RC5-Based Security in Wireless Sensor Networks: Utilization and Performance

JUHA KUKKURAINEN, MIKAEL SOINI, LAURI SYDÄNHEIMO
Rauma Research Unit, Department of Electronics
Tampere University of Technology
Kalliokatu 2, 26100 Rauma
FINLAND
juha.kukkurainen@tut.fi

Abstract: Information transferred in a wireless sensor network can be sensitive, hence it is vital to secure the data on the network. Enhancing the network's security affects the network operation, by increasing the data's handling time. In this paper, the trade-offs between enhanced security and sensor network performance is discussed. The studies concentrate on the increase in computation time and energy consumption as enhanced security features and levels are utilized. This paper presents, RC5-based encryption and CMAC authentication, used to achieve data confidentiality, freshness, replay protection, authentication, and integrity. These features enhance data security but can also decrease sensor network operability, because of the added load in computation and communication. By selecting a suitable algorithm and operation conditions for encryption and authentication, the data security in wireless sensor networks can be improved with minor resource losses.

Key–Words: consumption, KILAVI sensor network platform, performance trade-offs, wireless sensor network security

1 Introduction

Wireless sensor networks (WSNs) are based on physically small-sized sensor nodes exchanging mainly environment-related information with each other. WSNs have a very wide application area including home control, industrial sensing and environmental monitoring. Sensors typically have very limited power, memory, and processing resources and so interactions between sensors are limited to short distances and low data-rates. Sensor node energy-efficiency and sensor network data-transfer reliability are the primary design parameters.

Security is one other vital aspect in WSN applications. The implementation of security policies is a complex and challenging issue because of resource-constrained nodes. Short transmission distances reduce some of the security threats, but there are risks, for example, related to spoofing, message altering and replaying, and flooding and wormhole attacks [6]. It is important therefore to consider security solutions that guarantee data authenticity, freshness, replay protection, integrity and confidentiality.

Security measures should not significantly affect WSN operations. In Security Protocols for Sensor Networks (SPINS) over 90 % of security-related energy consumption is caused by extra communication [14]. It is estimated that each extra bit transmitted consumes an amount of power equal to that used in executing 800-1000 instructions in the processor [22]. The message size also affects reliability and scalability of the sensor network [19]. These points support the idea that message size, and the number of messages, should be minimized in order to achieve the goals of low-power, simplicity and reliability. There are methods that can be used to keep this sensitive information private. In asymmetric public key cryptosystems each node has a public key and a private key. The public key is published, while the private key is kept secret. Asymmetric public key cryptosystems such as the Diffie-Hellman key agreement or RSA signatures are typically too conservative in their security measures, adding too much complexity and protocol overhead to be usable in WSN solutions. The influence of public key cryptography on the lifetime of a sensor network node is evaluated in [15]. In symmetric cryptography the transmitter and the receiver of a message know and use the same secret key; the transmitter uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. Symmetric solutions are therefore more suitable for resource-constrained sensor networks, though they need special solutions for

security key pre-distribution. [17]. There are numerous key distribution mechanisms available for WSN applications, one is presented in [2].

In this paper, the well-known and well-understood RC5-based encryption and Cipher-based Message Authentication Code (CMAC) authentication are used to achieve security in data transfer. RC5 is a symmetric block cipher that is used in counter mode (CTR). The same simple RC5 algorithm can be used for encryption and decryption. RC5 is also employed in CMAC implementation. For these reasons, RC5 and CMAC are suitable for resource-constrained sensors.

This paper studies how these chosen security features affect sensor and sensor network performance. The focus is on security and performance trade-offs, especially in energy consumption and computation time. KILAVI platform [13, 18, 19, 20, 21] is used as a reference but all the results are easily exploited in other approaches as well. KILAVI is intended for low energy and low data-rate device control and monitoring in buildings, and is comprehensive with regard to the different functions and devices needed to implement an operative building network, with dynamic network set-up and maintenance. Message structure in KILAVI is compact in order to have low overhead protocol, and reduced channel reservation time in transmissions. The risk of transmission collisions increases in relation to channel reservation time.

The rest of this paper is organized as follows. Section 2 is a general discussion about security solutions in WSNs. Section 3 concentrates on chosen encryption and authentication principles. Section 4 introduces the KILAVI platform and its security features. Section 5 describes the cipher implementation. Section 6 presents the computation time used by the security features. Section 7 presents the amount of time and energy the used security features consume in data transmission. Section 8 analyzes the obtained results. Finally, section 9 concludes the paper.

2 Security Solutions in WSN

Although there are encryption algorithms that are faster and lighter compared with RC5, for example RC4, they lack the same degree of security [9]. For this reason it is not advisable to use them in wireless applications where confidential data is transferred between endpoints. For the time being, RC5 can be considered as one of the best ciphers in terms of overall performance, when used in nodes with limited memory and processing capabilities. As an industry standard, RC5 is also utilized in other types of applications [1, 11], and can be used in other types of wireless sensor networks than KILAVI [4]. As the network devices and the microcontrollers in them develop further and gain more memory, more complex ciphers can be utilized with fewer resources sacrificed.

