



Encrypt. Share. Protect.



Galaxkey API

Galaxkey API Reference and Integration

Version: 1.0

01/05/2016

Table of Contents

Section 1.	Copyright.....	3
Section 2.	Executive Summary.....	4
2.1.	About Galaxkey	4
Section 3.	Galaxkey API.....	5
3.1.	Galaxkey API Dll.....	5
Section 4.	API Reference.....	6
4.1.	GalaxkeyWrapper Methods	6
4.2.	GalaxkeyWrapper Property.....	8
Section 5.	C# Sample code.....	9
5.1.	Encryption :	9
5.2.	Decryption:.....	9

Section 1. Copyright

© Copyright Galaxkey® Limited. All rights reserved. Galaxkey Limited, the Galaxkey logo, Galaxkey are registered trademarks of Galaxkey Limited in the Europe and other countries. All other Trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Galaxkey Limited assumes no responsibility for any inaccuracies in this document. Galaxkey Limited reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Section 2. Executive Summary

2.1. About Galaxkey

Galaxkey provides an identity based email content security solution. When a user registers with Galaxkey, a security identity is created which is associated with the user's email address. This identity is managed and stored by Galaxkey in the cloud and is accessible to any Galaxkey enabled client on proper authorisation and authentication. The identities on the platform are completely secured with user provided unique credentials; ensuring secure user access. The administrators of corporate users have full control of their corporate user's identities and data.

With advances of networking and communication technologies, e-mail has established itself as a formal communication medium. E-mail as a medium has gone past the mere communication tool and is being used for communicating confidential information including official documents ranging from notices, invoices and confirmations to name a few. These developments also raise issues surrounding email security and privacy of data. E-mail is highly susceptible to various forms of attack, starting at the e-mail source point, onto the corporate server, including the stages in between, to the recipient's server and finally the destination point. The risk of attack increases multi-fold at each point. There are various solutions offered to counteract these vulnerabilities at various points. These solutions offer a partial solution. However, a robust solution that is able to address all the vulnerabilities and still maintain ultimate control with the end user, the sender and receiver is desirable. This is the solution Galaxkey provides.

Galaxkey offers security solutions using state of the art technology to secure e-mail end-to-end and not only in transit but at the source and at the destination while maintaining control with the end user through proper authentication.

Section 3. Galaxkey API

Galaxkey exposes its core functionality of encrypting and decrypting a file using Galaxkey platform in form of a Windows dll. The dll can be used to integrate Galaxkey's core functionality with any system that may want to use it in its data processing.

3.1. Galaxkey API Dll

Following are the details of the Galaxkey API

Attribute	Details
Name	GalaxkeyAPI.dll
Development Platform	Windows .NET
Software Requirements	Microsoft Windows with .NET Framework 4.0
64 bit / 32 bit	Dll is compiled for "x86" settings.
Connectivity Requirements	Using SOAP Over 443 to "*.galaxkey.com". In case of Hybrid setup, the connectivity to hybrid appliance endpoints over 443 is needed.
Logs	Logs are generated in file named "galaxkey_api.log" by the dll in "c:\Users\<loggedin user>\appData\Roaming\Galaxkey\Logs" folder.

Section 4. API Reference

GalaxkeyAPI Dll exposes “GalaxkeyWrapper” class which has following public methods and properties.

4.1. GalaxkeyWrapper Methods

1. **Encrypt:** This method is used for encrypting the given input file. Following is the method prototype

```
String encrypt(String owner, String emailList, String fileName, String strDestinationFile)
```

Parameter	Details
owner	Identity (email address) of the owner of the Galaxkey secured document.
emailList	Comma separated email address list of all the identities for which the document should be encrypted.
fileName	Fully qualified path of the source document to be encrypted.
strDestinationFile	File Path of the secured file which will be created. If this path is blank, File is created in same folder as that of source file along with added “.gmk” extension to original file.

Return Value:

If successful, the function returns filename with fully qualified path of the newly encrypted file. In case of failure the return string is blank. The reason for failure can be obtained by examining “LastError” property of the class.

Remarks:

The file encryption method follows following steps

- Check if the owner has valid identity.
- Get the public keys of all the registered email addresses in the list.

- If there are unregistered email addresses, “Invite” them to Galaxkey platform and get their public keys.
- Secure document using all public keys.
- Save secured document as the destination file with extension “.gmk”. Return the secured file path.

2. **Decrypt:** This method is used for decrypting the given file using the identity and its credentials supplied. Following is the method’s prototype

```
String decrypt(String strIdentity, String strPassword, String strFileName,
String strDestinationFile, bool bNotify)
```

Parameter	Details
strIdentity	Identity (email address) to be used to decrypt the input Galaxkey secured file.
strPassword	Password of the identity to be used for decrypting given Galaxkey secured document.
strFileName	Fully qualified path of the source document to be decrypted.
strDestinationFile	Fully qualified path of the destination file. If this is blank then the name of the original secured file is used for decrypted file. If the secured file is email and this parameter is blank, the name of gmk file is used as destination file name with .eml extension.
bNotify	True if the user wants to notify the owner of the document about opening the Galaxkey secured file. False if user does not want to. If it is set to false and owner of document has requested notification, the file will not be decrypted and the “LastError” will be set to “NOTIFICATION_REQUESTED”

Return Value:

If successful, the function returns filename with fully qualified path of the newly decrypted file. In case of failure the return string is blank. The reason for failure can be obtained by examining “LastError” property of the class.

Note: Galaxkey secured files with “Timeout” set cannot be decrypted using API.

Remarks:

Galaxkey API follows following steps to decrypt the given gxx file.

- Get the end point to get the private key for the identity.
- Get the identity’s private key.
- Load the gxx file and check if it is valid gxx file.
- Check if file is secured for given identity.
- Check if timeout is set for the gxx file.
- Check if notification is requested for the given identity by the owner.
- If notification is requested and bNotify is set to true, notify the owner about opening the gxx.
- Decrypt gxx to given destination folder.
- Return the file path of newly decrypted file.

4.2.GalaxkeyWrapper Property

Class GalaxkeyWrapper exposes only one property named “LastError”. This property gets updated every time an error occurs in execution.Calling program would get details of the reason why an operation failed in case the method call fails.

Section 5. C# Sample code

5.1.Encryption :

```
String strOwner = "name@domain.com";
String emailList = "user1@domain.com,user2@otherdomain.com";
String strFileName = @"c:\temp\test.doc";
String strDestinationFile = @"c:\temp\test.gxk";

GalaxkeyWrapper objWrapper = new GalaxkeyWrapper();
if (objWrapper.LastError.Length == 0)
{
    String strDestFile = objWrapper.encrypt(strOwner, emailList, strFileName,
strDestinationFile);
    if (String.IsNullOrEmpty(strDestFile) == true)
    {
        Console.WriteLine(objWrapper.LastError);
    }
    else
    {
        Console.WriteLine(strDestFile + " created succesfully.");
    }
}
else
{
    Console.WriteLine(objWrapper.LastError);
}
```

5.2.Decryption:

```
String strIdentity = "name@domain.com";
String strPassword = "user1@domain.com,user2@otherdomain.com";
String strFileName = @"c:\temp\test.doc.gxk";
String strDestinationFile = @"c:\temp\test.doc";
bool bNotify = false;

GalaxkeyWrapper objWrapper = new GalaxkeyWrapper();
if (objWrapper.LastError.Length == 0)
{
    String strFilePath = objWrapper.decrypt(strIdentity, strPassword, strFileName,
strDestinationFile, bNotify);
    if (String.IsNullOrEmpty(strFilePath) == true)
    {
        Console.WriteLine(objWrapper.LastError);
    }
    else
    {
        Console.WriteLine(fileName + " decrypted as " + strFilePath + " successfully.");
    }
}
else
{
    Console.WriteLine(objWrapper.LastError);
}
```