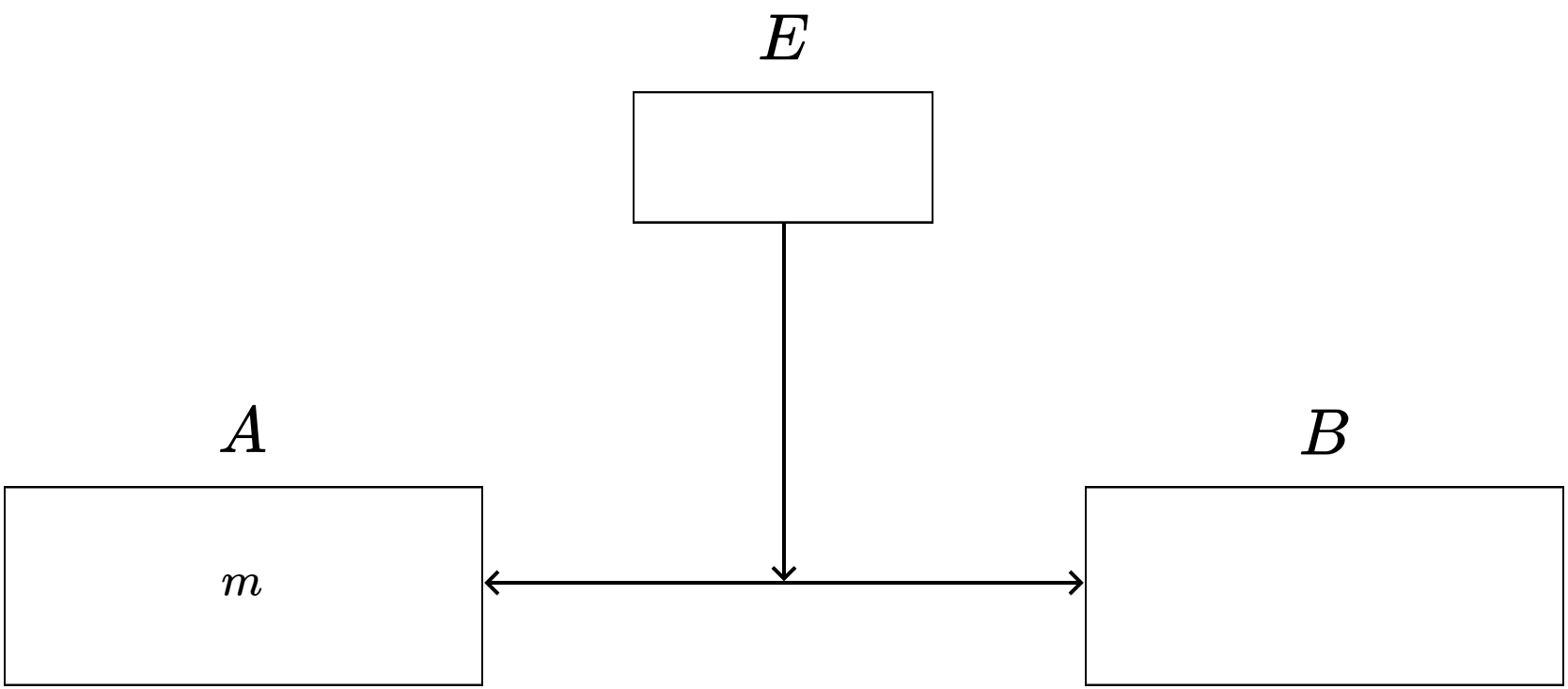