In a building environment, where data rates are low and payloads (states, measurements, control messages etc.) small, requirements for a cipher focus on data security and resource requirements. Memory consumption, time demanded for encryption and decryption operations (bit rotation, addition, multiplication, etc.) and enlarged payload in transmission are all examples of WSN node resource requirements. If the amount of protected data were larger, the operation speed of the cipher would play a more significant role. The RC5 cipher is known for its easy adaptability. Utilizing the RC5 encryption and decryption algorithms in an existing set of instructions is reasonably straightforward. All the necessary functions can be downloaded, modified and implemented free of charge. [16, 17]

Encrypting the message for data transmission is the only way to improve security if the transmission can not be secured physically by using wired transmissions or some other method. Obviously, these needs set limits to the usability of the application and hence are not taken into consideration in WSNs. In WSN solutions, the encryption is usually accomplished by using a symmetric cryptosystem such as RC5 or Advanced Encryption Standard (AES). Just like RC5, AES is also a block cipher and is used widely in ZigBee^{TM} (IEEE 802.15.4) [5]. Although the National Institute of Standards and Technology's (NIST) AES cipher is more widespread than RC5 and has inbuilt hardware support among some microcontroller manufacturers, AES is slower and has higher memory requirements than RC5, which makes RC5 a better cipher solution for devices with limited resources.

The reason why asymmetric-based cryptosystems are not used in WSNs is probably due to their slow execution times. The complexity that is required for making mathematically-paired keys demands calculation time and the extra resource-requirements for two separate keys: public and secret. The public key is used for message encryption and the secret key for decryption. In resource-constrained nodes, it is impractical to use a cryptosystem with high computation, communication and storage overheads. The good thing about asymmetric cryptosystems is that all the devices are digitally signed and hence the authenticity of each message can be noted easily. [14]

3 Security Issues

Section 2 introduced the features of the RC5 cipher and pointed out its benefits. This section presents message encryption and authentication using the RC5 cipher.

3.1 Encryption

Block-cipher type RC5 is a symmetric encryption algorithm that transforms a fixed-length block of unencrypted text into a block of encrypted text of the same length. This transformation takes place under the action of a user-provided secret key. Decryption is performed by applying the reverse transformation to the ciphertext block using the same secret key. [1, 17]

RC5 is suitable for resource-constrained sensor nodes for the following reasons. RC5 is a simple and fast cipher using only common microcontrollers operations; it has a low memory requirement making it suitable for sensor applications; the same lightweight algorithm can be used for both encryption and decryption; and heavy use of data-dependent rotations provides high security. The RC5 block cipher has built-in parameter variability that provides flexibility at all levels of security and efficiency. Table 1 presents the variety of parameters and values used in the RC5 operations. [16]

Table 1: RC5 parameters

Parameters	Values
Word size (w)	16, 32, 64 bits
Block size (2w)	32, 64, 128 bits
The number of rounds (r)	0 - 255
Key length (b)	0 - 2040 bits

There are three basic routines in RC5: key expansion, encryption, and decryption. Key expansion must be performed before the encryption can be initiated. In the key-expansion procedure, the user-provided secret key is expanded to fill a key table whose size depends on the number of rounds. The key table is then used in both encryption and decryption. The encryption routine is based on three basic operations: integer addition, bitwise XOR- (exclusive-or) and variable rotation operations. Figure 1 shows the first half of one RC5 encryption round. The plain text to be encrypted is divided into parts A and B. First, an XOR operation is performed on A and B. A bit rotation operation is executed on the output of XOR and B. Finally, the output of the bit rotation operation is added to the extended encryption key (K). This whole process is then repeated to complete one round. If the number of rounds to be used is 12 then the procedure as shown in Figure 1 is performed 24 times. [16, 17]

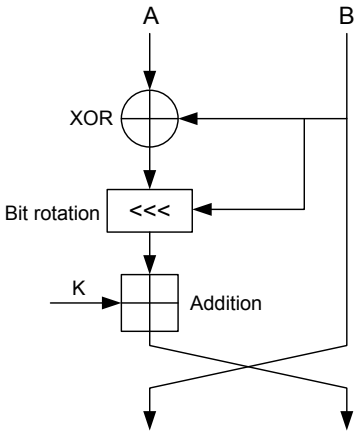

Figure 1: Basic operation of the RC5 algorithm, representing half of one round.

