

Правительство Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"
Московский институт электроники и математики им. А.Н. Тихонова
Департамент прикладной математики

УДК _____
№ госрегистрации _____
Инв. № _____

УТВЕРЖДАЮ

головой исполнитель

« _____ » _____ 2022 г.

МЕЖДИСЦИПЛИНАРНАЯ КУРСОВАЯ РАБОТА

Тема работы

по теме:

Исследование вопросов оптимизации методов анализа некоторых схем
шифрования сохраняющих формат
(промежуточный)

Руководитель курсовой работы _____ Д.Б. Фомин

Академический руководитель
образовательной программы _____ А.Б. Лось

Москва 2022

СПИСОК ИСПОЛНИТЕЛЕЙ

Выполнил студент

Щеглова П.Н.

СОДЕРЖАНИЕ

Введение	4
1 Шифрование с сохранением формата	5
1.1 Описание концепции	5
1.2 Действующие стандарты	6
1.2.1 FEA-1	7
1.3 Линейный метод	8
1.3.1 Их результаты	9

ВВЕДЕНИЕ

Утечка персональных данных является на данный момент серьезной проблемой, с которой в последнее время люди стали сталкиваться все чаще, так как их можно использовать в различных мошеннических схемах - номера кредитных карт, страховых полисов, в связи с чем остро встала задача безопасного хранения подобной информации. Данная ситуация увеличила важность шифрования с сохранением формата данных (format-preserving encryption), так как оно лучше подходит для хранения в базах данных, чем традиционные механизмы с использованием симметричного шифрования. Однако несмотря на исследования вокруг существующих алгоритмов и их постоянную модернизацию находятся способы показать их практическую нестойкость с помощью известных методов, таких как линейный и дифференциальный криптографический анализ, которые с небольшими доработками и уточнениями дешифруют шифртексты, использующие актуальные схемы типа FPE. В данной курсовой работе демонстрируется применение линейного метода анализа схем FPE с настройками на основе сети Фейстеля, а именно стандарта FEA-1, применение линейного метода в российской традиции и его сравнение с представленным в анализируемой статье, а также результаты эксперимента по нахождению линейного статистического аналога для входных и выходных последовательностей шифра.

1 Шифрование с сохранением формата

1.1 Описание концепции

Format-preserving encryption, сокращенно FPE, - тип шифрования, который подразумевает сохранение формата открытого текста в шифртексте, или более формально отображение из множества открытых текстов в то же самое множество. Примеры отображений: шифрование 16-значного номера банковской карты 16-значным числом; шифрование одного английского слова другим английским словом; шифрование n -битного числа n -битным числом (совпадает с определением n -битного блочного шифра).

Для конечных (в обобщении - счетных) множеств данный тип шифрования эквивалентен перестановке перенумерованных элементов множества. Истинно случайная перестановка является идеальным шифром FPE, однако для больших (бесконечных счетных) множеств невозможно предварительно сгенерировать и запомнить такую перестановку. Таким образом, проблема FPE состоит в том, чтобы сгенерировать псевдослучайную перестановку из секретного ключа таким образом, чтобы время вычисления для одного значения было небольшим (в идеале постоянным, но, что наиболее важно, меньшим, чем $O(N)$).

Алгоритм FPE можно реализовать с использованием сети Фейстеля. Сеть Фейстеля нуждается в источнике псевдослучайных значений для раундовых ключей, и выходные данные алгоритма AES могут использоваться в качестве этих псевдослучайных значений.

Чтобы реализовать алгоритм FPE с использованием AES и сети Фейстеля, можно использовать столько битов вывода AES, сколько необходимо, чтобы получить последовательность с длиной равной длине левой или правой половины сети Фейстеля. Если, например, в качестве раундового ключа требуется 24-битное значение, можно использовать 24 младших бита вывода AES.

При данном подходе выходные данные сети Фейстеля не обязательно сохраняют формат входных данных, поэтому итерации сети Фейстеля

повторяются с помощью cycle-walking до тех пор пока формат не совпадет. Поскольку можно настроить размер входов в сеть Фейстеля, можно сделать наиболее вероятным, что эта итерация завершится в среднем достаточно быстро. В случае номеров кредитных карт, например, существует 10^{15} возможных 16-значных номеров кредитных карт (с учетом избыточной контрольной цифры), а поскольку $10^{15}/approx 2^{49,8}$, с использованием 50-битной сети Фейстеля и cycle-walking можно создать алгоритм FPE, который в среднем зашифровывает довольно быстро.

