Comparison of Hardware and Software Based Encryption for Secure Communication in Wireless Sensor Networks

Miroslav Botta, Milan Simek, Nathalie Mitton

Comparison of Hardware and Software Based Encryption for Secure Communication in Wireless Sensor Networks

Miroslav Botta, Milan Simek, Nathalie Mitton

***Abstract*—This paper deals with the energy efficient issue of cryptographic mechanisms used for secure communication between devices in wireless sensor networks. Since these devices are mainly targeted for low power consumption appliances, there is an effort for optimization of any aspects needed for regular sensor operation. On a basis of utilization of hardware cryptographic accelerators integrated in microcontrollers, this article provides the comparison between software and hardware solutions. Proposed work examines the problems and solutions for implementation of security algorithms for WSN devices. Because the speed of hardware accelerator should be much higher than the software implementation, there are examination tests of energy consumption and validation of performance of this feature. Main contribution of the article is real testbed evaluation of the time latency and energy requirements needed for securing the communication. In addition, global evaluation for all important network communication parameters like throughput, delay and delivery ratio are also provided.**

***Keywords*—AES, encryption, Lightweight MESH, WSN, XTEA**

I. Introduction

WIRELESS sensor networks (WSNs) are networks of specific character and usability with a wide scope of applications. WSN has taken important place at information systems in last years, they are characteristic for their properties such as ultra-low energy consumption, low price and scalability.

Nowadays, the very popular standard for these networks is IEEE 802.15.4. It defines the PHY (PHYsical layer) and MAC (Medium Access Control) layer, but it does not define any specifications for upper layers, which depend on the used application. Based on the application and power resources, most networks based on the 802.15.4 standard have restrictive power consumption requirements. Because of this requirement, the hardware implements microprocessors with low computational power. Every included task has to take resources for as short as possible. This can be done with optimal software implementation when for additional processed function or task as a few processor cycles as possible are used. This can be done by the programmer or even by the native compiler. Another angle of view involves hardware optimization by means of parallel processing of independent tasks. Including hardware blocks optimized for precise task has advantage in short computational latency. These HW blocks also need power for their functionality.

In these days, the security of embedded and communication systems in general is a popular topic. The data security in automation systems requires high efficient approach with respect to the equilibrium of the implementation price and needed resources. Security features in WSN are restricted by the resources of the low computational performance. Because of that the lightweight security features using the hardware or software solutions were implemented.

This magnifies a question what method of optimization is more efficient when the energy consumption and computational resources are taken into account.

There is also proposed and evaluated the basic system for measuring the performance properties of implemented cryptographic solutions. At the end of work the testbed evaluation of energy efficiency and time needed for secure communication is included.

Section III describes a few basic security approaches in global. There is also a part which discusses the HW/SW solution differences and their pros and cons. Section IV describes the testbed and hardware configuration used for evaluation. Question of the final efficiency will be experimentally acquired and explained at the end of the work.

Manuscript received February 28, 2013. This research work is funded by projects SIX CZ.1.05/2.1.00/03.0072, EU ECOP CZ.1.07/2.3.00/20.0094, EU ECOP CZ.1.07/2.2.00/28.0062 and No. FRTI2/571, Czech Republic.

Miroslav Botta and Milan Simek is with the Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, Technicka 12, 61600 Brno, Czech Republic, (e-mail: botta@phd.feec.vutbr.cz, simek@feec.vutbr.cz).

Nathalie Mitton is with the Inria Lille-Nord Europe research center, Btiment A - Parc scientifique de la haute borne 40, avenue Halley 59650 Villeneuve d'Ascq, France (e-mail: nathalie.mitton@inria.fr).

II. Related Work

Some of related works [1] were devoted to the secure communication used in RFID systems and their performance.

There is also work with very similar thematic as the one proposed in our article. The [2] where authors compare the RC2, BLowfish, XTEA and AES computational performance and energy efficiency in personal digital assistants (PDA). They found out that RC2 encrypt faster and uses less energy than XTEA.

