

EJERCICIO 1:

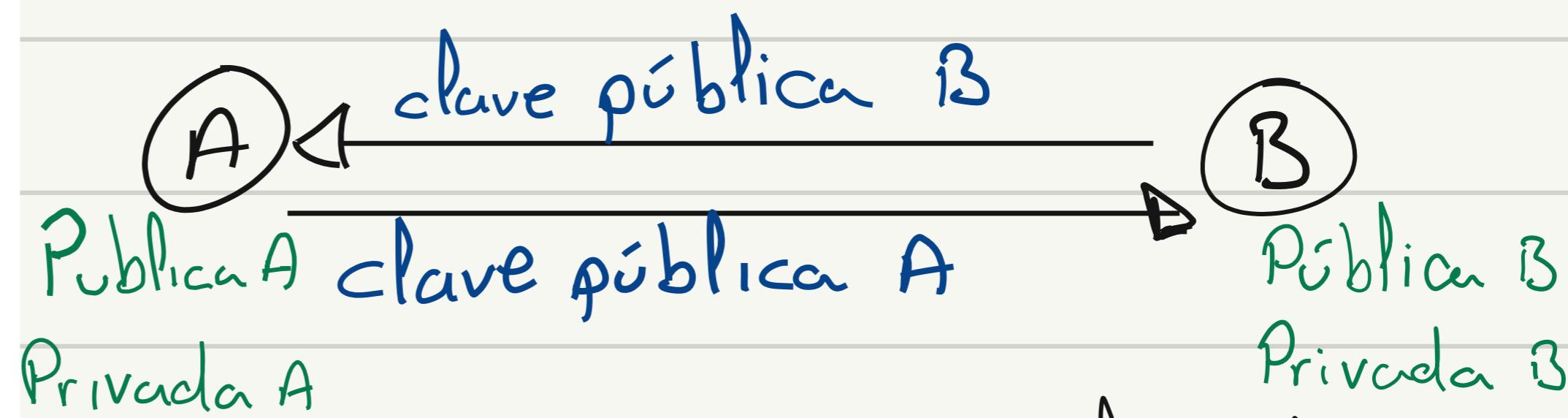
Definir el protocolo de comunicación entre Alice y Bob, de forma que Alice envíe un mensaje M a Bob de manera segura usando criptografía híbrida (es decir, que haya SOLO confidencialidad).

Precondición: Alice y Bob no han establecido comunicación previamente.



RSA

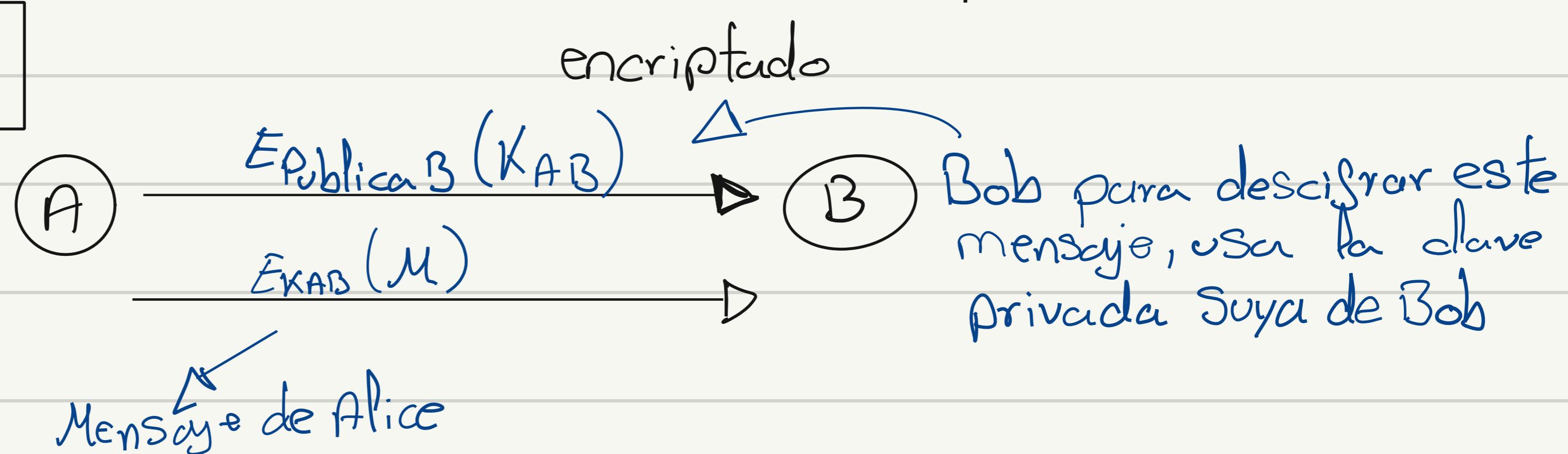
Tema 2



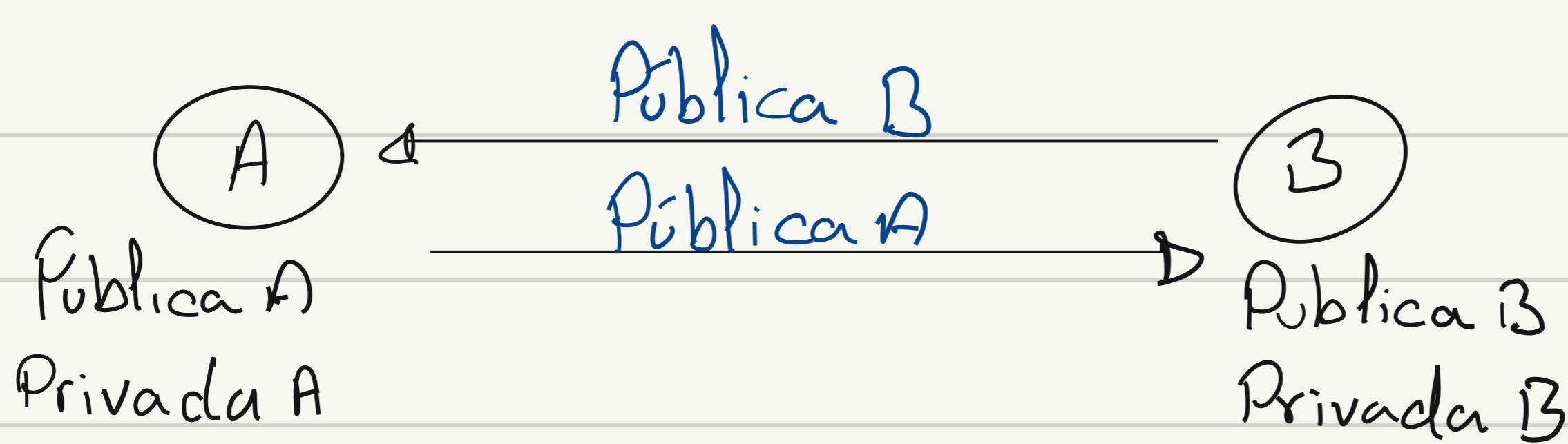
Se intercambian las claves públicas para saber de quién es el mensaje

1. Definir el problema asumiendo que se aplica RSA para el intercambio de claves.
2. Volver a definir el problema aplicando DH para el intercambio de claves.

