Energy Efficiency of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks

Xueying Zhang, Howard M. Heys, and Cheng Li
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, NL, A1B 3X5, Canada
Email: {xueying.zhang, hheys, licheng}@mun.ca

***Abstract*— In this paper, we examine the energy efficiency of symmetric key cryptographic algorithms applied in wireless sensor networks (WSNs) and in our study we consider both stream ciphers and block ciphers. We derive the computational energy cost of the ciphers under consideration by comparing the number of CPU cycles required to perform encryption. After evaluating a number of symmetric key ciphers, we compare the energy performance of stream ciphers and block ciphers applied to a noisy channel in a WSN. In conclusion, we recommend using a lightweight block cipher referred to as byte-oriented substitution-permutation network (BSPN), to achieve energy efficiency with a level of security suitable for wireless sensor networks.**

Keywords - wireless sensor networks; security; cryptographic algorithm; stream cipher; block cipher

I. Symmetric key Cryptography in WSNs

Selection of a suitable security scheme is critical in wireless sensor networks (WSNs) because of the open media broadcast communication and the limited energy supply of the sensor device [1]. To achieve the security requirements, several researchers have focused on evaluating cryptographic algorithms in WSNs [2][3] and proposing energy efficient ciphers [4][5]. Although the transmission of data is the most energy consuming activity in a wireless sensor node, it is also important to select an energy efficient cipher that will minimize the energy consumption of the energy constrained sensor node.

In this paper, we evaluate the energy performance of both stream ciphers and block ciphers. After comparing the number of CPU cycles of each cipher, we further analyze the ciphers when they are applied in a WSN with a poor channel quality. Our analysis results show that the lightweight block cipher, referred to as byte-oriented substitution-permutation network (BSPN), is the most energy efficient cipher among all the candidate symmetric key ciphers.

A. Security Requirements and Cryptography in WSNs

In WSNs, four major security requirements are integrity, confidentiality, authentication, and freshness [6]. To prevent the network from being attacked, a security scheme should be capable of protecting each data packet within the network from being eavesdropped (confidentiality), altered (integrity), spoofed (authentication), and replayed (freshness). Encryption is used to ensure the confidentiality. A message authentication code (MAC), functioning as a secure checksum, provides the data integrity and authentication in the network.

Symmetric key ciphers and asymmetric key ciphers are the two fundamental categories of ciphers. The security of asymmetric cryptography depends on the difficulty of a mathematical problem and the resulting algorithm consumes considerably more energy than symmetric key ciphers, which are constructed by iteratively applying simple cryptographic operations. Hence in WSNs, the symmetric key cipher is typically utilized to encrypt data during the transmission of sensor data, conforming to the limited energy source in the sensor device.

B. Symmetric Cryptography in WSNs

In WSNs, energy limitations make the security schemes focus on ciphers with efficient computational energy consumption. The two types of symmetric key ciphers, block ciphers and stream ciphers, have different features.

1) Size of Encryption Operands

The size of encryption operands is different for stream ciphers and block ciphers. Stream ciphers typically operate on one bit of plaintext data to produce one ciphertext bit. This is typically achieved by XORing plaintext bits with a pseudorandom sequence of bits called the keystream to produce the ciphertext bits. In contrast, block ciphers operate on a block of plaintext bits (typically, 64 bits or 128 bits) at one time to produce a block of ciphertext bits. When encrypting a large sequence of plaintext bits, stream ciphers can efficiently operate on variable lengths, while block ciphers may need to pad plaintext out to have a length that is a multiple of the block size. In WSNs, the resulting extra ciphertext bits result in increased transmission energy cost of the sensor node.

2) Security Considerations

Although the Advanced Encryption Standard (AES) [7] is the most widespread block cipher and is considered secure, other block ciphers with a suitable level of security can be also used considering the specific application environment. Compared to block ciphers, stream ciphers have not gained widespread confidence in their security strengths. However, stream ciphers are still being used in wireless communications

This work is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and funds from the Wireless Communications and Mobile Computing Research Center (WCMCRC) of Memorial University

due to their fast operation and flexible implementation.

