Введение в криптоанализ. Криптоанализ симметричных криптосистем: блочные шифры

Лекция

Ревизия: 0.1

История изменений

15.10.2009 – Версия 0.1. Первичный документ. Ковтун В.Ю.

Содержание

Лекция 4. Введение в криптоанализ. Криптоанализ симметричных криптосистем: блочные шифры. Часть 1

Вопросы

1. Введение в криптоанализ.
2. Криптоанализ симметричных блочных шифров.
3. Особенности реализации.

Введение

Основоположник современной криптографии Клод Шеннон, показал [102], что если на любой исходный текст наложить (т.е. сложить по модулю с текстом) ключ длины не меньшей, чем само сообщение, то такой шифр будет **нераскрываемым**: потенциальному злоумышленнику потребуется перебрать все возможные ключи и каждым из них попробовать расшифровать сообщение. Однако использование такого способа шифрования, получившего название **«одноразовых блокнотов»**, в большинстве случаев оказывается слишком дорогим и неоправданным. Это связано с тем, что нет смысла бороться за устойчивость системы защиты информации к взлому ниже некоторой «фоновой» вероятности, т.е. вероятности события, которое мы не в состоянии предотвратить [77]. Например, если вероятность выхода компании из бизнеса равна 2^{-30} (менее чем один из миллиона), то есть ли смысл для защиты информации, которая может нанести компании ущерб, сопоставимый с кризисом рынка, использовать алгоритм, вероятность вскрытия которого за приемлемое время составляет 2^{-100}?

Выбор необходимой степени защиты информации и средств ее обеспечения является важной задачей и должен учитывать ряд параметров: уровень секретности информации; ее стоимость; время, в течение которого она должна оставаться в тайне и т.д. Проблема защиты информационных ресурсов в настоящее время приобретает все более важное значение. Так, по данным отчета CSI/FBI Computer Crime and Security Survey 2005 [27], средний ущерб каждой компании, в которой в минувшем году была зафиксирована утечка конфиденциальных данных, составил 355,5 тыс. долларов (причем по сравнению с 2004 годом эта цифра возросла почти вдвое). По некоторым оценкам, экономические потери от злонамеренных атак на банковские системы по всему миру составляют ежегодно около 130 млрд. долларов.

Как известно [60], далеко не все присутствующие на рынке криптографические средства обеспечивают обещанный уровень защиты. Системы и средства защиты информации (СЗИ) характеризуются тем, что для них не существует простых и однозначных тестов, позволяющих убедиться в надежной защите информации. Например, для проверки работоспособности системы связи достаточно провести ее испытания. Однако успешное завершение этих испытаний не позволяет сделать вывод о том, что встроенная в нее подсистема защиты информации тоже работоспособна. Задача определения эффективности СЗИ при использовании криптографических методов защиты зачастую более трудоемкая, чем разработка СЗИ, требует наличия специальных знаний и более высокой квалификации, чем задача разработки. Часто анализ нового шифра является новой научной, а не инженерной задачей.

Эти обстоятельства приводят к тому, что на рынке появляются средства криптографической защиты информации, про которые никто не может сказать ничего определенного. При этом нередко разработчики держат криптоалгоритм (как показывает практика, часто легко взламываемый) в секрете. Однако задача точного определения используемого криптоалгоритма не может быть гарантированно сложной хотя бы потому, что он известен разработчикам. Кроме того, если нарушитель нашел способ преодоления защиты, то не в его интересах об этом заявлять. В результате пользователи таких СЗИ попадают в зависимость как минимум от разработчика. Поэтому обществу должно быть выгодно открытое обсуждение безопасности СЗИ

массового применения, а сокрытие разработчиками криптоалгоритма должно быть недопустимым [99].

Современная криптография — соревнование методов шифрования и криптоанализа. *Криптоанализом* (от греческого *kryptos* - "скрытый" и *analyein* -"ослаблять" или избавлять") называют науку восстановления (дешифрования) открытого текста без доступа к ключу. **Фундаментальное допущение криптоанализа, впервые сформулированное Кирхгоффом [34], состоит в том, что секретность сообщения всецело зависит от ключа, т.е. весь механизм шифрования, кроме значения ключа, известен противнику.** Как бы то ни было, секретность алгоритма не является большим препятствием: например, для определения типа программно реализованного криптографического алгоритма требуется лишь несколько дней инженерного анализа исполняемого кода.

Криптоанализ ставит своей задачей в разных условиях получить дополнительные сведения о ключе шифрования, чтобы значительно уменьшить диапазон вероятных ключей. Результаты криптоанализа могут варьироваться по степени практической применимости. Так, криптограф Ларс Кнудсен [36] предлагает следующую классификацию успешных исходов криптоанализа блочных шифров в зависимости от объема и качества секретной информации, которую удалось получить:

- Полный взлом – криптоаналитик извлекает секретный ключ.
- Глобальная дедукция – криптоаналитик разрабатывает функциональный эквивалент исследуемого алгоритма, позволяющий зашифровывать и расшифровывать информацию без знания ключа.
- Частичная дедукция – криптоаналитику удается расшифровать или зашифровать некоторые сообщения.
- Информационная дедукция – криптоаналитик получает некоторую информацию об открытом тексте или ключе.

Однако взлом шифра совсем не обязательно подразумевает обнаружение способа, применимого на практике для восстановления открытого текста по перехваченному зашифрованному сообщению. В научной криптологии другие правила [57]. **Шифр считается взломанным**, если в системе обнаружено слабое место, которое может быть использовано для более эффективного взлома, чем метод полного перебора ключей («brute-force approach»). Допустим, для дешифрования текста методом полного перебора требуется перебрать 2^{128} возможных ключей; тогда изобретение способа, требующего для дешифрования 2^{100} операций по подбору ключа, будет считаться взломом. Такие способы могут требовать нереалистично больших объемов подобранного открытого текста или памяти ЭВМ. Под **взломом** понимается лишь подтверждение наличия уязвимости криптоалгоритма, свидетельствующее о том, что свойства надежности шифра не соответствуют заявленным характеристикам. Как правило, криптоанализ начинается с попыток взлома упрощенной модификации алгоритма, после чего результаты распространяются на полноценную версию: прежде чем браться за взлом, например, 16-раундовой версии DES, естественно для начала попытаться взломать шифр с меньшим количеством раундов, чем указано в его спецификации (например, 8-раундовую версию шифра).

Попытка криптоанализа называется ***атакой***. Прежде чем классифицировать атаки, введем ряд обозначений: открытый текст будем обозначать буквой x, шифртекст - буквой y (в качестве x *може*т выступать любая последовательность битов: текстовый файл, оцифрованный звук, точечный рисунок и т.д.). Пусть для зашифровывания и расшифровывания, используются ключи k и k' соответственно (в симметричной криптографии $k = k'$); обозначим функцию зашифровывания E_k, расшифрования - $D_{k'}$. Тогда выполняются $E_k(x) = y$, соотношения $D_{k'}(y) = x$.

Известны четыре основных типа криптоаналитических атак. В каждом случае предполагается (согласно фундаментальному допущению Кирхгоффа), что криптоаналитик знает используемый алгоритм шифрования.

- **Атака на основе только шифртекста.** Криптоаналитик располагает шифртекстами $y_1, \ldots, y_m$, полученными из неизвестных открытых текстов $x_1, \ldots, x_m$ различных сообщений. Требуется найти хотя бы один из x_i, $i = \overline{1, m}$ (или соответствующий ключ k_i), исходя из достаточного числа m криптограмм, или убедиться в своей неспособности сделать это. В качестве частных случаев возможно совпадение ключей: $k_1 = \cdots = k_m$ или совпадение открытых текстов: $x_1 = \cdots = x_m$.
- **Атака на основе открытого текста.** Криптоаналитик располагает парами $(x_1, y_1), \ldots, (x_m, y_m)$ открытых и соответствующим им зашифрованных текстов. Требуется определить ключ k_i для хотя бы одной из пар. В частном случае, когда $k_1 = \cdots = k_m = k$, требуется определить ключ k или, убедившись в своей неспособности сделать это, определить открытый текст x_{m+1} еще одной криптограммы y_{m+1}, зашифрованной на том же ключе.
- **Атака на основе подобранного открытого текста** отличается от предыдущей лишь тем, что криптоаналитик имеет возможность выбора открытых текстов $x_1, \ldots, x_m$. Цель атаки та же, что и предыдущей. Подобная атака возможна, например, в случае, когда криптоаналитик имеет доступ к шифратору передающей стороны.
- **Атака на основе адаптивно подобранного открытого текста.** Это частный случай вышеописанной атаки с использованием подобранного открытого текста. Криптоаналитик может не только выбирать используемых шифруемый текст, но также уточнять свой последующий выбор на основе полученных ранее результатов шифрования.

Атаки с использованием известного или подобранного открытого текста встречаются чаще, чем можно подумать. **Необходимым требованием** к хорошему криптоалгоритму является способность противостоять таким атакам. Это означает, что рассекречивание некоторой информации, передававшейся по каналу связи в зашифрованном виде, не должно приводить к рассекречиванию другой информации, зашифрованной на этом ключе. Кроме того, указанное требование учитывает особенности эксплуатации аппаратуры и допускает некоторые вольности со стороны оператора или лиц, имеющих доступ к формированию засекреченной информации. В среде криптоаналитиков нельзя назвать неслыханными факты добычи открытого текста шифрованного сообщения или подкупа лица, которое должно будет зашифровать избранное сообщение. Применяются и «косвенные» методы осуществления атаки на основе подобранного шифртекста. Злоумышленник может убедить обладателя секретного ключа переслать некое сообщение, но в зашифрованной форме. Так, в [31] описан прием, использованный командованием военно-морского флота США во время Второй Мировой Войны перед битвой на Мидвее. Чтобы убедиться в правильности результатов работы по взлому японского военного шифра, криптоаналитики США попросили американский гарнизон, дислоцированный на Мидвее, сообщить по открытому незащищенному каналу о нехватке пресной воды. Спустя два дня было перехвачено секретное сообщение, в котором японцы, осуществлявшие мониторинг использованного канала, сообщали о проблемах с водой в некоем "AF". Благодаря этому американцы узнали, "AF" – кодовое обозначение Мидвея в шифрограммах противника. Атаки на основе подобранных текстов считаются наиболее опасными.

