Reminisce-to-Rescue

This repository includes the implementation of the R2R framework.

NOTICE: This is an academic proof-of-concept prototype and has not received careful code review. This implementation is NOT ready for production use.

Dependencies

To build R2R, one needs to install the following:

- Python v3.10
- Streamlit framework v1.30 (pip install streamlit)

To run locally

- Run command streamlit run query_interface.py

Set Bloom Filter Parameters

Under operation Manage Parameter, users can choose to set Bloom filters parameters mentioned below:

- Choose the type of hash function used to insert/lookup from the Bloom filter (SHA256 or SHA512), from the dropdown menu.
- Choose the number of security questions for encoding the private key, from the dropdown menu. Users can choose between 5 to 12 security questions. Note: If no values are updated, default values are considered.

Store Private Key into Bloom Filter

Under operation Store Private key, user can set below fields:

- Choose the security questions from the dropdown menu and fill in the memorized secret.
- Enter the private key required to be stored in Bloom filter, in the text box.
- Submit to store the private key and get information about the Bloom filter content, its size, and number of hash functions used, in the text area.

Retrieve Private Key from Bloom Filter

Under operation Retrieve Private key, user can set below fields:

- Fill the memorized secrets for the security questions chosen while storing the private key.
- Submit to retrieve the private key.

Note: Make sure to press enter after entering any details in the textbox.

False Positive Experiment

- To avoid false positives using padded ones, e.g, "1111" set padding_flag= True in the main.py file. Set padding value using padding variable.
- To avoid false positives using concatenated memorized secret answers, set padding_flag= False in the main.py file.