Контрольная работа
Вариант №4.

Роман Астраханцев, СКБ-171

21 февраля 2022 г.

Задача 1

Дана сеть Фейстеля, состоящая из 8 итераций, с длиной блока $n = 128$ бит. Из мастер ключа $K = (K_1, K_2, K_3, K_4)$, где $K_1, \dots, K_4 \in V_{64}$, итерационные ключи (на итерациях $1, 2, 3, \dots, 8$) получаются вырабаютываются как последовательность $K_3, K_2, K_4, K_1, K_3, K_2, K_4, K_1$. Обозначим за $E : V_{128} \times V_{256} \to V_{128}$ алгоритм зашифрования.

Описать трудоемкость, вероятность успеха, затраты по памяти и объём материала для методов тотального опробования и слайд-атаки.

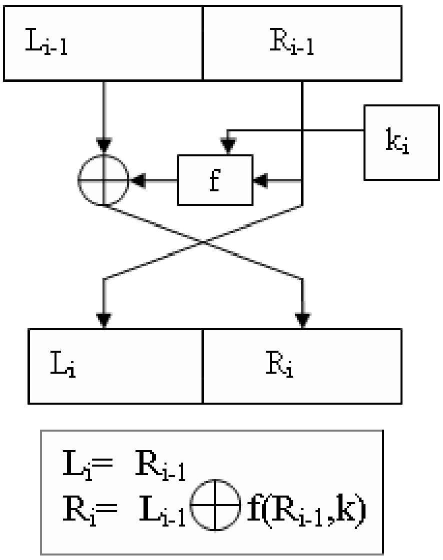

Рис. 1: Раунд сети Фейстеля

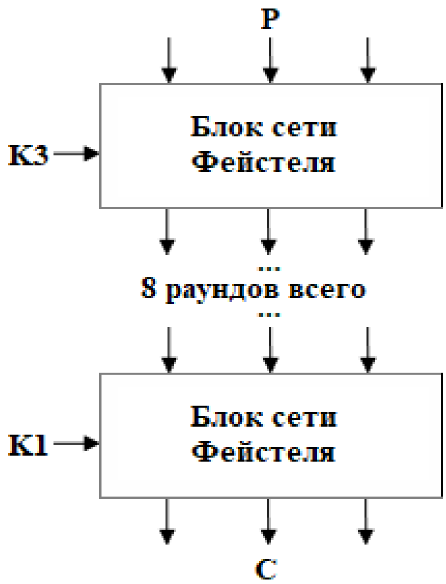

Рис. 2: Шифр из задачи

Метод тотального опробования

Для начала определим количество материала, необходимое для однозначного опредления ключа. Поскольку одной на паре (P, C) открытого и шифрованного текста, где $P, C \in V_{128}$, можно отбраковать 2^{128} ключей, то потребуется $\lceil \frac{256}{128} \rceil = 2$ различные пары: $(P_1, C_1), (P_2, C_2)$. Будем дальше считать, что они нам даны.

Алгоритм 1: Метод тотального опробования

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2)$
Выход: Ключ шифрования K

Для каждого $\tilde{k} \in V_{256}$
    Вычислить $B_1 = E(P_1, \tilde{k})$
    Если $B_1 = C_1$, **то**
        Вычислить $B_2 = E(P_2, \tilde{k})$
        Если $B_2 = C_2$, **то**
            Закончить алгоритм и вернуть $\tilde{k}$

Трудоёмокстьь Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M в битах. Тогда имеем

$$Q = 2^{256} + 2^{128} + 1 \approx 2^{256}$$

$$M = \alpha,$$

где α - количество памяти, необходимое для хранения локальных переменных алгоритма.

Вероятность успеха алгоритма $p = 1$, поскольку алгоритм гарантированно находит ключ шифрования.

