

Evaluation of Overheads in Security Mechanisms in Wireless Sensor Networks

Tanveer Zia, Albert Zomaya, and Nedal Ababneh
School of Information Technologies
University of Sydney
Email: {tanzia,zomaya,nedal}@it.usyd.edu.au

Abstract – Wireless sensor networks are being used in many commercial and military applications to collect real time and event driven data. Deployment nature of sensor networks makes them vulnerable to security threats. Due to the resource limitations traditional security measures are not enough to protect sensor nodes. Research in sensor network security domains has produced several security solutions. In this paper we have observed three recently introduced security mechanisms (1) TinySec (2) MiniSec, and (3) TripleKeys. We have studied these security mechanisms in terms of packet overheads and compared the packet transmission time, average latency and energy consumption. Our comparison shows that the packet overheads in TripleKeys are lesser compared to other two schemes. We have then used the 38 bytes packet size of TripleKeys for further analysis and calculated the packet delivery ratio, latency and energy consumption. We have observed that packet delivery ratio decreases when we increase the number of nodes while latency and energy increases.

I. INTRODUCTION

Sensor networks constituting large number of sensor nodes are becoming viable solution to many challenging domestic, commercial and military applications. Sensor networks collect and disseminate data from the fields where ordinary networks are unreachable for various environmental and strategic reasons.



Figure 1. MICA 2

For our evaluation we use UC Berkeley and Crossbow mote MICA 2 [1] as shown in figure 1. Configuration of a typical mote is 8MHz, 8-bit processor, with 128KB of instruction memory, 4KB of RAM, and 512KB of external flash memory, 2.7-3.3v, 2xAA power and the radio runs at 916 MHz as shown in Table I.

TABLE I. MICA2 SPECIFICATIONS

Specifications	BERKELEY'S MICA2
Processor	8Mhz, Atmel ATmega128L
Memory	128KB Program Flash and 4KB RAM
External storage	512KB Serial Flash
Default power	2.7-3.3v 2xAA
Sleep mode	<15 microA
Radio	916Mhz
LEDS	3 led indicators
Size	58 x 38 mm
Xm range	300m
OS	TinyOS

Due to the computation and power limitations sensor networks are more vulnerable to security threats. Security does not come free, adding heavy security measures in terms of computation power, limitation in energy and memory poses significant challenges in designing a light weight security solution against attacks on sensor networks.

In this paper we evaluate three security mechanisms in wireless sensor networks (1) TinySec, (2) MiniSec, and (3) TripleKeys. We study and compare the overheads in these security mechanisms and conclude that TripleKeys has got fewer overheads as compare to other mechanisms and it is potentially a viable security solution to many of the attacks in sensor networks. Our contribution in this paper is a thorough analysis and study of the three security mechanisms supported by the experimental results.

The rest of the paper is organized as follows. Section II summarizes the known attacks in sensor networks. In Section III we provide a brief summary of the three security mechanisms. Section IV presents the comparative analysis of TinySec, MiniSec and Triplekeys. In section V we provide our experimental results to determine delivery ratio,

average latency, and energy consumption in TripleKeys. Finally we conclude our paper in section VI.

II. ATTACKS ON SENSOR NETWORKS

Karlof and Wagner [2] have listed several attacks on sensor networks which are summarized in the followings section:

A. *Selective forwarding*

Selective forwarding is a way to influence the network traffic by believing that all the participating nodes in network are reliable to forward the message. In selective forwarding attack, malicious nodes simply drop certain messages instead of forwarding every message. Once a malicious node cherry picks on the messages, it reduces the latency and deceives the neighboring nodes that they are on a shorter route. Effectiveness of this attack depends on two factors. First the location of the malicious node, the closer it is to the base station the more traffic it will attract. Second is the percentage of messages it drops. When selective forwarder drops more messages and forwards less, it retains its energy level thus remaining powerful to trick the neighboring nodes.

B. *Sinkhole attacks*

In sinkhole attacks, adversary attracts the traffic to a compromised node. The simplest way of creating sinkhole is to place a malicious node where it can attract most of the traffic, possibly closer to the base station or malicious node itself deceiving as a base station. One reason for sinkhole attacks is to make selective forwarding possible to attract the traffic towards a compromised node. The nature of sensor networks where all the traffic flows towards one base station makes this type of attacks more susceptible.

C. *Sybil attacks*

A type of attacks where a node creates multiple illegitimate identities in sensor networks either by fabricating or stealing the identities of legitimate nodes. Sybil attacks can be used against routing algorithms and topology maintenance; it reduces the effectiveness of fault tolerant schemes such as distributed storage and dispersity. Another malicious factor is geographic routing where a Sybil node can appear at more than one place simultaneously.

