Криптоанализ симметричных криптосистем: поточные шифры. Криптоанализ по побочным каналам

Лекция

Ревизия: 0.1

Последняя модификация: Vlad Kovtun

Дата последней модификации: 31.01.2011 21:24:00

История изменений

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

Содержание

Лекция 4. Криптоанализ симметричных криптосистем: поточные шифры. Криптоанализ по побочным каналам. Часть 2

Вопросы

1. Криптоанализ симметричных поточных шифров.
2. Криптоанализ по побочным каналам.
3. Стойкость советского и американского стандартов симметричного шифрования.

Методы криптоанализа поточных шифров

Поточные шифры преобразуют открытый текст в шифртекст шифрованный побитово. Простейшая реализация поточного шифра представлена на рис. 1. Генератор гаммы выдает поток битов: $k_1, k_2, \ldots, k_i$, называемый **ключевым потоком**, или **бегущим ключом**, или **гаммой**. Гамма шифра и поток битов открытого текста $p_1, p_2, \ldots, p_i$ подвергаются операции XOR , в результате чего создается поток битов шифртекста $c_1, c_2, \ldots, c_i$, где $c_j = p_j \oplus k_j$ (этот режим шифрования называется **гаммированием**). При расшифровании для восстановления битов открытого текста над битами шифртекста и той же самой гаммой тоже выполняется операция XOR: $p_j = c_j \oplus k_j$.

Надежность схемы всецело зависит от генератора гаммы. Если он создает бесконечную строку нулей, шифртекст, очевидно, будет совпадать с открытым текстом, и вся операция бессмысленна.

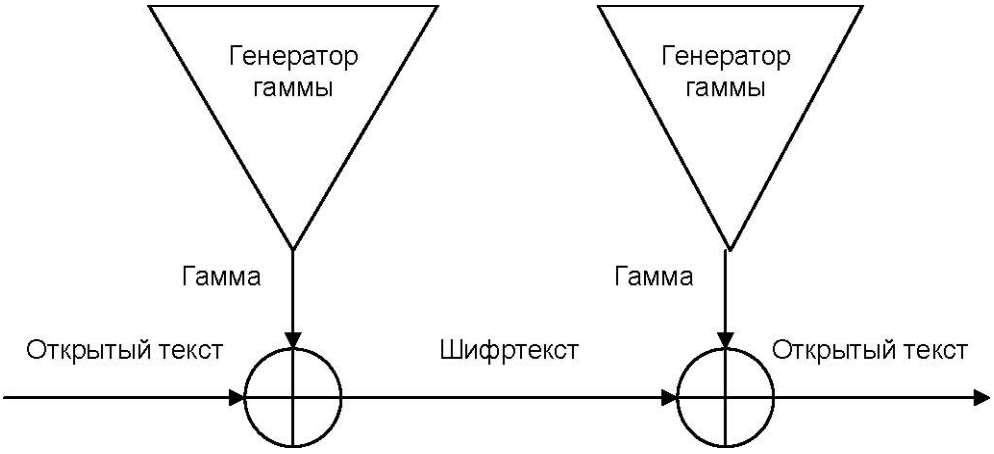

Рис. 1. Процесс зашифровывания и расшифровывания поточным шифром

Значительная часть предлагаемых поточных схем состоит из унифицированных узлов и блоков, напоминая тем самым выставку поделок из детского конструктора [96]. Основными деталями этого «криптографического конструктора» являются:

- регистры сдвига (на битах или двоичных векторах определенной размерности) с обратной связью (обычно линейной),
- дискретные (главным образом, булевы) функции усложнения,
- запоминающие устройства,
- узлы, реализующие неравномерное движение.

Регистры сдвига обеспечивают большой период и хорошие статистические свойства последовательностей, в то время как остальные детали «конструктора» используются для внесения элементов нелинейности в законы функционирования поточной схемы.

В **синхронных** поточных шифрах гаммирующая последовательность формируется независимо от потока открытого текста при зашифровывании и шифртекста при расшифровывании. **Главное свойство синхронного поточного шифра – отсутствие эффекта размножения ошибок.** Это свойство ограничивает возможность обнаружения ошибки при расшифровывании. Кроме того, противник имеет возможность

производить управляемые изменения шифртекста, совершенно точно зная, какие изменения в результате произойдут в соответствующем открытом тексте.

Самосинхронизирующиеся поточные шифры, благодаря свойству восстанавливать информацию после потери синхронизации, являются наиболее распространенным видом шифрования в дипломатических, военных и промышленных системах связи. В таких шифрах каждый бит гаммы представляет собой функцию фиксированного числа предыдущих битов шифртекста [58]. Наиболее распространенный режим функционирования самосинхронизирующихся поточных шифров – режим обратной связи по шифртексту.

Поточные шифры почти всегда работают быстрее и обычно обладают меньшей конструктивной сложностью, чем блочные шифры. Наиболее известный поточный шифр был разработан Р. Ривестом; это шифр RC4, который характеризуется переменным размером ключа и байт-ориентированными операциями. На один байт требуется от 8 до 16 действий, программная реализация шифра выполняется очень быстро. Независимые аналитики исследовали шифр, и он считается защищенным. RC4 [71] используется для шифрования файлов в таких изделиях, как RSA SecurPC. Он также применяется для защиты коммуникаций, например, для шифрования потока данных в Интернет-соединениях, использующих протокол SSL.

Хотя подавляющее большинство существующих шифров с секретным ключом с определенностью могут быть отнесены или к поточным или к блочным шифрам, теоретически граница между этими классами остается довольно размытой. Так, например, допускается использование алгоритмов блочного шифрования в режиме поточного шифрования (например, режимы CBF и OFB для алгоритма DES или режим гаммирования для алгоритма ГОСТ 28147-89 [83]).

Структура поточного ключа может иметь некоторые уязвимые места, которые позволят нападающему получить некоторую дополнительную информацию о ключе. Наиболее очевидно, если **период поточного ключа** (т.е. количество бит, после которых поточный ключ начинает повторяться) слишком мал, то нападающий может применить обнаруженные части поточного ключа для дешифрации других частей закрытого текста.

