An Analysis of Link Layer Encryption Schemes 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 focus on secure communication in wireless sensor networks (WSNs). Specifically, we investigate different factors which affect the energy cost of link layer cryptographic security schemes, such as the payload size, the source of the initialization vector, and the channel quality. We propose an approach to evaluate the performance of cryptographic communication schemes by developing an analysis model considering these factors. The appropriateness of this model is supported by simulation results. In conclusion, we recommend cipher feedback (CFB) mode for the cipher operation, thereby achieving energy efficiency without compromising the security.**

Keywords - wireless sensor networks; security; encryption; block cipher

I. CRYPTOGRAPHIC METHODS APPLIED TO WSNS

The confidentiality of data is critical in many wireless sensor networks (WSNs). In WSNs, the energy limitation of sensor nodes is a challenging constraint for all communication protocols including cryptographic schemes used to achieve confidentiality. Although some work has focused on the study of cryptographic algorithms in WSNs [1][2][3], little effort has been made to investigate the method or mode of operation applied to a cipher to achieve confidentiality in a WSN. TinySec [4] and SPINS [5] are examples of proposals which do explicitly recommend a mode of operation for block ciphers applied to WSNs.

In this paper, we explore the effect of packet size, the generation of the initialization vector and the channel quality on the energy consumption of cryptographic communication schemes in WSNs. As a result, we propose the encryption of a packet based on the use of a block cipher in CFB mode.

A. Encryption Methods

Security requirements in WSNs include four major parts: data confidentiality, data integrity, data authentication, and data freshness [5]. In cryptographic communication schemes, encrypted data (or ciphertext) takes the place of the original payload (or plaintext) to achieve the data confidentiality. A symmetric key cipher is considered the appropriate type of cipher to encrypt the data in a WSN, because it saves considerable energy cost over other types of ciphers [6]. A message authentication code (MAC) functions as the cryptographic checksum, providing both data integrity and data authentication. We take a packet as correctly transmitted only when the recalculated MAC value equals the transmitted MAC value, and assume the difference is caused either by a noisy channel or by a malicious attacker.

A mode of operation is a scheme to provide flexible implementation of a symmetric key block cipher when operating on a large bulk of data [7]. The operation mode determines how to use the block cipher to derive the ciphertext and has an impact on the communication energy cost in WSNs. Cipher block chaining (CBC) mode is a common selection for encrypting large amounts of data, and is proposed to be used in the TinySec scheme on a per-packet basis [4]. In [4], to reduce the size of ciphertext to the same number of bits as plaintext, the ciphertext stealing technique [8] is used. However the ciphertext size cannot be decreased below the block size when the amount of plaintext is less than the block size of the cipher. Counter mode, cipher feedback (CFB) mode, and output feedback (OFB) mode, make the encryption like a stream cipher, which generates the same number of ciphertext bits as plaintext bits. Counter mode is proposed to be used in the SPINS scheme [5], taking advantage of the efficiency and security of a stream cipher approach. Selection of an appropriate mode of operation for the block cipher is critical; otherwise potential problems might occur when applying the scheme in a realistic WSN, such as operation mode related security weaknesses, error propagation, and loss of data synchronization.

B. IV Generation

The initialization vector (IV) plays an important role in cryptography mechanisms, although the content of the IV itself is not kept confidential. Generally, the IV is XORed with the first data block before encryption, so that the plaintext data can be randomized thereby effectively eliminating the repetition of data input to the cipher - an important security consideration [7]. In counter mode, the IV is used to initialize the counter value and, since this must be done periodically to ensure synchronization between the ends of the communication, it is also important in this case, that IV is not repeated. In CFB mode, the IV can be used to reset the feedback at the block cipher input.

For security purposes, although the size of IV should ideally be similar to the block size of the cipher, to save the communication energy costs, some schemes reduce the size of IV. For instance, TinySec reduces the effective IV size to 16

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.

bits, which allows for the possibility of a repeated IV and therefore may be considered a potential weakness from a cryptographic point of view.

In WSNs, IV greatly affects the energy performance of a scheme because of two factors: (1) when transmitted between nodes it consumes communication energy and (2) the ciphertext will not be decrypted correctly if the IV is not reliably known by both communication parties.

