

Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт искусственного интеллекта   
Базовая кафедра №252 – информационной безопасности

**курсовая работа по предмету**

**«КРИПТОГрафические методы защиты информации»**

**Тема работы:** Криптоанализ алгоритма симметричного шифрования SM4

|  |  |  |
| --- | --- | --- |
| **Студент группы ККСО-01-20** |  | *Семин В.В.* |
| **Руководитель** |  | *Старший преподаватель*  *Хомутов Дмитрий*  *Геннадьевич* |

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc167682850)

[1 ОПИСАНИЕ АЛГОРИТМА ШИФРОВАНИЯ 6](#_Toc167682851)

[1.1 РАУНДОВАЯ ФУНКЦИЯ 6](#_Toc167682852)

[1.2 РАСШИРЕНИЕ КЛЮЧА 6](#_Toc167682853)

[1.3 WHITE-BOX РЕАЛИЗАЦИЯ SM4 7](#_Toc167682854)

[1.3.1 Этап инициализации 7](#_Toc167682855)

[1.3.2 White-box алгоритм раунда SM4 8](#_Toc167682856)

[2 ОПИСАНИЕ АЛГОРИТМОВ АТАКИ 14](#_Toc167682857)

[2.1 ЛИНЕЙНАЯ АТАКА НА 22-РАУДОВЫЙ SM4 14](#_Toc167682858)

[2.2 ДИФФЕРЕНЦИАЛЬНАЯ АТАКА НА 23-РАУНДОВЫЙ SM4 16](#_Toc167682859)

[3.3 КРИПТОАНАЛИЗ WHITE-BOX РЕАЛИЗАЦИЮ SM4 18](#_Toc167682860)

[3.3.1 Функция коллизий 18](#_Toc167682861)

[3.3.2 Восстановление функции Sij 20](#_Toc167682862)

[2.3.3 Восстановление линейной части кодирования Ei+1 22](#_Toc167682863)

[2.3.4 Восстановление маскированного раундового ключа 23](#_Toc167682864)

[2.3.5 Восстановление кодировщиков 24](#_Toc167682865)

[3 АНАЛИЗ СЛОЖНОСТИ РАССМОТРЕННЫХ АТАК 25](#_Toc167682866)

[3.1 АНАЛИЗ СЛОЖНОСТИ ЛИНЕЙНОЙ АТАКА НА 22 РАУДОВЫЙ SM4 25](#_Toc167682867)

[3.2 АНАЛИЗ СЛОЖНОСТИ ДИФФЕРЕНЦИАЛЬНОЙ АТАКИ НА 23-РАУНДОВЫЙ SM4 26](#_Toc167682868)

[3.3 ВРЕМЕННАЯ СЛОЖНОСТЬ КРИПТОАНАЛИЗА WHITE-BOX РЕАЛИЗАЦИИ 28](#_Toc167682869)

[4 ПРАКТИЧЕСКАЯ ЧАСТЬ 29](#_Toc167682870)

[4.1 ПРОВЕРКА ЛИНЕЙНОЙ АТАКА НА 22 РАУДОВЫЙ SM4 29](#_Toc167682871)

[4.2 ПРОВЕРКА ДИФФЕРЕНЦИАЛЬНОЙ АТАКИ НА 23-РАУНДОВЫЙ SM4 30](#_Toc167682872)

[4.3 ОЦЕНКА РЕЗУЛЬТАТОВ КРИПТОАНАЛИЗА WHITE-BOX РЕАЛИЗАЦИИ SM4 НА ВОЗМОЖНОСТЬ ИХ ПРИМЕНЕНИЯ ДЛЯ УСПЕШНОЙ АТАКИ 32](#_Toc167682873)

[ВЫВОД 33](#_Toc167682874)

[СПИСОК ЛИТЕРАТУРЫ 34](#_Toc167682875)

[ПРИЛОЖЕНИЕ А 35](#_Toc167682876)

# ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий и информационных систем безопасность данных приобретает первостепенное значение. Одним из ключевых элементов обеспечения безопасности информации является криптография, которая включает в себя методы шифрования и дешифрования данных. В Китае одним из стандартов шифрования, получивших широкое распространение, является алгоритм SM4. Он был разработан в 2006 году и принят в качестве национального стандарта (GBT.32907-2016) для защиты данных в различных приложениях, включая финансовые и правительственные системы.

Однако, несмотря на высокую степень защищенности алгоритма SM4, как и любой криптографический метод, он не застрахован от угроз криптоанализа. Криптоанализ представляет собой совокупность методов и техник, направленных на выявление уязвимостей криптографических алгоритмов с целью их взлома или ослабления. Изучение методов криптоанализа является важной научной задачей, так как позволяет не только проверять устойчивость существующих алгоритмов, но и разрабатывать более совершенные системы защиты данных.

Цель данной курсовой работы – рассмотреть различные методы криптоанализа, применимые к алгоритму SM4. В рамках исследования будут проанализированы наиболее известные и эффективные техники криптоанализа, такие как линейный и дифференциальный криптоанализ, а также рассмотрены новейшие подходы, появившиеся в последние годы.

Актуальность данной работы заключается в систематизации существующих знаний в области криптоанализа SM4 и выявлении перспективных направлений для дальнейших исследований. Результаты исследования могут быть полезны как для специалистов в области информационной безопасности, так и для разработчиков криптографических систем, стремящихся к созданию более надежных методов защиты данных.

Объект исследования – алгоритм SM4.