The [3] is devoted to the cryptanalytic attack on the XTEA with reduced round version, they considered that from the cryptanalysis point of view, the XTEA and TEA have close security level.

Our article extends research on the low computational power microprocessor widely used in wireless sensor networks.

III. Cryptographic features

In this section, approaches to the cryptographic features by means of practical implementation in real wireless sensor network will be described. Hardware and software alternatives, their brief description, pros and cons are also included.

A. AES - hardware block

The AES (Advanced Encryption Standard) hardware block implemented in AVR microprocessor ATmega128RFA1 [4] described later in section IV, is mainly characterized by its hardware accelerated encryption and decryption capabilities. It is compatible with AES-128 standard, using 128 bit key and data block size. AES also known as Rijndael is a block cipher and is based on a substitution/permutation network with fixed data block and also fixed encryption key size (128, 192 and 256 bits).

It supports two operation modes, the ECB (Electronic Code Book) and the CBC (Cipher Block Chaining). The main advantage of this block is the stand-alone operation ability, which is independent on other blocks. Encryption and decryption can be performed parallelly to other tasks. It uses the 16 MHz crystal clock provided by the transceiver. According to datasheet [4] the pure encryption/decryption interval of 16 data bytes without latency caused by writing values to the registers is 24 μs long. This event is signalized by the interruption.

B. XTEA - software solution

XTEA (eXtended Tiny Encryption Algorithm) is a symmetric key cryptography algorithm. It is extension to the old variant called TEA (invented by David Wheeler and Roger Needham), because it suffered from attack proposed in [5]. According to the [3], XTEA is fast block cipher of Feistel type algorithm.

It does not use predefined tables and S-boxes and also it does not need much initialization time. It works with 8 bytes data blocks and uses 16 bytes (128 bits) key. The main advantage of XTEA algorithm is its simplicity and efficiency with compact code size as can be seen from listing 1. This listing is obtained from open source code called ATMEL Lightweigt MESH software stack [6] which is described later in subsection IV-B.

Listing 1: XTEA algorithm - program code [6]

```
static void xtea(uint32_t text[2], uint32_t const
    key[4])
{
  uint32_t t0 = text[0];
  uint32_t t1 = text[1];
  uint32_t sum = 0;
  uint32_t delta = 0x9e3779b9;

  for (uint8_t i = 0; i < 32; i++)
  {
    t0 += (((t1 << 4) ^ (t1 >> 5)) + t1) ^ (sum +
        key[sum & 3]);
    sum += delta;
    t1 += (((t0 << 4) ^ (t0 >> 5)) + t0) ^ (sum +
        key[(sum >> 11) & 3]);
  }
  text[0] = t0;
  text[1] = t1;
}
```

C. Software vs. Hardware AES acceleration and security issues

As can be seen from the previous section, the software and hardware solutions are already implemented and used in practical application. The purpose of this work is to check the differences of these solutions by means of energy demands.

There is an option for HW block implementing AES cipher, but it has to be powered from limited source, which can mean some significance. Even the AES is theoretically stronger than XTEA, the integrated AES block can be security issue in some cases. For example, when microprocessor uses external radio chip, the security key has to be sent to it via SPI (Serial Peripheral Interface) in a plain text form (attacker who has access to the device can have SPI analyzer). In this case the software XTEA can offer better protection.

The hardware heterogeneity of network devices can create problem, because every device may not integrate such AES hardware. It means that XTEA is better solution in this kind of situation. This is basically the vendors' problem, because wireless sensor networks are mainly one purpose networks (do simple tasks and do it properly), so this has to be considered in design stage of development.

Big security issue is that the whole network uses same encryption key, some kind of safe key distribution method has to be included and also protection from reply attacks (for example sending alarm messages, old measurements and states) has to be implemented at application layer. From previous, it is not clear which solution is more efficient for use in WSN.

