

SEGURIDAD DE LA INFORMACIÓN

TEMA 3 - PARTE 1

ESQUEMAS, PROTOCOLOS Y MECANISMOS DE SOPORTE

(A LA SEGURIDAD DE APLICACIONES Y DE REDES)

Indice del tema

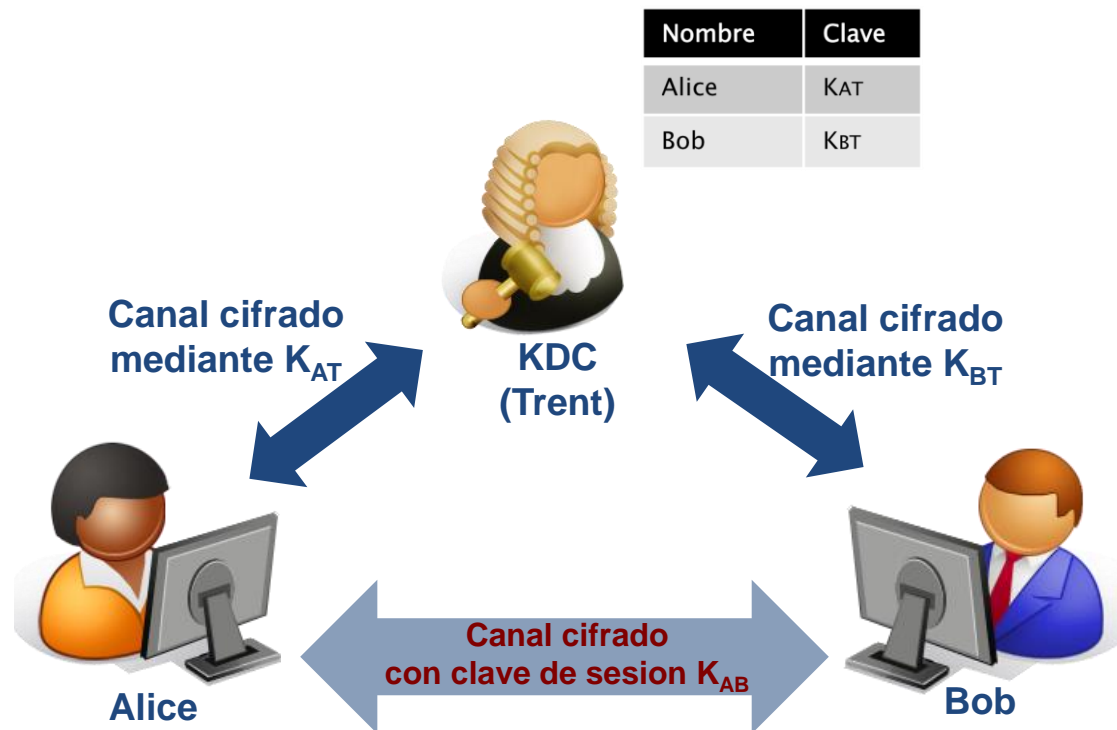
- **Gestión de las “claves”**
 - Protocolos de distribución de claves simétricas
 - Mecanismos e infraestructuras de administración de claves públicas
 - El caso del DNI-e
 - Mecanismo de Single Sign-On para Autenticación
- **Mecanismos de Control de Acceso**
 - DAC
 - MAC
 - RBAC
 - ABAC
 - Otros
- **Protocolos criptográficos avanzados**
 - Protocolos de división y compartición de secretos
 - Protocolos de compromiso de bit (bit-commitment)
 - Protocolos de lanzamiento de moneda
 - Protocolo de póker mental
- **Referencias bibliográficas**

Gestión de las Claves

Protocolos de distribución de claves simétricas

- Hay escenarios donde la utilización de la criptografía de clave pública para el intercambio de una clave de sesión K_{AB} no es posible, o simplemente no es conveniente
 - pero a pesar de ello, *Alice* y *Bob* van a seguir necesitando de alguna solución que les permitan, aún estando geográficamente lejanos, decidir esa clave de sesión K_{AB}
- En estos casos, la solución pasa por algún protocolo de **distribución centralizada de claves** para los usuarios del sistema
 - consiste en hacer uso de una tercera parte confiable (TTP), que en este caso se denomina **Centro de Distribución de Claves** (o **KDC** – *Key Distribution Center*)
- Existen diferentes protocolos que proporcionan una solución para ese escenario:
 - Yahalom, Needham-Schroeder, Otway-Rees, Kerberos, ...

