DATAEXPRESS NONSTOP

DXNS PGP

DXNS PGP is a Guardian based interface to the OpenPGP compliant GNU Privacy Guard (abbreviated GnuPG or GPG) port for Tandem NonStop OSS. The OSS GnuPG implementation was performed by the NonStop community and is a complete and free implementation of the OpenPGP standard. GnuPG runs in the OSS instance on TNS/E type CPUs.

The GnuPG c 1.2.4 port supports the following algorithms:

- Public Key: DSA, ELG, ELG-E, RSA, RSA-E, RSA-S
- Cipher: 3DES, CAST5, BLOWFISH, AED, AES192, AES256, TWOFISH
- Hash: MD5, RIPEMD16, SHA1, SHA256
- Compression: Uncompressed, BZIP, ZIP, ZLIB

DXNS PGP offers two versions:

- A standalone TACL executable version (DXGNUPG)
- A DXNS Pathway server controlled version (DXGNUAPI) – not covered in this document

DXGNUPG – GPG for TACL users

The DXGNUPG TACL version runs as a named process and accepts normal Guardian run commands to encrypt and decrypt files; import and export keys, display the fingerprint information from the key rings, check key signatures, generate keys, and remove keys from the public and private key rings.

DXGNUPG runs in an unattended mode but can also be run by users at a TACL prompt. The goal of the product is to provide an unattended mechanism for employing PGP encryption for Guardian users while eliminating, or minimizing, the need for Guardian users to access the NonStop OSS environment. The product may be run in standalone mode directly from a TACL prompt, from TACL command files, obey files or be started by a Guardian process capable of issuing a run command with the associated startup, assign, and param messages.

Files are stored and secured as normal Guardian Enscribe directory structure files and handed over to the OSS environment for encryption/decryption after which they are handed back to the Guardian system.

DXGNUPG utilizes the Tandem standard OSSTTY terminal emulator program to connect the Guardian based program with GNUPG running in the OSS instance.

DXNS FEATURES

- GUARDIAN BASED INTERFACE TO GNUPG
- OPENPGP COMPLIANT

The OSSTTY terminals are required for communication with GNUPG because the OSS OpenPGP version uses the standard input/output and error devices that are associated with C programs. The OpenPGP code will only communicate with a TTY type device which prevents users from doing input/output handling from unattended TACL programs, for example, or even from with standard NonStop inter-process communication. The OSSTTY program(s) is controlled by DXGNUPG and used to act as input and output terminal devices to GnuPG.

Appropriate command prompts, advisory or error messages are directed to the DXGNUPG home terminal. The encryption and decryption modes in particular are designed to be 'send and forget' with the "encrypt" or "decrypt" command being issued and no user involvement required after that point. In the case of decryption that means the secret passphrase is not entered by the users during a decrypt request. See below for more info on passphrases.

Other commands, such as key import, export, signature checking, key generation, all require user interactions which the DXGNUPG will prompt the home terminal for. These features and commands are not intended to be run in an unattended mode since they do require user participation.

User Passphrases

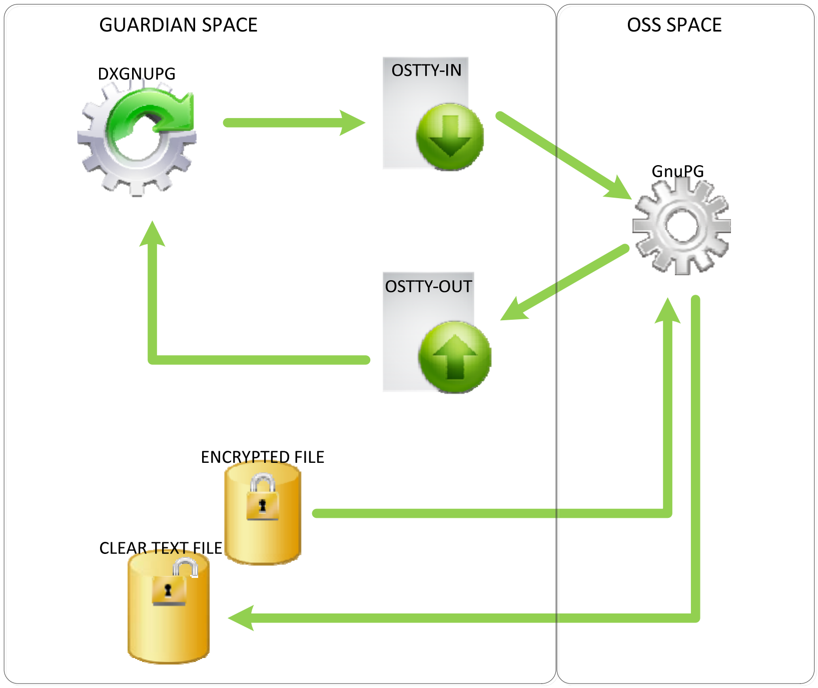
The passphrases for secret keys that are required to decrypt files are stored on a Guardian key structured file. These passphrases are encrypted on the passphrase file. Users must specify the recipient for decryption or encryption, or allow the 'default' user passphrase to be employed for file decryption or encryption. The user passphrase file is maintained by the DXGNUPG program and entries in it are created at the time user keys are generated using the GENKEY command. Entries are deleted in the same manner when keys are deleted from the key rings. This file is used specifically to allow unattended processing in a secure manner.

The DXGNUPG program requires:

- The GnuPG software available from the Open Source library section of the ITUGLIB website.
- Security access to execute the Guardian OSSTTY load object, generally found on $system.sys00
- Access to the OSS environment for setup

Decrypt command example:

```
dxgnupg/name $dxn1/decrypt  $disk1.encrypted.sample1   $disk1.cleartxt.sample1
```


GUARDIAN SPACE
OSS SPACE
DXGNUPG
OSTTY-IN
GnuPG
OSTTY-OUT
ENCRYPTED FILE
CLEAR TEXT FILE