IV. Hardware and software description

For testing the cryptographic performance by means of energy efficiency, the Atmel AVR microprocessor with hardware AES accelerator will be used. Software stack is Lightweight MESH which also originates from Atmel Corporation. All energy consumption measurements are done by AGILENT DSO-X 2002 oscilloscope. Next subsections describe in more detail all of these components and also describe methodology of all measurements.

A. Hardware device - deRFnode

Main component of this development board shown in Fig. 1 is the microprocessor Atmel ATmega128RFA1 which has all parts, like radio chip and AES HW block integrated, so there is no need for additional SPI communication with external interfaces. It is compliant with IEEE 802.15.4-std and communicates in ISM 2,4 GHz band. The power consumption of used microchip in active mode is according to the [7] approximately 18 mA and in sleep mode it is less than 2 μA.

B. Atmel Lightweight Mesh

This software stack [6] as its name indicates, is very small stack with proprietary network MESH functionality. It is relatively easy to use and portable, designed for various applications like home and industry automation, metering and remote control. It works with IEEE 802.15.4 compliant

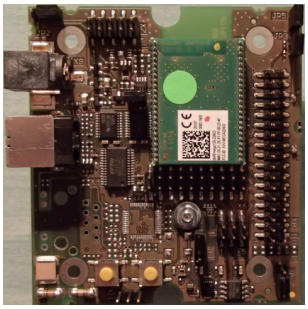

Fig. 1: Photography of deRFnode

transceivers [8]. This network stack is used because it is fully open source, it has already implemented XTEA and AES HW accelerated ciphering, and the most important fact is that, it offers the choice for selection of used cryptographic method.

V. Metrics and methodology used for measurement

In this subsection, the impact of encryption method and algorithms by means of energy efficiency of used power resources is proposed. The methodology and metrics used for final estimation is explained. For measuring all of these metrics, the Lightweight MESH network stack was slightly modified for signaling the ciphering event. Modification consisted of toggling of one microprocessors pin-out.

A. Energy consumption

For measuring the energy consumption, the oscilloscope was used for measuring the voltage drop U_R across serial attached resistor. A resistor is placed in series with hard power supply of $U_{supp} = +5$ Volts. The resistance was chosen $R = 10\Omega$ precisely. The reference multimeter was also attached for control purposes. This measurement is indicated by equation (1).

$$P = U_{supp} \cdot \frac{U_R}{R} \qquad [W] \tag{1}$$

The energy consumption E was finally computed using equation (2), where T is time needed for encryption of payload data and $P_{encryption}$ is power consumption.

$$E = P_{encryption} \cdot T \qquad [J] \tag{2}$$

B. Latency

One of the main metric in selection of suitable ciphering algorithm is its latency, respectively throughput. Of course, the memory demands at low power embedded devices are also important requirement. Latency/throughput is measured at network layer for variable packet size $1 \div MAX\_Payload_{SECURED}$ bytes. The parameter $MAX\_Payload_{SECURED}$ is given by maximum frame size (127 B), header (16 B), CRC part of frame (2 B) and MIC size (4 B).

Throughput values were computed by dividing the number of bytes encrypted by the time interval (measured at oscilloscope) required for ciphering.

VI. Testbed evaluation

In testbed evaluation shown in Fig. 2, the drop down voltage was measured across the serial attached resistor as was described earlier in section V-A.