1.2 Действующие стандарты

Существует множество реализованных алгоритмов типа FPE, к актуальным можно отнести разработанные в США FF1 и FF3-1, а также южно-корейские FEA-1 и FEA-2. Алгоритм FEA, представленный институтом исследований национальной безопасности (NSR), использует сети Фейстеля, аналогичные стандартам NIST, FF1 и FF3-1. Однако алгоритмы FF1 и FF3-1 используют блочные шифры как F-функции, в то время как FEA использует свои собственные специализированные функции. Эта особенность позволяет использовать более высокоскоростное шифрование по сравнению с другими алгоритмами, предназначенными для шифрования с сохранением формата. FEA может быть подходящим выбором, при шифровании конфиденциальной персональной информации, которая, как правило, имеет небольшой объем.

Разница между этими двумя алгоритмами состоит в том, что FEA-1 имеет размер настройки $128 - n$ бит (где n - размер входной последовательности), каждый с 12, 14 и 16 раундами при длине двоичного ключа 128, 192 и 256 соответственно. FEA-2 имеет фиксированный размер параметра настройки в 128 бит с 18, 21 и 24 раундами при длинах ключей 128, 192 и 256 соответственно.

1.2.1 FEA-1

Опишем подробнее стандарт, который анализируется в данной работе, а именно FEA-1:

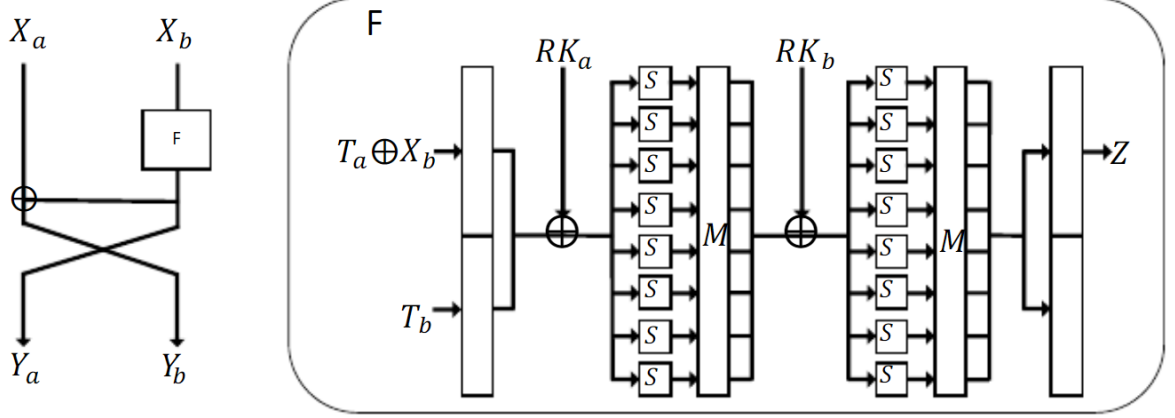


Рисунок 1 — Структура итерации FEA, на основе сети Фейстеля

На вход алгоритму подаются последовательности чисел из конечного множества, мощностью от 2^8 до 2^{128} , размер двоичного ключа K составляет $\in [128, 192, 256]$. Алгоритм представляет собой последовательное применение итераций сети Фейстеля, ее общая схема представлена слева на Рисунке 1. Входная последовательность итерации X делится на две равные части X_a и X_b , X_b передается на вход специализированной F-функции, общая схема которой обозначена справа на Рисунке 1. T_a и T_b - левая и правая половина настройки, принцип формирования которой будет описан далее, RK_a и RK_b - левая и правая половина раундового ключа, S - блок подстановки (в данной схеме применяются идентичные S -блоки), M - блок умножения на заданную матрицу.

Выбор настройки для каждой итерации происходит по следующему алгоритму: настройка T , битовый вектор длины $128 - n$, делится на две под-настройки $T_L = T_{[0:64-n_2-1]}$ и $T_R = T_{[64-n_2:128-n-1]}$ длины $64 - n_2$ и $64 - n_1$, соответственно. Полагаем $T_a^i = 0$ для каждой итерации и T_b^i для i -ой итерации, как:

$$T_b^i = \begin{cases} T_L & \frac{i}{2} \in N \\ T_R & \frac{i+1}{2} \in N \end{cases}$$

1.3 Линейный метод

Линейный метод криптографического анализа состоит из двух этапов:

а) Нахождение линейного статистического аналога для части исходного блочного шифра, это линейное соотношение связывает входные и выходные значения выбранной части алгоритма. Оно должно выполняться с вероятностью заметно отличающейся от случайной для возможности отличия этих двух вариантов событий.

