Appendix D Security solution use with classical shared folders

We demonstrate the user-interaction model with the proposed storage security solution in a classical GIS enterprise setup. Users *Scott* and *Alan*, log onto the ECMSDK server and access single-user files in their respective home folders and multi-user files placed in a project directory. These files must be decrypted for use and encrypted back for protection.

Single-user & Multi-user files and privileges: Scott's single-user files within his home folder are eight files (*cpg, dbf, prj, shp, shx, txt, xml and pdf*) of the ***site-of-significance*** shapefile dataset. For brevity, Alan does not have single-user files. There are two sets of multi-user shapefiles; ***nz-bounty-islands*** and ***nz-historic-places*** in the "`O:\projects\NewZealand\`" directory.

Scott should have access to the eight ***single-user*** files of ***site-of-significance*** shapefile dataset in his home folder and to the seven multi-user GIS files of the ***the nz-bounty-islands-polygons-topo-125k*** dataset. Alan should have access to all 14 multi-user files, but not to Scott's eight single user files. Table 2 details these requirements. These files require protection with encryption.

Single-user shapefiles	Privileges	
	Scott	Alan
site-of-significance (only 2 of 8 files need encryption) • site-of-significance.dbf • site-of-significance.shp	☑	
Multi-user shapefiles		
nz-bounty-islands-polygons-topo-125k (7 files)		☑
nz-historic-places (7 files)	☑	☑

Table 2. User privileges on single-user and multi-user shapefiles.

Protocol services: We demonstrate the security solution using network folder sharing via the ECMSDK SMB server. In enterprise GIS, this is the most common mode of accessing spatial data; clients access files on the server through mapped network shares and interact with files through GIS applications on the client's computer. However, working with the security solution using these other protocol servers (e.g., FTP, HTTP, Command line, etc.,) provided by ECMSDK product will be similar, since the DBFS enforces access control at the repository level and the execution of the encryption-decryption procedures depend on DBFS user sessions, that are created when the user uses any protocol service.

Begin with files already encrypted: For brevity, we begin from a point when the single-user and multi-user cryptographic keys for users Scott and Alan have been created and when Scott's single-user files and multi-user files are in encrypted form. Furthermore, the administrator has encrypted all eight multi-user files and user Scott has already specified that all the eight files (***cpg, dbf, prj, shp, shx, txt, xml and pdf***) belonging to the ***site-of-significance*** shapefile dataset need encryption by appending each the filenames with the `_enc` file suffix. In addition, we assume that Scott has subsequently logged out of DBFS upon which the files have been encrypted (see Fig. 1 and 2). Thereafter, we illustrate GIS users Scott and Alan logging to the ECMSDK. Subsequently, their single-user and multi-user shapefiles are decrypted for use, are available for multiple users during their DBFS sessions, and are later encrypted back for protection.

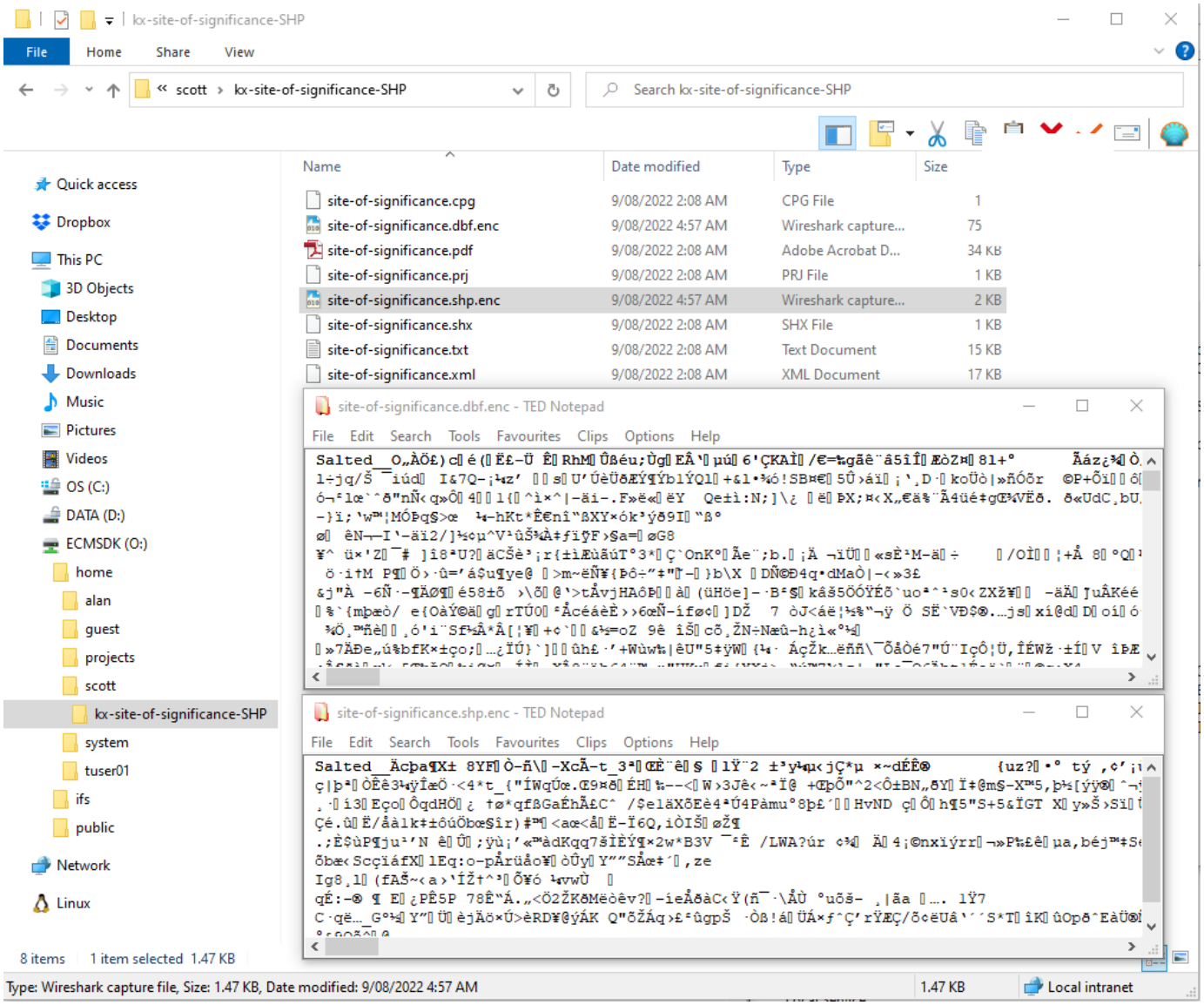

Figure 1: User Scott's single-user files (and multi-user files as shown in Fig 2.) are originally in encrypted form on the ECMSDK server.

