Lightweight Cryptography for Embedded Systems - A Comparative Analysis

Charalampos Manifavas[1], George Hatzivasilis[2], Konstantinos Fysarakis[2], and Konstantinos Rantos[3]

[1] Dept. of Applied Informatics & Multimedia, Technological Educational Institute of Crete, Heraklion, Crete, Greece
harryman@epp.teicrete.gr
[2] Dept. of Electronic & Computer Engineering, Technical University of Crete, Chania, Crete, Greece
{gchatzivasilis, kfysarakis}@isc.tuc.gr
[3] Dept. of Industrial Informatics, Technological Educational Institute of Kavala, Kavala, Greece
krantos@teikav.edu.gr

Abstract. As computing becomes pervasive, embedded systems are deployed in a wide range of domains, including industrial systems, critical infrastructures, private and public spaces as well as portable and wearable applications. An integral part of the functionality of these systems is the storage, access and transmission of private, sensitive or even critical information. Therefore, the confidentiality and integrity of the resources and services of said devices constitutes a prominent issue that must be considered during their design. There is a variety of cryptographic mechanisms which can be used to safeguard the confidentiality and integrity of stored and transmitted information. In the context of embedded systems, however, the problem at hand is exacerbated by the resource-constrained nature of the devices, in conjunction with the persistent need for smaller size and lower production costs. This paper provides a comparative analysis of lightweight cryptographic algorithms applicable to such devices, presenting recent advances in the field for symmetric and asymmetric algorithms as well as hash functions. A classification and evaluation of the schemes is also provided, utilizing relevant metrics in order to assess their suitability for various types of embedded systems.

1 Introduction

Embedded computer systems pervade our lives in various forms, from avionics to e-textiles, automobiles, home automation and wireless sensor nodes. Physically, Em-bedded Systems (ESs) range from miniature wearable nodes to large industrial installations of Programmable Logic Controllers (PLCs).

The security, (i.e. confidentiality, integrity and availability) of networked computer systems is not a novel concern but, in the context of ESs, their various intrinsic and often application specific characteristics render security techniques

developed for personal and enterprise systems unsatisfactory or even inapplicable. Such characteristics habitually include resource constraints (namely computational capabilities, memory and power), dynamically formulated and remotely managed or even unmanaged networking as well as operation in hostile environment and time-critical applications.

An additional differentiating factor of ES security is that applications often include direct interaction with the physical world. Consequently, a security incident might lead to asset damage or even personal injury and death. Furthermore, since ESs are often responsible for vital, time-critical applications where a delay or a speed-up of even a fraction of a second could have dire consequences. Mechanisms used to appropriately fortify embedded systems are based on robust cryptographic algorithms. However, the inherent limited capabilities of these resource-constrained devices dictate the use of light schemes.

This paper focuses on the design and implementation aspects of cryptographic mechanisms utilized in resource constrained embedded systems. Similar works on LWC were first carried out in 2007 [47, 48]. In [47], the authors evaluate hardware and software implementations for lightweight symmetric and asymmetric cryptography. In [48], the authors investigate lightweight hardware and software solutions for Wireless Sensor Networks (WSNs). In [49], the authors report new trends for lightweight hardware block and stream ciphers. In [50], hardware implementations of block ciphers are examined while in [51], the authors implement and evaluate 12 lightweight block ciphers. Cryptanalytic attacks on lightweight block ciphers were considered in [52].

2 Lightweight Cryptographic Mechanisms

Embedded devices often have inherent limitations in terms of processing power, memory, storage and energy. The cryptographic functionality that ESs utilize to provide tamper resistant hardware and software security functions has direct impact on the system's:

- Size: Memory elements constitute a significant part of the module's surface.
- Cost: Directly linked to the surface of the component.
- Speed: Optimized code provides results faster.
- Power Consumption: The quicker a set of instructions is executed, the quicker the module can return to an idle state or be put in sleep mode where power consumption is minimal.

Traditional cryptography solutions focus in providing high levels of security, ignoring the requirements of constrained devices. Lightweight cryptography (LWC) is a research field that has developed in recent years and focuses in designing schemes for devices with constrained capabilities in power supply, connectivity, hardware and software. Schemes proposed include hardware designs, which are typically considered more suitable for ultra-constrained devices, as well as software and hybrid implementations for lightweight devices.

- Hardware designs implement the exact functionality without redundant components. The main design goal is the reduction of the logic gates that are required to materialize the cipher. This metric is called Gate Equivalent (GE) [12]. A small GE predisposes that the circuit is cheap and consumes little power. For constrained de-vices an implementation including up to 3000 GE can be considered acceptable while for even smaller devices, like 4-bit microcontrollers, implementations of 1000 GE are being studied [12]. Energy consumption and power constraints are other significant factors. Energy consumption is important when a device is running on batteries while power constraints affect passive devices, like passive RFID tags, that must be connected to a host device to operate. Security attacks and relevant countermeasures that are correlated to power analysis are also considered in hardware designs.
- Software implementations typically only require a microprocessor to operate. The main design goals are the reduction of memory and processing requirements of the cipher. Implementations are optimized for throughput and power savings. Portability is their main advantage over hardware implementations.
- Hybrid schemes combine the two approaches exploiting the best features from both. Hardware implements the basic cipher functionality and software performs the data and communication manipulation. A common practice is the design of cryptographic co-processors. The throughput is mostly affected by the communication bandwidth between hardware and software components. Hybrid implementations target on specific communication applications, like RFID tags, portable devices and Internet servers.

2.1 Symmetric Cryptography

Lightweight and ultra-lightweight ciphers usually offer 80 to 128 bit security [12]. 80 bit security is considered adequate for constrained devices [23], like 4-bit micro-controllers and RFID tags, while 128 bits is typical for mainstream applications [1]. For one way authentication, 64 to 80 bit security would suffice [21].

