Performance Evaluation of Security Implementation in IEEE 802.15.4 Beacon Enabled Networks

by

Moazzam Khan

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Computer Science

University of Manitoba

Winnipeg

Performance Evaluation of Security Implementation in IEEE 802.15.4 Beacon Enabled Networks

By

Moazzam Khan

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

Thesis advisor

Jelena Mišić

Author

Moazzam Khan

Performance Evaluation of Security Implementation in IEEE 802.15.4 Beacon Enabled Networks

Abstract

The IEEE 802.15.4 specifications outline a class of wireless radios and protocols targeted at low power devices, personal area networks, and sensor devices. The IEEE 802.15.4 specification employs a number of well-known security services that can be implemented in such networks. Sensor network application developers need overhead statistics in choosing the security service that best suits securing a particular threat environment. For evaluating these security overheads on wireless sensor networks, I have simulated the IEEE 802.15.4 media access control layer including support for security features with interconnected clusters of nodes and also designed and simulated key exchange protocol. Clusters are interconnected via bridges implemented on cluster coordinators, where each cluster has to contribute a constant number of packets per second to the sink. Packets are protected with the Message Authentication Code that requires a secret key shared between each node and coordinator. I have modeled network behavior under regular sensing traffic, power management, and periodic key exchanges. Symmetric and Asymmetric key exchange protocols of varying complexity are considered, and the impact of their communication and computation complexity on the network lifetime is evaluated. My model and results can be used

to determine network and security parameters in IEEE 802.15.4 based wireless sensor network when application requirements are known.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. Jelena Mišić, for all of her support and guidance during my studies here at University of Manitoba. She is the one who made me interetsed in the field of network security and because of her great lectures during my undergraduate studies induced me to continue my graduate research with her. Thanks to Dr. Rasit Eskicioglu from department of Computer Science and Dr. Ekram Hossain from department of Electrical and Computer Engineering for accepting to be in my thesis defense committee, pointing out the mistakes and problems in the thesis, and their helpful comments to make the final version of the thesis better.

Many thanks to Fereshteh Amini and Hossein Pourreza for their insightful discussions and comments about my work and for being there for me when I needed help. Working as a team with Fereshteh during my early research has made it easier and accurate. I thank my Mom, Dad, my Sister and Brothers for their love and encouragement throughout all of my university studies. My appreciation goes to my friends at the Wireless Sensor Network's Lab. for their support and for making my time here more enjoyable.

This thesis is dedicated to my parents for their love, endless support and encouragement.

Chapter 1

Background and Related Work

Recently adopted IEEE 802.15.4 standard is poised to become the key enabler for low complexity, ultra low power consumption, low data rate wireless connectivity among inexpensive devices such as sensors. The standard defines the physical and mediam access control layer for low rate wireless personal area networks. The wireless personal area networks are used in sensing applications such as habitat monitoring, burglar alarms, inventory control, medical monitoring, emergency response and battlefield management and such applications sometime require reliable and secure data transfer.

Two network topologies are allowed by the standard, but both of them rely on the presence of a central controller device known as the PAN coordinator. In the peer-to-peer topology, devices can communicate with one another directly, as long as they are within the physical range. In the star-shaped topology, the devices must communicate through the PAN coordinator. The network uses two types of channel access mechanism, one based on slotted CSMA-CA algorithm in which the slots are

aligned with the beacon frames sent periodically by the PAN coordinator, and another based on unslotted CSMA-CA in which case there are no beacon frames. The beacon enabled mode and the star-based (in the text that follows will refer to star-based topology as cluster-based topology) hierarchical topology appear to be better suited to sensor network implementation than their peer-to-peer counterparts because the PAN coordinator can act as both the network controller and the sink to collect data from the sensor nodes. Within one cluster, time is organized in superframes which are delineated by beacons sent by the PAN coordinator. Superframe is further organized as an active part where nodes can transmit using CSMA-CA or TDMA (called Guaranteed Time Slots) and an inactive part where all nodes sleep. Larger areas under surveillance can be efficiently covered by interconnecting clusters in mesh topology through their coordinators. This feature is enabled through the existence of inactive superframe part since coordinator can then switch to another cluster and communicate as an ordinary node. When communication in foreign cluster is finished, coordinator returns to its own cluster.