Указанные криптоатаки относятся к классу ***пассивных*** [87]. Так классифицируются действия противника, который «пассивно изучает» шифрованные сообщения, может их перехватить и подвергнуть криптоанализу с целью получения информации об открытом тексте или ключе. Однако современные технические средства позволяют потенциальному противнику **«активно»** вмешиваться в процесс передачи сообщения. Обычно различают два типа активных атак, которые носят названия **имитации и подмены** сообщения. Атака имитации состоит в том, что противник «вставляет» в канал связи сфабрикованное им «шифрованное сообщение», которое на самом деле не

передавалось от законного отправителя к получателю. При этом противник рассчитывает на то, что получатель воспримет это сообщение как подлинное (аутентичное). Атака подмены состоит в том, что противник, наблюдая передаваемое по каналу связи подлинное сообщение от отправителя, «изымает» его и заменяет поддельным. Различные шифры могут быть более или менее уязвимыми к активным атакам. Способность самого шифра (без использования дополнительных средств) противостоять активным атакам обычно называют имитостойкостью шифра. Количественной мерой имитостойкости шифра служат вероятности успеха имитации и подмены соответственно. Эти вероятности определяют шансы противника на успех при навязывании получателю ложного сообщения.

Атаки можно также классифицировать по объему ресурсов, необходимых для их осуществления:

- Память – объем памяти, требуемый для реализации атаки;
- Время – количество элементарных операций, которые необходимо выполнить;
- Данные – необходимый объём открытых и соответствующим им зашифрованных текстов. В некоторых случаях эти параметры являются взаимозависимыми: например, за счет увеличения памяти можно сократить время атаки.

Способность криптосистемы противостоять атакам криптоаналитика называется ***стойкостью***. Количественно стойкость измеряется как сложность наилучшего алгоритма, приводящего криптоаналитика к успеху с приемлемой вероятностью. Универсальный метод прямого перебора множества всех возможных ключей позволяет получить оценку сверху для стойкости алгоритма шифрования. **Проблема всей современной криптографии** – это отсутствие нижней границы стойкости; длина ключа задаёт лишь общий объём пространства ключей, но всегда есть вероятность, ткнув пальцем в небо, угадать решение. **Относительное ожидаемое безопасное время** определяется как полупроизведение числа открытых ключей и времени, необходимого криптоаналитику для того, чтобы испытывать каждый ключ. В зависимости от целей и возможностей криптоаналитика меняется и стойкость. Различают **стойкость ключа** (сложность раскрытия ключа наилучшим известным алгоритмом), **стойкость бесключевого чтения**, **имитостойкость** (сложность навязывания ложной информации наилучшим известным алгоритмом) и вероятность навязывания ложной информации. Аналогично можно различать стойкость собственно криптоалгоритма, стойкость протокола, стойкость алгоритма генерации и распространения ключей [99].

В зависимости от сложности взлома алгоритмы обеспечивают различные степени защиты. Во главу угла ставится принципиальная возможность получения по перехвату некоторой информации об открытом тексте или использованном ключе. Существуют ***безусловно стойкие*** (или ***теоретически стойкие***), ***доказуемо стойкие*** и ***предположительно стойкие*** **криптоалгоритмы**.

Теоретически стойкие системы создают шифртексты, содержащие недостаточное количество информации для однозначного определения соответствующих им текстов (или ключей). В лучшем случае открытый текст может быть локализован в достаточно большом подмножестве множества всех открытых текстов, и его можно лишь «угадать» с ничтожно малой вероятностью. Для совершенного шифра открытый текст «локализуется» во всем множестве открытых текстов. Тем самым, для него сама задача расшифровывания становится бессмысленной. Никакой метод криптоанализа, включая полный перебор ключей, не позволяет не только определить ключ или открытый текст, но даже получить некоторую информацию о них. Алгоритм безусловно стоек, если восстановление открытого текста невозможно при любом объеме шифртекста, полученного криптоаналитиком. Безопасность безусловно стойких криптоалгоритмов основана на доказанных теоремах о невозможности раскрытия ключа. Как уже говорилось, принципиально не раскрываемые шифры (например, совершенно секретные системы Клода Шеннона, в которых ключ не может использоваться повторно, а его размер больше либо равен объему текста) неудобны на практике (симметричные криптосистемы с разовым использованием ключа требуют большой защищенной памяти для хранения ключей, системы квантовой криптографии требуют волоконно-оптических каналов связи и являются дорогими, кроме того, доказательство

их безопасности уходит из области математики в область физики [99]). В силу своей непрактичности и высокой ресурсозатратности абсолютно стойкие шифры применяются только в сетях связи с небольшим объемом передаваемой информации, когда есть возможность обеспечить всех абонентов достаточным запасом случайных ключей и исключить возможность их повторного применения: обычно это сети для передачи особо важной государственной информации.

Стойкость ***доказуемо стойких*** криптоалгоритмов определяется сложностью решения хорошо известной математической задачи, которую пытались решить многие математики и которая является общепризнанно сложной. В качестве примера можно привести системы DH (Диффи-Хеллмана) [19] и RSA (Ривеста-Шамира-Адельмана) [53, 54], основанные на сложностях дискретного логарифмирования и разложения целого числа на множители соответственно. Достоинством доказуемо стойких алгоритмов является хорошая изученность задач, положенных в их основу, а недостатком - невозможность в случае необходимости оперативной доработки криптоалгоритмов, т.е. отсутствие гибкости. Повышение стойкости достигается увеличением размера математической задачи или ее заменой, что, как правило, влечет цепь изменений в аппаратуре, используемой для шифрования.

Предположительно стойкие криптоалгоритмы основаны на сложности решения частной математической задачи, которая не сводится к хорошо известным задачам и которую пытались решить один или несколько человек. Примерами могут служить шифры ГОСТ 28147-89 [83], AES [52], FEAL [63]. Предположительно стойкие криптоалгоритмы характеризует сравнительно малая изученность математических задач, на которых базируется их стойкость. Однако такие шифры обладают большой гибкостью, что позволяет при обнаружении слабых мест не отказываться от алгоритмов, а проводить их доработку.

Создание новых методов криптоанализа и повышение эффективности существующих методов необходимы как для анализа стойкости криптографических средств, так и для разработки методов их вскрытия. Каждый новый метод криптоанализа приводит к пересмотру безопасности шифров, к которым он применим.

Последнее десятилетие ознаменовалось резким ростом числа открытых работ по криптологии, а криптоанализ становится одной из наиболее активно развивающихся областей исследований. Появился целый арсенал математических методов, представляющих интерес для криптоаналитика.

Универсальные методы криптоанализа

Если целью криптоаналитика является раскрытие возможно большего числа шифров (независимо от того, хочет ли он этим нанести ущерб обществу, предупредить его о возможной опасности или просто получить известность), то для него наилучшей стратегией является разработка универсальных методов анализа [99]. Но эта задача является также и наиболее сложной.

Метод полного перебора

Часто криптоаналитики вскрывают шифры на ЭВМ методом перебора ключей. В процессе криптоанализа приходится перебирать миллиард ключей со скоростью тысяча ключей в секунду.

Предположим, злоумышленнику известна одна или несколько пар (x, y). Пусть для простоты для любой пары (x, y) существует единственный ключ k, удовлетворяющий соотношению $E_k(x) = y$. Упорядочим множество возможных ключей (пространство ключей) и будем последовательно проверять ключи из K на выполнения равенства $E_k(x) = y$. Если считать проверку одного варианта ключа $k \in K$ за одну операцию, то полный перебор ключей потребует $|K|$ - число элементов в множестве. Пусть ключ в схеме шифрования выбирается случайно и равновероятно из множества K. Тогда с

вероятностью $\frac{1}{|K|}$ ключ будет угадан и трудоемкость метода полного перебора будет равна 1. Поэтому естественно в качестве оценки трудоемкости метода взять математическое ожидание случайной величины α, где α-число опробований до момента обнаружения использованного ключа. Поскольку α-равномерно распределенная случайная величина, то $M(\alpha)=\frac{|K|}{2}$.

Алгоритмы полного перебора допускают распараллеливание, что позволяет значительно ускорить нахождение ключа. Известно два направления в организации параллельного вычисления ключа [76].

Во-первых, построение конвейера. Пусть алгоритм соотношения $E_k(x)=y$ представим в виде детерминированной цепочки простейших действий (операций): $O_1, O_2, \ldots, O_N$.

Возьмем N процессоров $A_1, A_2, \ldots, A_N$, зададим их порядок и положим, что i-ый процессор выполняет три одинаковые по времени операции:

1) прием данных от $(i-1)$-го процессора;

2) выполнение операции O_i;

3) передача данных следующему $(i+1)$-му процессору.

Тогда конвейер из N последовательно соединенных, параллельно и синхронно работающих процессоров работает со скоростью $\frac{v}{3}$, где v-скорость выполнения одной операции процессором. Второе направление распараллеливания состоит в том, что множество K разбивается на непересекающиеся подмножества $K_1, K_2, \ldots, K_Q$. Система из Q машин перебирает ключи так, что i-ая машина осуществляет перебор ключей из множества K_i, $i=\overline{1,Q}$. Система прекращает работу, если одна из машин нашла ключ. Самой большой сложностью в изложенном подходе является организация деления ключевого множества. Однако если организовать поиск ключа таким образом, что при каждом очередном опробовании каждый из *N* процессоров стартует со случайной точки, то время опробования увеличится, но схема значительно упростится. Как показано в работе [84], среднее число шагов опробования N процессорами (машинами) ключей из множества K в этом случае составляет $\frac{|K|}{N}$.