II. Communication Model and Comparison Metric

A. Communication Model

In this paper, we assume a sensor network which includes three types of nodes: common sensor nodes, aggregators and a base station. The common sensor nodes sense data (such as temperature, smoke, humidity, etc.) and send it to an aggregator. The aggregator functions as the cluster head and is assumed to have a larger energy supply. After aggregating, data is sent to the base station, which has a continuous energy supply.

B. Cryptography Implemented in WSNs

Since a common sensor node is the most critically energy limited device in a WSN, we will focus on exploring the cryptographic scheme for the basic communication behavior of a sensor node: encrypting the sensor data and transmitting the ciphertext out. (This analysis can also be extended to the aggregator. However, it is assumed that the aggregator is not as energy constrained as the sensor node.) Confidentiality needs to be ensured for the link between a sensor node and an aggregator to ensure that potentially sensitive information is not obtained by an inappropriate party. Our research will focus on the data encryption at the link layer for data transmitted from the sensor node to the aggregator, and assumes the key has already been securely established. For a discussion on key distribution methods, see [4].

C. Scheme Comparison Metric

In a WSN, energy and security are two key considerations. Although security is the design goal, it is not practical to evaluate a cryptographic scheme by taking the security level as a metric. Although security schemes can be identified to have weaknesses, such flaws are not always evident or easily quantifiable. Hence, we shall assume that schemes using accepted cryptographic methods with reasonable block and key sizes are secure and the metric we shall use to evaluate the cryptographic communication scheme in a WSN is based on the energy cost of the scheme. We choose the amount of valid information transmitted during the sensor life as a metric, which is obtained under a given energy of the battery. To simplify the comparison, we consider only the energy costs of the security scheme in a common sensor node and ignore the energy requirement of other types of processing.

III. Cryptography Schemes

When a link layer encryption scheme is applied in a WSN, the IV must be known to both the communication sides. Hence, the origination and distribution of IV content are critical. In this section, we explore three methods that can be used to encrypt in WSNs and the associated IV agreement processes. Table I shows the notation, derived from TinySec [4], which will be used in the packet formats described for each scheme.

TABLE I. Notation used in the packet format

Symbol	Size (bit)	Description
START SYMBOL	N_{ss}	Start symbol used for medium access.
DEST	N_{hd} (sum)	Destination address of the receiver.
AM		Active message handler type.
LEN		Size of the packet.
IV	N_{iv}	Initialization vector.
PAYLOAD	N_{pld}	Payload, usually variable.
MAC	N_{mac}	Message authentication code.
SRC	32 bits (sum)	Source address of the sender
CTR		16-bit counter

1) CBC with IV in Each Packet (TinySec)

In applying CBC mode to encrypt, one approach would be to include IV in each packet transmitted as is done in TinySec, so that the receiver can recover the IV directly from the received packet to use in the decryption. The negative side of this scheme is that the IV requires extra bits to be transmitted, which increases the energy cost of each packet. The TinySec [4] packet format is shown in Fig. 1. The IV is taken from the fields from DEST to CTR. Since the effective size of the IV is much less than the block size, it is conceivable that, in some contexts, IV will be repeated over a long duration of time. Nevertheless, it is argued that the semantic security of this scheme is sufficient [4].

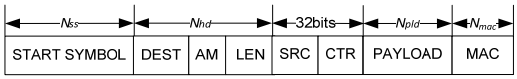


Fig. 1. Packet format of TinySec scheme

2) Counter Mode with Periodic IV Packet Synchronization

Using counter mode for encryption requires a periodic transfer of IV to ensure that the encryption and decryption process remain synchronized. Without the periodic transfer of IV, a lost data packet will result in a permanent loss of cipher synchronization. For this purpose, the IV can be communicated periodically within a special IV packet, separate from data packets. A newly received IV initializes the counter and subsequently the count increases by one after each block encryption. The SPINS scheme [5] proposes a similar approach, providing semantic security without transmission overhead.

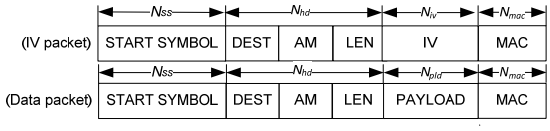


Fig. 2. Packet format of counter mode

In our study, it is assumed that a corrupted IV packet is simply discarded, resulting in corruption of the subsequent data packets until a new IV is successfully exchanged. (Note that an ARQ error control process could be applied to the IV packet, but our investigation shows that the acknowledgement process contributes little to the energy performance.) The

packet format is shown as Fig. 2, which includes both the IV packet and the data packet. This scheme reduces the energy cost of a packet over TinySec, but results in a high probability that the packet is not properly decrypted when the channel quality is poor. This occurs since (1) if the IV packet has been received with errors, subsequent packets cannot be decrypted correctly until the next IV packet is transmitted and (2) even if the IV packet is received correctly, one data packet that has an error and is discarded will affect the decryption of the following packets until the next IV packet is transmitted.