Wireless devices used for sensing the environment are low in computational power and memory resources. The bandwidth offered by IEEE 802.15.4 standard is low, since the standard allows the PAN to use either one of three frequency bands: 868-868.6MHz, 902-928MHz and 2400-2483.5MHz with raw data rates of 20kbps, 40kbps and 250kbps respectively. However, the bandwidth available to the application is further decreased due to CSMA-CA access with small backoff windows (default backoff window sizes without power saving mode are 8, 16, 32, 32, 32, respectively, for five allowed backoff attempts). Also, in downlink communications, PAN coordinator first

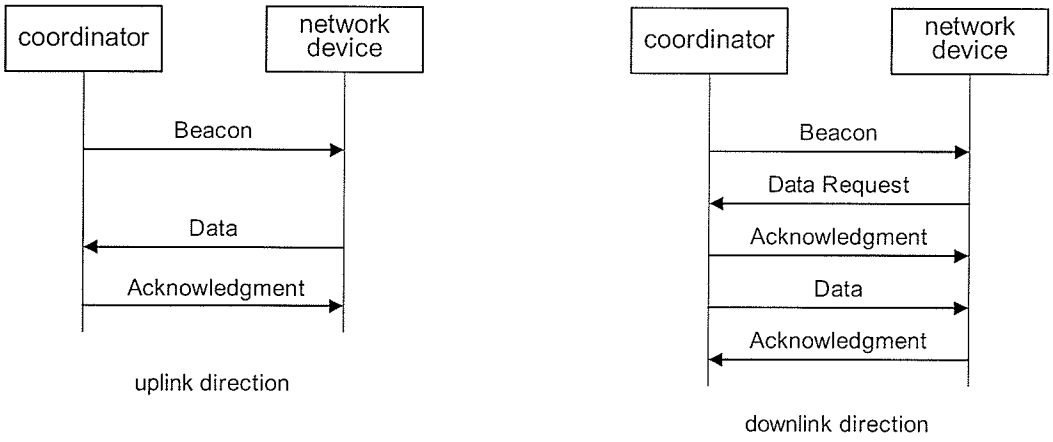


Figure 1.1: Data Transfer in 802.15.4 PAN in beacon enabled mode

has to advertise the packet in the beacon, then node has to send the request packet asking for downlink transmission, and finally downlink transmission can commence. Therefore, in the presence of many nodes in the cluster, effective bandwidth left to the application is less than 20% of the raw bandwidth [25].

Providing security services in such wireless sensor networks is a technical challenge. Algorithms for key exchange which naturally include authentication elements and addition of packet signature will further decrease the bandwidth available to the sensing application. Besides, complex computations often involved in public key cryptography might consume too much energy and memory resources. Therefore, goal of designing low power sensor devices forces security mechanism to fit under processing, memory and bandwidth constraints.

This chapter is organized as follows. In Section 1.1, I will explain the relationship between the sensor network architecture and its availability for both data collection and event sensing applications. Networks supporting secure applications, network

availability for sensing applications has the same importance as data integrity and data confidentiality. Section 1.1 explains the need of security in wireless sensor networks and which types of security techniques are considered in such networks. Detailed description of security features of IEEE 802.15.4 [3] based sensor networks is presented in Section 1.2. Section 1.3 discusses key exchange approaches currently used in WPANs.

1.1 Cluster-based wireless sensor networks

One of the most significant benefits of sensor networks is that they extend the computation capability to physical environments where human beings can not reach. However, energy possessed by sensor nodes is limited, which becomes the most challenging issue in designing sensor networks. The main power consumptions in sensor networks are computation and communication between sensor nodes. In particular, the ratio of energy consumption for communication and computation is typically in the scale of 1000 [20]. Therefore it is critical to enable collaborative information processing and data aggregation to prolong the lifetime of sensor networks. The choice of network topology in wireless sensor networks is still an open question. However, it seems that the choice of topology is an issue of trade off between the node simplicity and homogeneity versus the duration of network lifetime. For sensor networks covering large geographic areas it is difficult to replace sensors batteries when they are exhausted, and therefore, when nodes close to sink die the whole network is unavailable.

