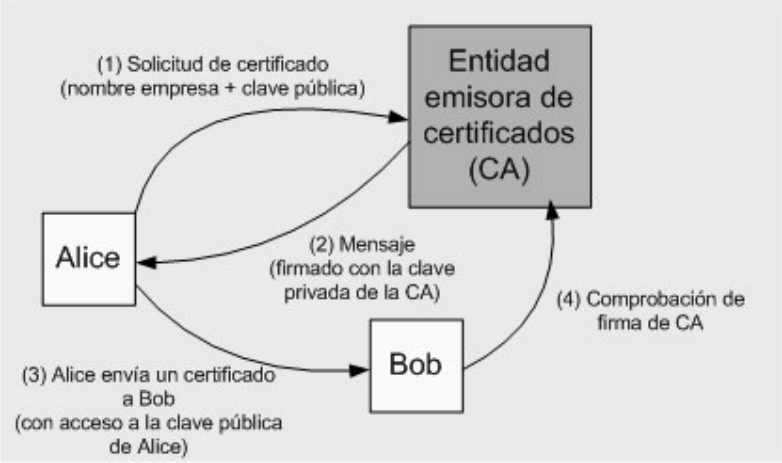
Criptografía, Claves y certificados con .Net FrameWork

El cifrado asimétrico utiliza un par de claves: pública y privada. Los datos cifrados con la clave privada pueden descifrarse únicamente con la clave pública correspondiente y viceversa.

Las claves públicas (tal y como sugiere su nombre) están disponibles para todo el mundo. Por el contrario, una clave privada se limita a una persona en concreto. El mecanismo de distribución por el que las claves públicas se transportan a los usuarios es el certificado. Los certificados suelen ir firmados por una entidad emisora de certificados (CA) para confirmar que la clave pública procede del sujeto que declara haberla enviado. La CA es una entidad de confianza mutua.

La implementación típica de certificación digital implica un proceso de firma del certificado. El proceso se muestra en la ilustración 1.



{Insert figure: REF – Digital Certification.gif}

Ilustración 1
Proceso de certificación digital

La secuencia de eventos que se muestra en la ilustración 1 es la siguiente:

1. Alice envía a una entidad emisora de certificados una solicitud de certificado firmado que contiene su nombre, su clave pública y quizás alguna información adicional.
2. La entidad emisora de certificados crea un mensaje a partir de la solicitud de Alice. La entidad emisora de certificados firma el mensaje con su clave privada, creando de este modo una firma independiente. A continuación, devuelve el mensaje y la firma a Alice. En conjunto, el mensaje y la firma forman el certificado de Alice.
3. Alice envía el certificado a Bob para darle acceso a su clave pública.

4. Bob comprueba la firma del certificado mediante la clave pública de la entidad emisora de certificados. Si la firma resulta ser válida, acepta la clave pública del certificado como la clave pública de Alice.

Como en el caso de la firma digital, cualquier destinatario con acceso a la clave pública de la CA puede determinar si la firma del certificado procede de una entidad emisora de certificados concreta. Este proceso no requiere ningún tipo de acceso a datos secretos. En el escenario anterior se supone que Bob tiene acceso a la clave pública de la entidad emisora de certificados. Tiene acceso a dicha clave si posee una copia del certificado de la CA que contiene la clave pública.

Certificados digitales X.509

Los certificados digitales X.509 no contienen únicamente el nombre de un usuario y la clave pública, sino también otra información acerca del usuario. Estos certificados son algo más que obstáculos en una jerarquía digital de confianza. Permiten a la CA proporcionar al destinatario de un certificado un medio de confianza de la clave pública del sujeto emisor del certificado y de otros datos acerca del mismo. Estos otros datos pueden ser, entre otras cosas, una dirección de correo electrónico, una autorización para firmar documentos de un determinado valor o la autorización para convertirse en una entidad emisora de certificados y firmar otros certificados.