3) CFB Scheme

A block cipher can be configured for CFB mode, where the data in a packet is encrypted by XORing the plaintext block with the output of the block cipher which has used the previous ciphertext block as input. We shall consider an approach that resets the feedback at the start of each packet by using the preceding ciphertext from the payloads of previous packets as an IV block to be fed to the block cipher input. Unlike other schemes, CFB scheme does not consume extra energy for either including IV bits in each packet or transmitting separate packets of IV frequently. The packet format is shown as Fig. 3. In this scheme, the current packet being decrypted depends on both the packet itself and the previous packets. Note the initial IV used in this scheme can be exchanged during the key establishment phase.

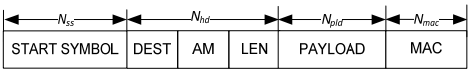


Fig. 3. Packet format of CFB scheme

4) Other Schemes

Other schemes combining cipher modes (eg. CBC, counter, and CFB) with different methods of IV agreement are possible. For example, it is possible to have a scheme which uses counter mode with an IV sent in every packet to re-initialize the counter value for decryption. Another example would be the application of CBC with IV periodically reset through the use of IV packets. Still another example is CFB mode which does not synchronize to a block at the start of each packet. In this paper, we focus on the CBC with an IV in each packet and counter mode with a periodic IV packet as these are similar to previous proposed schemes [4][5].

IV. Analysis of Encryption Schemes

In this section, we propose an approach to evaluate the energy performance of the link layer encryption schemes by developing an analysis model based on the assumption of fixed size packets. We focus on the energy cost of the sensor node.

A. General Analysis for Fixed Size Packets

1) Probability of the packet transmitted without errors

TABLE II. Packet Size for Different Schemes

Scheme	Type	Packet size
CBC (Tinysec)	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{iv} + N_{pld} + N_{mac}$
Counter with periodic IV	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{pld} + N_{mac}$
	IV	$N_{ivpkt} = N_{ss} + N_{hd} + N_{iv} + N_{mac}$
CFB	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{pld} + N_{mac}$

The probability that a packet has one or more bit errors is decided by two factors: the probability of error for each bit (p_e) and the size of the packet (N_{pkt}). The probability that a received packet has no errors (P_o) is expressed as

$$P_o = (1 - p_e)^{N_{pkt}}, \quad (1)$$

where it is assumed that bit errors occur randomly and independently. The determination of N_{pkt} is listed in Table II.

2) Energy Calculation

The energy cost of a sensor node mainly consists of two parts: the communication cost and the computation cost. For communication energy cost, we consider the transmitting energy cost; while for computation cost, we consider the encryption cost and MAC calculation cost. For the purposes of our analysis, some types of the energy are ignored, such as the energy cost when sensor is in the sleep mode, the computation costs of data processing, the key distribution costs, etc..

- Communication energy cost:

The energy cost of transmitting one packet (E_{xmt}) depends on the current in transmitting mode (I_{xmt}), the voltage (U), N_{pkt}, and the bit rate of transmission (R), and is given by

$$E_{xmt} = (I_{xmt} \times U \times N_{pkt})/R. \quad (2)$$

- Computation energy cost:

We consider two parts of computation energy cost introduced by the cryptographic scheme: encryption and MAC generation. We calculate the energy by determining the number of CPU cycles used to finish the cryptographic processing. Two factors impact the energy cost when the same sensor device is used: the cipher algorithm efficiency and the payload size. Since we are considering the use of a block cipher, the block size is an important parameter in the energy cost calculation, as this determines how many encryption operations are carried out. From the perspective of the sensor node, whose main function is to transmit collected sensor information, the encryption and MAC processing energy costs (E_{enc} and E_{mac}, respectively) are given by:

$$E_{enc} = (P_{cpu} \times C_{enc}/f_{cpu}) \times \lceil N_{pld}/b \rceil, \quad (3)$$

$$E_{mac} = (P_{cpu} \times C_{enc})/f_{cpu} \times \lceil (N_{pkt} - N_{ss} - N_{mac})/b \rceil, \quad (4)$$

where P_{cpu} and f_{cpu}represents the CPU's power and frequency respectively, b represents the block size, and C_{enc}represents the number of clock cycles required to encrypt one block. The symbol $\lceil . \rceil$ denotes the ceiling operator. Note that we are assuming that the MAC is applied across all fields of the packets except the start symbol and is produced using the block cipher in CBC mode [7]. Also the total energy cost of one packet depends on the type of the packet. For example, in the counter scheme with periodic IV, the IV packet energy cost does not include the encryption energy since the IV is not encrypted. This is summarized in Table III.