In this work, the RC5 encryption algorithm is used in counter mode (CTR). RC5 guarantees a different character string each time and thus replay protection and data freshness qualities are obtained. CTR-based RC5 encryption does not increase the amount of transmitted bits or the number of sent messages in normal operation. In other words, the lengths of encrypted and plain messages are the same. In CTR mode the node maintains a counter that is increased by one after each successful transmission. The receiving node must be synchronized with the transmitter to be able to decrypt the messages. Figure 2 presents the operation of encryption and decryption in CTR mode. K is the encryption key created in the key expansion procedure, C is the counter value and E is the RC5 encryption algorithm. CTR mode has the following benefits: high speed implementation, simplicity, arbitrary message length and a low rate of error propagation [8].

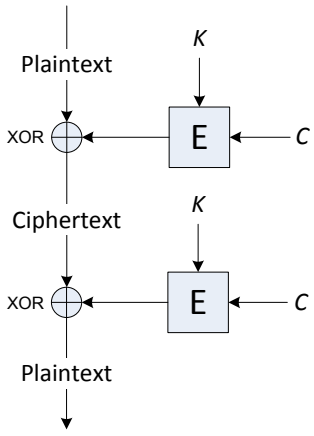

Figure 2: Encryption and decryption in CTR mode.

3.2 Authentication

Authentication may be used to check data integrity and authenticate communicating entities. A data integrity check makes sure that the message has not been altered by an adversary and an authentication check confirms the identity of the transmitter.

In this work, a CMAC algorithm is used for authentication. CMAC is an authentication algorithm defined by NIST. Figure 3 presents the operation principles of CMAC: on the left a situation where message length is an integer multiple of block size and next to it a situation where message length is not an integer multiple of block size.

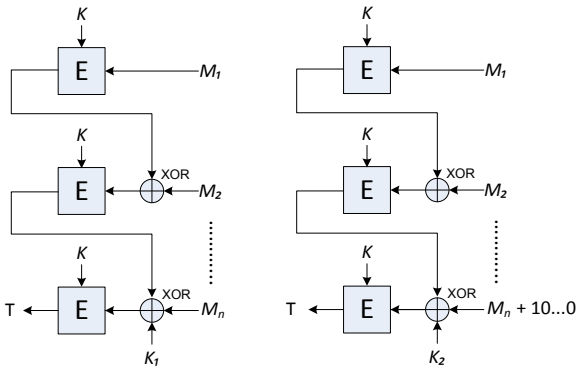

Figure 3: CMAC operation principles.

The CMAC algorithm depends on the choice of an underlying symmetric key block cipher, in this case RC5. The CMAC algorithm is thus a mode of operation of the block cipher (E). The CMAC key is the block cipher key (K); K is used to generate sub keys K_1 and K_2. The message M is divided into n blocks where $M_i\,(i = 1, 2, \ldots, n)$ is a block of the formatted message; and M_n is the last, possibly partial, block. T is the authentication result. [12]

CMAC is a simple variant of the Cipher Block Chaining MAC (CBC-MAC) and fixes its security deficiencies [10]. Whereas the basic CBC-MAC is only secure for messages of one fixed length (and that length must be a multiple of the block size), CMAC is secure across messages of any bit length. The security of MAC is directly related to its length - a suitable value is 32 bits [7].

4 KILAVI Platform

The two previous sections dealt with security in wireless applications and moreover, the RC5 block cipher. This section discusses the KILAVI platform as a workbench for studies on the influence that RC5 ciphering has on time and energy consumption.

4.1 Short Introduction

Building control and monitoring is best performed with application-specific sensor networking. KILAVI is an open manufacturer-independent platform developed for low data rate and low-energy building control and monitoring applications. KILAVI defines a set of functions and messages which are needed to enable co-operative networking between different devices and the common means for data collection and device control tasks. A master controls the network operation and all network nodes are alike. Nodes operate either in sensing mode (Sensor nodes) or forwarding mode (Intermediate nodes) depending, for example, on battery state. The basis for an operational building control platform comprises: centralized hierarchical architecture to enable resource concentration, compact messages to obtain robust networking, 433 MHz operating frequency to gain necessary operating distances with low-power consumption, multihop communication to enable large-scale networks and low-power sensors, and hybrid flooding to provide low overhead network management. KILAVI network architecture is shown in Figure 4. Performance evaluation has shown that this platform is energy-efficient, reliable and low-power. [13, 18, 19, 20, 21]

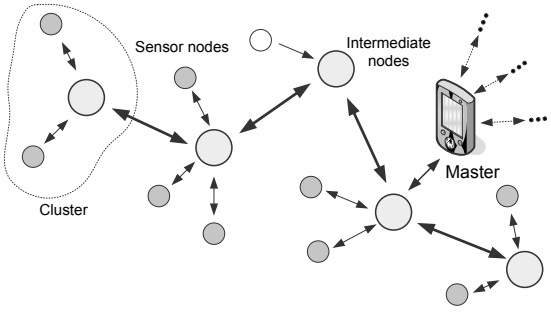

Figure 4: KILAVI network architecture.

