Energy Efficiency of Encryption Schemes Applied to Wireless Sensor Networks

Xueying Zhang, Howard M. Heys, and Cheng Li

Electrical and Computer Engineering

Faculty of Engineering and Applied Science

Memorial University

St. John's, Newfoundland, A1B 3X5, Canada

Emails: {xueying.zhang, hheys, licheng}@mun.ca

[1] This work is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Wireless Communications and Mobile Computing Research Centre (WCMCRC) of Memorial University.

[2] Based on "An Analysis of Link Layer Encryption Schemes in Wireless Sensor Networks" by X.Zhang, H.M.Heys, and C.Li, which appeared in Proceedings of *IEEE International Conference on Communications (ICC 2010)*, Cape Town, South Africa, May 2010, and "Energy Efficiency of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks" by X.Zhang, H.M.Heys, and C.Li, which appeared in Proceedings of *Biennial Symposium on Communications (QBSC 2010)*, Kingston, Canada, May 2010. ©2010 IEEE.

ABSTRACT

In this paper, we focus on the energy efficiency of secure communication in wireless sensor networks (WSNs). Our research considers link layer security of WSNs, investigating both the ciphers and the cryptographic implementation schemes, including aspects such as the cipher mode of operation and the establishment of initialization vectors. We evaluate the computational energy efficiency of different symmetric key ciphers considering both the algorithm characteristics and the effect of channel quality on cipher synchronization. Results show that the computational energy cost of block ciphers is less than that of stream ciphers when data is encrypted and transmitted through a noisy channel. We further investigate different factors affecting the communication energy cost of link layer cryptographic schemes, such as the size of payload, the mode of operation applied to a cipher, the distribution of the initialization vector, and the quality of the communication channel. A comprehensive performance comparison of different cryptographic schemes is undertaken by developing an energy analysis model of secure data transmission at the link layer. This model is constructed considering various factors affecting both the computational cost and communication cost and its appropriateness is verified by simulation results. In conclusion, we recommend using a block cipher instead of a stream cipher to encrypt data for wireless sensor network applications and using a cipher feedback scheme for the cipher operation, thereby achieving energy efficiency without compromising the security in WSNs.

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

1 INTRODUCTION

Secure data transmission is important in wireless sensor networks (WSNs) which can be used in a variety of scenarios ranging from medical applications to environment monitoring. The energy efficiency of security methods applied to WSNs has become important due to the open media and broadcast nature of WSN communication and the limited energy supply of the sensor device [1]. Security requirements in WSNs include four major aspects: data confidentiality, data integrity, data authentication, and data freshness [2]. To cater to these security requirements, cryptography is typically applied to the data and additional energy cost is introduced as a result. The energy cost includes both the computational cost and the communication cost and is affected by both the cryptographic algorithm (i.e., cipher) and the cryptographic implementation scheme[1] being used.

The energy limitation of a sensor device is a challenging constraint in WSNs. Researchers have focused on studying cryptographic algorithms which have direct impact on the computational energy cost in a WSN, by evaluating existing cryptographic algorithms [3][4] or proposing new energy efficient ciphers [5][6]. However, these studies are mainly based on the algorithm itself, independent of the cryptographic scheme. Other proposals focus on the cryptographic scheme such as the TinySec link layer security architecture [7], SPINS [2] and other encryption schemes based on, or developed from, TinySec [8][9][10][11][12]. Although these proposed schemes do consider energy efficiency in their recommendations for the application of cryptography, no effort has been made to effectively study the energy efficiency of a sensor node in a noisy environment, considering both the cryptographic algorithm and the cryptographic scheme as a whole.

In this paper, we explore the energy efficiency of secure data transmission at the link layer in WSNs by considering both the computational cost and communication cost. We investigate both the intrinsic factors and extrinsic factors which affect the computational cost of the cryptographic algorithms, including both stream ciphers and block ciphers. As well, we explore effects on the energy consumption of cryptographic schemes, from factors such as the packet size, the process used in the distribution of the initialization vector, and the channel quality. We present the integrated performance evaluation for different ciphers and schemes by developing an analysis model for a sensor node, which considers various factors. The appropriateness of the model is verified and supported by extensive simulation results. Based on our modeling and analysis, we propose an effective security scheme which utilizes a block cipher with a ciphertext feedback scheme to encrypt the data, thereby achieving enhanced energy efficiency as well as secure data transmission in WSNs.

[1] We use the term "cryptographic implementation scheme" or simply "cryptographic scheme" in this paper to refer to the method of applying a cipher to the plaintext data. For example, while a cryptographic system may use the block cipher Advanced Encryption Standard (AES [17]) as its cryptographic algorithm, the mode of operation applied (e.g. cipher block chaining (CBC) [30]) and the process associated with establishing the initialization vector (IV) are important components of the cryptographic scheme.

2 ENERGY COST IN A SECURE WSN

2.1 Cryptography in WSNs

In many WSNs, the confidentiality of data is often critical since the information transmitted by a sensor node may contain private information, such as the health condition of a patient. For this purpose, we focus on exploring symmetric key cryptographic algorithms and schemes for the basic communication behavior of a sensor node. Our network model assumes that a sensor node encrypts the sensing data and transmits the ciphertext out. The receiver can be a cluster head, an aggregator or the base station, which will decrypt the message for further analysis. Two fundamental categories of ciphers, symmetric key ciphers and asymmetric (or public) key ciphers use different mechanisms to achieve security: asymmetric cryptography depends on the difficulty of a mathematical problem and symmetric key cryptography focuses on the structure of simple iterative cryptographic operations [13]. Because significantly less energy is required, symmetric key ciphers are preferred for use in the encryption of data transmitted by a sensor node.

Symmetric key cryptographic algorithms include two types of ciphers, as shown in Figure 1: stream ciphers and block ciphers [13]. A stream cipher typically operates on one bit of data by XORing the generated keystream bit with a plaintext bit, whereas a block cipher operates on a block of data (typically 64 bits or 128 bits) by iterating through rounds of simple cryptographic operations, such as nonlinear substitution and linear transformation. A mode of operation is often applied to a block cipher to make the cipher's operation suitable for a given application. Such modes include cipher block chaining (CBC) mode, counter mode, and cipher feedback (CFB) mode [13].

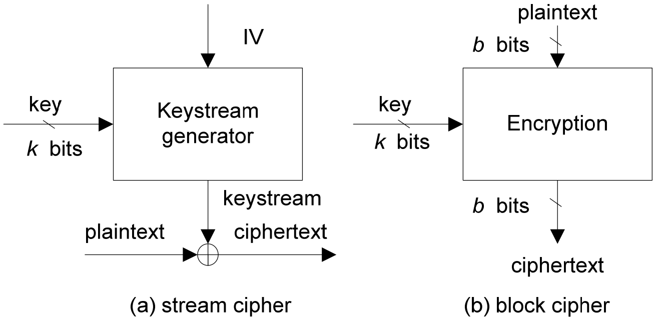


Fig. 1. Symmetric key cryptographic algorithms.

A symmetric key block cipher uses the same cipher key for both the encryption and decryption, which is assumed to be already known by the communicating parties. This cipher key can be embedded inside the sensor device before its deployment or established by a specialized key agreement at the beginning of communication [2]. Note that, in most applications, key establishment will be very infrequent and, as a result, the relatively high cost of a key agreement protocol can be expected to have little impact on the battery life of the sensor node in comparison to the ongoing encryption and transmission of sensor data. Hence, in our research, we assume the cipher key is established in a sensor node prior to our analysis and we do not consider the effects of key establishment.

Encryption is used to achieve the confidentiality between two communicating parties, preventing potentially sensitive information from being obtained by an inappropriate party. At the same time, a message authentication code (MAC) [13] can be transmitted in a packet, so that the receiver side can use it to judge the validity of a packet. In a security scheme, a MAC can provide both data integrity and data authentication. Using a MAC instead of a checksum or CRC is necessary in a WSN since both the channel quality and a malicious attack can influence the correctness of a received message. In our work, we assume that, at the receiver, to eliminate the potential possibility of malicious attack, a packet will be dropped if the MAC value calculated from the received data does not match the MAC accompanying the received data.

2.2 Energy Cost Calculation

The energy cost of a sensor node related to cryptographic operations consists of two parts: the communication cost and the computational cost. For communication energy cost, we consider the transmitting and receiving energy cost; while for computational cost, we consider the encryption cost and MAC calculation cost. In our research, the energy cost calculation is based on the structure of the Mica series sensor devices [14][15], which are representative products that have been widely studied and used in recent years. However, it is worth noting that the proposed analysis approach can be easily applied to other sensor products.

The operation of a sensor node mainly includes sensing data from the environment being monitored (such as temperature, heart rate, etc.), processing the collected data using a microprocessor (for example, the 8-bit Atmel Atmega128 CPU [16]), and transmitting messages out using small size packets (typically not larger than 30 bytes). Figure 2 shows a typical packet format of a sensor node derived from TinySec [7] and its notation is explained in

Table I. The initialization vector (IV) used by the encryption scheme can be transmitted with the payload (as shown in the figure) or be transmitted independently, determined by the specific cryptographic scheme adopted (which will be explicitly explained in the following sections).

TABLE I. NOTATION USED IN THE PACKET FORMAT

Symbol	Size (bits)	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 information.
PAYLOAD	N_{pld}	Payload, usually variable.
MAC	N_{mac}	Message authentication code.

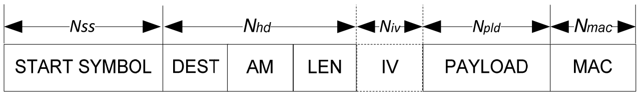


Fig. 2. Typical packet format for a sensor node

The cryptographic energy cost is calculated as the sum of the communication and computational energy costs as detailed in the following sections.

2.2.1 Communication Energy Cost

The energy cost of a sensor node for transmitting one packet (E_{tx}) depends on the current drawn from the battery by the transceiver circuitry in transmitting mode (I_{tx}), the voltage (U), the size of the packet (N_{pkt}), and the bit rate of transmission (R), and is given by

$$E_{tx} = \frac{I_{tx} \times U \times N_{pkt}}{R}. \tag{1}$$

Similarly, the energy cost of receiving one packet (E_{rx}) can be expressed as

$$E_{rx} = \frac{I_{rx} \times U \times N_{pkt}}{R}, \tag{2}$$

where I_{rx} is the current in receiving mode.