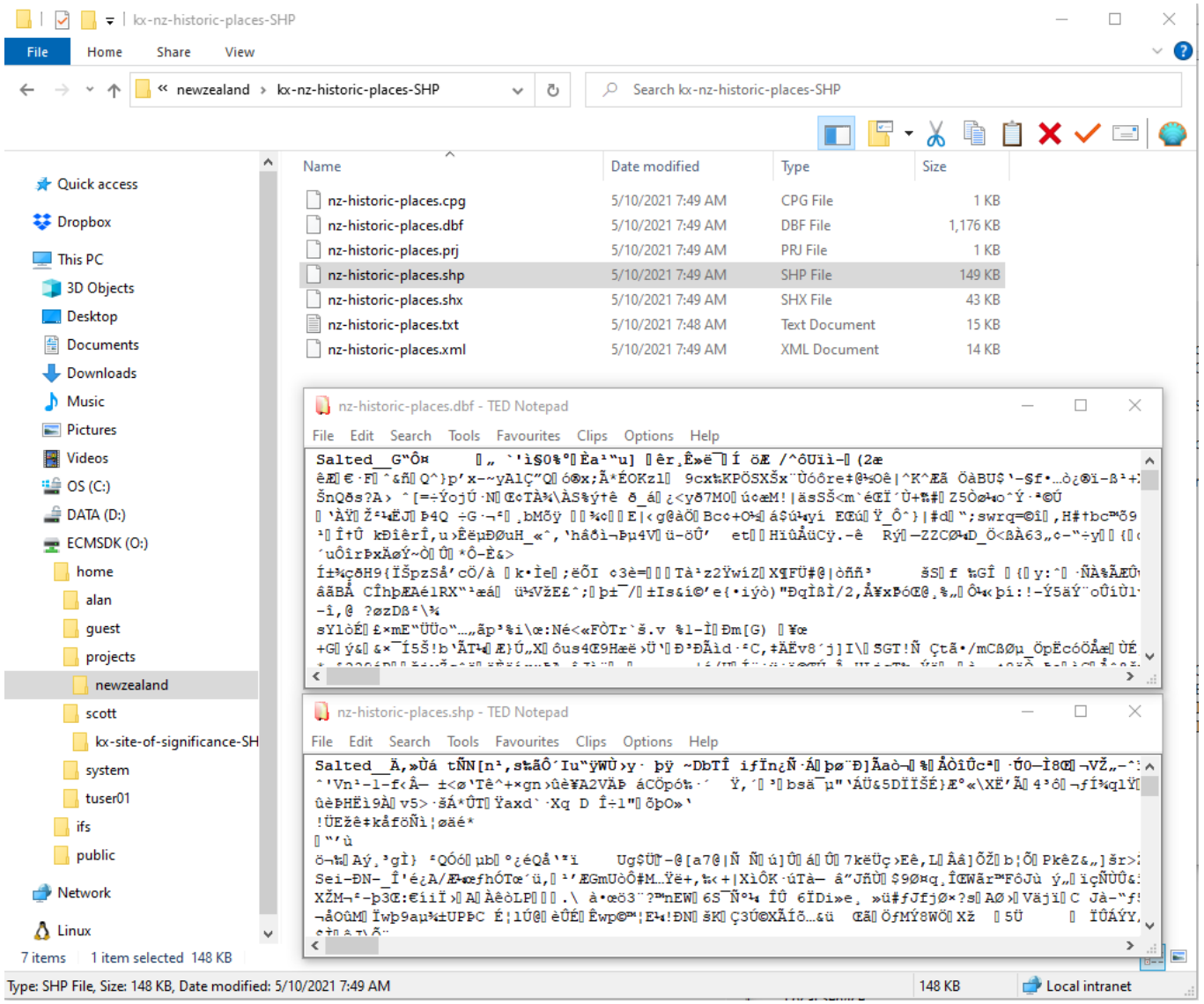

Figure 2: The seven multi-user GIS files to which Scott has privilege are also originally in encrypted form on the ECMSDK server.

Illustrating proposed solution using folder sharing and GIS application

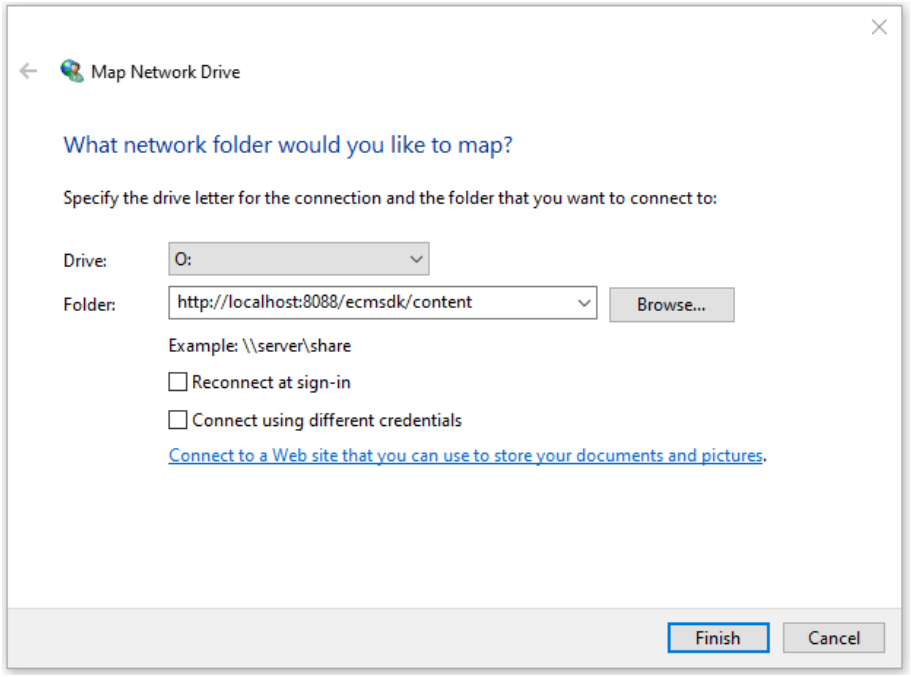

Figure 3: GIS user Scott chooses ECMSDK shared resource to map a network share.