3) Operation Modes

A mode of operation is used when encrypting a bulk of data by a block cipher. Five basic modes of operation for block ciphers include electronic codebook (ECB) mode, cipher block chaining (CBC) mode, cipher feedback (CFB) mode, output feedback (OFB) and counter mode [8]. Some modes are block oriented, such as ECB and CBC, which operate on plaintext of multiple block size. In contrast, some modes make the block cipher function as a stream cipher, such as CFB mode, OFB mode and counter mode, which can encrypt the plaintext bit by bit. Note that the mode of operation determines the size of ciphertext unit, which relates to the communication energy cost in a WSN.

4) Key Setup

Usually there is a key setup period included in the operation of symmetric key ciphers. For example, AES has a key expansion phase to generate round keys from the cipher key. Although the detailed operation of key setup is different for different ciphers, it can be complicated and take a relatively long time to finish. In WSNs, key setup is very infrequent since it will only follow the establishment of a new cipher key.

5) Keystream Setup

Keystream setup or setup of initialization vector (IV) takes place in symmetric key ciphers whenever a new IV is established. Stream ciphers periodically must re-initialize the keystream based on an updated IV to ensure that the transmitter and receiver are synchronized in their encryption and decryption processes, respectively. For block ciphers, all modes of operation, except ECB mode, use IVs, which are periodically updated to establish synchronization between encryption and decryption. When a block cipher is used in a stream cipher mode such as counter mode, the setup of the IV is equivalent to keystream setup. The energy cost of keystream setup for stream ciphers is much higher than that of block ciphers, which will be specifically discussed later in this paper. In a WSN, the energy cost of keystream setup is especially critical in the selection of stream ciphers and block ciphers.

C. Energy Efficiency of the Symmetric Key Cipher

The encryption computational energy cost of symmetric key ciphers (E_{enc}) for N_{PL} bits of plaintext can be calculated by

$$E_{enc} = (P_{cpu} \times C_{enc}/f_{cpu}) \times \lceil N_{PL}/u \rceil, \qquad (1)$$

where P_{cpu} and f_{cpu} are the power and frequency of the CPU, respectively, and C_{enc} is the number of CPU cycles required to perform an encryption of a block of size u. For a block cipher, u is the block size, while for stream ciphers u represents the keystream block size which is the amount of keystream produced at one time. The symbol $\lceil . \rceil$ denotes the ceiling operator. We can see that C_{enc} and u determine the computational energy cost of the cipher in the same CPU environment. For convenience, we use the number of CPU cycles per byte to evaluate the energy efficiency of the symmetric key ciphers. The number of cycles is obtained from an implementation of the cipher in assembly language on the ATmega128 CPU, a popular 8-bit microcontroller, which has been chosen in wireless sensor devices such as MICA2 [9].

II. Block ciphers in WSNs

A. Block Cipher Overview

In WSNs, a sensor node is an energy limited device, which transmits low entropy information with limited life period for many applications. A lightweight block cipher, with small block size and key size, is appropriate for this purpose, as it is energy efficient and provides sufficient security in WSNs. We choose four block ciphers as candidates and consider the different energy performance when applied in WSNs. The selected block ciphers are AES [7], Skipjack [10], Puffin [11] and BSPN[1] [12]. Characteristics of these ciphers are shown in Table I.

TABLE I. Characteristics of block ciphers

Block cipher	Block size	Key size	# Rounds
AES	128 bits	128 bits	10
Skipjack	64 bits	80 bits	32
Puffin	64 bits	128 bits	32
BSPN	64 bits	≥ 64 bits[2]	8

1) Advanced Encryption Standard

AES [7] is the most popularly deployed symmetric key cipher. Although, as we shall see, the energy cost per byte of AES is high, it is generally regarded as a secure choice when selecting ciphers for security schemes.

2) Skipjack

Skipjack [10] is designed to take the place of the Data Encryption Standards (DES). It is utilized for WSNs in the TinySec scheme [13] due to its energy efficiency. However, some research has shown that Skipjack has security weakness under certain cryptanalyses [14][15].