Three main approaches are followed in implementing lightweight ciphers. In the first case, researchers try to improve the performance of well-known and well-studied ciphers such as AES and DES. A state of the art AES [1] hardware implementation uses 2400 GE and is used as a benchmark for newer ciphers. In the second case, re-searchers design and implement new ciphers, specific for this domain. PRESENT [2] is such an example implemented for lightweight and ultra-lightweight cryptography and is one of the first ciphers that offer a 1000 GE implementation for ultra-constrained devices. In the third case, researchers mix features of several ciphers that are well studied and their individual properties are known.

The absence of decryption is another factor that can reduce the requirements of such ciphers, especially for ultra-lightweight cryptography. Hummingbird-2 [13] is a combination of cipher and protocol and adopts this strategy. This

approach is suitable for devices that need only one way authentication. Furthermore, some ciphers like KTANTAN [14] propose that the key should be hard-wired on the device to further reduce the GE due to the absence of key generation operations.

Block Ciphers. DES [15] is a traditional block cipher that can be used in constrained devices al-though due to its small key sizes, the security level is low. DESL is a lightweight version of the cipher that achieves 20% size reduction, DESX uses key whitening to increase the security level, while DESXL is the combination of the two variants [15].

Other traditional ciphers that are investigated in this field are AES [1], Camellia [62], CLEFIA [16] and IDEA [51]. Camellia is approved for use by the ISO/IEC and the projects NESSIE and CRYPTREC. The hardware implementation exceeds the 3000 GE bound while the software implementation is fast. CLEFIA is a 128-bit block size cipher and uses 128-, 192- and 256-bits keys. It was designed by SONY and is highly efficient both in hardware and software. It is standardized in ISO 29192-2. IDEA is used in PGP v2.0 and performs well in embedded software.

PRESENT [12] is a milestone in LWC and the comparison unit for lightweight ciphers. It is 128-bit block size cipher and uses 80- and 128-bits keys. It is standardized in ISO 29192-2 and is efficient in both hardware and software. PRESENT's novelties include the replacement of 8 distinct S-Boxes with a carefully selected single one and a fully wired diffusion layer without any algebraic unit.

Hummingbird-2 [13] is a promising ultra-lightweight cipher with a hybrid structure of block and stream cipher. It can optionally produce a message authentication code (MAC) for each message processed and form a one way authentication protocol. It encrypts data in high rates and its performance is better than PRESENT's. Two main drawbacks are the initialization process and the decryption function. In more detail, an initialization process is necessary before en/decryption for its stream property and, moreover, the performance decreases if many small messages are processed. Also, the encryption and decryption operations are different, therefore the en/decryption implementation is 70% larger than the encryption-only version.

The KATAN and KTANTAN family [14] produces low hardware footprint. KATAN uses a very simple key schedule mechanism and achieves 802 GE. KTANTAN is proposed for devices where the key is initialized once and remains unchanged, achieving 462 GE.

SEA [17] supports a scalable software implementation with low-cost encryption routines. It is parameterized in text, key, and processor size and can produce low memory requirements, small code size and a limited instruction set.

Newer lightweight block ciphers include TWINE [18], Klein [21], LED [20], LBlock [19], PUFFIN-2 [22], Piccolo [23], NOEKEON [51] and ITUbee [53]. TWINE, Klein, LED and LBlock balance tradeoffs between hardware and software implementations. TWINE achieves a good overall status as PRESENT.

PUFFIN-2 is faster and more lightweight in hardware than PRESENT for en/decryption implementations. Piccolo is the most lightweight block cipher in hardware and it requires 683 and 758 GE for 80 and 128 bit key size respectively. NOEKEON is reported in LWC for its compact and efficient software implementation. ITUbee is designed for lightweight software and achieves the best overall status in this domain.

SIMON and SPECK [54] have been designed by NSA. Although the ciphers are not publicly released, a performance evaluation was presented during the MIT 2013 Legal Hack-a-Thone. Both ciphers perform well in software and hardware. SIMON is better in hardware and SPECK is better in software. Nevertheless, Piccolo and ITUbee achieve a better overall status in hardware and software respectively.

Furthermore, domain specific ciphers include EPCBC [24] and PRINTcipher [25]. EPCBC is based on PRESENT and targets in Electronic Product Code (EPC) encryption applications. EPC aims to replace bar codes with low-cost passive RFIDS and is an industry standard by EPCglobal. PRINTcipher targets EPC and Integrated Circuit (IC) printing (i.e. used for the production and personalization of circuits).

Stream Ciphers. Stream ciphers are an alternative type of symmetric key ciphers and also well suited to constrained devices. Despite the evolution effort in the field of lightweight stream ciphers, they remain inferior to lightweight block ciphers. Their major draw-back is the lengthy initialization phase prior to first usage. Moreover, there are communication protocols that can't utilize stream ciphers. However, they are still in the foreground due to their simplicity and speed in hardware. They are often used in applications where the plaintext size is unknown.

Traditional stream ciphers RC4, A5/1 and E0 are considered insecure and should not be used in new applications [77]. AES in CTR mode is currently the only secure and widespread solution for stream encryption [77].

As for newer stream ciphers, the most notable are the finalists of the eSTREAM project [26]. eSTREAM was part of the ECRYPT Network of Excellence, targeted to deliver a small portfolio of promising stream ciphers. They considered two profiles of ciphers for different applications. Profile 1, includes ciphers for fast throughput in software, which are faster than the 128-bits AES-CTR. The finalist ciphers are the HC-128 [66], Rabbit [65], Salsa20 [66] and SOSEMANUK [66]. Profile 2, includes ciphers that are suitable for highly constrained environments and are more compact in hardware than the 80 bits AES. The finalists are Grain [3], Trivium [4] and MICKEY 2.0 [56]. All finalists are well-cryptanalyzed and are found secure against all attacks that are faster than the exhaustive key search attack.

In software, Salsa20/12 is reported as the most suitable for constrained devices. It uses 256-bit keys and 128-bit initialization vectors. The cipher utilizes only simple operations of addition, modulo 2^{32}, bit rotation and bitwise XOR,

which are efficient-ly implemented in software. Furthermore the encryption and decryption operations are identical.

