

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

# «МИРЭА – Российский технологический университет»

# РТУ МИРЭА

Институт искусственного интеллекта

Базовая кафедра №252 – информационной безопасности

ПРАКТИЧЕСКАЯ РАБОТА

# Тема практической работы: «IDEA OFB»

|  |  |  |
| --- | --- | --- |
| Студент группы ККСО-05-21 | Опанасюк Д.И. | *(подпись)* |
| Руководитель практической работы | старший преподаватель Плешаков А.С. | *(подпись)* |
| Работа представлена к защите | « k 2022 г. |  |
| Допущен к защите | « k 2022 г. |  |

Москва – 2022

**Оглавление**

1. Введение…………………….………………………………………………...3

2. Теоретическая часть……...…………………………………………………..4

3. Описание практической части…………………………….…………………7

4. Заключение…..………………………………………………………………14

5. Список литературы…….……………………………………………………15

**Введение**

В настоящее время используется большое количество самых разных алгоритмов блочного шифрования. Из всего многообразия мы расскажем лишь о тех алгоритмах, которые либо являются очень надежными, либо своим появлением оказали влияние на развитие криптографии. Одним из таких алгоритмов является IDEA.

Первый вариант шифра IDEA, предложенный Ксуеджа Лай и Джеймсом Масси, появился в 1990 году. Он назывался PES (Proposed Encryption Standard, предложенный стандарт шифрования). В следующем году, после демонстрации Бихамом и Шамиром возможностей дифференциального криптоанализа, авторы усилили свой шифр против такого вскрытия и назвали новый алгоритм IPES (Improved Proposed Encryption Standard, улучшенный предложенный стандарт шифрования). В 1992 году название IPES было изменено на IDEA (International Data Encryption Algorithm, международный алгоритм шифрования данных).

IDEA основывается на некоторых впечатляющих теоретических положениях и, хотя криптоанализ добился некоторых успехов в отношении вариантов с уменьшенным количеством этапов, алгоритм все еще кажется сильным.

Его сегодняшняя известность объясняется тем, что он является частью PGP.

**Теоретическая часть**

**Обзор**IDEA является блочным шифром, он работает с 64-битовыми блоками открытого текста. Длина ключа - 128 битов. Для шифрования и дешифрирования используется один и тот же алгоритм.

Как и другие, уже рассмотренные блочные шифры IDEA использует и запутывание, и рассеяние. Философия, лежащая в основе проекта, представляет собой "объединение операций из различных алгебраических групп". Смешиваются три алгебраические операции, и все они могут быть легко реализованы как аппаратно, так и программно.

Поразрядное сложение по модулю 2 (операция "исключающее ИЛИ"); операция обозначается через  .

Сложение беззнаковых целых по модулю 65536=; операция обозначается через .

Умножение целых по модулю 65537=+1, рассматриваемых как беззнаковые целые, за исключением того, что блок из 16 нулей рассматривается как ; операция обозначается через \odot.

Все операции выполняются над 16-битовыми субблоками. Эти три операции несогласованы в том смысле, что:

никакая пара из этих трех операций не удовлетворяет ассоциативному закону, например,



никакая пара из этих трех операций не удовлетворяет дистрибутивному закону, например,

.

Комбинирование этих трех операций обеспечивает комплексное преобразование входа, существенно затрудняя криптоанализ IDEA по сравнению с DES, который базируется исключительно на операции "исключающее ИЛИ".

Все эти операции (а в алгоритме используются только они, перестановки на битовом уровне не применяются) работают с 16-битовыми подблоками. (Этот алгоритм даже эффективнее на 16-битовых процессорах.)

**Описание**

Схема IDEA представлена на рисунке 7.6. 64-битовый блок данных делится на четыре 16-битовых подблока: , X_2, , . Эти четыре подблока становятся входными данными для первого этапа алгоритма. Всего в алгоритме восемь этапов. На каждом этапе четыре подблока подвергаются операциям XOR, сложениям и умножениям друг с другом и с шестью 16-битовыми подключами. Между этапами обмениваются местами второй и третий подблоки. Наконец, четыре подблока объединяются с четырьмя подключами в окончательном преобразовании.

На каждом этапе события происходят в следующей последовательности:

Перемножаются X_1 и первый подключ.

Складываются X_2 и второй подключ.

Складываются X_3 и третий подключ.

Перемножаются X_4и четвертый подключ.

Выполняется XOR над результатами этапов (1) и (3).

Выполняется XOR над результатами этапов (2) и (4).

Перемножаются результаты этапа (5) и пятый подключ.

Складываются результаты этапов (6) и (7).

