An Evaluation of Security Protocols on Wireless Sensor Network

Abu Shohel Ahmed
Helsinki University of Technology
aahmed@cc.hut.fi

Abstract

In recent years, Wireless Sensor Network (WSN) is employed in many application areas such as environmental monitoring, and battle field strategy planning. For many applications of WSN, security is an important requirement. However, security solutions in WSN differ from traditional networks due to resource limitation and computational constraints. This paper analyzes & compares five popular WSN protocols: Tinysec, LLSP, SPINS, LiSP, and LEDS with respect to security requirements, attack scenarios, and performance. The paper also presents application scenarios for general guidance.

KEYWORDS: Wireless Sensor Network, Security

1 Introduction

"A wireless sensor network (WSN) is a wireless network which is formed by distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions such as temperature and sound" [1]. WSN consists of independent small-size Sensor nodes. Each sensor node collects data and sends the information to different destinations. However, unlike wired network with large bandwidth capacity and processing power, WSN has some unique features [4, 12]. First, Sensor devices are limited in energy, computation, and communication capabilities. Second, Sensor nodes are usually deployed in unattended environment which makes it vulnerable to physical attack. Third, WSN is used to monitor physical environments. Fourth, unattended nature and wireless media increases the likelihood of various attacks[19]. All these features make a very demanding environment to provide security in WSN. In WSN, Public key cryptography is expensive in many cases, while symmetric key cryptography needs to be used with cautions. Communication cost is very high: each byte transmitted consumes around 800-1000 instructions [13], thus message overhead should be dealt with cautions.

Major parameters [5] for WSN security includes Key management, providing secrecy and authentication, ensure privacy, robustness against communication denial of service attack, secure routing, energy efficiency, and resilience to node capture. In addition, applications may require secure group management, intrusion detection, protection against traffic analysis, and secure data aggregation functionality. Although all these parameters are required in some scenarios, this paper uses a subset of parameters to evaluate and analyze the selected security protocols. The evaluation parameters are selected based on importance and likelihood of attacks. The following parameters are used in the paper to evaluate the security protocols: key management, availability, confidentiality, Authenticity, Integrity, performance overhead, and protection against common attacks.

WSN is used in many applications with varying security requirements. For example, environmental monitoring applications require less stringent security. In-network processing is more important in environmental monitoring applications to reduce the network contention. On the other hand, battlefield monitoring applications require high level of security. Therefore, security protocols should be selected based on applications characteristics, security demand and nodes processing power.

This paper analyzes five popular WSN security protocols: TinySec[11], LLSP[17], SPINS[6], LiSP[20] and LEDS[15] and presents application scenarios to meet specific security requirements. The paper also presents an application characteristics table which helps to select the appropriate security protocols. Rest of the paper is organized as follows. Section 2 describes the general security requirements of WSN. Section 3 provides the major features of selected security protocols. Section 4 analyzes and compares the selected security protocols. Section 5 provides discussion on example scenarios. Finally, Section 6 concludes the paper.

2 Security Parameters of WSN

A wireless sensor network is a special network with many constraints compared to a traditional computer network [5], which makes difficult to implement the existing security approaches of wired network. This section describes general security requirements of WSN, attack pattern and performance criteria to evaluate WSN. There have been numerous works on identifying general security requirements of WSN. However, we find the criteria in [14] as a complete set for evaluation.

2.1 Special security considerations for WSN

Resource Consumption: WSN has storage, memory and power limitations. An effective security mechanism should have limited size for the code and algorithm. For example, one common sensor type (TelosB) has a 16-bit, 8 MHz RISC CPU with only 10K RAM, 48K program memory, and 1024K flash storage [17]. In addition, when implementing a cryptographic protocol within a sensor the energy impact of security code must be considered. Energy consumption usually derives from two areas: computational costs and com-

munication costs. Computational cost relates to the cost incurred by calculation of hash functions and primitives while communication cost derives from additional byte transfer among sensor nodes. Usually communication cost is much higher than computational cost.

Resistance against unreliable transfer, conflicts and latency: The node communication is usually connectionless which can cause loss of information in critical security packets. In addition, synchronization issues can be critical as latency is a common phenomenon in WSN.

