В.П.Иващенко, Г.Г.Швачич, П.А.Щербина.

Некоторые аспекты защиты данных в многопроцессорных вычислительных системах.

Новые компьютерные технологи. Монография. –
Кривой рог:ДВНЗ «Криворожский национальный університет»,

2013

Некоторые аспекты организации информационной безопасности функционирования многопроцессорных вычислительных систем

На современном этапе все большую роль в дальнейшем развитии информационных ресурсов играют параллельные вычислительные системы и вычисления. Подобные системы находят применение в сфере экономических, технологических и других процессов. В связи с их развитием, внедрением и совершенствованием широкое распространение получили методы задания вреда таким ресурсам. Наибольший интерес вызывают проблемы исследования методов и средств защиты информации в параллельных вычислительных процессах. В настоящее время подобные исследования не приобрели надлежащего развития. Изучение и разработка подобной проблематики предоставит возможности для дальнейшего развития новых и уже существующих методов защиты информации. Использование существующих алгоритмов их доработка изменения оптимизация, а также дальнейшая программная и аппаратная реализация.

Таким образом, одной из основных проблем использования многопроцессорной и параллельной вычислительной системы является реализация методов защиты информации.

Проблема реализации методов защиты информации имеет два аспекта:

− разработку средств, реализующих криптографические алгоритмы,

− методику использования этих средств.

Предложенные далее криптографические методы могут быть реализованы либо программным, либо аппаратным способом.

Возможность программной реализации обуславливается тем, что все методы криптографического преобразования формальны и могут быть представлены в виде конечной алгоритмической процедуры.

При аппаратной реализации все процедуры шифрования и дешифрования выполняются специальными электронными схемами. Наибольшее распространение получили модули, реализующие комбинированные методы.

Анализируя некоторые аспекты информационной безопасности модульных вычислительных систем можно отметить следующее:

1.Вычислительную сеть многопроцессорной системы можно считать безопасной с точки зрения обработки информации, если в

ней предусмотрена централизованная система управляемых и взаимоувязанных препятствий, которая перекрывает с гарантированной надежностью (в соответствии с моделью потенциального нарушителя) количество возможных каналов несанкционированного доступа и угроз, направленных на потерю или модификацию информации, а также несанкционированное ознакомление с ней посторонних лиц [7].

2. При проектировании мероприятий защиты данных в параллельных вычислительных системах необходимо провести анализ мест хранения информации, интерфейса управления данными и каналами их передачи (локальными сетями или другими подобными средствами обмена данными). Причем основной аспект защиты данных должен быть, в первую очередь, направлен на защиту интерфейса управления данными, во вторую очередь, на защиту интерфейса обмена данными, и дальше на места хранения информации. Это объясняется наиболее уязвимыми местами для несанкционированного доступа в параллельных моделирующих средах, их своевременная защита позволит обеспечить надежность функционирования системы.

Цель данной главы монографии заключается в определении и использовании мероприятий по защите информации, которые могут быть эффективны при использовании многопроцессорных модульных вычислительных системах или при параллельных расчетах на многопоточных системах.

В данном параграфе монографии проведено сравнение методов защиты данных с последовательными системами, а также влияние применения различных криптографических методов на методы реализации защиты. В исследовании, в соответствии с некоторыми аспектами построения многопроцессорных систем, было рассмотрено и выявлено несколько ключевых элементов, которые требуют особенного внимания при разработке системы безопасности. Показано, что их основной выбор определяется отличиями от последовательных систем в теоретической и аппаратной реализации.

На современном этапе комбинированные средства шифрования включающие программно - аппаратные средства являются наиболее актуальными как в плане использования так и в плане разработок. Выбор метода реализации криптозащиты для параллельной вычислительной системы зависит от ее направленности и конструктивных особенностей, использование блуждающих ключей предоставляет возможность повысить универсальность системы, при невысоких потерях производительности.

6.6.1. Анализ информационной безопасности функционирования многопроцессорных вычислительных систем

Рассмотрим существующие виды информационных угроз.

Виды информационных угроз можно разделить на две больших группы:

1) отказы и нарушения работоспособности программных и технических средств;

2) преднамеренные угрозы, которые загодя планируются злоумышленниками для задания вреда.

