Journal of Network and Computer Applications 49 (2015) 15–50

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments

Jia Hao Kong [a,*], Li-Minn Ang [b], Kah Phooi Seng [b]

[a] Department of Electrical and Electronic Engineering, University of Nottingham, Malaysia Campus, 43500 Semenyih, Malaysia
[b] School of Engineering, Edith Cowan University, Joondalup, WA 6027, Australia

ARTICLE INFO

Article history:
Received 16 February 2014
Received in revised form
10 August 2014
Accepted 16 September 2014
Available online 14 October 2014

Keywords:
Symmetric cipher
Lightweight block cipher
WSN
RFID
WISP
IOT

ABSTRACT

Modern cryptographic algorithms play an irreplaceable role in data communication systems for various areas of applications. These algorithms are the backbone of data protection and secrecy for highly sensitive and classified data. The selection of a suitable crypto-algorithm will dynamically affect the lifespan and performance of a device in terms of battery-life, hardware memory, computation latency and communication bandwidth. In the current developments of the resource constrained environments, the trend is shifting towards lightweight algorithmic hardware designs. To select a suitable cryptographic algorithm for an application or an environment, the understandings of both the algorithmic requirements in terms of hardware and the specifications of the development platform intended has to be established. However, there are numerous ciphers in the literature that has various functionality, specifications and strength. Moreover, there are numerous literatures that cover the trend and specifications of security solutions in hardware constrained environment, employing known cryptographic algorithms. In this paper, we present a comprehensive survey of modern symmetric cryptographic solutions used in resource constrained environment (RCE), including literatures from the area of wireless sensor network (WSN), radio frequency identification (RFID), wireless identification and sensing platform (WISP) and other resource constrained platforms. This paper aims to provide a survey of the ciphers that were used in the past, and what are the ciphers that are currently active, and their respective specifications and applications in the area of modern world RCEs. On top of that, descriptive summaries of (a total of 100 symmetric ciphers) modern block ciphers (38), involution ciphers (6), lightweight block ciphers (28) and stream ciphers (28) are included and discussed, and an overview of the current contributions of various literatures, comparison and analysis of modern ciphers from the hardware and software perspective are also discussed.

Contents

[*] Corresponding author.
E-mail addresses: keyx9kjh@nottingham.edu.my (J.H. Kong), li-minn.ang@ecu.edu.au (L.-M. Ang), kp.seng@ecu.edu.my (K.P. Seng).

http://dx.doi.org/10.1016/j.jnca.2014.09.006
1084-8045/© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The term “resource constrained environment” (referred to as the RCE in this paper) is used in describing a hardware development platform that has very little amount of design space (e.g. battery-life, hardware memory, computation latency, communication bandwidth, etc.). Hence, designers are constrained by the resources available and minimalist approaches are often taken. Example of RCE platforms include the areas of wireless sensor network (WSN), radio frequency identification (RFID), wireless identification and sensing platform (WISP), Internet of Things (IOT), etc. The objective of this paper is to highlight discussions on ciphers that were used in the past, and what are the ciphers that we currently have, and where and how they are applied in our modern world hardware constrained environments and specifically, the resource constrained environments such as the WSN, RFID, WISP and IOT. A literature search showed that most if not all previous security related survey papers were targeted towards a specific resource-constrained application or environment such as:

1. Law et al. (2006) presented a survey on WSN ciphers.
2. Xiangqian et al. (2009) presented a survey on WSN security issues.
3. Yong et al. (2006) presented a survey on WSN security.
4. Roman et al. (2007) presented a survey on cryptographic primitives for WSN.
5. Juels (2006) presented a survey on RFID security issues.
6. Aragones-Vilella et al. (2007) presented a survey on RFID security techniques.
7. Langheinrich (2009) presented a survey on RFID privacy approaches.
8. Kai and Lina (2013) presented a survey on IOT security issues.

However, a survey combining different RCEs would be useful for the following reasons:

1. Most surveys mentioned the keyword “resource constrained environment” but do not discuss other environments such as the RFID or the WISP or the mobile platform all together. A combined RCE survey would give inclusive coverage to all known RCEs.
2. Most of these surveys relate back to the WSN topic or the RFID topic without consideration of other RCE topics. A combined RCE survey can provide a bigger picture of the RCE research scene, without isolation.
3. The current surveys are limited to a selected number of cryptographic algorithms, without consideration that some of the cryptographic solutions are able to be used in other RCEs (cross-platform,

due to the amount of resource they use are suitable for multiple environments).

4. Most surveys do not address the relationship between the cipher cost and the intended application's resource requirements. Most surveys are done by investigating how much the cryptographic cost is in terms of certain hardware platform but does not mention the "real-world" hardware used. With a combined survey, the readers would be able to see what the real resource limits are and what are the up-to-date implementation results in terms of hardware in contrast.
5. The combined RCE survey would provide readers the focal point to look for any RCE security discussion or topics. This serves as a main "hub" for readers to find security, cryptography, hardware implementation and RCE related discussions and papers.
6. In a more specific topic, the WSN survey topics are mainly focused to generalized sensor nodes. Special networks such as the visual sensor or multimedia sensor networks were not discussed all together and these visual nodes have different specifications and applications that needed addressing.
7. In a more specific topic, the RFID surveys found were very similar to WSN surveys. They are both presented in a way that selected cryptographic costs are discussed, without the inclusion of a large pool of cryptographic examples.
8. This combined survey paper has surveyed and reviewed a total (selected) 100 symmetric ciphers known from the literature and this would help readers to understand how much the "known" cryptographic costs are by referring to known implementation papers. This serves as a reference to researchers and designers who would want to understand approximately how much cost would incur when using a particular cryptographic solution.

Modern cryptographic algorithms play an irreplaceable role in data communication systems for various areas of applications. These algorithms are the backbone of data protection and secrecy for highly sensitive and classified data. To date, numerous cipher developments and new efficient ciphers are created to stand up to today's security challenges. In this paper, historic classical ciphers such as the Ceaser cipher or the substitution cipher or the any other ancient substitution ciphers are not discussed as they are not within the scope of topic. These ciphers are considered ancient and the fore-fathers to the historic developments of the current ciphers and will remain as education materials. From the current trend, ciphers proposed after the year 2000 (post millennium) were all inclined towards the developments of lightweight systems. Lightweight cryptography's existence is not to replace classical cryptography but it is tailored towards stringent requirements of constrained devices. The lightweight cryptographic algorithms are more popular ever since. This is due to the fact that the lightweight ciphers are designed to be more "resource friendly".

Today, many cryptographic algorithms are in the public. Some ciphers are considered obsolete but some are still active even though numerous successful cryptanalysis are disclosed. There are no ciphers with identical specifications or structural designs. Only similar blocks and resembling structures were seen and were mostly adopted from their predecessors. When these algorithms are translated into hardware, the hardware footprint is measured according to chip-area and memory occupancy. The problem lies at the search for a suitable cipher algorithm that performs and fits comfortably into the hardware constrained environment.

From the differences between hardware platform and application needs, it is undeniable that the selection of a suitable crypto-algorithm will dynamically affect the lifespan and performance of a device in terms of battery-life, hardware memory, computation latency and communication bandwidth. When the hardware resources are scarce, careful utilization of resources is essential. This shows that each module has a limited share of resources to use and to perform their respective functions.

In the current developments of the RCEs, the trend is moving forward towards lightweight algorithmic hardware designs. The reason behind this trend is due to the limitation of available resources pushing the boundaries to find critical measures to fully-utilize that small amount of resources. To select a suitable cryptographic algorithm fitted or optimized to an application or an environment, the understandings of both the algorithmic requirements in terms of hardware and the specifications of the development platform intended are required. With understanding of both different sides, designers and researchers are able to pick a suitable solution for any classes of hardware, with considerations of the security requirements.

When designing a system, the balance between security, cost and performance has to be accounted. A system could not have all design qualities weighted heavily without compromising the cost (in terms of resources consumed). In Gong (2010), the author had mentioned a very important relationship between the security, cost and performance of a hardware system. In terms of security, the cryptographic strength is defined by the key length. The longest key length that we have identified is 320-bits and the lowest is 54-bits. Besides the key-length, ciphers have round-based execution that gives the cipher additional confusion. The number of rounds for a specific cipher is determined by the original cipher designers, stating the optimum number of rounds for an optimum security in their early proposals. More rounds and longer key length translates to a safer system.

As for the performance of the system, it is mainly measured by the number of rounds that a cipher has to be executed. More rounds is equal to more computation overhead hence, system latency would be affected. On the other hand, a processing platform dynamically affects the speed of the system via structural support. A parallel processing mechanism boosts performance and lowers latency, compared to serial architecture, but the hardware cost is proportional. Thus, less cryptographic rounds and a parallel architecture increase the system's response and performance.

And lastly, cost of the system can be directly related to the choices of algorithms and the performance selection. Stronger and faster ciphers would require most costs. Powerful and fast processing architectures contributes to additional costs. Therefore in Fig. 1, the authors present an illustration of the relationships between the three qualities in RCE hardware design.

This paper presents a comprehensive survey of modern symmetric cryptographic solutions used in RCEs, covering literatures from the area of wireless sensor network (WSN), radio frequency identification (RFID), wireless identification and sensing platform (WISP) and other resource constrained platforms. This collective survey of the current literature will aid the researchers in the area of cryptographic implementation for RCEs and also a contribution in providing insights for researchers to better understand the RCE requirements and the current trend of cryptographic approaches in the area of symmetric block ciphers. On top of that, a comprehensive review of modern block ciphers (released from pre-2000 to post-2000), involution ciphers and lightweight block ciphers are included and discussed and an overview of the current contributions of various literatures, comparison and analysis of modern symmetric ciphers is presented in this paper. This paper aims to provide a survey of the ciphers that were used in the past, and what are the ciphers that are currently active, and their respective specifications and applications in the area of modern world RCEs. On top of that, descriptive summaries of (a total of 100 symmetric ciphers) modern block ciphers (38), involution ciphers (6), lightweight block ciphers (28) and stream ciphers (28) are included and discussed, and an overview of the current contributions of various literatures, comparison and analysis of modern ciphers from the hardware and software perspective are also discussed.

2. Resource constrained environments and cryptographic approaches

In this section, various literatures that surveys and reviews the cryptographic solutions in various RCEs and the design factors that contributes to the formation of an efficient system is studied and reviewed.

2.1. Wireless Sensor Network (WSN) RCE

A wireless sensor network is made up of many tiny sensor nodes or mates which are programmed to communicate via wireless medium. Due to the limitation of their physical size, sensor motes usually have limited amount of on-board resources such as: energy, storage, computation power and communication bandwidth. WSN can be divided into three groups with variation of applications.

- *Wireless Sensor Network* (*WSN*) **–** The WSN is a generic term for a network of motes with embedded sensors. WSNs normally have tiny sensors to monitor environmental variables such as the temperature, humidity, noise, pressure, etc. It is also understood that the choices of security used in WSN environmental application is influenced by the amount of energy the security architecture consumes. Lightweight and energy efficient cryptographic algorithms are generally preferred.
- *Wireless Multimedia Sensor Network* (*WMSN*) **–** The WMSN highlights the use of low-cost cameras in health care monitoring systems, incorporating applications that transmit data such as high-resolution still images and multimedia video and audio streaming. This is a kind of network composed of embedded audio and visual collection modules that requires the balancing between the energy costs, application purposes and security strength considerations.
- *Wireless Visual Sensor Network* (*WVSN*) – This type of network features the use of visual sensors or tiny visual devices for real-time environmental and surveillance purposes. The crucial area of consideration for WVSN is low latency of communication and image processing modules. The real-time systems are extremely resource constrained, making designers to find extreme measures without compromising too much on security.

The security and architectural design issues are not that different even though the application differs in the WSN area. The obvious requirements fall into the lightweight requirement, energy efficiency and extensive mote life-span category. This motivates researchers to find a balanced solution for security and resource efficiency.

As for the overview of the WSN, the paper by Jaydip Sen (2009) has mentioned four known constraints in WSNs:

1. Energy constraints
2. Memory limitations
3. High latency in communication
4. Unattended operation of networks

The author also provided an in-depth survey on security issues in WSNs, covering security requirements to attacks and countermeasures. This paper also covers all the known defences against all sorts of attacks at different levels of the OSI layer.

The survey paper (Roman et al., 2007) focuses on the cryptographic primitives for hardware-constrained sensor network and covers a wider selection of cryptographic primitives, including the public key cryptography (PKC) and hash functions. And according to the authors, microcontrollers are used in the WSN because of their cost-effectiveness. Microcontrollers are grouped into weak, normal and heavy-duty for their computing capabilities, clock speed and RAM size. Fig. 2 illustrates an overview of the architecture within a

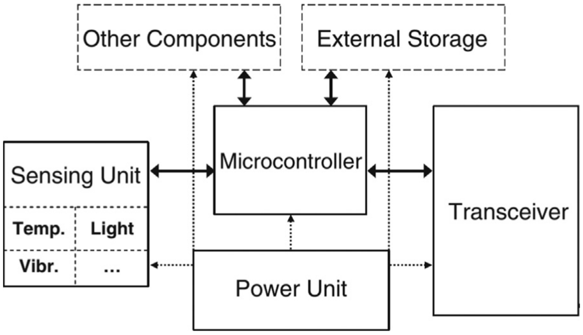

Fig. 2. An illustration of the architecture within a WSN node (Roman et al., 2007).

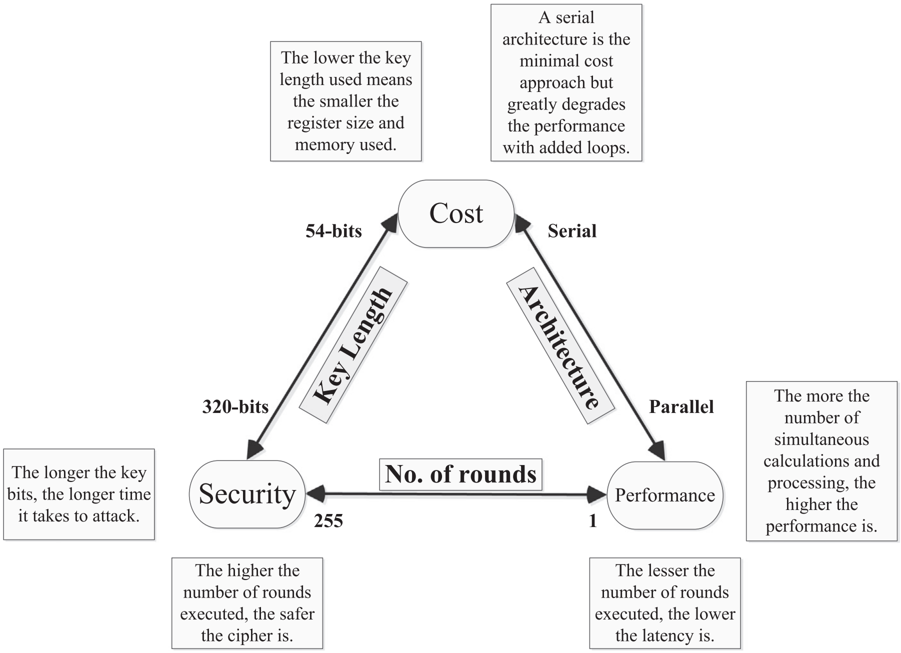
The lower the key length used means the smaller the register size and memory used.
A serial architecture is the minimal cost approach but greatly degrades the performance with added loops.
Cost
54-bits
Serial
Key Length
Architecture
320-bits
Parallel
The more the number of simultaneous calculations and processing, the higher the performance is.
The longer the key bits, the longer time it takes to attack.
Security
No. of rounds
Performance
255
1
The higher the number of rounds executed, the safer the cipher is.
The lesser the number of rounds executed, the lower the latency is.

Fig. 1. An illustration of the relationships between the three qualities in RCE hardware design.

WSN node, including connection of the microcontroller to other input/output components. It is also mentioned that suitability of some of the symmetric cryptographic primitives for some low-end microcontrollers are questioned. For instance, the AES and Twofish cipher are known to be optimized for 32-bit processors but some of the operations can be done using native 8-bit registers. Heavy-duty controllers such as the PXA271 or the ARM920T with a word size of 32-bit are a compatible with these ciphers. Ciphers like Skipjack fits perfectly into the MSP430 family because the operations used and the key schedule is in 16-bit words. The instruction memory and the RAM memory have to suffice for the storage of, program code, private key, intermediate values and other temporary data. This shows that choosing a cipher to match a microcontroller's resources is an important consideration.

Application-wise, the paper by Romer and Mattern (2004) had identified four classes of sensor nodes: brick, matchbox, grain and dust. This gives an overview of the physical size of the classes of sensor nodes and their respective resources onboard. Varying size and cost constraints of the sensor nodes directly result in varying limits on the energy and hardware available. The authors also provided a detailed classification of design spaces for various applications such as: ocean, avalanche, glacier, bird observation, herding, sniper localization, vital sign and various tracking applications. By identifying the application requirements and available design spaces, it can be understood how “constrained” the environment is and tailor a system which is lightweight enough, suitable for that application.

Johnson et al. (2009) reviewed the most recent specifications of sensor motes. This allows researchers to understand the current hardware platforms that they would be dealing when designing systems for WSN. Table 1 shows the hardware specifications of known motes.

The paper by Hempstead et al. (2008) provided a detailed analysis of hardware systems for sensor nodes, focusing on the architectural level of the processors used. The author mentioned a valid point that it would be difficult to judge the programmability, energy efficiency, and performance fairly without running the same benchmarking application on all these different systems. Among the general purpose architectures, the choice is not clear because of the different instruction set architecture (ISA) used, process technologies and even clock speeds. Hempstead summarized that despite all those differences, intelligent combination of circuit techniques, hardware architecture and application support can yield ultra-low power systems. Table 2 shows a summarized version of the table presented in Hempstead et al. (2008).

In the recent developments in the area of WVSN, the paper by Tavli et al. (2012) gave a general survey paper for the VSN area, mostly focusing on the hardware platforms. The relevant information extracted is the comparison of VSN hardware platforms. The hardware comparison is shown in Table 3.

Soro and Heinzelman (2009) explained in their survey that the unique characteristic of VSNs can be identified as below:

1. Resource requirements
2. Local processing
3. Real-time performance
4. Precise location and orientation information
5. Time synchronization
6. Data storage
7. Autonomous camera collaboration

Soro and Heinzelman's paper complements other VSN survey paper in understanding the area of VSN security. The important information is the hardware specification of the camera nodes are explained in detail. This information helps in the process to determine which cryptographic algorithm are able to be fitted into such environmental parameters. Table 3 shows the comparison of various visual sensor nodes and their respective specifications.

Table 1
The specifications of various sensor motes (Johnson et al., 2009).

Mote platform	μProcessor	Bus (bit)	Clock (MHz)	RAM (K)	Flash (K)	EEPROM (K)	Cost/node (USD)
TelosB (sensor)	TI MSP430F1611	16	4–8	10	48	1000	99
TelosB (w/o sensor)	TI MSP430F1611	16	4–8	10	48	1000	139
MicaZ	Atmel Atmega 128L	8	8	4	128	512	99
Mica2	Atmel Atmega 128L	8	8	4	128	512	99
SHIMMER	TI MSP430F1611	16	4–8	10	48+ microSD expansion	None	199
IRIS	Atmel Atmega 1281	8	8	8	640	4	115
Sun SPOT	Atmel AT91RM9200	32	180	512	4000	None	750
EZ-RF2480	TI MSP430F227432	16	16	1	1	None	99
EZ-RF2500	TI MSP430F227432	16	16	1	1	None	49

Table 2
The specifications of various controller architectures (Hempstead et al., 2008).

System	Architecture	Data path width	Memory (KB)
Atmel ATmega 128L	General purpose Off-the-shelf	8	132
TI MSP430	General purpose Off-the-shelf	16	10
SNAP/LE	General purpose Reduced instruction set computer	16	8
BitSNAP	General purpose Reduced instruction set computer (bit-serial data path)	16	8
Smart Dust	General purpose Reduced instruction set computer	8	3.125
Charm	Protocol processor	N/A	68
Michigan 1	General purpose	8	0.25
Michigan 2	General purpose	8	0.3125
Harvard	Event-driven accelerator	8	4

Table 3
The specifications of various controller architectures (Soro and Heinzelman, 2009; Tavli et al., 2012).

Camera node architecture	Processing unit	Memory	Image sensor
Cyclops	Atmel ATmega128L and CPLD-Xilinx XC2C256 CoolRunner	512 KB FLASH, 64 KB SRAM	ADCM-1700 Agilent Technology
MeshEye	Atmel ARM7TDMI Thumb (32-bit RISC) 55 MHz	64 KB SRAM, 256 KB FLASH External, MMC/SD FLASH	Two kilo-pixel imagers Agilent Technologies ADNS 3060 (grayscale) and one ADCM 2700 VGA (color)
Panopets	StrongARM (32-bit)	64 MB	Logitech 3000 USB Web-camera
Meerkats	XScale PXA255 (32-bit)	32 MB FLASH, 64 MB DRAM	Logitech 4000 USB Web-camera
Firefly mosaic	LPC2106 ARM7TDMI (32-bit) processor on IB, ATMEL Atmega1281 (8-bit) processor on NB	64 KB RAM, 128 KB FLASH on IB, 8 KB RAM, 128 KB, FLASH on NB	CMUCam3
Micrel eye	ATMEL FPSLIC SoC, with an AT40K MCU (8-bit)	36 KB onboard SRAM, 1 MB external SRAM	Omnivision OV7640
XYZ-ALOHA	RM7TDMI-based (32-bit) OKI ML67Q5002 on NB	32 KB RAM 256 KB FLASH onboard, 2 MB external RAM on NB	ALOHA imager
Citric	PXA270 (32-bit) on IB TI MSP430 (16-bit) on NB	64 MB SDRAM, 16 MB FLASH, on IB, 10 KB RAM, 1 MB FLASH on NB	Omnivision OV9655
Vision mote	ATMEL 9261 ARM 9 (32-bit)	128 MB FLASH, 64 MB DRAM	CMOS camera
SIMD-based architecture (single-instruction-multiple-data)	Philips IC3D Xetal (for low-level image processing), 8051 MCU (local host for high level image processing and control)	1792B RAM, 64 KB FLASH internal on 8051 MCU, dual port RAM 128 KB (shared memory by both processors)	VGA image sensor (one or two)
CMUCam3	ARM7TDMI (32-bit) 60 MHz	64 KB RAM, 128 KB FlASH on MCU, 1 MB AL4V8M440 FIFO Frame buffer flash (MMC)	Omnivision OV6620,

The work by Winkler and Rinner (2012) explained that the VSN is a middle group between distributed smart camera networks and WSNs. They also viewed a VSN node as a combination of processor, volatile and non-volatile memory, communication module and an image sensing unit, similar to WSNs. Other characteristics of VSN nodes are the constrained memory, limited computing, scarce battery power and the use of wireless multi-hop communication. The authors grouped the security problems into four areas: Data-centric, Node-centric, Network-centric and User-centric. However, this paper did not make much reference to the symmetric cryptography but did cite one of the most known security solutions for the WSN: Security Protocols for Sensor Network (SPINS).

In the area of multimedia applications, Akyildiz et al. (2007) had conducted a survey on wireless multimedia sensor networks (WMSN). This kind of multimedia network requires in-network processing. Techniques used to reduce data traffic would be preferred thus this suggests that the computational complexity of the security algorithm has to be minimal, allowing real-time applications, which points to the suggestion of using lightweight cryptographic solutions. A paper by Kundur et al. (2008) has presented findings in the area of Distributed Multimedia Sensor Network (DMSN). The authors provided an understanding of research opportunities in the field of DMSN. Focusing mostly on the distributed form of privacy paradigms, the authors stated the account for insider attacks, wireless communication limitations, multimedia security approaches, effective cryptographic and network paradigms have to be addressed.

Another survey in the area of WMSN by Guerrero-Zapata et al. (2010) provides the analysis of different security issues taken account in the design of WMSN platforms and protocols. Guerrero mentioned that asymmetric ciphers are worth revisiting as public key cryptography (PKC) simplifies the distribution and management of keys. However, this paper did not mention much about the use of symmetric ciphers and mostly focuses on the privacy schemes.

