УДК 004.056.2

А.Г. Проценко

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

ИССЛЕДОВАНИЕ БЫСТРОДЕЙСТВИЯ АЛГОРИТМОВ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ НА БАЗЕ ТЕХНОЛОГИИ .NET FRAMEWORK

Разработано программное средство, позволяющее производить исследование быстродействия алгоритмов обеспечения целостности данных, реализованных на базе технологии .NET Framework, и проведено исследование стандартных реализаций криптоалгоритмов .NET. Определены реализации алгоритмов цифровой подписи и хэш-функций, обладающие наибольшим быстродействием.

Ключевые слова: *алгоритмы обеспечения целостности данных, быстродействие, хэш-функции, криптопреобразования, цифровые подписи.*

Введение. Постановка задачи

В высокотехнологичном информационном обществе использование средств защиты информации является жизненно необходимым, поскольку это позволяет реализовать такие свойства защищаемых данных, как конфиденциальность, целостность, аутентичность, которые обеспечиваются с помощью методов криптографии.

Большинство задач защиты информации решаются специализированными программными средствами (ПС). Наряду с коммерческими ПС защиты информации, а также ПС, встраиваемыми в некоторые приложения, для выполнения специфических задач существует возможность использовать готовые программные реализации библиотек, реализующих криптографические алгоритмы защиты данных. Такие библиотеки не только предоставляют набор уже готовых криптоалгоритмов, но и определяют удобный интерфейс для расширения и модернизации уже существующих реализаций криптоалгоритмов и написания новых. Последний подход связан с тем, что большинство современных компаний стараются разработать универсальные криптографические интерфейсы и оградить разработчика программного обеспечения от самостоятельных реализаций сложных алгоритмов. Примером тому является интерфейсы Crypto API и Cryptography New Generation (CNG) семейства операционных систем Windows компании Microsoft, а встроенные средства технологии Microsoft .NET Framework предоставляют большой набор классов для осуществления различных криптопреобразований, позволяющих организовывать собственную систему криптозащиты данных [1].

Вопросы быстродействия криптоалгоритмов на базе .NET Framework рассматриваются в работе [2], однако предметом исследования является исключительно быстродействие алгоритмов обеспечения конфиденциальности данных. Исследования быстродействия других реализаций криптоалгоритмов .NET ранее не проводились.

Цель данной работы – провести исследование временных характеристик алгоритмов криптопреоб-

разования, используемых для обеспечения *целостности* данных, на основе технологии .NET Framework.

Обзор возможностей среды .NET по реализации криптоалгоритмов

Выбор криптоалгоритмов шифрования данных осуществлялся исходя из возможностей библиотек System.Security.Cryptography каркаса .NET Framework. Были включены в анализ реализации бесключевых хэш-функций и алгоритмов цифровых подписей, доступных для использования в каркасе .NET. Список доступных алгоритмов отличается для различных версий ОС Windows. Отличия в наборе доступных для использования классов представлены в табл. 1.

Как видно из таблицы, наибольший набор доступных алгоритмов цифровой подписи и хэш-алгоритмов реализован для ОС Windows Vista и Windows 7, в которых установлен обновлённый каркас шифрования данных CNG (Cryptography Next Generation). Однако стоит отметить, что для более ранних ОС .NET предоставляет реализации аналогичных криптоалгоритмов, выполненных полностью в managed-коде. Исключение составляют алгоритмы на эллиптических кривых. Реализации алгоритмов EC DSA и EC Diffie-Hellman доступны на данный момент только для ОС Windows Vista и выше.

Структура классов реализаций алгоритмов обеспечения целостности данных представлена на рис. 1.

Таблица 1

Отличия в наборе доступных для использования классов

Название криптоалгоритма		Windows XP	Windows Server 2003	Windows Vista, Windows 7
Алгоритмы цифровой подписи	RSA Crypto API	+	+	+
	DSA Crypto API	+	+	+
	EC DSA CNG			+
Хэш-функции	MD5 CryptoAPI	+	+	+
	MD5 CNG			+
	SHA1Crypto API	+	+	+
	SHA1 Managed	+	+	+
	SHA1 CNG			+
	SHA256 Crypto API		+	+
	SHA256 Managed	+	+	+
	SHA256 CNG			+
	SHA384 Crypto API		+	+
	SHA384 Managed	+	+	+
	SHA384 CNG			+
	SHA512 Crypto API		+	+
	SHA512 Managed	+	+	+
	SHA512 CNG			+
	RIPE MD160 Managed	+	+	+