Другая опасность состоит в том, что одна или несколько внутренних выходных последовательностей – часто выходы отдельных регистры сдвига с линейной обратной связью – могут **иметь корреляцию с объединенной гаммой** и атакованы при помощи линейной алгебры. Такие атаки называют «корреляционными атаками», или атаками «разделяй и властвуй» [58]. Основная идея состоит в выявлении определенной корреляции между выходом генератора и одной из его составных частей. Используя эту информацию, можно собирать данные о других промежуточных выходах до тех пор, пока генератор не будет взломан. К поточным шифрам также применим линейный и дифференциальный криптоанализ.

В обзоре [96] отмечаются следующие особенности работ, посвященных криптографическим исследованиям поточных шифров.

Если подавляющее число работ, посвященных блочным шифрам, ориентировано на анализ и синтез DES-подобных алгоритмов, то для поточных шифров нет такого "центра притяжения". Синтезированные решения и методы "взлома" поточных шифров отличаются значительным разнообразием.

Поскольку при разработке любых схем преобразования их стойкость определяется прежде всего стойкостью к известным на текущий момент аналитическим атакам, направленных на выявление в рассматриваемой схеме слабостей различного рода, представляется целесообразным рассмотрение наиболее распространенных и упоминаемых методов (алгоритмов) анализа для разработки требований к схемам поточного преобразования информации.

При рассмотрении методов анализа схем поточного преобразования все методы можно условно разделить на три класса:

- аналитические атаки;
- статистические;
- силовые.

К аналитическим атакам относят атаки, в которых алгоритм построения атаки основан на аналитических принципах вскрытия схемы. К статистическим атакам относятся атаки, основанные на оценке статистических свойств управляющей последовательности. К силовым атакам относятся атаки, основанные на принципе

полного перебора всех возможных комбинаций ключа; теоретически, при попытке вскрытия схемы преобразования данный вид атаки должен быть наиболее эффективным по сравнению с остальными предлагаемыми видами атак.

Класс аналитических атак можно разбить на два подкласса:

- методы анализа управляющей последовательности;
- методы анализа процедуры ключевой инициализации/реинициализации.

В силу специфики принципов построения ПШ основным видом атак на данные схемы в первом подклассе являются **корреляционные атаки**, основная идея которых состоит в нахождении корреляции между управляющей последовательностью и различными линейными комбинациями ключа (регистра сдвига). В качестве объекта исследования корреляционные атаки рассматривают нелинейную функцию, вносящую нелинейность в выходную последовательность регистра сдвига – таким образом, каждый раз, в зависимости от устройства применяемой нелинейной функции, реализации корреляционных атак будут различны и будут основаны на специфическом устройстве анализируемой функции.

Класс статистических атак делится на два подкласса:

- методы анализа статистических свойств управляющей последовательности;
- методы анализа сложности последовательности.

Методы первого подкласса направлены на выявление возможного дисбаланса в выходной последовательности схемы с целью нахождения способа предположения следующего бита выходной последовательности с вероятностью лучшей, чем при случайном выборе. Данные методы оперируют различными статистическими тестами, выбор необходимого и достаточного количества тестов – прерогатива аналитика. Методы второго подкласса направлены на выявление возможности генерации последовательности, аналогичной управляющей последовательности, каким-либо другим способом, сложность реализации которого была бы меньшей по сравнению со способом генерации управляющей последовательности; в идеале, найденный способ должен быть применимым на практике. Данные методы используют концепции линейной сложности, профиля линейной сложности, квадратичного размаха.

При проведении анализа схем подразумевается, что атака произошла успешно, если ее вычислительная сложность меньше, чем вычислительная сложность полного перебора всех ключевых комбинаций данной схемы.

Рассмотрим аналитические атаки как один из основных видов атак, применяемых к ПШ. Все аналитические атаки осуществляют при условии, что аналитику известно описание генератора (образующие полиномы, вид нелинейного преобразования), он обладает открытым и соответствующим ему закрытым текстом. Атаки данного класса эквивалентны атакам по известному открытому тексту. Задачей аналитика является определение применяемого ключа (начального заполнения). На рис. 2 представлены наиболее известные аналитические атаки, применяемые к синхронным ПШ.

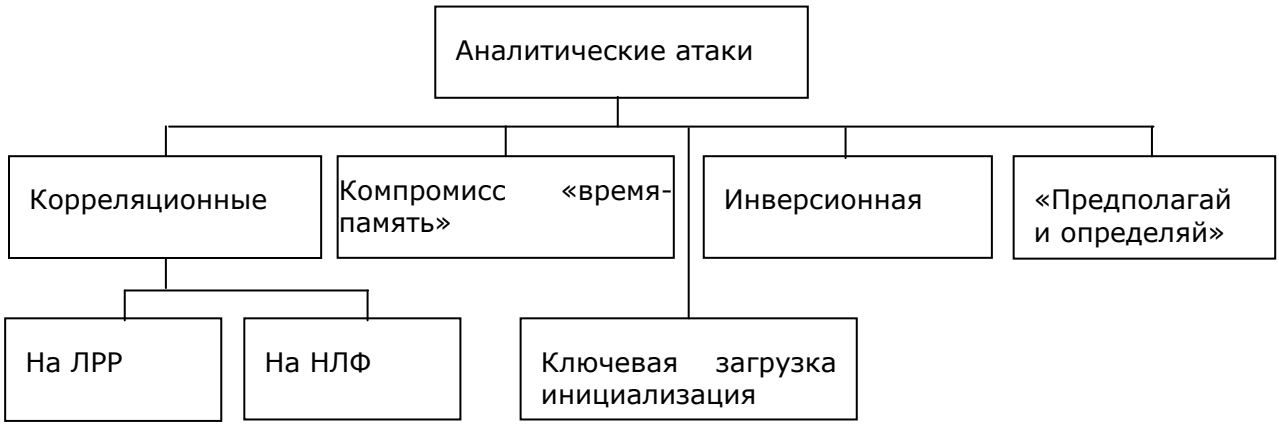

Рис. 2. Аналитические атаки, применяемые к синхронным ПШ

Если для блочных шифров нет "канонической" теории их синтеза и анализа, то для поточных шифров теория построения методов "взлома" сформировалась; также установлен набор требований, которым должны удовлетворять "хорошие" схемы. В идейном плане методы взлома поточных шифров сводятся к одному из следующих двух подходов:

- использование статистических связей (корреляционные атаки);
- линеаризация (сведение задачи поиска ключа к решению системы линейных уравнений).