3) Puffin

Puffin [11] is a recently proposed compact block cipher designed for hardware implementations. Puffin can resist differential and linear cryptanalysis and it is also resistant to related-key attacks and weak keys, which are two main insecurities of the key schedule.

4) Byte-wise SPN

Byte-wise SPN (BSPN) is a compact block cipher we suggest to use in WSNs [12], which provides moderate security to the energy limited environment. It has no apparent weaknesses and is resistant to both the differential and linear cryptanalysis attacks [12]. In the next section, we will show that it provides good energy performance applied in WSNs compared to other candidate block ciphers.

BSPN is a block cipher with an 8 byte block size and 64-bit (or larger) key size. It has 8 rounds of operation and each round of operation includes add round key, substitution and linear transformation as shown in Fig. 1. It uses an 8×8 S-box, which functions as a nonlinear transformation between 8 bits of input and 8 bits of output. The result of the linear transformation, U, is achieved by bitwise XORing the output bytes, V_i, of the other seven S-boxes after adding the round key, i.e., for byte i

$$U_i = \oplus_{j=1, j \neq i}^{8} V_j, 1 \leq i \leq 8. \qquad (2)$$

In the figure, the connection between each component represents one byte of data and S represents an 8×8 S-box.

[1] The cipher proposed in [12] is not named. In this paper we label the cipher as BSPN.

[2] The key size of BSPN could be easily extended to 80 or 128 bits.

The "Add Key" operation is achieved through bitwise XOR of the 64-bit data and the 64-bit round key. A description of the method to generate the round keys is given in [12].

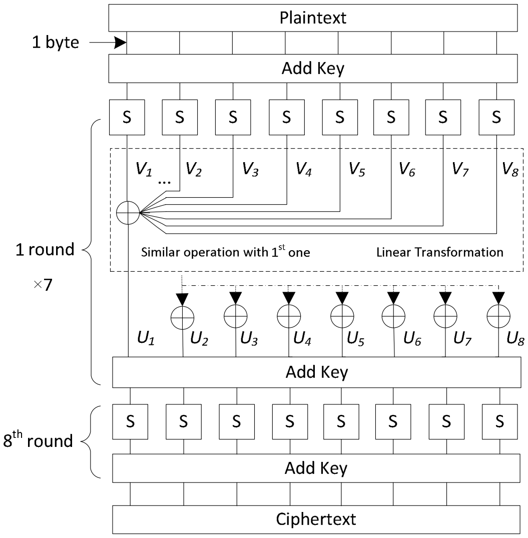

Fig. 1 Structure of cipher BSPN.

B. Implementation Comparison

We have implemented the four block ciphers in assembly language on the ATmega128 processor and the results are shown in Table II. We can see that BSPN requires the fewest number of CPU cycles per byte among the four block ciphers and thus has the lowest computational energy cost. The relationship between the CPU cycles and the size of plaintext is illustrated in Fig. 2. The total number of cycles to encrypt is based on operating the cipher in ECB mode. However, results for other modes, such as CBC mode or counter mode would be very similar. In this figure, BSPN achieves the best energy efficiency for all plaintext sizes. The number of CPU cycles of Puffin is noticeably higher than others because of its hardware design purpose. AES achieves a better result than Puffin, while slightly worse than the other two ciphers. Note that although the performance of Skipjack is slightly better than AES, it is vulnerable to cryptanalysis.

TABLE II. IMPLEMENTATION RESULT OF DIFFERENT BLOCK CIPHERS.

Block cipher	Block size (bits)	Cycles per block	Cycles per byte
BSPN	64	796	99
Skipjack	64	1482	186
AES	128	3266	204
Puffin	64	43418	5427

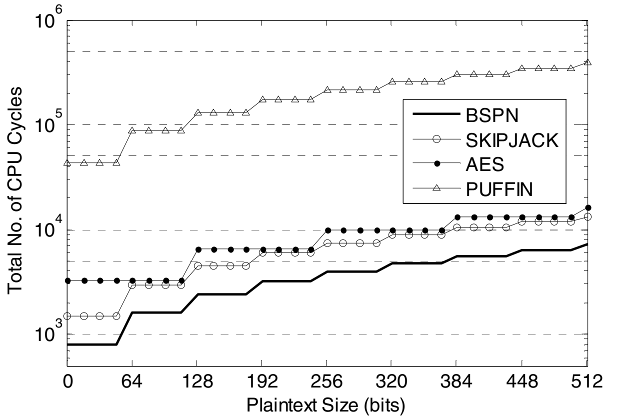

Fig. 2 Computational costs of block ciphers.

