en lugar de  $(c, f)$ . Con esto, la entidad  $E$  podría primero encriptar  $ID_E||d$  con la llave pública del notario para obtener  $c$ , y luego verificar que  $f$  es una firma válida para  $c$ .
- (3) Para el objetivo que se busca no es necesario encriptar el documento. El notario podría directamente firmar  $ID_E||d$ .

**Corrección:** Se considerará correcta una solución que tenga una respuesta a la pregunta (1) que indique que  $E$  no puede validar el documento porque no puede decriptar  $c$ . Las respuestas a las otras dos preguntas deben simplemente satisfacer los requerimientos del protocolo, no es necesario que sean equivalentes a lo que se menciona arriba. Obviamente, la respuesta a (2) sólo debe modificar el paso (d).