ЛЕГКОВЕСНАЯ КРИПТОГРАФИЯ
Часть 1

***Жуков Алексей Евгеньевич**, кандидат физико-математических наук, доцент, г. Москва*

Статья посвящена обзору алгоритмов малоресурсной (легковесной) криптографии, стойкость которых снижается незначительно, в отличие от объема требуемых ресурсов. Дан полный обзор литературы в области малоресурсной криптографии. Рассмотрены приложения малоресурсной криптографии. Дан анализ применения малоресурсных блочных и поточных шифров и хэш–функций. Приведено сравнение программных и аппаратных реализаций известных шифров. Показана эффективность использования малоресурсной криптографии с ассиметричных системах. Сделан вывод об обнадёживающих перспективах малоресурсной криптографии. Даны рекомендации по развитию направления малоресурсной криптографии в России.

***Ключевые слова**: криптография, мало ресурсная криптография, легковесная криптография, блочные шифры, стойкость легковесной криптографии, эффективность легковесной криптографии, программные шифры, аппратаные шифры, интернет вещей.*

LIGHTWEIGHT CRYPTOGRAPHY
Part 1

***Aleksey Zhukov**, Ph.D (in Math.),*
Associate Professor, Moscow

The algorithms lightweight cryptographic resistance which decreases slightly in contrast to the resources required are considered. The complete review of the literature in the field of lightweight cryptography is given. Applications of lightweight cryptographic are analyzed. The analysis of the application block and stream ciphers and hash functions are done. The comparison of software and hardware implementations of known ciphers is shown. The efficiency of using lightweight asymmetric cryptography systems is analyzed. The encouraging prospects lightweight cryptography are concluded. The recommendations for development of lightweight cryptography in Russia are given.

***Keywords**: cryptography, low resource cryptography, lightweight cryptography, block ciphers, resistance lightweight cryptography, efficiency lightweight cryptography, software ciphers, hardware ciphers, Internet of Things.*

Введение.

Развитие криптографии в течение последнего столетия во многом определялось развитием средств связи и информационных технологий. Так, открытие возможности передачи информации с помощью электрических сигналов и радиоволн привело к созданию в начале XX века дисковых шифраторов. Появление новых технологий, основанных на использовании полупроводников (первый полупроводниковый транзистор был создан в 1947 г.), привело к появлению шифров, реализуемых электронными логическими схемами, использующими, например, такие конструктивные узлы, как регистры сдвига. Возникновение компьютеров и соответствующей элементной базы (первые микропроцессоры появились в начале 70-х годов XX века) привело к созданию шифров нового поколения – блочных шифров.

Развитие криптографии, начиная с последней четверти XX века, в основном определялось и, по всей видимости, в ближайшем будущем будет определяться развитием Интернета и интернет-технологий. При этом определяющим на ближайшие годы направлением развития Интернета будет так называемый *Интернет Вещей* (Internet of Things, IoT), принятый комиссиями Европарламента и Совета Европы в качестве магистрального пути развития информационных и интернет технологий[1]. Это направление характеризуется пере-

1 www.internet-of-things-research.eu/documents.htm

ходом от Интернета Персональных Компьютеров к Интернету Вещей.

От Интернета Персональных Компьютеров к Интернету Вещей

> «The upcoming era of pervasive computing will be characterized by many smart devices that – because of the tight cost constraints inherent in mass deployments – have very limited resources in terms of memory, computing power, and battery supply.»
>
> Christof Paar, Axel Poschmann, et al.[2]

Всеми исследователями отмечается стремительный рост объема передаваемого интернет-трафика. В соответствии с данными презентации, сделанной в августе 2010 г. в исследовательском центре PARC корпорации Xerox вице-президентом компании Google Мариссой Майер, в 2002 году в сети находилось 5 эксабайтов данных (один эксабайт – 2^{60} байт), а в 2009 году уже 281 эксабайт, т.е. за 7 лет это число возросло в 56 раз. [2]. Но данные в сети создает не только человек. Уже сейчас 99.8% всех изготовленных (микро) процессоров используется во встроенных системах (all embedded CPUs 4,…,32 bit) и лишь 0.2% – в традиционных компьютерах (PC & workstation CPUs 32 bit) [3]. В последние годы наряду с традиционными интернет-устройствами, такими как персональные компьютеры, ноутбуки, смартфоны, стали появляться устройства бытовой техники, транспорта, а также различные датчики, имеющие доступ в Интернет. Это явление получило название «Интернет вещей»[3]. Интернет вещей представляет собой беспроводную самоконфигурирующуюся сеть между объектами типа бытовых приборов, транспортных средств, различных сенсоров и датчиков, а так же меток радиочастотной идентификации (**R**adio **F**requency **ID**entification, RFID).

Исследователи корпорации Cisco IBSG прогнозируют, что к 2015 году к Интернету будет подключено 25 миллиардов, а к 2020 году – 50 миллиардов различных устройств. В их число входят радиочастотные метки (RFID-Tags)[4], бесконтактные смарт-карты (smart cards), SIM-карты, средства системы глобальной мобильной связи (GSM, Global System for Mobile communications), средства автоматизированных систем управления технологическими процессами (SCADA – **S**upervisory for **C**ontrol **A**nd **D**ata **A**cquisition), беспроводные сенсоры (wireless sensors) в том числе имплантированные медицинские сенсоры и прочие устройства (medical sensors and defibrillators, insulin pumps, deep brain stimulators – стимулятор мозга, pacemakers – электронный стимулятор сердца), электронные паспорта (electronic passports) и прочие электронные средства идентификации личности (authentication of document owner by eID document), средства логистики (logistics) и др. средства автоматизации поставок (automated management of the supply chain), системы проведения банковских операций через Интернет (internet banking), автоматическая оплата пошлин, услуг, дорожных и прочих сборов (automatic tolls), общественный транспорт (public transportations), борьба с контрафакцией (prevention of counterfeiting), противоугонные системы автомобилей (car key systems), средства контроля за авиабагажом (airline luggage tracking), библиотечными книгами (library management) и т.д.

Легковесная криптография

Стремительное развитие указанных технологий делает чрезвычайно актуальными вопросы, связанные с их информационной безопасностью. Так, директор ЦРУ Дэвид Петрэус[5] заявил, что данные с подключенных к Интернету бытовых приборов можно использовать для составления максимально подробного досье на любого человека[6].

В презентации для MIT Media Lab, сделанной Исследовательской группой по доверенным системам (Trusted Systems Research Group)Агентства Национальной Безопасности США (National Security Agency) 30 января 2013 г. говорится: «RFID-технологии развиваются чрезвычайно быстро. Входя в состав систем определения точного местоположения, имеющих выход в глобальные сети связи, радиочастотные метки стали к 2013 г.

2 Грядущая эпоха всеобъемлющего распространения вычислительной техники будет характеризоваться многочисленными интеллектуальными устройствами, которые из-за жестких ограничений на стоимость, присущих массовому производству, будут иметь весьма ограниченные ресурсы с точки зрения памяти, вычислительной мощности и источников питания питания. – Кристоф Паар, Аксель Поршман и др.