Выделяют следующие основные группы причин сбоев и отказов в работе компьютерных систем :

нарушение физической и логической целостности структур данных, которые хранятся в оперативной и внешней памяти, что возникают вследствие старения или преждевременного износа их носителей;

нарушения, которые возникают в работе аппаратных средств из-за их старения или преждевременного износа;

нарушение физической и логической целостности структур данных, которые хранятся в оперативной и внешней памяти, что возникают вследствие некорректного использования компьютерных ресурсов;

нарушения, которые возникают в работе аппаратных средств из-за неправильного использования или повреждения, в том числе из-за неправильного использования программных средств;

не устраненные ошибки в программных средствах, не выявленные в процессе отладки и испытаний, а также что остались в аппаратных средствах после их разработки.[3]

Рассмотрим теории последовательного и параллельного программирования.

Параллельная программа - огромное количество параллельных процессов, которые взаимодействуют (что синхронизируют свою работу и обмениваются данными) с помощью передачи сообщений.

Идея распараллеливания вычислений основана на том, что большинство заданий могут быть разделена на набор меньших заданий, которые могут быть решены одновременно. Обычно параллельные вычисления требуют координации действий.

Параллельное программирование включает все черты больше традиционного, последовательного программирования, но в

параллельном программировании есть три дополнительных, четко определенных этапы.

Определение параллелизма : анализ задания с целью выделить подзадачи, которые могут выполняться одновременно

Выявление параллелизма : изменение структуры задания так, чтобы можно было эффективно выполнять подзадачи. Для этого часто требуется найти зависимости между подзадачами и организовать начальный код так, чтобы ими можно было эффективно управлять

Выражение параллелизма: реализация параллельного алгоритма в начальном коде с помощью системы обозначений параллельного программирования

Из предыдущих изложений можно сделать вывод, что основное отличие параллельной системы - это увеличение количества подзадач, которые пересылаются, которые могут выполняться одновременно и наличие отдельной системы управления этими подзадачами.

Рассмотрим зависимости параллельных технологий от аппаратных средств.

Часто выделяют три технологии обеспечения параллельной работы : симметричные многопроцессорные системы (SMP - symmetrical multiprocessing), кластерные конфигурации и распределены вычислительные системы (Grid). SMP требует поддержки как со стороны аппаратуры, так и со стороны операционной системы, а кластеры и Grid - среды больше зависят от организации сетевого взаимодействия [1].

С точки зрения ядра операционной системы поддержка кластеров и распределенных систем заключается в эффективной работе с сетью.

С некоторым упрощением любую современную высокопроизводительную вычислительную систему можно представить как огромное количество многопроцессорных вычислительных узлов, связанных одной или несколькими коммуникационными сетями.

6.6.2. Организация информационной безопасности ресурсов МВС

Вычислительную сеть можно считать безопасной в смысле обработки информации, если в ней предвидена централизованная система управляемых и взаимосвязанных препятствий, которые перекрывают с гарантированной прочностью заданное в соответствии с моделью потенциального нарушителя количество возможных каналов несанкционированного доступа и угроз, направленных на потерю или модификацию информации, а также несанкционированное ознакомление с ней посторонних лиц [4].

При проектировании защиты параллельных систем, необходимо провести анализ мест хранения информации, интерфейса управления данными, и каналами передачи (локальными сетями или другими подобными средствами обмена). Причем основной аспект защиты должен быть в первую очередь направлен на защиту интерфейса управления данными во вторую очередь, на защиту интерфейса обмена данными, и в дальнейшем на места хранения информации, поскольку они определяются как наиболее уязвимые места для несанкционированного доступа в параллельные системы их своевременная защита позволит обеспечить, надежность и отказостойкость системы.

Упрощенную систему защиты информационных ресурсов МВС в общем виде представлена на рис. 6.1.

При определении защиты интерфейса управления возможно использовать рекомендации корпорации Intel которая предлагает следующие уровни привилегий :

0 - ядро операционной системы;

1 - операционная система;

2 - системы программирования и базы данных;

3 - прикладные (предназначены для пользователя) программы.

4 - На уровне коммутации оптимально использовать в качестве криптографические так и аппаратные методы защиты;

5 – на уровне хранения желательно использовать криптографические и аппаратные методы в системе распределенного хранения информации.

Подобные системы предусматривают внешний доступ и управление соответственно должны использоваться и методы защиты удаленных подключений.

Поскольку система содержит большое количество средств защиты нужный мониторинг не только системы управления и коммутации, а также содержимого, записываемого на носителе.