K_{AB} = clave sesión



DH



Cada uno aplica algoritmo DH

$$K_{AB} = (Y_B)^{x_A} \mod q$$

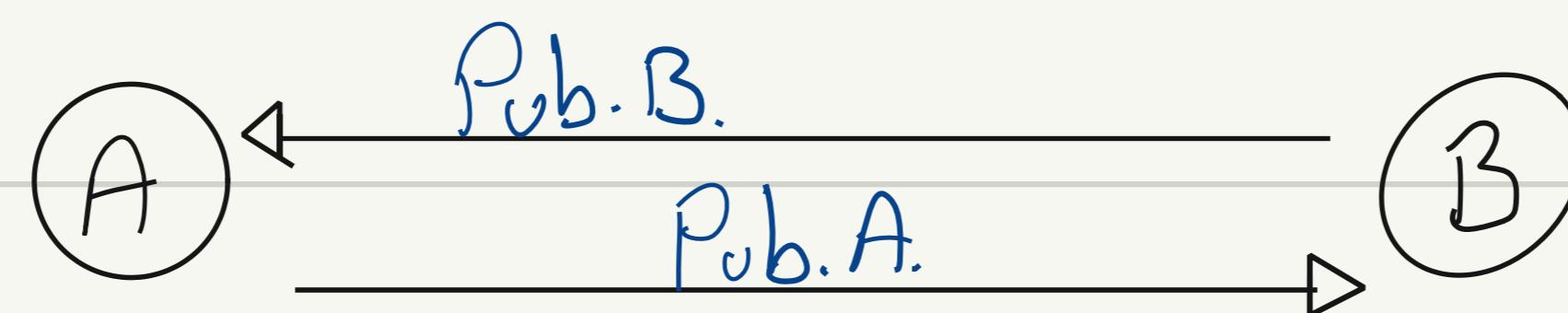
Clave privada A
Clave pública B
Valor común

$E_{KAB}(M)$

EJERCICIO 2:

Definir el protocolo de comunicación entre Alice y Bob, de forma que Alice envíe un mensaje M (de 2GB) a Bob de forma segura usando criptografía híbrida (es decir, que haya confidencialidad), pero esta vez aplicando firma digital para permitir a Bob verificar el origen real de los datos recibidos y su integridad.

Precondición: Alice y Bob no han establecido comunicación previamente.



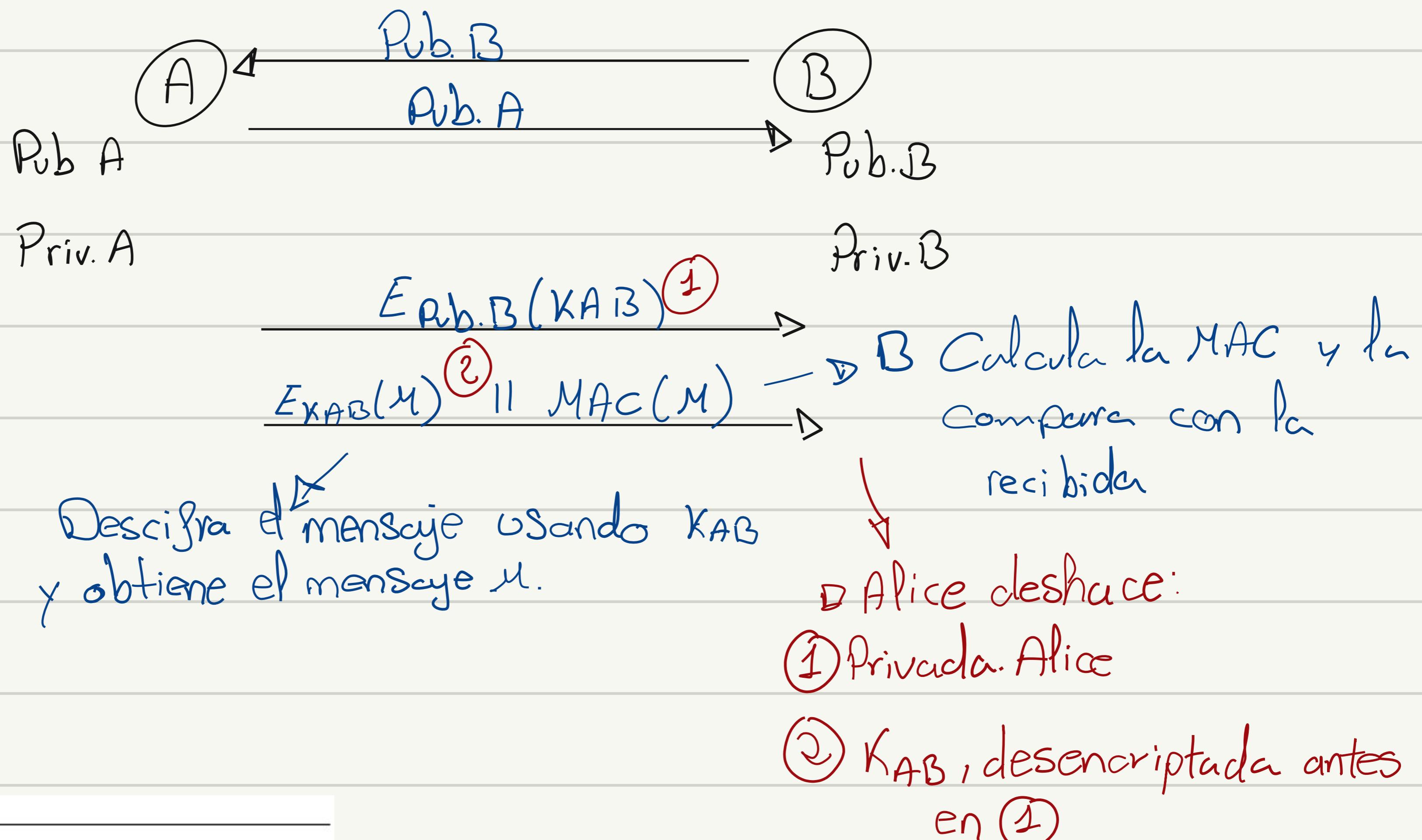
$E_{pub_B}(K_{AB})$

$E_{KAB}(M) \parallel E_{priv_A}(H(M))$

$E_{KAB}(E_{priv_A}(M)) ?$

EJERCICIO 3:

Definir el protocolo de comunicación entre Alice y Bob, de forma que Alice envíe un mensaje M (de 2GB) a Bob de forma segura (es decir, que haya confidencialidad), pero esta vez aplicando MAC para permitir a Bob a verificar el origen real de los datos recibidos y su integridad. Precondición: Alice y Bob no han establecido comunicación previamente.