Resistance against physical attacks: The sensor node may be deployed in an open environment which is favorable to adversaries. In addition, the node may be managed remotely which makes it difficult to detect physical tampering. Therefore, protocols should have built in mechanism to limit the damage in the event of node capture and other physical attacks.

2.2 Attacks on WSN

WSN is prone to various types of attacks due to the wireless nature. Mathew [18] defines possible attack types against sensor networks. The most common types of attack includes message injection or alteration, replay attack, node capture or tampering, Denial of Service (DOS), cipher-text collusion, and traffic analysis.

Denial of service: A standard attack on WSN is to jam the nodes. Jamming can be done by interference of the radio signals. Another possible way is to send continuous messages without following the link layer protocol rules. A compromised node can send continuous messages to overflow the network and to deplete the life time of other sensor nodes.

Replay attack: Attack using the same message being replayed. Usually data freshness guarantees protection against replay attack.

Physical attack/node capture: WSN usually operate in hostile outdoor environments which is prone to physical attacks. For example, attacker can extract cryptographic secrets, tamper with associated circuitry, modify programming with the sensors or replace them with malicious sensors under the control of attackers [21]. Recent work shows standard sensor nodes, such as MICA2 motes, can be compromised in less than one minute [9].

Node Replication Attacks: A node replication attack is quite simple: an attacker seeks to add a node to an existing sensor network by copying (replicating) the node ID of an existing sensor node [7]. A replicated node can severely disrupt the network performance such as corruption of packets, or mis-routing.

Injection Attack: Malicious node injects false message in the network. The false message could lead to wrong decision for the whole network. Usually, MAC is used for message authentication and integrity to protect against injection attack.

Intrusion Detection: Cryptography cannot protect against malicious nodes which are already compromised. Intrusion detection system (IDS) can detect the malicious behavior within the network. However, IDS should be deployed carefully in WSN as it consumes resources.

2.3 Security Requirements

Message Confidentiality: Message confidentiality is used to keep information secret from unauthorized parties. Normally, confidentiality is achieved through encryption. A good encryption technique prevents an adversary from recovering an encrypted message and prevents leaking of partial information about the data. Another major concern is optimal size of Initiation vector (IV). IV is used to provide randomness in the encryption process. Long IV provides good encryption at the expense of computational cost.

Message Integrity: A malicious node may add some fragments or manipulate the data within a packet. Message integrity ensures message is not altered in transit. Message integrity is usually provided by Message Authentication Code (MAC). Computing a MAC requires both sender and receiver share a secret key. The sender computes the MAC over the packet using the secret key and the receiver re-computes the packet.

Message Authentication: Message authentication ensures a receiver is capable of identifying the authenticity of the message. Since an adversary can easily inject a message, the receiver needs to ensure that data used in decision making process comes from authentic sources. In two party communications, message authentication can be achieved through Message Authentication code (MAC).

Freshness: Freshness guarantees that the data is recent and no old messages have been replayed. This is especially important for sensor network with shared symmetric key operation. Usually packets include a nonce or counter value to ensure data freshness.

Availability: The goal of security protocol is to consume less processing and communication power which ensure more availability. Besides, node compromise and DOS attack is usually common in large WSN, which results in data unavailability. Therefore, WSN should have mechanisms to tolerate the interference of malicious nodes. Techniques such as in-network processing, en-route filtering can be used to minimize the impact of unavailability.

Self-Organization: Self-Organization is an important requirement for WSN as many WSNs have no fixed infrastructure available for the purpose of network management. For many cases, the dynamics of the whole network inhibits the idea of pre-installation of a shared key between the base station and all sensors [16].

Secure Localization: Location information of a sensor node is often important when point identification is necessary. Besides, location information can prevent large scale attacks.

3 Features of TinySec, LLSP, SPINS, LiSP, and LEDS

This section introduces five selected security protocols of WSN. The section also describes major features of these protocols.

3.1 TinySec

TinySec is a link layer security protocols for WSN[11]. Link layer security provides an effective way to support passive communication (in network processing) among local nodes to remove overlapping communication with the base station. TinySec itself is very lightweight security implementation which creates minimal additional overhead to the existing TinyOS system. In addition, TinySec is fully compatible with higher level security protocols. It is fully transparent to upper layer security protocols. TinySec is based on software only cryptography methods. It supports two different security options: authenticated encryption (TinySecAE) and authentication only (TinySec-Auth) [11].