Соответственно, от стойких поточных схем требуется:

- большие периоды выходных последовательностей;
- хорошие статистические свойства выходных последовательностей («постулаты Соломона-Голомба» [26]);
- нелинейность (точнее, высокая линейная сложность) выходных последовательностей.

Исследования поточных схем протекают более динамично, чем исследования блочных шифров. В то время как исследования DES-алгоритма до недавнего времени шли без видимых продвижений, область поточного шифрования испытала множество "взлетов и падений", некоторые схемы, вначале казавшиеся стойкими, "рухнули" при последующем исследовании.

Вопросам исследования поточных шифров уделяется больше внимания в европейских криптографических центрах, в то время как в США больше уделяется внимание блочным шифрам.

Криптографические исследования поточных шифров явились источником ряда задач для фундаментальных направлений дискретной математики:

1. Задача анализа свойств регистров сдвига с линейной обратной связью стимулировали исследования линейных рекуррентных последовательностей над полями и кольцами.

2. Задача поиска статистических связей между входом и выходом узла, реализующего дискретную (булеву) функцию, и построение функций с заданными свойствами.

Криптоанализ по побочным каналам

В последнее время одним из самых актуальных направлений криптоанализа стало осуществление атак, использующих особенности реализации и рабочей среды. **Атаки по сторонним, или побочным, каналам** — это вид криптографических атак, использующих информацию, полученную по сторонним или побочным каналам. Под *информацией из побочных каналов* понимается информация, которая может быть получена с устройства шифрования и не является при этом ни открытым текстом, ни шифртекстом. При подготовке этого раздела использовались материалы доклада А.Е.Жукова на конференции РусКрипто2006 [85].

Почти все осуществленные на практике удачные атаки на криптосистемы используют слабости в реализации и размещении механизмов криптоалгоритма. Такие атаки основаны на корреляции между значениями физических параметров, измеряемых в разные моменты во время вычислений (потребление энергии, время вычислений, электромагнитное излучение и т.п.), и внутренним состоянием вычислительного устройства, имеющим отношение к секретному ключу. На практике атаки по побочным каналам на много порядков более эффективны, чем традиционные атаки, основанные только на математическом анализе. При этом атаки по побочным каналам используют особенности реализации (поэтому их иногда называют также называют атаками на реализацию - implementation attacks) для извлечения секретных параметров, задействованных в вычислениях. Такой подход менее обобщённый, поскольку привязан к конкретной реализации, но зачастую более мощный, чем классический криптоанализ.

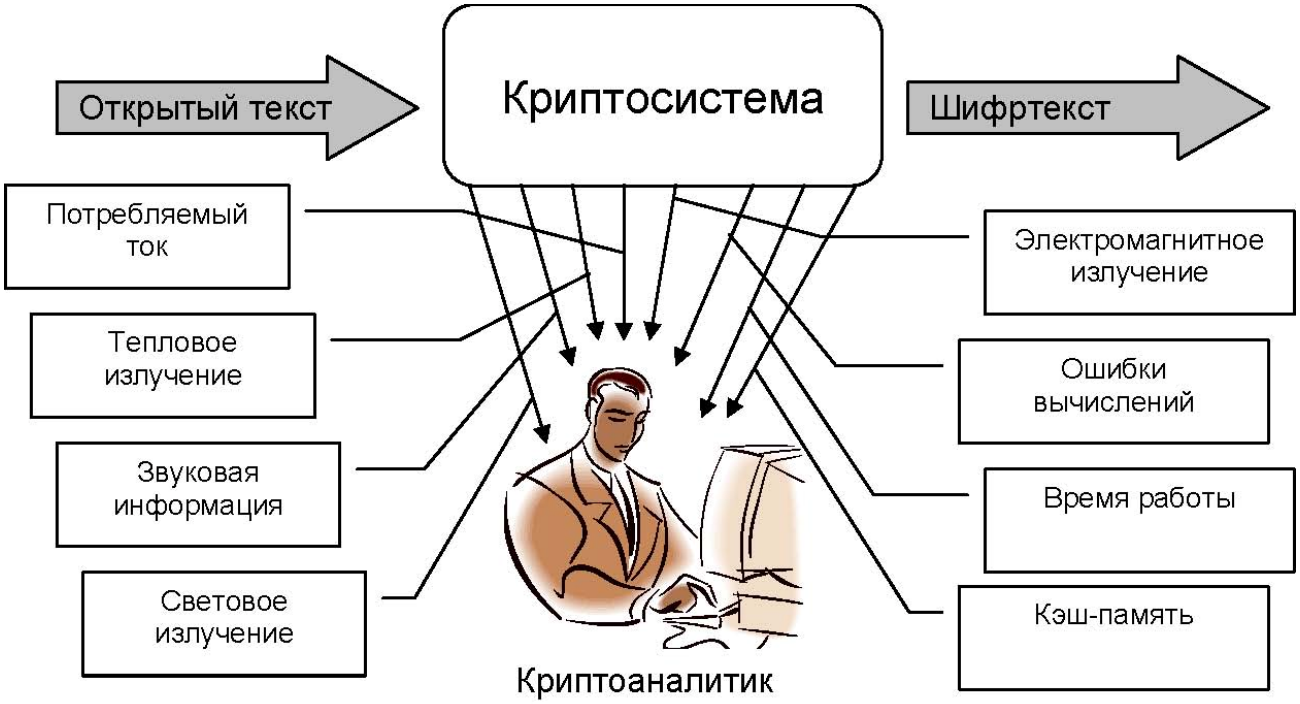

Рис. 3. Источники побочных каналов утечки информации

В последние годы резко возросло количество криптографических атак, использующих особенности реализации и рабочей среды. Например, противник может отслеживать энергию, потребляемую смарт-картой, когда она выполняет операции с закрытым ключом, такие, как расшифрование или генерация подписи. Противник может также замерять время, затрачиваемое на выполнение криптографической операции, или анализировать поведение криптографического устройства при возникновении определённых ошибок. Побочную информацию на практике собрать порой несложно, поэтому нужно обязательно учитывать такую угрозу при оценке защищённости системы.

