Рыбальченко В.А., Зорин Р.С., Ростенко Д.С.
Аппаратная реализация легковесных криптографических алгоритмов
шифрования для устройств с малой вычислительной мощностью

*Южный федеральный университет
(Россия, Таганрог)*

doi: 10.18411/lj-09-2020-04

idsp: ljjournal-09-2020-04

Аннотация

Статья является обзором малоресурсной криптографии. Отражена проблема увеличения ограничений на используемые вычислительные ресурсы, энергетические затраты, энергетические мощности. Обоснована актуальность использования легковесной криптографии. Дан анализ необходимости применения криптографии в сфере "Интернет вещей". Приведено сравнение аппаратных реализаций шифров.

Ключевые слова: легковесная криптография, RFID-метки, алгоритм шифрования, аппаратная реализация алгоритма шифрования.

Abstract

This article is a review of low-resource cryptography. The problem of increasing restrictions on the used computing resources, energy costs, and power capacity is reflected. The relevance of using lightweight cryptography is proved. An analysis of the need to use cryptography in the field of "Internet of things" is given. A comparison of hardware implementations of ciphers is given.

Keywords: lightweight cryptography, RFID tags, encryption algorithm, hardware implementation of the encryption algorithm.

Главной особенностью современного этапа развития Интернета является увеличение количества различных вычислительных устройств, имеющих доступ в интернет. В силу условий их эксплуатации, а также жестких ценовых ограничений, присущих массовому производству, эти устройства характеризуются значительными ограничениями по используемым ресурсам памяти, вычислительной мощности, источникам питания и так далее. Это приводит к ограничениям на используемые технологии и технологические решения, применяемые к низкоресурсным криптографическим средствам. Например, строгие ограничения накладываются на энергопотребление реализации криптографических алгоритмов для пассивных интеллектуальных устройств, таких как радиочастотные метки или бесконтактные смарт-карты.

Типичные ограничения, обнаруживаемые в низкоресурсной криптографии, заключаются в следующем:

- размер микросхемы, потребляемая мощность, время, затрачиваемое на выполнение программы (для аппаратной реализации);
- размер программного кода, размер оперативной памяти и время, затраченное на выполнение программы (для программной реализации).

Могут появиться и другие ограничения. Таким образом, в зависимости от конкретных условий применения разработанного средства важными могут быть такие характеристики, как полоса пропускания рабочих частот канала связи [1].

Для понимания назначения и сферы применения облегченных (низкоресурсных) криптографических алгоритмов необходимо остановиться на понятии "Интернет вещей". Итак, "Интернет вещей" - это беспроводная самонастраивающаяся сеть между такими объектами, как бытовая техника, транспортные средства, различные датчики и сенсоры, а также радиочастотные идентификационные (RFID) метки.

В связи с этим особую актуальность приобретает задача эффективной реализации алгоритмов защиты информации, обеспечивающих конфиденциальность и целостность данных. Очевидно, что в основе такой безопасности должны лежать криптографические методы защиты информации. Массовый характер применения и малые потоки передаваемых данных привели к необходимости использования алгоритмов "облегченной" криптографии, для реализации в устройствах с ограниченными вычислительными возможностями.

Реализация облегченной криптографии требует определенных требований: низкое энергопотребление; реализация надлежащего уровня безопасности; малый размер микросхемы; обработка небольших потоков информации с приемлемой производительностью; низкая цена устройства. В то же время криптографическая устойчивость алгоритмов не должна существенно снижаться.

Цель разработчика - найти баланс между производительностью, безопасностью и ценой.

Криптографические примитивы - это основные криптографические инструменты, обеспечивающие выполнение определенных криптографических сервисов. Они обычно делятся на примитивы с секретным ключом (симметричные примитивы), примитивы с открытым ключом (асимметричные примитивы) и примитивы без ключа [2].

Как упоминалось ранее, одной из технологий, использующих облегченную криптографию, является RFID. RFID-это метод автоматической идентификации объектов, при котором данные, хранящиеся в так называемых RFID-метках, считываются или записываются с помощью радиосигналов.

Любая RFID система состоит из считывателя и RFID метки [3]. Большинство RFID-меток состоят из двух частей: интегральной схемы для хранения и обработки информации и антенны для приема и передачи сигнала. По типу источника питания RFID-метки делятся на следующие категории:

1. Пассивные;
2. Активные;
3. Полупассивные.

Пассивные RFID-метки не имеют встроенного источника питания. Электрический ток, индуцируемый в антенне электромагнитным сигналом от считывателя, обеспечивает достаточную мощность для работы кремниевого чипа, помещенного в этикетку, и передачи ответного сигнала.

Активные RFID-метки имеют собственный источник питания и не зависят от энергии считывателя, поэтому они читаются на большом расстоянии, имеют большие размеры и могут быть оснащены дополнительной электроникой. Однако эти метки являются самыми дорогими, а время автономной работы ограничено.

Полупассивные RFID-метки, также называемые полуактивными метками, очень похожи на пассивные метки, но оснащены батареей, которая обеспечивает питание чипа.

Существуют как блочные, так и потоковые облегченные алгоритмы. В активных RFID-системах активно используются алгоритмы потокового шифрования [2]. На данный момент существует четыре описанных проточных LW-шифра, которые имеют относительно приемлемые характеристики. Это алгоритмы Mickey, Bean, Trivium и Grain v1. На алгоритме Grain v1 мы остановимся подробнее.