C. Discussion

Since the cipher is implemented by software (i.e., by assembly language), the number of CPU cycles is directly related to the architecture of the block cipher and instruction set of the CPU. The characteristics of the block cipher's substitution and linear transformation are two critical factors affecting the number of CPU cycles. These characteristics are summarized in Table III. The "Linear Trans. Unit" refers to the basic unit manipulated by the linear transformation. BSPN achieves the best energy performance among the four ciphers because of its efficiency of substitution and linear transformation for an 8 bit CPU. Note that for an 8 bit CPU, the number of cycles to change a value of one byte compared to changing one bit of the byte is the same, because they both include loading the byte value from the memory and writing it back.

TABLE III. CHARACTERISTICS OF SUBSTITUTION AND PERMUTATION.

Block cipher	Structure	S-box	Linear Trans. Unit
BSPN	SPN	8×8	8 bits
Skipjack	Feistel	8×8	8 bits
AES	SPN	8×8	8 bits
Puffin	SPN	4×4	1 bit

Puffin, as block cipher aimed for compactness, was designed for hardware purposes making it not appropriate for the software implementation. BSPN's use of the 8×8 S-box makes the substitution value look-up convenient in an 8 bit CPU. In contrast, a 4×4 S-box used in Puffin requires 2 memory accesses to update one byte of data during the substitution. Although in Puffin we can reconstruct the S-box to make it appropriate for the byte operation, extra memory needs to be allocated. BSPN performs a linear transformation on the data by XORing the output bytes of other S-boxes directly, which greatly reduces the complexity on an 8 bit CPU compared to the cipher Puffin, whose linear transformation is structured on a bitwise basis. Although part of the linear transformation can be optimized, it cannot change the result of taking a large number of cycles to execute.

The implementation of AES is an efficient 8-bit implementation based on using the "xtime" operation for MixColumns [16]. As shown in the table, both AES and BSPN have the 8×8 S-box and byte oriented linear transformation. The reason that AES costs 105 cycles more than BSPN per byte is because of the algorithm complexity of AES. It should be noted, however, that AES is designed for a 128-bit level of security for both key and block size, while BSPN is designed for a 64-bit level of security both in key and block size.

III. STREAM CIPHERS IN WSNS

We have selected three stream ciphers to compare: RC4, Sosemanuk and Salsa. RC4 is a popular stream cipher generating a small size (8 bit) keystream block to XOR with 8 bits of plaintext, and Sosemanuk and Salsa are from the eSTREAM project (Profile I), which are considered secure and designed for software purposes. Although there are also two other stream ciphers in the Profile I, Rabbit and HC-128, we do not consider them for the same reasons explained in [17]: Rabbit is patented and HC-128 is too complicated to be implemented efficiently on an 8 bit CPU.

Like the block ciphers, we have implemented RC4 in assembly language on ATmega128 and the cycles of the other two ciphers are taken from [17], which uses the same platform to compare stream ciphers. Table IV shows the characteristics and implementation results of the three stream ciphers. If not considering the keystream setup period, RC4 uses the fewest number of cycles to generate the keystream bytes for encryption.

TABLE IV. IMPLEMENTATION RESULT OF DIFFERENT STREAM CIPHERS.

