ЛЕГКОВЕСНАЯ КРИПТОГРАФИЯ
Часть 2

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

Статья является продолжением обзора малоресурсной (легковесной) криптографии, представленного в предыдущем номере журнала[1]. Показаны тенденции увеличения ограничений на используемые вычислительные ресурсы, ресурсы памяти, энергетическую мощность. Обоснована актуальность 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 в ассиметричных системах. Сделан вывод об обнадёживающих перспективах малоресурсной криптографии. Даны рекомендации по развитию направления малоресурсной криптографии в России.

***Ключевые слова**: криптография, малоресурсная криптография, легковесная криптография, блочные шифры, стойкость легковесной криптографии, эффективность легковесной криптографии, программные шифры, аппаратные шифры, интернет вещей.*

LIGHTWEIGHT CRYPTOGRAPHY
Part 2

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

Abstract. This article is a continuation of the review lightweight (low resource) cryptography, presented in the previous issue. The trend of increasing restrictions on the use computing resources, storage resources, energy capacity is shown. The urgency of algorithms lightweight cryptographic resistance which decreases slightly in contrast to the resources required is 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*

Начало в предыдущем номере.

Поточные шифры

Победители конкурса eSTREAM (2004-2008) алгоритмы поточного шифрования Grain v.1, MICKEY v.2, Trivium демонстрируют возможности низкоресурсной реализации, что входило в число критериев, по которым оценивались алгоритмы, представленные на конкурс по профилю 2 (алгоритмы, предназначенные для аппаратной реализации).

В международный стандарт ISO/IEC 29192-3 (Stream ciphers) включены два алгоритма:

- поточный шифр Enocoro (размер знака выходной гаммы – 8-бит, размер ключа – 80 или 128 бит);
- поточный шифр Trivium (размер знака выходной гаммы – 1-бит, размер ключа – 80 бит).

Попытаемся осуществить сравнительный анализ поточных шифров, как «общего пользования», так и специально разработанных «легковесных» алгоритмов.

1 Вопросы кибербезопасности. 2015. №1.

Таблица 4

Сравние результатов аппаратной реализации различных поточных шифров.

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

Алгоритм	Nk	Тактов на инициализацию	GE	CI/ byte
A2U2 (2011) [1]		56+5	284 [1] 226 [1]	
Enocoro-80 (2008) [2]			2,700	
Enocoro-128v2 (2010) [3]		4869.5	4,100	46.3
F-FCSR-H			4,800	
Grain-128		1137.5		31.2
Grain v.1 (2006) [4]	80	321	1,294[5] 2,599[5]	
MICKEY-128 v.2		56592.1	5,000	1231.4
Salsa20	128		3,842	
Snow 2.0		1086.0		5.0
Trivium (2005) [6]	80	1,333	749[7] 2,599[5] 3,488 5,504 1,294[5]	 8.0 0.125
WG-7 (2008) [8, 9]		10,084	1,097[10]	
AES-CTR		469.6		17.8

Программная реализация поточных шифров

Данные о программной реализации некоторых поточных шифров для 8-разрядных микроконтроллеров типа AVR отображены в таблице 5.

В то же время для процессоров общего назначения представленные алгоритмы демонстрируют весьма хорошие скорости работы. Так алгоритм Trivium на платформе x86 демонстрирует скорость порядка 4 тактов на байт, что почти в 5 раз выше скорости работы алгоритма AES, полученной для той же платформы.

Другой распространенной платформой для реализации легковесных блочных шифров являются ПЛИС (FPGA). Результаты реализаций ряда поточных алгоритмов на ПЛИС отображены в таблице 6. Для сравнения там же приводятся данные о реализации блочного шифра AES.

Таблица 5

Сравние результатов программной реализации на указанной выше платформе некоторых поточных шифров.

Обозначения: Nb – длина информационного блока (в битах); Nk – длина ключа (в битах).

