КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

N.I. Alishov, A.N. Alishov, M.G. Lutskiy, A.N. Mischenko, N.A. Sapunova

THE ARCHITECTURE OF HARDWARE OF IMPLEMENTATION OF METHODS OF INDIRECT STEGANOGRAPHY

Hardware for enciphering of the stream information by indirect steganography methods and illustrations of algorithms of operation of a cryptosystem are considered.
Key words: information protection, cryptography, steganography, associative memory, USB-device.

Представлені апаратні засоби для шифрування потокової інформації методами непрямої стеганографії та ілюстрації алгоритмів роботи криптосистеми.
Ключові слова: захист інформації, криптографія, стеганографія, асоціативна пам'ять, USB-пристрій.

Представлены аппаратные средства для шифрования потоковой информации методами косвенной стеганографии и иллюстрации алгоритмов работы криптосистемы.
Ключевые слова: защита информации, криптография, стеганография, ассоциативная память, USB-устройство.

УДК 004.056

Н.И. АЛИШОВ, А.Н. АЛИШОВ, М.Г. ЛУЦКИЙ, А.Н. МИЩЕНКО, Н.А. САПУНОВА

АРХИТЕКТУРА АППАРАТНЫХ СРЕДСТВ РЕАЛИЗАЦИИ МЕТОДОВ КОСВЕННОЙ СТЕГАНОГРАФИИ

Введение. При реализации процесса криптозащиты потока полезной информации исключительно программным методом, где обеспечивающее криптозащиту программное обеспечение (ПО) базируется на классической ЭВМ, построенной по фоннеймановской архитектуре, слабым звеном является сама архитектура системы, так как ПО функционирует только в рамках предопределенных архитектурных решений. Это может стать существенной проблемой, когда ЭВМ не сможет за отведенное время обработать определённый объём потока информации. Ещё одним недостатком программного метода является то, что никакой детерминированный алгоритм не может генерировать истинно случайные числа; он может только аппроксимировать некоторые свойства случайных чисел.

Для решения проблем, возникших при реализации программного подхода, предлагается перенести вычислительную нагрузку на внешнее устройство – аппаратную подсистему [1, 2], что позволит:

- снять вычислительную нагрузку с базовой системы− все действия по вычисл ению и обработке потоков информации возлагаются на внешнее устройство;
- обеспечить максимально возможную скорость обработки (шифрования/дешифрования) потоков полезной информации; предел скорости обработки ограничивается пропускной способностью канала связи;
- реализовать генерацию истинно случайных чисел в реальном масштабе времени;

- обеспечить скрытность – никакими программными средствами невозможно проследить процесс криптопреобразования; все действия выполняются исключительно аппаратной подсистемой в реальном масштабе времени.

Комплекс, объединяющий аппаратные и программные средства защиты информации, называется криптосистемой, в состав которой входят две взаимодействующие части: аппаратная подсистема – внешнее устройство; программная подсистема – программы, инсталлируемые на ЭВМ.

Множество ЭВМ с установленными криптосистемами, объединенных в сеть, будем называть крипторешением.

Постановка задачи. Рассматриваемое в данной статье крипторешение функционирует в соответствии с клиент-серверной сетевой архитектурой [1, 3]. Серверу должен быть задан статической IP-адрес [3] (или постоянное DNS-имя [4]). Во время установки программной подсистемы на ЭВМ клиентов задаётся сетевой IP-адрес или DNS-имя сервера.

В качестве алгоритма для обеспечения криптозащиты был выбран метод косвенной стеганографии [5], так как он основан на идеях шифра Вернама, или «схемы одноразовых блокнотов» (one-time pad) − единственной системы шифрования, для которой доказана абсолютная криптографическая стойкость [6]. Других шифров с этим свойством не существует, так как «схема одноразовых блокнотов» – самая безопасная криптосистема из всех известных. Аналогом «одноразовых блокнотов» является так называемая «книжная» стеганография, которая предусматривает использование книги в качестве секретного симметричного ключа. Для реализации алгоритмов косвенной стеганографии разработана архитектура и структурная организация аппаратно-программного комплекса.

Программная подсистема на хост-ЭВМ не выполняет сложных вычислительных преобразований, при этом нагрузка на ЭВМ, на которой установлена криптосистема, весьма незначительна. Программная подсистема лишь передает управляющие команды аппаратной подсистеме, а также перенаправляет потоки информации между подсистемами.

Решение задачи. Программные подсистемы на сервере и на системах клиентов отличаются.