D. *Wormholes*

In wormhole attacks an adversary positioned closer to the base station can completely disrupt the traffic by tunneling messages over a low latency link. Here an adversary convinces the nodes which are multi hop away that they are closer to the base station. This creates a sinkhole because adversary on the other side of the sinkhole provides a better route to the base station.

E. *Hello flood attacks*

Broadcasting a message with stronger transmission power and pretending that the HELLO message is coming from the base station. Message receiving nodes assume that the HELLO message sending node is the closest one and they try to send all their messages through this node. In this type of

attacks all nodes will be responding to HELLO floods and wasting the energies. The real base station will also be broadcasting the similar messages but only few nodes will have responding to it.

F. *DoS attacks*

Denial of service attacks occur at physical level causing radio jamming, interfering with the network protocol, battery exhaustion etc.

III. SECURITY MECHANISMS

Security in sensor networks has a number of challenges, some of which are: wireless communication among the nodes, lack of pre-existing infrastructure, dynamic topology changes, and resource constraints in terms of memory, energy, and low communication bandwidth. [3]

Primarily security in network systems depends on the key management and the strength of the encryption keys. In general we call this key establishment. Some of the well known key establishment techniques are [4, 5]: trusted-server scheme, self enforcing scheme, and key pre-distribution scheme. The trusted server scheme depends on a trusted server e.g., Kerberos [11]. Since there is no trusted infrastructure in sensor networks, therefore trusted-server scheme is not suitable in this case. The self-enforcing scheme depends on asymmetric cryptography using public keys. However, limited computation resources in sensor nodes make this scheme less desirable. Public key algorithms such as Diffie-Hellman [6] and RSA [7] as pointed out in [5, 8] require high computations resources which tiny sensors do not provide. The key pre-distribution scheme, where key information is embedded in sensor nodes before the nodes are deployed seems more desirable solution for sensor nodes. A simple solution is to store a master secret key in all the nodes and obtain a new pairwise key. In this case capture of one node will compromise the whole network. Storing the master key in tamper resistant sensor nodes increases the cost and energy consumption of sensors.

A. *TinySec*

TinySec [9] is a light weight secure link layer security solution for sensor networks. TinySec achieves low energy consumption and memory usage. However, it compromises the level of security. For example, it uses three type of keying mechanisms: (1) single network-wide shared key (2) per-link shared key, and (3) per-group shared key. Single network-wide shared key is easy to deploy but not robust to node compromise. Per-link shared keys between neighbouring nodes help degradation in the presence of compromised nodes, but needs a key distribution protocol and prohibit passive participation and local broadcast. Group shared keys also help in degradation in the presence of compromise nodes and support the passive participation, however this keying mechanism trades off robustness to node compromise for added functionality.

B. MiniSec

MiniSec [10] is a network layer security mechanism which operates in two modes: MiniSec (U), unicast designed for single-source communication, and MiniSec (B), tailored for multi-source broadcast communication. MiniSec claims low energy utilization on Telos [11] platform. MiniSec uses similar packet format as TinyOS [12] and replaces the 2-byte CRC from TinyOS with a 4-byte tag, as the tag protects the packet from tampering. MiniSec also eliminates the need of 1-byte group ID and uses different cryptographic keys.

C. TripleKeys

Secure TripleKey management scheme [13] is consisting of three keys: two pre-deployed keys in all nodes and one in-network generated cluster key for a cluster to address the hierarchical nature of sensor network.

K_n (network key) – Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. Nodes use this key to encrypt the data and pass onto next hop.

K_s (sensor key) – Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. Base station uses this key to decrypt and process the data and cluster leader uses this key to decrypt the data and send to base station.

K_c (cluster key) – Generated by the cluster leader, and shared by the nodes in that particular cluster. Nodes from a cluster use this key to decrypt the data and forward to the cluster leader.

Packet format in TripleKeys contains following fields:

ID	Keys	TS	Data	MAC
(3)	(3)	(1)	(29)	(4)

This gives 40 bytes of data packet to transmit. In the following section we evaluate the packet formats and overheads in TinySec, MiniSec and TripleKeys.

Secure triple keys provide an effective countermeasure against the attacks discussed in Section II. Secure triple key management scheme requires each node to have mutual authentication with its neighbors and cluster leaders. When a malicious or compromised node intends to impersonate a legitimate node, it does not have the authentication with its neighbors and the cluster leader. For this reasons a malicious node can not claim multiple identities. Therefore the Sybil attack is effectively countermeasure by the proposed routing mechanism. Selective forwarding, wormhole and sinkhole attacks are also defeated because nodes are authenticated by the base station and the cluster leader.

