An Energy Efficient Link-Layer Security Protocol for Wireless Sensor Networks

Leonard E. Lighfoot Jian Ren Tongtong Li
Department of Electrical & Computer Engineering
Michigan State University
East Lansing, MI 48824
email: {lightf10, renjian, tongli}@egr.msu.edu

Abstract— **In recent years, wireless sensor networks (WSNs) have found use in a variety of different applications including environmental monitoring, battlefield strategy planning, health monitoring, and so forth. With many of the applications involving communication of highly sensitive data, security becomes a primary concern. However, incorporating security features in WSNs is challenging due to the specific constraints such as limited memory and restricted energy supply. In this paper, we investigate the importance of the link-layer security service in WSNs and propose an energy efficient link-layer security protocol (LLSP) to reduce the energy consumption in the network. Comparing with the existing TinySec protocol, which provides node authentication, message confidentiality and access control, the proposed LLSP protocol provides replay protection while reducing the security overhead per packet by 17%. Throughout this paper we analyze and compare the performance as well as the energy consumption of TinySec and LLSP security protocols. Furthermore, we investigate the throughput of the system over both error-free and lossy channels using the LLSP protocol. In addition to stronger security measures, the simulation results demonstrate that LLSP outperforms TinySec in terms of energy reduction and throughput performance.**

I. Introduction

Wireless sensor networks (WSNs) are defined as networks consisting of independent, collaborating nodes that can sense, process and exchange data as well as act upon the data content. Independently each node is limited in its capability, but jointly the data-centric network can deliver time sensitive information to different destinations. WSNs have found their way into many applications as the manufacturing of small and low cost sensors has become technically and economically feasible. We can envision a future with thousands of nodes networked to operate unattended sensing tasks. Such networks are expected to be widely deployed in a variety of environments for commercial, civil, and military applications including large-scale acoustic surveillance, climate and habitat monitoring and ground target detection [9], [10].

By nature WSNs rely on a broadcast medium and is more vulnerable to security attacks than its wire counterpart due to the lack of physical boundary [2]. An adversary with an appropriate transceiver can eavesdrop, intercept, inject and alter the transmitted data. As a result, a security service is a necessity to ensure information confidentiality and effective access control in WSNs. Frequently, security services are abandoned in network design because of the limited capability of the sensor nodes. Consequently, this leave WSNs under security attacks, which could consume excessively more battery power and shorten the longevity of the WSNs.

The structure, design and vision of WSNs, introduces many challenges to implement an efficient security protocol. Typically, WSNs are constrained in energy and bandwidth since the sensors are small, low-power, and low-cost devices. These constraints conflict with the costly well-known security algorithms, which are designed for powerful workstations. Hence, to design secured protocols in WSNs the following elements are taken into consideration: a secure access control protocol, simple but effective data encryption/decryption algorithm, low communication overhead, and efficient and practical key management scheme.

Previous works, such as SNEP [1], provides a solid infrastructure to designing efficient and secure protocols for WSNs. However, SNEP is neither fully specified nor fully implemented. Also, researchers [11] have proven that SNEP is too expensive for the low-end devices. Other security protocols such as Cipher Block Chaining Message Authentication Code Protocol (CCMP) [12], provides strong message authentication, but requires a huge communication overhead per message packet. The TinySec [6], which specifically target sensor networks, is used as a foundation for the proposed link-layer security protocol.

In this paper, an energy efficient link-layer security protocol (LLSP) is proposed. Comparing with the TinySec protocol, the LLSP security protocol reduces the energy consumption by minimizing the security overhead for each message packet while maintaining security primitives. In an effort to prove the efficiency and security of the LLSP, the energy consumption and throughput of the LLSP and TinySec protocols are measured using the TinyOS simulator.

II. Security Requirements for Wireless Sensor Networks

Incorporating a security protocol in WSNs is a challenging, but necessary feature to include in the network design. An adequate security service can prevent malicious power consumption attacks, ensure effective access control, and provide

1-4244-0941-1/07/$25.00

information confidentiality [2]. Below are four basic security requirements for a superior link-layer security protocol.