Wireless sensor networks can carry two different types of sensing. First kind of

sensing is data collection where nodes in the network frequently communicate to report measurements. Depending on the application requirements, some collective sleep technique for all the nodes in the cluster can be used in order to extend the network lifetime. Data collection applications exploit spatial correlation of sensed data and in order to save bandwidth perform some kind of data aggregation. In peer-to-peer IEEE 802.15.4 architectures aggregation is performed in nodes which are conveying sensed data towards the sink. In cluster-based architectures aggregation occurs at the PAN coordinator and aggregated packets are conveyed to the next coordinator along the path possibly over more powerful link GTS(Guaranteed Time Slot) compared to the link type this is available to ordinary nodes CSMA-CA(Carrier sense multiple access with collision avoidance). From the aspect of available bandwidth, presence of GTS links between the cluster coordinators puts the cluster based networks in advantage over the peer-to-peer networks. Also, the aggregation done by the coordinator can be made much more secure than the aggregation in peer-to-peer network since coordinator is always aware about the identities of the nodes which participate in the aggregation (since this is done in the attachment process), while the set of neighbors in peer-to-peer network might depend on the type of the query. From the aspect of lifetime, it is reasonable to assume that PAN coordinators will have higher power resources than the ordinary nodes which combined with the GTS access will extend the lifetime of the network (since they will do the packet relaying).

In second kind of sensing, communication occurs only when some important event occurs. For applications where event detection is the target (e.g enemy troops movement, detection of noise level), sensors are required to be vigilant most of the time

which means that collective sleep of the nodes is prohibited. Event detection requires reporting only when an event occurs in contrast to data collection where communication of measurement is more frequent. In this case aggregation is avoided and it is important to deliver the sensed data to the sink within some time bound. In event detection applications, network availability and data integrity is much more critical than in data collection applications. Again, I argue that cluster-based architecture where PAN coordinators have higher power resources, GTS links for communication and reliable information about cluster members offers better availability and data integrity than peer-to-peer architecture.

Nodes in wireless sensor networks can directly communicate with nearby nodes. Nodes that are not within direct communication range use other nodes to relay message between them. Routing in such a multi hop network is challenging due to the lack of central control and the high dynamics of the network. Recent work has focused on discovering and maintaining routes that keep the connectivity between the nodes or, furthermore, that minimize the number of hops on a path. One important restriction of a wireless sensor network is that nodes are energy-constrained as they are normally powered by batteries. However, the algorithms that aim to minimize the path length may ignore fairness in routing, for example, the shortest-path routing is likely to use the same set of hops to relay packets for the same source and destination pair. This will heavily load those nodes on the path even when there exist other feasible paths. Such an uneven use of the nodes may cause some nodes to die earlier, thus creating holes in the network or worse, leaving the network disconnected.

Low available bandwidth to nodes, CSMA-CA access, data aggregation, and rout-

ing in wireless sensor networks based on IEEE 802.15.4 make the implementation of security a technical challenge. Even at the media access control layer it is possible to launch Denial of Service attack which will drastically increase the number of collisions and prevent and data communication (due to CSMA-CA access and small backoff windows). The processing, communication and aggregation cost of secure packets first increases both computational and communication overhead. In order to decrease this overhead all the security parameters and keying model under which the network will work are selected with great care so that the objective of both secure communication and longer network life is achieved. These two objectives are competing and trade off between them is necessary.

1.2 Overview of IEEE 802.15.4 security operations

IEEE 802.15.4, a link layer security protocol provides four basic security services: access control, message integrity, message confidentiality, and replay protection(AppendixA gives a description of these security techniques). The security requirements can be tuned by setting the appropriate control parameters of the protocol stack. If an application does not set any parameters, then security is not enabled by default. An application must explicitly enable security features.

