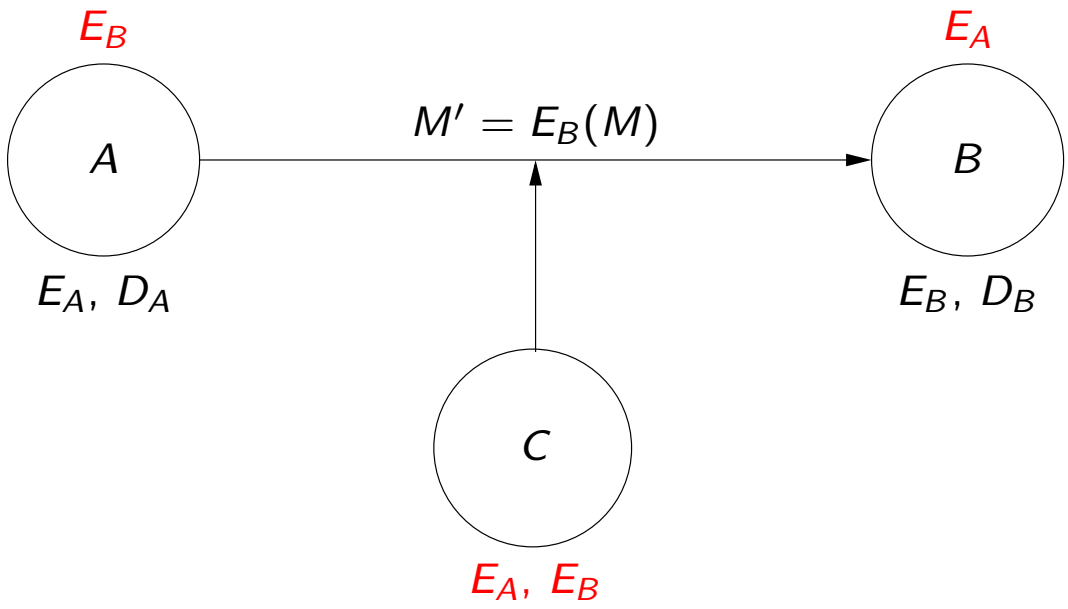
RSA: Autentificación y firma digitales

Escenario usual para RSA:



Dos preguntas a responder:

- Autentificación: ¿Cómo puede saber A si E_B es efectivamente la clave pública de B?
- Firma digital: ¿Cómo puede saber B si el mensaje M' fue efectivamente enviado por A?

RSA: Autentificación y firma digitales

Una propiedad fundamental de RSA: $E(D(M)) = M$

- Usamos esta propiedad para resolver los problemas de autentificación y firma digital

RSA: Autentificación y firma digitales

Una propiedad fundamental de RSA: $E(D(M)) = M$

- Usamos esta propiedad para resolver los problemas de autentificación y firma digital

Autentificación: Suponemos que existe una organización certificadora I

- Todos tienen acceso seguro a E_I

RSA: Autentificación y firma digitales

Una propiedad fundamental de RSA: $E(D(M)) = M$

- Usamos esta propiedad para resolver los problemas de autentificación y firma digital