IV. ANALYSIS AND EVALUATION OF TINYSEC, MINISEC AND

TRIPLEKEYS

Table II below shows the performance comparison among TinySec, MiniSec and TripleKeys. This comparison shows that the TripleKeys do not have any additional overheads. Also it overcomes the weaknesses of TinySec. As per our analysis TinySec is confusing because of the three different states of TinySec: (1) no TinySec (CRC), (2) TinySec-Auth and (3) TinySec-AE. Also TinySec assumes a message length of 8 bytes or more, it does not address the smaller messages. TinySec does not provide a secure localisation, secure routing mechanism while TripleKeys address these weaknesses. Although MiniSec provides a good level of security but the overheads are much more than TinySec and TripleKeys.

TinySec-Auth (Authentication only): +8b

Dest	AM	Len	AM	Src	Ctr	Data	MAC
(2)	(1)	(1)	(1)	(2)	(2)	(29)	(4)

TinySec-AE (Authentication and Encryption): +12b

Len	PCF	DSN	DstPAN	Dest	AM	Src	Data	MAC
(1)	(2)	(1)	(2)	(2)	(1)	(2)	(29)	(4)

MiniSec U or B: +15b

ID	Keys	TS	Data	MAC
(3)	(3)	(1)	(29)	(4)

Secure Triple keys: +9

Dest	AM	Len	Data	MAC
(2)	(1)	(1)	(29)	(4)

TABLE II. COMPARISON OF OVERHEADS IN TINYSEC, MINISEC AND TRIPLEKEYS

	Application Data (b)	Packet Overhead (b)	Total Size (b)	Time to transmit (ms)	Increase over TinyOS stack	Latency Overhead	Energy Overhead
TinySec-Auth	29	8	37	26.6	1.5%	1.7%	3%
TinySec-AE	29	12	41	28.8	8%	7.3%	10%
MiniSec U or B	29	15	44	30.7	12.8%	11.5%	12.2%
TripleKeys	29	11	40	28.3	6.3%	5.9%	8.2%

V. EXPERIMENTAL EVALUATION OF PACKET

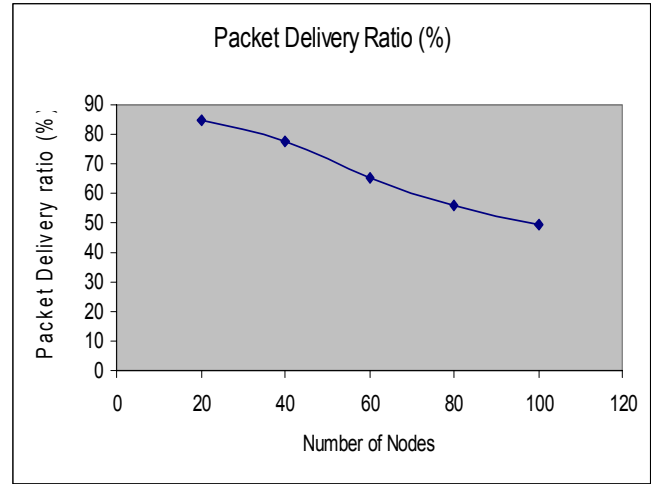
OVERHEADS IN TRIPLEKEYS

We have implemented the 40 bytes TripleKeys packet format in the sensor network simulator TOSSIM [14] to observe the packets in varying size of networks.

In our application scenario, all sensor nodes generate a new data message every 3s. A new data message generated periodically between 50 and 1000s of the simulation lifetime. All experimental results in this section are average of ten runs on different randomly-chosen scenarios; we assumed that the nodes are distributed uniformly.

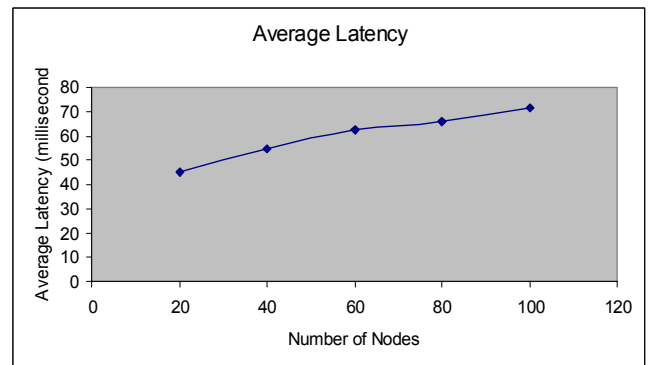
In the experiments, we have observed the delivery ratio, average latency, and energy consumption for the application over the course of the simulation period. We define the delivery rate as the percentage of total number of received data messages at the data sink divided by the total number of sent messages from all sources. Average latency is the average time a message takes to travel from any source to the data sink. The scenario that we have used in our simulation experiments consists of a 300m x 300m square area, covered with wireless sensor nodes. We compute the values of the above metrics in a scenario where the number of nodes varies from 20, 40, 60, 80 and 100 nodes; the radio transmission range of nodes is 200m. Figure 2 shows the observed results with respect to the number of nodes. Packet delivery ratio decreases when we increase the number of nodes while latency and energy increases.