3 Термин «Интернет вещей» был, по-видимому, впервые введен в обращение в 1999 г. Кельвином Эштоном (Kelvin Ashton), в то время главным технологом Массачусетского Технологического Института.

4 По данным IdTechEx'2011 в 2015 г. будет продано 2 миллиарда активных RFID-меток и триллион пассивных. Интересно, что крупнейшим потребителем RFID-меток является сеть супермаркетов Walmart в то время, как на втором месте идет министерство обороны США.

5 David Petraeus – американский генерал, с сентября 2011 г. по ноябрь 2012 г. занимал должность директора ЦРУ.

6 www.wired.com/2012/03/petraeus-tv-remote/

чрезвычайно мощным средством для идентификации, определения положения и слежения за отдельными людьми или объектами» – NSA WikiInfo.

Все это, конечно, позволит эффективнее обнаруживать террористов, но, вместе с тем, сбор данных коснется и подавляющего большинства законопослушных граждан. Добавим к этому многочисленные приложения, связанные с обработкой биометрических данных, персональных данных медицинского характера, важной финансовой информации и др. В связи с этим особенно актуальной становится задача эффективной реализации алгоритмов защиты информации, обеспечивающих конфиденциальность и целостность данных. Очевидно, что основу такой безопасности должны образовывать криптографические методы защиты информации. И основным средством обеспечения информационной безопасности в мире Интернета Вещей является так называемая «*легковесная криптография*» (lightweight cryptography, LWC).

В русском языке уже отчасти прижился термин «легковесная криптография» или даже «легкая криптография», хотя эти термины не кажутся слишком удачными. Слово «легковесный» имеет в русском языке оттенок «легкомысленный, несерьезный», а слово «легкий» вообще является синонимом к слову «простой». Между тем LWC отнюдь не является «простой». Аксель Пошманн, известный специалист по Lightweight Cryptography, характеризует ее словами «As light as a feather, and as hard as dragon-scales» – «Легка, как пух и прочна как чешуя дракона»[7].

Более удачными, на наш взгляд, являются термины «*низкоресурсная*» или «*малоресурсная криптография*», как более точно отражающие суть дела. Далее в работе мы будем использовать именно эту терминологию, оставляя в то же время аббревиатуру LWC.

Многочисленные исследования, посвященные низкоресурсной криптографии

Важность проведения исследований и развития средств LWC были особенно ясно осознаны в начале XXI века, тогда же появился и термин – LWC. К числу первых работ в области LWC можно отнести работы [4-6], которые были помимо прочего посвящены выработке требований, предъявляемых к средствам LWC, используемым в метках радиочастотной идентификации (RFID). Хотя возросший интерес к низкоресурсной криптографии для встроенных систем безусловно мотивируется развитием IoT, разработчики криптографических систем и раньше уделяли внимание экономии ресурсов, в особенности – размерам микросхемы. Так требования к алгоритмам, выдвигавшимся на конкурс eSTREAM по профилю 2 (аппаратно-ориентированные алгоритмы поточного шифрования) по сути, относились к требованиями, которые обычно предъявляют к средствам LWC.

Не смотря на огромное количество посвященных LWC публикаций, появившихся за последние 10 лет, на состоявшемся 11 июля 2012 г. совместном заседании рабочей группы по исследованию интеллектуальной архитектуры сетей и рабочей группы по компьютерной безопасности Национального института стандартов и технологий США (NIST SGIP-CSWG: Smart Grid Working Group – Cyber Security Working Group) говорилось о необходимости проведения исследований в области низкоресурсной криптографии, реализующей криптографическую защиту в миллионах устройств, снабженных весьма ограниченными вычислительными ресурсами.

Требования к средствам низкоресурсной криптографии

Как уже было отмечено, главной особенностью современного этапа развития Интернета является все возрастающее количество самых различных интеллектуальных устройств, имеющих доступ в Интернет. В силу условий их функционирования, а также жестких ценовых ограничений, свойственных массовому производству, эти устройства характеризуются значительными ограничениями на используемые ресурсы памяти, вычислительную мощность, источники питания и т.д. Отсюда следуют ограничения на используемые технологии и технологические решения, предъявляемые к средствам низкоресурсной криптографии.

Так, например, жесткие ограничения накладываются на энергозатратность реализации криптографических алгоритмов для пассивных интеллектуальных устройств таких, как радиочастотные метки или бесконтактные смарт-карты. В соответствии со стандартом ISO/IEC [8] пассивные RFID-метки должны иметь уровень энергопотребления не более 15 μW для того, чтобы гарантировать работу устройства в радиусе до 1 м. Послед-

7 см. Толкиен, «Властелин Колец»

8 ISO/IEC 18000-3:2004 Information technology – Radio frequency identification for item management – Part 3: Parameters for air interface communications at 13.56 MHz

нее, в свою очередь, ограничивает возможности, например, в распараллеливании вычислений с целью увеличения быстродействия алгоритма.

Другой пример ограничений дают системы автоматического осуществления дорожных сборов (платы за проезд по платным дорогам): для этих систем автомобиль, двигающийся с большой скоростью, должен быть идентифицирован (authenticate) считывающим устройством на значительном расстоянии (10-12 м.) и за весьма непродолжительное время (менее 10 мс.). Ясно, что в этом случае скорость работы значительно более существенны, чем размеры микросхемы или ее энергопотребление.

Таким образом, типичными ограничениями, встречающимися в низкоресурсной криптографии, являются: для аппаратной реализации – размер микросхемы, потребляемая энергия, время, затраченное на исполнение программы; для программной реализации – размер программного кода, размер оперативной памяти, время, затраченное на исполнение программы. Могут появляться и другие ограничения. Так в зависимости от конкретных условий применения разрабатываемого средства важной может оказаться такая характеристика, как ширина полосы рабочих частот канала связи.

Каждый проектировщик в области низкоресурсной криптографии должен стремиться найти баланс между безопасностью, ценой и производительностью (рис. 1). Обычно легко оптимизировать любые две из трёх целей разработки – безопасность и стоимость, безопасность и производительность, или стоимость и производительность; однако, очень тяжело оптимизировать эти три параметра одновременно. Например, безопасная и высокопроизводительная аппаратная реализация может быть достигнута на конвейерной архитектуре, устойчивой к утечке информации по побочным каналам, что ведет к увеличению размера микросхемы и соответственно росту ее стоимости. С другой стороны, можно спроектировать безопасное, недорогое оборудование имеющее, однако, ограниченную производительность.

Наконец, эффективность реализации того или иного преобразования на программном или аппаратном уровне оценивается по-разному. Для сравнения программных реализаций принято рассматривать требования к памяти и время работы, измеряемое в тактах процессора. Для аппаратной реализации критерием эффективности является прежде всего размер микросхемы и время работы в тактах процессора, хотя для очень многих приложений немаловажным фактором является энергопотребление устройства.

Отметим, что многие требования, предъявляемые к алгоритмам, предназначенным к использованию в низкоресурсных условиях, были закреплены в рамках международного стандарта ISO/IEC FDIS 29192 – Information technology – Security techniques – Lightweight cryptography.