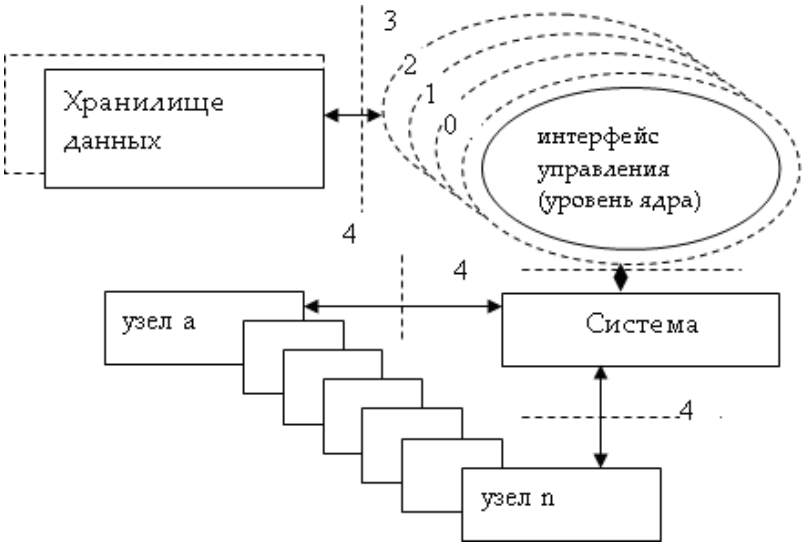

Рис. 6.1. Упрощенная система защиты информационных ресурсов МВС

Параметры системы коммутации при введении мероприятий защиты в параллельную систему будут меняться. Наиболее полная транспортная среда описана Н.И.Алишовим[4].

Основными параметрами транспортной среды с коммутацией пакетов являются количество коммутаторов (г) и длина заглавия (h), от которых непосредственно зависит время доставки массиву информации. Именно величины г и h предопределяют возможность адаптации времени доставки массиву данных D к конкретным умам функционирования дополнений. Поэтому процесс выбора необходимого времени доставки массиву данных объемом D байтов выходя из заданных значений величин r и h. а также из ограничений, существующих как для этих величин, так и для нескольких технических характеристик транспортной среды с коммутацией пакетов, назовем rh - оптимизацией.

$$T_{A\to B}^{D} = D(r+1) + \sum_{i=1}^{r+1} h_i$$

байтовым тактам

Также величина D будет изменяться и для систем многоканальной доставки данных с одинаковым и разностным количеством транзита но на постоянную величину которая будет непосредственно зависеть от времени обработки информации (например методы шифровки, архивации) как в сторону увеличения так и в сторону уменьшения D при увеличении времени обработки ядра и подчиненных узлов в случае использования распределенных хранилищ и соответственно дополнительных мероприятий, безопасности время обработки ядром еще больше увеличится. Соответственно количество байт- тактов изменится.

6.6.3. Организации безопасности информационных ресурсов в корпоративных сетях

Рассмотрим подсистему организации безопасности информационных ресурсов в корпоративных сетях, в частности РВС. Подсистема включает: данные, средства обработки (аппаратные и программные), активное компоненты (процессы и действия пользователей). Представим параллельную систему в виде схемы (рис. 6.2).

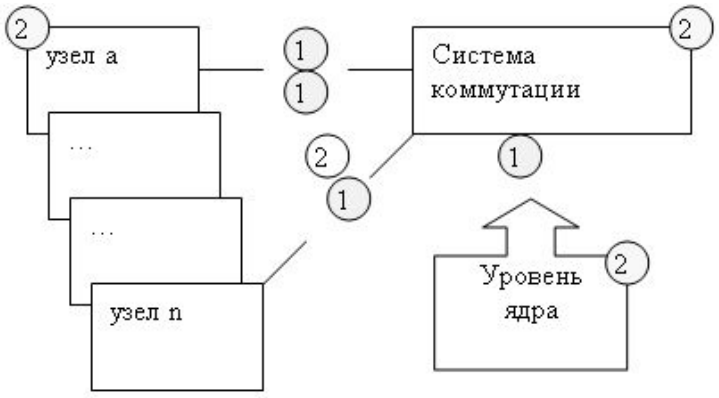

Рис. 6.2. Подсистема организации безопасности информационных ресурсов в корпоративных сетях

На схеме цифрами отмечены:

– аппаратные и программные средства обработки в системе коммутации;

– процессы и действия пользователей.