Атаки по побочным каналам классифицируются по следующим трём типам:

- по контролю над вычислительным процессом: *пассивные* и *активные*.
- по способу доступа к модулю: *агрессивные* (invasive), *полуагрессивные* (semi-invasive) и *неагрессивные* (non-invasive).
- по методу, применяемому в процессе анализа: простые – simple side channel attack (*SSCA*) и разностные – differential side channel attack (*DSCA*).

На сегодняшний день выделено более десяти побочных каналов. Атаки различаются по виду используемого побочного канала (рис. 2): атаки по времени исполнения (Timing Attacks), атаки по энергопотреблению (Power Analysis Attacks), атаки по ошибкам вычислений (Fault Attacks), атаки по электромагнитному излучению (ElectroMagnetic Analysis), атаки по ошибкам в канале связи (Error Message Attacks). Существуюти более изощренные виды атак: атаки по кэш-памяти (Cache-based Attacks), акустические атаки (Acoustic Attacks), атаки по световому излучению (Visible Light Attacks).

Атака по времени

Атака по времени – способ получения какой-либо скрытой информации путем точного измерения времени, которое требуется пользователю для выполнения криптографических операций. Это – самая первая из атак по побочным каналам, появившаяся в гражданской криптографии. Зачастую время обработки данных в криптосистемах немного изменяется в зависимости от входных значений (например, открытого текста или шифртекста). Это является следствием оптимизации производительности и широкого круга иных причин. Атака по времени основана на измерении времени, необходимого модулю шифрования для выполнения операции шифрования. Эта информация может вести к раскрытию информации о секретном ключе. Например, тщательно измеряя время, требуемое для выполнения операций с

секретным ключом, атакующий может найти точное значение экспоненты в алгоритме Diffie-Hellman [19].

Атаки по времени наделали много шума в прессе в 1995 году: закрытые ключи RSA могут быть восстановлены измерением относительных интервалов времени, затраченных на произведение криптографических операций. Эти атаки были успешно применены к карточкам с микропроцессорами и другим средствам надёжной идентификации, а также к серверам электронной коммерции в Сети.

Атаки по мощности

Атака по анализу мощности пригодна в основном для аппаратной реализации криптографических средств и успешно применяется при взломе смарт-карт и других систем, в которых хранится секретный ключ.

Чтобы измерить потребляемую схемой мощность, необходимо последовательно с цепью питания или заземления подключить резистор малого сопротивления (например, 50 Ом). Падение напряжения, деленное на сопротивление, даст силу тока. Современные лаборатории располагают оборудованием, способным производить цифровые измерения напряжения на исключительно высоких частотах (более 1 ГГц) и с превосходной точностью (ошибка менее 1%).

Атака по мощности может быть разделена на простую (Simple Power Analysis, SPA) и разностную (Differential Power Analysis, DPA). Целью SPA является получение информации о конкретных выполняемых инструкциях в системе и о конкретных обрабатываемых данных. В общем случае SPA может дать как сведения о работе устройства, так и информацию о ключе. Для осуществления этой атаки криптоаналитик должен располагать точными данными о реализации устройства. Этот метод использует непосредственные данные измерений, собранные во время выполнения криптографических операций. Согласно [13], простая атака по мощности для смарт-карт обычно занимает несколько секунд, в товремя как разностная атака по мощности может занять несколько часов.

В отличие от простых атак, разностные атаки, основанные на анализе потребляемой мощности, подразумевают не только визуальное представление потребляемой мощности, но также статистический анализ и статистические методы исправления ошибок для получения информации о ключах. Более того, DPA зачастую не нуждается в данных о конкретной реализации и в качестве альтернативы использует статистические методы анализа. Разностный анализ мощности – одно из самых мощных средств для проведения атак, использующих побочные каналы, причем эта атака требует очень маленьких затрат.

Атаки по ошибкам вычислений

Ошибки аппаратного обеспечения, появляющиеся во время работы соответствующего криптографического модуля, или ошибочные выходные данные могут стать важными побочными каналами и иногда существенно увеличивают уязвимость шифра к криптоанализу. Криптоанализ на основе формирования случайных аппаратных ошибок -это вид нападения на шифры в случае, когда предполагаемый нарушитель имеет возможность оказать на шифратор внешнее физическоевоздействие и вызвать одиночные ошибки в процессе шифрования одного блока данных. Атаки по ошибкам на криптографические алгоритмы изучаются с 1996 года, и с того времени почти все криптографические алгоритмы были подвержены атакам такого вида.

Осуществимость атаки по ошибкам (или, по крайней мере, ее эффективность) зависит от возможностей злоумышленника вызывать ошибки в системе специально или пользоваться сбоями естественного происхождения. Ошибки наиболее часто происходят из-за скачков напряжения, сбоев часов или из-за излучений различных типов. Рассмотрение вопроса стойкости к этому методу особенно актуально для шифраторов, применяемых в интеллектуальных электронных карточках. В основном ошибки классифицируются по следующим аспектам:

- Точность, которую нарушитель может достичь при выборе времени и места, где появляется ошибка во время работы криптографического модуля.
- Длина данных, на которые влияет ошибка; например, только один бит.
- Постоянство ошибки; является ли ошибка кратковременной или постоянной.

- Тип ошибки; такие как изменение одного бита; изменение одного бита, но только в одном направлении (например, с 1 на 0); изменение бита на случайное значение и др.