1.2.1 Addressing

For unique identification of a device in a network or cluster, addressing in IEEE 802.15.4 is accomplished via a 64-bit node identifier and a 16-bit network identifier. IEEE 802.15.4 supports a few different addressing modes. For example, a 16 bit

truncated address may be used in place of the full 64-bit node identifier in certain cases. This allows the sizes of the source and destination addresses vary between 0 and 10 bytes depending on whether truncated or full addresses are used, and whether or not the node sends to a broadcast address.

The specification defines four packet types for media access control layer:

1. Beacon packets
2. Data packets
3. Acknowledgment packets
4. Control packets

The specification does not support security for acknowledgment packets while security is optional for the rest of packets types depending on the need of application. Depending on the threat environment the application has a choice of security suites that control the type of security protection that is provided for the transmitted data. Each security suite offers a different set of security properties and results in different packet formats. The IEEE 802.15.4 specification defines eight different security suites outlined in (Table 1.1).

We can classify the suites by the properties they offer:

- No security
- Encryption only (AES-CTR)[1]

[1]AES-CTR: Advanced Encryption Standard in Counter Mode

Identifier	Security Suite Name	Access control	Data Encryption	Frame Integrity	Description
0x00	None	-	-	-	No Security
0x01	AES-CTR	X	X	-	Encryption Only
0x02	AES-CCM-128	X	X	X	Encryption & 128 bit MAC
0x03	AES-CCM-64	X	X	X	Encryption & 64 bit MAC
0x04	AES-CCM-32	X	X	X	Encryption & 32 bit MAC
0x05	AES-CBC-MAC-128	X	-	X	128 bit MAC
0x06	AES-CBC-MAC-64	X	-	X	64 bit MAC
0x07	AES-CBC-MAC-32	X	-	X	32 bit MAC

Table 1.1: Security Suites supported by 802.15.4 [Table 75 from [3]]

- Authentication only (AES-CBC-MAC)[2]
- Encryption and Authentication (AES-CCM)[3]

The specification supports MAC[4] of sizes that can be either of 4, 8 or 16 bytes long. The security feature of authentication is directly proportional to the length of MAC and it is very difficult for an adversary to break or guess a MAC of longer size. For example, with an 16 byte MAC, an adversary has a 2^{-128} chance of forging the MAC. The trade off is a larger packet size for increased protection against authenticity attacks.

[2]AES-CBC-MAC: Advanced Encryption Standard - Cipher Block Chaining - Message Authentication Code

[3]AES-CCM: Advanced Encryption Standard - Counter with CBC-MAC

[4]MAC: Message Authentication Code

The choice of secure authentication is tied with the addressing of devices in IEEE 802.15.4 protocol. Hence security suites are based on source and destination authentication addresses. Every device supporting IEEE 802.15.4 has an Access Control List (ACL) that control what security suite and keying information is used by each device. Each device can support up to 255 ACL entries. Each entry contains an 802.15.4 device address, a security suite identifier, and security material as shown in (Figure 1.2).

The *communication security material* is a persistent state necessary to execute the security suite. It consists of

- Cryptographic key
- Security Suite identifier
- Nonce state: that must be preserved across different packet encryption invocations.

Outgoing Frame packet and use of ACL

If security is enabled, the media access control layer looks up the destination address in its ACL table. If there is a match ACL entry, the security suite, and nonce specified in that ACL entry are used to encrypt or authenticate the outgoing packets. On the other hand in case of broadcast type of data packet where no specific destination address is mentioned, a default ACL entry is used, and this default entry matches all destination addresses