A recent surveys by Xiangqian et al. (2009) has provided a lot of insights for the security issues in WSNs. The authors covered topics like security goals in WSN, threats, attacks and security evaluations from the view point of OSI. The authors summarized that symmetric cryptography still remain superior to PKC in terms of execution speed and low energy cost. Despite having this advantage, key management for symmetric key cryptography is no easy task. Although recent proposals suggesting that PKC is still feasible by choosing appropriate parameters, symmetric key algorithm is still preferred. This paper also stated that issues within WSN can be categorized into seven areas, giving a complete picture of security problems in WSNs. The seven categories are:

1. Cryptography
2. Key management
3. Attack detection and prevention
4. Secure routing
5. Secure location security
6. Secure data fusion
7. Other security issues

The survey by Li and Gong (2008) addresses the state of the art in research on WSN security and future directions. This paper discusses the six security goals in WSNs:

1. Confidentiality
2. Integrity
3. Data origin
4. Entity authentication
5. Access control
6. Availability

The survey by Döser et al. (2011) discusses resource constrained security mechanisms that relates to WSN topics, stated design challenges and problems for ad hoc and sensor embedded systems.

Law et al. (2006) has provided a benchmarking of block ciphers for WSN applications. The author benchmarked ciphers like Skipjack, RC5, RC6, Rijndael, Twofish, MISTY1, KASUMI and Camellia. The authors had provided insights for security options in different scenarios such as low and high resource approach and high and low security approach for various applications. The authors concluded that Skipjack is a viable option for low-security applications and the MISTY1 is preferred when resources are low. The authors had made a specification comparison of sensors nodes, claiming that the rate of improvement is conservatively at a lower rate than Moore's law prediction. This further confirms the need for cheaper security designs.

Table 4
Hardware platforms used by Ganesan et al. (2003).

Platform	Word size (bits)	Clock frequency (MHz)	Architecture
Atmega 103	8	4	RISC
Atmega 128	8	16	RISC
M16C/10	16	16	CISC
StrongARM SA-1110	32	206	RISC
XScale PXA250	32	400	RISC
UltraSparc2	64/32	440	RISC

Similarly, Ganesan et al. (2003) had analyzed the ciphers: RC4, IDEA, RC5, MD5 and SHA. Note that MD5 (used in Pretty Good Privacy (PGP)) and SHA are hash functions. The authors used a few controllers in their experiment to search for the most efficient algorithm. They concluded that RC4 out-performs other ciphers in low-end processors (Atmega). Experimental measurements also showed that the cryptographic cost is uniform for each hardware and encryption class. Table 4 shows the hardware used.

In the area of lightweight cryptography, Jinwala et al. (2009) stated that the confidence in a security protocol is largely derived from the cipher. Hence, to ensure that ciphers operate comfortably without compromise, parameters such as the block-size, key-size and the number of rounds have to be investigated. The authors have selected TEA, XTEA, XXTEA ciphers and compared them with Skipjack on the MICA2 platform. The authors also concluded that XXTEA cipher is a suitable cipher for the WSN environment and the Skipjack cipher to be faster and energy efficient for the encryption and decryption operations even though the key expansion consumes higher energy. Fig. 3 shows their memory and CPU cycle results.

A paper by Xueying et al. (2010) covered a few of the symmetric key cryptographic algorithms (block and stream ciphers). The ciphers mentioned are: AES, Skipjack, Puffin, Byte-wise SPN, RC4, Sosemanuk and Salsa. The paper discussed a few key areas which affect the choices of ciphers:

- *The size of the encryption operands*, which contributes to the transmission cost in the sensor node.
- *The security considerations*, which is understood as the type of security used for any specific data or environmental application.
- *Modes of operation*, which is a trade-off balancing between the energy cost, application purpose and security considerations.
- *Key setup/expansion*, which is also a crucial area of consideration as this may contribute to high latency of the real-time system.
- *Key stream setup*, which is the area for stream ciphers. In WSN, the energy cost for key stream setup is accounted when ciphers are running in modes other than the ECB mode.

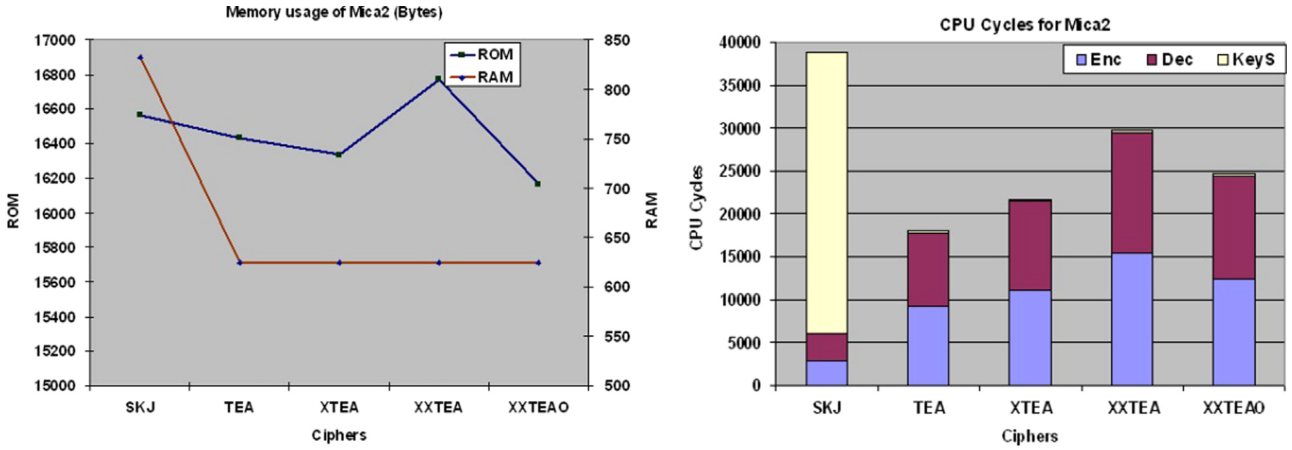

Fig. 3. (Left) Memory usage and (Right) CPU cycles used by various ciphers on the MICA2 platform (Jinwala et al., 2009).

- *The energy efficiency*, which is the key area when weighing the options of symmetric ciphers.

Table 5 shows a summary for the compilation of various surveys and papers in the area of WSN.

2.2. Radio Frequency Identification (RFID) RCE

The modern RFID system infrastructures are made up of three primary components RFID transponders (label), RFID readers or transceivers and backend electronic databases. RFID transponders are distinguished based on their operating frequency: low frequency (LF), high frequency (HF), ultra-high frequency (UHF) and microwave. Transponders are categorized by their powering techniques such as: passive, semi-passive and active. The most common devices are passive RFID tags, where a battery-less IC device harvests power from a nearby RFID reader (deriving their transmission power from the signal of an interrogating reader) and uses it to respond to the reader with an identification number. In this section, various topics within the RFID area will be discussed. Table 6 shows a comparison in terms of specifications on LF, HF and UHF tags.

Table 5
A compilation of survey literatures related to WSN.

Year	Authors	Reference	Paper characteristic
2003	Prasanth Ganesan, Ramnath Venugopalan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller, Mihail Sichitiu	Ganesan et al. (2003)	Analysis of selected ciphers and hardware platforms on encryption overhead for WSNs.
2004	Kay Romer, Friedemann Mattern	Romer and Mattern (2004)	Classification of design spaces and factors in WSNs.
2006	Yee Wei Law, Jeroen Doumen, Pieter Hartel	Law et al. (2006)	Benchmark of selected block ciphers for WSNs.
2007	Rodrigo Roman, Cristina Alcaraz, Javier Lopez	Roman et al. (2007)	Cryptographic primitives (symmetric and asymmetric, hardware and software) for WSNs.
2007	Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury	Akyildiz et al. (2007)	A survey on WMSNs.
2008	Mark Hempstead, Michael J. Lyons, David Brooks, Gu-Yeon Wei	Hempstead et al. (2008)	Architecture and circuit designs for WSN.
2008	Zhijun Li, Guang Gong	Li and Gong (2008)	A comprehensive survey of WSN security problems, focusing on key management topics.
2008	Deepa Kundur, William Luh, Unoma Ndili Okorafor, Takis Zourntos	Kundur et al. (2008)	Detailed investigation in the area of DMSN, focusing on distributed form of security and privacy.
2009	Michael Johnson, Michael Healy, Pepijn van de Ven, Martin J. Hayes, John Nelson, Thomas Newe, Elfed Lewis	Johnson et al. (2009)	Comparative review of various WSN motes.
2009	Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou	Xiangqian et al. (2009)	A broad survey on WSN security topic.
2009	Jaydip Sen	Sen (2009)	Security requirements, various attacks, countermeasures in WSNs.
2009	Devesh C. Jinwala, Dhiren R. Patel, Kankar S. Dasgupta	Jinwala et al. (2009)	Investigation of lightweight ciphers on MICA2 (WSN).
2009	Stanislava Soro, Wendi Heinzelman	Soro and Heinzelman (2009)	A complementary survey in the area of VSN.
2010	Xueying Zhang, Howard M. Heys, Cheng Li	Xueying et al. (2010)	The efficiency of symmetric key cryptographic algorithms in WSNs.
2011	Döser, Erman Jamal, Abdul Vestlund, Supervisor Christian	Döser et al. (2011)	Design challenges and problems for ad hoc and sensor embedded systems.
2012	Thomas Winkler, Bernhard Rinner	Winkler and Rinner, (2012)	Detailed technical report on security aspects in VSN.
2012	Bulent Tavli, Kemal Bicakci, Ruken Zilan, Jose M. Barcelo-Ordinas	Tavli et al. (2012)	A detailed survey of VSN platforms.

Table 6
A comparison between, LF, HF and UHF RFID tags (Industries, 2010).

Frequency	**Low-frequency (LF)** 125–135 Hz	**High-frequency (HF)** 13.56 MHz	**Ultra-high-frequency (UHF)** 850–960 MHz
Read range	~10 cm	~1 m	1~2 m
Penetration of materials	Excellent	Good	Poor
Water resistance	No	Some extent	Yes
Power source	Passive (inductive)	Passive (inductive)	Passive (propagation)
Data rate	Slow	Fast	Very fast
Reading multiple tags	Poor	Good	Very good
Applications	Animal identification, car immobilizers	Libraries, passports, baggage tracking, ticket payments	Case tracking, tolls, baggage tracking, asset tracking
Approximate price per tag (USD)	1	0.5	0.15

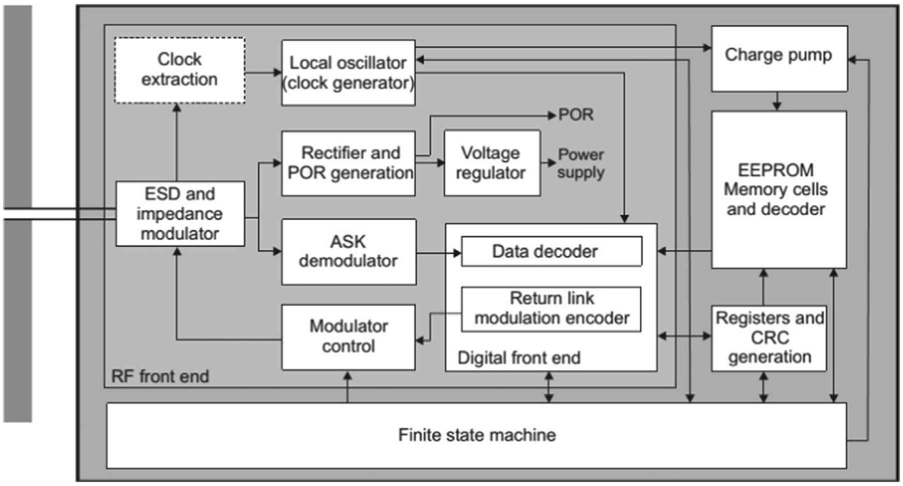
Clock extraction
Local oscillator (clock generator)
Charge pump
POR
Rectifier and POR generation
Voltage regulator
Power supply
EEPROM Memory cells and decoder
ESD and impedance modulator
ASK demodulator
Data decoder
Return link modulation encoder
Modulator control
Digital front end
Registers and CRC generation
RF front end
Finite state machine

Fig. 4. An illustration of the architecture within a HF/UHF RFID tag (Ranasinghe et al., 2006).

Dong-Liang et al. (2009) and Khan et al. (2009) identified a few key areas of RFID applications.

1. Supply chain automation
2. Asset management and tracking
3. Medical and healthcare applications
4. People identification, monitoring and tracking
5. Warehouse
6. Animal tracking
7. Toll collection systems
8. Ticketing system (cashless/wireless payment)
9. Sporting events

On the other hand, Langheinrich (2009) surveyed various privacy approaches for RFID systems. The author also identified four areas of RFID applications and their respective threats as:

1. Authentication – prone to counterfeiting, forging and theft.
2. Identification – prone to sniffing and unauthorized "listening".
3. Monitoring – prone to unauthorized tracking.
4. Alerting – prone to denial of service (DOS) to hide presence.

In this section, the highlight is given to the EPC tags (electronic product code). The two broadly adopted standards for this technology are the Electronic Product Code (EPC) Class-1 Generation 1 and Class-1 Generation 2 standards, which operate in the Ultra High Frequency (UHF) bands at 860–960 MHz and led by the EPCGlobal. The UHF tags are known to have the longest range (up to tens of meters) and are subject to ambient interference. In the paper by Khor et al. (2011), the lightweight cryptography is said to conform to the EPC Class-1 Generation-2 standard. Lightweight cryptographic functions such as the XOR, CRC and PRNG were mentioned and proposed as a part of the authors' solution to protect tags from eavesdropping and impersonation.

A general overview of the hardware components within a typical RFID transponder and a block diagram of a passive UHF/HF RFID label is discussed in Ranasinghe et al. (2006). The authors also stated that the current fabrication of Class I labels consists of around 1000–4000 logic gates while Class II labels may consist several thousand more gates. The authors further elaborated the three important components within the RFID: RF front-end, memory circuitry and the FSM logic circuitry (finite state machine). The authors propose an additional PUF circuit (Physical Unclonable Functions) which costs less than 1000 gates to tackle privacy and authentication issues. Fig. 4 illustrates the architecture within a UHF/HF tag extracted from Ranasinghe et al.

Juels (2005) stated that conventional symmetric cryptographic primitives are not able to be adopted in RFID because the RFID devices are computationally too weak to perform even basic symmetric-key cryptographic operations. However, Juels concluded in the same paper that their investigation proceeded under the assumption that symmetric-key cryptographic algorithms lie beyond the computational limited of RFID tags and stated that their proposed scheme might be adapted and even strengthen the RFID tags, capable of performing symmetric-key cryptography. The author also further elaborated that even if an RFID tag is unable to perform a full-version of primitive, a partial or a light-weighted form of crypto might still be deployable. Note that this paper was released in the year 2005 and this also points to the continuation of the search for lightweight crypto solution for RFID.

The Ph.D. thesis by David (2011) discussed two types of tags: silicon IC tags and printed tags. In the year 2008, 0.18 μm process was popular in IC fabrication and on the other hand, printed tags are conceived as the low-cost solution. The work performed comparison between the conventional silicon passive RFID tags and the printed ink passive RFID tags and information states the constraints and the resource boundaries of the RFID tags for security applications. Table 7 shows the security related system characteristic comparison.

Moreover, David also discusses some of the cryptographic solution known and their respective analysis. In the thesis, David mentioned some symmetric block ciphers. i.e.: TEA, SEA, mCRYPTON, KATAN, KTANTAN, DES, DESX, DESL, PRESENT, PRINTcipher, Lblock and Piccolo were further elaborated. Besides that, David has provided a classification of lightweight cryptographic algorithms. Fig. 5 shows an illustration of the said classification.

Rolfes et al. (2008) presented implementation results (using serialized architecture) requiring only 1000 Gate Equivalents (GE). By employing the PRESENT cipher, The authors presented three different architectures with the pipelined version achieving high throughput to area ratio but only claimed that the 1000 GE serial version is best suited for low-cost RFID devices and contactless smart cards.

Another paper by Juels and Weis (2005) has provided a very important piece of information. The UHF EPC tags claimed a popular ground because of its cost-effectiveness. But the obvious trade-off is the very constrained resources. Juels and Weis mentioned that an RFID tag may have a total of 1000–10,000 gates with a tight budget of 200–2000 specifically for security. Performing modular arithmetic over large fields or evaluating standardizing cryptographic operations such as the famous AES is not feasible.

Sarma et al. (2003) were the first to draw attention to lightweight cryptographic challenges emerging in the RFID field. From their work, it shows the convergence of the need for efficient hash functions and symmetric encryption schemes. They concluded that hardware implementation must be under 2000 GE, which includes the state memory and both encryption and decryption functionality. They also commented on the software implementation speed and size across all MCUs and CPUs should also be marked as a design consideration.

To investigate the resources available on tag, the authors have compiled a list of a few known models of RFID tags in the market by referring to Ibtechnology (2013), Deavours (2005), Bukkapatnam et al. (2005), and Corporation (2012). Table 8 shows the list of a few known RFID tags (LF, HF and UHF) and their resource specifications. Some of the latest RFID specifications can be found here: Corporation (2008b), Inc. (2013a, 2013b), and Corporation (2008a).

A survey by Aragones-Vilella et al. (2007) discussed the main approaches from the basic tag securities, symmetric key cryptography and the public key cryptography. The authors introduced a few protocols such as the OSK, YA-TRAP, and Hashing functions.

One survey (Juels, 2006) focuses more on the general overview of the security and privacy issues in RFID. Juels discussed a wide topic of RFID privacy issues and stated that in practice, commercial RFID devices are resource constrained leading to the deployment

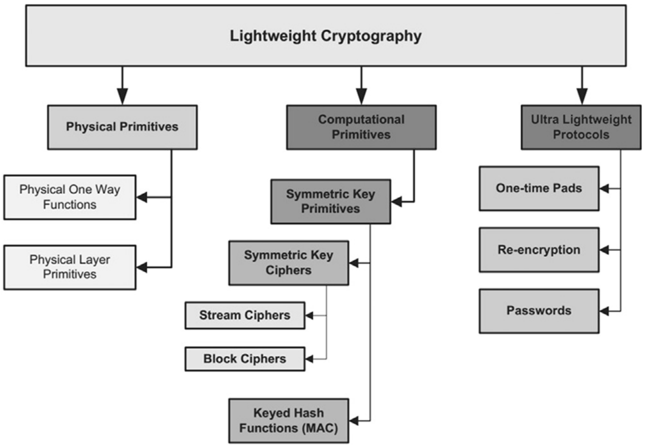

Fig. 5. The classification of lightweight cryptographic algorithms.

Table 7
Security related system characteristic comparison (table from David, 2011).

RFID tag types	Conventional silicon passive RFID tags	Printed ink passive RFID tags
Gates	2000 available for security primitive.	Less than 200 available for a security primitive.
Available memory	Most likely an EPC (Electronic Product Code) of 96–256 (refer to EPC global's tag data specification standard) and several hundred bits of user memory. Read–Write memory. Although further steps to reduce costs imply that it is likely to be Read Only memory (ROM).	Enough bits (96–256) to store a unique identifier. Additional bits may need to be implemented as ROM.
Power consumption	Tens of microwatts, and should not exceed that required for EEPROM read operation, so the tag read range requirements can be maintained. Currently EEPROM read operations require around 20–30 μW.	Few microwatts.
Performance	In North America, it is conceivable to allow a tag to expend around 400,000 clock cycles (based on a 1 MHz internal clock) during a 400 ms period (time constraint imposed by FCC regulations for UHF frequencies) for communications. In Europe under revised EN 302 208 regulations it is conceivable to allow a tag a maximum of 4 s for communications. Then performance appears to be mainly limited by user requirements and air interface protocols. Bit rates: 40–640 kbps (EPC global C1G2 protocol) Tag read rates of 200–1500 (demanded by end users)	Around 100 kbit/s will not be able to support complex anti-collision techniques and thus may only support the reading of few labels a second.
Read range	3–10 m for UHF and 200–500 mm for HF operation under FCC Regulations.	Much reduced read ranges. Currently UHF tags are not possible.
Communication Protocols used	The most prevalent standard for UHF tags is the CIG2 protocol. The multi-part ISO 18000 air interface standard defines protocols for a number of different frequencies; LF, HF and UHF. ISO 18000 Part 3 Mode 1 is possibly the most prevalent standard as of yet. The most commonly used HF standard, other than the ISO 18000, is ISO 14443 (types A and B).	No standardized protocols suitable for printed ink tags. New protocols based on the keeping their implementation to around 1200 transistors leaving around 800 transistors or 200 gates for security are needed
Tag IC footprint	Currently 18,000–30,000 gates for a Class I Generation 2 air interface protocol (C1G2) implementation. However these are expected to be simplified in the future to reduce the cost of tag ICs.	Around 2000 transistors (500 gates)
Available resources	32 bit random number generator (as required by the C1G2 protocol)	None.

Table 8
A compilation of specifications for various known RFID transponders (referred from Ibtechnology, 2013; Deavours, 2005; Bukkapatnam et al., 2005; Corporation, 2012).

Operating frequency	Transponder	Storage	User memory
LF			
125 kHz	Hitag1	256 bytes	192 bytes
125 kHz	Hitag S256/2048	256 bytes	248 bytes
125 kHz	Hitag2	32 bytes	16 bytes
125 kHz	EM4001/4102	8 bytes	5 bytes
125 kHz	MCRF200/123	16 bytes	14 bytes
HF			
13.56 MHz	Mifare 1k	1024 bytes	768 bytes
13.56 MHz	Mifare ProX	1024 bytes	768 bytes
13.56 MHz	SmartMX	1024 bytes	768 bytes
13.56 MHz	Mifare 4K	4096 bytes	3456 bytes
13.56 MHz	Ultralight	64 bytes	48 bytes
13.56 MHz	ICODE SLI/TagIT (ISO15693)	128 bytes	112 bytes
13.56 MHz	Mu-chip	128 bits	–
UHF			
902–928 MHz	Alien I2 (ALL-9250)	64 bits	–
902–928 MHz	Alien M (ALL-9254)	64 bits	–
902–928 MHz	Alien Squiggle (ALL-9238)	64 bits	–
860–960 MHz	IT36 Low profile Durable asset tag	TID=64 bits EPC=128 bits	512 bits
902–928 MHz	IT75 Low profile Durable asset tag	TID=64 bits EPC=128 bits	512 bits
865–868 MHz	IT76 Low profile Durable asset tag	TID=64 bits EPC=128 bits	512 bits
860–960 MHz	IT67 Enterprise Lateral transmitting (LT) tag	TID=64 bits EPC=240 bits	512 bits
860–960 MHz	IT65 Large rigid tag Gen2	TID=32 bits EPC=96 bits	0 bits
869/915 MHz	Tire tag insert	–	–
915 MHz	Container tag	–	–
902–928 MHz	Matrics/symbol Dual dipole	TID=112 bits EPC=128 bits	–
902–928 MHz	Matrics/symbol Single dipole	TID=112 bits EPC=128 bits	–

of weaker crypto-solutions. Hence, the vulnerability of the authentication protocols is exposed. Juels also mentioned that the RFID industry is still manufacturing cryptographically weak devices. There are only some devices like the Philips Mifare DESfire using the 3DES cipher. This shows that symmetric ciphers are feasible in RFID environment.

And lastly, the proposal by Poschmann et al. (2007) presented result of 1848 GE for the DESL version, proving that symmetric ciphers can be implemented under the constraint of 2000 GE. Note that due to this limitation, ciphers that fall below the 2000 GE mark are considered a candidate, suitable for the implementation in RFID environment. Table 9 shows a summary for the compilation of various surveys and papers in the area of RFID.

2.3. Wireless Identification and Sensing Platform (WISP) RCE

The WISP (Smith et al., 2006) was first developed under the project of Intel Research Seattle. WISPs have the similar capabilities of RFID tags with additional support for sensing and computing.

Compared to a RFID transponder, the WISP has a more powerful controller and spacious memory unit. Currently, there are three version of WISP (Contributors, 2013) and is shown in Table 10.

The most recent development is the WISP 5.0 but the information released is limited. Sample et al. (2008) give a complete description of the WISP, breaking down the WISP with detailed explanations from the analog front end, the modulation and demodulation, the digital section and power conditioning, packet coding and decoding to the power requirements and duty cycle. Fig. 6 shows an illustration of the hardware architecture and components within the WISP.

There are various proposals for the WISP's applications. For example, the paper by Yeager et al. (2008) has proposed using the WISP as a Passive Data Logger (PDL) and the paper by Wetherall (2008) has proposed the RFID sensor network (RSN) with the combination of WISP and readers. The paper by Czeskis et al. (2008)

Table 10
A table stating WISPs' version and their current status of development.

WISP name	MCU	Status
WISP 4.1DL (blue)	MSP-430F-2132	Ramping production
WISP 4.0DL (purple)	MSP-430F-2274	Deprecated
WISP 3.0	MSP-430F-2272	–
WISP G2.0 (red)	MSP-430F-2012	Limited use

Table 9
A compilation of survey literatures related to RFID.