A. Message Confidentiality

The goal of message confidentiality is to keep information secret from unauthorized parties. Typically, confidentiality is achieved with encryption. A proficient encryption scheme prevents an adversary from recovering an encrypted message and prevents an adversary from learning partial information about the encrypted message.

Due to the broadcast nature of WSNs, data encryption is extremely important. Wireless routed data makes it easier for eavesdroppers and adversaries to capture messages. Ideally, a strong encryption scheme is sought for wirelessly transmitted data, but generally, the stronger the encryption scheme, the more energy inefficient it becomes. This is due to the lengthy key storage and extra computational processing required for strong encryption algorithms.

B. Message Authentication and Access Control

Message authentication ensures a receiver is capable of detecting a modified or altered received message and is generally achieved by appending a message authentication code (MAC) to each transmitted packet. A MAC is essentially a one-way hash function with a secondary secret key input. A one-way hash function maps bit-strings of arbitrary finite length into strings of fixed length, such that it is computationally infeasible to find any input which hashes to any pre-specified output. A hash function also makes it is computationally infeasible to find any second input which has the same output as any specified input [13].

Due to the MAC's cryptographic computation, it can also be used to provide access control in WSNs. Access control imply that the link-layer protocol prevent unauthorized parties from participating in the network. A MAC allows legitimate nodes the ability to detect and reject messages from unauthorized nodes.

C. Replay Protection

Replay protection prevents messages from being re-sent to an authorized receiver. It ensures that all messages received by the receiver are fresh and has never been transmitted previously. If an adversary eavesdrops on a message between two authorized nodes and later replays the message to the receiver, the receiver will accept the message because the message is originally from an authorized sender. Due to the limited amount of state that each recipient can keep, replay protection is difficult to ensure.

A typical defense is to include a monotonically increasing counter with every message and reject messages with old counter values [2]. However, this method requires each recipient to maintain a table of the last value from every sender it receives. For RAM-constrained sensor nodes, this method is problematic for even modestly sized networks. Also, transmitting the counter value for each message packet is energy inefficient because it is widely known that communication is the dominant source in energy depletion for these sensor nodes.

III. TinyOS and TinySec Protocol

TinyOS is a flexible, application-specific operating system for sensor networks [3]. TinyOS consist of a simple FIFO task scheduler and numerous software components for radio communication, sensing, EEPROM access and other devices. A TinyOS application is assembled by linking multiple software components into an optimized binary, not in a binary kernel. Also, TinyOS is written in NesC [7], a C-based programming language. Over the past few years, the popularity of TinyOS has accelerated due to its support for several common sensor node platforms, such as Mica, Mica2, MicaZ, Telos, MSP430 and the AVRMote.

A. TOSSIM and PowerTOSSIM

The TinyOS environment has a built in simulator call TOSSIM [3] and energy simulator called PowerTOSSIM [5]. TOSSIM allows users to debug, test and analyze algorithms in a controlled and repeatable environment. PowerTOSSIM is an extension of TOSSIM and provides an accurate per node estimate of power consumption. In PowerTOSSIM, specific hardware peripherals such as radio, EEPROM, LEDs and CPU are instrumented to obtain a trace of each peripheral's activity during the simulation run time. PowerTOSSIM energy model is based on the Mica2 sensor node platform [5].

B. TinySec Protocol

TinyOS also has a standard security protocol call TinySec [6]. The TinySec protocol provides two options for security: *authentication only* and *authentication and encryption*. The authentication only security option authenticates the entire packet with a message authentication code, but the data payload is not encrypted. The authentication and encryption security option encrypts the data payload and authenticates the packet with a MAC.

The authentication and encryption security option uses a cipher block chaining (CBC) scheme called Skipjack along with a specially formatted 8-byte initialization vector (IV) to encrypt the data. The structure of the IV is $Dest||\text{AM}||Len||Src||Ctr$, where *Dest* is the destination address, AM is the active message handler type, *Len* is the data length, *Src* is the source address, and *Ctr* is the counter value. The active message handle types are similar to port numbers in TCP/IP and specifies the appropriate handler function to extract and interpret the message on the receiver. *Ctr* is a 2-byte counter that starts at 0 and increments by 1 after each message sent by the sender.

