Performance Evaluation of Block Ciphers for Wireless Sensor Networks

Kamanashis Biswas[1], Vallipuram Muthukkumarasamy[1], Xin-Wen Wu[1], and Kalvinder Singh[2]

[1] School of ICT, Griffith University, Gold Coast, Australia
{k.biswas, v.muthu, x.wu}@griffith.edu.au
[2] IBM, Australia Development Lab and Griffith University, Australia
kalsingh@ibm.edu.au

Abstract. Security is one of the major concerns in many Wireless Sensor Network (WSN) applications. A number of cryptographic algorithms have been developed to provide security services in WSNs. However, selecting an energy-efficient and lightweight cipher is a challenging task due to resource constrained nature of sensor nodes. Systematic evaluation of cryptographic algorithms is, therefore, necessary to provide a good understanding of the trade-off between security performance and operational cost. In this paper, we have examined five block ciphers: Skipjack, Corrected Block Tiny Encryption Algorithm (XXTEA), RC5, Advanced Encryption Standard (AES), and Chaotic-Map and Genetic-Operations based Encryption Algorithm (CGEA). The performance of these ciphers is evaluated on Arduino Pro and Mica2 sensor motes. Then the memory usage, operation time, and computational cost are compared. Finally, some recommendations are provided on evaluated block ciphers and implementation platforms.

Keywords: Block ciphers, Memory-and-energy-efficiency, Performance evaluation, Wireless sensor networks

1 Introduction

Wireless Sensor Networks consist of a large number of low-cost sensor nodes (SNs) which are randomly deployed in hostile environments [1]. These battery-powered SNs have low memory, weak processors, and limited communication capabilities. Therefore, to achieve energy efficiency in WSNs, a number of secure and lightweight block ciphers are proposed in the past few years. However, the experimental results show that many of these ciphers have poor performance compared to conventional ciphers [2]. Thus, performance evaluation is important to provide a benchmark of different cryptographic schemes.

The implementation of security policies has to maintain a trade-off between cost and performance. For example, many WSN applications require complex cryptographic algorithms to provide an enhanced level of security. However, the cost increases as powerful SNs are required to implement the crypto-system. Therefore, it is necessary to clearly understand the relationship between implementation cost and effectiveness. Table 1 presents a comparative view of costs, and hardware specifications of a number of commonly used sensor motes. The table shows that EZ430-RF2500 and Arduino Pro motes are less costly but they also have less memory. Hence, performance evaluation of cryptographic schemes on low-cost SNs is necessary to examine the feasibility of cost-effective platforms.

This paper presents an experimental evaluation of cryptographic algorithms based on actual sensor hardware. A number of block ciphers are implemented in Mica2 and

Table 1. Cost and hardware specification (2014)

Sensor Motes	Price (USD)	Package (incl.)	Micro-controller	Bus (bits)	Clock (MHz)	RAM (KB)	Flash (KB)	EEPROM (KB)
SHIMMER	226	2 boards	MSP430F1611	16	4-8	10	48	None
Waspmote	168	4 sensors	ATmega1281	8	14	8	128	4
IRIS	134	No board	ATmega1281	8	8	8	640	4
TelosB	102	3 sensors	MSP430F1611	16	4-8	10	48	1024
Mica2	99	N/Av.	ATmega128L	8	8	4	128	512
EZ430-RF2500	40	Board only	MSP430F2274	16	16	1	32	None
Arduino Pro (328)	25	nrf24L01 radio	ATmega328	8	8-16	2	32	1

Arduino Pro mote platforms in order to compare the memory efficiency, computational cost, and operation time. Finally, based on the experimental results, some critical insights are provided that will be useful to choose the best cryptographic algorithm and implementation platform.

The rest of the paper is organized as follows: Section 2 describes the related works. Section 3 presents an overview of evaluated block ciphers in WSNs. Section 4 details the implementation platforms. Performance evaluation and analysis are presented in Section 5. Section 6 discusses and concludes the paper.

2 Related Work

Law et al. [3] present a systematic evaluation framework that considers both security properties, and memory-and-energy-efficiency of the block ciphers for WSNs. The authors recommend Rijndael cipher to use for security and energy efficiency, whereas MISTY1 is suggested for storage and energy efficiency.

Passing et al. [4] use a testbed to analyze the performance of cryptographic hash functions (MD5 and SHA-1) as well as of encryption algorithm (AES). The results indicate that the cryptographic algorithm needs high execution times, for example, 1.67s for a single AES operation on a 1KB array.