Part 1: General
Part 2: Block ciphers
Part 3: Stream ciphers
Part 4: Mechanisms using asymmetric techniques

ISO/IEC 29192 является международным стандартом, определяющим средства низкоресурсной криптографии для обеспечения секретности, аутентичности, идентефикации, неотказуемости и ключевого обмена (data confidentiality, authentication, identification, non-repudiation, and key exchange).

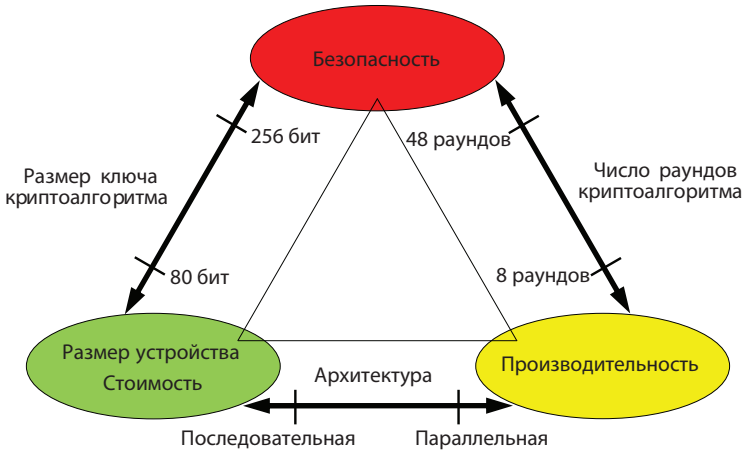

Рис. 1

Криптографические примитивы

Криптографические примитивы (cryptographic primitives) являются основным криптографическим инструментарием, который обеспечивает выполнение тех или иных криптографических сервисов. Обычно их подразделяют на примитивы с секретным ключом (симметричные примитивы), примитивы с открытым ключом (асимметричные примитивы) и бесключевые примитивы.

Примитивы с секретным ключом. Для таких примитивов криптографическая стойкость обеспечивается секретностью ключа, который должен оставаться неизвестным для всех участников информационного процесса, не имеющих соответствующих полномочий. В этот класс входят такие примитивы, как симметричные шифры (которые, в свою очередь, подразделяются на блочные шифры и поточные шифры), ключевые хэш-функции, называемые также кодами проверки подлинности сообщения или имитовставками (keyed hash function, **M**essage **A**uthentication **C**ode – MAC) а также криптографические генераторы псевдослучайных последовательностей.

Примитивы с открытым ключом. Примитивы из этого класса используют пары ключей (e,d) – (ключ шифрования, ключ расшифрования) или (открытый ключ, секретный ключ). Криптографическая стойкость обеспечивается секретностью секретного ключа, который должен оставаться неизвестным для всех участников информационного процесса, не имеющих соответствующих полномочий. В этот класс входят такие примитивы, как протоколы выработки и согласования ключей, асимметричные шифры (шифры с открытым ключом), схемы цифровой подписи и некоторые другие.

Бесключевые примитивы. В этот класс входят примитивы, не использующие ключей. Такие примитивы используются в таких криптографических сервисах, как аутентификация и обеспечение целостности информации. В этот класс входят такие примитивы, как однонаправленные подстановки (one-way permutation), хэш-функции, бесключевые хэш-функции или коды обнаружения модификации информации (unkeyed hash function, **M**odification **D**etection **C**ode – MDC), генераторы случайных последовательностей и проч.

Далее будут рассмотрены результаты низкоресурсной реализации основных криптографических примитивов. При этом мы попытаемся осуществить сравнительный анализ примитивов, как «общего пользования» (но получивших «экономную реализацию»), так и «облегченные» модификации примитивов общего пользования вместе со специально разработанными «легковесными» алгоритмами. Основной упор будет делаться на вопросы «легковесности», как специфические для LWC.

Безусловно важнейшей характеристикой криптоалгоритма является его криптостойкость. Однако анализ стойкости и «легковесных» вариантов классических алгоритмов и специально разработанных легковесных алго-

Рис. 2

ритмов принципиально ничем не отличается от анализа криптоалгоритмов общего вида. Он не имеет какой-то особой специфики и проводится обычным для современного криптоанализа образом – осуществляется проверка стойкости алгоритма относительно известных на сегодняшний день методов криптоанализа (линейный криптоанализ, разностный криптоанализ, корреляционный анализ и т.д.). Отметим, что для средств низкоресурсной криптографии очень важна их стойкость по отношению к анализу по побочным каналам (Side Channel Attacks), что объясняется условиями их эксплуатации. Надо сказать, что большинство алгоритмов (и облегченные варианты классических и специально разработанные «легковесные» алгоритмы) демонстрируют практическую стойкость, т.е. показано, что известные методы криптоанализа не позволяют «сломать» алгоритм за время меньшее, чем взлом грубой силой, требующий перебора всего ключевого пространства алгоритма (тотальное опробование). Редким исключением является, например, алгоритм ГОСТ 28147-89, для которого ряд исследователей получили результаты, позволяющие взламывать его быстрее полного перебора. Тем не менее, алгоритм демонстрирует практическую стойкость – на его взлом требуется время порядка 2^{191} операций шифрования при наличии 2^{64} пар открытый текст–шифртекст, что делает предложенную атаку неосуществимой на практике [7].

Основным характеристиками реализации криптографического алгоритма являются сложность реализации и скорость работы. Скорость – очень важная характеристика во многих (но не всех) применениях – в свою очередь зависит не только от частоты работы процессора, но и от размеров микросхемы (в случае аппаратной реализации) поскольку криптографические примитивы, как правило, весьма удобны для распараллеливания. В свою очередь сложность характеризуется размером микросхемы в GE[9] (в случае аппаратной реализации) или размером программного кода в байтах и размером требуемой оперативной памяти (в случае программной реализации). Всеобъемлющее сравнение различных реализаций зависит от очень многих параметров, включая технологию, архитектуру и т.д. Получение «абсолютно лучших» реализаций вряд ли возможно хотя бы в силу отсутствия методов получения нижних оценок схемной сложности даже для простейших преобразований. Однако, по оценкам некоторых авторов, для ряда схем достигнут предел в области минимизации по площади.

Симметричные и асимметричные криптоалгоритмы будут рассмотрены по отдельности, ввиду того, что они имеют различные области применения. Симметричные алгоритмы, в основном, служат для шифрования, проверки целостности сообщений, аутентификации, в то время как асимметричные алгоритмы используются в основном для управления ключами и обеспечения неотказуемости. Асимметричные алгоритмы требуют значительно большего объема вычислений по сравнению с симметричными как при аппаратной так и при программной реализации. Разрыв в производительности на устройствах с ограниченными ресурсами (например, для 8-разрядных микроконтроллеров) огромен. Так оптимизированный асимметричный алгоритм на эллиптических кривых (ECC) выполняется от 100 до 1000 раз медленнее, чем стандартный симметричный шифр и имеет на два-три порядка более высокое энергопотребление.