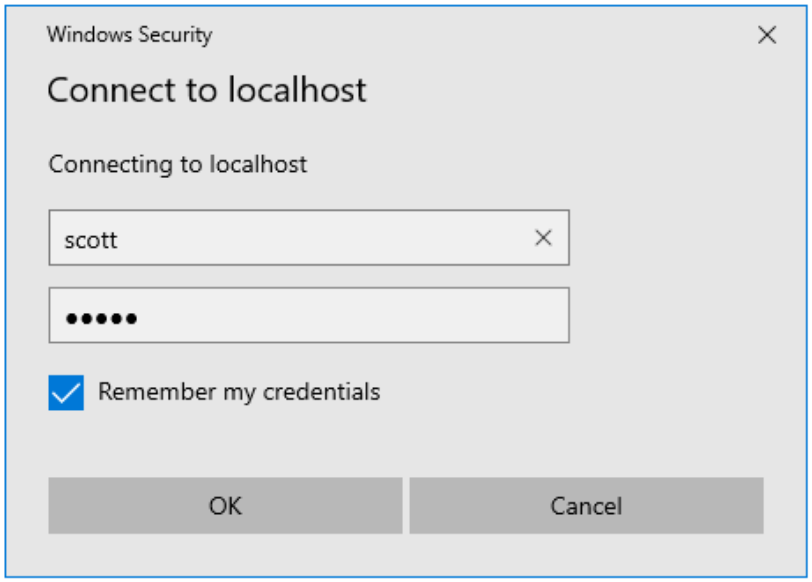

Figure 4: User "Scott" provides credentials for authentication to the ECMSDK protocol server; in this case, it is the ECMSDK SMB server.

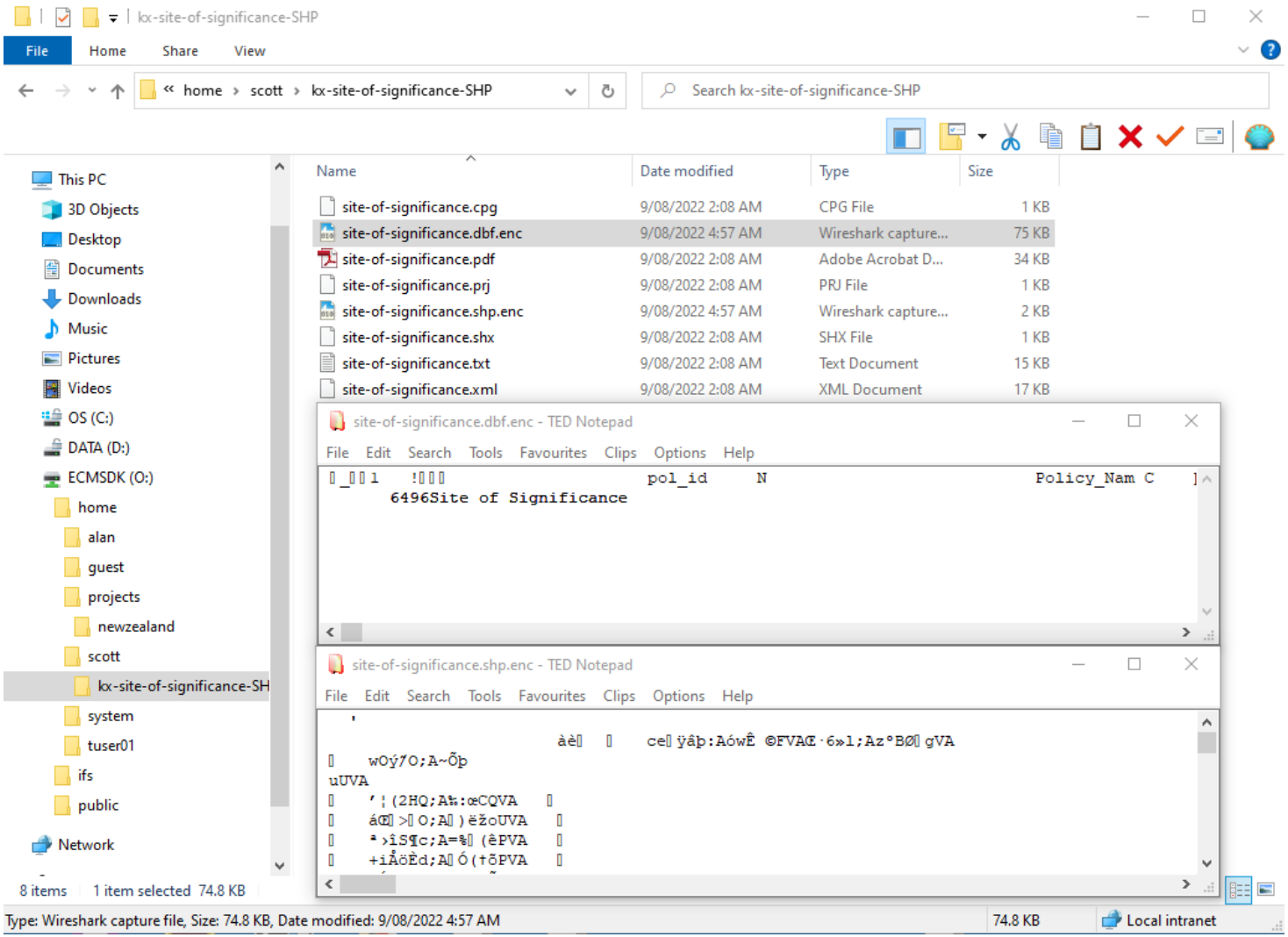

Figure 5: Once user Scott logs in to the ECMSDK, the user session-based trigger is fired, the files are decrypted for use. This includes his eight single-user files (shown decrypted) as well as seven multi-user GIS files to which Scott has privilege (shown decrypted in Fig. 6).

