Guia de Usuario

Proyecto Criptografia

March 2025

1 Requisitos Previos

Antes de ejecutar el programa, asegúrate de contar con los siguientes elementos:

- Python 3.9 o superior instalado en tu sistema.
- **Módulos de Python:** Asegúrate de tener instalados los siguientes módulos de Python:
  - **ssl:** Para manejar conexiones seguras.
  - **socket:** Para manejar conexiones de red.
  - **threading:** Para manejar múltiples conexiones simultáneamente.
  - **uuid:** Para generar identificadores únicos.
  - **json:** Para manejar archivos de configuración en formato JSON.

  Puedes instalar los módulos necesarios utilizando pip:
  `pip install ssl socket threading uuid json`
- Un archivo JSON con la configuración de conexiones.
- Un certificado SSL (cert.pem) y una clave privada (key.pem) para la comunicación segura.

2 Configuración del Archivo de Conexiones

El programa utiliza un archivo JSON para configurar los usuarios y su conexión.

Guarda este archivo JSON con un nombre como **connections.json**.

- **host:** Define la configuración del host principal, incluyendo la dirección IP, el puerto y un identificador único (UUID).
- **users:** Define una lista de usuarios con los que el host se conectará. Cada usuario tiene una dirección IP, un puerto y una lista de números que se utilizarán en el programa.

3 Ejecución del Programa

Paso 1: Inicializar el Host Principal

Para iniciar el host principal, crea un objeto ConnectionsFile y un MainUser:

```
from ConnectionsFile import ConnectionsFile

import NetworkUser

config = ConnectionsFile("connections.json") # Carga el archivo JSON

host = config.create_host() # Crea el usuario principal
```

Paso 2: Conectar con los Usuarios

Luego, conecta el host con los usuarios definidos en el archivo JSON:

```
config.connect_with_users(host)
```

El programa intentará conectar con cada usuario definido en la lista users. Si hay algún problema, mostrará mensajes de error.

4 Funcionalidad del Programa

Protocolos de Comunicación

El programa maneja varios protocolos para la transmisión de datos entre los usuarios de la red:

1. `REQUEST_CONNECTION`: Solicita una conexión a otro usuario.
2. `ACCEPT_CONNECTION`: Acepta una conexión entrante.
3. **MESSAGE**: Envia un mensaje de texto entre usuarios.
4. `INPUT_SHARE`: Comparte un valor secreto utilizando el protocolo de Shamir.
5. `PRODUCT_SHARE`: Comparte el resultado de una operación de multiplicación entre valores secretos.
6. `FINAL_SHARE`: Envia la parte final de un cálculo entre los usuarios conectados.

Cada protocolo tiene funciones `send_message()` y `receive_message()` para manejar el envío y recepción de datos.

Envío de Datos Secretos

Para enviar un número secreto a los usuarios conectados, se usa la función:

```
host.send_number(42) # Envía el número 42 a los usuarios usando el protocolo de Shamir
```

También se puede especificar un protocolo específico:

```
from NetworkProtocol import FinalShareProtocol
host.send_number(99, FinalShareProtocol)
```

Reconstrucción del Secreto

Una vez recibidas todas las partes necesarias, un usuario puede reconstruir el secreto:

```
secreto_reconstruido = host.reconstruct_secret()
print(f"Secreto final: secreto_reconstruido")
```

5 Posibles Errores y Soluciones

Si encuentras algún problema al ejecutar el programa, revisa la siguiente tabla con los errores más comunes y sus posibles soluciones:

Error	**Posible Causa**	**Solución**
`Error al leer el archivo JSON`	Archivo `connections.json` mal estructurado o inexistente.	Verifica la sintaxis JSON y que el archivo exista en el directorio correcto.
`No se ha definido el host.`	El archivo JSON no tiene una sección `host` válida.	Revisa que la sección `host` en el JSON contenga `ip`, `port` y `uuid`.
`No se pudo conectar con <IP>:<Puerto>`	El usuario no está en línea o el puerto está cerrado.	Asegúrate de que el usuario remoto esté ejecutando el programa y que el puerto esté disponible.
`Se agotaron los intentos de reconexión.`	Problema de conexión persistente.	Verifica la configuración de red, el firewall y que el host remoto esté accesible.