# *A Survey of the Current State of Lightweight Cryptography for the Internet of Things*

Wanican Julian Okello[1, a], Qingling Liu[1, b], Faizan Ali Siddiqui[1, c], Chaozhu Zhang[1, d]

[1]College of Information and Communications Engineering
Harbin Engineering University,
145-1Nantong Street, Nangang District, Heilongjiang, Harbin, 150001, PR China
[a]julian.okello@gmail.com, [b]liuqingling@hrbeu.edu.cn, [c]engr_faizan14@yahoo.com,
[d]zhangchaozhu@hrbeu.edu.cn

***Abstract*— The Internet of things offers a promising future for all stakeholders in technology; from researchers to consumers. It comprises a network of entities i.e. everyday objects; capable of sensing, processing, storing data and communicating with other entities. These entities or "things" can be connected to the internet and monitored or controlled through a service on another module like a mobile device or computer application. This model of ubiquitous computing presents a great challenge in the form of maintaining security i.e. confidentiality, integrity, authentication and non-repudiation of data as majority of the devices run on limited resources. Our paper examines the cryptographic solutions that have so far been developed, presents their strengths and weaknesses against each other and lays out some research gaps. We also present some notable institutions and groups that are researching and developing international standards in this field. The authors goal is that this paper is usable by all levels of people in the field, however, the main target audience are the new researchers on the topic and interested readers to get a complete layout of the field.**

***Keywords*— Lightweight-cryptography, Internet of Things, Symmetric key ciphers, Asymmetric key ciphers, Benchmarking tools, Performance optimization.**

## *I. INTRODUCTION*

Lightweight cryptography was proposed as a concept to manage security in resource limited devices and the IoT is a major sector with such devices. Lightweight cryptography is based on the conventional principles of cryptography but its aim is to design algorithms with a low footprint both in hardware and software. The concept of lightweight is extended to cryptanalysis and maintaining the balance between performance, security and cost of the algorithms. We present some performance statistics of lightweight ciphers and their resistance towards attacks. We conclude the paper by noting some major institutions and groups producing relevant research in this field.

### a. Security Requirements

For any cryptographic algorithm to be accepted as perfectly secure, it must satisfy four conditions;

- Confidentiality: Data is only accessed by sender or receiver
- Integrity: Data is not altered by unauthorized user
- Authentication: Data and user can be verified
- Non-repudiation: User cannot refute connection to sent data

By 2020, predictions from multiple sources like Gartner, Cisco and HP state that IoT devices will have risen to 50bn which is a staggering figure considering that security is still in the early stages of research. The Figure 1 below presents a grouping of sectors in which IoT is implemented hence, the importance of having secure and encrypted communication.

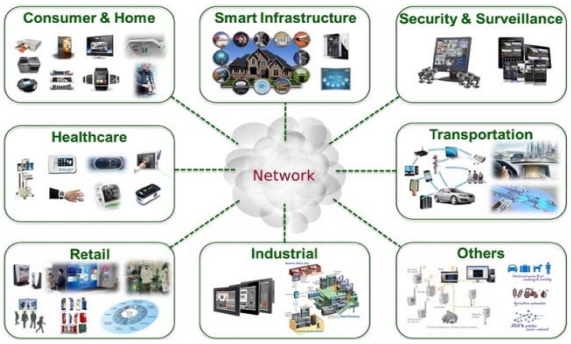


Figure 1: Ecosystem of the IoT Adapted from the Vivante Corporation

### b. Weight of the Algorithm

According to Saarinen and Engels [1], algorithms are weighed as follows;

i. Weight in Software:

- The time complexity of an algorithm is measured according to the number of clock cycles per byte and the latency.
- The memory complexity is measured according to size of RAM necessary to carry out the computation and the ROM space required to store the algorithms e.g. flash memory

ii. Weight in Hardware

- The consumption on physical devices like memory is measured in terms of logical gates with units of GE (Gate Equivalents) which are equivalent to one NAND gate.

