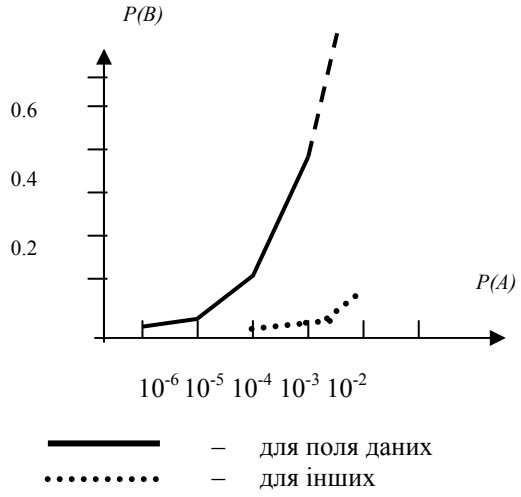

Рисунок 1– Залежність ймовірності порушення цілісності інформації в полях інформаційного кадру *P(B)* від ймовірності дії загрози *P(A)*

IV Висновок

Запропонована імітаційна модель дозволяє здійснювати прогнозування порушень безпеки інформації, яка циркулює в мережі передачі даних на канальному рівні відкритої системи обміну даними, за наявності апріорно заданих інформаційних загроз.

Література : 1. Гаценко О. Ю. Об оптимизации пакетов передачи данных в условиях информационного противоборства. // Электронное моделирование.—2000. Т. 22.– № 5.— С. 115 – 119. 2. Тарасенко В. П., Коваль С. М. Модельне прогнозування на канальному рівні характеристик відкритих систем обміну даними в умовах загроз. // Реєстрація, зберігання і обробка даних. – 2000, Т. 2.– № 4. – С. 82 – 90. 3. НД ТЗІ 1.1-003-99. Нормативний документ системи технічного захисту інформації. Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. 4. ДСТУ 3396.2-97. Захист інформації. Технічний захист інформації. Терміни і визначення. 5. Шварц М. Сети связи: протоколы, моделирование и анализ. – М.: Наука, 1992. – Ч. 1. – 336 с.

УДК 681.3.06

ИНТЕГРИРОВАННЫЕ МОДУЛИ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

Геннадий Гулак*, Виталий Вервейко, Юрий Горбенко, Сергей Полчанинов
**ДСТСЗИ СБУ, ХНУРЭ*

Анотація: **Розглядається задача створення інтегрованих модулів криптографічного перетворення та їх застосування.**
Summary: **The task of integrated crypto modules development and implementation is considered.**
Ключевые слова: **Система защиты информации, криптопровайдер, модуль криптографических преобразований.**

Введение

В настоящее время быстро расширяется спектр программного обеспечения, использующего средства криптографической защиты информации. Криптографические средства все более усложняются, что приводит к резкому усложнению программных продуктов, которые их используют. При появлении новых криптографических алгоритмов и стандартов приходится переделывать значительный объем программного обеспечения в соответствии с требованиями новых стандартов. В результате возникает задача упрощения применения средств криптографической защиты информации и упрощения перехода от использования одних

средств защиты к другим. Эта задача может быть успешно решена использованием интегрированных модулей криптографических преобразований. В зависимости от требований к уровню стойкости криптографической подсистемы интегрированные криптографические модули могут быть реализованы в программном или программно-аппаратном виде.

I Применение интегрированных криптографических модулей

Интегрированные модули криптографических преобразований – это программные библиотеки, реализующие различные функции криптографической защиты информации непосредственно или при помощи дополнительных аппаратных средств. К этим функциям обычно относят функции зашифрования и расшифрования, направленного шифрования, выработки и проверки цифровой подписи, выработки криптографических и однонаправленных хэш-функций, выработки общих секретов для протоколов разделения секрета и функции генерации случайных и псевдослучайных данных и ключей [1]. Дополнительными функциями интегрированных криптографических модулей являются хранение секретных ключей и разграничение доступа к ним, обеспечение целостности используемых дополнительных программных средств и др.

Для большего удобства применения модулей криптографических преобразований и ускорения их интеграции в разрабатываемые системы рекомендуется применять модули со стандартизированным интерфейсом. К стандартизированному интерфейсу предъявляется требование универсальности. Универсальность заключается в возможности реализации любой требуемой криптографической функции на базе данного интерфейса без его модификации. Дополнительно может выдвигаться требование распространенности интерфейса и простоты его использования. Широкая распространенность интерфейса заключается в его использовании в широком спектре программного обеспечения, а простота использования позволят упростить переход на этот интерфейс при модификации старых или разработке новых программных средств. Это позволяет не только облегчить разработку готовых программных продуктов, использующих криптографические средства защиты информации, но и устраняет необходимость в их изменении при смене используемой криптографической базы. При использовании модульного принципа построения можно выделить отдельную подсистему, предоставляющую разработчику необходимые криптографические функции. В этом случае, если возникает необходимость изменения или дополнения криптографической подсистемы, достаточно сменить один модуль криптографической защиты на другой или добавить новый модуль и автоматически изменяются используемые криптографические функции.

