Intra-Optimised Lightweight Enciphering Algorithm based on MQTT Protocol for Internet of Things Secure Application

Ronald Chiwariro[1]
Information Security and Cyber Forensics, Department of Information Technology,
SRM Institute of Science and Technology, Kattankulathur – 603 203, Chennai
Tamil Nadu, INDIA
[e-mail: chiwariro@gmail.com]

S. Rajendran[2]
Department of Information Technology,
SRM Institute of Science and Technology, Kattankulathur – 603 203, Chennai
Tamil Nadu, INDIA
[e-mail: rajendran.s@ktr.srmuniv.ac.in]

Abstract— Recent advances in technology have led to rapid growth of Internet of Things (IoT) systems which incorporate numerous miniaturized low powered devices with large numbers of sensors and actuators collecting and exchanging data autonomously over the internet generating enormous amounts of data that needs to be secured. Traditional encryption algorithms are not suitable due to great complexity and numerous rounds for encryption and decryption operations. There is however a rising need for elaborate lightweight encryption algorithms with less complexity for optimum security in resource constrained communication networks. In this paper, a lightweight encryption algorithm called Intra-Optimized Lightweight Enciphering (ILE) Algorithm is proposed. The proposed scheme is complemented by watchdogs who are deployed in the clusters to achieve optimum security for the overall generated clusters at less cost and is simulated on Message Queue Telemetry Transport (MQTT) protocol using Mosquito broker in Cooja simulator and the performance was evaluated. Results from simulations show that the proposed algorithm offers significant security, improved performance and power drain without compromising the quality of service and further a comparison was made with existing lightweight algorithms.

Keywords— ILE, MQTT, watchdogs, cluster, lightweight cryptography, constrained devices.

I. WIRELESS SENSOR FOSTERED INTERNET OF THINGS

Internet of things (IoT), also known to as 'Smart Object' Networks [1] refers to the internetworking of any physical devices, homes, vehicles or any other thing embedded with sensors, actuators, electronics and network connectivity. IoT devices are basically connected to the internet, can be sensed and therefore can be remote controlled. These intelligent devices have caused a paradigm shift to the way the environment interacts with technology. As rapidly evolving as it is, IoT has infiltrated many research areas, ranging from home automation, the healthcare sector, industrial automation, logistics, the mining sector, security and many others. The increase in general accessibility of broadband internet at lower costs has led to increased connectivity which is resulting in huge amounts of data being generated autonomously every second. Some of the data requires some varying levels of security as we see the application of IoT across all verticals.

The limitations surrounding the IoT environment include unstable network state, limited computation capability and low battery power [2] among others, leading to applications with limited security or non-secure applications which are prone many attacks. The other major problems with IoT objects is that they run autonomously in the field without constant supervision hence they are susceptible to tempering as well as their wireless nature which makes eavesdropping possible. In order to mitigate these constrains within an IoT ecosystem at the application layer, various data transfer protocols have been proposed in the recent years, which includes Message Queuing Telemetry Transport (MQTT and Advanced Message Queueing Protocol (AMQP) that use the queuing theory in publish/ subscribe pattern. Furthermore, the Extensible Messaging and Presence Protocol (XMPP) allows exchange of structured extensible messages and the Constrained Application Protocol (CoAP) uses a lightweight request/ response model. These data transfer protocols suffer bottle necks inherited from the constrained environments in which they operate hence security implementations remain a challenge in trying to keep them lightweight; which is why they were proposed in the first place.

II. RELATED WORK

There are numerous researches for data security and cryptographic primitives' optimization in IoT with various implementations as well. A number of security challenges are associated with systems which the protocols embodies default functionalities and settings. Additionally, many lightweight encryption algorithms have been proposed, with a large number of them having appealing features. However, there is still need for more research to further optimize these algorithms in order to make them suitable for implementation in low power and resource constrained IoT networks.