Stream cipher	Keystream block size	CPU cycles		Cycles per byte (encrypt)
		Setup	Encrypt	
RC4	8 bits	18787	31	31
Sosemanuk	640 bits	8739	8559	107
Salsa	512 bits	60	17812	279

The relation between the number of CPU cycles and the size of the plaintext is illustrated in Fig. 3, which is obtained on the assumption that the keystream setup period only happens once at the beginning of communication and then the keystream is generated continuously. In this figure, it can be seen that when the size of plaintext is smaller than 81 bytes, RC4 has the worst performance because the number of setup cycles is greatly larger than others. However, after that point, RC4 achieves the best energy performance since dramatically fewer cycles are needed to generate the keystream bytes for encryption.

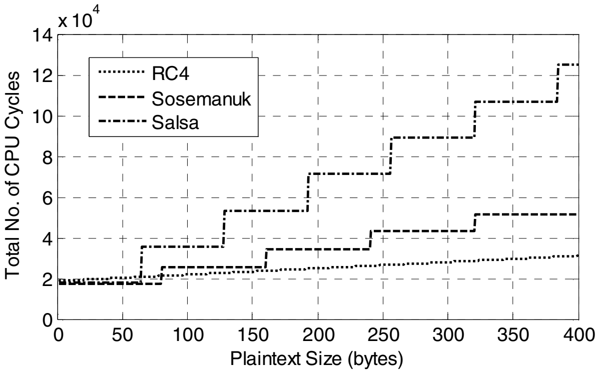

Fig. 3 Computational costs of stream ciphers.

IV. SELECTION BETWEEN STREAM CIPHER AND BLOCK CIPHER

In previous sections, we evaluated the symmetric key ciphers' energy performance without considering the application of the cipher to a protocol used in a noisy wireless channel resulting in a non-zero probability of bit errors. In this section, we further analyze the energy performance of the stream ciphers and block ciphers when they are applied in WSNs with poor channel quality.

A. Channel Quality Consideration

In a noisy communication channel, bit errors may result in packets being lost or corrupted and the resulting decryption may lose synchronization with encryption. This can be resolved by periodic resynchronization involving the transfer of an initialization vector (IV). The IV can be sent in each packet as in TinySec [13] or sent in a separate IV packet occasionally as implied by SPINS [6], in order to re-establish cryptographic synchronization. The resynchronization computational energy cost is different for different ciphers, which directly impacts the lifetime of the sensor device.

1) Block Cipher Synchronization

When using block ciphers, the computational cost can be calculated directly by the number of data blocks times the energy cost per block. That is, fixed computational energy is consumed for the given data block. The computational energy cost of IV synchronization for a block cipher only includes a few CPU cycles, such as loading and storing the new IV. The computational cost of key setup for block ciphers can be more substantial. However, we assume that key setup is very infrequent and base our analysis on the assumption that the round keys are already generated and stored.

2) Stream Cipher Synchronization

For stream ciphers, two stages are included in the operation of the cipher: (1) keystream setup based on an updated IV and (2) the keystream generation and encryption of plaintext. For most stream ciphers, the computational energy cost of IV synchronization (causing keystream setup) may take a considerable proportion of the total computational energy cost. This is particularly notable for RC4.

B. Comparison of Steam Cipher and Block Cipher

In this section, we compare the energy performance of the stream ciphers and block ciphers using the analysis model from our previous research [18]. Both the physical parameters and the packet formats for the data packet and IV packet are the same as shown in [18]. In the analysis, we apply the counter mode of operation to the block cipher, ensuring that it functions similarly to a stream cipher. This results in a fair comparison between stream ciphers and block ciphers and it is a suitable mode for WSN applications [6]. A separate IV packet is assumed to be sent periodically for the IV synchronization, where K is used to denote the synchronization period, that is, the number of data packets sent between IV packets. The data packet is accepted and considered valid only when the MAC recalculation is correct. In our analysis, we assume that for each block cipher case, the block cipher is used in CBC mode to generate the MAC. For stream cipher cases, the MAC is generated by AES used in CBC mode.