Алгоритм	Nk	Nb	Шифро-вание (тактов на блок)	Расшиф-рование (тактов на блок)	Размер кода в байтах	SRAM в байтах
Salsa20	128	512?	18,400	NA	1,452	280
LEX	128	320?	5,963	NA	1,598	304

Таблица 6.

Алгоритм	Nb	Nk	FPGA	Число Slices	Max. Freq. (MHz)	Проп. способн. (Mb/s)	Эфф. Mb/s/ Slice
A5/1			Virtex-II	32		188.3	5.88
Grain-128	1	128	Virtex-II	48	181	181	3.77
			Spartan-II	48		105	2.19
			xc3s50-5	50		196	3.92
Grain v1	1	80	xc3s50-5	44		196	4.45
MICKEY-128 2.0	1	128	xc3s50-5	176		223	1.27
			Virtex-II	190	200	200	1.05
			Virtex-E	167	170	170	1.02
MICKEY v2	1	80	xc3s50-5	115		233	2.03
Trivium	1	80	Virtex-II	41	207	207	5.05
			Spartan-II	40		102	2.55
			xc3s50-5	50		240	4.80
Trivium (x64)	64	80	xc3s400-5	344		13,504	39.26
Phelix			Virtex-II	1213	63	1000	0.82
			Spartan-II	1077		750	0.70
RC4			Virtex-II	140		121	0.86
AES			Spartan-II	124		2.2	0.02
			Virtex-II	146	123	358	2.45
			Virtex-II	1780	78	1000	0.56

Выводы. Поточные шифры

Сравнение реализаций блочных и поточных шифров показало, что вопреки распространенному убеждению, потоковые шифры не дают существенного преимущества в условиях ограниченных аппаратных или программных ресурсов приложения.

Зачастую поточные шифры весьма неудобны для легковесной реализации предназначенной для обработки очень небольших массивов информации, т.к. имеют, как правило, сравнительно большое время инициализации. Кроме того большинство поточных шифров требуют большого количества памяти для записи своего внутреннего состояния. Следовательно, эти шифры не могут обеспечить эффективное шифрование небольших объемов данных, что наиболее характерно для встраиваемых систем.

Становится очевидной задача разработки легковесных шифров, ориентированных на реализацию на 4-, 8-, и 16-битных микроконтроллерах и ПЛИС.

Хэш-функции

Для криптографической хэш-функции имеется международный стандарт – The Secure Hash Standard (SHA-256)[2]. Наиболее скоростная реализация алгоритма работает со скоростью 7,420 Mb/s (данные 2004 г.). Наиболее легковесная – имеет размер микросхемы 8,588 GE (представлено на конференции CHES 2010). Алгоритмы, представленные на конкурсе NIST SHA-3, также не удовлетворяют требованиям к LWC. Международных стандартов на легковесную хэш-функцию нет.

Ниже приводятся результаты аппаратной реализации для наиболее «легковесных» хэш-функций.

2 ISO 10118-3. IT. Security techniques. Hash functions. Part 3. Dedicated hash functions.

Таблица 7

Сравние результатов аппаратной реализации различных хэш-функций.

Обозначения: Nb – длина информационного блока (в битах); GE – условные логические элементы (gate equivalent); Cl/bl – число тактов работы алгоритма на блок (байт?) обработанной информации – мера скорости работы алгоритма, пропускная способность.

