Сохранение ключей в формате XML

Петелин А.Е.

```
RSACryptoServiceProvider rsa =
        new RSACryptoServiceProvider();
```

Запись в XML

```
//открытый и секретный ключи RSA
string publicPrivateKeyXML =
        rsa. ToXmlString(true) ;
//только открытый ключ RSA
string publicOnlyKeyXML =
        rsa.ToXmlString(false);
```

Чтение из XML

```
rsa. FromXmlString (publicPrivateKeyXML) ;
```

Цифровые сертификаты

Цифровой сертификат - это документ, который удостоверяет вашу личность в сообщениях или транзакциях через посредство Internet.
Существует множество коммерческих центров сертификации (например, Verysign), а сама сертификация существует во многих вариантах, отличающихся ценой и уровнем доверия.
В Microsoft Outlook для получения доступа к центру сертификации и создания сертификата, необходимо выбрать:
Сервис / Параметры / Безопасность /

Цифровая подпись

Асимметричные алгоритмы используются не только для достижения конфиденциальности, но и для аутентификации, контроля целостности и подтверждения обязательств.
Для цифровой подписи требуется «криптографический хеш».
Хеш - это функция, которая ставит в соответствие небольшой, фиксированного размера объем двоичных данных произвольному, сколь угодно большому объему входных данных. Хеш называют также дайджестом сообщения или отпечатком пальца.

Характеристики хорошей хеш-функции

Любая хорошая хеш-функция должна обладать следующими свойствами:

- входные данные могут обладать произвольными размерами;
- выходные данные всегда обладают небольшим, фиксированным размером, вытекающим из используемого алгоритма;
- функция быстро вычисляется;
- ее трудно обратить (то есть это односторонняя функция);
- вероятность возникновения коллизий невелика.

Хеш-алгоритмы, поддерживаемые в .NET

Наиболее часто используются хеш-алгоритмы **SHA-1** и **MD5**. Алгоритм SHA-1 создает 160-битовый хеш, а алгоритм MD5 - 128-битовый. Для достижения более высоких степеней безопасности могут использоваться **SHA-256**, **SHA-384** и **SHA-512**, создающие, соответственно, значения размером 256, 384 и 512 бит.

- HashAlgorithm
  - KeyedHashAlgorithm
    - HMACSHA1
    - MACRtipleDES
  - MD5
    - MD5CryptoServiceProvider
  - SHA1
    - SHA1CryptoServiceProvider
    - SHA1Managed
  - SHA256
    - SHA256Managed
  - SHA384
    - SHA384Managed
  - SHA512
    - SHA512Managed

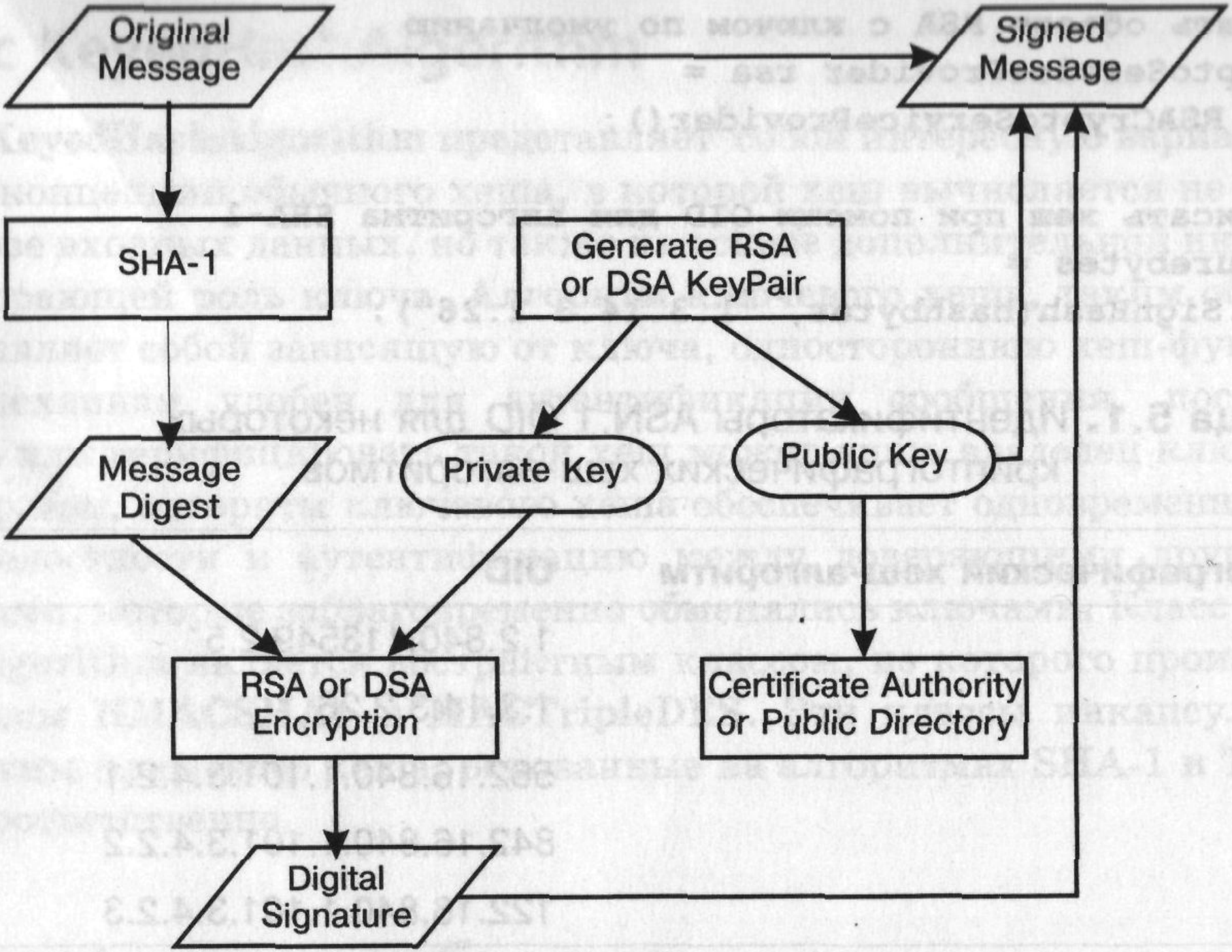
Классы, имена которых заканчиваются на CryptoServiceProvider, реализованы с использованием интерфейса CryptoAPI, предоставляемого операционной системой.
Классы, имени которых заканчиваются на Managed, реализованы полностью средствами контролируемого C#-кода, без использования CryptoAPI.
Класс HMACSHA1 производит ключевой хеш при помощи хеш-функции SHA1.
Класс MACTripleDES производит ключевой хеш при помощи шифровалия Triple DES.

