Towards an Ultra-lightweight Cryptosystem for IoT

Tasnime Omrani[1], Layth Sliman[2(✉)], Rabei Becheikh[1], Safya Belghith[1], and Belgacem Ben Hedia[3]

[1] Ecole Nationale d'Ingénieurs de Tunis, Université de Tunis El Manar, LR16RS07 Robotique, Informatique et Systèmes Complexes, BP 37 Le Belvédère, 1002 Tunis, Tunisia
[2] EFREI Engineering Institute, Villejuif, France
layth.sliman@gmail.com
[3] Atomic Energy Center, LIST Institute, Paris, France

Abstract. With the spread of Internet of Things (IoT), ensuring security and privacy proves essential. However, given the limited computation resources of IoT objects, the mechanisms and techniques used to secure data exchange in these environments should consider many constraints such as power consumption, size, execution time … etc. To cope with these constraints, new lightweight encryption systems have been proposed. This work is intended to compare some recently proposed lightweight cryptosystem in order to identify and suggest the necessary recommendations to achieve an ultra-lightweight cryptosystem adapted to IoT environment.

Keywords: Cryptography · Lightweight cryptosystem · Ultra-lightweight cryptography · IoT security

1 Introduction

IoT is all the rage nowadays, and for a good reason; it can help us out in almost every action in our lives and Businesses: from pocket-sized devices and other smart objects that we carry around or surrounding us at all times and everywhere, to smart factories, smart supply chains and Smart Cities.

In this context, data is exchanged continuously. Considering the fast and massive data exchange in IoT environments, as well as the need of getting IoT platforms highly intuitive, interactive and responsive, approaching IoT cryptography using existing solutions seems to be an expensive and rather ineffective alternative. In fact, with respect to resource limitations in IoT context, traditional cryptography solutions suffer from two major drawbacks: intensive memory and intensive computation resource usage. To cope with these drawbacks, new lightweight encryption systems have been proposed. However, up to date, no forma guide or recommendations have been issued on which one can relay to design a lightweight cryptosystem.

In this work, we will try to propose a set of recommendations which can be used to design a lightweight cryptosystem. The paper is organized as follows: in the first section, called state of the art, some popular lightweight cryptosystem are described. In

A. Abraham et al. (eds.), *Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016)*, Advances in Intelligent Systems and Computing 614, DOI 10.1007/978-3-319-60618-7_60

Sect. 3 the different methods described in the state of the art are analyzed and evaluated in order to identify and suggest the necessary recommendations to achieve an ultra-lightweight cryptosystem adapted to IoT environment. Based on the aforementioned evaluation and analysis, in Sect. 4, we suggest a set of recommendations then, in Sect. 5, we finish by a conclusion and we describe our future work.

2 State of the Art

A cryptosystem is considered as "Lightweight" if its software occupies less than 32 KB of ROM and it uses less than 8 kb of RAM and less than 3000 logic gates [1]. Since more than ten years, a whole heap of lightweight cryptosystems have been designed and proposed. In this state of the art we will limit our description to Present [2], considered as the reference approach of Lightweight cryptosystem, and the most recent lightweight cryptosystems, i.e. those which were proposed from 2015 up-to date.

2.1 Present

Present has been considered for long time as the reference approach of lightweight cryptosystem. Present follows Substitution-Permutation Network Architecture (SPN) [3, 4] with 32 rounds. As it is shown in Fig. 1, each round composed of an addition with sub-key, a substitution function using 4×4 S-box [5, 6] and permutation with bits.

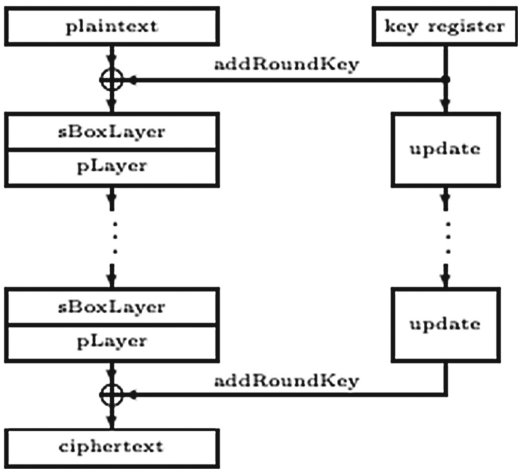

Fig. 1. Present cryptosystem [2]

Although Present has been designed to fit for computation resources limitations, it fails to meet some of IoT constraints, namely the high energy consumption because of a large number of rounds used (32 rounds), as well as the high computation resource usage caused by bits permutation function.

To overcome Present drawbacks many evolutions of Present architecture and algorithm have been proposed resulting in many new lightweight cryptosystems which are sought to be "lighter" than Present.