3.2 Energy Efficient Link-Layer security Protocol (LLSP)

LLSP is a link layer security protocol which ensures message authentication, access control, message confidentiality, and replay protection. It is based on the idea of TinySec. However, it uses different packet format and crypto structure. LLSP supports early rejection capability. It has also low performance overhead. However, it has low scalability as maintaining large networks is difficult with in node counter.

3.3 SPINS

SPINS is based on two secure building blocks: SNEP and μTESLA. SNEP provides data confidentiality, data authentication between two parties, and data freshness. μTESLA provides authenticated broadcast. The protocols achieves limited storage barrier by reuse of code for all crypto primitives (i.e., encryption, message authentication code, hash random number generator). In addition, to reduce the communication overhead, it shares the common state between communication parties. SNEP achieves semantic security by incorporating counter in both ends of the communication parties. However, to reduce the data transmission rate, the counter is not incorporated with the message which means a node in a large network needs to store a counter for each communication party. SNEP only supports Node-to-Base or Base-to-Node communication.

μTESLA provides authenticated broadcast. Unlike traditional authenticated broadcast which requires asymmetric keys to authenticate the initial packets, μTESLA provides security based on symmetric key approach with delayed disclosure of symmetric keys. This approach has difficulties with synchronization for a network with many nodes.

3.4 LiSP: A lightweight Security Protocol for Wireless Sensor Networks

LiSP [20] is a lightweight security mechanism, which is based on efficient rekeying technique. LiSP can be used for key management of large as well as small networks. The main features of LiSP includes efficient key broadcast without retransmission/ACK, authentication of key disclosure without incurring additional cost, ability to detect and recover lost keys, key refreshment without disrupting ongoing data encryption/decryption [20]. LiSP uses novel rekeying mechanism to periodically renew the shared key to solve the key stream reuse problem and maximize scalability/energy efficiency. It uses stream cipher for its cheap and fast processing, and use inexpensive crypto algorithms for key renewal operation. LiSP is very flexible: it requires very loose time synchronization which means it can operate in less reliable broadcast media. LiSP decomposes the entire network into clusters/sensing groups. Each group selects a node as group head (GH). One of the GH in the group acts as KS which controls the security of the group. In addition, LiSP uses intrusion detection system (IDS) to find malicious activities within the network.

3.5 LEDS

LEDS provides location aware end-to-end security. LEDS also provides end-to-end authentication and en-route filtering. It provides location aware key management. LEDS can be used in small as well as large networks. However, number of keys increases with cell size. In addition, LEDS does not support dynamic topology. LEDS divides the network in cell regions. If an event happens within a region, the event should be sensed by *T* nodes. First, each participating nodes agree on the report *M*. *T* nodes encrypt the event using the cell key. After that, each node computes a unique share C_u of *C* through predefined polynomial function LSSS [3]. C_u is an endorsement of the node *u* and can only be verifiable by the sink. All C_u are broadcast within cell and a leader is selected in the cell. The leader computes two MACs over all the *T* share of *C*, i.e, C_{share} which is verifiable by intermediate nodes. These two MAC is calculated using authentication keys shared among two intermediate nodes of upstream report-auth cells of *u*. Then C_{share} is broadcast to the report forward path nodes which contains *T+1* different MACs. Sending of duplicate reports are prevented using techniques[13]. Intermediate nodes check the validity of the message by using MACs. If the report passes the validation test, the message is again broadcast to other intermediate nodes. The sink can verify the report using the *t+1* valid MACs and checking the endorsement of *T* nodes[15]. As long as there are no more than *T-t* invalid shares, sink can always recover the report.

3.6 Major features

Table 1 lists the major features of selected security protocols.

4 Analysis of security in TinySec, LLSP, SPINS, LiSP, and LEDS

This section analyzes the security protocols based on the evaluation criteria defined in section 2.

4.1 Key management / Self re-organization

Key management ensures self re-organization of a network. If network parameters change or a node is compromised, key

Protocol	Type	Key management	Location aware	In-Network process	Scalable
TinySec	Hop-to-hop	No	No	Yes	Partial
LLSP	Hop-to-hop	No	No	Yes	low
SPINS	Node-to-Base	Yes	No	No	low
LiSP	Node-to-GH	Yes	Partial	Partial	Partial
LEDS	End-to-End	Yes	Partial	Partial	Partial

Table 1: Major Features

management can be used to re-organize the network to a safe state.