In hardware, Grain [3] and TRIVIUM [4] are the more accepted ones and have been reported as the most suitable for constrained devices. The key size of Grain is 80 bits, while the related Grain-128 supports 128-bit keys, and the IV 64 bits and it re-quires about 1300 GE to implement. TRIVIUM comes up with an 80 bit secret key, an 80 bit IV and about 2600 GE to implement. It was designed as an exercise in ex-ploring how far a stream cipher can be simplified without sacrificing its security, speed or flexibility. It is standardized in ISO 29192-1, as is ENOCORO [5] which has an equivalent GE size in hardware.

Several newer ciphers are proposed based on eSTREAM candidates, like BEAN [45], QUAVIUM [46] and WG-7 [27]. BEAN is based on Grain. It supports binary output production without the need of additional hardware that was needed in Grain. The weak output function leads to an efficient distinguisher and a state-recovery attack [61]. In software, it takes less time to generate the keystream while using the same amount of memory. QUAVIUM is a scalable extension of TRIVIUM. It uses four TRIVIUM-like shift registers in coupling connection instead of three shift regis-ters in series connection of the original TRIVIUM. The hardware implementation is larger than TRIVIUM and the software implementation is faster.WG-7 is the new version of WG cipher that was candidate in eSTREAM. It produces larger throughput than the other candidates and requires less memory. The key size is 80 bits, while the IV is 81 bits and is parameterized for RFID tags [27].

A2U2 [28] is a domain specific stream cipher. It was designed for the extremely resource limited environment of printed electronic RFID tags and is based on the principles of KATAN for efficient hardware. The smaller version requires less than 300 GE.

Hash Functions. Hash functions are another research field of LWC. The standardized or widely-used MD5 (8001 GE) [30], SHA-1 (5527 GE) [29] and SHA-2 (10868 GE) [30] and ARMADILLO (4353 GE) [70] are too large to fit in hardware constrained devices (i.e. more than 3000 GE). After the release of the PRESENT cipher there were many efforts to build novel lightweight hash functions based on PRESENT design principles [32], like C-PRESENT (4600 GE), H-PRESENT (2330 GE) and PRESENT-DM (1600 GE).

The NIST's SHA-3 competition [33] in 2012 defined a new function to replace the older SHA-1 and SHA-2. The finalists [34] were BLAKE, Grostl, JH, Skein and Keccak, with the latter being the winner. Unfortunately, the SHA-3/Keccak and the other finalists aren't much more compact than the previous SHA functions. At this time, all SHA-3 finalists require more than 12000GE for 128 bit security. The SHA-3 competition has helped our understanding of hash functions significantly and led to a new design trend of hash functions with sponge constructions. Keccak is such a function and a lightweight implementation of a constrained Keccak version [35] was later announced at 2520 5090 GE.

Other new lightweight hash functions with sponge constructions are SQUASH (6328 GE) [36], GLUON (2071 GE) [37], Quark (1379 GE) [38], Photon (1120 GE) [39] and Spongent [40]. Spongent is the most lightweight hash function family known so far. Its smallest implementations require 738, 1060, 1329, 1728 and 1950 GE for 88, 128, 160, 224 and 256 bit respectively. It is based on a sponge construction instantiated with a PRESENT-type permutation, following the hermetic sponge strategy.

All the state of the art ciphers and hash functions that are mentioned should be extensively tested for security vulnerabilities before being widely used.

2.2 Asymmetric Cryptography

Asymmetric algorithms and protocols must also be adapted to operate on devices with the aforementioned resource limitations. This is an elaborate task, since asymmetric ciphers are computationally far more demanding than their symmetric counterparts and are usually used with powerful hardware. The performance gap is wider on constrained devices such as 8-bit microcontrollers. Even an optimized asymmetric algorithm e.g. elliptic-curve cryptography (ECC) is 100 to 1000 times slower than a standard symmetric algorithm like AES which correlates to two or three orders-of-magnitude higher power consumption.

Traditional Asymmetric Cryptosystems. Traditional public key cryptography is based on one-way trapdoor functions. These functions are based on a set of hard mathematical problems. There are three well established cryptosystems:

1. RSA, Rabin [42] based on the Integer Factorization Problem (FP)
2. ECC [6] / HECC [7] based on Elliptic Curve Discrete Logarithm Problem (ECDLP)
3. ElGamal [74] based on the Discrete Logarithm Problem In Finite Fields (DLP)

RSA is the most popular algorithm for asymmetric cryptography and supports key sizes from 1024 to 4096 bits. As such, it is used as a benchmark for the various public key cryptosystems researchers propose. However its large hardware footprint and its resource demanding implementations led researchers to seek for other algorithms for applications in constrained devices.

Rabin is quite similar to RSA. One main difference is the complexity of the factorization problems that they rely upon. Rabin is proven to be as hard as the integer factorizations problem, while RSA is not. Also, the encryption for Rabin is faster but the decryption is less efficient. WIPR [42] is a low-resource implementation of Rabin in hardware. The implementation shares several architectural principles with the SQUASH hash function. It requires 4682 GE and fits on RFID tags and wireless sensor nodes. BluJay [43] is a hybrid Rabin-based scheme that is suitable for lightweight platforms and is based on WIPR and Hummingbird-2. The encryption is significantly faster and more lightweight

than RSA and ECC for the same level of security. The hardware implementation requires less than 3000 GE.

ECC [6] and HECC [7] are considered the most attractive cryptosystems for em-bedded systems. They present smaller operand lengths and relatively lower computational requirements. Their main advantage is the fact that for the same level of security they offer shorter keys compared to RSA, which leads to smaller internal state requirements. As the level of security increases, RSA key sizes grow much faster than ECC. ECC also produces lightweight software implementations due to its memory and energy savings. The most known software implementations [8] are the TinyECC and the WMECC.

HECC is a generalization of elliptic curves. A hyper elliptic curve of genus 1 is an elliptic curve. As the genus increases, the arithmetic of encryption gets more complicated, but it needs fewer bits for the same level of security. HECC's operand size is at least a factor of two smaller than the ECC one. The curves of genus 2 are of great interest for the research community as higher genus curves suffer from security at-tacks. HECC has better performance than ECC and is more attractive in resource constrained devices.