EJERCICIO 4:

Teniendo en cuenta el siguiente protocolo:

1. $T \rightarrow A: E_{KAT}\{B, KAB, E_{KBT}\{C, KAB, KBC, E_{KCT}\{C, KBC\}\}}$ -- suponemos que A ya tiene la clave K_{AB} de una transacción previa
2. $A \rightarrow B: E_{KBT}\{C, KAB, KBC, E_{KCT}\{C, KBC\}\}$
3. $B \rightarrow C: E_{KCT}\{C, KBC\}$
4. $B \rightarrow A: E_{KAB}\{"holá ID=A"$
5. $B \rightarrow C: E_{KCT}\{C, KBC\}$
6. $C \rightarrow B: E_{KBC}\{"holá ID=B"$

y que T tiene compartido una clave secreta K_{CT} con C, K_{BT} con B y una clave K_{AT} con A, analizar el protocolo, contestando a las siguientes preguntas:

1. ¿Qué hace A cuando recibe el mensaje de T (punto 1 y punto 2)?
2. ¿Qué hace B cuando recibe el mensaje de A (punto 3, punto 4 y punto 5)?
3. ¿Qué tipo de criptografía se está aplicando en este protocolo?
4. ¿Hay autenticación en los puntos 2, 3, 4, 5 y 6? Razonar la respuesta.

① A descifra el mensaje y obtiene la clave de sesión K_{AB} y a quién le tiene que enviar el mensaje (B)

$$E_{KBT}\{C, K_{AB}, \dots\}$$

② B descifra el mensaje y tiene que enviar a "C" el resto del mensaje, además de obtener las claves de sesión de K_{AB} y K_{BC}

$$E_{KCT}\{C, K_{BC}\} \rightarrow \text{Envío a C}$$

$$E_{KAB}\{"holá ID=A"\} \rightarrow \text{Envío a A}$$

③ Criptografía Simétrica (solo se hace uso de claves de sesión).

④ Sí, la clave de sesión me garantiza autenticación, pero no es adecuado el procedimiento puesto que no garantiza integridad como hace la cript. asimétrica con la firma digital (usando claves privadas)

EJERCICIO 5:

Analizar el siguiente protocolo:
 1. $T \rightarrow A: T, A, E_{K_{pub}}\{H(K_{AB})\}, E_{priv}\{H(T, B, K_{AB})\}$
 2. $A \rightarrow B: A, B, E_{K_{AB}}\{"Estamos aprendiendo criptografía aplicada"\}$

contestando a las siguientes preguntas:

1. ¿Qué hace A cuando recibe el mensaje de T?
2. ¿Qué tipo de criptografía se está aplicando en este protocolo?
3. ¿Quién genera la clave de sesión?
4. ¿T se autentica a A? Razonar la respuesta.
5. ¿A se autentica con respecto a B? Razonar la respuesta.
6. ¿Crees que hay un coste de computación elevado en el punto 1?

① "A" descifra el mensaje y obtiene K_{AB} y verifica la firma de T.
 Luego le envía a "B" un mensaje cifrado con la clave K_{AB} .

- ② Híbrida y cuyo mensaje se ha cifrado con firma digital.
- ③ La clave es generada por "T".
- ④ Sí, con su firma digital $E_{priv,T}\{H(T, B, K_{AB})\}$
 • Y no solo se autentica, sino que hay integridad y se verifica que "T" es el remitente al 100%.
- ⑤ Sí, si recibe de "T" la clave K_{AB} .
- ⑥ Al usar cript. híbrida, los costes computacionales son relativamente bajos.

EJERCICIO 6:

Teniendo en cuenta el siguiente protocolo:

1. $A \rightarrow B: B, "Hola B", MAC(__)$
2. $B \rightarrow A: A, "Hola A", MAC(__)$

contestando a las siguientes preguntas:

1. ¿Es necesario que A y B tengan una clave de sesión previamente acordada?
2. ¿Qué hace B cuando recibe el mensaje de A (punto 1)? Razonar la respuesta.
3. ¿Qué hace A cuando recibe el mensaje de B (punto 2)? Razonar la respuesta.
4. ¿Qué servicio de seguridad se está aplicando?
5. Rediseña el protocolo para que, en vez de usar una MAC, usen firma digital.

Nota: "MAC" Siempre usa claves simétricas (claves de sesión)

① Sí, de lo contrario no se podría hacer la MAC.

② / ③ Comparan las MACs tras recibir los mensajes y verifican que sean iguales

por el hash

④ MAC proporciona integridad y, por tanto, autenticación.

⑤ $A \rightarrow B: B, "Hola B", K_{priv,A}\{H("Hola B")\}$

$B \rightarrow A: A, "Hola A", K_{priv,B}\{H("Hola A")\}$

EJERCICIO 7:

Teniendo en cuenta el siguiente protocolo:

1. $A \rightarrow B: B, "Hola B", MAC_{KAB}(B, "Hola B")$ – situación normal

Supongamos que un Man-in-the-Middle (MitM = Mallory) intercepta el canal de comunicación y modifica el mensaje "Hola B" justo en el punto 1, tal que:

1. $A \rightarrow Mallory \rightarrow B: B, "Adios B", MAC_{KAB}(B, "Hola B")$

Contestar a las siguientes preguntas:

1. ¿Qué hace B cuando recibe el mensaje de A (del protocolo alterado por Mallory)?
2. ¿Qué servicio de seguridad se está aplicando?
3. ¿Crees que el MitM pueda rehacer el MAC (del punto 2 y 3)? Razonar la respuesta.
4. ¿Cómo se resolvería el problema del MitM en el protocolo de arriba?

- ① Verifica el origen del mensaje y que la MAC no ha sido alterada.
- ② Integridad y autenticación (pero no es certificado por culpa del atacante).
- ③ No, si no sabe la clave de sesión K_{AB} .
- ④ Cifrando el mensaje y firmando digitalmente.

Tema 3

Teniendo en cuenta el siguiente protocolo:

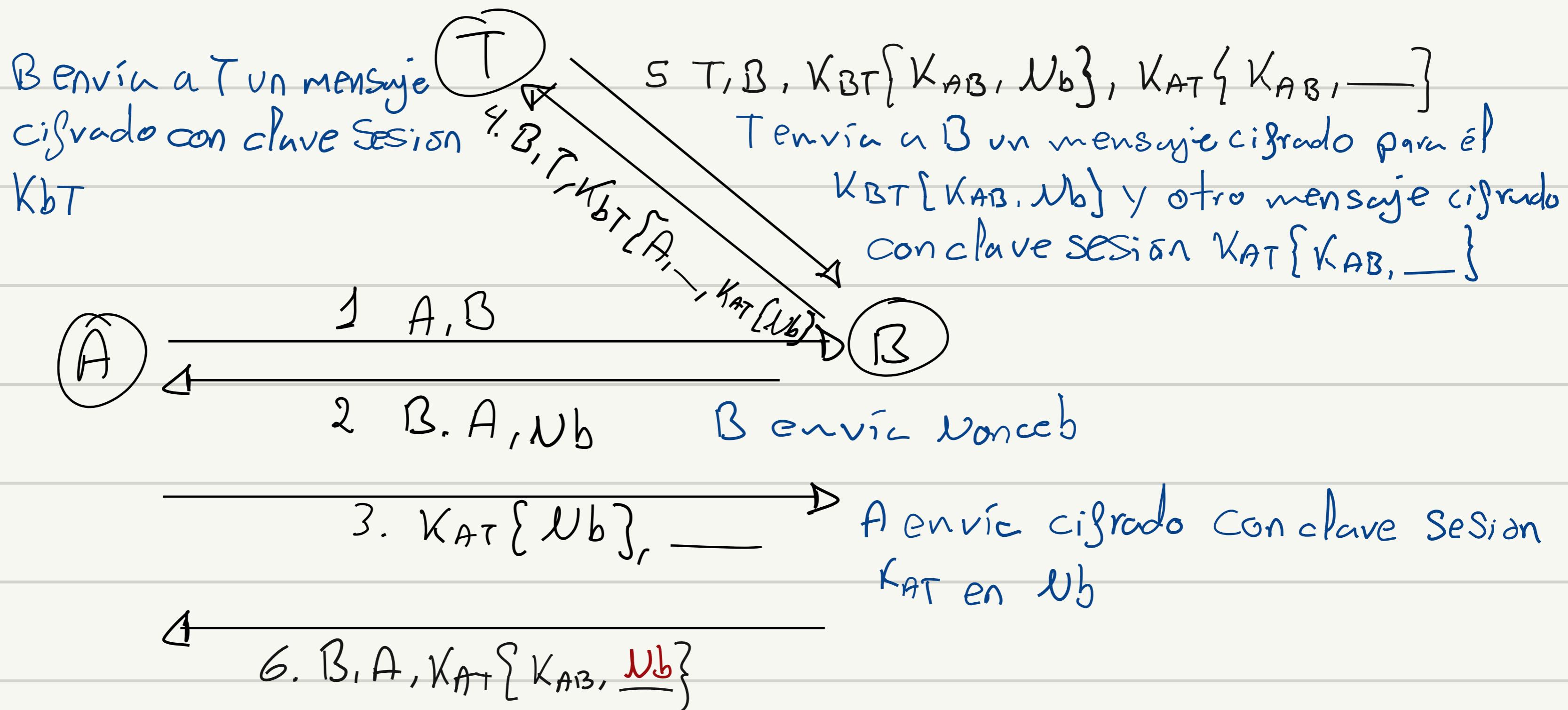
1. $A \rightarrow B : A, B$
2. $B \rightarrow A : B, A, N_b$
3. $A \rightarrow B : K_{AT}\{N_b\}, \dots$
4. $B \rightarrow T : B, T, K_{BT}\{A, \dots, K_{AT}\{N_b\}\}$
5. $T \rightarrow B : T, B, K_{BT}\{K_{AB}, N_b\}, K_{AT}\{K_{AB}, \dots\}$
6. $B \rightarrow A : B, A, K_{AT}\{K_{AB}, \dots\}$

y que Trent tiene compartido una clave secreta K_{BT} con B y una clave K_{AT} con A.

Analizar el protocolo, contestando a las siguientes preguntas:

1. ¿Qué significa cada parámetro del protocolo y para qué sirve?
2. ¿Qué problemas existen a la hora de trabajar con las claves K_{BT} y K_{AT} ?
3. Completar el protocolo anterior para evitar ataques de tipo *replay*.
4. ¿Crees que está completamente controlado todos los posibles ataques de reenvío en todo el protocolo? Es decir, ¿Bob controla el ataque? ¿Y Alice?
5. ¿Hay desafío y respuesta? Si es así, dónde.
6. ¿De qué tipo es el protocolo? ¿pull o push?

①



② El atacante puede interceptar el canal y capturar todos los mensajes y suplanta la identidad de Bob y Alice si consigue K_{BT} y K_{AT} respectivamente

③ Introducimos un "nonce" en la comunicación 6 para comprobar tras llegar el mensaje a Alice si el nonce que le llegó antes con el de ahora se mantienen iguales.

④ Sí, está controlado tanto en A como en B a través de N_b

⑤ No existe desafío - respuesta

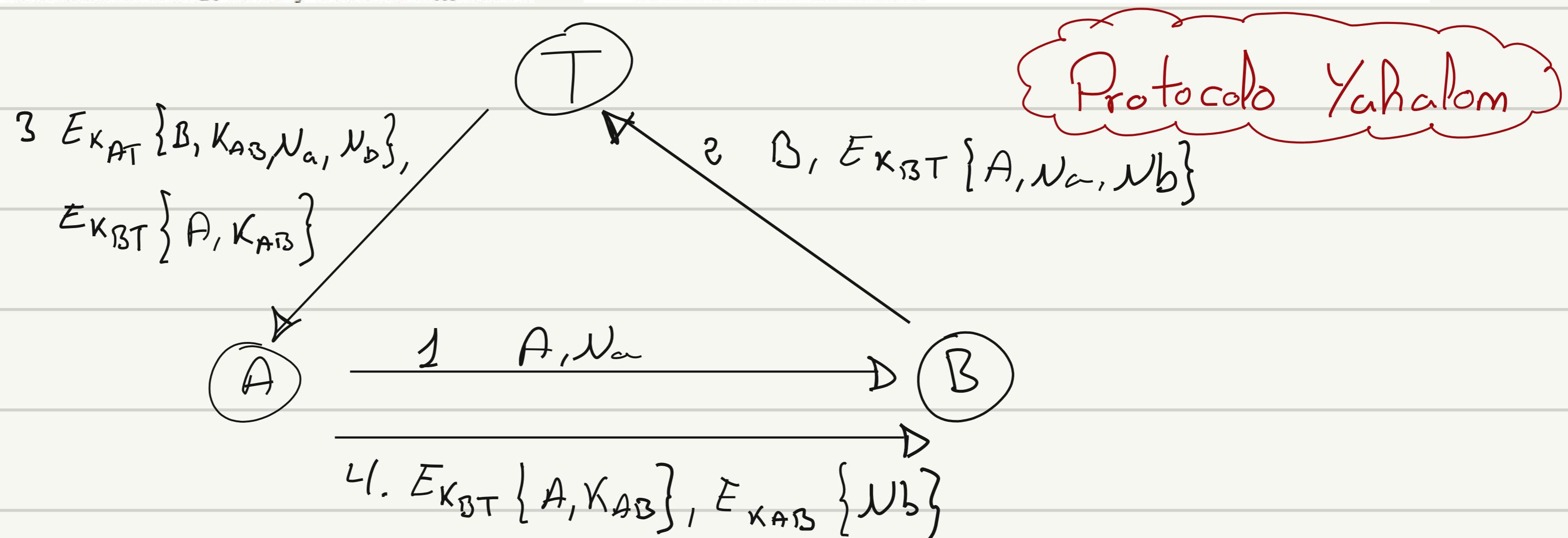
⑥ Push

②

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow T : B, E_{K_{BT}}\{A, N_a, N_b\}$
3. $T \rightarrow A : E_{K_{AT}}\{B, K_{AB}, N_a, N_b\}, E_{K_{BT}}\{A, K_{AB}\}$
4. $A \rightarrow B : E_{K_{BT}}\{A, K_{AB}\}, E_{K_{AB}}\{N_b\}$

