EJERCICIOS SEGURIDAD EN SERVICIOS Y APLICACIONES

TEMA 2 – PROTOCOLOS BÁSICOS

EJERCICIO 1:

Teniendo en cuenta el siguiente protocolo:

1. T --> A : EKAT{B, EKBT{A, KAB}}
-- suponemos que A ya tiene el KAB de una transacción previa
2. A --> B : EKBT{A, KAB}
3. B --> A: EKAB{"hola otra vez"}

Y que T tiene compartido una clase secreta KBT con B y una clave KAT con A.
Analizar el protocolo, contestando a las siguientes preguntas:

1. **¿Qué hace A cuándo recibe el mensaje de T (punto 1 y punto 2)?**
   Reenvía el mensaje cifrado con KBT a B

2. **¿Qué hace B cuándo recibe el mensaje de A (punto 2 y punto 3)**
   Obtiene la clave de sesión KAB y la usa para firmar el mensaje que posteriormente le manda a A con el contenido "hola otra vez"

3. **¿Qué tipo de criptografía se está aplicando en este protocolo?**
   Criptografía simétrica.

EJERCICIO 2:

Analizar el siguiente protocolo:

T --> A : T, A, EKpubA{T, B, KAB}

A --> B : A, B, EKAB{"hola"}

1. **¿Qué hace A cuándo recibe el mensaje de T?**
   Obtiene la clave de sesión KAB y la usa para firmar el mensaje que le manda a B

2. **¿Qué tipo de criptografía se está aplicando en este protocolo?**
   Criptografía híbrida

3. **¿T se autentica a A?**
   No, porque en ningún momento usa alguna clave KAT

4. **¿A se autentica con respecto a B?**
   Si, porque usa la clave de sesión entre A y B, KAB

EJERCICIO 3:

Definir el protocolo de comunicación entre Alice y Bob, de forma que Alice envíe un mensaje M a Bob de forma segura (es decir, que haya confidencialidad).
Precondición: Alice y Bob no han establecido comunicación previamente.

- Definir problema con RSA

A → B: pubA
B → A: pubB
A calcula KAB
A → B: EpubB(KAB)
B obtiene KAB
A → B: EKAB(M)

- Definir el problema con DH

A → B: pubA
B → A: pubB
B calcula KAB = pubA ^ privB mod q
A calcula KAB = pubA ^ privA mod q
A → B: EKAB(M)

EJERCICIO 4:

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

A → B: pubA
B → A: pubB
A → B: EpubB(KAB)
A → B: EKAB(M) || EprivA(H(M))

En este caso se esta aplicando firma digital (misma solución Ej 5).
En caso de aplicar MAC (Ej 6) solo habría que hacer la siguiente modificación:
A→ B: EKAB(M) || MAC(M)

EJERCICIO 7:

Teniendo en cuenta el siguiente protocolo:

1. A → B: B, M, EKpriv_A (B, H(M))

2. B → A: A, EKpriv_B(A, H(M))

Analizarlo contestando a las siguientes preguntas:

1. **¿Qué hace B cuándo recibe el mensaje de A (punto 1)?**
   Obtiene el mensaje M y H(M), le aplica un hash al mensaje obtenido para comprobar que ambos hash coinciden, después manda este H(M) aplicado a A firmado con su clave privada.
2. **¿Qué hace A cuándo recibe el mensaje de B (punto 2)?**
   Aplica un hash al mensaje original y lo compara con el hash recibido de B, para comprobar que el mensaje original no ha sido modificado
3. **¿Qué servicio de seguridad se está aplicando realmente?**
   Integridad y no repudio de origen

EJERCICIO 8:

Teniendo en cuenta el siguiente protocolo:

1. A → T: T, B, M, EKpriv_A(T, B, H(M))

2. T → B: A, B, M, EKpriv_T (A, B, H(M))

3. T → A: A, B, EKpriv_T (A, B, H(M))

Analizarlo contestando a las siguientes preguntas:

1. **¿Qué hace T cuándo recibe el mensaje de A (punto 1)?**
   Envía el H(M) a A y B y el M a B, todo esto firmado con la Kpriv_T
2. **¿Qué hace B cuándo recibe el mensaje de T (punto 2)?**
   Obtiene el mensaje M y hash del mensaje H(M), puede comprobar integridad.
3. **¿Qué verifica A cuándo recibe el mensaje de T (punto 3)?**
   Verifica que el hash del mensaje H(M) recibido y el hash del mensaje original sean iguales.
4. **¿Qué servicio de seguridad se está aplicando realmente?**
   Integridad y no repudio de origen y destino