ElGamal [74] is of no interest for resource constrained platforms. The computation is more intensive than RSA and encryption produces a 2:1 expansion in size from plaintext to ciphertext. It is also considered vulnerable to some types of attacks, like chosen ciphertext attacks.

Alternative Asymmetric Cryptosystems. Alternative public key cryptosystems (APKCs) [9] that are based on other mathematical features have become popular due to their performance and their resistance against quantum computing. These alternative cryptosystems are based on:

- Hash-Based Cryptography The Merkle signature scheme (MSS) [41] is a crypto-system which uses typical hash functions
- Lattice-Based Cryptography NTRU [10] is the most popular scheme which is based on the Shortest Vector Problem
- Code-Based Cryptography McEliece [72] is a popular scheme based on error-correcting codes
- Multivariate-Quadratic (MQ) Cryptography MQ [73] is based on the problem of solving multivariable quadratic equations over finite fields

An MSS implementation with the AES-based hash function [41] has smaller code size and faster verification process than RSA and ECC. Moreover, the signature generation is faster than RSA and comparable to ECC. MSS may gain ground in lightweight asymmetric cryptosystems due to the evolution of lightweight hash function design.

NTRU [10, 11] is the most promising cryptosystem of all APKCs. Encryption and decryption use only simple polynomial multiplications, which makes them very fast compared to traditional cryptosystems. NTRU is highly efficient, suitable for embedded systems and provides a level of security comparable to RSA and ECC. In hard-ware implementations [11], NTRU is 1,5 times faster

compared to ECC for the same level of security and only has 1/7 of its memory footprint. The hardware implementation requires almost 3000 GE. In software implementations [10], NTRU is 200 times faster in key generation, almost 3 times faster in encryption and about 30 times faster in decryption compared to RSA. On the other hand, NTRU produces larger output, which may impact the performance of the cryptosystem if the number of transmitted messages is crucial. It is considered safe when the recommended parameters are used [76]. NTRU can be efficiently used in embedded systems because of its easy key generation process, its high speed and its low memory usage. The system is now adopted by the IEEE P1363 standards under the specifications for lattice-based public-key cryptography as well as IEEE P1363.1 and ANSI X9.98 Standard for use in the financial services industry.

The main drawback of McEliece [72] and MQ [73] cryptosystems is the use of large keys. In comparison to 1924 bit RSA, MQ requires 9690 bytes for the public key and 879 bytes for the private key. Key sizes impact on the computations that are performed, the speed, the key storage and the output's size. The advantage of these systems is the fast encryption and decryption process that makes them suitable for high performance applications where messages must be assigned in real time.

3 Evaluation

We analyze the features of different cryptographic solutions and propose the more suitable ones for different types of embedded devices. Based on the devices' capabilities we categorize the solutions in four groups: ultra-lightweight, low-cost, lightweight and specific domain. Ultra-lightweight implementations fit in the most constrained devices (in computation capability, memory, power), like the standard 8051 microcontroller and the ATtiny45. Low-cost devices (e.g. ATmega128) are cheap and perform a little better than ultra-lightweight ones. Lightweight devices include the rest of the devices reported in LWC. As specific domains we consider the EPC encryption applications and IC-printing.

3.1 Hardware implementations

The hardware implementations are categorized based on chip area. Ultra-lightweight implementations occupy up to 1000 logic gates, low-cost implementations occupy up to 2000 logic gates and lightweight implementations occupy up to 3000 logic gates. The best implementations in each group are selected based on the figure of merit (FOM) metric [70]. FOM is considered as a fair metric to compare the energy efficiency of different implementations; the higher the value, the better.

$$FOM = throughput\,[Kbps]/area\ squared\,[GE^2] \tag{1}$$

Block ciphers are better than stream ciphers in the three general groups of devices. Hash functions perform efficiently in low-cost and lightweight devices. Asymmetric cryptography is feasible only in lightweight devices. For ultra-lightweight and low-cost devices the key establishment mechanisms based only

on symmetric cryptography can be applied. For the domain specific applications, the PRINTcipher achieves a better overall status than EPCBC and A2U2.

For ultra-lightweight devices the block ciphers PRINTcipher, KTANTAN, Piccolo, SIMON, SPECK and LED, and the stream ciphers A2U2 and TRIVIUM, as well as the hash functions PHOTON and Spongent are implemented. For block ciphers, PRINTcipher is considered insecure for wide use as it is designed for a specific application domain and ignores several types of general attacks. SIMON and SPECK have not been released and the authors cannot reason about their security. KTANTAN is only appropriate in applications where the key is hardcoded on the device. LED consumes high energy per bit and is inefficient. Piccolo achieves a good overall status and is the most suitable cipher in this category. Regarding stream ciphers, A2U2 achieves the best FOM but as a new cipher it is not extensively cryptanalyzed, therefore the standardized TRIVIUM appears to be the optimal choice. For hash functions, Spongent is the most lightweight choice but PHOTON produces higher FOM.

Targeting low-cost devices, the block ciphers PRESENT, TWINE, KATAN, Klein, DESL, EPCBC, LBlock, PUFFIN-2, the stream cipher Grain, and the hash functions DM-PRESENT, D-QUARK, and U-QUARK are implemented (as well as the ones for ultra-lightweight devices). For block ciphers, PRESENT is standardized and is considered the best solution. TWINE performs similar to PRESENT but is a new cipher. KATAN, Klein, DESL, EPCBC, LBlock and PUFFIN-2 have worse performance. For stream ciphers, Grain performs better than the ultra-lightweight TRIVIUM. For hash functions, Spongent is the best while DM-PRESENT, D-QUARK, and U-QUARK produce worse FOM metrics.

