Software Diversified Services

1322 - 81st Ave NE
Minneapolis, MN 55432

t 763 571 9000
e info@sdsusa.com
w www.sdsusa.com

SDS E-Business Server™

How to decrypt PGPArchives (*.pga) and Self-Decrypting Archives (*.sda.exe)

PGParchives and self-decrypting archives (SDAs) are for receiving and decrypting data at systems where no E-Business Server software is installed. A PGParchive (*filename*.pga) can be decrypted on z/OS TSO, z/OS USS, Linux, Unix, or Windows® platforms by means of the PGParchive reader, a freely distributable part of the SDS E-Business Server. An SDA (*filename*.sda.exe) can decrypt itself on Windows.

PGParchives and SDAs use *conventional* encryption. The encrypter/sender needs to separately and securely supply a passphrase.

For PGPArchives, the receiver needs to download the reader utility from *www.sdsusa.com/ebusiness/pgparchives*.

Decrypting PGParchives on Linux/Unix

The sender supplies the archive file and the passphrase.

Copy the `ebsreader` file to the Linux/Unix system.

Decrypt with a command like the following:

```
./ebsreader <filename.pga> [<target_directory>] [--overwrite]
```

<*filename.pga*> is the PGParchive to be decrypted.

Output goes to the current directory or to the <*target_directory*>.

`--overwrite` means that the reader will not prompt before overwriting an existing file.

At execution, the reader will prompt for the passphrase.

Decrypting PGParchives on Windows

The sender supplies the archive file and the passphrase.

Copy `EBSreader.exe` to the Windows system.

To decrypt, double-click on the archive file (typically `*.pga`).

Windows will prompt for an opener program. Direct it to `EBSreader.exe`

Answer the reader's prompt for the passphrase.

Then browse the Windows file system to specify an output directory.

Or, at a command line, execute a command like the following:

```
EBSreader.exe <filename.pga> [<target_directory>] [--overwrite] [--no-gui]
```

<*filename*> is the PGParchive to be decrypted.

Output goes to the current directory or to the `<target_directory>`.

`--overwrite` means that the reader will not prompt before overwriting an existing file.

`--no-gui` means the reader will not generate Windows diaologue boxes.

At execution, the reader will prompt for the passphrase.

Decrypting SDAs on Windows

The sender supplies the archive file and the passphrase.

Double-click on the archive file, typically `*.sda.exe`.

Answer the prompt for the passphrase.

Then browse the Windows file system to specify the output directory.

Decrypting PGParchives on z/OS USS

The sender supplies the archive file and the passphrase.

Copy the *pgpreader* file to the z/OS USS system.

Decrypt with a command like the following:

`./pgpreader <filename.pga> [<target_directory>] [--overwrite]`

<*filename.pga*> is the PGParchive to be decrypted.

Output goes to the current directory or to the <*target_directory*>.

`--overwrite` means that the reader will not prompt before overwriting an existing file.

At execution, the reader will prompt for the passphrase.

Decrypting PGParchives in z/OS TSO/batch

The sender supplies the archive file and the passphrase.

Copy the dataset member *READER* into a loadlib, probably by a TSO receive operation.

On TSO, decrypt with a command like the following:

`TSO CALL 'hlq.PGP.LOAD(READER)'`

The READER will prompt for the name of the archive file, then its passphrase.

Note that the output reproduces the original file names, and the first node of a file name becomes the HLQ. Invalid HLQs will cause security and access problems. The problem needs to be avoided at the encryption end, by renaming files.

To decrypt with a batch job, see the example JCL below.

```
//READER  JOB (),'NAME',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID,
//     MSGLEVEL=(1,1)
//*-------------------------------------------------------------*/
//* Sample job to decrypt PGParchive on MVS                     */
//* First statement in SYSIN is PGA file name                   */
//* Second statement in SYSIN is passphrase (case sensitive)    */
//*-------------------------------------------------------------*/
//RDR          EXEC PGM=READER,PARM='2>&1'
//STEPLIB      DD   DISP=SHR,DSN=HLQ.PGP.LOAD
//SYSIN        DD   *
'MY.ARCHIVE.PGA'
hello
/*
```

Moving PGPArchives across platforms

From elsewhere, to z/OS:

- Decryption does not translate ASCII character data to EBCDIC. Any translation needed to be done at the encryption end, before creating the archive.
- Decryption does not translate line breaks from one encoding to another. z/OS will likely not recognize line breaks from ASCII machines, though Windows data decrypted in USS may prove an exception.
- Decryption reproduces original file names. On z/OS TSO/batch, those file names need to obey z/OS TSO naming requirements.
- On z/OS TSO/batch, no directory structure or subdirectory files are preserved. Decryption outputs all the files as sibling datasets.
- On z/OS TSO/batch, original files residing in subdirectories are only restored if they were encrypted with the option `--discard-paths`.

From z/OS, to elsewhere:

- Decryption does not translate EBCDIC character data to ASCII. Any translation needed to be done at the encryption end, before creating the archive.
- Decryption does not translate line breaks from one encoding to another. Other systems will likely not recognize line breaks from z/OS machines.
- An archive created in TSO/batch will not contain any directory structure. All the decrypted output files will be siblings.

Between Windows and Linux/Unix:

- Decryption does not translate line breaks from one encoding to another. After decryption on Unix, character data from Windows will still have Windows line breaks: `0x0D 0x0C` After decryption on Windows, character data from Linux/Unix will still have Linux/Unix line breaks: `0x0C`

PGP Reader, What version?

To learn the version number of a PGParchive reader, enter one of the following commands:

Linux/Unix: `ebsreader --version`

Windows: `EBSreader.exe --version`

z/OS USS: `pgpreader --version`