Для параллельной системы аппаратных и программных средств требуется больше, с каждым дополнительным модулем вычисления, которое еще больше усложняет систему защиты, снижая в некоторой степени общую вычислительную мощность системы в целом.

Для защиты данных используется ряд методов, таких как:

Алгоритмы запутывания - используются хаотические переходы в разные части кода, внедрения ошибочных процедур – «пустышек», неженаты циклы, искажение количества реальных параметров процедур ПО, разброс участков кода по разным областям ОЗУ и тому подобное

Алгоритмы мутации - создаются таблицы соответствия операндов - синонимов и замена их друг на друга при каждом запуске программы по определенной схеме или случайному образу, случайные изменения структуры программы.

Алгоритмы компрессии данных - программа упаковывается, а потом распаковывается по мере выполнения.

Алгоритмы шифровки данных - программа шифруется, а потом расшифровывается по мере выполнения.

Вычисление сложных математических выражений в процессе отработки механизма защиты - элементы логики защиты зависят от результата вычисления значения какой-либо формулы или группы формул.

Методы затруднения дизассемблирования - используются разные приемы, направленные на предотвращение дизассемблирования в пакетном режиме.

Методы затруднения отладки - используются разные приемы, направленные на осложнение отладки программы.

Эмуляция процессоров и операционных систем - создается виртуальный процессор и операционная система (не обязательно реально существующие) и программа-переводчик из системы команд IBM в систему команд созданного процессора или ОС, после такого перевода ПО может выполняться только с помощью эмулятора, который резко затрудняет исследование алгоритма ПО.

Нестандартные методы работы с аппаратным обеспечением - модули системы защиты обращаются к аппаратуре ЭВМ, минуя процедуры операционной системы, и используют малоизвестные или недокументируемые ее возможности.

Из рассмотренного выше можно заключить, что основным отличительным звеном уязвимости параллельных систем, являются апаратно-программные средства системы коммутации (так как увеличивается их число) в соответствии с требованиями решаемых задач.

Потому для средств коммутации актуальным является применение методов шифрования данных.

Существует симметричное (традиционное) и асимметричное шифрование данных нет смысла останавливаться на данных методах они известны и применяются во многих последовательных системах. Но наиболее перспективным для исследования и дальнейших модификаций в параллельных системах является метод блуждающих ключей.

6.6.4. Защита данных при помощи метод блуждающих ключей

Проблема распределения ключей является наиболее острой в больших информационных системах к которым возможно отнести большую часть параллельных. Частично эта проблема решается (а точнее снимается) за счет использования открытых ключей. Но наиболее надежные криптосистемы с открытым ключом типа RSA достаточно трудоемки, а для шифрования мультимедийных данных и вовсе не приспособлены.

Оригинальные решения проблемы "блуждающих ключей" активно разрабатываются специалистами. Эти системы являются некотором компромисом между системами с открытыми ключами и обычными алгоритмами, для которых требуется наличие одного и того же ключа и у отправителя и получателя.

Рассмотрим суть идеи метода. После того, как ключ использован в одном сеансе по некоторому правилу он изменяется другим. Это правило должно быть известено и отправителю, и получателю. Зная правило, при получении очередного сообщения получатель тоже меняет ключ. Если правило изменения ключей придерживается и отправителем и получателем, то в каждый момент времени они имеют одинаковый ключ. Постоянное изменение ключа затрудняет раскрытие информации злоумышленником.

Основное задание в реализации этого метода - выбор эффективного правила изменения ключей. Наиболее простой путь - генерация случайного списка ключей. Изменение ключей осуществляется в порядку списке. Однако, очевидно список придется каким-то образом передавать.

Другий вариант - использование математических алгоритмов, основанных на так называемых перебирающих последовательностях. На огромном количестве ключей путем одной и той же операции над элементом выходит другой элемент. Последовательность этих операций позволяет переходить от одного элемента к другому, пока не будет перебрано все множество.

Реализация криптографических методов

Проблема реализации методов защиты информации имеет два аспекта (рис. 6.3):

1. разработку средств, реализующихх криптографические алгоритмы,
2. методику использования этих средств.

Рис. 6.3. Схема реализации криптографических методов защиты данных

Каждый из рассмотренных криптографических методов может быть реализован или програмным, или аппаратным способом.