For lightweight devices, the block ciphers Hummingbird-2, AES, DESXL, DESX and CLEFIA, the stream cipher QUAVIUM, the hash functions H-PRESENT, Keccak, S-QUARK and SQUASH, and the asymmetric cryptosystems NTRUencrypt and GPS-4/4-F are implemented; in addition to the ones for ultra-lightweight and low-cost devices. For block ciphers, AES is the best choice. The standardized PRESENT and CLEFIA are also appropriate. The variants DESXL and DESX can also be applied as they offer higher level of security than DES. Hummingbird-2 is another promising candidate. For stream ciphers TRIVIUM achieves higher FOM than Grain while the new cipher QUAVIUM isn't well cryptanalyzed. The hash function DM-PRESENT achieves by far the best FOM for all the relevant proposals. Keccak as the new SHA-3 function can also be used. S-QUARK and SQUASH produce poor performance. For the asymmetric cryptosystems, NTRU appears to be the most suitable.

Table 1 summarizes the best hardware implementation of each examined cipher that requires less than 3000 GE. The implementations are sorted by the FOM metric.

3.2 Software implementations

The software implementations are categorized based on the ROM and RAM requirements. Ultra-lightweight implementations require up to 4 KB ROM and 256 bytes RAM, low-cost implementations require up to 4 KB ROM and 8

KB RAM and lightweight implementations require up to 32 KB ROM and 8 KB RAM. The best implementations in each group are selected based on the combined metric (CM) [51]. CM indicates the tradeoff between implementation size and performance and smaller values are better.

$$CM = (code\ size\ [bits] * encryption\ cycle\ count\ [cycles])/block\ size\ [bits] \quad (2)$$

Again, the block ciphers are more efficient than stream ciphers in the three general groups of devices. In software, asymmetric cryptography materializes specific key exchange schemes and communication protocols, like SSL. Due to the complexity of implementing this functionality, cryptographic libraries are utilized to enhance the robustness of an application. Hash functions are embodied to these schemes and protocols. Compact libraries, like CyaSSL [71] which is specifically designed for em-bedded devices, are suitable for lightweight implementations. Thus, individual primitive implementations are mainly proposed for block and stream ciphers.

For ultra-lightweight devices, the block ciphers Camellia, SEA, Hummingbird-2, AES, NOEKEON, IDEA, Klein, PRESENT, TWINE, KATAN, DESL and DESXL, and the stream ciphers Rabbit, WG-7, TRIVIUM and Grain are implemented. For block ciphers, Camellia achieves the best CM by far. AES is also efficient in this domain. SEA, Hummingbird-2 and NOEKEON achieve a good overall status. IDEA, Klein, PRESENT, TWINE, KATAN, DESL and DESXL perform poor in such devices. For stream ciphers, Rabbit is the best proposal. WG-7 performs well, but as a new cipher, it isn't well cryptanalyzed. TRIVIUM and Grain have poor performance.

For low-cost devices, the block ciphers ITUbee and Hummingbird, and the stream cipher Salsa20 are implemented (in addition to the ones for ultra-lightweight devices). All ciphers perform well but achieve lower CM metric than the proposed ones in ultra-lightweight devices. For lightweight devices, the block ciphers DES and DESX, and the stream ciphers HC128 and AES in CTR mode are additionally implemented. All ciphers perform poor in such devices and are inferior to the proposals in ultra-lightweight and low-cost devices.

Table 2 summarizes the best software implementation of each examined cipher that requires less than 32 KB ROM and 8 KB RAM. The implementations are sorted by the CM metric.

4 Conclusions

The aim of this paper was to provide a comparative analysis on lightweight cryptographic algorithms designed for resource-constrained devices. The inherently limited capabilities of these systems in terms of computing power, memory, storage and energy resources, inevitably limit the effectiveness and the applicability of well-established cryptographic mechanisms designed for systems where such resource constraints are not a significant concern.

Such an extensive analysis is considered essential to those planning on utilizing such mechanisms in newly designed systems or applications running on

resource constrained devices. As demonstrated in this work, there is on-going research on various aspects of lightweight cryptography. The evaluation of the robustness and efficiency of pre-existing as well as newly proposed schemes poses a major challenge to research and development efforts. Overcoming the aforementioned challenges, how-ever, is necessary for realizing the ubiquitous computing future.

Acknowledgement

This work was funded by the General Secretarial Research and Technology (G.S.R.T.), Hellas under the Artemis JU research program nSHIELD (new embedded Systems arcHItecturE for multi-Layer Dependable solutions) project. Call: ARTEMIS-2010-1, Grand Agreement No: 269317.

References

1. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.: Pushing the Limits: a Very Compact and a Threshold Implementation of AES. In: Advances in Cryptology EUROCRYPT 2011 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, vol. 6632, pp. 69 (2011)
2. Poschmann, A.: Lightweight Cryptography Cryptographic Engineering for a Pervasive World. PhD Dissertation, Faculty of Electrical Engineering and Information Technology, Ruhr-University Bochum, Germany (2009)
3. Hell, M., Johansson, T., Meier, W.: Grain a Stream Cipher for Constrained Environments. International Journal of Wireless and Mobile Computing, vol. 2, No 1/2007, pp. 86- 93 (2007)
4. De Canniere, C., Prenel, B.: Trivium Specifications. eStream Project, http://www.ecrypt.eu.org/stream/triviump3.html (2008)
5. Watanabe, D., Ideguchi, K., Kitahara, J., Muto, K., Furuichi, H.: Enocoro-80: A Hardware Oriented Stream Cipher. In: Third International Conference on Availability, Reliability and Security (ARES 08), vol., no., pp.1294,1300, 4-7 March 2008 (2008)
6. Hein, D., Wolkerstorfer, J., Felber, N.: ECC is ready for RFID a Proof in Silicon. Selected Areas In Cryptography. LNCS, vol. 5381/2009, pp. 401-413 (2009)
7. Roman, R., Alcaraz, C., Lopez, J.: A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes. Journal of Mobile Networks and Applications, vol. 12, issue 4, August 2007 (2007)
8. Nizamuddin, N., Ch, S. A., Nasar, W., Javaid, Q.: Efficient Signcryption Schemes based on Hyperlliptic Curve Cryptosystem. In: 7th International Conference on Emerging Technologies (ICET), pp. 1-4 (2011)
9. Guneysu, T., Heyse, S., Paar, C.: The Future of High-Speed Cryptography: New Computing Platforms and New Ciphers. In: Proceedings of the 21st edition of the Great Lakes Symposium on VLSI (GLSVLSI'11) (2011)
10. Shen, X., Du, Z., Chen, R.: Research on NTRU Algorithm for Mobile Java Security. In: International Conference on Scalable Computing and Communications; The Eighth International Conference on Embedded Computing 2009, SCALCOM-EMBEDDEDCOM'09, pp 366-369 (2009)