В общем, успешная атака по ошибкам на криптографические модули или устройства требует двух шагов: шаг создания ошибки и шаг использования ошибки. Ошибки могут быть вызваны в смарт-картах путем внешнего влияния на нее и помещения ее в неправильные условия. Некоторые из них – аномальное и внезапное понижение или повышение напряжения, частоты, температуры, излучения, освещения и др.

Разностный анализ по ошибкам состоит в изучении результата работы алгоритма шифрования в нормальных и ненормальных условиях при одном и том же входе (открытом тексте). Ненормальные условия обычно получаются созданием ошибки в процессе (кратковременная ошибка) или перед процессом (постоянная ошибка) работы. Разностный анализ по ошибкам широко изучены с теоретической точки зрения и кажутся применимыми почти ко всем симметричным криптосистемам.

Для ГОСТ 28147-89 была показана возможность раскрытия ключа и таблиц подстановок с помощью криптоанализа на основе формирования случайных аппаратных ошибок [93]. DES, RC5 и другие шифры также являются уязвимыми по отношению к этому виду криптоанализа, поэтому при их использовании необходимо обеспечить защиту аппаратуры от навязывания сбоев.

Атаки по электромагнитному излучению

Выполнение вычислительных операций на компьютере сопряжено с выделением электромагнитного излучения. Измеряя и анализируя это излучение, нарушитель может получить значительную информацию о выполняющихся вычислениях и используемых данных. Атаки по электромагнитному анализу могут быть также разделены на две большие категории: простые (SEMA) и дифференциальные (DEMA).

Стойкость советского и американского стандартов симметричного шифрования

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 $\mathbf{Z}_2^{48} \times \mathbf{Z}_2^{32} \to \mathbf{Z}_2^{32}$, 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 алгоритмприобретает 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.

Использование новых технологий в криптоанализе

Этот раздел посвящен самым необычным и достаточно спорным подходам к исследованию криптографических систем. На данный момент эти методы не привели к сколько-нибудь серьезным прорывам во взломе шифров и представляют в большей степени академический интерес, чем практический. Тем не менее, эти методы заслуживают внимания хотя бы из-за своей оригинальности; кроме того, не исключено, что со временем их значение в криптологии возрастет.

Нейронные сети

Криптосистему можно рассматривать как «черный ящик», т.е. устройство или программу, о внутренней структуре которой ничего не известно, но, подавая сигналы команды или данные на вход, можно получить реакцию на выходе. Задача криптоанализа – идентификация этой системы, т.е. определение ее структуры на основе сигналов, поступающих на ее вход и получаемых на выходе. Одним из инструментов решения этой задачи могут являться нейронные сети, теория которых изложена в [82].

Искусственная нейронная сеть— это математическая модель, а также устройства параллельных вычислений, представляющие собой систему из соединенных и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно исключительно просты, особенно в сравнении с процессорами, используемыми в персональных компьютерах. Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. Тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи. Понятие возникло при изучениипроцессов, протекающих в мозге при мышлении, и при попытке смоделировать эти процессы. Полученные модели называются искусственными нейронными сетями (ИНС). Схема простой нейросети изображена на рис. 3. Черным обозначены входные элементы, белым — выходной элемент.

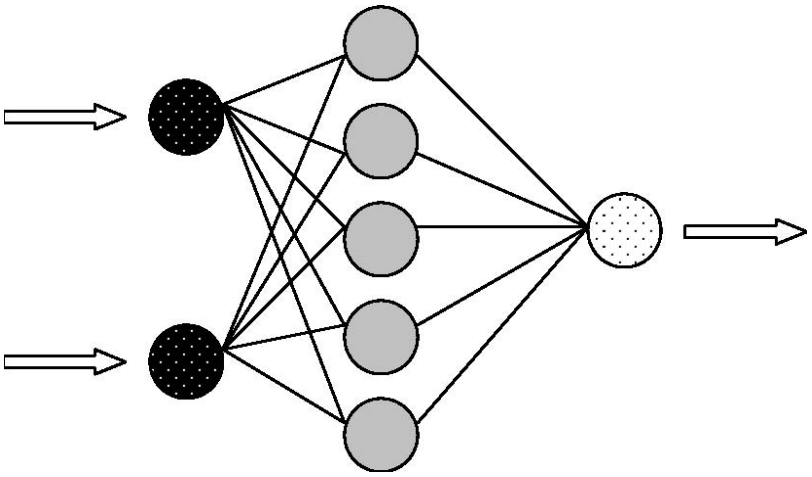

Рис. 4. Нейронные сети

Брюс Шнайер в своей книге «Прикладная криптография» [58] склоняется к пессимизму в отношении к применимости нейронных сетей в криптоанализе: «Процесс взлома не оставляет места обучению: вы либо раскрываете ключ, либо нет. (По крайней мере, это верно при вскрытии любого надежного алгоритма). Нейронные сети хорошо работают в структурированных средах, допускающих обучение, но не в высокоэнтропийном, предположительно случайном мире криптографии». Тем не менее, исследования в этом направлении продолжаются.

В статье [Al-Ubaidi] разработана так называемая «атака черного ящика» на классические и поточные криптосистемы, основанная на построении модели «нейро-распознавателя черного ящика» (Black-Box Neuro Identifier). Преследуются две цели: во-первых, определение ключа на основе открытых и соответствующим им зашифрованных текстов; во-вторых, создание нейро-модели исследуемой криптосистемы.

Идентификация системы заключается в определении правил функционирования системы-«черного ящика» по входным и выходным данным и аппроксимации неизвестной функции моделью нейронной сети. Первым этапом является выбор нейронной модели, которая характеризуется определенной архитектурой и алгоритмом обучения. Выбор осуществляется методом проб и ошибок. Множество известных пар открытых текстов и криптограмм разделяется на два подмножества, одно из которых используется для обучения сети, а другое – для проверки соответствия полученной модели заданному критерию точности. Оптимальной считается модель, имеющая минимальное число нейронов и при этом удовлетворяющая критерию.