TinySec and LLSP do not include key management. However, TinySec is not limited to any particular keying mechanism [11] and it relies on keying mechanism from other protocols. TinySec can be used with single network wide shared key. However, this keying mechanism is not suitable to protect against node capture attack. If an adversary compromises a single node, she can injects and eavesdrop the message. To protect against this possible threat, we can use a shared key for each particular communication. However, this technique impedes the in-network processing capabilities.

SPINS includes key exchange protocol. In SNEP, communicating parties share a master secret key [6] which is preloaded during installation process. SNEP derives the encryption key, K_{AB}= $FX_{AS}(1)$ and MAC keys, $K1_{AB}$= $FX_{AS}(2)$ from shared master secret key using pseudo-random function. If one of these temporary keys are revealed during transmission process, the next key can be derived from the master secret. SNEP assumes the communication pattern is predominantly based on node-to-base station or vice versa, thus node-to-node keys are not required. The key distribution nature of SNEP is quite naive. First, it only supports node-to-base station security. Second, a captured node can easily send false message to the base station. Third, the design severely limits in-network processing capability of WSN for false messages.

μTESLA, provides authenticated broadcast using delayed disclosure of symmetric keys which introduces asymmetry in key transfer. μTESLA requires the base station and nodes are loosely time synchronized and each node knows the upper bound of synchronization error [6]. Based on loose time synchronization, authentication keys are broadcast after a time epoch to all receivers. When a node receives a key, it can verify the correctness of the key using one way hash function. In the initial phase, sender first generates one way key chain using one way function which provides forward key checking mechanism. When a new receiver joins the network, receiver receives authenticated keys, time synchronization information, and key disclosure information from the sender. These information are used for future sender-receiver communication. When a broadcast data packet arrived in the receiver, the data is stored in the memory till the keys are disclosed. When the key is disclosed, the node checks the authenticity of the keys. If the key is authenticated, receiver verifies the MAC of the stored message using the key. One precondition is to have a initial commitment key to check the future MAC keys broadcast from the sender to receiver. μTESLA provides many advantages. First, it achieves authenticated broadcast without digital signature. Second, disclosure of keys after a time interval saves energy. However, this approach suffers with increasing sender in a network. As the number of sender increases, receiver need memory space for each sender to buffer the data. In addition, μTESLA approach is not efficient to update keys in a large network.

LiSP uses an efficient rekeying mechanism. LiSP key broadcast does not require retransmission/ACK. The keys can be implicitly authenticated using the one way hash function which supports recovery of lost keys. LiSP decomposes the entire network into clusters/sensing groups and each group selects a node as group head (GH). One of the GH in a group acts as a KS which controls the security of the group. All KS in a network forms a internal network among themselves for keying and intergroup communication. LiSP uses two types of keys: 1) Temporal key (TK) for encryption/decryption 2) a sensor specific master key(MK), used by KS to uni-cast TK to the individual sensor. The KS executes the entity authentication when a new sensor joins the group. After successful authentication, the new node gets a MK and TK. Efficient and Secure TK distribution is always a concern. LiSP achieves the above requirement by 1) Generate TKs by utilizing crypto function, 2) distribute TK before used 3) Perform TK buffering in all sensors 4) verify the authenticity of the received TK. Unlike SPINS, LiSP - TKs are distributed after a certain interval using broadcast (reliable transmission is not required) which conserves resources regardless of network size. In SPINS, Keys are always distributed before data is encrypted with the key which means only keys are stored at each node. LiSP also supports seamless encryption in changing key environment. Therefore, LiSP can work with very loose time synchronization and does not require reliable broadcast at the link layer. In addition, LiSP is also energy efficient and robust to DOS attack. Besides, LiSP minimizes the impact of TK revelation through group TK management. However, it is vulnerable to node capture attack. Capture of master key reveals all the subsequent keys in a particular region of the network.