Алгоритм	Nb	GE Публикация реализации	Cl/bl	Примечание
AES128-based [12]	128	> 4,400	> 1,032	
DM-AES				
H-AES		>9,800	>1,032	
Luffa-224/256		10,157		
		25,833		
Luffa-384		13,168		
		34,401		
Luffa-512		16,720		
		40,715		
MAME	256	8,100[13]	96	
MD4	128	7,350	456	[14,15]
MD5	128	8,400	612	[14,15]
C-PRESENT (2008)	192	8,048[15]	108	
		4,600[15]	3,338	
DM-PRESENT-80	64	1,600	547	[15]
[12] (2008)		2,213	33	[15]
DM-PRESENT-128 [12]	128	1,886	559	[15]
(2008)	64	2,530	33	[15]
H-PRESENT-128	128	2,330	559	[15]
(2008)		4,253	32	[15]
QUARK [AHMN'10] (2010)		1,379		
		2,296		
D-QUARK	160	1,702	547	
[AHMN'10]				
T-QUARK	224	2,296	33	
[AHMN'10]				
U-QUARK	128	1,379	33	
[AHMN'10]				
SHA-1	160	6,812		
		8,120	1,274	[14,15]
		54,133		
SHA-224/256[3]	256	10,868	1,128	[15] [14]
		11,484		
		22,025		
SHA-384/512		43,330		
		23,146		
SQUASH	32	6,303		
[20]	64	6,328	104,114	[16]
	128	2,646 [17]		
WH-16	512	460 [18]		
Cubehash8/1	512	7630 [19]		

3 FIPS 180-3, Secure Hash Standard, U.S. Department of Commerce, 2008.

Выводы. Хэш-функции

Хотя инициированный NIST конкурс на SHA-3 завершился 2 октября 2012 года, когда NIST объявил, что Keccak будет новым SHA-3 хеш-алгоритмом для легковесной криптографии, но полезных результатов не принес[4]. Задачей конкурса был выбор хэш-функции общего вида и финалисты конкурса не удовлетворяют требованиям к LWC. В частности, размеры микросхем, реализующих рассматриваемые алгоритмы хэширования весьма далеки от требуемой границы в 2,000 GE. Финалисты конкурса демонстрируют показатели в пределах 11,000-140,000 GE.

В то же время хорошие показатели имеют хэш-функции, базирующиеся на легковесных блочных шифрах. Хорошие результаты показывают также хэш-функции семейства QUARK.

Криптография с открытым ключом

В отличии от алгоритмов блочного шифрования, в настоящее время нет большого выбора асимметричных криптоалгоритмов, пригодных для малоресурсной реализации.

В стандарт ISO/IEC 29192-4 (Mechanisms using asymmetric techniques) включены следующие схемы легковесной асимметричной криптографии:

- cryptoGPS (по именам авторов – Girault, Poupard, Stern) – однонаправленный (one-way) механизм аутентификации, основанный на дискретном логарифме над эллиптической кривой. Алгоритм обеспечивает криптостойкость эквивалентную стойкости 80-битной симметричной криптосистемы. Известна его аппаратная реализация, требующая 724 тактов работы для схемы размером 2876 GE[5].
- ALIKE (Authenticated Lightweight Key Exchange) – однонаправленный механизм ключевого обмена, базирующийся на шифровании. Размер кода для процессора 8051 core – 1.6 Кбайт, время срабатывания на частоте 31 MHz – 80 мсек..
- Механизм выработки цифровой подписи IBS. Известны характеристики его программной реализации:

Алгоритм	Размер кода [Byte]	RAM [Byte]	Время работы [ms]	Энергия [μJ]
IBS Выработка подписи	54,308	858	896	12,370
IBS Проверка подписи	55,374	922	5,610	77,400

4 www.nist.gov/itl/csd/sha-100212.cfm

5 ISO 29192-4. IT. Security techniques. Lightweight cryptography. Part 4: Mechanisms using asymmetric techniques

Аппаратная реализация асимметричных криптоалгоритмов

Особенно актуальной становится аппаратная реализация малогабаритного процессора для работы с асимметричными криптоалгоритмами на стандартных эллиптических кривых ECC. Разработаны аппаратные реализации малогабаритных процессоров для работы таких алгоритмов. Для уменьшения площади процессора, жертвуя гибкостью, выбирается работа с конкретной эллиптической кривой над стандартным двоичным полем, что обусловлено ограниченными возможностями встроенного устройства. Устройство работает со стандартными двоичными полями, которые обеспечивают как краткосрочную безопасность (113 бит), так и среднесрочную безопасность (193 бит).