Incoming Frame packet and use of ACL

On packet reception the media access control defined by IEEE 802.15.4 examine

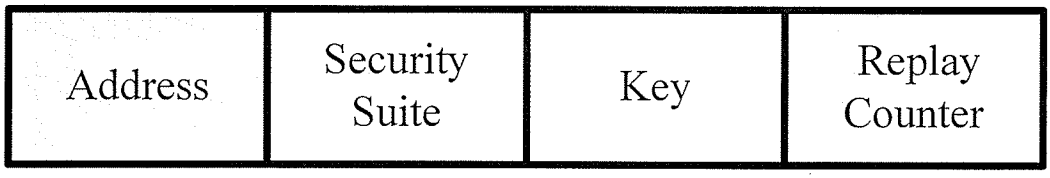


Figure 1.2: Access control List entry

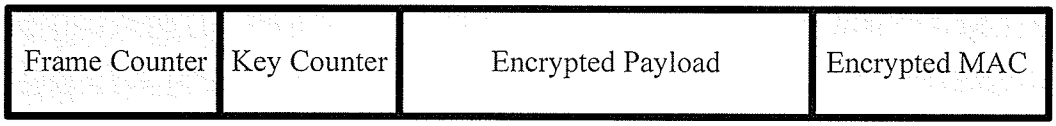


Figure 1.3: Frame format after adding security features

flag fields in the packet to determine if any security suite has been applied to that packet. If no security was applied, packet is passed to upper layer. Otherwise, media access control layer finds appropriate ACL entry corresponding to Sender's address. It then applies the appropriate security suite, and replay counter to the incoming packet. General structure of secured frame is shown in (Figure 1.3).

1.2.2 No Security

This is the simplest security suite. Its inclusion is mandatory in all radio chips. It does not have any security material and operates as the identity function. It does not provide any security guarantees.

1.2.3 AES-CTR

This suite provides confidentiality protection using the AES (Advanced Encryption Standard) block cipher with counter mode. To encrypt data under counter

mode, the breaks the plain text packet into 16-byte blocks $p_1, \ldots p_n$ and computes $c_i = p_i \oplus E_k(x_i)$. Each 16-byte block uses its own varying counter, which we call x_i. The recipient recovers the original plaintext by computing $p_i = c_i \oplus E_k(x_i)$. Clearly the recipient needs the counter value x_i in order to reconstruct p_i.

The x_i counter, known as nonce or IV, is composed of

- a static flag field
- the sender's address
- 3 separate counters: a 4 byte frame counter that identifies the packet, a 1 byte key counter field: The key counter is under application control and can be incremented if the frame counter ever reaches its maximum value, a 2 byte block counter that number the 16 byte blocks within the packet.

For employing security, requirement is that the nonce must never repeat within the lifetime of any single key, hence frame and key counters are introduced to prevent nonce reuse. The 2 byte block counter ensures that each block will use a different nonce value.

In summary, the sender includes the frame counter, key counter, and encrypted payload into the data payload field of the packet as show in (Figure 1.3)

1.2.4 AES-CBC-MAC

This suite provides integrity protection using CBC-MAC. The sender can compute either a 4,8 or 16 byte MAC using the CBC-MAC algorithm, leading to three different AES-CBC-MAC variants. The MAC can only be computed by parties with the

symmetric key. The MAC protects packets headers as well as the data payload. The sender appends the plain text data with the MAC. The recipient verifies the MAC by computing the MAC and comparing it with the value included in the packet.

1.2.5 AES-CCM

This security suite uses CCM mode for encryption and authentication. Broadly, it first applies *integrity* protection over the header and data payload using CBC-MAC and then *encrypts* the data payload and MAC using AES-CTR mode. As such, AES-CCM includes the fields from both the authentication and encryption operations: a MAC, and the frame and key counters. These fields serve the same function as above. Just as AES-CBC-MAC has three variants depending on the MAC size, AES-CCM also has three variants.

1.2.6 Replay Protection

A receiver can optionally enable replay protection when using a security suite that provides confidentiality protection. This includes AES-CTR and all of the AES-CCM variants. The recipients use the frame and key counter as a 5 byte *replay counter*, with the key counter occupying the most significant byte of this value. The recipient compares the replay counter from the incoming packet to the highest seen, as stored in the ACL entry. If the incoming packet has a larger replay counter than the stored one, then the packet is accepted and the new replay counter is saved. If, however, the incoming packet has a smaller value, the packet is rejected and application is notified of the rejection. I refer to this counter as the replay counter, even though it is the

same counter as the nonce, which used for confidentiality. The replay counter is not exposed to the application to use.