Autentificación: Suponemos que existe una organización certificadora I

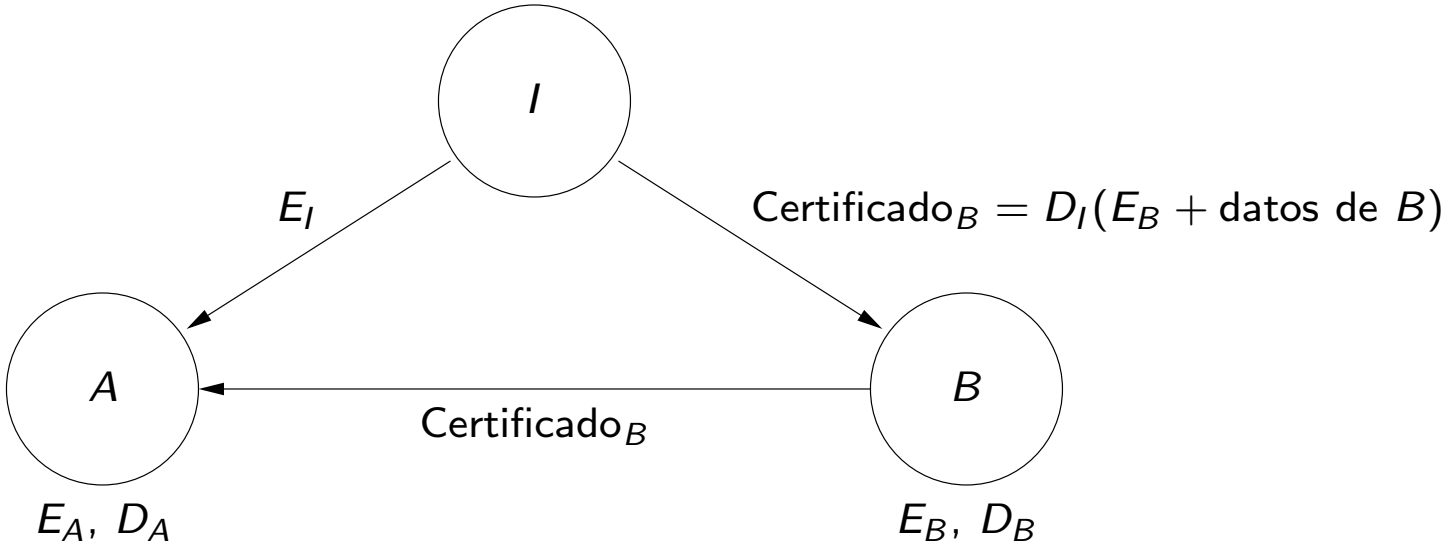
- Todos tienen acceso seguro a E_I

¿Cómo puede ser implementada la organización certificadora?

- ¿Por qué podemos suponer que tenemos acceso seguro a E_I?

RSA: Autentificación

Organización certificadora funciona de la siguiente forma:

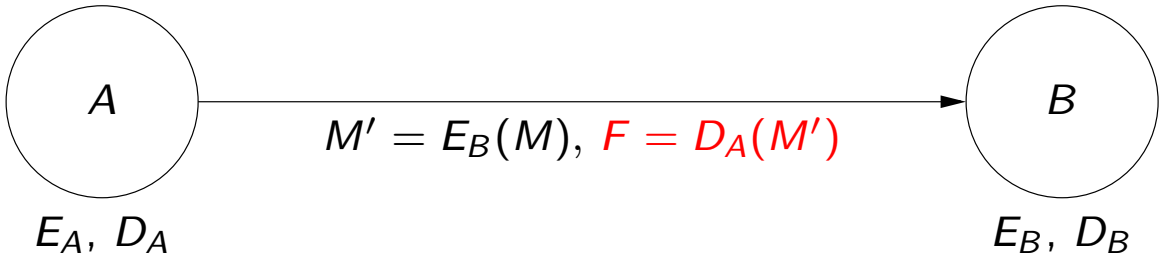


Tenemos entonces el siguiente protocolo:

- I entrega a B un certificado Certificado$_B$
- Para saber la clave pública de B (y los datos relevantes de B), A utiliza E_I(Certificado$_B$)

RSA: Firma digital

Envío de mensajes firmados: A envía tanto el mensaje M' como una firma F:



B recibe M' y F:

- Verificación: ¿Es cierto que $M' = E_A(F)$?
- Mensaje a leer: $D_B(M')$

RSA: Implementación

Para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

Primero vamos a resolver (2).

Máximo común divisor

Sea MCD(a, b) el máximo común divisor de los números a y b

- ¿Cómo podemos calcular MCD(a, b)?

Proposición

Si $b > 0$, *entonces* $MCD(a, b) = MCD(b, a \bmod b)$

Máximo común divisor

Sea MCD(a, b) el máximo común divisor de los números a y b

- ¿Cómo podemos calcular MCD(a, b)?

Proposición

Si $b > 0$, *entonces* $MCD(a, b) = MCD(b, a \bmod b)$

Demostración: Vamos a demostrar que un número c divide a a y b si y sólo si c divide a b y $a \bmod b$.

- De esto se concluye que MCD$(a, b) =$ MCD$(b, a \bmod b)$

Máximo común divisor

Sabemos que $a = \alpha \cdot b + (a \bmod b)$

($\Rightarrow$) Suponga que $c|a$ y $c|b$.

Dado que $(a \bmod b) = a - \alpha \cdot b$, concluimos que $c|(a \bmod b)$.

($\Leftarrow$) Suponga que $c|b$ y $c|(a \bmod b)$.