	Authors	Reference	Paper characteristic
2005	Ari Juels	Juels (2005)	Proposal for a new lightweight security protocol as an effort towards minimalist cryptography.
2005	Ari Juels, A. Weis	Juels and Weis (2005)	Proposal for an augmented version of Hopper and Blum (HB) protocol against Learning Parity with Noise (LPN).
2006	Ari Juels	Juels (2006)	A broad survey on RFID security and privacy issues.
2006	Damith C. Ranasinghe, Peter H. Cole	Ranasinghe et al. (2006)	The introduction to RFID authentication problems and the proposal of the addition PUF circuit within RFID chip.
2007	J. Aragones-Vilella, A. Martinez-Balleste, A. Solanas	Aragones-Vilella et al. (2007)	Survey and analysis of basic security techniques, symmetric-key and public-key cryptography in RFID.
2008	Carsten Rolfes, Axel Poschmann, Gregor Leander, Christof Paar	Rolfes et al. (2008)	The proposal of three separate architectures of PRESENT implementation for resource constrained RFID.
2009	Marc Langheinrich	Langheinrich (2009)	A brief survey on RFID privacy approaches.
2009	Dong-Liang Wu, Wing W.Y. Ng, Daniel S. Yeung, Hai-Lan Ding	Dong-Liang et al. (2009)	A brief survey on current areas of RFID applications.
2009	M. Ayoub Khan, Manoj Sharma, Brahmanandha Prabhu R.	Khan et al. (2009)	A brief survey on current areas of RFID applications.
2011	Jing Huey Khor, Widad Ismail, Mohammed I. Younis, M.K. Sulaiman, Mohammad Ghulam Rahman	Khor et al. (2011)	A brief description of authentication and privacy problems in RFID and a proposal of a new fingerprint-based mutual authentication protocol.
2011	Mathieu David	David (2011)	Ph.D. thesis on the aspect of lightweight cryptography for RFID tags.
2012	Markku-Juhani O. Saarinen, Daniel Engels	Saarinen and Engels (2012)	The overview of design specifications, criteria and goals of a cipher for RFID.
2007	Axel Poschmann, Gregor Leander, Kai Schramm, Christof Paar	Poschmann et al. (2007)	The proposal of a new version of DES (namely the DESL) with claims of 1848 GE used, suitable for implementation in RFIDs.

is a security application related paper. The authors tackle a unique problem of defending against ghost-and-leech attacks against RFID tags and other contactless cards. The ghost-and-leech attack may be referred to as proxying, relay or man-in-the-middle attack. The authors have added a level of protection against the ghost-and-leech attack by limiting the context in which the contactless card can communicate with a reader. Although this literature has no direct relationship to symmetric cryptography, but it is worth mentioning that the authors have contributed efforts on finding a system to defend against man-in-the-middle attack incorporating WISP.

In Chae and Daniel (2007), the authors have implemented RC5 using WISP. Their implementation of RC5 32/12/16 (32-bit word, 12 rounds, 16-bytes key) has shown that conventional cryptography is no longer out of reach of a UHF tag. The WISP tag has 256 bytes of RAM and 8 Kbytes of flash. The authors have proven that the RC5 is able to operate under such constrained environment and further shown that symmetric cryptography is feasible on an RF-powered UHF WISP by providing the first implementation of conventional cryptography on WISP.

They stated that the ASIC approach is efficient in terms of power consumption and cost but are limited to a narrow set of applications, making it time consuming when realizing actual applications. They selected RC5 32/12/16 simply because of the cipher's simplicity and small memory requirements. Due to limited RAM size (256 bytes), minimizing instruction stack size and careful memory planning is the approach taken. RC5 requires expanded key table of size $2(r+1)$ words. With the 12 rounds, the expanded key table results to a total of $2(12+1)=26$ words ($1\text{ word}=4$ bytes), giving us a total of 104 bytes (the original key is 16 bytes). With 104 bytes used, only 152 bytes remained for the stacks. The authors have suggested that the other possible approach is to use pre-computed expanded key table and remain stored in the ROM or flash. They stated that in their implementation, the extended key table is computed once in every hardware reset, meaning that the key table is computed during the first active cycle and kept in RAM unless the WISP reaches brownout voltage.

Other than the implementation issues regarding the RC5 algorithm's memory requirements, operation issues such as reduced operating range for reliable flash writes and duty cycling are discussed. Fig. 7 shows a typical lifecycle of WISP with RC5. It can be observed that the WISP spends most of its time within the Power Save Mode (known as the LP4). The only period that the WISP is active when is there is a reader in range. While in Chae et al. the execution time for the three operations of RC5 on WISP tags are presented, operating at a small distance of 30 cm. If the reader-tag distance is beyond 60 cm, WISP does not function except for a few sporadic reads. This information is useful as it gives us a clearer idea of what are the events that occur when a WISP is in distance with a reader, which is similar to the behavior of an RFID system.

Table 11 shows a broader comparison in terms of device resources and properties.

Table 12 shows a summary for the compilation of various surveys and papers in the area of WISP.

2.4. Internet of Things (IOT) RCE

The term Internet of Things (abbreviated as IOT) refers to the interconnectivity of embedded computers. IOT usually extends its definition of the connectivity between devices and computers beyond the normal machine-to-machine communication, by offering advanced services, systems and even functionality. The very idea of having devices with computing capability to be connected to larger networks is a very significant idea and poses new challenges in any form of modern applications and devices. Applications that researchers have identified for the IOT includes: environmental monitoring, energy management, industrial and asset management, home automation, healthcare monitoring systems, etc.

However, integration with the Internet implies that the devices will have an IP address as a unique identifier. IOT devices will inherit the security threats of a generic computer because it is no longer a simple and isolated device. IOT devices inherently obtain

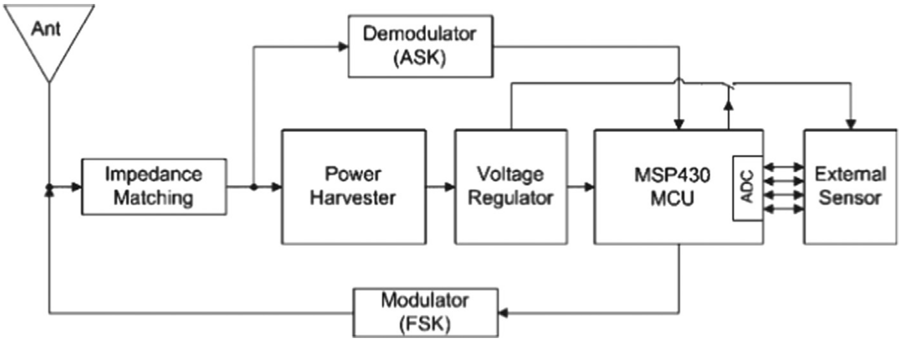

Fig. 6. An illustration of the WISP platform and its components (Sample et al., 2008; Smith et al., 2006).

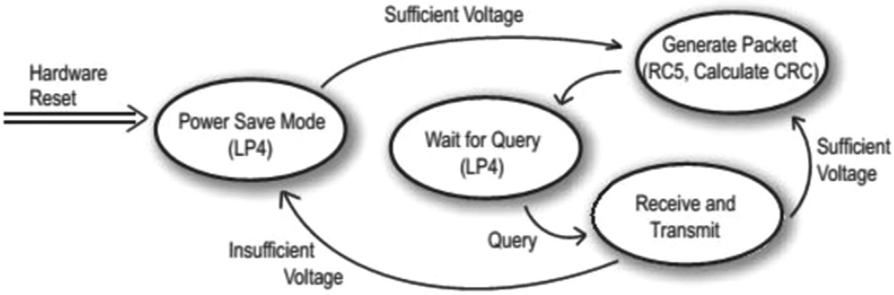

Fig. 7. An illustration of the WISP life cycle (Chae and Daniel, 2007).

Table 11
A comparison of WISP with other RFID devices (Chae and Daniel, 2007).

Platform	μProcessor	Power Source	Bus (bit)	Clock (MHz)	Storage (Memory)	Read range	Cost (USD)
WISP 4.1DL (blue) (Contributors, 2013)	MSP-430F-2132	UHF RF	16	12 (@2.7 V) 16 (@3.3 V)	8 KB+256 bytes flash 512 bytes RAM	~3 m (10 ft)	–
WISP 4.0DL (purple) (Contributors, 2013)	MSP-430F-2274	UHF	16	12 (@2.7 V) 16 (@3.3 V)	32 KB+256 bytes flash 1 KB RAM	~3 m (10 ft)	–
WISP 3.0 (Contributors, 2013)	MSP-430F-2272	UHF	16	12 (@2.7 V) 16 (@3.3 V)	32 KB+256 bytes flash 1 KB RAM	~3 m (10 ft)	–
WISP G2.0 (red) (Contributors, 2013)	MSP-430F-2012	UHF	16	12 (@2.7 V) 16 (@3.3 V)	2 KB+256 bytes flash 128 bytes RAM	~3 m (10 ft)	–
EPC Gen1	State machine	UHF RF	–	–	64 bits	3–7.5 m	Active: 25–100 Passive: 0.05–5
EPC Gen2	State machine	UHF RF	–	–	96/128 bits	3–7.5 m	Active: 25–100 Passive: 0.05–5
DemoTag (HF) (Aigner, 2006)	Atmel ATMega128 microcontroller	Battery	8	16	4 KB EEPROM 4 KB SRAM 128 KB flash	–	~720
DemoTag (UHF) (Aigner, 2006)	Atmel ATMega128 microcontroller	Battery	8	16	4 KB EEPROM 4 KB SRAM 128 KB flash	–	~965
Mica2	Atmel Atmega 128L	Battery	8	8	128 KB ROM 4 KB RAM 512 KB flash	40–50 m	99
Microchip	MCRF202	Inductive	12	0.4	96/128 bits	1.3–10.2 cm	
Proxmark3 (Westhues, 2007)	CPU: Atmel AT91SAM7S256 FPGA: Xilinx Spartan-II	USB	32	16	256 KB flash 64 KB SRAM	–	399
RFID Guardian (RF40) (Rieback et al., 2006)	CPU: CPU Babyboard with BlackFin BF547 FPGA: Altera EP3C16E Cyclone III Series	Battery	32	520	16 MB flash 64 MB SRAM	0.5 m	–

Table 12
A compilation of literatures related to WISP.

Year	Authors	Reference	Paper Characteristic
2006	Joshua R. Smith, Alanson P. Sample, Pauline S. Powledge, Sumit Roy, Alexander Mamishev	Smith et al. (2006)	WISP's proposal
2007	Hee-Jin Chae, Daniel J. Yeager, Joshua R. Smith, Kevin Fu	Chae and Daniel (2007)	Symmetric Cryptography (RC5) on WISP UHF Tag
2008	Alanson P. Sample, Daniel J. Yeager, Pauline S. Powledge, Alexander V. Mamishev, Joshua R. Smith	Sample et al. (2008)	Detailed discussion of the WISP design
2008	Daniel J. Yeager, Pauline S. Powledge, Richa Prasad, David Wetherall, Joshua R. Smith	Yeager et al. (2008)	WISP as a PDL (Passive Data Logger)
2008	Michael Buettner, Ben Greenstein, Alanson Sample, Joshua R. Smith, David Wetherall	Wetherall (2008)	RSN (RFID sensor network)
2008	Alexei Czeskis, Karl Koscher, Joshua R. Smith, Tadayoshi Kohno	Czeskis et al. (2008)	Defense against ghost-and-leech attacks

the security problems of any computer connected to the Internet. IOT devices are mostly embedded computing systems that have the nature of low-power radios and low-computing power. In this paper, the survey of cryptographic solutions extends to the IOT RCE area because of the nature and the specification of the IOT devices. For example, the survey by Kai and Lina (2013) states some of the security problems in IOT, which can be divided into three layers: Application, Network and Perception layer. All these layers can be protected by using secret key algorithms, like a typical WSN, which includes any cryptographic approaches. In the perception layer, devices such as the RFID, sensor, GPS, Bluetooth and Zigbee will have a role in providing the "real" hardware for data collection, data processing and data transmission. These are

Table 13
A qualitative performance comparison between RC5 and AES (Doomun and Soyjaudah, 2009).

Rank	Size optimized			Speed optimized		
	Code memory	Data memory	Speed	Code memory	Data memory	Speed
Performance by key setup						
1	RC5	Rijndael	Rijndael	RC5	Rijndael	Rijndael
2	Rijndael	RC5	RC5	Rijndael	RC5	RC5
Performance by encryption						
1	RC5	RC5	Rijndael	RC5	RC5	Rijndael
2	Rijndael	Rijndael	RC5	Rijndael	Rijndael	RC5

the components that requires the aid of security algorithms (symmetric or asymmetric), which is also mentioned by Kai and Lina. The IOT RCE is an emerging research area which deserves much attention from researchers to develop new security and cryptographic solutions.

2.5. *Other reviews or surveys on cipher candidates*

Although John Jacob (John, 2012) did not specifically mention in which area of the RCE his paper was referring to, but he did mentioned examples like RFID systems, WSNs and Smart Cards. John surveyed four block ciphers: AES, DESL, HIGHT and PRESENT and three stream ciphers: Trivium, Grain and Humming-bird. A comparison paper by Doomun and Soyjaudah (2009) analyzed the RC5 and the AES cipher for WSN in terms of cipher complexity, performance parameters, execution time and energy consumption. Table 13 shows a qualitative comparison by the authors.

On the other hand, the work by Good et al. (2006) presented a good review on selected stream ciphers. The ciphers reviewed are filtered in terms of security and costs, and finally the author concluded by categorizing the stream ciphers according to their suitability for low resource hardware environment as shown in Table 14.

3. Taxonomy of symmetric cryptographic solutions in resource constrained environments

To identify the types of ciphers known, we have grouped the currently known ciphers by the period of their release. To further distinguish the ciphers, the structural designs of the ciphers are classified. Within the symmetric block cipher class, ciphers can often be grouped by their structural anatomy: substitution-permutation network (SPN) and Feistel network (FN). Note that not all ciphers falls into the SPN or the FN class.

3.1. *Modern symmetric key block ciphers*

In this paper, we would like to categorize the pre-millennium (pre-2000) and post-millennium (post-2000) block ciphers as modern symmetric key block ciphers. The reason is simply because of the relevance of those ciphers in today's cryptographic development. For example, the advanced encryption standard (AES) is still relevant and safe despite being released since the year 2000. The AES can be found in a wide-variety of hardware platforms, from small smart cards to high performance computers. Even in the recent developments of x86 based microprocessor and the solid state devices, the AES is widely used. Before the birth of AES, the famous data encryption standard (DES) was proposed in the year 1979, International Data Encryption Algorithm (IDEA) in the year 1991 and the RC5 in the year 1994. These ciphers are not to be over-looked and they are still prominent algorithms in the cryptographic scene.

3.2. *Involution ciphers*

Involution ciphers came into the spotlight with paper titled "The ANUBIS block cipher" by Barreto and Rijmen (2001a). The idea of involution operations is not new, i.e. bitwise XOR and transposition of a matrix is an involution operation. But when the idea is used for involution operations within ciphers, this has impacted the cipher design and the efficiency of hardware implementation. Involution ciphers play an important role when a design environment is so constrained that only the forward encryption hardware or codes are being included in the system. In some applications, forward encryption would suffice as the back-end server, data center or sink would handle the decryption of messages. But for applications that require on-site encryption and decryption, a decryption code or hardware has to be present. In RCEs, most cases reflect towards a single direction security approach, having only the forward encryption and puts weight on the backend network. With the assistance of involution ciphers, the forward-encryption hardware can be reused for decryption purposes. With identical circuit for encryption and decryption, the resources are effectively halved, compared to the original approach to implement both encryption and decryption circuit.

3.3. *Lightweight symmetric block ciphers*

There are no obvious factors to group which ciphers are considered in the "lightweight" class. We have conducted a wide area of research to determine which are considered "lightweight" ciphers. Most of the cipher proposals claimed that the proposed algorithm is lightweight. The foundations of such claims are not clear but the motivation is very obvious: for the implementation of RCE devices. When the resource constrained environments (RFID in particular) were at the center of the research focus, the aim was to find a cryptographic solution to be resource friendly and to occupy as little resource as possible. Hence, the term "lightweight cipher" was given to such an algorithm. In this section, we have chosen to classify the lightweight block ciphers as they were, by referring to each cipher's original proposal and the claims of the cipher's designer.

3.4. *Stream ciphers*

Stream ciphers are formally considered as symmetric ciphers. In the paper by Good et al. (2006), the authors had made a good explanation and depiction of how a typical stream cipher works. The nature of the stream ciphers involves generating a sequence of random and secure bits which is known as the "keystream". The "keystream" is combined with the plain-text or the cipher-text using bit-wise XOR operations. Fig. 8 shows an illustration of the components of a stream cipher.

During the year 2004, a project initiated by the European Commission, under the Information Societies Technology (IST

Table 14
Categories of stream ciphers for low-resource hardware. (Good et al., 2006).

Category	Candidate stream ciphers for low resource hardware
Lowest resource, high speed (approx. 100 Mbps)	Grain, Trivium
Low resource, moderate speed (approx. 10 Mbps)	Mosquito, Sfinks, Hermes8
Moderate resource, very high speed (approx. 1 Gbps)	Phelix
High resource or broken (Insecure)	ABC, Achterbahn, Dicing, Dragon, F-FCSR, HC-256, MAG, MICKEY, Mir-1, NLS, Polar Bear, Pomaranch, Py ("Roo"), salsa20, Sosemanuk, SSS, TSC-3, WG, Yamb
Commercial i.e. "Not free for all"	CryptMT, Decim, Edon80, Frogbit, Lex, Rabbit, Trbdk3, Vest, ZK-crypt

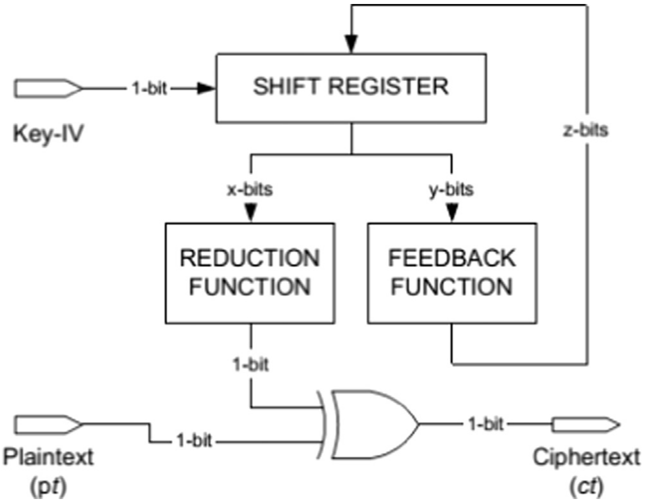

Fig. 8. An illustration of a generic stream cipher (Good et al., 2006)

Table 15
The stream ciphers under the eSTREAM portfolio.

Profile 1 (software)	Profile 2 (hardware)
HC-128	Grain
Rabbit	MICKEY
Salsa 20/12	Trivium
SOSEMANUK	

Program) named the “eSTREAM” in search for good stream cipher candidates. 34 stream ciphers were submitted and evaluated. The stream ciphers that made up the eSTREAM portfolio are shown in Table 15. Note that profile 1 are meant for software applications with high throughput requirements and profile 2 are meant for hardware applications with constrained resources in terms of storage, gate count or power consumption.

3.5. *Substitution-Permutation Network (SPN)*

A Substitution-Permutation Network (often referred to as the SPN) is a network that takes in blocks of plaintext and keys, applies alternating rounds of substitution layers (S-boxes) and permutation layers (P-boxes) to produce the final ciphertext. The S and P boxes consist of common operations that are “friendly” towards hardware implementation. As for the decryption, it is simply done by reversing the whole process: inverse S-boxes, P-boxes and a reverse order of the same round keys. A SPN based cipher, is basically a network of S and P boxes. The main advantage of a SPN is that the structure is simple and easy to analyze. One of the significant ciphers made that uses the SPN structure is the AES cipher. Other ciphers including the mCRYPTON, PRINCE, ANUBIS, KHAZAD and many more also uses the SPN structure. However, S-layer or P-layer alone does not contribute much cryptographic strength. A good design of SPN together with alternating S and P-layers (satisfying Shannon's confusion and diffusion properties) will produce a cipher that passes the strict avalanche criterion (SAC) and remain strong against chose-cipher text attack.

3.6. *Feistel network (FN)*

A Feistel cipher is known as a symmetric structure used in construction of block ciphers. It is also commonly referred to as the Feistel network or FN in short. Many uses this structure by having the advantage of identical or very similar encryption and decryption operations with only a reversal of key schedule.

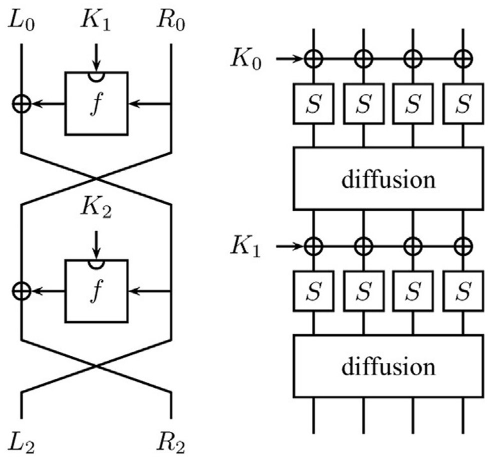

Fig. 9. An illustration of the SPN and FN.

Therefore, the size of the circuitry and the codes are nearly halved. Ciphers such as the SEA, TEA, DES, DESL and CLEFIA have Feistel structure. Fig. 9 (De Canniere et al., 2006) has been presented as an illustration of both the SPN and the FN.

Table 16 shows the taxonomy of symmetric block cryptographic algorithms covered in this paper.

4. Modern symmetric key block ciphers

4.1. *3DES (triple DES, TDEA)*

3DES, sometimes referred to as the triple DES, is inspired by the original cipher: DES. 3DES provides a relatively simple fix against brute-force attack by simply increasing the key size. In search for a more recent implementation results, we refer to Prasun Ghosal and Manish Biswas (2010). The intention is to search for a more recent implementation results which would give a more accurate data on the current implementation development. The author has presented numerous implementation results. By using Vertex5 XCVLX50, Number of Slices (NOS) is 266. Using Vertex5 XC5V1 × 110T, NOS is 266. These two results are the smallest known implementation figures.

4.2. *3-Way*

Created by Daemen et al. (1994), the 3-Way cipher is designed to be very efficient in various hardware platforms from 8-bit processors to specialized hardware. The 3-Way cipher is designed to have some mathematical features, enabling almost all the decryption to be done in exactly the same circuits as the encryption. In the paper “A New Approach Towards Block Cipher Design”. The authors presented a CMOS model for the 3-Way cipher, claiming to reach a throughput of 1 Gbit/s using standard technology (1.2 μ CMOS). There are no hardware occupancy reports of the 3-Way cipher.

4.3. *A5/3 (also known as the KASUMI)*

The A5/3 cipher or the KASUMI block cipher (modified and derived version of MISTY cipher) is known for its application in on 3GPP GSM wireless networks. With 64-bit blocks and 128-bit keys, the A5/3 was developed by Mitsubishi Electric Corporation. However, it is claimed that the A5/3 is unsafe and prone to related key attack. The A5/3 (Dunkelman et al., 2010) algorithm was developed for third generation GSM telephony in 2002 and it is claimed to be already implemented widely in headset set and telephones.

Table 16
Taxonomy of 100 symmetric block ciphers.