ECC processor performance for scalar multiplication.		
Field	Total area (GE)	Source
$GF(2^{113})$	10,113	[B'06]
$GF(2^{131})$	11,970	[B'06]
$GF(2^{163})$	15,094	[B'06]
$GF(2^{193})$	17,723	[B'06]
$GF(2^{67})^2$	12,944	[B'06]
$GF(2^{131})$	14,735	[B'06]
$GF(p_{100})$	18,720	[G'05]
$GF(2^{191})$	23,000	[W'05]
$GF(p_{166})$	30,333	[OSS'04]

Основной причиной выбора бинарных полей, а не простых полей является отсутствие переноса младших разрядов в старшие при выполнении арифметических операций в поле, что хорошо подходит для аппаратной реализации. Второй причиной является простота структуры для операции возведения в квадрат, которая является основной операцией используемой в алгоритмах для нашего процессора.

Для полей $GF(2^{163})$ наиболее компактные реализации эллиптической криптографии используют порядка 15,400 GE (2009 г.). Наиболее быстрая реализация эллиптической криптографии над тем же полем дает 83,300 операций умножения в секунду (12 μs на операцию) и требует порядка 154,000 GE.

Наиболее быстрой микросхемой из упоминаемых в открытой литературе, является микросхема, работающая с 1024-битным модулем и выполняющая 6,350 модулярных возведений в степень в секунду (157.4 μs на одно возведение). Однако,

при этом, микросхема содержит 923,000 GE и, работая на частоте 140 MHz, потребляет 1.619 W энергии. Для 3,248-битных модулей результаты значительно хуже.

Для сравнения алгоритм RSA для обеспечения секретности, эквивалентной 128-битному ключу в симметричных криптосистемах, должен работать с 3,248-битными модулями. Возведение в степень в этом случае требует порядка 5,000 операций модулярного умножения. Доступные коммерческие решения, работающие на частоте более 300 MHz с микросхемами размера около 40,000 GE и порядка 12,000 бит оперативной памяти (RAM), выполняют в секунду не более 50 возведений в степень по 1024-битному модулю (Hellion ModExp Core, STD256).

В то же время наиболее компактные реализации алгоритма RSA используют порядка 8,000 GE, но могут выполнять лишь 5 модулярных возведений в степень в секунду (Helion ModExp Core, TINY32).

Программная реализация

Использование сочетания программного и аппаратного обеспечения может существенно увеличить производительность криптографических алгоритмов с открытым ключом при минимальной площади микросхемы. В работе [EKPPU'07] описана реализация криптографии с открытым ключом, которая может быть использована в низкоуровневых 8-битных процессорах для обеспечения надлежащей безопасности приложения низкого уровня. Реализация была осуществлена на 8-разрядном микроконтроллере ATmega128L, который включает 128 Кбайт внутрисистемной перепрограммируемой флэш-памяти, 4 Кбайт программируемой ПЗУ (EEPROM), и 4 Кбайт внутренней SRAM. Микроконтроллер может работать с дополнительными 64 Кбайт внешней памяти.

Эффективность реализации арифметики в конечных полях, в особенности операции умножения в конечном поле, определяют общую эффективность криптографических алгоритмов на эллиптических кривых. Поскольку асимметричные криптосистемы в среднем на три порядка медленнее, чем симметричные криптосистемы, основной упор делается на скорость, а не на размер кода.

Elliptic Curve Cryptography (ECC) над конечным полем порядка 2^{256} обеспечивает секретность, эквивалентную 128-битному ключу в симметричных криптосистемах. Основной операцией в эллиптической криптографии является модулярное произведение. Наиболее быстрой реализацией этой операции в простом 256-битном поле является реализация на микросхеме Xilinx Virtex-4 SX55 FPGA board, дающей 25.000 операций умножения в секунду (40 μs на операцию) (представлено на конференции CHES 2008). Однако такая схема использует 24,574 логических модулей (logic slices) and 512 цифровых сигнальных процессоров (DSP units).

