Лабораторные работы по курсу Защита информации (часть 1).

МГТУ им. Баумана
Кафедра ИУ-9
Юдаев П.В., 2014

Лабораторная работа №3. Поиск коллизии хэш функции, основанной на блочном шифре. Реализация атаки на блочный шифр AES в режиме CBC при наличии оракула правильного окончания блока.

1 Задание

а) На лекции рассказано, что для значения хэш функции, реализованной как симметричный шифр с известным ключом, можно найти прообраз, расшифровав значение хэш функции этим шифром. Требуется найти коллизию для каждой из хэш функций на основе шифра AES-128, не используя парадокс дней рождения. Ответ для каждой задачи дать в виде одной строки: 16-байтовые значения x и y, разделенные пробелом, в HEX кодировке.
1) $H(x,y) = AES(y,x) \oplus y$
2) $H(x,y) = AES(x,x) \oplus y$
3) $H(x,y) = AES(y, x \oplus 1^{128}) \oplus y$
4) $H(x,y) = AES(x \oplus 1^{128}, x) \oplus y$
5) $H(x,y) = AES(y,y) \oplus x \oplus 1^{128}$
6) $H(x,y) = AES(y,y) \oplus x \oplus y$
7) $H(x,y) = AES(x,x) \oplus x \oplus y$
8) $H(x,y) = AES(y,x) \oplus AES(y,y) \oplus y$
9) $H(x,y) = AES(x,x) \oplus AES(x,y) \oplus x$
10) $H(x,y) = AES(x,x) \oplus AES(x, y \oplus x)$
Обозначения: $AES(y,x)$ - шифрование сообщения x ключом y шифром AES-128. $\oplus$ - побитовое сложение по модулю два.

Формат ответа одной задачи:
`00000000000000000000000000000000 ABCDEF11111111111111111111111111`

б) Дана программа или функция, которая представляет собой оракул правильного окончания блока при шифровании AES в режиме CBC с длиной блока 128 бит. Как известно, при использовании блочного шифра в режиме CBC исходный текст дополняется до дли-

ны блока. Если длина дополнения равна 1 байту, то значение этого байта равно 1 (0x01). Если длина дополнения равна 2 байтам, то значение каждого байта дополнения равно 2 (0x0202). И так далее: если длина дополнения равна 15 байтам, то они равны 0x0F...0F. Если же длина текста кратна длине блока, то добавляется еще один целый блок (16 байт) из нулей, 0x00...00.

Оракул расшифровывает присланный шифротекст, и если окончание блока правильное, возвращает $true$, иначе $false$.

Считать шифротекст из стандартного потока ввода. Первый блок шифротекста - это вектор инициализации. Напечатать в поток стандартного вывода весь исходный текст (без дополнения до длины блока). Все вводимые и выводимые данные представлены в HEX кодировке.

Пример работы программы:

Ввод (одна строка):
`000102030405060700010203040506079 56e58be27c1ff7cba8ca72f6b1a6d6f9f2de07510e062ab1683faabdb175d11`
Вывод:
`var_example_here_now`

2 Рекомендации по выполнению задания

а) Пользуясь материалами лекции, построить предположение о связи между x и y в случае коллизии. Проверить это предположение, пользуясь онлайн калькуляторами AES и XOR, или написать свой код для проверки. Предъявить пары (x_1, y_1), (x_2, y_2), значения хэш функции от которых совпадают.

Ссылки на онлайн калькуляторы шифра AES:
`http://testprotect.com/appendix/AEScalc`,
`http://seit.unsw.adfa.edu.au/staff/sites/lpb/src/AEScalc/AEScalc.html`
и калькулятор XOR:
`http://www.miniwebtool.com/bitwise-calculator/`

б) Пусть $b_{15}b_{14}...b_2b_1b_0$ - предпоследний блок шифротекста, $a_{15}a_{14}...a_2a_1a_0$ - последний блок исходного текста, дополненного до длины блока.
Значение последнего байта a_0 можно узнать следующим образом. Если значение b_0 изменить на величину $a_0 \oplus 0x01$:
$b'_0 := b_0 \oplus (a_0 \oplus 0x01)$,
то последний байт расшифрованного оракулом текста изменится так же:
$a'_0 := a_0 \oplus (a_0 \oplus 0x01) = 0x01$

и 0x01 будет верное окончание блока расшифрованного текста. Оракул вернет $true$. Для всех остальных значений b'_0 окончание блока будет (почти наверняка) неверным и оракул вернет $false$. Всего 256 возможных значений a_0. Значит, за 256 попыток найдем значение последнего байта исходного текста a_0.

Далее ищем значение предпоследнего байта a_1. Меняем значения двух последних байтов второго с конца блока шифротекста так, чтобы, если два последних байта открытого текста равны a_1a_0, блок расшифрованного оракулом текста кончался бы на 0x0202. Всего 256 возможных значений a_1. Значение a_0 уже известно, поэтому его не меняем:
$b'_1 := b_1 \oplus (a_1 \oplus 0x02)$, $b'_0 := b_0 \oplus (a_0 \oplus 0x02)$.
Чтобы узнать третий с конца байт этого блока исходного текста, меняем значения трех байтов шифротекста, и так далее. Так можно последовательно узнать весь исходный текст, не находя ключ шифра.