Блочные шифры

Наиболее активная и наиболее продуктивная деятельность по разработке низкоресурсных криптоалгоритмов происходила в области алгоритмов блочного шифрования. За последние 10 лет было предложено множество низкоресурсных решений. При этом развитие этой области низкоресурсной криптографии шло по двум направлениям:

- эффективная реализации известных алгоритмов блочного шифрования (с, возможно, их небольшой модификацией в сторону «облегчения», но при условии сохранения или незначительного снижения их криптографических свойств).
- разработка новых блочных шифров, ориентированных на оптимальную реализацию на микропрограммном или аппаратном уровне.

Некоторые исследователи считают, что уже есть достаточно большой выбор «облегченных» блочных криптоалгоритмов, пригодных для прак-

9 Площадь микросхемы обычно измеряется в μm^2, однако этот параметр сильно зависит от используемых технологий и библиотек стандартных ячеек (standard cell library). Для того, чтобы иметь возможность сравнивать микросхемы, изготовленные по разным технологиям, размеры принято измерять в условных логических элементах (Gate Equivalent – GE). За один условный логический элемент принимается площадь, занимаемая элементом NAND с двумя входами. При этом «ультралегкой» (ultra-lightweight) называется реализация, требующая менее 1000 GE, «низкостоимостной» (low-cost) – требующая не более 2000 GE и «легковесной» (lightweight) – не более 3000 GE [1].

тического применения. В частности, в международный стандарт ISO/IEC 29192-2 (Block ciphers) включены два алгоритма:

- блочный шифр PRESENT (размер информационного блока – 64-бит, размер ключа – 80 или 128 бит);
- блочный шифр CLEFIA (размер информационного блока – 128-бит, размер ключа – 128, 192 или 256 бит).

В качестве «точки отсчета» для сравнения тех или иных реализаций блочных шифров приведем различные реализации «эталонного» алгоритма – блочного шифра AES.

Реализации AES.

Аппаратная реализация. Наиболее скоростная реализация алгоритма AES демонстрирует скорость до 70 Гбит/сек [8]. Такая реализация использует конвейерную архитектуру процессора и требует более 250,000 GE. В то же время наиболее компактная реализация этого алгоритма требует порядка 2,400 GE [9].

Программная реализация. Для стандартных процессоров известна реализация алгоритма AES, обеспечивающая скорость 7.6 тактов на байт на процессоре Intel Core 2 Q9550 или 6.9 тактов на байт на процессоре Intel Core i7 [10].

Программно-аппаратная реализация. Принятие стандарта AES вызвало разработку дополнительных команд для процессоров семейства Intel (2008 г.) Сходное расширение PadLock engine существует в микропроцессорах от VIA Technologies. Целью данного расширения является ускорение приложений, использующих шифрование по алгоритму AES что обеспечивает скорость шифрования порядка 0.75 тактов на байт [11].

Аппаратная реализация симметричных шифров

В таблице 1 приведены данные об аппаратной реализации наиболее известных блочных шифров.

Таблица 1

Обозначения: Nb – длина информационного блока (в битах); Nk – длина ключа (в битах); R – число раундов (циклов шифрования); GE – условные логические элементы (Gate Equivalent); Cl/bl – число тактов работы алгоритма, затрачиваемое на шифрование одного информационного блока – мера скорости работы алгоритма, пропускная способность; SPN – SP сеть; Feistel – схема Фейстеля; Lai-Massey – схема Лая-Месси. Алгоритм общего пользования, «облегченный» вариант алгоритма общего пользования, специально разработанный легковесный криптографический алгоритм.

Алгоритм Публикация алгоритма Структура	Nb	Nk	R	GE Публикация реализации	Cl/bl	
AES-128* (2001)[10] SPN	128	128	10	3,100[12] 3,488[13] 2,400[9] 3400[13] 5400	160 1,032 226 1,032 54	
CAMELLIA (2001) [14] Feistel	128	128	18 24	11,350[14]		
		192				
		256				

10 FIPS PUB 197: Advanced Encryption Standard, U.S. Department of Commerce, November 2001.

Алгоритм Публикация алгоритма Структура	Nb	Nk	R	GE Публикация реализации	Cl/bl	
CLEFIA-128* (2007) [15] GFN	128	128	18	2,678 2,893 2,996 4,950[15] 5,979[15] 2,488[16] 2,604[16]	 36 18	
		192		8,536[15]	22	
		256		8,482[15]	26	
DES* (1976) [11] Feistel	64	56	16	2,309[16] 3,000 [17]	144 28	
3-DES* (ANS X9.52, 1998) [18] Feistel [11]	64	168	48	2,309[16]		
DESX (2007) [16] Feistel	64	184	16+2	2,629[19]	144	
DESL (2007) [16] Feistel	64	56	16	1,848[16]	144	
DESXL (2007) [16] Feistel	64	184	16+2	2,168[19]	144	
GOST (1989) ГОСТ 28147-89 [12] Feistel	64	256	32	651[20] 800[20] 1000[20]	264 264 32	
HIGHT (2006) [21] GFN	64	128	32	2,608 [LLYC'09] 3,048[21]	34 34	
Hummingbird-2 (2012) [22]				3,220		
ICEBERG (2004) [23]	64	128	16	7,732[24] 5800	16	

11 FIPS 46, «Data encryption standard», U.S. Department of Commerce/National Bureau of Standards, National Technical Information Service, 1977. (Revised as FIPS 46-1:1988, FIPS 46-2:1993, FIPS 46-3:1999).

12 ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М.: 1989.

Алгоритм Публикация алгоритма Структура	Nb	Nk	R	GE Публикация реализации	Cl/bl	
IDEA* (1990) [25] Lai-Massey	64	128	8.5	44,708		
KASUMI (1999) [13] Feistel network	64	64 128	8	2,990		
KATAN (2009) [26]	32	80	254	802[26]	256	
	48		254	927[26]		
	64		254	1054[26]	254	
KTANTAN (2009) [26]	32	80		462[26]		
	48			588[26]		
	64			688[26]	254	
KLEIN (2011) [27] SPN	64	64	12	1365 1220[27]	96 207	
		80	16	2629[27] 1478[27]	17 271	
		96	20	2769[27] 1528[27]	21 335	
LBLOCK (2011) [28] Feistel	64	80	32	1320[28]		
LED (2011) [29] SPN	64	64	32	966[29] 2,695[29] 688[29] 2,354[29]	1,248 32 1,248 32	
		80	48	1,040[29] 2,780[29] 690[29] 2,354[29]	1,872 48 1,872 48	
		128	48	1,265[29] 3,036[29] 700[29] 2,354[29]	1,872 48 1,872 48	

13 3GPP Confidentiality and Integrity Algorithms; KASUMI Specification (10), 2011. www.3gpp.org

Алгоритм Публикация алгоритма Структура	Nb	Nk	R	GE Публикация реализации	Cl/bl	
mCrypton (2005) [30] SPN	64	64	12	2,420[30]	13	
		96		2,681 [30]	13	
		128		2,949 [30] 4,108	13	
MIBS (2009) [31] Feistel	64	64	32	1,396[31]		
		80		1,530[31]		
Piccolo (2011) [32] GFN	64	80	25	1,048[33] 1,499[33] 616[33] 1,051[33]	432 27 432 27	
		128	31	1,338[33] 1,776[33] 654[33] 1,083[33]	528 33 528 33	
				683 758 (1,260)		
PRESENT (2007) [34] SPN	64	80	31	1,075[35] 1570[34] 1000[36] 1169[36] 27,028[35]	547 32 563 563 1	
		128		1,391[35] 1884[35] 996	559 32	
PRINCE (2012) [37]	64	128	12	3,491 3779[37]	12	
PRINT (2010) [38]	48	80		503[38] 402[38]	48	
	96	160		967		
TEA (1994) [39] Feistel	64	128	64	1984[40]		