Выводы. Криптография с открытым ключом

В настоящее время наиболее «легковесными» являются алгоритмы асимметричной криптографии, работающие с эллиптическими кривыми (ECC – Elliptic Curves Cryptography). Типичными значениями параметров различных процессоров, предназначенных для вычислений с эллиптическими кривыми, являются величины в пределах 10-40 μW и 10,000-20,000 GE. Близкими к ним являются и значения параметров у процессоров, предназначенных для вычислений с гиперэллиптическими кривыми (HECC – HyperElliptic Curves Cryptography).

Таким образом, по сравнению с симметричной криптографией, асимметричная криптография с безопасной длиной ключа, реализованная на аппаратном уровне, по-прежнему требует значительно больших ресурсов (по крайней мере, 10000 дополнительных логических элементов), при этом реализуется вполне приемлемая скорость работы. Для криптографии с открытым ключом верхняя граница на размер микросхемы устанавливается в 15,000 GE.

Что касается программной реализации, то тщательная оптимизация алгоритмов позволяет микроконтроллерам выполнять операции ассиметричной криптографии менее чем за 1 секунду, что вполне достаточно для большинства приложений. При этом программно-аппаратная реализация, по-видимому, создаёт наилучший баланс между размером и скоростью для многих распространенных вычислительных приложений.

Основные подходы и принципы к разработке и реализации легковесных криптоалгоритмов

Как отметил президент IACR Барт Пренель в своем докладе «Stream Ciphers and Lightweight Cryptography» на международном семинаре в Пекине (июнь 2011 г.) нет одного оптимального решения, подходящего для использования в различных приложениях и встроенных системах – радиочастотных метках, бесконтактных смарт-картах, сенсорах, сопроцессорах для 8-битных процессоров.

Одним из двух основных направлений развития низкоресурсной криптографии является эффективная реализация известных алгоритмов шифрования (возможно, с их небольшой модификацией). В последнем случае допускается частичное снижение безопасности до уровня, вполне достаточного для большинства приложений.

При реализации криптографических алгоритмов разработчики устройств с ограниченными техническими ресурсами должны принимать во внимание свойства целевой платформы и условия, в которых оборудование будет функционировать.

Так, очень важное значение имеет выбор архитектуры. Например, при аппаратной реализации алгоритма последовательные вычисления уменьшают размер схемы и потребляемую микросхемой мощность (но, естественно, увеличивают время работы). С другой стороны малое время обработки информации ведет к уменьшению потребляемой энергии. Это важно, особенно для устройств с батарейным питанием, потому что время обработки информации непосредственно взаимосвязано с энергопотреблением. В современные микроконтроллеры можно ввести различные режимы отключения питания и энергосбережения после окончаний вычислений. Таким образом, быстрое выполнение алгоритма может снизить потребление энергии и продлить время работы батарейного устройства. Поиск сбалансированного решения является непростой задачей. Необходимо, например, исследование зависимости скорости работы криптоалгоритма от размера микросхемы (area for speed – serial/parallel).

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.

Другим направлением в развитии легкой криптографии помимо эффективной реализации или небольшой модификации известных алгоритмов шифрования, является разработка новых шифров, ориентированных на оптимальную реализацию на аппаратном уровне.

Как указывалось выше, задачей проектирования средств легковесной криптографии является нахождение компромисса между имеющимися ограничениями на используемые ресурсы и криптографической стойкостью разрабатываемого алгоритма (с учетом условий, в которых будет функционировать оборудование, для которого этот алгоритм разрабатывается). Даже выбор основных параметров алгоритма (размер информационного блока, размер ключа, размер внутреннего состояния у алгоритмов поточного шифрования, размер конечного поля в случае асимметричных криптоалгоритмов) требуется сделать в пределах обозначенных ограничений. Так, ограничения на используемые ресурсы делают привлекательным разработку криптоалгоритмов с малыми размерами информационного блока и ключа. Однако в этом случае криптографический алгоритм подвержен различным атакам (например, атаки, связанные с парадоксом о днях рождения). Поэтому, например, блочный шифр с малыми размерами информационного блока категорически не рекомендуется использовать в режиме ECB.