Предмет исследования – возможность проведения успешной атаки на рассматриваемый алгоритм.

Задачи исследования:

1. Изучить доступную в открытых источниках информацию об атаках на SM4.
2. Оценить сложность выполнения данных атак по памяти и по времени.
3. Оценить практическую возможность проведения рассмотренных атак.

Гипотеза исследования. На данный момент в открытом доступе уже существует атака на SM4, способная значительно ослабить его криптографическую стойкость.

В данной курсовой работе будут использованы следующие методы исследования.

1. Аналитический метод. Этот метод будет использоваться для изучения существующих научных публикаций, статей и других источников, связанных с алгоритмом SM4 и методами его криптоанализа. Анализ позволит систематизировать и обобщить имеющиеся данные о различных техниках атак и их эффективности.
2. Математическое моделирование. Для понимания работы методов криптоанализа и их воздействия на алгоритм SM4 будут использоваться математические модели. Это позволит детально рассмотреть процессы, происходящие при применении различных техник криптоанализа, и оценить их эффективность.
3. Экспериментальный метод. В рамках работы будут проведены экспериментальные исследования, включающие моделирование атак на алгоритм SM4. Это позволит на практике проверить теоретические выводы и оценить уязвимости алгоритма. Эксперименты будут включать реализацию различных методов криптоанализа и анализ полученных результатов.

# 1 ОПИСАНИЕ АЛГОРИТМА ШИФРОВАНИЯ

Описание алгоритма представлено в стандарте [1]. Алгоритм имеет размер блока 128 бит и размер ключа 128 бит. Его структура – несбалансированная сеть Фейстеля (за каждый раунд преобразуется 32 бита блока) с 32 раундами.

## 1.1 РАУНДОВАЯ ФУНКЦИЯ

Раундовая функция имеет вид (1.1).

|  |  |
| --- | --- |
|  | (1.1) |

Здесь – блок открытого текста, – раундовый ключ, – подстановка.

Подстановка является суперпозицией подстановки линейной подстановки и нелинейной подстановки , то есть .

В свою очередь имеет вид (1.2), а имеет вид (1.3).

|  |  |
| --- | --- |
|  | (1.2) |
|  | (1.3) |
| Результатом является блок зашифрованного текста .  На Рисунке 1.1 представлена схема -того раунда SM4.    Рисунок 1.1 - Схема -того раунда SM4. |  |

## 1.2 РАСШИРЕНИЕ КЛЮЧА

Расширение ключа происходит по формуле (1.4)

|  |  |
| --- | --- |
|  | (1.4) |

Здесь , где – ключ, – константы; – константы; - подстановка, идентичная подстановке за исключением того, что вместо используется , имеющая формулу (1.5).

|  |  |
| --- | --- |
|  | (1.5) |

## 1.3 WHITE-BOX РЕАЛИЗАЦИЯ SM4

White-box криптография – раздел криптографии, посвящённый защите криптографических алгоритмов от восстановления ключа в экстремальных условиях, когда атакующий имеет полный доступ к окружению и деталям исполнения алгоритма. Например, у атакующего есть доступ к исполняемому файлу, который он может проанализировать, или он обладает исходным кодом программы шифрования, или он может считывать информацию из памяти шифрующего устройства и перехватывать вызовы в его процессоре во время исполнения шифрования.

В наше время white-box криптография имеет множество реальных сценариев применения, таких как ТВ-боксы, мобильные телефоны, игровые консоли.

В статье [6] представлена одна из последних на данный момент white-box реализаций SM4. Она будет рассмотрена в данном пункте.

### 1.3.1 Этап инициализации

Линейное преобразование , используемое в SM4 можно представить в матричной форме (1.6).

|  |  |
| --- | --- |
|  | (1.6) |

Здесь , , – матричные блоки размером битов:

Расширим матричное представление (1.6) с до битов (формула 1.7, 0 обозначает нулевую матрицу размером битов).

|  |  |
| --- | --- |
|  | (1.7) |

### 1.3.2 White-box алгоритм раунда SM4

1. Вычислить .

Перед расчётами необходимо снять выходное скремблирующее кодирование предыдущего раунда и наложить новое скремблирующее кодирование. Чтобы удостоверится в том, что последовательные операции Исключающего ИЛИ могут быть применены напрямую, необходимо удостоверится в том, что линейная часть скремблирующего кодирования является постоянной. Рисунок 1.2 показывает процесс вычисления .

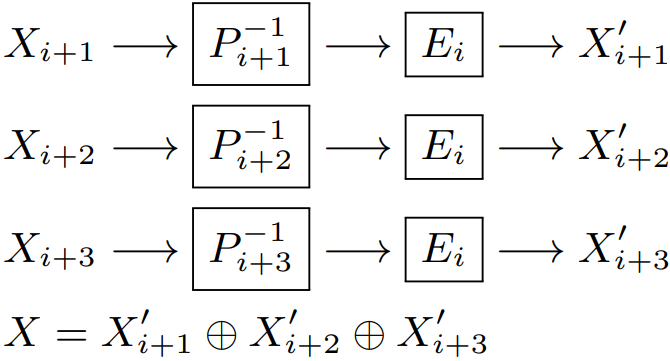


Рисунок 1.2 - Процесс вычисления .

В данном алгоритме для выходного скремблирующего кодирования используется обратимое аффинное преобразование (1.8).

|  |  |
| --- | --- |
|  | (1.8) |

В данном уравнении – обратимая матрица из бита, – 32-битная константа.