II Архитектура криптографической подсистемы Windows

Модульный принцип построения подсистемы защиты информации реализован в операционных системах семейства Windows. В качестве базовых интегрированных модулей криптографических преобразований используются криптопровайдеры (CSP, Cryptographic Service Providers). Для разработки был взят интерфейс криптопровайдеров CryptoSPI [2] – как наиболее универсальный и широко распространенный. Дополнительно фирма Microsoft в операционных системах семейства Windows реализовала специализированный интерфейс CryptoAPI, через который пользовательское приложение может получить доступ к интегрированным в операционную систему криптопровайдерам. Архитектура криптографической подсистемы Microsoft Windows представлена на рисунке 1.

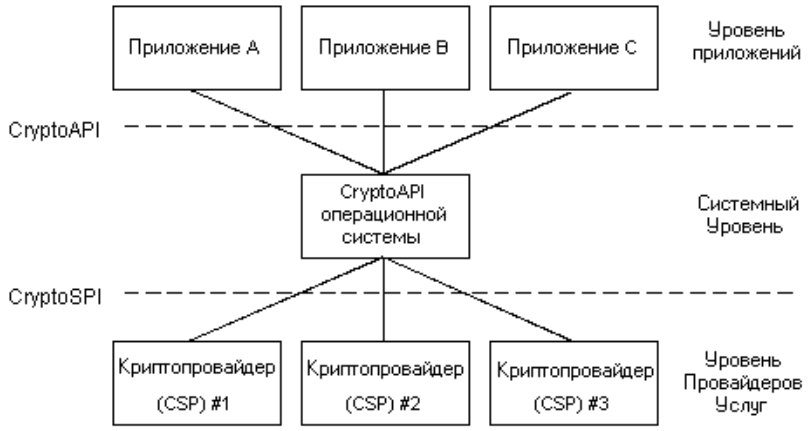

Рисунок 1 – Архитектура криптографической подсистемы Windows

Применение интегрированных криптографических модулей, стандартно поставляемых с операционной системой Windows, не может обеспечить гарантированную безопасность, так как в них используются алгоритмы, не сертифицированные к применению на территории Украины. Сами криптографические модули (криптопровайдеры) не могут быть сертифицированы как программное обеспечение, так как поставляются без исходных кодов и не могут быть просто проанализированы на наличие недокументированных функций (программных закладок). Для проведения анализа необходимо осуществить дизассемблирование большого объема программного обеспечения, а затем провести тематические исследования на правильность реализации соответствующих криптографических функций. Кроме того, стандартное криптографическое программное обеспечение не использует аппаратных средств для выработки и хранения ключей и параметров, что значительно снижает их уровень стойкости. Поэтому важной является задача интегрирования в операционную систему Windows своих криптографических модулей с разрешенными алгоритмами или действующими ГОСТами, возможностью выработки ключей, в том числе с помощью аппаратных генераторов случайных чисел, возможностью их хранения на различных носителях в защищенном виде, а также возможностью реализации криптографической подсистемы в аппаратном виде.

III Особенности архитектуры криптопровайдеров

Для повышения уровня защищенности интегрированного криптографического модуля используется специальная архитектура, которая позволяет создать такие элементы, как изолированная область памяти и контейнеры секретных ключей. Эти элементы позволяют не давать приложению пользователя никаких секретных данных и значительно усложнить попытки получения таких данных сторонними приложениями. При работе с функциями CryptoSPI приложения и операционная система используют только дескрипторы объектов (handle). Использование дескрипторов, а не указателей, позволяет реализовать один из главных элементов архитектуры криптопровайдеров – изолированную область памяти. В изолированной области памяти криптопровайдер может хранить различные секретные параметры (например секретные ключи). При этом получить секретные параметры в открытом виде невозможно. Доступ к этой памяти осуществляется только при помощи специальных дескрипторов, что позволяет реализовать модель, в которой из приложения невозможно определить, где хранятся ключи и алгоритмы криптопреобразований. При программной реализации криптопровайдера возможно создание области памяти, изолированной только от приложений, а при программно-аппаратной – как от приложений, так и от операционной системы. Обычно при программной реализации в качестве изолированной памяти используется память, которая не включается в страничный файл подкачки. Возможна программно-аппаратная реализация, когда криптографические алгоритмы и секретные ключи хранятся в специализированном устройстве, криптографические преобразования выполняются в этом же устройстве, а программная часть выполняет только трансляцию вызовов интерфейса CryptoSPI в вызовы данного устройства. Это позволяет обойти программные закладки, которые могут присутствовать в операционной системе Windows.

