Seguridad en Redes

Resolución de ejercicios

4.06.2025

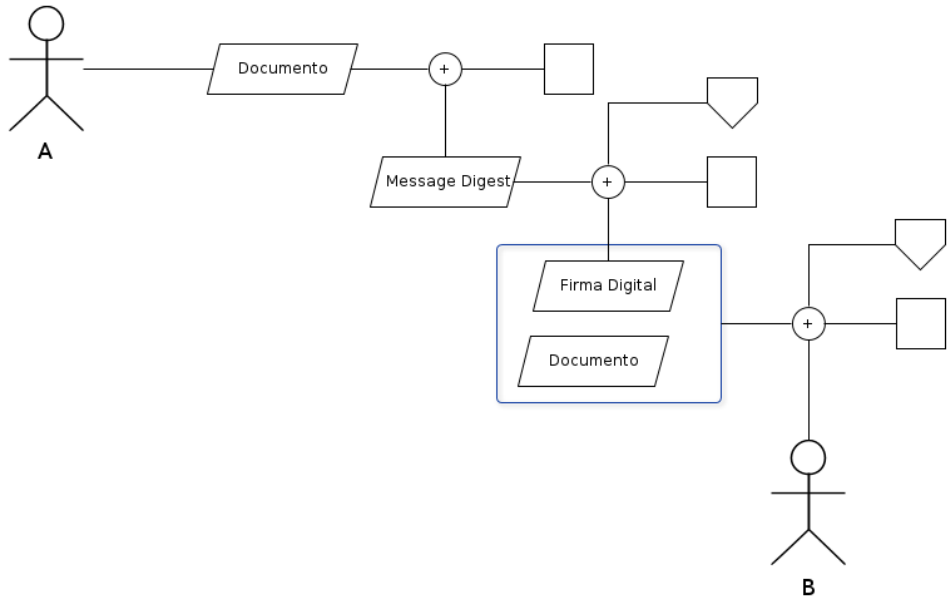
1. Primer ejercicio

1.1. Enunciado

A desea enviarle un mensaje a B de carácter importante. A quiere asegurarse de que nadie excepto B pueda leerlo y que B pueda confiar en que A fue quién envió el mensaje.

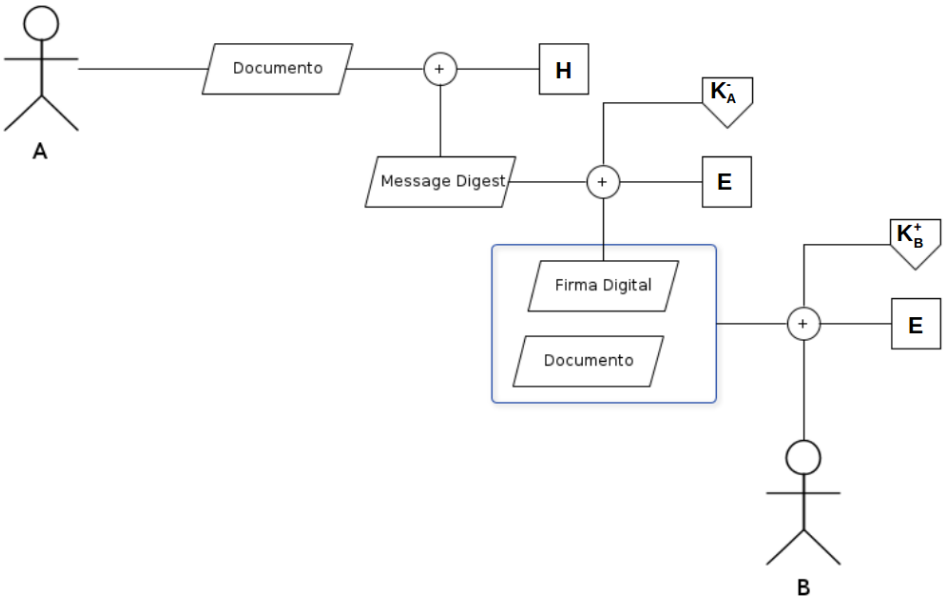
a. Basandose en el esquema a continuación, completando los cuadrados con algoritmos y las casitas con los parámetros adicionales que estos toman, explique cómo puede hacer A para construir el mensaje usando criptografía asimétrica, de manera de garantizar las propiedades de Confidencialidad y Autenticación.

b. Explique qué debe hacer B para verificar que A fue quién envió el mensaje.



1.2. Resolución

a. Sean K_A^-, la clave privada de A; K_B^+, la clave pública de B; E un algoritmo de encripción de clave asimétrica; y H una función de Hash, para garantizar Confidencialidad y Autenticación hay que hacer:



b. Dado el criptograma C que recibe B, se debe desencriptar usando un algoritmo de clave asimétrica D con su clave privada K_B^-, obteniendo $D_{K_B^-}(C) = M + F$, dónde M es el documento original que envía A, y F, la firma digital. Como B es el único capaz de desencriptar el mensaje cifrado C, entonces queda garantizada la Confidencialidad. Luego, para verificar la firma, B debe desencriptar la firma usando la clave pública de A (K_A^+) y comparar aplicando la misma función de Hash al documento: $D_{K_A^+}(F) == H(M)$, y así queda garantizado la autenticación de A sobre el mensaje enviado.