4.2 Security in KILAVI

The dominating traffic pattern in WSN is many-to-one where sensors communicate with a central unit. Thus, centralized security architecture and symmetrical end-to-end keys between a master and sensors are a natural choice. In KILAVI, the central node manages, delivers and updates keys with every node. Nodes have one authentication and one encryption key to enable secure communication with the central node. Nodes in forwarding mode do not interpret messages except for the information related to the forwarding or storing of data. This is simple from the sensor node perspective and a low overhead from the networking perspective. [13, 21]

KILAVI uses RC5 encryption in counter mode and the CMAC authentication presented in Section 3. More KILAVI security features including security levels, key exchanging procedure and counter synchronization are presented in [18]. In addition to the secure device registration method presented in [18], KILAVI also utilizes a safety feature where the network master must be set to registration mode by the user for the new device to become a network member. This procedure prevents unwanted devices joining the network without approval from the network caretaker (user). The master's registration state operates in two ways: only one device at a time can join the network and if a registration request is not received by the master within a certain period of time, the master resumes its prior operations. These actions prevent any adversary devices joining the network without user acceptance and hence improve network security.

Table 2 summarizes KILAVI security parameters. The justifications for the selected parameters can be found in detail from [18].

Table 2: KILAVI security parameters.

KILAVI security	Parameters
Word size (w)	16 bits
Block size (2w)	32 bits
Number of rounds (r)	12
Key length (b)	128 bits
Counter length	32 bits
MAC length	32 bits

Further, in KILAVI the increment due to the MAC field is only 32 bits, if security features are used. These very small packet length increments do not significantly affect sensor network reliability as shown in [19].

Memory space is usually very limited in sensor network nodes. The implemented solution with chosen parameters for encryption and authentication is lightweight: RC5 code size is 716 bytes and MAC size is 366 bytes in length. The total program memory size is therefore 1082 bytes.

5 Implementation

The encryption and authentication functions presented were implemented on an 8-bit ATmega644PV-10PU microcontroller unit (MCU). The controller has 64 Kbytes of Flash program memory, 4 Kbytes of RAM, and was running a self-made, event-driven set of instructions written in C [3]. These instructions included a version of the RC5 symmetric algorithm modified from the RC5 reference implementation and the necessary functions needed for making transmissions with the nRF905 radio circuit. These functions included initializations for port operations required for the SPI bus operation and signalling between the microcontroller and the radio.

Comparing the memory requirements, presented previously, with the controller capabilities, it can be noted that only about 1,65 % of program memory was occupied for security purposes. This, of course, is due to the large capacity of the microcontroller and would play a more significant role if less memory was available. Such a memory requirement sets limits for the variety of microcontrollers usable in KILAVI platform and other equivalent applications. [3]

Being more than sufficient for embedded devices in WSNs, RC5 has a downside when used in similar microcontrollers to those used in this paper. In an 8-bit environment, 32-bit data-dependent rotations are slow and hence costly to perform. Therefore, better efficiency can be achieved when 32-bit microcontrollers are used for RC5 ciphering.

6 Effect of Enhanced Security on Computation

Section 3 presented the principles of chosen security measures. This section presents, based on practical measurements, the time consumed by key expansion, encryption and authentication computation operations with different security-related parameters. Section 8 also considers enhanced security from the data transmission perspective.

6.1 Key Expansion

The first routine of RC5 is key expansion. Key expansion is executed before actual encryption and its operation was presented in section 3. Figure 5 shows how much time the key expansion routine for the encryption key consumes with different word lengths as a function of the number of rounds.

6.2 Encryption and Decryption

After the key expansion, encryption and decryption procedures can be initiated in a manner presented in section 3. Figure 6 and Figure 7 present the time taken to encrypt messages of various data length with different word lengths and with different numbers of encryption rounds.

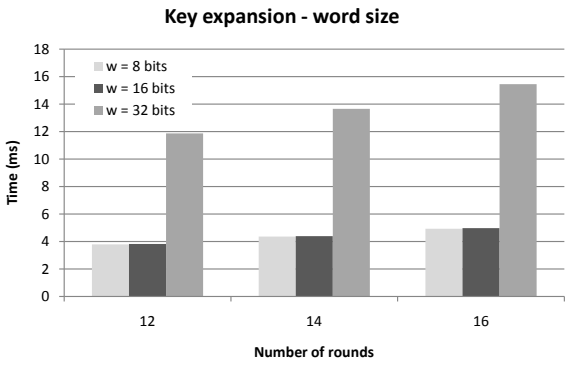

Figure 5: Key expansion time with different word lengths and numbers of rounds (clock speed 8 MHz).