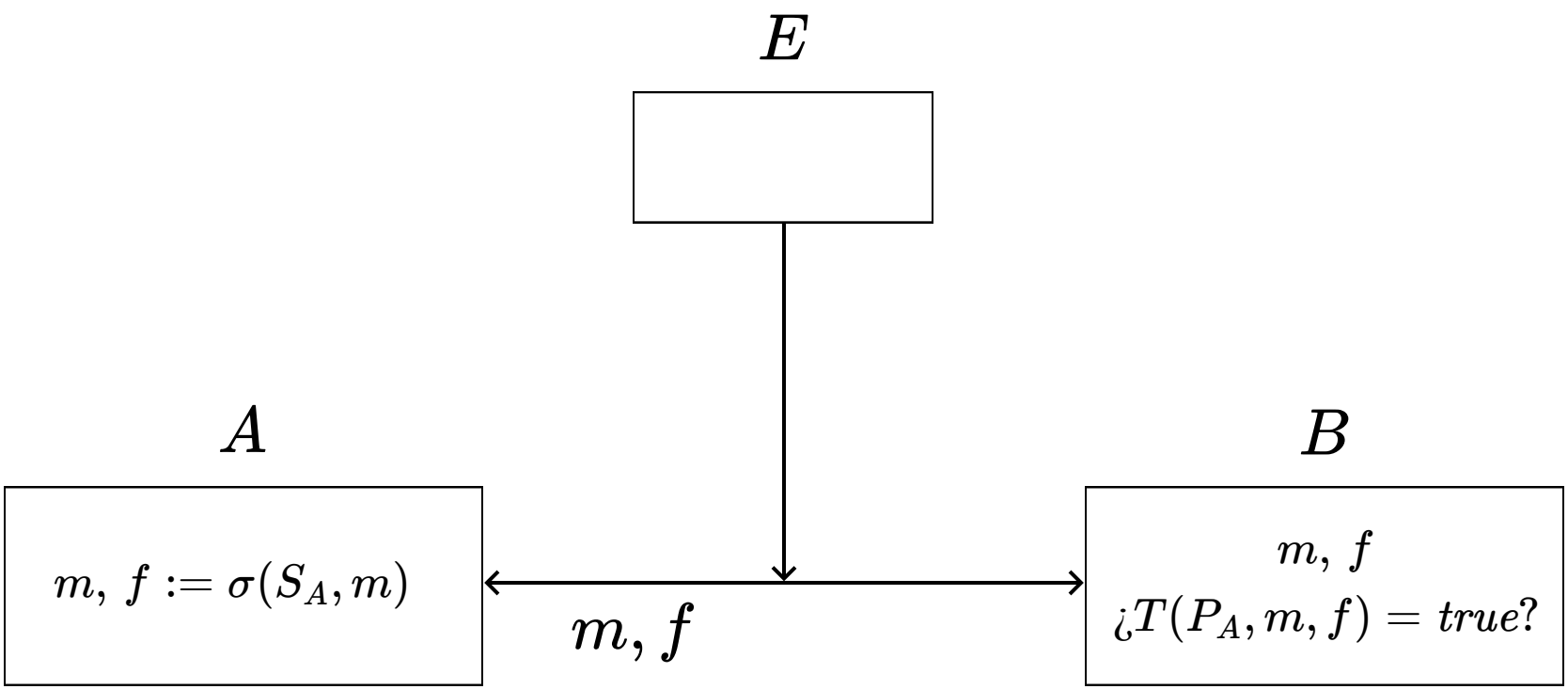
IIC3253