Слайд-атака

Заметим, что алгоримт зашфирования E представим как $E = G \circ G$, где $G : V_{128} \times V_{256} \to V_{128}$ – работа первых 4 раундов сети Фейстеля представленного в задаче шифра. Точно так же, как и в методе тотального опробования, после нахождения слайд-пары необоходимо будет доопробовать найденный ключ. В общем итоге для восстановления ключа нам потребуется $\lceil \frac{256}{128} \rceil = 2$ различные пары $(P_1, C_1), (P_2, C_2)$ открытого и шифрованного текста $(P_i, C_i \in V_{64})$. Будем дальше считать, что они нам даны. Также будем считать, что нам дана возможность по любому открытому тексту получить его зашифрованную версию, иными словами для любого открытого текста P мы можем вычислить $E(P, K)$ даже не зная K (это может быть заранее полученный корпус из пар открытый-закрытый тексты).

Алгоритм 2: Метод скольжения

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2)$

Выход: Ключ шифрования K

1. Принять $i = 1$

2. **Пока** *ключ не найден или* $i > 2^{64}$

3. $i = i + 1$

4. Выберем случайно $P \in V_{128}$ – открытый текст

5. Посчитаем $C = E(P, K)$

6. Выберем случайно $P' \in V_{128}$ – другой открытый текст

7. Посчитаем $C' = E(P', K)$

8. **Если** *пары* (P, C) *и* (P', C') *совпали,* **то**

9. Перейти на новую итерацию цикла

10. Решим уравнение $P' = G(P)$ и результат занесём в K_{first}

11. Решим уравнение $C' = G(C)$ и результат занесём в K_{last}

12. **Если** $K_{first} = K_{last}$, **то**

13. Доопробуем ключ K_{first} на парах $(P_1, C_1), (P_2, C_2)$ и в случае успеха вернём ключ K_{first}

Алгоритм 2 был сформулирован как вероятностный, чтобы продемонстрировать его основные характеристики. Детеременированная версия алгоритма (вероятность успеха которой равна 1) легко получается заменой случайного выбора на перебор всевозможных значений.

Возвращаясь к рассуждениям о получении шифртекста по открытому тексту, стоит заметить, что данная формулировка алогритма используется тот факт, что объём построенного заранее корпуса данных должен быть не менее 2^{64} пар открытый-закрытый тексты. В таком случае согласно парадоксу дней рождений вероятность успеха алгоритма $p \gg 0.9999$.

Трудоёмоксть Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M в битах. Тогда имеем

$$Q = 2^{64} \cdot 2q$$

$$M = \alpha,$$

где q – это сложность решения уравнения $P' = G(P)$ относительно ключа k, α - количество памяти, необходимое для хранения локальных переменных алгоритма.

Задача 2

Дан алгоритма ГОСТ 28147-89 («Магма»). Обозначим через $H_i = F(X, K_i)$ - результат зашифрования $X \in V_{64}$ одной итерацией алгоритма ГОСТ на ключе $K_i \in V_{32}, i \in \overline{0,7}$. Через T обозначим финальную перестановку алгоритма ГОСТ. Для преобразований H_i и T справедливы следующие равенства.

$$H_i^{-1} = TH_iT$$
$$T^2 = TT = E$$

Зашифрование алгоритмом ГОСТ выглядит следующим образом

H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
H_7	H_6	H_5	H_4	H_3	H_2	H_1	$H_0 T$

Рис. 3: Схематичная работа алгоритма ГОСТ

Описать трудоемкость, вероятность успеха, затраты по памяти и объём материала для методов Исобе и Динура-Данкельмана-Шамира.

Метод Исобе

Для начала определим количество материала, необходимое для однозначного опредления ключа. Поскольку одной на паре (P, C) открытого и шифрованного текста, где $P, C \in V_{64}$, можно отбраковать 2^{64} ключей, то потребуется $\lceil \frac{256}{64} \rceil = 4$ различные пары: $(P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)$. Будем дальше считать, что они нам даны.