3) Expected Number of Valid Data Bits

As discussed before, we choose the total number of data bits successfully transmitted (N_{total}) by the sensor node for a given energy level as the metric to evaluate the performance of different schemes. Its expected value can be expressed as

$$N_{total} = n \times N_{pld} \times P_{valid}, \quad (5)$$

which depends on three factors: the number of the transmitted data packets, n, the payload size, N_{pld}, and the probability that the packet is correctly decrypted, P_{valid}. Given parameters of

the packet and the sensor node energy, the first two parameters can be calculated directly. In the following section, we will focus on the factor P_{valid}, which varies for different schemes.

B. Probability of Valid Packets for Different Schemes

The probability that a packet is successfully received and decrypted (P_{valid}) strongly depends on the specific scheme applied. In determining an expression for P_{valid}, we assume that the bit errors occur independently with a probability given by the bit error rate p_e. We consider now the probability for different schemes.

1) CBC with IV in Each Packet (TinySec)

In this scheme, since an IV is included in each packet, the packet can be decrypted correctly when every bit of the packet is received correctly. Also, since the TinySec scheme uses CBC mode of operation and the ciphertext stealing technique, P_{valid} is given by P_o in (1) with

$$N_{pld} = \begin{cases} b & , \quad \text{if } N_{ptext} < b \\ N_{ptext}, & \text{if } N_{ptext} \geq b \end{cases}, \tag{6}$$

where N_{ptext} represents the number of plaintext bits.

2) Counter Mode with Periodic IV Packet

In this scheme, the probability that a data packet of size N_{pkt} is correctly received is

$$P_{data} = (1 - p_e)^{N_{pkt}}, \tag{7}$$

and for an IV packet of size N_{ivpkt} is

$$P_{iv} = (1 - p_e)^{N_{ivpkt}}. \tag{8}$$

The probability that the data packet can be decrypted correctly is based on the probability that the previous IV packet and all previous data packets following the IV packet are successfully received and is given by

$$P_{valid} = \frac{P_{iv} \times (P_{data} - {P_{data}}^{K+1})}{K \times (1 - P_{data})}, \tag{9}$$

where K represents the number of data packets sent for each IV packet. We can see that P_{valid} is determined by both K and the channel quality. If the channel is very noisy, there will be high energy cost due to the fact that bit errors result in a lot of data being discarded when MAC verification fails.

3) CFB Scheme

For our CFB scheme, the probability of a data packet being decrypted correctly is given by

$$P_{valid} = (P_o)^{traceback+1}, \tag{10}$$

where the parameter $traceback$ is determined by

$$traceback = \begin{cases} \lceil N_{pld}/b \rceil & \text{if } N_{pld} < b \\ 1 & \text{if } N_{pld} \geq b \end{cases}. \tag{11}$$

The parameter $traceback$ represents the number of previous packets whose ciphertext is used for the IV and therefore affects the decryption of the current packet. The $traceback$ value will be large for a small payload size, which will decrease the probability of the current packet being decrypted correctly. However, the small payload size will make the packet itself more likely to be transferred correctly, which counteracts the effect of the large $traceback$ value.

C. Analysis Results

1) Parameters

The values of the parameters we have used in the analysis are listed in Table IV. We use the block cipher Skipjack [9] as the cipher applied in all schemes since TinySec specifies this cipher. However, similar results can be shown for the Advanced Encryption Standard (AES) [10]. We have programmed Skipjack in assembly language on the ATmega128 CPU to determine the number of cycles required for encryption. We choose an arbitrary value for the battery energy E_{total} as this is sufficient for comparing schemes. Physical parameters for the sensor device are derived from the specifications of the commercial product MICA2 [11]. Also, we use several BER values to represent the different channel qualities.

