Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.471 – 479

REVIEW ARTICLE

Security Techniques for Constructing Wireless Sensor Networks: A Review

Bharati N. Sawant[1], Bhupendra S. Chordia[2]

[1]Master Student Computer Science & Engineering, SSVPS'S B.S.Deore College of Engineering, India

[2]Assistant Professor, Computer Science & Engineering, SSVPS'S B.S.Deore College of Engineering, India

[1] bharatisawant26@gmail.com; [2] chordiabs@yahoo.com

Abstract— Efficient design and implementation of Wireless Sensor Networks remain one of the most challenging researches due to their wide range of applications. As the use of wireless sensor networks (WSN) grow vastly, so it should require effective security mechanisms. These sensor networks may interacts with the sensitive data and operating in hostile unattended environments so its security concerns should be address from the beginning of the system design are getting much preference. Currently there is enormous research is present in the security field of wireless sensor network. Thus, in this review we mainly highlight the major topics in wireless sensor network security, and present security protocols for sensor networks.

Keywords: Sensor network security, secure communication architecture

I. INTRODUCTION

A Wireless Sensor Networks (WSNs) are heterogeneous systems containing large number of small devices called sensor nodes and actuators with general-purpose computing elements. These networks should consist of thousands of low cost, low power and self-organizing nodes which are highly distributed either inside the system or very close to it. There are Various applications of WSN includes ocean and wildlife monitoring, monitoring of manufactured machinery, building safety, earthquake monitoring environmental observation, military applications, manufacturing and logistics, forecast systems, health, home and office application and a variety of intelligent and smart systems. The WSNs topology can be vary from a simple star network to the advanced multi-hop wireless mesh network [1].

In case of wireless sensor network, the communication among the sensor nodes is done using wireless transceivers. The attractive features of the wireless sensor networks promises many researchers to work on various issues related to these types of networks. However, while the routing strategies and wireless sensor network modeling are getting much preference, the security issues are yet to receive extensive focus [1] [2].

A. Basic model of wireless sensor network

A typical simple wireless sensor network is shown in fig.1. In which, a complete wireless sensor network usually consists of one or more base stations or gateway, a number of sensor nodes, and the end user. Sensor nodes are use to measure physical quantities such as temperature, position, humidity, pressure etc. The outputs of those sensor nodes are wirelessly

transmit to the base station for data collection, analysis, and logging. End users may also be able to receive and manage the data from the sensor via a website from long-distance or applications in console terminal.

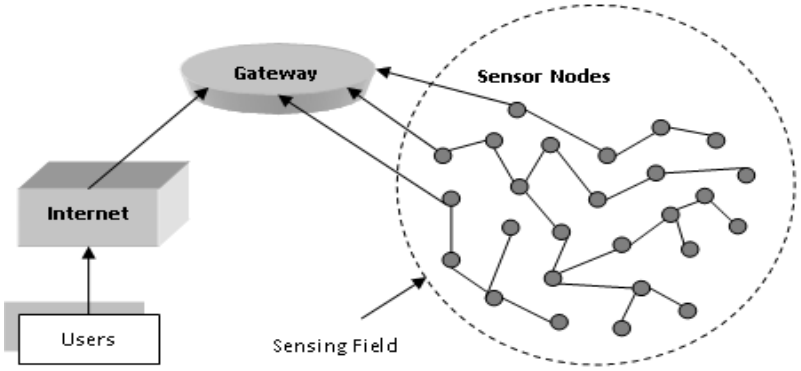

Fig. 1: A simple wireless sensor network

B. **The General Characteristics of WSN:** Ability to cope with node failures, Mobility of nodes, Dynamic network topology, Communication failure, Heterogeneity of nodes, Withstand to harsh environmental conditions, Easy way to use, Unattended operation, Scalability to large scale of deployment, etc. [1].