TEMA 3: PROTOCOLOS DE SEGURIDAD

EJERCICIO 1:

Teniendo en cuenta el siguiente protocolo:

1. A → B : A

2. B → A: Nb

3. A → B: KAT{Nb}

4. B → T: KBT{A, KAT{Nb}}

5. T → B: KBT{Nb}

Y que Trent tiene compartido una clase secreta KBT con B y una clave KAT con A. Analizar el protocolo, contestando a las siguientes preguntas:

1. **¿Qué significa cada parámetro del protocolo y para qué sirven?**
   A: Identidad de A
   B: Identidad de B
   Nb: Nonce generado por B
   KAT: Clave secreta de A con T, para cifrar los mensajes entre A y T
   KBT: Clave secreta de B con T, para cifrar los mensajes entre B y T

2. **¿Qué principales problemas existen a la hora de trabajar con KBT y KAT?**

   Que Trent podría llegar a ser un único punto de fallo

3. **¿Hay algún problema de seguridad en el punto 4 del protocolo? Razonar la respuesta.**

   No existe autenticación, puesto que la única que conoce la identidad de A es T y en ningún momento lo autentica con B

4. **En caso afirmativo (punto 3), mencionar alguna forma de resolver el problema.**
   Añadiendo la identidad de A en el paso 5, T → B: KBT(A, Nb)

EJERCICIO 2:

Teniendo en cuenta el siguiente protocolo:

1. **A --> B: A, Na**
2. **B --> T: B, EKBT{A, Na, Nb}**
3. **T --> A: EKAT{B, KAB, Na, Nb), EKBT{A, KAB}**
4. **A --> B: EKBT{A, KAB}, EKAB{Nb}**

Y que Trent tiene compartido una clase secreta KBT con B y una clave KAT con

A. Analizar el protocolo, contestando a las siguientes preguntas:

1. **¿Quién inicializa el proceso de negociación de la clave de sesión KAB?**
   A

2. **¿Quién inicializa el proceso de negociación de la clave KAB con T?**
   B

3. **¿Qué modelo sigue este protocolo (PULL o PUSH)?**
   PUSH extendido (Yahalom)

4. **¿Existe posibilidad de ataques de repetición?**
   No, porque para ello se usan los nonces, para evitar mensajes repetidos

5. **¿Tiene B alguna forma de verificar que el mensaje viene de Alice y que está realmente hablando con ella?**
   Sí, porque A reenvía el nonce de B que ha obtenido de T, la cual solo reenvía este nonce a A.
   Sería como la respuesta de un desafío que no ha establecido B, es decir, A solo ha podido obtener este Nb mediante T.

EJERCICIO 3:

Teniendo en cuenta el siguiente protocolo:

1. **A -->B: A, EKAB{NA}**
2. **B -->A: EKAB{NA, NB}**
3. **A -->B: EKAB{NB}**
4. **B -->A: EKAB{KAB', NB'}**

Analizar el protocolo, contestando a las siguientes preguntas:

1. **¿Se debe reordenar los mensajes para que tenga sentido el protocolo?**
   No

En caso afirmativo, reordenarlo.

2. **¿Qué se hace en los puntos 1-3 del protocolo?**
   Intercambio de nonces. Una especie de desafío-respuesta.

3. **¿Qué significa la clave KAB'?**
   La nueva clave que se va a usar para el canal entre A y B

4. **¿Qué tipos de ataques podrían existir si asumimos previamente la existencia de M (Mallory)? Razonar la respuesta.**
   Replicación del mensaje del paso 4, ya que este no contiene el nonce de A

5. **¿Cómo se podría resolver dichos ataques? Razonar la respuesta.**
   Añadiendo el nonce de A en el mensaje del paso 4.

6. **¿Qué sentido tendría usar un MAC en este protocolo?**
   Para garantizar la integridad de los datos y la autenticación, aunque se podría seguir dando ataque de replicación de mensajes.

EJERCICIO 4:

Supongamos ahora que en vez de aplicar criptografía simétrica para la gestión de claves KAB, 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 (Na y Nb) que les permitan a cada entidad (A y B) verificar el freshness de las transacciones.