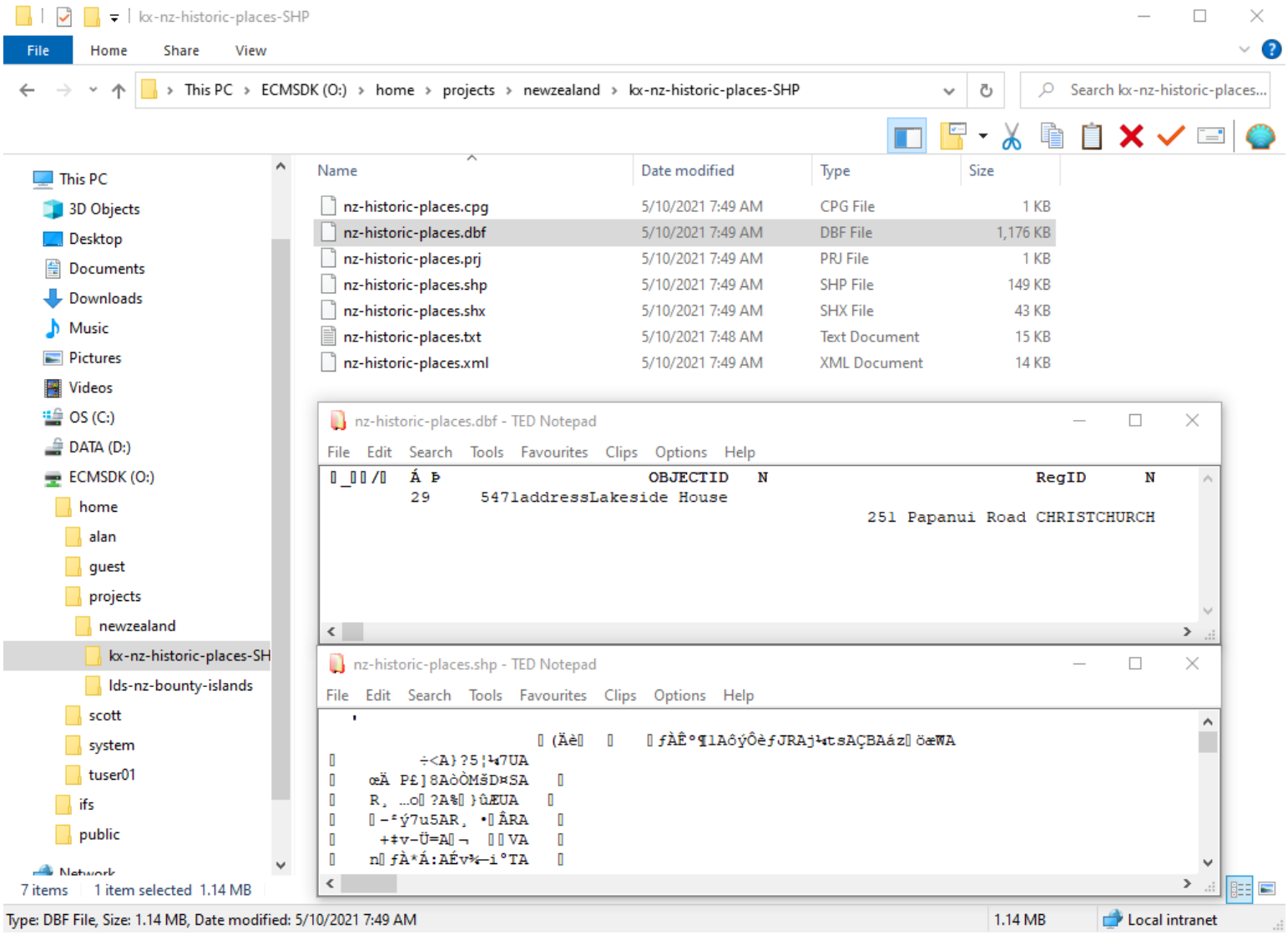

Figure 6: The seven multi-user GIS files to which user Scott has privilege are also decrypted for use. In addition, Scott's actions on other user's folders and files are protected by ECMSDK ACM as Scott cannot access user Alan's home folder which is private.