C. Need To Secure WSN:

1) Wireless sensor networks have many applications in homeland, military security and other areas in such area many networks have mission critical tasks.
2) Security is critical for networks which deployed in hostile environments.
3) Most actively monitor their surroundings and it is often easy to infer information other than the data monitored.
4) Such unwanted information often results in privacy breaches of the people in environment.
5) Moreover the wireless communication employed by sensor networks suffers eavesdropping and packet injection by an adversary.
6) The above factors demands security for wireless sensor networks at design time to operation safety privacy of sensitive data and privacy for people in sensor environments.
7) Providing security in sensor networks is even critical than MANETs (Mobile Ad-hoc Networks) due to the resource limitations of sensor nodes.

II. SECURITY TECHNIQUES FOR WIRELESS SENSOR NETWORKS

There are several researches which represents several techniques to secure wireless sensor networks which are summarized as below:

A) MoteSec-Aware

In 2013, Yao-Tung Tsou and Chun-Shien Lu and *et.al* present a security mechanism called MoteSec-Aware which is build on network layer for wireless sensor networks with the focus to provide secure network protocol and data access control using Virtual Counter Manager (VCM) with synchronized incremental counter approach and applied to detect the jamming and replay attacks using the symmetric key cryptography with AES in OCB mode. Also for access control they have used the Key-Lock Matching (KLM) method for preventing the unauthorized access. In their research they have applied MoteSec-Aware for TelosB prototype sensor platform which running on TinyOS 1.1.15, and conduct field of experiments and TOSSIM-based simulations for evaluating the performance of MoteSec-Aware system. They produce output of this research by using feasible and efficient MoteSec-Aware mechanism shows less energy consumption while communication, and fulfill the high level of security than other several state-of-the-art methods [3].

i) Important conditions for MoteSec-Aware System

While doing research they find out that for tackling critical security, their system i.e. MoteSec-Aware must fulfill following conditions:

- **Data Confidentiality:** The basic property of a secure communication protocol which allows keeping the data secret from unauthorized reading
- **Replay and Jamming Detection:** There must be insurance of recent communication data and not replaying adversary as well jamming the data

- **Data Authentication:** It must prevent an opponent from spoofing packets. And Message Authentication Code (MAC) should use for each packet to verifying origination and transmission
- **DoS-Resilience**: For depleting energies the DoS must have attacks, by resisting in particular for resource limited sensor nodes
- **Data Access**: There must be detection and prevention of adversary from accessing data storage on nodes

ii) Model design of MoteSec-Aware System

As shown in fig.2 the design of MoteSec–Aware system has two bottom layers raw hardware and hardware abstraction layers which provides all basic services and components of limited resources and having TinyOS above them which surrounds by MoteSec–Aware core which is build up by many materials such as Memory Data Access Control Policy (MDACP), Event Handler, VCM, Query Logic, and Key Pool. Basically they applied MoteSec–Aware on TelosB by using symmetric key cryptosystem with communication key to prevent data confidentiality.

For protection against outside network messages and inside memory data leakage they applied AES-OCFA and MDACP strategies. AES-OCFA included two procedures for justifying DoS and detecting replay or jamming attack. They modified their previous method Constrained Function-based Authentication (CFA) and fixed with the AES in OCB a more efficient mode and two times faster than CBC-MAC mode which produce a cipher text providing simultaneously data secrecy and authenticity.