При помощи описанной системы в работе [3] осуществлялся криптоанализ классического полиалфавитного шифра Вижинера и поточных шифров. При установке порога ошибки на уровне 10^{-5} удалось получить модель криптосистем, выдающую результат со 100%-й точностью. К преимуществам такого подхода по сравнению с другими современными методами относится независимость результата от используемого естественного языка и его статистических характеристик, поскольку для получения знаний система использует адаптивный обучающий процесс.

В статье [29] говорится о применении нейронных сетей для взлома DES. В процессе обучения использовалось 2240 пар открытых текстов и криптограмм, что позволило получать результаты с точностью до 98%. В планы авторов также входит криптоанализ AES с использованием разработанной стратегии.

Генетические алгоритмы

Один из первых шифров на основе задачи об укладке ранца был предложен Меркли и Хеллманом в 1978 [46]. Это была одна из первых попыток создания системы шифрования с открытым ключом. Несмотря на то, что проблема укладки ранца относится к классу NP-полных, было показано, что большинство версий алгоритма являются нестойкими. В 1983 г. Брикел предложилспособ взлома криптосистемы на основе ранца низкой плотности [14]. Год спустя Шамир разработал полиномиальный алгоритм для атаки на исходную «рюкзачную» криптосистему [61]. После этого было предложено множество других систем на основе алгоритма укладки ранца: несколько последовательных рюкзаков, рюкзаки Грэм-Шамира (Garham-Shamir) и др. [58]. Для всех этих систем были разработаны методы вскрытия. В статье [66] предлагается еще один метод криптоанализа шифров на основе алгоритма укладки ранца; отличительной особенностью такого подхода является его универсальность, т.е. возможность применения к любой версии «рюкзачной» криптосистемы, а также простота работы. В метод базируется на использовании генетических алгоритмов.

Генетические алгоритмы были разработаны Джоном Холландом и представляют собой модификацию так называемого «эволюционного программирования» [30]. Идея Холланда заключалась в том, чтобы создать алгоритм поиска на основе механизмов естественного отбора, известного из биологии, или алгоритм «направленного» случайного поиска. На этапе инициализации процедуры создается популяция возможных решений. На основе этой популяции выводится новое поколение решений, которое, в свою очередь, служит «исходным материалом» для очередного поколения, и т.д. Цикл генетического алгоритма в общем виде представлен на рис. 4 и включает стадии отбора, скрещивания и мутации. Лучшие представители поколения отбираются для воспроизводства популяции; таким образом, по предположению, каждое новое поколение должно содержать лучшие решения, чем предыдущее. Во многих случаях это соответствует действительности [30].

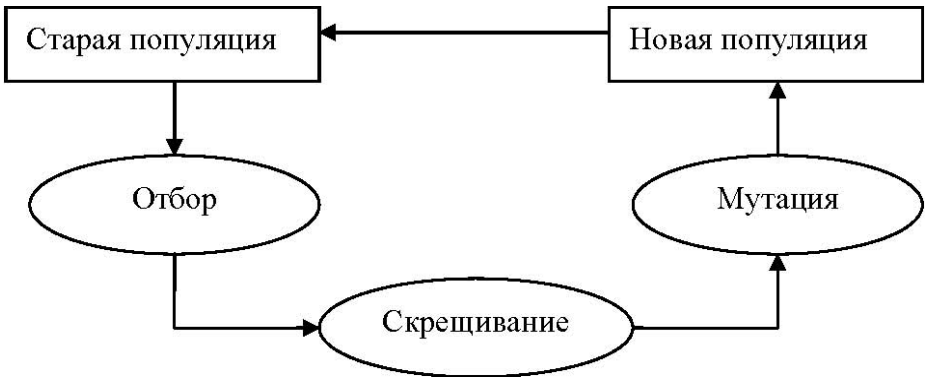

Рис. 5. Схема эволюции популяции

Популяция состоит из набора бинарных строк. Каждая бинарная строка представляет решение проблемы и называется хромосомой. Первой стадией генетического алгоритма является отбор. В процессе отбора определяются строки, которые будут использоваться при создании нового поколения. «Родители» выбираются произвольно, однако «лучшие особи» популяции имеют больший шанс оказаться выбранными. Таким образом алгоритм продвигается в самом перспективном направлении поиска. Следующая стадия – скрещивание. Скрещивание заключается в том, что для пары отобранных строк длины r каждая выбирается произвольным образом число $s \in \{1, \ldots, r\}$. «Родители» обмениваются битами от $s+1$ до r, таким образом получаются хромосомы потомков (рис. 4).

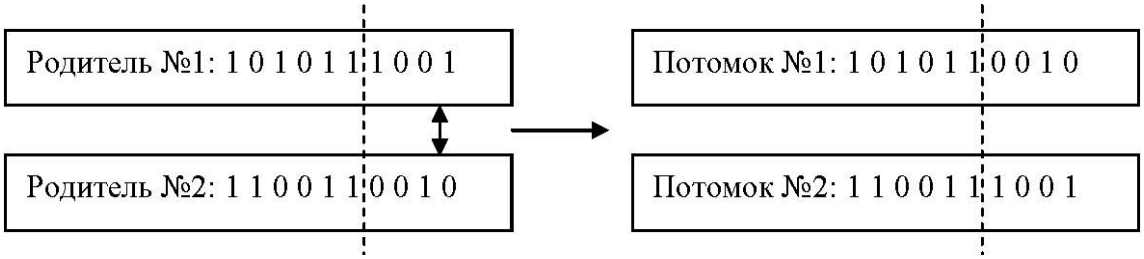

Рис. 6. Передача генетической информации от «родителей» к «потомкам»

Заключительная стадия – мутация. При инициализации алгоритма устанавливается фиксированная маленькая вероятность мутации, которой подвергаются вновь образовавшиеся хромосомы (рис.5).

Эти стадии повторяются до достижения условия выхода из цикла (таким условием может быть, к примеру, превышение максимального количества популяций).

