ДИФФЕРЕНЦИАЛЬНЫЙ КРИПТОАНАЛИЗ НА ОСНОВЕ СБОЕВ УСТРОЙСТВА (DIFFERENTIAL FAULT ANALYSIS).

Доклад подготовил Бурдасов Вячеслав, 015гр.

Дифференциальный криптоанализ – метод, с помощью которого анализируются различия между парами открытых текстов на основе отличий в результирующих парах шифротекстов. Эти различия могут быть использованы с целью поставить в соответствие вероятности различным ключам и обнаружить наиболее возможный из них. Обычно этот метод реализуется на большом количестве открытых текстов с одинаковым отличаем. Для DES-подобных криптосистем в качестве такого отличая выбрано значение, полученное посредством вычисления операции XOR над двумя открытыми текстами. Покажем, как эти отличия могут быть проанализированы и использованы для решения задачи нахождения ключа.

Как известно, на вход F-функции DES подаются 32-битное входное значение (R_0) и выборка 48 бит ключа. Вход функции расширяется за счет E-блоков до 48-битного значения и происходит реализация процедуры XOR над расширенным входом функции и 48-битным раундовым ключом. Результат помещается в S-боксы и значение на выходе S-боксов подвергается процедуре преобразования, реализующей перестановку битов.

Несмотря на то, что алгоритм DES кажется нелинейным в отношении входных бит, при одновременном изменении отдельных комбинаций входных бит соответствующие изменения с большой вероятностью после нескольких раундов претерпевают и промежуточные биты. Это свойство описывают посредством результата выполнения операции XOR над двумя открытыми текстами, операции XOR для двух значений промежуточных раундов и соответствующими им вероятностями.

DES содержит S-боксы, которые являются нелинейными таблицами. Знание значения XOR между двумя парами входных текстов не может гарантировать знания значений операции XOR над парами соответствующих выходных значений. Как бы то ни было, знание результата выполнения операции XOR над любыми двумя входными значениями приводит к распределению вероятности значений возможных результатов выполнения операции XOR над соответствующей парой выходных значений. Такое распределение представлено в таблице 1. В этой таблице описано распределение вероятностей получить определенное значение в результате применения операции XOR над двумя парами выходных (из блока S1) значений при возможных значениях операции XOR над входными значениями.

Вышеуказанное свойство может быть использовано для определения бит ключа. Если мы можем найти результат выполнения операции XOR для двух выходных блоков данных функции F последнего раунда, то сможем указать наиболее вероятное значение результата применения операции XOR для блока входных данных. Таким образом, с определенной вероятностью, мы можем указать значения XOR для входных и выходных блоков каждого S-бокса последнего раунда, а, следовательно, и значения некоторых бит ключа.

Input XOR	Output XOR															
	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
2_x	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
3_x	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
4_x	0	0	0	6	0	10	10	6	0	4	6	4	2	8	6	2
5_x	4	8	6	2	2	4	4	2	0	4	4	0	12	2	4	6
6_x	0	4	2	4	8	2	6	2	8	4	4	2	4	2	0	12
7_x	2	4	10	4	0	4	8	4	2	4	8	2	2	2	4	4
8_x	0	0	0	12	0	8	8	4	0	6	2	8	8	2	2	4
9_x	10	2	4	0	2	4	6	0	2	2	8	0	10	0	2	12
A_x	0	8	6	2	2	8	6	0	6	4	8	0	4	0	2	10
B_x	2	4	0	10	2	2	4	0	2	6	2	6	6	4	2	12
C_x	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
D_x	6	6	4	8	4	8	2	6	0	6	4	6	0	2	0	2
E_x	0	4	8	8	6	6	4	0	6	6	4	0	0	4	0	8
F_x	2	0	2	4	4	6	4	2	4	8	2	2	2	6	8	8
									⋮							
30_x	0	4	6	0	12	6	2	2	8	2	4	4	6	2	2	4
31_x	4	8	2	10	2	2	2	2	6	0	6	2	2	4	10	8
32_x	4	2	6	4	4	2	2	4	6	6	4	8	2	2	8	0
33_x	4	4	6	2	10	8	4	2	4	0	2	2	4	6	2	4
34_x	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
35_x	2	2	4	0	8	0	0	0	14	4	6	8	0	2	14	0
36_x	2	6	2	2	8	0	2	2	4	2	6	8	2	4	10	0
37_x	2	2	12	4	2	4	4	10	4	4	2	6	0	2	2	4
38_x	0	6	2	2	2	0	2	2	4	6	4	4	4	6	10	10
39_x	6	2	2	4	12	6	4	8	4	0	2	4	2	4	4	0
$3A_x$	6	4	6	4	6	8	0	6	2	2	6	2	2	6	4	0
$3B_x$	2	6	4	0	0	2	4	6	4	6	8	6	4	4	6	2
$3C_x$	0	10	4	0	12	0	4	2	6	0	4	12	4	4	2	0
$3D_x$	0	8	6	2	2	6	0	8	4	4	0	4	0	12	4	4
$3E_x$	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
$3F_x$	4	8	4	2	4	0	2	4	4	2	4	8	8	6	2	2