LEDS incorporate location aware key management framework where each key is bind with location information. Localization helps to isolate the impact for a compromised key. LISP assumes all sensor nodes are preloaded with two master keys. From these preloaded keys, other keys are derived and the original keys are erased after key generation. This approach helps to prevent master key disclosure using compromised nodes. LEDS uses three types of keys: 1) two unique secret keys shared between node and sink for providing node-to-sink authentication, 2) cell key shared with other nodes for data confidentiality 3) a set of authentication keys shared with the nodes and reporting path nodes to provide cell-to-cell authentication and en-route bogus data filtering.

LEDS uses symmetric key to provide end-to-end encryption and hop-to-hop authentication. However, this scheme generates many keys. For a network cell size of 5, it requires 21 keys. It does not support dynamic topology. In addition, change of cell dynamics is not possible after key generation.

4.2 Security Requirements

The paper compares the strength and weakness of the security protocols from the viewpoint of general security requirements: Encryption, Authentication, Message Integrity, Availability and Secure localization. Table 2 provides comparison of five protocols against major security requirements.

TinySec provides authentication, message integrity and encryption. TinySec uses block encryption techniques using CBC mode. However, to limit the packet size, TinySec implements CBC mode using cipher text stealing [8]. TinySec uses eight byte IV (actually two random bytes, others are already provided information).Two byte IV cannot provide total randomness. Thus CBC mode is used to provide reliable protection against cipher text collusion with possible repetitions of IV. Usually RC5 or Skipjack is used as an encryption algorithm. However, TinySec is independent of any encryption algorithm. TinySec uses four byte MAC to provide authentication and message integrity. TinySec only provides hop-to-hop security or link layer security which can be used as a building block for upper layer security protocols. In hop-to-hop security, a misbehaved intermediary node can jeopardize the security of the whole network.

LLSP also provides the authentication, message integrity and encryption in the Link layer. LLSP uses ten byte IV, of which four bytes are used as counter. Four byte randomness provides stronger IV than TinySec. Strong randomness provides better protection against cipher text collusion attack. LLSP provides encryption using AES-CBC mode and four byte MAC is calculated using CBC-MAC algorithm.

SNEP provides data confidentiality, two party authentication and Message integrity. Encryption is provided using block cipher in CTR mode while randomness of the encryption comes from the counter value shared between the two communication parties. Although CTR mode provides fast execution of encryption, ensuring randomness in IV is a major challenge. Without a random IV, it would be a major target for cipher text collusion attack. In SNEP, randomness of IVs comes from the counter. However, counter synchronization among nodes is difficult in many scenarios. In SNEP, keys are shared between base station and node which means only node-to-base station encryption, authentication and message integrity is possible.

LiSP provides message encryption based on stream cipher. It generates key streams using Key stream (TK, nodeID, IV) and Xor this with plaintext. LiSP ensures that key stream is never reused by blending its node id in the key stream generation, periodic refresh of TKs, and increment of IVs. LISP incorporates the IV in the message packet rather than implicit IV management (i.e,SPIN) which solves the synchronization problem of long message size. To provide authentication and integrity, MAC is used where mac= MAC (keyed | nodeID |IV|P). LiSP ensures limited amount of availability by dividing the whole network into groups. Each group has its own key. If compromise happens in a group, other groups remain unaffected. Efficient periodic rekeying mechanism also ensures the availability of data in case of compromise of TKs for a particular session.

LEDS provides authentication, integrity, confidentiality and secure localization. LEDS includes end-to-end data security, interleaved cell-by-cell en-route filtering, and sink verification capability. Data authentication is managed by endorsement of t(i.e., minimum no. of endorsement to validate report), which ensures that adversary cannot define a data as authentic without capturing less than t nodes. In traditional approach, a single captured node can send authentic information to the sink node. LEDS has improved this authentication level to a threshold value. The authentication check is performed at each stage of the intermediate nodes. This ensures in-network processing of false messages. LEDS ensures availability by increasing the prevention level from compromised nodes. It prevents the report disruption attack[13] and selective forwarding attack [10] by dividing the encrypted report into a number of shares. Using a threshold property of LSSS [3], sink can always recover the report from a subset of t valid shares. One-to-many data forwarding approach removes the vulnerability of single node capture attack and ensures availability of data. LEDS manages confidentiality by end-to-end encryption. Message is encrypted using cell key which is only shared between the cell nodes and sink. Therefore, only the compromising node within a cell can break the message confidentiality.