IV. Design of Link Layer Security Protocol

In order for WSNs to be widely deployed, an adequate security protocol must be integrated into the sensor network design. The proposed LLSP security protocol addresses the security issues mentioned in this paper. In addition, the LLSP security protocol guarantees message authentication, access

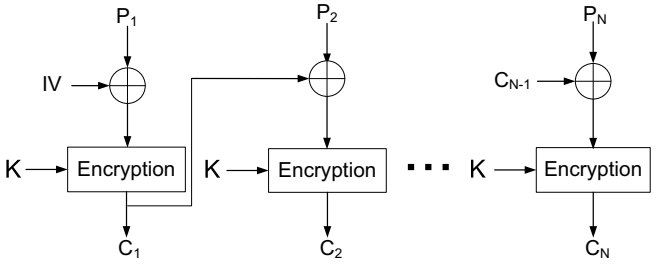


Fig. 1. Cipher Block Chaining (CBC) mode encryption

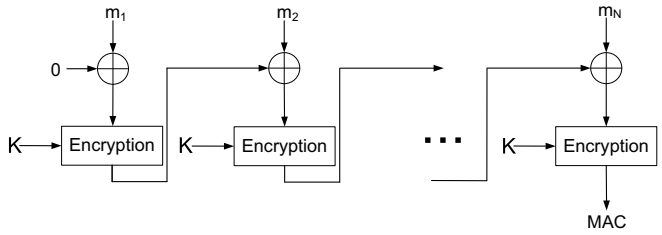


Fig. 2. Cipher Block Chaining Message Authentication Code (CBC-MAC)

control, message confidentiality and replay protection. In the following subsections we discuss the security services of the LLSP security protocol.

A. Message Confidentiality

Data encryption is a method of achieving message confidentiality when transmitting data through an unsecured medium. For the LLSP security protocol, we propose the Advance Encryption Standard (AES) with cipher block chaining (CBC) mode of operation as the data encryption scheme. A depiction of the CBC encryption scheme is shown in Figure 1, where AES is the encryption scheme. As shown in the figure, the initial plaintext *P1* is bitwise exclusive-or (XOR) with an initialization vector (IV). The result is encrypted with a shared key *K* to produce the initial ciphertext *C1*. The second round of the encryption scheme is identical to the first round except the resulting ciphertext *C1* is used as the IV. This process is repeated for the desired number of rounds of the encryption method.

The unique design of AES-CBC provides semantic security, which implies that encrypting the same plaintext twice, will produce two different ciphertexts. Semantic security is achieved by adding a unique IV to the encryption scheme. As shown in Figure 1, the IV is a side input to the encryption algorithm. The IVs are used to provide variation to the encryption process when there is little or no variation between the set of messages.

B. Message Authentication

Due to the unreliability and random characterization of wireless channels, they are more vulnerable to transmission errors than its wired counterpart. Typical communication protocols provide packet error checking with a cyclic redundancy check (CRC). However, a known flaw of CRC is that it does not protect against malicious modifications or forgery of packets.

To guarantee message authentication and access control, LLSP uses a message authentication code (MAC). Message authentication prevents unauthorized nodes from participating in the network and ensures received messages are not altered, thus inherently assuring the message contains no errors. A MAC is essentially a cryptographically secure checksum of a message. Computation of the MAC is based on a cryptographic hash function and a secret shared key between the sender and receiver.

Prior to packet transmission, the MAC is computed and appended to the message *m*. The receiver and sender share a secret key, therefore once the receiver receives the message, the receiver can recalculate the MAC. If the two MACs are equal, then the receiver keeps the packet and discards the packet if the MACs are not equal. The hash function ensures that, if an adversary alters a valid message or injects a malicious message, the receiver will reject the message because the receiver is unable to recompute the correct MAC value.