2.2.2 Computational Energy Cost

We consider two parts of computational energy cost introduced by cryptographic operation: encryption and MAC generation. Since a sensor node is manufactured to be low power and low cost, most devices use an 8-bit microprocessor for data processing, such as one from the Atmel AVR ATmega series [14]. We calculate the energy by determining the number of CPU cycles to finish the cryptographic processing. Two factors impact the computational energy cost: the cipher algorithm efficiency and the payload size. When considering the use of a symmetric key cipher, the block size is an important parameter in the energy cost calculation, as this determines how many encryption operations are to be carried out. From the perspective of a sensor node, whose main function is to transmit collected sensor information, the encryption and MAC processing energy costs per packet (E_{enc} and E_{mac} , respectively) are given by:

$$E_{enc} = \left(P_{cpu} \times \frac{C_{enc}}{f_{cpu}}\right) \times \left\lceil \frac{N_{pld}}{b} \right\rceil \tag{3}$$

and

$$E_{mac} = \left(P_{cpu} \times \frac{C_{mac}}{f_{cpu}}\right) \times \left\lceil \frac{N_{pkt} - N_{ss} - N_{mac}}{b_{mac}} \right\rceil, \tag{4}$$

where P_{cpu} represents the CPU's power, f_{cpu} represents the CPU's clock frequency, C_{enc} and C_{mac} represent the number of clock cycles required to encrypt one block for the encryption scheme and the MAC, respectively, and the packet parameters (N_{pld}, N_{pkt}, N_{ss}, and N_{mac}) are defined in Table I. For a block cipher, b is the block size, while for a stream cipher, b represents the keystream block size which is the amount of keystream produced at one time to be used in the bit-wise XOR of b bits in parallel. The symbol $\lceil\cdot\rceil$ denotes the ceiling operator. Note that we assume that the MAC is produced using a block cipher of block size b_{mac} in CBC mode [13] and is applied across all fields of the packets except the start symbol.

2.3 Performance Metrics

In a WSN, energy and security are two key considerations. Although security is a design goal, it is not practical to evaluate a cryptographic algorithm or scheme by taking the level of the security as a metric. Although security schemes may be identified to have weaknesses, such flaws are not always evident or easily quantifiable. Hence, we shall assume that schemes using accepted cryptographic algorithms and methods with reasonable block and key sizes are secure, and the metric we shall use to evaluate the cryptographic performance in a WSN is based on the energy consumption. We choose as a metric the amount of valid information successfully transmitted from the sensor node per Joule of energy consumed in the transmission process and refer to this metric as “energy efficiency”. Because the main focus of this paper is on evaluating cryptographic schemes, we consider only the energy costs related to security operation in a sensor node and ignore the energy requirement of other types of processing.

The valid information successfully transmitted refers to the data in packets which are received and decrypted correctly. That is to say, if a packet is transmitted with a corruption of its start symbol or packet header, the packet is considered to be not valid since the receiver will not receive it. As well, if other fields of packet are corrupted, the MAC verification will fail and the data will be discarded and also considered not valid. Many factors will influence the energy efficiency of a sensor node. In the following sections, we will analyze it from both the perspective of cryptographic algorithm and the applied cryptographic scheme.

3 CRYPTOGRAPHIC ALGORITHMS IN WSNS

Data encryption is an important aspect of applying security to a WSN. Although the transmission of data is typically 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 sensor node. In WSNs, energy limitations make the security schemes focus on ciphers with efficient computational energy consumption. Hence, the symmetric key cipher is typically utilized to encrypt data for transmission. In this section, we analyze the computational cost of symmetric key ciphers in a WSN, including analyses of both block ciphers and stream ciphers.

3.1 Factors Affecting the Computational Cost

Before evaluating the energy consumption of different cryptographic algorithms in a sensor node, we first discuss different factors which directly affect the computational energy cost. These factors can be divided into two types: (1) intrinsic nature of the algorithm structure; (2) extrinsic factor of channel quality, which influences the frequency of cryptographic resynchronization. Both the intrinsic and extrinsic factors interactively affect the computational energy cost of cryptographic algorithms in a WSN.

3.1.1 Intrinsic Factors: Characteristics of the Algorithms

(1) Structure of the Cryptographic Algorithm

Cryptographic algorithms may vary significantly in structure. Many block ciphers, such as Advanced Encryption Standard (AES) [17], make use of a number of rounds of operations such as substitutions (S-boxes) and linear transformations. To satisfy some special needs, such as low circuit area and limited resources, some lightweight ciphers have been designed [18]. For many applications in WSNs, a sensor node is an energy limited device transmitting low entropy information with a limited life span. Hence, lightweight ciphers with energy efficiency are good candidates for such applications. Lightweight ciphers are often designed to provide just sufficient security, which may not offer as much as provided by AES [17]. For example, a lightweight block cipher may use a smaller block size and key size than AES.

(2) 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 keystream to produce the ciphertext bits. For practicality, blocks of plaintext bits can often be XORed with keystream bits in parallel. In contrast, block ciphers process an entire 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 straightforwardly operate on variable lengths. However, block ciphers may need to pad plaintext out to have a length that is a multiple of the block size and in WSNs the resulting extra ciphertext bits may lead to an increased transmission energy cost to the sensor node.

(3) Key Setup

Usually there is a key setup period 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 varies for different ciphers, it can be complicated and take relatively long time to finish. However, in WSNs, key setup is typically very infrequent since it will only follow the establishment of a new cipher key. Hence, the energy consumption introduced by key setup is very small over the lifetime of the sensor node.

(4) Initialization Vector (IV) Setup

Setup of the initialization vector (IV) takes place in symmetric key ciphers whenever a new IV is established. For example, stream ciphers must periodically 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, most modes of operation [13] 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 the setup for the keystream. The energy cost of IV setup for stream ciphers is typically much higher than that of block ciphers, and as we shall see, in a WSN, the energy cost of IV setup is especially critical in the determination of whether to use a stream cipher or a block cipher.

3.1.2 Extrinsic Factors: Quality of Communication Channel

In a noisy communication channel, bit errors may result in packet loss or corruption and the resulting decryption may lose synchronization with the encryption process. This can be resolved by periodic resynchronization via the transfer of a new IV. However, the energy cost of resynchronization will directly impact the lifetime of the sensor device and is influenced by the channel condition since poor channel quality may lead to the need of frequent resynchronization. The computational cost of resynchronization will be different for block ciphers and stream ciphers due to the variation in the IV setup phases. In this section, we focus on the computational energy cost of cryptographic resynchronization for different ciphers, with its communication cost being explicitly investigated in Section 4.

(1) Block Cipher Resynchronization

When using block ciphers, the computational cost of encryption can be calculated directly by the number of data blocks times the energy cost per block. That is, fixed computational energy is consumed per data block encrypted. The computational energy cost of IV setup for a block cipher typically includes only a few CPU cycles to complete operations 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 used by a block cipher are already generated and stored.

(2) Stream Cipher Resynchronization

For stream ciphers, two phases are included in the operation of the cipher: (1) keystream setup based on an updated IV (that is, IV setup) and (2) keystream generation and encryption of plaintext. For most stream ciphers, the computational energy cost of resynchronization (causing IV setup) may take a considerable proportion of the total computational energy cost. This is particularly noticeable for RC4 [19], which will be presented in our analysis later.

3.2 Symmetric Key Cipher Evaluation

Our evaluation of the cryptographic algorithms includes two parts: (1) consideration of intrinsic factors only, that is, evaluating the symmetric key cipher's energy performance directly; (2) consideration of extrinsic factors, which means applying the cipher to a protocol used in a noisy wireless channel.

3.2.1 Considering Intrinsic Factors

As a low cost device, a sensor node usually employs an 8-bit microcontroller to implement different kinds of operations. Hence, in our evaluation, we use the number of CPU cycles per byte to evaluate the computational energy efficiency of symmetric key ciphers. The number of cycles is obtained from an implementation of the cipher in assembly language on the ATmega128 CPU [16], a popular 8-bit microcontroller, used in wireless sensor devices such as Mica2 [15].

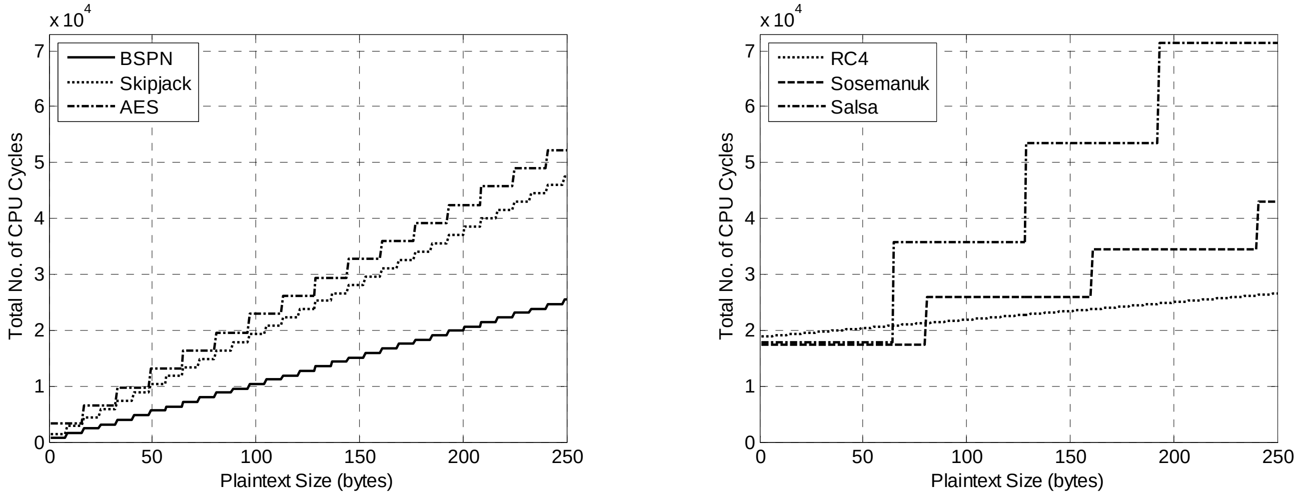
(1) Block Cipher Comparison

We consider the energy performance of three block ciphers when applied in WSNs. The selected block ciphers are AES [17], Skipjack [20], and Byte-oriented Substitution Permutation Network (BSPN) (first proposed in [21] and applied to WSNs in [22]). We have selected these three ciphers since they represent different characteristics of symmetric key cryptographic algorithms: AES shows the most popularity and trustworthiness for encryption, Skipjack has been proposed for application to WSNs (such as in TinySec) and BSPN is a novel cipher designed explicitly for efficient implementation by 8-bit microcontrollers. Characteristics and implementation results of these ciphers are shown in Table II. From the results, we can see that the performance of Skipjack is slightly better than AES, however, Skipjack has been noted to have security weakness under certain cryptanalyses [24][25]. Among the three block ciphers, BSPN requires the fewest number of CPU cycles per byte and thus has the lowest computational energy cost. As a lightweight block cipher, BSPN is conjectured to provide suitable security to sensor nodes, capable of resisting important attacks such as linear cryptanalysis and differential cryptanalysis [21].