A comparative performance analysis [5] of RC6, AES, and Scalable Encryption Algorithm (SEA) shows that SEA requires less memory compared to AES and RC6 ciphers, whereas AES and RC6 achieve best performance in terms of execution time and bandwidth usage respectively.

The computational energy cost of symmetric key ciphers is derived and compared with respect to different block and payload size [6]. Furthermore, the authors recommend to use the Byte-oriented Substitution-Permutation Network (BSPN) cipher to provide both security and energy efficiency.

The effects of symmetric block ciphers on WSN performance and behaviour are analyzed to identify critical network parameters in [7]. AES, RC5, and Skipjack ciphers are implemented on MicaZ and TelosB motes as well as important trade-offs are provided both qualitatively and quantitatively.

Trad et al. [8] measured and compared the memory efficiency, operation time, and energy consumption of AES, RC5, and RC6 algorithms in Mica2 sensor motes. The experimental outcomes show that RC5 is the most suitable block cipher in terms of time and energy-efficiency.

In addition to conventional cryptographic algorithms, several lightweight block ciphers such as HIGHT [9], Simple Lightweight Encryption Scheme [10], and Lightweight Security Protocol [11] are implemented on Mica2 motes. These algorithms are energy-efficient and provide a good level of security in WSNs.

This work implements a number of block ciphers on two different hardware platforms and investigates the security performance and operational cost. The experimental results show the effectiveness of the evaluated block ciphers.

Table 2. Cipher parameters used in experiments

Block Ciphers	Key Length (bits)	Rounds	Block Length (bits)
Skipjack	80	32	64
XXTEA	128	14	64
RC5	128	14	64
AES-128	128	10	128
AES-256	256	14	128
CGEA	256	N/Av.	128

3 Overview of Implemented Block Ciphers

Skipjack, XXTEA, RC5, AES, and CGEA ciphers are described in this Section. Table 2 lists the parameters adopted for each block cipher in our experiments.

Skipjack uses an 80-bit key over 64-bit data blocks. It implements an unbalanced Feistel network with 32 rounds. Biiham et al. presented an attack against 31 of 32 rounds using impossible differential cryptanalysis [12]. Moreover, short key length makes Skipjack vulnerable to exhaustive key search attack.

XXTEA has 64-bit block length and 128-bit key length. It implements an unbalanced Fiestel network with variable number of rounds (6 to 32 full cycles). The last reported attack against the full-round XXTEA presents a chosen plaintext attack based on differential cryptanalysis using 2^{59} queries [13].

RC5 is a flexible block cipher with variable parameters: block size (32, 64, or 128-bits), key size (0 to 2040-bits), and number of rounds (0 to 255). It is a widely used block cipher in WSNs. However, 12-round RC5 with 64-bit blocks is vulnerable to a differential attack using 2^{44} chosen plaintext [14].

AES is an iterative block cipher based on a substitution-permutation network and has 128 bits fixed block size. It operates on a 4 × 4 array of bytes. AES running on 10, 12, and 14 rounds for 128, 192, and 256-bits key respectively is vulnerable [15].

CGEA is a lightweight block cipher that uses chaotic map to generate pseudorandom bit sequence [16]. The 256-bit blocks of the sequence is used as key to encrypt or decrypt 128-bit data blocks. The algorithm implements XOR, mutation, and crossover operations on plaintext to generate the ciphertext. Instead of using rounds, it performs crossover operation for each byte of data in plaintext.

4 Implementation Environmnet

4.1 Hardware Specification

Arduino Pro is a microcontroller board based on ATmega168/328 [17]. In experiments, USB powered Arduino Pro (328) motes are used with following configurations: Operating voltage - 3.3V, Clock speed - 8MHz, RAM - 2KB, FLASH - 32KB, EEPROM - 1KB, Radio unit - nrf24L01, and Data rate - 19.2 Kbps.

Mica2 is a low-power sensor mote based on ATmega128L processor [18]. USB powered Mica2 motes are used in the experiments with following configurations: Operating voltage - 3.3V, Clock speed - 8MHz, RAM - 4KB, Flash - 128KB, EEPROM - 512KB, Radio unit - CC1000, and Data rate - 19.2 Kbps.