Алгоритм Публикация алгоритма Структура	Nb	Nk	R	GE Публикация реализации	Cl/bl	
XTEA (1998)	64	128	32	2,335[41] 3,490[26]		
TWINE (2012) [42] GFN	64	80	36	1503[42] 1799[42] 1116[42]	36 36 540	
		128		1866[42] 2285[42]	36 36	
SEA (2006) [43] Feistel	96	96	Var.	3,758[24] 1333[24]	93	
	108	108		4,003[43]	111	
	120	120		4,673[43]	1600	
	132	132		5,071[43]	121	
	144	144		5,761[43]		
SIMON (2012) [44]	32 48 64 96 128	64 72 96 128 144 192 256		763 838 1317 [44]		
SPECK (2012) [44]	32 48 64 96 128	64 72 96 128 144 192 256		884 984 1396 [44]		

Выводы. Блочные шифры. Аппаратная реализация.

Результаты реализации сильно рознятся, в зависимости от задачи, поставленной разработчикам:

- Наиболее компактными являются реализации алгоритмов ГОСТ 28147-89 – 615 GE, KATAN – 802-1054 GE (в зависимости от размера информационного блока), KTANTAN – 462-688 GE (в зависимости от размера информационного блока), Piccolo – 683 GE, PRESENT – 1075 GE, PRINT – 402-967 GE (в зависимости от размера информационного блока), SIMON&SPECK – 763-1396 причем для алгоритмов KATAN, KTANTAN, PRINT, SIMON и SPECK «рекордные» результаты дали реализации с 32- или 48-битными информационными блоками, что является уязвимостью для многих методов криптоанализа.

Реализации остальных алгоритмов показали результаты существенно большие 1000 GE – общепринятой верхней границы размера микросхемы для симметричного криптоалгоритма.

- Размеры микросхемы существенным образом зависят от архитектуры реализации, которая, в свою очередь, диктуется финальными целями, как то оптимизация по площади, оптимизация по скорости и т.д. Если первая достигается за счет использования сериальной архитектуры (area for speed – serial/parallel), обрабатывающей информацию байтами, высокая скорость достигается за счет распараллеливания и конвейерной обработки данных, что неминуемо приводит к увеличению размера микросхемы. Итеративная архитектура (iterative).

Увеличение производительности микросхемы за счет увеличения ее размеров (18 тактов на информационный блок при размере схемы 5979 GE против 36 тактов на информационный блок при размере схемы 4950 GE для алгоритма CLEFIA-128.

Типичным распределением ресурсов при аппаратной реализации является распределение ресурсов при аппаратной реализации алгоритма PRESENT (1570 GE): ключевая информация – 30%, S- и P-блоки – 29%, запись внутреннего состояния – 25%, операция XOR – 11%. При этом 55% ресурсов уходят на регистры памяти (ключ + внутреннее состояние) [34].

Программная реализация блочных шифров

Во многих приложениях, где стоимость и уровень потребления энергии имеют решающее значение, вычислительные устройства представляются в форме небольших и недорогих процессоров. В настоящее время на мировом рынке наибольшая доля вычислительных устройств приходится на 8-битные контроллеры. Эти небольшие вычислительные устройства как правило имеют ограниченную несколькими десятками килобайт программную память, менее 1 Кбайт статической ОЗУ, небольшую тактовую частоту (порядка нескольких МГц), маленькие размеры регистров, ограниченный набор выполняемых операций.

В этом контексте эффективность использования ресурсов, измеряемая в основном по потребляемой памяти, является более важной характеристикой, чем пропускная способность, особенно потому что большая часть встроенных приложений (систем) обрабатывают весьма малые объемы информации. Тем не менее, особенно для устройств с батарейным питанием, низкая вычислительная сложность тоже может иметь большое значение, потому что время обработки информации непосредственно взаимосвязано с энергопотреблением. Таким образом, быстрое выполнение алгоритма может снизить потребление энергии и продлить время работы батарейного устройства.

Эти небольшие вычислительные устройства имеют весьма ограниченную программную память (флэш-память), оперативную память, небольшую тактовую частоту, маленькие размеры регистров, ограниченный набор арифметических возможностей.

Далее приводится сравнение отдельных блочных шифров по таким параметрам, как размер кода и объем требуемой оперативной памяти [45]. Приводимые данные были получены для 8-разрядных микроконтроллеров типа AVR, которые являются распространённым семейством 8-разрядных RISC-микроконтроллеров. Результаты реализаций указанных алгоритмов отображены в таблице 2.

В таблице 2 приведены данные о программной реализации на указанной выше платформе различных блочных шифров.

Таблица 2.

Обозначения: Nb – длина информационного блока (в битах); Nk – длина ключа (в битах); R – число раундов (циклов шифрования); SPN – SP сеть; Feistel – схема Фейстеля; Lai-Massey – схема Лая-Месси.

Алгоритм **Структура**	**Nb**	**Nk**	**R**	**Вычислительные устройства**	**Размер кода в байтах** **ROM**	**SRAM в байтах** **RAM**	**Enc** **(cyc/ byte)**	**Dec** **(cyc/ byte)**	**PUBL.**
AES-128 (2001) SPN	128	128	10		2,606	224	415	464	[45]
					1659	33			[46]
					1,912	432	125	181	[47]
					1,143		1,489	1,930	[48]
					2,747		991	1,296	[48]

Алгоритм Структура	Nb	Nk	R	Вычислительные устройства	Размер кода в байтах ROM	SRAM в байтах RAM	Enc (cyc/ byte)	Dec (cyc/ byte)	PUBL.
IDEA (1990) Lai-Massey	64	128	8.5		596 836	0 232	338	1,924	[45] [46]
SEA (2006) Feistel	96	96	Var.		2,132 426 1904	0 24 249	805	805	[45] [46] [27]
TEA & (1994) Feistel	64	128	64		1,140 648 1354	0 24 13	784	784	[45] [46] [49]
XTEA					1160 1394	 11			[50] [49]
KATAN32					5,816	1,881			[49]
KATAN48					7,076	1,969			[49]
KATAN64					8,348 3260 2628	1,953 625 202			[49] [27] [27]
KTANTAN 32					10,516	614			[49]
KTANTAN 48					11, 764	702			[49]
KTANTAN 64					16,252	790			[49]
LED 64,128					2,648	41			[49]
DES (1976) Feistel	64	56	16		4,314	0	1,079	1,019	[45]
DESL (2007) Feistel	64	56	16	ATMEL ATMega128	3,098		1,046	986	[45]
DESXL (2007) Feistel	64	184	16+2	ATMEL ATMega128	3,192 820	0 48	1,066	995	[45] [46]
HIGHT (2006) Feistel	64	128	32		5,672 402 2510 2050	0 32 117 132	371	371	[45] [46] [27] [27]
PRESENT (2007) SPN	64	80	31	 ATmega163	936 1,000 841 854	0 18 16	1,340 6,936 80,784	1,405 8,197	[45] [46] [51] [35]
		128							

