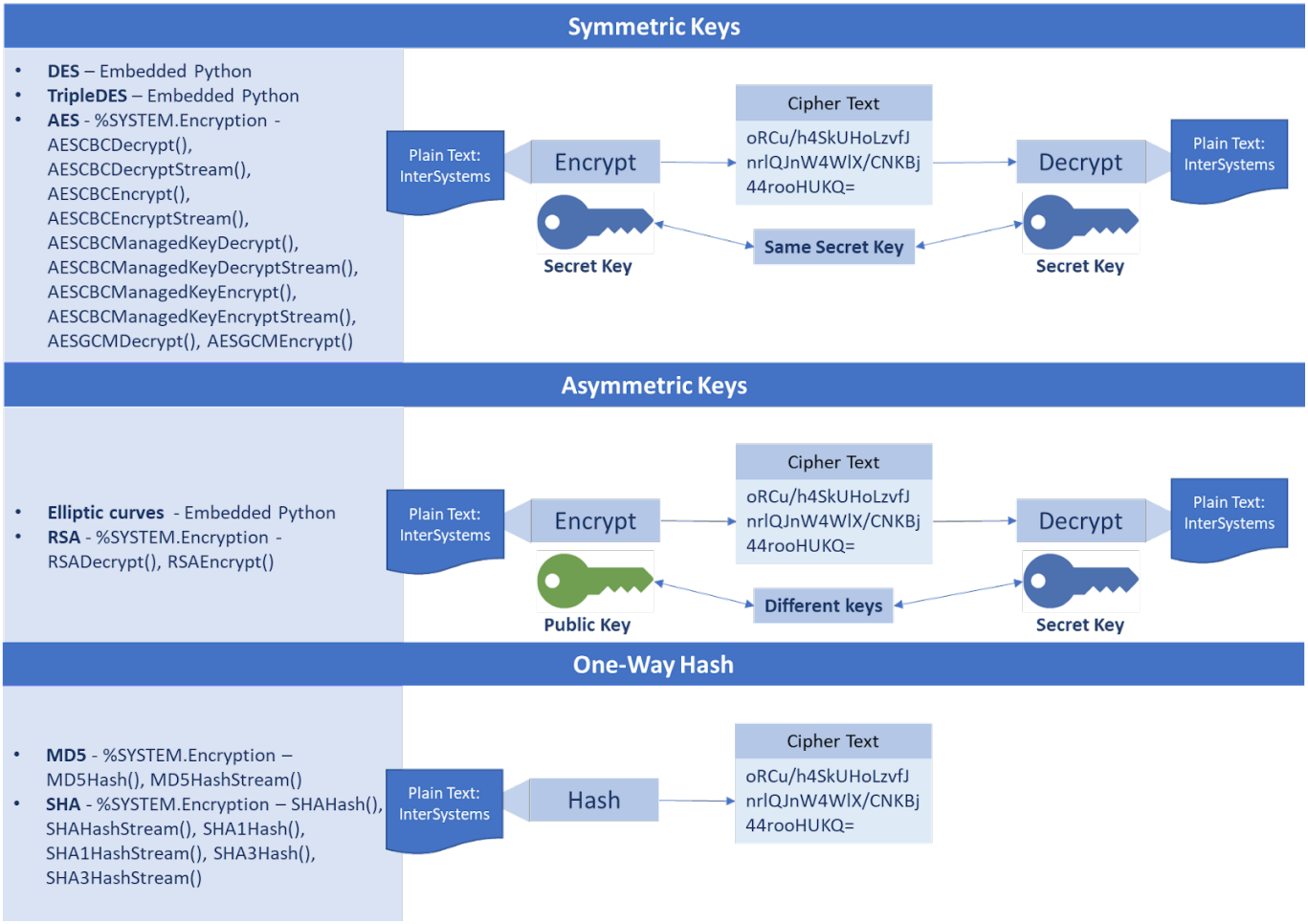
Artigo
Danusa Calixto · Jul. 11, 2022 10min de leitura

Open Exchange

Dominando a classe %SYSTEM.Encryption



O InterSystems IRIS tem um suporte excelente para operações de criptografia, descriptografia e hashing. Na classe %SYSTEM.Encryption (https://docs.intersystems.com/iris20212/csp/documatic/%25CSP.Documatic.c...), há métodos de classes para os principais algoritmos no mercado.