Очевидно, что предпочтение в использовании получают преобразования, требующие меньшего размера памяти вычислительного устройства (или меньшего числа логических элементов для их реализации). Так выбор S-блоков размера 4x4 предпочтительнее S-блоков размера 8x8, так, как 8-битные S-блоки требуют для своей реализации в среднем 1,000 GE (и уж никак не меньше 120 GE) в то время как 4-битные могут быть реализованы со сложностью 21 – 39 GE, что в среднем ведет к уменьшению размеров микросхемы в десятки раз [15]. Однако 4-битные S-блоки должны быть выбраны очень тщательно, так как они криптографически слабее, чем 8-битные. Тем не менее, за счет тщательного отбора, возможно достичь соответствующего уровня безопасности.

Алгоритм выработки цикловых ключей должен порождать их in-place, т.е. цикловые ключи не должны требовать предвычислений.

В последнее время наметилась тенденция использовать в разрабатываемом легковесном алгоритме широко распространенные и хорошо исследованные элементарные преобразования (арифметические и логические операции и т.д.). Используемые операции должны допускать различные способы реализации в зависимости от имеющихся в наличии ресурсов. Примером могут послужить так называемые ARX-шифры, построенные из преобразований трех видов: сложения по mod 2^n (Addition), циклического сдвига (Rotation) и операции побитового сложения (Xor). В настоящее время известен целый ряд шифров, относящихся к этой категории, однако требуется дальнейшее исследование криптографических свойств таких преобразований и требований, предъявляемых к их композиции. Важным направлением исследований становится поиск дру-

гих легко реализуемых преобразований, подходящих для использования в криптографических алгоритмах.

Тенденции развития легковесной криптграфии

Прежде всего, во всех обзорах по низкоресурсной криптографии отмечается, что в настоящее время нет общей теории разработки LWC-алгоритмов (возможно и не будет). Целый ряд авторов предлагает выделить разработку сверхлегких криптографических алгоритмов (ultra-lightweight algorithms) в отдельное направление криптографии. При этом усиливается расхождение между программно- и аппаратно-ориентированными легковесными криптоалгоритмами. Фундаментальное различие в требованиях, предъявляемыми ресурсными ограничениями к программно- и аппаратно-ориентированным легковесным криптоалгоритмам было продемонстрировано в работе [15]. В частности, там было показано, что блочный шифр PRESENT чрезвычайно удобен для легковесной аппаратной реализации, но требует значительных ресурсов при программной реализации.

Поскольку основной задачей низкоресурсной криптографии является минимизация затрачиваемых ресурсов, важным направлением является многофункциональность – возможность с помощью одной микросхемы осуществлять шифрование, реализацию выработки имитовставки (MAC), генерацию псевдослучайной последовательности (PRNG) и т.д. При этом разрабатываемые алгоритмы должны обеспечить эффективную обработку небольших объемов данных, что наиболее характерно для встраиваемых систем.

Наконец, важной проблемой для LWC-алгоритмов являются атаки по побочным каналам (side-channel attacks). Эти вопросы пока мало исследованы, но учитывая условия, в которых будут работать алгоритмы, велика вероятность успеха таких атак. Так что перспективным направлением является разработка для низкоресурсных криптоалгоритмов контрмер против атак по побочным каналам.