2. Segundo ejercicio

2.1. Enunciado

Suponga que una compañía necesita implementar un sistema que garantice la autenticidad de sus clientes usando un sólo servicio instalado en un servidor.

a. Se sabe que ssh admite autenticación por clave pública y privada mediante el método de challenge-response. Explique dónde y cuántas claves públicas y privadas deberían instalarse para que se pueda garantizar la autenticidad de los clientes usando ssh.

b. Suponga ahora que los clientes necesitan garantizar la autenticidad del servidor de la compañía, ¿Dónde deberían instalarse las claves públicas y privadas?

c. También se puede garantizar autenticidad estableciendo una conexión SSL/TLS que usa un handshake seguro intercambiando certificados digitales. Explique cómo cambian la soluciones de los incisos **a.** y **b.**

2.2. Resolución

a. En el caso de SSH, las claves deben instalarse previamente en cada cliente y servidor, dependiendo de quién debe garantizar la autenticidad y sobre quién. Para que se pueda garantizar la autenticidad de N clientes ($C_i, 1 \leq i \leq N$), cada uno de ellos debe generar un par de claves pública y privada ($K_{C_i}^+, K_{C_i}^-$). Luego, deben instalarse las N claves públicas $K_{C_i}^+$ en el servidor asociandose a cada uno de los clientes.

b. Para garantizar la autenticidad del servidor, éste debe generar un par de claves pública y privada (K_S^+, K_S^-) y, luego, debe instalarse K_S^+ en cada cliente.

c. Primero que nada, todas las máquinas (clientes y servidor) tienen que tener instalado un certificado de una autoridad certificante (CA) que contiene su clave pública: K_{CA}^+.
En el handshake SSL/TLS la autenticación del servidor es obligatoria. Éste debe generar un par de claves pública y privada (K_S^+, K_S^-), generar su certificado, hacerlo firmar por CA e instalarlo en el servidor para su posterior envío en el Handshake SSL.
Luego, para garantizar la Autenticidad de los clientes C_i, cada uno de estos debe generar un par de claves pública y privada $(K_{C_i}^+, K_{C_i}^-)$, construir un certificado con la clave pública $(K_{C_i}^+)$ y hacerlo firmar por la autoridad certificante de manera que lo asocie con el nombre del cliente (C_i). El certificado que se debe instalar en cada cliente, a manera de ejemplo, contendría las siguientes piezas de información:

$$(\, C_i;\, K_{C_i}^+;\, F_{CA,C_i}),\ \text{con}\ F_{CA,C_i} = E(H(C_i + K_{C_i}^+), K_{CA}^-)\,)$$

Donde E es un algoritmo de encripción de clave asimétrica y H es una función de Hash. Luego, se instala cada certificado en cada cliente para que se envíe durante el Handshake SSL y que el servidor pueda garantizar la autenticidad usando K_{CA}^+ para validar el certificado y, luego, $K_{C_i}^+$ para autenticar al cliente.
Si no se necesita autenticar clientes se omite la parte correspondiente. En el handshake SSL/TLS la autenticación del cliente es opcional.

3. Tercer ejercicio

3.1. Enunciado

La empresa Security First expone una API usando el protocolo HTTP. Dicha API tiene un método *getPrice* que devuelve el precio de frutas usando la siguiente url:

`http://api.securityfirst.com/?method=getPrice&fruit=Pera`.

Además, la empresa tiene su propio servidor DNS autoritativo del dominio `securityfirst.com`.

a. Es necesario agregar un método nuevo a la API que haga lo mismo que el getPrice pero que provea el servicio sólo a usuarios autenticados brindando **integridad** y **no repudio** del lado del servidor sobre los pedidos de los usuarios.

1. Explique qué información tiene que tener el cliente y el servidor para que esto sea posible.
2. Muestre una posible url que permita procesar el pedido.
3. Explique qué es lo que tiene que hacer cada parte para garantizar lo solicitado.

Nota: Es requisito para lo anterior no usar encripción sobre la totalidad del mensaje y no manejar sesiones.

3.2. Resolución

a. La idea aca es extender la funcionalidad usando la firma digital dado que nos piden Integridad y No repudio.

   1. Cada cliente debe tener generada una clave púbilica y su respectiva clave privada. El server debe tener la clave pública de cada cliente autorizado a consumir la API. Se asume que este intercambio se realizó previamente y de manera segura. Además, cliente y servidor deben ponerse de acuerdo en los algortimos a usar para implementar firma digital. Una posible combinación es RSA para criptografía asimétrica y SHA-3 para el cálculo de hashes.
   2. `http://api.securityfirst.com/?method=getPrice&fruit=Pera&signature=FIRMA&clientId=33`
   3. El cliente debe realizar un hash con el valor de los parámetros method, fruit y clientId (**Integridad**). Ese hash debe ser usado para la firma digital (**No repudio**), para eso usa su clave privada. La firma digital se utiliza donde esta la palabra FIRMA en el punto 2. El servidor, al recibir el pedido debe validar la firma digital usando la clave pública del cliente. Esa clave la obtiene buscandola por clientId en la colección de usuarios autorizados que almacena (**Autenticación**). Finalmente, debe tomar el valor de los parámetros method, fruit y clientId para calcular el hash, compararlo con lo incluido en la firma digital y asi validar integridad.

3.3. Notas finales

a. Pero... ¿Qué sucede si alguien conoce la nueva URL propuesta?. La puede ejecutar cuantas veces quiera, como si estuviera realizando nuevas peticiones para conocer el precio de un producto. Si por cada ejecución del método la empresa Security First cobra un valor por la operación alguien pudiera generar costos para los clientes, que estos pudieran negar no se cumple el (NO REPUDIO). Esto se conoce como (ataque de repetición). Para su estudio independiente los invitamos a buscar más información al respecto en $https://en.wikipedia.org/wiki/Replay_attack$. Como dijimos al principio siempre debemos estar alertas: La seguridad consiste en hacer que el riesgo se reduzca a niveles aceptables, debido a que el riesgo es inherente a cualquier actividad y nunca puede ser eliminado.