Таблица 1. Распределение вероятностей получить определенное значение в результате применения операции XOR над двумя парами выходных (из блока S1) значений при возможных значениях операции XOR над входными значениями. Таблица взята из статьи Эли Байхэма (Eli Biham), Ади Шамира (Adi Shamir) "Differential Cryptanalysis of DES-like Cryptosystems", The Weizmann Institute of Science Department of Applied Mathematics.

Как известно, число циклов преобразования в полнораундовой схеме DES равно шестнадцати. Возникает вопрос: почему именно 16? Быть может, было бы достаточно 3, 5, 7? Исчерпывающий ответ на этот вопрос дает применение для анализа его криптостойкости дифференциального криптоанализа, впервые реализованного для атаки на алгоритм FEAL-4 [Murphy S. The Cryptanalysis of FEAL-4 With 20 Chosen Plaintexts].

Известно, что после пяти циклов преобразования каждый бит шифротекста является функцией всех битов ключа и открытого текста. После восьми циклов преобразования наблюдается пик лавинного эффекта – шифротекст представляет собой случайную функцию всех битов ключа и открытого текста [Konheim A.C. Cryptography: A Primer]. На данный момент перед разработчиками криптографических алгоритмов не стоит задача создания идеального в отношении криптостойкости алгоритма (цель, которая не достижима в принципе), она формулируется иначе: создать алгоритм, эффективная атака на который вызвала бы б*о*льшие издержки, чем стоимость зашифрованной информации. Так, реализация brute-force-атаки (атаки грубой силой) на алгоритм ГОСТ 28147-89 (стандарт России) в силу высокой разрядности ключа (256 бит) на данный момент (и, скорее всего, в ближайшие несколько лет) совершенно бесперспективна.

Из истории современной криптографии известно, что метод дифференциального криптоанализа был засекречен, что подтверждено публичными заявлениями самих разработчиков [Coppersmith D. The Data Encryption Standard (DES) and its Strength against Attacks]. Именно здесь и лежит ответ на вопрос о количестве раундов DES: доказано, что трудоемкость атаки на открытом

тексте для DES (при любом количестве текстов) с числом раундов, меньшим 16, ниже, чем трудоемкость силовой атаки.

Рассмотрим более подробно один из видов дифференциального криптоанализа – дифференциальный криптоанализ на основе отказов устройства.