Теперь зафиксируем свойство алгоритма ГОСТ, которое поможет нам в построении эффективного алгоритма получения ключа. Пусть (X, Y) – пара входа-выхода на 4 итерациях алгоритма ГОСТ, а $K_i, K_{i+1}, K_{i+2}, K_{i+3} \in V_{32}$ – итерационные ключи этих 4 итераций.

Свойство 1 (Четырёх операций)**.** При известных (X, Y) и при фиксации ключей K_i, K_{i+1} (или K_{i+2}, K_{i+3}) конкретными значениями, два других ключа определяются однозначно.

Будем считать, что нам дана возможность по любому открытому тексту получить его зашифрованную версию, иными словами для любого открытого текста P мы можем вычислить $E(P, K)$ даже не зная K (это может быть заранее полученный корпус из пар открытый-закрытый тексты).

Обозначим за $F_K^{[i,j]}(P)$ результат зашифрования на ключе K алгоритмом ГОСТ, начиная с итерации с номером i, и заканчивая итерацией с номером j $(1 \le i \le j \le 32)$ текста P.

Алгоритм 3: Метод Исобе

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)$

Выход: Ключ шифрования K

Принять $i = 1$

Пока *ключ не найден или* $i > 2^{32}$

    $i = i + 1$

    Выберем случайно $P \in V_{64}$ – открытый текст

    Посчитаем $C = E(P, K)$

    Для каждого $(S, T) \in V_{128}$

        `/* тут` S `и` T `- это внутренние состояния после 4 и 12 итераций соответсвенного */`

        Для каждого $(K_4, K_5) \in V_{64}$

            По свойству 1 находим $(K_6, K_7) \in V_{64}$ по известным T, C, K_4, K_5

            Обозначаем $K' = (K_4, K_5, K_6, K_7)$

            Вычисляем $V = F_{K'}^{[5,8]}(S)$

            Заносим в память по адресу V значение ключа K'

        Для каждого $(K_0, K_1) \in V_{64}$

            По свойству 1 находим $(K_2, K_3) \in V_{64}$ по известным P, S, K_0, K_1

            Обозначаем $K'' = (K_0, K_1, K_2, K_3)$

            Вычисляем $U = F^{-1}{}_{K''}^{[9,12]}(T)$

            Извлекаем из памяти по адресу U ключ K'

            Обозначаем $K = (K'', K') \in V_{256}$

            Доопробуем ключ K на парах $(P_i, C_i), i \in \overline{1,4}$

            Если *доопробование успешно,* **то**

                Вернуть ключ K и завершить алгоримт

Алгоритм 3 был сформулирован как вероятностный, чтобы продемонстрировать его основные характеристики. Детеременированная версия алгоритма (вероятность успеха которой равна 1) легко получается заменой случайного выбора на перебор всевозможных значений.

Возвращаясь к рассуждениям о получении шифртекста по открытому тексту, стоит заметить, что данная формулировка алогритма используется тот факт, что объём построенного заранее корпуса данных должен быть не менее 2^{32} пар открытый-закрытый тексты. Вероятность попадания в неподвижную точку преобразования HTH^{-1} равна 2^{-32}. Из-за этого при наличии корпуса размером 2^{32} мы в среднем попадём в 1 неподвижную точку и в среднем доопробуем ровно 1 ключ.

Трудоёмоксть Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M будем измерять в битах. Тогда имеем

$$Q = 2^{32} \cdot 2^{128} \cdot (2^{64} + 2^{64}) = 2^{225}$$

$$M = 2^{64} \cdot 128 + \alpha = 2^{71} + \alpha,$$

где α - количество памяти, необходимое для хранения локальных переменных алгоритма.

Вероятность успеха (наличия на доступном материале неподвижной точки) равна

$$p = 1 - (1 - \frac{1}{2^{32}})^{2^{32}} \approx 1 - e^{-1} \approx 0.63.$$

Методы Динура-Данкельмана-Шамира

В работе Динура, Данкельмана и Шамира были предложены модификации метода Исобе. В основе их лежит использование метода согласования для одной пары открытого-шифрованного текста не на 16 итерациях, как в методе Исобе, а для двух пар открытого-шифрованного текста на 8 итерациях.