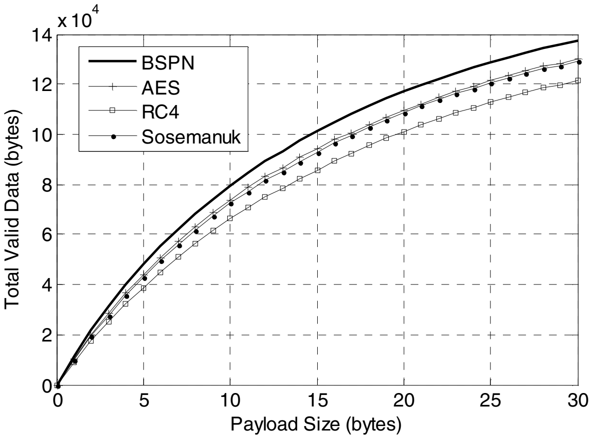

Fig. 5 Energy performance of different ciphers (BER = 10^{-4}).

We analyze the amount of total valid data transferred from a sensor to an aggregator node or base station in a WSN given a fixed energy supply. The resulting total valid data for different ciphers is shown in Fig. 5, where K = 5 and the bit error rate (BER) is 10^{-4}. Bit errors are randomly and independently generated. Fig. 5 represents a meaningful relative comparison of ciphers. We can see that the BSPN cipher achieves the best energy performance. Although RC4 takes the fewest number of cycles per byte, it shows the worst energy performance under a noisy channel due to its large number of keystream setup cycles.

The value of *K* impacts the amount of valid transferred information together with the size of payload. One packet transmitted with an error will lead to the packet being discarded and, hence, the following packets cannot be decrypted correctly until the next IV packet comes. This encourages the value of *K* to be small. On the other hand, small *K* consumes significant communication energy cost to transfer the IV packet, which will also decrease the energy performance. An optimal value of *K* exists to balance these two factors. The effect of period *K* and payload size on the amount of valid data is shown as Fig. 6, where the cipher BSPN is applied and BER is 10^{-4}. It can be seen that as the payload size increases, the amount of total valid data can obtain a maximum value with an optimal selection of period *K*.

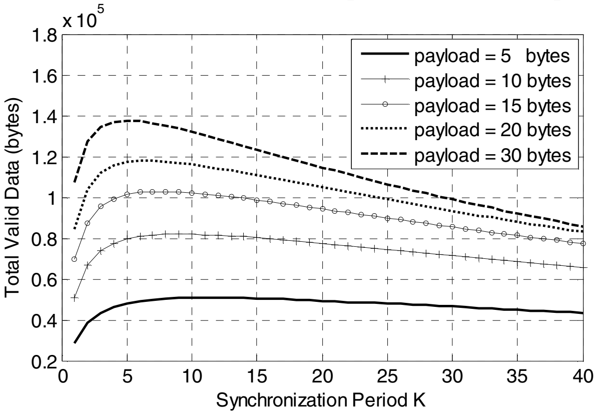

Fig. 6 Effects of period and payload size on throughput (BSPN, BER = 10^{-4}).

The optimal value of *K* varies a little for different ciphers given other parameters fixed. Fig. 7 shows the effect of different ciphers, which is obtained under the condition of payload size of 25 bytes and BER equal to 10^{-4}. We can see that the optimal value changes according to different ciphers being used, but in all cases it is in the range of 5 – 8.

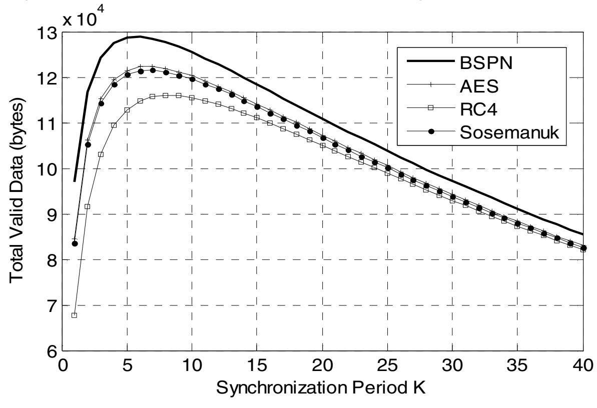

Fig. 7 Optimal *K* for different ciphers (payload size = 25 bytes, BER=10^{-4}).