Dado que $a = \alpha \cdot b + (a \bmod b)$, tenemos que $c|a$. $\square$

Cálculo de máximo común divisor

De lo anterior, concluimos la siguiente identidad:

$$\text{MCD}(a, b) = \begin{cases} a & b = 0 \\ \text{MCD}(b, a \bmod b) & b > 0 \end{cases}$$

Podemos usar esta identidad para generar un algoritmos recursivo para calcular el máximo común divisor.

- Este algoritmo puede ser extendido para calcular s y t tales que $\text{MCD}(a, b) = s \cdot a + t \cdot b$
- Vamos a usar este algoritmo para calcular los coeficiente d y e usados en RSA.

Algoritmo extendido para calcular el máximo común divisor

Suponga que $a \geq b$, y defina la siguiente sucesión:

$$\begin{aligned} r_0 &= a \\ r_1 &= b \\ r_{i+1} &= r_{i-1} \bmod r_i \qquad (i \geq 2) \end{aligned}$$

Algoritmo extendido para calcular el máximo común divisor

Suponga que $a \geq b$, y defina la siguiente sucesión:

$$\begin{aligned} r_0 &= a \\ r_1 &= b \\ r_{i+1} &= r_{i-1} \bmod r_i \qquad (i \geq 2) \end{aligned}$$

Calculamos esta sucesión hasta un número k tal que $r_k = 0$

- Tenemos que $\text{MCD}(a, b) = r_{k-1}$

Algoritmo extendido para calcular el máximo común divisor

Al mismo tiempo calculamos sucesiones s_i, t_i tales que:

$$r_i \quad = \quad s_i \cdot a + t_i \cdot b$$

Tenemos que: $\text{MCD}(a, b) = r_{k-1} = s_{k-1} \cdot a + t_{k-1} \cdot b$

Algoritmo extendido para calcular el máximo común divisor

Al mismo tiempo calculamos sucesiones s_i, t_i tales que:

$$r_i = s_i \cdot a + t_i \cdot b$$

Tenemos que: $\text{MCD}(a, b) = r_{k-1} = s_{k-1} \cdot a + t_{k-1} \cdot b$

Sean:

$$s_0 = 1 \qquad t_0 = 0$$
$$s_1 = 0 \qquad t_1 = 1$$

Algoritmo extendido para calcular el máximo común divisor

Al mismo tiempo calculamos sucesiones s_i, t_i tales que:

$$r_i = s_i \cdot a + t_i \cdot b$$

Tenemos que: $\text{MCD}(a, b) = r_{k-1} = s_{k-1} \cdot a + t_{k-1} \cdot b$

Sean:

$$s_0 = 1 \qquad t_0 = 0$$
$$s_1 = 0 \qquad t_1 = 1$$

Se tiene que:

$$r_0 = s_0 \cdot a + t_0 \cdot b$$
$$r_1 = s_1 \cdot a + t_1 \cdot b$$

Algoritmo extendido para calcular el máximo común divisor

Dado que $r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i-1} \bmod r_i$, tenemos que:

$$r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i+1}$$

Por lo tanto:

$$s_{i-1} \cdot a + t_{i-1} \cdot b = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot (s_i \cdot a + t_i \cdot b) + r_{i+1}$$

Concluimos que:

$$r_{i+1} = (s_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot s_i) \cdot a + (t_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot t_i) \cdot b$$

Algoritmo extendido para calcular el máximo común divisor

Dado que $r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i-1} \bmod r_i$, tenemos que:

$$r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i+1}$$

Por lo tanto:

$$s_{i-1} \cdot a + t_{i-1} \cdot b = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot (s_i \cdot a + t_i \cdot b) + r_{i+1}$$

Concluimos que:

$$r_{i+1} = (s_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot s_i) \cdot a + (t_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot t_i) \cdot b$$

Definimos entonces:

$$s_{i+1} = s_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot s_i$$

$$t_{i+1} = t_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot t_i$$

Algoritmo extendido para calcular el máximo común divisor

Ejemplo

Vamos a usar el algoritmo para $a = 60$ y $b = 13$.