Обозначим за $F_K^{[i,j]}(P)$ результат зашифрования на ключе K алгоритмом ГОСТ, начиная с итерации с номером i, и заканчивая итерацией с номером j $(1 \leq i \leq j \leq 32)$ текста P.

Для начала опишем вспомогательный алгоритм, который позволяет восстановить ключ по известным парам (X, Y) и (X', Y') - входам в выходам на первых 8 итерациях алгоритма ГОСТ. На этапе доопробования нам потребуется дополнительно 2 пары $(P_1, C_1), (P_2, C_2)$

Алгоритм 4: Метод Динура-Данкельмана-Шамира по внутренним состояниям

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2)$
Входы в выходы на первых 8 итерациях алгоритма ГОСТ (X, Y) и (X', Y')

Выход: Ключ шифрования K

1. **Для каждого** $I \in V_{64}$
2. **Для каждого** $(K_0, K_1) \in V_{64}$
3. По свойству 1 находим $(K_2, K_3) \in V_{64}$ по известным X, I, K_0, K_1
4. Обозначаем $K' = (K_0, K_1, K_2, K_3)$
5. Вычисляем $I' = F_{K'}^{[1,4]}(X')$
6. Заносим в память по адресу I' значение ключа K'
7. **Для каждого** $(K_4, K_5) \in V_{64}$
8. По свойству 1 находим $(K_6, K_7) \in V_{64}$ по известным I, Y, K_4, K_5
9. Обозначаем $K'' = (K_4, K_5, K_6, K_7)$
10. Вычисляем $I'' = {F^{-1}}_{K''}^{[5,8]}(Y')$
11. Извлекаем из ячеки по адресу I'' ключ K'
12. Обозначаем $K = (K', K'') \in V_{256}$
13. Доопробуем ключ K на парах $(P_i, C_i), i \in \overline{1,2}$
14. **Если** *доопробование успешно,* **то**
15. Вернуть ключ K и завершить алгоримт.

Трудоёмкость Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M будем измерять в битах. Тогда имеем

$$Q = 2^{64} \cdot \left(2^{64} \cdot \left(\frac{1}{8} + \frac{1}{8}\right) + 2^{64} \cdot \left(\frac{1}{8} + \frac{1}{8}\right) + 2^{64}\right) = 1.5 \cdot 2^{128}$$

$$M = 2^{64} \cdot 128 + \alpha = 2^{71} + \alpha,$$

где α - количество памяти, необходимое для хранения локальных переменных алгоритма.

Вероятность успеха алгоритма $p = 1$, поскольку алгоритм гарантированно находит ключ шифрования, при известных промежуточных стотяниях X, X', Y, Y'.

Модификация 1

Первая модификация использует возможность того, что открытый текст P является неподвижной точкой первых 8 итераций алгоритма ГОСТ.

Будем считать, что нам дана возможность по любому открытому тексту получить его зашифрованную версию, иными словами для любого открытого текста P мы можем вычислить $E(P, K)$ даже не зная K (это может быть заранее полученный корпус из пар открытый-закрытый тексты).

Алгоритм 5: Метод Динура-Данкельмана-Шамира

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2)$
Выход: Ключ шифрования K

Принять $i = 1$
Пока *ключ не найден или* $i > 2^{64}$
    $i = i + 1$
    Выберем случайно $P \in V_{64}$ – открытый текст
    Посчитаем $C = E(P, K)$
    Обозначим $\widetilde{C}$ перестановку 32-битных полублоков C
    Обозначим $\widetilde{P}$ перестановку 32-битных полублоков P
    Обозначим $(X, Y) = (P, P)$
    Обозначим $(X', Y') = (\widetilde{C}, \widetilde{P})$
    Применим алгоритм 4 на $(P_1, C_1), (P_2, C_2)$ и $(X, Y), (X', Y')$
    Если *ключ был найден,* **то**
        Вернуть ключ, полученный из алгоритма 4 и завершить алгоримт.