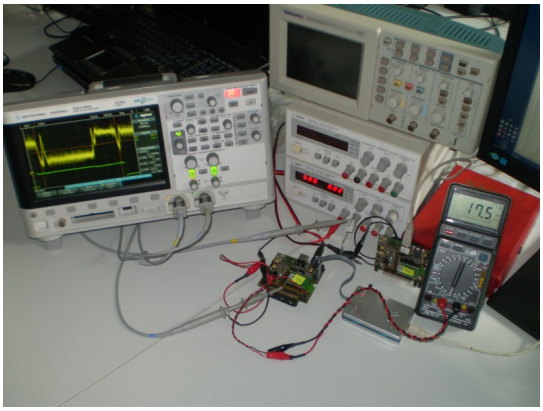

Fig. 2: Photo of testbed evaluation workspace

The original source code of the Lighweight MESH code was slightly modified for measuring the encryption/decryption time intervals. Also the simple application for sending frames with variable payload size was developed for this purpose, but it is not further described in article. Every time the encryption process begins/ends, the output pin of the microprocessor at the connector attached on the deRFnode board was put in the HIGH/LOW level. Using this approach the oscilloscope with triggered capability was used and encryption time intervals were measured manually with precision of microseconds. For every encryption time interval measurement of software XTEA and hardware accelerated AES, the true root mean square voltage bounded by the HIGH/LOW level of the switched pin with the oscilloscope was measured. All measurements and calculated values are in detail shown in Fig. 3, 4 and 5.

In Fig. 3 the throughput of the pure encryption process is shown. This throughput was calculated by number of encrypted bytes divided by the whole time interval needed for encryption. As can be seen, every time, the peeks for every

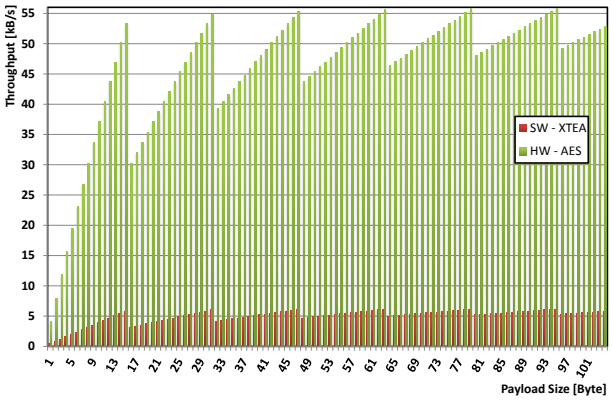

Fig. 3: Graph - Dependency of encryption throughput on the payload size

16 Bytes of data occur. It is caused by padding of data blocks to the key size of 128 bits.

The following Fig. 4 shows power consumption when encryption is in process. There are only little changes during the whole measurement.

Table I contains only selected sample measurements of the whole set of payload lengths. This table and oscilloscope printscreens in Fig. 6 show how the encryption process and related throughput/energy consumption depend on the payload length.

The beginning, respectively the end of encryption process is signalized with the output pin (*put in HIGH/LOW level*) connected to the oscilloscope second channel (*green line*) in Fig. 6. The Fig. 6a shows sample encryption process measurement of the 31B payload with the XTEA algorithm. The influence of the other payload sizes is seen in the duration of XTEA encryption process. On the other hand, using the hardware AES block has significant improvement in the encryption process. The Fig. 6b – Fig. 6h show the physical utilization of this block.

As can be seen from the graphs in Fig. 3 and Fig. 5 there are special cases in which it is useful to send more data. This achievement can be used for example when optimization in network systems is such a big concern.