Серверная программная подсистема выполняет следующие функции:

1. Работа с контейнером-ключом предполагает такие действия:

- установка контейнера-ключа поочерёдно на каждом аппаратном устройстве для клиентов;
- замена контейнера-ключа в работающем крипторешении, выполняемая в таких режимах:
  - автоматический (максимальная криптозащита) – контейнер-ключ используется только дважды – для передачи открытой информации равной размеру контейнера-ключа и нового контейнера-ключа, путем смешивания массива данных исходного ключа;
  - текущий контейнер-ключ постоянно используется, пока не будет дана команда «создать новый». При этом либо меняются случайным образом началь-

ные точки шифрования в массиве, либо осуществляется смешивание данных массива контейнера-ключа с использованием дополнительного массива истинно случайных чисел или псевдослучайных последовательностей. При этом в соответствии с расписанием указывается график создания нового контейнера-ключа;

- режим *multicast* – передача контейнера-ключа с сервера доверенным клиентам для снижения нагрузки на канал связи.

2. Обработка потоков полезной информации (рис. 1).

3. Задание режима работы с приложениями для использования криптосистемы:

- создание списка приложений;
- шифрование всего сетевого трафика.

Клиентская программная подсистема реализует:

1) получение от сервера нового контейнера-ключа и передачу его аппаратной подсистеме (рис. 2; Алгоритм 4);
2) обработку потоков полезной информации;
3) получение от сервера списка приложений, для обеспечения, безопасности которых используется криптосистема.

Аппаратная подсистема предлагаемого решения включает:

1. Цифровой сигнальный процессор (ЦСП) [7], реализующий следующие алгоритмы работы аппаратной подсистемы:

- обработка двух потоков информации одновременно:
  - из приложения в сеть (шифрование),
  - из сети в приложение (дешифрирование);
- работа с контейнером-ключом:
  - генерировать новый основной контейнер-ключ и сохранять его во встроенной flash-памяти и в программной подсистеме (рис. 2; Алгоритм 1);
  - заполнить контейнер-ключ незашифрованным потоком из программной подсистемы (рис. 2; Алгоритм 2);
  - генерировать новый контейнер-ключ и сохранять его во встроенной flash-памяти и передавать программной подсистеме новый контейнер-ключ, зашифрованный активным (текущим или старым) контейнером-ключом (рис. 2; Алгоритм 3);
  - заполнить новый контейнер-ключ зашифрованным потоком, получаемым из программной подсистемы (рис. 2; Алгоритм 4).

2. Твердотельное перезаписываемое постоянное запоминающее устройство (flash-память) [8] – для хранения секретного контейнера-ключа.

На данный момент планируется использовать комплект физических микросхем памяти, общим объемом 128 Gb, логически разделив их на четыре равных сектора. В первых двух секторах хранятся:

- основной контейнер-ключ (ОКК) – классический массив (индексы присвоены последовательно), хранящийся в памяти внешней аппаратной подсистемы. Заполняется потоком истинно случайных чисел, поступающих с АЦП (array[*i*]={“*j*”}). Каждая ячейка массива содержит один байт;