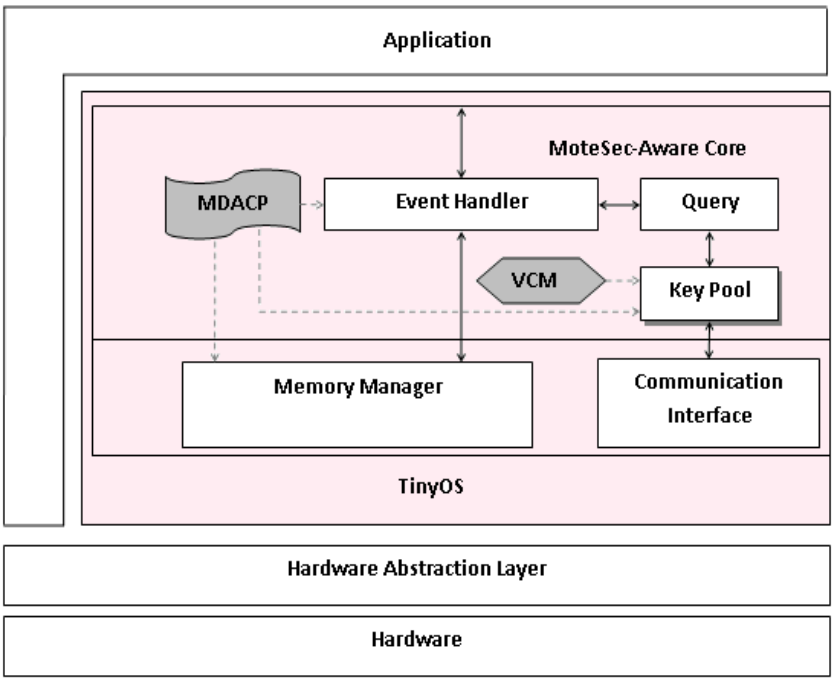

Fig. 2: MoteSec-Aware Protocol Stack

B) Security Protocols for Sensor Networks: SPINS

Adrian Perrig and *et.al* in 2001 optimize and report a set of protocols designing for resource constrained environments and wireless communication called as "SPINS" with following two building blocks [4].

i) SNEP- Sensor Network Encryption Protocol

It provides basic primary objectives such as Data confidentiality which is used in every security protocols, facilitate two-party data authentication, and also evidence of the data freshness with following advantages.

1. It only adds 8 bytes per message i.e. low communication overhead.
2. Similar to other cryptographic protocols it uses a counter, but by avoiding the transmitting counter value through keeping state at both end points it makes a difference.
3 Prevention of eavesdroppers from interrupting the message content with the encrypted message is a very strong semantic security phenomenon.
4 This is very simple protocol that provides Data Authentication, Replay Prevention, and Weak Message Freshness.

However the more energy requires for sending data over the RF channel. So, to face this problem SNEP build another cryptographic mechanism by achieving semantic security having no additional transmission overhead. SNEP actually operate by combination of two mechanisms as it relies on the shared counter between sender and receiver for the block cipher with in counter mode (CTR). Both of communicating members are share the counter and then increment it after an each block, so they does not need to sent counter with the message. Also achieving two-party authentication and data integrity, SNEP uses a message authentication code (MAC).

Applicative properties of SNEP:

- **Semantic security:** The counter value is increases after each message and long enough which is never repeats within the lifetime of node therefore the same message is encrypted differently at each time.
- **Data authentication:** The receiver becomes assured about the originating of message from the claimed sender, only when the MAC verifies correctly.
- **Replay protection:** Replaying old messages prohibited by the counter value in the MAC (absence of the counter in the MAC, indicates replaying of messages due to adversary).
- **Weak freshness:** Enforcement of a message ordering and yields weak freshness due to verification of message correctly and sending it after previous one by receiver.
- **Low communication overhead:** Does not need to send the counter state in each message as it is set to be at each end point.

ii) µTESLA (Micro-Timed Efficient Stream Loss-tolerant Authentication)

Asymmetric digital signatures are impractical for sensor networks for the authentication, they need long signatures with the high communication overhead of 50-1000. Thus earlier TESLA protocol provided efficient authenticated broadcast However, TESLA not designed for sensor networks. In 2001 Adrian Perrig and et al. proposed µTESLA to solve the following difficulties of TESLA in sensor networks:

- TESLA authenticating initial packet with a digital signature is too expensive for sensor nodes. µTESLA allows only the symmetric mechanisms.
- Key disclosed in each packet requires large amount of energy for sending and receiving. µTESLA disclosed the key at once per epoch.
- Storing a "one-way key chain" make a sensor node expensive. µTESLA restricts the number of authenticated senders.

Mechanism with example behind µTesla system:

The basic idea behind µTesla is to achieving asymmetric cryptography via delayed the disclosure of the symmetric keys. With this condition a sender will broadcast a message which generated with a secret key. After a some period of time, sender will disclose the secret key while the receiver having an obligation for buffering the packet up to the secret key has been disclosed. So after these disclosure the receiver authenticate the packet, with condition that the packet must be received before the key was disclosed. But µTesla suffers from one limitation that some initial information need to unicast to each of sensor nodes before authentication of broadcast messages that can begin cryptography by delaying disclosure of symmetric keys. In this situation a sender will broadcast a message that generated with a secret key.