Для хранения секретных ключей криптопровайдеры используют логические объекты, называемые контейнерами ключей. В каждом вызове функции CryptoSPI содержится дескриптор секретного контейнера ключей, который может не использоваться при отсутствии необходимости в секретном ключе (например при проверке цифровой подписи). Обычно в контейнерах хранятся ключевые пары (секретный и открытый ключи) или только секретные ключи, используемые для выработки цифровой подписи и обмена секретным сеансовым ключом. Физически контейнеры ключей могут находиться в произвольном месте: в реестре или в виде отдельных файлов на жестком диске, на дискете, на смарт-карте и т. д. Контейнеры хранятся в защищенном виде и для доступа к ним требуются секретные данные (например пароль или PIN-код). При этом приложение пользователя может не знать, где находится контейнер и какие данные криптопровайдер потребует от пользователя для доступа к нему. После получения доступа к секретному контейнеру он отображается в изолированную область памяти и для доступа к нему пользовательской программе выдается дескриптор. Дополнительно контейнер ключей может содержать данные о пользователе, время действия секретных ключей и другие требуемые параметры. Содержимое контейнеров ключей является постоянным от сеанса к сеансу, и изменяется только при смене ключей.

Кроме контейнеров ключей используются также временные объекты, которые находятся в изолированной памяти криптопровайдера, не сохраняются на диске и существуют только во время сеанса работы. В таких объектах хранятся сеансовые ключи, общие секреты, хэш-функции и т. д.

IV Создание нового криптопровайдера

При создании нового криптопровайдера от разработчика требуется только выполнение библиотеки в виде модуля со стандартным интерфейсом и произвольной внутренней структурой. Это позволяет создавать

криптопровайдеры, работающие с разнообразным аппаратным обеспечением (аппаратные шифраторы, генераторы случайных чисел, разнообразные носители ключей). Если использовать специальные технические средства (например, сетевые карты с аппаратным шифратором и памятью для секретных ключей), то возможно создание высокозащищенной информационной системы на базе операционной системы Windows. При отсутствии требований высокой защищенности возможна реализация криптопровайдера только программными средствами. В этом случае возможно создание только частично контролируемой информационной системы. Разработка криптопровайдера начинается с выбора криптографических алгоритмов [3] и форматов данных, которые будут использоваться в нем. Интерфейс модуля криптографических преобразований является универсальным и позволяет использовать практически все известные криптографические алгоритмы и базовые протоколы. После этого необходимо реализовать выбранные алгоритмы и форматы данных в соответствии с интерфейсом CryptoSPI. Если криптопровайдер будет использовать специализированное оборудование, нужно написать соответствующие драйвера. После завершения разработки криптопровайдера необходимо его интегрировать в операционную систему. Рекомендованная Microsoft процедура инсталляции криптопровайдера состоит из следующих пунктов:

- копирование файлов криптопровайдера в системную папку;
- создание записей в реестре;
- создание и запись в реестр цифровой подписи криптопровайдера.

Последний пункт является самым сложным, т. к. для получения цифровой подписи разрабатываемого криптопровайдера необходимо обращаться в Государственный Департамент США, что невозможно по нескольким причинам. Эту проблему можно решить при помощи генерации своих ключей цифровой подписи, встраивания открытого ключа в операционную систему и выработки на них цифровой подписи длиной 1024 бита по алгоритму RSA таким образом, чтобы программы проверки целостности и подлинности «признали» эту программу, как свою. В этом случае возможны некоторые проблемы с проверкой стандартного программного обеспечения Windows, а полное отключение функций проверки целостности значительно понижает надежность операционной системы. Кроме того, возможно использование криптопровайдера без встраивания в операционную систему. Основным недостатком такого способа использования криптографического модуля является отсутствие высокоуровневых протоколов (таких как SSL, TLS) и функций управления сертификатами, реализованных в операционной системе. Эти элементы необходимо создавать в программе, которая будет использовать криптографический модуль. Для устранения этого недостатка может быть создана дополнительная библиотека, которая обеспечивает управление сертификатами и реализует расширенные криптографические протоколы.

Заключение

В большинстве стандартного программного обеспечения под операционную систему Windows, использующего криптографические функции, используется интерфейс CryptoAPI. Зарегистрировав свой криптопровайдер вместо стандартного, можно полностью заменить используемую в этих программах криптографическую базу (алгоритмы криптографических преобразований, схемы управления ключами и др.). Это позволяет применять собственные криптографические модули в широком спектре уже существующего программного обеспечения без его модификации.

В настоящее время разработан ряд криптопровайдеров, на основе которых реализован ряд защищенных технологий с необходимым уровнем стойкости криптографической подсистемы. Некоторые из криптопровайдеров используют специализированные аппаратные средства (генераторы случайных чисел).

Литература: 1. А. Менезис, П. ван Оршот, С. Ватсон. Прикладная криптография // CRC Press, 1996. Глава 5. 2. Microsoft Developer Network Library – January 2001. Platform SDK (Security). Microsoft Corporation. 3. Schnier B. Applied Cryptography. Second Edition: protocols, algorithms and source code in C. Published by John Wiley & Sons. Inc, New York: Chichester Brisbane Toronto Singapore, 1996.