For LLSP security protocol, we propose the Cipher Block Chaining Message Authentication Code (CBC-MAC) to provide message authentication and access control. A depiction of the CBC-MAC encryption scheme is shown in Figure 2. As shown, the CBC-MAC uses a scheme similar to CBC, but the IV is initialized to zero. The similar computation method allows for code reuse thus reducing the memory usage.

C. Replay Protection

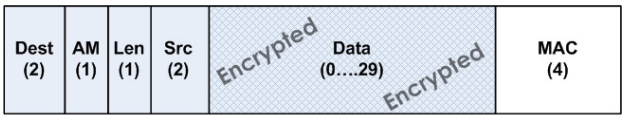
In a replay attack, an adversary eavesdrops between two authorized sensor nodes and replay the message to the receiver at a later time. Typically, a counter value is used to maintain record of the received messages from a node. If the authorized receiver has a record of the received message, it can detect a replayed message and reject it. But, if there is no record of the received message then the receiver will accept the message again, consequently increasing the energy consumption. For large WSNs it is impractical for each node to maintain record of each senders message count. However, if the sensor node has knowledge of the network topology and power efficient routing, then counters are only necessary for the number of nodes directly in its communication range. Typically, the number of sensors in a communication range is small. Therefore, it is practical for a sensor node to maintain a counter for each node that is in its communication range.

The proposed LLSP security protocol maintains a synchronous 4-byte counter between the sender and receiver pair. The feedback shift register (FSR) is used to update the 4-byte counter. Recall that the TinySec security protocol maintains a 2-byte counter by transmitting the counter value in each message packet. The FSR design allows energy to be saved by eliminating the transmission of the counter bytes in each message packet.

D. Packet Format

The packet format for the LLSP security protocol is based on the packet format of the TinySec protocol. Figure 3, depicts both security protocols. The two packet formats differ in the absence of the counter bytes in the LLSP packet format. Although the counter value is not included in the packet format

(a) TinySec – Authentication & Encryption packet format



(b) Link Layer Security Protocol packet format

Fig. 3. TinySec protocol and Link Layer Security Protocol packet formats. The byte size of each field is indicated below the label. In both packet formats, the grided area is encrypted.

for LLSP, it is included in the IV format and needed for the computation of the MAC. The structure of the IV consists of *Dest*, *AM*, *Len*, *Src*, and *Ctr* bytes appended together. The counter value is included in the structure of the IV to add variation to the encryption, which will reduce the risk that the IV is repeated. IV reuse may severely compromise the security of the network. Similarly, the counter value is included in the computation of the MAC. Calculation of the MAC for the LLSP protocol is shown in Equation 1,

$$MAC = H(K, Dest \| AM \| Len \| Src \| Ctr \| Data), \quad (1)$$

where *Data* is the encryption of the sensor reading or other information.

As shown in the packet format, the packet header (*Dest*, *AM*, *Len*, and *Src*) and the MAC are not encrypted. The benefits of not encrypting the packet header and MAC outweighs the benefits of keeping them a secret. Early rejection, a power saving mechanism for sensor nodes, where the nodes turns off its sensor radio after determining the message is not addressed to it. However, if the packet header is encrypted, early rejection cannot be invoked until the packet header is decrypted. If an adversary wants to disturb the sensor network, it can repeatedly transmit messages to the sensor thus forcing the sensor to consume energy by decrypting pointless messages. Early rejection can also be achieved if the MAC is not encrypted. This allows the receiver to determine the authenticity of the message with minimum energy consumption since the decision can be made without requiring the decryption of the data packet. The LLSP security protocol also allows the receiver to determine the number of lost packets based on the correct counter value that generates the MAC.

Furthermore, the LLSP security protocol reduces the energy consumption without decreasing the security, by removing the 2-byte counter values from the security overhead. Each sender and receiver pair maintain a synchronous counter generated through a FRS. As a result, the counter value is not transmitted with the message packet. In [8], the authors state that the transmission of 1 bit consumes about as much power as executing 8,000-1,000 instructions for the Mica2 mote. LLSP's 2-byte security overhead reduction is equivalent to not executing 12,800-16,000 instructions for each packet, which equates to less consumption of energy and an increase in network lifespan.