11. Kamal, A. A., Youssef, A. M.: An FPGA Implementation of the NTRUEncrypt Cryptosystem. In: 2009 International Conference on Microelectronics (ICM), pp. 209-212 (2009)
12. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Proceedings of Workshop Cryptographic Hardware and Embedded Systems (CHES 07) (2007)
13. Engels, D., Saarinen, M.-J. O., Schweitzer, P., Smith, E. M.: The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. The 7th Workshop of RFID Security and Privacy (RFIDSec 2011), Amherst, Massachusetts, USA(2011)
14. Canniere, C., Dunkelman, O., Knezevic, M.: KATAN & KTANTAN A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gai, K., (eds.) CHES 2009. LNCS, vol. 5747, pp. 272-288. Springer, Heidelberg (2009)
15. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New Lightweight DES Variants. In: Proceedings of Fast Software Encryption 2007 FSE 2007, LNCS, vol. 4593, pp. 196 210. Springer, Berlin (2007)
16. Akishita, T., Hiwatari, H.: Very Compact Hardware Implementations of the Blockcipher CLEFIA. Sony Corporation, Technical Paper, June 2011, http://www.sony.co.jp/Products/cryptography/clefia/download/data/clefia-hwcompact- 20110615.pdf (2011)
17. Standaert, F.-X., Piret, G., Gershenfeld, N., Quisquater, J.-J.: SEA: A Scalable Encryption Algorithm for Small Embedded Applications. Smart Card Research and Advanced Applications. In: Proceedings of CARDIS 2006, LNCS, vol. 3928, pp. 222-236. Springer, Verlag (2006)
18. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A Lightweight, Versatile Block Cipher. ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146-169, November 28-29 (2011)
19. Wu, W.,Zhang, L.: LBlock: A Lightweight Block Cipher. Applied Cryptography and Network Security, LNCS, vol. 6715/2011, pp. 327-344. Springer(2011)
20. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED Block Cipher. Cryptographic Hardware and Embedded Systems CHES 2011, LNCS, Vol. 6917/2011, pp. 326-341. Springer (2011)
21. Gong, Z., Nikova, S., Law, Y.-W.: KLEIN: A New Family of Lightweight Block Ciphers. In: Proceedings of the 7th Workshop on RFID Security and Privacy, LNCS, available via http://rfid-cusp.org/rfidsec/. Springer (2011)
22. Wang, C., Heys, H. M.: An Ultra Compact Block Cipher for Serialized Architecture Implementations. In: Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2009), St. John's, Newfoundland, May 2009. (2009)
23. Shibutani, K., Isobe, T., Hiwarati, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An Ultra- Lightweight Blockcipher. Cryptographic Hardware and Embedded Systems CHES 2011, LNCS, vol. 6917/2011, pp. 342-357. Springer (2011)
24. Yap, H., Khoo, K., Poschmann, A., Henricksen, M.: EPCBC A Block Cipher Suitable for Electronic Product Code Encryption. CANS 2011, LNCS, vol. 7092, pp. 76-97. Springer- Verlag Berlin Heidelberg(2011)
25. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M. J. B.: PRINTcipher: A Block Cipher for IC-Printing. Cryptographic Hardware and Embedded Systems CHES 2010, LNCS, vol. 6225/2010, pp. 16-32. Springer (2010)
26. eSTREAM Web Page, http://www.ecrypt.eu.org/stream
27. Luo, Y., Chai, Q., Gong, G., Lai, X.: A Lightweight Stream Cipher WG-7 for RFID Encryption and Authentication. IEEE Global Telecommunications Conference 2010 (GLOBECOM 2010), pp. 1-6 (2010)

28. David, M., Ranasinghe, D. C., Larsen, T.: A2U2: A Stream Cipher for Printed Electronics RFID Tags. IEEE International Conference on RFID 2011, pp. 176-183 (2011)
29. O'Neill, M.: Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In: Dominikus, S., Aigner, M. (eds.) RFIDSec 2008, http://events.iaik.tugraz.at/RFIDSec08/Papers/ (2008)
30. Feldhofe, M., Rechberger, C.: A Case Against Currently Used Hash Functions in RFID Protocols. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops, LNCS, vol. 4277, pp. 372-381. Springer, Heidelberg (2006)
31. Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y.: Hash Functions and RFID Tags: Mind the Gap. CHES'08 Proceedings of the 10th international workshop on Cryptographic Hardware and Embedded Systems, pp. 283-299. Springer- Verlag Berlin, Heidelberg (2008)
32. SHA-3 Contest, http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/submissions_rnd3.html
33. Gaj, K., Homsirikamol, E., Rogawski, M., Shahid, R., Sharif, M. U.: Comprehensive Evaluation of High-Speed and Medium Speed Implementations of Five SHA-3 Finalists Using Xilinx and Altera FPGAs. The 3rd SHA-3 Candidate Conference, Washington, D. C., March 22-23 2012 (2012)
34. Kavun, E. B., Yalcin, T.: A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. In: Proceedings of the 6th Workshop on RFID Security (RFIDSec'11), Istanbul, Turkey, June 7-9 2010 (2010)
35. Shamir, A.: SQUASH A New MAC with Provable Security Properties for Highly Constrained Devices such as RFID Tags. Nyberg, K. (ed.) FSE 2008, LNCS, vol. 5086, pp. 144-157 (2008)
36. Berger, T. P., D'Hayer, J., Marquet, K., Minier, M., Thomas, G.: The GLUON Family: A Lightweight Hash Function Family Based on FCSRs. Mitrokotsa, A., Vaudenay, S., (eds.) AFRICACRYPT 2012, LNCS, vol. 7374, pp. 306-323 (2012)
37. Aumasson, J.-P., Henzen, L., Meier, W., Naya-Plasencia, M.: QUARK: A
38. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011, LNCS, vol. 6841, pp. 222-239 (2011)
39. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: SPONGENT: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES 2011, LNCS, vol. 6917, pp. 312-325 (2011)
40. Rohde, S., Eisenbarth, T., Dahmen, E., Buchmann, J., Paar, C.: Fast Hash-Based Signatures on Constrained Devices. In: Proceedings of the 8th Smart Card Research and Advanced Application IFIP Conference CARDIS 2008, September 8-11 (2008)
41. Oren, Y., Feldhofer, M.: WIPR A Low-Resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In: Basin, D. A., Capkun, S., Lee, W. (eds.) WISEC, ACM, pp. 59-68 (2009)
42. Saarinen, M.-J. O.: The BlueJay Ultra-Lightweight Hybrid Cryptosystem. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW), pp. 27-3, May 24-25 (2012)
43. Reddy, A. V.: A Cryptanalysis of the Tiny Encryption Algorithm. Tuscaloosa, the University of Alabama, Master Thesis (2003)
44. Kumar, N., Ojha, S., Jain, K., Sangeeta, L.: BEAN: a Lightweight Stream Cipher. In: Proceedings of the 2nd International Conference on Security of Information and Networks (SIN '09), pp. 168-171 (2009)