Taxonomy of symmetric block ciphers	Substitution Permutation Network (SPN)	Feistel Network (FN)	Others
Symmetric block ciphers	(4.2) 3-Way (4.4) AES (4.9) CRYPTON (4.10) CURUPIRA-1 (4.11) CURUPIRA-2 (4.27) RAINBOW (4.33) SERPENT (4.34) SHARK (4.36) SQUARE	(4.1) 3DES (4.3) A5/3 (4.5) Blowfish (4.6) Camelia (4.7) CAST-128 (4.8) CAST-256 (4.12) DEAL (4.13) DES (4.14) FEAL (4.15) FEAL-N/NX (4.16) GOST (4.17) ICE (4.19) Khafre (4.20) Khufu (4.21) LOKI97 (4.22) Lucifer (4.23) MAGENTA (4.24) MARS (4.25) MISTY-1 (4.26) MISTY-2 (4.28) RC2 (4.29) RC5 (4.30) RC6 (4.31) SEED 128 (4.32) SEED 192/256 (4.35) Skipjack (4.38) Twofish	(4.18 IDEA) (4.37) Threefish
Involution ciphers	(5.1) ANUBIS (5.2) ARIA (5.3) ICEBERG (5.4) Khazad (5.5) PP-1 (5.6) PUFFIN	–	–
Lightweight block ciphers	(6.6) EPCBC (6.10) KLEIN (6.13) LED (6.14) mCRYPTON (6.16) NOEKEON (6.18) PRESENT (6.19) PRINCE (6.20) PRINTcipher	(6.1) CLEFIA (6.2) DESL (6.3) DESX (6.4) DESXL (6.5) DST40 (6.7) HIGHT (6.12) LBlock (6.15) MIBS (6.17) Piccolo (6.21) SEA (6.24) TEA (6.25) TWINE (6.26) TWIS (6.27) XTEA (6.28) XXTEA	(6.8) KATAN (6.9) KEELOQ (6.11) KTANTAN (6.22) SIMON (6.23) SPECK
Stream ciphers	–	–	(7.1) A2U2 (7.2) A5/1 (7.3) A5/2 (7.4) AES-CTR (7.5) Edon80 (7.6) GRAIN (7.7) HC-128 (7.8) HC-256 (7.9) Hermes8 (7.10) LEX (7.11) MICKEY-128 (7.12) Phelix (7.13) Polar Bear (7.14) Rabbit (7.15) RC4 (7.16) Salsa20/12 (7.17) SFINKS (7.18) SNOW3G (7.19) SNOWv1 (7.20) SNOWv2 (7.21) SOSEMANUK (7.22) TinyStream (7.23) Trivium (7.24) VEST (7.25) WG-7

Table 16 (*continued*)

Taxonomy of symmetric block ciphers	Substitution Permutation Network (SPN)	Feistel Network (FN)	Others
			(7.26) WG-8 (7.27) WG-16 (7.28) ZUC

4.4. Advanced Encryption Standard (AES, Rijndael)

The Advanced Encryption Standard, termed the AES, is sometimes referred to as the Rijndael. The Rijndael is a well known and well established cipher and had caught researchers' attention soon after winning the AES title. The AES has a great impact in today's modern cryptography. There are many hardware implementation papers on the AES but in this paper, only a few are mentioned. A paper by Rouvroy et al. (2004) reported results by using the Xilinx XC2S30-6 FPGA, the occupied slices amount to 163 and by using the Xilinx XC2V40-6 FPGA, the occupied slices amount to 146. On the other hand, the work by Good and Benaissa (2005) showed their smallest AES processor, with only 124 slices occupied on a Spartan-II XC2S15-6 FPGA. Another small AES hardware implementation, the paper by Kong et al. (2013) presented a small AES FPGA processor, using only 100 slice flip-flops in terms of memory.

4.5. Blowfish

Blowfish cipher is a 64-bit block cipher and is open for public use. Although Blowfish is known to be susceptible to attacks, it is still being used in computers. By referring to the most recent literature, Blowfish cipher (Kumar and Baskaran, 2010) is implemented at a cost of 1088 slice flip flops, the smallest among other Blowfish implementations.

4.6. Camelia

Derived from MISTY1 and developed by Mitsubishi and NTT Japan, the Camelia (Aoki et al., 2000, 2001;) is based on a Feistel structure with 18 and 24 rounds, with respect to 128-bit and 256-bit keys. According to Aoki et al. (2000), this block cipher is designed to be suitable for both software and hardware implementations, ranging from low-cost smart cards to high-speed networks. In an implementation paper by Huiju and Heys (2007), the authors presented a compact hardware implementation of Camelia with concurrent error detection, yielding a result of 1052 slices on a Virtex-E v1000efg860 FPGA.

4.7. CAST-128

The CAST-128 (Adams, 1997) cipher is a 64-bit block cipher developed by Carlisle Adams was published as document RFC 2144 during 1997. One prominent application of this CAST cipher is in the popular e-mail ciphering tool Pretty Good Privacy (PGP). In the paper by Sugawara et al. (2007), the author have presented their implementation results based on ASIC. Using their own method of hardware implementation and S-box minimization, Sugawara et al. managed to present circuits of 26.4–39.5K-gates using 0.13 μm and 0.18 μm technology.

4.8. CAST-256

Similar with CAST-128, the CAST-256 (Adams and Gilchrist, 1999) cipher shares the same attributes and structures but with a different block size of 128-bit. In the paper by Riaz and Heys (1999), the author claimed that the hardware required for a CAST-256 implementation using the XC40200 FPGA is about 5052 CLBs or 72% of the available CLBs.

4.9. Crypton

CRYPTON (Hong et al., 1999) was proposed as a candidate for the AES and is described to be efficient in both hardware and software implementations. In the same paper, the authors did mention that "Gates" means a 2 input NAND gate, which is equivalent to 4 transistor and understood as the Gate Equivalent (GE). The authors have presented their results on the full-round model with 46,259 GE, and the two-round model is 18,322 GE, with both area optimized.

4.10. CURUPIRA-1

The CURUPIRA cipher (Barreto and Simplício, 2007) is proposed as a special purpose block cipher for platforms that have constrained power and processing time such as the mobile and sensor network platforms. The paper by Kitsos et al. (2008b) has given a benchmark of the CURUPIRA via hardware implementation. The authors have reported a result of 9450 GE using 0.13 μm process and throughput of 960 kbps at 100 kHz. The authors also state the applications of the CURUPIRA in the RFID are not feasible at the moment but mentioned that it is still suitable for other resource constrained environment such as the WSN.

4.11. CURUPIRA-2

As for the CURUPIRA-2 cipher (Simplíciojr et al., 2008), it is designed with a faster key-schedule, with a trade-off of lower level security against related-key attacks. Similar with CURUPIRA-1, the CURUPIRA-2 also operates on 96-bit blocks and uses 96, 144 and 192-bit keys. There are no hardware implementation examples using the CURUPIRA-2 cipher.

4.12. DEAL

The DEAL cipher is originally developed by Lars R. Knudsen (Knudsen, 1998) as a proposal to the AES. Like the other candidates submitted to the AES contest, the DEAL cipher is a feistel cipher which uses DES as the round function. Therefore, in the report by Lars R. Knudsen (1998), the author mentioned that the DEAL cipher can be implemented using the existing DES-hardware or DES-software. However, there are no hardware implementation examples for this particular cipher.

4.13. Data Encryption Standard (DES)

The DES was first designed by the IBM during the 1970s and was later replaced by Triple-DES and ICE cipher. Due to the reason that the DES was once the prominent standard in the area of cryptography, the research conducted on the DES is vast and many hardware proposals were made. In this section, only the smallest hardware for DES is mentioned here. The paper by Saqib et al. (2004) reported an FPGA implementation of the DES cipher. The authors used the Xilinx XCV400e FPGA and reported a result of 117 CLB slices used for the DES implementation.

4.14. FEAL-4

The FEAL cipher (Miyaguchi, 1991) is also known as the "Fast data Encipherment Algorithm" was published by Akihiro Shimizu and Shoji Miyaguchi from Nippon Telegraph and Telephone (NTT). The FEAL-4 cipher has 4 rounds and 64-bit keys. The FEAL cipher was designed as software for 8-bit microprocessors in IC cards, and the merit of the initially developed FEAL-8 was that it was faster than DES. Despite having various weaknesses found in FEAL, there are no known implementation examples.

4.15. Feal-N/Nx

During 1990, by using the FEAL-8 as a base, the FEAL-N/NX (Miyaguchi, 1991) was developed. The FEAL-N or the FEAL-NX is the variant of FEAL-4 cipher. FEAL-N has a key length chosen by the user and the FEAL-NX has a larger 128-bit key. However, cryptanalytic attack has been found via exhaustive key search. However, there are no hardware implementation literatures.

4.16. GOST

The GOST block cipher is a Soviet and Russian government standard symmetric key block cipher developed during 1989. Also known as the GOST 28147-89, it is a well-known 256-bit block cipher which is a plausible alternative for AES-256 and triple DES. In a recent paper by Poschmann et al. (2010), the authors have presented result of 650 GE, suggesting applications for the low-area hardware environments.

4.17. ICE

Designed by Matthew Kwan during 1997, the ICE cipher (also known as the Information Concealment Engine) is a 16-round Feistel cipher designed for speed and simplicity. Other than the ICE cipher, Kwan also described a faster variant the Thin-ICE and an open-ended variant with varying key-size, the ICE-n. In a paper by Fournaris et al. (2003), the author presented their work on implementation of the ICE cipher on the XILINX v600efg900 FPGA, occupying 5331 CLB slices (@42%) and running at 29.1 MHz.

4.18. International Data Encryption Algorithm (IDEA)

The IDEA (Hoffman, 2007) cipher was first published in 1991 and was also a candidate in the NESSIE project. However, IDEA did not replace DES, but was incorporated into Pretty Good Privacy (PGP) after the BassOmatic cipher in PGP was found insecure. Comparing with cryptosystems like the AES, IDEA cipher is considered obsolete today. The IDEA cipher is mentioned in this paper because, although it is obsolete, IDEA cipher still serves as a teaching tool, to help students to understand DES and AES. There are no known implementation results.

4.19. Khafre

Designed by Ralph Merkle in 1989 while working at Xerox's Palo Alto Research Center, the Khafre (Merkle, 1991) cipher is said to have better "key-agility" due to its non-key-dependent S-boxes but shows slower encryption. The Khafre is a unique Feistel cipher with 64-bit block and a key size of 512 bits. However, there are no known implementation results using this cipher in applications.

4.20. Khufu

Also designed by Ralph Merkle in 1989, Khufu (Merkle, 1991) (similar to Khafre) is an unusual block cipher with 64-bit block and a key size of 512 bits, tailored for the time-consuming key-setup. In contrast of Khafre cipher, the Khufu cipher performs better in bulk encryption of large amounts of data. Despite having interesting fetures, there are no known hardware implementation models.

4.21. LOKI97

LOKI97 is originally proposed by Lawrie Brown and Josef Pieprzyk in the year 1998. LOKI97 was the improved version of LOKI89 and LOKI91 with larger keyspace and harden key schedule. LOKI97 was suggested to be used in 8-bit processor, ATMs, HDTVs, voice and satellite applications. The reason that LOKI97 is mentioned here is because LOKI97 was submitted to NIST as an AES candidate. However, there are no known hardware implementation results.

4.22. Lucifer

Lucifer (Feistel and Ibm, 1971) is a direct precursor of the DES. It was developed by Horst Fesitel and his colleagues at IBM during the 1971. This cipher is obsolete but it is mentioned here because this cipher led to the formation of DES after the NSA reduced the key size to 56-bits and the block size to 64-bits. The practical example of this cipher is that the Lucifer was commercially used for electronic banking in the 1970s.

4.23. Magenta

The MAGENTA cipher (Huber and Wolter, 1996) is also known as the Multifunctional Algorithm for General-purpose Encryption and Network Telecommunication. The MAGENTA cipher was developed under the initiative to create a chip that is capable of running at high data rates. The paper by Huber and Wolter (1996) presented a high speed implementation of the MAGENTA cipher using 0.6 μm CMOS technology, with 1 Gbps and 60 mm^2 chip area. It is also mentioned that the MAGENTA is suitable for high-speed application such as the ATM networks.

4.24. Mars

The MARS cipher (Burwick et al., 1999) by the IBM was selected as the fifth and last in the AES competition. MARS has a fixed block size of 128-bits with variable key length of 128, 192 and 256 bits. In the specification document for MARS (Burwick et al., 1999), the cell count for the MARS implemented is approximately 70,000 cells while a typical DES implementation is approximately 28,000 cells on an 8-bit processor.

4.25. MISTY-1

The MISTY-1 cipher was developed in 1995 by Mitsuru Matsui, Ichikawa Tetsuya, Sorimachi Toru, Tokita Toshio, and Yamagishi Atsuhiro (Matsui, 1997). The MISTY-1 cipher is a Feistel network with a variable number of rounds (multiple of 4). MISTY-1 is later replaced by it successors: MISTY-2 and KASUMI, which is also covered in this article. The work by Yamamoto et al. (2008) has presented a small hardware for MISTY-1 64-bit cipher using the Fujitsu 0.18-μm CMOS standard cell library. The simulated result shows a gate size of 3.95 Kgates, focusing on the optimization of the FO/FI functions. On the other hand, the paper by Kitsos et al. (2005) presented results of 4735 CLB slices using XCV1000 FPGA and 4039 CLB slices on XCVII3000 FPGA.

4.26. MISTY-2

Also designed by Matsui (1997), the MISTY-2 has the similar FO function as the MISTY-1. The difference between MISTY-2 and MISTY-1 is that the MISTY-1 has 8 rounds and MISTY-2 has 12 rounds. However, there are no hardware implementation examples for MISTY-2 cipher known.

4.27. Rainbow

Rainbow is a 128-bit block cipher that accepts a 128-bit key. Rainbow is similar to the SQUARE and SHARK. In the paper by Lee and Kim (1997), the RAINBOW cipher is implemented on a Pentium 400 MHz using windows 95, compiled via visual C++. No hardware implementation data were found in the literature.

4.28. RC2

The RC2 cipher is a cipher designed by Ron Rivest in 1987. RC2 is a 64-bit block cipher with a variable size key. The RC2 performs faster than DES and can be made more secure or less secure than DES against exhaustive key search with suitable key sizes. The paper by Knudsen et al. (1998) described the cipher and further presented attempts to use differential and linear cryptanalysis on the RC2. There are no examples of hardware platforms running RC2.

4.29. RC5

The RC5 is also a cipher designed by Ron Rivest which stands for “Ron’s Code” (Rivest, 1994). A paper that presents the hardware implementation of RC5 cipher using FPGA by Rashidi (2012) has reported using Quartus II 9.1 with Stratix II FPGA EP2S15F484C3 to implement the RC5 cipher, showing results of 17% logic utilization (1787 ALUT) and a maximum clock frequency of 175.69 MHz. Besides the work by Rashidi, Koch et al. (2006) used XC4VLX25 FPGA for RC5 cipher, showing logic utilization of 87% (9388 slices) and the work by Yoshikawa and Sakaue (2011) used XC5VLX30/50, with 2488 slices (51%) and 8893 LUTS (46%). And lastly, the paper by Bevi et al. (2012) showed that 1698 slices occupied using a Virtex II Pro FPGA with their proposed 12-stage pipeline architecture.

4.30. RC6

The RC6 cipher is based on the RC5 cipher. It was designed to meet the strict requirements of the AES competition. In paper by Riaz and Heys (1999), the authors presented an FPGA implementation of the RC6 cipher. The FPGA used is the XC40200 FPGA and the hardware needed for the RC6 encryption is 4944 CLBs for the RC6 core plus 704 CLBs for key storage purposes and another 64 CLBs for storing the 128-bit input data. Other than this, there is also some data flow and control logic overhead (on the order of 750 CLBs). So, the total FPGA resources required is about 6450 CLBs or 91% of the available CLBs in the target device (Riaz and Heys, 1999). On the other hand, the results presented by Beuchat (2003) which has the smallest hardware footprint is reported at 1560 slices using XC2V1000 (1 round), 10,288 slices using XC2V3000-6 (20 rounds), and 1918 slices using XCV2000E-6 (20 rounds). And lastly, the paper by Shareef et al. in 2008 (Shareef et al., 2008) used the Xilinx Virtex II (XC2V10000) FPGA with encryption only sequence, yielding 494 slices via iteration mode.

4.31. SEED (SEED 128)

The SEED (Agency, 1999) cipher is broadly used in the South Korean industries. Developed by Korea Internet & Security Agency (KISA), the SEED cipher is a 128-bit block cipher with 16-round Feistel structure, recommended for mobile devices and smart cards. In the paper by Young-Ho et al. (2000), the authors presented an FPGA implementation of the SEED cipher. It was synthesized using the FPGA Compiler of SYNOPSYS with ALTERA 10K library and simulation was performed in ALTERA MAX+PLUS 11. The design occupies 80% chip area of ALTERA I0KE. On the other hand, a more recent paper by Yi et al. (2009) presented a result of 6.4 Gbps throughput on a Virtex-V XC5LX110T while occupying 53% of the chip area. The number of slice LUTs used is 36,678 (53%) and the number of slice flip-flops used is 5314 (7%).

4.32. SEED 192/256

The SEED cipher was later updated with 2 different key lengths, proposed by Jeong et al. (2009). The idea for the SEED cipher addition is to cater for various platforms, capable of supporting 192 and 256 key-bits. The logic lies at that with longer keys, the more secure the cipher is. There are no hardware implementation examples using the SEED-192 or SEED-256. Given that the new SEED ciphers are similar with just different key-lengths, the hardware examples can be referred to: Yi et al. (2009) and Young-Ho et al. (2000).

4.33. Serpent

Serpent was one of the finalists in the AES contest. Serpent (Anderson et al., 2000) was engineered to satisfy the requirements of the AES. It is mentioned that Serpent is faster than DES and would be secure as Triple DES. In the same paper, the authors mentioned that Serpent was suited for smart card applications. On the implementation area, we refer to the work by Elbirt and Paar (2000) with the smallest design on an XCV100 FPGA. The reported hardware utilization is 5511 CLB slices.

4.34. Shark

SHARK (Rijmen et al., 1996) was considered as one of the predecessors of Rijndael. However, SHARK is considered unsecure after being discovered that it can be broken using an interpolation attack. Soon after this discovery, the SHARK’s successor, Rijndael was born. In the original paper by Vincent et al. the authors claimed that SHARK runs 4 times faster than SAFER and IDEA on a 64-bit architecture via C-implementation.

4.35. Skipjack

Skipjack (National Institute of Standards and Technology, 1998) is one of the few ciphers developed by the National Security Agency (NSA). It is based on an unbalanced Feistel network running at 32 rounds and 64-bit blocks. The implementation proposed by Kong et al. (2011) is considered the smallest Skipjack in hardware, claiming that only 76 slices of flip-flop were used.

4.36. Square

Proposed by Daemen et al. (1997), the SQUARE block cipher was first published in 1997 and has 128-bit block size and 128-bit key size. The SQUARE cipher led to development of the Rijndael Key Schedule and which was adopted by the AES. In the paper by Daemen et al. (1997), the author presented an implementation of the SQUARE cipher using Motorola’s M68HC05 microprocessor, occupying 547 bytes or ROM and 36 bytes of RAM, together with 7500 cycles for 1 execution of the SQUARE (including the key schedule).

4.37. Threefish

The THREEFISH cipher is a block cipher designed as a part of the SKEIN hash function. The description of the Threefish cipher can be found in the paper (Ferguson et al., 2008). For the hardware example of the Threefish cipher, the paper by At et al. (2013) has presented a Threefish hardware implementation using the Xilinx Virtex-6 (XC6VLX75T-2) FPGA with a result of 277 silces at 267 MHz. The forward encryption only circuit is 145 slices at 294 MHz.

4.38. Twofish

The TWOFISH block cipher (Schneier et al., 1999) is derived from Blowfish cipher. The Twofish cipher is one of the five finalists of the AES contest. This cipher is one of the few ciphers included in the OpenPGP standard. For practical hardware example, Chodowiec and Gaj (1999) has presented 2 versions of the Twofish implementation. The smallest Xilinx FPGA device mentioned by the authors that is able to implement the circuit is the Xilinx XC4028, with the maximum number of CLBs equal to 1024, and the equivalent number of logic gates equal to 28,000. The best result for the area optimized combinational design is at 888 CLBs and the sequential design is at 926 CLBs.

4.39. Summary of ciphers

Table 17 shows the collective information and specifications about the 38 selected modern symmetric key block ciphers.

5. Involution cipher

5.1. Anubis

According to the original literature (Barreto and Rijmen, 2001a), the ANUBIS is said to be much more scalable than most modern ciphers, in the sense of being very fast while avoiding excessive storage space (for both code and tables) and expensive or unusual instructions built in the processor; this makes it suitable for a wide variety of platforms. The same structure also favors extensively parallel execution of component mappings, and its mathematical simplicity tends to make analysis easier. Unfortunately, currently there are no clear figures on the performance, hardware area or gate count in ASIC or FPGA.

5.2. ARIA

ARIA is a 128-bit block cipher and was originally proposed in the year 2005 by a group of South Korean researcher from the Academia Research Institute and Agency, namely the ARIA. In the paper by Jinsub et al. (2006), the authors had presented their compact design of ARIA cipher. The report results are a total of 43,760 GE for the ARIA128 and 13,893 GE for the ARIA32.

5.3. ICEBERG

ICEBERG is also based on a fast involution structure. The original authors in Standaert et al. (2004) stated that the ICEBERG can be as secure as AES and other NESSIE candidates and also very efficient for reconfigurable hardware implementations. ICEBERG is said to offer free opportunities to defeat most side-channel attacks by using adequate encryption modes. In the same paper, the authors have presented HW costs with a total of 704 LUTs.

5.4. KHAZAD

For KHAZAD (Barreto and Rijmen, 2001b), it is also similar to ANUBIS and it is said to be much more scalable than most modern ciphers. This also makes it very suitable for a wide variety of platforms. The same structure also favors extensively parallel execution of component mappings, and its mathematical simplicity tends to make analysis easier. And KHAZAD currently has no clear figures on the performance, hardware area or gate count in ASIC or FPGA.

5.5. PP-1

The scalable PP-1 (Bucholc et al., 2010) cipher is described to be a simple, efficient and secure involution block cipher. The cipher is mentioned to be used on resource constrained platform, especially with a very limited amount of memory. Due to the fact that PP-1 uses only very simple arithmetic operations, the cipher can be implemented on other platforms, such as smart cards, TV decoders and even mobiles. The original authors claimed that they could not find any significant constraint or any hidden weaknesses.

5.6. PUFFIN

The PUFFIN cipher (Huiju et al., 2008) is also based on an involution SPN structure. The identical path for encryption and decryption with a simple key schedule makes it capable of on-the-fly sub-key generation. It is mentioned that the ASIC implementation based on a 0.18 μm CMOS standard cell design requires only 2600 gates with a throughput up to 700 Mbps. In this same paper, it is also mentioned that it is proposed to be implemented in smart cards and RFID tags because comparisons were made and closely matched to those in their respective area.

5.7. Summary of ciphers

Table 18 shows the collective information about the 6 selected involution ciphers.

6. Lightweight block ciphers

In this section, we are reviewing the known lightweight block ciphers. This section will discuss about the background and implementation details which impacts the current literature on compact lightweight block cipher designs for RCEs.

6.1. CLEFIA

CLEFIA (Akishita and Hiwatari, 2012) by SONY Corporation is a highly secure and efficient block cipher algorithm that delivers advanced copyright protection and authentication. CLEFIA is said to be able to perform even in restrictive environments such as smart cards and mobile devices. The reported hardware implementation results are a total of 2604 GE for the Type-2 CLEFIA, using 0.13 μm CMOS ASIC library.

6.2. DESL

DESL (DES Lightweight extension) is based on the classical DES. In the paper by Poschmann et al. (2007), the DESL has a single S-Box being used 8-times repeatedly. Other than that, the DESL is optimized to resist common attacks such as the linear and differential cryptanalysis. For the implementation results, the author has presented a fair amount of GE used with 1848 GE with 1 GE equivalent to 4 transistors.

Table 17
Specifications of the 38 selected modern symmetric key block ciphers.