#

Algoritmos IRIS e tipos de criptografia/descriptografia

Como você pode ver, as operações são baseadas em chaves e incluem 3 opções:

- **Chaves simétricas**: as partes que executam as operações de criptografia e descriptografia compartilham a mesma chave secreta.
- **Chaves assimétricas**: as partes que conduzem as operações de criptografia e descriptografia compartilham a mesma chave secreta para a criptografia. No entanto, para a descriptografia, cada parceiro tem uma chave privada. Essa chave não pode ser compartilhada com outras pessoas, pois é uma prova de identidade.
- **Hash**: é usado quando você só precisa criptografar, e não descriptografar. É uma abordagem comum para

o armazenamento das senhas dos usuários.

Diferenças entre a criptografia simétrica e assimétrica

- A criptografia simétrica usa uma única chave, compartilhada entre as pessoas que precisam receber a mensagem, enquanto a criptografia assimétrica usa um par de chaves públicas e uma chave privada para criptografar e descriptografar as mensagens durante a comunicação.
- A criptografia simétrica é uma técnica antiga, enquanto a criptografia assimétrica é relativamente nova.
- A criptografia assimétrica foi criada para complementar o problema inerente da necessidade de compartilhar a chave em um modelo de criptografia simétrica, sem precisar compartilhar a chave usando um par de chaves pública-privada.
- A criptografia assimétrica leva mais tempo do que a criptografia simétrica.

Principais diferenças	Criptografia simétrica	Criptografia assimétrica
Tamanho do texto cifrado	Texto cifrado menor do que o arquivo de texto simples original.	Texto cifrado maior do que o arquivo de texto simples original.
Tamanho dos dados	Usada para transmitir dados grandes.	Usada para transmitir dados pequenos.
Uso de recursos	A criptografia simétrica de chaves funciona com baixo uso de recursos.	A criptografia assimétrica requer alto consumo de recursos.
Comprimento da chave	Tamanho de 128 ou 256-bit.	Tamanho de RSA 2048-bit ou superior.
Segurança	Menos segura devido ao uso de uma única chave para criptografia.	Muito mais segura já que duas chaves diferentes estão envolvidas na criptografia e descriptografia.
Número de chaves	A criptografia simétrica usa uma única chave para a criptografia e descriptografia.	A criptografia assimétrica usa duas chaves diferentes para a criptografia e descriptografia
Técnicas	É uma técnica antiga.	É uma técnica moderna.
Confidencialidade	Uma única chave para a criptografia e descriptografia tem o risco de ser comprometida.	Duas chaves são criadas separadamente para a criptografia e descriptografia, o que elimina a necessidade de compartilhar uma chave.
Velocidade	A criptografia simétrica é uma técnica rápida.	A criptografia assimétrica é mais lenta em termos de velocidade.
Algoritmos	RC4, AES, DES, 3DES e QUAD.	RSA, Diffie-Hellman e ECC.

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

Usando a classe %SYSTEM.Encryption para criptografia, descriptografia e hash

Para treinar o suporte do IRIS para as operações de criptografia, descriptografia e hash, acesse https://github.com/yurimarx/cryptography-sample e siga estas etapas:

1. Use o git pull ou clone o repositório em qualquer diretório local

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

2. Abra um terminal Docker nesse diretório e execute:

```
$ docker-compose build
```

3. Execute o contêiner IRIS:

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

4. Abra o terminal IRIS:

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

IRISAPP>
```

5. Para a criptografia assimétrica com RSA, execute:

```
IRISAPP>Set ciphertext = ##class(dc.cryptosamples.Samples).DoRSAEncrypt("InterSystems
")
IRISAPP>Write ciphertext
Ms/eR7pPmE39KBJu75EOYIxpFEd7qqoji61EfahJE1r9mGZX1NYuw5i2cPS5YwE3Aw6vPAeiEKXF
rYW++WtzMeRIRdCMbLG9PrCHD3iQHfZobBnuzx/JMXVc6a4TssbY9gk7qJ5BmlqRTU8zNJiiVmd8
pCFpJgwKzKkNrIgaQn48EgnwblmVkxSFnF2jwXpBt/naNudBguFUBthef2wfULl4uY00aZzHHNxA
bi15mzTdlSJu1vRtCQaEahng9ug7BZ6dyWCHOv74O/L5NEHI+jU+kHQeF2DJneE2yWNESzqhSECa
ZbRjjxNxiRn/HVAKyZdAjkGQVKUkyG8vjnc3Jw==
```

6. Para a descriptografia assimétrica com RSA, execute:

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

7. Para a criptografia simétrica com AES CBC, execute:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoAESCBCEncrypt("InterSystems")
8sGVUikDZaJF+Z9UljFVAA==
```

8. Para a descriptografia simétrica com AES CBC, execute:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoAESCBCDecrypt("8sGVUikDZaJF+Z9UljFVAA=
=")
InterSystems
```

9. Para uma abordagem antiga com hash MD5, execute:

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

10. Para uma abordagem recomendada com hash SHA, execute:

```
IRISAPP>Do ##class(dc.cryptosamples.Samples).DoSHAHash("InterSystems")
+X0hDlyoViPlWOm/825KvN3rRKB5cTU5EQTDLvPWM+E=
```

11. Para sair do terminal:

```
Digite HALT ou H (não diferencia maiúsculas de minúsculas)
```

Sobre o código-fonte

1. Sobre a chave simétrica

```
# para usar com criptografia/descriptografia simétrica
ENV SECRETKEY=InterSystemsIRIS
```

No Dockerfile, foi criada uma chave do ambiente para ser usada como chave secreta em operações simétricas.

2. Sobre a chave assimétrica

para usar com criptografia/descriptografia assimétrica, execute openssl req -new -x509 -sha256 -config example-com.conf -newkey rsa:2048 -nodes -keyout example-com.key.pem -days 365 -out example-com.cert.pem
No Dockerfile, foram geradas uma chave privada e uma chave pública para serem usadas com operações assimétricas.

3. Criptografia simétrica

```
// Amostra de chaves simétricas para criptografia
ClassMethod DoAESCBCEncrypt(plaintext As %String) As %Status
{
  // converter para utf-8
  Set text=$ZCONVERT(plaintext,"O","UTF8")

  // definir uma chave secreta
  Set secretkey = $system.Util.GetEnviron("SECRETKEY")
  Set IV = $system.Util.GetEnviron("SECRETKEY")

  // criptografar um texto
  Set text = $SYSTEM.Encryption.AESCBCEncrypt(text, secretkey, IV)
  Set ciphertext = $SYSTEM.Encryption.Base64Encode(text)

  Write ciphertext
}
```

A operação AES CBC Encrypt foi usada para criptografar textos.
A codificação base64 retorna os resultados como um texto organizado/legível para o usuário.

4. Descriptografia simétrica

```
// Amostra de chaves simétricas para descriptografia
ClassMethod DoAESCBCDecrypt(ciphertext As %String) As %Status
{
  // definir uma chave secreta
  Set secretkey = $system.Util.GetEnviron("SECRETKEY")
  Set IV = $system.Util.GetEnviron("SECRETKEY")

  // descriptografar um texto
  Set text=$SYSTEM.Encryption.Base64Decode(ciphertext)
  Set text=$SYSTEM.Encryption.AESCBCDecrypt(text,secretkey,IV)
```

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

A operação AES CBC Decrypt foi usada para descriptografar textos.
A decodificação base64 retorna o texto criptografado a um binário, para ser usado na descriptografia.

5. Criptografia assimétrica