45. Tian, Y., Chen, G., Li, J.: QUAVIUM a New Stream Cipher inspired by TRIVIUM. Journal of Computers, vol. 7, no. 5, pp. 1278-1283, May 2012, doi: 10.4304/jcp.7.5.1278-1283. (2012)
46. Eisenbarth, T., Paar, C., Poschmann, A., Kumar, S., Uhsadel, L.: A Survey of Lightweight Cryptography Implementations. IEEE Design and Test of Computers 2007, vol. 24, issue 6, pp 522-533 (2007)
47. Roman, R., Alcaraz, C., Lopez, J.: A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes. Journal of Mobile Networks and Applications, vol. 12, issue 4, August 2007. (2007)
48. Paar, C., Poschmann, A., Robshaw, M. J. B.: New Design in Lightweight Symmetric Encryption. RFID Security, vol. III, pp. 349-371 (2009)
49. Kitsos, P., Sklavos, N., Parousi, M., Skodras, A. N.: A Comparative Study of Hardware Architectures for Lightweight Block Ciphers. Journal of Computers and Electrical Engineering, vol. 38, issue 1, pp. 148-160, January 2012. (2012)
50. Eisenbarth, T., Gong, Z., Guneysu, T., Heyse, S., Indesteege, S., Kerckhof, S., Koeune, F., Nad, T., Plos, T., Regazzoni, F., Standaert, F.-X., Oldenzeel, Loic van Oldeneel: Compact Implementation and Performance Evaluation of Block Ciphers in ATiny Devices. ECRYPT Workshop on Lightweight Cryptography, Louvain-la-Neuve, Belgium (November 2011), and AFRICACRYPT 2012, LNCS, vol. 7374, pp. 172-187. Springer (2012)
51. Anjali, A. P., Saibal, K. P.: A Survey of Cryptanalysis Attacks on Lightweight Block Ciphers. IRACST International Journal of Computer Science and Information & Security (IJCSITS), vol. 2, no 2, April 2012. (2012)
52. Karakoc, F., Demirci, H., Harmanci, A. E.: ITUbee: a Software Oriented Lightweight Block Cipher. May 6 (2013)
53. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: Performance of the SIMON and SPECK Families of Lightweight Block Ciphers. In: MIT 2013 Legal Hack-a-Thone. http://iauth.org/wpcontent/uploads/2013/02/SIMONSPECKperformance_13feb2013.pdf (2013)
54. Mentens, N., Genoe, J., Preneel, B., Verbauwhede, I.: A Low-Cost Implementation of Trivium. In: ECRYPT Workshop, SASC The state of the Art of Stream Ciphers, pp. 197-204 (2008)
55. Good, T., Benaissa, M.: Hardware Performance of eStream Phase-iii Stream Cipher Candidates. In: State of the Art of Stream Ciphers Workshop (SASC 2008), pp. 163173, February (2008)
56. Kitsos, P., Sklavos, N., Parousi, M., Skodras, A. N.: A Comparative Study of Hardware Architectures for Lightweight Block Ciphers. In: Journal of Computers and Electrical Engineering, vol. 38, issue 1, pp. 148-160, January (2012)
57. Zhilyaev, S.: Evaluating a New MAC for Current and Next Generation RFID. Master Thesis, University of Massachusetts Amherst, http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1477&context=theses (2010)
58. Kaps, J.-P., Gaubatz, G., Sunar, B.: Public Key Cryptography in Sensor Networks Revisited. In: Castellucia, C., Hartenstein, H., Paar, C., Westhoff, D. (eds.), Proceeding of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks ESAS 2004. LNCS, vol. 3312, pp. 218. Springer-Verlag (2004)
59. Agren, M.: On Some Symmetric Lightweight Cryptographic Designs. PhD Dissertation, Department of Electrical and Information Technology, Faculty of Engineering, LTH, Lund University (2012)