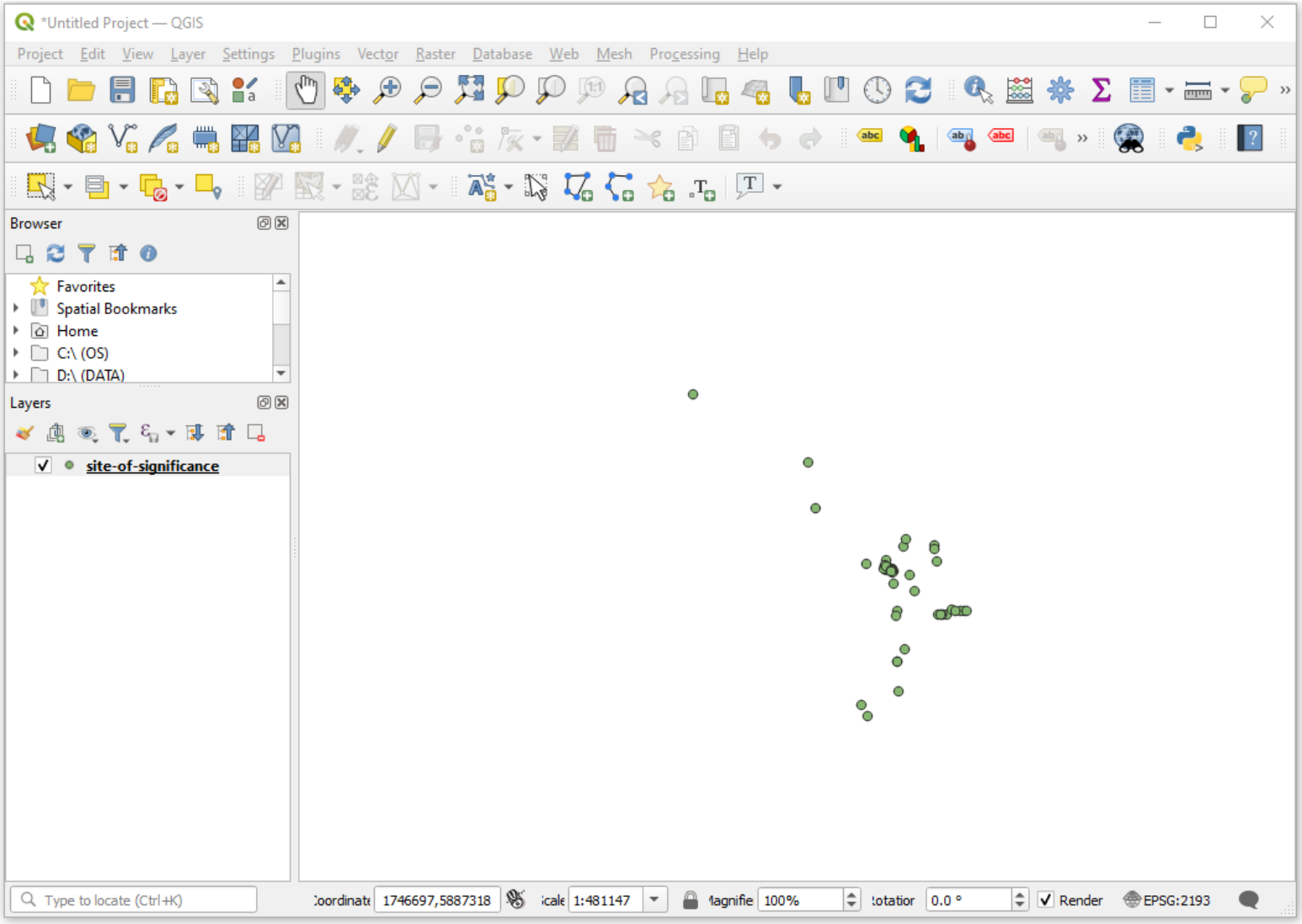

Figure 8: Scott can now remove the ***.enc*** file extension, access, and work with the shapefiles using GIS software such as QGIS.

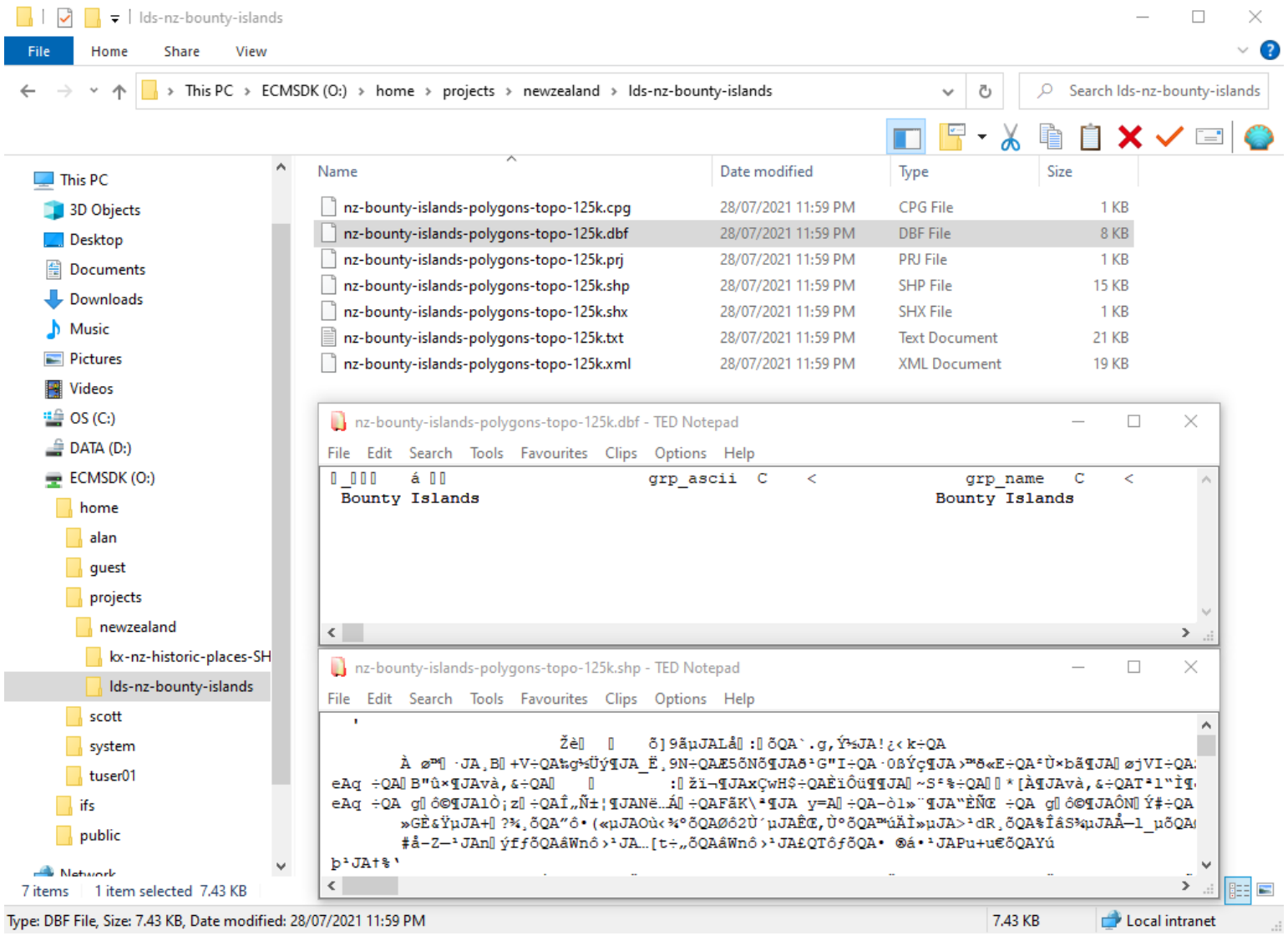

Figure 9: GIS user Alan's login decrypts the remaining seven multi-user GIS files to which only he should have access. The seven multi-user files already decrypted for Scott are not decrypted again. Alan can now access the decrypted contents of the multi-user GIS files.