```
 // Amostra de chaves assimétricas para criptografia
ClassMethod DoRSAEncrypt(plaintext As %String) As %Status
{
   // obter certificado público
   Set pubKeyFileName = "/opt/irisbuild/example-com.cert.pem"
   Set objCharFile = ##class(%Stream.FileCharacter).%New()
   Set objCharFile.Filename = pubKeyFileName
   Set pubKey = objCharFile.Read()
   // criptografar usando RSA
   Set binarytext = $System.Encryption.RSAEncrypt(plaintext, pubKey)
   Set ciphertext = $SYSTEM.Encryption.Base64Encode(binarytext)

   Return ciphertext
}
```

É preciso obter o conteúdo do arquivo da chave pública para criptografar com RSA.
A operação RSA Encrypt foi usada para criptografar textos.

6. Descriptografia assimétrica

```
 // Amostra de chaves assimétricas para descriptografia
ClassMethod DoRSADecrypt(ciphertext As %String) As %Status
{
   // obter chave pública
   Set privKeyFileName = "/opt/irisbuild/example-com.key.pem"
   Set privobjCharFile = ##class(%Stream.FileCharacter).%New()
   Set privobjCharFile.Filename = privKeyFileName
   Set privKey = privobjCharFile.Read()
   // obter ciphertext em formato binário
   Set text=$SYSTEM.Encryption.Base64Decode(ciphertext)
   // descriptografar texto usando RSA
   Set plaintext = $System.Encryption.RSADecrypt(text, privKey)
   Return plaintext
}
```

É preciso obter o conteúdo do arquivo da chave privada para descriptografar com RSA.
A operação RSA Decrypt foi usada para descriptografar textos.

7. Texto com hash usando MD5 (abordagem antiga)

```
// Amostra de hash
ClassMethod DoHash(plaintext As %String) As %Status
{
  // converter para utf-8
  Set text=$ZCONVERT(plaintext,"O","UTF8")

  // hash de um texto
  Set hashtext = $SYSTEM.Encryption.MD5Hash(text)

  Set base64text = $SYSTEM.Encryption.Base64Encode(hashtext)
  // converter para texto hex para seguir as melhores práticas
  Set hextext = ..GetHexText(base64text)
  // retornar usando minúsculas
  Write $ZCONVERT(hextext,"L")
}
```

A operação MD5 Hash fará a criptografia do texto e não será possível descriptografar.
O hash com MD5 não é recomendado para novos projetos por não ser considerado seguro. Por isso, foi substituído por SHA. O InterSystems IRIS oferece compatibilidade com SHA (veja nosso próximo exemplo).

8. Texto com hash usando SHA (abordagem recomendada)

Usaremos o método de hash SHA-3 para esta amostra. De acordo com a documentação do InterSystems, esse método gera um hash usando um dos Algoritmos de Hash Seguro dos EUA - 3. Consulte a Publicação 202 dos Federal Information Processing Standards para saber mais.

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

  // hash de um texto
  Set hashtext = $SYSTEM.Encryption.SHA3Hash(256, text)

  Set base64text = $SYSTEM.Encryption.Base64Encode(hashtext)
  // converter para texto hex para seguir as melhores práticas
  Set hextext = ..GetHexText(base64text)
  // retornar usando minúsculas
```

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

No método SHA, é possível definir o comprimento do bit usado em uma operação de hash. Quanto maior o número de bits, mais difícil é decifrar o hash. No entanto, o processo de hashing também fica mais lento. Na amostra, usamos 256 bits. Você pode escolher estas opções de comprimento de bit:

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

#Melhores Práticas #Segurança #InterSystems IRIS
Confira o aplicativo relacionado no InterSystems Open Exchange

---

URL de origem:https://pt.community.intersystems.com/post/dominando-classe-systemencryption