2.1 Lightweight Cryptography

The need for security in resource-constrained IoT networks gave rise to research in lightweight cryptographic primitives. The primitives include both symmetric and asymmetric block ciphers and stream ciphers, a variety of lightweight hash functions and message authentication codes (MACs), which aim to offer improved performance over conventional cryptographic standards. The main difference between conventional algorithms and these lightweight primitives is that lightweight primitives are intended for limited applications, and they assume similar power limitations to the attacker. However, this fact should not be confused to mean that the lightweight algorithms are less secure but, contemporary research should use improvements that result in designs that balance between security, improved efficiency and resource usage [3].

2.2 Lightweight Block Ciphers

Advanced Encryption Standard (AES) [4], particularly AES-128 has been simplified in some implementations to improve their efficiency. One of the early lightweight block cipher designs [5] targeted constrained hardware environments. Some designs come in suites such as SIMON and SPECK [6], which are lightweight block ciphers aiming for simplicity, flexibility and performance to the underlying hardware and software. Some algorithms that have been fairly implemented in constrained environments due to their simple round structure are RC5 [7],TEA [8], XTEA [9] and HIGHT [10]. Other lightweight block ciphers are listed in [11]. Particularly the performance benefits in all these algorithms are realized through the design choices that include smaller block and key sizes, simpler rounds and key schedules among others.

2.3 Lightweight Stream Ciphers

Mainly Lightweight Stream ciphers were restricted to resource constrained hardware applications. One such stream cipher is Grain [12], due to its implementation flexibility and authentication support. Another design that was well analysed is Trivium [13] which makes use 80-bit keys. However, Mickey [14] provides limited implementation flexibility when analysed against Grain and Trivium as well as being susceptible to scheduling and power analysis caused by irregular clocking.

2.4 Lightweight Hash Functions

Similarly, the constraints in constrained environments apply to hash functions. This is mainly contributed by the internal state sizes and levels of power consumptions. Lightweight hash functions, such as PHOTON [15], Quark [16], SPONGENT [17], and Lesamnta-LW [18] were as a result of trying to balance between power consumption and efficiency. The general usage of lightweight hash functions differs from that of conventional hash functions in a variety of ways [19].

2.5 Lightweight Message Authentication Codes

Message authentication code (MAC) uses the combination of a message and a secret key to generate a tag that is used for authentication and integrity checks on messages. The minimum recommended Tag sizes are to be at least 64 bits. Some examples of lightweight MAC algorithms are Chaskey [20], TuLP [21], and LightMAC [22].

III. INTRA-OPTIMIZED LIGHTWEIGHT ENCIPHERING ALGORITHM (ILE)

The primary objective is to improve the security of IoT systems in particular those using the publish/ subscribe pattern and to offer a security mechanism for the resource constrained devices whilst limiting the complexities for encryption and decryption without compromise on the underlying security of the algorithm. The proposed algorithm is intended to offer high security in IoT domain in particular to use cases that involve highly constrained devices where network connections are erratic and power management is of essence. The Intra-Optimized Lightweight Enciphering (ILE) algorithm is a symmetric key block cipher incorporating an 80-bit key and 80-bit plain-text following both the Substitution and Permutation Network (SPN) and feistel structure in order to achieve Shannon's requirements for diffusion and confusion as well balance with computational complexity for the encryption and decryption operations by making maximum use of the same registers for both operations. However, in symmetric key algorithm the encryption process comprises of encryption rounds, each round depends on some computational capacities to create disarray and an increase in number of rounds guarantees improved security. Cryptographic algorithms [23] are normally optimised at the range of 10 to 20 rounds on average to keep the encryption procedure sufficiently rigid to attain higher level of diffusion and confusion. The proposed algorithm is limited to just six rounds, hence to improve the energy efficiency, individual round of encryption incorporates mathematical operation that operate on data blocks of 4 bits. To complement the algorithm, the scheme should employ watchdogs for detection of intrusions. The algorithm goes through two fundamental stages; Key expansion and Encryption on the publisher side followed by Key expansion