Алгоритм Структура	Nb	Nk	R	Вычислительные устройства	Размер кода в байтах ROM	SRAM в байтах RAM	Enc (cyc/ byte)	Dec (cyc/ byte)	PUBL.
KASU					1,264	24			[46]
LBlock					3,568	13			[49]
NOEK					364	32			[46]
KTAN					338	18			[46]
KLEIN					1,268	18			[46]
KLEIN64					5,486 2582	36 105			[49] [27]
KLEIN80					5,676 2672	38 107			[49] [27]
KLEIN96					5,862 2782	39 109			[49] [27]
Humm	16	256			1,532 2646 1822	 150 116	2,887	2,606	[48] [27] [27]
mCRYP					1,076	28			[46]
mCRYPTON64					2,726	18			[49]
mCRYPTON 96					2,834	20			[49]
mCRYPTON 128					3,108	24			[49]
MIBS64					3,184	29			[49]
MIBS80					3,138	16			[49]
CLEFIA 128					4,780	180			[49]
CLEFIA 192					5,010	268			[49]
CLEFIA 256					4,924	268			[49]
TWINE				ATmega163	1,304 2216	414 23	271	271	[42] [49]
PRINT	48	80			490 1,250		10,415 5,013	10,575 5,229	[48] [48]
	96	160			761 2,397		20,448 10,828	20,624 10,916	[48] [48
Piccolo128					2,510			91	[49]
Piccolo80					2,434			79	[49]
SKIPJACK					6628 2566 1542	19 133 130			 [27] [27]

Другой распространенной платформой для реализации легковесных блочных шифров являются ПЛИС (FPGA). Результаты реализаций на указанной платформе ряда блочных алгоритмов отображены в таблице 3.

Таблица 3.

Обозначения: Nb – длина информационного блока (в битах); Nk – длина ключа (в битах); R – число раундов (циклов шифрования); SPN – SP сеть; Feistel – схема Фейстеля; Lai-Massey – схема Лая-Месси.

Алгоритм **Структура**	**Nb**	**Nk**	**R**	**FPGA**	**Число Slices**	**Тактов на блок**	**Max. Freq. (MHz)**	**Проп. спос. (Mb/s)**	
AES-128 (2001) SPN	128	128	10	XC2S30-5	222	46		139	
				Spartan-3 XC3S50-5	393	534		16.86	
				Spartan-II XC2S30-6	522		60	166	[52]
				Spartan-3 XC3S2000-5	17,425		196.1	25,107	[53]
				Spartan-2 XC2S15-6	264	3900	67	2.2	[53]
				Spartan-2 XC2V40-6	1,214		123	358	[54]
				Spartan-3	1,800		150	1700	[55]
				Spartan-II	124			2.2	
				Virtex-II	146		123	358	
				Virtex-II	1780		78	1000	
				XCV1000-4 [56]	2151			390	
				XCVE2000-8 [57]	446			1000	
				XC2S30-5 [52]	222			139	
				XC2S30-6 [52]	222			166	
				VIRTEX2300E [58]	542			1450	
				XCV-100-4 [59]	496			417	
				XCV-600E-8 [59]	496			743	
				XCV812E [60]	2744			258.5	
				Virtex-2 [61]	1780			1000	
				XC3S500E -4 [62]	326			270	
CAMELLIA [11]	128			Spartan-3 XC3S50-5	318	875		18.41	
				XC4000XL [14]	874				

HIGHT (2006) Feistel	64	128	32	Spartan-3 XC3S50-5	91	160		65.48	
Hummingbird (2010)	16	256		Spartan-3 XC3S200-5	273		40.1	160.4	
ICEBERG (2004)	64	128		Virtex-2	631		–	1,016	[63]
IDEA (1990) Lai-Massey	64	128	8.5						
PRESENT (2007) SPN	64	80	31	Spartan-3 XC3S50-5	117	256		28.46	
				Spartan-3 XC3S400-5	176		258	516	
				Spartan-3 XC3S400-5	202		254	508	[35]
				Spartan-3E XC3S500	271		–	–	[64]
		128		Spartan-III XCS400-5	202		254	508	[35]
SEA (2006)	96	96	Var.						
Feistel	126			Virtex-II XC2V4000	424		145	156	[65]
TEA & XTEA (1994) Feistel	64	128	64	Spartan-3 XC3S50-5	254	112	62.6	36	
				Virtex-5 XC5VLX85-3	9,647		332.2	20,645	

Выводы. Блочные шифры. Программная реализация.

Результаты реализации сильно рознятся, в зависимости от задачи, поставленной разработчикам. Таблицы показывают, что подходящие блочные шифры, такие как Piccolo, TWINE, XTEA и AES имеют хорошую производительность, несмотря на компромисс между размером кода и количеством тактов на блок. Также мы видим, что аппаратно-ориентированные шифры (LED, PRESENT, KATAN и KTANTAN) показывают худшие результаты.

Продолжение в следующем номере журнала

Литература

1. Manifavas C., Hatzivasilis G., Fysarakis K., Rantos K. Lightweight Cryptography for Embedded Systems - A Comparative Analysis, SETOP'2013.
2. Mayer M. Innovation at Google: the physics of data, PARC Forum, 2009. URL: www.slideshare.net/PARCInc/innovation-at-google-the-physics-of-data
3. Paar C. Light-Weight Cryptography for Ubiquitous Computing. Securing Cyberspace Workshop IV: Special purpose hardware for cryptography –Attacks and Applications. University of California at Los Angeles, December 4, 2006.
4. Sarma S. E., Weis S. A., Engels D. W. Radio-frequency identification: Security risks and challenges. CryptoBytes 6, 1, pp. 2–9, 2003.
5. Sarma S. E., Weis S. A., Engels D. W. RFID systems and security and privacy implications. In CHES 2002, Lecture Notes in Computer Science v. 2523, pp. 454–469, 2003.
6. Weis S. A., Sarma S. E., Rivest R. L., Engels D. W. Security and privacy aspects of low-cost radio frequency identification systems. In Security in Pervasive Computing 2003, Lecture Notes in Computer Science v. 2802, pp. 201–212, 2004.
7. Dinur I., Dunkelman O., Shamir A. Improved Attacks on Full GOST. Report 2011/558, available via http://eprint.iacr.org/, 2011.
8. Hodjat A., Verbauwhede I. Minimum Area Cost for a 30 to 70 Gbits/s AES Processor. In IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2004), pp. 498–502, 2004.
9. Moradi A., Poschmann A., Ling S., Paar C., Wang H. Pushing the Limits: A Very Compact and a Threshold Implementation of AES", EUROCRYPT 2011, LNCS 6632, pp. 69-88, Springer-Verlag, 2011.