Рассмотрим формулировку задачи об укладке ранца. Дано множество предметов различного веса; спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению? Более формально задача формулируется так: дан набор значений $M_1, M_2, \ldots, M_n$ и суммарное значение S требуется вычислить значения b_i такие, что:

$S = b_1 M_1 + b_2 M_2 + \cdots + b_n M_n$.

Здесь b_i может быть либо нулем, либо единицей. Значение $b_i = 1$ означает, что i-й предмет кладут в рюкзак, а $b_i = 0$ -не кладут. Отсюда очевидным образом вытекает представление содержимого рюкзака в виде хромосом, биты которых соответствуют значениям b_i. Функция выбора «лучших хромосом» оценивает близость веса конкретного рюкзака к заданному числу. Значения функции располагаются в диапазоне $[0,1]$, где 1 означает точное совпадение с искомым весом. Если вес одного рюкзака превышает целевое значение S на некоторое число x, а вес другого, напротив, меньше требуемого на то же число x, то «лучшим» считается последний рюкзак. Более формально эта функция описывается ниже:

1. Вычислить максимальное расхождение, которое может возникнуть между произвольной хромосомой и целевым значением S: $\Delta_{\max} = \max\left(S, \tilde{S} - S\right)$, где $\tilde{S}$ - сумма всех компонентов, которые можно использовать при укладке рюкзака

2. Вычислить вес рюкзака, соответствующего текущей хромосоме, и обозначить S'.

3. Если $S' \le S$, то «качество» хромосомы оценивается значением: $\alpha = 1 - \sqrt{\frac{|S' - S|}{S}}$.

4. Если $S' > S$, то $\alpha = 1 - \sqrt{\frac{|S' - S|}{\Delta_{\max}}}$.

Общий алгоритм, примененный в статье для криптоанализа задачи об укладке ранца, имеет вид:

1. Создается случайная популяция двоичных хромосом.

2. Для каждой хромосомы вычисляется значение α (функция оценки).

3. На основе полученных коэффициентов происходит естественный отбор.

4. К выбранным на 3-м этапе особям применяется скрещивание

5. Потомки подвергаются мутации.

6. Новая популяция анализируется, выделяются «лучшие хромосомы».

Процесс прервется, когда количество поколений превысит определенное заданное число; «лучшие хромосомы» будут использованы для взлома шифра. Для простого шифра подстановки, исследованного в статье, алгоритм показал хорошие результаты:

для достижения оптимальной точки, т.е. секретного ключа, алгоритму потребовалось исследовать в среднем не более 2% всего ключевого пространства.

Генетические алгоритмы успешно применяются в криптоанализе перестановочных и подстановочных шифров [44, 67].

Квантовые компьютеры

С помощью квантового компьютера можно проводить вычисления, которые не реализуемы на сегодняшних (классических) компьютерах [89, 94]. В квантовой физике состояние частицы характеризуется так называемой волновой функцией ψ, которая

принимает комплексные значения, называемые амплитудами. Аргументом волновой функции является время, а также некоторый набор физических параметров (например, координаты частицы). В теории алгоритмов принято оперировать конечными объектами. Поэтому от принятых в физике бесконечномерных моделей перейдем к конечномерным, считая аргументы волновой функции дискретными.

Для простоты рассмотрим случай, когда волновая функция $\psi(x,t)$ зависит только от координаты x и времени t, которое пока зафиксируем. Пусть сначала x *може*т принимать только два значения: 0 и 1. Это соответствует случаю, когда частица может находиться только в двух различных точках. Можно посмотреть на этот случай с другой стороны. Предположим, в нашем распоряжении только один бит для хранения информации о местонахождении частицы, которая находится где-то на отрезке $[0,1]$.

Тогда мы примем $x=0$, если она находится в левой половине этого отрезка, и $x=1$, если в правой. Рассмотрим две вспомогательные функции: $|0\rangle$ и $|1\rangle$. Первая равна 1 при $x=0$ и нулю при $x=1$, а вторая наоборот. Тогда любую волновую функцию $\psi(x)$ можно единственным образом записать в форме $\lambda_0|0\rangle+\lambda_1|1\rangle$. Эта запись соответствует разложению вектора двумерного комплексного пространства по базису $|0\rangle$, $|1\rangle$. Если считать эти базисные вектора ортогональными и единичной длины, то получим, что волновая функция есть вектор двумерного комплексного пространства с выделенным ортонормированным базисом. Такая частица называется *квантовым битом*, или *кубитом*.

Единственный способ узнать, в какой точке находится частица в данным момент времени – это измерить ее волновую функцию. Измерение даст нам любой из векторов $|0\rangle$ и $|1\rangle$ - каждый с вероятностью $|\lambda_0|^2$ и $|\lambda_1|^2$ соответственно. λ_0 и λ_1 — это *амплитуды*, связанные условием нормировки: $|\lambda_0|^2+|\lambda_1|^2=1$ (суммарная вероятность равна единице).

Система из n кубитов имеет пространство состояний размерности 2^n. Именно этот экспоненциальный рост пространства состояний в зависимости от числа частиц даёт преимущество в скорости вычислений на квантовых компьютерах в сравнении с классическими.

В 1994 году Питер Шор открыл так называемый «ограниченно-вероятностный» алгоритм факторизации [64], который позволяет разложить на множители число N за полиномиальное время $O(\log^3 N)$, затратив $O(\log N)$ места на квантовом компьютере. В большинстве алгоритмов, включая алгоритм Шора, используется стандартный способ сведения задачи разложения к задаче поиска периода функции. Шор использует квантовый параллелизм для получения суперпозиции всех значений функции за один шаг.

Затем он производит квантовое преобразование Фурье, результатом которого, как и для классического преобразовании Фурье, является функция, аргумент которой кратен величине, обратной периоду. С высокой вероятностью измерение состояния возвращает период, который, в свою очередь, служит для разложения целого числа *N*. Вышесказанное раскрывает суть квантового алгоритма в очень упрощённом виде. Проблема заключается в том, что квантовое преобразование Фурье основано на быстром преобразовании Фурье и, таким образом, в большинстве случаев дает только приблизительный результат.