A. Key Expansion

The key expansion process is the first part as the keys are used for subsequent encryption rounds. The secret key is used to generate round keys for subsequent encryption or decryption rounds. It is important to note that the whole security of the algorithm is dependent upon the key. In this regard, essential measures must be considered to make the disclosure of the key on a need to know basis. The proposed scheme is an 80-bit cipher, which requires an 80-bit key for every round of encryption and decryption. A cipher key of 80-bits is taken as input key by the cipher block. The Block, after operations creates diffusion as well as confusion and in the process creates six special keys from this initial key. These keys should be utilized as a part of the encrypting/ decrypting process and are sufficiently rigid to remain unclear amid attacks. The key expansion process includes the following:

- Initially, the 80-bit cipher key is divided into multiple blocks of 4-bits.

- The bits for each function are obtained by carrying out initial substitutions of the segments of cipher key as given in the equation:

$$Kbif = \|^{5}_{j=1} \quad Kc5\,(j-1) + I \qquad (1)$$

Where i = 1 to 5 for first 5 round keys as depicted in Fig in 2.

- Matrix transformation are carried out to obtain round keys, K1, K2, K3 K4, K5 and K6 into six arrays of 16 bits round keys (Kr). The arrangement of these bits is shown in equations (2), (3), (5), (6) and (7).

$$K1 = i4 \oplus i3 \oplus i2 \oplus i1 \oplus i5 \oplus i6 \oplus i7 \oplus i8 \oplus i12 \oplus i11 \oplus i10 \oplus i9 \oplus i13 \oplus i14 \oplus i15 \oplus i16 \qquad (2)$$

$$K2 = j1 \oplus j5 \oplus j9 \oplus j13 \oplus j14 \oplus j10 \oplus j6 \oplus j2 \oplus j3 \oplus j7 \oplus j11 \oplus j15 \oplus j16 \oplus j12 \oplus j8 \oplus j4 \qquad (3)$$

$$K3 = k1 \oplus k2 \oplus k3 \oplus k4 \oplus k8 \oplus k7 \oplus k6 \oplus k5 \oplus k9 \oplus k10 \oplus k11 \oplus k12 \oplus k16 \oplus k15 \oplus k14 \oplus k13 \qquad (4)$$

$$K4 = l13 \oplus l9 \oplus l5 \oplus l1 \oplus l2 \oplus l6 \oplus l10 \oplus l14 \oplus l15 \oplus l11 \oplus l7 \oplus l3 \oplus l4 \oplus l8 \oplus l12 \oplus l16 \qquad (5)$$

$$K5 = m16 \oplus m11 \oplus m7 \oplus m3 \oplus m2 \oplus m6 \oplus m4 \oplus m10 \oplus m9 \oplus m15 \oplus m11 \oplus m7 \oplus m16 \oplus m15 \oplus m14 \oplus m1 \qquad (6)$$

An XOR operation is carried out among the six round keys to obtain the seventh round key as shown in equation