4.2 Software Specification

The source code of each block cipher is written in Arduino IDE to compile and upload on Arduino Pro motes. In our experiments, two built-in library functions (*microsecondsToClockCycles()*, and *Serial.print()*) are used to obtain and print CPU cycles and encryption time.

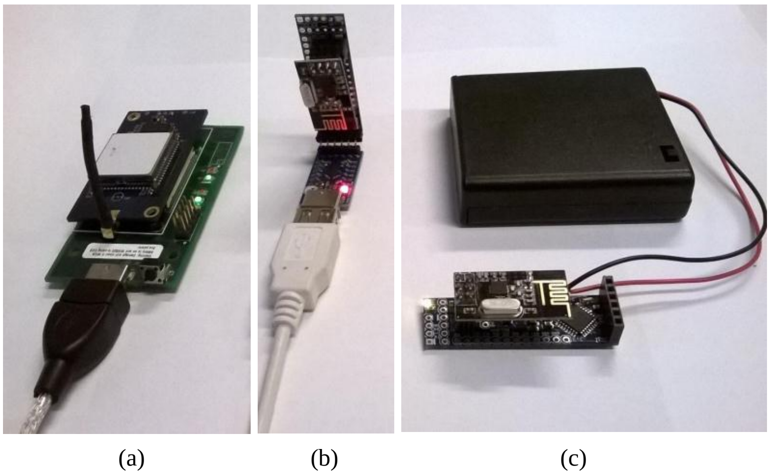

Fig. 1. Experimental setup: (a) USB powered Mica2 mote, (b) USB powered Arduino Pro mote with programmer board, (c) Battery powered Arduino Pro mote

A high-level component-based programming language (nesC) [19] is used to implement the ciphers on Mica2 motes. The *LocalTime.get()*, and *prinf()* functions are used to get the execution time, whereas the CPU cycles are obtained by using the ATEMU [20].

Finally, the avr-size and avr-objdump utilities are used to measure the memory usage on Arduino Pro and Mica2 motes respectively. These two utilities display the header information of object files. The information includes the size of RAM and ROM in terms of text, data, and bss section.

5 Performance Evaluation and Analysis

This Section presents a comparative performance analysis of optimized Skipjack, XXTEA, RC5, AES, and CGEA block ciphers implemented on Mica2 and Arduino Pro motes. To make the comparison, three crucial parameters have been selected: memory consumption, computational cost, and operation time.

Table 3. Memory consumption (bytes)

Block Ciphers	Mica2		Arduino Pro	
	RAM	ROM	RAM	ROM
Skipjack	3096	8658	398	4952
XXTEA	542	6312	226	4112
RC5	682	6110	350	3184
AES-128	1074	6296	814	3692
AES-256	1822	7932	1014	4190
CGEA	664	6268	548	3228

Table 4. Computational cost (cycles)

Block Ciphers	Mica2	Arduino Pro
Skipjack	9820	12672
XXTEA	24064	30464
RC5	53014	61504
AES-128	37525	43200
AES-256	80344	88896
CGEA	67786	76212

5.1 Memory Consumption

Memory consumption is a significant performance metric that can be used to select encryption algorithms with less memory overhead. Table 3 shows the amount of memory consumed by each block cipher on Mica2 and Arduino Pro platforms. It can be seen that Skipjack and AES-256 require more memory compared to other algorithms. The memory requirements of AES-128 is slightly lower compared to AES-256, whereas RC5 is the lightest among all algorithms.

Critical observations- Both Skipjack and AES uses a big S-box of 256-bytes, and as a result the algorithms occupy a significant amount of memory. XXTEA, RC5, and CGEA require less memory for execution and is, therefore, suitable for memory constrained SNs like Arduino Pro. One important observation is that the implementation of AES-256 on Arduino Pro mote shows a message regarding low available memory. Therefore, cryptographic algorithms that use excessive memory may experience stability problem on Arduino Pro platform.

5.2 Computational Cost

The energy efficiency of an algorithm can be calculated from it's computational complexity. Assuming the energy consumption per CPU cycle is fixed, the amount of consumed energy per byte can be computed by measuring the number of CPU cycles required to process one byte of plaintext. However, Table 4 shows the total number of CPU cycles required by each algorithm to encrypt 32 bytes data. It can be seen that Skipjack is the most energy efficient block cipher, whereas the performance of AES-256 is worst among all algorithms. It is also noted that AES-128 performs two times better than AES-256.

