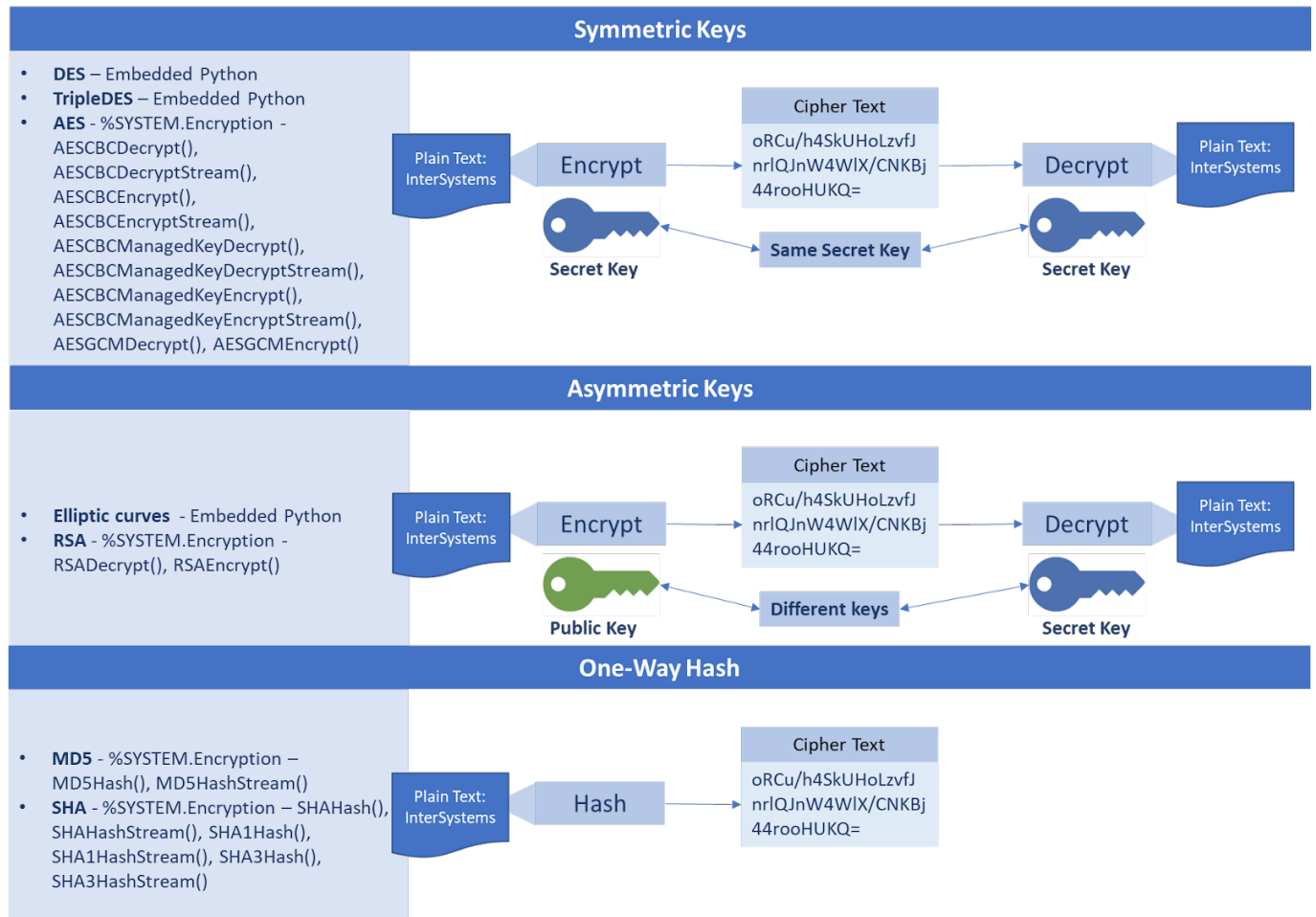


Article

[Yuri Marx](#) · May 13, 2022 8m read

[Open Exchange](#)

Mastering the %SYSTEM.Encryption class



The InterSystems IRIS has excellent support for encryption, decryption and hashing operations. Inside the class %SYSTEM.Encryption (<https://docs.intersystems.com/iris20212/csp/documatic/%25CSP.Documatic.c...>) there are class methods for the main algorithms on the market.

IRIS Algorithms and Encrypt/Decrypt types

As you can see, the operations are based on keys and include 3 options:

- Symmetric Keys: the parts running encrypt and decrypt operations share the same secret key.
- Asymmetric Keys: the parts conducting encrypt and decrypt operations share the same secret key for encryption. However, for decryption, each partner has a private key. This key cannot be shared with other people, because it is an identity proof.
- Hash: used when you do not need to decrypt, but only encrypt. It is a common approach when it comes to storing user passwords.