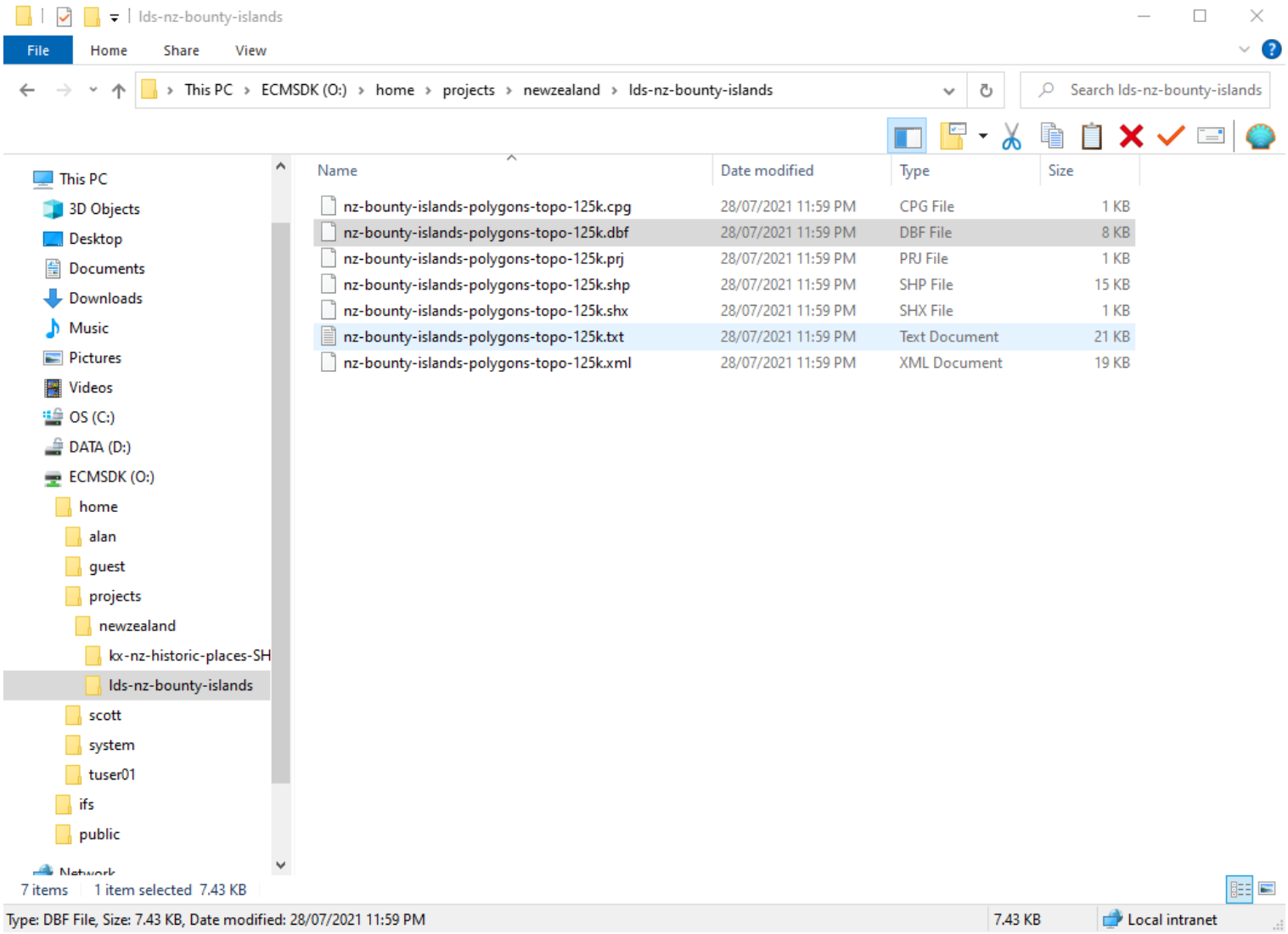

Figure 10: User Alan's logout encrypts only those multi-user GIS files not in use by Scott, as Scott has active DBFS user session. Scott can continue using the seven multi-user files.

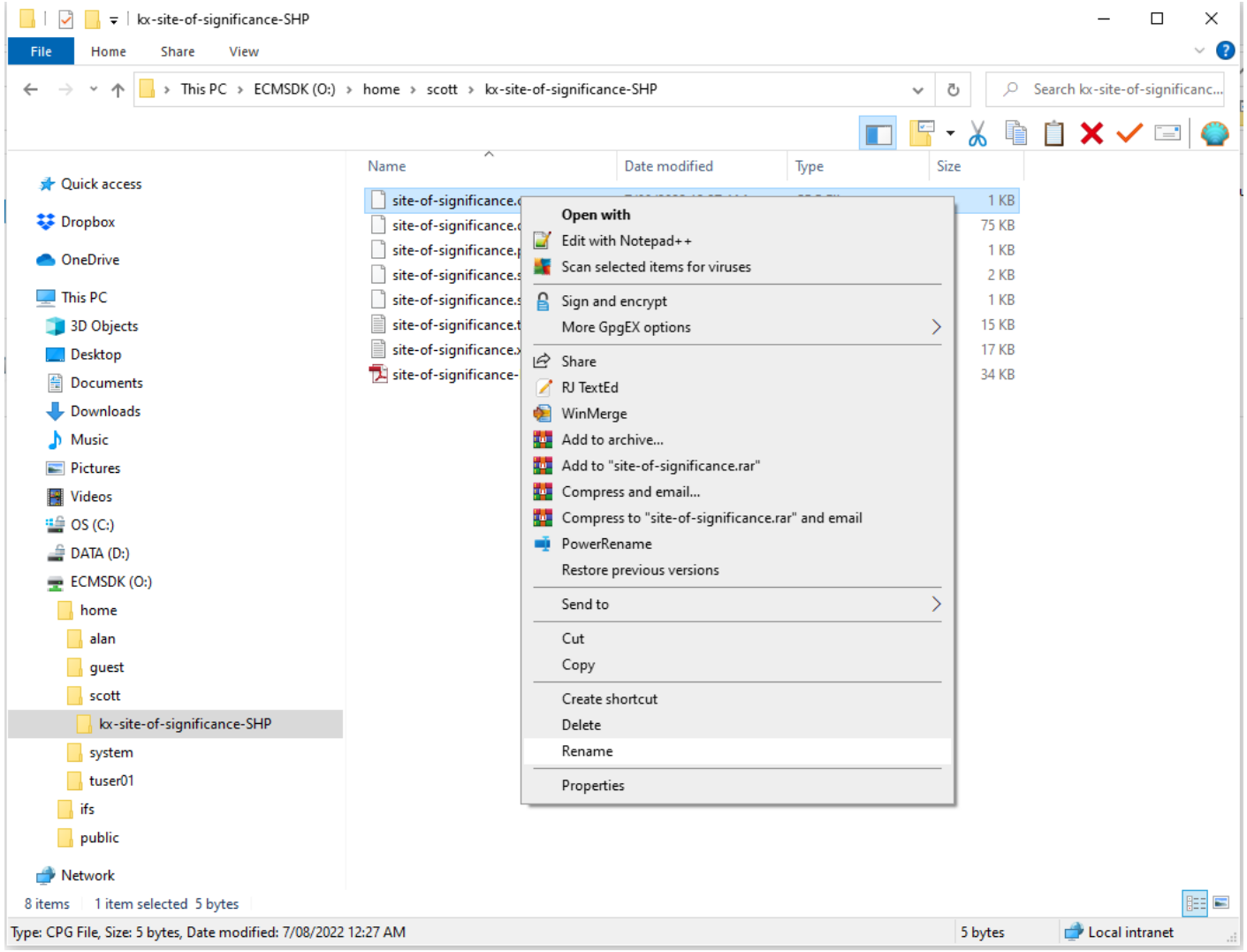

Figure 11: After using the single user files, Scott can specify them for encryption again by adding an *.enc* file extension to them.