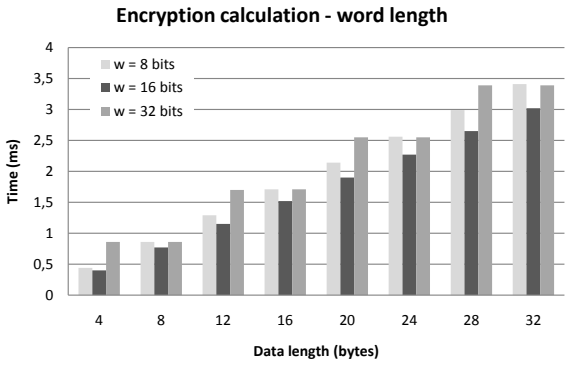

Figure 6: Encryption calculation for different word lengths (clock speed 8 MHz and 12 encryption rounds).

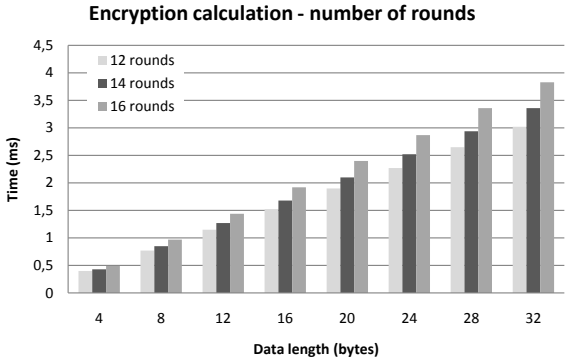

Figure 7: Encryption calculation time for different numbers of rounds (clock speed 8 MHz and word length 16 bits).

6.3 Authentication

Authentication is used to check data integrity and confirm the identity of a sender. Figure 8 and Figure 9 present MAC calculation times for variable message sizes with different word lengths when the microcontroller clock speed is set to 1 MHz and 8 MHz, respectively.

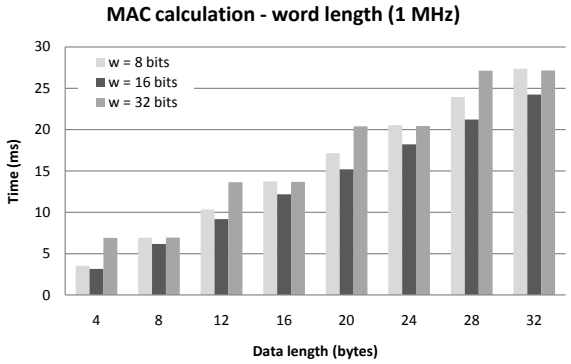

Figure 8: MAC calculation time for different word lengths (clock speed 1 MHz and 12 encryption rounds).

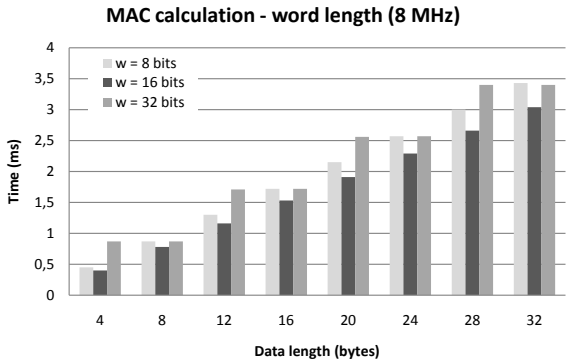

Figure 9: MAC calculation time for different word lengths (clock speed 8 MHz and 12 encryption rounds).

7 Effect of Enhanced Security on Communication

This section considers how selected security features affect sensor data transmission including the computational parts presented in section 6. Computation and transmission time, and sensor energy consumption are studied. Message lengths used are 4 - 32 bytes which are typical in KILAVI.

7.1 Computation and Transmission Time

Figure 10 and Figure 11 show how time is distributed in typical message transmission (with nRF905), between message authentication (MAC calculation, and SPI and RF transmission of 32-bit code), encryption (encryption calculation), and actual data payload (SPI and RF transmission).

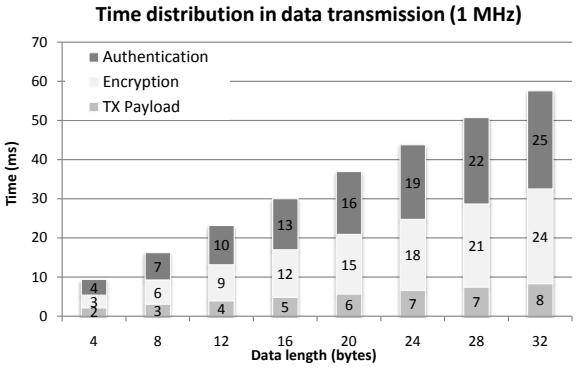

Figure 10: Time distribution in data transmission between MAC, encryption and data payload (clock speed 1 MHz, 12 encryption rounds and 16-bit word length).