- En el modus operandi general de este tipo de protocolos, cada usuario del sistema comparte, de inicio, una **clave secreta** con el KDC
 - mediante algún proceso de registro o inscripción del usuario ante el KDC



KDC: modelos y protocolos

- El uso de un KDC se basa en el uso de claves jerárquicas, de manera que se requieren al menos dos niveles de claves

Canal cifrado
mediante K_{AT} , K_{BT}

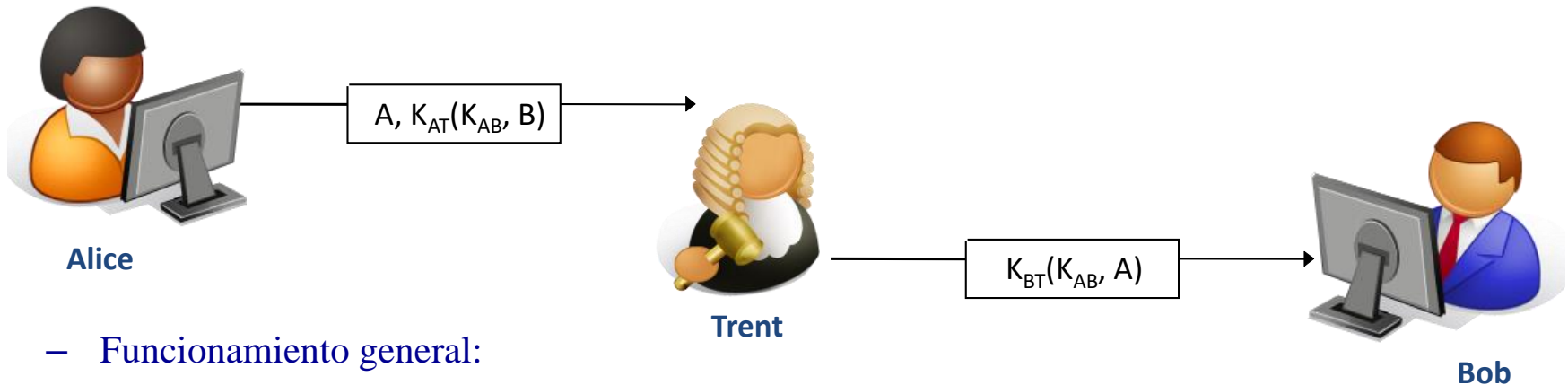


Canal cifrado
mediante K_{AB}

- La mayoría de las técnicas de distribución de claves se adaptan a situaciones, escenarios y aplicaciones específicas, de manera que son diversos los esquemas que se integran a entornos locales donde todos los usuarios tienen acceso a un **servidor común de confianza**
- Hay muchos modelos de distribución de claves:
 - **Simples**
 - **Genéricos**, y dentro de los genéricos, nos podemos encontrar:
 - Los modelos PULL o modelos PUSH, o sus combinaciones