Перемножаются результаты этапа (8) и шестой подключ.

Складываются результаты этапов (7) и (9).

Выполняется XOR над результатами этапов (1) и (9).

Выполняется XOR над результатами этапов (3) и (9).

Выполняется XOR над результатами этапов (2) и (10).

Выполняется XOR над результатами этапов (4) и (10).

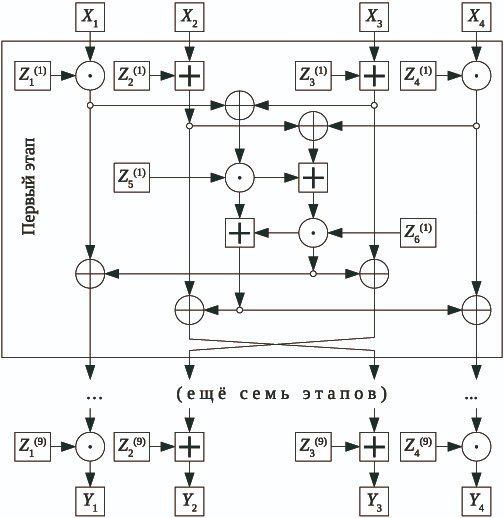


Схема алгоритма IDEA

Выходом этапа являются четыре подблока - результаты действий (11), (12), (13) и (14). Поменяйте местами два внутренних подблока (но не в последнем этапе), и вы получите исходные данные для следующего этапа.

Затем выполняется заключительный, 9-й этап преобразования:

Перемножаются  и первый подключ.

Складываются X_2 и второй подключ.

Складываются X_3 и третий подключ.

Перемножаются X_4и четвертый подключ.

Наконец четыре подблока снова соединяются, образуя шифртекст.

**Описание практической части**

Пример 1

*Выполнить первый раунд зашифрования алгоритмом IDEA. Ключ: 1D52 34BC 891C 9C9B 1CC2 4363 A32B 132C, блок открытого текста: 89C1 B11D 63F0 FF23. Ключ и открытый текст задаются шестнадцатеричными последовательностями. Каждые четыре последовательных шестнадцатеричных разряда WXYZ обозначают 16-битный блок .*

Решение.

Вначале выберем подключи первого раунда. Это первые 6 16-битных блоков ключа.

В обозначениях рисунка имеем: X_1=89C1, X_2=B11D, ,, , Z_2^{(1)}=34BC, Z_3^{(1)}=891C, Z_4^{(1)}=9C9B, Z_5^{(1)}=1CC2, Z_6^{(1)}=4363

Будем обозначать результат i-го шага в алгоритме через A_i.

Все операции будем проводить в шестнадцатеричной системе счисления. Результат применения одного раунда обозначим через X'_1, X'_2, X'_3, X'_4.

A_1 = X_1 \odot Z_1^{(1)} = 89C1\odot 1D52 = (89C1\cdot 1D52)~(\mod 10001) = ED0CA_2 = X_2 \boxplus Z_2^{(1)} = B11D \boxplus 34BC = (B11D+34BC)~(\mod 10000) = E5D9

A_3 = X_3 \boxplus Z_3^{(1)} = 5E1B \boxplus 891C = (63F0+891C)~(\mod 10000) = ED0C

A_5 = A_1 \oplus A_3 = ED0C \oplus ED0C = 0000



A_7 = A_5 \odot Z_5^{(1)} = 0000 \odot 1CC2 = (10000\cdot 1CC2)~(\mod 10001) = E33F



A_9 = A_8 \odot Z_6^{(1)} = BB16\odot 4363 = (BB06\cdot 4363)~(\mod 10001) = B418











Создание подключей  также относительно несложно. Алгоритм использует всего 52 подключа (по шесть для каждого из восьми циклов и еще четыре для заключительного этапа). Сначала 128-битовый ключ делят на восемь 16-битовых подключей. Это -- первые восемь подключей для алгоритма (шесть подключей -- для первого цикла и первые два подключа -- для второго цикла). Затем 128-битовый ключ циклически сдвигается влево на 25 бит и снова делится на восемь подключей. Первые четыре из них используют во втором цикле; последние четыре -- в третьем цикле. Ключ снова циклически сдвигается влево еще на 25 бит для получения следующих восьми подключей и т.д., пока выполнение алгоритма не завершится.

