1. Identifying private information

Note: For simplification, any of the private information listed here does not store on server side.

A. Use storage as an owner (Main Folder - Application_Data\Server_Directory_Data)

I. Using privacy based MFA (Main Folder\Storage_ID\MFA_Device)

-> All data stored in MFA_Device folder should be consider as private data and they are responsible in making sure the privacy based MFA works.

II. Encrypted Files Component (Main Folder\Storage_ID\Encrypted_Files)

-> When you navigate through the folder, you should see a lot of random directory names, those are random file name that send to the server instead of original file name.

i. Encryption Keys & ED25519SK

-> Any folder that has the name of “Keys” or “ED25519SK”, that folder and its corresponding items should be consider as private data and these data was used to perform encryption or decryption on a certain file.

ii. File Name

-> Within the subfolder, you should be seeing a file named “FileName.txt” or “FileName”. This file is responsible for storing the original selected file name and its extension.

iii. ED25519PK

-> Any folder that has the name of “ED25519PK”, that folder and its corresponding items is public. However, server does not hold it as server is not the verifier. This information though it’s not private, you will need to do backup else decryption of file can’t be done properly.

III. Storage ED25519SK (Main_Folder\Storage_ID)

-> Any file that has “rootSK” is consider as a private data. This data was responsible for authenticating with the server.

B. Use storage as an outsider (Main Folder – Application_Data\Other_Directory_Data)

I. Access ED25519SK (Main_Folder\Storage_ID)

-> Any file that has “SK” is consider as a private data. This data was responsible for authenticating with the server.

2. Identifying public information

Note: The information that store on server side is all publicly disclosable data.

A. Storage_ID

-> The Storage_ID typically comes after either Application_Data\Server_Directory_Data or Application_Data\Other_Directory_Data, these data are publicly disclosable data.

B. Random File Name (Random_File_ID)

-> Any random file name typically comes after either Application_Data\Server_Directory_Data\Storage_ID\Encrypted_Files or Application_Data\Other_Directory_Data\Storage_ID\Encrypted_Files, these data are publicly disclosable data.

C. MFA_Device_ID

-> Any random file name typically comes after either Application_Data\Server_Directory_Data\Storage_ID\MFA_Device or Application_Data\Other_Directory_Data\Storage_ID\MFA_Device, these data are publicly disclosable data.

D. Access_ID

-> Any content resides within either Application_Data\Server_Directory_Data\Storage_ID\Allowed_User\ or Application_Data\Other_Directory_Data\Storage_ID\Access_ID.txt is/are publicly disclosable data.

E. ED25519PK or PK

-> Any content resides within the “ED25519PK” folder or any content resides within the file that contains the name of “PK” is all publicly disclosable data.

3. System Backup

A. Storage Owner

-> Server_Directory_Data backup that had been provided in the application backups all the data reside after Application_Data\

Server_Directory_Data\ folder and turns it into a zipped file into any selected destination.

-> Encryption Keys backup that had been provided in the application backups all the encryption keys that was used in encrypting/decrypting the files.

-> There's a selection in which you can choose to not include the "ED25519SK" which was used in signing the encrypted file content. You need to choose this selection if you wished to let others able to decrypt the file content but not signing the file.

-> File Name backup that had been provided in the application backups all original selected file name. The application by default sends a random file name instead of the original file name to the server. Backing up the file name is essential else during decryption of file, it won't decrypt properly.

B. Other Users

-> The same thing applies to other users. However, the need in backing up is not really needed compare to storage owner.

4. Online backup options (relies on problematic and occasionally weak passwords)

Note: These "no-access" storage is not really "no-access" as you hand in to the provider your password so that they can encrypt/decrypt data on your behalf. The thing that's worth mentioning here is that passwords will be deemed to get leaked. Hence, it's only a matter of time before your passwords are in there regardless if they are properly deal with then store in database.

For more details refers to this link in checking if your passwords/credentials have been leaked. (https://haveibeenpwned.com/Passwords)

A. Sync.com

B. Storj

5. Online backup options (WhatsApp/Signal/Session/Threema)

Note: You can use these platforms in storing your sensitive and private information that was used in the storage. However, it's not recommended to put too much stress onto these platforms as they are not designed for sharing videos, files and images.

If you overburden them, they will need to find other ways to scale up.

1st example, WhatsApp turns the users into a product by selling their data to Facebook. Facebook then in turn sells the data to anyone who wants them which could be both a privacy and security concern.

2nd example, Signal introduces a controversial project which is cryptocurrency payment. Not to mention, there's a time in which they go close sourced instead of open sourcing the code. For more information, please watch this video in Youtube. (https://www.youtube.com/watch?v=tJoO2uWrX1M)

Side Note: Open Source refers to anyone can view the code and know how it works and close source refers the exact opposite.

6. Offline backup option

Note: Once you do any system backup, you can go with old style by storing the important and private information into a hard disk or drive. This way you won't be putting burden onto WhatsApp,Signal,Session,Threema. I don't object that you backup through Signal/WhatsApp/Threema/Session but at the least try not to put huge files onto them.