978-1-5090-5957-7/17/$31.00 ©2017 IEEE

- Time complexity is measured in terms of throughput rate i.e. bits per second for a particular frequency usually 100Hz.
- Latency is also taken into account just like for the software consumption.
- Power consumption is a metric universal to both hardware and software forms of weighing an algorithm since lightweight devices have limited power. The units used are Watts.

## *II. CRYPTOGRAPHIC SOLUTIONS*

Lightweight Algorithms can be classified according to mode of implementation i.e. hardware and software, or the architecture i.e. symmetric and asymmetric.

### a. Implementation Modes

i. **Software Optimized Algorithms:** Example of such algorithms include; HC-128, Rabbit, SOSEMANUK, Salsa, Speck, Zorro, Robin, Pride, RC5, LEA, SPARX, and RoadRunneR. The main drawback of this approach is that, in case an attacker gains access to the device either physically or over the network, they may be able to alter the software and mount an attack. Also, the software may have to be updated regularly which consumes the already limited bandwidth and processing power of the device.

ii. **Hardware Optimized Algorithms**: These algorithms are designed to optimize hardware resources and are usually easier to build because of the nature of encryption when translated to hardware becomes logic Gates that can be implemented by logic circuits[2]. The advantage of this approach is that even if an attacker gains access to the physical device, it is difficult to modify the state of the cipher without destroying or altering the whole device.

### b. Symmetric and Asymmetric Algorithms

i. **Symmetric Key Algorithms:** Symmetric key algorithms use a common key $k$, to encrypt and decrypt data. This presents a simplified and faster operation however the main drawback is that the key has to be shared in a secure manner between the two users. Symmetric key algorithms are distributed as; block ciphers, stream ciphers and hash functions[3]. Block ciphers encrypt blocks of data at a time while stream ciphers encrypt a single bit of data with one bit of Keystream. Hash functions on the other hand are primarily meant to provide message authentication by generating a unique code i.e. hash value for every message key pair $(m, k)$. Both hash functions and stream ciphers can however be implemented from block ciphers, therefore, block ciphers seem to enjoy a preference on the research scene[4]. Notable examples of symmetric key algorithms are: AES, Simon and Speck from the NSA; stream ciphers; Grain128a, Hummingbird, Snow3G, and Trivium hash functions; Photon, QUARK, and Spongent. A more comprehensive list can be found at[5].

ii. **Asymmetric Key Algorithms:** Require two keys $(Pu_K, Pr_K)$, a public $Pu_K$ key that can be shared over a non-secure channel and a private key $Pr_K$ that does not need to be shared at all. The strength in this approach is that the keys are more secure, however, it can be costly for constrained devices because a host needs to have a global registry of all public keys of other hosts its communicating with. Another shortfall is that most public key systems are proprietary unlike private key systems that are open source. They are also generally slower and more complex than other algorithms therefore less preferred. The most notable example of lightweight asymmetric algorithms is NTRU[6].

**Elliptic Curve Cryptography:** A major part of asymmetric key cryptography is ECC which is based on algebraic representations of elliptic curves over a Finite Field. ECC requires much smaller keys and has a smaller footprint as compared to non-ECC primitives[7]. Some notable schemes include ECDH, ECDSA, ECIES. The elliptic curve E over the field K is defined as the set of solutions to the equation;

$$E: F(x,y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0 \quad (1)$$

$$where\ a_i \in K$$

## *III. OPEN RESEARCH ISSUES*

### a. Cryptanalysis

Cryptanalysis refers to the analysis of cryptographic systems with a goal of revealing hidden aspects like the private key, the state of the cipher and the ciphertext or plaintext. Based on some of the available work[8] there is still a great need for lightweight algorithms to be thoroughly understood in terms of weaknesses and vulnerabilities to attacks. Our work classifies the analysis into two categories; generic attacks and shortcut attacks.

i. **Generic Attacks**: They are independent of the internal state of a cipher and can only be thwarted by carefully choosing a key size, block size or internal state size to make the attack computationally non-feasible. We list some of the generic attacks;

An **exhaustive key attack** also known as brute force attack aims to discover a key $k_i$ such that for a cipher text $c_i$ , this key can reproduce a plain text $p_i$ using the algorithm $E_k(.)$. ie $E_k(p_i) = c_i$. Theoretically, the attacker will try to exhaust all possible keys $k_i \in K$. But according to current standards, any cipher that can computationally be broken by an exhaustive key search is considered insecure. The theoretical limit against this attack is $2^{128}$ which implies that key sizes of 128 and above are theoretically unbreakable using present day technology.