Возможность программной реализации обусловливается тем, что все методы криптографического превращения формальны и могут быть представлены в виде конечной алгоритмической процедуры.

При апаратной реализации все процедуры шифрования и дешифрования выполняются специальными электронными средствами. Наибольшее распространение получили модули, реализующие комбинированные методы.

Перейдем к практической реализации данных методов.

Возьмем несколько базовых алгоритмов шифрования применяемых на сегодняшний день [5, 6].

Алгоритм DES

DES (Data Encryption Standart) – это симметричный алгоритм шифрования, т. е. один ключ используется как для зашифровки, так и для расшифровки сообщений. Разработан фирмой IBM и утвержден правительством США в 1977 как официальный стандарт. DES имеет блоки по 64 бит и основан на 16-ти кратной перестановке данных, также для зашифровки использует ключ в 56 бит. Существует несколько режимов DES, например Electronic Code Book (ECB) и Cipher Block Chaining (CBC). 56 бит - это 8 семибитовых ASCII символов, т. е. пароль не может быть длиннее 8 букв. Если вдобавок использовать только буквы и цифры, то количество возможных вариантов будет гораздо меньше максимально возможных 256.

Рассмотрим один из шагов алгоритма DES. Входной блок данных делится пополам на левую (L') и правую (R') части. После этого формируется выходной массив так, что его левая часть L'' представлена правой частью R' входного, из 32-битового слова R' с помощью битовых перестановок формируется 48-битовое слово. К полученному 48-битовому слову и 48-битовому раундовому ключу применяется операция XOR . Результирующее 48-битовое слово разбивается на 8 6-битовых групп, каждая 6-битовая группа с помощью соответствующего S-box'a заменяется на 4-битовую группу и из полученных восьми 4-битовых групп составляется 32-битовое слово. К полученному слову и L' применяется XOR, в результате получается R''. Можно убедиться, что все проведенные операции могут быть обращены, и расшифровка может осуществляться за число операций, линейно зависящее от размера блока. После нескольких таких проходов можно считать, что каждый бит выходного блока шифровки может зависеть от каждого бита сообщения.

Алгоритм «тройной DES»

Так как текст, зашифрованный двойным DES (встреча на середине (meet in the middle)), оказывается хрупким при криптографической атаке, то текст шифруется 3 раза DES. Таким образом, длина ключа возрастает до 168-бит (56x3). Не всегда применение тройного DES означает увеличение уровня безопасности сообщения. Типы тройного шифрования DES:

– DES-EEE3: шифруется 3 раза с 3 различными ключами.

– DES-EDE3: 3 DES операции шифрование - дешифрование - шифрование с 3 различными ключами.

– DES-EEE2 и DES-EDE2: как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

В табл. 6.1 приведено сравнение различных видов DES шифрования.

Таблица 6.1

Сравнение различных видов DES шифрования

# Шифрования	# Ключей	Вычисление (Computation)	Хранение (Storage)	Тип атаки
Одиночный	1	256	-	known plaintext
Одиночный	1	238	238	chosen plaintext
Одиночный	1	-	256	chosen plaintext
Двойной	2	2112	-	known plaintext
Двойной	2	256	256	known plaintext
Двойной	2	-	2112	chosen plaintext
Тройной	2	2112	-	known plaintext
Тройной	2	256	256	256 chosen plaintext
Тройной	2	2(120- t)	-	2t known plaintext
Тройной	2	-	256	chosen plaintext
Тройной	3	2112	256	known plaintext
Тройной	3	256	2112	chosen plaintext

Алгоритм ГОСТ

ГОСТ предусматривает 3 режима шифрования (простая замена, гаммирование, гаммирование с обратной связью) и один режим выработки имитовставки. Первый из режимов шифрования

предназначен для шифрования ключевой информации и не может использоваться для шифрования других данных, для этого предусмотрены два других режима шифрования. Режим выработки имитовставки (криптографической контрольной комбинации) предназначен для имитозащиты шифруемых данных, то есть для их защиты от случайных или преднамеренных несанкционированных изменений.

Алгоритм построен по тому же принципу, что и DES – это классический блочный шифр с секретным ключом – однако отличается от DES'а большей длиной ключа, большим количеством раундов, и более простой схемой построения самих раундов.