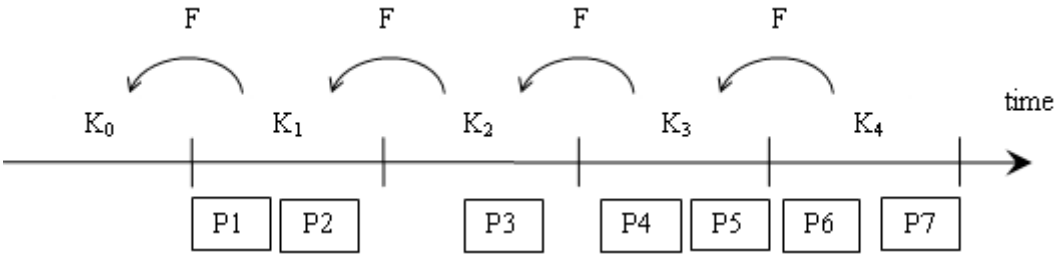

Fig. 3: An example of µTESLA

Assume that the receiver node is loosely time synchronized and knows K0 (a commitment to the key chain) in an authenticated way. As the fig. 3 shows, Packets P1 and P2 sent in interval 1 contain a MAC with key K1. Packet P3 contain a MAC using key K2. So far, the receiver cannot authenticate any packets yet. Let us assume that packets P4, P5, and P6 are all lost, also the packet that discloses key K1, so the receiver can still not authenticate P1, P2, or P3. At interval 4 the base station broadcasts key K2, to which the node authenticates by verifying K0 = F (F(K2)), and hence also knows K1 = F(K2), so it can authenticate the packets P1, P2 with K1, and P3 with K2. Instead of adding a disclosed key to each data packet, the key disclosure is

independent from the packets broadcast, and is tied to time intervals. In the context of μTESLA, the sender broadcasts the current key periodically in a special packet.

C). TinySec

TinySec is a link-layer security architecture for wireless sensor networks that is part of the official TinyOS release. It generates secure packets by encrypting data packets using a group key shared among sensor nodes and calculating a MAC for the whole packet including the header. It provides two modes of operation for communication namely, authenticated encryption and authentication only. Authentication is only the default mode of operation, where the payload in the TinyOS packet is not encrypted; each packet is simply enhanced with a MAC. In the authenticated encryption mode the payload is encrypted before the MAC is computed on the packet. The key distribution mechanism is left out and must be implemented as a separate part of the software. The TinySec architecture is shown in fig. 4 [5] [6].

In 2004 Karlof and *et.al* introduce TinySec i.e."Link Layer Security Architecture for Wireless Sensor Networks" the replacement for the unfinished SNEP, the first fully-implemented link layer security architecture for wireless sensor networks which provide In their design, they leverage recent lessons learned from design vulnerabilities in security protocols for other wireless networks such as 802.11b and GSM. Conventional security protocols tend to be conservative in their security guarantees, with small memories, weak processors, limited energy, and sensor networks cannot afford this luxury. TinySec addresses these extreme resource constraints with careful design; they explore the tradeoffs among different cryptographic primitives and use the inherent sensor network limitations to their advantage when choosing parameters to find a sweet spot for security, packet overhead, and resource requirements. TinySec is portable to a variety of hardware and radio platforms. Their experimental results on a 36 node distributed sensor network application clearly demonstrate that software based link layer protocols are feasible and efficient, adding less than 10% energy, latency, and bandwidth overhead [5].