Inicialmente:

$$\begin{array}{lll} r_0 = 60 & s_0 = 1 & t_0 = 0 \\ r_1 = 13 & s_1 = 0 & t_1 = 1 \end{array}$$

Entonces tenemos que:

$$\begin{aligned} r_2 &= r_0 \bmod r_1 \\ s_2 &= s_0 - \lfloor \frac{r_0}{r_1} \rfloor \cdot s_1 \\ t_2 &= t_0 - \lfloor \frac{r_0}{r_1} \rfloor \cdot t_1 \end{aligned}$$

Algoritmo extendido para calcular el máximo común divisor

Example (Continuación)

Por lo tanto:

$$r_2 = 8 \qquad s_2 = 1 \qquad t_2 = -4$$

Y el proceso continua:

$$\begin{array}{lll} r_3 = 5 & s_3 = -1 & t_3 = 5 \\ r_4 = 3 & s_4 = 2 & t_4 = -9 \\ r_5 = 2 & s_5 = -3 & t_5 = 14 \\ r_6 = 1 & s_6 = 5 & t_6 = -23 \\ r_7 = 0 & s_7 = -13 & t_7 = 60 \end{array}$$

Tenemos que: $1 = 5 \cdot 60 + (-23) \cdot 13$

Inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \bmod n$

Inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \, mod \, n$

Para el caso de RSA: d es inverso de e en módulo $\phi(N)$

- En el ejemplo: 37 es inverso de 13 en módulo 60

Inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \, mod \, n$

Para el caso de RSA: d es inverso de e en módulo $\phi(N)$

- En el ejemplo: 37 es inverso de 13 en módulo 60

¿Todo número tiene inverso modular?

- No: 2 no tiene inverso en módulo 4

Inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \, mod \, n$

Para el caso de RSA: d es inverso de e en módulo $\phi(N)$

- En el ejemplo: 37 es inverso de 13 en módulo 60

¿Todo número tiene inverso modular?

- No: 2 no tiene inverso en módulo 4

¿Bajo qué condiciones a tiene inverso en módulo n?

- ¿Cómo podemos calcular este inverso?

Inverso modular: Existencia

Teorema

a tiene inverso en módulo n si y sólo si $MCD(a, n) = 1$

Inverso modular: Existencia

Teorema

a tiene inverso en módulo n si y sólo si $MCD(a, n) = 1$

Demostración: ($\Rightarrow$) Suponga que b es inverso de a en módulo n

- Entonces: $a \cdot b \equiv 1 \bmod n$

Se deduce que $a \cdot b = \alpha \cdot n + 1$, por lo que $1 = a \cdot b - \alpha \cdot n$

Concluimos que si $c|a$ y $c|n$, entonces $c|1$

- Por lo tanto c debe ser igual a 1, de lo que concluimos que $\text{MCD}(a, n) = 1$

Inverso modular: Existencia

($\Leftarrow$) Suponga que $\text{MCD}(a, n) = 1$

Ejecutando el algoritmo extendido para el cálculo del máximo común divisor obtenemos s y t tales que:

$$1 \quad = \quad s \cdot n + t \cdot a$$

Por lo tanto: $a \cdot t \equiv 1 \bmod n$

- Concluimos que a tiene inverso en módulo n

Cálculo de exponentes e y d en RSA

Recuerde que en RSA: $N = P \cdot Q$ y $\phi(N) = (P-1) \cdot (Q-1)$

- Tenemos que generar e y d tales que $e \cdot d \equiv 1 \bmod \phi(N)$

Tenemos los ingredientes necesarios para generar e y d:

genere al azar un número e
while $\text{MCD}(e, \phi(N)) > 1$ **do**
    genere al azar un número e
calcule s y t tales que $1 = s \cdot \phi(N) + t \cdot e$
sea $d \in \{0, \ldots, \phi(N) - 1\}$ tal que $d \equiv t \bmod \phi(N)$
return (e, d)

RSA: Implementación (continuación)

Como mencionamos anteriormente, para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

RSA: Implementación (continuación)

Como mencionamos anteriormente, para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

Ya resolvimos (2), nos falta resolver (1) y (3).

RSA: Implementación (continuación)

Como mencionamos anteriormente, para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

Ya resolvimos (2), nos falta resolver (1) y (3).

- ¿Cómo se resuelve (3)? No es difícil hacer esto

RSA: Implementación (continuación)

Como mencionamos anteriormente, para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

Ya resolvimos (2), nos falta resolver (1) y (3).

- ¿Cómo se resuelve (3)? No es difícil hacer esto
- Lamentablemente (1) está fuera del alcance de este curso ...

Teorema de Wilson: Una aplicación de la materia

Teorema (Wilson)

Un número $n \geq 2$ es primo si y sólo si $(n-1)! \equiv (n-1) \bmod n$.

Ejemplos:

- $3! \bmod 4 = 6 \bmod 4 = 2$
- $6! \bmod 7 = 720 \bmod 7 = 6$
- $8! \bmod 9 = 40320 \bmod 9 = 0$

Teorema de Wilson: Una aplicación de la materia

Teorema (Wilson)

Un número $n \geq 2$ es primo si y sólo si $(n-1)! \equiv (n-1) \bmod n$.