In **table lookup attacks**, the attacker knows the key length $n_k$therefore for all possible keys of that length, he pre-computes a table of cipher text for a like message. When he intercepts a matching

ciphertext, all he has to do is look up the corresponding key. This attack is possible if the attacker can maintain enough memory to store $2^{n_k}$blocks of ciphertext.

**The dictionary attack** completely dismisses key recovery by collecting enough plaintexts and corresponding ciphertext therefore when the attacker recovers a ciphertext, he looks up a corresponding plaintext. This can only work for a large plaintext, ciphertext dictionary and assuming the same key is used for all these pairs.

**A time-memory trade off (TMTO)** attack has five major parameters; N -search space/key size/initial state size, P – Pre-computation time, M -Memory required, T-Time for online phase, D -Data available to the attacker. In block ciphers a TMTO attack is possible through two steps for a cipher represented as; The block cipher represented as a reduced function of encryption is; $f(k) = R(E_k(p_0))$.

The pre-computation Phase $X_{ij} = f(X_{ij-1})$ This equation generates an $m \times t$ matrix of m chains of t elements $for\ j = 1 \dots t, i = 1 \dots m.\ X_{i0} = SP_1$where $SP_n$are starting points for m.

The second phase is the online phase where; the endpoints are also defined where $EP_i = f^t(SP_i)$. The pairs are collected in a table $(SP_i, EP_i)_{i=1}^{m}$. Therefore, for an intercepted ciphertext $c_0$, a value $Y_1 = R(c_0) = f(k)$ where $Y_1$is computed until it matches some endpoint. Eventually a time, memory function is modelled to reveal the key state.

ii. **Non-Generic Attacks**

The two most common non-generic cryptanalytic attacks are linear and differential attacks which manipulate the internal state of the cipher using mathematical models to recreate the key, internal state, plaintext and ciphertext. They are the more difficult to defend against than generic attacks.

**Linear attacks:** Was first developed by Matsui in 1993 to break DES[9] and it uses linear models to create a relationship between the plaintext, ciphertext and the key in order to derive some definitions of the secret key. For a non-linear function $S\{0,1\}^n \rightarrow S\{0,1\}^m$, we consider a set of inputs $X = (x_{n-1} \dots x_0)$ produces an output set $Y = (y_{m-1} \dots y_0) = s(X)\ of\ S(.)$. The attacker tries to find an $n$ bit binary value $\alpha$ and an $m$-bit binary value $\beta$ such that the equation $\alpha.X = \beta.Y$ holds for the bound $^{2n}/_{2}.\alpha\ and\ \beta$ are considered linear approximations and can be used to generate a set of equations that reveal the key state.

**Differential Attacks:** It was invented by Biham and Shamir[10] to reveal weaknesses in the block cipher FEAL. It forwards the argument that by analyzing the difference caused in the output of a system, when a particular difference is applied to the input, the state of the system can be revealed. A difference Δ between two elements $p\ and\ c \in B$ can be expressed as; $\Delta d = \Delta(p, c) = p \otimes c^{-1}$; where $c^{-1}$ denotes the inverse of c with respect to the group operator ⊗.

**Algebraic Attack:** Very recently introduced and exploits the fact that most cryptographic systems can be described as a binary system with multivariate non-linear equations therefore solving the equations reveals the secret key. Solving these equations is deemed NP-hard however some theories like Grobner bases and linearization are being researched to be applied as a solution[11].