Входное скремблирующее кодирование может быть представлено формулой (1.9).

|  |  |
| --- | --- |
|  | (1.9) |

В этом уравнении – обратимое аффинное преобразование размером битов.

В конечном итоге, нам надо сохранить , где может быть представлено как обратимая матрица размером бита и -битным вектором. Вычисление состоит из 3 аффинных преобразований и 2 32-битных операций Исключающее ИЛИ.

1. Вычисление по таблице. Для примера возьмём первый раунд .

На основе ключа первого раунда создаются следующие 4 таблицы с входом 8 бит и выходом 64 бита (уравнения 1.10).

|  |  |
| --- | --- |
|  | (1.10) |

В данных уравнениях , где , – 8-битные случайные числа, – композиция сдвиговой матрицы и таблицы выходного скремблирующего кодирования , – матрица бит. Выход S-бокса определяется значением 4-битного вектора

|  |  |
| --- | --- |
|  | (1.10) |

Чтобы проиллюстрировать процесс использования справочных таблиц в деталях, возьмём 4-биный случайный вектор . Пусть выходом S-бокса будет: . Байты, отмеченные , представляют «полезную» информацию. Справочные таблицы для первого раунда выглядят следующим образом:

Применим Исключающее ИЛИ на результат таблицы:

Пусть выходом S-бокса стандартного алгоритма SM4 будет . После линейного преобразования L он может быть представлен уравнением (1.11).

|  |  |
| --- | --- |
|  | (1.11) |

Нужно удостоверится, что возможно получить результат, совпадающий с тем результатом, который получается после применения преобразования в стандартном алгоритме SM4, с помощью применения Исключающего ИЛИ на выход справочных таблиц. Поэтому нужно включить сдвиговую матрицу в для .

В этом выражении – линейная часть справочной таблицы выходного скремблирующего кодирования, которая состоит из 8 матриц ; – постоянная часть выходного скремблирующего кодирования, которая состоит из 8 8-битных векторов, конкатенированных вместе. Для завершения раунда требуется извлечь «полезную» информацию, то есть 4 байта, отмеченных из 64-битного результата Исключающего ИЛИ и сложить их по модулю 2 с . Вычисление состоит из 4 справочных таблиц с 8-битным входом и 64-битным выходом и 3 64-битных операций Исключающего ИЛИ. Рисунок 1.3 показывает вычисление .

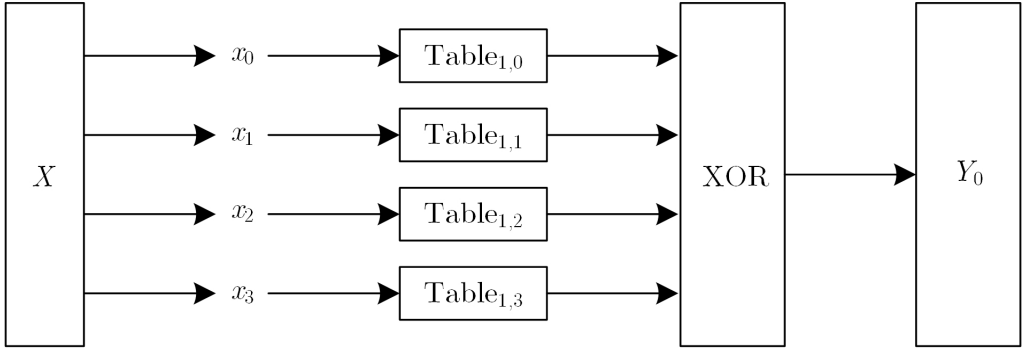


Рисунок 1.3 – Вычисление .

1. Вычислить /

Пусть результат справочной таблицы в предыдущем пункте равен , где – «полезная» информация. Вычисление происходит по формулам (1.12).

|  |  |
| --- | --- |
| ′  Инверсия справочной таблицы выходного скремблирующего кодирования → | (1.12) |

Здесь и . В конечном счёте, вычисление требует два аффинных преобразования и одну операцию Исключающего ИЛИ над 32-битным вектором.

В схему было добавлено внешнее кодирование. Поэтому, 128-битный вход в алгоритм сначала подвергается внешнему кодированию и только после этого попадает в процесс white-box имплементации. После 32 раундов итеративного шифрования он снова подвергается внешнему кодированию, чтобы получить настоящий шифртекст. Внешнее кодирование аннулируется аффинным преобразованием первого и последнего раундов, поэтому результат white-box алгоритма и стандартного SM4 имеет тот же вход и выход. Внешнее кодирование происходит на защищённой платформе.

# 2 ОПИСАНИЕ АЛГОРИТМОВ АТАКИ

## 2.1 ЛИНЕЙНАЯ АТАКА НА 22-РАУДОВЫЙ SM4

Статья [2] предлагает линейную атаку на 22 раундовый SM4. Для этого сначала строится линейная аппроксимация раундов 2-19, описанная уравнениями (2.1) – (2.6).

|  |  |
| --- | --- |
| ; раунд 5 | (2.1) |
| ; раунд 6 | (2.2) |
| ; раунд 10 | (2.3) |
| ; раунд 11 | (2.4) |
| ; раунд 15 | (2.5) |
| ; раунд 16 | (2.6) |

Здесь С учётом того, что равно :

|  |  |
| --- | --- |
| ; раунд 5 | (2.7) |
| ; раунд 6 | (2.8) |
| ; раунд 10 | (2.9) |
| ; раунд 11 | (2.10) |
| ; раунд 15 | (2.11) |
| ; раунд 16 | (2.12) |

Сложив уравнения (2.7) – (2.12), получим уравнение линейной аппроксимации раундов 2-19 (2.13).

|  |  |
| --- | --- |
|  | (2.13) |

Автор утверждает, что данная аппроксимация имеет отклонение .