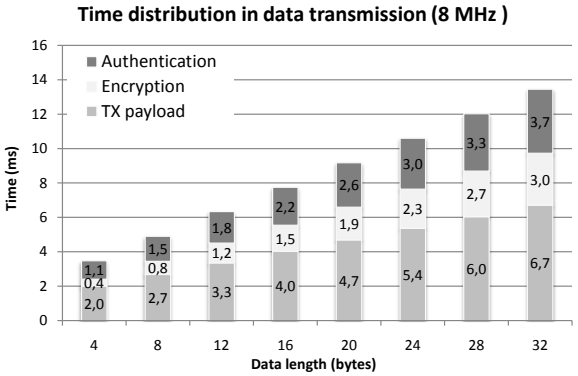

Figure 11: Time distribution in data transmission between MAC, encryption and data payload (clock speed 8 MHz, 12 encryption rounds and 16-bit word length).

7.2 Sensor Node Energy Consumption

Table 3 presents current consumption values used in the following calculations. Current consumption values were measured with prototype sensors (Atmel's ATmega644PV-10PU MCU and Nordic Semiconductor's nRF905 radio).

Table 3: Current consumption of prototype nodes.

Operation	RC5/MAC	SPI	Radio
MCU	active	active	pwr save
SPI	non-active	active	non-active
Radio	pwr down	pwr down	active
I@1 MHz	0,9 mA	1,1 mA	9 mA
I@8 MHz	4,15 mA	4,3 mA	9 mA

By using these measured current consumption (I) values, measured voltage (U) of 2,989 V and previous measured time values (t), energy consumption (E) can be calculated with (1)

$$E = P \cdot t = U \cdot I \cdot t \tag{1}$$

Figure 12 and Figure 13 present energy distribution in typical message transmission between encryption, message authentication, and actual data payload.

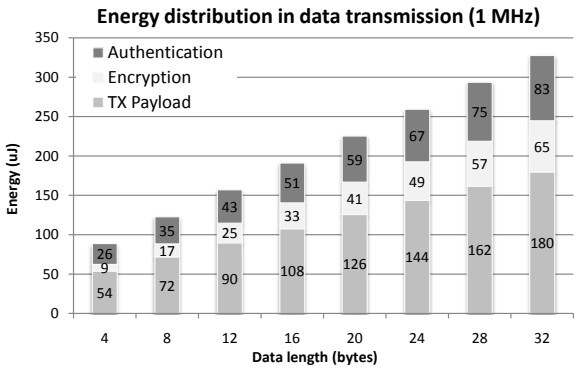

Figure 12: Energy distribution in data transmission between MAC, encryption and data payload (clock speed 1 MHz, 12 encryption rounds and 16-bit word length).

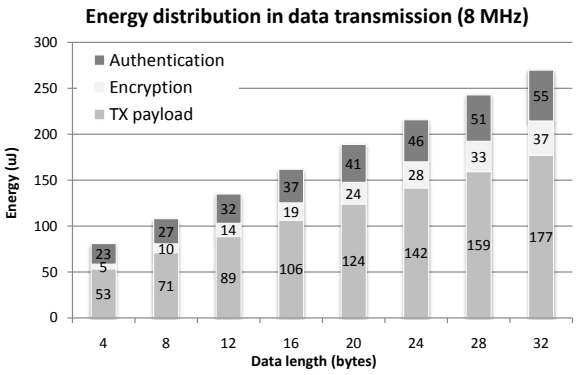

Figure 13: Energy distribution in data transmission between MAC, encryption and data payload (clock speed 8 MHz, 12 encryption rounds and 16-bit word length).

8 Analysis

This section analyzes the results individually presented in sections 6 and 7. More comprehensive analysis of the results is presented in section 9.

8.1 Encryption and MAC Calculations

From Figure 8 and Figure 9, it can be seen that the crystal oscillator frequency is inversely proportional to computation time. Therefore, increasing the oscillator frequency from 1 MHz to 8 MHz decreases the computation times (t) by 87.5 %. At the same time, energy consumption decreases by about 42 % independent of the message length. Energy consumption $(E = P \cdot t)$ does not decrease by the same amount, because the power consumption is approximately 4-times higher with an 8 MHz oscillator speed. The number of rounds affects security strength. From Figure 7 it can be observed that additional rounds (n) increase the computation time by around $\mathrm{n} \cdot 5$ %. Optimal word length can save 10 % to 25 % of computation time depending on the message length (see Figure 6 and Figure 8).

In Figure 6, Figure 8 and Figure 9, the effects of block size can also be seen, as the timed results for the 32-bit word length increase in steps of 8 bytes. These steps are formed by the data lengths of 4 and 8 bytes, 12 and 16 bytes, 20 and 24 bytes and 28 and 32 bytes. This is due to the features of the RC5 block cipher and the fact that a block is the same size as two words.

