МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное бюджетное профессиональное образовательное учреждение
Свердловской области
«Уральский государственный колледж имени И.И. Ползунова»
Отдел информационных технологий

CRAB

Студенты:
Кайгородов А.А.
Буртасов А.О.
Группа: ЗИ-313
Преподаватель:
Соколов И.П.

Что такое Crab?

Crab — это блочный шифр, разработанный Бартом Калиски и Мэттом Робшоу из лаборатории RSA на первом семинаре FSE в 1993. Crab был разработан, чтобы продемонстрировать, как идеи хеш-функций могут быть использованы для создания быстрого шифрования. Алгоритм шифрования очень похож на MD5.

Общие сведения

Краб имеет необычно большой размер блока 8192 бит. Его создатели предложили использовать 80-битный ключ, но шифр мог использовать любой размер ключа . Авторы не указали фактическое расписание ключей , только то, что ключ используется для генерации двух больших наборов подразделов : перестановка чисел от 0 до 255 и массив из 2048 32-разрядных чисел. Блок разделен на 256 32-битных субблоков, которые переставляются в начале. Затем алгоритм делает четыре прохода по данным, каждый раз применяя одно из четырех преобразований, адаптированных из MD5 .

Алгоритм

1. Разбить X на 256 32-битных подблока: $X_0, X_1, \ldots, X_{255}$
2. Переставить подблоки X_i согласно P_i
3. *for* $r = 0$ *to* 3
   for $g = 0$ *to* 63
4. $A = X_{(4g)<<<2r}$
5. $B = X_{(4g+1)<<<2r}$
6. $C = X_{(4g+2)<<<2r}$
7. $D = X_{(4g+3)<<<2r}$
8. *for* $s = 0$ *to* 7
9. $A = A \oplus F_r(B, C, D, S_{512r+8g+s})$
10. $TEMP = D$
11. $D = C$
12. $C = B$
13. $B = A <<< 5$
14. $A = TEMP$
15. $X_{(4g)<<<2r} = A$
16. $X_{(4g+1)<<<2r} = B$
17. $X_{(4g+2)<<<2r} = C$
18. $X_{(4g+3)<<<2r} = D$
19. Переставить $X_0, X_1, \ldots, X_{255}$ для формирования зашифрованного текста

Для шифрования 1024-байтного блока X

$X <<< b$ означает левое вращение на b бит

Функции F_r применяются 8 раз при обработке каждой из 64 независимых групп в каждой итерации, они используют булевы функции f_r, который зависят от итерации. В остальном F_r одинаковы и могут быть представлены следующим образом:

$$F_r(B, C, D, S) = (B + f_r(B, C, D) + S)$$

Функции f_r совпадают с функциями из MD5:

$$\mathrm{f}_0(B, C, D) = (B \wedge C) \vee (\neg B \wedge D),$$
$$\mathrm{f}_1(B, C, D) = (B \wedge D) \vee (\neg D \wedge C),$$
$$\mathrm{f}_2(B, C, D) = B \oplus C \oplus D,$$
$$\mathrm{f}_3(B, C, D) = B \oplus (\neg C \vee D),$$

Где $\oplus, \wedge, \vee, \neg$ побитовые логические операции XOR, AND, OR и NOT соответственно, все операции по модулю 2^{32}

Дешифрование происходит по обратному алгоритму.

Генерация подключей

Генерация подключей — это задача, которая может быть решена различными способами. Начальная перестановка P может быть сгенерирована с использованием ключа K вариациями методов, представленных в Knuth; один вариант того, как массив перестановок P может быть сгенерирован из 80-битного ключа K, представлен ниже.

1. Инициализировать $K_0, K_1, \ldots, K_9$ первыми 10-ю битами K
2. $for\ i = 10\ to\ 255$
3. $\quad K_i = K_{i-2} \oplus K_{i-6} \oplus K_{i-7} \oplus K_{i-10}$
4. $for\ i = 0\ to\ 255,\ P_i = i$
5. $m = 0$
6. $for\ j = 0\ to\ i$
7. $\quad for\ i = 256\ to\ 1\ step\ -1$
8. $\quad\quad m = (K_{256-i} + K_{257-i}) mod\ i$
9. $\quad\quad K_{257-i} = K_{257-i} <<< 3$
10. $\quad$ Переставить P_m и P_{i-1}

Массив S из 2048 32-битных слов может быть сгенерирован так же, из того же 80-битного ключа или из другого ключа. Авторы предупреждают, что эти детали должны «рассматриваться как мотивационные; вполне могут быть альтернативные схемы, которые являются более эффективными и предлагают улучшенную безопасность"

Безопасность

Рассмотрим перевернутый 26-й бит (0x04000000) одного из 256 текстовых слов. Этот конкретный бит часто затрагивает только три слова в первой итерации. При следующих трех итерациях число затронутых слов будет увеличиваться в 4 раза, что приведет к $3 \times 4 \times 4 \times 4 = 192$ затронутым словам, оставив 64 слова нетронутыми. Это происходит с экспериментальной вероятностью в $0.096 = 2^{-3.4}$, что немедленно приводит к тому, что для различающей нужно не больше дюжины выбранных блоков открытого текста.

Анализ атаки с восстановлением ключа (англ. Key-recovery attack) немного сложнее из-за отрывочного характера описания генерации ключа. Если предположить, что ключ может быть эффективно восстановлен из перестановки P, авторы считают, что для атаки с восстановлением ключа не потребуется более 2^{16} выбранных блоков открытого текста и пренебрежительных вычислительных усилий.

Спасибо за внимание!!!