**b. Performance optimization**

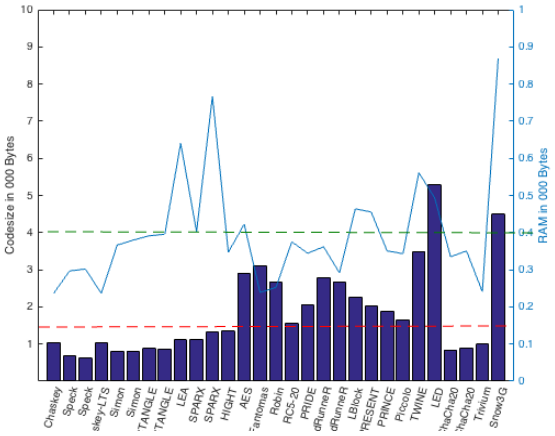


Figure 2: Codesize and RAM Consumption

The previous sections have mentioned a lot about performance of lightweight cryptographic algorithms, however there is still a lot of room for improvement when devices like RFID chips and sensor nodes are placed in the picture. Notable works on performance optimization of lightweight algorithms include [13]. The Figures 2 and 3 represents the average performance of selected lightweight ciphers as extracted from the FELICS[14] tool. The figure 2 shows the average codesize and RAM consumption of lightweight ciphers which when compared to a conventional cipher like AES with codesize x and RAM consumption Y is much better.

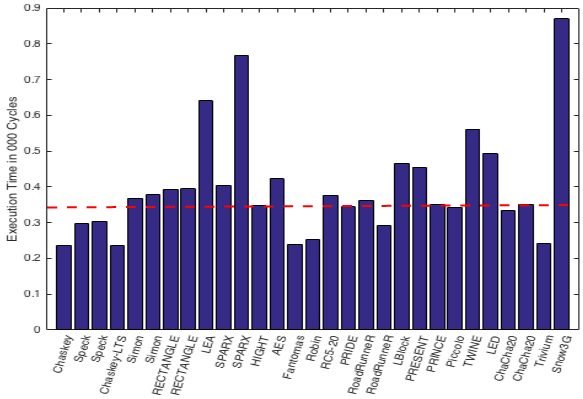


Figure 3: Execution Time of selected Lightweight Ciphers

**c. Benchmarking Tools**

In order to accurately measure and analyze the performance of an algorithm, a consistent

benchmarking tool is required. The most popular tools are briefly discussed here in order of release;

1. **XBX for SUPERCOP:** The eXternal Benchmarking eXtension[15] is an improvement of a previous tool SUPERCOP that could only run on Shell tools with the POSIX standard that cannot be implemented on limited resource devices like PDAs, mobile phones and platforms like FPGAs. XBX also seeks to converge the diversity of results that come from researchers analyzing the same algorithm but on different platforms, implementations and CPUs. Three devices are required, PC, XB Harness (XBH), XB Device. The XBH connects the PC running an XB Software (XBS) to the XBD. XBX tests their implementation on Atmel microcontroller family of devices but theoretical it can be implemented on any hardware platform. The extension is written in shell and Perl scripts as a form of software components. The tool also presents a way for Hardware abstraction from all the platform specific code through the Hardware Abstraction Layer (HAL)

2. **ATHENa**: Automated Tool for Hardware Evaluation[16] and is majorly focused on analyzing good practices for benchmarking hardware based cryptographic tools especially FPGAs. It was inspired by the eBACS project, is written in Perl, and is platform independent with a Perl interpreter that is available for free. The two main hardware vendors analyzed are Xilinx and Altera which accounted for 90% of the market use by time of implementation. Project ATHENa boasts its main merits as automated generation and optimization of results, automated verification of implementations through synthesis and batch tests. The only relative shortfall is that the source code is not revealed which according to the authors protects intellectual property rights but narrows transparency in criticizing the implementations.

