Ejercicio 1

Enunciado

La empresa Tecnosalud S.A. está desarrollando un chip que se puede alojar dentro del cuerpo de una persona y permite almacenar su historia clínica, para poder brindar un mejor tratamiento en caso de ocurrir algún accidente. Como dicha información varía a lo largo del tiempo el chip debe permitir la modificación de datos luego de su instalación.
Tras realizar un modelado de amenazas, se decide:

- Que el chip mantenga separada información inmutable (como nombre, fecha de nacimiento, grupo sanguíneo), información de contacto (nombre de contacto, teléfono, email, etc), e información médica. El primer grupo de información no podrá ser modificada; el segundo sí; y en el tercero solo se permitirá agregar información nueva.
- Que la información deberá estar cifrada de alguna manera.
- Que, sin un dispositivo autorizado, no será posible leer la información.
- Que, si un dispositivo es robado, deberá poder inutilizarse el uso del mismo.
- Que la información esté almacenada en el chip y nunca viaje a servidores donde sea almacenada de forma centralizada.

Para ello se decide desarrollar un sistema con un controlador online que arbitre el acceso a la información de los dispositivos:

- Cada chip y cada lector tendrá un número de serie y una clave simétrica asociada. La clave es única por chip.
- El comando central (controlador) posee una base de datos con todos los números de serie y su clave asociada.
- Cuando un dispositivo quiere acceder a la información del chip, consigue su número de serie (utilizando un protocolo similar a RFID –los chips informan públicamente su número de serie), y solicita permiso para leerlo al comando central.
- El permiso, si es concedido, consiste en un token que representa una capacidad que el comando central otorga al dispositivo. Con esa capacidad, el dispositivo puede obtener del chip la información que solicita, o actualizar/agregar información nueva (algunos dispositivos solo pueden leer, otros también actualizar información).
- Solo los dispositivos pueden comunicarse con el comando central. Los chips solo se conectan a corta distancia física.

a. Describir un posible protocolo enumerando los mensajes que intercambiarían CH (chip), DS (dispositivo) y CC (comando central), para que el segundo acceda a información del primero, y que debería hacer/verificar cada parte.(1 punto).
b. Explicar cómo ocurriría en el protocolo anterior que un dispositivo robado no siga accediendo indefinidamente a información de chips. ¿Es posible evitar que vuelva a leer chips para los cuales ya había recibido permiso anteriormente? (1 punto).
c. Explicar por qué no es posible que alguien simule ser un dispositivo y acceda a la información del chip. (1 punto).
d. Explicar por qué el protocolo no puede ser offline. ¿Qué requerimiento no puede cumplirse si el protocolo no contase con un comando central? (1 punto).

Ejercicio 2

Enunciado

Una importante cadena de supermercados decidió construir su propio sistema de puntos de venta, que corre en todas las cajas de todas sus sucursales, y está integrado a los sistemas de stock y contabilidad.

El sistema está compuesto por una aplicación nativa que corre en una distribución de Linux modificada para actuar como un Kiosk (al iniciarse se ejecuta la aplicación desarrollada y no hay forma de ejecutar otras aplicaciones), un servicio REST expuesto en la red interna que expone la funcionalidad del sistema, y que utiliza una base de datos PostgreSQL y otros microservicios para completar su funcionamiento.

Para utilizar la aplicación, cada cajero dispone de una tarjeta, que al ser pasada por un lector se convierte en un número de serie, y debe ingresar un pin numérico de 8 dígitos que sólo él conoce. El servicio expone un recurso a fin de iniciar sesiones: POST /session, que recibe como mensaje un document JSON con el siguiente formato:

```
{
    "user": "<número de serie>",
    "credentials": "<información de autenticación>"
}
```

y retorna un identificador para la nueva sesión o un error si algo falló. Adicionalmente, en la intranet no está permitido el uso de HTTPs por temas de balance de carga y auditoría, así que la llamada al recurso se realizará utilizando HTTP sin seguridad adicional.

A partir de una sesión de modelado de amenazas, se determinó que los dos ataques más peligrosos, y contra los cuales la solución debe ser segura, son:

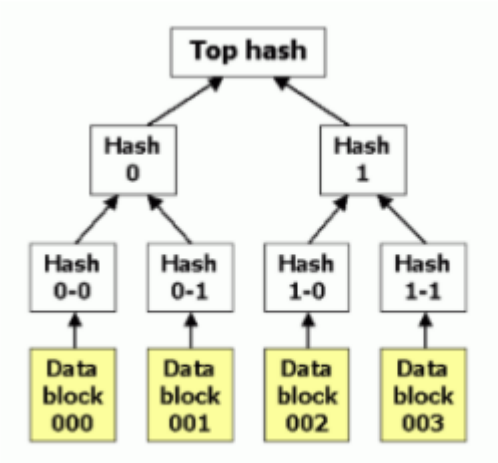
- El administrador de las bases de datos quiere impersonar a un cajero.
- El administrador de red (que puede capturar mensajes) quiere impersonar a un cajero.