TABLE II. CHARACTERISTICS OF BLOCK CIPHERS

Block cipher	Block size	Key size	# Rounds	Cycles per block	Cycles per byte
AES	128 bits	128 bits	10	3266	204
Skipjack	64 bits	80 bits	32	1482	186
BSPN	64 bits	≥ 64 bits	8	796	99

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. BSPN is the most efficient among the three ciphers because of its efficiency of linear transformation for an 8-bit CPU [22].



(a) Computational costs of block ciphers.

(b) Computational costs of stream ciphers.

Fig. 3. Comparison of cipher computational costs.

The relation between the number of CPU cycles and the size of the plaintext is illustrated in Figure 3(a). From this figure, we can see that the computational energy cost of BSPN is the smallest among the three block ciphers. Note that the steps in the figure are caused by the block encryption nature of the cipher. The implementation of AES is an efficient 8-bit implementation based on using the “xtime” operation for MixColumns [26]. Although AES requires more cycles per bytes than BSPN, it should be noted, that AES is designed for a 128-bit key and block size, while BPSN is designed for a 64-bit block size, although its key size can be easily extended to 128 bits [22].

(2) Stream Cipher Comparison

We have selected three stream ciphers for comparison: RC4 [19], Sosemanuk [27] and Salsa [28]. 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)[2] and are considered secure and designed for software purposes.

TABLE III. CHARACTERISTICS OF 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

We have implemented RC4 in assembly language on the ATmega128 and the results for the other two ciphers are taken from [29], which uses the same platform to compare stream ciphers. Table III shows the characteristics and implementation results of the three stream ciphers. From the results, we can see that, if not considering the IV setup period, RC4 uses the fewest number of cycles to generate the keystream bytes for encryption. However, the energy efficiency of the stream ciphers is significantly influenced by the resynchronization frequency as we shall see in Section 3.2.2. The relation between the number of CPU cycles and the size of the plaintext is illustrated in Figure 3(b), which is obtained on the assumption that the IV setup phase only happens once at the beginning of communication and then the keystream is generated continuously. As the number of bytes increases, RC4 achieves the best energy performance since dramatically fewer cycles are needed to generate the keystream bytes for encryption.

[2] Although there are also two other stream ciphers selected by the eSTREAM project for Profile I, Rabbit and HC-128, we do not consider them for the same reasons explained in [29]: Rabbit is patented and HC-128 is too complicated to be implemented efficiently on an 8-bit CPU.

3.2.2 Considering Extrinsic Factors

We now consider the energy performance of block ciphers and stream ciphers in a noisy communication environment. We apply the counter mode of operation to the block cipher to ensure 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 [2]. By using block ciphers and stream ciphers, we focus on the energy efficiency by analyzing the amount of valid data transmitted per Joule. The analysis is based on the Mica sensor device [15], and uses the *Periodic IV without ACK* scheme (which will be discussed in Section 4.2.3) for the IV distribution. This scheme functions by transmitting the IV within an independent packet periodically to resynchronize the cipher. In the analysis, bit errors are randomly and independently generated. Figure 4 shows the performance comparison result of different ciphers, which is obtained under the bit error rate (BER) of 10^{-4} with an IV packet sent for every $K = 5$ data packets. Using the MAC, data packets that are detected to be corrupted are discarded and thereby reduce the energy efficiency. Other details of the analysis can be found in the following sections, such as the parameters listed in Table VI and packet format illustrated in Figure 8. For generating the MAC used in each data packet, we assume that for each block cipher case, the block cipher is used in CBC mode to generate the MAC [13]. For the stream cipher cases, the MAC is generated by AES used in CBC mode. The analysis results show that the BSPN cipher achieves the best energy efficiency and, in general, a sensor node can transmit more valid data per Joule using block ciphers. This occurs because BSPN takes a modest number of clock cycles to encrypt each byte and virtually no computational energy consumption is associated with the overhead of updating the IV. Although RC4 takes the fewest number of cycles per byte to encrypt, it shows the worst energy performance under a noisy channel due to its large number of IV setup cycles.

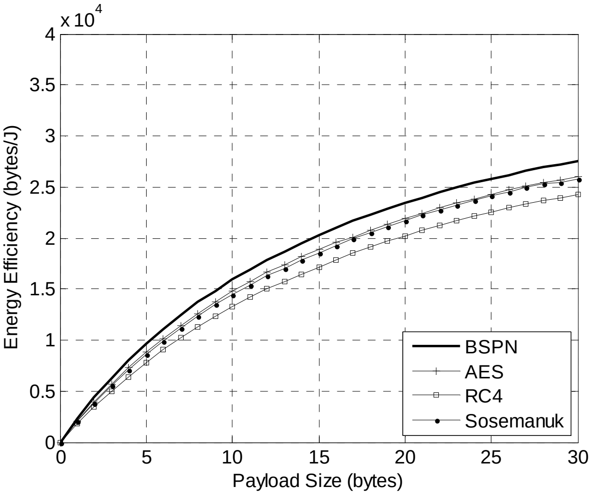


Fig. 4. Energy efficiency of different ciphers ($K = 5$, BER = 10^{-4}).

4 CRYPTOGRAPHIC SCHEMES IN WSNS

After analyzing the energy efficiency of cryptographic algorithms, we now investigate cryptographic schemes in a WSN, which directly impact the communication energy cost for a wireless sensor node. A cryptographic scheme specifies how to use a cryptographic algorithm to achieve secure data transmission. This includes features such as the mode of operation and the IV distribution. The cryptographic schemes to be considered will all be based on block ciphers.

In secure communication schemes, encrypted data (or ciphertext) takes the place of the original payload (or plaintext) to achieve the data confidentiality. A message authentication code (MAC) functions as the cryptographic checksum, providing both data integrity and data authentication. At the receiver, we accept a packet as correctly transmitted only when the MAC value calculated from the data received in the packet equals the received MAC value and assume that any difference is caused by either a noisy channel or a malicious attacker.

4.1 Factors Affecting the Communication Cost

Before evaluating the energy cost of different cryptographic schemes, we first discuss several factors which will affect the communication energy cost when cryptography is applied to a WSN.

(1) Mode of Operation

A mode of operation is a method to make use of a symmetric key block cipher when operating on a large bulk of data [13]. Cipher block chaining (CBC) mode [30] functions like a chain, each block of ciphertext is obtained by encrypting the result of the previous ciphertext block XORed with the current plaintext block, except that the first block is generated by XORing plaintext with the IV. Counter mode [30] and cipher feedback (CFB) mode [30] function like stream ciphers, which generate keystream by encrypting the value of a counter and a feedback register, respectively. Both the counter and feedback register are initialized by IV, and the difference is that the counter value is incremented by 1 for each block, while the feedback register stores the previous block of ciphertext. The three modes of operation are illustrated in Figure 5 and each mode needs an IV for performing the encryption.

The mode of operation determines how to use the block cipher to derive the ciphertext and has an impact on the communication energy cost in WSNs. 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 [7]. To reduce the size of ciphertext to the same number of bits as plaintext, the ciphertext stealing technique [31] 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 and CFB mode make the encryption process similar to a stream cipher, which generates the same number of ciphertext bits as plaintext bits and counter mode is proposed to be used in WSNs by the SPINS scheme [2]. Selection of an appropriate mode of operation for the block cipher is critical: issues such as mode-related security weaknesses, error propagation, and loss of data synchronization must be considered in the context of practical WSN applications.

There are also many advanced modes of operation proposed in recent years, which integrate the MAC generation with the encryption mode of a block cipher such as counter with CBC-MAC (CCM) mode [32]. Some modes of operation can also provide encryption, integrity and authentication in one pass of a block cipher operation such as offset codebook (OCB) mode [33] and Galois/counter (GCM) mode [34]. Some researchers propose to use advanced modes in WSNs to accelerate the speed of operation and save the computational energy cost of MAC generation. However, it is worth noting that, in a WSN, the overall energy cost will not be significantly affected: as our analysis results will show later, communication energy cost is the major factor affecting the encryption schemes applied to a sensor node and the added cost of computing the MAC is comparatively insignificant.

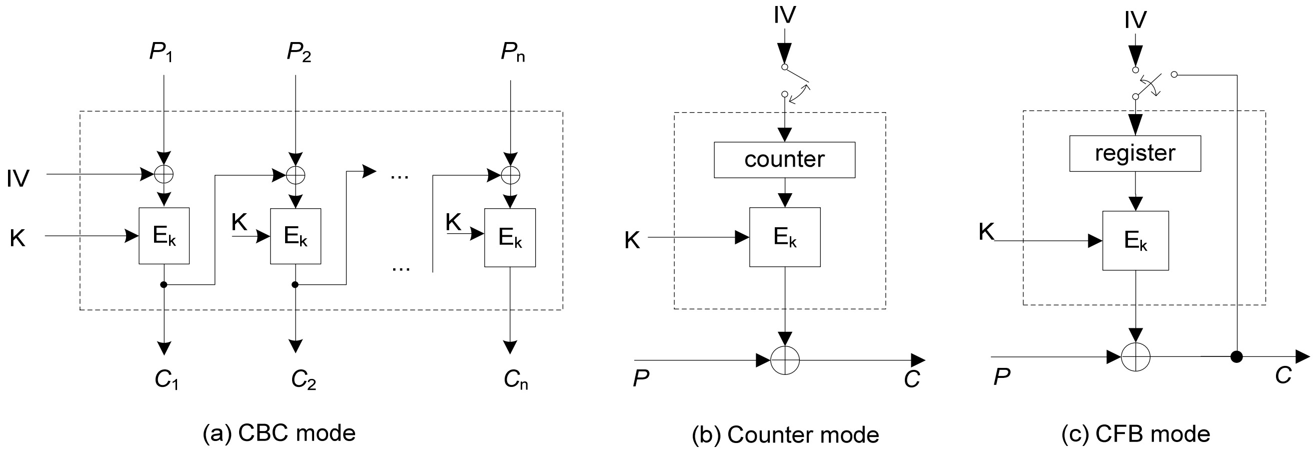


Fig. 5. Modes of operation of block ciphers.