1.3 Key Management Models

Key management is the process by which keys are generated, stored, protected, transferred, updated and destroyed. Keying refers to the process of deriving common secret keys among communicating parties. Pre-deployed keying refers to the distribution of key(s) to the nodes before their deployment. Pairwise keying involves two parties agreeing on and communicating with a session key after deployment, while group keying involves more than two parties using a common group key. Group keying is important for multicasting.

Keying model that is most appropriate for an application depends on the threat model that an application faces and what type of resources it is willing to expend for key management. Depending on application types, key management models can be discussed under following parameters: (i) network architecture (ii) communication styles such as pair-wise (unicast), group-wise (multicast) or network-wise (broadcast), (iii) security requirements such as authentication, confidentiality or integrity, and (iv) keying requirements such as pre-distributed or dynamically generated pairwise, group-wise or network-wise keys. The constrained energy budgets and the limited computational and communication capacities of sensor nodes make use of public cryptography very challenging in large-scale sensor networks. At present, the most practical approach for bootstrapping secret keys in sensor networks is to use predeployed keying in which keys are loaded into sensor nodes before they are deployed.

Several solutions based on pre-deployed keying have been proposed in the literature including approaches based on the use of a global key shared by all nodes, approaches in which every node shares a unique key with the base station, and approaches based on random key sharing. In wireless sensor network, nodes use pre-distributed keys directly, or use keying materials to dynamically generate pair-wise and group-wise keys. Challenge is to find an efficient way of distributing keys and keying materials to sensor nodes prior to deployment. Solutions to key distribution problem in WSN can use one of the following popular approaches:

1.3.1 Probabilistic keying models

In probabilistic solutions, key-chains are randomly selected from a key-pool and distributed to sensor nodes. For example Random pair-wise key scheme [11] addresses unnecessary storage problem. In this scheme, each sensor node stores a random set of Np pair-wise keys to achieve probability p that two nodes are connected. At key setup phase, each node identity is matched with Np other randomly selected node with probability p. A pair-wise key is generated for each node pair, and is stored in every node's key-chain along with the identity of corresponding node. Similarly [13] also proposed probabilistic key pre-distribution scheme which relies on probabilistic key sharing among the nodes of a random graph.

1.3.2 Deterministic keying models

In deterministic solutions, deterministic processes are used to design the key-pool and the key-chains to provide better key connectivity. For example [8] suggested that

all possible link keys in a network of size N can be represented as an $N \times N$ key matrix. It is possible to store small amount of information to each sensor node, so that every pair of nodes can calculate corresponding field of the matrix, and uses it as the link key. Multiple space key pre-distribution scheme [12] improves the resilience of Blom's scheme. It uses a public matrix G and a set of ω private matrices D. Polynomial based key pre-distribution scheme [9] distributes a polynomial share (a partially evaluated polynomial) to each sensor node by using which every pair of nodes can generate a link key.

1.3.3 Hybrid keying models

Finally, hybrid solutions use probabilistic approaches on deterministic solutions to improve scalability and resilience. Polynomial pool-based key pre-distribution scheme [21] considers the fact that not all pairs of sensor nodes have to establish a key. It combines Polynomial based key pre-distribution scheme [9] with the key-pool idea in [13, 11] to improve resilience and scalability.

1.3.4 Group-wise keying models

In hierarchical WSNs, sensor nodes require group-wise keys to secure multicast messages. One approach is to use secure but costly asymmetric cryptography. Some studies [10], IKA2 [30] have used a Diffie-Hellman based group key transport protocol. However recently, more work on the public key cryptography protocols (e.g Elliptic curve cryptography) regarding evaluation and efficiency measurments on sensor node platforms showed optimistic results [32],[16]. In an hierarchical network, where a base

station share pair-wise keys with all the sensor nodes, base station can intermediate establishment of group-wise keys. Such protocols like Localized Encryption and Authentication Protocol (LEAP) [35] provide a mechanism to generate group-wise keys incorporating node location during pair-wise key establishment phase.