4.3 Protection against Attacks

The major goal of security protocol is to protect applications from common attacks. For our comparison we have chosen a subset of common attacks: Replay, Injection, Alternation, Node capture, Node replication, and DOS attack. Table 3 Provides an attack protection matrix of five protocols.

TinySec includes a two byte counter to provide randomness in the IV. However, these two bytes are not used for replay protection. TinySec prevents Injection and alternation attack using MAC. However, four byte MAC can only provide limited amount of protection against MAC forgery. On the other hand, LLSP uses four byte counter to protect against replay attack. TinySec and LLSP both provide some resistance against DOS attack using in-network filtering of false messages. However, both of these protocols are vulnerable to cipher-text collusion attacks if IV is reused.

SNEP uses implicit counter in both ends to protect against replay attack. SNEP uses MAC to protect against injection and alternation attack. In SNEP, Counter synchronization attack is possible by sending bogus message to a node to perform counter synchronization. Besides counter synchronization problem, cipher text collusion is possible. SNEP has no protection against node capture or node replication attack. SNEP provides no mechanism to identify a tampered master key.

The major type of attack against LiSP could be modification of Temporary Key(TK) management. However, LiSP is effective against modification attack on TKs. As TKs are generated using one way hash function, any modification of the keys will be rejected in authentication test by the receiver.

Protocols	Authentication	Message Integrity	Encryption	Availability
TinySec	CBC-MAC	CBC-MAC	CBC-RC5/Skipjack	
LLSP	MAC	CBC-MAC	AES-CBC	
SNEP	CBC-MAC	MAC	CTR-RC5	
μTESLA	one way hash function	MAC		
LiSP	MAC, key authentication by one way hash function	MAC	Stream cipher	partial
LEDS	End-to-End & Node-to-End by MAC	End-to-End, MAC	End-to-End, MAC	Partial

Table 2: Security Requirements

Attack	TinySec	LLSP	SNEP	LiSP	LEDS
Replay	No	Yes	Yes	Yes	
Injection	Maybe	Maybe	Partial	Medium	Strong
Alternation	Maybe	Maybe	Partial	Medium	Strong
Node capture				Partial	Medium
DOS	Low	Low		Partial	Medium

Table 3: Attack Protection

In LiSP, KS can easily detect the DOS attack if more than 50% node fails in the authentication tests. In addition, Replay attack is also protected in LiSP using unique IV. LiSP uses intrusion detection System (IDS) to identify possible malicious nodes. Thus a good IDS can prevent node capture and node replication attack. Cipher-text collusion attack is also removed in LiSP using truly random key-stream. However, LiSP is vulnerable to single node attack, if the group head or KS is under attack all other nodes in the group suffers the problem.

Hop by hop security can not protect against injection and alternation attack in the intermediate nodes. LEDS overcomes this problem by using end-to-end security. It also provides protection against node capture attack by removing master key after key setup and use of endorsement system for an event. Removal of master key also ensures protection against node replication attack. LEDS also provides protection against report distortion and selective forwarding attacks. Event sensing nodes send the message to multiple next report auth cells. Therefore, compromise of one intermediate node will not prevent the report being sent to the sink or distort the whole report. To make a successful attack, the adversary needs to capture a certain number of nodes. LEDS uses location aware key setup which provides a reasonable protection against a large scale DOS attack on the whole network.

4.4 Performance Overhead

WSN operates in strictly energy and memory limited environment. One of the major design goals of WSN security protocols is to limit the overhead cost to implement the security. The measurement criteria for performance overhead consists of energy overhead, communication overhead, computational overhead and Memory space. Table 4 provides performance overhead figures of the security protocols.

TinySec increases the packet length by 1 to 5 bytes. Longer packet length requires high bandwidth, increased latency, and energy consumption. In TinySec, the additional packet size increases the latency by 8%[11]. On the other hand, the performance of LLSP protocol is quite good. It only adds 3 bytes of additional overhead per packet which is quite good comparing to the modest security services it provides (i.e., message authentication, replay protection, confidentiality and error checking). This means it has 3% less latency and 17% less energy consumption than TinySec while throughput is increased by 6% in ideal conditions[17].