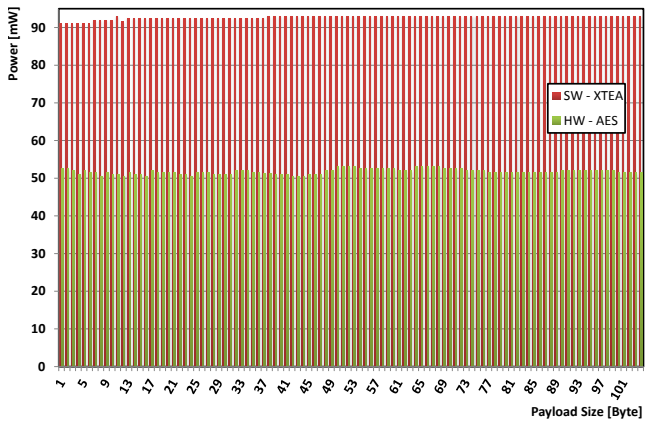

Fig. 4: Graph - Dependency of electrical power on the payload size

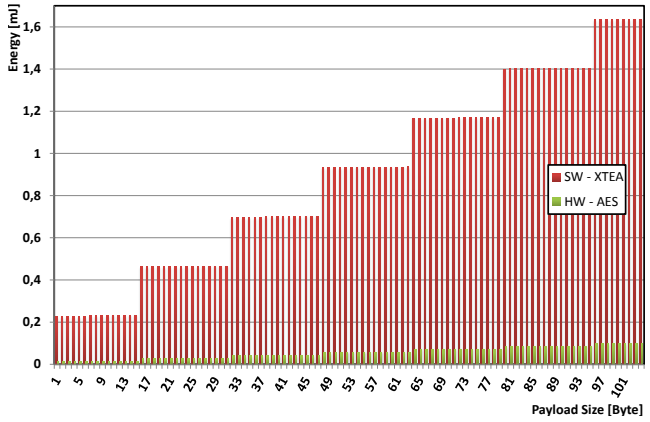

Fig. 5: Graph - Dependency of energy need for encryption/decryption on the payload size

TABLE I: Energy consumption and throughput for variable frame size with software XTEA and hardware AES encryption

	SW-XTEA			HW-AES		
Payload [B]	Time [ms]	Throughput [kB/s]	Energy [mJ]	Time [ms]	Throughput [kB/s]	Energy [mJ]
1	2,48	0,39	0,23	0,24	4,03	0,01
15	2,51	5,83	0,23	0,28	53,26	0,01
16	4,99	3,13	0,46	0,52	30,21	0,03
31	5,03	6,02	0,47	0,55	54,82	0,03
32	7,51	4,16	0,69	0,79	39,33	0,04
47	7,55	6,08	0,70	0,83	55,35	0,04
48	10,03	4,68	0,93	1,07	43,75	0,06
63	10,06	6,11	0,94	1,11	55,59	0,06
64	12,54	4,98	1,17	1,35	46,34	0,07
79	12,58	6,13	1,17	1,38	55,74	0,07
80	15,05	5,19	1,40	1,63	48,04	0,08
95	15,09	6,15	1,40	1,66	55,86	0,09
96	17,57	5,34	1,63	1,90	49,26	0,10
104	17,59	5,77	1,64	1,92	52,85	0,10

