Пользуясь полученными представителями классов, посчитаем количество классов с фиксированным рангом r. Прослеживается закономерность в количествах классов для рангов, близких к n. Так, для $r = n - 1$ при $n \geq 3$ имеется 3 класса, для $r = n - 2$ при $n \geq 6$ имеется 15 классов, что подтверждает известные вычисления. Установлено, что для $r = n - 3$ имеется 69 классов при $n \geq 9$ и для $r = n - 4$ имеется 342 класса при $n \geq 12$. Отсюда можно сделать предположение, что количество классов с фиксированным рангом начиная с некоторого момента перестает меняться.

ПРИМЕНЕНИЯ ГЕНЕРАТОРОВ КОНСТАНТ ДЛЯ ЗАЩИТЫ ЦИФРОВЫХ СХЕМ

В.В. СЕРГЕЙЧИК, А.А. ИВАНЮК

Схемная обфускация представляет собой методику защиты цифровых устройств от обратного проектирования. Одним из методов схемной обфускации цифровых устройств является внедрение генераторов констант (ГК). ГК — разновидность непрозрачных предикатов. ГК представляют собой схемы, генерирующие одно логическое значение постоянно. Сложность реализации ГК заключается в создании нераспознаваемых и неминимизируемых средством синтеза схем. Примером ГК может быть элемент памяти, в котором вход асинхронного сброса дополнительно соединен с выходом вентилем *and*. Некоторые применения ГК: вместо сигнальных источников ‘0’ и ‘1’ в схеме; мнимое объединение несвязанных узлов схемы мультиплексором (при подключении к селекторному входу ГК всегда будет выбираться один и тот же путь); создание обфусцированных вариантов основных логических вентилей с использованием ГК и мультиплексора. Второй способ применения был экспериментально изучен для FPGA на тестовых наборах комбинационных схем ISCAS-85 и последовательностных схем ITC-99 с использованием описанных выше примитивов ГК, которые внедрялись в участки, не входящие в критический путь. При этом синтезаторы (Xilinx xst 8.1 и Leonardo Spectrum 2007), во-первых, не смогли минимизировать ГК, во-вторых, сгенерировали неверную информацию о критическом пути (проложив путь через никогда не выбираемый маршрут в мультиплексоре с ГК). Дополнительная задержка, вносимая ГК, равна сумме задержек мультиплексора и двух проводящих линий. Таким образом, с помощью ГК можно не только усложнить анализ схемы, но и ввести в заблуждение атакующего относительно временных характеристик схемы. Кроме того, дополнительно ГК можно использовать для внедрения водяных знаков в конфигурируемые логические блоки FPGA.

ЭЛЕМЕНТЫ КРИПТОАНАЛИЗА БЛОЧНОГО АЛГОРИТМА ШИФРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКОГО ХАОСА

А.В. СИДОРЕНКО, Д. А. ЖУКОВЕЦ

Широкое распространение информационных технологий практически во всех сферах жизнедеятельности человека способствует появлению новых задач, связанных с обеспечением необходимой степени защиты информации.

Среди различных методов защиты информации и обеспечения ее целостности выделяются криптографические методы. Одним из перспективных направлений в современной криптографии является разработка алгоритмов шифрования на основе динамического хаоса [1]. Применение динамического хаоса для систем защиты информации обусловлено способностью с помощью хаотических отображений обеспечивать скрытость передачи зашифрованной информации и, прежде всего, в блочных шифрах. Детерминизм хаоса способствует шифрованию информации, а его случайность делает систему стойкой к вскрытию.

Использование же криптографических средств в практически реализуемых системах неразрывно связано со стойкостью алгоритмов шифрования. В вероятностных терминах стойким считается алгоритм, в котором перехват зашифрованных сообщений не приводит к появлению точки единственного принятия решения об используемом ключе или переданном открытом сообщении.

При традиционном подходе стойкость алгоритма шифрования определяется стойкостью к известным видам криптографических атак, применяемых с целью прочтения, замены зашифрованного сообщения или вычисления ключа шифрования. При этом дифференциальный и линейный методы криптоанализа относятся к наиболее известным. Дифференциальный метод заключается в анализе пар открытого и зашифрованного текстов, между которыми существует определенная разность, вычисляемая, как правило, при помощи операции сложения по модулю два. Анализируя вероятности появления определенных разностей на выходе одного раунда преобразования в зависимости от разности на его входе, выделяют дифференциальные характеристики, при которых наиболее вероятная разность на выходе одного раунда соответствует определенной разности на входе следующего раунда. Далее проводится анализ пар текстов по накоплению статистики о возможном значении ключа шифрования или открытого текста.

В данной работе приведены основные параметры результатов алгоритма поиска правильных пар текстов по заданному дифференциалу для проведения анализа n-раундового алгоритма шифрования с использованием динамического хаоса.

Литература

1. *Сидоренко А.В., Мулярчик К.С. Шакинко И.В*. Вестник БГУ. Сер. 1 Физика, математика, информатика. № 4. 2012. С. 44–50.

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ТЕСТИРОВАНИЯ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Е.В. СТАВЕР

Одной из актуальных задач криптографии является задача исследования статистических свойств бинарных последовательностей, используемых для создания ключей криптографических алгоритмов. Так в основу данной раззработки положен американский стандарт SP 800-90B. Документ SP 800-90B определяет требования к тестированию случайных последоввательнстей, полученных с физических датчиков. Данный программный продукт представляет собой пакет для тестирования битовых последовательностей, согласно документу NIST SP800-90B. Пакет состоит из динамической библиотеки с открытым интерфейсом для тестирования и программы-оболочки над ней. Среда разработки — Microsoft Visual Studio Express 2008, используемый язык программирования — C++

2 Функциональные возможности

Возможность выбрать набор из нескольких тестов для оценки заданной случайной последовательности:

ApproximateEntropy

ChiSquare

Collision

Сохранение результатов тестирования последовательностей в LOG-файл.

Результаты тестирования передаются в Excel и отображаются в виде диаграммы с выводом основных параметров тестов и результатом прохождения или не прохождения набора тестов последовательности.