In the key expansion procedure (see Figure 5), computation time can decrease by as much as 68 %, if optimal word length is used (in this case 16-bits). The interesting part of the measurement for the key expansion is the similar results with word lengths of 8 and 16-bits, whereas the word length of 32-bits consumes a lot more calculation time. The reason for this behaviour is the 8-bit MCU's ability to utilize a few of the 8-bit registers as 16-bit registers. This is accomplished by handling the 16-bit register in 8-bit parts; upper and lower. [3]

8.2 Message Transmission

Here the results are considered from a message transmission perspective. This is important in addition to the security calculations aspect. From Figure 10 and Figure 11, it can be seen that by increasing the oscillator frequency from 1 MHz to 8 MHz, the computation time decreases by 63 % to 77 % depending on message length. Messages tested were from 4 to 32 bytes long. At the same time energy consumption decreases by 8.7 % to 18 %. The data transmission time does not vary as a function of the oscillator frequency (excluding SPI transmission from microcontroller to radio).

8.3 Security versus No-Security: Transmission of a Single Message

In this subchapter, 8 MHz oscillator frequency is used for the reasons given above. From Figure 11, it can be seen that transmission of a single message without security takes 2.0 ms to 6.7 ms, depending on the message length (4 B to 32 B). With the security features operating, this time increases by between 74 % and 100 % (3.5 ms to 13.4 ms), again depending on the message length. From Figure 13, it can be seen that from the energy perspective, transmission of a single message without security consumes 53 to 177 μJ depending on the tested message length. The security features increase energy consumption by 52 % in all cases.

8.4 Security versus No-Security: Sensor Network Operation

In this subchapter, the above results are mirrored in a real-world KILAVI-based data collection system where a sensor wakes up and transmits measured data at specific intervals and otherwise stays in sleep mode (here, $\mathrm{I_{SLEEP}}$ is 10 μA) most of the time. If a sensor transmits messages at a rate of 1 message per second, and security features are used, then the total duration of one operation cycle increases by 0.15 % to 0.67 % and the energy consumption increases by 33 % to 45 % depending on the message length (4 B to 32 B). If a sensor transmits messages at a rate of 1 message per minute, then the total time taken for one operation cycle increases by between 0.003 % and 0.011 % if security features are used, and the increase in energy consumption is between 1.5 % and 4.7 %, both depending

on message length (4 B to 32 B). With longer transmission intervals, the increase in energy consumption caused by the security features is negligible, as seen in Figure 14.

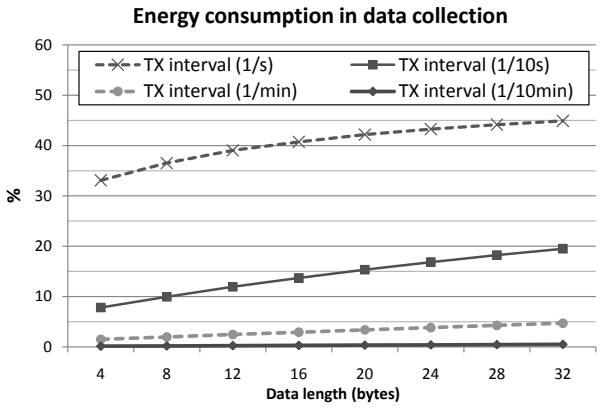

Figure 14: The increase in energy consumption caused by security feature utilization in a data collection system where status information is sent periodically (TX interval).

9 Conclusions

KILAVI uses Rivest's nominal version (RC5-32/12/16) for encryption and decryption, and CMAC for authentication. In this paper, the effects of these security features on WSN operation were studied.

It can be concluded that it is beneficial to use high operation frequencies, in this case from the internal oscillator, if security features are implemented in WSN solutions. Due to high oscillator frequency, MCU's faster operation compensates for the added processing which results from enlarged buffer lengths. Shorter buffer lengths are one reason for faster computation times at sensor nodes and shorter end-to-end delays in multihop communication.

It can be seen from the results that the increase in computation time caused by added security is negligible in sensor networks when using typical transmission intervals. Therefore, KILAVI network nodes in forwarding mode do not become congested due to increased computation and resulting delay. Further, energy-scarce nodes in sensing mode typically operate with large transmission intervals and therefore the energy consumption increase caused by added security is tolerable and does not substantially affect sensor lifetime.

KILAVI uses very short messages. In building control applications messages are not normally frequent and so the utilization of security features presented causes neither congestion in network operation nor significant increases in sensor node energy consumption.

Although we managed to present a lightweight solution using optimized C code, lighter and faster encryption and authentication algorithms are achievable when Assembly code is used in optimization [7]. This area of research can be considered later on when KILAVI matures and more thorough real-life measurements are performed.

None of the results presented in this paper are restricted to the KILAVI platform. Similar results can be achieved with other devices and platforms where an 8-bit microcontroller is used. The measurements did not contain any KILAVI-specific messages or functions and hence the results may be applied more widely. Likely work in the future includes implementation of the security enhancements presented in this paper with an operational KILAVI sensor network.