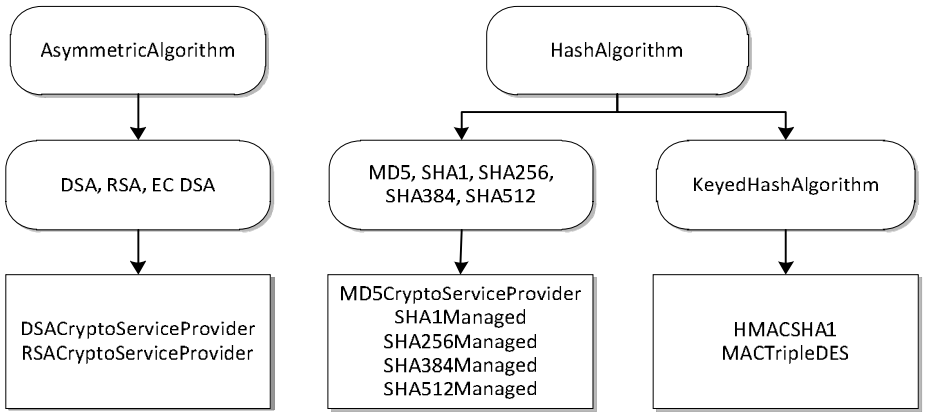

Рис. 1. Структурная схема классов реализаций алгоритмов обеспечения целостности данных

Базовые классы AsymmetricAlgorithm и HashAlgorithm предоставляют интерфейс, реализуемый всеми классами алгоритмов цифровых подписей и хэш-алгоритмов.

Класс KeyedHashAlgorithm расширяет реализации доступных хэш-функций возможностью осуществлять вычисление хэшей документов с использованием дополнительного ключа. В остальном наследники данного класса копируют функциональность соответствующих реализаций бесключевых хэш-функций.

Для классов криптографии .NET существует разделение на среды, в которых был реализован код

алгоритма. Выделяются классы реализаций для встроенных каркасов Windows CryptoAPI, CNG (Cryptography Next Generation) и Managed. Первые два типа выступают оболочками для библиотек Windows, тогда как managed-реализации выполняются исключительно виртуальной машиной .NET, поэтому для данных классов возможна потеря быстродействия за счёт операции компилирования байт-кода .NET в машинный код компьютера.

Более подробно структура классов криптографии .NET описана в работе [2].

Изложение и анализ результатов

Для исследования быстродействия криптоалгоритмов обеспечения целостности данных производилось измерение времени, затрачиваемого на произведение криптопреобразований в зависимости от размера документа.

В качестве операционной системы для проведения исследования была выбрана ОС Microsoft Windows 7, которая на данный момент обладает наибольшим набором доступных для исследования криптоалгоритмов. Поскольку в данной операционной системе наряду с встроенными managed-классами доступны реализации CryptoAPI и CNG, для бесключевых хэш-функций был проведён сравнительный анализ быстродействия различных реализаций одного криптоалгоритма.

В качестве характеристики времени, затрачиваемого на процесс шифрования, здесь и далее использовалась временная единица тик (tick). Microsoft в своём каркасе определяет тик, как наименьшую единицу измерения времени: 1 тик приравнивается к 100 наносекундам.

При выполнении одноразового анализа на графиках видны резкие пики, обусловленные в основном распределением процессорного времени на системные нужды (ресурсы ОС и фоновых процессов). Так как криптопреобразования – ресурсоёмкая задача, доступность ресурсов системы, таких, как процессорное время и оперативная память, влияет на результат анализа. Для сглаживания результирующих данных производилось усреднение их значений по результатам десяти замеров.

В каркасе .NET Framework алгоритмы бесключевой хэш-функции MD5 представлены реализациями Crypto API и CNG. Результаты анализа быстродействия показаны на рис. 2.

Реализация криптоалгоритма на базе CNG работает несколько медленнее аналогичной реализации для Crypto API, однако разница во времени криптопреобразования незначительна даже при обработке документов большого объёма.

Хэш-функция SHA1 представлена в .NET тремя реализациями, одна из которых полностью написана в managed-коде. Как видно из рис. 3, managed-реализация криптоалгоритма работает приблизительно в 2,5 раза медленнее, чем реализации того же алгоритма CNG и Crypto API. Разница в производительности алгоритмов CNG и Crypto API незначительна.