3. **BLOC**: The BLOC Project[17] was commenced on October 1st 2011 and funded by the French National Research Agency with an aim of analyzing the performance of lightweight block ciphers on constrained environments. The main metrics considered were; security models and proofs, cryptanalysis, design and security arguments, and performance evaluations. The main shortcoming of this project was an error in computation of the RAM requirement, hence erroneous results for that metric. The project was shut down on March 31st 2016

4. **FELICS**: The Fair Evaluation of Lightweight Cryptographic Systems (FELICS) is the most popular and recent benchmarking tool developed by Dinu [14]. It can be extended for multiple cipher implementations and on their webpage submissions are accepted The tool also provides a grading system for cipher performance in the form of Figure of Merit (FOM) that can be used to compute data presentations from different perspectives.

However, currently there are no tools that universally provide cryptanalysis except statistical test suites like the NIST Test Suite[18] and the FIPS (2001) standard both by NIST and not very new. It is also noted that not much analysis of ECC systems are available.

## IV. NOTABLE RESEARCH GROUPS

Numerous companies, vendors, academic units, governments and individuals have embarked on researching lightweight cryptography, however, only a few are mentioned basing on either their reputation or contributions.

### a. Academic Institutions/Research Groups

1. Crypto Lux: Crypto Lux[19] is a research group under the university of Luxembourg headed by Prof. Alex Biryukov with several staff, post-doctoral and PhD students. Some of their most notable contributions to lightweight cryptography include a recent win for a password hashing competition in 2015. Their presence is also notable at various international conferences and crypto competitions.

2. ECRYPT NET: The European Network of Excellence for Cryptography ECRYPT[20] is a collaboration between six universities, two companies and seven associate companies whose goal is to develop advanced cryptographic algorithms for the internet of things and the cloud. They are arguably one of the most active contributors to this field with projects like eSTREAM and benchmarking tools like eBACS.

### b. Government Institutions

1. NSA: The National Security Agency of the USA is also contributing positively to the lightweight cryptography platform. Simon and Speck[21], the two most recent and popular lightweight block ciphers were developed by two NSA members. In 2014 at the WHSR[22], a group of 20 top officials from the NSA discussed a proper adoption process of lightweight cryptography in order to protect future implementations from disastrous outcomes.

2. NIST LWC-Forum: The National Institute of Standards and Technology, a US based institution has always been very instrumental in researching cyber security and has now formed a team called the Lightweight Cryptography Forum[23] that is researching the need for lightweight crypto algorithms. So far, they have organized a conference and have drafted a report for NISTIR 8114 for lightweight cryptography in August 2016.

### c. International Institutions

1. IEFT: The Internet Engineering Task Force (IETF) is an international body that defines internet standards and considering the IoT will be an IP based technology, some of the most vital standards that will eventually affect cryptographic algorithms are defined by the IETF. Ishaq's paper[24] describes a survey of IEFT standards for the IoT, though it's

important to note that at such an early stage numerous new revisions are expected.

**d. Independent Institutions and Vendors**

1. Google: Google Incorporated is a popular tech company mostly for its web technologies. Recently the project Android Things code named Brillo[25], an OS for IoT devices was developed.

2. Sony: Sony Corporation mostly known for its hardware products and appliances is becoming active in the cryptographic world and recently developed a block cipher called CLEFIA[26]. In 2012, CLEFIA was adopted as an international standard for lightweight cryptography by the ISO/IEC.

3. Intel: Most popularly known for its processors in PCs, this company happens to be very diverse and are now producing microprocessors capable of running lightweight algorithms[27].

## V. CONCLUSION

This survey analyzed lightweight cryptography as a solution to security in the IoT, presenting the different classifications of algorithms and a case for each classification. A few lightweight algorithms were analyzed for performance, while presenting the main cryptanalysis techniques against them. We also presented outstanding research opportunities and some notable groups and institutions to watch and follow in this standard. The main contribution of this paper is a wholesome presentation on the perspective of data security for the IoT with analysis for some algorithms using a very recent and approved benchmarking tool and pointing out influential researchers in this field. This paper can be expanded to analyze other aspects of security like IoT network security and physical device security.