Dado esto, y los mensajes siguientes:

1. **A, B**
2. **B, A**
3. **Kpub_A{Na, Nb, B}**
4. **Kpriv_T{Kpub_B, B}**
5. **Kpub_B{Nb}**
6. **Kpriv_T{Kpub_A, A}**
7. **Kpub_b{NA, A}**

Se pide

1. **Reorganizar los mensajes teniendo en cuenta la existencia de Trent (T) - como mediador entre A y B-, y la existencia de sus claves Kpriv_T y Kpub_T.**
   Para realizar el ejercicio, es fundamental identificar y declarar el origen y el destino de cada mensaje (ej: A --> B), y asumir que T conoce las claves públicas de A y B, y se encarga de enviarlos cuando ellos son necesitados.

   - A → T: A, B
   - T → A: Kpriv_T(Kpub_B, B)
   - B →T: B, A
   - T → B: Kpriv_T(Kpub_A, A)
   - A → B: Kpub_B(Na, A)
   - B → A: Kpub_A(Na, Nb, B)
   - A → B: Kpub_B(Nb)

2. **¿Qué hace realmente el protocolo? ¿Cuál es su objetivo final?**
   Es una especie de challenge-response donde se intercambian los nonces pero previamente han tenido que solicitar a T la clave pública del otro punto.

EJERCICIO 5:

Teniendo en cuenta el siguiente protocolo:

1. **A --> B: Kpub_A**
2. **B --> A: Kpub_B**
3. **A --> B: Kpub_B{KAB}**

Se pide:

1. **Optimizar el protocolo para que Alice pueda verificar que la clave pública es genuina y pertenece a Bob.**
   Si se intercambian los certificados
   A → B: Kpub_A || CertB_A
   B → A: Kpub_B || CertA_B

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.**

EJERCICIO 6:

Teniendo en cuenta el siguiente protocolo:

1. **CA1 --> A: CertA_CA1**
2. **CA2 --> B: CertB_CA2**
3. **A --> B: CertA_CA1**
4. **B --> A: CertB_CA2**

Asumiendo que A conoce CA1 y tiene su clave pública Kpub_CA1 y B conoce CA2 y tiene su clave pública Kpub_CA2, se pide contestar a las siguientes preguntas:

1. **¿Puede A verificar el certificado CertB_CA2 si CA1 =! CA2? Razonar la respuesta.**
   No, porque si son diferentes A no tiene manera de comparar CertB_CA2 con ningún otro dato o certificado
   Solo sería posible si tiene alguna forma de conseguir la Kpub_CA2
2. **¿Existe alguna forma de que A y B puedan compartir las claves de forma confiable?**
   Si, con cualquier algoritmo de compartición de clave, ya que se tratan de claves públicas.

Otra solución posible es con una CA común que verifique tanto CertA_CA1 como CertB_CA2, de esta manera se crearía una cadena de confianza.

TEMA 3: PROTOCOLOS AVANZADOS

EJERCICIO 1:

Considerando el protocolo de división de secretos y M= 110010101100:

a) **dividir el mensaje en dos sombras.**
Dividir el mensaje en dos trozos: 110010 101100

b) **Realizar la misma actividad, pero para seis sombras.**
Igual, pero dividiendo en 6 (12/6=2), cada trozo tiene 2 números

c) **Supongamos que una de las sombras no llega al destino. ¿Qué ocurre?**
No se puede descifrar el mensaje, se necesita tenerlas todas para conocer el mensaje completo.

EJERCICIO 2:

Considerando el protocolo de compartición de secretos y las siguientes condiciones:
• k=3 • D=263

a) **dividir el mensaje en 5 sombras, ocultando el valor original del dato (D).**
$q(x) = a0 + a1\ x + a2\ x^2$
$a0 = 263$
$a1 = 456$
$a2 = 945$ $q(x) = 263 + 456\ x + 945\ x^2$

$q(1) = 1664$, $q(2) = 4955$, $q(3) = 10136$, $q(4) = 17207$, $q(5) = 26168$

b) **Recuperar el mensaje suponiendo que se han recibido sólo tres sombras de 5.**
Quedaría un sistema de 3 ecuaciones con 3 incógnitas, del que se puede sacar a0 que es el mensaje original.

c) **En el caso extremo que se reciban 2 sombras cualesquiera, ¿cuáles serían los polinomios? ¿Qué problema existe?**
Tendríamos solo 2 ecuaciones por tanto no podríamos resolver el sistema.

EJERCICIO 3:

Considerando el protocolo de bit-commitment y el uso de las funciones hash:

a) **¿Qué ocurre si Alice envía a Bob H(R1,b), R1?**
   Bob no puede conocer el contenido de b, pero más tarde si Alice le manda b en claro puede aplicar hash a b y R1 y comparar con el hash que había recibido antes para ver que se mantiene la integridad de los datos.