Hereafter, we show a brief description of the most recent lightweight cryptosystems designed to cope with Present limitations, namely Midori [7], RaodRunner [8], Rectangle [9] and Simeck [10, 11].

2.2 Midori

Midori encrypts 64 or 128 bits message with 128 bits key. The message is realized in the form of matrix with 4 rows and 4 columns of nybbles for 64 bits size message and bytes for 128 bits size message. It follows Substitution-Permutation Network structure with 14 or 18 rounds dor Midori-64, Midori-128 respectively.

As it is shown in Fig. 2, each round containing an addition with involutive Sbox 4*4, a permutation before and after Sbox function and finally a multiplication with a binary involutive matrix.

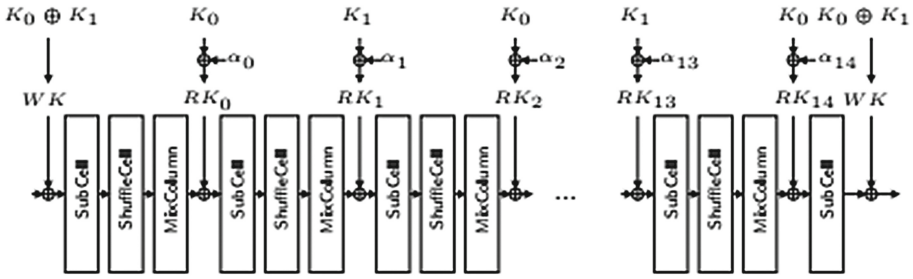

Fig. 2. Midori cryptosystem [7]

Using involutive components seems to be a good choice since it allows to use the same components for both encryption and decryption and hence reduce memory and logic gates use.

Furthermore, Midori uses a simple function to generate the sub-keys. Again, this allows to reduce memory, logic gates and time usage. The function is described as follows:

For Midori128: RKi = K $\oplus$ βi

For Midori64: Ki = K (i mod 2) $\oplus$ αi avec $\alpha i = \beta i$ pour $0 \leq i \leq 14$

2.3 RoadRunner

The main goal of RoadRunner is to reduce memory use. This is insured by a reduced code size.

RoadRunner encrypts a 64 bits message size with an 80 or a 128 key size. It follows Feistel structure [12] with 10 or 12 rounds for RR-80 and RR-128 respectively, and with SPNas a function of Feistel structure.

As shown in Fig. 3 RoadRunner uses Key Whitening method [13]. Key Whitening method allows to apply to the block an XOR with a sub-key before and after each round. This increases the security level.

In RoadRunner each Feistel function is composed of 4 bits S-box layers and 3 linear layers (according to the formula 1) and three layers of mixing with the sub-key.

$$(x <<< i) \oplus (x <<< j) \oplus (x <<< k) \tag{1}$$

To reduce memory usage the key is divided into sub-keys of 32 bits which are used one by one in a circular manner.

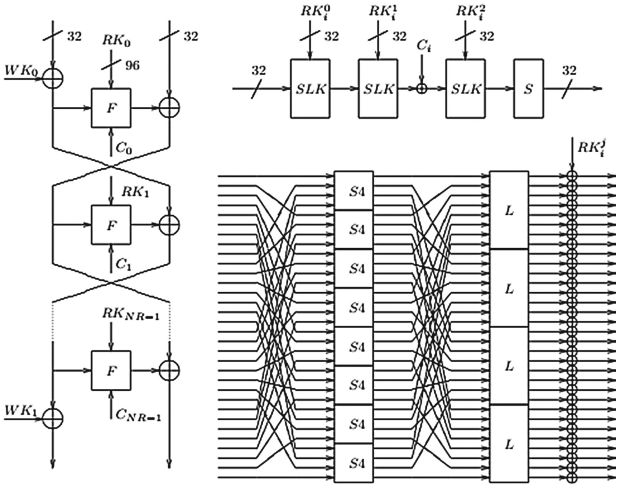

Fig. 3. RoadRunner cryptosystem [8]

2.4 Rectangle

It's based on a bit-slicing design [14]. The objective of using a bit-slicing design is to increase the system's flexibility with regard to hardware implementation.

Rectangle encrypts 64 bits message with a 64 or 128 bits key. The message is realized in the form of matrix with 4 rows and 16 columns of bits. It follows SPN structure with 25 rounds containing an addition with sub-key and a substitution function using s-box and a rotation function which circularly shifts all the lines with a given value. The S-Box is applied simultaneously on the 16 columns. The objective is to reduce the number of clock cycle.

The keySchedule is a simple function that uses only shifts and XORs to reduce logic gates number.