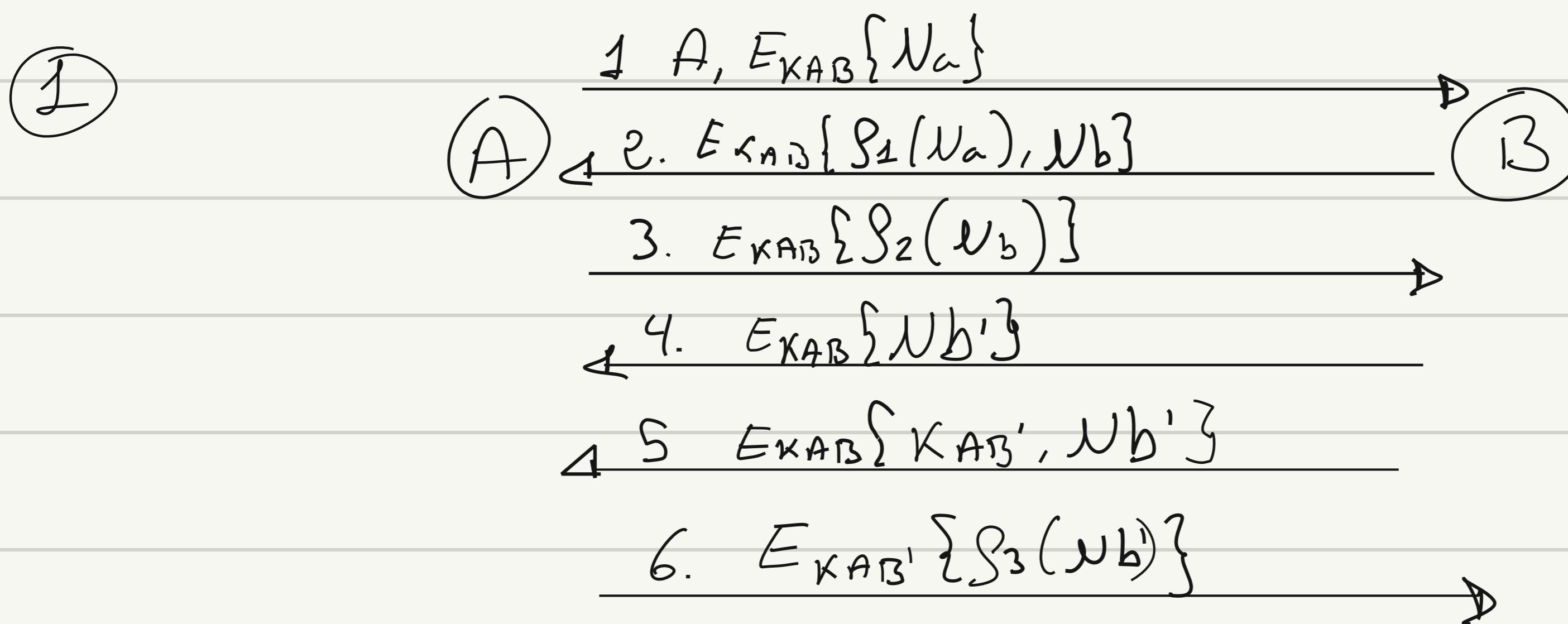
y que Trent tiene compartido una clave secreta K_{BT} con B y una clave K_{AT} con A.

1. ¿Quién inicializa el proceso de negociación de la clave de sesión K_{AB} ?
2. ¿Quién inicializa el proceso de negociación de la clave K_{AB} con T?
3. ¿Qué modelo sigue este protocolo (PULL o PUSH)?
4. ¿Existe posibilidad de ataques de repetición?
5. ¿Tiene Bob alguna forma de verificar que el mensaje viene de Alice y que está realmente hablando con ella?



- ① A (trama 4)
- ② B
- ③ Push (B contacta con T) → "Protocolo Yacelon"
- ④ No, porque tanto A como B reciben sus "nonces" y el del otro.
↳ Nota A en la trama 1 envía su "nonce" sin cifrar, arriesgándose a que alguien lo modifique.
- ⑤ Por parte sí, dado el nonce Nb de Alice.

- ③
- | | |
|--|---|
| 1. $A \rightarrow B: A, E_{KAB}\{N_a\}$ 2. $B \rightarrow A: E_{KAB}\{f_1(N_a), N_b\}$ 3. $A \rightarrow B: E_{KAB}\{f_2(N_b)\}$ 4. $A \rightarrow B: E_{KAB}\{f_3(N_b)\}$ 5. 6. $B \rightarrow A: E_{KAB}\{N_b\}$ 7. $B \rightarrow A: E_{KAB}\{K_{AB}, N_b\}$ | 1. ¿Se debe reordenar los mensajes para que tenga sentido el protocolo? En caso afirmativo, reordenarlo. 2. ¿Por qué se aplica f_1 , f_2 y f_3 ? 3. ¿Qué significa la clave K_{AB} ? ¿Por qué se envía en el paso 5 un N_b ? 4. ¿Qué sentido tendría usar un código MAC en este protocolo? |
|--|---|