Реализация такого параллелизма предполагает различные решения. Самое очевидное решение - создание компьютерного вируса для распространения программы-взломщика в глобальной сети. Идея впервые была опубликована в [72]. Вирус должен использовать периоды простоя компьютера (по данным исследований, компьютер простаивает 70-90% времени) для осуществления перебора по множеству ключей. Рано или поздно один из зараженных компьютеров обнаружит искомый ключ (необходимо предусмотреть механизм оповещения злоумышленника); с ростом производительности компьютеров и скорости распространения вирусов угроза успешного исхода такой атаки растет.

В [51] описаны более оригинальные идеи распараллеливания вычислений: «Китайская лотерея», создание «криптоаналитических» водорослей и животных. Китайская лотерея предполагает, что в каждый радиоприемник и телевизор встроена микросхема, запрограммированная на автоматическую проверку различных множеств ключей после получения по эфиру пары открытый текст/шифртекст. Использование биотехнологий в перспективе сделает возможным осуществление более эффективных атак. Рассмотрим вымышленное животное под названием "DESозавр". Оно состоит из оптически прозрачных биологических клеток, умеющих тестировать возможные ключи. По какому-то широковещательному оптическому каналу в клетки передаются пары

открытый текст/шифртекст. Решения переносятся к органам речи DESозавра специальными клетками, путешествующими по кровеносной системе животного. В доисторические времена средний динозавр состоял примерно из 10^{14}клеток (без микробов). Если каждая клетка может выполнять миллион шифрований в секунду, вскрытие 56-битового ключа займет 710^{-4} сек, а 64-битового - не более 0,2 сек. Другой биологический подход заключается в создании методами генной инженерии криптоаналитических водорослей, умеющих вскрывать криптографические алгоритмы методом полного перебора. Водоросли могут покрывать большие пространства, что теоретически позволит создать некое подобие распределенного компьютера с огромным числом процессоров.

Теперь рассмотрим случай, когда криптоаналитик осуществляет атаку на основе только шифртекста. При осуществлении попытки определения ключа шифра по криптограмме путем ее расшифрования на разных ключах требуется некоторым образом анализировать выходные данные алгоритма и проверять их «осмысленность». Сегодня в качестве объекта шифрования может выступать графический файл или программа. В этом случае задача определения «осмысленности» выходных данных становится очень трудной. Рассмотрим более простой случай, а именно -защиту передаваемых текстовых сообщений. Когда известно, что открытый текст представляет собой предложение на естественном языке, проанализировать результат и опознать успешный исход дешифрования сравнительно несложно, тем более что нередко криптоаналитик располагает некоторой априорной информацией о содержании сообщения [87]. Требуется по небольшому отрезку текста решить, что собой представляет дешифрованный текст: осмысленное сообщение или набор случайных символов. Однако вручную выполнить анализ множества фрагментов дешифрированных текстов невозможно. Поэтому задачу выделения осмысленного текста (то есть обнаружение правильно дешифрированного текста) решают с помощью ЭВМ. В этом случае используют теоретические положения, разработанные в конце XIX века петербургским математиком Марковым А.А., -так называемые цепи Маркова [75].

Тем не менее, возможно, что несколько вариантов пройдут критерий на открытый текст. К. Шеннон привел следующий пример. Криптограмму WNAJW, полученную при использовании сдвигового шифра для шифрования текста на английском языке, порождают два открытых текста RIVER и ARENA, отвечающим ключам F (=5) и W (=22). При этом один из ключей является истинным, а другой – ложным. Для сдвигового шифра одинаковые криптограммы порождают и более длинные слова SULPHUR (сера) и PRIMERO (запал, учебник). Аналогичные примеры имеются и для русского языка: АГАТА – ОСОБО, БЕДНЯГ – КОНЧИМ, КОНОПЛЕЮ – ОТСТУПИВ и т.д. В [87] получены оценки числа ложных ключей для произвольного шифра замены. Так, среднее число ложных ключей θ_L относительно всех возможных шифртекстов длины L определяется формулой:

$$\theta_L = \sum_{v \in V^L} p(v) \cdot \theta_L(v),$$

где V^L -множество криптограмм длины L, $p(v)$ -вероятность появления криптограммы v, $\theta_L(v)$ -число ложных ключей, соответствующих данной криптограмме.

Противник заинтересован в получении некоторой вероятностной информации об исходном тексте сообщения. Например, известный факт написания текста на английском языке предоставляет криптоаналитику некоторую априорную информацию об этом сообщении даже до анализа шифровки. В этом случае он заранее знает, что слово «HELLO» является более вероятным началом сообщения, чем набор букв «FGHKM». Поэтому одной из целей криптоанализа может являться увеличение информации, относящейся к каждому возможному сообщению. Предположим, противник перехватил шифровку «ABCCD» и знает (или предполагает), что использованный шифр – это шифр простой замены. Анализ шифровки позволяет сделать вывод, что исходное сообщение состоит из пяти букв, причем на третьей и четвертой позициях стоит одна и та же буква, а остальные отличны от нее и различны между собой. Противник не может считать, что это сообщение «HELLO», потому что имеются и другие возможные сообщения, например, «TEDDY». Однако апостериорные

вероятности таких открытых текстов возрастают относительно их априорных вероятностей. В то же время апостериорная вероятность таких открытых текстов, как «PEACE» или «GATES», снижается до нуля вне зависимости от их априорных вероятностей. По Шеннону, криптосистема является совершенной, если после анализа закрытых текстов апостериорные вероятности возможных открытых текстов остаются такими же, какими были их априорные вероятности [102].

Ограничения применения

Можно подумать, что с ростом мощности компьютеров разрядность ключа, достаточная для обеспечения безопасности информации против атаки методом полного перебора, будет неограниченно расти. Однако это не так. Существуют фундаментальные ограничения вычислительной мощности, наложенные структурой вселенной: например, скорость передачи любого сигнала не может превышать скорость распространения света в вакууме, а количество атомов во Вселенной (из которых, в конечном счете, состоят компьютеры) огромно, но конечно. Так, например, в [18] описаны два фундаментальных ограничения:

1. **Предел, основанный на выделяемой Солнцем энергии.** Все вычисления потребляют энергию. Согласно принципам классической термодинамики и статистической механики, потребление энергии при осуществлении необратимого преобразования (вычисления) имеет порядок $k \cdot T$, где T – температура окружающей среды (по абсолютной шкале Кельвина), а *k*– постоянная Больцмана (равная 1.4 10^{-23} Дж/K). Мощность излучения Солнца составляет приблизительно 3.86 10^{26} Вт; таким образом, за весь свой предполагаемый период существования – 10 млрд. лет, или 310^{17} секунд – Солнце выделит около 10^{44} Дж). Предположим, температура – окружающей среды *T*= 10^{6} градусов, тогда энергозатраты на одну операцию составляют $1.4 \cdot 10^{-29}$ Дж. Значит, количество вычислительных операций, которые можно осуществить с использованием всей выделяемой солнцем энергии, равно выделяемой мощности, поделенной на количество энергии, необходимой для осуществления одной операции, т.е. всего 10^{73} операций. Такое количество операций потребовалось бы на взлом ключа из 73 десятичных цифр (или около 250 бит) методом прямого перебора при грубом предположении, что для проверки одного значения ключа необходима всего одна операция (на самом деле -сотни операций). Для справки, количество атомов солнечной системы – около $_{10}{}^{60}$.

2. **Предел, основанный на массе Земли.** Масса Земли составляет порядка $6 \cdot 10^{24}$ кг. Масса протона – $1{,}6 \cdot 10^{27}$ кг , т.е. Земля содержит приблизительно $4 \cdot 10^{51}$ протонов. Сопоставим каждому протону отдельный компьютер и примем за скорость выполнения операции на таком компьютере время, за которое луч света проходит расстояние, равное диаметру этого протона ($\frac{10^{-15}\,м}{3 \cdot 10^{10}\,м/с} = \frac{1}{3} \cdot 10^{-25}$ с). Таким образом, каждый компьютер может выполнять 3 10^{25} операций в секунду. Если все эти компьютеры будут работать параллельно, их суммарное быстродействие составит 4 $10^{51} \cdot 3\ 10^{25}$ операций в секунду, т.е. 10^{77} операций в секунду. Возраст Вселенной приблизительно 10 млрд. лет, т.е. 3 10^{17} секунд. За весь период существования Вселенной такие гипотетические компьютеры размером с протон смогли бы выполнить 3 10^{94} операций. При описанных в п. 1 предположений относительно количества операций, необходимых на проверку значения ключа, такое количество операций позволит взломать ключ длиной 95 десятичных цифр, или 320 бит.

Таким образом, минимальный размер ключа, необходимый для защиты информации от атак злоумышленника, будет расти по мере повышения быстродействия компьютеров; тем не менее, приведенные вышевычисления показывают, что существует возможность выбрать такую длину ключа, что атаку методом полного перебора будет осуществить в принципе невозможно, вне зависимости от повышения вычислительной мощности компьютеров или успехов в области классической теории алгоритмов.

