Make sure to save all the scripts in the same directory and set the necessary permissions (e.g., chmod +x script\_name.sh) to make them executable.

To run the full scenario and test cases:

1. Run the create\_test\_files.sh script to create the test files.
2. Run the install\_tools\_dependencies.sh script to install the necessary tools and dependencies.
3. Run the backup.sh script and provide the required parameters (source directory, backup directory, encryption key, number of days).
4. Run the restore.sh script and provide the required parameters (backup directory, restore directory, decryption key).
5. Run the scp.sh script and provide the required parameters (backup directory, remote user, remote host).
6. Enable the incremental backup functionality if needed (update the code in the menu.sh script).
7. Select the appropriate options from the menu script (menu.sh) to perform the desired actions.

Remember to update the script placeholders ("<source\_directory>", "<backup\_directory>", "<encryption\_key>", "<days>", "<remote\_user>", "<remote\_host>") with the actual values.

Please note that the above scripts provide a framework for implementing the backup, restore, and SCP functionality. You will need to fill in the actual logic within the functions to perform the desired operations according to your specific requirements.

SCP function

1. Ensure that you have SSH access to the remote server and have the necessary permissions to copy files to the destination directory.
2. Replace <remote\_user> with the actual username of the remote server.
3. Replace <remote\_host> with the hostname or IP address of the remote server.
4. Replace /path/to/remote/backup/directory/ with the actual path to the destination directory on the remote server where you want to copy the backup.
5. Run your backup script (backup.sh) with the appropriate command-line parameters.
6. Monitor the output of the script and check for any error messages related to the scp command.
7. After the script execution completes, verify that the backup file (file\_tar\_file.gpg) is copied to the specified remote server and the destination directory.
8. Optionally, you can manually connect to the remote server and check if the backup file is present in the correct location.

By following these steps, you can test the scp function in your backup script and ensure that the backup file is successfully copied to the remote server.