Ingenius. Revista de Ciencia y Tecnología
ISSN: 1390-650X
revistaingenius@ups.edu.ec
Universidad Politécnica Salesiana
Ecuador

Mendoza T., Julio César
DEMOSTRACIÓN DE CIFRADO SIMETRICO Y ASIMETRICO
Ingenius. Revista de Ciencia y Tecnología, núm. 3, 2008, pp. 46-53
Universidad Politécnica Salesiana
Cuenca, Ecuador

Disponible en: http://www.redalyc.org/articulo.oa?id=505554806007

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org
Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Julio César Mendoza T.
Ingeniería de Sistemas – Quito

RESUMEN

En el presente artículo se presenta una breve introducción a la criptografía sin profundizar en las matemáticas que soportan los algoritmos criptográficos; simplemente se abordará al cifrado a un nivel muy básico, para mostrar una visión de los distintos tipos (simétricos y asimétricos) y realizar una demostración de cifrado aplicando ambas técnicas mediante la aplicación del algoritmo RSA y 3DES en Visual Basic.NET. El propósito de éste articulo es mostrar técnicas sencillas, fáciles de integrar en aplicaciones y que tengan un impacto mínimo en la facilidad de empleo.

Para finalizar se mencionan las ventajas y desventajas de la utilización del cifrado simétrico y asimétrico.

INTRODUCCIÓN

La criptografía es una palabra que proviene de las palabras griegas Kriptos (ocultar) y graphos (escritura). Literalmente significa "escritura oculta", es decir mantener seguro los secretos mediante la codificación de los mensajes para hacerlos ilegibles, de tal manera que solo pueda verla aquel receptor que el emisor desea.

En terminología de cifrado:

- El mensaje original recibe el nombre de texto claro.
- El mensaje codificado se denomina texto cifrado.
- El proceso de convertir el texto claro en texto cifrado se denomina cifrado.
- El proceso de recuperar el texto claro a partir del texto cifrado se denomina descifrado.

Una parte importante de la aplicación de los algoritmos criptográficos son las claves. Una clave criptográfica es similar a una llave física que se usa para cerrar o abrir una puerta. Para cada tipo de cerradura existe una llave con una forma específica que se ajuste a aquella con cierta longitud apropiada capaz de girar y abrir la cerradura. [1] Las claves criptográficas son similares a las llaves físicas de muchas maneras. Cada algoritmo criptográfico necesita una clave con la extensión correcta (número correcto de bits). Se puede procesar un algoritmo criptográfico con cualquier clave que tenga la longitud apropiada, pero solo la que tenga el patrón correcto de bits hará que el algoritmo descifre la información cifrada.

El cifrado ayuda a garantizar: confidencialidad, autenticación e integridad. Los algoritmos de cifrado se clasifican en simétricos y asimétricos.

Una parte importante de la aplicación de los algoritmos criptográficos son las claves. Una clave criptográfica es similar a una llave física que se usa para cerrar o abrir una puerta. Para cada tipo de cerradura existe una llave con una forma específica que se ajuste a aquella con cierta longitud apropiada capaz de girar y abrir la cerradura.

A. Criptografía Simétrica

La criptografía simétrica utiliza la misma clave para cifrar y descifrar el mensaje de datos, es decir se basa en un secreto compartido. Es por esta razón que la seguridad de este proceso depende de la posibilidad de que una persona no autorizada consiga la clave de sesión o clave secreta.

Los algoritmos criptográficos simétricos tienen dos versiones: cifrador en bloque y cifrador en flujo. Una cifra es una palabra para describir un algoritmo de cifrado. Los cifradores en bloque codifican datos en bloques pequeños de longitud fija de 64 bits de longitud. Hay muchos cifradores en bloque que incluyen DES, 3-DES, RC2, RC5, RC6 y Rijndael (conocido como AES)

DES es un algoritmo simétrico cuyas abreviaturas significa "Estándar de cifrado de datos" o "*Data Encryption Estandar*" y es un cifrador en bloque de 64 bits de longitud. La palabra triple hace referencia a la forma que funciona el cifrado; en primer lugar se cifra el texto en claro; a continuación, éste resultado se vuelve a cifrar, lo que da lugar a que el texto claro se cifre en tres ocasiones y el resultado es un robusto cifrado de 192 bits, determinando el número total de posibles claves. Por ejemplo, una clave de 192 bits tiene 2192 posibles valores

La variante [1] más simple del Tripe DES funciona de la siguiente manera:

$$C = E^{k_3}_{DES}\left(D^{k_2}_{DES}\left(E^{k_1}_{DES}(M)\right)\right)$$

• Donde M es el mensaje a cifrar y k1, k2 y k3 las respectivas claves DES.

