Criptografía y Seguridad Computacional **2017-01**

Tarea 4: 19/05/2017

Entrega: 30/05/2017

Reglas

- Las tareas se pueden discutir de a 2 personas. Sin embargo, cada estudiante tiene la obligación de escribir su propia solución.
- Cualquier material no visto en clases utilizado para solucionar la tarea debe ser citado. Dicho esto, está completamente prohibido buscar soluciones a los problemas en la web.
- La entrega de la tarea se realiza mediante correo electrónico firmado a `gnmonsalve@uc.cl` y con copia a `fekrell@uc.cl` con el asunto `[IIC3253] Entrega Tarea 3` a las 23:59 pm del día de la entrega. Cada estudiante sólo puede asumir que su tarea ha sido correctamente entregada si recibe un correo electrónico de respuesta correctamente firmado por algún miembro del cuerpo docente.
- Usted debe entregar un archivo pdf generado en LaTeX. Se sugiere utilizar el template publicado en la página web del curso.
- Problemas 1,2,3 y 4 son *obligatorios*. Es decir, cada problema de esta tarea es un requisito para aprobar el curso.

Problema 1 (15pts)

Calcule a mano lo siguiente,

1. (3pts) Los últimos 2 dígitos de 7^{35688}. Hint: $\phi(\prod P_i^{e_i}) = \prod P_i^{e_i-1}(P_i-1)$, para P_i primo y $e_i \in \mathbb{N}$.
2. (4pts) $[233^{230000022} \mod 35]$
3. (4pts) $[46^{51} \mod 55]$ usando el Teorema del resto chino.

Problema 2 (10pts)

Sean $p, N \in \mathbb{Z}$ tales que $p \mid N$. Demuestre que para cualquier entero x, $[[x \mod N] \mod p] = x \mod p$, pero que no es necesariamente cierto que $[[x \mod p] \mod N] = x \mod N$.

Problema 3 (15pts)

Sean N, e tales que $\mathsf{mcd}(e, \phi(N)) = 1$ y asuma que existe un adversario probabilista $\mathcal{A}$ que corre en tiempo t tal que

$$\Pr[\mathcal{A}(e, N, x^e \mod N) = x] = 0{,}01$$

en donde x es uniforme en $\mathbb{Z}_N^*$. Construya un adversario probabilista $\mathcal{A}'$ que corra en tiempo polinomial en t y $||N|| = \log N$ tal que:

$$\Pr[\mathcal{A}'(e, N, x^e \mod N) = x] = 0{,}99$$

para todo x (es decir, x no es necesariamente elegido uniformemente, por lo que no hay ninguna garantía de que $\mathcal{A}$ retorne x con probabilidad 0.01).

Hint: dado $x^e \mod N$ para cualquier x en $\mathbb{Z}_N^*$, genere un elemento y^e tal que y sea uniformemente aleatorio en $\mathbb{Z}_N^*$.

Problema 4 (20pts)

En este problema construiremos un test aleatorio para determinar si un número es primo. Si el número es primo, entonces el test *siempre* retornará `primo`. En cambio, si el número es compuesto, el test retornará `compuesto` con alta probabilidad.

Sabemos que si N es primo, entonces $a^{N-1} = 1 \mod N$ para todo $a \in \mathbb{Z}_N^*$ ($\phi(N) = N - 1$). Un posible test sería entonces elegir a uniforme es $\mathbb{Z}_N^*$ y ver si $a^{N-1} \neq 1$ (en tal caso decimos que a es un *testigo* de que N es compuesto). Si $a^{N-1} \neq 1$ retornamos "compuesto", en otro caso repetimos. Si al cabo de varias repeticiones no hemos retornado compuesto, entonces retornamos "primo".

Sin embargo, el test no necesariamente funciona para todo N pués, lamentablemente, para infinitos números compuestos N no existe ningún testigo.

En este ejercicio construiremos un test mejor. El test se basa en encontrar un *testigo fuerte*. Tales testigos (definidos más adelante), a diferencia de los anteriores, son abundantes en $\mathbb{Z}_N^*$ para todo N (excepto potencias perfectas).