TABLE III. ENERGY COST OF DIFFERENT PACKET TYPES

Packet Type	Energy per packet
Data packet	$E_{data} = E_{enc} + E_{xmt} + E_{mac}$
IV packet	$E_{iv} = E_{xmt} + E_{mac}$

TABLE IV. PARAMETERS USED IN THE ANALYSIS

Object	Parameter	Value	Unit
Block cipher (Skipjack)	C_{enc}	1482	cycles
	b	64	bits
Sensor board	E_{total}	5	J
	P_{cpu}	13.8	mW
	f_{cpu}	8	MHz
	I_{xmt}	27	mA
	U	3.3	V
	R	38400	bps
Packet	N_{ss}	8	bytes
	N_{hd}	4	bytes
	N_{iv}	b	bits
	N_{pld}	from 1 to 30	bytes
	N_{mac}	4	bytes
	K	10	packets
Channel	BER	5×10^{-4},10^{-4},10^{-5},0	-

2) Analysis Results for Different Schemes

The analysis results of the three schemes are shown in Fig. 4. These figures illustrate the expected total amount of valid data given a fixed energy according to different payload sizes. In the analysis, we also present the results under different channel bit error rates. As expected, as BER increases, the total amount of valid data decreases due to the necessity of discarding many corrupted packets. All cryptographic schemes have the same trend for the relationship between BER and the amount of valid data transferred.

a) CBC with IV in Each Packet

Fig. 4(a) shows the results for the specific format of TinySec, which uses CBC and includes the IV in each packet. As seen in the figure, the slope decreases as the payload size increases. This occurs because the ratio of IV size to the packet size decreases. This trend is the same in other schemes as well. The curve from payload size 1 to 8 bytes forms a straight line instead of an arc, as a result of CBC mode requiring at least one block to encrypt. Hence, since Skipjack uses a 64 bit block size, for payload sizes of 1 to 8 bytes, the plaintext is padded out to 64 bits and a full 64 bits of ciphertext size will be produced. When the payload size is larger than the block size, the transmitted ciphertext size is the same as the plaintext size, since the ciphertext stealing can be used.

b) Counter Mode with Periodic IV Scheme

For the counter mode scheme with periodic IV synchronization, Fig. 4(b) shows, at the high bit error rate, the number of valid data bits decreases dramatically compared to the lower bit error rate. This indicates that the periodic IV approach has a relatively poor performance in a poor quality channel. When the channel quality is not good, both the data packet and IV packet have a large probability of being discarded due to error, which will impact the decryption of the following data packets which relies on synchronization of the counter value between transmitter and receiver.

c) CFB Scheme

For the cipher feedback scheme, Fig. 4(c) shows steps in the curve are apparent for small payloads, particularly when the bit error rate increases. The occurrence of steps is influenced by the *traceback* value, which relates to both the payload size of the packet and the block size of the cipher. Since, in the analysis, we use the cipher of block size 64 bits, the steps stop at the payload size equal to 8 bytes. For the payload size larger than 8 bytes, the curve presents a continuous arc. We can see that the *traceback* does not impact the performance of the scheme very much.

D. Simulation Results for Fixed and Variable Size Packets

To verify the suitability of previous the analysis models, we have performed simulations for both fixed size packets and variable size packets. Simulation results of packets with variable size are shown in Fig. 5. Each scheme is simulated using a binomial distribution of different variances. As can be seen, the resulting curves are very similar to the curves determined by the analysis model. Similar simulation outcomes were also observed for fixed size packets and variable sized packets following the truncated geometric, Poisson, and uniform distributions. We conclude that the analysis model approximates well the energy cost behavior for sensor nodes for a large variety of packet size distributions.

V. Comparison of Encryption Schemes

In this section, we compare the performance of the three different schemes. We use the analytical result to evaluate each scheme, because we have found through simulations that the analysis is representative of results for many packet distributions.

A. Error-free Channels

Fig. 6 compares the cryptographic schemes utilizing the cipher Skipjack under the condition of an error-free channel. We can see that counter mode scheme with periodic IV achieves better results than the CBC scheme with an IV included in each packet. This is because periodic IV schemes can do the IV agreement without losing counter synchronization when the channel is error-free. In contrast, a scheme such as TinySec transmits the IV with every packet, which costs more energy than the periodic IV scheme. The CFB scheme also does not transmit IV with the packet, and achieves a slightly better result than the counter mode scheme. Compared to TinySec, the CFB scheme achieves an improvement between 9% - 65%, depending on the packet size with improvement being most notable for smaller payloads. Compared to the counter mode scheme, the CFB scheme achieves an improvement between 5% and 15%.