Далее левая часть преобразуется для расширения на 22 раунда:

|  |  |
| --- | --- |
|  | (2.14) |

Здесь – правая часть уравнения (2.13), – результат шифрования 22 раундовым алгоритмом.

Выражение (2.14) имеет форму (2.15)

|  |  |
| --- | --- |
|  | (2.15) |

– функция чётности выражения, зависящая от открытого текста и закрытого. – функция, являющаяся композицией нескольких функций .

Так как в данной атаке количество активных s-боксов 6 для раундов 1, 20 и 8 для 21,22, то зависит от 112 битного вектора , где , , , . Соответственно, представляется формулой (2.16).

|  |  |
| --- | --- |
|  | (2.16) |

Алгоритм атаки следующий:

1. Инициализировать вектор из элементов, каждый из которых соответствует возможному значению битов части текста/шифртекста, использующегося в вычислении .
2. Для каждой пары векторов , вычислить чётность (чётность равна 1, если равно 0, -1 иначе). Сохранить результат в матрице , то есть
3. Вычислить чётность для каждого и прибавить её к соответствующему элементу .
4. Вычислить отклонение .
5. Если , то выбрать соответствующий вариант ключа и предположить

Если , то выбрать соответствующий вариант ключа и предположить

1. После получения значений остаётся перебрать 15 бит для получения ключа.

## 2.2 ДИФФЕРЕНЦИАЛЬНАЯ АТАКА НА 23-РАУНДОВЫЙ SM4

В работе [5] показана дифференциальная атака на 23-раундовый SM4, которая строится на основе 19-раундовой дифференциальной характеристики, рассчитанной авторами статьи.

Данная характеристика выглядит следующим образом. Пусть . Определим ( – это вероятность того, что при разности входов функцией будет получена разность выходов ). Автор статьи утверждает, что множество имеет мощность . Это означает, что возможно получить примерно характеристик данным способом. Определим . Таблица в Приложении А показывает данные характеристики более подробно. В ней , – входная разность и – выходная разность.

Атака на 23-раундовый SM4 начинается с применения дифференциальной характеристики к первым 19 раундам. Для этого понадобится выбрать открытые текста с разностью , где . Сгенерируем два набора и , в каждом из которых открытых текста (формула 2.17).

|  |  |
| --- | --- |
|  | (2.17) |

Здесь – 3 фиксированных слова, – принимают все возможных значений. Назовём пару структурой. Одна структура содержит различных открытых текстов и может создать около кандидатов-пар открытых текстов.

После применения характеристики выходная разность раунда 18 – это . А значит, входная разность функции в раунде 19 равна . Для S-бокса SM4 существует 127 возможных выходных разностей для каждой ненулевой входной разности. Следовательно, функция имеет примерно возможных вариантов. Пусть – множество всех вариантов. Тогда выходные разности последующих раундов (19-22) должны принадлежать соответственно, где обозначает неизвестное слово.

Алгоритм атаки:

1. Выбрать структур. Мы получим пар открытых текстов, используя открытых текстов.
2. Для каждой пары открытых текстов вычислить разность соответствующих шифртекстов и проверить, принадлежит ли первое слово шифртекста множеству . Если это не так, то отбрасываем пару. После этого должно остаться примерно пар.
3. Для каждого предположения о значении нулевого байта , то есть , делаем следующее:
   1. Для каждой оставшейся пары шифртекста пусть , . Частично расшифруем по формуле (2.18), чтобы получить разность нулевого S-бокса в раунде 22., и вычислим по формуле (2.19). Если предположение о значении верно, то для правильных пар . Если это не так, отбрасываем пару. После этого теста должно остаться примерно пар шифртекстов.

|  |  |
| --- | --- |
|  | (2.18) |
|  | (2.19) |

* 1. Для каждого из оставшихся 3 байт , 4 байт , 4 байт продолжаем делать предположения. Для каждого из этих предположений также вычисляем и . Как и в пункте а отбрасываем, неподошедшие пары на каждом шаге. После этих тестов должно остаться примерно пар шифртектов.
  2. Предположим значение . Поступим так же, как в пункте а. Примерно часть пар уже была отброшена, из-за выбора множества . Следовательно, после теста около пар шифртестов останется.
  3. Для каждого из оставшихся 3 байт делаем предположения и повторяем аналогично шагу а. После этого должно остаться около пар шифртекстов.

1. Выведите 128-битное предположение о значении , если оно имеет максимальное число оставшихся пар после шага 3.

## 3.3 КРИПТОАНАЛИЗ WHITE-BOX РЕАЛИЗАЦИЮ SM4

В статье [7] представлена атака на white-box реализацию SM4 из статьи [6].

### 3.3.1 Функция коллизий

На рисунке 3.1 представлена функция коллизий, которая принимает на вход два 32-битных параметра во второй части раунда шифрования в white-box реализации SM4 и выводит – операцию ветки в первой части следующего раунда шифрования.