V. CONCLUSION

In this paper, we have investigated the energy performance of symmetric key cryptographic algorithms when security is applied to the link layer of wireless sensor networks. We evaluate the energy efficiency by comparing the number of CPU cycles per byte for different symmetric key ciphers, including both stream ciphers and block ciphers. We further analyze the ciphers according to their characteristics and the effect of the channel quality when applied in WSNs. Finally, we conclude from the analysis results that the lightweight block cipher, BSPN, achieves good performance, providing energy efficiency as well as suitable security for sensor nodes in a WSN.

REFERENCES

[1] Y. Wang, G. Attebury, B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol.8, no.2, pp. 2-23, 2006

[2] W. K. Koo, H. Lee, Y. H. Kim and D. H. Lee, "Implementation and Analysis of New Lightweight Cryptographic Algorithm Suitable for Wireless Sensor Networks," in *Proc of 2008 Information Security and Assurance (ISA 2008)*, pp.73-76, Korea, April 2008.

[3] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy Analysis of Public-key Cryptography for Wireless Sensor Networks," *in Proc of 2005 Pervasive Computing and Communications (PerCom 2005)*, pp. 324-328, Germany, March 2005.

[4] M. Henricksen, "Tiny Dragon - An Encryption Algorithm for Wireless Sensor Networks," *in Proc of 10th High Performance Computing and Communications, (HPCC '08)*, p.p. 795-800, 25-27 Sept. 2008.

[5] R. Tahir, M. Y. Javed, M. Tahir and F. Imam, "LRSA: Lightweight Rabbit Based Security Architecture for Wireless Sensor Networks," *in Proc of 2008 Intelligent Information Technology Application (IITA '08),* vol.3, pp. 679-683, China, Dec. 2008.

[6] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," *ACM Wireless Networks*, vol. 8, no. 5, pp. 521-534, Sept. 2002.

[7] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," *Federal Information Processing Standard (FIPS) 197*, Nov. 2001.

[8] A.J. Menezes, P. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.

[9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," *in Proc. of ACM ASPLOS IX*, pp. 93-104, Nov. 2000.

[10] "SkipJack and KEA Algorithm Specifications," National Institute of Standards and Technology, Mai.1998.

[11] H. Cheng, H.M. Heys, and C. Wang, "PUFFIN: A Novel Compact Block Cipher Targeted to Embedded Digital Systems", *in Proc of Euromicro Conference on Digital System Design (DSD 2008),* Parma, Italy, Sep. 2008.

[12] A. Youssef, S.E. Tavares, and H.M. Heys, "A New Class of Substitution-Permutation Networks", *in Proc of Workshop on Selected Areas in Cryptography(SAC '96)*, Queen's University, Kingston, Ontario, Aug. 1996.

[13] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a Link Layer Security Architecture for Wireless Sensor Networks," in *Proc of the 2nd international Conference on Embedded Networked Sensor Systems* (SenSys '04), pp. 162-175, New York, November 2004.

[14] L. Granboulan, "Flaws in Differential Cryptanalysis of Skipjack," *Lecture Notes in Computer Science 2355: Fast Software Encryption*, Springer-Verlag, pp. 81-98, 2002.

[15] L. R. Knudsen, M. J. B. Robshaw and D. Wagner, "Truncated Differentials and Skipjack," *Lecture Notes in Computer Science 1666: Advances in Cryptology – CRYPTO'99*, Springer-Verlag, pp. 790, 1999.

[16] J. Daemen, V. Rijmen, *The Design of Rijndael: AES - the Advanced Encryption Standard*, Springer-Verlag New York, 2002.

[17] G. Meiser, T. Eisenbarth, K. Lemke-Rust, and C. Paar, "Efficient Implementation of eSTREAM Ciphers on 8-bit AVR Microcontrollers," *in Proc of 2008 Industrial Embedded Systems (SIES 2008)*, pp. 58-66, June 2008.

[18] X. Zhang, H.M. Heys, and C. Li, "An Analysis of Link Layer Encryption Schemes in Wireless Sensor Networks," *in Proc of IEEE International Conference on Communications (ICC 2010), Cape Town, May 2010.*