V. Evaluation

A. Security Analysis

The LLSP security protocol focuses on three data security services: message authentication, replay protection, and message confidentiality. There are many cryptographic algorithms that provide all three security services and LLSP is not limited to any cryptographic algorithm. It is strongly recommended that only well-known symmetric cryptographic algorithms be applied to ensure sensor network security and implementation efficiency.

The length of the IV significantly affect the security and energy consumption of the security protocol. If the IV is too long there will be unnecessary bits added to the packets, which translates to significant cost in energy and bandwidth. At the same time, if the IV is too short there is a risk of IV reuse and then the security is compromised. The LLSP security protocol uses a 10-byte IV structure, where only 6 bytes are transmitted in each packet. The 4-byte counter value is not transmitted and is used to add variation to the encryption process and reduce the risk of IV reuse.

The length of the MAC is directly related to the security of the MAC. Conventional security protocols uses MAC length sizes of 8, 10, 12, 16 and 20 bytes [4]. However, for a WSN, a 4-byte MAC is sufficient. A 4-byte MAC implies that an adversary has a 1 and 2^{32} chance of blind forging. In other words, an adversary needs to repeatedly send packets to an authorized receiver about 2^{31} times to achieve the correct MAC. Typically, sending 2^{31} packets to a receiver with higher bandwidth is trivial, but for a WSN, the application bandwidth is much smaller. As an example, the Mica2 sensor node that features a low-powered radio from Chipcon, can transmit at a data rate of 19.2kbps. At this slow speed, an adversary can only transfer 40 forgery attempts per second. Thus, it would take about 20 months for the adversary to transfer 2^{31} forgeries. Furthermore, the Mica2 sensor node operating at full power can only run for two weeks before it uses all of its battery resources; therefore exceeding the battery life by more than 40 times.

Similar to most security protocols, the LLSP security protocol increases the computational and energy cost for each packet transmission. There are two major contributions to these costs: the extra computation time and energy needed for cryptography, and the larger packet size due to the security

overhead. Fast symmetric cryptosystems such as AES-CBC, ensure only a modest increase in process and RAM. As shown in Figure 3, to add authentication and encryption to a message packet only requires 10 bytes of security overhead. However, even for non-secure operations, the destination address, active message type and data length fields are necessary and cost 4 bytes of communication overhead. Furthermore, a checksum is generally performed on message packets to detect transmission errors. A simple checksum such as CRC-16, requires a minimum of 2 bytes of communication overhead. In order to do this, the source address should be specified in the message packet which contributes to at least another 1 byte; thus a total of 7 bytes of communication overhead is necessary for non-secure operations. The LLSP security protocol only requires an additional 3 bytes of overhead compared to typical non-secure operation. The 3-byte increase is very modest considering the additional services (message authentication, replay protection, confidentiality, and error checking) that is provided with LLSP security protocol.

B. Performance Analysis

1) Implementation: The LLSP and TinySec security protocols were both implemented with the TinyOS operating system and written in the nesC programming language. The energy consumption and throughput of the LLSP and TinySec security protocols were both measured with the PowerTOSSIM simulator environment. The focus of this paper is on the performance of the security overhead, thus the energy consumption is based solely on the transmission of the security overhead (zero bytes of data payload) on an error-free radio channel.

The performance of the throughput is based on a data payload of 8 bytes per message and is simulated on two different channels: an error-free radio channel and a bit-error lossy radio channel. The error-free radio channel measures the throughput in an ideal environment, where every bit transmitted is received without error. In contrast, the lossy model measures the throughput in a practical environment, where every bit transmitted is not received perfectly. The lossy radio model places each node in a directed graph with TOSSIM's LossyBuilder tool [14].