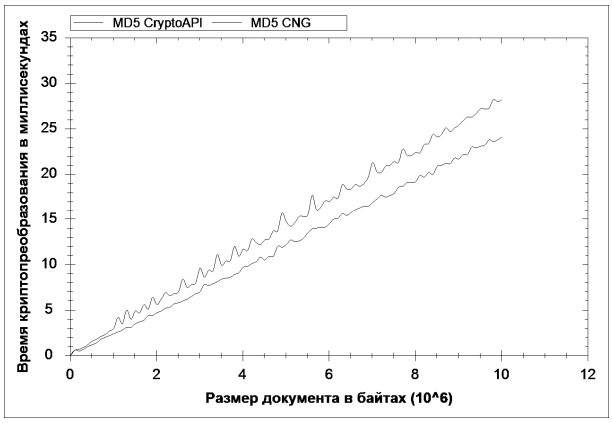

Рис. 2. Результаты анализа реализаций хэш-функции MD5 для документов разного объёма

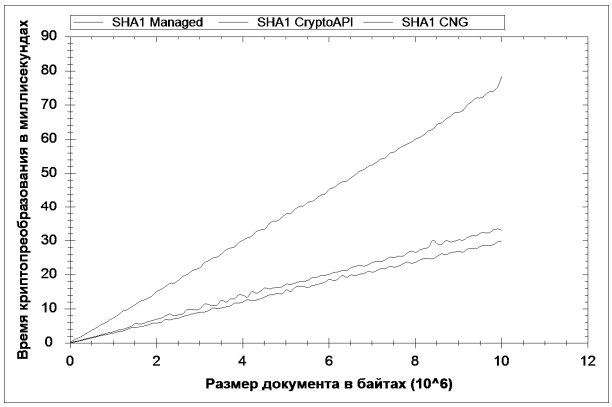

Рис. 3. Результаты анализа реализаций хэш-функции SHA1 для документов разного объёма

На рис. 4 представлен результат оценки быстродействия бесключевой хэш-функции SHA256. Реализация алгоритма .NET в managed-коде работает в 3 раза медленнее аналогичных реализаций встроенных средств Windows. Стоит отметить, что для использования данного алгоритма в ОС Windows XP доступна только реализация managed, поскольку поддержка данного криптоалгоритма на базе Crypto API была добавлена только в ОС Windows Server 2003. Никакой разницы в быстродействии неуправляемых реализаций данного алгоритма (CNG и Crypto API) не отмечено.

На рис. 5 представлены результаты анализа быстродействия криптоалгоритма SHA384. Данная хэш-функция также имеет три реализации в .NET, две из которых реализованы встроенными средствами Windows и только одна доступна в использовании для Windows XP. Как и в предыдущем случае, время работы криптоалгоритмов растёт линейно пропорционально размеру обрабатываемого документа и время работы managed-реализации пример-

но в 3 раза выше превышает время работы встроенных реализаций Windows.

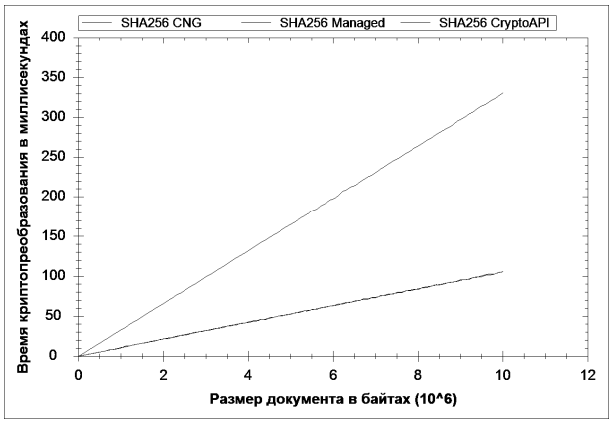

Рис. 4. Результаты анализа реализаций хэш-функции SHA256 для документов разного объёма

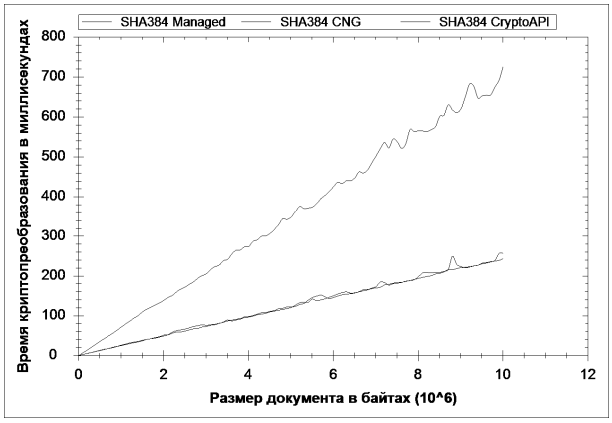

Рис. 5. Результаты анализа реализаций хэш-функции SHA384 для документов разного объёма

Результаты анализа производительности алгоритма SHA512 можно увидеть на рис. 6. Как видно из рисунка, также как и для предыдущих реализаций наблюдается чёткая линейная зависимость времени криптопреобразования от объёма документа.