60. Cakiroglu, M.: Software Implementation and Performance Comparison of Popular Block Ciphers on 8-bit Low-Cost Microcontroller. International Journal of the Physical Sciences, vol. 5, issue 9, pp. 1338-1343, 18 August (2010)
61. Rinne, S., Eisenbarth, T., Paar, C.: Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers. (2011)
62. Bos, J. W., Osvik, D. A., Stefan, D.: Fast Implementations of AES on Various Platforms. SPEED-CC Software Performance Enhancement for Encryption and Decryption and Cryptographic Compilers. (2009)
63. Boesgaard, M., Vesterager, M., Christensen, T., Zenner, E.: The Stream Cipher Rabbit 1. Available via http://www.ecrypt.eu.org/stream/p3ciphers/rabbit/rabbit_p3.pdf. (2010)
64. Meiser, G., Eisenbarth, T., Lemke-Rust, K., Paar, C.: Software Implementation of eSTREAM Profile I Ciphers on Embedded 8-bit AVR Microcontrollers. In: Workshop Record State of the Art of Stream Ciphers (SASC 07) Also submitted in: The eSTREAM Project. (2007)
65. Otte, D.: AVR-Crypto-Lib, http://www.das-labor.org/wiki/AVR-Crypto-Lib/en (2009)
66. Engels, D., Fan, X., Gong, G., Hu, H., Smith, E. M.: Hummingbird: Ultra-Lightweight Cryptography for Resource-Constrained Devices. In: Proceedings of the 14th International Conference on Financial Cryptography and Data Security-FC, LNCS, 2010, vol.6054/2010, pp. 3-18. (2010)
67. Yu, Y., Yang, Y., Fan, Y., Min, H.: Security Scheme for RFID Tag. Technical report, Auto-ID Labs white paper WP-HARDWARE-022
68. Badel, S., Dagtekin, N., Nakahara, J., Ouafi, K., Reffe, N., Sepehrdad, P., Susil, P., Vaudenay, S.: ARMADILLO: A Multi-Purpose Cryptographic Primitive Dedicated to Hardware. In: Mangard, S., Standaert, F.-X. (eds.), CHES 2010, LNCS, vol. 6225, pp. 398412. Springer, Heidelberg (2010)
69. wolfSSL Inc., CyaSSL embedded ssl library, http://yassl.com/yaSSL/Products-cyassl.html
70. Shoufan, A., Wink, T., Molter, G., Huss, S., Strentzke, F.: A Novel Processor Architecture for McEliece Cryptosystem and FPGA Platforms. In: Proceedings of the 20th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP 2009), pp. 98-105 (2009)
71. Yang, B.-Y., Cheng, C.-M., Chen, B.-R., Chen, J.-M.: Implementing Minimized Multivariate PKC on Low-resource Embedded Systems. In: Clark, J. A., Paige, R. F., Polack, F. C., Brooke, P. J. (eds.) Proceedings of the Third international conference on Security in Pervasive Computing (SPC'06), pp. 73-88. Springer, Berlin, Heidelberg (2006)
72. Gamal, T. E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31, 4, 469472(1985).
73. Howgrave-Graham, N., Silverman, J. H., Whyte, W.: Choosing Parameter Sets for NTRUEncrypt with NAEP and SVES-3. In: Menezes, A. (ed.) Proceedings of the 2005 international conference on Topics in Cryptology (CT-RSA'05), pp. 118-135. Springer, Berlin, Heidelberg (2005)
74. Bjorstad, T. E.: An Introduction to New Stream Cipher Designs. In: 25th Chaos Communication Congress (2008)

Table 1. Hardware implementations ($< 3000GE$)

Cipher	**Key Size (bits)**	**Throughput (Kbps at 100 KHz higher is better)**	**GE (lower is better)**	**FOM (higher is better)**
PRINTcipher [25]	80	100	503	3952
Piccolo [23]	80	237.04	1136	1836
PRESENT [12]	80	200	1570	811
TWINE [18]	80	178	1503	787
KTANTAN [14]	80	25.1	688	530
SIMON [54]	96	15.8	763	271
KATAN [14]	80	25.1	1054	226
Klein [21]	64	30.9	1220	208
SPECK [54]	96	12	884	153
HummingBird-2 [13]	128	68.9	2159	147
DESL [15]	56	44.4	1848	130
EPCBC [28]	96	12.12	1008	119
LBlock [19]	80	200	1320	115
AES [1]	128	56.64	2400	98
DESXL [15]	184	44.4	2168	95
DESX [15]	184	44.4	2629	64
CLEFIA [16]	128	39	2488	63
PUFFIN-2 [22]	80	5.2	1083	44
LED [20]	80	3.4	1040	32
A2U2 [28]	56	50	284	620
Grain [56]	80	100	1294	597
TRIVIUM [55]	80	100	2017	245
QUAVIUM [46]	80	-	2372	-
DM-PRESENT [32]	64	387.88	2530	605.98
Spongent [40]	88	17.78	1127	139
PHOTON [39]	80	15.15	1168	111.13
D-QUARK [38]	160	18.18	2819	22.88
H-PRESENT [32]	128	11.45	2330	21.09
U-QUARK [38]	128	11.76	2392	20.56
Keccak-f[400] [35]	128	8	2520	12
S-QUARK [38]	224	3.13	2296	5.93
SQUASH [59]	64	0.2	2646	0.29
NTRUencrypt [60]	57	292.2	2850	359
GPS-4 / 4-F (PRESENT) [2]	80	107.23	2143	233

Table 2. Software implementations ($< 32KBROM, < 8KBRAM$)

Cipher	**Key Size (bits)**	**ROM (bytes - lower is better)**	**RAM (bytes - lower is better)**	**Throughput (Kbps at 4MHz higher is better)**	**CM (lower is better)**
Camellia [62]	128	1262	12	8000	631
SEA [47]	96	2132	0	39	21439
ITUbee [53]	80	400	186	109	21513
AES [64]	128	1912	432	256	29875
TWINE [18]	80	1304	414	118	44173
Hummingbird-2 [13]	128	2227	114	200/172	44400
NOEKEON [51]	128	364	32	21.7	66876
IDEA [51]	128	836	232	31	107765
Klein [21]	80	1268	18	42	120757
PRESENT [47]	80	936	0	23.8	156823
TWINE [18]	80	792	191	13.6	232575
KATAN [51]	80	338	18	3.5	380582
DESL [63]	56	3098	0	30.6	404918
DESXL [47]	184	3192	0	30.4	425483.6
Hummingbird [68]	256	2950	1064	26.5	445081
DES [51]	56	4314	0	29.6	581918
DESX [63]	184	4406	0	29.4	598871
Rabbit [65]	128	1714	216	8421	814
WG-7 [27]	80	1100	0	192	45650
Salsa20 [47]	128	1452	280	111	58181
TRIVIUM [67]	80	424	36	6	281960
Grain [67]	80	778	20	6.4	480026
HC128 [66]	128	23100	4556	189.5/189.6	487446
AES [66]	128	6664	88	40/34	654633