General information

This version employs single unified codebase to implement LSABE-MA client and LSABE-MA server.

LSABE-MA server is implemented using flask https://flask.palletsprojects.com/en/2.0.x/quickstart/ . Please install it with pip at the system where server will be running: pip install flask

LSABE-MA client is based on requests module https://docs.python-requests.org/en/master/

Please install it with pip at the system where client will be running: pip install requests

The server

Functions

The server implements the following functions:

1. **Store** – receive the cyphertext over REST API and store it to memory cash and local file storage
2. **Search** – receive the trapdoor and transformation key over REST API, apply trapdoor algorithm to all messages in the memory cash, apply transformation algorithm to all matching messages, return the list of partially description messages to the client
3. **Clear-messages** – deletes all messages from memory cash and local file storage.
4. **Global-setup** – receive MSK and PP over REST API and store them. Note: server has encoded default MSK and PP that match client default MSK and PP. This call is optional.
5. **Authority-setup** – receive authority secret key, public key and attributes over REST API and store them. Note: server has encoded default values for authority with id=1 that match client defaults. This call is optional.
6. **On startup** the server loads cyphertexts from local file storage to memory cash.

The server is implemented using LSABE_MA and LSABE_AUTH classes delivered earlier. Some additional features were added to serialization and deserialization, but the core was left intact.

Running

In the package root folder LSABE-MA-2 run the following commands:

```
export FLASK_APP=lsabe_ma_srv
export FLASK_RUN_PORT=5000              ← This is the port the server will be using
export FLASK_RUN_HOST=0.0.0.0           ← This means ‘bind to all network interfaces’
python -m flask run
```

Please ensure that security settings, firewall, antivirus do not block network traffic. It is also possible to run client and server on the same computer using local host interface:

```
export FLASK_APP=lsabe_ma_srv
python -m flask run
```

This will bind server to default interface 127.0.0.1:5000

The client

Functions

The client is based on lsabe_ma application delivered earlier. Several methods are now accepting additional –url parameter. If this parameter is provided, the application won't use local storage but rather call the server over REST API.

Client code includes default MSK, PP and authority secret key, public key and attributes for authority with id=1. They match default values for the server so initioal setup may be omitted.

Running

Suggested initial testing sequence:

```
python -m lsabe_ma --keygen --authority-id 1 --sec-attr "attribute-1"  --GID "user-1"
```

```
python -m lsabe_ma --encrypt --authority-id 1 --msg "Searchable encryption is good" --kwd Searchable encryption --url http://127.0.0.1:5000
```

```
python -m lsabe_ma --encrypt --authority-id 1 --msg "This is unrelated message" --kwd unrelated message --url http://127.0.0.1:5000
```

```
python -m lsabe_ma --search --authority-id 1 --GID "user-1" --kwd Searchable --url http://127.0.0.1:5000
```

```
python -m lsabe_ma --search --authority-id 1 --GID "user-1" --kwd ENCRYPTION --url http://127.0.0.1:5000
```

The following call will execute bulk encryption of messages from the file 100.txt. Each line shall contain comma-separated message text and keywords

```
python -m lsabe_ma –bulk-encrypt 100.txt --authority-id 1 --url http://127.0.0.1:5000
```

Please note that –url parameter **must** include protocol keyword (http)