Grain v1 - это симметричный алгоритм синхронного потокового шифрования, в первую очередь ориентированный на аппаратную реализацию. Схема шифрования состоит из трех блоков:

1. Регистр сдвига с линейной обратной связью (LFSR).

2. Регистр сдвига с нелинейной обратной связью (NFSR).
3. Выходной функции $h(x)$.

Следует пояснить, что регистр сдвига обратной связи представляет собой последовательность битов, которая сдвигается вправо на одну цифру в каждом такте шифрования, причем выход из самого правого бита является выходом генератора, а вход самого левого бита подается значение, вычисленное как функция отдельных битов регистра. Ключ шифрования поточного шифра вводится в регистр перед генерацией битового потока.

Первоначально 80 бит ключа загружаются в NFSR, 64-битный вектор инициализации, который дополняется 16 единицами, в LFSR. После 160 циклов простоя, в течение которых будут выполнены функции $f(x)$ на LFSR и $g(x)$ на NFSR, регистры будут полностью обновлены:

$$f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80}, \quad (1)$$

$$g(x) = 1 + x^{18} + x^{20} + x^{28} + x^{35} + x^{43} + x^{47} + x^{52} + x^{59} + x^{66} + x^{71} + (x^{80} + x^{17}x^{20}) + \\ + (x^{43}x^{47} + x^{65}x^{71}) + (x^{20}x^{28}x^{35}) + (x^{47}x^{52}x^{59}) + (x^{17}x^{35}x^{52}x^{71}) + (x^{20}x^{28}x^{43}x^{47}) + \\ + (x^{17}x^{20}x^{59}x^{65}) + (x^{17}x^{20}x^{28}x^{35}x^{43}) + (x^{47}x^{52}x^{59}x^{65}x^{71}) + (x^{28}x^{35}x^{43}x^{47}x^{52}x^{59}). \quad (2)$$

Тогда аргументы функции $h(x)$ будут равны 4 битам из LFSR и 1 биту из NFSR. В результате на выход поступает бит, в процессе вычисления которого задействуется результат функции $h(x)$ и сумма битов из NFSR.

Структура шифра Grain v1 представлена на рисунке 1. Шифр сконструирован таким образом, что невозможно атаковать быстрее, чем полный поиск ключей. Так что лучшая атака должна иметь порядок сложности 2^{80} .

Рисунок 1- Структура алгоритма шифрования Grain v1

Алгоритм работы Grain v1 был перенесен на язык описания и моделирования электронных схем Verilog, далее, используя программное обеспечение Xilinx ISE был произведена синтез аппаратной части Grain v1. Для этого была выбрана программируемая логическая интегральная схема Spartan-6 XC6SLX4, на ней были проведены тесты по шифрованию.

Пропускная способность Grain v1, по сравнению с другими алгоритмами, достаточно велика. Результаты аппаратной реализации других потоковых шифров [1] приведены в таблице 1 для сравнения с Grain v1.

Таблица 1.

Сравнение результатов аппаратной реализации различных поточных шифров.

Алгоритм	Nk	Тактов на инициализацию	GE	Cl/byte
Enocoro-128 v2	128	4869	4100	46.3
Grain v1	80	321	1294	31.2
Mickey-128 v2	128	56592	5000	1231.4
Trivium	80	1333	3488	8.0
AES-CTR	128	470	4400	17.8

Обозначения: Nk - длина ключа (в битах), GE - условные логические элементы (Gate Equivalent), Cl/byte - число тактов работы алгоритма на байт зашифрованной информации – мера скорости работы алгоритма, пропускная способность.

Рассмотренные криптографические алгоритмы и их реализация построены и применяются для обеспечения информационной безопасности соответствующих систем с учетом их взаимодействия с интернетом и сетевыми технологиями [4].

Проанализировав результаты аппаратной реализации различных алгоритмов шифрования, становятся видны преимущества алгоритма Grain v1. Он занимает наименьшее количество логических элементов среди рассмотренных шифров. Так же ему требуется меньшее количество тактов на инициализацию, и он имеет достаточно хороший показатель скорости работы, что позволяет использовать его для различных устройств "Интернета вещей".

1. Жукова, А.Е. Легковесная криптография. Части 1 и 2/ А.Е. Жукова. [Электронный ресурс]. – URL: https://cyberrus.com/wp-content/uploads/2015/05/vkb_10_01.pdf
2. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ П.П. Урбанович. - Минск: БЕТУ, 2016. - 220 с.
3. LW-криптография: шифры для RFID-систем. [Электронный ресурс]. – URL: <https://habr.com/ru/post/119700/>
4. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. - Минск: БГТУ, 2011. - 399 с.

Шавтикова Л.М.**Реализация алгоритма kNN на языке программирования Python**Северо-Кавказская государственная академия
(Россия, Черкесск)

doi: 10.18411/lj-09-2020-05

idsp: ljjournal-09-2020-05

Аннотация

В данной статье рассматривается реализация одного из простых алгоритмов машинного обучения - k-nearest neighbors (кближайших соседей) для задачи классификации на Python с использованием библиотек. Важной частью машинного обучения является предсказание значений на новых данных.

Ключевые слова: машинное обучение, алгоритм, выборка, диаграмма, графика, визуализация.

Abstract

This article discusses the implementation of one of the simple machine learning algorithms - k-nearest neighbors for the classification problem in Python using libraries. An important part of machine learning is predicting values on new data.

Keywords: machine learning, algorithm, selection, diagram, graph, visualization.