Firmas digitales

Firma digital con una clave pública



Firma digital con una clave pública



Firma digital con una clave pública

- A está firmando un mensaje m, para cualquiera que lo necesite
- $\sigma(S_A, m)$ utiliza la clave secreta de A para generar una firma f de m, de manera tal que solo A puede firmar
- $T(P_A, m, f)$ verifica si f es una firma válida del mensaje m por el usuario A
- $T(P_A, m, f)$ utiliza la clave pública de A, de manera que cualquiera puede verificar si f es una firma válida

Firmas digitales con RSA

Suponga que $P_A = (e, N)$ y $S_A = (d, N)$ son las claves pública y privada de un usuario A

Para cada $m \in \{0, \ldots, N-1\}$, sabemos que

$$Dec(S_A, Enc(P_A, m)) = m$$

Firmas digitales con RSA

Pero también tenemos que

$Enc(P_A, Dec(S_A, m)) =$

Firmas digitales con RSA

Pero también tenemos que

$$
\begin{aligned}
Enc(P_A, Dec(S_A, m)) &= (m^d \bmod N)^e \bmod N \\
&= (m^d)^e \bmod N \\
&= m^{d \cdot e} \bmod N \\
&= (m^e)^d \bmod N \\
&= (m^e \bmod N)^d \bmod N \\
&= Dec(S_A, Enc(P_A, m)) \\
&= m
\end{aligned}
$$

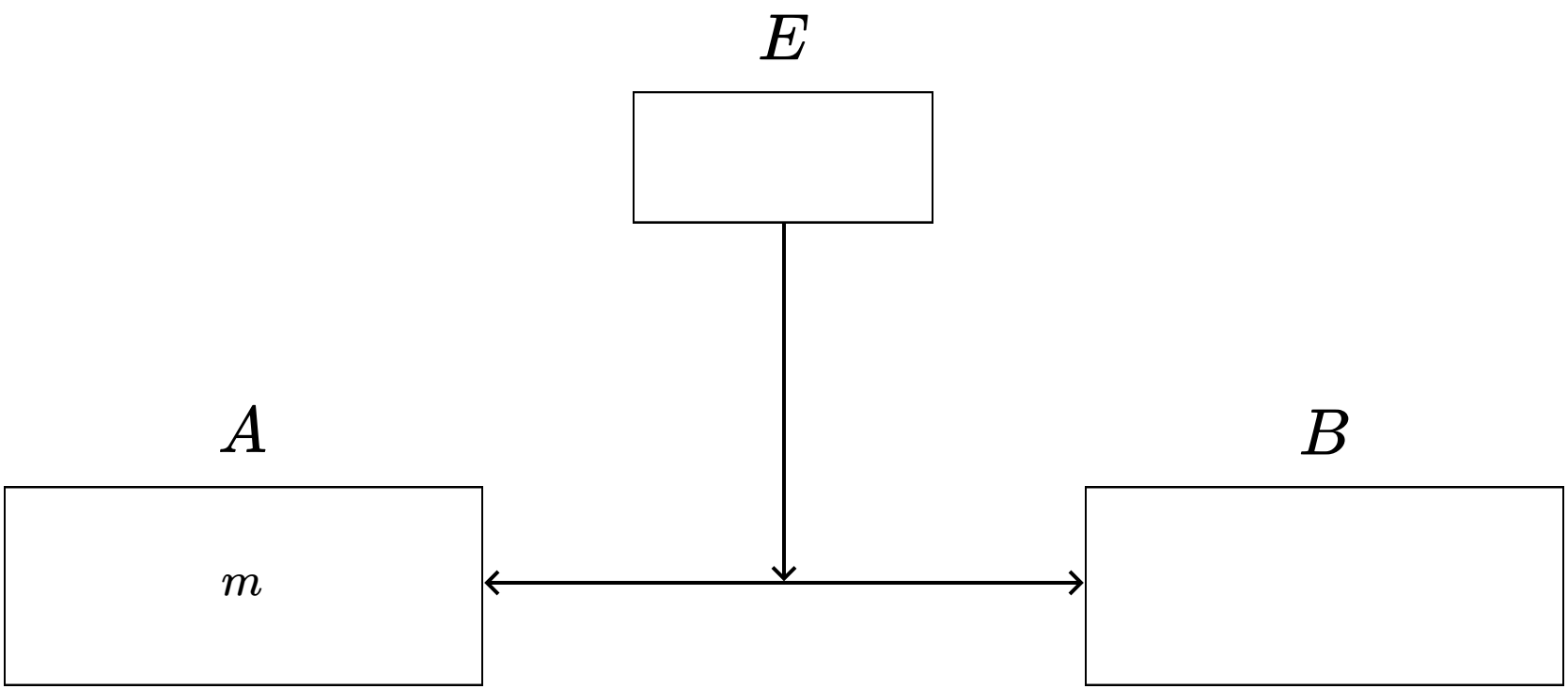
Firmas digitales con RSA

Definimos entonces la firma del mensaje m por el usuario A como:

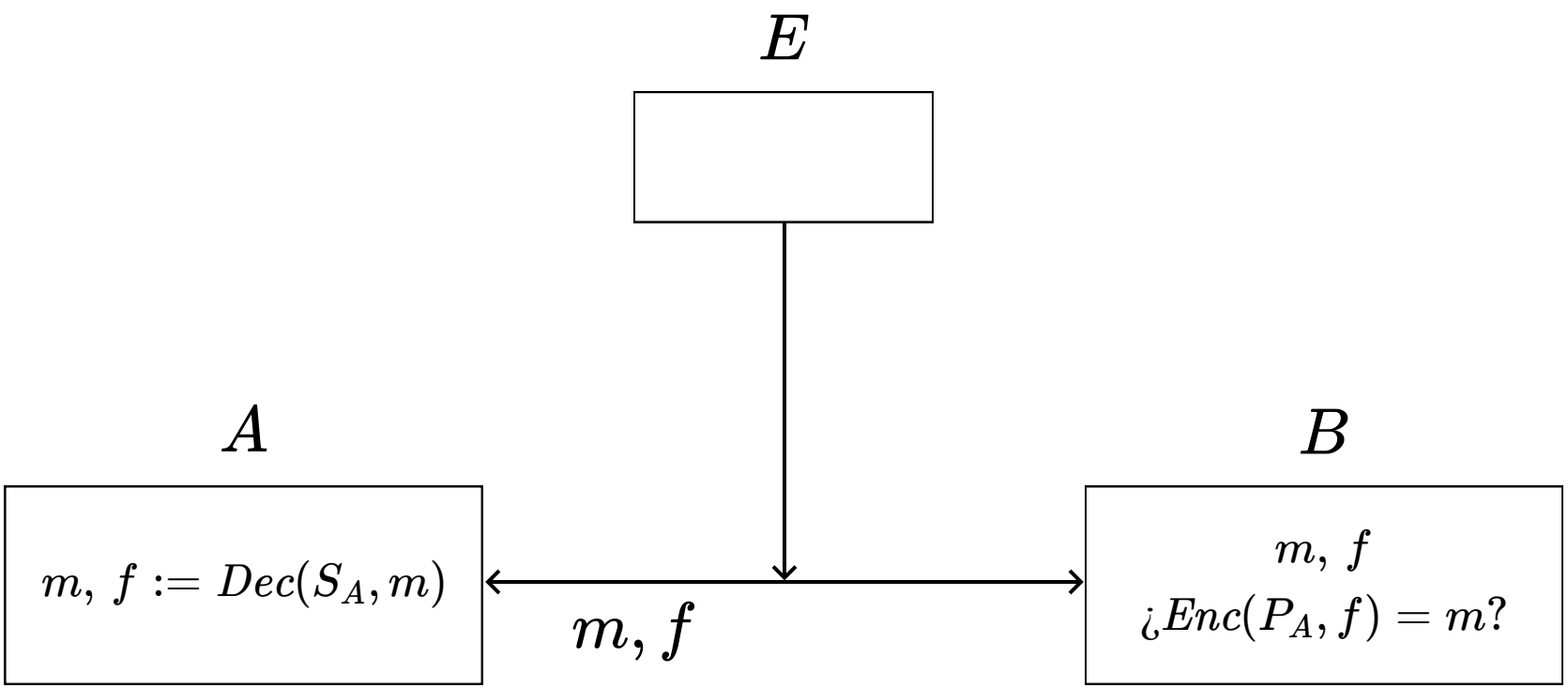
$$f := Dec(S_A, m)$$

Solo A puede generar esta firma. Cualquier usuario puede verificar si A firmó un mensaje usando la clave pública P_A de A