Los certificados X.509 y muchos otros tienen un periodo de validez. Un certificado puede caducar y perder su validez. Una entidad emisora de certificados puede revocar un certificado por diversos motivos. Para controlar las revocaciones, la CA mantiene y distribuye una lista de certificados revocados denominada Lista de revocaciones de certificados (CRL). Los usuarios de la red tienen acceso a la lista para determinar la validez de un certificado.

Almacenes de certificados

Los certificados se almacenan en ubicaciones seguras denominadas almacenes de certificados. Un almacén de certificados puede contener certificados, CRL y Listas de certificados de confianza (CTL). Cada usuario tiene un almacén personal (denominado "MY store") donde se almacenan los certificados del usuario. El almacén MY store puede implementarse físicamente en varias ubicaciones, incluido el Registro, en un equipo local o remoto, un archivo de disco, una base de datos, un servicio de directorio, un dispositivo inteligente u otra ubicación.

Aunque cualquier certificado puede almacenarse en el almacén MY store, debería reservarse para los certificados personales de un usuario, es decir, los certificados que se utilizan para firmar y descifrar los mensajes de ese determinado usuario.

Además del almacén MY store, Windows mantiene los siguientes almacenes de certificados:

- **CA y ROOT**. Este almacén contiene los certificados de las autoridades emisoras de certificados en las que el usuario confía la emisión de certificados a otros usuarios. El sistema operativo proporciona un conjunto de certificados de entidades emisoras de certificados de confianza y los administradores pueden agregar otros.
- **Otro**. Este almacén contiene los certificados de otras personas con las que el usuario intercambia mensajes firmados.

CryptoAPI proporciona funciones para administrar certificados. Sólo es posible tener acceso a estas API a través de código no administrado. Además, CAPICOM es una API basada en COM para CryptoAPI, a la que se puede tener acceso a través de interoperabilidad COM.

Más información

Para obtener más información, consulte "Cryptography, CryptoAPI, and CAPICOM" en MSDN (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/Security/cryptography_cryptoapi_and_capicom.asp) (en inglés).

Criptografía

La criptografía se utiliza para proporcionar lo siguiente:

- **Confidencialidad**. Para garantizar que se mantiene la privacidad de los datos. La confidencialidad suele lograrse mediante el cifrado. Los algoritmos de cifrado (que utilizan claves de cifrado) se utilizan para convertir texto normal en texto cifrado y el algoritmo de descifrado equivalente se utiliza para convertir el texto cifrado de nuevo a texto normal. Los algoritmos de cifrado simétricos utilizan la misma clave para cifrar y descifrar, mientras que los algoritmos asimétricos utilizan un par de claves: pública y privada.
- **Integridad de los datos**. Para garantizar la protección de los datos frente a modificaciones accidentales o deliberadas (maliciosas). La integridad suelen proporcionarla los códigos de autenticación de mensajes o valores hash. Un valor hash es un valor numérico de longitud fija derivado de una secuencia de datos. Los valores hash se utilizan para comprobar la integridad de los datos que se envían a través de canales no seguros. Se compara el valor hash de los datos recibidos con el valor hash de los datos que se enviaron para determinar si se alteraron los datos.
- **Autenticación**. Para garantizar que los datos proceden de un determinado usuario. Los certificados digitales se utilizan para proporcionar autenticación. Las firmas digitales suelen aplicarse a valores hash, ya que suelen ser considerablemente menores que los datos de origen que representan.

Opciones técnicas

- Utilice un valor hash cuando necesite comprobar que los datos no han sufrido alteraciones durante el tránsito.
- Utilice un valor hash con clave cuando necesite demostrar que una entidad conoce un secreto sin enviar el secreto y volverlo a recibir, o cuando desee defenderse de intercepciones durante el tránsito mediante un valor hash sencillo.
- Utilice el cifrado cuando desee ocultar datos que se envían en un medio no seguro o al establecer la persistencia de los datos.
- Utilice un certificado cuando desee comprobar la identidad de la persona que declara ser propietario de la clave pública.
- Utilice el cifrado simétrico para mayor velocidad y cuando las dos partes conocen la clave.
- Utilice el cifrado asimétrico cuando desee intercambiar datos de forma segura a través de un medio no seguro.
- Utilice la firma digital cuando desee autenticar sin repudio.
- Utilice un valor salt (número aleatorio generado criptográficamente) para defenderse de los ataques de diccionario.