2.5 Simeck

As shown in Table 1, there are 10 variants of this system. We distinguish between these method by the massage size and the size of the encryption key.

This cryptosystem is considered as a good option in case if we are only concerned by the hardware implementation cost (in term of logic gates) and not so much by memory usage. Simeck uses a circular shift; which requires only a material wiring. It also uses bit-by-bit operations (XOR, AND) which are flexible operation. However, bit-by-bit operations are usually memory greedy. Figure 4 shows a round of Simeck.

Sub-keys generation function consists of bit-by-bit circular shift operations with XOR. Subsequently, sub-keys are interdependent. However, this function reduce logic gates usage.

Table 1. Combination

Block size	Ket size	Rounds number
32	64	32
48	72	36
48	96	63
64	96	42
64	128	44
96	96	52
96	144	54
128	128	68
128	196	69
128	256	72

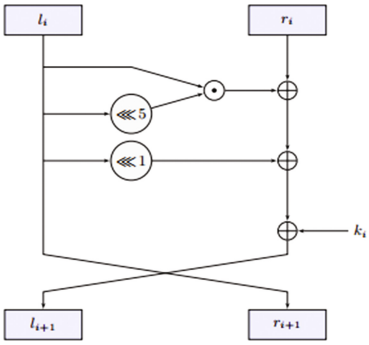

Fig. 4. Simeck cryptosystem [10]

3 Simulations

Lightweight cryptosystems can be evaluated, with regard to software implementation, in term of RAM and ROM memory usage. In the other side and as far as the hardware implementation is evaluated, the evaluation is done in term of gates equivalent.

In the following we proceed into an evaluation of the abovementioned cryptosystems.

The evaluation is intended to analyze the studied systems in term of memory and logic gates used by each cryptosystem. As a result of the simulation, the shortcomings related to IoT constraints of the different studied cryptosystems will be identified and argued. The simulation, along with the state of the art study as well as a proper analysis, should allow us to provide some recommendations that help-up in designing ultra-light cryptosystems, i.e., cryptosystems that cope with IoT constraints.

3.1 Evaluation Software

The simulation is achieved using a personal computer with an Intel core i7 processor, with a 2.20 GHz clock speed and an 8 GB of RAM, under the operating system Ubuntu 12.4.

As shown in Fig. 5, PRESENT does not take into consideration the memory use constraint since it keeps in the RAM 8.682 KB which exceeds the aforementioned maximum value of light weight cryptosystem (KB). This drawback has been overcome by Rectangle, Midori, Simek and RoadRunner. We can notice that Rectangle maintains the lowest values of memory RAM use comparing to the studied cryptosystems and it consumes 2.2 kb, a value very near to the minimum value recorded by Simeck which is 1.9.

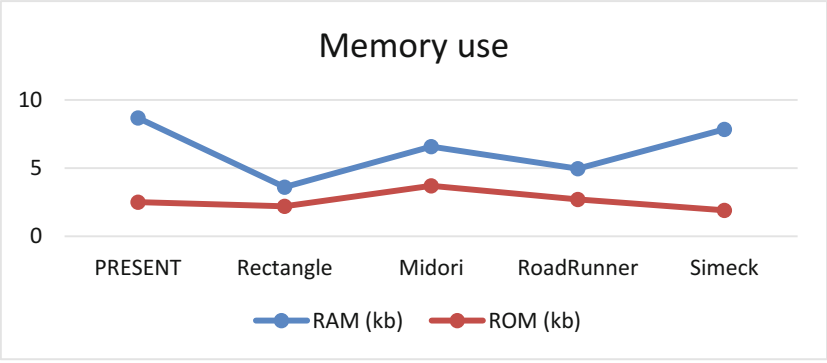

Fig. 5. Software evaluation

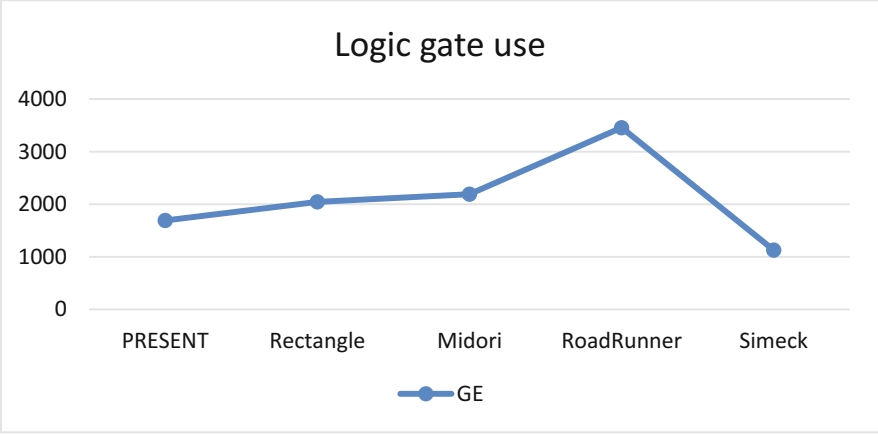

Fig. 6. Hardware evaluation

3.2 Hardware Evaluation

The hardware simulation was run using VHDL description language embedded in Xilinx system design, Spartan-IIE Field-Programmable Gate Array family and xc2s300e-6pq208 device.