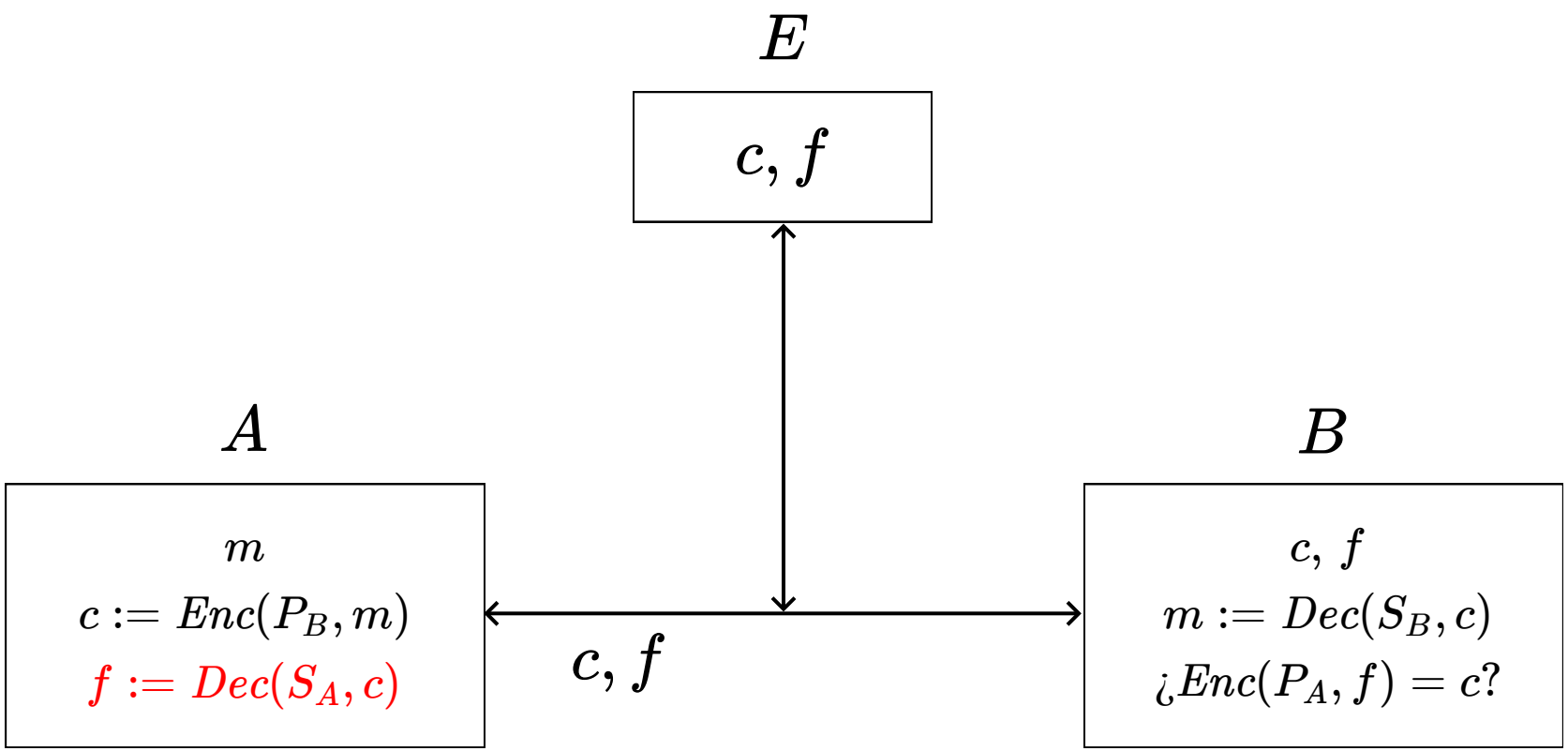
El esquema de firmas digitales con RSA



El esquema de firmas digitales con RSA



A puede firmar para B



¿Qué problema tiene el esquema anterior?