Modern block ciphers	Year	Designer	Network type	Block size	Key size (bits)	Rounds	Application/design environment
3DES (Triple DES, TDEA)	1998	ANS X9.52	FN	64	56, 11, 168	48	Electronic payment systems, Microsoft Outlook
3-Way	1994	Joan Daemon	SPN	96	96	11	Various hardware platforms from 8-bit processors to specialized hardware
A5/3 KASUMI	2000	Mitsubishi	FN	64	128	8	GSM Handsets, 3GPP GSM wireless networks.
AES (Advanced Encryption Standard, Rijndael)	1998	Vincent Rijmen, Joan Daemen	SPN	128	128, 192, 256	10, 12, 14	Archive and compression tools, Disk encryption, Security for communications in Local Area Networks, INTEL and AMD processors
Blowfish	1993	Bruce Schneier	FN	64	32–448	16	Computers
Camellia	2000	Mitsubishi, NTT	FN	128	128, 192, 256	18, 24	Low-cost smart cards, high-speed networks
CAST-128	1996	Carlisle Adams, Stafford Tavares	FN	64	40–128	12, 16	Pretty Good Privacy (PGP)
CAST-256	1998	Carlisle Adams, Stafford Tavares, Howard heys, Michael Wiener	Generalized FN	128	128, 160, 192, 224, 256	48	Pretty Good Privacy (PGP)
CRYPTON	1998	Chae Hoon Lim	SPN	128	128, 192, 256	12	ATM, high throughput systems, high speed networks
CURUPIRA-1	2007	Paulo S.L.M. Barreto, Marcos A. Simpl´ıcio Jr.	SPN	96	96, 144, 192	10, 11, 14, 17, 18, 23	Sensor and mobile networks or systems heavily dependent on tokens or smart cards.
CURUPIRA-2	2008	Marcos A. Simpl´ıcio Jr., Paulo S.L.M. Barreto, Tereza C.M.B. Carvalho, Cintia B. Margi, and Mats N□aslund	SPN	96	96, 144, 192	10, 11, 14, 17, 18, 23	Sensor networks.
DEAL	1998	Lars Knudsen	Nested FN	128	128. 192, 256	6, 8	Can be used in DES-based hardware and software environment.
DES (Data Encryption Standard)	1979	IBM	Balance FN	64	56	16	Webmail/email encryption, Online banking, Electronic funds transfers, Internet-based VoIP, Wireless LAN, Personal communications (e.g., secure instant messaging), Laptop hard-drive encryption
FEAL-4	1987	Akihiro Shimizu and Shoji Miyaguchi (NTT)	FN	64	64	32	Designed as software for 8-bit microprocessors in IC cards
FEAL-N/NX	1990	Akihiro Shimizu and Shoji Miyaguchi (NTT)	FN	64	128	32	Designed as software for 8-bit microprocessors in IC cards
GOST	1994	USSR	FN	64	256	32	Software and low-area hardware, passive RFID tags.
ICE	1997	Matthew Kwan	FN	64	64*n*	8, 16*n*	Software, compatible interface with DES.
IDEA (International Data Encryption Algorithm)	1991	Xuejia Lai, James Massey	Add-Rotate-XOR	64	128	8.5	Pretty Good Privacy (PGP)
Khafre	1989	Ralph Merkle	FN	64	512	16++	–
Khufu	1989	Ralph Merkle	FN	64	512	16	–
LOKI97	1998	Lawrie Brown, Jennifer Seberry, Josef Pieprzyk	FN	128	128, 192, 256	16	8-bit processors, ATM, HDTV, B-ISDN, voice applications, Satellite applications.
Lucifer	1984	Sorkin, IBM	FN	64	56	16 Feistel rounds	Electronic banking
MAGENTA	1998	Michael Jacobson Jr., Klaus Huber	FN	128	128, 192, 256	6 or 8	Suitable for applications in ATM, HDTV, B-ISDN, voice and satellite applications.
MARS	1999	Carolynn Burwick, Don Coppersmith, Edward D'Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas Jr., Luke O'Connor, Mohammad Peyravian, David Safford, Nevenko Zunicof	FN	128	128–448	16	–
MISTY-1	1995	Matsui	Nested FN	64	128	8	Software stored in IC cards, hardware used in fast ATM networks, S/MIME-based e-mail application, Government services, Secure Web Access System, HTTP and TCPs.
MISTY-2	1995	Matsui	Nested FN	64	128	12	Software stored in IC cards, hardware used in fast ATM networks, S/MIME-based e-mail application, Government services, Secure Web Access System, HTTP and TCPs.
RAINBOW	1997	Chang-Hyi Lee, Jeong-Soo Kim (Samsung Advanced Institute of Technology)	SPN	128	128–256	7	ATM, HDTV, B-ISDN, Satellite and such as Smart Cards using 8-bit processor.
RC2	1987	Ron Rivest	Source-heavy FN	64	8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112. 120, 128	16 mix+2 mash	–
RC5	1994	Ron Rivest	Feistel-like network	32, 64, 128	0–2040	1–255	Any application.
RC6	1998	Ron Rivest, Matt Robshaw, Ray Sidney, Yiqun Lisa Yin	FN	128	128, 192, 256	20	Any application

Table 17 (*continued*)

Modern block ciphers	Year	Designer	Network type	Block size	Key size (bits)	Rounds	Application/design environment
SEED 128	1998	KISA	Nested FN	128	128	16	Mobile devices and smart cards
SEED 192/256	1998	KISA	Nested FN	128	128	16	Mobile devices and smart cards
Serpent	1998	Ross Anderson, Eli Biham, Lars Knudsen	SPN	128	128, 192, 256	32	Low cost smartcard applications,
SHARK	1996	Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bossalaers, Erik De Win	SPN	64	128	6	Software based
Skipjack	1998	NSA	Unbalanced FN	64	80	32	Secured phones
SQUARE	1997	Joan Daemen, Vincent Rijmen	SPN	128	128	8	–
Threefish	2008	Bruce Schneier, Niels Ferguson, Stefan Lucks, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, Jesse Walker	Add-Rotate-XOR	256, 512, 1024	256, 512, 1024	72, 80	–
Twofish	1998	Bruce Schneier	FN	128	128, 192, 256	16	OpenPGP

Table 18
Specifications of the 6 selected involution ciphers.

Involution cipher	Year	Designer	Network type	Block size	Key size (bits)	Rounds	Application/design environment
Anubis	2000	Vincent Rijmen, Paulo S.L.M. Barreto	SPN	128	128, 169, 192, 224, 256, 288, 320	12, 13, 14, 15, 16, 17, 18	FPGA (involution architecture suggests hardware implementation will be efficient)
ARIA	2003	Academia Research Institute and Agency, South Korea	SPN	128	128, 192, 256	12, 14, 16	IPsec, Secure Real-time Transport Protocol
Khazad	2000	Vincent Rijmen, Paulo S.L.M. Barreto	SPN	64	128	8	FPGA (involution architecture suggests hardware implementation will be efficient)
ICEBERG	2004	Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, Jean-Didier Legat	SPN	64	128	16	Reconfigurable systems, FPGA
PUFFIN	2008	Huiju Cheng, Howard M. Heys, Cheng Wang	SPN	64	128	32	Embedded digital systems, smart cards and RFID tags
PP-1	2010	Krzysztofbucholc, Krzysztofchmiel, Annagrocholewska-Czuryło, Ewaidzikowska, Izabelajanicka-Lipska, Januszstokłosa	SPN	$N=t*64$, where $t=1, 2, 3\ldots$ (64, 128, 192, 256)	N, $2N$	11, 22, 32, 43	Resource constrained environment, smart cards, TV decoders, mobiles.

6.3. *DESX*

DESX is a variant of DES. DESX has increased strength unlike the predecessor DES. The difference is that the DESX had included a technique called the key-whitening technique. The intention is to increase the key size of DES without substantially altering the DES algorithm. The paper by Poschmann et al. (2007) mentioned that by using the linear cryptanalysis at a clock speed of 500 kHz, it will take more than 80 years to crack DESX. In the same paper, the implementation result shows the DESX design occupies 2629GE.

6.4. *DESXL*

The DESXL (Leander et al., 2007) is basically a combination of DESX and DESL. The DESXL is usually viewed as a modified version of DES due to design costs. It is considered a sounder option to modify a well-studied and established algorithm than completely use a new cipher, which is more costly in many cases. It is also reported that the implementation results for the DESXL is 2169 GE, which outperforms the DESX in terms of hardware occupancy.

6.5. *DST40*

DST40 (Bono et al., 2005) is also known as the digital signature transponder. The reason that we mention the DST within the lightweight cipher section is because the DST is used in the various wireless authentication applications in the form of RFID device. But unfortunately, the DST40 manufacture by the Texas Instrument is cracked by Bono et al. (2005), using an array of 16 FPGAs to perform brute-force attack in less than an hour. This vulnerability has affected SpeedPass payment transponders and automobile ignition keys. This is then fixed by using a metallic shield to prevent unauthorized scanning of DST tags.

6.6. *EPCBC*

EPCBC (Yap et al., 2011) stands for electronic product code (EPC) block cipher and is said to be suitable for the application for electronic product code encryption. The EPC uses low-cost passive RFID tags and also a 96-bit (12 bytes) unique identifier, which the EPCBC aims for the cipher block size. The EPCBC is found to be secure against related-key differential or boomerang attacks due to the reason of its optimized key schedule. The EPCBC-96 occupies 1333 GE area and the EPCBC-48 occupies 1008 GE in VHDL based on the UMC L180 0.18 μm.

6.7. *HIGHT*

HIGHT (Hong et al., 2006) is proposed to provide low-resource environment such as the RFID tags or the ubiquitous sensor network (USN). The presented implementation requires 3048 gates using 34 clocks. The throughput achieved is at 150.6 Mbps, running at an 80 MHz frequency. The security strength is described to be abundant and an all-pass NIST statistical test results were presented in the same paper for added confidence.

6.8. KATAN

KATAN (Canni et al., 2009) is motivated by the need of a compact design with a minimal number of gates, suited for RFID and resource constrained environments. KATAN is a 524 rounds cipher with 32, 38 and 64 block sizes. The KATAN cipher is expected to be secure against the related-key differential attacks, algebraic attacks, differential and linear attacks, combined attacks related-key attacks and slide attacks. The implementation results showed the smallest design size would be the KATAN32 with a total of 802GE.

6.9. KEELOQ

KEELOQ was originally created by Professor Gideon Kuhn and used in many remote keyless entry systems by companies like: Toyota, Volvo, Honda, Fiat and Jaguar. The specification of Keeloq is it accepts 64-bit keys and encrypts 32-bit block, by executing its single-bit non-linear feedback shift register (NLFSR). It is mentioned that the Keeloq itself is more vulnerable to a replay attack by jamming the channel while intercepting the codes. Keeloq is also vulnerable to brute-force attack and poses threats to all known car, building access control and systems that uses Keeloq. This cipher is worth mentioning as Microchip (Marneweck, 2011) had made it clear that the Keeloq is secure provided that the implementation conditions are met. There is no known data on the implementation results.

6.10. KLEIN

KLEIN (Gong et al., 2012) is mentioned to have a better advantage in the software performance on legacy sensor platforms, originally designed for WSNs and RFID tags. It provides moderate security and was tested against various cryptanalysis, which most lightweight ciphers are weak against. In the paper, the authors have pointed out that in hardware implementations for RFID, a complete system (including the analog systems) would have a total of between 1000 and 10,000 GE occupied. For the security circuits, it should be about 2000 GE of occupancy.

6.11. KTANTAN

KTANTAN (Canni et al., 2009) has a lot of common properties with KATAN. Similar with KATAN, the KTANTAN is also suited for resource constrained environments. The only difference between KATAN and KTANTAN is the key schedule. The KATAN family has an 80-bit key loaded into a register which is then repeatedly clocked. While in the KTANTAN family, the key is “hard-coded” or “burnt” with the only possible flexibility of the choices for sub-key bits. The KTANTAN is as secure as KATAN and the smallest implementation results reported is 462GE for KTANTAN32.

6.12. LBlock

The paper that describes the LBlock cipher (Wu and Zhang, 2011) had claimed that the LBlock cipher requires 1320 GE for hardware implementation and is secure against known attacks. This satisfies the regular limitation of 2000 GE in RFID applications. On the other hand, with additional RAM, an area-optimized LBlock implementation would only require 866.3 GE. Furthermore, the original authors mentioned that the design goal for LBlock is to provide cryptographic measures for RCEs such as the RFID tags and sensor networks.

6.13. Light Encryption Device (LED)

The Light Encryption (Guo et al., 2011) Device (LED) is an AES-like cipher. This design of the LED cipher is aimed at very compact hardware implementation while maintaining software-friendly properties. The standard cell library UMCL18G212T3 based on the UMC L180 0.18 μm was used for the simulation. The result for flexible key implementation is 966 GE for LED64, 1040 GE for LED80, 1116 GE for LED96 and 1265 GE for LED128. On the other hand, the result for hard-wired keys (without key-update) is 688 GE for LED64, 690 GE for LED80, 695 GE for LED96 and 700 GE for LED128.

6.14. Miniature CRYPTON (mCRYPTON)

mCRYPTON (Lim and Korkishko, 2006) is also known as the miniature CRYPTON from its predecessor: CRYPTON. In the original paper, it is reported that the prototype implementation requires around 3.5 kGE to 4.1 kGE for both encryption and decryption. In another paper (Plos et al., 2012), the authors claim that the most compact mCRYPTON implementation requires 2709 GE using a 130 nm CMOS process technology from Faraday.

6.15. MIBS

MIBS (Izadi et al., 2009) is a Feistel network cipher with block size of 64-bits. In the same paper, the hardware implementation result is reported to be 1400 GE, which is lower than the 2000 GE requirement for RFID security. The implementation was done in a standard cell library based on TSMC 018 μm CMOS technology. In the same paper, the authors showed that MIBS is secure against differential and linear cryptanalysis.

6.16. NOEKEON

NOEKEON (Daemen et al., 2000) is described to be resistant against cryptanalysis of known attacks. Compared to other ciphers, NOEKEON has compact code size and runs efficiently on various platforms. Other advantages worth mentioning are NOEKEON is ultra-compact and fast in dedicated hardware implementation and needs very low RAM requirements in software implementation. In the paper, it is also mentioned that the cipher is suited to be implemented on 32-bit processors and also 8-bit processors, suitable for RCEs. However, in the same paper, no hardware implementation results were mentioned. The original authors did present the code size of 332 bytes for NOEKEON encryption and decryption sequences, with 712 cycles at a bit rate of 5.1 Mbit/s.

6.17. Piccolo

The authors for Piccolo cipher (Shibutani et al., 2011) claims that their cipher is capable of offering security against the meet-in-the-middle attacks (MITM) and the related-key differential attacks (RKA). They offer implementation requirements of 683 GE and 758 GE for the 80 and 128-bit key mode respectively. They also claimed that the Piccolo has achieved the best performance with respect to energy consumption and is among one of the most compact lightweight ciphers known.

6.18. PRESENT

PRESENT (Bogdanov et al., 2007) is described to be “ULTRA” lightweight and is a substitute for the AES in resource constrained environments. With an 80-bit key, it is said to be more than enough for most low-security applications. It is also reported that the implementation results can achieve as low as 1570 GE, which is lower than the 2000 GE for RFID.

6.19. PRINCE

The PRINCE cipher (Borghoff et al., 2012) is optimized with respect to latency when implemented in hardware. The authors claimed that the PRINCE allows data encryption within one clock cycle with competitive implementation results to known solutions. The Cadence NCVVerilog 06.20-p001 is used in the implementation simulation process. The PRINCE ciphers scored 8260GE using the 45 nm generic NANGATE open cell library, 7996 GE using the 90 nm Faraday library from UMC and 8679 GE using the 130 nm Faraday library from UMC. The authors had made comparisons with PRESENT, LED and AES with scores of 63,942 GE, 109,811 GE, and 135,051 GE.

6.20. PRINTcipher

The keyword “IC-printing” was mentioned when the authors described the PRINTcipher (Knudsen et al., 2010). The PRINTcipher is the product of the infant technological advancement for IC-printing; using silicon inks for high-definition printing process. IC-printing is used in the fabrication of cheap RFID tags so this explains the motivation for adding some simple security functionality. In the same paper, the PRINTcipher has shown implementation results of 402 GE and 726 GE respectively for PRINTcipher-48 and PRINTcipher-96 in serial encryption.

6.21. Scalable Encryption Algorithm (SEA)

The Scalable Encryption Algorithm (SEA) (Standaert et al., 2006) poses as a suitable solution for low-cost encryption, RFID environment and any power or space-limited applications. The cipher is made mostly targeting small embedded applications, secure against linear and differential attacks. The implementation was made on an ARM-based 32-bit RISC processor and Atmel ATtiny 8-bit processor, aiming for processors with limited instruction set like: AND, OR, XOR, Rotate and Modular-Add. The authors claimed that the SEA cipher is simple and an implementation can be done within a few hours via assembly coding.

6.22. SIMON

SIMON (Beaulieu et al., 2012) is a cipher proposal by the NSA. There have been no decisions made regarding the public release of SIMON. For this reason, the original paper shows performance data but does not describe the algorithms themselves. SIMON is worth mentioning because from the results, we can observe that the smallest design (SIMON 48/96) scored 782 GE, which is competitive with other lightweight ciphers.

6.23. SPECK

SPECK (Beaulieu et al., 2012) is a cipher proposal by the NSA. There have been no decisions made regarding the public release of SPECK. For this reason, the original paper shows performance data but does not describe the algorithms themselves. From the performance paper, we can observe that the smallest design (SPECK 48/96) scored 882 GE, which is competitive with other lightweight ciphers as well.

6.24. Tiny Encryption Algorithm (TEA)

The tiny encryption algorithm (TEA) (Needham, 1995) was created by Wheeler and Needham with the motivation for a tiny but fast and secure cryptographic algorithm. This algorithm is worth mentioning because soon after the TEA was introduced, the XTEA and the XXTEA is proposed for added security and implementation efficiency.

6.25. TWINE

The design rational for TWINE (Tomoyasu Suzaki, 2011) is to focus on the mixed environments of hardware and software resource constrained environment. The cryptanalysis results show that a full-round TWINE is security sufficient. In the same paper, the author had made comparison to a wider list of lightweight ciphers. The author did mentioned in this paper saying that they have implemented TWINE on both hardware and software (ASIC, FPGA and 8-bit MCU), claiming that the results are around 1500GE.

6.26. TWIS

TWIS (Ojha et al., 2009) is inspired by CLEFIA and it is much more resource efficient and cryptographically on par with CLEFIA. However, there is a literature by Li (2010) pointing out that the TWIS cipher has weaknesses and vulnerability. However, there is no known implementation data.

6.27. Xtended TEA (XTEAe)

XTEA is the first version of the block TEA's successor and it is designed to correct the weaknesses in smaller rounds of TEA. Presented along with XTEA was a variable-width block cipher named Block TEA. However, block TEA is proven weak and XXTEA was proposed to rectify the problem. The paper by Kaps (2008) has presented a high speed implementation of XTEA operating at 20.6 Gbps on FPGA and 36.6 Gbps on ASIC. However, XTEA is vulnerable to related key differential attack and related key rectangle attack (Lu, 2009; Ko et al., 2004).

6.28. XXTEA (corrected block tiny encryption algorithm)

Corrected block TEA is often referred to as XXTEA is a cipher designed to correct the weaknesses in the original cipher: Block TEA. However, there is not much information regarding the hardware implementation results for this cipher and the XXTEA is vulnerable to chosen plaintext attack (Yarrkov, 2010).

6.29. Summary of ciphers

Table 19 shows the collective information about the 28 selected lightweight block ciphers.

7. Stream ciphers

In this section, we are reviewing the known stream ciphers, which is also a type of symmetric ciphers. There are some literatures that mentioned the term “lightweight stream ciphers” specifically for the type of stream ciphers used in RCEs such as the RFID environment. This section will discuss about the background and implementation details which impacts the current literature on stream cipher designs for RCEs. Note that only selected stream ciphers are chosen to be mentioned in this paper.

7.1. A2U2

The A2U2 stream cipher was designed by David et al. (2011) with the purpose of tackling RFID security challenges. The authors claimed that their design uses only 284 GE. This is very promising because it is very low compared to the 2 kGE mark. In the same paper, the work by Hamalainen et al. (2006) is shown at the 3100 GE mark as comparison. The authors mentioned the method they used for area optimization is through processing of shorter word lengths, repeated use of hardware (i.e. S-box), increasing the number of rounds before the ciphertext is generated to maintain security at the compromise of speed, using shorter keys, block sizes, feedback functions, register lengths and permutation-substitution boxes.

7.2. A5/1

A5/1 stream cipher was used in Global System for Mobile Communications (GSM). One of the hardware results found (Batina et al., 2004) for the A5/1 were implemented with only 64 slice flip-flops and the total equivalent gate count is 932 gates using Xilinx Virtex-HQ800 FPGA. On the other hand, the hardware results by Galanis et al. (2005) has shown only 32 slices used on a Virtex-II 2V250FG256 FPGA. However, it is claimed that this cipher is broken (Biryukov et al., 2001).

7.3. A5/2

A5/2 stream cipher is the variant of A5 stream cipher. It is claimed that A5/1 is stronger than A5/2 and the A5/2 is a deliberate weakening of the A5 algorithm for export. However, the A5/2 is considered weak and insecure and there are no hardware examples of this stream cipher.

7.4. AES-CTR

The AES-CTR is also known as the AES Counter mode. In this mode, the AES operates like a stream cipher. In some literatures, it is known as the AES CM. In terms of hardware implementation examples, one of the work by Good and Benaissa (2006) has implemented the AES-CTR using ASIP design on a Xilinx XC2s15 FPGA, amounting to 120 slices.

7.5. Edon80

The Edon80 stream cipher is work by Gligoroski et al. (2008). There are not many hardware implementation examples but the one identified is the work by Kasper et al. (2006). Kasper et al. had shown good results using the Edon80 stream cipher. By using Xilinx Spartan-2 XC2S15-6 FPGA, the Edon80 consumes only 52 CLBs running at 106 MHz and by using Xilinx Virtex-4 XC4VLX15-12, the Edon80 consumes only 45 CLBs running at 286 MHz. On top of that, the authors had also presented their version of CMOS ASIC implementation, amounting to approximately 2922 GE.

7.6. GRAIN

The GRAIN stream cipher by Hell et al. (2007) is a stream cipher designed for constrained hardware environments. In the same paper, Hell et al. has presented some hardware results using Altera FPGA which amounts to 4008 gates. On the other hand, in the paper by Jafarpour et al. (2011), the authors had implemented GRAIN into a Spartan-3 FPGA, yielding result of 227 slice flip-flop and 188 4-input LUTs utilized.

7.7. HC-128

The HC-128 stream cipher is designed by Wu (2008). The HC-128 stream cipher is the newer version and the simplified version of the HC stream cipher and the internal state is 128-bits. This cipher is worth mentioning in this paper because the author claims that this cipher has no weak keys, no hidden flaws and it is software efficient but no hardware implementation results are found.

7.8. HC-256

The HC-256 is the original HC stream cipher proposed by Wu (2004) during 2004. This cipher uses 256-bit keys and 256-bit initialization variables. Despite having HC-128 and HC-256 stream ciphers available to research community, there are no known hardware implementation efforts using this cipher. However, there is one interesting paper that uses the HC-256 stream cipher for image encryption, which can be found in the paper by Jolfaei et al. (2012).

7.9. Hermes8

Hermes-8 stream cipher is claimed to be low-complexity and low-power. The designer of this stream cipher (Kaiser, 2006) and in the same paper, Kaiser had proposed 2 models of Hermes8: an 80-bits and 128-bits version. In terms of hardware implementation, Kitsos and Kaiser (2007) has shown that their architecture is able to perform using Spartan-2 XC2S100-6 FPGA with 697 CLBs used and VIrtex-4 XC4VFX12-11 FPGA with 715 CLBs used. And secondly, the work by Good et al. (2006) has shown implementation results using Xilinx Spartan-2 XC2S30-5 FPGA with only 190 CLBs used.

7.10. LEX

LEX stream cipher is a simple stream cipher that uses AES. The original idea for this stream cipher is to extract parts of the internal state at certain rounds and outputs them. This can be applied to any block ciphers and the LEX mainly depends on the strength of the cipher's round function and the strength of the key schedule. The LEX stream cipher is designed by Biryukov (2005). However, there are no known hardware implementation models or examples for this stream cipher.

7.11. MICKEY-128

The MICKEY-128 stream cipher is a cipher by Babbage and Dodd (2008). MICKEY stands for Mutual Irregular Clocking Keystream Generator. In term of hardware example, only one particular paper is found regarding hardware implementation of the MICKEY-128 stream cipher, which is the paper by Kitsos (2005). The results presented was the implementation of MICKEY-128 using Xilinx XCV50ECS144 FPGA, with only 167 CLBs utilized, running at 170 MHz.

7.12. Phelix

Phelix is a stream cipher designed by Whiting et al. (2005) and it is also a high-speed stream cipher with a built-in MAC functionality. The author has presented analysis on Phelix on the Pentium CPU platform but the simulations were done in C code. As for hardware implementation literature, the work by Junjie and Heys (2007) has shown hardware implementation example using ASIC, with 12,400 2-input NAND gates used. And the paper by Good et al. (2006) has shown 4 sets of results: 1198 slices on XC2S100-5, 1077 slices on XC2S100-5, 264 slices on XC2S30-5 and 250 slices on XC2S30-5.

7.13. Polar Bear

The Polar Bear stream cipher (Nada and ŇAslund, 2005) is a stream cipher that uses Rijndael (AES) for key initialization and RC4 for steady state dynamic table. From the hardware implementation's point of view, Good et al. (2006) has noted in a paper that the Polar Bear stream cipher would be large because even with one round of AES, the large footprint has already been established. There are no known hardware implementation results at the moment.

Table 19
Specifications of the 28 selected lightweight block ciphers.