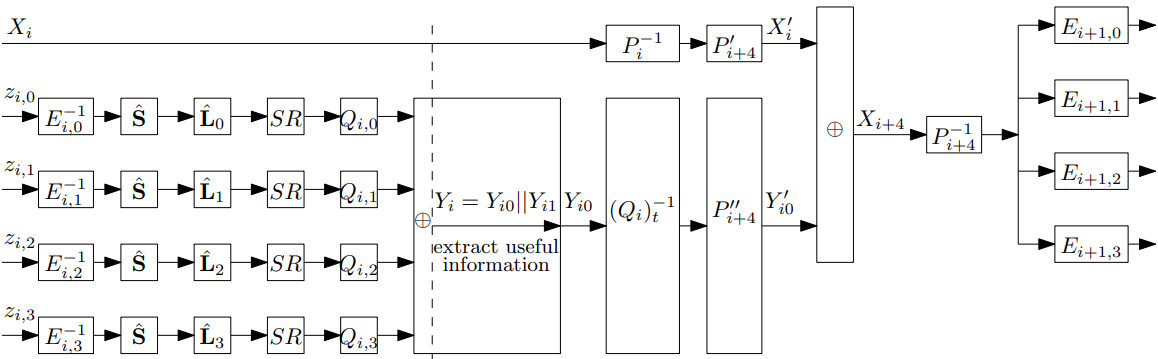
**

Рисунок 3.1 – Функция коллизий.

Заметим, что и – диагональные аффинные преобразования, и – обратимые аффинные преобразования размером бит, и – оригинальных входной байт в -ый оригинальный S-бокс -ого раунда зашифрования в защищённой форме с .

Функция коллизий может быть упрощена до той, которая представлена на Рисунке 3.2.

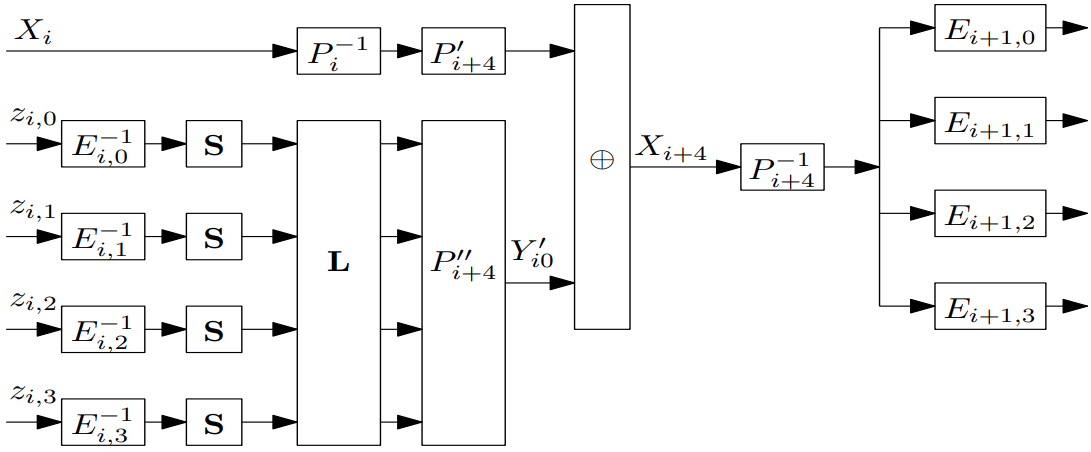
**

Рисунок 3.2 – Упрощённая функция коллизий.

Установим , такое что , и обозначим константу как . Пусть – оригинальное 32-битное значение сразу после операции при входе . Тогда

.

При функция коллизий будет иметь вид (2.20).

|  |  |
| --- | --- |
|  | (2.20) |

Кроме этого, можно выразить как конкатенацию 4 байтовых функций(формула 2.21).

|  |  |
| --- | --- |
|  | (2.21) |

Определим функцию формулой (2.22)

|  |  |
| --- | --- |
|  | (2.22) |

### 3.3.2 Восстановление функции Sij

Воспользуемся коллизией (2.23) для восстановления и .

|  |  |
| --- | --- |
|  | (2.23) |

Здесь . Если рассмотреть уравнения (1.6) и (2.23), то получаем

Где – соответствующий байт . Так как – биекция, имеет место следующее выражение:

Определим и и тогда получаем формулу (2.24).

|  |  |
| --- | --- |
|  | (2.24) |

Так как и – биекции, можно найти 256 коллизий. После удаления , получается 255 пар , удовлетворяющих уравнению (2.23), каждое сиз которых даёт уравнение вида (2.24). При этом, мы используем и другие функции чтобы сгенерировать похожие уравнения с разными коэффициентами в . Наконец, получаем линейных уравнений с 512 неизвестными (уравнение (2.25)).

|  |  |
| --- | --- |
|  | (2.25) |

Определим и Тем самым уменьшим количество неизвестных до 510, тогда система примет вид 2.26.

|  |  |
| --- | --- |
|  | (2.26) |

4 × 255 уравнений дают линейную систему ранга 509, и в такой линейной системе уравнений все остальные неизвестные могут быть выражены как функция одного из них, например Таким образом, существуют коэффициенты и , такие что и . Из этого получаем уравнения (2.27).

|  |  |
| --- | --- |
|  | (2.27) |

Далее восстанавливаем полным перебором пар , верифицируя корректность уравнением из определения :

Так как – обратимое аффинное преобразование размером , функция выше имеет алгебраическую степень не более 1. Для неправильной пары будет получен неправильный кандидат , который будет аффинным эквивалентом , то есть существует матрица и 8-битный вектор , такой что при . Функцию можно выразить, как

В этом случае с большой вероятностью у функции алгебраическая степень больше 1.