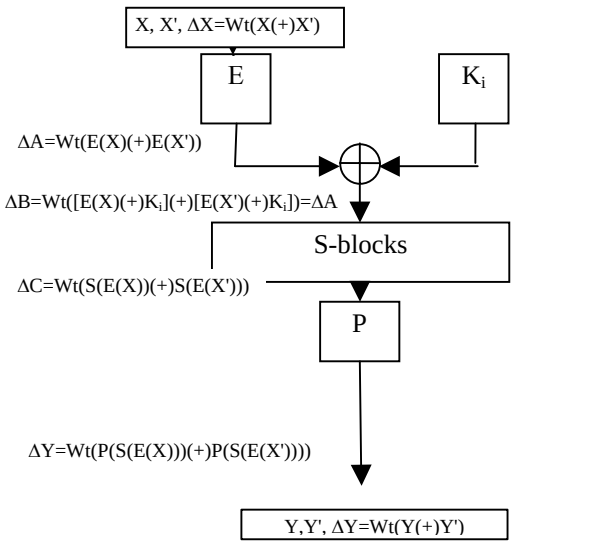
В 1996г. специалисты из Bellcore объявили о новом эффективном методе криптографического анализа, призванном эффективно раскрывать секретные ключи, хранящиеся в памяти портативных шифрующих/расшифровывающих устройств, например Smart Card или PCMCIA, сохранность ключа в которых обеспечивается за счет уникальных характеристик технологии TEMPSET. Эффективность DFA (Differential Fault Analysis) во многих случаях была доказана высокой - для раскрытия ключа DES понадобилось проанализировать 200 шифротекстов, хранящихся в памяти устройства.

Известно, что некоторые виды излучения (например, ионизирующее или радиоактивное) приводят к сбоям в работе электронного оборудования. Криптоаналитик облучает шифрующее/расшифровывающее устройство с целью вызвать инверсию одного или нескольких бит в памяти на промежуточном уровне криптографического преобразования. Для блочных шифров конструкции Фейстеля инверсия возникает на одном из циклов преобразования. Важно знать, что ни позиция инвертированного бита, ни номер цикла криптоаналитику также не известны. Сбой в работе приводит к искажению шифротекста на выходе устройства. Задача криптоаналитика – раскрыть секретный ключ, анализируя искаженные и неискаженные шифротексты. Помимо раскрытия секретного ключа, хранящегося в памяти устройства шифрования/расшифрования, можно попытаться решить и более сложную задачу – идентифицировать неизвестный криптоалгоритм, включая определение неизвестной функции цикла, подключи и даже S-блоки.

Рассмотрим данный метод на примере криптоанализа DES. Предположим, имеется два различных шифротекста, полученных при шифровании одного и того же открытого текста на фиксированном ключе. Известно, какой шифротекст получен в результате инверсии одиночного бита в процессе шифрования. Под инверсией подразумевается следующее: бит правой половины входного блока на одном из 16 циклов DES-преобразования меняет исходное значение 0 на 1 или наоборот. Причем, позиция бита и номер цикла преобразования выбираются случайно и имеют равномерное распределение.

На первом этапе необходимо установить номер цикла преобразования, на котором произошла инверсия. Предположим, инверсия произошла на последнем, 16-м цикле DES-преобразования. Если инверсия произошла в левой половине блока, то два шифротекста будут различаться в одном бите, локализованном в правой половине блока. Эта информация нам ничего, кроме позиции инвертированного бита, не скажет. Различие левых половин блока шифротекста определяется выходами тех S-блоков (одного или двух), на входе которых появился инвертированный бит.

Рассмотрим следующую схему:



Один цикл DES-преобразования для двух входных блоков X и X'.

Пусть задана пара входов X и X' с несходством ΔX (расстояние Хэмминга). Выходы Y и Y' известны => известно и несходство ΔY. Также известна перестановка с расширением E и P-блок. Отсюда также известны ΔA и ΔC. Значения на выходе XOR нам не известны, но известно несходство $\Delta B=Wt([E(X)(+)K_i](+)[E(X')(+)K_i])$ – оно равно ΔA (ведь на данном этапе преобразование производится посредством одного и того же подключа K_i (операция XOR: $K_i(+)K_i \equiv 0$!). ***Доказано, что для любого заданного ΔA не все значения ΔC равновероятны.*** Комбинация ΔA и ΔC позволяет нам предположить значения битов для $E(X)(+)K_i$ и $E(X')(+)K_i$, а,

так как нам известны X и X', то мы можем получить информацию о K_i.