Атака по ключам

Согласно [88], одной из причин ненадежности криптосистем является использование слабых ключей. Согласно уже упомянутому принципу Кирхгофа, стойкость криптоалгоритма определяется секретностью ключа. Слабый ключ – это ключ, не обеспечивающий достаточного уровня защиты или использующий в шифровании закономерности, которые могут быть взломаны. Обычно считается, что алгоритм симметричного шифрования должен по возможности не иметь слабых ключей. Если это невозможно, то количество слабых ключей должно быть минимальным, чтобы уменьшить вероятность случайного выбора одного из них. Тем не менее, все слабые ключи должны быть заранее известны, чтобы их можно было отбраковать в процессе создания ключа.

Генераторы случайных чисел – ещё одно уязвимое место криптографических систем [59]. Это означает, что, если для генерации ключей используется криптографический слабый алгоритм, независимо от используемого шифра вся система будет нестойкой. Качественный ключ, предназначенный для использования в рамках симметричной криптосистемы, представляет собой случайный двоичный набор. Если требуется ключ разрядностью n, в процессе его генерации с одинаковой вероятностью должен получаться любой из 2^n возможных вариантов. Генерация ключей для асимметричных криптосистем – процедура более сложная, т.к. ключи, применяемые в таких системах, должны обладать определенными математическими свойствами. Например, в случае системы RSA модуль шифрования представляет собой произведение двух больших простых чисел.

Рассмотрим понятие криптографически стойкого генератора псевдослучайных кодов, или, для краткости, псевдослучайного генератора, которое было введено Блюмом и Микали [25]. Пусть $g:\{0,1\}^n \to \{0,1\}^{q(n)}$ -функция, вычислимая за полиномиальное от n время, $q(n)$-некоторый полином. Такая функция называется **генератором**. Генератор g является **псевдослучайным**, если порождаемые им последовательности неотличимы никаким полиномиальным вероятностным алгоритмом от случайных последовательностей той же длины $q(n)$. В 1989-1990 гг. Импальянцо, Левин и Луби [78] и Хостад [22] доказали, что псевдослучайные генераторы существуют тогда и только тогда, когда существуют односторонние функции.

С помощью псевдослучайных генераторов можно строить стойкие криптосистемы. Хорошие генераторы случайных чисел сложны в разработке, так как их надёжность часто зависит от особенностей аппаратного и программного обеспечения. В связи с этим ведется поиск способов построения эффективных псевдослучайных генераторов на основе различных криптографических предположений. Для генерации ключевой информации, предназначенной для использования в рамках симметричной криптосистемы, используются следующие методы (в порядке возрастания качества) [88]:

- Программная генерация, предполагающая вычисление очередного псевдослучайного числа как функции текущего времени, последовательности символов, введенных пользователем, особенностей его клавиатурного почерка и т.д.;
- Программная генерация, основанная на моделировании качественного псевдослучайного генератора с равномерным законом распределения;
- Аппаратная генерация с использованием качественного псевдослучайного генератора;
- Аппаратная генерация с использованием генераторов случайных последовательностей, построенных на основе физических генераторов шума и качественных псевдослучайных генераторов.

Показателем эффективности служит количество операций, затрачиваемых на вычисление каждого очередного бита псевдослучайной последовательности. Псевдослучайный генератор характеризуются периодом, разбросом, а также необходимостью его инициализировать. Малый период и плохой разброс относятся к

недостаткам псевдослучайного генератора. В случае малого периода (когда псевдослучайных значений, вырабатываемых генератором, меньше, чем возможных значений ключа) злоумышленник может сократить время поиска ключа, перебирая не сами ключи, а псевдослучайные значения, и генерируя из них ключи. Невысокое качество программных методов формирования объясняется также необходимостью защиты от разрушающих программных воздействий.

Лучший способ генерации множества случайных битов – извлечение их из естественно случайных событий реального мира [58]. Часто такой метод требует наличия специальной аппаратуры, но можно реализовать его и на компьютерах. Дж. Б. Эгнью предложил генератор случайных битов, который можно встроить в СБИС [2]. Это конденсатор металл-изолятор-полупроводник. Два таких конденсатора помещаются рядом друг с другом, а случайный бит определяется разностью зарядов этих конденсаторов. Другой генератор случайных битов генерирует поток случайных битов, используя нестабильность частоты свободно колеблющегося осциллятора [21]. Коммерческая микросхема фирмы AT&T генерирует случайные числа, опираясь именно на это явление [5], а генератор М. Гьюда [28] собирает случайные числа из физических явлений (например, радиоактивного распада). В качестве случайных величин можно также рассматривать интервалы между нажатиями клавиш клавиатуры. Главный недостаток подобных систем – возможные закономерности в генерируемой последовательности. Используемые физические процессы могут быть случайны, однако использование измерительных инструментов может привести к появлению проблем: смещения, отклонения или корреляции между битами. Обойти эти недостатки можно, используя не один, а несколько случайных источников. В качестве случайных событий Брюс Шнайер предлагает рассматривать, например:

- моменты нажатия на клавиши;
- моменты поступления команд от мыши;
- номер сектора, время дня и задержку поиска для каждой дисковой операции;
- фактической положение курсора мыши;
- номер текущей строки развертки монитора;
- содержимое текущего выводимого на экран изображения;
- содержимое таблиц файловой системы FAT;
- момент окончания загрузки процессора;
- время поступления сетевых пакетов и т.д.

Применяя к этим событиям однонаправленную функцию, можно сохранять полученные случайные величины в накопителе (пуле) и при необходимости их извлекать.

Counterpane опубликовала новый класс атак на генераторы случайных чисел [33], основанный на работе компании над коммерческими моделями. Одна из самых неожиданных находок была в том, что определённые генераторы случайных чисел могут быть надёжными при использовании с одной целью, но ненадёжными для другой; обобщение анализа надёжности опасно.

Частотный анализ

На протяжении веков дешифрованию криптограмм помогает частотный анализ появления отдельных символов и их сочетаний. Вероятности появления отдельных букв в тексте сильно различаются. Для русского языка, например, буква "о" появляется в 45 раз чаще буквы "ф" и в 30 раз чаще буквы "э". Анализируя достаточно длинный текст, зашифрованный методом замены, можно по частотам появления символов произвести обратную замену и восстановить исходный текст.

В таблице 1 [75] приведены относительные частоты появления русских букв.

Таблица 1. Относительные частоты появления русских букв

Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота
о	0,09	в	0,038	з	0,016	ж	0,007

е, ё	0,072	л	0,035	ы	0,016	ш	0,006
а	0,062	к	0,028	б	0,014	ю	0,006
и	0,062	м	0,026	ь, ъ	0,014	ц	0,004
н	0,053	д	0,025	г	0,013	щ	0,003
т	0,053	п	0,023	ч	0,012	э	0,003
с	0,045	у	0,021	й	0,01	ф	0,002
р	0,04	я	0,018	х	0,009		

Относительная частота появления пробела или знака препинания в русском языке составляет 0,174. Кроме того, порядок букв в словах и фразах естественного языка подчиняется определенным статистическим закономерностям. Частотный анализ также учитывает частоту появления различных буквосочетаний: например, пара стоящих рядом букв «ся» в русском языке более вероятна, чем «цы», а «оь» не встречается никогда. Для большинства естественных языков такая статистика документирована.

Эти принципы широко применяются в распространенных сегодня программах по подбору паролей. Возможные методы подбора пароля (могут применяться в совокупности) [86]:

- неоптимизированный перебор;
- перебор, оптимизированный по словарям вероятных паролей;
- перебор, оптимизированный на основе встречаемости символов и биграмм;
- перебор, ориентированный на информацию о подсистеме аутентификации ОС. Если ключевая система ОС допускает существование эквивалентных паролей, при переборе из каждого класса эквивалентности опробуется всего один пароль;
- перебор с использованием знаний о пользователе. Как правило, опробуются пароли, использование которых представляется наиболее вероятным.

Если программа перебора вначале подбирает наиболее вероятные пароли, а менее вероятные оставляет на потом, то перебор сокращается в десятки и сотни раз. В [80] приводится ряд результатов, полученных при подборе пароля. Числа, указанные в первой колонке таблицы 2, соответствуют сложности полного перебора. Однако применялся оптимизированные перебор, а в первом случае пароль представлял собой два английских слова, записанных без пробела. Таким образом, время перебора сократилось во много раз. Во втором же случае пароль состоял из трехстрочных английских букв, двух заглавных английских букв и одной цифры и был абсолютно бессмысленным.

Таблица 2. Сложность перебора паролей с учетом особенностей языка

Сложность перебора	Время перебора	Тип процессора
2, 08 10^{11}	15 минут	486DX/4-100
5, 68 10^{10}	8 часов	Pentium-120

Частотный криптоанализ использует статистические и лингвистические методы для получения дополнительной информации о ключе, а аналитические методы предполагают математическое изучение алгоритма шифрования. Каждый новый метод криптоанализа добавляет новые требования к алгоритмам шифрования. Так, частотный метод, в котором по распределению символов в шифртексте выдвигаются гипотезы о ключе шифрования, породил требование равномерного распределения символов в шифртексте. С ростом сложности алгоритмов постепенно стал доминировать математический подход. Такая тенденция проявилась особенно отчетливо во время Второй Мировой Войны, когда взлом шифров потребовал применения нетривиальных математических выкладок.

Методы криптоанализа симметричных криптосистем

Методы криптоанализа блочных шифров

Блочный шифр представляет собой отображение векторных пространств над полем из двух элементов вида $F : \mathbf{Z}_2^m \times \mathbf{Z}_2^n \to \mathbf{Z}_2^m$, где ключ $k \in \mathbf{Z}_2^n$, а блоки открытого и зашифрованного текста $X_i, Y_i \in \mathbf{Z}_2^m$. Идея, лежащая в основе большинства итерационных блочных шифров, состоит в построении криптографически стойкой системы путем последовательного применения относительно простых криптографических преобразований. Принцип многоразового шифрования с помощью простых криптографических преобразований был впервые предложен Шенноном в работе [102]: он использовал с этой целью преобразования перестановки и подстановки. Первое из этих преобразований переставляет отдельные символы преобразуемого информационного блока, а второе – заменяет каждый символ (или группу символов) из преобразуемого информационного блока другим символом из того же алфавита (соответственно группой символов того же размера и из того же алфавита). Узлы, реализующие эти преобразования, называются, соответственно, ***P-блоками*** (*P-box, permutation box*) и ***S-блоками*** (*S-box, substitution box*).