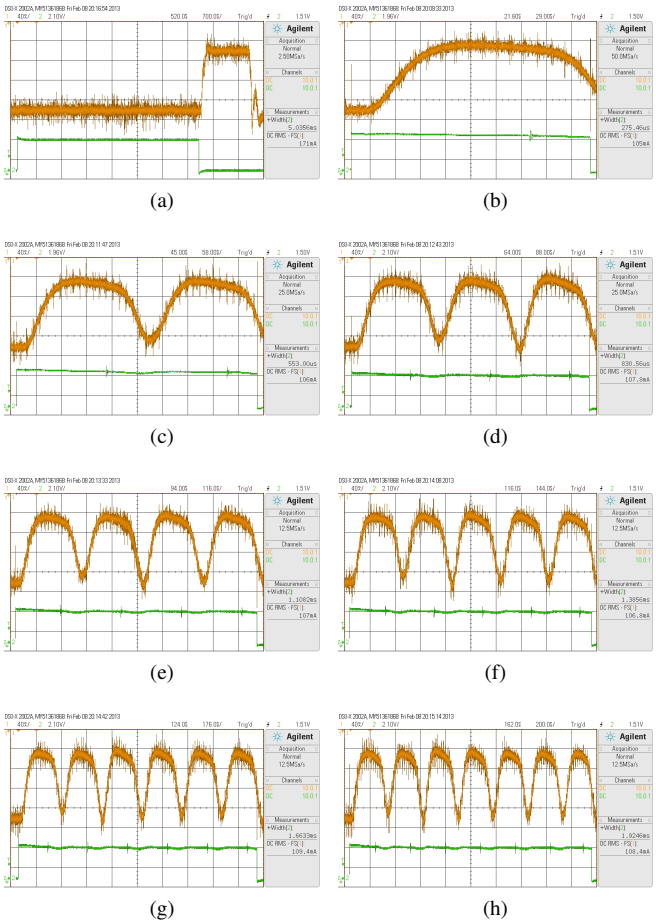

Fig. 6: Sample measurements shown from oscilloscope for variable frame size encryption with software XTEA and hardware AES: (a) *XTEA* 31B payload, (b) *HW-AES* 15B payload, (c) *HW-AES* 31B payload, (d) *HW-AES* 47B payload, (e) *HW-AES* 63B payload, (f) *HW-AES* 79B payload, (g) *HW-AES* 95B payload, (h) *HW-AES* 104B payload

VII. Conclusion and future work

Since we are living in information age, the security and authenticity of provided information is very important issue. This importance grows with needs for sensing and actuation in home and industry environments.

Proposed testbed evaluation was mainly conducted for estimation of the properties and possibilities for optimization of communication in wireless sensor network system using the Lightweight MESH stack from Atmel. In this work, the optimization by means of security features and its impact on the performance and energy consumption was investigated.

From the previous graphs it is obvious that payload length directly influences the performance of encryption process. Low computational power device that is destined for wireless sensor networks was used for testbed. These networks have parameters really limited by the strict requirements for energy consumption. Notable difference in energy consumption of SW and HW implementation is also seen. This testbed evaluation was mainly conducted for investigating the possibilities of encryption power on the low cost microcontroller.

We can say that after considering measured and computed characteristics that were proposed in this work, the HW acceleration has significant impact on the microprocessor performance. HW accelerator is stand-alone module that works in parallel with base microprocessor, which also brings efficiency to the computation.

Energy consumption was used as general rule for evaluation of the optimal approach to the securing the network communication.

Future work will be aimed on the implementation and testing the performance of the other cryptographic methods (in Lighweight MESH stack), especially software implementation of AES-128 because of needed interoperability between various types of hardware devices that can be used in the network system.

References

[1] J. Yu, G. Khan, and F. Yuan, "Xtea encryption based novel rfid security protocol," in *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, may 2011, pp. 000 058 –000 062.

[2] C. Hager, S. Midkiff, J.-M. Park, and T. Martin, "Performance and energy efficiency of block ciphers in personal digital assistants," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, march 2005, pp. 127 – 136.

[3] J. Hernandez and P. Isasi, "New results on the genetic cryptanalysis of tea and reduced-round versions of xtea," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2, june 2004, pp. 2124 – 2129 Vol.2.

[4] Atmel, "Atmega128rfa1: 8-bit avr microcontroller with low power 2.4ghz transceiver for zigbee and ieee 802.15.4," datasheet, 2001. [Online]. Available: http://www.atmel.com/Images/doc8266.pdf

[5] J. Kelsey, B. Schneier, and D. Wagner, "Related-key cryptanalysis of 3-way, biham-des,cast, des-x, newdes, rc2, and tea," in *DES, RC2, and TEA, Proceedings of the 1997 International Conference on Information and Communications Security*. Springer-Verlag, 1997, pp. 233–246.

[6] Atmel, "Lightweight mesh software stack," online source code, September 2012. [Online]. Available: http://www.atmel.com/tools/LIGHTWEIGHT_MESH.aspx

[7] D. Elektronik, "Radio modules derfmega128 22a02—22c02," datasheet, July 2011. [Online]. Available: http://www.dresden-elektronik.de/funktechnik/uploads/media/deRFmega128-22A02-C02-DBT-en.pdf

[8] Atmel, "Atmel avr2130: Lightweight mesh developer guide," online, September 2012. [Online]. Available: http://www.atmel.com/Images/doc42028.pdf