Criptografía en .NET

El espacio de nombres **System.Security.Cryptography** proporciona servicios criptográficos como codificar y descodificar datos, aplicar algoritmos hash, generar números aleatorios y autenticar mensajes.

.NET Framework proporciona implementaciones de numerosos algoritmos criptográficos estándar que pueden extenderse fácilmente gracias a la jerarquía de

herencia bien definida, formada por clases abstractas que definen los tipos básicos de algoritmos: simétricos, asimétricos y algoritmos hash, junto con las clases de algoritmos.

Tabla 1: Muestra los algoritmos para los que .NET Framework proporciona clases de implementación estándar.

Algoritmos simétricos	Algoritmos asimétricos	Algoritmos hash
DES (Estándar de cifrado de datos)	DSA (Algoritmo de firma digital)	HMAC SHA1 (Código de autenticación de mensajes basado en hash que utiliza el algoritmo hash SHA1)
TripleDES (Estándar de cifrado de datos triple)	RSA	MAC Triple DES (Código de autenticación de mensajes mediante Triple DES)
Rijndael		MD5
RC2		SHA1, SHA256, SHA384, SHA512 (Algoritmo hash seguro mediante diversos tamaños de valor hash)

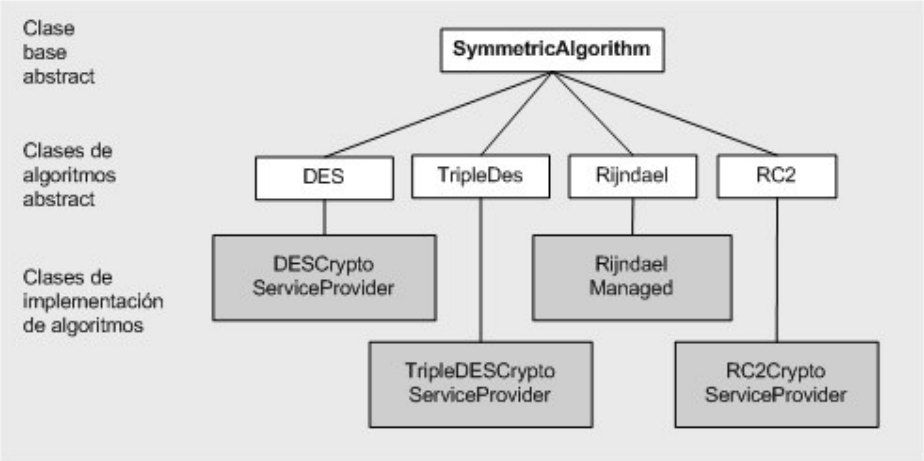
Compatibilidad con algoritmos simétricos

.NET proporciona las siguientes clases de implementación que ofrecen algoritmos simétricos de cifrado de claves secretas:

- DESCryptoServiceProvider
- RC2CryptoServiceProvider
- RijndaelManaged
- TripleDESCryptoServiceProvider

Nota: Las clases que tienen la terminación "CryptoServiceProvider" son contenedores que utilizan los servicios subyacentes del proveedor de servicios de cifrado (CSP) y las clases que tienen la terminación "Managed" se implementan en código administrado.

La ilustración 2 muestra la jerarquía de herencia que adopta .NET Framework. La clase base del tipo de algoritmo (por ejemplo, **SymmetricAlgorithm**) es abstracta. Un conjunto de clases abstractas para los distintos algoritmos se derivan de la clase base de algoritmo, también de tipos abstracto. Las clases de implementación de algoritmos proporcionan implementaciones concretas del algoritmo seleccionado; por ejemplo, DES, Triple-DES, Rijndael y RC2.