(2) Initialization Vector

The initialization vector (IV) plays an important role in cryptographic mechanisms. Although the content of the IV itself is not required to be confidential, the use of a particular IV value should not be repeated. In CBC mode, the IV is used to produce the first block of ciphertext, so that the data can be randomized, thereby effectively eliminating the repetition of data input to the cipher - an important security consideration [13]. In counter mode, the IV is used to initialize the counter value and, since this must be done periodically to ensure synchronization between both 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. Although the size of IV should ideally be similar to the block size of the cipher for security purposes, to save the communication energy costs, some schemes reduce the size of IV. For instance, TinySec [7] reduces the effective IV size to 16 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 communicating parties.

(3) Payload Size and Channel Quality

To prevent malicious attacks, a packet will be discarded when the MAC verification fails. However, poor channel quality may also lead to an unsuccessful MAC verification. Also, a packet with longer payload size will more likely lead to the packet being corrupted in a noisy channel. Hence, both the payload size of a packet and the quality of the communication channel will influence the amount of data that needs to be retransmitted and consequently will affect the energy efficiency.

4.2 Description of Cryptographic Schemes

When a link layer cryptographic scheme is applied in a WSN, the IV is generated by the transmitter and must be known to both communication sides. Hence, the generation and distribution of IV content are critical. In this section, we explore several typical schemes that can be used for encryption in WSNs and the associated IV distribution processes. Typically, IV distribution methods in WSNs have been of two types: (1) including IV in each packet being transmitted [7] and (2) transmitting IV in an independent packet periodically [2]. Here we also present another IV distribution method, which is based on previous ciphertext collection and does not require the designated IV information to be separately transmitted. We have selected several cryptographic schemes encompassing different modes of operation and IV distribution approaches, and analyze them in terms of their energy efficiency when applied to WSNs. First we describe the schemes.

4.2.1 CBC Mode + Implicit IV Scheme

In this scheme, the IV is included in each packet and CBC mode is applied to a block cipher for encryption. We use *implicit IV* as the name to represent a general approach which transmits the IV in each packet so that the receiver can recover the IV directly from the received packet to use in the decryption. In this scheme, the IV size is the same as the block size of the cipher. The disadvantage of this scheme is that extra bits are to be transmitted, which increases the communication energy cost of each packet. The packet format is shown in Figure 6. In the *implicit IV* scheme, the packet will be decrypted correctly when it is transmitted and received without error (i.e., the MAC verification passes).

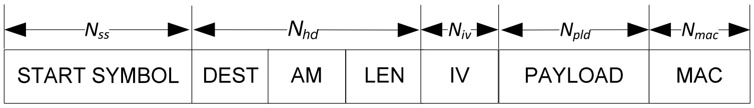


Fig. 6. Packet format of implicit IV scheme.

4.2.2 TinySec Scheme

TinySec [7] uses information in each packet to determine the IV and applies CBC mode to a block cipher for encryption. Although similar to the *implicit IV* approach, it minimizes the number of transmitted bits by using some information for IV that is already available in the packet. The packet format is shown in Figure 7, where SRC is the source address of the sender and CTR is a 16-bit counter. This scheme reduces the energy cost by reducing the number of bits being transmitted. 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 [7]. The energy cost is also minimized by applying CBC mode with the ciphertext stealing technique [31] to encrypt the data.

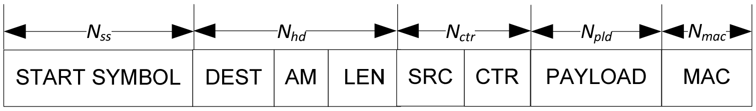


Fig. 7. Packet format of TinySec scheme.

4.2.3 Counter Mode + Periodic IV without ACK Scheme

This scheme uses counter mode of a block cipher for encryption, which requires a periodic transfer of IV to ensure that the encryption and decryption processes stay synchronized. The SPINS scheme [2] proposes a similar approach, providing semantic security without transmission overhead. However, without the periodic transfer of IV, a lost data packet will result in a permanent loss of cipher synchronization. For this purpose, a sensor node will periodically transmit a special IV packet (separate from data packets) to indicate a new IV for use in encryption. The transferred IV functions as initializing the counter of both communicating sides and subsequently the count increases by one for each block of data encrypted and decrypted.

In our study, it is assumed that a corrupted IV packet is simply discarded, resulting in improper decryption of the subsequent data packets until a new IV is successfully exchanged. The packet format is shown as Figure 8, which includes both the IV packet and the data packet. This scheme reduces the energy cost of a packet over the schemes that include IV in each packet, 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.

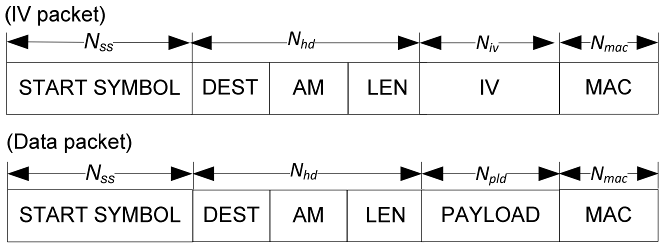


Fig. 8. Packet format of periodic IV without ACK scheme.

4.2.4 Counter Mode + Periodic IV with ACK Scheme

This scheme is similar to the previous scheme except that a stop-and-wait ARQ protocol is used to guarantee the IV packet is being correctly received. That is, the sensor node periodically transmits the IV packet and judges whether the IV packet has been successfully transferred by receiving an acknowledgement from the receiver. In each IV distribution process, a sensor node will keep sending the IV packet until it receives the acknowledgement from the receiver. Counter mode is also used in this scheme. Packet formats are shown in Figure 9 for the IV packet, the acknowledgement (ACK) packet and the data packet. In the ACK packet, the “ACK” field represents the acknowledgement information. The energy cost will be higher than the *periodic IV without ACK* scheme, because two new causes of energy consumption are introduced by the ARQ scheme: first, the reception of the ACK packet will consume communication energy and, second, the IV packet may be retransmitted several times if the channel quality is particularly poor. It is worth noting that, although the ARQ protocol ensures that the IV packet is correctly received, corrupted data packets will still be discarded, resulting in a loss of synchronization in the decryption counter and, hence, the subsequent data packets will not be decrypted correctly until the next IV is distributed.

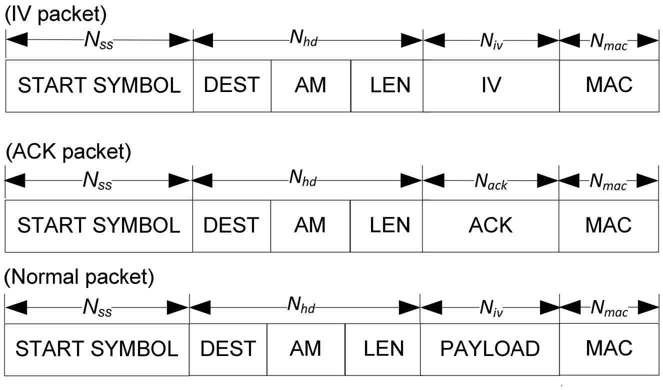


Fig. 9. Packet format of periodic IV with ACK scheme.

4.2.5 CFB Mode + Ciphertext Based IV Scheme

We propose this cryptographic scheme based on using previously transmitted ciphertext as the IV for the encryption of a data packet. A block cipher can be configured by 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 *b* bits of the preceding ciphertext (that is, encrypted payload) from the payloads of previous packets as an IV block to be fed to the block cipher input. If the previous data packet had at least *b* bits of payload, (i.e. $N_{pld} \geq b$), the last *b* bits of ciphertext from the packet is used as the IV; if the previous data packet had $N_{pld} < b$ then the N_{pld} ciphertext bits are used in the IV along with b-N_{pld} bits retrieved from the earlier packet(s). Unlike other schemes, this scheme does not consume extra energy for either including IV bits in each packet or transmitting separate IV packets. The packet format is shown in Figure 10. In this scheme, the current packet being decrypted depends on both the packet itself and the previous packets. Note that the initial IV used to produce the first bits of ciphertext can be exchanged during the key establishment phase.

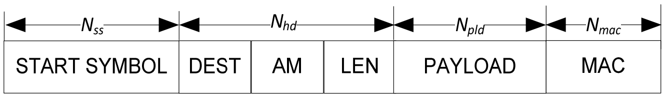


Fig. 10. Packet format of CFB scheme.

4.2.6 Other Schemes

Other schemes combining cipher modes (e.g. CBC, counter, and CFB) with different methods of IV distribution are possible but are not investigated in this paper. 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 or with an IV collected from previously transmitted ciphertext. 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. Likewise, advanced modes (providing authentication and encryption simultaneously) can also be used with different IV distribution schemes. In this paper, we have chosen representative cryptographic schemes to investigate.

5 Analysis of Cryptographic Schemes

In this section, we first focus on constructing an analysis model based on the assumption of fixed size packets to evaluate the energy performance of the link layer cryptographic schemes. Subsequently, the study will be extended through simulation experiments to variable size packets under different probability distributions. In our analysis, we take the binary symmetric channel as the channel model to study the energy performance of different encryption schemes in the presence of noise.

5.1 General Analysis for Fixed Size Packets

5.1.1 Probability of Error-free Packets

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. The probability that a received data packet has no errors (P_{data}) is expressed as

$$P_{data} = (1 - p_e)^{N_{pkt}} \tag{5}$$

under the assumption that bit error occurs randomly and independently. The determination of N_{pkt} is listed in Table IV for the different cryptographic schemes. Expressions similar to (5) exist for IV and ACK packets.

TABLE IV. Packet sizes for different schemes

Scheme	Type	Packet size
CBC + Implicit IV	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{iv} + N_{pld} + N_{mac}$
TinySec	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{ctr} + N_{pld} + N_{mac}$
Counter + Periodic IV without ACK	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{pld} + N_{mac}$
	IV	$N_{ivpkt} = N_{ss} + N_{hd} + N_{iv} + N_{mac}$
Counter + Periodic IV with ACK	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{pld} + N_{mac}$
	IV	$N_{ivpkt} = N_{ss} + N_{hd} + N_{iv} + N_{mac}$
	ACK	$N_{ackpkt} = N_{ss} + N_{hd} + N_{ack} + N_{mac}$

Scheme	Type	Packet size
CBC + Implicit IV	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{iv} + N_{pld} + N_{mac}$
CFB + Ciphertext based IV	Data	$N_{pkt} = N_{ss} + N_{hd} + N_{pld} + N_{mac}$

5.1.2 Energy Cost Calculation

The energy cost calculation has already been discussed in Section 2.2. The total energy cost of one packet depends on the type of the packet. For example, in the schemes with periodic IV, the IV packet and ACK packet energy cost does not include the encryption energy since the IV and ACK information is not encrypted. However, for all packet types, the MAC is calculated with the resulting energy cost. This is summarized in Table V. For the IV packet and ACK packet in the calculation of E_{tx} and E_{rx}, N_{pkt} is replaced by N_{ivpkt} and N_{ackpkt}, respectively, as shown in Table IV.

