**СРСП 8.** Атака «грубой силы», метод «встречи по середине». Анализ слабых ключей.

**Форма и содержание отчета**

Отчет должен содержать:

1. Титульный лист установленного образца.
2. Описание комплексного поиск возможных методов доступа.
3. Сравнительный анализ рассмотренных терминалов защищенной информационной системы.
4. Перечень использованных информационных источников.

Отчет выполняется на листах формата А4 рукописным или печатным способом. При рукописном оформлении отчета текст следует излагать четким почерком чернилами черного цвета.

**Атака «грубой силы» или метод полного перебора.** Если известен алгоритм шифрования и есть хотя бы одна пара открытый – шифрованный текст, то самым естественным способом анализа, который сразу приходит в голову, является последовательное опробование всех возможных вариантов ключа, которые могли быть использованы. Опробование производят до тех пор, пока зашифрование открытого текста на очередном ключе не приведет к получению имеющегося шифрованного сообщения. Такой способ анализа в разных источниках литературы имеет разные названия, например «Метод полного перебора» или «Метод грубой силы» или «Метод атаки в лоб» или «Brut-force атака». У этого метода есть одно неоспоримое преимущество: рано или поздно искомый ключ будет найден и для этого будет необходим минимальный набор данных. Быстрота нахождения ключа будет зависеть от длины используемого секретного ключа и от вычислительной мощи, которая есть в наличии у аналитика. А также от доли везения. Ведь может случиться так, что искомый ключ встретится одним из первых.

Вместе с тем нам известно, что одним из важных свойств информации является ее своевременность. Поэтому применение метода полного перебора на практике легко реализуется, но как правило не используется. Так, например, когда разрабатывался алгоритм шифрования DES, длина его фактического секретного ключа была определена в 56 бит. То есть для того, чтобы перебрать все возможные варианты секретных ключей, необходимо было сделать 256 опробований. С помощью имевшихся в то время вычислительных средств это можно было бы сделать за несколько десятков лет! Конечно, с той поры как был разработан алгоритм шифрования DES, в развитии вычислительной технике произошел огромный скачок и вычислительные мощности возросли в тысячи раз. Сегодня с использованием мощных вычислительных кластеров задача по поиску секретного ключа для алгоритма DES может быть решена за несколько минут. В связи с тем, что вычислительная мощность с каждым днем неумолимо растет, стандарт DES был заменен на новый стандарт AES (Advanced Encryption Standard), где длина секретного ключа возросла до 128 бит. Так или иначе, в криптографии принято время анализа с помощью метода полного перебора считать эталонным. Что это означает? Это значит, что если аналитику удастся провести анализ алгоритма шифрования быстрее, чем это можно сделать с помощью полного перебора, то данный алгоритм шифрования будет считаться уязвимым, в связи с чем его использовать для шифрования данных будет нецелесообразно.

Задача поиска секретного ключа шифрования методом полного перебора хорошо распараллеливается и легко может быть реализована для многопроцессорных вычислительных систем.

**Метод встречи посередине.** Метод встреча посередине применим к алгорит-

мам шифрования, в которых используется два различных ключа К. Это может быть достигнуто в том случае, если секретные подключи появляются с какой-то периодичностью, или, например, если было произведено двойное зашифрование данных, то есть сначала данные зашифровали на одном ключе К1, а затем полученный результат шифрования еще раз зашифровали на другом секретном ключе К2.

Пусть нам известна пара открытый – закрытый текст, зашифрованная подобным образом. В этом случае, необходимо произвести зашифрование открытого текста на всех возможных значениях ключа К1. Параллельно с этим необходимо произвести дешифрование закрытого текста на всех возможных значениях ключа К2. Та пара ключей (К1, К2), для которой результат шифрования открытого текста и результат дешифрования закрытого текста совпадут, и будет являться искомой.

Как видно из объяснений, анализ на основе метода «встреча посередине» может быть распараллелен и реализован с использованием распределенных многопроцессорных вычислений. В качестве примера работы метода можно рассмотреть варианты анализа двойного алгоритма DES или, например, анализа алгоритма ГОСТ 28147-89, в котором один и тот же ключ фактически используется четыре раза.