Para explicar como funciona el cifrado simétrico se utilizará 3DES en el siguiente ejemplo (Figura A1), donde se seguirá los siguientes pasos:

1. Una vez digitado el "*Texto Claro*" y la "*Clave*", se presiona el botón "*Cifrado*" y automáticamente se cifrará el texto utilizando el algoritmo Triple-DES y se mostrará el resultado en el cuadro "*Texto Cifrado*"

2. Posteriormente se presiona el botón "*Descifrar*" y se descifrará con la misma clave automáticamente el "*Texto cifrado*" y se mostrará el resultado en el cuadro "*Texto Claro*"

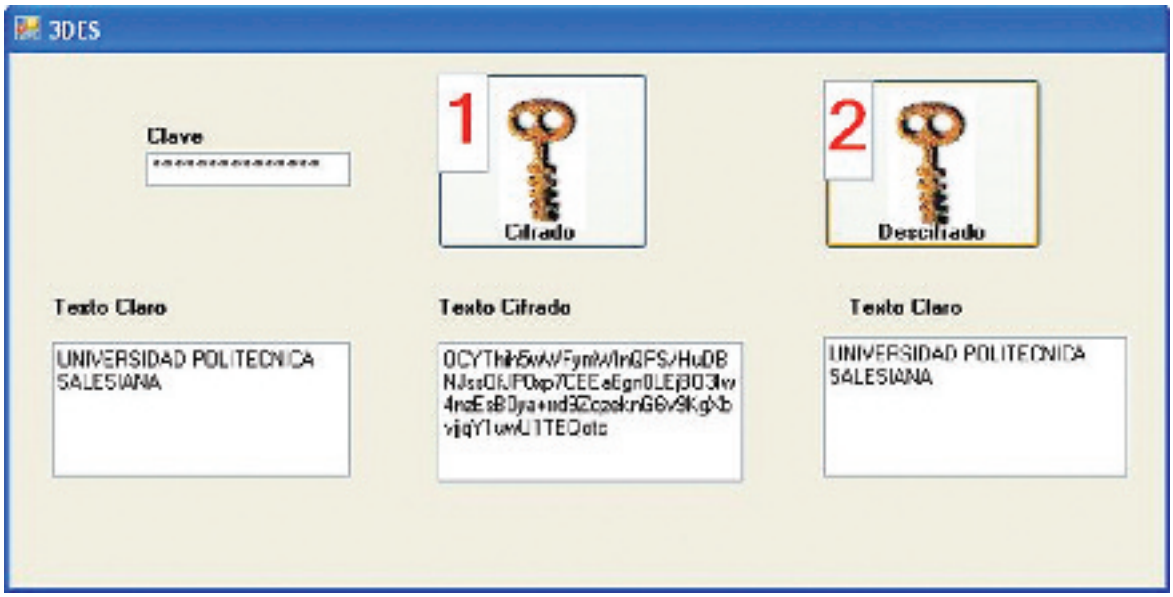

Figura A1. Cifrado Simétrico

A continuación se presenta las funciones, utilizadas en cada uno de los pasos realizados anteriormente:

1. Encriptar(textoClaro, clave): ésta función cifra la cadena "*textoClaro*" utilizando una cadena "*clave*" de 24 caracteres de longitud y muestra su resultado en el cuadro de texto "*Texto Cifrado*".

```
Function Encriptar(ByVal textoClaro As String, ByVal clave As String) As String
    Dim des As New TripleDESCryptoServiceProvider()
    Dim uCodifica As New UnicodeEncoding()
    Dim textoClaroByte() As Byte = uCodifica.GetBytes(textoClaro)
    Dim textoCifradoByte As New MemoryStream()
    Dim slt(0) As Byte
```

```
Dim pdb As New PasswordDeriveBytes(clave, slt)
Dim claveByte() As Byte = pdb.GetBytes(24)
des.Key = claveByte
des.IV = pdb.GetBytes(8)
Dim csEncrypted As New CryptoStream(textoCifradoByte, des.CreateEncryptor(), _
CryptoStreamMode.Write)
csEncrypted.Write(textoClaroByte, 0, textoClaroByte.Length)
csEncrypted.FlushFinalBlock()
Return Convert.ToBase64String(textoCifradoByte.ToArray())
End Function
```

2. Decrypt (textoCifrado, clave): ésta función descifra la cadena "*textoCifrado*" utilizando la misma cadena "*clave*" y asigna su resultado en el cuadro de texto "*Texto Claro*"

```
Function Decrypt(ByVal textoCifrado As String, ByVal clave As String) As String
    Dim des As New TripleDESCryptoServiceProvider()
    Dim uCodifica As New UnicodeEncoding()
    Dim textoCifradoByte() As Byte = Convert.FromBase64String(textoCifrado)
    Dim textoClaroMem As New MemoryStream()
    Dim textoCifradoMem As New MemoryStream(textoCifradoByte)
    Dim arreglo(0) As Byte
    Dim pdb As New PasswordDeriveBytes(clave, arreglo)
    Dim claveByte() As Byte = pdb.GetBytes(24)
    des.Key = claveByte
    des.IV = pdb.GetBytes(8)
    Dim desDecifrado As New CryptoStream(textoCifradoMem, des.CreateDecryptor(),
    _ CryptoStreamMode.Read)
    Dim varEscritura As New StreamWriter(textoClaroMem)
    Dim varLectura As New StreamReader(desDecifrado)
    varEscritura.Write(varLectura.ReadToEnd)
    varEscritura.Flush()
    desDecifrado.Clear()
    des.Clear()
    Return uCodifica.GetString(textoClaroMem.ToArray())
    End Function
```

B. Criptografía Asimétrica

Los algoritmos asimétricos son diferentes a los simétricos en un sentido muy importante. Cuando se genera una clave simétrica, simplemente se escoge un número aleatorio de la longitud apropiada. Al generar claves asimétricas el proceso es más complejo.

Los algoritmos asimétricos se llaman asimétricos porque en lugar de usar una sola clave para realizar la codificación y la decodificación, se utilizan dos claves diferentes: una para cifrar y otra para descifrar. Estas dos claves se encuentran asociadas matemáticamente, cuya característica fundamental es que una clave no puede descifrar lo que cifra.

Cuando se completa la generación de una clave asimétrica se define una clave de cifrado (clave pública) y una clave de descifrado (clave privada); la primera puede ser conocida por todo el mundo, pero, de otro lado se debe tener mucho cuidado en ocul-

tar la clave privada. Las claves asimétricas tienen la sorprendente propiedad de que lo que se está cifrando con una clave sólo se puede descifrar con la otra.

Existen pocos algoritmos asimétricos. El algoritmo Diffie-Hellman se basa en las matemáticas de logaritmos discretos y aunque no es tan exitoso como el RSA es un algoritmo de uso común. RSA fue inventado en el MIT por Rivest, Shamir y Adleman puede cifrar un mensaje de máximo 116 bytes (58 caracteres) y se basa en la factorización de dos números primos

Para explicar como funciona el cifrado asimétrico se utilizará RSA en el siguiente ejemplo (Figura A2), donde se seguirá los siguientes pasos:

1. Una vez digitado el texto en claro procedemos a presionar el botón "*Generar clave pública / privada*" cuyo resultado se observa en el cuadro de texto "*Par Claves*". Para ello matemáticamente se seguirán los siguientes pasos:

   - Seleccione dos números primos largos p y q de manera que $p \neq q$.
   - Calcule $n = pq$.
   - Calcule $\phi(n) = (p - 1)(q - 1)$.
   - Seleccione un entero positivo e tal que el $1 < e < \phi(n)$ tales que e y $\hat{E}(n)$ sean Primos entre sí .
   - Calcule d tal que $de = (mod\ \phi\ (n))$

$$d = \left(\frac{x(p-1)(q-1)+1}{e}\right)$$

$$de = mod(p \quad 1)(q \quad)$$

Donde:

- La clave privada será d y la clave pública será e.. Adicionalmente el parámetro n debe hacerse público.
- La clave pública consiste en: n, el módulo y e, el exponente público (a veces exponente de cifrado).
- La clave privada consiste en: n, el módulo, el cual es público y aparece en la clave pública; d, el exponente privado (a veces el exponente de descifrado), el cual debe permanecer oculto.

2. Luego presionamos el botón "*Cifrar Datos con clave pública*" y el texto en claro se cifrará utilizando el algoritmo RSA y se mostrará el resultado en el cuadro "*Texto cifrado*". Para ello internamente se aplicará:

$$c = m^e \quad (mod\ n)$$

Donde:

- m es el mensaje, c es el texto cifrado correspondiente a m

3. Seguidamente se presiona el botón "*Descifrar datos con clave privada*" y se mostrará su resultado en el cuadro "*Texto descifrado*". Para ello matemáticamente se aplica:

$$m \equiv c^d \pmod{n}$$
$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$$
$$m^{ed} = \pmod{p} \wedge m^{ed} = m \pmod{q}$$
$$m^{ed-1} = m \pmod{pq}$$ Así
$$c^d \equiv m \pmod{n}$$

A continuación se presenta las funciones [2] utilizadas en cada uno de los pasos realizados anteriormente:

1. CrearClaves() : ésta función crea y devuelve un nuevo par de claves pública/ privada. El par de claves pública/privada es una estructura de ocho campos; ésta estructura es una cadena XML y debe mantenerse de forma confidencial porque la clave privada es la que se utiliza para descifrar los datos.

```
Function CrearParClaves() As String
    Dim crsa As New RSACryptoServiceProvider()
    CrearParClaves = crsa.ToXmlString(True)
    crsa.Clear()
End Function
```

