Explanation of the functions used in RHA-512 Hashing:

1. **hashlib**: This is a Python library that provides various cryptographic hash functions, including SHA-512. We import it at the beginning of the script.
2. **calculate\_sha512(file\_path)**: This is a custom function that calculates the SHA-512 hash of a file. It takes the path to the file as an argument. Here's what this function does step by step:

a. **hashlib.sha512()**: Creates an SHA-512 hash object.

b. **with open(file\_path, 'rb') as file**: Opens the specified file in binary read mode ('rb') as a context manager, ensuring that the file is properly closed after use.

c. Inside the **with** block, we read the file in chunks of 8KB (8192 bytes) at a time using **file.read(8192)**. This is done to efficiently handle large files without loading the entire file into memory.

d. **sha512\_hash.update(data)**: Updates the SHA-512 hash object with the data read from the file. This operation is performed iteratively as each chunk of data is read.

e. The loop continues until there is no more data to read from the file.

1. **return sha512\_hash.hexdigest()**: After reading and updating the hash object with the entire file, this function returns the hexadecimal representation of the SHA-512 hash.
2. Finally, the script calls the **calculate\_sha512** function with the path to the file you want to hash, and it prints the resulting SHA-512 hash to the console.

This code ensures that the SHA-512 hash is calculated efficiently and accurately for the specified file.

Top of Form