$$K6 = K1 \oplus K2 \oplus K3 \oplus K4 \oplus K5 \qquad (7)$$

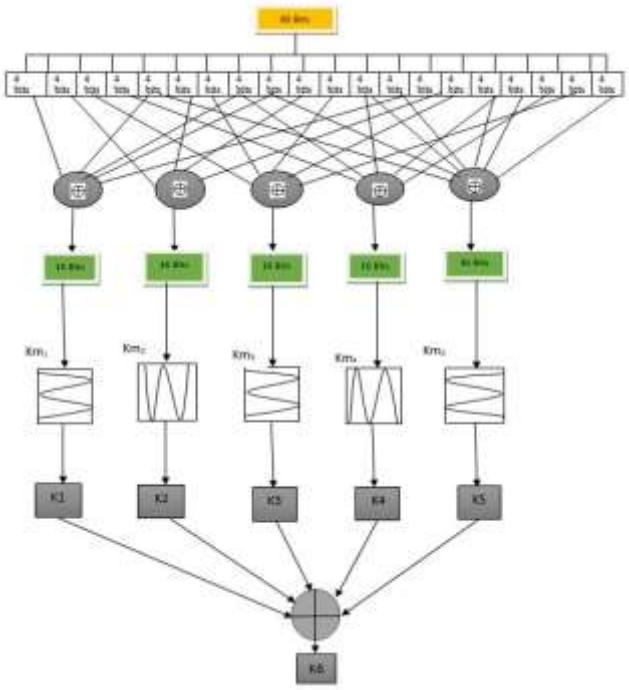


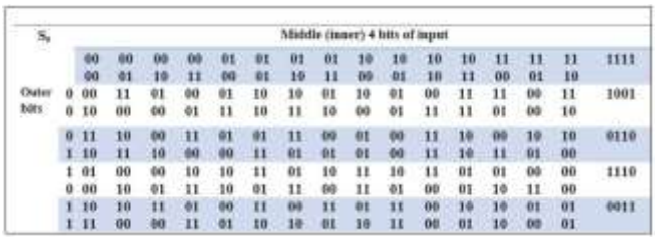
Fig. 1. *Key Expansion*

B. Encryption

As soon as round keys are generated we can begin the encryption process; we create disarray by some logical operations which are swapping, left shifting and substitution. The plaintext is divided into blocks which will be encrypted by the round keys in turn. The blocks are Px0-15, Px16-31, Px32-47, Px48-63, PX64-79. Bitwise XNOR Process is carried out amid the individual round keys (K1 – K5) which were generated earlier during the key expansion phase. This is done to the blocks to obtain Px0-15 to Px64-79 cipher text blocks respectively. The blocks are then finally XORed together with the last round key to obtain the final cipher text.

C. S-box Configuration

The Substitution box (S-Box) constitutes an important module of symmetric-key algorithms. To create a disarray of the plain text and coded text (cipher text) to make it difficult to decode. It is however necessary to choose carefully to avoid cryptanalysis. The substitution box s-box takes some number of m input bits and however transform them to some number of output bits, n. The number of output bits, *n*: an m × n S-Box can be implemented with 2^m words of n bits.



S_0	Middle (inner) 4 bits of input															
	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	1111
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	
Outer bits 0	00	11	01	00	01	10	10	01	10	01	00	11	11	00	11	1001
0	10	00	00	01	11	10	11	10	00	01	11	11	01	00	10	
0	11	10	00	11	01	01	11	00	01	00	11	10	00	10	10	0110
1	10	11	10	00	00	11	01	01	01	00	11	10	11	01	00	
1	01	00	00	10	10	11	01	10	11	10	11	01	01	00	00	1110
0	00	10	01	11	10	01	11	00	11	01	00	01	10	11	00	
1	10	10	11	01	00	11	00	11	01	11	00	10	10	01	01	0011
1	11	00	00	11	01	10	10	01	10	11	00	01	10	00	01	

Fig. 2. *S – Boxes Configuration [4]*

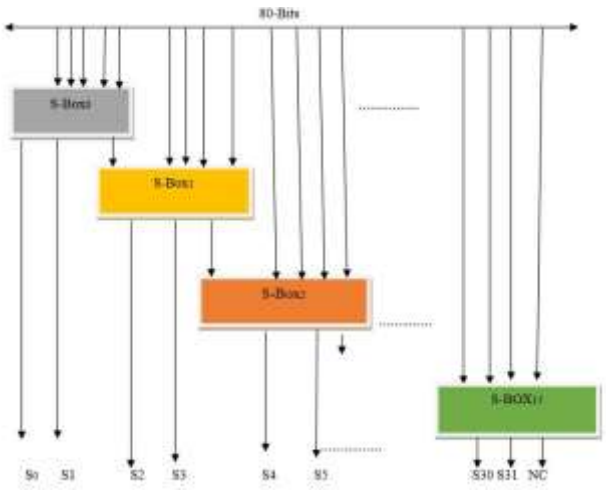


Fig. 3. *S - Box Configuration*

IV. Experimental Results and Analysis

Figures 5, 6 and 7 are results from validations of our algorithm in Cooja Simulator environment with standard testing methods for assessment against AES-128 and HIGHT algorithms. A