TABLE V. ENERGY COST OF DIFFERENT PACKET TYPES

Packet Type	Energy per packet
Data packet	$E_{data} = E_{enc} + E_{mac} + E_{tx}$
IV packet	$E_{iv} = E_{tx} + E_{mac}$
ACK packet	$E_{ack} = E_{rx} + E_{mac}$

5.1.3 Expected Number of Valid Data Bits

We choose the energy efficiency, η, as defined by the average number of data bits successfully transmitted per Joule by the sensor node as the metric to evaluate the performance of different schemes. This can be expressed as

$$\eta = n \times N_{pld} \times P_{valid} \tag{6}$$

which depends on three factors: the number of transmitted data packets, n, the payload size, N_{pld}, and the probability that the packet is correctly received and decrypted, P_{valid}. Given parameters of the packet and an energy of 1 Joule for each sensor node, 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.

5.2 Probability of Valid Packets for Different Schemes

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

5.2.1 CBC Mode + Implicit IV Scheme

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. Since CBC mode is used, P_{valid} is given by P_{data} in (5) with

$$N_{pld} = \left\lceil \frac{N_{ptext}}{b} \right\rceil \times b \quad , \tag{7}$$

where N_{pld} and N_{ptext} represent the number of payload bits and plaintext bits, respectively, and *b* is the cipher block size. This implies that the payload size is increased due to the application of CBC mode.

5.2.2 TinySec Scheme

Since the TinySec scheme uses CBC mode of operation with the ciphertext stealing technique, P_{valid} is given by P_{data} in (5) with

$$N_{pld} = \begin{cases} b \quad , & \text{if } N_{ptext} < b \\ N_{ptext}, & \text{if } N_{ptext} \geq b. \end{cases} \tag{8}$$

In this case, only when the amount of plaintext is less than one block is the payload size increased due to application of the encryption.

5.2.3 Counter Mode + Periodic IV without ACK Scheme

In this scheme, the probability that a data packet of size N_{pkt} is correctly received is given by P_{data} in (5) with $N_{pld} = N_{ptext}$ and, for an IV packet of size N_{ivpkt}, the probability that the packet is correctly received is

$$P_{iv} = (1 - p_e)^{N_{ivpkt}} . \tag{9}$$

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

$$P_{valid} = \frac{P_{iv} \times \left(P_{data} - P_{data}{}^{K+1}\right)}{K \times (1 - P_{data})} \quad , \tag{10}$$

where K represents the number of data packets sent for each IV packet and is assumed to be constant. 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.

5.2.4 Counter Mode + Periodic IV with ACK Scheme

Since the acknowledgement ensures the receiver side knows the initial value of the IV, the probability of the data packet being correctly decrypted (P_{valid}) depends on the probability of the data packet being transmitted correctly (P_{data}) and the period of the IV packet (K). Similar to the *periodic IV without ACK* scheme, if one data packet is lost or transmitted with error, the counter value will lose synchronization and the following packets will not be decrypted correctly. The factor K affects how many of the following packets will lose synchronization. Based on receiving the IV packet correctly, the probability that the data packet can be decrypted correctly for this scheme is expressed by

$$P_{valid} = \frac{P_{data} - P_{data}^{K+1}}{K \times (1 - P_{data})} \quad . \tag{11}$$

Note that two parameters, P_{iv} (probability of the IV packet transmitted correctly) and P_{ack} (probability of the ACK packet transmitted correctly), directly impact the number of successfully transmitted data packets (n in (6)), although they do not occur in the expression for P_{valid}. The less energy consumed by the IV distribution process, the more energy can be provided to the data transmission. That is to say, with large value of P_{iv} and P_{ack}, more energy can be used to transmit data packets, which directly results in large values of n and η in (6). On the contrary, if the channel is very noisy, the IV packet and ACK packet are very likely to be corrupted and there will be much more energy cost during the IV distribution.

5.2.5 CFB Mode + Ciphertext Based IV Scheme

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

$$P_{valid} = ([P_{data}])^{\gamma+1}, \tag{12}$$

where the parameter γ is determined by

$$\gamma = \begin{cases} \left\lceil {}^{b}/_{N_{pld}} \right\rceil & \text{if } N_{pld} < b \\ 1 & \text{if } N_{pld} \geq b \end{cases} \tag{13}$$

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

5.3 Analysis Results

In this section, we present the results of the analysis model applied to the identified cryptographic schemes.

TABLE VI. PARAMETERS USED IN THE ANALYSIS

Object	Parameter	Value	Unit
Block cipher (AES)	C_{enc}	3266	cycles
	b	128	bits
Sensor board	P_{cpu}	13.8	mW
	f_{cpu}	8	MHz
	I_{tx}	27	mA
	I_{rx}	10	mA
	U	3.3	V
	R	38400	bps
Packet	N_{ss}	8	bytes
	N_{hd}	4	bytes
	N_{iv}	16	bytes
	N_{ctr}	4	bytes
	N_{ack}	1	bytes
	N_{pld}	from 1 to 30	bytes
	N_{mac}	4	bytes
	K	10	packets
Channel	BER (p_e)	5×10^{-4},10^{-4},10^{-5},0	-

5.3.1 Parameters

The values of the parameters we have used in the analysis are listed in Table VI. We use the cipher AES, as it is the most applied block cipher today. However, similar outcomes would result for other symmetric key ciphers. Physical parameters for the sensor device are derived from the specifications of the commercial product Mica2 [14]. Also, we use several bit error rate values to represent the different channel qualities.

5.3.2 Analysis Results for Different Schemes

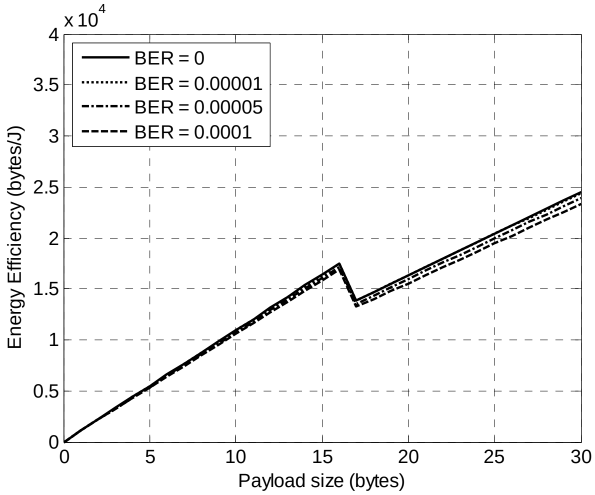
The analysis results for the five schemes are shown in Figure 11. These figures illustrate the energy efficiency for different payload sizes. In the analysis, we also present the results under different channel bit error rates. As expected, as BER increases, for all schemes the energy efficiency decreases due to the necessity of discarding many corrupted packets.

(1) CBC + Implicit IV Scheme

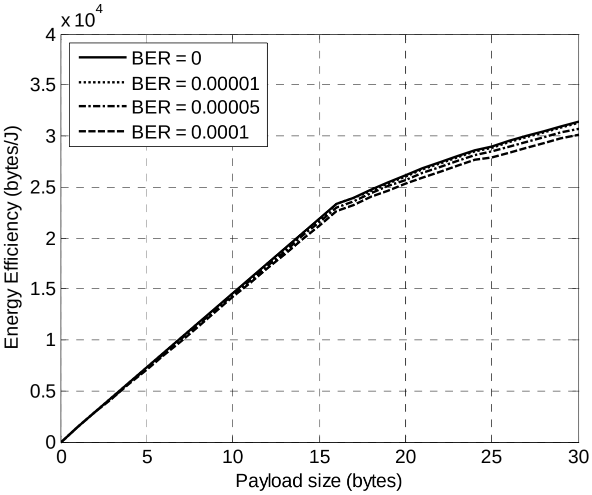
In Figure 11(a), we can see that the curve for the *implicit IV* scheme shows a sawtooth shape, because CBC mode is used in this scheme and the size of encryption result is always an integer multiple of the block size. The extra bits being transmitted due to padding the plaintext data out to a multiple of a block size consume energy while providing no informational content to the communication.