Ejemplos:

- $3! \bmod 4 = 6 \bmod 4 = 2$
- $6! \bmod 7 = 720 \bmod 7 = 6$
- $8! \bmod 9 = 40320 \bmod 9 = 0$

¿Cómo se demuestra este teorema?

- La noción de inverso modular es la pieza clave

Teorema de Wilson: Demostración

($\Leftarrow$) Por contradicción, suponga que $n \geq 2$, $(n-1)! \equiv (n-1) \bmod n$ y n no es primo.

Como n no es primo, existe $a \in \{2, \ldots, n-1\}$ tal que $a|n$.

Tenemos que $(n-1)! = a \cdot (a-1)! \cdot \left(\prod_{i=a+1}^{n-1} i \right)$

- Sea $\alpha = (a-1)! \cdot \left(\prod_{i=a+1}^{n-1} i \right)$

Teorema de Wilson: Demostración

Por hipótesis, concluimos que:

$$a \cdot \alpha \equiv (n-1) \bmod n$$

Por lo tanto:

$$a^2 \cdot \alpha^2 \equiv 1 \bmod n$$

Concluimos que a^2 tiene inverso en módulo n.

- Obtenemos una contradicción ya que $\text{MCD}(a^2, n) \neq 1$

Teorema de Wilson: Demostración

($\Rightarrow$) Sea $n \geq 2$ un número primo.

Para demostrar que $(n-1)! \equiv (n-1) \bmod n$ necesitamos demostrar algunos resultados intermedios.

Lema

Si p es un número primo y $a \cdot b \equiv 0 \bmod p$, entonces $a \equiv 0 \bmod p$ ó $b \equiv 0 \bmod p$.

Teorema de Wilson: Demostración

($\Rightarrow$) Sea $n \geq 2$ un número primo.

Para demostrar que $(n-1)! \equiv (n-1) \bmod n$ necesitamos demostrar algunos resultados intermedios.

Lema

Si p es un número primo y $a \cdot b \equiv 0 \bmod p$, entonces $a \equiv 0 \bmod p$ ó $b \equiv 0 \bmod p$.

Ejercicio

Demuestre el lema.

Teorema de Wilson: Demostración

Corolario

Si p es un número primo y $a^2 \equiv 1 \bmod p$, entonces $a \equiv 1 \bmod p$ ó $a \equiv (p-1) \bmod p$.

Teorema de Wilson: Demostración

Corolario

Si p es un número primo y $a^2 \equiv 1 \bmod p$, entonces $a \equiv 1 \bmod p$ ó $a \equiv (p-1) \bmod p$.

Ejercicio

Demuestre el corolario.

Teorema de Wilson: Demostración

Corolario

Si p es un número primo, $a \cdot b \equiv 1 \bmod p$ y $a \cdot c \equiv 1 \bmod p$, entonces $b \equiv c \bmod p$

- *En particular, si $b, c \in \{1, \ldots, p-1\}$, entonces $b = c$*

Teorema de Wilson: Demostración

Corolario

Si p es un número primo, $a \cdot b \equiv 1 \bmod p$ y $a \cdot c \equiv 1 \bmod p$, entonces $b \equiv c \bmod p$

- *En particular, si $b, c \in \{1, \ldots, p-1\}$, entonces $b = c$*

Ejercicio

Demuestre el corolario.

Teorema de Wilson: Demostración

Tenemos los ingredientes necesarios para demostrar el teorema.

- Para $n = 2$ y $n = 3$ es cierto, supongamos que $n > 3$

Teorema de Wilson: Demostración

Tenemos los ingredientes necesarios para demostrar el teorema.

- Para $n = 2$ y $n = 3$ es cierto, supongamos que $n > 3$

Usando los resultados anteriores concluimos lo siguiente (¿por qué?):

$$\left(\prod_{i=2}^{n-2} i\right) \equiv 1 \bmod n$$

Teorema de Wilson: Demostración

Tenemos los ingredientes necesarios para demostrar el teorema.

- Para $n = 2$ y $n = 3$ es cierto, supongamos que $n > 3$

Usando los resultados anteriores concluimos lo siguiente (¿por qué?):

$$\left(\prod_{i=2}^{n-2} i\right) \equiv 1 \bmod n$$

Concluimos que $(n-1)! \equiv (n-1) \bmod n$. □