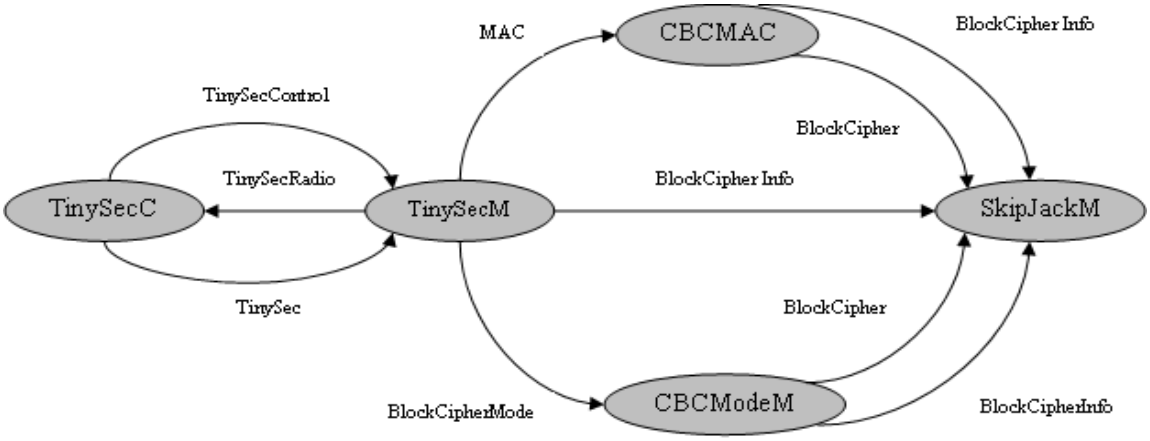

Fig. 4: Relationship between components in TinySec

D). ZigBee

ZigBee is global standards of communication protocol. In 2005 the relevant task force under IEEE 802.154, for data communications dealing business and consumer devices proposed ZigBee. It is design for low power consumption enabling batteries to last forever. The ZigBee standard provides network, security, and application support services operating on top of the IEEE 802.15.4 Medium Access Control (MAC) and Physical Layer wireless standard. It employs a group of technologies to enable scalable, self-organizing, self-healing networks that can manage various data traffic patterns. ZigBee is a low-cost, low power, wireless mesh networking standard. The low cost allows the technology to widely deployed in wireless control and monitoring applications, in which the low power usage allows longer life with smaller batteries, and the mesh networking which promises high reliability and larger range [7] [6].

i) ZigBee General Characteristics

- Global operation in the 2.4GHz frequency band according to IEEE 802.15.4
- Regional operation in the 915MHz (America) and 868MHz (Europe)
- Low power consumption, with battery life ranging from months to years
- Maximum data rates allowed for each of these frequency bands are fixed as 250kbps at 2.4GHz, 40kbps at 915MHz, and 20kbps at 868MHz
- Channel access using Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)

- Yields high throughput and low latency for low duty cycle devices like sensors and controls
- Multiple topologies: star, peer-to-peer, mesh
- Addressing space of up to 64 bit IEEE address devices and 65,535 networks
- Fully reliable "hand-shake" data transfer protocol
- Range: 50m typical (5-500m based on environment)

ii) ZigBee Protocol Stack Architecture

The ZigBee protocol stack architecture consists of a set of blocks which called as layers. In which each layer can carry outs a specific set of services for their above layer, so a data entity gives a data transmission service and a management entity gives all other services. In which each service entity exposes an interface to their upper layer through a service access point (SAP) each of which supports a number of service primitives to get the required functionality.

The ZigBee protocol stack architecture is shown in fig. 5, it is based on OSI model (standard Open Systems Interconnection seven-layer model) but it defines only those layers that relevant to achieving functionality in the intended market space. The IEEE 802.15.4-2003 standard defines the two lower layers one as the physical (PHY) layer and second is the medium access (MAC) sub-layer. The ZigBee Alliance builds on this basic foundation by providing the network (NWK) layer and the framework for the application layer, which contain application support sub-layer (APS), ZigBee device objects (ZDO) and manufacturer-defined application objects.

IEEE 802.15.4-2003 contains two PHY layers that can operate in two separate frequency ranges as 868 MHz and 2.4 GHz. The lower frequency PHY layer covers both the 868 MHz European band and 915 MHz band that used in countries such as the United States and Australia etc. The higher frequency PHY layer is virtually worldwide used. The IEEE 802.15.4-2003 MAC sub-layer controls access to the radio channel using a CSMA-CA mechanism. It takes charge of transmitting beacon frames, synchronization and providing a reliable transmission mechanism.