Number of Nodes	Packet Delivery Ratio (%)
20	84.51
40	77.43
60	65.25
80	55.78
100	49.13



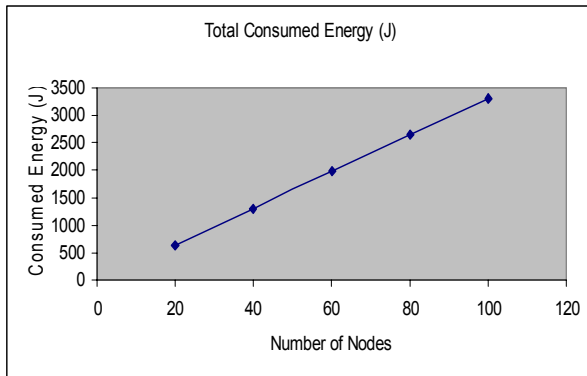
(a) Packet Delivery ratio (%)

Number of Nodes	Average Latency (millisecond)
20	44.87
40	54.54
60	62.42
80	66.13
100	71.45



(b) Average Latency (ms)

Number of Nodes	Consumed Energy (J)
20	638.15
40	1308.19
60	1978.15
80	2648.03
100	3317.73



(c) Energy Consumption (J)

Figure 2. Delivery ratio, average latency, and energy consumption in TripleKeys

As shown in Table II, if we convert the delivery ratio, average latency and energy consumption in per packet transmission, TripleKeys overheads increase 3.1% in latency and 4.7% in energy on TinyOS stack. These overheads are much lesser than TinySec and MiniSec.

VI. CONCLUSION

Wireless sensor networks have promising future to many critical and sensitive applications such as environmental monitoring, military applications, healthcare, surveillance and habitat monitoring etc. Sensor nodes have limited resources in terms of energy and memory. Besides resource limitations sensor networks are left unattended after deployment which poses additional security threats. Security solutions for sensor networks have to be simple with fewer overheads. In this paper we have studied three security mechanisms in sensor networks (1) TinySec (2) MiniSec, and (3) TripleKeys. Increase in overheads over the TinyOS stack is shown in the comparison table. We have provided experimental evaluation on packet delivery ratio, latency and energy consumption in TripleKeys and have observed that TripleKeys overheads are comparatively lesser than other two security mechanisms. The goal of this paper is to provide a comparative study of the security mechanisms in wireless sensor networks and their effect on the performance of the network.

REFERENCES

- [1] MICA2: Wireless Measurement System. Crossbow Technologies Inc. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf [Viewed on 2 May 2007]
- [2] C. Karlof and D. Wagner, *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*, University of California at Berkeley, USA 2003.
- [3] T. Roosta, S. Shieh, and S. Sastry, taxonomy of Security Attacks in Sensor Networks and Countermeasures, *In The First IEEE International Conference on System Integration and Reliability Improvements. Hanoi, Vietnam, 13-15 December 2006*.
- [4] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney, A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks, *ACM CCS 2003*.
- [5] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shiang Chen, and Pramod K. Varshney, A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge, *IEEE InfoCom 2004*.
- [6] W. Diffie and M. E. Hellman, "New directions in cryptography. *IEEE transactions on information theory*, 22:644-654, 1976.
- [7] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120-126, 1978
- [8] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar. SPINS: Security Protocols for Sensor Networks, in *Wireless Networks Journal (WINE)*, September 2002
- [9] C. Karlof, N. Shastry and D. Wagner, "TinySec: A Link layer Security Architecture for Wireless Sensor Networks", *SenSys'04*, November 3-5 2004, Baltimore, Maryland, USA
- [10] M. Luk, G. Mezzour, A. Perrig, and V. Gilgor, "MiniSec: A Secure Sensor Network Communication Architecture", *IPSN'07*, April 25-27, 2007, Cambridge, Massachusetts, USA
- [11] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN/SPOTS*, 2005.
- [12] TinyOS, 2007, viewed on 8 May 2007 <http://www.tinyos.net/>
- [13] T. A. Zia, and A. Y. Zomaya, A Secure Triple-Key Management Scheme for Wireless Sensor Networks, In the proceedings of the *IEEE INFOCOM 2006*, April 23-24, 2006, Barcelona, Spain
- [14] P. Levis and N. Lee, TOSSIM: A Simulator for TinyOS Networks, UC Berkeley, September 2003, Berkeley, CA