**Acknowledgement**

This paper is funded by the Harbin Engineering University under the Major National Scientific Instrument and Equipment Development Project of China (2013YQ290489).

**References:**

[1] M.-J. O. Saarinen and D. Engels, 'A Do-It-All-Cipher for RFID : Design Requirements ( Extended Abstract )', *IACR Cryptol. ePrint Arch.*, vol. 2012, 2012.

[2] S. S. Mansouri, *Design and Implementation of Efficient and Secure Lightweight Cryptosystems*. 2014.

[3] A. Poschmann, 'Lightweight Cryptography', 2007.

[4] A. . Fallis, 'Understanding Cryptography', *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[5] A. and P. Biryukov, 'Lightweight Cryptography Lounge', 2015. [Online]. Available: http://cryptolux.org/index.php/Lightweight_Cryptography.

[6] 'The NTRU Project'. [Online]. Available: https://tbuktu.github.io/ntru/.

[7] S. Kumar, 'ELLIPTIC CURVE CRYPTOGRAPHY', Ruhr-University Bochum, 2006.

[8] J. Borghoff, 'Cryptanalysis of Lightweight Ciphers', no. December, p. 198, 2010.

[9] M. Matsui, 'Linear Cryptanalysis Method for DES Cipher', *Springer-Verlag*, 1998.

[10] E. Biham and A. Shamir, 'Differential Cryptanalysis of the Data Encryption Standard', no. 1993, 2009.

[11] M. V. M. Walter, 'Algebraic methods in analyzing lightweight cryptographic symmetric primitives', Technical University of Darmstadt, 2012.

[12] M. Knezevic, 'Lightweight Cryptography : from Smallest to Fastest Trade-offs in HW Performance', 2015.

[13] P. Yalla, J. Kaps, and A. Hight, 'Lightweight Cryptography for FPGAs', 2006.

[14] D. Dinu, A. Biryukov, and J. Großsch, 'FELICS – Fair Evaluation of Lightweight Cryptographic Systems'.

[15] C. Wenzel-benner and J. Gr, 'XBX : eXternal Benchmarking eXtension for the SUPERCOP crypto benchmarking framework', pp. 1–13.

[16] K. Gaj, J. Kaps, V. Amirineni, M. Rogawski, E. Homsirikamol, and B. Y. Brewster, 'ATHENa – Automated Tool for Hardware EvaluatioN : Toward Fair and Comprehensive Benchmarking of Cryptographic Hardware using FPGAs'.

[17] M. M. Mickaël Cazorla, Kevin Marquet, 'The BLOC Project', 2013. .

[18] A. Rukhin, J. Soto, and J. Nechvatal, 'A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications', no. April, 2010.

[19] 'CryptoLUX'. [Online]. Available: https://www.cryptolux.org/.

[20] ECRYPT, 'Network of Excellence in Cryptology'. [Online]. Available: http://www.ecrypt.eu.org/ecrypt1/. [Accessed: 17-Jun-2015].

[21] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, 'Simon and Speck: Block Ciphers for the Internet of Things', 2015.

[22] 'Washington Homeland Security Roundtable'. [Online]. Available: http://www.whsroundtable.org/. [Accessed: 19-Apr-2017].

[23] M. S. Turan, 'NIST ' s Lightweight Crypto Project'.

[24] I. Ishaq, D. Carels, G. Teklemariam, J. Hoebeke, F. Abeele, E. Poorter, I. Moerman, and P. Demeester, *IETF Standardization in the Field of the Internet of Things (IoT): A Survey*, vol. 2, no. 2. 2013.

[25] Google, 'Brillo'. [Online]. Available: https://developers.google.com/brillo/. [Accessed: 17-Jun-2015].

[26] T. Shirai, K. Shibutani, and T. Akishita, 'The 128-bit Blockcipher CLEFIA'.

[27] Intel, 'Intel IoT Platform'. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/overview.html. [Accessed: 19-Apr-2017].