The responsibilities of the ZigBee NWK layer include mechanisms used to join and leave network, to apply security to frames and to route frames to their intended destinations. Therefore, the discovery and maintenance of routes between devices devolve to the NWK layer and the discovery of one-hop neighbors and the storing of pertinent neighbor information done at this NWK layer. The NWK layer of a ZigBee coordinator is responsible for starting a new network, when appropriate, and assigning address to newly associated devices.

The ZigBee application layer consists of the APS, the Application Framework (AF), the ZDO and the manufacturer-defined application objects. The responsibilities of the APS sub-layer include maintaining tables for binding, which is the ability to match two devices together based on their services and their needs, and forwarding messages between bound devices. The responsibilities of the ZDO include defining the role of the device within the network (e.g., ZigBee coordinator or end device), initiating and responding to binding requests and establishing a secure relationship between network devices. The ZOD is also responsible for discovering devices on the network and determining which application services they provide.

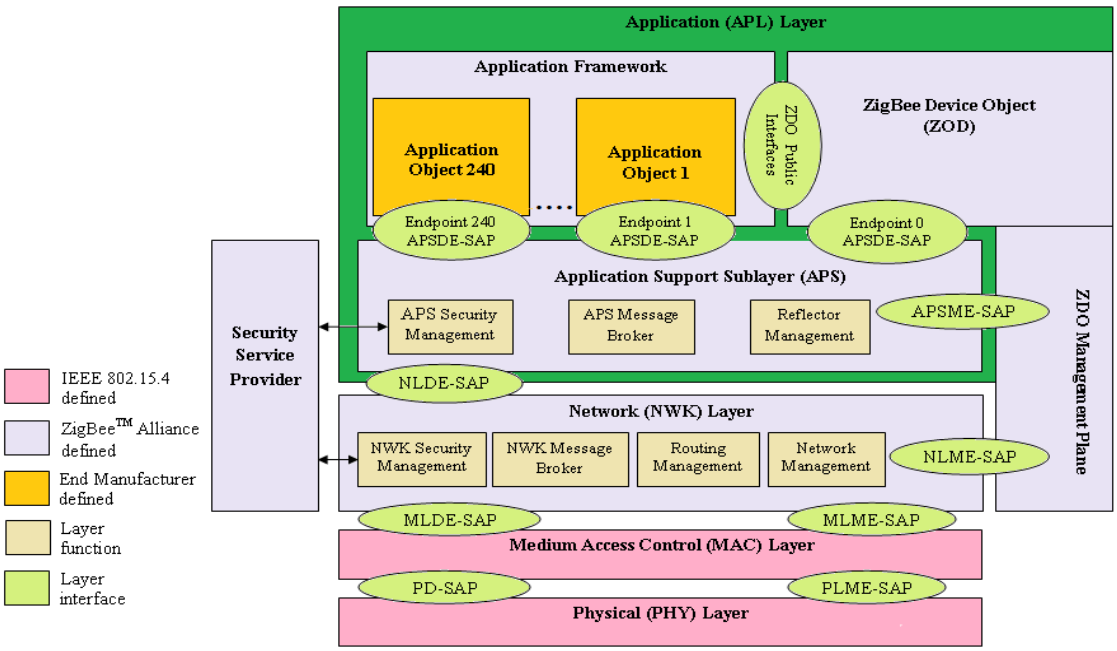

Fig. 5: ZigBee stack architecture

E) MiniSec

In 2007 G. Mezzour, A. Perrig, and V. Gligor M. Luk introduce MiniSec, a secure network layer protocol for wireless sensor networks. MiniSec has two different operating modes for unicast and broadcast communication between sensor nodes, called MiniSec-U and MiniSec-B respectively and uses the OCB-encryption scheme for both encryption and authentication. They also provide semantic security by the use of a counter as a nonce. In the case of the unicast mode two synchronized counters are kept at the sender and at the receiver, while in the broadcast mode the authors propose the use of a Bloom-filter2 based mechanism that precludes per-sender state [8] [6].