a. Definir cómo implementaría el servicio de autenticación, explicitando como mínimo (1 punto):
    i. Cómo guardaría en la base de datos la información que permita autenticar a los usuarios.
    ii. Cómo quedaría conformado el valor de “credentials” en el mensaje del servicio.
    iii. Cómo utilizaría la información recibida y la almacenada para autenticar al cajero.
b. Explicar por qué el sistema resultante es inmune a los escenarios de ataque mencionados. (3 puntos)

Ejercicio 3

Enunciado

Cuando se necesita verificar parcialmente la integridad de un conjunto de datos, obtener el hash de todo el mensaje es ineficiente. Considerar, por ejemplo, un mensaje de 1TB de tamaño, en donde interesa verificar la integridad de un bloque de 1MB.
Para este tipo de problemas, se utiliza una construcción llamada Merkle Tree, que consiste en un un árbol donde cada hoja lleva el resultado de aplicar la función de hash a un subconjunto del mensaje, y cada nodo lleva el resultado de aplicar la función de hash a sus nodos hijos inmediatos. El siguiente ejemplo muestra un árbol binario, con un mensaje dividido en 4 partes:



En este caso, H0 = h(H0-0 || H0-1), H1 = h(H1-0 || H1-1) y Top Hash = h(H0 || H1)
Suponiendo que la función de hash utilizada es libre de colisiones, explicar:

1. ¿Por qué para verificar la integridad del bloque 001, solo es necesario calcular H0-1? (0.5 puntos)
2. ¿Por qué para verificar la segunda mitad del mensaje NO alcanza con verificar H1-0 y H1-1? (0.5 puntos)
3. Un Merkle Tree, sin más modificaciones, puede no ser resistente a segundas preimágenes. ¿Qué condiciones se tienen que dar para que el mensaje M'=H0-0 || H0-1 || H1-0 || H1-1 resulte en el mismo Top Hash? (1 punto)
4. Si alguien pudiera encontrar una segunda preimagen de alguno de los bloques, podría generar una segunda preimagen del Top Hash. Explicar cómo haría esto, y por qué no representa una debilidad en la primitiva. (1 punto)

Ejercicio 1

a. Protocolo de comunicacion

(1) DS → CH : pide id (por RFID)

(2) CH → DS : id_{CH} || timestamp

(3) DS → CC : id_{CH} || timestamp || ids

(4) CC → DS : $\{\{Cap || id_{CH} || ids || timestamp\}_{CH}\}_{DS}$

(5) DS → CH : $\{Cap || id_{CH} || ids || timestamp\}_{CH}$ || informacion || ids (necesario?)

b. Si un dispositivo es robado, se borra el id de la base de datos entonces deja de ser dispositivo autorizado. Por otro lado, el timestamp evita que se vuelvan a leer los chips porque el chip verifica el timestamp para verificar si ese pedido ya fue atendido.

c. Si un atacante desea similar ser un dispositivo deberia obtener su ID. Sin embargo, esto es imposible porque si pide el ID de algun dispositivo al CC, este devuelve el ID encriptado bajo la clave del dispositivo. Luego, como el atacante no tiene la clave, no lo podria desencriptar.

d. El protocolo no podria ser offline porque esto implicaria que cada chip tenga que almacenar los ids de los dispositivos y las claves. Luego, si un atacante roba un chip, tendria toda esta informacion.

Ejercicio 2

a. Usar PBKDF2

Ejercicio 3

a. Si hubiese algun cambio en el bloque 001, H0-1 tambien se veria modificado. Esto se debe a que cualquier cambion en la entrada de una funcion de hash implica una modificacion en lo salida de la misma. Por lo tanto, es suficiente con chequear H0-1 para ver si se altero el bloque 001.

b. No alcanza porque H1-0 y H1-1 solo verifican la integridad de un bloque y no de la concatenacion. Suponiendo que nos dan B003 antes que B002, H1-0 y H1-1 van a ser distintos a los que obtenemos si los bloques estan en orden. Si bien los hashes son distintos, no hubieron modificaciones en los mensajes. Por lo tanto, es importante llevarlo al siguiente nivel y chequear el hash de la concatenacion.

c. Suponiendo que cada bloque es de longitud n y que los hashes son de longitud $n/2$, entonces separamos el mensaje M' en dos bloques $H0\text{-}0 \| H0\text{-}1$ y $H1\text{-}0 \| H1\text{-}1$. Luego, hacemos el hash de cada bloque: $H(H0\text{-}0 \| H0\text{-}1)$ y $H(H1\text{-}0 \| H1\text{-}1)$. Finalmente, hasheamos todo y obtenemos $H(H(H0\text{-}0 \| H0\text{-}1) \| H(H1\text{-}0 \| H1\text{-}1)) = H(H0 \| H1)$.

d. Se puede tomar como mensaje $M' = H0\text{-}0 \| H0\text{-}1 \| H1\text{-}0 \| H1\text{-}1$, al igual que el punto anterior.