KDC: modelos y protocolos – Modelo Simple

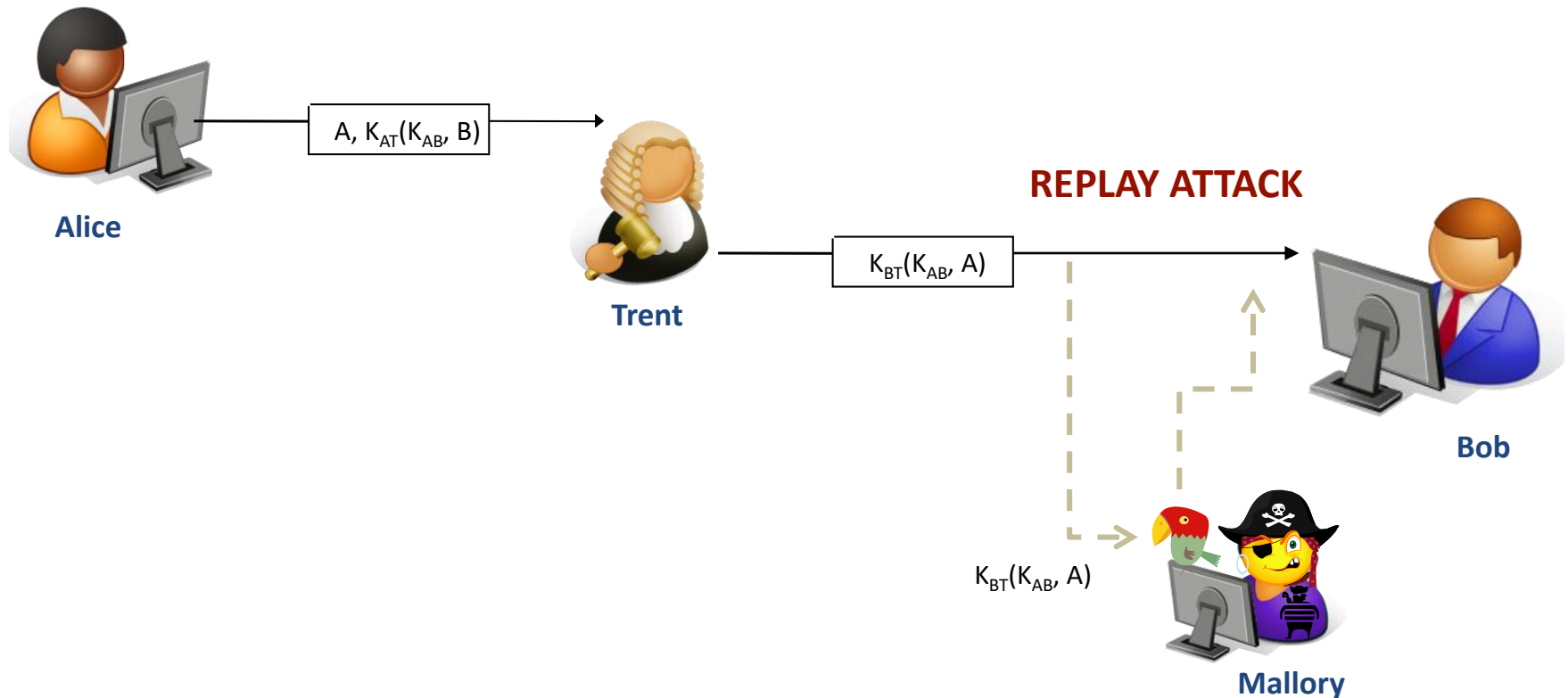
- El protocolo “**La Rana de la Boca Grande**” es un ejemplo de modelo simple para la distribución de claves:



- Funcionamiento general:
 - **Paso 1:** A genera una clave de sesión K_{AB} y se la envía al KDC
 - El mensaje incluye la identidad de A, la identidad de B y la clave de sesión cifrada con el K_{AT}
 - **Paso 2:** El KDC verifica la identidad de A y reenvía la K_{AB} a B cifrado con K_{BT}
 - **Paso 3:** B verifica la identidad de KDC por la K_{BT} y obtiene la clave de sesión
- Como se puede observar existe **validación de identidad**:
 - Las claves con el KDC son secretas, por lo que nadie más habría sido capaz de cifrar la clave secreta K_{AB} , además existe autenticación de cada parte involucrada

KDC: modelos y protocolos – Modelo Simple

- Sin embargo, existe un **fallo de seguridad**:



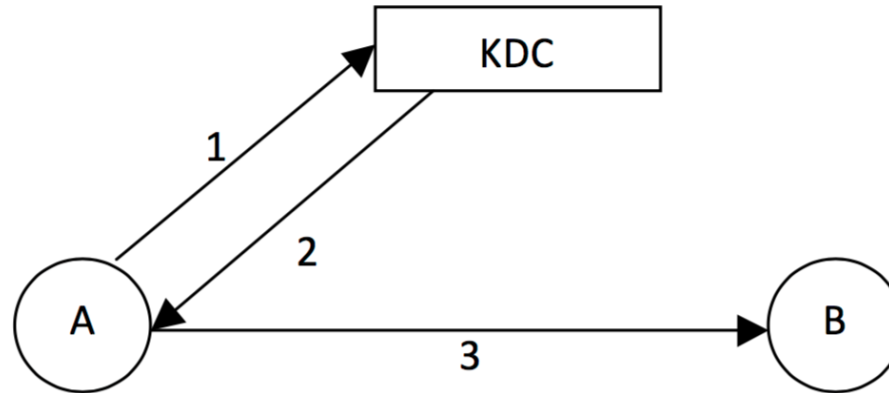
Si Mallory intercepta el canal y captura todos los mensajes de KDC a B, entonces es posible que Mallory cause un ataque de repetición (ataque replay), y, por consiguiente un ataque de Denegación de Servicio (DoS) sin necesidad de que éste derive K_{AB} o K_{BT}