An important design consideration for security protocols based on symmetric keys is the degree of key sharing between the nodes in the system. At one extreme, we can have network-wide keys that are used for encrypting data and for authentication. This key sharing approach has the lowest storage costs and is very energy-efficient since no communication is required between nodes for establishing additional keys. However, it has the obvious security disadvantage that the compromise of a single node will reveal the global keys. At the other extreme, we can have a key sharing approach in which all secure communication is based on keys that are shared pairwise between two nodes. From a security point of view, this approach is ideal since the compromise of a node does not reveal any keys that are used by the other nodes in the network. However, under this approach, each node will need a unique key for every other node that it communicates with. Moreover, in many sensor networks, the immediate neighbors of a sensor node cannot be predicted in advance; consequently, these pairwise shared keys will need to be established after the network is deployed. A unique issue that arises in sensor networks that needs to be considered while selecting a key sharing approach is its impact on the effectiveness of in-network processing. Particular key exchange mechanism may reduce the effectiveness of in-network processing.

IEEE 802.15.4 compliant devices can share network key such that each cluster share only one key among all devices to exchange data and for authentication pur-

poses. This will ease the key management and memory overhead issues but this comes at the cost of lower security. Similarly IEEE 802.15.4 compliant device can also support pairwise key exchange that improves the over all security of network where any two devices exchanging data will share a different key. This improved robustness of network security comes at a cost, particularly in the overhead of key management. A device communicating with many devices in a network has to have different keys for each corresponding communicating device which will increase the memory overhead on resource scarce devices used in the network.

1.3.5 Key updates

Key management scheme is at the heart of securing such networks. Key management schemes for sensor networks can classified broadly into static and dynamic keying based on administrative key updates after network deployment. While static schemes assume no updates, dynamic ones provide for post deployment key updates. The general security and performance objective of key management schemes include minimizing number of keys stored per sensor node, providing rich logical pairwise connectivity, and enhancing network resilience to node capture.

Static Keying Schemes

Static keying management schemes (a.k.a key-predistribution) perform key management functions statically prior to or shortly after the deployment of the network. Administrative keys are generated at the sensor manufacturing time or by the base station upon network bootstrapping. Key assignment to nodes may be performed

on random basis or may take place based on some deployment information. Once generated and assigned, keys are pre-distributed to nodes. The main feature of static key management is the fact that the above key management cycle take place only once at or prior to initialization. Accordingly lost keys due to node capture and or failure are not compensated.

Dynamic Keying Schemes

The main feature of dynamic key management schemes is repeating the key management process either periodically or on demand to respond to node capture. After initial keying, key generation, assignment and distribution might take place (in a process known as rekeying) to create new keys that replaces the keys assumed lost or revealed to attacker so that the network is refreshed and the attacker loses information earned by node capture. Another advantage of dynamic keying is that upon adding new nodes, unlike static keying, the probability of network capture does not necessarily increase. Various dynamic key management techniques have been proposed with different key management responsibility taken by different network component.

1.3.6 Limitations of IEEE 802.15.4 standard from the security aspect

Higher layers will determine when the security is employed at media access control layer by any device and provide all keying material necessary to provide the security services. Key management, device authentication and freshness protection may be provided by the higher layers but is not addressed in IEEE 802.15.4 standard. The

management and establishment of keys is the responsibility of the implementer of higher layers.