Critical Observations- The key size and number of rounds play a significant role in computational complexity. The implementation of AES-128 block cipher reduces more than half of computational cost required by AES-256 due to small size of key and less number of rounds. It is also noted that RC5 consumes more CPU cycles compared to AES-128 in spite of having the same size key. The reason is that RC5 executes 14 rounds, whereas AES-128 uses 10 rounds only.

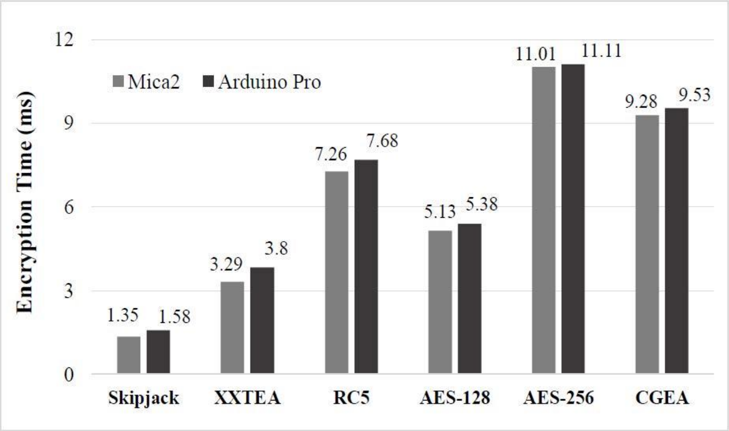

Fig. 2. Encryption Time

5.3 Operation Cost

Operation speed indicates time efficiency and is defined in terms of encryption time and communication time. Encryption time is the amount of time spent to encrypt the plaintext, whereas the time required to encrypt and successfully send the ciphertext is defined as communication time. Fig. 2 shows the execution time required to encrypt 32 bytes data. It can be seen that Skipjack is more than 7 and 6 times faster compared to AES and CGEA ciphers respectively. In addition, AES-128 cipher reduces more than half of AES-256 encryption time. The same results are obtained for communication time experiment as shown in Fig. 3.

Critical observations- Skipjack algorithm is most efficient since it generates the shortest expanded key among all block ciphers. Similarly, the use of 128-bits key in AES-128 shows better performance compared to AES-256 cipher. XXTEA also requires low encryption time since the cipher is structured with simple XOR and shift operations. The longer word size (32-bit) leads to longer execution times for both key setup and encryption phases in RC5. The CGEA block cipher also takes significant amount of encryption time to perform crossover operations by repeatedly swapping the values at different memory locations.

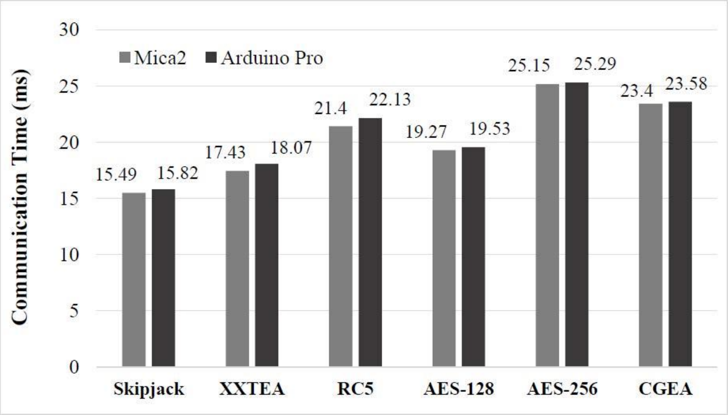

Fig. 3. Communication Time

6 Discussion and Conclusions

According to the experimental results, RC5 is the most memory efficient block cipher. XXTEA and CGEA are also potential candidates for memory constrained SNs like Arduino Pro. On the other hand, Skipjack shows best performance in terms of operation time and computational cost. XXTEA and AES-128 ciphers also consume low energy. However, from a security perspective, Skipjack is a high risk algorithm because of shorter key length. Similarly, XXTEA and RC5 are vulnerable to a number of security attacks such as timing attack, and chosen plaintext attack. Moreover, 128-bits key is not secure against quantum attack. The quantum computing systems are able to break 128-bits key with time 2^{64} [21]. However, AES-256 and CGEA would still be secure against exhaustive search due to 256-bits key length. Therefore, we recommend to use AES-256 or CGEA block ciphers when security is a priority. RC5, XXTEA, and AES-128 ciphers can be used for the applications that require minimum level security.