Lightweight block ciphers	Year	Designer	Network type	Block size	Key size (bits)	Rounds	Application/design environment
CLEFIA	2007	SONY	FN	128	128, 192, 256	18, 22, 26	Restrictive environments such as: smart cards and mobile devices, copyright protection and authentication.
DESL	2007	Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm	FN	64	56	16	Passive RFIDs, resource constrained environment
DESX	2007	Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm	FN	64	184	16	Passive RFIDs, resource constrained environment
DESXL	2007	Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm	FN	64	184	16	Passive RFIDs, resource constrained environment
DST40		Texas Instruments	Unbalanced-FN	40	40	200	Exxon-Mobil Speed-pass Payment System
EPCBC	2011	Huihui Yap, Khoongming Khoo, Axel Poschmann, Matt Henricksen	SPN	48, 96	96	32	Electronic Product Code (EPC)
HIGHT	2006	Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, Seongtaek Chee	Feistel-like network	64	128	32	RFID tags, USN (ubiquitous sensor network)
KATAN	2009	Christophe De Canni’ere, Orr Dunkelman, Miroslav Knězevi´c	NLFSR-based	32, 48, 64	80	254	RFID, resource constrained environment
KEELOQ	1985	Frederick Bruwer, Gideon Kuhn, Willem Smit	Unbalanced Feistel, NLFSR-based	32	64	528	Cars, garage, remote door opener
KLEIN	2010	Zheng Gong, Svetla Nikova, Yee Wei Law	SPN	64	54, 80, 96	12, 16, 20	WSN, RFID
KTANTAN	2009	Christophe De Canni’ere, Orr Dunkelman, Miroslav Knězevi´c	NLFSR-based	32, 48, 64	80	254	RFID, resource constrained environment
LBlock	2011	Wenling Wu, Lei Zhang	FN	64	80	32	WSN, RFID
LED (Light Encryption Device)	2011	Jian Guo, Thomas Peyrin, Axel Poschmann and Matt Robshaw	SPN	64	64–128	32, 48	RFID
mCRYPTON (miniature CRYPTON)	2006	Chae Hoon Lim, Tymur Korkishko	SPN	64	64, 96, 128	12	Low-cost RFID tags and sensors
MIBS	2009	Maryam Izadi, Babak Sadeghiyan, Seyed Saeed Sadeghian, Hossein Arabnezhad Khanooki	FN	64	64, 80	32	RFID tags and sensor networks
NOEKEON	2000	Joan Daemen, Michaël Peeters, Gilles Van Assche, Vincent Rijmen	SPN	128	128	16	Resource constrained environment, WSN
Piccolo	2011	Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, Taizo Shirai	Generalized FN	64	80, 128	25, 31	RFID tags and sensor nodes
PRESENT	2007	Orange Labs, Ruhr University Bochum and the Technical University of Denmark	SPN	64	80, 128	31	Resource constrained environment
PRINCE	2012	Julia Borghoff, Anne Canteaut, Tim Guneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Soren S. Thomsen, Tolga Yalcin	SPN	64	128	12	Pervasive computing
PRINTcipher	2010	Lars Knudsen, Gregor Leander, Axel Poschmann, Matthew J.B. Robshaw	SPN	48, 96	80, 160	48, 96	Printable Circuits
SEA (Scalable Encryption Algorithm)	2006	Francois-Xavier Standaert, Gilles Piret, Neil Gershenfeld, Jean-Jacques Quisquater	FN	48, 96, 144, 6b		$3n/4+2^*$ $(nb+[b/2])$	RFID, Smart Cards
SIMON	2012	Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, Louis Wingers	–	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192, 256	–	Low-power limited gate devices (such as RFID and similar devices)
SPECK	2012	Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, Louis Wingers	–	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192, 256	–	Low-power limited gate devices (such as RFID and similar devices)
TEA (Tiny Encryption Algorithm)	1994	Roger Needham, David Wheeler	FN	128	64	64 Feistel rounds	Microsoft’s Xbox Gaming Console, as Hash function
TWINE	2011	Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi	Generalized FN	64	80, 128	36	Resource constrained environment (hardware and software)
TWIS	2009	Shri Kant Ojha, Naveen Kumar, Kritika Jain, Sangeeta	Generalized FN	128	128	10	Resource constrained applications
XTEA (eXtended TEA)	1997	Roger Needham, David Wheeler	FN	128	64	64 Feistel rounds	FGPA, SIC, low-power RFID tags and WSN
XXTEA (corrected block tiny encryption algorithm)	1998	Roger Needham, David Wheeler	Unbalanced FN	Arbitrary, at least two words (64 bits)	128	Depends on the block size, 6–32 full cycles	Legacy hardware systems (embedded)

7.14. Rabbit

The Rabbit stream cipher was one of eStream Portfolio designed for software environment. The designers, Boesgaard et al. (2008) had claimed that no weaknesses are found for the Rabbit stream cipher in the original paper. To show the practically of this stream cipher, a hardware framework designed by Stefan (2010) has shown a version of resource efficient design and a direct implementation of the Rabbit using Xilinx Virtex-5 LXT FPGA with a result of 568 slices used.

7.15. RC4

RC4 stream cipher was originally designed by Ron Rivest during 1987 and the RC4 is known to be used in SSL, TLS and the wireless WEP. Despite being one of the oldest stream ciphers around, a hardware implementation example for RC4 is shown by Kitsos et al. (2003) has shown promising results of only 138 slices used on a Xilinx XC2V250 FPGA.

7.16. Salsa 20/12

The Salsa20 stream cipher designed by Bernstein (2008) comes in a variety of variants. In this paper, we are presenting one of the Salsa20 family which is the Salsa 20/12 stream cipher. Salsa 20/12 is recommended for applications where speed is the bigger concern and the confidence is not the main issue. There are a few implementation efforts found in the literature. The paper by Junjie and Heys (2007) presented a total of 194 CLB slices and 4 BRAMs used in a Xilinx 2V250fg256 FPGA. As for the most compact design on an ASIC platform, the implementation consumed 14,100 2-input NAND gates, which is consider as 14,100 GE. And on the other hand, the work by Jarosław (2013) has shown implementation of Salsa20 on Spartan-3 using 14,838 slices and on Spartan-6 using 5079 slices.

7.17. SFINKS

The SFINKS stream cipher is designed by (Braeken et al.) for hardware applications and offering medium-term security. The authors have implemented the SFINKS stream cipher into a Xilinx Spartan XC3S5000 FPGA at 172 MHz. In the same paper, the authors have claimed to produce more results by producing other versions of the design for achieving various trade-offs between gate count and encryption speed.

7.18. SNOW3G

The SNOW 3G stream cipher is the stream cipher used for the 3GPP encryption algorithms, the UEA2 and UIA2. The SNOW stream ciphers (the previous versions are the SNOW v1 and SNOW v2) were designed by Force (2006). As for hardware examples, the work by Kitsos et al. (2008a) has used ASIC as the implementation platform. However, the hardware footprint was not mentioned but the throughput achieved is 7968 Mbps. On the other hand, Helion Technology (Ltd, 2009) has presented their version of SNOW 3G using Virtex-5 with a good result of 164 slices used. And lastly, the paper by Lingchen et al. (2012) has provided additional information on the hardware footprint. The authors claimed that they have “optimized” the previous SNOW 3G works. By using Virtex-5, the results are 188 slices and 356 slices for the 2 versions of SNOW 3G implementations that they had designed.

7.19. SNOW v1

The first version of SNOW stream cipher was released during 2001 (Ekdahl and Johansson, 2001). The author’s claimed that the SNOW stream cipher is suitable for telecommunication protocols. In the same paper, the authors presented a C implementation of the SNOW stream cipher but no hardware implementation is known at the moment.

7.20. SNOW v2

The SNOW v2 stream cipher is actually one of the two stream ciphers chosen for the ISO/IEC standard. The paper by Ekdahl and Johansson (2003) was released during 2003 and claimed that the SNOW v2 is more secure and operates faster in software as compared to SNOW v1. In the same paper, the authors presented a C implementation of the SNOW stream cipher but no hardware implementation known at the moment.

7.21. SOSEMANUK

The SOSEMANUK stream cipher by Berbain et al. (2008) uses from basic designs from SNOW v2 and some transformations from SERPENT cipher. The name “SOSEMANUK” means “snow snake” in Cree Indian Language because of its fundamental designs depending on SNOW and SERPENT. The SOSEMANUK is also one of the ciphers selected for the eSTREAM Portfolio. The author has shown some of their effort in implementing the SOSEMANUK into CISC and RISC. However, there are no known hardware implementation literatures regarding this cipher.

7.22. TinyStream

The TinyStream is a steam cipher designed by Tieming et al. (2010) and the authors claimed that the TinyStream is designed for communication security applications in resource constrained environments such as the WSN. In the same paper, the authors have claimed that their implementation of the TinyStream on the TinyOS has consumed 65,024 bytes of ROM and 1,659,184 bytes of RAM.

7.23. Trivium

Trivium is a hardware oriented synchronous stream cipher and it is claimed to be able to provide a flexible trade-off between speed and area (De Cannière, 2006). One paper from the literature has shown that the Trivium cipher can be implemented in FPGA and AVR chips. The work by Jafarpour et al. (2011) has shown 456 slice flip-flops and 311 number of 4 input LUTs used, using a Spartan 3 FPGA. Another paper by Mora-Gutiérrez et al. (2013) has shown an implementation example featuring CMOS technology (simulated with Modelsim and synthesized with Synopsys). The authors have shown that the power consumption can be reduced by more than 20% with only a slight penalty in terms of area and operation frequency.

7.24. VEST

The VEST stream cipher is designed to be hardware-friendly according to O’neil et al. (2005). The authors claimed that the VEST stream cipher is also suitable for low-cost hardware environment such as the RFID system, making it suitable for resource constrained environment. There are a few known variations of the VEST cipher that accepts variable key-size and initialization variable, namely the VEST-4, VEST-8 VEST-16 and VEST-32. The authors had shown a few results using FPGA and ASICs. The VEST-4 stream cipher was designed to achieve at least 80-bit security while occupying an area of approximately 4K gates in ASIC or approximately. The other example would be the VEST-32 stream cipher that was designed to achieve at least 256-bit, occupying an area of around 20K gates in

ASIC or around 2K logic cells and 1K register cells in the Altera Stratix-II FPGA.

7.25. WG-7

The WG-7 (Welch-Gong) stream cipher was first introduced in 2010 by Yiyuan et al. (2010). The key size of the WG-7 is 80 bits and the initialization variable is set to 81 bits. The authors have implemented WG-7 on a 4-bit and 8-bit microcontroller. However, this is the only literature found that offers implementation of WG-7 on MARC4 ATAM893-D using 1097 lines of code and on ATmega8 using 1100 bytes of flash memory.

7.26. WG-8

WG-8 is a stream cipher proposed by Fan et al. (2013a). The authors mentioned that the recent work has demonstrated the tower field constructions for finite field arithmetic in the AES S-box and also the WG-16 stream ciphers (Fan and Gong, 2013). The work by Yang et al. (2013) experimented with three different tower field constructions for WG-8 and analyzed their hardware results. For LUT approach with a CMOS 65 nm, the smallest design amounts to 3942GE.

7.27. WG-16

The WG-16 stream cipher is designed by Fan and Gong (2013) in an effort to address better security confidence from its predecessors. This stream cipher is fairly new and not many hardware example of this cipher is available at the moment. However, the same authors have produced some results to support their findings on this new WG-16 stream cipher. For FPGA platform, the smallest design amounts to 398 slices. Another paper from Fan et al. (2013b) showed results with 478 slices using FPGA and 12,031 GE using 65 nm ASIC.

7.28. ZUC

ZUC stream cipher is the core of the 3GPP (3rd Generation Partnership Project) confidentiality algorithm 128-EIA3 and the 3GPP confidentiality algorithm 128-EEA3. These algorithms are used as security services in LTE networks enabling high demands

Table 20
Specifications of the 28 selected stream ciphers.

Stream Ciphers	Year	Designer	Key length	Initialization variable (IV)	Application/design environment
A2U2	2011	Mathieu David, Damith C. Ranasinghe, Torben Larsen	56	5	Proposed for RFID environment, Printed electronic RFID tags
A5/1	1989	–	54	22	Global System for Mobile Communications (GSM)
A5/2	1989	–	54	114	Global System for Mobile Communications (GSM)
AES-CTR	2010	Niels Ferguson and Bruce Schneier	128, 192, 256	128, 192, 256	–
Edon80	2008	Gligoroski, Danilo Markovski, Smile Knapskog, SveinJohan	80	80	eSTREAM Profile: Hardware and resource constrained environment
GRAIN	2007	Martin Hell, Thomas Johansson and Willi Meier	80	64	Resource constrained environment, RFID, WSN, smart cards
HC-128	2008	Hongjun Wu	128	128	Proposed for efficient software application
HC-256	2004	Hongjun Wu	256	256	Proposed for efficient software application
Hermes8	2005	Ulrich Kaiser	128	296	Suited for low-complexity and low-power systems, suited for 8-bit micro-computers, dedicated hardware, embedded system
LEX	2005	Alex Biryukov	128	128	–
MICKEY-128	2008	Steve Babbage, Matthew Dodd	80	80	Proposed for resource constrained hardware
Phelix	2004	Doug Whiting, Bruce Schneier, Stefan Lucks, and Frederic Muller	128	128	Proposed for efficient software and hardware implementation
Polar Bear	2005	Johan Håstad and Mats Näslund	128	248	Proposed for efficient software implementation
Rabbit	2003	Martin Boesgaard, Mette Vesterager, Thomas Christensen, Erik Zenner	128	64	eSTREAM Profile: Software Environment
RC4	1987	Ron Rivest	40–2048	–	SSL, TLS, wireless WEP
Salsa20/12	2008	Daniel J. Bernstein	128	64	Typical encryption application, Fast encryption applications (lower security confidence)
SFINKS	2005	An Braeken, Joseph Lano, Nele Mentens, Bart Preneel and Ingrid Verbauwhede	80	80	Restricted Hardware Environments, Resource Constrained Environments.
SNOW 3G	2010	Thomas Johansson and Patrik Ekdahl	128	128	LTE (3GPP), chosen as the cipher for UEA2 and UIA2
SNOW v1	2001	Thomas Johansson and Patrik Ekdahl	128, 256	64	Telecommunication Protocols, Software
SNOW v2	2003	Thomas Johansson and Patrik Ekdahl	128	128	Telecommunication Protocols, Software
SOSEMANUK	2008	C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin and H. Sibert	128	128	Dedicated to software applications
TinyStream	2010	Tieming Chen, Liang Ge, Xiaohao Wang, Jiamei Cai	128	–	Resource Constrained Environment, WSN systems
Trivium	2006	Christophe De Canni‘ ere	80	80	High performance hardware application with limited resources
VEST	2005	O’Neil, S., Gittins, B., Landman, H.	Variable	Variale	Resource constrained environments, RFID Smart card
WG-7	2010	Y. Luo, Q. Chai, G. Gong, and X. Lai	80	81	Proposed for RFID environment, RFID tags
WG-8	2013	Xinxin Fan, Kalikinkar Mandal, and Guang Gong	80	80	Proposed for resource constrained smart devices, IOT
WG-16	2013	Xinxin Fan and Guang Gong	128	128	4G (LTE) mobile networks
ZUC	2011	3GPP (3rd Generation Partnership Project)	128	128	4G (LTE) mobile networks

Table 21
A listing of selected lightweight block ciphers and their respective hardware results.

Block ciphers	Reference	Logic process (μm)	GE area (gate equivalent)	Throughput at 100 kHz (Kbps)	Latency (cycles per block)
AES	Moradi et al. (2011)	0.13	2400	56.64	226
AES128	Feldhofer et al. (2005)	0.35	3400	12.40	–
AES128	Feldhofer et al. (2005)	0.35	3400	12.4	–
AES128	Hamalainen et al. (2006)	0.13	3100	0.08	–
AES128	Satoh et al. (2001)	0.11	5400	–	–
CLEFIA (compact-approach)	Eisenbarth and Kumar (2007)	0.09	4993	355.56	36
CLEFIA Type-1 (ENC)	Akishita and Hiwatari (2012)	0.13	2678	73	176+128
CLEFIA Type-1 (ENC/DEC)	Akishita and Hiwatari (2012)	0.13	2781	73	176+128
CLEFIA Type-2 (ENC)	Akishita and Hiwatari (2012)	0.13	2594	67	192+128
CLEFIA Type-2 (ENC/DEC)	Akishita and Hiwatari (2012)	0.13	2678	67/70	(192/184) +128
CLEFIA Type-3 (ENC/DEC)	Akishita and Hiwatari (2012)	0.13	2604	39/40	(328/320) +224
CLEFIA Type-3(ENC)	Akishita and Hiwatari (2012)	0.13	2488	39	328+224
CURIPURA-1	Kitsos et al. (2012)	0.09	8334	960	10
CURIPURA-2	Kitsos et al. (2012)	0.0	7334	960	10
DES	Poschmann et al. (2007), Leander et al. (2007)	0.18	2309	5.55	144
DESL	Poschmann et al. (2007), Leander et al. (2007)	0.18	1848	5.55	144
DESL	Kitsos et al. (2012)	0.09	2762	400	16
DESX	Poschmann et al. (2007), Leander et al. (2007)	0.18	2629	–	144
DESXL	Poschmann et al. (2007), Leander et al. (2007)	0.18	2168	–	144
DESXL	Kitsos et al. (2012)	0.09	3082	400	16
EPCBC48	Yap et al. (2011)	0.18	1008	12.12	396
EPCBC96	Yap et al. (2011)	0.18	1333	12.12	792
HIGHT	Hong et al. (2006)	0.25	3048	188.20	–
HIGHT	Kitsos et al. (2012)	0.09	3901	200	32
KATAN32	Canni et al. (2009)	0.13	802	12.5	–
KATAN48	Canni et al. (2009)	0.13	927	18.8	–
KATAN64	Canni et al. (2009)	0.13	1054	25.1	–
KLEIN64	Gong et al. (2012)	0.18	1220	30.9	207
KLEIN80	Gong et al. (2012)	0.18	1478	23.62	271
KLEIN96	Gong et al. (2012)	0.18	1528	19.1	335
KTANTAN32	Canni et al. (2009)	0.13	462	12.5	–
KTANTAN48	Canni et al. (2009)	0.13	588	18.8	–
KTANTAN64	Canni et al. (2009)	0.13	688	25.1	–
LBlock	Wu and Zhang (2011)	0.18	1320	200	–
LED128	Guo et al. (2011)	0.18	1265	3.4	1872
LED128 (HW-key)	Guo et al. (2011)	0.18	700	3.42	1872
LED64	Guo et al. (2011)	0.18	966	5.1	1248
LED64 (HW-key)	Guo et al. (2011)	0.18	688	5.13	1280
LED80	Guo et al. (2011)	0.18	1040	3.4	1872
LED80 (HW-key)	Guo et al. (2011)	0.18	690	3.4	1872
LED96	Guo et al. (2011)	0.18	1116	3.4	1872
LED96 (HW-key)	Guo et al. (2011)	0.18	695	3.42	1872
mCRYPTON128	Lim and Korkishko (2006)	0.13	2949	492.3	–
mCRYPTON64	Lim and Korkishko (2006)	0.13	2420	492.3	–
mCRYPTON96	Lim and Korkishko (2006)	0.13	2681	492.30	–
MIBS64	Izadi et al. (2009)	0.18	1396	200	32
Piccolo (ENC) (round)	Shibutani et al. (2011)	0.13	1496 (incl. key register=360GE)	237	27
Piccolo (ENC+DEC) (round)	Shibutani et al. (2011)	0.13	1634 (incl. key register=360GE)	237	27
Piccolo (ENC) (serial)	Shibutani et al. (2011)	0.13	1043 (incl key register=360GE)	14.8	432
Piccolo (ENC+DEC) (serial)	Shibutani et al. (2011)	0.13	1103 (incl key register=360GE)	14.8	432
PRESENT	Kitsos et al. (2012)	0.09	1704	206.4	31
PRESENT128	Yap et al. (2011)	0.18	1339	12.12	528
PRESENT128	Poschmann (2009)	0.18	1391	11.45	559
PRESENT80	Bogdanov et al. (2007)	0.18	1570	200.00	–
PRESENT80	Rolfes et al. (2008)	0.18	1075	11.7	547
PRESENT80	Yap et al. (2011)	0.18	1030	12.4	516
PRESENT80	Rolfes et al. (2008)	0.35	1000	11.4	–
PRINTcipher48 (HW-key)	Knudsen et al. (2010)	0.18	402	6.25	768
PRINTcipher96 (HW-key)	Knudsen et al. (2010)	0.18	726	3.13	3072
PUFFIN	Kitsos et al., (2012)	0.09	2303	200	32
PUFFIN	Huiju et al. (2008)	0.18	2577	–	32
SEA	Mace et al. (2007)	0.13	3758	103	93
SIMON (128/128) (best)	Beaulieu et al. (2012)	0.13	1274	12.9	–
SIMON (128/128) (area-minimizing)	Beaulieu et al. (2012)	0.13	1218	3.2	–

Table 21 (*continued*)

Block ciphers	Reference	Logic process (μm)	GE area (gate equivalent)	Throughput at 100 kHz (Kbps)	Latency (cycles per block)
SIMON (32/64) (area-minimizing)	Beaulieu et al. (2012)	0.13	523	5.5	–
SIMON (48/96) (best)	Beaulieu et al. (2012)	0.13	782	15	–
SIMON (48/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	738	5.0	–
SIMON (64/128) (best)	Beaulieu et al. (2012)	0.13	1011	18.1	–
SIMON (64/128) (area-minimizing)	Beaulieu et al. (2012)	0.13	939	4.5	–
SIMON (64/96) (best)	Beaulieu et al. (2012)	0.13	854	19	–
SIMON (64/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	801	4.8	–
SIMON (96/96) (best)	Beaulieu et al. (2012)	0.13	985	12.5	–
SIMON (96/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	946	4.2	–
SPECK (128/128) (best)	Beaulieu et al. (2012)	0.13	1501	21.6	–
SPECK (128/128) (area-minimizing)	Beaulieu et al. (2012)	0.13	1292	2.7	–
SPECK (32/64) (area-minimizing)	Beaulieu et al. (2012)	0.13	582	4.6	–
SPECK (48/96) (best)	Beaulieu et al. (2012)	0.13	882	12.5	–
SPECK (48/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	794	4.2	–
SPECK (64/128) (best)	Beaulieu et al. (2012)	0.13	1128	14.2	–
SPECK (64/128) (area-minimizing)	Beaulieu et al. (2012)	0.13	997	3.6	–
SPECK (64/96) (best)	Beaulieu et al. (2012)	0.13	984	15.7	–
SPECK (64/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	860	3.9	–
SPECK (96/96) (best)	Beaulieu et al. (2012)	0.13	1126	13.8	–
SPECK (96/96) (area-minimizing)	Beaulieu et al. (2012)	0.13	1012	3.4	–
TWINE (64/128) (ENC)	Tomoyasu Suzaki (2011)	0.09	1866	178	36
TWINE (64/128) (ENC+DEC)	Tomoyasu Suzaki (2011)	0.09	2285	178	36
TWINE (64/80) (ENC)	Tomoyasu Suzaki (2011)	0.09	1503	178	36
TWINE (64/80) (ENC+DEC)	Tomoyasu Suzaki (2011)	0.09	1799	178	36
TWINE (64/80) (ENC)(serial)	Tomoyasu Suzaki (2011)	0.09	1116	11.8	540
XTEA	Kitsos et al. (2012)	0.09	3490	200	32
XTEA	Kaps (2008)	0.13	2521	–	32

of mobile TV, streaming and downloading. Currently there are a few known hardware implementation papers in the literature. The work by Kitsos et al. (2011) has shown implementation results of 385 slices using Xilinx Virtex-5 FPGA. On the other hand, the work by Liu et al. (2013) presented results of 328 slices using Virtex-6 and 350 slices using virtex-5. Other results include the work by Wang et al. (2011) with 356 slices, and finally the work by Lingchen et al. (2012) with 395 slices.

7.29. Summary of ciphers

Table 20 shows the collective information about the 28 selected stream ciphers.

8. Discussion, comparison and analysis of the symmetric ciphers

In this section, discussions and comparisons are made focusing on the lightweight block ciphers and stream ciphers. Most of these ciphers the designed and tailored for resource constrained application specifically. These ciphers are very recent and most of them are still active in service. Most lightweight block ciphers and stream ciphers are newer than traditional and modern block ciphers. Moreover, all the ciphers known to-date are bound to be broken once computational power outgrows the block key-length and more efficient techniques are found to crack the ciphers within a short period. Due to this sole reason, it is common practice to consider only the strongest and the most recent ciphers known. In this section, only the lightweight block ciphers (which were discussed in Section 6) and the stream ciphers (which were discussed in Section 7) are chosen for this discussion.