Из-за намного большей длины ключа ГОСТ гораздо устойчивей DES'а к вскрытию «грубой силой» – путем полного перебора по множеству возможных значений ключа.

Функция шифрования ГОСТа гораздо проще функции шифрования DES'а, она не содержит операций битовых перестановок, коими изобилует DES и которые крайне неэффективно реализуются на современных универсальных процессорах (хотя очень просто аппаратно - путем разводки проводников в кристалле или на плате). В силу сказанного, при вдвое большем количестве раундов (32 против 16) программная реализация ГОСТа на процессорах Intel x86 более чем в 2 раза превосходит по быстродействию реализацию DES'а. Естественно, сравнивались близкие к оптимуму по быстродействию реализации.

Из других отличий ГОСТа от DES'а надо отметить следующие:

На каждом раунде шифрования используется «раундовый ключ», в DES'е он 48-битовый и вырабатывается по относительно сложному алгоритму, включающему битовые перестановки и замены по таблице, в ГОСТе он берется как фрагмент ключа шифрования. Длина ключа шифрования в ГОСТе равна 256 битам, длина раундового ключа - 32 битам, итого получаем, что ключ шифрования ГОСТа содержит 256/32=8 раундовых ключей. В ГОСТе 32 раунда, следовательно, каждый раундовый ключ используется 4 раза, порядок использования раундовых ключей установлен в ГОСТе и различен для различных режимов.

Таблица замен в ГОСТе - аналог S-блоков DES'а - представляет собой таблицу (матрицу) размером 8x16, содержащую число от 0 до 15. В каждой строке каждое из 16-ти чисел должно встретиться ровно 1 раз. В отличие от DES'а, таблица замен в ГОСТе одна и та же для всех раундов и не зафиксирована в стандарте, а является сменяемым секретным ключевым элементом. От качества этой таблицы зависит качество шифра. При «"сильной»" таблице

замен стойкость шифра не опускается ниже некоторого допустимого предела даже в случае ее разглашения. И наоборот, использование «слабой» таблицы может уменьшить стойкость шифра до недопустимо низкого предела. Никакой информации по качеству таблицы замен в открытой печати России не публиковалось, однако существование «слабых» таблиц не вызывает сомнения - примером может служить «тривиальная» таблица замен, по которой каждое значение заменяется самим собой. Это делает ненужным для компетентных органов России ограничивать длину ключа – можно просто поставить недостаточно «сильную» таблицу замен.

Шифр Blowfish

Blowfish – это 64-битный блочный шифр разработанный Шнайером (Schneier) в 1993 году. Это шифр Файстела (Feistel), и каждый проход состоит из зависимой от ключа перестановки и зависимой от ключа с данными замены. Все операции основаны на операциях XOR и прибавлениях к 32-битным словам (XORs and additions on 32-bit words). Ключ имеет переменную длину (максимально 448 бит) и используется для генерации нескольких подключевых массивов (subkey arrays). Шифр был создан специально для 32-битных машин и существенно быстрее DES.

Шифр RC5

RC5 – это довольно быстрый блочный шифр разработанный Ривестом для RSA Data Security. Этот алгоритм параметричный, т.е. с переменным размером блока, длиной ключа и переменным числом проходов. Размер блока может быть 32, 64, или 128 битов. Количество проходов в промежутке от 0 до 2048 бит. Параметричность такого рода дает гибкость и эффективность шифрования.

RC5 состоит из ввода ключа (key expansion), шифрования и дешифровки. При вводе ключа вводятся также количество проходов, размер блока и т.д. Шифрование состоит из 3 примитивных операций: сложения, побитового XOR и чередования (rotation). Исключительная простота RC5 делает его простым в использовании. RC5 текст, также как и RSA, может быть дописан в конец письма в зашифрованном виде.

Безопасность RC5 основывается на зависящих от данных чередования и смешивания результатах различных операций. RC5 с размером блока 64 бита и 12 или более проходов обеспечивает хорошую стойкость к дифференциальному и линейному криптоанализу.

Шифр IDEA

IDEA (International Data Encryption Algorithm) - это вторая версия блочного шифра, разработанная К. Лейем (Lai) и Д. Месси (Massey) в конце 80-х. Это шифр, состоящий из 64-битных повторяющихся блоков со 128-битным ключом и восемью проходами (rounds). Хотя этот шифр не является шифром Файстела, дешифровка выполняется по тому же принципу, что и шифрование. Структура шифра была разработана для легкого воплощения как программно, так и аппаратно, и безопасность IDEA основывается на использовании трех не совместимых типов арифметических операций над 16-битными словами. Скорость программного IDEA сравнима со скоростью DES.