i) MiniSec-U

In unicast mode, MiniSec requires each pair of nodes in the network to share two keys: K_{AB} and K_{BA} for $A \rightarrow B$ and $B \rightarrow A$ communication, respectively. A 32-bit counter that is increase for each new message is assign to each key to guarantee semantic security. Counter C_{AB} is use for key K_{AB} and counter C_{BA} for key K_{BA}. Only the last x bits of the counter value are included in each packet to save the energy of transmitting more bits. Both sender and receiver keep track of the counters, which have to be synchronized on both sides. The receiver can accept only messages with a counter value greater than this in the previous messages. However, the counters can be desynchronized and a counter resynchronization protocol is needed.

Unless it is known before deployment which pairs of nodes are going to use unicast communication, each node in the network should maintain a counter for each possible sender (i.e. its neighbours), resulting in high memory overhead and making counter resynchronization very expensive. These problems also dictate the use of a different mechanism for the broadcast case.

ii) MiniSec-B

Two mechanisms are used in MiniSec-B to provide semantic security and replay protection. The first one requires time synchronization among the nodes and divides time in epochs $E1, E2, E3, \ldots$ The number of the current epoch is used as the nonce for OCB-encryption. When a node receives a packet, it attempts decryption twice; one with the current epoch number and one with the immediately previous epoch number. The epoch length is defined in a way that compensates for time synchronization errors and network latency.

The second mechanism defends against replay attacks within the current epoch. Each sender nodes keeps a counter which is incremented for each new message. At the end of each epoch the counter is reset, which means it can be shorter than the counter in MiniSec-U (the authors found that it was sufficient to use an 8-bit counter). The receiver keeps two alternating Bloom filters, one for the current epoch and one for the previous epoch. Each time it receives a packet it queries the corresponding Bloom filter and if the query returns true, the packet is considered to be a replay. The problem, however, is that the Bloom filters may cause false positives, causing a legitimate packet to be rejected as a replayed packet.