Так как на каждом этапе преобразования участвует 48-битный подключ K_i исходного 56-битного секретного ключа, то раскрытие K_{16} позволяет восстановить 48 бит ключа. Остальные 8 бит ключа можно восстановить посредством силовой атаки (перебора) или же анализируя 15-й раунд преобразования. Последний вариант позволяет эффективно атаковать DES в режиме EDE.

Несходство различных пар открытых текстов приводит к несходству получаемых шифротекстов с определенной вероятностью. Вероятности можно оценить, построив таблицу, строки которой представляют собой всевозможные входы сумматора по модулю 2, столбцы – возможные результаты суммирования, а элемент на пересечении столбца и строки представляет собой частоту появления конкретного результата при указанных входах.

Пара открытых текстов, соответствующих несходству с более высокой вероятностью, подсказывает правильный ключ последнего раунда. Правильный ключ определяется статистически – один и тот же из подключей мы будем встречать чаще, чем все остальные.

Описанный метод работает и в тех случаях, когда инверсия возникает внутри F-функции и процедуры генерации подключей. Успешное применение метода дифференциального криптоанализа может иметь место и для проведения атаки на такие блочные шифры, как IDEA, RC5 и Feal. Некоторые блочные шифры, например, Khufu, Khafre и Blowfish, используют заданный секретный ключ для генерации S-блоков. В этом случае описанный выше метод позволяет раскрыть не только ключи, но и сами S-блоки. Т.о. данный алгоритм во многих случаях предлагает эффективную атаку на неизвестный метод криптографического преобразования. Так, к примеру, детали криптоалгоритма Skipjack составляют государственную тайну и засекречены Агентством национальной безопасности США. Однако аппаратная реализация криптоалгоритма в виде микросхемы Clipper входит в состав многих коммерческих систем шифрования, например, Fortrezza PCMCIA. Рассмотрим криптоаналитическую атаку на подобный алгоритм.

Предположим, что ключи хранятся в памяти с асимметричной динамикой сбоев, т.е переход бита из значения 1 в 0 происходит с вероятностью, отличной от вероятности перехода из 0 в 1 (примером может служить EEPROM-память). Также мы предположим, что мы можем вводить в криптографическое устройство некоторый открытый текст m и в результате шифрования на ключе k, хранящемся в энергонезависимой памяти, получить на выходе шифротекст c.

Реализация атаки:

1. неизвестный секретный ключ k, хранящийся в памяти, используется для шифрования фиксированного открытого текста m_0. Далее устройство подвергается облучению, в результате которого некоторые биты ключа меняют свое значение с 1 на 0. В таком случае на выходе мы имеем серию шифротекстов на изменяющихся ключах: c_0, c_1, …, c_f. Изменение шифротекста – переход от c_i к c_{i+1} обусловлено направленной инверсией некоторых битов ключа (1→0). Иными словами, заменой k_i на k_{i+1}, $k_i \neq k_{i+1}$; Поскольку $wt(k) \approx n/2$, где n-кол-во бит ключа, можем предположить, что $f \approx n/2$ и c_f – результат шифрования m_0 на нулевом ключе. Это явный признак вырождения ключа.
2. теперь мы можем восстановить исходный ключ k_0:
   пусть нам известен некоторый ключ k_{i+1}, предположительно получаемый (согласно модели сбоев) из k_i направленной заменой какого-либо бита, $wt(k_{i+1}(+)k_i)=1$). Тогда для раскрытия ключа k_i достаточно зашифровать m_0 на последовательности ключей k'_{i+1}, получаемых из k_{i+1} путем замены единиц на нули. Если результат шифрования равен c_i – ключ восстановлен.

Таким образом, трудоемкость метода оценивается как $O(n^2)$ операций шифрования.

Использованная литература:

1. Современная криптография. А.Л. Чмора (“Гелиос АРВ” Москва 2001г.).
2. Differential Cryptoanalysis of the Data Encryption Standard. Biham E., Shamir A. (Spring-Verlag 1999y).
3. Differential Cryptoanalysis of DES-like Cryptosystems. Biham E., Shamir A. (Spring-Verlag 1991y).