As per the hardware experiments and as shown in Fig. 6, Simeck maintains the least value of logic gates. Although RoadRunner has perfectly respected the constraints of memory use, clearly, it does not give the same importance to hardware implementation constraints.

4 Recommendations

As a result of our study, we could obtain the following list of recommendations to help designing a cryptosystem that copes with IoT constraints regarding memory and logic gates use while keeping a good level of security:

- To design an effective cryptosystem with regard to hardware cost and flexibility, we should use bitwise operations [9].
- Based on our simulation, we can conclude that using word oriented operations (in contrast to bit oriented operations) can help reducing memory usage (by compering the unit responsible of the "addition with a subkey" in Midori which is word oriented, and this of Simeck which is bit oriented, we can notice that Simeck's uses more memory than that of Midori (0.347 kb vs. 0.232 kb).
- With regard to energy and memory consumption. It's better to use 4 bits Sbox [7] . A 4 bits S-box allows to store 2^4 values while an 8 bits S-box stores 2^8 values. This requires 16 times more memory than 4 bits Sbox and requires more energy to find a value out of 256 possibilities.
- With Feistel architecture, the round function is applied only on the half of message. Thus, it can be implemented with low power consumption, less logic gates and easier decryption implementation (like in Simeck).
- In order to use components that provide operation for both encryption and decryption, so that we reduce the memory logic gate use. To this end we can use involutive component (like in Midori).
- Using Keyschedule function increases security. However, it may increase memory usageand processing time. Therefore, we can reduce memory usage by avoiding sub-keys generation function (like in RoadRunner).

5 Conclusion

In this paper we have conducted an experimental analysis of some existing and widely used lightweight cryptosystems. The analysis has been done in order to identify a set a recommendations that can be used to design the cryptosystem dedicated to IoT environments, characterized by the limited resources and the need of fast response.

As a next step, we will proceed to a more detailed analysis that involve a larger spectrum of cryptosystems and more criteria such as algorithms complexity and the adequacy of each cryptosystem to different kinds of content.

References

1. Charalampos, M., George, H., Konstantinos, F., Konstantinos, R.: Lightweight cryptography for embedded systems - a comparative analysis. In: 6th International Workshop on Autonomous and Spontaneous Security, SETOP 2013. LNCS. Springer, Heidelberg (2013)
2. 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: 9th International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
3. Kim, G.H., Kim, J.N., Cho, G.Y.: Symmetry structured SPN block cipher algorithm. In: 11th International Conference on Advanced Communication Technology, ICACT 2009 (2009)
4. Heys, H.M., Tavares, S.E.: Substitution-permutation networks resistant to differential and linear cryptanalysis. J. Cryptol. **9**, 1–19 (1996)
5. Al Dabbagh, S.S.M., Al Shaikhli, I.F.T.: Security of PRESENT S-box. In: 2012 International Conference on Advanced Computer Science Applications and Technologies (2013)
6. Saarinen, M.J.O.: Cryptographic Analysis of All 4 × 4-Bit S-Boxes. Springer, Heidelberg (2012)
7. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: a block cipher for low energy. In: Advances in Cryptology–ASIACRYPT 2015. Springer, Heidelberg (2015)
8. Baysal, A., Sühap, Ş.: RoadRunner: a small and fast bitslice block cipher for low cost 8-bit processors. LightSec (2015)
9. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., VerBauwhede, I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci. China Inf. Sci. (2015)
10. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The Simeck family of lightweight block ciphers. In: Cryptographic Hardware and Embedded Systems–CHES 2011. Springer, Heidelberg (2014)
11. Kölbl, S., Roy, A.: A Brief Comparison of Simon and Simeck, IACR Cryptology ePrint Archive (2014)
12. Knudsen, L.R.: Practically secure Feistel ciphers. Arhus University, Denmark. Springer, Heidelberg (1994)
13. Schneier, B.: Applied Cryptography, pp. 366–367. Wiley, New York (1996)
14. Grabher, P., Großschädl, J., Page, D.: Light-weight instruction set extensions for bit-sliced cryptography. International Association for Cryptologic Research (2008)