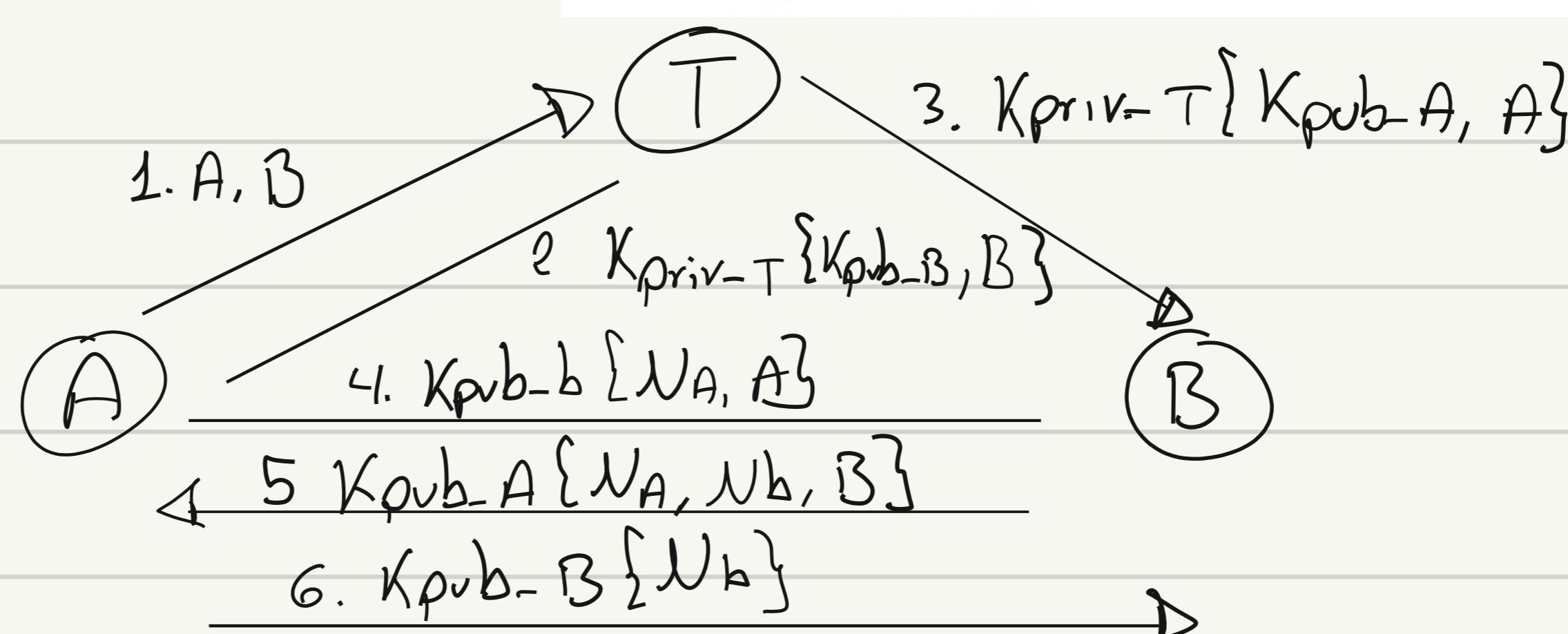


- ② S_1, S_2, S_3 corresponden a las respuestas de un desafío.
- ③ K_{AB}' es una nueva clave de sesión generada por B y se envía N_b' como parte del desafío-respuesta de S_3 .
- ④ Al pasar los mensajes cifrados por una función MAC, junto a la misma clave simétrica, ya no solo garantizamos la autenticidad del mensaje, sino que también su integridad.

- ④
- Supongamos ahora que en vez de aplicar criptografía simétrica para la gestión de claves K_{AB} , aplicamos sólo y únicamente criptografía asimétrica para:
- (i) gestionar las claves públicas de cada entidad A y B, y desde T, y
 - (ii) para buscar la forma de compartir un nonce (N_a y N_b) que les permitan a cada entidad (A y B) verificar el "freshness" de las transacciones.

- ✗ 1. A, B
- ✗ 2. $K_{pub_A}\{N_a, N_b, B\}$
- ✗ 3. $K_{priv_T}\{K_{pub_B}, B\}$
- ✗ 4. $K_{pub_B}\{N_b\}$
- ✗ 5. $K_{priv_T}\{K_{pub_A}, A\}$
- ✗ 6. $K_{pub_B}\{N_A, A\}$

1. Reorganizar los mensajes teniendo en cuenta la existencia de Trent (T) - como mediador entre A y B - , y la existencia de sus claves K_{priv_T} y K_{pub_T} . Para realizar el ejercicio, es fundamental identificar el origen y el destino de cada mensaje (ej: $A \rightarrow B$) y asumir que T (i) conoce las claves públicas de A y B y (ii) se encarga de enviarlas cuando ellos las solicitan.
2. ¿Qué hace realmente el protocolo? ¿Cuál es su objetivo final?
3. ¿Para qué sirve el N_A ?



② Simula los certificados digitales, donde T funcionaría como la CA.

③ Para controlar los ataques replay por el lado de A.

5

Teniendo en cuenta el siguiente protocolo:

1. Alice → Bob: Bob, M, $S_{Alice}(Bob, H(M))$
2. Bob → Alice: Alice, $S_{Bob}(Alice, H(M))$

1. Analizar el protocolo y determinar su principal objetivo.
2. ¿Crees que hay suficiente evidencia para asegurar el no repudio tanto en el lado del origen como en el destino?

① No-Repudio → Alice quiere dejar constancia a Bob que el mensaje firmado es de ella, enviándose sin encriptar también.

② Si:
 → Alice-Bob: (por lo dicho antes)
 → Bob-Alice: Sí, Bob acusa a Alice de no haberle llegado nada, Alice tiene evidencias de que Bob sabe "M", puesto que este le envió de vuelta a ella un mensaje cifrado S_{Bob} con el mensaje "M" firmado.

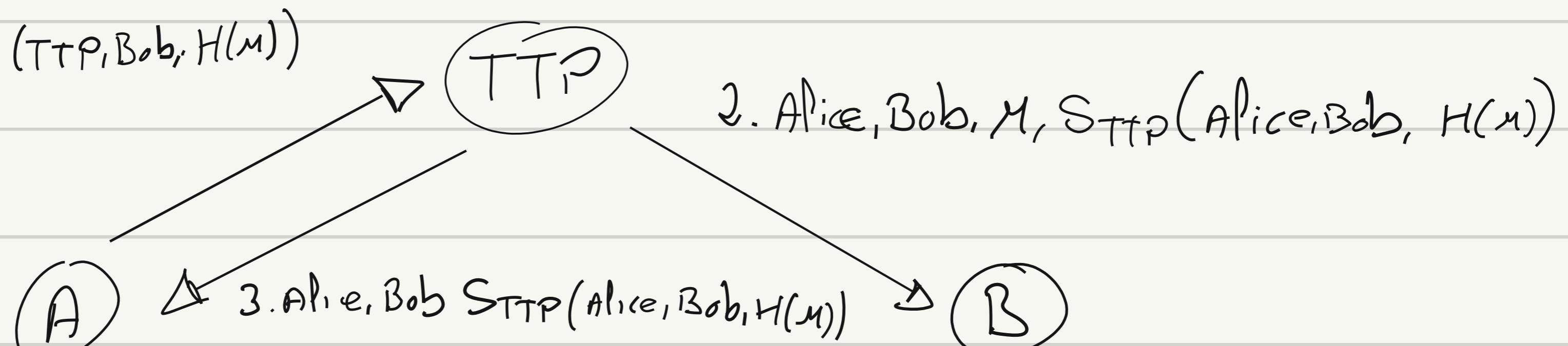
6

Si suponemos que el canal es fiable y las entidades NO son honestas, entonces se introduce una tercera persona confiable para este nuevo protocolo, tal que:

1. Alice → TTP: TTP, Bob, M, $S_{Alice}(TTP, Bob, H(M))$
2. TTP → Bob: Alice, Bob, M, $S_{TTP}(Alice, Bob, H(M))$
3. TTP → Alice: Alice, Bob, $S_{TTP}(Alice, Bob, H(M))$

1. Analizar el protocolo y determinar su principal objetivo.
2. ¿Crees que hay suficiente evidencia para asegurar el no repudio tanto en el lado del origen como en el destino? ¿Quién firma esas evidencias?

1. TTP, Bob, M, $S_{Alice}(TTP, Bob, H(M))$



2. Alice, Bob, M, $S_{TTP}(Alice, Bob, H(M))$

3. Alice, Bob, $S_{TTP}(Alice, Bob, H(M))$

① No-Repudio: Todos reciben los mensajes hasta sin encriptar.