Для того, чтобы проверить, имеет ли функция алгебраическую степень 1 или меньше, вычислим её производную в точке 01

Проверим, является ли постоянной для всех вариантов входа. Вероятность того, что она окажется постоянной для неправильной пары равна , так что неправильные предположения могут быть быстро убраны.

После этого используем уравнение (2.27) для восстановления полным перебором по и походим образом восстанавливаем и другими уравнениями.

### 2.3.3 Восстановление линейной части кодирования Ei+1

Так как – обратимое аффинное преобразование, его можно выразить, как , где линейная часть обратимая матрица и – 8-битная константа.

Получив 32-битный вход в функцию коллизий, обозначим 32-битное значение после операции следующим образом:

Так как известно и уже восстановлено, можно вычислить Выход функции коллизий

Здесь .

Рассмотрим вывод , при входе (формула (2.28)).

|  |  |
| --- | --- |
|  | (2.28) |

Далее, выбираем 32-битный вход в функцию коллизий, такой, что выход после операции равен 0, это можно сделать легко, выбрав , такой, что

Следовательно, имеем выход (2.29)

|  |  |
| --- | --- |
|  | (2.29) |

Наконец, складываем по модулю 2 уравнения (2.28) и (2.29), чтобы получить . Из этого уравнения возможно восстановить .

### 2.3.4 Восстановление маскированного раундового ключа

В этом пункте будет получено значение . Пусть – функция коллизий -ого раунда. Для неё, основываясь на выражениях (2.22) и (2.23), определим

Здесь – постоянная часть, которая может быть легко вычислена.

Так как - обратимое аффинное преобразование размером , функция выше имеет алгебраическую степень не более 1. Для неправильного предположения функция определена, как

имеет алгебраическую степень больше, чем 1.

Для определения степени можно использовать то же тест, что и в пункте 2.3.2.

В результате возможно также восстановить для , изменяя определение функции . Так как , то мы можем получить , где

### 2.3.5 Восстановление кодировщиков

В данном пункте будет показано частичное или полное восстановление кодировщиков .

Сначала восстанавливаем линейную часть. Для этого рассмотрим разность выводов пар входов:

Постоянная часть может быть легко восстановлена, так как

используют одну линейную часть , но способа восстановить и на данный момент не существует.

# 3 АНАЛИЗ СЛОЖНОСТИ РАССМОТРЕННЫХ АТАК

## 3.1 АНАЛИЗ СЛОЖНОСТИ ЛИНЕЙНОЙ АТАКА НА 22 РАУДОВЫЙ SM4

Введём несколько определений из статьи [3].

Циркулятная матрица. Матрица является циркулянтной, если каждый вектор строки (столбца) сдвинут на один элемент вправо относительно предыдущего вектора строки (столбца).

Блочно-циркулянтная матрица. Матрица является m-блочно-циркулянтной, если она является циркулянтной по блокам, и количество блоков в каждой строке (или столбце) равно m.

Уровневая циркулянтная матрица.

1. Матрица является циркулянтной уровня 1 типа (n), если она является циркулянтной размером n.
2. Матрица является циркулянтной уровня 2 типа (m, n), если она является m-блочно-циркулянтной, и каждый блок сам является циркулянтной матрицей размером n.
3. Матрица является циркулянтной уровня 3 типа (m, n, o), если она является m-блочно-циркулянтной, блоки которой являются циркулянтными уровня 2 типа (n, o).
4. Матрица является циркулянтной уровня q типа (m, n, o, ...), если она является m-блочно-циркулянтной, блоки которой являются циркулянтными уровня q-1 типа (n, o, ...).

Рассмотрим потребление памяти и времени для каждого пункта алгоритма атаки, описанное в [2].

1. Для того, чтобы хранить вектор , состоящий из элементов требуется бит памяти.
2. Матрица M имеет циркулянтную природу. Это следует из утверждения, рассмотренного в [3], о том, что матрица размерами , определённая как , где некоторая функция, является циркулянтной уровня типа . Следовательно, расчёт и хранение всей таблицы не требуется. С учётом этого необходимо вычислить различных значений . Исходя из формулы (2.16), на это понадобится операций зашифрования 22-раундовым SM4, что примерно равно операций. Кроме этого, нужно выделить бит для хранения полученного результата.
3. В шаге 3 используются простые операции извлечения битов и инкрементации. На них понадобится операций.
4. В шаге 4 при прямом подходе необходимо операций. Однако, циркулянтная природа позволяет значительно уменьшить количество операций. Для этого можно использовать методы, описанные в [3]. При этом сложность шага 4 уменьшается до
5. Шаг 6 имеет сложность

В результате, для атаки требуется примерно операций зашифрования 22-раундовым SM4, арифметических операций, бит памяти. Согласно лемме, рассмотренной в [4], понадобится пар открытый/закрытый текст для высокой вероятности успеха .

## 3.2 АНАЛИЗ СЛОЖНОСТИ ДИФФЕРЕНЦИАЛЬНОЙ АТАКИ НА 23-РАУНДОВЫЙ SM4

Рассмотрим анализ сложности алгоритма из [5].

Установлено, что для большинства дифференциальных характеристик, описанных в Приложении А, вероятность . Для остальных она больше, чем . Так что с точки зрения атакующего можно предположить, что вероятность каждой дифференциальной характеристики .