Один из принципов создания IDEA – затруднить дифференциальный криптоанализ. Ни одна линейная криптоаналитическая атака не закончилась успешно, как и не было выявлено алгебраически слабых мест. Самый полный анализ провел Daemen. Он открыл большой класс 251 слабых ключей, при использовании которых в процессе шифрования ключ может быть обнаружен и восстановлен. Однако, в IDEA существует 2128 возможных вариантов ключей, поэтому это открытие не влияет на практическую безопасность шифра.

Шифр RSA

RSA (авторами являются Rivest, Shamir и Alderman) – это система с открытым ключом (public-key), предназначенная как для шифрования, так и для аутентификации. Была разработана в 1977 году. Она основана на трудности разложения очень больших целых чисел на простые сомножители.

RSA очень медленный алгоритм. Для сравнения: на программном уровне DES по меньшей мере в 100 раз быстрее RSA, на аппаратном – в 1000-10000 раз, в зависимости от выполнения.

Алгоритм RSA состоит в следующем:

– Для двух очень больших целых чисел P и Q определяются N=PQ и M=(P - 1)(Q - 1).

– Выбирается случайное целое число D, взаимно простое с M, и вычисляется E = (1 mod M)/D.

– D и N публикуются как открытый ключ, а E сохраняется в тайне.

– Пусть S – сообщение. Его длина определяется значением выражаемого им целого числа и находится в интервале (1, N). S превращается в шифровку возведением в степень D по модулю N и отправляется получателю S ' = (SD mod N).

– Получатель сообщения расшифровывает его, возведя в степень E (число E ему уже известно) по модулю N, т. к. S = ((S ')D mod N) = (SDE mod N).

Шифрование PGP

Pretty Good Privacy(PGP) - это программный пакет разработанный Филипом Циммерманом (Philip Zimmerman), который обеспечивает шифровку почты и файлов. Циммерман взял существующие криптосистемы и криптографические протоколы и разработал бесплатную (freeware) программу для различных платформ. Она обеспечивает шифрование сообщений, цифровые подписи и совместимую почту (email compatibility).

Алгоритмы, используемые для шифрования сообщений - это RSA для передачи ключа и IDEA для самого шифрования сообщений. Цифровые подписи достигаются при использовании RSA для подписи и MD5 для вычисления дайджеста сообщения (message digest). PGP использует ZIP компрессию, а также маскирует координаты и данные отправителя, что немного осложняет процесс анализа трафика. Совместимость почты достигается путем использования Radix-64 конвертации (conversion).

Таким образом, в ходе исследования показано, что потребность в использовании высокопроизводительных вычислений во всем мире относится к фундаментальным факторам развития стратегического потенциала и имеет важное научно-техническое и народно-хозяйственное значение. На сегодняшний день известны два основных метода повышения производительности и быстродействия вычислительных систем:

– использование все более совершенной элементной базы;

– параллельное выполнение вычислительных операций.

Первый способ связан с весьма значительными капиталовложениями. Опыт фирмы CRAY, создавшей суперкомпьютер на базе арсенида галлия, показал, что разработка принципиально новой элементной базы для высокопроизводительных вычислительных систем является непосильной задачей даже для таких именитых корпораций. Второй способ стал доминировать после объявления в США правительственной программы «Ускоренная стратегическая компьютерная инициатива» (ASCI).

На сегодняшний день оказался более перспективным следующий подход. Для построения суперкомпьютеров берутся серийные микропроцессоры, снабженные каждый своей локальной памятью, и соединяются посредством некоторой коммуникационной

среды. У такой архитектуры достоинств много: при необходимости можно добавлять процессоры, увеличивая производительность такого кластера; если ограничены финансовые возможности или заранее известна требуемая вычислительная мощность, то легко подбирать требуемую конфигурацию системы. Название таких систем подчеркивает теоретически неограниченную масштабируемость устройств такого класса.

Список источников

1. Лацис А.О. Как построить и использовать суперкомпьютер / А.О. Лацис. – М.: Бестселлер, 2003. – 240 с.

2. Гергель В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем: учеб. пособие / В.П. Гергель, Р.Г. Стронгин. – Н.Новгород: Н.НГУ, 2003. – 184 с.