Трудоёмкость Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M будем измерять в битах. Тогда имеем

$$Q = 2^{64} \cdot 1.5 \cdot 2^{128} = 1.5 \cdot 2^{192}$$

$$M = 2^{64} \cdot 128 + \alpha = 2^{71} + \alpha,$$

где α - количество памяти, необходимое для хранения локальных переменных алгоритма.

Возвращаясь к рассуждениям о получении шифртекста по открытому тексту, стоит заметить, что данная формулировка алогритма используется тот факт, что объём построенного заранее корпуса данных должен быть не менее 2^{64} пар открытый-закрытый тексты. Вероятность попадания в неподвижную точку преобразования H равна 2^{-64}. Из-за этого при наличии корпуса размером 2^{64} мы в среднем попадём в 1 неподвижную точку.

Вероятность успеха (наличия на доступном материале неподвижной точки) равна

$$p = 1 - (1 - \frac{1}{2^{64}})^{2^{64}} \approx 1 - e^{-1} \approx 0.63.$$

Модификация 2

Первая модификация использует возможность того, что шифрованный текст является неподвижной точкой первых преобразования HTH^{-1}. В таком случае у нас имеется соответсвие (P, C) между между входом и выходом на 16 итерациях алгоритма ГОСТ.

Будем считать, что нам дана возможность по любому открытому тексту получить его зашифрованную версию, иными словами для любого открытого текста P мы можем вычислить $E(P, K)$ даже не зная K (это может быть заранее полученный корпус из пар открытый-закрытый тексты).

Алгоритм 6: Метод Динура-Данкельмана-Шамира

Вход: Пары открытого и шифрованного текста $(P_1, C_1), (P_2, C_2)$
Выход: Ключ шифрования K

1. Принять $i = 1$
2. **Пока** *ключ не найден или* $i > 2^{32}$
3. $\quad$ $i = i + 1$
4. $\quad$ Выберем случайно $X \in V_{64}$ – открытый текст
5. $\quad$ Посчитаем $Y = E(X, K)$
6. $\quad$ **Для каждого** $Z \in V_{64}$
7. $\quad\quad$ Применим алгоритм 4 на $(P_1, C_1), (P_2, C_2)$ и $(X, Z), (Z, Y)$
8. $\quad\quad$ **Если** *ключ был найден,* **то**
9. $\quad\quad\quad$ Вернуть ключ, полученный из алгоритма 4 и завершить алгоримт.

Трудоёмкость Q этого алгоритма будем измерять в количествах зашифрования, а необходимую для работы алгоритма память M будем измерять в битах. Тогда имеем

$$Q = 2^{32} \cdot 2^{64} \cdot 1.5 \cdot 2^{128} = 1.5 \cdot 2^{192}$$

$$M = 2^{64} \cdot 128 + \alpha = 2^{71} + \alpha,$$

где α - количество памяти, необходимое для хранения локальных переменных алгоритма. Количетсво памяти можно сократить до 2^{36} адресов (хранятся 128-битные вектора) с помощью техники «угадывай и определя» (guess and determine), которая заключается в построении дерева возможных ключей с частичным перебором их значений.

Возвращаясь к рассуждениям о получении шифртекста по открытому тексту, стоит заметить, что данная формулировка алогритма использует тот факт, что объём построенного заранее корпуса данных должен быть не менее 2^{32} пар открытый-закрытый тексты. Вероятность попадания в неподвижную точку преобразования HTH^{-1} равна 2^{-32}. Из-за этого при наличии корпуса размером 2^{32} мы в среднем попадём в 1 неподвижную точку.

Вероятность успеха (наличия на доступном материале неподвижной точки) равна

$$p = 1 - (1 - \frac{1}{2^{32}})^{2^{32}} \approx 1 - e^{-1} \approx 0.63.$$