Предполагается, что для корректного ключа останется примерно правильных пар для корректного ключа. Однако, для неправильного предположения о подключах ожидаемое количество оставшихся пар после шага 3 алгоритма примерно .

Для данного анализа будет использована концепция «отношение сигнал-шум». Для определения подходящего значения . Отношение сигнал-шум определено, как отношение вероятности правильного ключа быть предложенным правильной парой к вероятности случайного ключа быть предложенным случайной парой при заданной начальной разности. Отношение сигнал-шум может быть вычислено по формуле (3.1).

|  |  |
| --- | --- |
|  | (3.1) |

Здесь – количество предположенных битов ключа, – вероятность дифференциальной характеристики, – среднее число ключей, предложенное учтённой парой и – отношения количества учтённых пар ко всем парам.

В рассматриваемой атаке предполагается значение 128 бит подключа. Вероятность каждой дифференциальной характеристики . Для каждого теста в шаге 3 есть возможных ключей, и учтённая пара должна удовлетворять 32-битному условию., поэтому . В шаге 2 11-битное условие используется для того, чтобы отбросить пары, поэтому . Следовательно отношение сигнал-шум равно

При этом значении примерно 3-4 правильных пар понадобится для успешной дифференциальной атаки.

Поэтому, выбираем . При этом значении ожидается примерно оставшихся текстов при неправильном ключе и 4 оставшихся текста при правильном.

Следовательно, для успешной атаки понадобится выбранных открытых текстов.

Основную часть временной сложности составляет шаг 3. В пункте 3(а) пар шифртекстов обрабатываются с кандидатов на . Так что сложность примерно 23-раундовых шифрования SM4. Аналогично, сложность 3(б) примерно , 3(в) , 3(г) . Следовательно, общая временная сложность около 23-раундовых SM4 зашифрований.

## 3.3 ВРЕМЕННАЯ СЛОЖНОСТЬ КРИПТОАНАЛИЗА WHITE-BOX РЕАЛИЗАЦИИ

При восстановлении существуют кандидатов для полного перебора, и для того, что проверить, что производная является постоянной, нужно проверить входов. Для каждого неправильно предположения вероятность того, что производная – константа, равна примерно . Следовательно, мат ожидание теста . Поэтому ожидаемое время восстановления . Восстановление происходит полным перебором , поэтому временная сложность равна . Следовательно, ожидаемая временная сложность восстановления всех 4 около .

Временная сложность при восстановлении линейной части выходного кодирования пренебрежительно мала. Ожидаемая временная сложность восстановления около , так что ожидаемая сложность восстановления   и временная сложность восстановления маскированных ключей в последних 31 раундах .

# 4 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 4.1 ПРОВЕРКА ЛИНЕЙНОЙ АТАКА НА 22 РАУДОВЫЙ SM4

Проверка данного алгоритма была начата с проверки правдивости перехода от уравнения (2.13) к уравнению (2.14). Действительно, переход оказался верным:

Кроме этого, проведена проверка верности выражения (2.14) с помощью программной реализации алгоритма шифрования SM4 на языке программирования Python 3.12. Для этого собрано 10 выборок по 10000 случайных пар текст/шифртекст, зашифрованных на одном ключе, и для каждой пары подсчитана доля побитовых совпадений левой и правой части уравнения (2.14). Результаты представлены в Таблице 4.1

Таблица 4.1 – Результаты проверки верности уравнения (2.14)

|  |  |
| --- | --- |
| Выборка | Доля совпадений |
| 1 | 0,41050 |
| 2 | 0,41004 |
| 3 | 0,40998 |
| 4 | 0,40827 |
| 5 | 0,40914 |
| 6 | 0,40767 |
| 7 | 0,40970 |
| 8 | 0,40966 |
| 9 | 0,41024 |
| 10 | 0,40886 |

Доля совпадений во всех выборках примерно одна и та же. И она значительно отклонена от . Это показывает, что линейная аппроксимация, представленная уравнением (2.14), действительно приближает 22-раундовый SM4 с высокой точностью.

## 4.2 ПРОВЕРКА ДИФФЕРЕНЦИАЛЬНОЙ АТАКИ НА 23-РАУНДОВЫЙ SM4

Проверка представленного в Приложении А семейства характеристик состоит из двух этапов.

На первом этапе написана программа на языке C++ для поиска всех членов множества . Получено (что равно ) уникальных членов данного множества, что совпадает с результатами, представленными в статье [5].

На втором этапе написана программа на языке Python 3.12 для проверки всех 16129 характеристик. Для этого сгенерированы 10 случайных пар текст/шифртекст. Каждый открытый текст суммируется по модулю 2 с вектором . Полученный вектор шифруется. Далее находится разность между шифртекстами изначального текста и полученного сложением с входным дифференциальным вектором. Данный выходной дифференциальный вектор побитово сравнивается с , и высчитывается доля совпадений. Данные действия повторяются для всех 16129 значений . Результаты были поделены на 16 выборок по 10000 результатов сравнения (в последней выборке их 11290). Доли совпадений для всех выборок представлены в Таблице 4.2.