Differences Between Symmetric and Asymmetric Encryption

- Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetric encryption uses a pair of public keys and a private key to encrypt and decrypt messages when communicating.
- Symmetric encryption is an old technique while asymmetric encryption is relatively new.
- Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in a symmetric encryption model, eliminating the need to share the key by using a pair of public-private keys.
- Asymmetric encryption takes relatively more time than symmetric encryption.

Key Differences	Symmetric Encryption	Asymmetric Encryption
Size of cipher text	Smaller cipher text than the original plain text file.	Larger cipher text than the original plain text file.
Data size	Used to transmit big data.	Used to transmit small data.
Resources Utilization	Symmetric key encryption works on low usage of resources.	Asymmetric encryption requires high consumption of resources.
Key Length	128 or 256-bit key size.	RSA 2048-bit or higher key size.
Security	Less secure due to the usage of a single key for encryption.	Much safer as two different keys are involved in encryption and decryption.
Number of keys	Symmetric Encryption uses a single key for encryption and decryption.	Asymmetric Encryption uses two different keys for encryption and decryption.
Techniques	It is an old technique.	It is a modern technique.
Confidentiality	A single key for encryption and decryption has chances of the key being compromised.	Two keys are separately made for encryption and decryption which removes the need to share a key.
Speed	Symmetric encryption is a fast technique.	Asymmetric encryption is slower in terms of speed.
Algorithms	RC4, AES, DES, 3DES, and QUAD.	RSA, Diffie-Hellman, ECC algorithm.

Source: <https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>

Using the %SYSTEM.Encryption class to do Encrypt, Decrypt and Hash

To exercise IRIS support to Encrypt, Decrypt and Hash operations, go to <https://github.com/yurimarx/cryptography-samples> and follow these steps:

1. Clone/git pull the repo into any local directory

```
$ git clone https://github.com/yurimarx/cryptography-samples.git
```

2. Open a Docker terminal in this directory and run:

```
$ docker-compose build
```

3. Run the IRIS container:

```
$ docker-compose up -d
```

4. Open IRIS Terminal:

```
$ docker-compose exec iris iris session iris -U IRISAPP
```

```
IRISAPP>
```

5. To do RSA Encrypt for asymmetric encryption execute this:

```
IRISAPP>Set ciphertext = ##class(dc.cryptosamples.Samples).DoRSAEncrypt("InterSystems")
IRISAPP>Write ciphertext
Ms/eR7pPmE39KBJu75EOYIxpFE7qgoji6lEfahJElr9mGZXlNYuw5i2cPS5YwE3Aw6vPAeiEKXF
rYW++WtzMeRIRdCMbLG9PrCHD3iQHfZobBnuzx/JMXVc6a4TssbY9gk7qJ5BmlqRTU8zNJiiVmd8
pCFpJgwKzKkNrIgaQn48EgnwblmVxxSFnF2jwXpBt/naNudBguFUBthef2wfULL4uY00aZzHHNxAbi15mzTdlSJulvRtCQaEahng9ug7BZ6dyWCHOv74O/L5NEHI+jU+kHQeF2DJneE2yWNESzqhSECaZbRjjxNxiRn/HVAKyZdAjkgQVKUkyG8vjnc3Jw==
```

6. To do RSA Decrypt for asymmetric decryption run this:

```
IRISAPP>Set plaintext = ##class(dc.cryptosamples.Samples).DoRSADecrypt(ciphertext)
IRISAPP>Write plaintext
InterSystems
```

7. To do AES CBC Encrypt for symmetric encryption perform this:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoAESCBCEncrypt("InterSystems")
8sGVUikDZaJF+Z9Ul jFVAA==
```

8. To do AES CBC Decrypt for symmetric encryption complete this:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoAESCBCDecrypt("8sGVUikDZaJF+Z9Ul jFVAA=")
InterSystems
```

9. To do MD5 hash for an old hash approach conduct this:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoHash("InterSystems")
rOs6HXfrnbEY5+JBdUJ8hw==
```

10. To do SHA hash for recommended hash approach follow this:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoSHAHash("InterSystems")
+X0hdlyoViPlWom/825KvN3rRKB5cTU5EQTDLvPWM+E=
```

11. To exit the terminal, do any of the following:

Enter HALT or H (not case-sensitive)

About the source code

1. About the Symmetric key

```
# to use with symmetric encrypt/decrypt  
ENV SECRETKEY=InterSystemsIRIS
```

In the Dockerfile, there was created an Environment key to be used as secret key on symmetric operations.

2. About the Asymmetric key

```
# to use with asymmetric encrypt/decrypt  
RUN openssl req -new -x509 -sha256 -config example-com.conf -newkey rsa:2048 -nodes  
example-com.key.pem -days 365 -out example-com.cert.pem
```

In the Dockerfile were generated a private key and a public key to be used with asymmetric operations.

