On the Development of Adaptive and Self-dependent Secure Routing Protocol (ASSP) for Wireless Sensor Networks

Jamal N. Al-Karaki 1, Samer Khasawneh 1, Mohammad Alrousan 2

1 jkaraki@hu.edu.jo, sakhasawneh06@cit.just.edu.jo

The Hashemite University, Zarka, Jordan

2 alrousan@just.edu.jo

Jordan University of Science and Technology, Irbid, Jordan

ABSTRACT

In Wireless Sensor Networks (WSNs), sensors may be deployed in hostile environments or in some security-sensitive applications exposing them to tampering and to many several security issues. As such, developing secure routing protocols for sensor networks is a crucial issue. The heart of security protocols is the key distribution algorithm. Many key distribution protocols were proposed in WSNs, but most of these protocols lack the adaptivity to network conditions. In this paper, we propose a novel hierarchical and secure routing protocol with innovative key distribution for WSNs. The proposed protocol offers a tradeoff between the desired degree of security and energy consumption by controlling two parameters: key length threshold and the length of the new generated key, hence the adaptive feature. We use a uniform attack distribution to model attack frequency for the sake of evaluating the performance of the protocol response. When compared to other protocols (e.g., Trusted BS and a pair wise key pre-distribution), our protocol shows enhanced performance with low overhead.

Keywords

Wireless Sensor Networks, Secure Routing, Key Distribution, Hierarchical routing, Adaptive.

1. INTRODUCTION

Wireless ad hoc networking is very attractive since it provides ubiquitous connectivity without the need for fixed infrastructure. Wireless Sensor Networks (WSNs) is a class of wireless ad hoc networks in which sensor nodes collect, process, and communicate data acquired from the physical environment to an external Base-Station (BS) as shown in Figure 1, hence allowing for monitoring and control of various physical parameters [1].

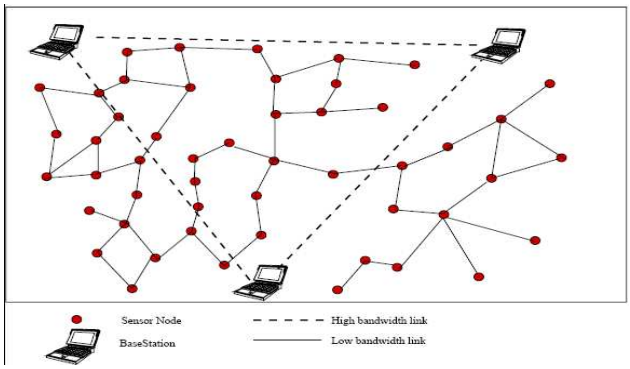


Figure 1: General architecture of wireless sensor networks

As such, WSNs provide low cost solutions to many real life problems. They could be deployed in a field with large density to collect precious information with a relatively low cost.

g

When sensor nodes are deployed in hostile environments or in some security-sensitive military applications especially those related to homeland security, some security measures such as confidentiality and integrity are highly required. These applications require the underlying sensor network to be secure even though it usually has to operate in a harsh and unattended environment. In such environments, WSNs can be subject to eavesdropping, disruption of network services, and manipulation of individual sensors. The security requirements in sensor networks share some commonalities with a typical computer network, but also have their own unique requirements. Hence, the implementation of traditional computer security techniques is invalid for the wireless sensor networks. In literature, large number of routing protocols were designed for WSNs [2] but few of them consider the network security as their main goal. Recently, secure routing protocols for WSNs especially in military and security-sensitive applications received much attention

The design of secure protocols for sensor networks is complicated due to the limited network resources. For example, limited computation power makes complex public key cryptography not suitable for the sensor networks. Furthermore, sensors have limited memory and storage space and cannot maintain a session key with every other sensor in the network. In addition, nodes have energy limitations. So their residual energy must be utilized for an efficient secure routing. Also, typical sensor networks have limited communication bandwidth, so we cannot waste it by flooding the network with control packets to establish/re-establish a secure communication link. Sensor nodes are prone to vulnerability to capture: because they maybe deployed in hostile areas, the requirements for low cost sensor nodes conflict with making them robust to tampering.

In summary, there exist several challenges that hinder providing a key management protocol that is suitable for sensor networks. The severe resource constraints, the ad hoc nature of sensor networks, the unreliable communication medium and the lack of trusted entity like the Certificate Authority (CA) must be addressed in order to achieve secure communication in WSNs.

. Several protocols that tried to balance between the security requirements and network efficiency were proposed. Many of these protocols utilize concepts from wired networks which make them not suitable for wireless networks due to the special and unique characteristics of wireless networks. Some protocols require extensive computations and as such drain sensor nodes'

energy quickly and decrease the network lifetime dramatically. Also public cryptography algorithms require high processing capabilities and will not work on sensors with simple and low processing units.

The rest of the paper is organized as follows. In Section 2, we shed light on the related work. The network model is presented in section 3. **In Section 3, security attacks on the proposed clustered architecture and their countermeasures are presented.** The proposed secure routing protocol is presented in Section 4. A security-energy tradeoff analysis is also presented in this section.. Performance evaluation of the proposed protocol is presented in Section 5. Concluding remarks and some future work is discussed in Section 6.

2. RELATED WORK

The trusted base station is a key distribution scheme that is proposed in [3]. The base station works as key distribution center (KDC) that is responsible for supplying the session keys. Every two nodes that require a secure communication demand a session key from the base station. After they agree on the key they start to communicate securely. Trusted base station does not utilize the available bandwidth efficiently since every secure communication must be proceeded by key establishment phase. Moreover, the BS is overloaded by answering key requests and this will introduce extra packet delay.

Eschenauer and Gligor proposed a random key predistribution scheme [4]. In this scheme a pool of keys is established where every node in the network is assigned a subset of keys from that pool. A pair of nodes can communicate if they share a key between them, so a key discovery phase is necessary for each node to determine which neighbor(s) share the same key with thereby can communicate securely with. In order for two nodes that doesn't share common key to communicate, a secure path is established between them, where any two nodes on that path share a common key.

In Eschenauer and Gligor scheme, if a link between two nodes is attacked then the security of other links that share the same common key with the attacked link are jeopardized. To address this problem multiple key reinforcement scheme is proposed. The link security is enhanced by sending the link common key over multiple independent paths and the final key is the XOR for all the keys received from the independent paths.

SPINS [5] is a protocol that is built using two secure building blocks: SNEP and µTESLA. SNEP offers data confidentiality, authentication, integrity, and freshness. µTESLA offers broadcast data authentication and is also used on regular networks . Since communications in WSN's has three scenarios: node-to base station, base station-to-node, and base station-to-all nodes all traffic pass through the Base station.

Using location pre-deployment knowledge [6] we can manage key generation and distribution. One example is security in grid sensor networks. If the sensors are arranged in grids we know sensor neighbors, so we can generate a key between that sensor and one of its neighbors so that they can communicate securely.

The previous protocols offer a certain degree of security that is independent of the type of danger the network vulnerable to. To the best of our knowledge there is no protocol that offers adaptive degree of security that depends on the attacks the network suffers (i.e., no balance between the degree of security needed and the network efficiency and lifetime). In this paper, we propose a novel security routing protocol that can adapt to network conditions and preserve the nodes energy consumption. A key management module is proposed to securely share/establish the cryptographic keys among sensors, as well as, supporting a secure geographic forwarding protocol. Moreover, a security-energy tradeoff module will reconfigure the sensor node based on the level of residual energy, thus making a tradeoff between security and energy-efficiency is introduced. The motivation behind the proposed protocol is the unpredictable deployment of sensors therefore it is possible that some parts of the network are jeopardized to serious attacks while other parts of the network are relatively safe.

3. THE SYSTEM MODEL

In this section, we briefly introduce the system model along with assumptions used to develop the proposed security protocol. Our model assumes homogeneous WSNs where sensor nodes are assumed resource-limited. In contrast, BSs do not have such resource limitations. When a sensor field contains more than one BS, we assume that each BS is equipped with the same radio receiver/transmitter as sensor nodes, as well as a more powerful RF interface to communicate with other BSs. The network architecture is envisioned as a two layer architecture where nodes play different roles in each layer. The result is a clustered network where the first layer includes data-collection sensor nodes, while the second (virtual) layer consists of control nodes, called ClusterHeads (CHs). Clusters in this two-tier routing system rely on CHs for managing cluster topology, routing information, and so on. Our proposed protocol operates on a similar clustered network and can utilize any hierarchical routing protocol. The protocol assumes that the sensors are deployed in a region and the clusters are formed before running this protocol as in [7] [8]

We also introduce special purpose nodes called Key Generation Nodes (KGNs) that are deployed in the field along with the normal nodes. Each cluster must contain a key generation node (KGN) that is responsible for generating new keys, or re-keying when a node is compromised.

Authorized sensor nodes in the network share a symmetric key that is used to encrypt all communication messages. Recall that the entire network is divided into multiple clusters, where each cluster has its own symmetric key shared among all nodes inside the cluster. The shared key is refreshed after sometime in order to insure high level security. Rekeying means that the shared key is changed/redistributed whenever a sensor joins or leaves the network. For its proper operation, rekeying must be protected by predeploying the keys in each sensor node using tamper-proofing techniques and where only computationally inexpensive cryptography can be employed.

The two fundamental security operations are the encryption and decryption. These operation are computationally intensive operation that require powerful computations unit. However sensor nodes don't have these capabilities. For this reason, the selection of the encryption/decryption algorithm is crucial. After studying a variety of encryption/decryption algorithms we agree that RC5 [9] is the best choice for the encryption\decryption process for several reasons. RC5 is a fast and has low memory and computation requirements compared to the majority of security algorithms. Nevertheless, RC5 is adaptive to the key length which is an essential feature that is desirable in our key distribution techniques.

4. PROTOCOL OPERATION AND PHASES

This section is dedicated to describe the functionality of the proposed protocol. The functionality of the proposed protocol is divided into three phases: establishing session key, obtaining pools of keys and distributing the keys.

4.1.1 Phase 1: Establishing session key

For the sake of clarification and without loss of generality, we will assume that LEACH is the underlying routing protocol. After the advertisement phase, the KGN sends a join message to the cluster head using a special message format so that the cluster head can determine that this node is a KGN rather than a normal node (assume KGNs have a special IDs appended to the join message) . Recall that each cluster contains one and only one KGN in addition to normal nodes and CH. Thus, the proposed protocol runs in rounds. Nodes are periodically and dynamically elected to work as CHs. Depending on its locations, each CH contacts the nearest KGN to obtain a pool of keys. As shown in Fig.2(a), if two different CH's contact KGN1 and KGN2 each KGN replay by different set of keys for example E(Kse([K_{i1}, K_{i2},…, $K_{imax\text{-}pool\text{-}size}$])) and E(Kse([$K_{j1}$, K_{j2},…, $K_{jmax\text{-}pool\text{-}size}$])). Note that this replay message is encrypted using the session key Kse.

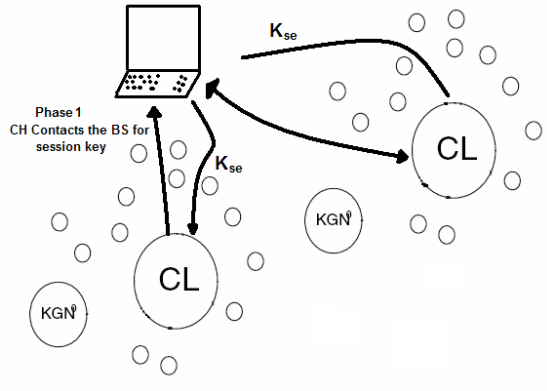


Figure 2 (a): Establishing a session key

4.1.2 Phase 2: Obtaining pools of keys

In this phase, each CH contacts the nearest KGN to get a pool of keys, selects subsets of pool, encrypts it using the session key obtained from the BS and broadcast a subset to each node that belongs to its cluster. The CH and the KGN that belong to one cluster are aware of all cluster keys. A graphical representation for this phase is shown in Fig.2(b).

4.1.3 Phase 3: Distributing the keys to the nodes

The final step before the nodes can start transmitting is to distribute the keys to the nodes. Each cluster head transmits a subset Sc [k_1,k_2,k_3…] of the previously received pool (from the KGN) to every node belongs to its cluster. The subset is Sc chosen randomly from the original pool. Its worth to mention that the subset is encrypted using Kse Fig. 2(c).

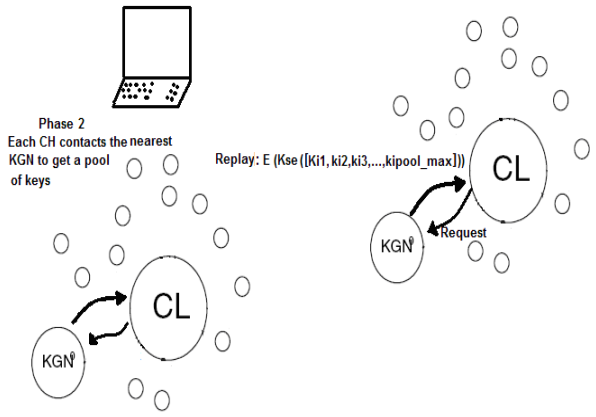


Figure 2 (b): obtain pools of keys.

Nodes that share a secret key can communicate with each other so each node must share a key with the CH because in LEACH each node belong to specific cluster transmit data to in every round (TDMA schedule). Since CH and KGN are aware of the cluster keys all in-cluster nodes can transmit to the CH. In the startup (First round) nodes are given keys with a minimum length (in our simulation we assume keys are 8-bit at startup) and a probability of 1 (max probability). Since LEACH run in rounds after a given time a new round need to be invoked and the cluster heads will be changed. Hence, the previous round keys no longer exists. It is the KGNs responsibility to provide the round keys. Each node wish to transmit data to the CH (according to the TDMA for sure) select random key from its own pool and encrypt these data using RC5 encryption algorithm. When a new round is invoked the same process is done.

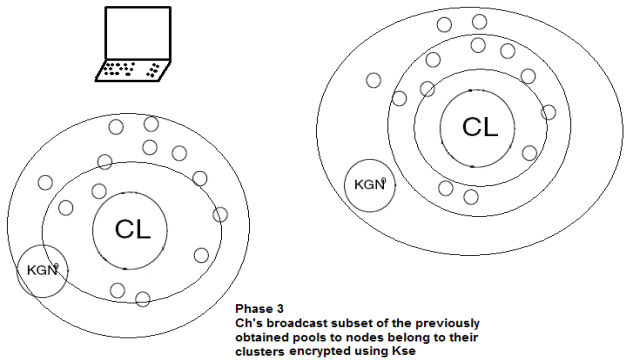


Figure 2 (c): distributing keys to nodes

In the proposed protocol, the KGN use a data structure called Max Priority Queue to store the keys length along with a value corresponding to each length. This data structure is used to keep track of the keys currently in use. If a node is compromised the value corresponding to key length is decreased by a prespecified value. If this value goes beyond a certain threshold the KGN start generating keys with larger length to increase the security level. In this way the proposed protocol offers an adaptive mechanism that balances the trade-off between achieving a secure communication and preserving sensor nodes energy. The flowchart shown in Fig3 shows the protocol steps.

Figure 3. The protocol operation flowchart

Table 2: Simulation symbols

5. PERFORMANCE EVALUATION

In this section we present the performance evaluation results of the proposed protocol, called Adaptive Self-dependent Secure

routing Protocol (ASSP). Although normal sensor are susceptible to a variety of attacks, in our simulation we assume a simple attack model in which a random node is chosen to be compromised and the attack frequency is modeled using uniform distributed. We are not interested how a specific node is compromised rather we study the protocol response to the node attack. Unless mentioned otherwise, we assume S to be equal 0.1. We have developed our simulator using C# language.

We compare our ASSP protocol to two of the most known protocols trusted BS and random pairwise pre distribution. Each single simulation was run more than one time to verify its correctness. Also we study the effect of attack frequency, threshold value and new key length. Through the simulation we used the symbols presented in table 2.

Symbol	Meaning
ASSPx: x is between (0.1-0.9)	ASSP protocol with x threshold
ASSPy: y is a power of 2 (2-32)	ASSP protocol where the new generate key= y times the old one.
ASSPz: z is integer larger than zero	ASSP protocol where the attack occurs every z seconds

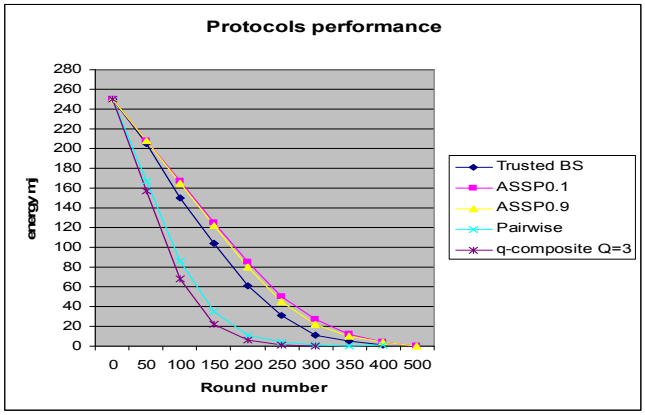
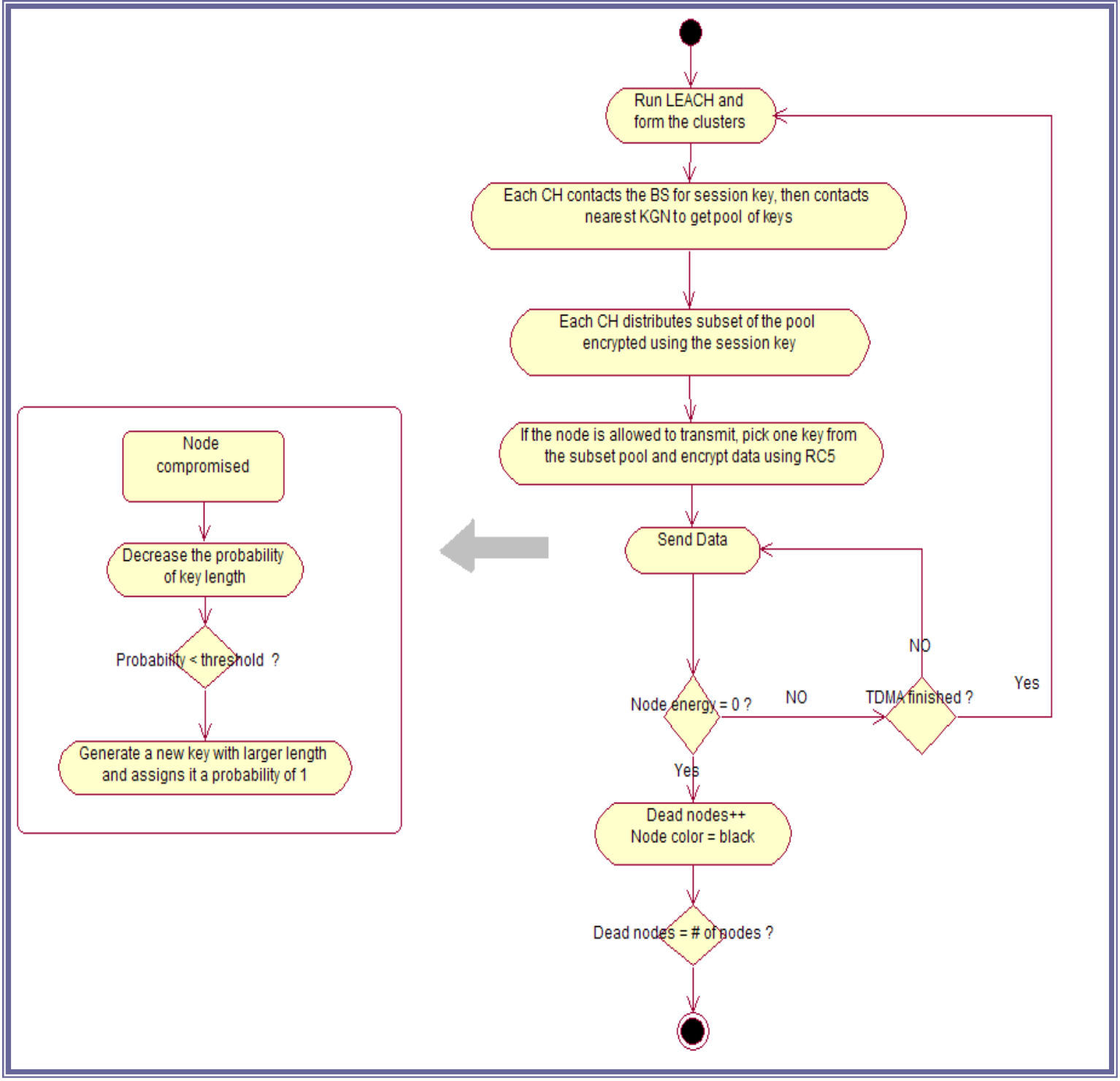


Figure 4: ASSP, trusted BS, and random pairwise performance



First, we compare the performance for 2 versions of ASSP with trusted BS and random pairwise. Fig.4 shows that ASSP0.9 (worst case in terms of threshold) slightly outperforms trusted BS and performs better than random pairwise. There are two versions of ASSP tested here: ASSP0.1 and ASSP0.9 where the value of 0.1 and 0.9 indicates the threshold. Pairwise protocol doesn't perform well in this case. This is explained as follows: random pairwise assumes that two nodes can communicate securely if and only if they share a common key, so that on each round the CH must share a key with every node on its cluster. But if the CH and the transmitting node doesn't share a key, the transmitting node tries to discover a neighbor node that share a key with it and also with the CH, if found it send the data to this neighbor and the neighbor takes care of sending the data to the CH. If the node doesn't find this neighbor, it and the CH must contact the BS to establish the key. Note that ASSP0.9 performs worse than ASSp0.1. This is because in ASSp0.9 after 1 attack the generation of new larger length keys is triggered while ASSp0.1 is the best case because 9 attacks must occur on a specific KGN cluster before the generation of new larger length keys is triggered.

Fig.5 shows the first and last node death for each one of the previously mentioned protocols. It is obvious that ASSP0.1 and ASSP0.9 have a higher network lifetime than trusted BS and random pairwise predistribution.

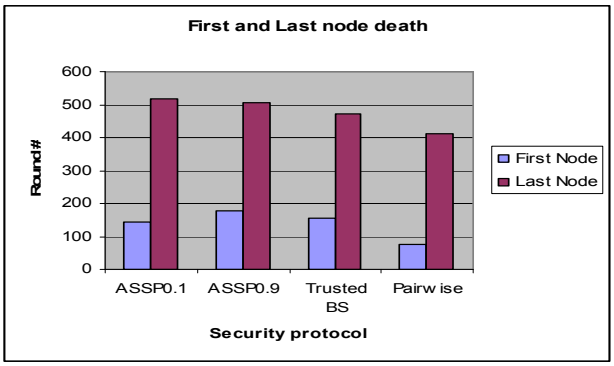


Figure 5: First and least node death for the simulated protocols

For the sake of storage analysis, we assume that each CH in ASSP contacts the nearest KGN to obtain a pool of keys this pool is 5 keys, for each node the CH selects randomly 3 keys and distributes them to each node in the cluster, so at any time all non cluster nodes contain 3 keys. Using Trusted BS any node must share a key with the CH, this key is obtained directly from the BS, so at any time the node stores only one key. Each CH must share a key with every node in the cluster in order to achieve a secure communication on random pairwise predistribution. We performed a test to determine the average number of nodes on each cluster (for example CH1 has 13 nodes in the cluster) because under pairwise protocol each node must share a key with CH in order to communicate securely. The simulation in Fig.6 shows that CH1 has on average 11 keys stored on its memory (this number of keys is less than the number of nodes in the cluster because the node that doesn't share a key with the CH performs shared key discovery to find a neighbor that shares a key with the CH and so no need to share a key with the CH directly). Fig.7 shows the average number of keys each CH stores during the simulation.

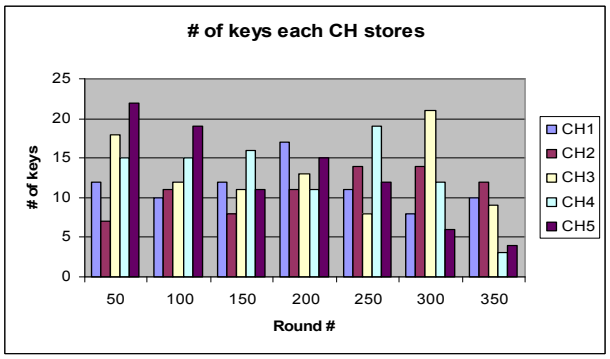


Figure 6: Number of keys stored on each CH for different rounds

From the above discussion its obvious that the random pairwise requires the largest storage requirement while Trusted BS requires the minimum storage requirement with only one key stored in the memory where ASSP requires three keys to be stored in nodes' memories, which is less than the Trusted BS scheme.

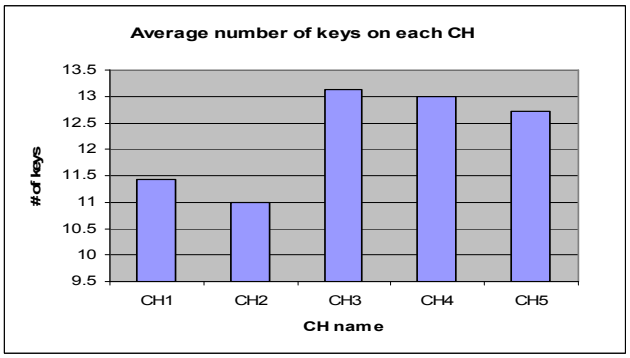


Figure 7: The Average number of keys each CH stores during the simulation

Now we focus on ASSP performance for different threshold values, key lengths and attack frequencies. For this purpose, we vary the values of the controlled parameters, namely, threshold, new key length and the frequency of the attacks. We set the attack frequency to 100 and new key length to 8 times larger than old ones for all thresholds.

A) ASSP with different thresholds

By varying threshold values, we are interested in studying the threshold effect on sensor network energy. We measure the average network power when the threshold is on the worst case (0.9), the best case (0.1) and the average case (0.5). The threshold value (e.g. 0.9) means that after one successful attack the cluster KGN start generating larger length keys while 0.1 threshold means that after 9 successful attacks the cluster KGN start generating larger keys. Hence, the network where threshold value is 0.1 to have higher average energy because the keys are smaller in size and thus the packet size is smaller (i.e. smaller packet size means less transmitting and receiving power dissipation).

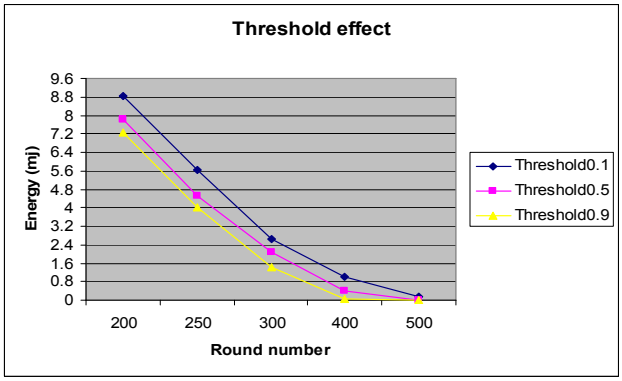


Figure 8: Threshold effect on network power on different LEACH rounds

Fig.8 shows the energy dissipation differences when best, average, and worst case threshold values are used (ASSP0.1, ASSP0.5, and ASSP0.9). We are showing values after round 200 when network stabilizes. It is noted that energy differences between the different thresholds in some cases is small. This can be justified by saying that ASSP0.9 will generate larger keys faster than ASSP0.1 and thus the network exchanges larger messages.

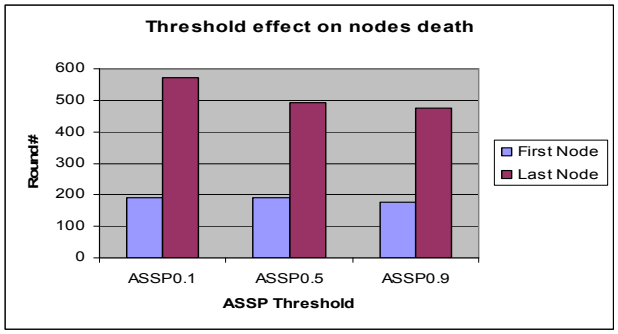


Figure 9: First and last node death for different ASSP thresholds

Next, we focus on network lifetime. From Fig.9, it is obvious that the network that implemented ASSP0.1 has a higher lifetime since the last node lived for 570 rounds where ASSP0.5 and ASSP0.9 has a lower lifetime and the last node dead before round 500.

B) ASSP with different attack frequencies

Recall that our proposed protocol responds to node compromise by generating keys with larger lengths. Since higher length keys means more bits to travel on the network and also means higher power consumption upon sending or receiving messages, the frequency of attacks affect the network performance and lifetime. To clarify this, we vary the frequency in which attacks occur to determine how it affects the network lifetime. It is expected that the network lifetime will decrease as the attack frequency increases since more attacks means larger key lengths and thus more power consumption.

Fig.10 shows that ASSP100s (which means that attacks occur every 100 second) has the highest average power consumption when compared to ASSP10s and ASSP30s because nodes will die earlier due to power exhaustion. Fig.11 shows network lifetime and confirms with the results and conclusions of Figure 10. That is, nodes with higher attack frequencies will die sooner than nodes with less attack frequency leading to shorter network lifetime. In this simulation, we set nodes' threshold to 0.9 and new key length to 8 times larger than old ones for the different attack frequencies.

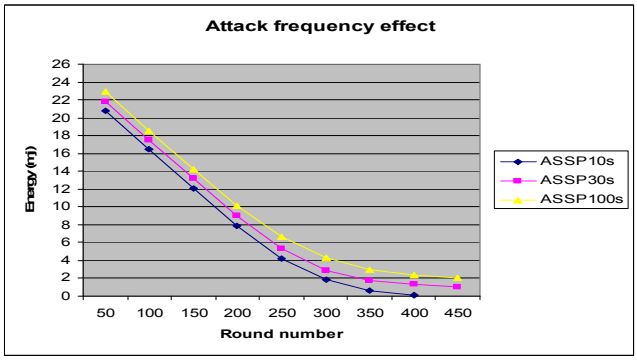


Figure 10: ASSP performance with different attack frequencies

C) ASSP with different new key lengths

When an attack is detected, ASSP responds by generating keys with larger lengths. We now focus on the effect of the newly generated key length on sensor network lifetime. We can achieve higher degrees of security by generate new keys with larger lengths. The choice of the new key length depends on the degree of the security desired and the network lifetime required. The results bellow show different versions of ASSP, ASSP1 (the new key is 2x the old) ASSP8 (the new key is 8x the old) and ASSP32 (the new key is 32x the old one). ASSP1 has the best performance and network lifetime since it generates keys with length equal the double of the old key length (old key means the key that was used by the compromised node). These results are depicted in Fig12 and Fig 13.

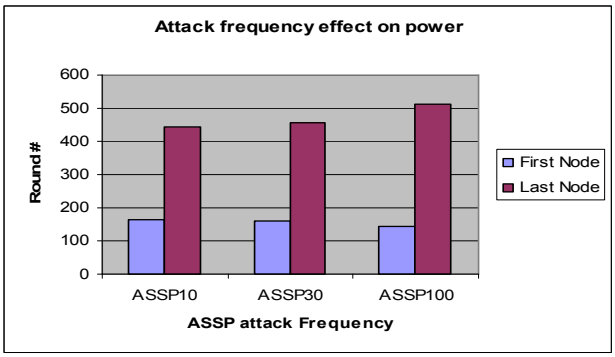


Figure 11: Network lifetime for different attack frequencies

ASSP32 has the lowest network lifetime since it generates keys that are 32 times larger in length then the keys used by the compromised nodes. In this simulation, we set nodes' threshold to 0.9 and attack frequency to 100 for all keys length.

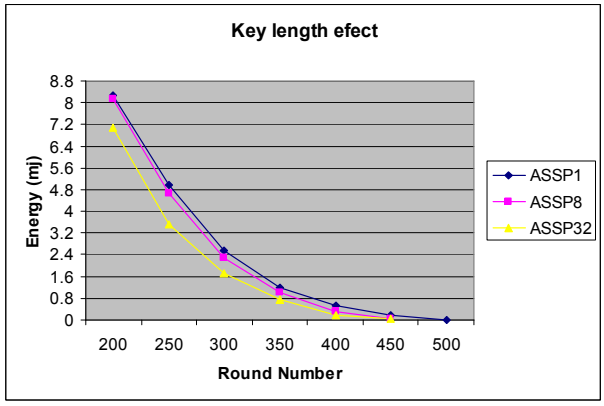


Figure 12: ASSP performance for different generated keys lengths

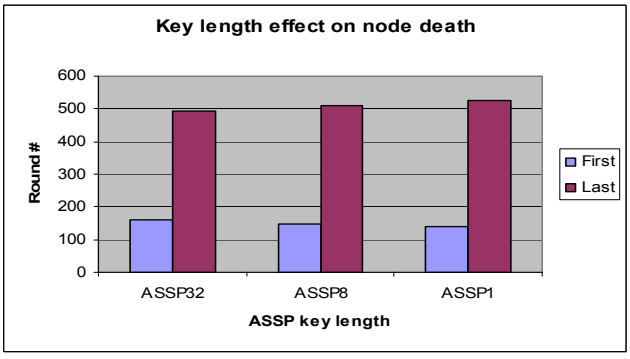


Figure 13. Network lifetime for different generated keys lengths

6. CONCLUSION AND FUTURE WORK

In this paper we presented ASSP as a novel approach to achieve secure communication via introducing a key distribution paradigm in sensor networks. ASSP has the capability to adapt to network condition thereby offer a wide range of tradeoffs between the desired security level and the network lifetime. This is achieved by controlling three values: the threshold (determine when generating new keys with larger length is triggered), the

new key length (controls the newly generated keys length) and the attack frequency (the expected attack frequency the network maybe jeopardized to). Simulation results show that our protocol outperforms the Trusted BS and the random pairwise predistribution in terms of network lifetime. The future work will study and compare the delay of ASSP with the comparable protocols.

REFERENCES

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*: 2002; 102/114.

[2] J. N. Al-Karaki, A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", *IEEE Wireless Communications*, 2/24, PP. 1536-1284, 2004.

[3] S. P. Miller, C. Neuman, J. I. Schiller, and J. H. Saltzer. Kerberos authentication and authorization system. In *Project Athena Technical Plan*, page section E.2.1, 1987.

[4] Laurent Eschenauer and Virgil D. Gligor. A key-anagement scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communication security*, pages 41–47, November 2002.

[5] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D.Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual ACM International Conference on Mobile computing and Networks(MobiCom 2001)*, July 2001.

[6] Dijiang Huang, Manish Mehta, Deep Medhi, Lein Harn, Location aware Key Management Scheme for Wireless Sensor Networks, ACM workshop on Security of ad hoc and sensor networks 2004.

[7] Wendi RA HE, Anantha CH, Hari BA. Energy-Efficient Communication Protocol for Wireless Sensor Networks. IEEE System Sciences 2000; 2:10.

[8] J.N Al-karaki R.Ul-Mustafa, A.E. Kamal.,"Data Agggregation in Wireless Sensor Networks – Exact and Approximate Algorithms," Proc. IEEE Wks. High Perf. Switching and Routing 2004, Phoenix, AZ, Apr. 18-21, 2004.

[9] William Stallings, Cryptography and network security principles and practices, 3rd edition 1999 (book).