2) Energy costs: To analytically estimate the cost of the cryptographic service, we first calculate the effect of packet lengths between the LLSP and TinySec security protocols. Recall that TinySec security overhead is 2 bytes larger than the LLSP security overhead. Longer security overhead affect the sensor network in several ways: first, it reduces bandwidth; second, it increases latency because of the fairly slow communication channels; third, it increases the energy consumption because the communication radio must be turned on for a longer period when transmitting longer overhead. We first calculate and compare the expected latency due to the LLSP and TinySec security overheads. Table I shows the extra time needed to transmit a packet using TinySec. We expect the LLSP to reduce the latency by approximately 3%. Note that sending a packet involves more than just sending the associated header and data payload. Media access control information such as start symbol and synchronization bytes are also transmitted. This will cause a discrepancy between the theoretical and simulated energy overhead cost for LLSP security protocol.

	Application Data (bytes)	Packet Overhead (bytes)	Total Size (bytes)	Time to Transmit (ms)	Latency Reduction
TinySec	24	44	68	28.3	-
LLSP	24	42	66	27.5	2.95%

TABLE I
TABLE LISTING THE EXPECTED LATENCY REDUCTION DUE TO DECREASE IN SECURITY OVERHEAD

	Theoretical (mJ)	Simulated (mJ)
TinySec Protocol	0.322	0.5283
Link Layer Protocol	0.269	0.4510
Improvement	16.67%	14.63%

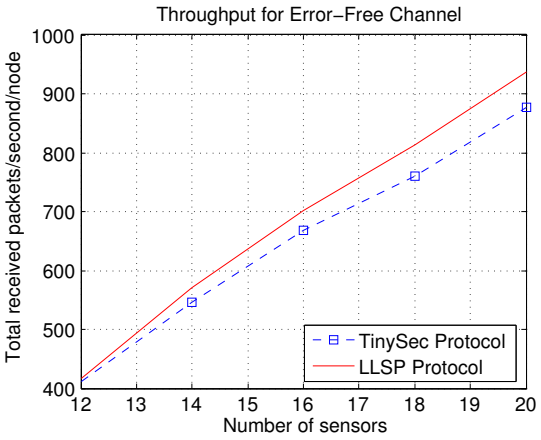
TABLE II
ENERGY CONSUMPTION RESULTS OF TRANSMITTING SECURITY OVERHEAD.

To determine the theoretical energy overhead cost for the LLSP and TinySec security protocols, the power is calculated using Equation 2

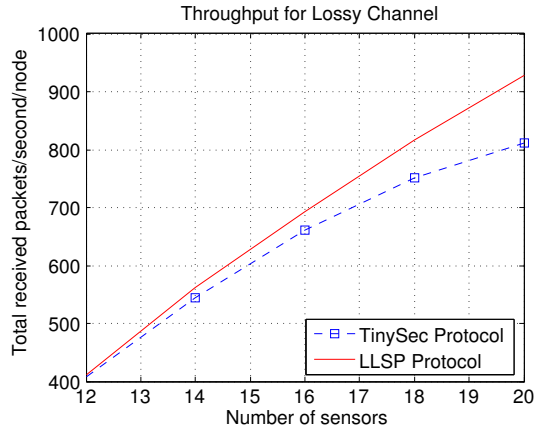
$$P = I * V, \tag{2}$$

where P is the power, I is the transmission current and V is the voltage. For all cases, the transmission current used is 21.48mA and the voltage is 3v. The energy is computed by multiplying the power found in Equation 2, by the packet transmission time. The packet transmission time depends on the data rate of the Mica2 sensor node platform, which has a data rate of 19.2 kbps. Table II summaries the theoretical results and shows that the LLSP security protocol reduces the energy cost of the security overhead by approximately 17%. This result is expected since the reduction in bytes of the security is approximately 17% as well.

The simulation results in Table II are based on the PowerTOSSIM simulator environment. The measurement of the energy consumption for each protocol is based on the transmission of a message packet consisting of only the security overhead. The results shown in Table II, sshow only a 15% improvement between the two protocols. The discrepancies between the theoretical and simulated results are due to the hidden additional bytes of data transmitted during the simulation calculations. During simulation, the sender node always transmits start symbol bytes and pulse strength bytes. These additional bytes cause for only a 15% reduction in energy cost for the security overhead of the LLSP security protocol.



(a) Throughput for Error-Free Channel.



(b) Throughput for Lossy Channel.

Fig. 4. Throughput, plotted as a function of the number of sensors.