Key setup operation in SNEP is relatively expensive (4 ms) than encryption (1.1 ms) and MAC (1.28 ms) operation. SNEP requires 29% more energy than usual transmission to achieve the security[6]. μTESLA requires additional memory, as it stores the data until the key is released. Experimental result shows, μTESLA requires 17.8 ms to check the buffered messages[6]. Besides, memory consumption, broadcasting of keys need communication cost. However, periodic disclosures of μTESLA key messages can be combined with routing messages to avoid large transmission cost.

In LiSP, major performance overhead generates from the key setup operation. Let C_H denotes the hash computation cost. Experimental result shows, initial setup of computation cost for the key server in per group is n.C_H. *"Each node computes on average less than three hash functions per TK disclosure, even in severe attacks"* [20]. LiSP also consumes less memory than μTESLA, as it does not store data; only keys are stored. LiSP uses an efficient broadcast scheme which results in transmission cost of only 22%(in number of bytes) of unicast case [20]. However, LiSP uses IDS which incurs additional performance overhead.

LEDS has a large key storage overhead to maintain different keys for node-to-sink and cell-to-cell authentication. For example, a network with cell size 5 requires 21 keys[15]. However, the major advantage is the number of key is independent of network size. LEDS also increases the communication cost as every authentic report contains *T+1* MACs [15]. Besides, the division of encrypted report into a set of unique shares increases the message length. Furthermore, the communication pattern is one-to-many which increases the communication overhead significantly.

Security Protocols	Energy overhead	Communication overhead	Computational overhead	Memory
TinySec	TinySec-AE 10% energy overhead [11]	5 bytes per packet		
LLSP	17% less energy than TinySec [17]	3 bytes overhead per packet		
SPINS	29% overhead than usual transmission[6]	6 bytes per packet		2 K extra memory
LiSP		Better than unicast.	Three hash computation per key disclosure per node	
LEDS		Increases packet size. Packets are sent to multiple paths	Computational cost of calculating MACs and hash.	Key storage increase.

Table 4: Performance Figure

5 Recommendation and Discussion

WSN is used in a variety of environments. The suitability of a security protocol is dependent on the application. In this section, we will consider two application case scenarios: Battle field surveillance and Environmental monitoring.

Battle Field Surveillance: Battle Field Surveillance applications require sensors to be deployed in hostile environments. An adversary can capture a sensor node or introduce his own malicious nodes inside the network. Therefore, security protocols should be resilient against a subset of compromised sensor node attack. In addition, WSN needs to be deployed in hostile environments by random distribution. Therefore, dynamic key management protocol is required for battle field surveillance applications[2]. LEDS can protect against node capture attacks by removing primary master secret from the node after key setup. In addition, each sensing message is endorsed by multiple sensors. This prevents the message injection and compromised node attacks up to a threshold value. Multiple path message propagation also ensures availability in case of intermediate node capture attack. However, LEDS lacks the support of dynamic routing and network topology. Therefore LEDS is not suitable for battle field surveillance. However, networks with predictable network structure in harsh environment can be protected using LEDS. On the other hand, LiSP supports key management in dynamic routing environment. However, Lisp has no resilience against node capture attacks. Therefore LiSP is not appropriate for battle field surveillance applications.

Environmental Monitoring: WSN application for environmental monitoring requires large number of sensor nodes distributed over a large geographic area. The nodes are installed carefully beforehand and a defined network topology can be used. Potential threat for node compromise and injection attack is less common rather data processing in the internal network is more important. Therefore, a security protocols with fixed topological structure and in-network processing capability can be used. TinySec and LLSP both provides in-network processing capability, and link layer security with pre-distributed keys. Therefore, TinySec and LLSP provide reasonable security for environmental monitoring applications. However, SPINS is not suitable for large environmental applications due to IV re-synchronization and temporary key rekeying problem. On the other hand, LEDS incurs large communication overhead which blocks performance of a large environmental monitoring application.

Protocol name	Application Characteristics
TinySec	In-network processing & local broadcast. Resource constraint environment Can be combined with higher level security protocols
LLSP	In-network processing applications Resource constraint environment Small sized network
SPINS	Small sized network μTESLA can be used to setup authenticated routing. Communication pattern is node-to-base or vice versa.
LiSP	WSN with many of nodes Applications where security demand is node to Group head
LEDS	End-to-End secure applications Physical attack protection is important Localization is important.

Table 5: Application Characteristics

Table 5 defines application characteristics of five WSN security protocols. These characteristics can be used to select security protocols for a particular application.