(2) TinySec Scheme

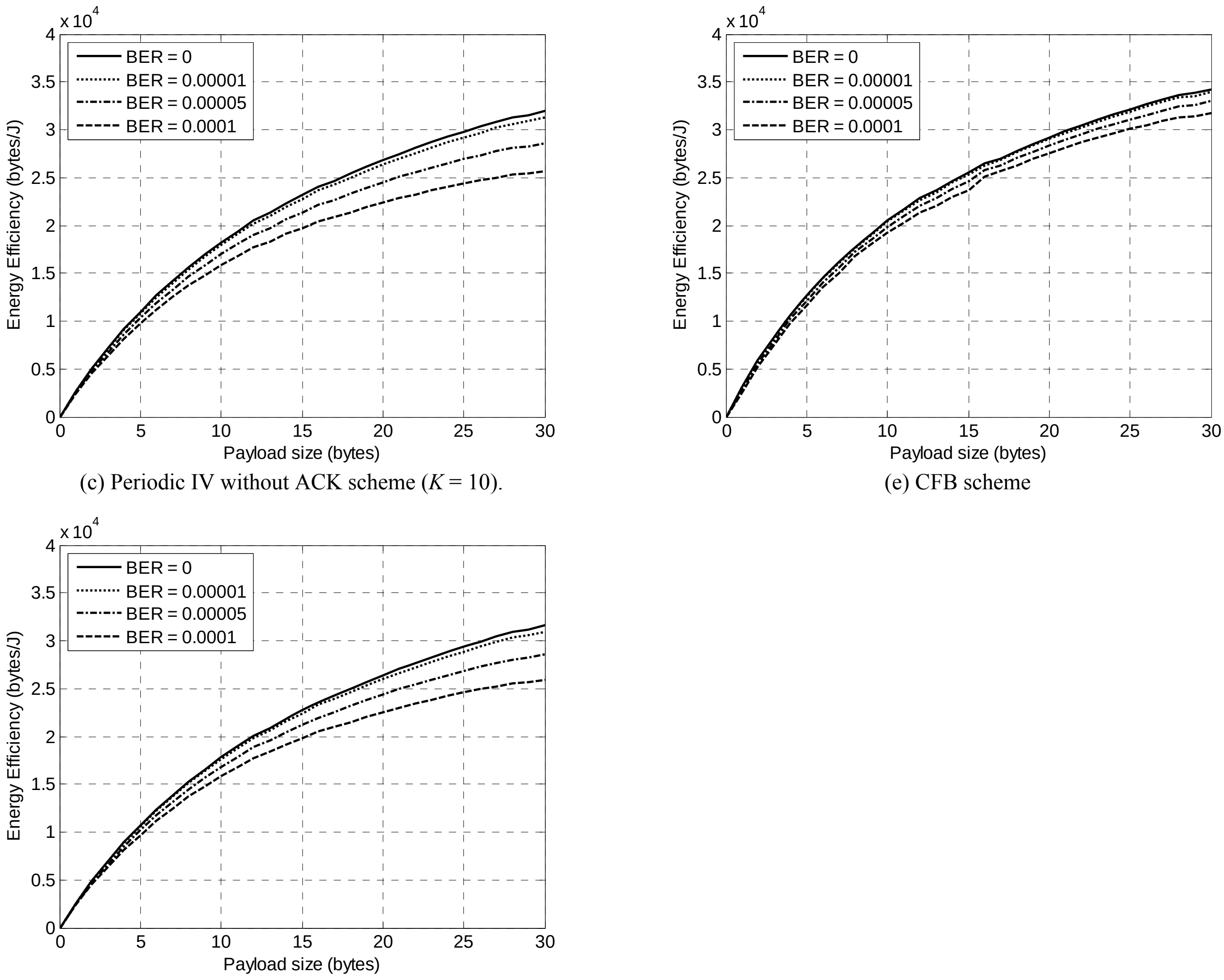
Figure 11(b) shows the results for the specific format of TinySec, which uses CBC (with ciphertext stealing) and includes the IV in each packet. As seen in the figure, the slope decreases as the payload size increases. This occurs because the ratio of the IV size to the packet size decreases. Since AES uses a 128-bit block size, for payload sizes of 1 to 16 bytes, the plaintext is padded out to 128 bits and a full 128 bits of ciphertext will be produced resulting in a straight line for the corresponding portion of the graph. When the payload size is larger than the block size, the transmitted ciphertext size is the same as the plaintext size, since ciphertext stealing can be used.



(a) Implicit IV scheme



(b) Tinysec scheme



(c) Periodic IV without ACK scheme (K = 10).

(e) CFB scheme

(d) Periodic IV with ACK scheme (K = 10).

Fig. 11. Analysis for different channel qualities.

(3) Counter Mode + Periodic IV without ACK scheme

In Figure 11(c), it can be seen that, for the *periodic IV without ACK* scheme applying counter mode, at high bit error rate, the energy efficiency decreases dramatically compared to lower bit error rate. This indicates that the periodic IV approach has a relatively poor performance in a poor channel conditions. 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.

(4) Counter Mode + Periodic IV with ACK scheme

As shown in Figure 11(d), the trend of this scheme is similar to the *periodic IV without ACK* scheme. The acknowledgement does not improve the performance of the scheme very much. This is because when the channel quality is poor, the data packet also has a large probability of being corrupted, which will impact the following data packets until a new IV is established.

(5) CFB Mode + Ciphertext based IV scheme

For the CFB scheme, we can see from the Figure 11(e) that the CFB scheme achieves better energy efficiency for all the payload sizes compared to other schemes. We can also see that, compared to the *periodic IV* schemes, there is little spread in the curves when bit error rate increases.

5.3.3 Other Considerations

(1) Effect of Block Size

Consider the *implicit IV* scheme which includes the IV in each packet. When CBC mode (without ciphertext stealing) is used on a block cipher, the block size of the cipher influences the energy efficiency. Since the block cipher processes data block by block, zeroes will be added to the end of the data if the payload size is not an integer multiple of the block size. For CBC mode, a cipher of larger block size tends to cause more energy inefficiency since extra communication energy cost is consumed to transmit the added dummy bits. Take a 7-byte payload for example: 72 bits of zeroes need to be added for a 128-bit block cipher while only 8 bits are needed for a 64-bit block cipher. When the payload size equals a multiple of the block size there is no extra communication energy cost introduced. When used in CBC mode, compared to cipher AES with a 128-bit block size, a 64-bit cipher such as BSPN causes fewer extra bits to be transmitted resulting in higher energy efficiency across the different payload sizes. For modes, such as counter mode and CFB mode, which do not result in extra bits of ciphertext to be transmitted to fill out blocks, the effect of different block sizes is very small, only being dependent on the efficiency of the block cipher.

(2) Effect of IV Size

The size of IV directly impacts the energy cost when IV is included in the transmitted packet. As is exploited by the TinySec scheme, reducing the IV size can lead to a better energy performance. Figure 12 shows the effects of different IV sizes for the condition of BER equal 10^{-4} using the *implicit IV* scheme with CBC mode for AES. It can be seen that the energy efficiency increases when the IV size decreases. However, for security purposes, the IV size should be large enough to ensure that the IV is not repeated.

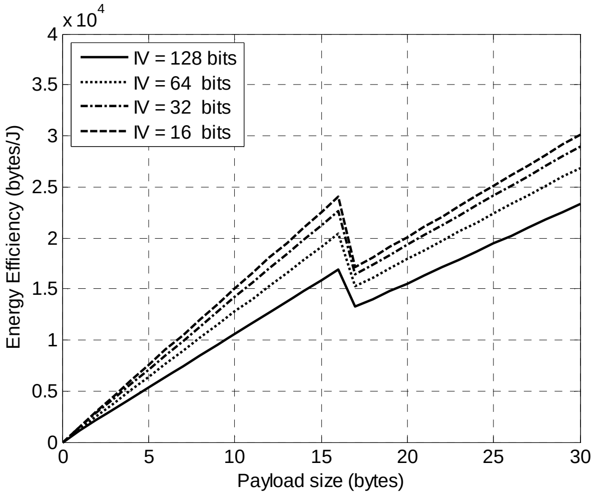


Fig. 12. Analysis for different IV sizes (implicit IV scheme, BER=10^{-4}).

(3) Effect of IV Period

In the schemes which periodically transmit an IV packet to establish synchronization, the value of *K*, together with the size of the payload, impacts energy efficiency. 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, smaller *K* causes significant communication energy cost to transfer the IV packet, which will also decrease the energy performance. It is desirable to determine the optimal value of *K* to balance these two factors. The effect of period *K* and the payload size on the energy efficiency is shown in Figure 13, which is plotted for the *periodic IV without ACK* scheme. In this figure, the cipher AES is applied and BER is 10^{-4}. It can be seen that, as the payload size increases, the energy efficiency can obtain a maximum value with an optimal selection of period *K*. The energy efficiency can be expressed as

$$\eta = \left(\frac{K}{K \times E_{data} + E_{iv}} \times \frac{P_{iv} \times \left(P_{data} - P_{data}^{K+1} \right)}{K \times (1 - P_{data})} \right) \times N_{pld} , \qquad (14)$$

where E_{data} and E_{iv} represent the energy cost for transmitting a data packet and an IV packet, respectively. The optimal value of *K* for maximizing the value of η can be derived by solving $d\eta / dK = 0$ for *K*. Figure 14 illustrates the optimal period *K* according to payload sizes under different BER conditions. From this figure, we can see that the optimal period becomes smaller when the channel quality is poor.

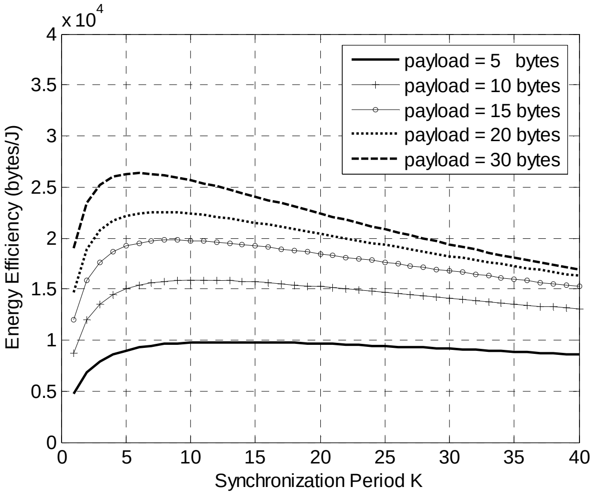


Fig. 13. Effects of period and payload size (AES, BER = 10^{-4}).

