Fase I

Carlos Zamora Sanz
Juan Soliveres Olivares
Abdel Awati Lasshab

Descripción

· Aplicación para la compartición de archivos multimedia

· Aplicación de escritorio

· Implementación en Java

· Almacenamiento de los archivos en la nube mediante la API de Google Drive

· La implementación se realizará en NetBeans

Cliente Java

· Se solicitará Usuario/Contraseña

· Se seleccionarán los ficheros a conservar

· Se realizará el cifrado y formato de compresión de los ficheros

· Se seleccionará la política de copias de seguridad

· Se desconfiará totalmente del servidor (conocimiento 0)

Servidor Java

· Conservará los datos referentes a cada usuario de forma segura

· Almacenará los ficheros de forma segura

· Posteriormente realizará las llamadas necesarias a la API de Google Drive

Algoritmos a utilizar

· **AES** con una clave de 256 bits

· **RSA** con una clave de 2048 bits

· **SHA-3** con 512 bits como función HASH

Notación

· Utilizaremos $L_K(a)$ para definir los primeros K bits de la cadena binaria a

·Utilizaremos $R_K(a)$ para definir los últimos K bits de la cadena binaria a

·En ambos casos se mantiene el orden original de la cadena

Registro

Usuario

HASH(Password + SAL)

KU_{PuB}^{RSA}

$E_{D_{256}(HASH(Pass+SAL))}^{AES}(\boldsymbol{KU_{Priv}^{RSA}})$

Enviamos

Almacenamos:

Usuario

L_{256}(HASH(Password + SAL))

SAL

KU_{PuB}^{RSA}

$E_{D_{256}(HASH(Pass+SAL))}^{AES}(\boldsymbol{KU_{Priv}^{RSA}})$

Login I

Usuario

Enviamos

Usuario

Almacenamos:

L_{256}(HASH(Password + SAL))

SAL

KU_{PuB}^{RSA}

SAL

Recibimos

$E_{D_{256}(HASH(Pass+SAL))}^{AES}(\boldsymbol{KU_{Priv}^{RSA}})$

Login II

Usuario

L_{256}(HASH(Password + SAL))

Comparamos

Usuario

L_{256}(HASH(Password + SAL))

Cifrado de ficheros

K_{File}^{AES}

Generamos: $E_{K_{File}^{AES}}^{AES}(\mathbf{File})$

$E_{KU_{PuB}^{RSA}}^{RSA}(\boldsymbol{K_{File}^{AES}})$

Enviamos

Almacenamos: $E_{K_{File}^{AES}}^{AES}(\mathbf{File})$

$E_{KU_{PuB}^{RSA}}^{RSA}(\boldsymbol{K_{File}^{AES}})$

Drive

Descifrado de ficheros

Desciframos:

$$D^{RSA}_{KU^{RSA}_{Priv}}[\boldsymbol{E}^{\boldsymbol{RSA}}_{\boldsymbol{KU}^{\boldsymbol{RSA}}_{\boldsymbol{PuB}}}(\boldsymbol{K}^{\boldsymbol{AES}}_{\boldsymbol{File}})] = K^{AES}_{File}$$

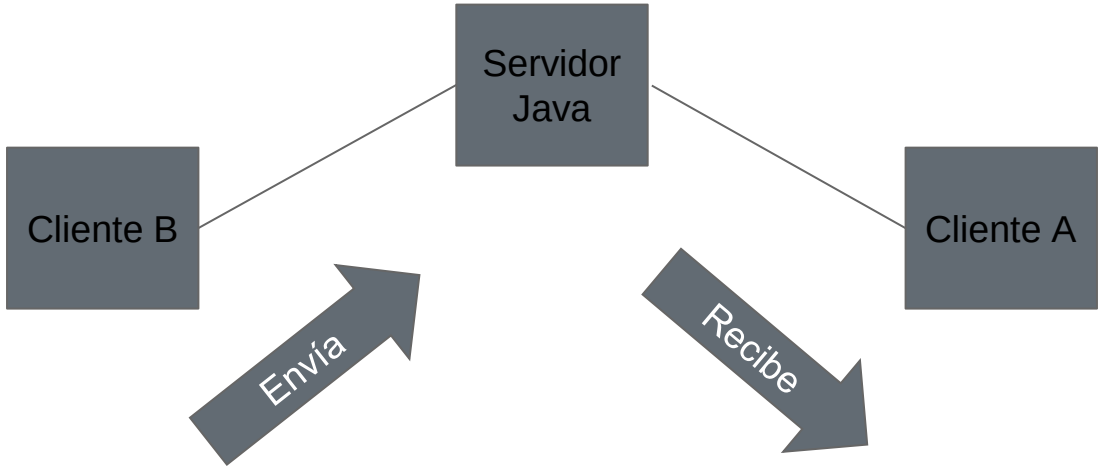
$$D^{AES}_{K^{AES}_{File}}[\boldsymbol{E}^{\boldsymbol{AES}}_{\boldsymbol{K}^{\boldsymbol{AES}}_{\boldsymbol{File}}}(\textbf{File})] = \text{File}$$

Recibimos

$$E^{AES}_{K^{AES}_{File}}(\textbf{File})$$

$$E^{RSA}_{KU^{RSA}_{PuB}}(\boldsymbol{K}^{\boldsymbol{AES}}_{\boldsymbol{File}})$$

Compartición de ficheros



$$E^{RSA}_{KA^{RSA}_{PuB}}(\boldsymbol{K^{AES}_{File}})$$

Cifrado con la clave pública de A

$$D^{RSA}_{KA^{RSA}_{PuB}}[\boldsymbol{E^{RSA}_{KA^{RSA}_{PuB}}}(\boldsymbol{K^{AES}_{File}})] = K^{AES}_{File}$$

$$D^{AES}_{K^{AES}_{File}}[\boldsymbol{E^{AES}_{K^{AES}_{File}}}(\mathbf{File})] = \text{File}$$

Librerías a utilizar

Seguridad

· Utilizaremos java.security.*

· Utilizaremos java.crypto.*

· Se añadirán futuras librerías conforme a las necesidades del proyecto

Interfaz

· Utilizaremos java.awt.*