Наибольший прогресс в разработке методов раскрытия блочных шифров был достигнут в самом конце XX века и в основном связан с появлением в начале 90-х годов двух методов – метода разностного криптоанализа и метода линейного криптоанализа.

Статистический метод

Задачей статистического метода криптоанализа является разработка алгоритмов определения неизвестного ключа (или части ключа) $k \in \mathbf{Z}_2^n$. Рассмотрим базовые принципы и понятия статистического метода для блочных шифров; более развернутые описания представлены в работах [92] и [40], использованных в данной работе.

Реализации статистического метода криптоанализа для ряда блочных шифров позволяют получать оценки эффективности алгоритмов определения секретного ключа лучше, чем оценки метода полного перебора ключей. Входом алгоритма является некоторое число пар (X_i, Y_i), $i = \overline{1, N}$ открытого и шифрованного текста, полученных в результате применения отображения F с ключом k. Такие пары будем называть **материалом** [92] и обозначим буквой M. Объем материала соответствует числу пар (X_i, Y_i): $|M| = N$. Предполагается, что открытые тексты X_i, $i = \overline{1, N}$ выбраны случайно, равновероятно и независимо из всего пространства $\mathbf{Z}_2^m$.

Важнейшей частью статистических методов анализа являются **процедуры статистической классификации** (ПСК) [40], предназначенные для поиска неизвестного параметра по доступным случайным наблюдениям. Функции распределения вероятностей для наблюдений зависят от этого параметра. Идея ПСК заключается в том, что, если эти распределения вероятностей различны, то при достаточно большом числе наблюдений можно с определенной долей уверенности определить закон распределения наблюдений, а, значит, и искомый параметр.

Доступными случайными наблюдениями в нашем случае является материал M, а неизвестным параметром - часть ключа или некоторые линейные комбинации битов ключа. Обозначим множество, в котором принимает значения неизвестный параметр, через $\mathbf{\Gamma}$, $|\mathbf{\Gamma}| = s \le 2^n$.

Каждая ПСК определяет разбиение всего пространства наблюдений на $T > 1$, непересекающихся областей: $M = \bigcup_{i=1}^{T} M_i$, $M_i \cap M_j = 0$ при $i \ne j$, $i, j = \overline{1, T}$. Области M_i, $i = \overline{1, T}$, называют **областями принятия решений** [92], причем для заданного наблюдения $m \in M$ сложность алгоритма определения номера $i(m)$, такого, что $m \in M_{i(m)}$, считается малой. Для каждой области M_i ПСК также определяет упорядоченный список объема s', $1 \le s' \le s$ элементов множества $\mathbf{\Gamma}$: $\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,s'}$, при этом $\gamma_{i,j_1} \ne \gamma_{i,j_2}$ при $j_1 \ne j_2$.

Для определения неизвестного параметра из $\mathbf{\Gamma}$ выполняются следующие действия. Сначала по полученному наблюдению $m \in M$ нужно определить номер области принятия решений $i(m)$. Затем последовательно перебирают параметры из $\mathbf{\Gamma}$: $\gamma_{i(m),1}, \gamma_{i(m),2}, \ldots, \gamma_{i(m),s'}$ и проверяют, является ли значение j –го параметра, $j = \overline{1, s'}$ искомым или нет. Алгоритм проверки включает два этапа:

1. доопределение оставшейся части ключа;
2. проверка, правильно ли определен весь ключ. Первый шаг отсутствует, если в качестве неизвестного параметра выступает весь ключ.

В этом случае $\mathbf{\Gamma} = \mathbf{Z}_2^n$. Как правило, для доопределения оставшейся части ключа используется полный перебор все оставшихся неизвестными битов ключа. Если параметром является часть ключа или невырожденная линейная комбинация битов ключа $\mathbf{\Gamma} = \mathbf{Z}_2^{n^*}$, $1 \le n^* \le n$, потребуется перебрать 2^{n-n^*} вариантов.

Проверка того, правильно ли доопределен весь ключ, осуществляется следующим образом: для пар $(X_i, Y_i) \in M$, $X_i, Y_i \in \mathbf{Z}_2^m, i = \overline{1, N}$ открытого и шифрованного текста из

доступного материала M проверяют, выполнены или нет равенства: $F(X_i, k^*) = Y$, где $k^* \in \mathbf{Z}_2^n$ опробуемый вариант всего ключа. Ложный ключ, как правило, отсеивается уже на первых шагах проверки. На самом деле проверку достаточно осуществить для d первых пар открытого и зашифрованного текста, где число d таково, что для любого набора из d различных открытых текстов X_i, $i = \overline{1,d}$ и для любых двух различных ключей $k_1 \neq k_2 \in \mathbf{Z}_2^n$ найдется такой номер $j \in \{1, \ldots, d\}$, что $F(X_i, k_1) \neq F(X_i, k_2)$ (минимальное d_0, удовлетворяющее этому условию, называют **расстоянием единственности шифра** *F*).

Алгоритмы определения ключа сравнивают по трем параметрам: N-объем используемого материала, Q_0 -средняя трудоемкость работы алгоритма и π_0 - надежность алгоритма. Q_0 и π_0 зависят от того, какие открытые тексты были случайно выбраны, и от искомого ключа. Трудоемкость соответствует математическому ожиданию числа шагов алгоритма при случайном выборе открытых текстов и случайном, равновероятном и независимом от выбора открытых текстов выборе ключа. Надежность алгоритма равна математическому ожиданию вероятности того, что процедура выдаст правильный результат в предположении, что ключ выбран в пространстве $\mathbf{Z}_2^n$ случайно, равновероятно и независимо от выбора открытых текстов. Между параметрами Q_0 и π_0 существует прямая зависимость: чем выше надежность, тем больше трудоемкость, и наоборот.

Метод разностного (дифференциального) анализа

При написании этого раздела использовались материалы семинара А.Е. Жукова по спектральным характеристикам булевых функций.

Метод разностного анализа сочетает в себе обобщение идеи общей линейной структуры с применением вероятностно-статистических методов исследования. Этот метод относится к атакам по выбранному открытому тексту. Попытки применить разностный анализ к известному открытому тексту в большинстве случаев приводили к резкому увеличению требуемого материала. Метод был разработанный в 1990 году израильскими математиками Э. Бихамом и А. Шамиром [11]. Д. Копперсмит утверждает [16], что этот метод был известен команде разработчиков DES алгоритма еще в начале 70-х годов, но был засекречен. Идея, близкая к методу дифференциального анализа, была опубликована до работы Э. Бихама и А. Шамира в 1990 году С. Мерфи [47].

Пусть некоторый блочный шифратор с длиной блока m задается отображением $F : \Xi \times (K_1 \times \cdots K_2) \to Y$, где $F = F_R \circ F_{R-1} \circ \cdots \circ F_2 \circ F_1$. При этом $k_i \in K_i$ получаются по некоторой схеме из общего ключа k или выбирается независимо и равновероятно для каждого цикла. Пространство открытых текстов Ξ снабжено групповой операцией $\otimes$, и для каждого $X \in \Xi$ в Ξ существует элемент $X^{-1} \in \Xi$, обратный к X относительно операции $\otimes$. Выходной информационный блок $(i-1)$-го цикла является входным блоком i-го цикла, т.е. $X(i) = Y(i-1)$, для $i = \overline{2,R}$; открытый текст $X = X(1)$, зашифрованный текст $Y = Y(R)$.

Пусть одноцикловое преобразование F_j-криптографически слабое. Сделанное предположение вполне допустимо (идея Шеннона о суперпозиции простых шифров для получения сложного шифра [102]). Отметим, что под **слабым криптографическим преобразованием** $F : \Xi \times K \to Y$ мы будем понимать такое криптографическое , =(преобразование $F(X, k) = Y$, для которого по известным величинам $Y = F(X, k)$, $Y^* = F(X^*, k)$ и $\Delta X = X \otimes (X^*)^{-1}$ можно, не зная X и X^*, определить множество K', $|K'| << |K|$, такое, что $k \in K'$.

Пусть X и X^*- открытые тексты. Два открытых текста определяют последовательность разностей $\Delta X(0), \Delta X(1), \ldots, \Delta X(R)$, где $\Delta X(0) = \Delta X = X \otimes (X^*)^{-1}$; $\Delta X(i) = X(i+1) \otimes (X^*(i+1))^{-1}$, $i = \overline{1, R-1}$; $\Delta X(R) = Y \otimes (Y^*)^{-1}$. Тогда для любого $1 \le i \le R$

и любой пары (α, β) можно определить вероятность $P_{\alpha\beta}^{(i)} = \mathbf{P}\{\Delta X(i) = \beta \mid \Delta X(0) = \alpha\}$ условии, что вход X и все одноцикловые ключи k_i выбраны случайно, независимо и равновероятно. Пара (α, β) возможных значений вектора $(\Delta X(0), \Delta X(i))$ называется **дифференциалом i-го цикла** [84].