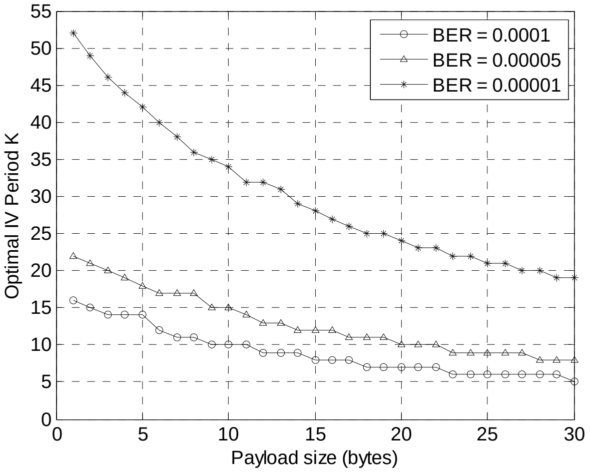
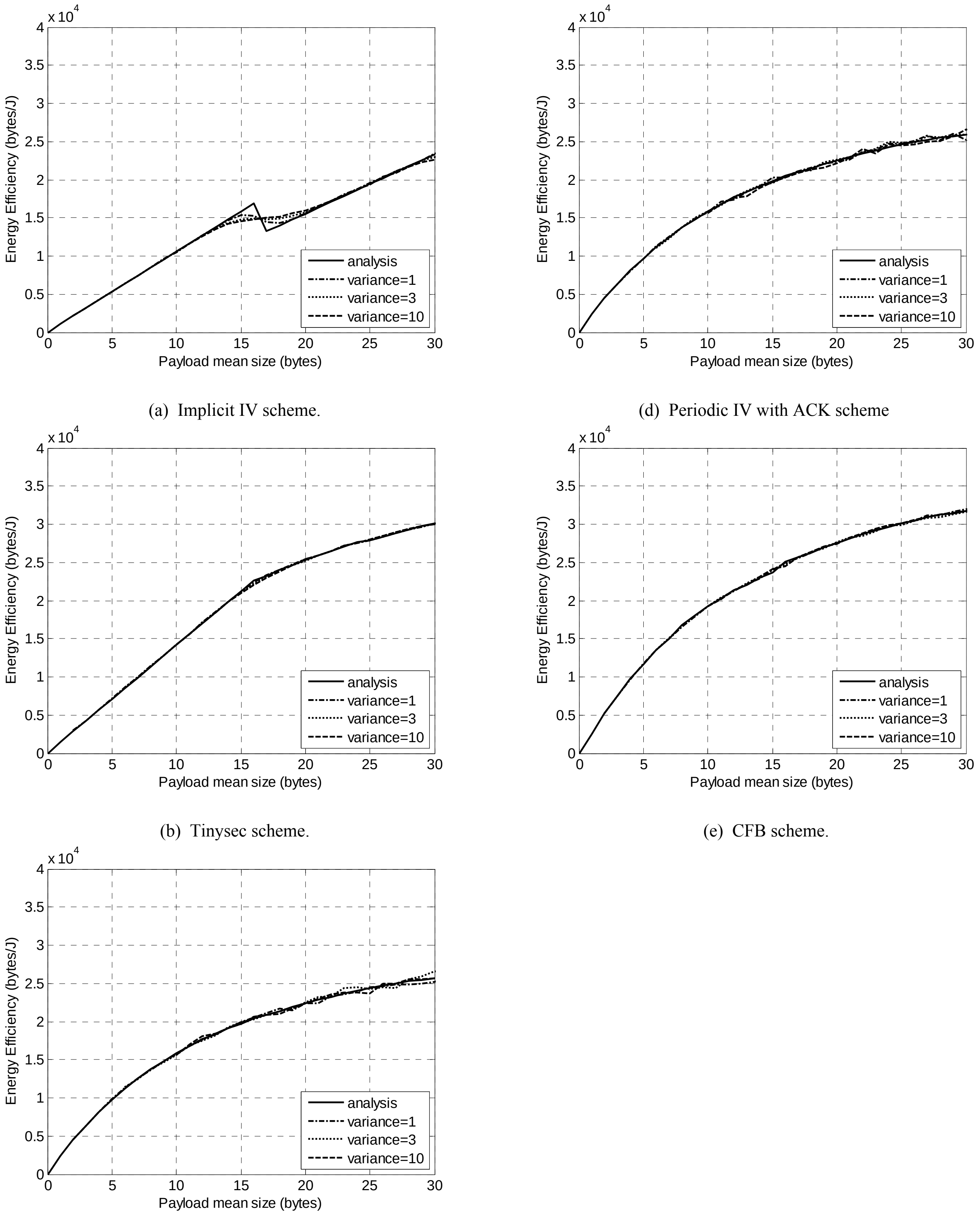


Fig. 14. Optimal *K* for different channel qualities.

5.4 Simulation Results for Variable Size Packets

To verify the suitability of the analysis model, we have performed extensive simulations for both fixed size packets and variable size packets. In our work, each scheme has been simulated using several types of probability distributions for payload sizes with different variances. The distributions considered are the binomial distribution, Poisson distribution, geometric distribution, and discrete uniform distribution. For example, in Figure 15, each scheme is simulated using a binomial distribution with different variances. We can see that they are very similar to the results of the analysis model which is based on fixed size payloads. Simulation results for other probability distributions also show that the resulting curves are very similar to the curves determined by the analysis model. This is illustrated in Figure 16, where variable size packets with different distributions are used to simulate the CFB scheme and the results are presented along with fixed size analysis results. We conclude that the analysis model approximates well the energy cost behavior for sensor nodes for a large variety of packet size distributions.



(a) Implicit IV scheme.

(d) Periodic IV with ACK scheme

(b) Tinysec scheme.

(e) CFB scheme.

(c) Periodic IV without ACK scheme.

Fig. 15. Energy efficiency simulation with binomial distribution.

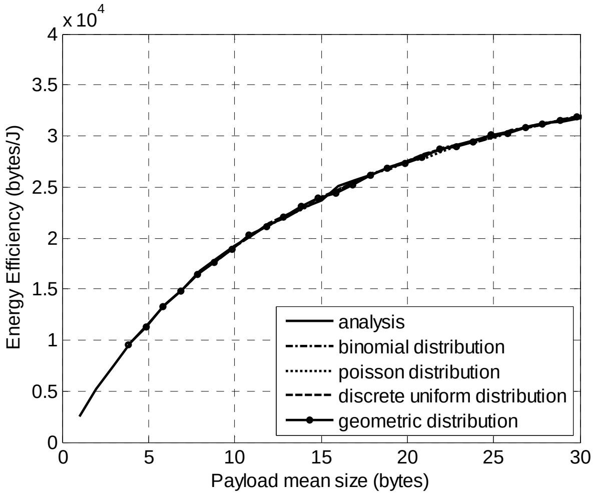


Fig. 16. Analysis for CFB scheme with different distributions (variance = 10, BER=10^{-4}).

6 Comparison of Cryptographic Schemes

In this section, we compare the performance of the five 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. We also calculate the improvement of the CFB scheme compared to other schemes, which is obtained by

$$\text{Improvement} = \frac{\eta_A - \eta_B}{\eta_B} \tag{14}$$

where η_A is the energy efficiency of the CFB scheme and η_B is the energy efficiency of the scheme we choose for comparison.

6.1 Error-free Channels

Figure 17 compares the cryptographic schemes utilizing the cipher AES under the condition of an error-free channel. We can see that schemes of IV transmitted periodically (where we have used K = 10) achieve better results than the schemes of IV included in each packet. This is because *periodic IV* schemes can do the IV distribution without losing counter synchronization when the channel is error-free. In contrast, schemes transmitting the IV with every packet cost more energy than the periodic IV schemes. The CFB scheme also does not transmit IV within the packet and achieves a slightly better result than the *periodic IV* schemes.

The performance improvement of the CFB scheme compared to other schemes under the error-free channel is shown in Figure 18. The CFB scheme achieves significant improvement over the other schemes. For example, for small payloads of less than 10 bytes, the CFB scheme has an improvement of 40% or more over the TinySec scheme. In general, the exact value of improvement depends on the packet size with improvement being most notable for smaller payloads.

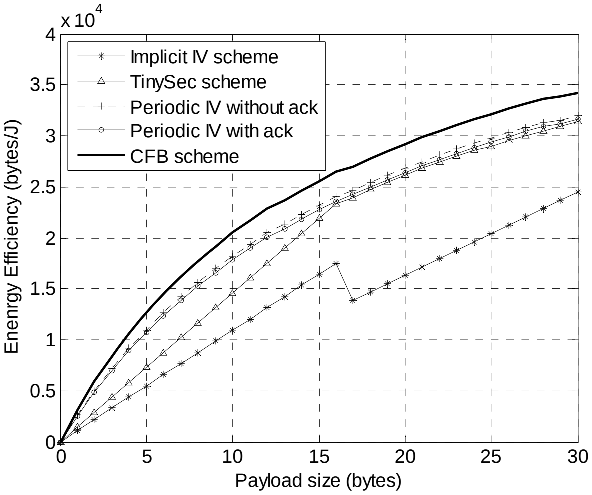


Fig. 17. Comparison of schemes (BER = 0).

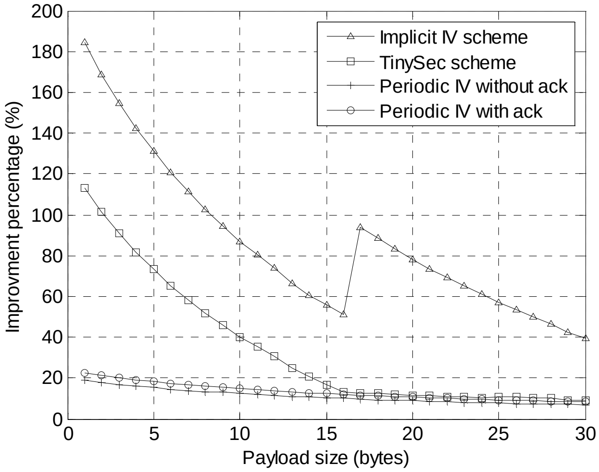


Fig. 18. Improvement of CFB scheme (BER = 0).

6.2 Noisy Channels

The results change significantly in a channel with noise, as shown in Figure 19 which is based on BER of 10^{-4}. The energy efficiency of *periodic IV* schemes (for which we set K = 10) 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 TinySec scheme which includes the IV information in every packet achieves 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 *implicit IV* scheme achieves the worst performance because sending the full size IV in each packet introduces too much communication energy cost. The CFB scheme still achieves the best performance result among all these 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 successfully decrypted packet.

Figure 20 shows the performance improvement for varying payload sizes under the noisy channel of BER equal to 10^{-4}. The comparison result illustrates that the CFB scheme achieves significantly better performance than all other schemes even when the channel is noisy. Typically, improvements are most significant for smaller size payloads.

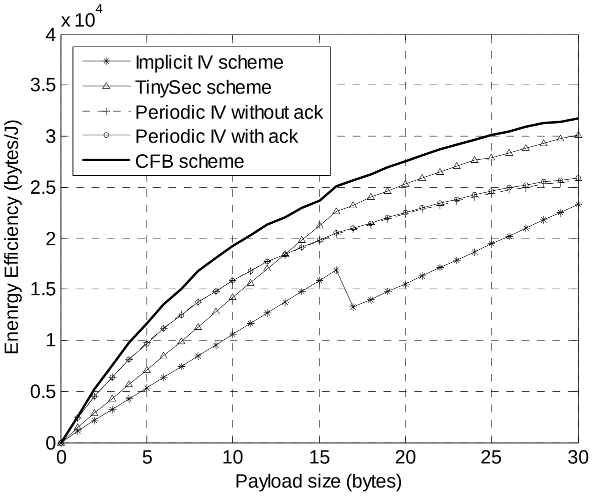


Fig. 19. Comparison of schemes (BER = 10^{-4}).