EJERCICIO 4:

Considerando el protocolo de póker mental: a) generalizar el problema para 4 personas.

A → B: Ekpub_A(Mi)

B → A: Ekpub_B(Ekpub_A(Mj)) siendo j = 1…5

A: Dkpriv_A(Ekpub_B(Ekpub_A(Mj))) = Ekpub_B(Mj)

A → B: Ekpub_B(Mj)

B: Dkpriv_B(Ekpub_B(Mj)) = Mj

B → C: Ekpub_A(Mk) siendo k = 1…5

C → A: Ekpub_C(Ekpub_A(Mk))

A: Dkpriv_A(Ekpub_C(Ekpub_A(Mk)) = Ekpub_C(Mk)

A →C: Ekpub_C(Mk)

C: Dkpriv_C(Ekpub_C(Mk))

C → D: Ekpub_A(Ml) siendo l = 1…5

D → A: Ekpub_D(Ekpub_A(Ml))

A: Dkpriv_A(Ekpub_D(Ekpub_A(Ml)) = Ekpub_D(Ml)

A →D: Ekpub_D(Ml)

D: Dkpriv_D(Ekpub_D(Ml)) = Ml

D → A: Kpub_A(Mm) -- Envía 5 cartas de las restantes, que será la mano de Alice

TEMA 4: PROTOCOLOS

EJERCICIO 1:

Dado el siguiente protocolo:

1. **A → B: E_{Ks1}{data1} + E_{Ks2}{data2} + $E_{Kpub\_B}{K_{s1}}$ + $E_{Kpub\_C}{K_{s2}}$ + H(data1) + H(data 2) + E_{Kpriv_A}{H(H(data1)+H(data2))}**
2. **B → C: E_{Ks2}{data2} + $E_{Kpub\_C}{K_{s2}}$ + H(data1) + E_{Kpriv_A}{H(H(data1)+H(data2))}**

Analizar el protocolo, contestando a las siguientes preguntas:

1. **¿Qué tipo de protocolo se está aplicando aquí?**

   SET con firma dual

2. **¿Qué tipo de servicios se están ofreciendo con este protocolo?**

   Confidencialidad, Autenticación de origen y destino, Integridad, No repudio

3. **¿Podría existir un ataque de DoS? Razonar la respuesta. En caso afirmativo, indicar cómo se podría evitar la amenaza.**

   Si, puesto que cualquier mensaje se podría replicar sin problema ya que ninguno contiene nonces o marcas de tiempo. Para evitar habría que añadir nonces.

EJERCICIO 2:

¿Cómo se podría adaptar el protocolo de firma dual a un sistema sanitario?

El paciente le pasa toda la información al doctor, este solo es capaz de ver la que necesita y el resto, que no puede ver, se la reenvía a la administración.

El sistema cuenta con tres actores importantes:

- **Un paciente.**
- **El doctor.**
- **Administración.**

¿Qué utilidad tendría aquí la firma dual en este escenario? Razonar la respuesta

Ocultar datos privados del individuo al doctor (dni, num seguridad social…) pero haciendole llegar la información que si necesita (edad, nombre, altura, peso…).

EJERCICIO 3:

Dado el siguiente protocolo:

1. **Alice → Bob: Bob, L, C, NO**
2. **Bob → Alice: Alice, L, NR**
3. **Alice → TTP: Bob, L, K, proc_K**
4. **TTP → Bob: Alice, Bob, L, K, pub_K**
5. **TTP → Alice: Alice, Bob, L, K, pub_K**

Donde:

- **L: representa la secuencia del protocolo y se basa de un entero.**
- **S: indica la firma digital.**
- **C: el cifrado de un mensaje, es decir: E_K{M}.**
- **K: La clave simétrica aplicada en el cifrado de M.**
- **NO = S_{Alice}*(Bob, L, H(C)).***
- **NR = S_{Bob}*(Alice, L, H(C)).***

Contestar a las siguientes preguntas:

1. **¿Qué tipo de protocolo se está aplicando aquí?**

   Distribución de claves simétricas (en este caso renovación de clave)

2. **¿Qué tipo de servicios se están ofreciendo con este protocolo?**

   Autenticación, No repudio, Integridad

3. **¿Qué contenido debería tener proc_K y pub_K?**

   Proc_K: A,B – Petición de clave de sesión nueva

   Pub_K: EkAT(K2) || EkBT(K2)

4. **¿Podría existir un ataque de DoS? Razonar la respuesta. En caso afirmativo, indicar cómo se podría evitar la amenaza.**

   No, porque se usa el entero L que debe ser único (sirve a modo de nonce).