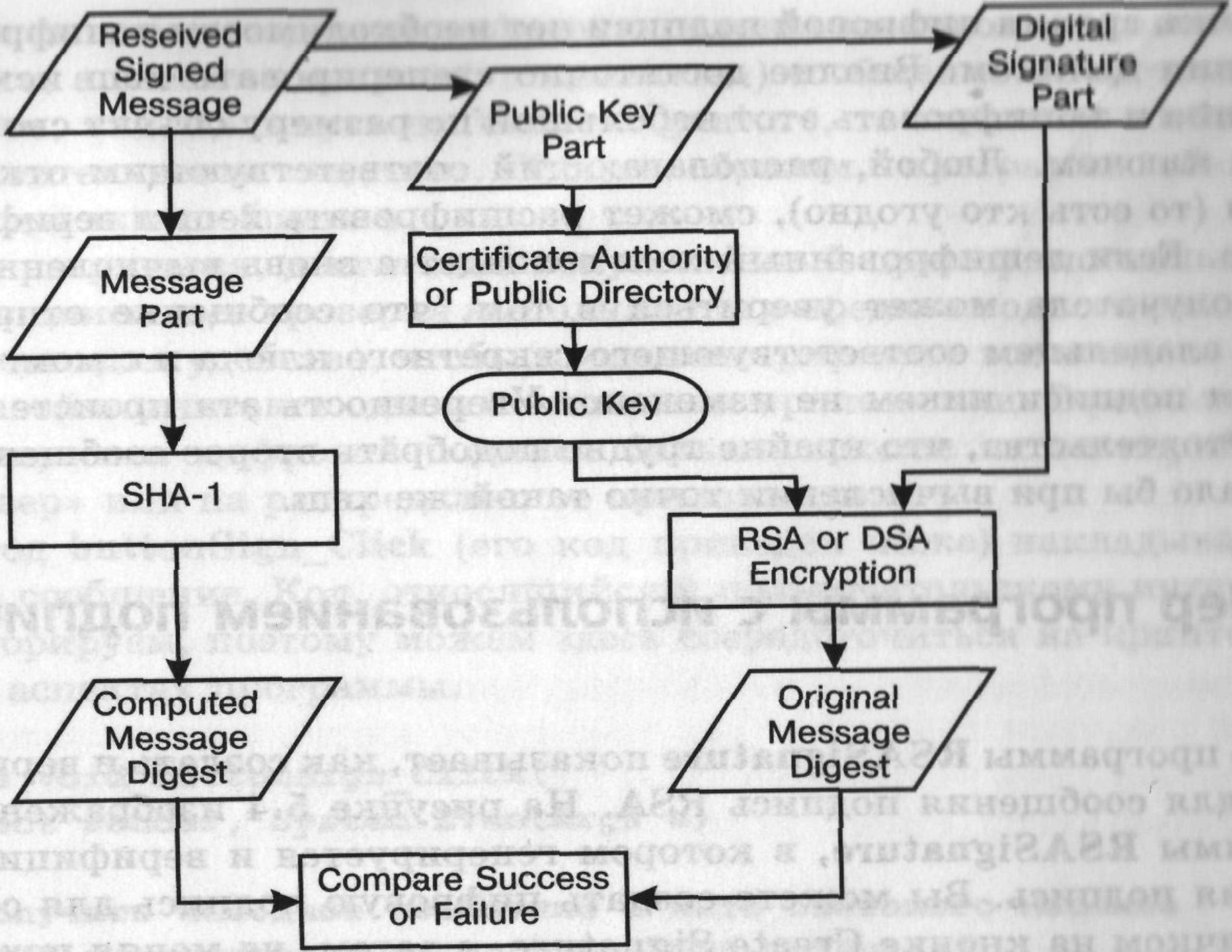
Метод вычисления хеша в .NET

```
HashAlgorithm sha1 = new
SHA1CryptoServiceProyider();
byte[]  sha1Hash =
sha1.ComputeHash(messageByteArray);
```

Идентификаторы объектов

Хеш-алгоритм	OID
MD5	1.2.840.113549.2.5
SHA-1	1.3.14.3.2.26
SHA-2	562.16.840.1.101.3.4.2.1
SHA-3	842.16.840.1.101.3.4.2.1
SHA-5	122.16.840.1.101.3.4.2.1


Original Message
Signed Message
SHA-1
Generate RSA or DSA KeyPair
Message Digest
Private Key
Public Key
RSA or DSA Encryption
Certificate Authority or Public Directory
Digital Signature


Reseived Signed Message
Digital Signature Part
Public Key Part
Message Part
Certificate Authority or Public Directory
Public Key
SHA-1
RSA or DSA Encryption
Computed Message Digest
Original Message Digest
Compare Success or Failure