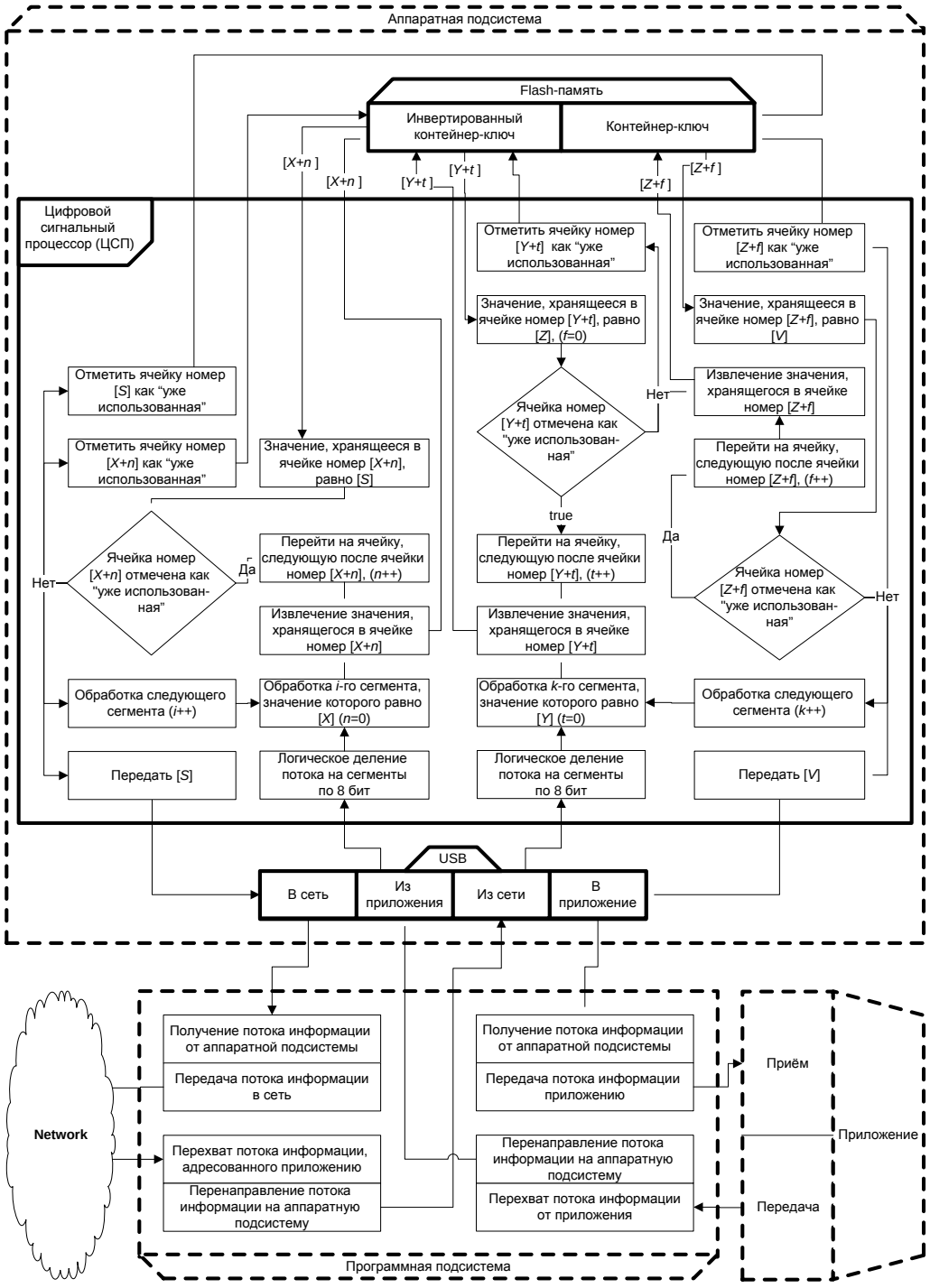

РИС. 1. Обработка потоков полезной информации криптосистемой

▪ инвертированный контейнер-ключ (ИКК) – ассоциативный массив, хранящийся в памяти внешней аппаратной подсистемы. Заполняется совместно с ОКК. Индексы (номера ячеек) ОКК становятся значениями, хранящимися в ячейках ИКК, а значения, хранящиеся в ячейках ОКК, становятся индексами ИКК – номерами ячеек (array[j]={“i”}).

Оставшиеся два сектора зарезервированы для создания соответственно:

▪ альтернативного контейнера-ключа (АКК) – будет задействован после окончания использования ОКК;

▪ инвертированного альтернативного контейнера-ключа – будет задействован после окончания использования ИКК.

Контейнерам-ключам присваиваются индексы (номера), чтобы различать:

▪ текущий контейнер-ключ, – используемый в данный момент для шифрования/дешифрования потоков информации;

▪ альтернативный контейнер-ключ – хранящийся в памяти и ожидающий использования.

Контейнер-ключ с меньшим индексом является текущим, а с большим – альтернативным.

3. Аналоговый генератор белого шума [9] – для получения истинно случайных чисел.

4. Аналого-цифровой преобразователь (АЦП) [9] – для оцифровки аналогового сигнала, поступающего от генератора белого шума.

Описание алгоритма работы. У отправителя и получателя имеются одинаковые секретные контейнеры-ключи. Поток информации, подлежащий защите, условно делится на сегменты (побайтно), после чего производится замена (по определенному алгоритму) сегментов полезной информации соответствующими сегментами секретного контейнера-ключа. В результате получается образ исходной информации такого же размера. Сегменты образа полезной информации в реальном времени передаются по каналу связи. При получении адресатом образ подвергается обратному преобразованию: его сегменты заменяются соответствующими сегментами секретного контейнера-ключа, выполняется зеркальный Алгоритм (см. рис. 1).

