REDES DE COMUNICACIONES II

Práctica 2 - Seguridad y criptografía

Volver a: Prácticas

2 Funcionalidad

Como ya se ha comentado, el cliente del servicio SecureBox deberá interaccionar con el servidor, con el fin de permitir subir y descargar archivos firmados y cifrados.

Para ello, el cliente deberá permitir:

- Gestionar (importar, listar y borrar) una identidad digital asociada a un único token de autenticación.
- Cifrar y firmar un archivo, dirigido a algún otro usuario del sistema.
- Enviar un archivo al servicio SecureBox.
- Recibir un archivo almacenado en SecureBox, comprobando tras su descarga su firma digital, y descifrando su contenido.

Para poder hacer estas operaciones criptográficas, es necesario crear una identidad digital previa, que nos proporcione un par de claves pública y privada.

Funcionalidad del cliente

Las opciones de la línea de comandos que debe soportar el cliente son:

Opción	Descripción
Gestión de usuarios e identidades	
--create_id *nombre email* [*alias*]	Crea una nueva identidad (par de claves púlica y privada) para un usuario con nombre *nombre* y correo *email*, y la registra en SecureBox, para que pueda ser encontrada por otros usuarios. *alias* es una cadena identificativa opcional.
--search_id *cadena*	Busca un usuario cuyo nombre o correo electrónico contenga *cadena* en el repositorio de identidades de SecureBox, y devuelve su ID.
--delete_id *id*	Borra la identidad con ID *id* registrada en el sistema. Obviamente, sólo se pueden borrar aquellas identidades creadas por el usuario que realiza la llamada.
Subida y descarga de ficheros	
--upload *fichero*	Envia un fichero a otro usuario, cuyo ID es especificado con la opción --dest_id. Por defecto, el archivo se subirá a SecureBox firmado y cifrado con las claves adecuadas para que pueda ser recuperado y verificado por el destinatario.
--source_id *id*	ID del emisor del fichero.
--dest_id *id*	ID del receptor del fichero.
--list_files	Lista todos los ficheros pertenecientes al usuario
--download *id_fichero*	Recupera un fichero con ID *id_fichero* del sistema (este ID se genera en la llamada a *upload*, y debe ser comunicado al receptor). Tras ser descargado, debe ser verificada la firma y, después, descifrado el contenido.
--delete_file *id_fichero*	Borra un fichero del sistema.
Cifrado y firma de ficheros local	
--encrypt *fichero*	Cifra un fichero, de forma que puede ser descifrado por otro usuario, cuyo ID es especificado con la opción --dest_id.
--sign *fichero*	Firma un fichero.

--enc_sign *fichero*	Cifra y firma un fichero, combinando funcionalmente las dos opciones anteriores.

Ejemplos de uso

Imaginemos dos usuarios, de nombres Pablo López, con ID GGHHII y Antonio Chicharro, de ID XXYYZZ, desean enviarse un archivo de forma segura a través de SecureBox. Suponiendo que ambas identidades, por supuesto, han sido previamente creadas y registradas en el sistema, la secuencia de comandos que deberían seguir es similar a la siguiente:

```
# python securebox_client.py --search_id Antonio
Buscando usuario 'Antonio' en el servidor...OK
3 usuarios encontrados:
[1] Antonio Chicharro, antonio.chicharro@estudiante.uam.es, ID: XXYYZZ
[2] Antonio López, antonio.lopez@estudiante.uam.es, ID: AABBCC
[3] Juan Antonio Barrera, juanantonio.barrera@estudiante.uam.es, ID: DDEEFF
```

En este momento sería necesario asegurarse de que el ID del Antonio que buscamos es XXYYZZ, confirmándolo con él por otro medio. Una vez satisfechos, podríamos ya enviarle un fichero a través de SecureBox:

```
 # python securebox_client.py --upload foto1.jpg --dest_id XXYYZZ
Solicitado envio de fichero a SecureBox
-> Firmando fichero...OK
-> Recuperando clave pública de ID XXYYZZ...OK
-> Cifrando fichero...OK
-> Subiendo fichero a servidor...OK
Subida realizada correctamente, ID del fichero: AABBCCDDEEFF
```

A partir de este momento, el fichero está disponible en el servidor para que sea recuperado por Antonio. Para ello, tendríamos que comunicarle el ID del fichero, AABBCCDDEEFF, para que el pudiera recuperarlo con un comando similar al siguiente:

```
# python securebox_client.py --download AABBCCDDEEFF --source_id GGHHII
Descargando fichero de SecureBox...OK
-> 33245 bytes descargados correctamente
-> Descifrando fichero...OK
-> Recuperando clave pública de ID GGHHII...OK
-> Verificando firma...OK
Fichero descargado y verificado correctamente
```

Volver a: Prácticas