trade off was made between required security level and the resources required to achieve it. Performance metrics such as power and energy consumption, latency, and throughput are critical in determining the applicability of any lightweight algorithm within the IoT constrained environment. Furthermore, the efficiency of lightweight algorithms should be reflected in register, RAM and ROM usage as reflected in Table 1. Throughput was measured by considering the rate at which cipher text was being produced. ILE showed high throughput compared to the other two algorithms. In IoT real-time applications, Latency is especially a relevant factor to consider, for example automotive applications where quick responses to triggered actions are required. It can be measured by considering time between the initial request of a process and production of the output. This was measured by taking into consideration the difference between initial request for encryption of payload and the reply that yielded the equivalent cipher text. Likewise, power consumption was also recorded over a time period. The algorithm uses less power which is a desirable quality in lightweight cryptography. However, power consumption depends on many other such as the threshold voltage, clock frequency besides the algorithm used.

Table 1. Software Metrics

Cipher	Device	Block Size	Key Size	Code Size	Ram	Cycles (Enc)	Cycles (Dec)
AES	AVR	64	128	1570	-	2739	3579
HIGHT	AVR	64	128	5672	-	2964	2964
ILE	Atmega328	80	80	1228	15	2480	2860

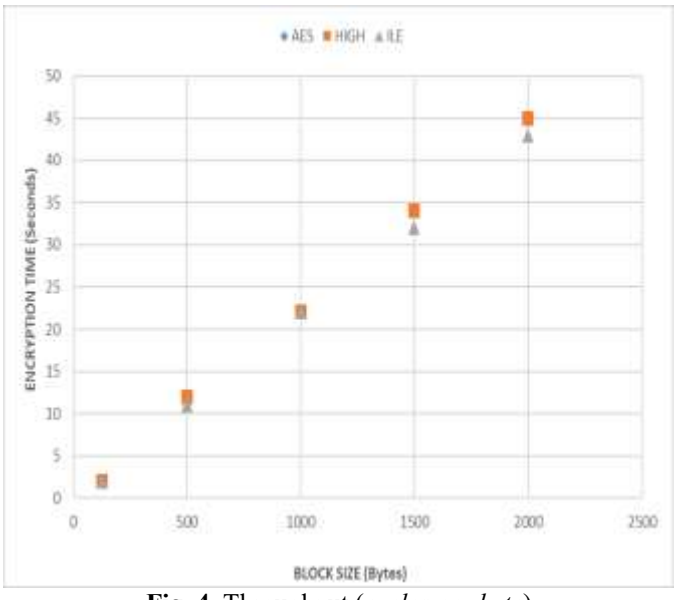


Fig. 4. Throughput (*cycles per byte*)

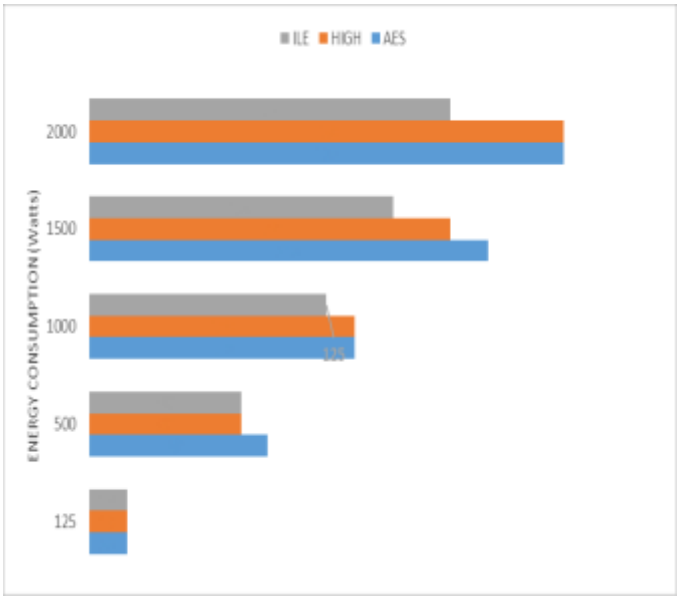


Fig. 5. Power/ Energy Consumption (*W*)