Utilzaremos los siguientes resultados para demostrar que los elementos que no son testigos fuertes son menos de la mitad de los elementos en $\mathbb{Z}_N^*$:

- Sea $\mathbb{G}$ es un grupo finito. Sea $\mathbb{H} \neq \emptyset$, tal que $\mathbb{H} \subseteq \mathbb{G}$ y todo $x, y \in \mathbb{H}$, $xy \in \mathbb{H}$. ($\mathbb{H}$ es un subgrupo de $\mathbb{G}$).
- Si $\mathbb{H}$ es un subgrupo estricto de $\mathbb{G}$, entonces $|\mathbb{H}| \leq |\mathbb{G}|/2$.

Sea N impar, entonces $N-1$ puede ser descrito como $2^r u$ con $u \in \mathbb{N}$ impar y $r \geq 1 \in \mathbb{N}$ (ej, $8 = 2^3 \cdot 1$, $10 = 2^1 \cdot 5$, $28 = 2^2 \cdot 7$, $2000 = 2^4 \cdot 125$). Notar que si para algún i, $a^{2^i u} = \pm 1$, entonces $a^{2^j u} = 1 \mod N$ para todo $j > i$ ($a^{2^j u} = (a^{u 2^i})^{2^{j-i}}$).

Definición 1. *Decimos que $a \in \mathbb{Z}_N^*$ es un* testigo fuerte *de que N es compuesto si $a^u \neq \pm 1 \mod N$ y para todo $i \in \{1, ..., r-1\}$, $a^{2^i u} \neq -1 \mod N$.*

1. Describa un algoritmo de tiempo polinomial que dado N impar retorne $r \geq 1$ y u impar, tal que $N - 1 = 2^r u$.
2. Demuestre que si N es primo, entonces los únicos elementos x tales que $x^2 = 1 \mod N$ son $\{-1, 1\}$ (raices cuadradas de $1 \mod N$). Hint: Si N es primo y $N \mid ab$ entonces $N \mid a$ o $N \mid b$.
3. Demuestre que si N es primo impar, entonces N no tiene testigos fuertes. Es decir, para todo a o bien $a^u = \pm 1$, o $a^{2^i u} = -1$ para algún $i \in 1, ..., r$. Hint: para a arbitrario en $\mathbb{Z}_N^*$, defina j como el mínimo entero tal que $a^{2^j u} = 1 \mod N$. Demuestre por inducción en j y use el resultado anterior.
4. Sea NT el conjunto que contiene los elementos de $\mathbb{Z}_N^*$ que *no son testigos fuertes*. Demuestre que $[-1 \mod N] \in \mathsf{NT}$. (Demostraremos luego que $|\mathsf{NT}| \leq |\mathbb{Z}_N^*|$)
5. Sea i el máximo entero tal que existe $a \in \mathsf{NT}$ tal que $a^{2^i u} = -1 \mod N$, y definamos $\mathsf{GNT} = \{a | a^{2^i u} = \pm 1\}$. Demuestre que $\mathsf{NT} \subseteq \mathsf{GNT}$.
6. Demuestre que GNT es un subgrupo de $\mathbb{Z}_N^*$.
7. Considere $N = N_1 \cdot N_2$ tal que $\mathsf{mcd}(N_1, N_2) = 1$. Sea $a \in \mathsf{GNT}$ tal que $a^{2^i u} = -1 \mod N$. Considere un elemento $b \in \mathbb{Z}_N^*$ que se puede describir como ($a \mod N_1, 1 \mod N_2$) utilizando el teorema del resto chino. Demuestre que b no pertenece a GNT. Hint: Use el teorema del resto chino sobre $-1 \mod N$.

8. Concluya que si N es compuesto impar y no es potencia potencia perfecta de un entero ($N \neq Y^i$, para todo $Y, i \in \mathbb{N}$), entonces $\mathbb{Z}_N^*$ tiene al menos $|\mathbb{Z}_N^*|/2$ testigos fuertes.

9. Asumiendo la existencia de un algoritmo PerfectPower que determina si N es potencia perfecta de un entero, construya un test de tiempo polinomial en el largo de N tal que si N es primo, el test siempre retorne `primo`, pero si N es compuesto, el test retorna `primo` con probabilidad negligible en el largo de N.

Problema 5 (Opcional)

Describa un algoritmo de tiempo polinomial para determinar si un número es potencia perfecta de un entero.