Алгоритм Шора разложения чисел на множители явился, пожалуй, главным достижением в области квантовых вычислительных алгоритмов. Это был не только

крупный успех математики. Именно с этого момента началось усиленное финансирование работ по созданию квантовых компьютеров.

Эффективность алгоритма Шора была поставлена под сомнение японскими учеными из компании SHARP. Дело в том, что как сам Шор, так и все математики, работающие в области квантовых алгоритмов, говорят о количестве операций, хотя практически важно именно время расчета, которое и определили японцы [55]. Дискретное преобразование Фурье (ДПФ) на квантовом компьютере выполняется как чередующиеся друг за другом преобразования Адамара [94] над отдельными кубитами и операции условного поворота фазы в одном кубите j в зависимости от состояния другого кубита к на угол $\theta = \frac{\pi}{2^{k-j}}$. Если число кубитов равно n, то минимальный угол равен $\frac{\pi}{2^{k-j}}$. Пусть на эту операцию мы затрачиваем время $\tau_{\min}$, тогда на операцию поворота на угол для $\frac{\pi}{2}$ для соседних кубитов мы затратим время порядка $\tau = \tau_{\min} 2^n$. Экспоненциальная зависимость, полученная японскими учеными, сводит на нет преимущества алгоритма Шора во времени расчета.

Однако сотрудник ФТИАН Леонид Федичкин указал на опубликованную в 1996 году работу финских авторов [7]. Они исследовали влияние шума на точность ДПФ. Как показано в [55], такое преобразование требует экспоненциально большого (по отношению к числу кубитов n) динамического диапазона углов θ. Что будет, если поворот на малые углы забивается шумом? Оказалось, что требуемая точность операции фазового сдвига в состоянии кубита допускает устранение операции поворота фазы на малые углы. Расчеты Федичкина показывают, что исключение этой операции сохраняет полиномиальную зависимость времени выполнения алгоритма Шора от количества кубитов n.

Так как алгоритм Шора работает только на квантовом компьютере, в настоящее время не существует технических средств, позволяющих за полиномиальное время разложить достаточно большое число на множители. Таким образом, самым важным вопросом остаётся создание квантового компьютера. Алгоритм Шора чрезвычайно прост и довольствуется гораздо более скромным аппаратным обеспечением, чем то, которое понадобилось бы для универсального квантового компьютера. Поэтому вероятно, что квантовое устройство для разложения на множители будет построено задолго до того, как весь диапазон квантовых вычислений станет технологически осуществимым. На сегодняшний день есть конкретные результаты. Так, IBM продемонстрировала использование созданного в лабораториях компании семикубитового квантового компьютера для факторизации чисел по алгоритму Шора. Хотя решённаяим задача вряд ли способна поразить воображение (компьютер верно определил, что делителями числа 15 являются числа 5 и 3), это самое сложное вычисление за всю историю квантовых компьютеров.

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 информационных системах. Курс лекций. И.Д. Горбенко. ХНУРЭ. 2002.

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

3. Al-Ubaidy M. K. I. Black-box attack using neuro-identifier // Cryptologia, Oct 2004.

4. ANSI X3.92. American National Standard for Data Encryption Algorithm (DEA). American National Standards Institute, 1981.

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

13. Black J., Urtubia H. Side-channel attacks on symmetric encryption schemes: the case for authenticated encryption // Proc of 11th USENIX Security Symposium, 2002. P. 327-338.

14. Brickell E. Solving Low Density Knapsacks // Advances in Cryptology: Proceedings of CRYPTO. New York: Plenum Press, 1984. P. 25 -37.

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

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.

26. Golomb S.W. Shift Register Sequences. Holden-Day, San Francisco, 1967.

29. Haranadh Gavara, Harendra Kumar Mishra, Surendra Kumar Y. Cryptanalysis using Neural Networks // Available via http://netlab.cs.iitm.ernet.in/cs650/2006/TermPapers/Group9.pdf

30. Holland J. Adaptation in Natural and Artificial Systems. Ann Arbor MI: University of Michigan Press. 1975.

44. Matthews R. The use of genetic algorithms in cryptanalysts // Cryptologia. 17(2), 1993.

46. Merkle R.C., Hellman M.E. Hiding Information and Signatures in Trapdoor Knapsacks // IEEE transactions on Information Theory. V. 24, n. 5, Sep 1978. P. 525-530.

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

55. Saito A., Kioi K., Akagi Y., Hashizume N., Ohta K. Actual computational time-cost of the Quantum Fourier Transform in a quantum computer using nuclear spins. // Quantum Physics, abstract quant-ph/0001113

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).

61. Shamir A. A Polynomial -Time Algorithm for Breaking the Basic Merkle -Hellman Cryptosystem // Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science, 1982. P. 145 – 152.

64. Shor P. W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring // In Proceedings, 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, November 20–22, 1994, IEEE Computer Society Press. P. 124–134.

66. Spillman R. Cryptanalysis of knapsack ciphers using genetic algorithms // Cryptologia, 17(1), 1993.

67. Spillman R., Janssen M., Nelson B., Kepner M. Use of a Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers // Cryptologia. 17(1), 1993. P. 31-44.

71. What is RC4? // RSA Security Laboratories; available via http://www.rsasecurity.com/rsalabs/node.asp?id=2250

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.

82. Галушкин А. Теория нейронных сетей. М.:ИПРЖР, 2000.

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

85. Жуков А.Е. Криптоанализ по побочным каналам (Side Channel Attacks). // Материалы конференции РусКрипто – 2006. // Опубликовано: http://ruscrypto.ru/sources/publications

89. Китаев А., Шень А., Вялый М. Классические и квантовые вычисления. М.: МЦНМО, 1999.

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

94. Ожигов Ю.И. Квантовые вычисления. Учебно-методическое пособие. М.: МГУ, факультет ВМиК, 2003.

96. Основные тенденции развития открытой криптографии (обзор по заказу cryptography.ru) // Опубликовано: geo.com.ru