6 Conclusion

This paper analyzes five security protocols of wireless sensor network. Each protocols provide certain levels of security. TinySec and LLSP provides link layer security with low performance overhead. LiSP and SPINS is based on symmetric key distribution protocols which ensures relatively low overhead and reasonable security. LEDS is a location aware

end-to-end security protocol which provides high degree of security in the expense of computational and communication cost. LEDS is effective against end-to-end data confidentiality, encryption, and reasonable level of node capture attacks. The paper provides a comprehensive comparison of these security protocols and derives an application characteristics list. The list can be used as a guidance to select security protocols. As identification of appropriate security requirements and selection of security protocols is a complex task, we expect this analysis will help the application designer to choose the appropriate security protocols.

References

[1] Wireless sensor network. http://en.wikipedia.org/wiki/Wireless_Sensor_Networks.

[2] Wireless sensor networks, part 2: Limitations. http://webhosting.devshed.com/c/a/Web-Hosting-Articles/Wireless-Sensor-Networks-part-2-Limitations/.

[3] A. Shamir. How to Share a Secret. In *Communications of the ACM, 22(11)*, Nov 1979.

[4] A. Wood and J. Stankovic. Denial of Service in Sensor Networks. In *IEEE Computer, 35(10):54*, Octobar 2002.

[5] Adrian Perrig and John Stankovic and and David Wagner. Security in Wireless Sensor Networks. In *Communication of the ACM*, volume 47, page 53, June 2004.

[6] Adrian Perrig and Robert Szewczyk and J.D. Tygar and Victor Wen and David E. Culler. SPINS: security protocols for sensor networks. In *ACM Wireless Networks*, volume 8, 2002.

[7] B. Parno and A. Perrig and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *In Proceedings of IEEE Symposium on Security and Privacy*, May 2005.

[8] Bruce Schneier. Applied Cryptography, Second Edition. In *John Wiley & Sons*, 1996.

[9] C. Hartung and J. Balasalle and R. Han. Node compromise in sensor networks: The need for secure systems. In *Technical report CUCU98804, Department of Computer science, University of Colorado at Boulder*, 2004.

[10] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks:Attacks and Countermeasures. In *Elsevier AdHoc Networks Journal*, 2003.

[11] Chris Karlof and Naveen Sastry and David Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *SenSys,ACM*, November 2004.

[12] D. Carman and P. Kruus and B. Matt . Constraints and approaches for distributed sensor network security. In *NAI Labs Tech. Report #00010*, 2000.

[13] H. Yang and F. Ye and Y. Yuan and S. Lu and W. Arbaugh. Toward Resilient Security in Wireless Sensor Networks. In *In Proc. of ACM MOBIHOC*, 2005.

[14] John Paul Walters and Zhengqiang Liang and Weisong Shi and Vipin Chaudhary. Wireless Sensor Network Security: A Survey, Security in Distributed, Grid, and Pervasive Computing. In *Auerbach Publications,CRC Press*, 2006.

[15] Kui Ren and Wenjing Lou and Yanchao Zhang. LEDS: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks. In *Mobile Computing, IEEE Transactions*, volume Voloume: 7, Issue: 5, page 585, May 2008.

[16] L. Eschenaur and V.D. Gligor. A key management scheme for distributed sensor neotworks. In *In Proceddings of the 9th ACM conference on Computer and communications security, ACM Press*, page 41, 2002.

[17] Leonard E. Lighfoot and Jian Ren and Tongtong Li. An Energy Efficient Link-Layer Security Protocol for Wireless Sensor Networks. In *IEEE EIT*, 2007.

[18] Matthew N. Vella. SURVEY OF WIRELESS SENSOR NETWORK SECURITY.

[19] R. Anderson and M. Kuhn. Tamper resistance-a cautionary note. In *In the Second USENIX workshop on Electronic Commerce proceedings, Oakland, California*, 1996.

[20] Taejoon Park and Kang G. Shin. LiSP: A Lightweight Security Protocol for Wireless Sensor Networks. In *ACM Transaction*, June 2004.

[21] X.Wang and W. Gu, S. Chellappan and Dong Xuan. Search-based physical attacks in sensor networks: Modelling and defense. In *Technical report, Dept. Of Computer Science and Engineering, The Ohio state University*, February 2005.