Камышников Игорь Георгиевич
2006/03/28

Эссе по курсу «Защита информации», кафедра радиотехники, Московский Физико-Технический Институт (МФТИ ГУ)
www.re.mipt.ru/infsec

Обзор Microsoft CryptoAPI

1 Введение

Многие программисты в процессе своего профессионального роста рано или поздно сталкиваются с проблемой защиты своих программ от взлома. Он проявляется в самых разных местах готового программного продукта, начиная от попытки нахождения схем генерирования серийных номеров и заканчивая гораздо более серьезными атаками: попытками дешифрования конфиденциальных данных пользователя, хранящихся на всевозможных носителях или передающихся по различным сетям. Предположим, например, вы создали прекрасный клиент электронной почты, но в нем не хватает маленькой детали: письма на диске хранятся в открытом виде. Может быть, в этом ничего страшного и нет, но когда пользователю приходят в письмах важные данные, или он просто не желает, чтобы кто-то прочитал его почту, что остается делать программистам?

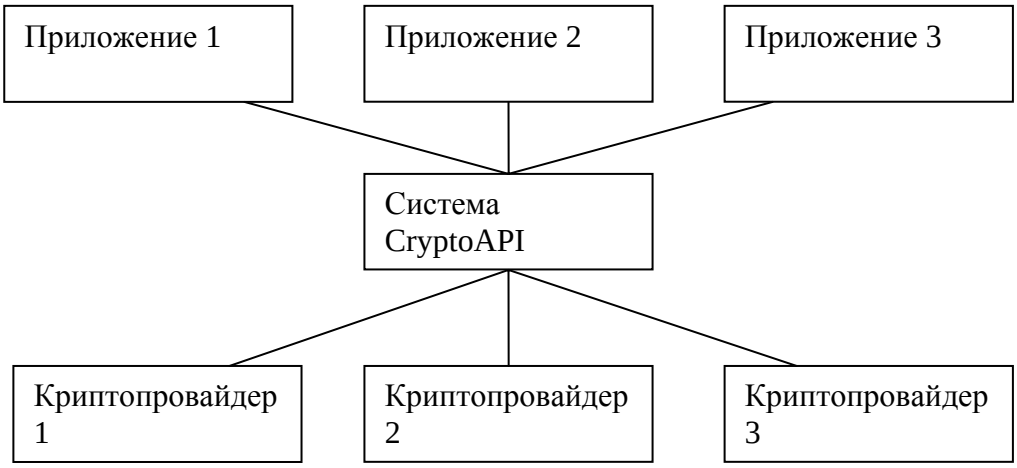
Часто они начинают сами придумывать схемы шифрования, хранения паролей и т.п. Хорошо, если программист знаком с наукой и действительно понимает то, что делает. Вообще говоря, разработка надежной системы безопасности программного обеспечения может занять много времени, которое могло бы пойти на улучшение функциональности, выявление и устранения других ошибок, которые, как ни крути, встречаются даже в дорогих продуктах от весьма серьезных производителей.

На сегодняшний день одной из самых популярных операционных систем является Microsoft Windows. И Microsoft предоставляет встроенные средства шифрования данных, начиная с Windows 95 OSR2. Этот блок API (Application Programming Interface) носит название Microsoft CryptoAPI.

2 Структура и основные положения CryptoAPI

Концепция CryptoAPI подразумевает сокрытие от программиста всех тонкостей и нюансов процесса шифрования данных. Работа этого API осуществляется через так называемые криптопровайдеры (*CSP* от англ. *Cryptographic Service Provider*). По своей сути они являются отдельными приложениями (DLL или вообще отдельные сервисы), которые написаны независимо от других приложений. Они подписаны цифровой подписью, так что система время от времени может проверять их подлинность.

Следует отметить гибкость такого решения. Например, при смене криптопровайдра на более современный (скажем, поддерживающий биометрический ввод данных) достаточно просто сменить название используемого криптопровайдера, поскольку весь основной интерфейс стандартизован.



Каждый разработчик может сам написать и внедрить свой криптопровайдер.

2.1 Типы функций CryptoAPI

Вся архитектура CryptoAPI может быть разделена три основыне части

- Базовые функции
- Функции шифрования и для работы с сертификатами
- Функции для работы с сообщениями

2.1.1 Базовые функции

В эту группу входят функции для выбора и подключения к криптопровайдеру, генерации и хранения ключей, обмена ключами. Сюда так же входят возможности управления параметрами ключа, такими как: режим сцепления блоков, инициализационный вектор, а также так называемый salt value. На данный момент Microsoft поддерживает только CBC и ECB режимы сцепления блоков, хотя константы для режимов CFB и OFB уже заведены и вероятно, что в скором веремени эти режимы появятся. С выходом Windows Vista появится новая версия криптосистемы от Microsoft.