**Анализ слабых ключей**.

Одной из причин ненадежности криптосистем является использование слабых ключей. Фундаментальное допущение криптоанализа, которое впервые сформулировал Огюст Кирхгоф, состоит в том, что секретность сообщения всецело зависит от ключа: механизм шифрования, кроме ключа, известен противнику. Секретность алгоритма не является большим препятствием – для определения типа программно реализованного криптографического алгоритма требуется лишь несколько дней инженерного анализа исполняемого кода. «Слабый ключ» – это ключ, не обеспечивающий достаточного уровня защиты или использующий в шифровании закономерности, которые могут быть выявлены. Алгоритм шифрования не должен иметь слабых ключей. Если это невозможно, то количество слабых ключей должно быть минимальным, чтобы уменьшить вероятность случайного выбора одного из них; все слабые ключи должны быть известны заранее, чтобы их можно было отбраковать в процессе создания ключа.

Генераторы случайных чисел – еще один источник угрозы для стойкости криптосистемы. Если для генерации ключей используется криптографический слабый алгоритм, то 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 , то в процессе его генерации с одинаковой вероятностью должен получаться любой из 2n возможных вариантов. Генерация ключей для асимметричных криптосистем – процедура более сложная, так как ключи, применяемые в таких системах, должны обладать определенными математическими свойствами. Например, в случае системы RSA модуль шифрования представляет собой произведение двух больших простых чисел.

Для алгоритма des, из-за того, что первоначальный ключ изменяется при получении подключа для каждого этапа алгоритма, определенные первоначальные ключи являются слабыми. Вспомните, первоначальное значение расщепляется на две половины, каждая из которых сдвигается независимо. Если все биты каждой половины равны 0 или 1, то для всех этапов алгоритма используется один и тот же ключ. Это может произойти, если ключ состоит из одних 1, из одних 0, или если одна половина ключа состоит из одних 1, а другая - из одних 0. Кроме того, у два слабых ключа обладают другими свойствами, снижающими их безопасность.

Четыре слабых ключа показаны в шестнадцатиричном виде в Табл. 1. (Не забывайте, что каждый восьмой бит - это бит четности.)

Табл. 1.
Слабые ключи DES

|  |  |
| --- | --- |
| Значение слабого ключа (с битами четности) | Действительный ключ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0101 | 0101 | 0101 | 0101 | 0000000 0000000 |
| 1F1F | 1F1F | 0E0E | 0E0E | 0000000 FFFFFFF |
| E0E0 | E0E0 | F1F1 | F1F1 | FFFFFFF 0000000 |
| FEFE | FEFE | FEFE | FEFE | FFFFFFF FFFFFFF |

Кроме того, некоторые пары ключей при шифровании переводят открытый текст в идентичный шифротекст. Иными словами, один из ключей пары может расшифровать сообщения, зашифрованные другим ключом пары. Это происходит из-за метода, используемого DES для генерации подключей - вместо 16 различных подключей эти ключи генерируют только два различных подключа. В алгоритме каждый из этих подключей используется восемь раз. Эти ключи, называемые полуслабыми ключами, в шестнадцатиричном виде приведены в Табл. 2.

Табл. 2.
Полуслабые пары ключей DES

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 01FE | 01FE | 01FE | 01FE | и | FE01 | FE01 | FE01 | FE01 |
| 1FE0 | 1FE0 | 0EF1 | 0EF1 | и | E01F | E01F | F10E | F10E |
| 01E0 | 01E0 | 01F1 | 01F1 | и | E001 | E001 | F101 | F101 |
| 1FFE | 1EEE | 0EFE | 0EFE | и | FE1F | FE1F | FE0E | FE0E |
| 011F | 011F | 010E | 010E | и | 1F01 | 1F01 | 0E01 | 0E01 |
| E0FE | E0FE | F1FE | F1FE | и | FEE0 | FEE0 | FEE1 | FEE1 |

Ряд ключей генерирует только четыре подключа, каждый из которых четыре раза используется в алгоритме. Эти возможно слабые ключи перечислены в Табл. 2.