Таблица 4.2 – Результаты проверки семейства характеристик из Приложения А.

|  |  |
| --- | --- |
| Выборка | Доля |
| 1 | 0.49200 |
| 2 | 0.49254 |
| 3 | 0.49186 |
| 4 | 0.49246 |
| 5 | 0.49128 |
| 6 | 0.49175 |
| 7 | 0.49322 |
| 8 | 0.49244 |
| 9 | 0.49158 |
| 10 | 0.49245 |
| 11 | 0.49197 |
| 12 | 0.49186 |
| 13 | 0.49226 |
| 14 | 0.49228 |
| 15 | 0.49238 |
| 16 | 0.492300 |

Доля совпадений во всех выборках примерно одна и та же. И она отклонена от . Это означает, что рассмотренное семейство дифференциальных характеристик действительно хорошо аппроксимирует 19 раундов SM4.

## 4.3 ОЦЕНКА РЕЗУЛЬТАТОВ КРИПТОАНАЛИЗА WHITE-BOX РЕАЛИЗАЦИИ SM4 НА ВОЗМОЖНОСТЬ ИХ ПРИМЕНЕНИЯ ДЛЯ УСПЕШНОЙ АТАКИ

Рассмотренная в статье [7] техника криптоанализа white-box реализации SM4 из статьи [6] методом использования функции коллизий позволяет снять большинство операций, до тех пор, пока не останутся булевы маски. Это значительно уменьшает уровень безопасности реализации. Кроме этого, данная техника позволяет восстановить маскированные ключи. Однако, восстановление исходного ключа на данный момент считается невозможным.

Следовательно, несмотря на существенное снижение уровня безопасности, восстановление исходного ключа шифрования остается нерешенной проблемой, что ограничивает полную компрометацию системы.

# ВЫВОД

Целью работы было рассмотреть различные методы криптоанализа, применимые к алгоритму SM4. В ходе проведенного исследования были рассмотрены такие методы, как линейный и дифференциальный криптоанализ, а также криптоанализ white-box реализации.

Анализ показал, что ни один из рассмотренных методов криптоанализа не демонстрирует значительного ослабления безопасности полных версий алгоритма SM4. Линейная и дифференциальная атаки, будучи мощными инструментами в теории криптоанализа, применимы лишь к редуцированным версиям SM4, содержащим 22 и 23 раунда соответственно. Даже при этом они обладают достаточно высокими требованиями по памяти, времени и количеству пар текст/шифртекст, что делает их малоэффективными в реальных условиях.

Кроме того, криптоанализ white-box реализации SM4, хотя и позволяет ослабить безопасность системы, не приводит к полной компрометации. Этот вид криптоанализа восстанавливает только маскированные ключи с неизвестной случайной маской, что существенно ограничивает возможность успешной атаки на защищенную систему.

Таким образом, гипотеза о существовании в открытом доступе эффективной атаки на полный алгоритм SM4, способной значительно ослабить его криптографическую стойкость, не подтвердилась. На текущий момент SM4 остается надежным криптографическим стандартом, однако продолжение исследований в области криптоанализа остается актуальным для поддержания и улучшения уровня его безопасности.

.

# СПИСОК ЛИТЕРАТУРЫ

1. GMT 0002-2012 SM4分组密码算法 [Электронный ресурс]. Режим доступа: <https://github.com/guanzhi/GM-Standards/blob/master/GMT%E5%AF%86%E7%A0%81%E8%A1%8C%E6%A0%87/GMT%200002-2012%20SM4%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81%E7%AE%97%E6%B3%95.pdf> (Дата доступа 14.03.24).
2. Taehyun Kim, Jongsung Kim, Seokhie Hong, Jaechul Sung. Linear and Differential Cryptanalysis of Reduced SMS4 Block Cipher [Электронный ресурс]. Режим доступа: https://eprint.iacr.org/2008/281.pdf (Дата доступа 14.03.24).
3. Collard, Baudoin; Standaert, François-Xavier; Quisquater, Jean-Jacques. Improving the Time Complexity of Matsui's Linear Cryptanalysis [Электронный ресурс]. Режим доступа: <http://hdl.handle.net/2078.1/81795> (Дата доступа 24.05.24)
4. M. Matsui, Linear Cryptanalysis Method for DES Cipher [Электронный ресурс]. Режим доступа: <https://link.springer.com/chapter/10.1007/3-540-48285-7_33> (Дата доступа 25.05.24)
5. Bozhan Su, Wenling Wu, Wentao Zhang. Differential Cryptanalysis of SMS4 Block Cipher [Электронный ресурс]. Режим доступа: <https://eprint.iacr.org/2010/062.pdf> (Дата доступа 25.05.24)
6. YAO Si, CHEN Jie. A New Method for White-box Implementation of SM4 Algorithm [Электронный ресурс]. Режим доступа: <http://www.jcr.cacrnet.org.cn/EN/10.13868/j.cnki.jcr.000373> (Дата доступа 26.05.24)
7. Jiqiang Lu, Jingyu Li. Cryptanalysis of a Type of White-Box Implementations of the SM4 Block Cipher [Электронный ресурс]. Режим доступа: <https://eprint.iacr.org/2021/1586.pdf> (Дата доступа 26.05.24)

# ПРИЛОЖЕНИЕ А

Таблица А.1 – Семейство 19-раундовых дифференциальных характеристик.

| Раунд |  |  |  |  | Вероятность |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  | 1 |
| 3 |  |  |  |  | 1 |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  | 1 |
| 7 |  |  |  |  | 1 |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  | 1 |
| 11 |  |  |  |  | 1 |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  | 1 |
| 15 |  |  |  |  | 1 |
| 16 |  |  |  |  |  |
| 17 |  |  |  |  |  |
| 18 |  |  |  |  | 1 |
| 19 |  |  |  |  | 1 |