К базовым функциям также можно отнести криптографическую функцию генерации случайных данных. Это наиболее важная функция во всей криптосистеме, потому что она пользуется для генерации ключей. Вся ответственность за ее реализацию ложится на разработчиков криптопровайдера. И если он работает в паре с каким-нибудь оборудованием, она может использовать его возможности. В случае, когда криптопровайдер реализован чисто программно, данными, которыми он используется для генерации случайной последовательности, могут быть системное время, позиция курсора мыши на экране компьютера, некоторое состояние устройства, например буфер ввода/вывода клавиатуры, или количество выполняемых задач в операционной системе.

Основные функции в этой группе:
CryptAcquireContext, CryptReleaseContext, CryptGenKey, CryptDestroyKey, CryptExportKey, CryptImportKey, CryptDeriveKey, CryptGenRandom.

2.1.2 Функции шифрования и для работы с сертификатами

В эту группу входят все функции для хеширования данных, шифрования и расшифрования, а также функции для использования сертификатов, основной задачей которых является предоставление доступа к открытому ключу. CryptoAPI поддерживает сертификаты спецификации X.509, в которые входит информация о версии сертификата, его серийном номере, периоде действия, алгоритмах шифрования публичного ключа и др. Вообще, в CryptoAPI сертификаты представляют из себя большую область, требующую

отдельного изучения. Но их функции можно использовать уже при простом подписывании данных.

Основные функции в этой группе:
CryptEncrypt, CryptDecrypt, CryptCreateHash, CryptDestroyHash, CryptHashData, CryptSignHash, CertOpenStore, CertCloseStore, CertFindCertificateInStore.

2.1.3 Функции для работы с сообщениями

Под сообщениями в CryptoAPI понимаются данные в стандартизованном формате PKCS #7, разработанном RSA Laboratories. Функции для работы с ними делятся на две части: высокоуровневые функции (упрощенные) и низкоуровневые. Для работы с последними необходимо ознакомиться со спецификацией на сайте http://www.rsasecurity.com.

2.2 Ключи шифрования в CryptoAPI

Ставшее традиционным разделение ключей шифрования на два типа здесь так же присутствует. CryptoAPI предоставляет методы для работы с сессионными (симметричными ключами) и с открытыми ключами. Все ключи управляются и используются при помощи неких идентификаторов, и приложение не получает открытого доступа к ним.

2.2.1 Сессионные ключи

Сессионные ключи меняются от сессии к сессии, и криптопровайдер не сохраняет их на диски и прочую энергонезависимую память. Но ведь есть ситуации, когда приложению требуется получить в каком-то виде ключ, чтобы, скажем, передать его по незащищенному каналу передачи данных или зашифровать файл и рашифровать его через некоторое время. Для этого в системе имеются функции обмена ключами:

- *CryptExportKey*, которая для случая экспорта сессионного ключа зашифровывает его, чаще всего с помощью открытого ключа принимащей стороны. Этот ключ носит название *key exchange public key* (открытый ключ для обмена ключами).
- *CryptImportKey*, которая служит для импорта на приемной стороне сессионного ключа.

Сессионные ключи могут генерироваться не только случайным образом, но и по каким-нибудь данным. В последнем случае гарантируется, что один и тот же CSP будет возвращать один и тот же ключ по одинаковым данным. Такая возможность используется для генерирования ключей по паролям.

2.2.2 Открытые ключи

Открытые ключи и их закрытые части для несимметричного шифрования в отличие от сессионных ключей хранятся криптопровайдером в контейнерах ключей в зашифрованном виде. Реализованы они могут быть самыми различными способами: файлы на дисках, ключи в реестре или другие, более изощренные устройства подключаемые к компьютеру. Тем не менее, этими ключами также можно обмениваться. Для этого используются те же самые функции, что и для сессионых ключей.

Уже сейчас можно проследить, что программист имеет возможность:

- Подключиться к некой криптографической службе.
- Запросить у нее хранилище пар ключей шифрования
- Передать открытый ключ приемной стороне

- Сгенерировать сессионный ключ
- Экспортировать сессионный ключ и передать приемной стороне
- Зашифровать данные и рашифровать на приемной стороне
- Подписать данные

Такого набора функций уже достаточно, чтобы наладить защищенный канал передачи данных или зашифровать данные пользователя, используя самые соверменные алгоритмы.