Выберем пару (α, β), для которой величина $P_{\alpha\beta}^{(R-1)}$ принимает максимальное значение, и пару (X, X^*), такую, что $\Delta X = \alpha$. Для одноциклового шифра F_R, полагая $\Delta X(R-1) = \beta$ и зная истинные значения $Y = F(X, k)$, $Y^* = F(X^*, k)$, определим множество вероятных одноцикловых ключей K'. Если теперь эту процедуру провести для различных пар (X, X^*), удовлетворяющих условию $\Delta X = \alpha$, то ключи, наиболее часто встречающиеся в множествах K', можно считать кандидатами в истинный ключ R-го цикла шифрования. Ключ всей системы находим с помощью перебора оставшихся неизвестными разрядов ключа системы или с использованием особенностей процедуры выработки цикловых ключей из ключа всей системы.

Для того чтобы описанная процедура приводила к корректным результатам, необходимо, чтобы для данной системы шифрования выполнялась

Гипотеза о статистической эквивалентности:

$$\mathbf{P}_{\alpha\beta}^{(R-1)}\{\Delta X(R-1) = \beta \mid \Delta X(0) = \alpha\} \approx \Pr\{\Delta X(R-1) = \beta \mid \Delta X(0) = \alpha, k_1 = \omega_1, \ldots, k_{R-1} = \omega_{R-1}\}$$

для почти всех значений частей ключа, используемых в циклах шифрования $(\omega_1, \ldots, \omega_{R-1})$, где $\Pr\{\theta\}$ обозначает вероятность события θ.

Возможность эффективного применения метода разностного анализа существенно зависит от выбора групповой операции, относительно которой определяются разности Δ. Чаще всего в качестве таковой выбирается операция сложения булевых векторов. Однако в отдельных случаях неудачный выбор операции может приводить к нарушению гипотезы о статистической эквивалентности, в результате чего становится невозможным вычисление вероятностей.

Эффективность метода разностного анализа существенно зависит от выбора характеристики, с помощью которой он проводится. Свойства подстановки P на разностный анализ систем типа DES не влияют. В то же время, даже изменение порядковой нумерации S-блоков (без изменения их строения) может сильно ослабить DES. При неудачном подборе этой нумерации DES-16 раскрывается за 2^{46} опробований. Стойкость системы DES к методу разностного анализа может также уменьшаться при замене операции векторного сложения на другие арифметические операции, при замене S-блоков на случайно выбранные и даже при внесении минимальных изменений в один S-блок.

Почти сразу после появления первых работ по разностному криптоанализу начались поиски условий, при которых та или иная криптографическая система остается устойчивой по отношению к этому методу. Так как разностный анализ основан на использовании неравновероятности в распределении значений разности двух шифртекстов полученных из пары открытых текстов, имеющих некоторую фиксированную разность, то очевидно, что если все возможные значения разностей двух шифртекстов будут появляться с близкими (в идеале – с равными) вероятностями, то метод разностного анализа не сможет работать.

Как показано в [92], метод разностного анализа развивался в следующих направлениях:

1. Вскоре после изобретения метода разностного анализа было предложено использовать разностные характеристики для случая, когда операцию покоординатного суммирования векторов из $\mathbf{Z}_2^{32}$ (для DES-алгоритма) по модулю 2 заменяют на операцию суммирования этих векторов по модулю 2^{32}, рассматривая их как 2-адическую запись целых чисел [9].
2. В 1993 году израильские математики Бен-Аройа и Бихам [8] предложили искать разностные характеристики, считая, что ключ принимает не все возможные значения, а лишь значения из некоторого подмножества. Этот метод получил название «*метод условных дифференциалов*».

3. В 1994 году датский математик Ларс Кнудсен [37] предложил строить по аналогии с обычным разностным методом криптоанализа метод усеченных дифференциалов. Идея Кнудсена заключается в том, чтобы "следить" в разностной характеристике лишь за частью бит векторов, участвующих в соотношении.
4. В 1994 году швейцарский криптограф Лаи [37, 39] показал, что для построения метода криптографического анализа вместо пар можно использовать разностные характеристики высших порядков: $\underset{a\in L}{\oplus} F(x\oplus a)=d$, где L L -пространство в $\mathbf{Z}_2^n$, $\dim L\geq 2$.
5. В 1998 году Бихам, Бирюков и Шамир [10] заметили, что для построения метода криптографического анализа можно использовать дифференциалы, имеющие не повышенную вероятность появления, а пониженную, еще лучше -нулевую (*невозможные дифференциалы*). Именно этим методом была обнаружена слабость в криптографическом алгоритме Skipjack [65] -первом и пока единственным алгоритме, авторство которого официально признано Агентством Национальной Безопасности США.
6. Для шифров итерационного типа редко удается построить разностную характеристику, имеющую гарантированно большую вероятность. В 1999 году

Вагнер [69] предложил использовать не пары, а четверки открытых текстов с заданным набором разностей. Для построения таких четверок Вагнер разработал специальный метод, названный **«методом прямоугольника»**. Развитие этого метода получило название **«метода бумеранга»**.

Метод линейного анализа

Подобно разностному анализу, линейный криптоанализ является комбинированным методом, сочетающим в себе поиск линейных стат. аналогов для уравнений шифрования, статистический анализ имеющихся открытых и шифрованных текстов, использующий также методы согласования и перебора. Этот метод исследует статистические линейные соотношения между отдельными координатами векторов открытого текста, соответствующего шифртекста и ключа и использует эти соотношения для определения статистическими методами отдельных координат ключевого вектора.

На сегодняшний день метод линейного криптоанализа позволил получить наиболее сильные результаты по раскрытию ряда итерационных систем блочного шифрования, в том числе и системы DES. В отличие от метода разностного анализа, метод линейного криптоанализа в неявном виде появился еще в работе [47], где он успешно применялся при анализе системы блочного шифрования FEAL [63]. В работе же [43] этот подход был впервые четко формализован, а затем успешно применен к анализу системы DES.

Пусть имеется блочный шифр $F:\mathbf{Z}_2^n\times\mathbf{Z}_2^k\to\mathbf{Z}_2^n$, отображающий при фиксированном ключе $k\in\mathbf{Z}_2^k$ вектор открытого текста $p\in\mathbf{Z}_2^n$ в вектор шифртекста $c\in\mathbf{Z}_2^n$. Предполагается, что открытый текст выбирается случайно и равновероятно, а подключи в каждом раунде независимы друг от друга. Сложность атаки связана с количеством необходимых известных открытых текстов, т.к. для любой пары (открытый текст, шифртекст) требуется небольшое количество вычислений для реализации алгоритма.

Эта атака может быть проведена, если имеется только шифртекст. Мацуи были получены следующие результаты:

- если известно, что открытый текст на английском языке, представленный в ASCII кодах, то при 8-раундовый DES вскрывается при 2^{29} известных шифртекстах;
- если открытый текст является случайным ASCII кодом, то 8-раундовый DES вскрывается при 2^{37} известных шифртекстах.

Метод линейного криптоанализа является частным случаем общего статистического метода и основан на использовании выражений вида

$$\langle Y(t_1),a\rangle\oplus\langle Y(t_2),b\rangle=\langle k,c\rangle, \qquad (3)$$

где $a, b \in \mathbf{Z}_2^m$, $c \in \mathbf{Z}_2^n$, $0 \le t_1 < t_2 \le R$ (при этом пару векторов (a, b) называют линейной характеристикой), $\langle g, h \rangle = \sum_i g_i \cdot h_i$ - скалярное произведение векторов g и h одинаковой размерности. В методе линейного криптоанализа используют только те выражения типа (3), которые выполняются с вероятностью, отличной от $\frac{1}{2}$. Вероятность, с которой выполняется равенство (3), удобно записывать в виде

$$P\{\langle Y(t_1), a \rangle \oplus \langle Y(t_2), b \rangle = \langle k, c \rangle\} = \tfrac{1}{2} + \varepsilon, \qquad (4)$$

где $-\frac{1}{2} \le \varepsilon \le \frac{1}{2}$. Число ε называют **преобладанием** (deviation, bias).

Рассмотрим простейший (но не самый эффективный) вариант метода линейного криптоанализа - алгоритм определения одного бита ключа, называемый также **«алгоритм 1 Мацуи»** [92]. Этот алгоритм использует выражение типа (3) при $t_1 = 0$ и $t_2 = R$, т.е. подставляются непосредственно открытый и зашифрованный тексты:

$$\langle X_i\ a \rangle \oplus \langle Y_i, b \rangle = \langle k, c \rangle,\ i = \overline{1, N},$$

где (X_i, Y_i) -пары открытого и зашифрованного текста, известные криптоаналитику. Метод линейного криптоанализа в данном варианте определяет линейную комбинацию $\langle K, c \rangle$ битов ключа. Подсчитывается мощность множества:

$$N_0 = \left| i \mid \langle X_i\ a \rangle \oplus \langle Y_i, b \rangle = 0, i = \overline{1, N} \right|.$$

Процедура статистической классификации разбивает все пространство наблюдений M на две области M_0 и M_1 ($M = M_0 \cup M_1$): к области M_0 относят все те наблюдения, для которых $N_0 \ge \frac{N}{2}$, а к области M_1 - те наблюдения, для которых $N_0 < \frac{N}{2}$.

Алгоритм использует принцип максимума правдоподобия (этот алгоритм называют **«алгоритмом 1 Мацуи»**): если $N_0 \ge \frac{N}{2}$ и в равенстве (4) $\varepsilon > 0$ или $N_0 < \frac{N}{2}$ и в равенстве (4) $\varepsilon < 0$, то алгоритм выдает значение $\langle k, c \rangle = 0$, в противном случае $\langle k, c \rangle = 1$. Остальные биты ключа определяют методом перебора.

Улучшенный «алгоритм 2 Мацуи» [92] представляет собой естественное обобщение алгоритма 1 и заключается в использовании выражений вида (3) при $t_2 > R$ и/или $t_1 > 0$.