3. Symmetric Encrypt

```
// Symmetric Keys sample to encrypt  
ClassMethod DoAESCBCEncrypt(plaintext As %String) As %Status  
{  
    // convert to utf-8  
    Set text=$ZCONVERT(plaintext,"O","UTF8")  
  
    // set a secret key  
    Set secretkey = $system.Util.GetEnviron("SECRETKEY")  
    Set IV = $system.Util.GetEnviron("SECRETKEY")  
  
    // encrypt a text  
    Set text = $SYSTEM.Encryption.AESCBCEncrypt(text, secretkey, IV)  
    Set ciphertext = $SYSTEM.Encryption.Base64Encode(text)
```

Write ciphertext

The operation AES CBC Encrypt was used to encrypt texts.

Base64 Encode returns the results as a pretty/readable text to the user.

4. Symmetric Decrypt

```
// Symmetric Keys sample to decrypt
ClassMethod DoAESCBDecrypt(ciphertext As %String) As %Status
{
    // set a secret key
    Set secretkey = $system.Util.GetEnviron("SECRETKEY")
    Set IV = $system.Util.GetEnviron("SECRETKEY")

    // decrypt a text
    Set text=$SYSTEM.Encryption.Base64Decode(ciphertext)
    Set text=$SYSTEM.Encryption.AESCBDecrypt(text,secretkey,IV)

    Set plaintext=$ZCONVERT(text,"I","UTF8")
    Write plaintext
}
```

The operation AES CBC Decrypt was used to decrypt texts.

Base64 Decode returns the encrypted text to a binary one, so it can be used to decrypt.

5. Asymmetric Encrypt

```
// Asymmetric Keys sample to encrypt
ClassMethod DoRSAEncrypt(plaintext As %String) As %Status
{
    // get public certificate
    Set pubKeyFileName = "/opt/irisbuild/example-com.cert.pem"
    Set objCharFile = ##class(%Stream.FileCharacter).%New()
    Set objCharFile.Filename = pubKeyFileName
    Set pubKey = objCharFile.Read()

    // encrypt using RSA
    Set binarytext = $System.Encryption.RSAEncrypt(plaintext, pubKey)
    Set ciphertext = $SYSTEM.Encryption.Base64Encode(binarytext)

    Return ciphertext
}
```

}

It is necessary to get the public key file content to encrypt with RSA.
The operation RSA Encrypt was used to encrypt texts.

6. Asymmetric Decrypt

```
// Asymmetric Keys sample to decrypt
ClassMethod DoRSADecrypt(ciphertext As %String) As %Status
{
    // get private key
    Set privKeyFileName = "/opt/irisbuild/example-com.key.pem"
    Set privobjCharFile = ##class(%Stream.FileCharacter).%New()
    Set privobjCharFile.Filename = privKeyFileName
    Set privKey = privobjCharFile.Read()
    // get ciphertext in binary format
    Set text=$SYSTEM.Encryption.Base64Decode(ciphertext)
    // decrypt text using RSA
    Set plaintext = $System.Encryption.RSADecrypt(text, privKey)
    Return plaintext
}
```

It is necessary to get the private key file content to decrypt with RSA.
The operation RSA Decrypt was used to decrypt texts.

7. Hash text using MD5 (old approach)

```
// Hash sample
ClassMethod DoHash(plaintext As %String) As %Status
{
    // convert to utf-8
    Set text=$ZCONVERT(plaintext,"O","UTF8")
    // hash a text
    Set hashtext = $SYSTEM.Encryption.MD5Hash(text)
    Set base64text = $SYSTEM.Encryption.Base64Encode(hashtext)
    // convert to hex text to following best practices
    Set hextext = ..GetHexText(base64text)
    // return using lowercase
}
```

```
Write $ZCONVERT(hextext,"L")
}
```

The operation MD5 Hash will encrypt the text, and it will not be possible to decrypt it.

Hash using MD5 is not recommended for new projects because it is considered insecure. That is why it was replaced by SHA. The InterSystems IRIS supports SHA (our next example will demonstrate it).

8. Hash text using SHA (recommend approach)

We will use the SHA-3 Hash method for this sample. According to InterSystems documentation, this method generates a hash using one of the U.S. Secure Hash Algorithms - 3. (See Federal Information Processing Standards Publication 202 for more information.).

```
// Hash using SHA
ClassMethod DoSHAHash(plaintext As %String) As %Status
{
    // convert to utf-8
    Set text=$ZCONVERT(plaintext,"O","UTF8")

    // hash a text
    Set hashtext = $SYSTEM.Encryption.SHA3Hash(256, text)

    Set base64text = $SYSTEM.Encryption.Base64Encode(hashtext)
    // convert to hex text to following best practices
    Set hextext = ..GetHexText(base64text)
    // return using lowercase
    Write $ZCONVERT(hextext,"L")
}
```

For the SHA method, it is possible to set the bit length used on a hash operation. The greater the number of bits, the more difficult it is to crack the hash. However, the hashing process slows down too. In this sample we used 256 bits. You can choose these options for bit length:

- 224 (SHA-224)
- 256 (SHA-256)
- 384 (SHA-384)
- 512 (SHA-512)

[#Best Practices](#) [#Security](#) [#InterSystems IRIS](#)

[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/mastering-systemencryption-class>