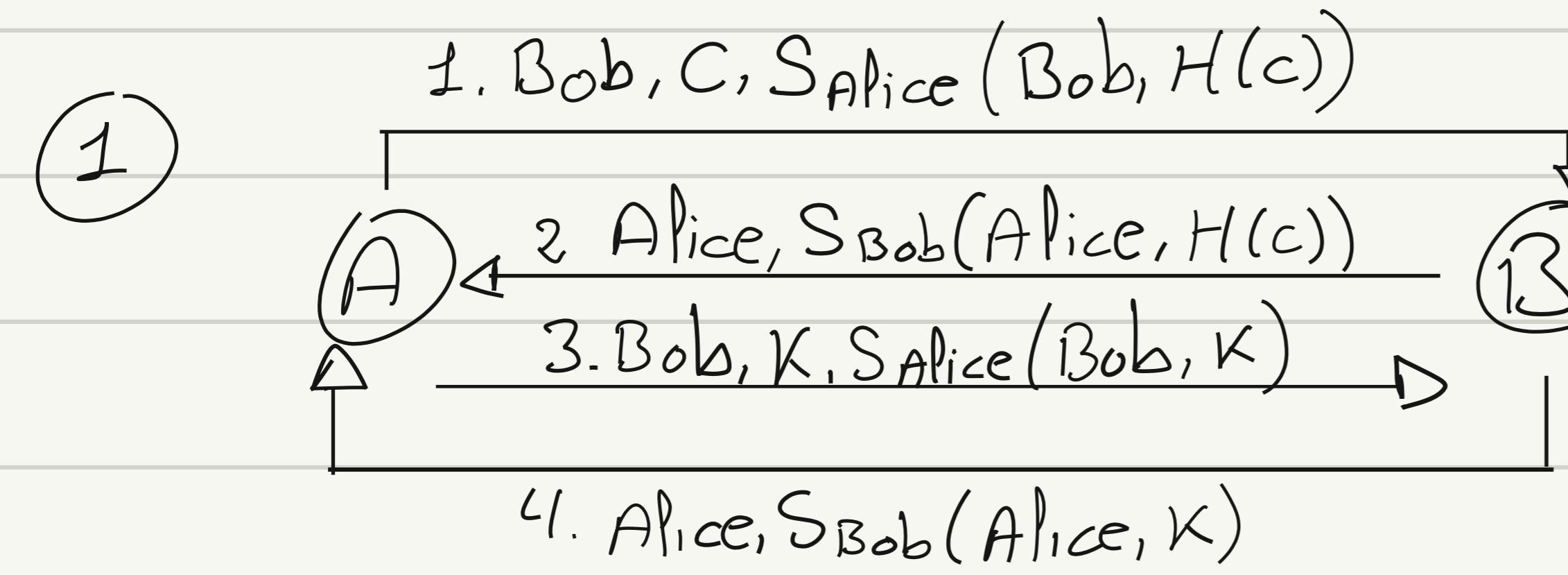
② No, a TTP le falta la prueba de recibo por parte de B.

7

Si suponemos que el canal NO es fiable y las entidades NO son honestas, pero no se introduce una tercera persona confiable para este nuevo protocolo, tal que:

1. Alice → Bob: Bob, C, $S_{Alice}(Bob, H(C))$
2. Bob → Alice: Alice, $S_{Bob}(Alice, H(C))$
3. Alice → Bob: Bob, K, $S_{Alice}(Bob, K)$
4. Bob → Alice: Alice, $S_{Bob}(Alice, K)$

1. Analizar el protocolo, y determinar qué es C en el punto 1 del protocolo y K en el punto 3 de dicho protocolo.
2. Determinar el principal objetivo de este protocolo, razonando la respuesta.
3. En el punto 3 y 4 no se realiza el $H(K)$ para la firma, es decir: $S_{Alice}(Bob, H(K))$ o $S_{Bob}(Alice, H(K))$. Explicar las razones.



• "C" es un criptograma
de un mensaje cifrado
por A.

• "K" es una clave de sesión
con la que se firmó el
mensaje original asociado
a C.

② No-Repudio

③ Seguramente porque la clave de

Sesión ya es pequeña

↳ Pero esto es una mala práctica!!

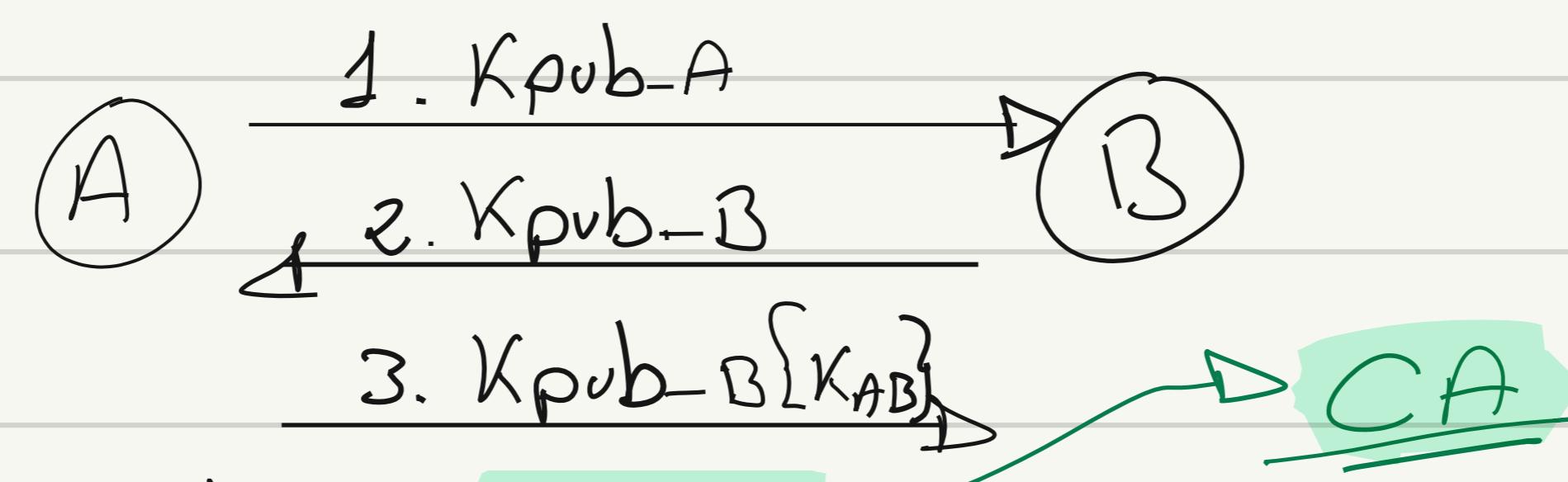
Se debe primero pasar el mensaje por una función
"Hash" y luego si, firmar.