3. Beowulf Introduction & Overview [Електронний ресурс]. – Режим доступу: http://www.beowulf.org.

4. Алишов Н.И. Развитые методы взаимодействия ресурсов в распределенных системах / Н.И. Алишов. – К.: Сталь, 2009. – 448 с.

5. Ковалевский В., Максимов В. Криптографические методы. // КомпьютерПресс. - 1993. - № 5. - с. 31-34.

6. Мафтик С. Механизмы защиты в сетях ЭВМ. - М.: Мир, 1996.

7. Малюк А.А. Информационная безопасность – концептуальные и методологические основы защиты информации / А.А. Малюк. – М.: Горячая линия – Телеком, 2004. – 280 с.

Перечень обозначений к разделу 6

MPP - (Massively Parallel Processing). − массово-параллельная архитектура класс архитектур параллельных вычислительных систем. Особенность архитектуры состоит в том, что память физически разделена.

PCI шиной - Peripheral component interconnect шина ввода/вывода для подключения периферийных устройств к материнской плате компьютера.

Fast Ethernet — является эволюционным развитием классической технологии Ethernet.

ПЭВМ − сокр. персональная электронно вычислительная машина.

ЭВМ −электронная вычислительная машина.

ВКК - векторно-конвейерные компьютеры.

БИС - большая интегральная схема.

SMP - (Symmetric Multi Processing) Параллельные компьютеры с общей памятью.

MPP - (Massively Parallel Processing) Массивно-параллельные компьютеры с распределенной памятью.

MPI - (Message Passing Interface интерфейс передачи сообщений) программный интерфейс для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу.

PVM - (Parallel Virtual Machine) параллельная виртуальная машина) общедоступный программный пакет, позволяющий объединять разнородный набор компьютеров в общий вычислительный ресурс («виртуальную параллельную машину») и предоставляющий возможности управления процессами с помощью механизма передачи сообщений.

Стек протоколов TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть нтернет.

Linux (Ли́нукс) — общее название Unix-подобных операционных систем, основанных на одноимённом ядре

Flops (флопс) (FLoating-point Operations Per Second) — внесистемная единица используемая для измерения производительности компьютеров,

Channel bonding - соединение (обьединение) каналов

VLAN (Virtual Local Area Network) — логическая («виртуальная») локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения.

CPU – (central processing unit) центральное процессорное устройство ЦПУ

ATX (Advanced Technology Extended) — форм-фактор персональных настольных компьютеров.

DVD (Digital Versatile Disc — цифровой многоцелевой диск.

DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки узла) — сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

ПВК - персональный вычислительный кластер.

NetBIOS (Network Basic Input/Output System) — протокол для работы в локальных сетях, разработан в виде интерфейса, который не зависит от фирмы-производителя.

UPS - (Uninterruptible Power Supply) бесперебойный блок питания.

LINPACK — программная библиотека содержит набор подпрограмм для анализа и решения плотных систем линейных алгебраических уравнений.

SONET (Synchronous Digital Hierarchy) - Синхронная цифровая иерархия — это система передачи данных, основанная на синхронизации по времени передающего и принимающего устройства.

Fibre Channel (FC) — волоконный канал, семейство протоколов для высокоскоростной передачи данных.

СОДУ - системам обыкновенных дифференциальных уравнений.

Grid -кластерные конфигурации и распределены вычислительные системы.

NUMA (NUMA – Non-uniform Memory Access) — «неравномерный доступ к памяти» или «Архитектура с неравномерной памятью») — схема реализации компьютерной памяти, используемая в мультипроцессорных системах, когда время доступа к памяти определяется её расположением по отношению к процессору.

Алгоритм DES (Data Encryption Standart) – это симметричный алгоритм шифрования.

Blowfish – это 64-битный блочный шифр разработанный Шнайером (Schneier) в 1993 году.

RC5 – алгоритм параметричный, т.е. с переменным размером блока, длиной ключа и переменным числом проходов.

IDEA (International Data Encryption Algorithm) - это вторая версия блочного шифра.

RSA (авторами являются Rivest, Shamir и Alderman) – это система с открытым ключом (public-key), предназначенная как для шифрования, так и для аутентификации.

PGP (Pretty Good Privacy) — компьютерная программа, также библиотека функций, позволяющая выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации.