



CTX-Encryption-Module User Guide

Contents

CTX-Encryption-Module User Guide..... 1

Contents 2

Versions 3

 Document Revisions 3

 Module Versions 3

Preface 4

 About this Manual 4

 Audience 4

 Related Material 4

 Abbreviations used in this Document..... 4

1 Requirements 5

2 Import CTX-Encryption-Module **Error! Bookmark not defined.**

3 Publish Flows..... **Error! Bookmark not defined.**

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
08/09/2021	1.0	First release

Module Versions

This version of the CTX-Encryption-Module User Guide is relevant up to version 1.0 of the CTX-Encryption-Module GitHub module.

Preface

About this Manual

This document provides a guide on how to deploy the CTX-Encryption-Module in your Cortex system.

Audience

This document is intended for those who require the use of CTX-Encryption-Module.

Related Material

Document
CTX-Encryption-Module – Deployment Plan
CTX-Encryption-Module.studiopkg

Abbreviations used in this Document

AES256 Advanced Encryption Standard (256 bits key length)

PGP Pretty Good Privacy


FQ Fully-Qualified

Requirements

This document details all the steps required to deploy the CTX-Encryption-Module module.

Requirements:

- 7zip installed at the default directory if using AES256 [Download \(7-zip.org\)](#)
- GnuPG v2.3.2 (or newer) installed if using PGP encryption [GnuPG - Download](#)
- Minimum Cortex v6.4 installed on the Cortex Application Server
- PowerShell V5 installed on the Cortex Server

 This module can run with either 7zip or GnuPG installed for AES256 or PGP respectively. The LivePortal flow does not require both to function properly while each subtask requires its respective software.

1 Encryption-Module Overview

The Encryption Module (EM) consists of a LivePortal (LP) flow **Encryption-Module-LP** and two subtasks: **EM-AES** and **EM-PGP**, each one handling both encryption and decryption for either AES256 or PGP respectively. The LP flow allows a user to interactively encrypt or decrypt file(s) whereas the subtasks can be integrated into any other flow with the required variables passed through. AES256 and PGP require 7zip and GnuPG to be installed on the Cortex server; please refer to the requirements section to ensure they are available.

1.1 Encryption Methods

1.1.1 AES256

AES256 is a version of AES (Advances Encryption Standard), and comes in either a 128-bit, 192-bit or 256-bit version. This specifies the length of the encryption key, with 256 being the strongest level of encryption. AES is fast and is best suited for closed systems and large databases.

1.1.2 PGP

PGP is just as strong as AES but adds an additional layer of security. This makes it useful for sharing information across open networks. It uses more computational resources, so it can be slower than AES and thus works better for individual files. This module uses password encryption for PGP currently, and not public-private key cryptography.

2 LivePortal UI Flows

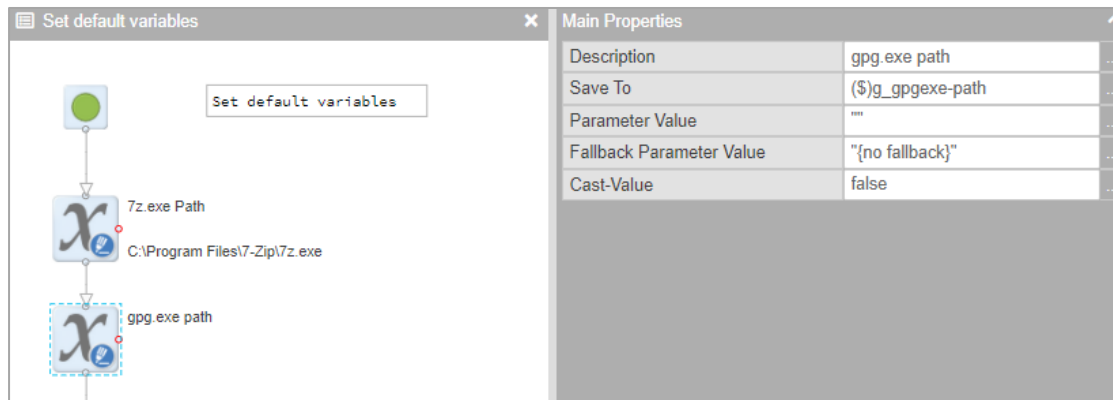
2.1.1 Overview and Configuration

This flow allows the user to interactively encrypt, or decrypt files based on their chosen method between AES256 or PGP in LivePortal.

At-least one of the encryption programs must be installed (7zip or GnuPG). Depending on which program has been installed, the user must first configure the file-path for either the **7z.exe** or **gpg.exe** file in the **(\$)SetVariables** state. The default path for these files is:

- C:\Program Files\7zip\7z.exe
- C:\Program Files (86)\GnuPG\bin\gpg.exe

When one of these is not installed, the corresponding *Set Variable* block must be left empty, as shown below.




Main Properties	
Description	gpg.exe path
Save To	(\$)g_gpgexe-path
Parameter Value	""
Fallback Parameter Value	"{no fallback}"
Cast-Value	false

2.1.2 User Homepage

Once the .studiopkg has been imported and published, either run the flow from LivePortal, or navigate to, and run the **Encryption-Module-LP** flow in gateway. The homepage UI will prompt the user to choose the method, either AES256 or PGP, and the operation, whether to encrypt or decrypt. If only one of either 7zip or GnuPG has been configured, only that method will be available.

Files can either be uploaded to the Cortex server – this allows multiple files to be operated on, or a single, FQ file-path on the Cortex application server can be provided.

 For security purposes, any files uploaded onto Cortex will be removed from the server after they have been operated on at the end of the flow execution, so no copy is left behind.

Encryption Module

Encryption Methods:

There are 2 types of encryption methods available: AES256 and PGP.

Please choose the method of encryption below and then whether you wish to Encrypt or Decrypt a file using the chosen method.
You may upload the file(s) or enter the full path of the single file you wish to operate on. All uploaded files will be removed from Cortex at the end of the operation.

Method

AES256

Operation

Encrypt

Browse File...

FilePath

If both an uploaded file and a file-path is provided, the uploaded file(s) will be prioritised and taken over the supplied file-path instead; not both. Successfully uploaded files will show a green circle next to them. Orange indicates the file is still being uploaded and red means that there has been an error. Please remove the file and try again if that is the case.

Method

AES256

Operation

Encrypt

Browse File...

● SSLPost.txt

✕ Remove

● CoE_LP-Analysis.txt

✕ Remove

FilePath

2.1.3 Encryption Key and Destination Folder

The next UI will prompt the user for a case-sensitive password. It is recommended that a secure, memorable password is used for encryption as there is no way to recover these files if the password is forgotten. The file path for the destination folder where the encrypted/decrypted files will be placed once the operation takes place. Non-existing folders specified in the file-path will automatically be created, e.g; specifying *C:\temp\newfolder\newfolder2* will create both *newfolder* and *newfolder2*.

Encryption Module


Please provide the destination path for the encrypted file to be saved to, and chose a secure password for encryption.
Note: The encrypted file complexity is determined by how secure (i.e complex) your password is.

Password *

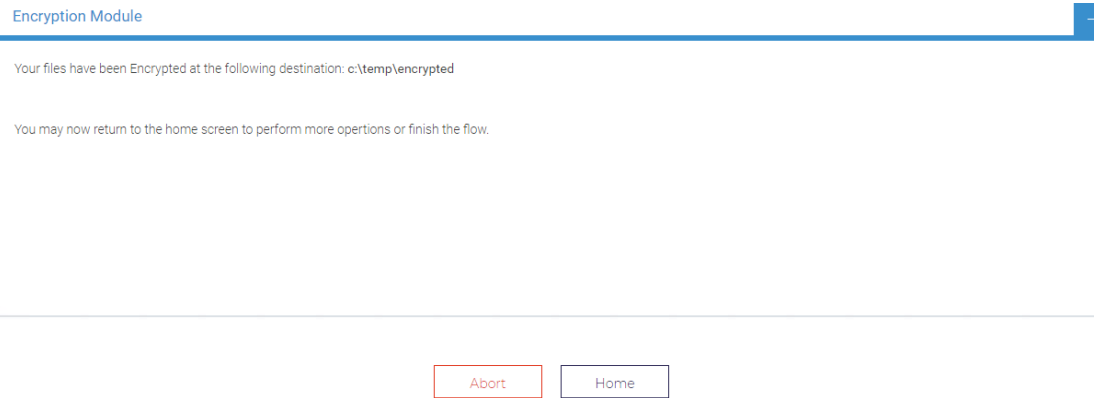
SecurePassword

Destination folder *

C:\temp\newfolder\newfolder2

 If you wish to save on your local machine, please copy over the files afterwards, or save to a shared network drive. The user must also have permissions to the destination folder.

Clicking OK will start the encryption (or decryption) and show a success message as shown below, at which point the user can abort the flow or return to the homepage to perform more operations. If an error occurred, the specific error message that PowerShell supplies will be shown to the user.



3 Encryption-Module Subtasks

3.1 EM-AES

3.1.1 Overview

This Subtask handles both the encryption and decryption of a file using AES256 encryption via PowerShell. It takes in a list of file-paths and returns a list of output-messages corresponding to the PowerShell response. Each element in the output results list corresponds to the file-path in the same element level of the input file-path list. The output will show the encrypted/decrypted file destination, and a message of **“Everything is OK”** if the file was encrypted successfully. Any errors will be reflected in the output results list if present.

3.1.2 Inputs

All the inputs are mandatory.

Input Variables	Type	Description
EM-AES_i_Operation	Text	Operation to perform (Encrypt/Decrypt). Example: Encrypt
EM-AES_i_File-Locations	List	A list containing the fully qualified paths for each file to encrypt or decrypt Example: A List with the 1 st element as “C:\temp\test.txt”
EM-AES_i_Destination-Folder-Path	Text	Path of the folder the encrypted/decrypted file is to be saved to Example: “C:\temp\encrypted”
EM-AES_i_Encryption-Key	Text	The password used for the 7-zip encryption
EM-AES_i_7zip-path	Text	The file path for the “7zip.exe” file in the 7zip installation folder
EM-AES_i_PS-Domain	Text	The domain for the PowerShell User
EM-AES_i_PS-Username	Text	The username for the PowerShell User
EM-AES_i_PS-Password	Text	The password for the PowerShell User

3.1.3 Outputs

Input Variables	Type	Description
EM-AES_o_Results-List	List	<p>Contains the results from the PowerShell script. Successful results will show the destination file path of the new archive and "Everything is OK".</p> <p>Unsuccessful results will show the error warnings.</p> <p>The results list is in chronological order of file-paths provided.</p> <p>Example Encryption Result (1 file):</p> <pre>["7-Zip 19.00 (x64) : Copyright (c) 1999-2018 Igor Pavlov : 2019-02-21", "Scanning the drive:", "1 file, 3 bytes (1 KiB)", "Creating archive: C:\\temp\\decrypted\\test.7z", "Add new data to archive: 1 file, 3 bytes (1 KiB)", "Files read from disk: 1", "Archive size: 206 bytes (1 KiB)", "Everything is Ok"]</pre>

3.2 EM-PGP

3.2.1 Overview

This subtask handles both encryption and decryption using PGP via a GnuPG module ran through PowerShell. It takes in a list of file-paths and returns an output list containing destination file-path for each of the encrypted/decrypted file that was passed in.

3.2.2 Inputs

All the inputs are mandatory.

Input Variables	Type	Description
EM-PGP_i_Operation	Text	Operation to perform (Encrypt/Decrypt). Example: Encrypt
EM-PGP_i_File-Locations	List	A list containing the fully-qualified paths for each file to encrypt or decrypt Example: A List with the 1 st element as "C:\temp\test.txt"
EM-PGP_i_Destination-Folder-Path	Text	Path of the folder the encrypted/decrypted file is to be saved to Example: "C:\temp\encrypted"
EM-PGP_i_Encryption-Key	Text	The password used for the PGP encryption
EM-PGP_i_GnuPG-path	Text	The file path for the "gpg.exe" file in the GnuPG installation folder
EM-PGP_i_PS-Domain	Text	The domain for the PowerShell User
EM-PGP_i_PS-Username	Text	The username for the PowerShell User
EM-PGP_i_PS-Password	Text	The password for the PowerShell User

3.2.3 Outputs

Input Variables	Type	Description
EM-PGP_o_Results-List	List	Contains the results from the PowerShell script. Successful results will show the destination file-path of the encrypted/decrypted file and unsuccessful results will show the corresponding error warning. The results list is in chronological order of file-paths provided. Example (1 file): ["C:\\temp\\test.txt.gpg"]