10. Kasper E., Schwabe P. Faster and Timing-Attack Resistant AES-GCM. CHES 2009, LNCS 5747, pp. 1–17, 2009.
11. Preneel B. Perspectives on Lightweight Cryptography," Inscript 2010, Shanghai, China, 20-24 October 2010.
12. Hamalainen P., Alho T., Hannikainen M., Hamalainen T. D. Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core. In Euromicro Conference on Digital System Design, IEEE Computer Society, pp. 577-583, 2006.
13. Feldhofer M., Wolkerstorfer J., Rijmen V. AES Implementation on a Grain of Sand. IEE Proceedings Information Security, vol. 15, no. 1, pp. 13-20, 2005.
14. Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms – Design and Analysis. Selected Areas in Cryptography (SAC), LNCS v. 2012, 2001, pp. 39-56, 2001.
15. Shirai T., Shibutani K., Akishita T., Moriai S., Iwata T., The 128-bit Blockcipher CLEFIA. In Proceedings of Workshop on Fast Software Encryption 2007 — FSE 2007, Lecture Notes in Computer Science v. 4593, pp. 181-195, Springer-Verlag, 2007
16. Akishita T., Hiwatari H. Very Compact Hardware Implementations of the Blockcipher CLEFIA. Selected Areas in Cryptography 2011– SAC 2011, LNCS 7118, pp. 278–292, 2012.
16. Leander G., Paar C., Poschmann A., Schramm K. New lightweight DES variants. In Fast Software Encryption - FSE 2007, Lecture Notes in Computer Science v. 4593, pp. 196-210, 2007.
17. Verbauwhede I., Hoornaert F., Vandewalle J., De Man H. Security and performance optimization of a new DES data encryption chip. IEEE Journal of Solid-State Circuits, 23(3), pp. 647-656, 1988.
18. Coppersmith D., Johnson D.B., Matyas S.M. A proposed mode for triple-DES encryption. IBM Journal of Research and Development, 40, pp. 253–261, 1996
19. Poschmann A., Leander G., Schramm K., Paar C. New lighweight crypto algorithms for RFID. In Proceedings of The IEEE International Symposium on Circuits and Systems 2007 - ISCAS 2007, pp. 1843-1846, 2007.
20. Poschmann A., Ling S., Wang H. 256 Bit Standardized Crypto for 650 GE GOST Revisited, In CHES 2010, Lecture Notes in Computer Science v. 6225, pp. 219-233, 2010.
21. Hong D., Sung J., Hong S., Lim J., Lee S., Koo B., Lee C., Chang D., Lee J., Jeong K., Kim H., Kim J., Chee S. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In Cryptographic Hardware and Embedded Systems - CHES 2006, Lecture Notes in Computer Science v. 4249, pp. 46-59, 2006.
22. Engels D., Saarinen M.-J. O., Schweitzer P., Smith E. M. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. The 7th International Work-shop on RFID Security and Privacy - RFIDSec 2011, Lecture Notes in Computer Science v.7055, A. Juels and C. Paar (Eds.), Berlin, Germany: Springer-Verlag, pp. 19-31, 2012.
23. Standaert F.-X., Piret G., Rouvroy G., Quisquater J.-J., Legat J.-D. Iceberg : An involutional cipher efficient for block encryption in reconfigurable hardware. In FSE, Lecture Notes in Computer Science v. 3017, pp. 279-299, 2004.
24. Mace F., Standaert F.-X., Quisquater J.-J. ASIC Implementations of the Block Cipher SEA for Constrained Applications. In Proceedings of the Third International Conference on RFID Security – RFIDSec, pp. 103-114. Citeseer, 2007.
25. Lai X., Massey J. L. A proposal for a new block encryption standard. In Advances in Cryptology - EUROCRYPT '90, Lecture Notes in Computer Science v. 473, pages 389-404. Springer, 1990.
26. J.-P. Kaps. Chai-tea, cryptographic hardware implementations of xtea. In INDOCRYPT 2008, volume 5365 of LNCS, pages 363-375. Springer-Verlag, 2008.
27. Gong Z., Nikova S., Law Y. W. KLEIN: A new family of lightweight block ciphers. The 7th International Workshop on RFID Security and Privacy – RFIDSec 2011, Lecture Notes in Computer Science v. 7055, pp. 1-18, 2011.
28. Wu W., Zhang L. LBLOCK: A lightweight block cipher. In Applied Cryptography and Network Security - ACNS 2011, Lecture Notes in Computer Science v. 6715, pp. 327-344, 2011.
29. Guo J., Peyrin T., Poschmann A., Robshaw M. J. B. The LED block cipher. The 13th International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2011, Lecture Notes in Computer Science v. 6917, pp. 326-341, 2011.
30. Lim C. H., Korkishko T. mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In Workshop on Information Security Applications - WISA 2005, Lecture Notes in Computer Science v. 3786, pp. 243-258, 2005.
31. Izadi M., Sadeghiyan B., Sadeghian S. S., Khanooki H. A. MIBS: A New Lightweight Block Cipher. In Cryptology and Network Security - CANS 2009, Lecture Notes in Computer Science v. 5888, pp. 334-348, 2009.
32. Shibutani K., Isobe T., Hiwatari H., Mitsuda A., Akishita T., Shirai T. Piccolo: An ultra-lightweight blockcipher. The 13th International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2011, Lecture Notes in Computer Science v. 6917, pp. 342-357, 2011.
33. Hiwatari H., Shibutani K., Isobe T., Mitsuda A., Akishita T., Shirai T. Compact Hardware Implementations of the Ultra-Lightweight Block Cipher Piccolo. ECRYPT Workshop on Lightweight Cryptography - November 2011. Pp. 49-70
34. Bogdanov A., Knudsen L., Leander G., Paar C., Poschmann A., Robshaw M., Seurin Y., Vikkelsoe C. PRESENT - An Ultra-Lightweight Block Cipher. In Proceedings of Workshop on Cryptographic Hardware and Embedded Systems — CHES 2007, Lecture Notes in Computer Science v. 4727, pp. 450-466, 2007.
35. Poschmann A. Lightweight Cryptography - Cryptographic Engineering for a Pervasive World. Ph.D. Thesis, Ruhr University Bochum, 2009.
36. Rolfes C., Poschmann A., Leander G., Paar C. Ultra-Lightweight implementations for smart devices - security for 1000 gate equivalents. In Proceedings of the 8th Smart Card Research and Advanced Application IFIP Conference CARDIS 2008, Lecture Notes in Computer Science v. 5189, pp. 89-103, 2008.
37. Borgho J., Canteaut A., Guneysu T., Kavun E. B., Knezevic M., Knudsen L. R., Leander G., Nikov V., Paar C., Rechberger C., Rombouts P., Thomsen S. S., Yalcin T. PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications - Proc. of Advances in Cryptology, LNCS, vol.7658, Dec. 2012,pp. 208-225.
38. Knudsen L., Leander G., Poschmann A., Robshaw M. J. B. PRINTcipher: A Block Cipher for IC-Printing. The 12th International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2010, Lecture Notes in Computer Science v. 6225, pp. 16-32, 2010.