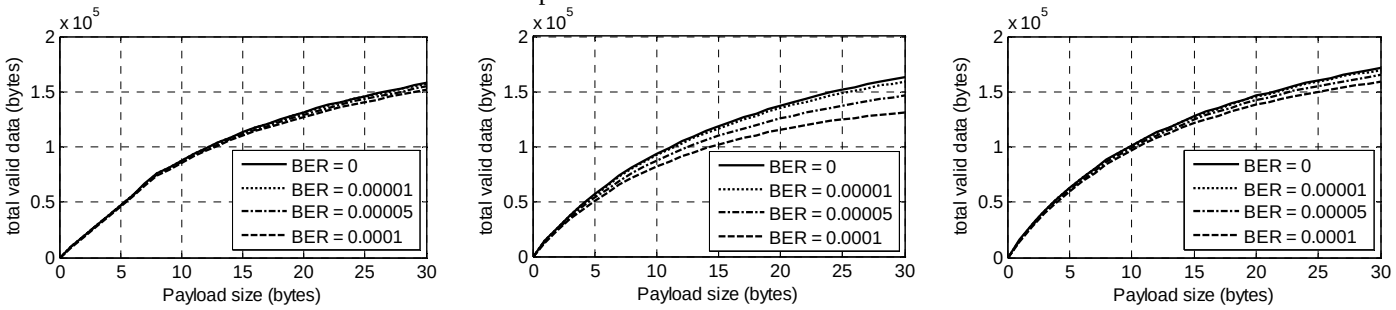
B. Noisy Channels

The results change significantly in a channel with noise, as shown in Fig. 7 which is based on BER of 10^{-4}. The number of valid data bits of the counter mode scheme with periodic IV dramatically decreases, because the successful decryption of the data packet depends on both the IV packet and previous data packets to be correctly received. In contrast, the CBC scheme (TinySec) which includes the IV in every packet achieved much better results. This can be explained by noting that, in the noisy channel, including an IV in each packet results in a larger probability to decrypt the packet correctly, because it only depends on the packet received to have no errors. The CFB scheme achieves better results than both of these two schemes, as it reduces the extra communication energy cost of an IV in each packet with the small expense of introducing a modest decrease in the probability of a valid decrypted packet.

The comparison result illustrates that the CFB scheme can still achieve better performance even when the channel is noisy. Compared to TinySec, the CFB scheme achieves an improvement between 5% and 50% depending on packet sizes. Similarly, compared to the counter mode scheme, CFB achieves improvement between 10% and 21%. In both cases, improvements are most significant for smaller size payloads.

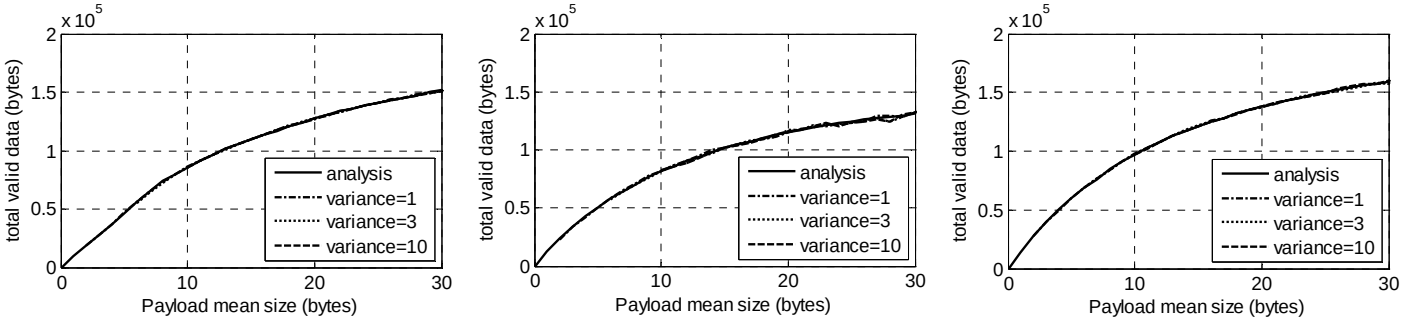
C. Encryption Algorithm

Although the communication energy cost mainly affects the life of the sensor, computation energy also contributes to the total energy cost. Also, security is another factor that needs to be considered. Although the previous discussions use Skipjack as the block cipher, for most applications, AES is considered to be secure and efficient. We have implemented AES in assembly language on ATmega128 and the result is shown in Table V.