Контейнер-ключ представляет собой классический массив случайных чисел с неограниченным количеством строк. Все ячейки массива пронумерованы по порядку. Таким образом, для получения из контейнера-ключа значения ячейки достаточно знать её адрес. Однако определение адреса ячейки по ее значению связано с поиском по всему массиву, что ощутимо сказывается на общей скорости обработки информации.

Для решения этой проблемы создаётся инвертированный, по отношению к оригинальному, ассоциативный массив, заполняемый одновременно с оригинальным контейнером-ключом. В инвертированном контейнере-ключе значения ячеек из оригинального контейнера-ключа становятся номерами ячеек, а номера ячеек из оригинального контейнера-ключа, соответственно, − значениями, хранящимися в ячейках ассоциативного массива.

Теперь, передав инвертированному контейнеру-ключу в качестве адреса ячейки значение, получаемое из сети, можно сразу определить номер ячейки в оригинальном контейнере-ключе (см. рис. 1).

Для реализации алгоритма работы выбирается определённое количество идентичных физических устройств (аппаратных подсистем).

Настройка крипторешения выполняется поэтапно.

I. Инициализация физических устройств.

1. На сервере устанавливается серверная программная подсистема.

2. К серверу подключается первое физическое устройство с возможностью генерации случайных чисел.

3. В программной подсистеме сервера запускается «Алгоритм 1» (рис. 2; Алгоритм 1), в результате чего генерируется поток истинно случайных чисел, который сохраняется:

- во встроенной flash-памяти аппаратной подсистемы как основной контейнер-ключ или инвертированный контейнер-ключ;
- в программной подсистеме.

4. К серверу подключается следующее (любое из партии) физическое устройство – аппаратная подсистема для клиентов.

5. На сервере запускается «Алгоритм 2» (рис. 2; Алгоритм 2), в результате чего программная подсистема передаёт аппаратной подсистеме незашифрованный поток (контейнер-ключ), а аппаратная подсистема сохраняет его во встроенной flash-памяти как основной контейнер-ключ и инвертированный контейнер-ключ. «Алгоритм 2» повторяется для остальных физических устройств (аппаратных подсистем).

II. Инсталляция крипторешения.

1. На сервере выбирается режим для обеспечения безопасности приложений, для которых будет использоваться крипторешение:

- создаётся список приложений;
- задается режим шифрования всего сетевого трафика.

2. Устанавливаются криптосистемы (программные и аппаратные подсистемы) на клиентских ЭВМ.

3. Клиентские ЭВМ регистрируются на сервере и получают список приложений, сетевой трафик с которых подлежит защите.

III. Эксплуатация крипторешения.

1. Обмен зашифрованными потоками информации.

2. Замена контейнера-ключа:

- сервер выполняет «Алгоритм 3» и рассылает клиентским ЭВМ зашифрованный новый контейнер-ключ;
- клиенты выполняют «Алгоритм 4» и инсталлируют (сохраняют) новый контейнер-ключ.

Сетевая реализация. Сетевая реализация предлагаемого крипторешения базируется на клиент-серверной архитектуре и может использоваться для защиты сетевого трафика между двумя ЭВМ либо всего сетевого периметра.