б) Отбрасывание ложных ключей с использованием найденного вероятностного соотношения.

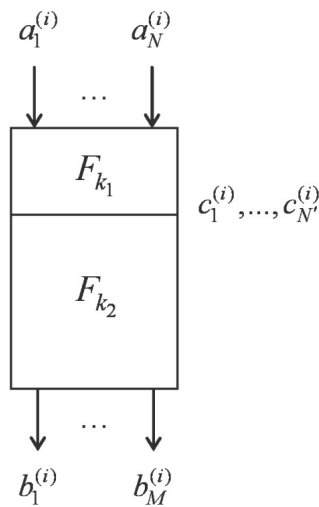


Рисунок 2 — Схема разбиения алгоритма на два блока для проведения линейного криптографического анализа

Перейдем к описанию метода:

Пусть схема шифропреобразования с общим ключом K разбита на две последовательные части F_{K_1} и F_{K_2} , как показано на Рисунке 2. В первой из них используется небольшая часть исходного ключа K_1 , во второй, соответственно, большая K_2 (при этом часть K_1 может совпадать

с частью K_2). Пусть также найдено линейное соотношение

$$c_1^{(i)} L'_1 + \dots + c_N^{(i)} L'_N \simeq b_1^{(i)} L''_1 + \dots + b_N^{(i)} L''_N,$$

которое, независимо от значения K_2 , выполняется с вероятностью $P = \frac{1+\delta}{2}$, где $\delta \neq 0$.

Булевы величины L'_j и L''_s , $j, s \in \{1, \dots, N\}$ (маска найденного линейного соотношения) известны; $c_1^{(i)}, \dots, c_N^{(i)}$ - промежуточный шифртекст, между двумя блоками шифропреобразования; $b_1^{(i)}, \dots, b_N^{(i)}$ - известный итоговый шифртекст, $a_1^{(i)}, \dots, a_N^{(i)}$ - известный открытый текст, $i \in 1, \dots, T$, где T - количество материала.

Пусть K'_1 - доля ключа K_1 , от которой зависит левая часть в соотношении 1.3. Если при опробовании K'_1 выполнимость соотношения с вероятностью $P = \frac{1+\delta}{2}$ не подтверждается, то соответствующее значение K'_1 отбраковывается. Чем больше при этом T и $|\sigma|$, тем большая доля значений K'_1 будет отбракована, вплоть до однозначного определения K'_1 .

1.3.1 Их результаты

Линейный криптоанализ основан на вероятностны линейных соотношениях или линейных приближениях. Пусть $F : F_2^n \rightarrow F_2^m$ - функция, возможно зависящая от ключа. Линейные различители строятся на основе линейных приближений с большой абсолютной корреляцией. Линейное приближение для F определяется двумя масками $(u_1, u_2) \in F_2^n \oplus F_2^m$, и их корреляция равна:

$$C_{u_1, u_2}^F = 2 \cdot P(u_1^\top F(x) = u_2^\top x) - 1 = \frac{1}{2^n} \sum_{x \in F_2^n} (-1)^{u_1^\top F(x) + u_2^\top x},$$

где вероятность считается от равновероятно распределенного $x \in F_2^n$. Если $u_1 \neq 0$, то корреляция равновероятно распределенной функции сконцентрирована вокруг нуля с стандартным отклонением $2^{-n/2}$. Следовательно, если корреляция C значительно превосходит $2^{-n/2}$, то различителем можно считать вычисление корреляции для $q = \Theta(1/c^2)$ пар масок и сравнение получившего значения с теоретическим.

На первый взгляд стандарты FEA-1 и FEA-2 кажутся стойкими к линейному криптоанализу, особенно когда их раундовые функции F заменены на равновероятные случайные функции. Основное замечание, которое можно эксплуатировать в атаках на такие шифры, состоит в том, что шифр оказывается нестойким, если настройка (его часть) считается частью входных данных.

Если внимательно рассмотреть две итерации FEA-1, то можно заметить, что настройка T_L - произвольная постоянная, а T_R считается переменной входа. Если это не так, то для проведения атаки T_R должна быть известной.