3 Стандартные криптопровайдеры и алгоритмы

CryptoAPI предоставляет следующие стандартные криптопровайдеры:

- Microsoft Base Cryptographic Provider
- Microsoft Strong Cryptographic Provider
- Microsoft Enhanced Cryptographic Provider
- Microsoft AES Cryptographic Provider
- Microsoft DSS Cryptographic Provider
- Microsoft Base DSS and Diffie-Hellman Cryptographic Provider
- Microsoft DSS and Diffie-Hellman/Schannel Cryptographic Provider
- Microsoft RSA/Schannel Cryptographic Provider

Все эти CSP отличаются друг от друга своими типама, которые определяются набором параметров, включающим:

- алгоритм обмена сессионным (симметричным) ключом.
- алгоритм вычисление цифровой подписи
- формат цифровой подписи
- схема генерирования сессионного ключа по хешу
- длина ключа

На текущий момент CryptoAPI имеет 8 стандартных типов криптопровайдеров. Всех их можно разделить на две группы по алгоритму, используемому для вычисления цифровой подписи: RSA и DSS – и на три группы по алгоритму обмена сессионным ключом: RSA, DH (Diffie-Hellman) и KEY (Key Exchange Algorithm).

Ниже представлена сравнительная таблица длин ключей в битах, используемых в Base Cryptographic Provider и AES Cryptographic Provider.

Алгоритм	Base Cryptographic Provider	AES Cryptographic Provider
RSA формирование подписи	512	1024
RSA алгоритм обмена ключами	512	1024
RC2 блочное шифрование	40	128
RC4 потоковое шифрование	40	128
DES	56	56
Triple DES (2 ключа)	Не поддерживается	112
Triple DES (3 ключа)	Не поддерживается	168

Между тем в CryptoAPI это лишь стандратные длины ключей. В этой системе имеется такое понятие, как алгоритм шифрования, и многие алгоритмы поддреживают сразу несколько длин ключей. Длины ключей RSA алгоритмов вычисления цифровой подписи и обмена ключами могут варьироваться от 384 до 16384 бит с интервалом в 8 бит.

Также поддерживаются алгоритмы шифрования AES (128, 192, 256) и вычисления хешей MD2, MD5, SHA, MAC, MAC (хеширование с ключом).

4 Хранение ключей

По умолчанию пары ключей в Windows хранятся в директории на диске в зашифрованных файлах. Шифрование происходит с помощью ключа полученного на основе пароля пользователя. Последнии уязвимости в этом шифровании, публично доступные в Интернете, отноятся к Windows 2000. До соответствующего обновления Windows в ней использовалось 40-битное симметричное шифрование, которое оказалось слабым. После него длина ключа стала равна 56 битам.

Так же в интернете можно встретить описание уязвимостей в CryptoAPI и формате хранения ключей за 1998 год. Более свежих найти не удается.

В Windows XP пары ключей лежат в директории типа C:\Documents and Setting\<UserName>\Application Data\Microsoft\Crypto. Была найдена интересная статья [5] о том, как можно избавить себя от хранения зашифрованных ключей у себя на диске, пользуясь монтированием директорий в файловой системе NTFS.

5 Применение

Если полистать страницы в интернете, где используется этот CryptoAPI, то можно обнаружить, что едва ли кто об этом пишет. Конечно, сама Microsoft пользуется этим API в своих продуктах Internet Explorer, Outlook Express и некоторых других. Тем не менее, в интернете (как в русском, так и иностранном сегментах) можно найти множество пособий о том, как пользоваться этой системой. Ведь если задуматься, то Microsoft не ограничивает использование этой системы только своими криптопровайдерами, у которых также есть свои недостатки (наиболее очевидный – это то, что пары ключей у простых пользователей хранятся на дисках в реестре системы, и поэтому на них возможно проводить атаки; статью об этом можно найти в ссылках на источники). Например, можно разрабатывать оборудование, защищенное от взлома, которое будет работать с новым криптопровайдером. Примером может служить продукт ЗАО АВЕСТ под названием AVEST CSP. Этот криптопровайдер позволяет использовать российский стандратный алгоритм шифрования ГОСТ 28147-89 и некоторые белорусские стандарты вычисление хешей и цифровых подписей. Этот криптопровайдер поддерживает оборудование для хранения пар ключей, такое как Rainbow iKey™1000, Rainbow iKey™1032 и др.

Сейчас Microsoft разрабатывает новый продукт CNG – Cryptography API: Next Generation. Оно будет доступно с выходом Windows Vista и станет заменой CryptoAPI, предоставляя еще более систематизированный подход к шифрованию данных.

6 Источники

1. http://msdn.microsoft.com/library/en-us/seccrypto/security/cryptography_portal.asp. Microsoft Developer Network (MSDN)
2. http://www.aves.by/crypto/csp.htm. КРИПТОПРОВАЙДЕР AVEST CSP
3. http://www.rsdn.ru/article/crypto/usingcryptoapi.xml. Использование CryptoAPI (2004 год, Юрий Николаев, RSDB Magazine #5-2004)
4. http://www.insecure.org/sploits/microsoft.private-key.protections.html. Microsoft Private Key Recovery (Peter Gutmann, 25 Jan 1998)
5. http://www.softwaresecretweapons.com/jspwiki/Wiki.jsp?page=EncryptedFileSystem Encrypted File System (последние изменения в 2006 году)