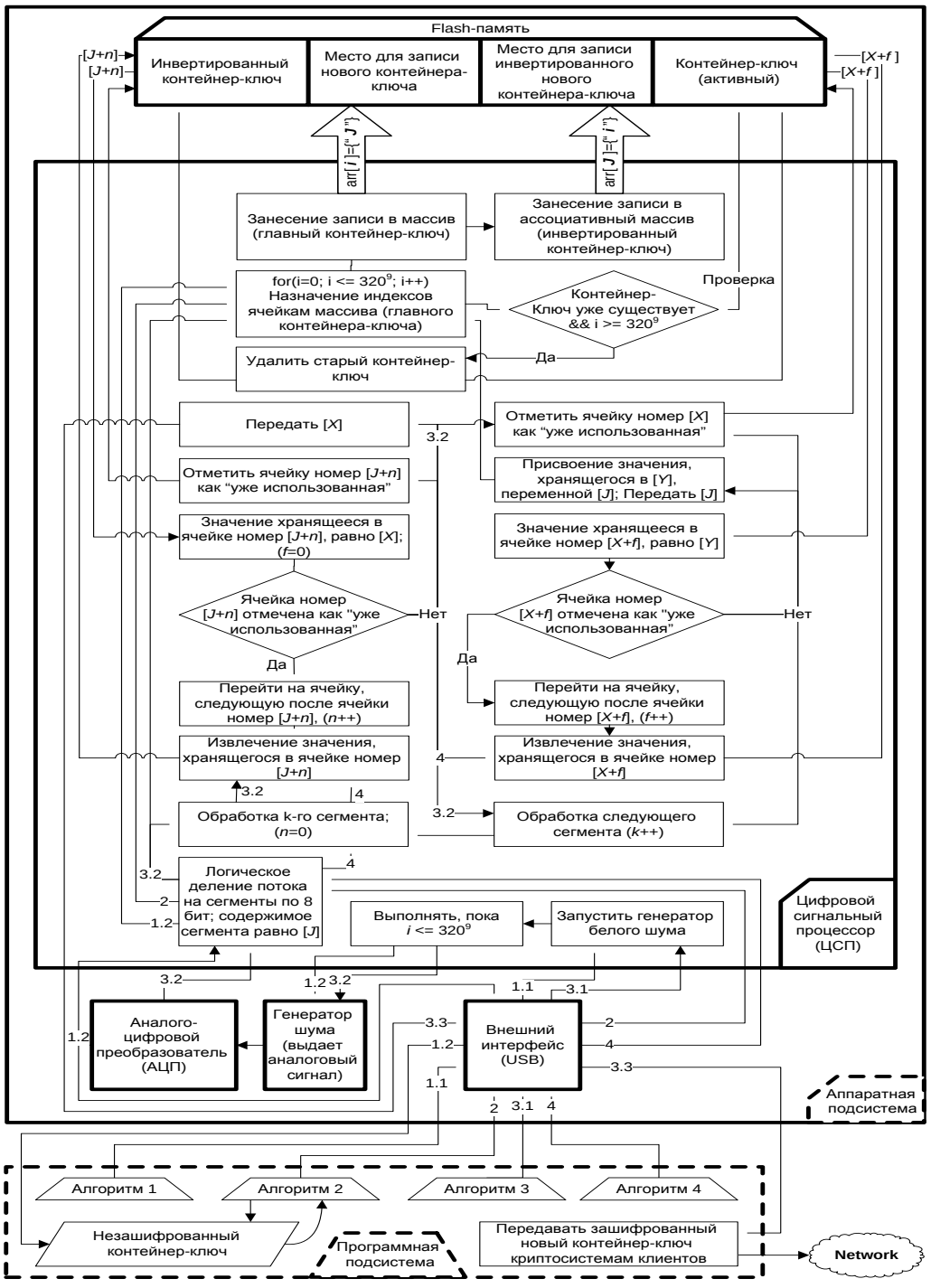

РИС. 2. Алгоритмы работы с контейнером-ключом

В случае установки криптосистем на нескольких ЭВМ взаимодействие между ними выполняется в стандартном режиме, за исключением того, что передаваемые потоки информации являются зашифрованными. При организации передачи данных между клиентами и сервером выполняются соответствующие протоколы шифрования/дешифрования потоковой информации.

Заключение. Таким образом, использование предлагаемой криптосистемы позволяет реализовать защиту (шифрование/дешифрирование) передаваемого потока информации как для конкретного (заданного) приложения, так и для всего сетевого трафика в целом.

1. *Жельников В.* Криптография от папируса до компьютера. – М.: АВР, 1996. – 56 с.
2. *Сборка* и перевод зарубежных исследований. Поточные шифры. Результаты зарубежной открытой криптологии. – М., 1997. – 390 с.
3. *Кудрявцева С.П., Колос В.В.* Навчальний посібник – К.: Вид. дім «Слово», 2005. – 400 с.
4. *Ли Альбитц.* DNS и BIND. – М.: Символ, 2008. – 712 с.
5. *Алишов Н.* Косвенная стеганография // Intern. Book Series “INFORMATION SCIENCE & COMPUTING” (Sofia: ITHEA). − 2009. − N 11. − P. 53−58.
6. *Шеннон К.* Работы по теории информации и кибернетике. – М.: Изд-во иностр. лит-ры, 1963. – 830 с.
7. *Interfacing* Micron MT9V022 Image Sensors to Blackfin Processors (2006). – http://www.analog.com/uploadedfiles/application_notes/213595899ee_258r262006.pdf
8. *Incorporated* Flash Memory. – http://ieeexplore.ieee.org/iel5/7440/20223/00934447.pdf
9. *Design* and Implementation of a True Random Number Generator Based on / Michael Epstein, Laszlo Hars, Raymond Krasinski, etc. – http://www.hars.us/Papers/CHES2003-epstein-hars-krasinski-rosner-zheng.pdf

Получено 16.04.2010