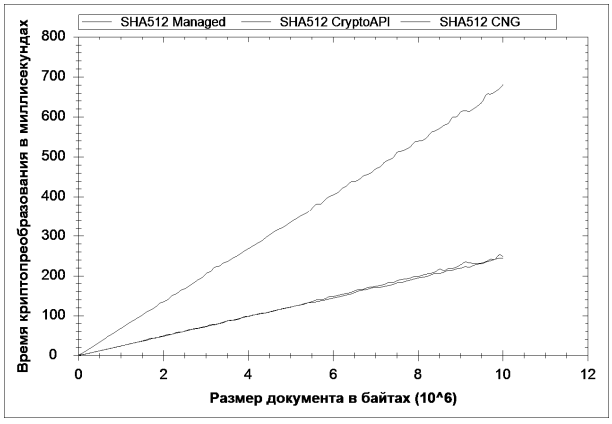

Рис. 6. Результаты анализа реализаций хэш-функции SHA512 для документов разного объёма

Результат сравнительного анализа быстродействия рассмотренных хэш-функций на базе CNG представлен на рис. 7.

Наилучшим быстродействием обладают алгоритмы MD5 и SHA1, показывая наименьшее время работы среди всех рассмотренных алгоритмов хэш-функций. Алгоритмы SHA384 и SHA512 показывают практически одинаковое время работы, наибольшее среди рассмотренных алгоритмов. Время работы этих алгоритмов в общем случае превышает время работы алгоритмов SHA1 и MD5 приблизительно в 10 раз, однако данные реализации обеспечивают гораздо большую криптозащищённость, ввиду использования б*о*льшего размера дайджеста и оперирования внутренними состояниями большего размера. Одинаковое время работы реализаций SHA384 и SHA512 обусловлено незначительной разницей в реализации алгоритмов на практике. Алгоритм SHA256 показывает в среднем время работы в 3 раза больше, чем у реализаций SHA1 и MD5, однако гораздо меньше, чем у реализаций SHA384 и SHA512.

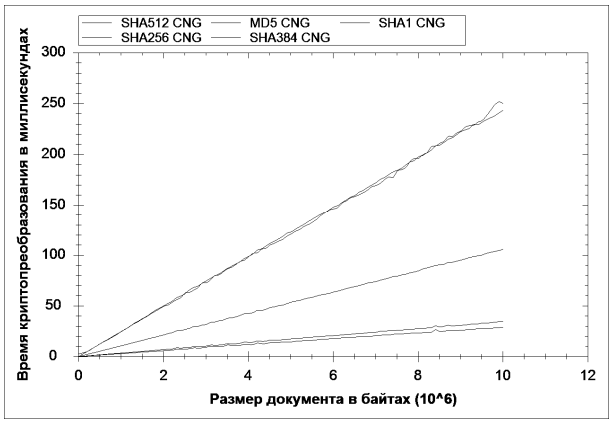

Рис. 7. Сравнительный график результатов анализа реализаций бесключевых хэш-функций для документов разного объёма

На рис. 8 представлены результаты анализа быстродействия алгоритмов цифровой подписи .NET. В существующей версии доступны три реализации алгоритмов цифровой подписи: RSA, DSA и DSA на эллиптических кривых (ECDSA).

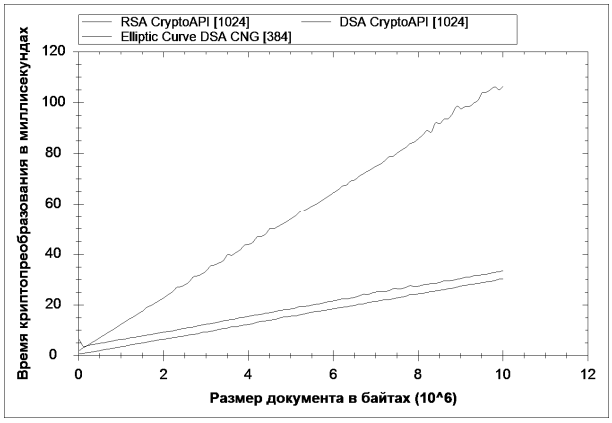

Рис. 8. Результаты анализа реализаций алгоритмов цифровой подписи для документов разного объёма

Алгоритм DSA с длиной ключа 1024 бита показал наиболее высокое быстродействие по сравнению с другими рассмотренными алгоритмами, хотя разница во времени работы с алгоритмом RSA незначительна и с ростом объёма документа не изменяется. Алгоритм ECDSA показывает значительный рост времени работы по сравнению с другими алгоритмами цифровой подписи. Для документа объёмом 10 Мбайт время работы алгоритма ECDSA в 5 раз выше, чем у других алгоритмов. Однако стоит заметить, что это единственная реализация алгоритмов цифровой подписи на эллиптических кривых.