Таким образом, метод линейного криптоанализа сводится к построению таких векторов $a, b \in \mathbf{Z}_2^m$, $c \in \mathbf{Z}_2^n$ чтобы модуль ε в выражении (4) был как можно больше. При этом надежность и трудоемкость метода определяются этой величиной $|\varepsilon|$, т.е. для построенных векторов $a, b \in \mathbf{Z}_2^m$, $c \in \mathbf{Z}_2^n$ надо уметь оценивать $|\varepsilon|$ в выражении (4).

Эту задачу метод линейного криптоанализа решает, используя итерационную структуру блоковых шифров. Первым шагом при этом является анализ в каждом раунде алгоритма шифрования нелинейных элементов. Для каждого нелинейного элемента вычисляют для всех $a' \in \mathbf{Z}_2^k$, $b' \in \mathbf{Z}_2^l$ вероятности $P\{\langle X', a' \rangle = \langle S(X'), b' \rangle\} = \frac{1}{2} + \varepsilon'$ при случайном равновероятном выборе X' из $\mathbf{Z}_2^n$. Для получения оценки на преобладание для всего алгоритма рассматривают последовательности «согласованных» линейных соотношений для соседних раундов (т.е. соотношения, когда выходная линейная комбинация предыдущего раунда совпадает с входной линейной комбинацией последующего раунда).

За годы развития линейного криптоанализа был предложен ряд его обобщений. В работе [32] предложена идея использовать несколько линейных соотношений (этот метод называют методом кратного приближения), но конкретной конструкции, как это реализовать, не предложено, за исключением случая, когда во все линейные соотношения входит одна и та же линейная комбинация битов ключа. И лишь недавно появилась статья [12], в которой предложен статистический расчет метода кратных приближений. В работе [38] предложено использовать нелинейные приближения вместо линейных, однако не указано, как эффективно искать такие приближения и как

рассчитывать их преобладания. В работе [35] предложено использовать линейный метод криптоанализа в сценарии атаки с выбором открытого текста. Также заслуживает внимания метод решеточного криптоанализа, предложенного Ростовцевым в статье [97]. Метод должен быть эффективен для тех шифров, в которых не каждый разряд ключа сцеплен с каждым разрядом шифртекста, например, для шифров с псевдослучайным выбором слов ключа.

Стойкость советского и американского стандартов симметричной криптографиии

DES, Triple DES и AES. В 1973-74 гг. Национальное Бюро Стандартов США (NBS) опубликовало документы, содержащие требования к криптографическому алгоритму, который мог бы быть принят в качестве стандарта шифрования данных в государственных и частных учреждениях. В 1976 г. в качестве такового стандарта был утвержден алгоритм, разработанный фирмой IBM. В 1977 г. этот стандарт был официально опубликован и вступил в силу как федеральный стандарт шифрования данных – Data Encryption Standard или сокращенно DES [4].

В самом схематичном виде DES представляет собой 16-цикловой итерационный блочный шифр. DES работает с блоками данных разрядностью 64 бита с использованием 56-разрядного ключа. Применяемые преобразования – поразрядное сложение по модулю два, подстановки и перестановки. Алгоритм выработки 48-битовых цикловых ключей из 56-битового ключа системы и ряд преобразований служат для обеспечения необходимого перемешивания и рассеивания перерабатываемой информации, однако при анализе DES чаще всего играют не самую существенную роль.

В 1999 г. на конференции, организованной RSA, компания Electronic Frontier Foundation взломала ключ DES менее чем за 24 часа. Одной из замен DES, получившей широкое распространение, стал алгоритм Triple DES. В этом случае алгоритм DES выполняется трижды, при этом используются 3 ключа, каждый из которых состоит из 56 битов (что, по сути, соответствует использованию 168-битового ключа). Тем не менее, криптоаналитики обнаружили способ, позволяющий сделать атаку прямого перебора эквивалентной атаке на 108-битовый ключ. Второй проблемой является значительное снижение скорости зашифрования и расшифрования данных.

В ответ на проблемы с длиной ключа и производительностью, проявившиеся в Triple DES, многие криптографы и компании разработали новые блочные шифры. Наиболее популярными предложениями стали алгоритмы RC2 и RC5 корпорации RSA Data Security, IDEA компании Ascom, Cast компании Entrust, Safer компании Cylink и Blowfish компании Counterpane Systems. Коммерческие альтернативы DES получили определенное распространение, но ни одна из них не стала стандартом.

В 2001 г. на смену DES и Triple DES пришел стандарт AES (Advanced Encryption Standard), действующий и по сей день. Шифр AES основан на алгоритме Rijndael [52], разработанном бельгийцами Д. Дейменом и В. Райменом. Он быстрый, простой, защищенный, универсальный и хорошо подходит для реализации на смарт-картах. Rijndael – это итерационный блочный шифр, имеющий архитектуру «Квадрат». Шифр имеет переменную длин у блоков и различные длины ключей. Длина ключа и длина блока могут быть равны независимо друг от друга 128, 192 или 256 битам. В стандарте AES определена длина блока, равная 128 битам.

ГОСТ 28147-89 [83]. Отечественный стандарт шифрования носит официальное название «Алгоритм криптографического преобразования ГОСТ 28147-89». Как явствует из его номера, стандарт был принят в СССР в 1989 г. Если охарактеризовать алгоритм ГОСТ в самом общем виде, то он является блочным шифром, построенным по схеме Фейстеля с 32 циклами шифрования. Длина информационного блока – 64 бита, длина ключа – 256 бит.

Основные отличия алгоритма ГОСТ от алгоритма DES – в строении функции, которая осуществляет отображение $\mathbf{Z}_2^{32} \times \mathbf{Z}_2^{48} \to \mathbf{Z}_2^{32}$, и алгоритме выработки цикловых ключей. И в том и в другом случае преобразования, используемые в алгоритме ГОСТ, проще для программной реализации. В статье [81] рассматривается устойчивость алгоритмов ГОСТ и AES к известным видам криптоанализа, в особенности к линейному и разностному методу. По оценкам разработчиков шифра Rijndael, уже на четырех раундах шифрования этот алгоритм приобретает достаточную устойчивость к указанным видам криптоанализа. Теоретической границей, за которой линейный и дифференциальный виды криптоанализа теряют смысл, является рубеж в 6-8 раундов в зависимости от размера блока. Согласно спецификации, в шифре предусмотрено 10-14 раундов.

Следовательно, шифр Rijndael устойчив к указанным видам криптоанализа с определенным запасом. Дать оценку устойчивости алгоритма ГОСТ 28147-89 к конкретным видам криптоанализа невозможно без спецификации узлов замен, так как качество этого шифра существенным образом зависит от качества использованных узлов. Однако исследования близких по архитектуре шифров с заданными таблицами подстановок (DES) показали, что криптоанализ шифра с 16 раундами в принципе осуществим, однако требует очень большого числа исходных данных, а при 20-24 раундах становится теоретически бесполезным. ГОСТ предусматривает 32 раунда шифрования, и этого количества хватает с запасом, чтобы успешно противостоять указанным видам криптоанализа.

Исследования [81] показывают, что российский стандарт не уступает по стойкости американскому AES. Делается вывод, что оба сравниваемых шифра обладают достаточной стойкостью к известным видам криптоанализа. Молдовян, напротив, утверждает [93], что ГОСТ является устаревшим алгоритмом шифрования из-за его подверженности атакам на основе аппаратных ошибок. К стандарту, который придет на смену ГОСТ 28147-89, автор предъявляет следующие требования:

- высокое быстродействие при аппаратной и программной реализации,
- стойкость ко всем известным видам криптоанализа, включая атаку на основе формирования аппаратных ошибок,
- невысокая стоимость аппаратной реализации,
- полная открытость алгоритма.

Заключение

Чтобы снизить вероятность непредсказуемого "обвала" вновь разработанного криптоалгоритма, необходимо заблаговременное проведение криптографических исследований. Разработка любого шифра предусматривает оценку его стойкости к достаточно разнообразным типам криптоаналитических нападений. Как относиться к заявляемым оценкам стойкости с учетом того, что их получение обычно является довольно сложной задачей? Это зависит от того, кто дает оценку [93]. Стойкостьшифра рассматривается как разработчиком, так и критиком (криптоаналитиком). Оценки разработчика шифра можно считать корректными, если он делает некоторые допущения в пользу криптоаналитика. Оценки разработчика будут опровергнуты, если кто-либо укажет другой способ криптоанализа, для которого вычислительная сложность получается меньше заявляемой.

Оценки критика являются корректными, если он не занижает значение стойкости по предлагаемому им лучшему методу криптоанализа. Оценки критика будут опровергнуты, если кто-либо найдет и укажет принятые криптоаналитиком существенные допущения, учет которых приводит к значительному увеличению сложности предлагаемого криптоаналитического нападения. Таким образом, если криптоаналитик предлагает корректный вариант атаки, который вычислительно реализуем по оценкам, то практическая проверка должна быть положительной.

В обоих случаях риск того, что оценки будут скомпрометированы, тем меньше, чем больше специалистов анализировали алгоритм, чем выше их квалификация и чем больше времени они уделили анализу. Поэтому открытая публикация криптоалгоритмов, их исследование и публичное обсуждение являются необходимыми.