EJERCICIO 4:

Adaptar el siguiente protocolo del ejercicio 3 al protocolo de firma dual (basado en una comunicación estrella – no en línea).

Dicho de otro modo, combinar ambos protocolos y contestar a la siguiente pregunta:

1. **¿Qué tipo de servicios se están ofreciendo con este protocolo combinado?**

   Autenticación, No repudio, Integridad, Confidencialidad

TEMA 5: FIREWALLS, VPNs Y COMUNICACIÓN INÁLAMBRICA

EJERCICIO 1:

Supongamos que tenemos dos nodos (A y B) que desean establecer comunicación punto a punto mediante el protocolo IPSec. Contestar a las siguientes preguntas:

- **¿Qué tipo de protocolo de IPSec se debería aplicar si se desea confidencialidad?, ¿y autenticación?, ¿y ambos?**

  Para confidencialidad ESP, para autenticación ESP o AH, para ambos IKE

- **¿Por qué es esencial definir un código de SPI para cada asociación de seguridad? Razonar la respuesta.**

  Para identificarla

- **¿En la asociación de seguridad se especifica el modo de comunicación (túnel o transporte) entre ambos nodos?**

  No, eso lo realizan las Security Policy (SP) o también algunas SAD

- **¿Crees que toda comunicación o protocolo que se establezca en capas superiores de la capa de red iría por la VPN? Si es así, ¿existe alguna forma de especificar el tipo de protocolo para que vaya solo este tipo de comunicación por la VPN?**

  No

- **Mencionar al menos dos situaciones o escenarios en el que se recomendaría aplicar una VPN, y razonar la respuesta.**

  Para proteger los paquetes de ataques externos

TEMA 5: PROTOCOLOS – TLS

EJERCICIO 1:

Especificar (de manera formal) la fase de "sesión" de TLS (v1.2), considerando la existencia de dos nodos, uno funcionando como cliente (A) y el otro como servidor (B):

1. A → B: Client Hello
2. B → A: Server Hello
3. A → B: ClientKeyExchange, ChangeCipherSpec
4. B → A: ServerKeyExchange, ChangeCipherSpec

Una vez definida la fase de sesión, completar el protocolo de TLS (v1.2) para contemplar el envío seguro por la fase de "conexión", de forma que A le envía a B el siguiente mensaje: "*Hola servidor B, te envío este secreto por TLSv1.2*".

5. A → B: Ek(M)

EJERCICIO 2:

Especificar (de manera formal) la fase de "sesión" de TLS (v1.3), considerando la existencia de dos nodos, uno funcionando como cliente (A) y el otro como servidor (B):

1. A → B: ClientHello, ClientKeyExchange
2. B → A: ServerHello, ServerKeyExchange, ChangeCipherSpec
3. A → B: ChangeCipherSpec

Una vez definida la fase de sesión, completar el protocolo de TLS (v1.2) para contemplar el envío seguro por la fase de "conexión", de forma que A le envía a B el siguiente mensaje: "Hola servidor B, te envío este secreto por TLSv1.3".

A → B: Ek(M)

EJERCICIO 3:

Considerando que A (el cliente) y B (el servidor) usan TLSv1.2, especificar de manera formal el protocolo de intercambio de clave establecido entre A y B, bajo las siguientes condiciones iniciales:

• A y B deciden negociar la clave de sesión de forma que el cliente (A) envía cifrado con RSA el pre-shared master key, y

• A no sabe de antemano la clave pública.

La formalización seguiría el siguiente formato:

1. A → B: ClientHello: (Na)
2. B → A: ServerHello: (Kpub, Nb)
3. A → B: ClientKeyExchange: (Epub(pre-shared master key), Na)

Realizar el mismo ejercicio, pero en lugar de usar RSA, aplicar DH.

A → B: Na

B → A: G, N, Yb, EprivB(G, N, Yb, Nb)

A → B: Ya

EJERCICIO 4:

Considerando que A (el cliente) y B (el servidor) usan TLSv1.2, especificar de manera formal el protocolo de intercambio de clave establecido entre A y B, bajo las siguientes condiciones iniciales:

• A y B deciden negociar la clave de sesión con DHE firmado con RSA, y

• A no sabe de antemano la clave pública.

La formalización seguiría el siguiente formato:

1. A → B: Na
2. B → A: G, N, Yb, EprivB(G, N, Yb, Nb)
3. A → B: Ya
4. A: Obtiene pre-shared master key
5. B: Obtiene pre-shared master key