As discussed in Section 2.2, the hardware specification and requirements for RFID environment is confined to an amount of 200–2000 GE for security applications. To show that which current cryptographic proposal falls into the limitation of such a hardware budget, Table 21 has presented a large listing of known light weight cryptographic ciphers with their implementation specifications. Note that the AES is included in the list as it is notably the most popular cipher despite comparing it to the other lightweight cryptographic implementations. A recent paper by Moradi et al. (2011), showed an AES hardware occupancy of only 2400GE. This is the smallest known AES implementation in terms of GE. The smallest block cipher implementation found is the work by Knudsen et al. (2010) using the PRINTcipher48, showing occupancy of only 402 GE, and the second smallest block cipher implementation found is the work by Canni et al. (2009) with their KTANTAN cipher, showing an implementation result of 462 GE. For the highest throughput, the work by the work by Kitsos et al. (2012) using the CURIPURA-1 cipher. The authors reported a throughput at 960 kbps at 100 kHz. The lowest throughput reported is work by Hamalainen et al. (2006) using the AES cipher, at a throughput of 0.08 kbps. As for the circuit latency, the lowest latency reported is the work by Kitsos et al. (2012) using the CURIPURA-1 and CURIPURA-2 cipher, achieving a latency of 10 cycles per block. Table 21 shows the collective hardware implementation results for lightweight block ciphers and Table 22 shows the collective hardware implementation results for stream ciphers.

Table 22
A listing of selected stream block ciphers and their respective hardware results.

Stream ciphers	Reference	Xilinx FPGA (slices)		Altera FPGA (LE)		Equiv. gate estimation (GE)
		FPGA	Slices	FPGA	LE	
A2U2	David et al. (2011)	–	283	–	–	–
A2U2	Hamalainen et al. (2006)	–	–	–	–	3100
A5/1	Galanis et al. (2005)	2V250FG256	32	–	–	–
A5/1	Batina et al. (2004)	Virtex-HQ800	932	–	–	–
AES	Satoh et al. (2001)	–	–	–	–	5398
AES	Feldhofer et al. (2005)	–	–	–	–	3400
AES-A	Good et al. (2006)	–	–	–	–	6000
AES-B	Good et al. (2006)	XC2S15-5	242	–	–	10,426
AES-CTR	Good and Benaissa (2006)	XC2S15	120	–	–	–
E0	Galanis et al. (2005)	2V250FG256	895	–	–	–
Edon80	Gligoroski et al. (2008)	XC2S15-6	52	–	–	–
Edon80	Gligoroski et al. (2008)	XC4VLX15-12	45	–	–	
Edon80	Gligoroski et al. (2008)	–	–	–	–	2922
F-FCSR-H	Good and Benaissa (2007)	–	–	–	–	4760
Grain	Good et al. (2006)	XC2S15-5	48	EP1C3T-C7	191	1714
GRAIN	Jafarpour et al. (2011)	Spartan-3	227	–	–	–
Grain-128	Good and Benaissa (2007)	–	–	–	–	1857
Grain-128, x16	Good and Benaissa (2007)	–	–	–	–	3189
Grain-128, x32	Good and Benaissa (2007)	–	–	–	–	4617
Grain-128, x4	Good and Benaissa (2007)	–	–	–	–	2129
Grain-128, x8	Good and Benaissa (2007)	–	–	–	–	2489
Grain80	Good and Benaissa (2007)	–	–	–	–	1294
Grain80 ,x16	Good and Benaissa (2007)	–	–	–	–	3239
Grain80, x4	Good and Benaissa (2007)	–	–	–	–	1678
Grain80, x8	Good and Benaissa (2007)	–	–	–	–	2191
Helix	Galanis et al. (2005)	2V250FG256	418	–	–	–
Hermes8	Good et al. (2006)	XC2S30-5	190	EP1C3T-C7	645	5022
Hermes8	Kitsos and Kaiser (2007)	XC2S100-6	697	–	–	–
Hermes8	Kitsos and Kaiser (2007)	XC4VFX12-11	715	–	–	–
MICKEY128	Good and Benaissa (2007)	–	–	–	–	5039
MICKEY-128	Kitsos (2005)	XCV50ECS144	167	–	–	–
Mosquito-A	Good et al. (2006)	XC2S30-5	298	EP1C3T-C7	530	6844
Mosquito-B	Good et al. (2006)	XC2S15-5	190	EP1C3T-C7	431	4178
Phelix, ½ RND	Good and Benaissa (2007)	–	–	–	–	13,159
Phelix, 1 RND	Good and Benaissa (2007)	–	–	–	–	15,032
Phelix-A	Good et al. (2006)	XC2S100-5	1198	EP1C3T-C7	1772	20,404
Phelix-B	Good et al. (2006)	XC2S100-5	1077	EP1C3T-C7	1455	18,080
Phelix-C	Good et al. (2006)	XC2S30-5	264	EP1C3T-C7	1697	12,314
Phelix-D	Good et al. (2006)	XC2S30-5	~250	–	–	~8800
RC4	Galanis et al. (2005)	2V250FG256	140	–	–	–
RC4	Kitsos et al. (2003)	XC2V250	138	–	–	–
Salsa20, 16H	Good and Benaissa (2007)	–	–	–	–	16,394
Salsa20, 1H	Good and Benaissa (2007)	–	–	–	–	12,126
Salsa20, 32H	Good and Benaissa (2007)	–	–	–	–	18,626
Salsa20, 4H	Good and Benaissa (2007)	–	–	–	–	12,914
Salsa20/12	Jarosław (2013)	Spartan-3	14,838	–	–	–
Salsa20/12	Jarosław (2013)	Spartan-6	5079	–	–	–
Salsa20/12	Junjie and Heys (2007)	2V250fg256	194	–	–	–
Salsa20/12	Junjie and Heys (2007)	–	–	–	–	14,100
Sfinks-A	Good et al. (2006)	XC2S30-5	334	EP1C3T-C7	556	5904
Sfinks-B	Good et al. (2006)	XC2S30-5	334	EP1C3T-C7	517	4910
Sfinks-C	Good et al. (2006)	XC2S30-5	319	EP1C3T-C7	508	3946
SNOW3G	Ltd (2009)	Virtex-5	164	–	–	–
SNOW3G	Lingchen et al. (2012)	Virtex-5	188	–	–	–
SNOW3G	Lingchen et al. (2012)	Virtex-5	356	–	–	–
SOSEMANUK	Good and Benaissa (2007)	–	–	–	–	18,819
Trivium	Good et al. (2006)	XC2S15-5	40	EP1C3T-C7	327	2682
Trivium	Good and Benaissa (2007)	–	–	–	–	2599
Trivium	Jafarpour et al. (2011)	Spartan-3	456	–	–	–
Trivium, x4	Good and Benaissa (2007)	–	–	–	–	2660
Trivium, x16	Good and Benaissa (2007)	–	–	–	–	3185
Trivium, x32	Good and Benaissa (2007)	–	–	–	–	3787
Trivium, x64	Good and Benaissa (2007)	–	–	–	–	4921
Trivium, x8	Good and Benaissa (2007)	–	–	–	–	2801
W7	Galanis et al. (2005)	2V250FG256	608	–	–	–
WG-16	Fan and Gong (2013)	–	398	–	–	–
WG-16	Fan et al. (2013b)	–	478	–	–	–
WG-16	Fan et al. (2013b)	–	–	–	–	12,031
WG-8	Jafarpour et al. (2011)	Spartan-3	311	–	–	–
WG-8 (LUT)	Yang et al. (2013)	XC3S1000	85	–	–	–
WG-8 (LUT)	Yang et al. (2013)	XC3S1000	207	–	–	–
WG-8 (LUT)	Yang et al. (2013)	–	–	–	–	1786
WG-8 (LUT)	Yang et al. (2013)	–	–	–	–	3942

Table 22 (*continued*)

Stream ciphers	Reference	Xilinx FPGA (slices)		Altera FPGA (LE)		Equiv. gate estimation (GE)
		FPGA	Slices	FPGA	LE	
WG-8 (TF 1)	Yang et al. (2013)	XC3S1000	83	–	–	–
WG-8 (TF 1)	Yang et al. (2013)	XC3S1000	279	–	–	–
WG-8 (TF 1)	Yang et al. (2013)	–	–	–	–	7523
WG-8 (TF 1)	Yang et al. (2013)	–	–	–	–	42,762
WG-8 (TF 2)	Yang et al. (2013)	XC3S1000	83	–	–	–
WG-8 (TF 2)	Yang et al. (2013)	XC3S1000	306	–	–	–
WG-8 (TF 2)	Yang et al. (2013)	–	–	–	–	3162
WG-8 (TF 2)	Yang et al. (2013)	–	–	–	–	22,668
WG-8 (TF 3)	Yang et al. (2013)	XC3S1000	114	–	–	–
WG-8 (TF 3)	Yang et al. (2013)	XC3S1000	470	–	–	–
WG-8 (TF 3)	Yang et al. (2013)	–	–	–	–	2981
WG-8 (TF 3)	Yang et al. (2013)	–	–	–	–	19,882
ZUC	Kitsos et al. (2011)	Virtex-5	385	–	–	–
ZUC	Liu et al. (2013)	Virtex-6	328	–	–	–
ZUC	Liu et al. (2013)	Virtex-5	350	–	–	–
ZUC	Wang et al. (2011)	–	356	–	–	–
ZUC	Lingchen et al. (2012)	–	395	–	–	–

9. Conclusion

This paper has provided a comprehensive review of modern symmetric cryptographic solutions for resource constrained environments and 100 selected symmetric block ciphers (38 block ciphers, 6 involution ciphers, 28 lightweight block ciphers and 28 stream ciphers). This paper has provided collective surveys from other literatures on the topic of hardware, design requirements, and security trends of RCEs.

From the 100 symmetric ciphers it is clearly observed that issues such as the trends and techniques to design effective and resource efficient symmetric cryptographic solutions (cryptographic implementation) for resource constrained environments are addressed. The reviewed RCE specifications together with the specifications of the hardware environment have given abundant information to readers from a designer's point of view. This paper is able to greatly aid the researchers and designers in terms of understanding and estimating the cryptographic design costs and applications concerning the RCEs.

It can be seen that the many ciphers mentioned are simulation or software based. Hardware results will be able to tell the readers approximately how much is the cryptographic cost for a particular cipher. A fair comparison for all known ciphers would be having all ciphers implementation on a common hardware platform but this is not feasible because there are too many ciphers known in the current development. Moreover, not all researchers use the same design and development environment. This makes it hard to compile a collective result of the hardware implementation reports. The best method would be to gather the reports and results as they are and present them for researchers and designers for their own references to their own respective work, which is what this paper is hoping to achieve.

We hope that this paper will encourage more researchers to design cryptographic primitives and area-efficient and resource-friendly hardware architectures that are suitable for RCE applications. And this paper enables more researches complying with RCE constraints and allowing them to conform to the trends observed, allowing researchers to fairly compare their cryptographic implementations to other works.

References

Adams C. The CAST-128 encryption algorithm [Online]. Network Working Group available: ⟨http://tools.ietf.org/search/rfc2144⟩; 1997.

Adams C, Gilchrist J. The CAST-256 encryption algorithm [Online]. Network Working Group; 1999.

Agency KIS. A design and analysis of 128-bit symmetric block cipher (SEED). KISA; 1999.

Aigner M. RFID tag emulators for HF and UHF frequency range [Online]. Available: ⟨http://www.iaik.tugraz.at/content/research/rfid/tag_emulators/⟩; 2006 [accessed 05.05.13].

Akishita T, Hiwatari H. Very compact hardware implementations of the blockcipher CLEFIA. In: Proceedings of the 18th international conference on selected areas in cryptography. Toronto, Ont., Canada. Springer-Verlag Berlin, Heidelberg, 2012.

Akyildiz IF, Melodia T, Chowdhury KR. A survey on wireless multimedia sensor networks. Comput Netw 2007;51:921–60.

Anderson RJ, Biham E, Knudsen LR. Serpent and smartcards. In: Proceedings of the the international conference on smart card research and applications. Springer-Verlag London, UK.; 2000.

Aoki K, Ichikawa T, Kanda M, Matsui M, Moriai S, Nakajima J, & et al. 2000. Specification of Camelia – a 128-bit block cipher. NTT and Mitsubishi Electric Corporation 2000–2001: Nippon Telegraph and Telephone Corporation, Mitsubishi Electric Corporation.

Aoki K, Ichikawa T, Kanda M, Matsui M, Moriai S, Nakajima J, et al. Camellia: a 128-bit block cipher suitable for multiple platforms — design and analysis. In: Stinson D, Tavares S, editors. Selected areas in cryptography. Berlin, Heidelberg: Springer; 2001.

Aragones-Vilella J, Martínez-Ballesté A, Solanas A. A brief survey on RFID privacy and security. In: Proceedings of the world congress on engineering; 2007.

At N, Beuchat J-L, Okamoto E, San I, Yamazaki T. Compact Hardware Implementations of ChaCha, BLAKE, Threefish, and Skein on FPGA. IEEE Trans Circuits Syst I: Regul Pap 2013:1–14.

Babbage S, Dodd M. The MICKEY stream ciphers. In: Robshaw M, Billet O, editors. New stream cipher designs. Berlin, Heidelberg: Springer; 2008.

Barreto, PSLM, Rijmen V. The Anubis block cipher [Online]. Available: ⟨http://www.larc.usp.br/~pbarreto/anubis.zip⟩; 2001a.

Barreto, PSLM, Rijmen. The Khazad legacy-level block cipher [Online]. Available: ⟨http://www.larc.usp.br/~pbarreto/khazad-tweak.zip⟩; 2001b.

Barreto PSLM, Simplício MA. CURUPIRA, a block cipher for constrained platforms. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC'2007. Belém, PA; 2007.

Batina L, Lano J, Mentens N, Ors SB, Preneel, B. & Verbauwhede, I. Energy, performance, area versus security trade-offs for stream ciphers. In: The state of the art of stream ciphers: workshop record. Brugge, Belgium; 2004. p. 9.

Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L. 2012. Performance of the SIMON and SPECK families of lightweight block ciphers. Agency NS ⟨http://iauth.org/wp-content/uploads/2013/01/SimonSpeckPerformance1.pdf⟩.

Berbain C, Billet O, Canteaut A, Courtois N, Gilbert H, Goubin L, et al. Sosemanuk, a fast software-oriented stream cipher. In: Robshaw M, Billet O, editors. New stream cipher designs. Berlin, Heidelberg: Springer; 2008.

Bernstein D. The Salsa20 family of stream ciphers. In: Robshaw M, Billet O, editors. New stream cipher designs. Berlin, Heidelberg: Springer; 2008.

Beuchat J-L. FPGA implementations of the RC6 block cipher. In: Cheung P, Constantinides G, editors. Field programmable logic and application. Berlin, Heidelberg: Springer; 2003.

Bevi AR, Sheshu SSV, Malarvizhi S. FPGA based pipelined architecture for RC5 encryption. In: Proceedings of the 2012 second international conference on Digital Information and Communication Technology and it's Applications (DICTAP), 16–18 May, 2012. p. 214–9.

Biryukov A. A new 128-bit key stream cipher LEX. In eSTREAM, ECRYPT stream cipher project, report 2005/013; 2005.

Biryukov A, Shamir A, Wagner D. Real time cryptanalysis of A5/1 on a PC. In: Goos G, Hartmanis J, Van Leeuwen J, Schneier B, editors. Fast software encryption. Berlin, Heidelberg: Springer; 2001.

Boesgaard M, Vesterager M, Zenner E. The rabbit stream cipher. In: Robshaw M, Billet O, editors. New stream cipher designs. Berlin, Heidelberg: Springer; 2008.

Bogdanov A, Knudsen LR, leander G, Paar C, Poschmann A, Robshaw MJ, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of the 9th international workshop on cryptographic hardware and embedded systems. Vienna, Austria: Springer-Verlag; 2007.

Bono SC, Green M, Stubblefield A, Juels A, Rubin AD, Szydlo M. Security analysis of a cryptographically-enabled RFID device. In: Proceedings of the 14th conference on USENIX security symposium, vol. 14. Baltimore, MD: USENIX Association; 2005.

Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçın., PRINCE: a low-latency block cipher for pervasive computing applications. In: Proceedings of the 18th international conference on the theory and application of cryptology and information security. Beijing, China. Berlin, Heidelberg: Springer-Verlag; 2012.

Braeken A, Lano J, Mentens N, Preneel B. SFINKS: a synchronous stream cipher for restricted hardware environments. eSTREAM, ECRYPT stream cipher project, Report 2005/026. Available at: 〈http://www.ecrypt.eu.org/stream〉.

Bucholc K, Chmiel K, Grocholewska-Czury #322 A, Idzikowska E, Janicka-Lipska I, Stok J, et al. Scalable PP-1 block cipher. Int J Appl Math Comput Sci 2010;20:401–11.

Bukkapatnam S, Govardhan JM, Hariharan S, Rajamani V, Gardner B, Contreras A. Report of work conducted under the aegis of celdi strategic research grant 2005 "experimental test bed for performance evaluation of RFID systems". Sensor (RFID) networks and complex manufacturing systems monitoring (commsens): Laboratory for RFID research. Stillwater, OK: Oklahoma State University; 2005.

Burwick C, Coppersmith D, D'avignon E, Gennaro R, Halevi S, Jutla C, et al. MARS – a candidate cipher for AES. 〈http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.6084&rep=rep1&type=pdf〉; 1999. p. 1–62.

Christophe Cannière, Orr Dunkelman and Miroslav Knežević. KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Proceedings of the 11th international workshop on cryptographic hardware and embedded systems. Lausanne, Switzerland. Springer-Verlag; 2009.

Chae H-JAY, Daniel J, Smith Joshua R, Fu, Kevin. Maximalist Cryptography and Computation on the WISP UHF RFID Tag. In: Proceedings of the conference on RFID security, July 2007.

Chodowiec P, Gaj K. Implementation of the twofish cipher using FPGA devices. Technical Report. Electrical and computer engineering. George Mason University; 1999.

Contributors, W. WISP [Online]. Wikispaces: tangient LLC. Available: 〈http://wisp.wikispaces.com/〉; 2013. [accessed 28.05.13].

Corporation, IT. Intermec RFID tags & media – meeting the scalable RFID challenge [Online]. 〈http://www.intermec.com/products/rfid/tags_inserts_and_smart_labels/index.aspx〉; 2012 [accessed 21.06.13].

Corporation N. EPC RFID Label Selector Guide [Online]. Available: 〈http://nashua.com/Resources/ProductSheets/10049-EPCSelecGuide.pdf〉; 2008a.

Corporation N. Item level tagging selector guide [Online]. 〈http://nashua.com/ProdAndServices/RFID_ItemLevel.aspx?Selected=LabelTrans〉; 2008b [accessed 21.06.13].

Czeskis A, Koscher K, Smith JR, Kohno T. RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In: Proceedings of the 15th ACM conference on computer and communications security. Alexandria, VA, USA. ACM; 2008.

Daemen J, Govaerts R, Vandewalle J. A new approach to block cipher design. In: Anderson R, editor. Fast software encryption. Berlin, Heidelberg: Springer; 1994.

Daemen J, Knudsen L, Rijmen V. The block cipher square. In: Biham E, editor. Fast software encryption. Berlin, Heidelberg: Springer; 1997.

Daemen J, Peeters M, Van AsscheG, Rijmen V. Nessie Proposal: NOEKEON [Online]. 〈http://gro.noekeon.org〉; 2000.

David, M. 2011. Lightweight cryptography for passive RFID tags [Ph.d. thesis]. Technology Platform Section, Department of Electronic Systems, Aalborg University.

David M, Ranasinghe DC, Larsen T. A2U2: a stream cipher for printed electronics RFID tags. 2011 IEEE International Conference on RFID (RFID), 12–14 April 2011. p. 176–83.

De Cannière C. Trivium: a stream cipher construction inspired by block cipher design principles. In: Katsikas S, López J, Backes M, Gritzalis S, Preneel B, editors. Information security. Berlin, Heidelberg: Springer; 2006.

De Canniere C, Biryukov A, Preneel B. An introduction to block cipher cryptanalysis. Proc IEEE 2006;94:346–56.

Deavours DD. UHF EPC tag performance evaluation. RFID Alliance Lab, University of Kansas; 2005.

Dong-Liang W, Ng WWY, Yeung DS, Hai-Lan D. A brief survey on current RFID applications. In: International conference on machine learning and cybernetics, 12–15 July 2009. p. 2330–5.

Doomun MR, Soyjaudah K. Analytical comparison of cryptographic techniques for resource-constrained wireless security. Int J Netw Secur. 2009;9:82–94.

Döser E, Jamal A, Vestlund SC. Security mechanisms in resource-constrained computer systems [Online]; 2011.

Dunkelman O, Keller N, Shamir A.A practical-time attack on the A5/3 cryptosystem used in third generation gsm telephony; 2010.

Eisenbarth T, Kumar S. A survey of lightweight-cryptography implementations. IEEE Des Test Comput 2007;24:522–33.

Ekdahl P, Johansson T. SNOW – a new stream cipher. In: Proceedings of first open nessie workshop. Ku-Leuven; 2001.

Ekdahl P, Johansson T. A new version of the stream cipher SNOW. Revised papers from the 9th annual international workshop on selected areas in cryptography. Springer-Verlag; 2003.

Elbirt AJ, Paar C. An FPGA implementation and performance evaluation of the Serpent block cipher. In: Proceedings of the 2000 ACM/SIGDA eighth international symposium on Field programmable gate arrays. Monterey, CA, USA: ACM; 2000.

Standaert F-X, G.P., Rouvroy G, Quisquater J-J, Legat J-D. ICEBERG: an involutional cipher efficient for block encryption in reconfigurable hardware. In: Springer B, editor. Proceedings of FSE 2004, the fast software encryption workshop, February 5–7, 2004. New Delhi. Berlin: Springer; 2004.

Fan X, Gong G. Specification of the stream cipher WG-16 based confidentiality and integrity algorithms [Online]. 2013.

Fan X, Mandal K, Gong G. WG-8: a lightweight stream cipher for resource-constrained smart devices. In: Singh K, Awasthi A, editors. Quality, reliability, security and robustness in heterogeneous networks. Berlin, Heidelberg: Springer; 2013a.

Fan X, Zidaric N, Aagaard M, Gong G. Efficient hardware implementation of the stream cipher WG-16 with composite field arithmetic. In: Proceedings of the 3rd international workshop on trustworthy embedded devices. Berlin, Germany. New York, NY, USA: ACM; 2013b.

Feistel H, Ibm. Block cipher cryptographic system 1971.

Feldhofer M, Wolkerstorfer J, Rijmen V. AES implementation on a grain of sand. IEE Proc Inf Secur 2005;152:13–20.

Ferguson N, Lucks S, Schneier B, Whiting D, Bellare M, Kohno T, et al. 2008. The skein hash function family [Online]. Available: 〈http://www.skein-hash.info/sites/default/files/skein1.1.pdf〉 2013].

Force GT. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2; 2006 [Document 2: SNOW 3G specification].

Fournaris AP, Sklavos N, Koufopavlou O. VLSI architecture and FPGA implementation of ICE encryption algorithm. In: Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003, vol. 1, 14–17 December 2003. p. 88–91.

Galanis MD, Kitsos P, Kostopoulos G, Sklavos N, Goutis CE. Comparison of the hardware implementation of stream ciphers. Int Arab J Inf Technol (IAJIT) Coll Comput Inf Soc 2005;2:267–74.

Ganesan P, Venugopalan R, Peddabachagari P, Dean A, Mueller F, Sichitiu M. Analyzing and modeling encryption overhead for sensor network nodes. In: Proceedings of the 2nd ACM international conference on wireless sensor networks and applications. San Diego, CA, USA. ACM; 2003.

Gligoroski D, Markovski S, Knapskog S. The stream cipher Edon80. In: Robshaw M, Billet O, editors. New stream cipher designs. Berlin, Heidelberg: Springer; 2008.

Gong, G. Lightweight cryptography for RFID systems. In: Proceedings of the international conference on cryptology in India (Tutorial Talk). Hyderabad, India; 2010.

Gong Z, Nikova S, Law YW. KLEIN: a new family of lightweight block ciphers. In: Proceedings of the 7th international conference on RFID security and privacy. Amherst, MA. Springer-Verlag; 2012.