Firmar un mensaje m puede ser lento si m es un mensaje muy largo

Para solucionar este problema, se puede firmar $h(m)$ en lugar de firmar m, donde h es una función de hash

Firmas de Schnorr

Vamos a ver un segundo esquema para firmas digitales que está basado en el problema del logaritmo discreto

Se puede aplicar en cualquier grupo donde el problema de calcular el logaritmo discreto es difícil

La definición de las firmas de Schnorr

Suponemos dado un grupo finito $(G, *)$ y un elemento $g \in G$ tal que $|\langle g \rangle| = q$

- G, g y q son públicos
- Como vimos antes, se debe tener que $|G|$ y q son números grandes

Además suponemos dada una función de hash h

La definición de las firmas de Schnorr

La clave privada de un usuario A es $x \in \{1, \ldots, q-1\}$ y su clave pública es g^x

La definición de las firmas de Schnorr

La clave privada de un usuario A es $x \in \{1, \ldots, q-1\}$ y su clave pública es g^x

El usuario A quiere firmar un mensaje m

La definición de las firmas de Schnorr

A firma m de la siguiente forma:

1. Genera al azar $k \in \{1, \ldots, q-1\}$ y calcula $r = g^k$
2. Calcula $e = h(r \| m)$ usando r como un string
3. Calcula $s = k + e \cdot x$ interpretando e como un número natural
4. La firma de m es (e, s)

La verificación de una firma de Schnorr

Se puede verificar que (e, s) es una firma de m generada por A de la siguiente forma:

1. Calcule $\alpha = g^s = g^{k+e \cdot x} = g^k * g^{e \cdot x} = g^k * (g^x)^e$
2. Calcule $\beta = \alpha * ((g^x)^e)^{-1} = g^k * (g^x)^e * ((g^x)^e)^{-1} = g^k$
3. Verifique si $e = h(\beta \| m) = h(g^k \| m)$

¿Qué ventajas tienen las firmas de Schnorr?

Son más pequeñas que *otras* firmas digitales

Son fáciles de combinar si varios usuarios deben firmar un mensaje

¿Cómo pueden firmar un documento dos usuarios?

Suponga que A y B deben firmar un mensaje m

- Por ejemplo, un pago que debe ser autorizado por ambos usuarios

¿Cómo puede hacer esto usando RSA?

- ¿Es posible tener **una** clave pública para verificar que una firma es válida?

Resolviendo el problema con firmas de Schnorr

Suponemos que:

- La clave privada de A es x_1 y su clave pública es g^{x_1}
- La clave privada de B es x_2 y su clave pública es g^{x_2}

La clave pública para verificar la firma de m por ambos usuarios es $g^{x_1} * g^{x_2} = g^{x_1+x_2}$

Resolviendo el problema con firmas de Schnorr

A y B firman m de la siguiente forma:

1. A genera al azar $k_1 \in \{1, \ldots, q-1\}$ y calcula $r_1 = g^{k_1}$
2. B genera al azar $k_2 \in \{1, \ldots, q-1\}$ y calcula $r_2 = g^{k_2}$
3. *Ambos* calculan $e = h((r_1 * r_2) \| m)$ usando $r_1 * r_2$ como un string

Resolviendo el problema con firmas de Schnorr

4. A calcula $s_1 = k_1 + e \cdot x_1$ interpretando e como un número natural
5. B calcula $s_2 = k_2 + e \cdot x_2$ interpretando e como un número natural (de la misma forma que A)
6. Ambos calculan $s = s_1 + s_2$, y la firma de m es (e, s)

Resolviendo el problema con firmas de Schnorr

¿Cómo puede un usuario verificar que (e, s) es una firma de m generada por A y B?

¿Cómo se puede generalizar esta idea para n usuarios?