{Insert figure: REF – Crypto Object Model.gif}

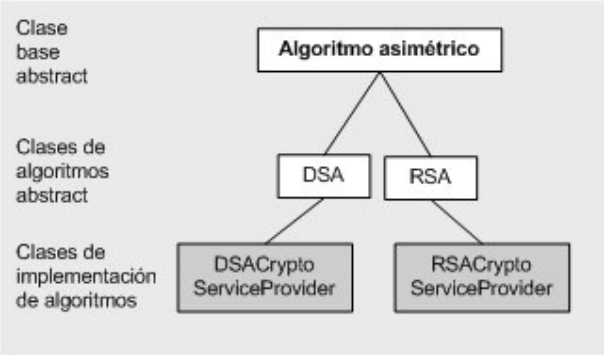
Ilustración 2
La jerarquía de herencia de la clase crypto simétrica

Compatibilidad con algoritmos asimétricos

.NET proporciona los siguientes algoritmos de cifrado asimétricos (clave pública/privada) a través de la clase base abstracta (**System.Security.Crytography.AsymmetricAlgorithm**):

- DSACryptoServiceProvider
- RSACryptoServiceProvider

Se utilizan para firmar y cifrar datos digitalmente. En la ilustración 3 se muestra la jerarquía de herencia.



{Insert figure: REF–Crypto Object Model (Asymmetric).gif}

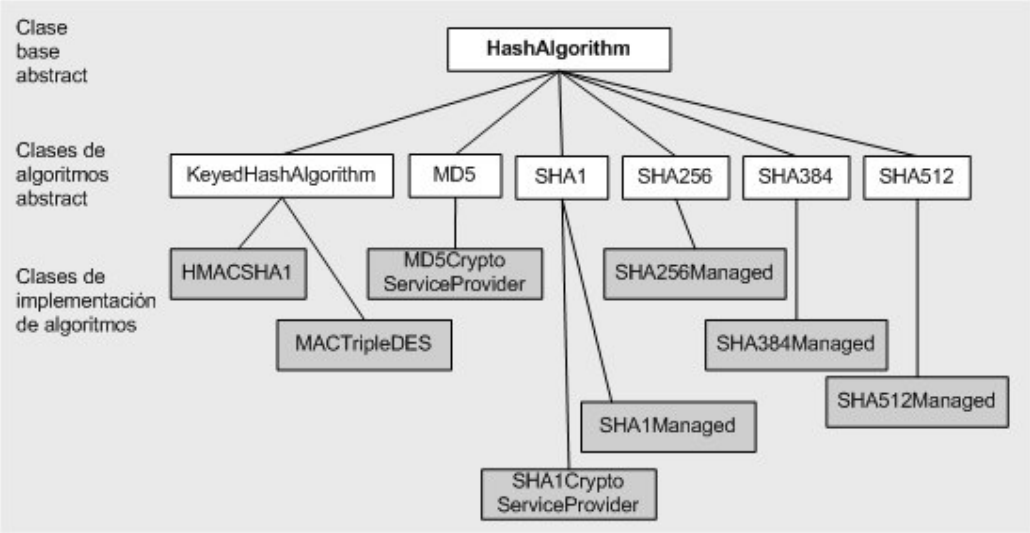
Ilustración 3
La jerarquía de herencia de la clase crypto asimétrica

Compatibilidad con algoritmos hash

.NET proporciona los siguientes algoritmos hash:

- SHA1, SHA256, SHA384, SHA512
- MD5
- HMACSHA (Algoritmo hash con clave)
- MACTripleDES (Algoritmo hash con clave)

La ilustración 4 muestra la jerarquía de herencia de las clases de algoritmos hash.



{Insert figure: REF – Crypto Object Model (Hashing).gif}

Ilustración 4
La jerarquía de herencia de la clase crypto hash

Resumen

La criptografía es una tecnología importante para crear aplicaciones Web seguras. Este apéndice ha tratado algunos de los conceptos básicos de los certificados y la criptografía y ha introducido algunas de las clases expuestas por el espacio de nombres **System.Security.Cryptography**, que permiten incorporar con mayor facilidad soluciones de seguridad criptográfica en las aplicaciones .NET.

Para obtener más información acerca de la criptografía en .NET, busque en MSDN la página titulada ".NET Framework Cryptography Model" (en inglés).