8

Teniendo en cuenta el siguiente protocolo:

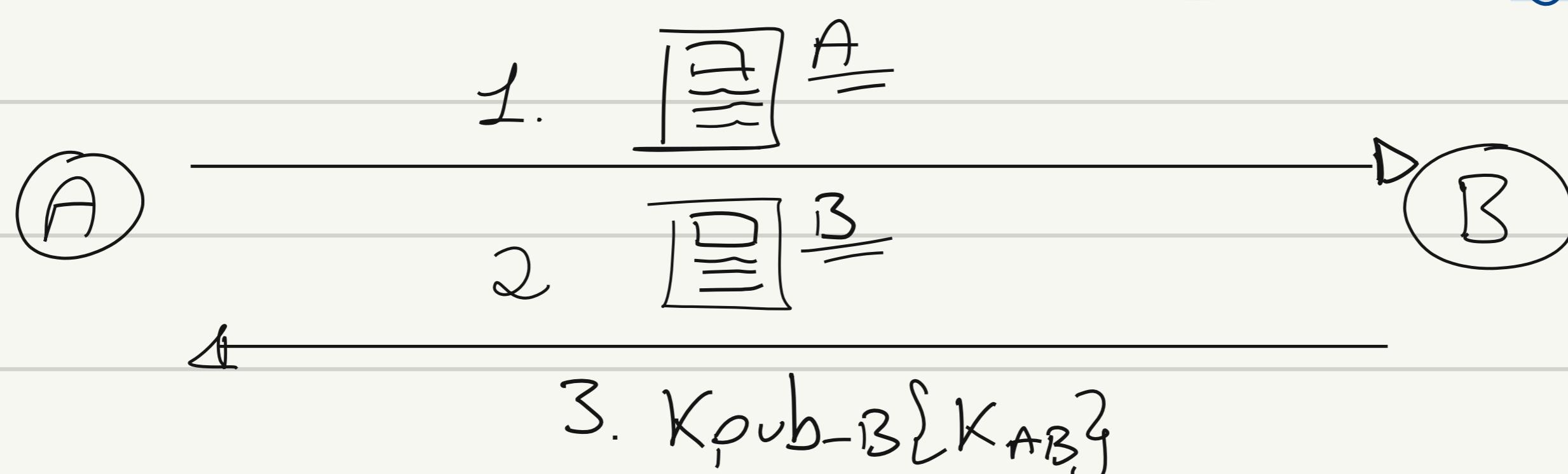
1. A → B: K_{pub_A}
2. B → A: K_{pub_B}
3. A → B: $K_{pub_B}\{K_{AB}\}$

1. Optimizar el protocolo para que Alice pueda verificar que la clave pública es genuina y pertenece a Bob.
2. ¿Qué hace Alice y Bob cuando recibe la clave pública asumiendo la modificación del punto 1? Establecer la secuencia de acciones que toma cada parte para la verificación.

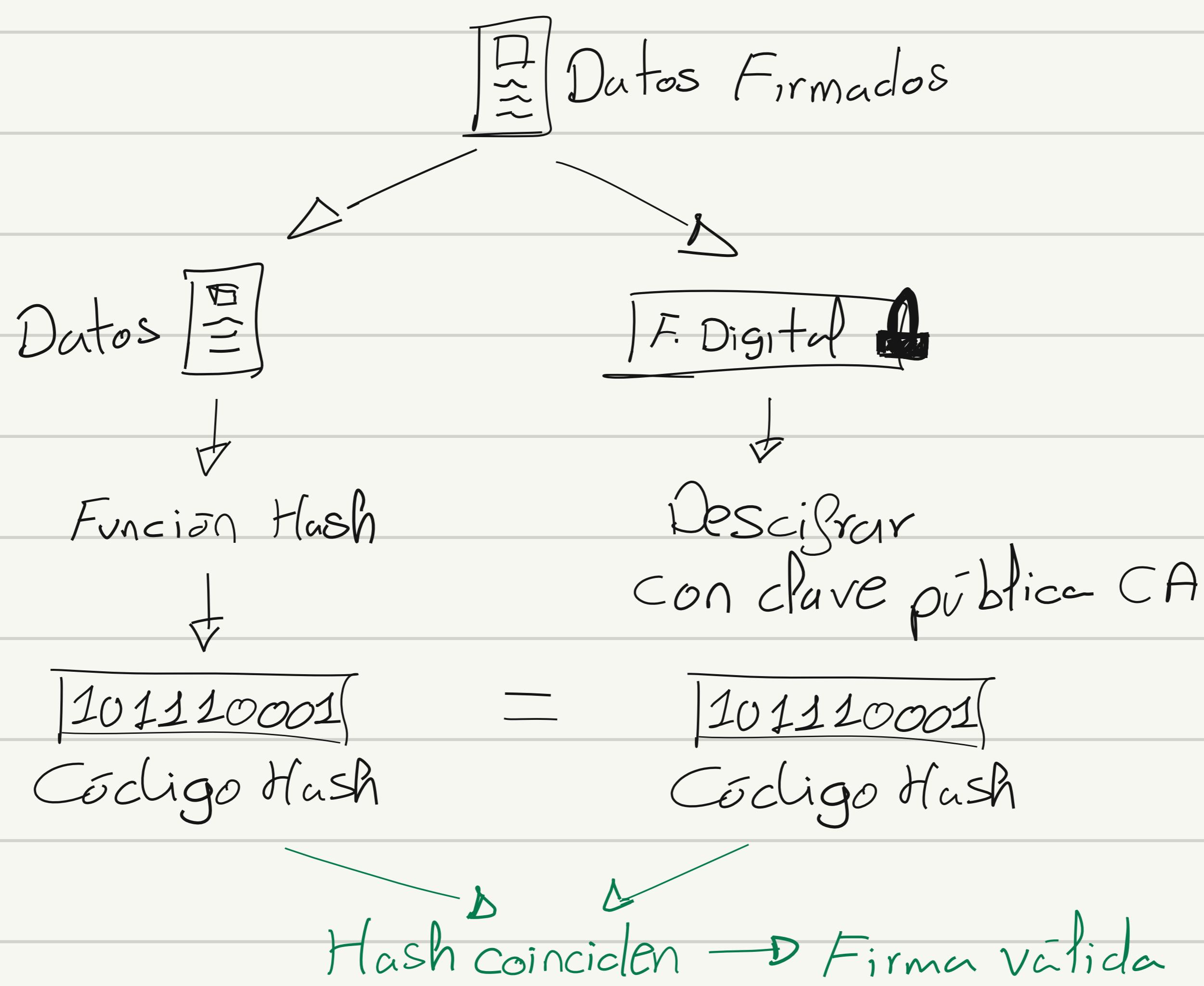


① Con la ayuda de un tercero podemos garantizar la
veracidad de las claves públicas

↳ Certificados Digitales



② Alice verifica la firma de la CA.



⑨

Teniendo en cuenta el siguiente protocolo:

1. $CA_1 \rightarrow A : Cert_{A_CA_1}$
2. $CA_2 \rightarrow B : Cert_{B_CA_2}$
3. $A \rightarrow B : Cert_{A_CA_1}$
4. $B \rightarrow A : Cert_{B_CA_2}$

y asumiendo que A conoce CA_1 y tiene su clave pública $K_{pub_CA_1}$ y B conoce CA_2 y tiene su clave pública $K_{pub_CA_2}$, se pide contestar a las siguientes preguntas:

1. ¿Puede A verificar el certificado $Cert_{B_CA_2}$ si $CA_1 \neq CA_2$? Razonar la respuesta.
2. ¿Existe alguna forma de que A y B puedan compartir las claves de forma segura?

① Sí, siempre que A pueda obtener la clave pública de CA_2
- Pero si previamente A no conoce la clave, Seguramente
descongrie del certificado

② Se pueden pasar las claves como quieran.

↳ La mejor manera es que ambos certificados fueran
certificados por un tercero.