3) Throughput: To measure the maximum throughput of the TinySec and LLSP security protocols, the total number of successful received packet in the network were calculated for a 30 second time period. In this experiment, the network was configured such that each sensor node in the network simultaneously transmit packets. Since the number of senders affects the channel utilization, the number of senors in the network is varied. This allows a complete characterization of the throughput at different regimes. The results of the error-free channel and lossy channel are in Figure 4.

First we investigate the throughput performance of the error-free channel. As shown in Figure 4(a), the throughput of the LLSP security protocol gradually begins to outperform the TinySec security protocol as the number of sensors in the network increase. With 20 sensors in the network, the LLSP security protocol increased the throughput by approximately 6%. Recall that in an error-free channel both protocols receive all packets without error. As a result, a small number of packets are required to be retransmitted due to the CSMA/CA protocol. However, the performance of the throughput of a lossy channel shown in Figure 4(b), illustrates a greater improvement in throughput. As the number of sensors increases, the throughput of the LLSP security protocol outperforms the TinySec security protocol by 13%. The increase in the throughput is due to the fact that a lossy channel models bit errors, therefore more packets require retransmission.

VI. Conclusion

The proposed LLSP protocol is an energy-efficient and secure link-layer protocol for WSNs. In comparison with the TinySec protocol, the LLSP security protocol reduces the energy consumption per message packet by 15%. This is achieved by disregarding the 2-byte counter in the communication overhead and maintaining a synchronous counter for each sender and receiver pair with a feedback shift register. In addition to the reduction in energy consumption, this design also increases the throughput for both the error-free channel and the lossy channel. Overall, the LLSP security protocol is a simple, but secure protocol that is easily integrated into the existing applications with minimal application overhead.

References

[1] A. Perrig, R. Szewczyk, V, Wen, D. Culler, and J.D. Tygar, "SPINS: Security Protocols for Sensors Networks," *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, 2001.

[2] J. Ren, T. Li and D. Aslam, "A Power Efficient Link Layer Protocol (LLSP) for Wireless Sensor Network," *Military Communication Conference*, 2005, pp 1002-1007.

[3] P. Levis, N. Lee, M. Welsh and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp 126-137.

[4] R. Venugopalan, P. Ganesan, P. Peddabachagari, A. Dean, F. Mueller and M. Sichitiui, "Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis," *2003 International Conference on Compliers, Architectures and Synthesis for Embedded Systems*, 2003, pp 188-197.

[5] V. Scnayder, M. Hempstead, B. Chen, G. Allen and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp 188-200

[6] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems* , 2004, pp 162-175.

[7] D. Gay, P. Levis R. Behren M. Welsh, E. Brewer and D. Culler, "The nesC Language: A Holistic Approach to Network Embedded Systems," *Programming Language Design and Implementation (PLDI)*, 2003.

[8] J. Hill, R. Szewczyk, S. Hollar, A. Woo, D. Culler and K. Pister, "System Architecture Directions for Networked Sensors," *Proceedings of ACM ASPLOS IX*, 2000.

[9] C. Chong, and S.P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of IEEE*, 2003, pp 1247-1256.

[10] A. Bharathidasas, and V. Anand, "Sensor Networks: An Overview," *Technical report, Dept. of Computer Science, University of California at Davis*, 2002.

[11] X. Luo, K. Zheng, Y. Pan, and Z. Wu, "Encryption Algorithms Comparisons for Wireless Networked Sensors," *IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp 1142-1146.

[12] IEEE standard 802.11i, "Draft supplement to standard for telecommunications and information exchange between systems LAN/MAN specfific requirements-Part 11: Wireless Medium Access Control (MAC) and physical layer specifications: Specifications for Enhanced Security," 2002.

[13] B. Preneel, and P. van Oorschot, "On the Security of Iterated Message Authentication Codes," *IEEE Transactions on Information Theory*, 1999, pp 188-199.

[14] P. Levis, and N. Lee, "TOSSIM: A Simulator for TinyOS Network," *http://www.cs.berkeley.edu/~pal/pubs/nido.pdf*.