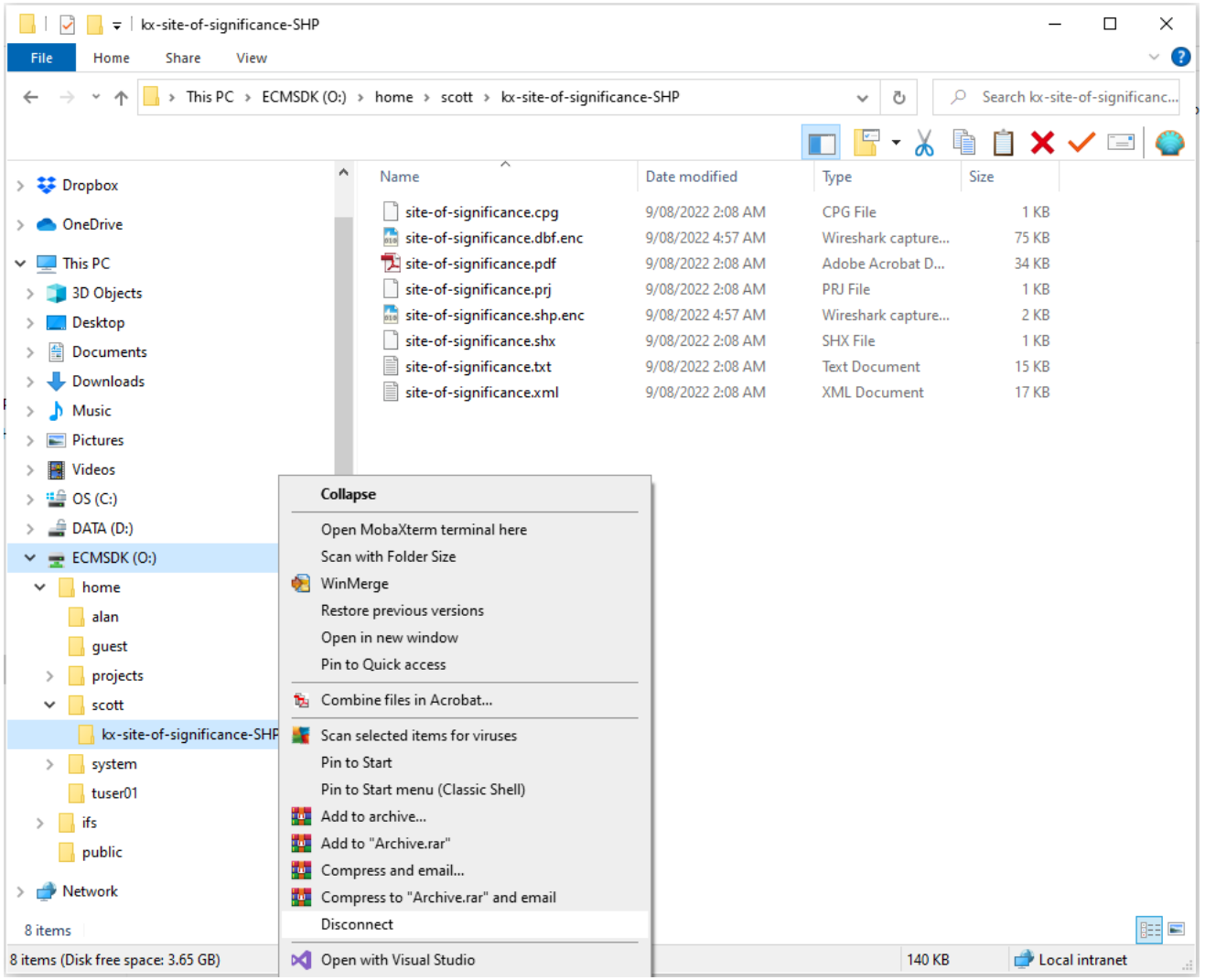

Figure 12: User Scott logs out by disconnecting from the connected ECMSDK shared drive.