Good T, Benaissa M. AES on FPGA from the fastest to the smallest. In: Rao J, Sunar B, editors. *Cryptographic hardware and embedded systems – CHES* 2005. Berlin, Heidelberg: Springer; 2005.

Good T, Benaissa M. AES as stream cipher on a small FPGA. In: Proceedings of the 2006 IEEE international symposium on circuits and systems, 2006. ISCAS 2006, 21–24 May 2006. 4 pp.

Good T, Benaissa M. Hardware results for selected stream cipher candidates of stream ciphers. SASC 2007, workshop record; 2007.

Good T, Chelton W, Benaissa M. Review of stream cipher candidates from a low resource hardware perspective. ECRYPT eSTREAM; 2006.

Guerrero-Zapata M, zilan R, Barceló-Ordinas J, Bicakci K, Tavli B. The future of security in wireless multimedia sensor networks. Telecommun Syst 2010;45:77–91.

Guo J, Peyrin T, Poschmann A, Robshaw M.The LED block cipher. In: Proceedings of the 13th international conference on cryptographic hardware and embedded systems. Nara, Japan. Springer-Verlag; 2011.

Hamalainen, P., Alho, T., Hannikainen, M. & Hamalainen, T.D. Design and implementation of low-area and low-power AES encryption hardware core. In: Proceedings of the 9th EUROMICRO conference on Digital System Design: architectures, methods and tools, 2006. DSD 2006. 0-0 0 2006. p. 577–83.

Hell M, Johansson T, Meier W. Grain: a stream cipher for constrained environments. Int J Wire Mob Comput 2007;2:86–93.

Hempstead M, Lyons MJ, Brooks D, Wei G-Y. Survey of hardware systems for wireless sensor networks. J. Low Power Electron. 2008;4:11–20.

Hoffman N. A simplified IDEA algorithm. Cryptologia 2007;31:143–51.

Hong D, Sung J, Hong S, Lim J, Lee S, Koo B, et al. HIGHT: a new block cipher suitable for low-resource device. In: Proceedings of the 8th international workshop on Cryptographic Hardware and Embedded Systems – CHES 2006. Springer-Verlag Berlin, Heidelberg; 2006.

Hong E, Chung J-H, Lim CH. Hardware design and performance estimation of the 128-bit block cipher crypton. In: Proceedings of the first international workshop on cryptographic hardware and embedded systems. Springer-Verlag London, UK; 1999.

Huber K, Wolter S. Telekom's MAGENTA algorithm for en-/decryption in the Gigabit/sec range. In: 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, vol. 6, 7–10 May 1996. p. 3233–5.

Huiju C, Heys HM. Compact hardware implementation of the block cipher Camellia with concurrent error detection. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2007. CCECE 2007, 22–26 April 2007. p. 1129–32.

Huiju C, Heys HM, Cheng W. PUFFIN: a novel compact block cipher targeted to embedded digital systems. In: Proceedings of the 11th EUROMICRO conference on Digital System Design architectures, methods and tools, 2008. DSD '08. 3–5 September 2008. p. 383–90.

Ibtechnology. List of supported tag types and key features [Online]. Available: 〈http://www.ibtechnology.co.uk/pdf/tag_types.pdf〉; 2013 [accessed 21.06.13].

Inc, AD. HF RFID inlays [Online]. 〈http://rfid.averydennison.com/product_cat/hf-rfid-inlays/〉; 2013a [accessed 21.06.13].

Inc AD. UHF RFID inlays [Online]. Available: 〈http://nashua.com/Resources/ProductSheets/10049-EPCSelecGuide.pdf〉; 2013b [accessed 21.06.13].

Industries A. RFID selection guide. Version 1; 2010.

Izadi M, Sadeghiyan B, Sadeghian SS, Khanooki HA. MIBS: a new lightweight block cipher. In: Proceedings of the 8th international conference on cryptology and network security. Kanazawa, Japan. Berlin, Heidelberg: Springer-Verlag; 2009.

Jafarpour A, Mahdlo A, Akbari A, Kianfar K. Grain and Trivium ciphers implementation algorithm in FPGA chip and AVR micro controller. In: 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 4–7 December 2011. p. 656–8.

Jarosław S. Low-cost hardware implementations of Salsa20 stream cipher in programmable devices. J Pol Saf Reliab Assoc 2013;4:8.

Jeong K, Choi J, Lee Y, Lee C, Sung J, Park H, et al. Update on SEED: SEED-192/256. In: Park J, Chen H-H, Atiquzzaman M, Lee C, Kim T-H, Yeo S-S, editors. Advances in information security and assurance. Berlin, Heidelberg: Springer; 2009.

Jinsub P, Young-Dae K, Sangwoon Y, Younggap Y. Low power compact design of ARIA block cipher. In: Proceedings of 2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006, 21–24 May 2006. 4 p.–316.

Jinwala DC, Patel DR, Dasgupta KS. Investigating and analyzing the light-weight ciphers for wireless sensor networks. INFOCOMP J Comput Sci 2009;8:12.

John J. Cryptography for resource constrained devices: a survey. Int J Comput Sci Eng 2012;4:1766–70.

Johnson M, Healy M, Van De venP, Hayes MJ, Nelson J, Newe T, et al. A comparative review of wireless sensor network mote technologies. In: Sensors, 2009 IEEE, 25–28 October 2009. p. 1439–42.

Jolfaei A, Vizandan A, Mirghadri A. Image encryption using HC-128 and HC-256 stream ciphers. Int J Electron Secur Digit Forensic 2012;4:19–42.

Juels A. Minimalist cryptography for low-cost RFID tags (extended abstract). In: Blundo C, Cimato S, editors. Security in communication networks. Berlin, Heidelberg: Springer; 2005.

Juels A. RFID security and privacy: a research survey. IEEE J Sel Areas Commun 2006;24:381–94.

Juels A, Weis S. Authenticating pervasive devices with human protocols. In: Shoup V, editor. Advances in cCryptology – CRYPTO 2005. Berlin, Heidelberg: Springer; 2005.

Junjie Y, Heys HM. Hardware implementation of the Salsa20 and Phelix stream ciphers. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2007. CCECE 2007, 22–26 April 2007. p. 1125–8.

Kai Z, Lina G. A survey on the Internet of things security. In: Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS), 2013, 14–15 December 2013. p. 663–7.

Kaiser U. 2006. Hermes8: a low-complexity low-power stream cipher. IACR Cryptology ePrint Archive 2006. p. 19.

Kaps, J-P. Chai-Tea, Cryptographic hardware implementations of xTEA. In: Proceedings of the 9th international conference on cryptology in india: progress in cryptology. Kharagpur, India. Springer-Verlag Berlin, Heidelberg; 2008.

Kasper M, Kumar E, Lemke-Rust K, Paar C. A Compact implementation of Edon80 ⋆ [Online]; 2006.

Khan MA, Sharma M, R BP. A survey of RFID tags. Int J Recent Trends Eng 2009;1:4.

Khor J, Ismail W, Younis M, Sulaiman MK, Rahman M. Security problems in an RFID system. Wirel Pers Commun 2011;59:17–26.

Kitsos, P. On the hardware implementation of the mickey-128 stream cipher. IACR cryptology ePrint archive; 2005.

Kitsos P, Galanis MD, Koufopavlou O. A RAM-based FPGA implementation of the 64-bit MISTY1 block cipher. In: Proceedings of IEEE International Symposium on Circuits and Systems. ISCAS 2005, vol. 5, 23–26 May 2005. p. 4641–4.

Kitsos P, Kaiser U. A high-speed hardware implementation of the Hermes8-128 stream cipher. In: 18th European Conference on Circuit Theory and Design, 2007. ECCTD 2007, 27–30 August 2007. p. 364–7.

Kitsos P, Kostopoulos G, Sklavos N, Koufopavlou, O. Hardware implementation of the RC4 stream cipher. In: IEEE 46th Midwest symposium on circuits and systems, 2003, vol. 3, 27–30 December 2003. p. 1363–6.

Kitsos P, Selimis G, Koufopavlou O. 2008a. A high performance ASIC implementation of the SNOW 3G stream cipher. In: Proceedings of the 16th International Conference on Very Large Scale Integration (VLSI-SoC 2008). Rhodes Island, Greece.

Kitsos P, Selimis G, Koufopavlou O, Skodras AN. A hardware implementation of CURUPIRA block cipher for wireless sensors. In: 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, 2008. DSD '08, 3–5 September 2008. 2008b. p. 850–3.

Kitsos P, Sklavos N, Parousi M, Skodras AN. A comparative study of hardware architectures for lightweight block ciphers. Comput Electr Eng 2012;38:148–60.

Kitsos P, Sklavos N, Skodras AN. An FPGA implementation of the ZUC stream cipher. In: Proceedings of the 14th Euromicro conference on Digital System Design (DSD), 2011, August 31 2011–September 2. p. 814–7.

Knudsen L, Leander G, Poschmann A, Robshaw MJB. PRINTcipher: a block cipher for IC-printing. In: Proceedings of the 12th international conference on cryptographic hardware and embedded systems. Santa Barbara, CA. Springer-Verlag Berlin, Heidelberg; 2010.

Knudsen L, Rijmen V, Rivest R, Robshaw MB. On the design and security of RC2. In: Vaudenay S, editor. Fast software encryption. Berlin, Heidelberg: Springer; 1998.

Knudsen LR. DEAL – a 128-bit block cipher. NIST AES proposal; 1998.

Ko Y, Hong S, Lee W, Lee S, Kang J-S. Related key differential attacks on 27 rounds of XTEA and full-round GOST. In: Roy B, Meier W, editors. Fast software encryption. Berlin, Heidelberg: Springer; 2004.

Koch D, Korber M, Teich JU. Searching RC5-keys with distributed reconfigurable computing. In: Plaks TP, editor. ERSA 2006. Las Vegas, NV, USA: CSREA Press; 2006. p. 42–8.

Kong JH, Ang L-M, Seng KP. A very compact AES-SPIHT selective encryption computer architecture design with improved S-box. J Eng 2013:26.

Kong JH, Ang LM, Seng KP, Ong FT. Low-complexity two instruction set computer architecture for sensor network using Skipjack encryption. In: Proceedings of International Conference on Information Networking (ICOIN), 26–28 January 2011. p. 472–7.

Kumar PK, Baskaran K. An ASIC implementation of low power and high throughput blowfish crypto algorithm. Microelectron J 2010;41:347–55.

Kundur D, Luh W, Okorafor UN, Zourntos T. Security and privacy for distributed multimedia sensor networks. Proc IEEE 2008;96:112–30.

Langheinrich M. A survey of RFID privacy approaches. Personal Ubiquitous Computing 2009;13:413–21.

Law YW, Doumen J, Hartel P. Survey and benchmark of block ciphers for wireless sensor networks. ACM Trans Sen Netw 2006;2:65–93.

Leander G, Paar C, Poschmann A, Schramm K. New lightweight DES variants. In: Proceedings of the 14th international conference on Fast Software Encryption. Luxembourg, Luxembourg. Springer-Verlag Berlin, Heidelberg. 2007.

Lee C-H, Kim J-S. The new block cipher rainbow. Samsung Advanced Institute of Technology; 1997.

Li, BSAWWALZAY. Some observations on TWIS block cipher. IACR Eprint archive; 2010.

Li Z, Gong G. A survey on security in wireless sensor networks. Waterloo, Ont., Canada: University of Waterloo; 2008.

Lim CH, Korkishko T. mCrypton: a lightweight block cipher for security of low-cost RFID tags and sensors. In: Proceedings of the 6th international conference on Information Security Applications. Jeju Island, Korea. Springer-Verlag Berlin, Heidelberg; 2006.

Lingchen Z, Luning X, Zongbin L, Jiwu J, Yuan M. Evaluating the optimized implementations of SNOW3G and ZUC on FPGA. In: Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 25-27 June 2012. p. 436–42.

Liu Z, Gao N, Jing J, PL. 2013. HPAZ: a High-throughput Pipeline Architecture of ZUC in Hardware. IACR Cryptology ePrint Archive; 2012. p. 461.

Ltd, HT. SNOW3G stream cipher core for Xilinx FPGA; 2009.

Lu J. Related-key rectangle attack on 36 rounds of the XTEA block cipher. Int J Inf Secur 2009;8:1–11.

Mace F, Standaert F-X, Quisquater J-J. ASIC implementations of the block cipher SEA for constrained applications. RFID security – RFIDsec, Malaga, Spain. 2007.

Marneweck, K. 2011. An introduction to KEELOQ® Code hopping. Available: 〈http://www.microchip.com/stellent/groups/techpub_sg/documents/appnotes/en010992.pdf〉.

Matsui M. Block encryption algorithm MISTY. In: Fast Software Encryption (FSE 1997). Springer-Verlag Berlin, Heidelberg; 1997. p. 64–74.

Merkle RC. Method and apparatus for data encryption. Google Patents; 1991.

Miyaguchi S. The FEAL cipher family. In: Menezes A, Vanstone S, editors. Advances in Cryptology-CRYPT0'90. Berlin, Heidelberg: Springer; 1991.

Mora-Gutiérrez JM, Jiménez-Fernández CJ, Valencia-Barrero M. Low power implementation of trivium stream cipher. In: Ayala J, Shang D, Yakovlev A, editors. Integrated circuit and system design. power and timing modeling, optimization and simulation. Berlin, Heidelberg: Springer; 2013.

Moradi A, Poschmann A, Ling S, Paar C, Wang H. Pushing the Limits: A Very Compact and a Threshold implementation of AES. In: PATERSON K, editor. Advances in cryptology – EUROCRYPT 2011. Berlin, Heidelberg: Springer; 2011.

Nada, J.H. & ŇAslund, M.The stream cipher polar bear [Online]; 2005.

National Institute of Standards and Technology, N. SKIPJACK and KEA algorithm specifications version 2.0. National Institute of Standards and Technology (NIST); 1998.

Needham DWAR. TEA, a tiny encryption algorithm. Springer-Verlag Berlin, Heidelberg; 1995. 97–110.

O'neil S, Gittins B, Landman H. VEST hardware-dedicated stream ciphers. eSTREAM, ECRYPT Stream Cipher Project; 2005.

Ojha S, Kumar N, Jain K, Sangeeta. TWIS – a lightweight block cipher. In: Prakash A, Sen Gupta I, editors. Information systems security. Berlin, Heidelberg: Springer; 2009.

Plos T, Dobraunig C, Hofinger M, Oprisnik A, Wiesmeier C, Wiesmeier J. Compact hardware implementations of the block ciphers mCrypton, NOEKEON, and SEA. In: Galbraith S, Nandi M, editors. Progress in cryptology – INDOCRYPT 2012. Berlin, Heidelberg: Springer; 2012.

Poschmann A, Leander G, Schramm K, Paar C. New light-weight crypto algorithms for RFID. In: Proceedings of IEEE International Symposium on Circuits and Systems, 2007, ISCAS 2007, 27–30 May 2007. p. 1843–6.

Poschmann A, Ling S, Wang H. 256 Bit standardized crypto for 650 GE – GOST revisited. In: Mangard S, Standaert F-X, editors. Cryptographic hardware and embedded systems, CHES 2010. Berlin, Heidelberg: Springer; 2010.

Poschmann AY. Lightweight cryptography: cryptographic engineering for a pervasive world. Verlag: European University; 2009.

Prasun Ghosal MB, Manish Biswas. Hardware implementation of TDES crypto system with on chip verification in FPGA. The Computing Research Repository (CoRR) J Telecommun 2010;1:113–7.

Ranasinghe DC, Lim D, Cole PH, Devadas S. A low cost solution to authentication in passive RFID systems. 2006.

Rashidi B. FPGA implementation of optimized the 64-bit RC5 encryption algorithm. Elixir Int J 2012;51:10700–3.

Riaz M, Heys HM. The FPGA implementation of the RC6 and CAST-256 encryption algorithms. In: Proceedings of the 1999 IEEE Canadian conference on electrical and computer engineering, vol. 1, 9–12 May 1999. p. 367–72.

Rieback MR, Gaydadjiev GN, Crispo B, Hofman RFH, Tanenbaum AS. A platform for RFID security and privacy administration. In: Proceedings of the 20th conference on Large Installation System Administration. Washington, DC. USENIX Association; 2006.

Rijmen V, Daemen J, Preneel B, Bosselaers A, Win ED. The Cipher SHARK. In: Fast software encryption, third international workshop. Springer-Verlag Berlin, Heidelberg; 1996.

Rivest RL. The RC5 encryption algorithm. In: Proceedings of the second international workshop on Fast Software Encryption (FSE) 1994, 1994. p. 86–96.

Rolfes C, Poschmann A, Leander G, Paar C.Ultra-Lightweight Implementations for Smart Devices — Security for 1000 Gate Equivalents. In: Proceedings of the 8th IFIP WG 8.8/11.2 international conference on Smart Card research and advanced applications. London, UK. Springer-Verlag Berlin, Heidelberg; 2008.

Roman R, Alcaraz C, Lopez J. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. Mob Netw Appl 2007;12:231–44.

Romer K, Mattern F. The design space of wireless sensor networks. Wirel Commun IEEE 2004;11:54–61.

Rouvroy G, Standaert FX, Quisquater JJ, Legat JD. Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications. In: Proceedings of international conference on Information Technology: Coding and Computing, 2004. ITCC 2004, vol. 2. 5–7 April 2004. p. 583–7.

Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, Eita Kobayashi. TWINE: a lightweight. Versatile block cipher; 2011.

Saarinen M-J O, Engels DW. A do-it-all-cipher for rfid: design requirements (extended abstract). IACR cryptology ePrint archive; 2012. p. 317.

Sample AP, Yeager DJ, Powledge PS, Mamishev AV, Smith JR. Design of an RFID-based battery-free programmable sensing platform. IEEE Trans Instrum Meas 2008;57:2608–15.

Saqib NA, Rodr´Iguez-Henriquez F, D´Iaz-P´Erez A.. A compact and efficient FPGA implementation of the DES algorithm. In: International conference on Reconfigurable Computing and FPGAs (ReConFig'04). Colima, Mexico; 2004.

Sarma SE, Weis SA, Engels DW. RFID systems and security and privacy implications. In: Revised papers from the 4th international workshop on cryptographic hardware and embedded systems. Springer-Verlag Berlin, Heidelberg; 2003.

Satoh A, Morioka S, Takano K, Munetoh S. A compact Rijndael hardware architecture with S-box optimization. In: Proceedings of the 7th international conference on the theory and application of cryptology and information security: advances in cryptology. Springer-Verlag Berlin, Heidelberg; 2001.

Schneier B, Kelsey J, Whiting D, Wagner D, Hall C, Ferguson N. The twofish encryption algorithm: a 128-bit block cipher. New York, NY, USA: John Wiley & Sons, Inc.; 1999.

Sen J. A survey on wireless sensor network security. Int J Commun Netw Inf Secur 2009;1:24.

Shareef FF, Hashim AT, Shareef WF. Compact hardware implementation of FPGA based RC6 block cipher. J Eng Appl Sci 2008;3:598–601.

Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, Shirai T. Piccolo: an ultra-lightweight blockcipher. In: Proceedings of the 13th international conference on cryptographic hardware and embedded systems. Nara, Japan. Springer-Verlag Berlin, Heidelberg; 2011.

Simplíciojr M, Pslm B, Tcmb C, Cb M & M, N. The CURUPIRA-2 block cipher for constrained platforms: specification and benchmarking. In: 1st international workshop on Privacy in Location-Based Applications – PiLBA '08. Malaga, Spain; 2008.

Smith J, Sample A, Powledge P, Roy S, Mamishev A. A wirelessly-powered platform for sensing and computation. In: Dourish P, Friday A, editors. UbiComp 2006: ubiquitous computing. Berlin, Heidelberg: Springer; 2006.

Soro S, Heinzelman W. A survey of visual sensor networks. Adv Multimed 2009:2009.

Standaert F-X, Piret G, Gershenfeld N, Quisquater J-J. SEA: a scalable encryption algorithm for small embedded applications. In: Proceedings of the 7th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications. Tarragona, Spain. Springer-Verlag; 2006.

Stefan D. Hardware framework for the rabbit stream cipher. In: Bao F, Yung M, Lin D, Jing J, editors. Information security and cryptology. Berlin, Heidelberg: Springer; 2010.

Sugawara T, Homma N, Aoki T, Satoh A. A high-performance ASIC implementation of the 64-bit block cipher CAST-128. In: Proceedings of IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007. 27–30 May2007. p. 1859–62.

Tavli B, Bicakci K, Zilan R, Barcelo-Ordinas J. A survey of visual sensor network platforms. Multimed Tools Appl 2012;60:689–726.

Tieming C, Liang G, Xiaohao W, Jiamei C. TinyStream: A Lightweight and Novel Stream Cipher Scheme for Wireless Sensor Networks. In: Proceedings of the 2010 international conference on Computational Intelligence and Security (CIS), 11–14 December 2010. p. 528–32.

Wang L, Jing J, Liu Z, Zhang L, Pan W. Evaluating optimized implementations of stream cipher ZUC algorithm on FPGA. In: Qing S, Susilo W, Wang G, Liu D, editors. Information and communications security. Berlin, Heidelberg: Springer; 2011.

Westhues J. A Test Instrument for HF/LF RFID [Online]. Available: 〈http://cq.cx/proxmark3.pl〉; 2007 [accessed 05.05.13].

Wetherall, MBABGAASAJRSAD. Revisiting smart dust with RFID sensor networks. In: Proceedings of the 7th ACM workshop on Hot Topics in Networks (Hotnets-VII), October 2008.

Whiting D, Schneier B, Lucks S, Muller F. Phelix: fast encryption and authentication in a single cryptographic primitive. eSTREAM, ECRYPT Stream Cipher Project Report; 2005.

Winkler T, Rinner B. Security and privacy protection in visual sensor networks: a survey. University of Klagenfurt; 2012.

Wu H. A new stream cipher HC-256. In: Roy B, Meier W, editors. Fast software encryption. Berlin, Heidelberg: Springer; 2004.

Wu H. The stream cipher HC-128. In: Matthew R, Olivier B, editors. *New stream cipher designs*. Springer-Verlag Berlin, Heidelberg; 2008.

Wu W, Zhang L. LBlock: a lightweight block cipher. In: Proceedings of the 9th international conference on Applied cryptography and network security. Nerja, Spain. Springer-Verlag Berlin, Heidelberg; 2011.

Xiangqian C, Kia M, Kang Y, Pissinou N. Sensor network security: a survey. Commun Surv Tutor IEEE 2009;11:52–73.

Xueying Z, Heys HM, Cheng L. Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks. In: 2010 25th Biennial Symposium on Communications (QBSC), 12–14 May 2010. p. 168–72.

Yamamoto D, Yajima J, Itoh K. A very compact hardware implementation of the MISTY1 block cipher. In: Oswald E, Rohatgi P, editors. Cryptographic Hardware and Embedded Systems – CHES 2008. Berlin, Heidelberg: Springer; 2008.

Yang G, Fan X, Aagaard M, Gong G. Design space exploration of the lightweight stream cipher WG-8 for FPGAs and ASICs. In: Proceedings of the workshop on embedded systems security. Montreal, Quebec, Canada. ACM New York, NY, USA; 2013.

Yap H, Khoo K, Poschmann A, Henricksen M. EPCBC: a block cipher suitable for electronic product code encryption. In: Proceedings of the 10th international conference on cryptology and network security. Sanya, China. Springer-Verlag Berlin, Heidelberg; 2011.

Yarrkov, E. Cryptanalysis of XXTEA. IACR Eprint archive; 2010.

Yeager DJ, Powledge PS, Prasad R, Wetherall D, Smith JR. Wirelessly-charged UHF tags for sensor data collection. In: Proceedings of 2008 IEEE International Conference on RFID, 16–17 April 2008. p. 320–7.

Yi J, Park K, Park J, Ro W. Fully pipelined hardware implementation of 128-bit SEED block cipher algorithm. In: Becker J, Woods R, Athanas P, Morgan F, editors. Reconfigurable computing: architectures, tools and applications. Berlin, Heidelberg: Springer; 2009.

Yiyuan L, Qi C, Guang G, Xuejia L. A lightweight stream cipher WG-7 for RFID encryption and authentication. In: Proceedings of the Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 6–10 December 2010. p. 1–6.

Yong W, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. Commun Surv Tutor IEEE 2006;8:2–23.

Yoshikawa M, Sakaue K. Dedicated hardware for RC5 cryptography and its implementation. In: World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOM'11). Las Vegas, NV, USA; 2011.

Young-Ho S, Jong-Hyeon K & Dong-Wook, K. Hardware implementation of 128-bit symmetric cipher SEED. Proceedings of the second IEEE Asia Pacific Conference on ASICs. AP-ASIC 2000. p. 183–6.