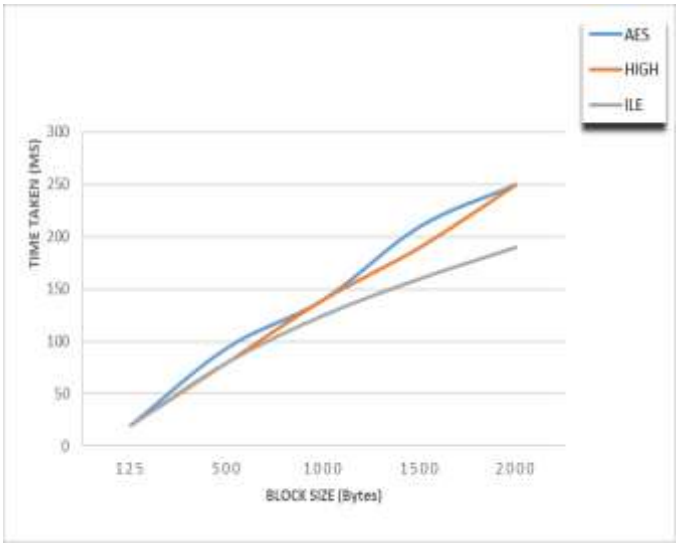


Fig. 6. Latency (*cycles/ packet*).

V. Conclusion

In this paper Intra-Optimized Lightweight Encipher, a data encryption algorithm for use in resource constrained IoT environments to mitigate the resource challenges of adopting Transport Layer Security (TLS) on erratic networks was presented. MQTT was used for implementation and testing of AES-128, HIGHT and ILE using Mosquito Broker in Contiki Cooja Simulator. The overall performance of the proposed algorithm was evaluated and the results show that ILE algorithm had better performance in terms of throughput, latency and power dissipation for both encryption and decryption operations. We further recommend use cases that require additional security to further complement the algorithm with a well-organised distributed clustering algorithm for high security integrity through the use of watchdog mechanisms for

intrusion detection. With the data generated by IoT autonomously everyday there is great need to be keen on privacy and protection issues to personal and corporate information. Adoption of platform supported security mechanisms to data both in transit and storage remains critical to the rapid acceptance of IoT by ordinary users since concerns remain on security and privacy issues.

REFERENCES

[1] P.L.L.P. Pan Wang, Professor Sohail Chaudhry, S. Li, T. Tryfonas, and H. Li, "The Internet of Things: a Security Point of View", Internet Research, Vol. 26, no. 2, pp. 337–359, 2016.

[2] Tobias Heer, Oscar Garcia-Morchon, Rene Hummen, Sye Loong Keoh, Sandeep S. Kumar, and Klaus Wehrle. "Security Challenges in the IP-based Internet of Thing". Journal on Wireless Personal Communications, Springer. December 2011, Volume 61, Issue 3, pp 527-542

[3] National Institute of Standards and Technology, NIST Report on Lightweight Cryptography, NISTIR 8114, March 2017, https://doi.org/10.6028/NIST.IR.8114.

[4] U.S. Department of Commerce, Advanced Encryption Standard (AES), Federal Information Processing Standards (FIPS) Publication 197, November 2001.

[5] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., and Vikkelsoe, C., PRESENT: An Ultra-Lightweight Block Cipher. Proc. 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007), Vienna, Austria, September 10-13, 2007, LNCS 4727, pp. 450-466.