Acknowledgements: The research was supported by the Finnish Funding Agency for Technology and Innovation.

References:

[1] J.–V. Aguirre, R. Álvarez, J. Sánchez and A. Zamora, Silence Detection in Secure P2P VoIP Multiconferencing, *Proc. 5th WSEAS International Conference on Information Security and Privacy*, Venice, Italy, November 2006

[2] B. Arazi, I. Elhanany, O. Arazi and H. Qi, Revisiting Public-Key Cryptography for Wireless Sensor Networks, *IEEE Computer Magazine*, Vol. 38, No. 11, November 2005.

[3] Atmel Corporation, *ATmega644: 8-bit Microcontroller with 64K Bytes In-System Programmable Flash*, [Online] Available: http://www.atmel.com/dyn/resources/prod_documents/doc2593.pdf

[4] F. Graziosi, L. Pomante and D. Pacifico, A Middleware-Based Approach for Heterogeneous Wireless Sensor Networks, *12th WSEAS International Conference on Communications*, Heraklion, Greece, July 2008

[5] J. A. Gutiérrez, E. H. Callaway Jr and R. L. Barrett Jr, Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4, *Standards Information Network IEEE Press*, USA, 2003, 155 p

[6] C. Karlof and D. Wagner, Secure Routing in Wireless Sensor Networks: Attacks and countermeasures, *Proc. IEEE International Workshop*

on Sensor Network Protocols and Applications, May 2003, pp. 113–127.

[7] C. Karlof, N. Sastry and D. Wagner, TinySec: A Link Layer Security Architecture for Wireless Sensor Networks, *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems*, November 2004.

[8] H. Lipmaa, P. Rogaway and D. Wagner, Comments to NIST concerning AES Modes of Operations: CTR-Mode Encryption, *Symmetric Key Block Cipher Modes of Operation Workshop*, Baltimore, MD, USA, October 2000.

[9] X. Luo, K. Zheng, Y. Pan and Z. Wu, Encryption algorithms comparisons for wireless networked sensors, *IEEE lntemational Conference on Systems, Man and Cybernetics* , 2004

[10] A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, *CRC Press, Inc.*, Boca Raton, FL, USA, 1996.

[11] S. Milanovic and N. E. Mastorakis, Building a Strategic m-Commerce Services Platform, *Proc. 4th WSEAS Internatioal Conference on ISA 2004*, Miami, Florida, April 2004.

[12] National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, *National Institute of Standards and Technology*, Special Publication 800-38B, May 2005.

[13] P. Oksa, M. Soini, L. Sydänheimo and M. Kivikoski, Kilavi platform for wireless building automation, *Energy and Buildings Journal*, Vol. 40, No. 9, 2008, pp. 1721–1730.

[14] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J.D. Tygar, SPINS: Security Protocols for Sensor Networks, *Proc. 7th Annual ACM International Conference on Mobile Computing and Networks*, Rome, Italy, July 2001, pp. 189–199.

[15] K. Piotrowski, P. Langendoerfer and S. Peter, How public key cryptography influences wireless sensor node lifetime, *Proc. 4th ACM workshop on Security of ad hoc and sensor networks*, October 2006.

[16] R.L. Rivest, The RC5 Encryption Algorithm, *Proc. 2nd International Workshop on Fast Software Encryption*, Leuven, Belgium, December 1994, pp. 86–96.

[17] RSA Laboratories, RSA Laboratories: Frequently Asked Questions About Today's Cryptography, *RSA Security Inc.*, Version 4.1, 2000.

[18] H. Sikkilä, M. Soini, L. Sydänheimo and M. Kivikoski, KILAVI Wireless Communication Protocol for the Building Environment - Security Issues, *Proc. 10th IEEE International Symposium of Consumer Electronics*, St. Petersburg, Russia, June-July 2006.

[19] M. Soini, L. Sydänheimo and M. Kivikoski, Reliability and Scalability of Wireless Kilavi Building Control Platform, *Proc. 11th WSEAS International Conference on Communications*, Agios Nikolaos, Crete Island, Greece, July 2007

[20] M. Soini, H. Sikkila, P. Oksa, L. Sydänheimo and M. Kivikoski, KILAVI Wireless Communication Protocol for the Building Environment - Network Issues, *Proc. 10th IEEE International Symposium of Consumer Electronics*, St. Petersburg, Russia, June–July 2006.

[21] M. Soini, J. van Greunen, J. Rabaey and L. Sydänheimo, Beyond Sensor Networks: ZUMA Middleware, *Proc. IEEE Wireless Communication and Networking Conference*, Hong Kong, China, March 2007.

[22] R. Szewczyk, A. Woo, S. Hollar, D.E. Culler and K.S.J. Pister, System architecture directions for networked sensors, *Proc. 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, USA, November 2000, pp. 93–104.