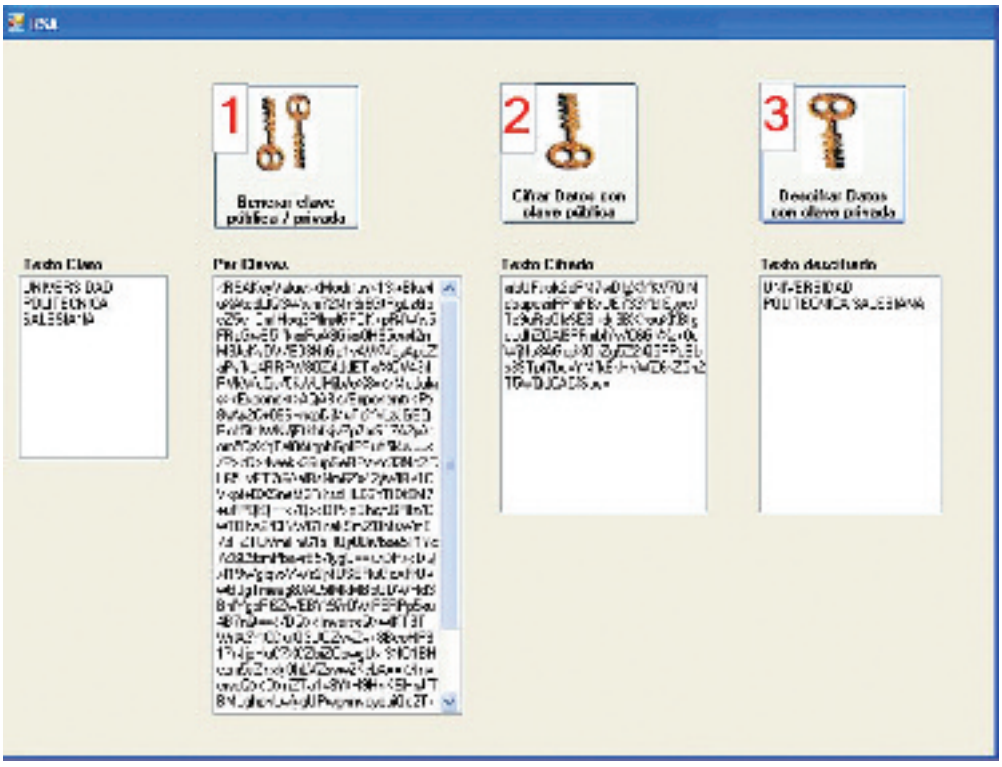

Figura A2. Cifrado Asimétrico

2. ObtenerClavePublica(clave): ésta función extrae la clave pública del par de claves y la devuelve en forma de cadena.

   Encriptar(textoClaro, clavePublica): esta función cifra el cuadro de texto "*Texto claro*" utilizando la clave pública y devueve una cadena al cuadro de texto "*Texto Cifrado*". Esta función puede cifrar una cadena de hasta 58 caracteres como máximo.

```
Function ObtenerClavePublica(ByVal clave As String) As String
    Dim crsa As New RSACryptoServiceProvider()
    crsa.FromXmlString(clave)
    Return crsa.ToXmlString(False)
End Function


Function Encriptar(ByVal textoClaro As String, ByVal clavePublica As String) As
String
    Dim crsa As New RSACryptoServiceProvider()
    Dim textoClaroByte() As Byte
    Dim textCifradoByte() As Byte
    Dim uCodifica As New UnicodeEncoding()
    crsa.FromXmlString(ClavePublica)
    textoClaroByte = uCodifica.GetBytes(textoClaro)
    TextCifradoByte = crsa.Encrypt(TextoClaroByte, False)
    Encriptar = Convert.ToBase64String(TextCifradoByte)
    crsa.Clear()
 End Function
```

3. Decrypt(textoCifrado, clavePrivada): ésta función descifra el cuadro de texto "*Texto Cifrado*" utilizando la clave privada y devuelve el texto descifrado.

```
Function Decrypt(ByVal textoCifrado As String, ByVal clavePrivada As String) As
String
    Dim crsa As New RSACryptoServiceProvider()
    Dim textoClaroByte() As Byte
    Dim textoCifradoByte() As Byte
    Dim uCodifica As New UnicodeEncoding()
    crsa.FromXmlString(clavePrivada)
    textoCifradoByte = Convert.FromBase64String(textoCifrado)
    textoClaroByte = crsa.Decrypt(textoCifradoByte, False)
    Decrypt = uCodifica.GetString(textoClaroByte)
    crsa.Clear()
 End Function
```

C. Ventajas y Desventajas

El beneficio del uso de un algoritmo simétrico radica en el procesamiento rápido para encriptar y desencriptar un alto volumen de datos. Las principales desventajas de los métodos simétricos son la distribución y gestión de claves.

Los beneficios de la criptografía asimétrica son la solución a los problemas de la criptografía simétrica, pues las claves públicas pueden ser distribuidas con toda tranquilidad, no valen de nada sin las claves privadas. La principal desventaja de la criptografía asimétrica es el procesamiento intenso y lento.

La combinación de ambas técnicas produce una solución segura, rápida, compacta, escalable, mayor facilidad de gestión de claves, resistencia a la interceptación.

El cifrado simétrico es una eficaz táctica de almacenamiento de información sensible en una base de datos, un registro o archivo El cifrado asimétrico se lo emplea muy frecuente para pasar con seguridad una clave privada, que posteriormente, será la que se utilice para cifrar y/o descifrar otra información.

"Suele ser un error muy frecuente pensar que los algoritmos de cifrado deben ser secretos para resultar seguros. Los algoritmos de cifrados utilizados son de dominio público y el código fuente asociado también. Sin embargo, siguen siendo seguros porque requieren que el usuario proporcione la clave secreta" [3].

BIBLIOGRAFÍA:

[1] Nash Andrew, Infraestructura de Claves públicas, Ed. McGraw Hill, México 2004.
[2] http://www.rsa.com/node.aspx?id=2972145
[3] De Jemas, .NET Framework Ed. McGraw Hill, México 2004.