Табл. 3.
Возможно слабые ключи DES

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1F | 1F | 01 | 01 | 0E | 0E | 01 | 01 |  | E0 | 01 | 01 | E0 | F1 | 01 | 01 | F1 |
| 01 | 1F | 1F | 01 | 01 | 0E | 0E | 01 |  | FE | 1F | 01 | E0 | FE | 0E | 01 | F1 |
| 1F | 01 | 01 | 1F | 0E | 01 | 01 | 0E |  | FE | 01 | 1F | E0 | FE | 01 | 0E | F1 |
| 01 | 01 | 1F | 1F | 01 | 01 | 0E | 0E |  | E0 | 1F | 1F | E0 | F1 | 0E | 0E | F1 |
| E0 | E0 | 01 | 01 | F1 | F1 | 01 | 01 |  | FE | 01 | 01 | FE | FE | 01 | 01 | FE |
| FE | FE | 01 | 01 | FE | FE | 01 | 01 |  | E0 | 1F | 01 | FE | F1 | 0E | 01 | FE |
| FE | E0 | 1F | 01 | FE | F1 | 0E | 01 |  | E0 | 01 | 1F | FE | F1 | 01 | 0E | FE |
| E0 | FE | 1F | 01 | F1 | FE | 0E | 01 |  | FE | 1F | 1F | FE | FE | 0E | 0E | FE |
| FE | E0 | 01 | 1F | FE | F1 | 01 | 0E |  | 1F | FE | 01 | E0 | 0E | FE | 01 | F1 |
| E0 | FE | 01 | 1F | F1 | FE | 01 | 0E |  | 01 | FE | 1F | E0 | 01 | FE | 0E | F1 |
| E0 | E0 | 1F | 1F | F1 | F1 | 0E | 0E |  | 1F | E0 | 01 | FE | 0E | F1 | 01 | FE |
| FE | FE | 1F | 1F | FE | FE | 0E | 0E |  | 01 | E0 | 1F | FE | 01 | F1 | 0E | FE |
| FE | 1F | E0 | 01 | FE | 0E | F1 | 01 |  | 01 | 01 | E0 | E0 | 01 | 01 | F1 | F1 |
| E0 | 1F | FE | 01 | F1 | 0E | FE | 01 |  | 1F | 1F | E0 | E0 | 0E | 0E | F1 | F1 |
| FE | 01 | E0 | 1F | FE | 01 | F1 | 0E |  | 1F | 01 | FE | E0 | 0E | 01 | FE | F1 |
| E0 | 01 | FE | 1F | F1 | 01 | FE | 0E |  | 01 | 1F | FE | E0 | 01 | 0E | FE | F1 |
| 01 | E0 | E0 | 01 | 01 | F1 | F1 | 01 |  | 1F | 01 | E0 | FE | 0E | 01 | F1 | FE |
| 1F | FE | E0 | 01 | 0E | FE | F0 | 01 |  | 01 | 1F | E0 | FE | 01 | 0E | F1 | FE |
| 1F | E0 | FE | 01 | 0E | F1 | FE | 01 |  | 01 | 01 | FE | FE | 01 | 01 | FE | FE |
| 01 | FE | FE | 01 | 01 | FE | FE | 01 |  | 1F | 1F | FE | FE | 0E | 0E | FE | FE |
| 1F | E0 | E0 | 1F | 0E | F1 | F1 | 0E |  | FE | FE | E0 | E0 | FE | FE | F1 | F1 |
| 01 | FE | E0 | 1F | 01 | FE | F1 | 0E |  | E0 | FE | FE | E0 | F1 | FE | FE | F1 |
| 01 | E0 | FE | 1F | 01 | F1 | FE | 0E |  | FE | E0 | E0 | FE | FE | F1 | F1 | FE |
| 1F | FE | FE | 1F | 0E | FE | FE | 0E |  | E0 | E0 | FE | FE | F1 | F1 | FE | FE |

Прежде, чем порицать DES слабые ключи, обратите внимание на то, что эти 64 ключа - это крошечная часть полного набора из 72057594037927936 возможных ключей. Если вы выбираете ключ случайно, вероятность выбрать один из слабых ключей пренебрежимо мала. Если вы настоящий параноик, можете всегда проверять "на слабость" сгенерированный ключ. Некоторые думают, что нечего и беспокоиться на этот счет. Другие утверждают, что проверка очень легка, почему бы ее и не выполнить.