Литература / References

1. David M., Ranasinghe D. C., Larsen T. A2U2 - A Stream Cipher for Printed Electronics RFID Tags. IEEE International Conference on RFID 2011. pp. 176-183.
2. Watanabe D., Ideguchi K., Kitahara J., Muto K., Furuichi H., Kaneko T. Enocoro-80: A Hardware Oriented Stream Cipher. 2008 Third International Conference on Availability, Reliability and Security, ARES 2008, Proceedings, pp. 1294–1300, 2008.
3. 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).
4. Hell M., Johansson T., Meier W. Grain: A Stream Cipher for Constrained Environments. International Journal of Wireless and Mobile Computing, vol. 2, no. 1, pp. 86-93, 2007.
5. Good T., Benaissa M. Hardware Results for selected Stream Cipher Candidates. State of the Art of Stream Ciphers 2007 (SASC 2007), Workshop Record, February 2007.
6. Canniere C. D., Preneel B. TRIVIUM - a stream cipher construction inspired by block cipher design principles. eStream, ECRYPT Stream Cipher Project, Report 2005/030, 2005. http://www.ecrypt.eu.org/stream/trivium.html
7. Mentens N., Genoe J., Preneel B., Verbauwhede I. A Low-cost Implementation of Trivium. In ECRYPT Workshop, The State of the Art of Stream Ciphers — SASC 2008, pages 197–204, 2008.
8. Luo Y., Chai Q., Gong G., Lai X. WG-7: A Lightweight Stream Cipher with Good Cryptographic Properties. IEEE Global Communications Conference GLOBECOM'2010, pp. 1-6, 2010.
9. Nawaz Y., Gong G. WG: A family of stream ciphers with designed randomness properties. Information Sciences, v. 178, no. 7, pp. 1903-1916, 2008.
10. Lam C.H., Aagaard M., Gong G. Hardware Implementations of multi-output Welch-Gong ciphers. Preprint, University of Waterloo, 2009.
11. Bogdanov A., Knezevic M., Leander G., Toz D., Varici K., Verbauwhede I. Spongent: A lightweight hash function. CHES 2011, Lecture Notes in Computer Science v. 6917, pp. 312–325, 2011.
12. Bogdanov A., Leander G., Paar C., Poschmann A., Robshaw M., Seurin Y. Hash Functions and RFID Tags: Mind the Gap -10. International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2008, Washington, USA. August 10 - 13, 2008. Pages 283-299.
13. H. Yoshida, D.Watanabe, K. Okeya, J. Kitahara, J.Wu, O. Kucuk, and B. Preneel. MAME: A Compression Function With Reduced Hardware Requirements. In P. Paillier and I. Verbauwhede, editors, Cryptographic Hardware and Embedded Systems — CHES 2007, volume 4727 of Lecture Notes in Computer Science, pages 148–165. Springer-Verlag, 2007.

14. M. Feldhofer and C. Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. In First International Workshop on Information Security—IS 2006, volume 4277 of Lecture Notes in Computer Science, pages 372–381. Springer-Verlag, 2006.
15. Poschmann A. Lightweight Cryptography - Cryptographic Engineering for a Pervasive World. Ph.D. Thesis, Ruhr University Bochum, 2009.
16. F. Gosset, F.-X. Standaert, and J.-J. Quisquater. FPGA Implementation of SQUASH. In Twenty-ninth Symposium on Information Theory in the Benelux, 2008.
17. S. Zhilyaev. Evaluating a new MAC for current and next generation RFID, M.S. thesis, Graduate School of the University of Massachusetts, Amherst, 2010.
18. K. Yüksel. Universal hashing for ultra-low-power cryptographic hardware applications. M.S. thesis, Worcester Polytechnic Institute, Citeseer, 2004.
19. Preneel B. Perspectives on Lightweight Cryptography," Inscript 2010, Shanghai, China, 20-24 October 2010.
20. Shamir, A. SQUASH - a new MAC with provable security properties for highly constrained devices such as RFID tags. In FSE 2008, Lecture Notes in Computer Science v. 5086, pp. 144–157, 2008.
21. 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.
22. 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.
23. Zhukov A.E. Lightweight Cryptography, Cybersecurity Issues (In Russia), 2015, No 1(9). Part 1.