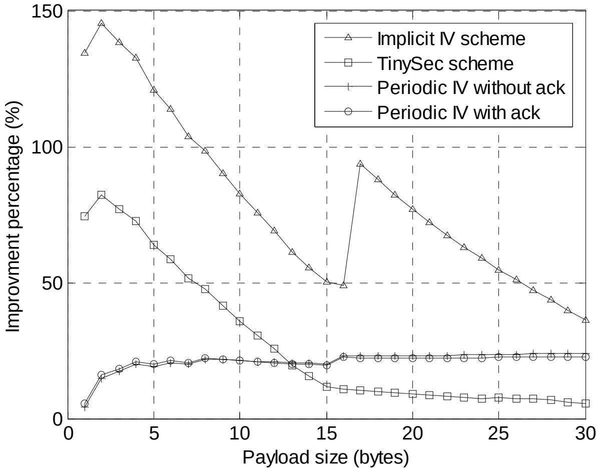


Fig. 20. Improvement of CFB scheme (BER = 10^{-4}).

In Figure 21, we explore the performance of the various schemes as the BER of the channel varies. Fixed payload sizes of 20 bytes are assumed. For all schemes, the energy efficiency drops dramatically as BER increases to large values. It is clear that for most ranges of BER, the CFB scheme provides greater energy efficiency than other schemes. The only exception to this occurs when the BER becomes very high (> 5 × 10^{-4}). In such cases, the small overhead and independent nature of each packet decryption allows the TinySec scheme to perform slightly better.

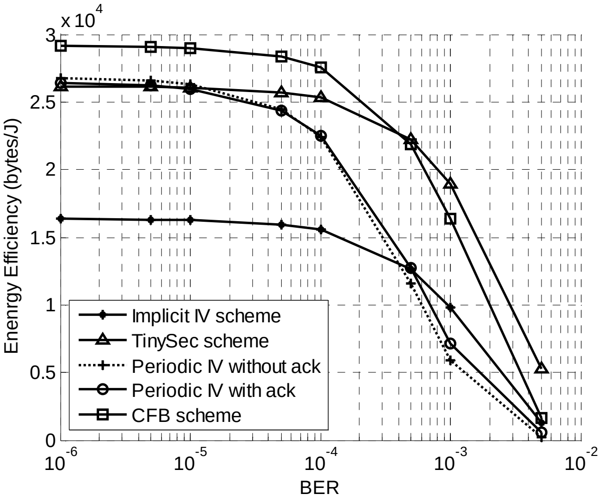


Fig. 21. Comparison of schemes for varying BER (payload size = 20 bytes).

6.3 Effect of Cryptographic Algorithm

To consider the effect of the cryptographic algorithm on the overall energy efficiency, we have applied different ciphers to the CFB scheme and *periodic IV without ACK* scheme with $K = 10$. For the CFB scheme, we also consider the case of no computational energy cost for calculating the MAC, which represents a lower bound for the energy cost of an advanced mode of operation which integrates encryption and authentication. The results are shown in Figure 22. We can see that both the cryptographic algorithm and cryptographic scheme affect the final result. However, the effect of using different ciphers is relatively small. Clearly, the effect of using different schemes is more significant. For example, there is a small difference between AES and BSPN when using the same cryptographic scheme. From the figure, we can also see that the effect of computational energy cost for MAC is also small, since little difference is shown for the CFB scheme using AES whether the computational energy cost of the MAC is included or not.

As a broadly accepted secure cipher, AES seems to be a good choice for WSNs. However, our analysis is based on a model with a one way nature of the communication: the sensor node exclusively encrypts and transmits data packets. It is worth noting that the decryption of AES is much less efficient than its encryption. Our implementation result shows that, compared to the 3266 cycles needed to encrypt a plaintext block, 49864 cycles are needed to decrypt a ciphertext block. In some scenarios, a sensor node may need both the function of encryption and decryption and the large number of cycles for decryption may dramatically decrease the energy performance of the sensor node. For example, if data is transmitted to a sensor node encrypted using AES in CBC mode, then the sensor node must use the AES decryption operation to decrypt the data. If communication in a WSN is bidirectional resulting in a significant amount of decryption operations, AES should either not be used or used in a mode such as counter or CFB that only requires the encryption functionality.

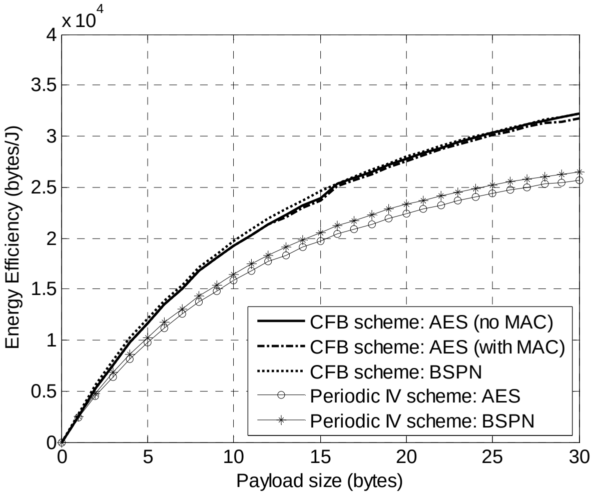


Fig. 22. CFB scheme with different ciphers (BER = 10^{-4}).

7 CONCLUSION

In this paper, we have investigated the performance of secure data transmission at the link layer in wireless sensor networks, considering both the cryptographic algorithms and cryptographic schemes. The energy efficiency will be directly affected by many factors, such as the algorithm utilized, the mode of operation, the IV distribution, the packet size and the bit error rate of the channel. We proposed using the amount of valid data transferred per Joule from the sensor node as the metric to evaluate the energy efficiency when cryptographic schemes are applied to sensor node communication. We have compared the energy efficiency among several typical cryptographic schemes by developing an analysis model. Our results suggest that a cryptographic scheme using a symmetric key block cipher, such as AES, in cipher feedback mode achieves better performance for a wide range of channel qualities and provides significant improvement in energy efficiency compared to other schemes.

REFERENCES

[1] 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.

[2] 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, 2002.

[3] 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 *International Conference on Information Security and Assurance (ISA 2008)*, pp.73-76, Busan, Korea, Apr. 2008.

[4] 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 *IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, pp. 324-328, Hawaii, Mar. 2005.

[5] M. Henricksen, "Tiny Dragon - An Encryption Algorithm for Wireless Sensor Networks," in Proc. of *IEEE International Conference on High Performance Computing and Communications, (HPCC '08)*, pp. 795-800, Dalian, China, Sep. 2008.

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

[7] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," in Proc. of *ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162-175, Baltimore, Maryland, Nov. 2004.

[8] T. Li, H. Wu, X. Wang, and F. Bao, "SenSec Design. Technical Report-TR v1.1," InfoComm Security Department, Institute for Infocomm Research, 2005, [online]. *Available: http://icsd.i2r.a-star.edu.sg/staff/tieyan/SecureSensor/papers/SenSec-TR-I2R.pdf.*

[9] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a Secure Sensor Network Communication Architecture," in Proc. of *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2007)*, pp. 479-488, Cambridge, Massachusetts, Apr. 2007.

[10] D. Jinwala, D. Patel, and K. Dasgupta, "FlexiSec: A Configurable Link Layer Security Architecture for Wireless Sensor Networks", *Journal of Information Assurance & Security: Special Issue on Information Assurance and Data Security*, Dynamic Publishers, vol. 4, no. 6, pp. 582-603, 2009.

[11] L. Casado and P. Tsigas, "ContikiSec: A Secure Network Layer for Wireless Sensor Networks under the Contiki Operating System," *Lecture Notes in Computer Science 5838: Identity and Privacy in the Internet Age*, Springer-Verlag, pp. 133-147, 2009.

[12] S. Li, T. Li, X. Wang, J. Zhou, and K. Chen, "Efficient Link Layer Security Scheme for Wireless Sensor Networks", *Journal of Information And Computational Science*, Binary Information Press, vol.4, no.2, pp. 553-567, 2007.

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

[14] "Wireless Microsensor Mote", [online]. *Available: http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf*.

[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," in Proc. of *ACM International Conference on Architectural Support for Programming Languages and Operating Systems, (ASPLOS IX)*, pp. 93-104, Cambridge, Massachusetts, Nov. 2000.

[16] "8-bit Microcontroller with 128K Bytes In-System Programmable Flash", [online]. Available : http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.

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

[18] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A Survey of Lightweight Cryptography Implementations", *IEEE Design & Test of Computers: Special Issue on Secure ICs for Secure Embedded Computing*, IEEE, vol 24, no 6, pp. 522-533, 2007.

[19] B. Schneier, *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 1996.

[20] "SkipJack and KEA Algorithm Specifications (Version 2.0)," *National Institute of Standards and Technology*, May 1998.

[21] A. Youssef, S.E. Tavares, and H.M. Heys, "A New Class of Substitution-Permutation Networks", presented at *Workshop on Selected Areas in Cryptography (SAC '96)*, Kingston, Canada, Aug. 1996.

[22] X. Zhang, H.M. Heys, and C. Li, " Energy Efficiency of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks," in Proc. of *25th Biennial Symposium on Communications (QBSC'10)*, pp. 168-172, Kingston, Canada, May 2010.

[23] 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.

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

[25] 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. 165-180, 1999.

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

[27] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert, "SOSEMANUK: A fast software-oriented stream cipher." [Online]. *Available: http://www.ecrypt.eu.org/stream/p3ciphers/sosemanuk/sosemanuk p3.pdf.*

[28] D. J. Bernstein, "Salsa20." [Online]. *Available: http://www.ecrypt.eu.org/stream/p3ciphers/salsa20/salsa20 p3.zip.*

[29] G. Meiser, T. Eisenbarth, K. Lemke-Rust, and C. Paar, "Efficient Implementation of eSTREAM Ciphers on 8-bit AVR Microcontrollers," in Proc. of *International Symposium on Industrial Embedded Systems (SIES'08)*, pp. 58-66, Montpellier, France, Jun. 2008.

[30] National Institute of Standards and Technology (NIST), "Recommendation for Block Cipher Modes of Operation Methods and Techniques," *NIST Special Publication 800-38A*, 2001.

[31] National Institute of Standards and Technology (NIST), "Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode," *Addendum to NIST Special Publication 800-38A*, 2010.

[32] National Institute of Standards and Technology (NIST), "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality," *NIST Special Publication 800-38C*, 2007.

[33] P. Rogaway, M, Bellare, J, Black, and T. Krovetz, "OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption", Aug. 2001, [Online]. *Available:* *http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ocb/ocb-spec.pdf* .

[34] National Institute of Standards and Technology (NIST), "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", *NIST Special Publication 800-38D*, 2007.