39. Wheeler D. J., Needham R. M. TEA, a tiny encryption algorithm. In Fast Software Encryption – FSE 94, Lecture Notes 68. Watanabe D., Okamoto K., Kaneko T. A Hardware-Oriented Light Weight Pseudo-Random Number Generator Enocoro-128v2. The 2010 Symposium on Cryptography and Information Security, SCIS 2010, 3D1-3, 2010 (in Japanese).
40. P. Israsena. Securing ubiquitous and low-cost RFID using tiny encryption algorithm. In 1-st International Symposium on Wireless Pervasive Computing, 2006, pages 4. IEEE, 2006.
41. Yu Y., Yang Y., Fan Y., Min H. Security Scheme for RFID Tag. Technical Report, Auto-ID Labs white paper WP-HARDWARE-022.
42. Suzaki T., Minematsu K., Morioka S., Kobayashi E. TWINE: A lightweight block cipher for multiple platforms. In Selected Areas in Cryptography - SAC 2012, Selected Areas in Cryptography. Volume 7707 of Lecture Notes in Computer Science., Springer (2013), pp 339–354, 2012.
43. Standaert F.-X., Piret G., Gershenfeld N., Quisquater J.-J. SEA: A scalable encryption algorithm for small embedded applications. In Smart Card Research and Advanced Applications - CARDIS 2006, Lecture Notes in Computer Science v. 3928, pp. 222-236, 2006.
44. Beaulieu R., Shors D., Smith J., Treatman-Clark S., Weeks B., Wingers L. Performance of the SIMON and SPECK families of lightweight block ciphers. Tech. rep., National Security Agency, May 2012.
45. Eisenbarth T., Kumar S. S., Paar C., Poschmann A., Uhsadel L. A Survey of Lightweight Cryptography Implementations. IEEE Design & Test of Computers - Special Issue on Secure ICs for Secure Embedded Computing vol 24, no 6, pp 522-533, 2007
46. Thomas Eisenbarth, Zheng Gong, Tim Guneysu, Stefan Heyse, Sebastiaan Indesteege, Stephanie Kerckhof, Francois Koeune, Tomislav Nad, Thomas Plos, Francesco Regazzoni, Francois-Xavier Standaert, Loic van Oldeneel tot Oldenzeel. Compact Implementation and Performance Evaluation of Block Ciphers in ATtiny Devices. ECRYPT Workshop on Lightweight Cryptography - November 2011. pp. 70-86.
47. J. W. Bos, D. A. Osvik, D. Stefan. "Fast Implementations of AES on Various Platforms." SPEED-CC – Software Performance Enhancement for Encryption and Decryption and Cryptographic Compilers, 2009. Cryptology ePrint Archive, Report 2009/501 (2009), http://eprint.iacr.org/ 8.
48. Tino Kaufmann, Axel Poschmann. Enabling Standardized Cryptography on Ultra-Constrained 4-bit Microcontrollers. ECRYPT Workshop on Lightweight Cryptography - November 2011. pp. 255-276.
49. Cazorla M., Marquet K., Minier M. Survey and Benchmark of Lightweight Block Ciphers for Wireless Sensor Networks, Report 2013/295, 2013.
50. Rinne S., Eisenbarth T., Paar C. Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers. Software Performance Enhancement for Encryption and Decryption (SPEED 2007), Amsterdam, NL June 11-12, 2007.
51. M. Vogt, A. Poschmann, and C. Paar. Cryptography is Feasible on 4-Bit Microcontrollers - A Proof of Concept. In International IEEE Conference on RFID, pages 267–274, Orlando, USA, April 2009.
52. P. Chodowiec and K. Gaj. Very Compact FPGA Implementation of the AES Algorithm. In C.D. Walter, Ç.K. Koç, and C. Paar, editors, Cryptographic Hardware and Embedded Systems – CHES 2003, number 2779 in Lecture Notes in Computer Science, pages 319–333. Springer-Verlag, 2003.
53. T. Good and M. Benaissa. AES on FPGA from the Fastest to the Smallest. In J.R. Rao and B. Sunar, editors, Cryptographic Hardware and Embedded Systems — CHES 2005, number 3659 in Lecture Notes in Computer Science, pages 427–440. Springer-Verlag, 2005.
54. G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat. Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael Very Well Suited for Small Embedded Applications. In International Conference on Information Technology: Coding and Computing—ITCC 2004, pages 583–587. IEEE Computer Society, 2004.
55. P. Bulens, F.-X. Standaert, J.-J. Quisquater, P. Pellegrin, and G. Rouvroy. Implementation of the AES-128 on Virtex-5 FPGAs. In S. Vaudenay, editor, Progress in Cryptology — AFRICACRYPT 2008, pages 16–26, 2008.
56. Labbe A., Perez A. AES Implementation on FPGA: Time and Flexibility Tradeoff. Proceedings of FPL, pp. 836- 844, 2002.
57. Saggese G.P., Mazzeo A., Mazzocca N., Strollo A.G.M. An FPGA-Based Performance Analysis of the Unrolling, Tiling, and Pipelining of the AES Algorithm. Lecture Notes in Computer Science, volume 2778, pages 292–302, 2003.
58. Standaert F.-X., Rouvroy G., Quisquater J.-J., Legat J.-D. Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvements and Design Tradeoffs. Cryptographic Hardware and Embedded Systems - CHES 2003, pages 334–350, 2003.
59. Segredo A., Zabala E., Bello G. Diseno de un procesador criptografico Rijndael en FPGA. X Workshop IBERCHIP, pages 64–65, 2004.
60. Rodriguez-Henriquez F., Saqib N. A., Diaz-Perez A. D. 4.2 Gbit/s Single-Chip FPGA Implementation of AES Algorithm. IEE Electronic Letters, volume 39 (15), pages 1115–1116, July 2003.
61. Zambreno J., Nguyen D., Choudhary A. Exploring area/delay tradeoffs in an AES FPGA implementation. FPL 2004, Lecture Notes in Computer Science, volume 3203, pp.575-585, 2004.
62. Adib S. E., Raissouni N. AES Encryption Algorithm Hardware Implementation Architecture: Resource and Execution Time Optimization. International Journal of Information & Network Security (IJINS) Vol.1, No.2, June 2012, pp. 110-118.
63. F.-X. Standaert, G. Piret, G. Rouvroy, and J.-J. Quisquater. FPGA Implementations of the ICEBERG Block Cipher. The VLSI Journal, 40(1):20–27, 2007.
64. X. Guo, Z. Chen, and P. Schaumont. Energy and Performance Evaluation of an FPGABased SoC Platform with AES and PRESENT Coprocessors. In Embedded Computer Systems: Architectures, Modeling, and Simulation, volume 5114 of Lecture Notes in Computer Science, pages 106–115. Springer-Verlag, 2008.
65. F. Macé, F.-X. Standaert, and J.-J. Quisquater. FPGA implementation(s) of a Scalable Encryption Algorithm. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 16(2):212–216, 2008.
67. Panasenko S., Smagin S. Lightweight Cryptography: Underlying Principles and Approaches. International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011, pp. 516-520.
68. S. Zhilyaev. Evaluating a new MAC for current and next generation RFID, M.S. Thesis, Graduate School of the University of Massachusetts, Amherst, 2010.