[6] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L., The SIMON and SPECK Families of Lightweight Block Ciphers, IACR Cryptology ePrint Archive, 2013.

[7] Rivest, R.L., The RC5 Encryption Algorithm. Proc. Second International Workshop on Fast Software Encryption (FSE 1994), Leuven, Belgium, December 14–16, 1994, LNCS 1008, pp. 86-96.

[8] Wheeler, D.J., and Needham, R.M., TEA, A Tiny Encryption Algorithm. Proc. Second International Workshop on Fast Software Encryption (FSE 1994), Leuven, Belgium, December 14–16, 1994, LNCS 1008, pp. 363-366.

[9] Needham, R.M., and Wheeler, D.J., Tea extensions, Technical Report, Computer Laboratory, University of Cambridge, October 1997.

[10] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S; Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In L. Goubin and M. Matsui, editors, Proceedings of CHES 2006, LNCS, volume 4249, pages 46–59, Springer-Verlag, 2006.

[11] Biryukov, A., and Perrin, L., Lightweight Block Ciphers, https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers, [accessed January 10, 2018]

[12] Hell, M., Johansson, T., and Meier, W., Grain: A Stream Cipher for Constrained Environments, International Journal of Wireless and Mobile Computing (IJWMC), 2007, Vol. 2, (1), pp. 86-93.

[13] De Cannière, C., and Preneel, B., Trivium: 'New Stream Cipher Designs - The eSTREAM Finalists' (Springer, 2008), LNCS 4986, pp. 244-266.

[14] Babbage, S., and Dodd, M., The MICKEY Stream Ciphers: 'New Stream Cipher Designs - The eSTREAM Finalists' (Springer, 2008), LNCS 4986, pp. 191-209.

[15] Guo, J., Peyrin, T., and Poschmann, A., The PHOTON Family of Lightweight Hash Functions. Proc. 31st Annual International Cryptology Conference (CRYPTO 2011), Santa Barbara, CA, USA, August 14-18, 2011, LNCS 6841, pp. 222-239.

[16] Aumasson, J.P., Henzen, L., Meier, W., and Naya-Plasencia, M., Quark: A Lightweight Hash, Journal of Cryptology, 2013, Vol. 26, (2), pp. 313-339.

[17] Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., and Verbauwhede, I., SPONGENT: A Lightweight Hash Function. Proc. 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 – October 1, 2011, LNCS 6917, pp. 312-325.

[18] Hirose, S., Ideguchi, K., Kuwakado, H., Owada, T., Preneel, B., and Yoshida, H., A Lightweight 256-Bit Hash Function for Hardware and Low-End Devices: Lesamnta-LW. Proc. 13th International Conference on Information Security and Cryptology (ICISC 2010), Seoul, Korea, December 1-3, 2010, LNCS 6829, pp. 151-168.

[19] Poschmann, A.Y.: Lightweight Cryptography: Cryptographic Engineering for a Pervasive World. Ph.D. Thesis, Ruhr University Bochum, 2009.

[20] Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., and Verbauwhede, I., Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. Proc. 21st International Conference on Selected Areas in Cryptography (SAC 2014), Montreal, QC, Canada, August 14-15, 2014, LNCS 8781, pp. 306-323.

[21] Gong, Z., Hartel, P., Nikova, S., Tang, S.H., and Zhu, B., TuLP: A Family of Lightweight Message Authentication Codes for Body Sensor Networks, Journal of Computer Science and Technology, 2014, Vol. 29, (1), pp. 53-68.

[22] ISO, ISO/IEC 29167-10:2015, Information Technology - Automated Identification and Data Capture Techniques - Part 10: Crypto Suite AES-128 Security Services for Air Interface Communications, 2015.

[23] S. Khan, M. S. Ibrahim, M. Ebrahim, and H. Amjad, "FPGA implementation of secure force (64-bit) Low Complexity Encryption Algorithm", International Journal of Computer Network and Information Security, Vol. 7, No. 12, p. 60, 2015.