KDC: modelos y protocolos – Modelo Simple

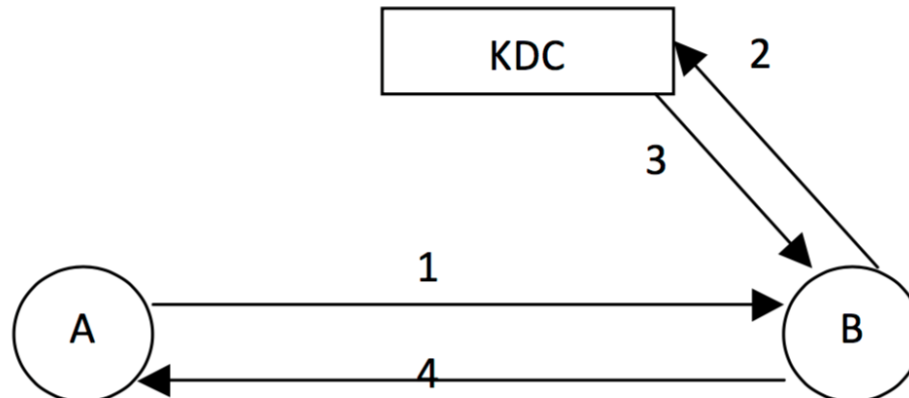
- Para resolver el problema anterior, se pueden hacer uso de alguno de los mecanismos existentes:
 - **Marca de tiempo:** incluir en cada mensaje una marca de tiempo (un sello de tiempo) de forma que pueda descartar mensajes obsoletos
 - Problema: los relojes nunca están perfectamente sincronizados en toda una red
 - **Nonce / único:** incluir un número aleatorio único para cada mensaje enviado, de forma que cada parte de la comunicación debe siempre recordar todos los únicos enviados o recibidos, y rechazar cualquier mensaje que contenga un único previamente usado
 - Problema: si una de las partes pierde la lista de nonce / únicos, es susceptible a ataques replays
 - **Combinación de ambas** estrategias para limitar el tiempo que pueden recordarse los únicos, pero el protocolo se volverá más complicado

KDC: modelos y protocolos – Modelos Genéricos

- Modelo **PULL** para la distribución de claves:

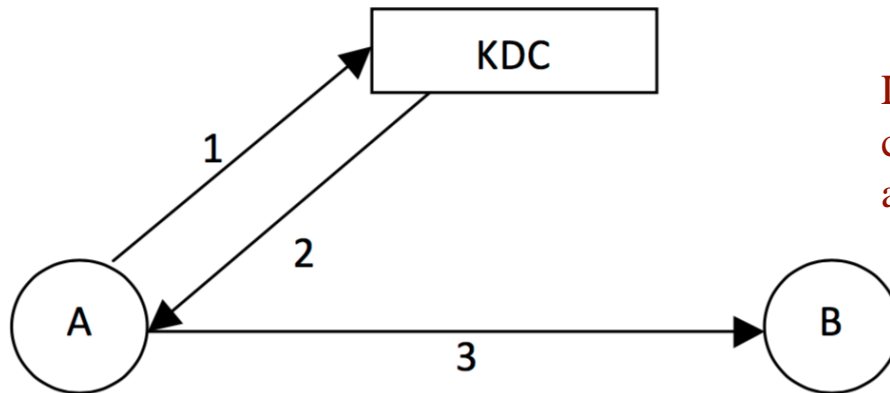


- Modelo **PUSH** para la distribución de claves:



KDC: modelos y protocolos

- Modelo **PULL** para la distribución de claves:



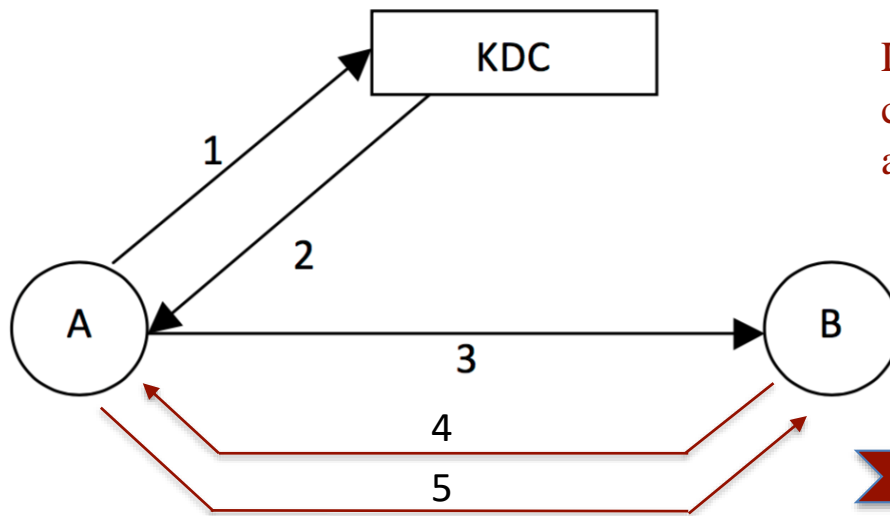
La entidad A desea tener comunicación segura con B, por lo que contacta con el KDC primero antes de comunicarse con B

– Funcionamiento general:

- Paso 1:** A solicita una clave de sesión K_{AB} al KDC
 - El mensaje incluye la identidad de A, la identidad de B y un valor **N1 (sello de tiempo, valor aleatorio)**
- Paso 2:** El KDC le contesta a A con un mensaje cifrado mediante la clave maestra K_{AT} , de manera que solamente A puede leer dicho mensaje y con ello, sabe, además, que el KDC es el único que pudo haberlo generado
 - El mensaje contiene la clave K_{AB} , N1, y un mensaje cifrado para B con el K_{AB}
- Paso 3:** A obtiene la información recibida y reenvía el mensaje a B para que pueda obtener el K_{AB} también

KDC: modelos y protocolos

- Modelo **PULL** para la distribución de claves:



La entidad A desea tener comunicación segura con B, por lo que contacta con el KDC primero antes de comunicarse con B

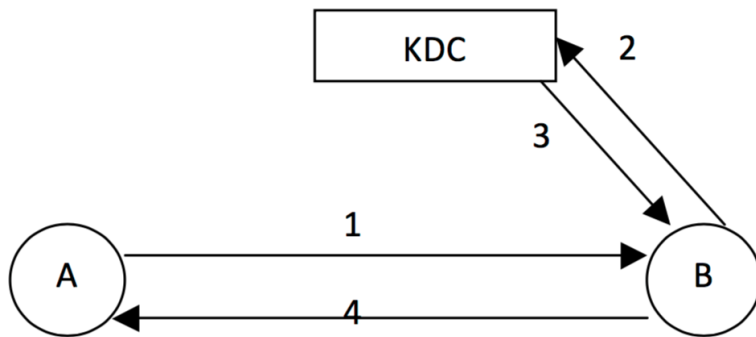
desafío-respuesta
"challenge-response"

– Funcionamiento general:

- Paso 4:** B utiliza la K_{AB} para cifrar un valor único N2 y se lo envía a A
- Paso 5:** A recibe el valor N2, le aplica una transformación $f(N2)$, lo cifra con K_{AB} y lo transmite a B

KDC: modelos y protocolos

- Modelo **PUSH** para la distribución de claves:

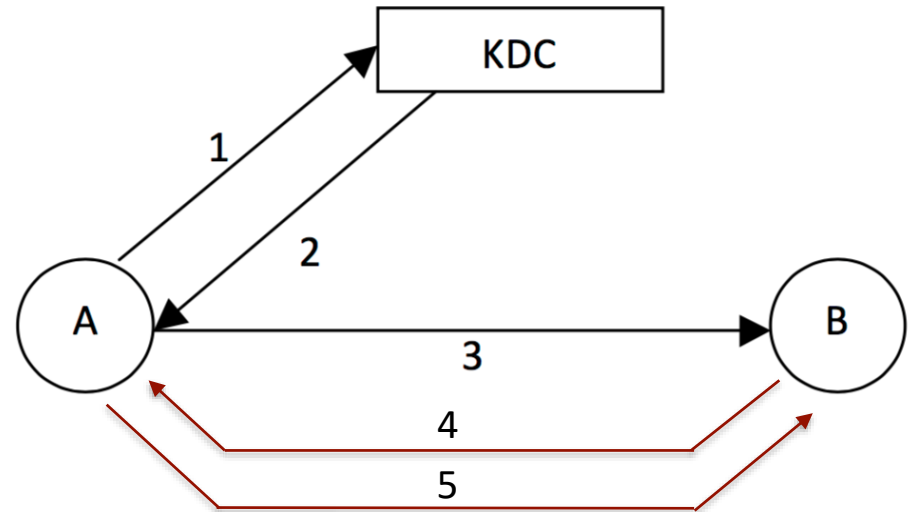


La entidad A contacta primero con la entidad B a fin de que ésta última sea la encargada de solicitar al KDC la clave correspondiente

– Funcionamiento general:

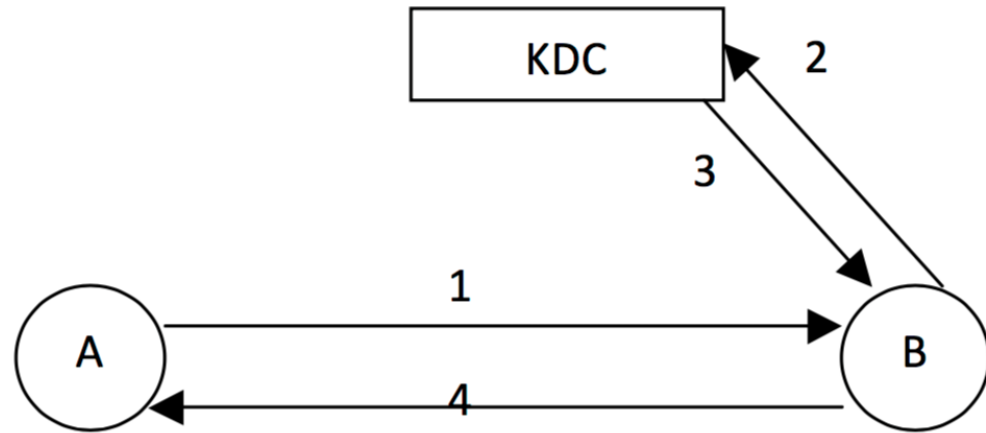
- Paso 1:** A solicita conexión segura con B, a B.
 - Le manda, como mínimo, su identidad, la identidad de B y un nonce
- Paso 2:** B reenvía dicha solicitud a KDC para que éste genere la K_{AB}
- Paso 3:** KDC verifica las identidades y el freshnesses de los mensajes, genera la K_{AB} , y dicha información se reenvía a B cifrada con la correspondiente K_{XT}
- Paso 4:** B reenvía dicha solicitud a A para que obtenga K_{AB}
- Paso 5 (opcional):** A establece un desafío y respuesta

Ejercicio 1 - PULL



- **Paso 1:** A solicita una clave de sesión K_{AB} al KDC
 - El mensaje incluye la identidad de A, la identidad de B y un valor **N1 (sello de tiempo, valor aleatorio)**
- **Paso 2:** El KDC le contesta a A con un mensaje cifrado mediante la clave maestra K_{AT} , de manera que solamente A puede leer dicho mensaje y con ello, sabe, además, que el KDC es el único que pudo haberlo generado
 - El mensaje contiene la clave K_{AB} , N1, y un mensaje cifrado para B con el K_{AB}
- **Paso 3:** A obtiene la información recibida y reenvía el mensaje a B para que pueda obtener el K_{AB} también
- **Paso 4:** B utiliza la K_{AB} para cifrar un valor único N2 y se lo envía a A
- **Paso 5:** A recibe el valor N2, le aplica una transformación $f(N2)$, lo cifra con K_{AB} y lo transmite a B

Ejercicio 2 - PUSH



- **Paso 1:** A solicita conexión segura con B, a B.
 - Le manda, como mínimo, su identidad, la identidad de B y un nonce
- **Paso 2:** B reenvía dicha solicitud a KDC para que éste genere la K_{AB}
- **Paso 3:** KDC verifica las identidades y el freshnesses de los mensajes, genera la K_{AB} , y dicha información se reenvía a B cifrada con la correspondiente K_{xT}
- **Paso 4:** B reenvía dicha solicitud a A para que obtenga K_{AB}
- **Paso 5 (opcional):** A establece un desafío y respuesta