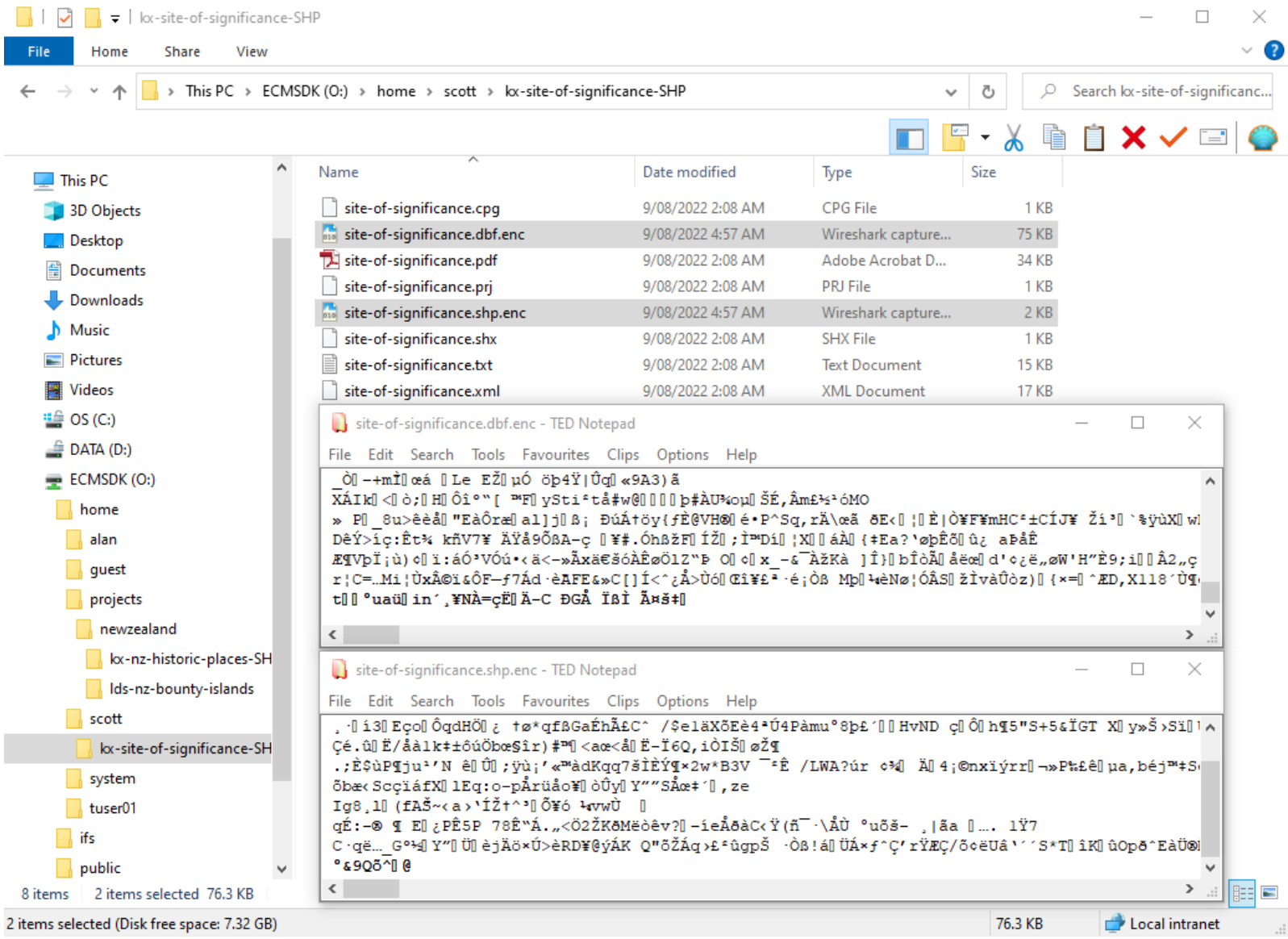

Figure 13: Upon logout, Scott's single-user GIS files are encrypted back for protection.

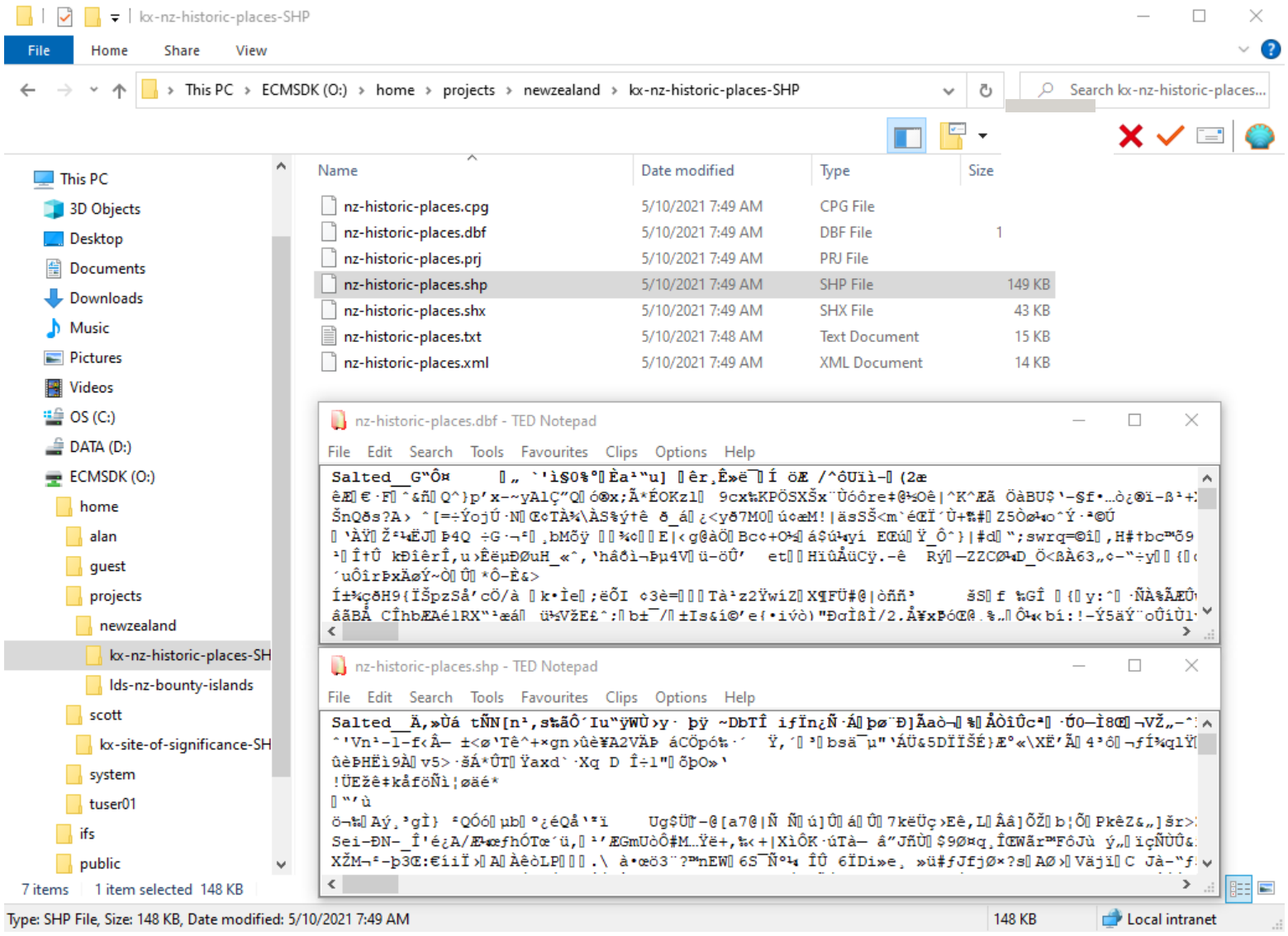

Figure 14: Scott's multi-user GIS files are also encrypted back for protection. The multi-user files are encrypted back only when the last privileged user with active sessions (i.e. Scott) logs out.