Для уменьшения возможного ущерба, вызванного несвоевременной заменой криптоалгоритма, потерявшего свою стойкость, желательна периодическая перепроверка стойкости криптоалгоритма. То обстоятельство, что любую задачу отыскания способа раскрытия некоторой конкретной криптосистемы можно переформулировать как привлекательную математическую задачу, при решении которой удается использовать многие методы той же теории сложности, теории чисел и алгебры, привело к раскрытию многих криптосистем. С развитием математики и средств вычислительной техники стойкость криптоалгоритма может только уменьшаться. Если влияние роста мощности компьютеров на стойкость алгоритмов еще можно предсказать с той или иной степенью точности (до настоящего момента каждое десятилетие скорость вычислений вырастала на порядок), то оценить перспективы научного прогресса не под силу даже ученым-криптографам с мировым именем. Так, в 1977 году Рон Ривест заявил, что разложение на множители 125-разрядного числа потребует 40 квадриллионов лет [24]. Однако уже в 1994 г. было факторизовано число, состоящее из 129 двоичных разрядов! Как видно, предсказания – дело неблагодарное, поэтому в данной статье авторы ограничились изложением фактов, касающихся современного

состояния и тенденций развития криптоанализа. Хочется надеяться, что этот обзор позволил заинтересованному читателю получить общее представление о теме; более подробная информация доступна в источниках, использованных при написании данной статьи.

Литература

1. Криптографическая защита информации в АСУ СН. Курс лекций. В.И. Долгов. ХВУ. 1998.

2. Криптографическая защита информации в информационных системах. Курс лекций. И.Д. Горбенко. ХНУРЭ. 2002.

3. Теория информации. Курс лекций. В.И. Долгов. ХВУ. 1998.

4. Брюс Шнайер. Прикладная криптография. 2-ое издание. Протоколы, алгоритмы и исходные тексты на языке С. Доступно: http://nrjetix.com/r-and-d/lectures

2. Agnew G.B. Random Sources for Cryptographic Systems // Advances in Cryptology – EUROCRYPT'87 Proceedings, Springler-Verlag, 1988. P. 77-81.

5. AT&T. T7001 Random Number Generator // Data Sheet, Aug 1986.

8. Ben-Aroya I., Biham E. Differential Cryptanalysis of Lucifer // CRYPTO'93, Springer.

9. Berson T.A. Differential Cryptanalysis Mod with Applications to MD5, EUROCRYPT'92, Lecture Notes in Computer Science, v. 658, Springer, 1992, pp. 71-80.

10. Biham E., Biryukov A., Shamir A. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials // EUROCRYPT'99, 1999, pp.12-23.

11. Biham E., Shamir A. Differential Cryptanalysis of DES-like cryptosystems // Journal of Cryptology. 1991. V.4. No. 1. P. 3-72.

12. Biryukov A., De Canniere C., Quisquarter M. On Multiple Linear Approximations // 2004. Available via http://www.iacr.eprint.org.

16. Coppersmith D. The data encryption standard (DES) and its strength against attacks // Technical Report RC 18613 (81421), IBM Research Division, December, 1992.

19. Diffie W., Hellman M. New directions in cryptography // IEEE Trans. Inform. Theory, Vol 22, 6 (1976). P. 644–654.

21. Fairfield R.C., Mrotenson R.L., Koulthart K.B. An LSI Random Number Generator (RNG) // Advances in Cryptology: Proceedings of CRYPTO 84, Springler-Verlag, 1985.

22. Fiat A., Shamir A. How to prove yourself: practical solutions to identification and signature problems // Proc. Crypto'86, Lect. Notes in Comput. Sci. V. 263, 1987. P. 186-194.

24. Gardner M. A New Kind of Cipher That Would Take Millions of Years to Break // Scientific American, v.237, n.8, Aug 1977. P. 120 – 124.

27. Gordon L.A., Loeb M.P., Lucyshyn W., Richardson R. CSI/FBI Computer Crime and Security Survey 2005 // Computer Security Institute Publications, 2005.

28. Gude M. Concept for a high-Performance Random Number Generator Based on Physical Random Phenomena // Frequenz, v. 39, 1985. P. 187-190.

31. Kahn D. The Codebreakers: The Story of Secret Writing. MacMillan, New York, 1967.

32. Kaliski B.S., Robshaw M.J.B. Linear cryptanalysis using multiple approxiamtions // In Proceedings of Advances of Cryptology - CRYPTO`94, (Desmedt Y. Ed.), Lect. Notes in Comp. Sci. Springler-Verlag, 1994. V. 950. P. 26-39.

33. Kelsey J., Schneier B., Wagner D., Hall C. Cryptanalytic Attacks on Pseudorandom Number Generators // Fast Software Encryption, Fifth International Workshop Proceedings (March 1998), Springer-Verlag, 1998, pp. 168-188. Available via: http://www.counterpane.com/pseudorandom_number.html

34. Kerckhoffs A. La cryptographie militaire // Journal des sciences militaires, vol. IX. P. 5-38, Jan. 1883, (P. 161-191, Feb. 1883).

35. Knudsen L.R., Mathiassen J.E. A chosen-plaintext linear attack on DES // In Proceedings of Fast Software Encryption - FSE'2000, (Schneier B. ed.), Lect. Notes in Comp. Sci. Sprinler-Verlag, 2001. V. 1978. P. 262-272

36. Knudsen L.R. Block Ciphers – Analysis, Design, Applications // Ph.D. dissertation, Aarhus Unversity, Nov 1994.

37. Knudsen L.R. Truncated and Higher Order Differentials // Fast Software Encryption, Second International Workshop, Belgium, December, 1994, Springer. P. 196-211.

38. Knudsen L.R., Robshaw M.J.B. Non-Linear Approximation in Linear Cryptanalysis // In Proceedings of Advances of Cryptology - EUROCRYPT'96, (Maurer U. ed.), Lect. Notes in Comp. Sci. Springer-Verlag, 1996. V. 1070. P. 224-236.

39. Lai X. Higher Order Derivatives and Differential Cryptanalysis // Communications and Cryptography, Kluwer Academic Publishers, 1994. P. 227-233.

40. Lehmann E.L. Testing statistical hypotheses. John Wiley, 1959. (Русский перевод: Леман Э. Проверка статистических гипотез. М.: Наука, 1979.)

43. Matsui M., Yamagishi A. A new method for known plaintext attack of FEAL cipher // In Proceedings of Advances in Cryptology - EUROCRYPT'92. Lect. Notes in Comp. Sci. Berlin: Springler-Verlag, 1992. V. 658. P. 1-91.

47. Murphy S. The cryptanalysis of FEAL-4 with 20 chosen plaintexts // Journal of Cryptology, 1990, v. 3, No. 2. P. 145-154.

51. Quisquarter J.-J., Desmedt Y.G. Chinese Lotto as an Exhaustive Code-Breaking Machine // Computer, v.24, n.11, Nov 1991. P. 14-22.

52. RIJNDAEL description. Submission to NIST by Joan Daemen, Vincent Rijmen // Available via http://csrc.nist.gov/encryption/aes/round1/docs.htm

53. Rivest R.L., Shamir A., Adleman L.M. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems // Communications of the ACM, v.21, n.2, Feb 1978. P. 120-126.

54. Rivest R.L., Shamir A., Adleman L.M. On Digital Signatures and Public Key Cryptosystems // MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan1979.

57. Schneier B. A self-study course in block-cipher cryptanalysis // Cryptologia, v.24, n.1, Jan 2000. P. 18-34.

58. Schneier B. Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons Inc., 1996. (Русский перевод: Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002.)

60. Schneier B. Snake Oil, Crypto-Gram // February, 1999. Available via http://www.counterpane.com/Crypto-Gram.html

63. Shimizu R., Miyaguchi S. Fast Data Encipherment Algorithm FEAL // Transactions of IEICE of Japan, v. J70-D, n. 7, Jul 87. P. 1413-1423.

65. SKIPJACK and KEA Algorithm Specifications // Version 2.0, 1998. Available via: http://csrc.nist.gov/CryptoToolkit/skipjack/skipjack-kea.htm

69. Wagner D. The Boomerang Attack // Proceedings of Fast Software Encryption'99, Springer Verlag, Lecture Notes in Computer Science, v. 1636, 1999. P. 156-170.

72. White S.R. Covert Distributed Processing with Computer Viruses. // Advances in Cryptology - CRYPTO'89 Proceedings, Springler-Verlag, 1990. P. 616-619.

75. Алексеев А. Криптография и криптоанализ: вековая проблема человечества. // Опубликовано: http://infocity.kiev.ua/hack/content/hack008.phtml

77. Баричев С. Основной вопрос криптографии // Chief Inforamtion Officer – руководитель информационной службы. #5 (37), 2005, с. 93-95.

78. Варновский Н.П. О стойкости схем электронной подписи с аппаратной поддержкой. Технический отчет. Лаборатория МГУ по математическим проблемам криптографии, 1997.

81. Винокуров А., Применко Э. Сравнение российского стандарта шифрования, алгоритма ГОСТ 28147-89, и алгоритма Rijndael, выбранного в качестве нового стандарта шифрования США // «Системы безопасности», М., изд. «Гротэк», 2001, №№1,2.

83. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.

86. Загнетко А. Информация доступная и недоступная. // Опубликовано: 19.06.2006: http://pda.cio-world.ru/?action=article&id=273907

87. Зубов А.Ю. Криптографические методы защиты информации. Совершенные шифры: Учебное пособие. М.: Гелиос АРВ, 2005.

88. Иванов М.А. криптографические методы защиты информации в компьютерных системах и сетях // М.: КУДИЦ-ОБРАЗ, 2001.

92. Логачев О.А., Сальников А.А., Ященко В.В. Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.

93. Молдовян Н. Каким быть новому стандарту шифрования? //"Компьютерра" №2 от 18.01.2000.

97. Ростовцев А.Г. Решеточный криптоанализ // Безопасность информационных технологий. -М.: Изд. МИФИ, 1997, №3. -с. 53-55.

99. Ростовцев А.Г., Михайлова Н.В. Методы криптоанализа классических шифров // 1998. Опубликовано: http://crypto.hotbox.ru/download/cryptoan.zip

102. Шеннон К.Э. Работы по теории информации и кибернетике // М.: И.Л., 1963.