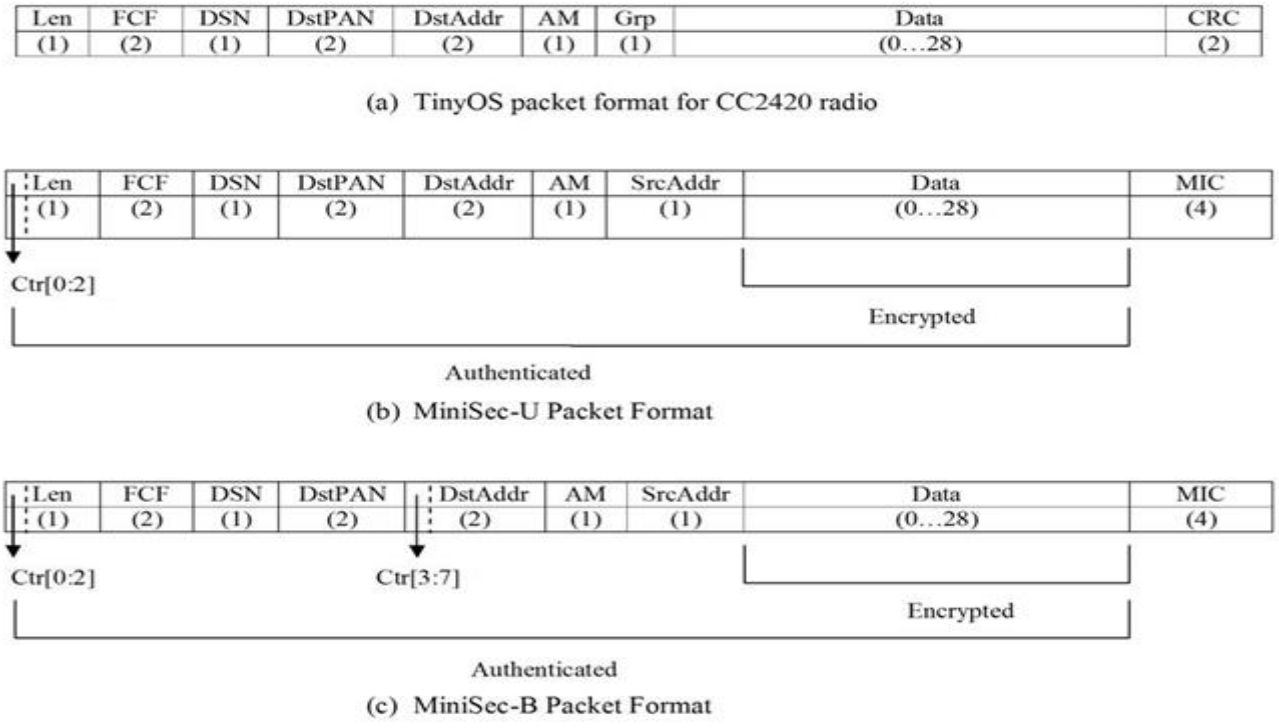

Fig. 6: MiniSec packet format in the unicast and broadcast modes

iii) MiniSec packet format

Fig. 6 shows the packet formats for MiniSec-U and MiniSec-B compared to the TinyOS packet format for the CC2420 radio (compliant with IEEE 802.15.4). Like in TinySec, the Group ID has been removed from the header, since access control is achieved through the use of different cryptographic keys. The 2-byte CRC is replaced by a 4-byte MIC (Message Integrity Code). The difference between MiniSec-U and MiniSec-B is that for the unicast mode, only $x = 3$ bits of the counter are sent in the packet header, while for the broadcast mode the whole counter has to be sent.

III. COMPARISON OF SECURITY TECHNIQUES

TABLE I: COMPARISON OF SECURITY TECHNIQUES IN WSNS (*N*: NUMBER OF NODES; φ: PACKET LOSS RATE; *I*: BYTES OF THE IV) [3]

Techniques	Replay Detection	Jamming Detection	DoS Resilience	Memory Access Control	Packet Security Overhead	Communication Cost
MoteSec-Aware	Yes	Yes	Yes	Yes	No	$O(N)$
SPINS	Yes	Yes	No	No	Counter resynchronization	$O(N\varphi^c)(c:constant)$
TinySec	No	No	No	No	With 8-byte IV	$O(N + NI)$
Zigbee	Yes	No	No	No	With 8-byte IV	$O(N + NI)$
MiniSec	Yes	Yes	No	No	Few bits of the IV	$O(N + NI)$

IV. CONCLUSION

Security is serious issue and complicated enough to set up in different parts of Wireless Sensor Networking (WSN) system, such as development of security mechanism and making it efficient, low consumption of energy, cost effective and above review highlights different techniques for WSNs which allowed or encourage you to expand your research and new challenges in research. In this present literature, we focus a brief survey on wireless sensor network, its characteristics, need for security, also represent various security techniques for WSN.

REFERENCES

[1] Wei Hong and David E. Culler, "Wireless Sensor Networks: Introduction," *Communications Of The ACM*, vol. 47, no. 6, pp. 30-33, June 2004.

[2] S. Kun, L. An, N. Peng, and M. Douglas, "Securing network access in wireless sensor networks," in *Proc. International Conference on Wireless Network Security*, 2009, pp. 261–268.

[3] Yao-Tung Tsou, Sy-Yen Kuo, and Chun-Shien Lu, "MoteSec-Aware: A Practical Secure Mechanism for Wireless Sensor Networks," *IEEE Trans. Wireless Communication*, vol. 12, no. 6, pp. 2817-2829, June 2013.

[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proc. International Conference on Mobile Computing and Networking*, 2001, pp. 189–199.

[5] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proc. International Conference on Embedded Networked Sensor Systems*, 2004, pp. 162–175.

[6] I Krontiris, T Dimitriou, H Soroush, and M Salajegheh. (2008) WSN Link-layer Security Frameworks. PDF. [Online]. http://www.web-portal-system.de/wps/wse/dl/showfile/rannenberg/5305/LinkLayer_Sec_bookch.pdf

[7] ZigBee Alliance, Zigbee specifications, Technical Report Document 053474r06, 2005.

[8] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proc. International Conference on Information Processing in Sensor Networks*, 2007, pp. 479–488.