Расшифрование осуществляют аналогичным образом, за исключением того, что порядок использования подключей --- обратный, и ряд значений подключей заменяется на обратные значения. Порядок использования подключей при зашифровании и расшифровании приведён в таблице 7.10. Через -a обозначается противоположный к a по модулю 65536 элемент, через a^{-1} --- обратный по умножению по модулю 65537



Таблица 1

Пример 2

Имея ключ 1D52 34BC 891C 9C9B 1CC2 4363 A32B 132C для алгоритма IDEA, найти подключи четвертого раунда расшифровки. Порядок записи бит числа --- от младшего к старшему.

Отметим, что существует два способа записи бит в памяти ЭВМ: от старшего к младшему (Big-Endian) и от младшего к старшему (Little-Endian). Big-Endian использовался в ранних архитектурах ЭВМ, и для сохранения обратной совместимости остался в сетевых протоколах и некоторых форматах файлов. Little-Endian используется практически во всех современных архитектурах, поэтому мы используем его в нашей задаче.

Решение. Согласно таблице 1, нам нужно найти подключи {Z_1^{(6)}}^{-1},-Z_2^{(6)},-Z_3^{(6)},{Z_4^{(6)}}^{-1},, Z_6^{(5)}.

Поэтому необходимо сначала найти необходимые нам подключи для 5-го и 6-го раундовZ_1^{(6)},,Z_3^{(6)},,Z_5^{(5)},Z_6^{(5)}

Обозначим i-й 16-битный блок ключа, n раз циклически сдвинутого влево на 25 бит, через K_i^{(n)}. Тогда данный нам исходный ключ состоит из блоков K_1^{(0)},\ K_2^{(0)},\ \ldots,\ K_8^{(0)}нужный нам подключ Z_i^{(j)}равен некоторому блоку K_m^{(n)}. Выпишем это соответствие, записывая индексы друг под другом:

n: 000000001111111122222222333333334444444455555555

m: 123456781234567812345678123456781234567812345678

j: 111111222222333333444444555555666666777777888888

i: 123456123456123456123456123456123456123456123456

В таблице отмечены требуемые нам подключи зашифрования. Имеем:

Z_1^{(6)}=K_7^{(3)},\ Z_2^{(6)}=K_8^{(3)}, Z_3^{(6)}=K_1^{(4)}, Z_4^{(6)}=K_2^{(4)}, Z_5^{(5)}=K_5^{(3)},  Z_6^{(5)}=K_6^{(3)}.

Для выполнения циклического сдвига запишем наш ключ в бинарном виде. Для этого переведём каждый шестнадцатеричный блок в двоичную систему счисления:

1D52_{16} = 0001110101010010_2; \quad 34BC_{16}=0011010010111100_2;\quad \ldots

и выпишем блоки последовательно, начиная от младшего бита к старшему:

0100101010111000\quad 0011110100101100\quad 0011100010010001\quad 1101100100111001 \\
    0100001100111000\quad 1100011011000010\quad 1101010011000101\quad 0011010011001000Для получения блока K_5^{(3)}нужно сдвинуть ключ на 3\* 25=75 бит влево. В полученной битовой последовательности биты с 1 по 16 --- блок K_1^{(3)}, c 17 по 32 --- блок K_2^{(3)}, ..., с 65 по 80 --- блок K_5^{(3)}. Таким образом, в исходном ключе блок K_5^{(3)} располагается с 75+65-128=12 по 75+80-128=28 биты: 1100000111101001.

Итак, K_5^{(3)} = 1001011110000011_2 = 38787_{10}=9383_{16}.

Следующие 16 бит (0110000111000100) составляют ключK_6^{(3)} = 0010001110000110_2 = 9094_{10}=2386_{16}. Следом идут K_7^{(3)} = 1001001101110001_2 = 37745_{10}=9371_{16}, K_8^{(3)}=1001100001010011_2 = 38995_{10}=9853_{16}.

Для получения K_1^{(4)}$ и $K_2^{(4)}нужно сдвинуть ключ на 100 бит влево. В полученной битовой последовательности биты с 1 по 16 --- блок K_1^{(4)}, а с 17 по 32 --- блок K_2^{(4)}. Имеем:

K_1^{(4)} = 1100101000110010_2 = 51762_{10}=CA32_{16},

K_2^{(4)} = 0010000100110010_2 = 8498_{10}=2132_{16}.

K_1^{(4)} = 1100101000110010_2 = 51762_{10}=CA32_{16},



Теперь мы можем найти подключи для расшифрования:

{Z_1^{(6)}}^{-1} = {K_7^{(3)}}^{-1} = 37745^{-1}~\mod 65537 = 53013_{10} = CF15_{16};

-Z_2^{(6)} = -K_8^{(3)} = -38995~\mod 65536 = 26541_{10}=67AD_{16};







Z_6^{(5)} = K_6^{(3)}=2386_{16}.