Учреждение образования “Белорусский государственный технологический yниверситет”

Практическая работа №14 «Изучение стандартных средств для реализации приложений, использующих симметричное и ассиметричное шифрование с использованием библиотеки System.Security.Cryptography»

Выполнила:

студентка 2 курса, 7 группы

Колядко Яна Дмитриевна

Барковский Евгений Валерьевич

Цель**:** изучить модель криптографии .NET Framework, основные классы и структуры данных, разработать приложение для шифрования файлов, использующих симметричные и ассиметричные алгоритмы шифрования.

1. Какие симметричные алгоритмы шифрования Вы знаете?

- Простая перестановка

- Одиночная перестановка по ключу

- Двойная перестановка

- Перестановка «Магический квадрат»

1. Какие ассиметричные алгоритмы шифрования Вы знаете?

• RSA (аббревиатура от Rivest, Shamir и Adelman, фамилий создателей алгоритма) — алгоритм, в основе которого лежит вычислительная сложность факторизации (разложения на множители) больших чисел. Применяется в защищенных протоколах SSL и TLS, стандартах шифрования, например в PGP и S/MIME, и так далее. Используется и для шифрования данных, и для создания цифровых подписей.

• DSA (Digital Signature Algorithm, «алгоритм цифровой подписи») — алгоритм, основанный на сложности вычисления дискретных логарифмов. Используется для генерации цифровых подписей. Является частью стандарта DSS (Digital Signature Standard, «стандарт цифровой подписи»).

• Схема Эль-Гамаля — алгоритм, основанный на сложности вычисления дискретных логарифмов. Лежит в основе DSA и устаревшего российского стандарта ГОСТ 34.10–94. Применяется как для шифрования, так и для создания цифровых подписей.

• ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм, основанный на сложности вычисления дискретного логарифма в группе точек эллиптической кривой. Применяется для генерации цифровых подписей, в частности для подтверждения транзакций в криптовалюте Ripple.

1. Основное назначение библиотеки System.Security.Cryptography?

Пространство имен System. Security. Cryptography открывает программный доступ к самым разнообразным криптографическим сервисам, с помощью которых приложения могут шифровать и дешифровать данные, обеспечивать их целостность, а также обрабатывать цифровые подписи и сертификаты.

1. Влияет ли размер ключа на криптостойкость алгоритма?

Влияет.

Сильным называют шифрование с длиной симметричного ключа 128 бит. На любом современном оборудовании реконструкция такого ключа занимает времени в миллионы раз больше, чем возраст Вселенной. Это технически невозможное мероприятие, если нет каких-либо дополнительных данных, например сведений о характерных настройках средства ЭЦП, использованного при генерации ключа. Теоретически такие сведения у “взломщика” могут быть, и тогда реконструкция даже сильного ключа может быть технически возможной.

Для ключей несимметричного шифрования получить столь простую формулу, как для симметричных ключей, как правило, не удается. Алгоритмы несимметричного шифрования еще не до конца изучены (в этом нет ничего удивительного, поскольку по сей день не изучены даже свойства таких “простых” математических объектов, как простые числа). Поэтому при использовании несимметричного шифрования говорят об относительной криптостойкости ключей. Понятно, что, как и для симметричных ключей, их криптостойкость зависит от длины, но выразить это соотношение простой формулой для большинства алгоритмов пока не удалось. Обычно относительную криптостойкость оценивают по эмпирическим данным, полученным опытным путем.

1. Назовите основные классы библиотеки System.Security.Cryptography?

1. **SymmetricAlgorithm** является абстрактным базовым классом, от которого наследуют другие классы, специфичные для конкретных алгоритмов. К числу поддерживаемых симметричных алгоритмов относятся Data Encryption Standard (DES), RC2, Rijndael, Advanced Encryption Standard (AES) и Triple Data Encryption Standard (TripleDES).

2. К общеизвестным асимметричным алгоритмам относятся **Digital Signature Algorithm** (DSA) и RSA. Эти алгоритмы в конечном счете являются производными от абстрактных классов DSA и RSA, в свою очередь производных от AsymmetricAlgorithm. Так как эти алгоритмы очень сложны, им нужны вспомогательные классы, производные, например, от AsymmetricKeyExchangeFormatter и AsymmetricSignatureFormatter.

3. За обмен сессионными ключами в .NET отвечают классы **RSAOAEPKeyExchangeFormatter/Deformatter** и **RSAPKCS1KeyExchangeFormatter/Deformatter.** Они унаследованы от базовых классов AsymmetricKeyExchangeFormatter/Deformatter, предоставляющих методы CreateKeyExchange и DecryptKeyExchange для шифрования и дешифрации сессионных ключей, соответственно.

4. Пространство имен Cryptography содержит базовый класс **HashAlgorithm** и производные классы, поддерживающие алгоритмы MD5, SHA1, SHA256, SHA384 и SHA512.

Вывод**:** в данной лабораторной работе изучили модель криптографии .NET Framework, основные классы и структуры данных, разработали приложение для шифрования файлов, использующих симметричные и ассиметричные алгоритмы шифрования.