(a) CBC scheme (TinySec). (b) Counter mode IV scheme. (c) CFB scheme.

Fig. 4. Throughput analysis under different BER conditions.



(a) CBC scheme (TinySec). (b) Counter mode scheme. (c) CFB scheme.

Fig. 5. Throughput analysis for different schemes with binomial distribution.

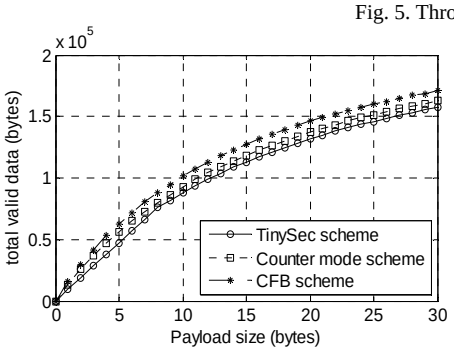


Fig. 6. Comparison of schemes with BER = 0.

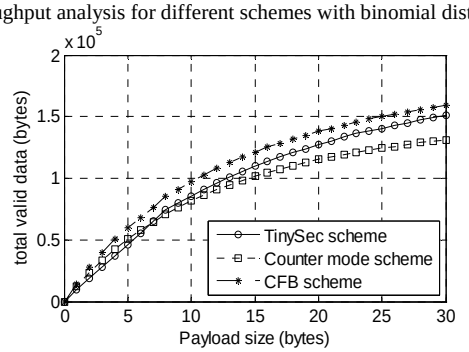


Fig. 7. Comparison of schemes with BER = 10^{-4}.

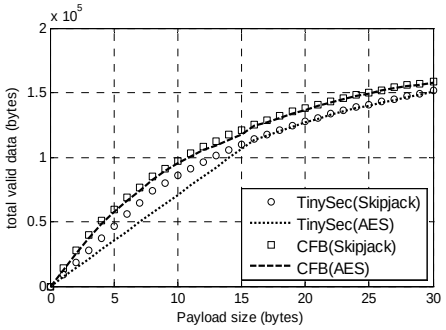


Fig. 8. Cipher comparison with BER = 10^{-4}.

Skipjack, a block cipher of 64 bit block size, is utilized in the TinySec scheme [4] as it is considered the most energy efficient cipher compared to other ciphers with similar parameters. However, several weaknesses in Skipjack have been identified [12][13]. AES, with a 128 bit block size, is a widely accepted secure cipher. Although the energy cost of an AES encryption computation is about two times more than Skipjack, one AES encryption produces about twice the number of ciphertext bits for large payloads. In Fig. 8, the analysis results of TinySec and the CFB scheme using both Skipjack and AES are presented. From the figure, we can see that there is little difference between the two ciphers except the Skipjack scheme when the payload size is small, which is because of the ciphertext stealing technique being used. Hence, the cipher AES appears to be good choice when considering both the security level and energy consumption.

TABLE V. CIPHERS USED IN SCHEMES

Cipher	Block size	Cycles
Skipjack	64 bits	1482
AES	128 bits	3266

VI. CONCLUSION

In this paper, we investigate the performance of link layer encryption schemes in wireless sensor networks, which are directly affected by the packet size, the IV agreement, and the bit error rate of the channel. We propose using the amount of valid data transferred from the sensor node given a fixed energy supply as the metric to evaluate the performance when cryptographic schemes are applied to the sensor node communication. We develop an analysis model and a performance comparison is made between the cipher feedback scheme and other schemes. The results suggest that the cipher feedback scheme achieves better performance for a range of channel qualities and for small payload sizes provides a large relative improvement in the number of the bits that can be successfully transferred for a given energy.

REFERENCES

[1] 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.

[2] Y. W. Law, J. Doumen, and P. Hartel, "Survey and Benchmark of Block Ciphers for Wireless Sensor Networks," ACM *Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 65-93, Feb. 2006.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] National Institute of Standards and Technology (NIST), "Proposal To Extend CBC Mode By Ciphertext Stealing," *Special Publication (SP) 800-38A*, May. 2007.

[9] "SkipJack and KEA Algorithm Specifications," *National Institute of Standards and Technology*, May 1998.

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

[11] 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.

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

[13] 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.