This paper presents a comparative performance analysis of Skipjack, XXTEA, RC5, AES, and CGEA block ciphers. It is noted that Arduino Pro requires slightly more execution and communication time compared to Mica2 which is negligible. Therefore, it will be cost-effective to use Arduino Pro for common WSN applications such as environmental monitoring instead of time critical and high-memory-demand applications. Our future works will evaluate the performance of stream ciphers and compare the results with block ciphers.

References

1. Biswas, K., Muthukkumarasamy, V., Sithirasenan, E.: Maximal clique based clustering scheme for WSNs. In: IEEE ISSNIP, pp. 237--241, Melbourne (2013)
2. Cazorla, M., Marquet, K., Minier, M.: Survey and benchmark of lightweight block ciphers for WSNs. In: SECRYPT, (2013)
3. Law, Y., and Doumen, J., Hartel, P.: Survey and benchmark of block ciphers for WSNs. ACM Trans. Sen. Netw. 2(1), 65--93 (2006)
4. Passing, M., Dressler, F.: Experimental performance evaluation of cryptographic algorithms on sensor nodes. In: IEEE MASS, pp. 882--887, Vancouver (2006)
5. Çakırolu, M., Bayilmi, C., Özcerit, T., Çetin, Ö.: Performance evaluation of scalable encryption algorithm for WSNs. J. of Sci. Research and Essays. 5(9), 856--861 (2010)
6. Zhang, X., Heys, H. M., Li, C.: Energy efficiency of symmetric key cryptographic algorithms in WSNs. In: QBSC Symposium, pp. 168--172, Kingston (2010)
7. Antonopoulos, Ch. P., Petropoulos, Ch., Antonopoulos, K., Triantafyllou, V., Voros, N. S: The effect of symmetric block ciphers on WSN performance and behavior. In: IEEE WiMob, pp. 799--806, Barcelona (2012)
8. Trad, A., Bahattab, A. A., Othman, S. B.: Performance trade-offs of encryption algorithms

for WSNs. In: WCCAIS, pp. 1--6, Hammamet (2014)

9. Koo, W. K., Lee, H., Kim, Y. H., Lee, D. H.: Implementation and analysis of new lightweight cryptographic algorithm suitable for WSNs. In: ISA, pp. 73--76, (2008)
10. Biswas, K., Muthukkumarasamy, V., Sithirasenan, E., Singh, K.: A simple lightweight encryption scheme for WSNs. In: Chatterjee, M., Cao, J., Kothapalli, K., Rajsbaum, S. (eds.) ICDCN. LNCS, vol. 8314, pp. 499--504. Springer, Heidelberg (2014)
11. Biswas, K.: Lightweight security protocol for WSNs. In: International Symposium on WoWMoM, pp. 1--2, Sydney (2014)
12. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: J. of crypt. 18(4), 291--311 (2005)
13. Yarrkov, E.: Cryptanalysis of XXTEA, http://eprint.iacr.org/2010/254.pdf.
14. Biryukov, A., Kushilevitz, E.: Improved cryptanalysis of RC5. In: N., Kaisa (eds.) EUROCRYPT'98. LNCS, vol. 1403, pp. 85--99. Springer, Heidelberg (1998)
15. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full AES. In: Lee, D. H., Wang, X. (eds.) ASIACRYPT. LNCS, vol. 7073, pp. 344--371. Springer, Heidelberg (2011)
16. Biswas, K., Muthukkumarasamy, V., Singh, K.: An encryption scheme using chaotic map and genetic operations for WSNs. J. IEEE Sens. 15(11), 2801 -- 2809 (2015)
17. Arduino Inc.: Arduino Pro. https://www.arduino.cc/en/Main/ArduinoBoardPro.
18. Crossbow Technology: MICA2, Wireless Measurement System. http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/ mica2.pdf.
19. Gay, D., Levis, P., Behren, R., Welsh, M., Brewer, E., Culler, D.: The nesC Language: A holistic approach to networked embedded systems. ACM SIGPLAN Notices. 38(5), 1--11 (2003)
20. Polley, J., Blazakis, D., Mcgee, J., Rusk, D., Baras, J. S.: ATEMU: a fine grained sensor network simulator. In: IEEE SECON, pp. 145--152, (2004)
21. Wu, D.: Introduction to cryptography. In: Lecture Notes, Stanford University, http://crypto.stanford.edu/~dwu4/notes/CS255LectureNotes.pdf

View publication stats