Заключение

1. В ходе данной работы было разработано программное средство для исследования быстродействия алгоритмов обеспечения целостности данных на базе технологии Microsoft .NET Framework.

2. В ходе анализа быстродействия было отмечено, что классы реализаций, написанные полностью в managed-коде, работают в среднем в 3 раза дольше для документов того же объёма, чем аналогичные, использующие встроенные возможности Windows.

3. Наибольшим быстродействием среди реализаций хэш-функций обладают алгоритмы MD5 и SHA1, однако они обеспечивают наименьшую защищённость.

4. Наименьшее быстродействие среди реализаций хэш-функций показали алгоритмы SHA384 и SHA512, время работы может превышать время работы самых быстрых алгоритмов в 10 раз. Однако данные алгоритмы имеют наибольшую криптостойкость.

5. Алгоритмы цифровых подписей RSA и DSA обладают соизмеримым быстродействием при одинаковых длинах ключа.

6. Алгоритм DSA на эллиптических кривых работает значительно дольше аналогов. Для установки цифровой подписи на документ одного объёма для алгоритма ECDSA потребуется в 5 раз больше времени по сравнению с реализациями DSA и RSA. В то же время, при этом обеспечивается более высокая криптостойкость для одинаковых значений длины ключа.

7. Разработанное программное средство может быть использовано при проведении лабораторных занятий для изучения алгоритмов шифрования данных по дисциплинам направления «Информационная безопасность».

Дальнейшие исследования могут быть связаны со сравнительным анализом быстродействия алгоритмов обеспечения конфиденциальности и целостности данных, реализованных в разных каркасах программирования (например, .NET и Java).

Список литературы

1. Авдошин С.М. Криптотехнологии Microsoft / С.М. Авдошин, А.А. Савельева // Приложение к журналу «Информационные технологии». – 2008. – №9. – С. 23-30.

2. Проценко А.Г. Исследование быстродействия алгоритмов шифрования на базе технологии .NET Framework / А.Г. Проценко, И.В. Лысенко // Системи обробки інформації. – Х.: ХУПС, 2011. – Вип. 4(94). – С. 175-181.

3. Нортроп Т. Разработка защищенных приложений на VisualBasic .NET и VisualC#.NET: Учебный курс Microsoft / Т. Нортроп. – М.: Русская Редакция, 2007. – 688 с.

4. Использование криптографии с помощью API CNG в Windows Vista. [Электронный ресурс]. – Режим доступа к ресурсу: http://msdn.microsoft.com/ru/magazine/cc163389.aspx.

5. Шнайер Б. Прикладная криптография: протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М.: Триумф, 2002. – 820 с.

Поступила в редколлегию 17.10.2011

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ДОСЛІДЖЕННЯ ШВИДКОДІЇ АЛГОРИТМІВ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ НА БАЗІ ТЕХНОЛОГІЇ .NET FRAMEWORK

О.Г. Проценко

Розроблено програмний засіб, що дозволяє проводити дослідження швидкодії алгоритмів забезпечення цілісності даних, що реалізовані на базі технології .NET Framework, і досліджені стандартні реалізації криптоалгоритмів .NET. Визначені реалізації алгоритмів цифрового підпису та хеш-функцій, що мають найбільшу швидкодію.

Ключові слова: *алгоритми забезпечення цілісності даних, швидкодія, хеш-функції, криптоперетворення, цифрові підписи.*

TESTING OF THE PERFORMANCE OF DATA INTEGRITY ALGORITHMS BASED ON .NET FRAMEWORK TECHNOLOGY

O.G. Protsenko

The application developed, which allows studying performance of data integrity algorithms based on .NET Framework technology and performance of standard implementations of .NET algorithms is investigated. Digital signatures and hash-functions biggest performance realizations are determined.

Keywords: *data integrity algorithm, performance, hash-functions, crypto converting, digital signatures.*