УДК 004.4

Возможности реализации алгоритмов ЭЦП на основе .NET Framework Security

Белова С.В.
Белорусский национальный технический университет

Аутентификация сообщений защищает две обменивающиеся сообщениями стороны от любой третьей, но не обеспечивает защиту каждой из сторон от другой. Поэтому в ситуациях, когда нет полного доверия между отправителем и получателем, требуется нечто большее, чем аутентификация. Наиболее привлекательное решение проблемы – использование электронной цифровой подписи (ЭЦП).

Технология применения электронной цифровой подписи предполагает наличие сети абонентов, посылающих друг другу подписанные электронные документы. Для каждого абонента генерируется пара ключей: открытый и личный. Личный ключ хранится абонентом в тайне и используется для формирования ЭЦП. Берется исходное сообщение и создается его хеш-код при помощи одного из алгоритмов хеширования. Затем хеш-код шифруется при помощи личного ключа отправителя сообщения. Результат шифрования и есть ЭЦП. Открытый ключ известен всем другим пользователям и предназначен для проверки (верификации) ЭЦП получателем документа.

В библиотеке классов .NET Security поддерживаются асимметричные алгоритмы DSA и RSA. В основе всех асимметричных алгоритмов лежит класс AsymmetricAlgorithm. Класс AsymmetricAlgorithm располагается в пространстве имен System Security.Cryptography и является абстрактным классом. Из него производятся классы алгоритмов: RSA и DSA, которые также являются абстрактными. Из классов RSA и DSA затем производятся классы RSACryptoServiceProvider и DSACryptoServiceProvider, которые обеспечивают реализацию алгоритмов. Эти классы являются оболочками для Microsoft Crypto API. Способы работы с электронной цифровой подписью с помощью классов DSACryptoServiceProvider и RSACryptoServiceProvider практически идентичны. Публичные методы и свойства классов DSACryptoServiceProvider и RSACryptoServiceProvider также во многом аналогичны. Методы позволяют, например, вычислить хеш-код сообщения и подписать его, верифицировать заданную подпись, сравнив ее с подписью, вычисленной для заданного сообщения и т.д. Конструктор класса автоматически генерирует ключевую информацию в момент создания экземпляра. С помощью публичных свойств можно получить имя алгоритма обмена ключами, разрешенные размеры ключей, размер ключа в битах и другие параметры ЭЦП.