Chapter 2

Symmetric and Asymmetric Key Exchange Models

2.1 Security services provided by ZigBee alliance

In previous Chapter we have outlined number of problems in achieving the target of secure communication in wireless sensor networks. IEEE 802.15.4 cluster based wireless sensor network provides higher bandwidth links for inter-coordinator communication, and allows higher power resources at the coordinator but still during deployment of such network needs a lot effort in choosing the right keying model based on the application requirements. Since key exchange protocols require downlink communications from the PAN coordinator to the ordinary nodes it will consume major bandwidth. Therefore, period of key exchange is a crucial design parameter which has to match both security requirements and bandwidth requires for the sensing application. Also, addition of the Message Authentication Code at the end of the

packet decreases the bandwidth which is left to the application and affects the complexity of the aggregation. We expect that the future work in this area will deliver the reasonable tradeoff between the level of security and application bandwidth in large sensor networks implemented over interconnected IEEE 802.15.4 clusters. The IEEE 802.15.4 standard addresses good security mechanisms but still does not addresses the type of keying mechanism to be used employing supported security techniques.

Zigbee alliance [4] is an association of companies working together to enable wireless networked monitoring and control products based on IEEE 802.15.4 standard. After the acceptance of 802.15.4 as IEEE standard, Zigbee alliance is mainly focused on developing network and Application layer issues. Zigbee alliance is also working on Application Programming Interfaces (API) at network and link layer of IEEE 802.15.4. Alliance also introduces secure data transmission in wireless sensor network that are based on IEEE 802.15.4 specification but most of this work is in general theoretical descriptions of security protocol at network layer. There is no specific study or results published or mentioned by Zigbee alliance in regards to which security suite perform better in different application overheads. Zigbee alliance has recommended both symmetric and asymmetric key exchange protocols for different networking layers. Asymmetric key exchange protocols that mainly rely on public key cryptography are computationally intensive and their feasibility in wireless sensor networks is only possible with devices that are resource rich both in computation and power.

Keyed Hash function for Message Authentication

A hash function is a way of creating a small digital fingerprint of any data. Cryptographic hash function is a one-way operation and there is no practical way to calculate a particular data input that will result in a desired hash value thus is difficult to forge. A practical motivation for constructing hash functions from block ciphers is that if an efficient implementation of block cipher is already available within a system (either in hardware or in software), then using it as the central component for a hash function may provide latter functionality at little additional cost. IEEE 802.15.4 protocol supports a well known block cipher AES and hence Zigbee Alliance specification also relied on AES. Zigbee alliance suggested the use of Matyas-Meyer-Oseas [23] as the cryptographic hash function that will be based on AES with a block size of 128 bits.

Mechanisms that provide integrity checks based on a secret key are usually called Message Authentication Codes (MACs). Typically, message authentication codes are used between two parties that share a secret key in order to authenticate information transmitted between these parties. Zigbee alliance specification suggest the keyed hash message authentication code (HMAC) as specified in the FIPS Pub 198 [2]. A Message Authentication code or MAC takes a message and a secret key and generates a $MACtag$, such that it is difficult for an attacker to generate a valid (message, tag) pair and are used to prevent attackers forging messages. The calculation of $MacTag$ (i.e HMAC) of data $MacData$ under key $MacKey$ will be shown as follows

$$MacTag = MAC_{MacKey}MacData$$

2.1.1 Symmetric-Key Key Establishment Protocol (SKKE)

Key establishment involves two entities, an initiator device and a responder device, and is prefaced by a trust-provisioning step. Trust information (e.g., a master key) provides a starting point for establishing a link key and can be provisioned in-band or out-band. In the following explanation of the protocol we assume unique identifiers for initiator device's as U and for Responder Device (PAN Coordinator) as V. The master key shared among both devices is represented as $Mkey$.

We will divide Symmetric-Key Key Establishment Protocol (SKKE) between initiator and responder in following major steps.

Exchange of ephemeral data

Figure 2.1 illustrates the exchange of the ephemeral data where the initiator device U will generate the Challenge QEU. QEU is a statistically unique and unpredictable bit string of length $challengeLen$ by either using a random or pseudorandom string for a challenge $Domain\ D$. The challenge domain D defines the minimum and maximum length of the Challenge.

$$D = (minchallengeLen, maxchallengeLen)$$

Initiator device U will send the Challenge QEU to responder device which upon receipt will validate the Challenge QEU by computing the bit-length of bit string Challenge QEU as $challengeLen$ and verify that

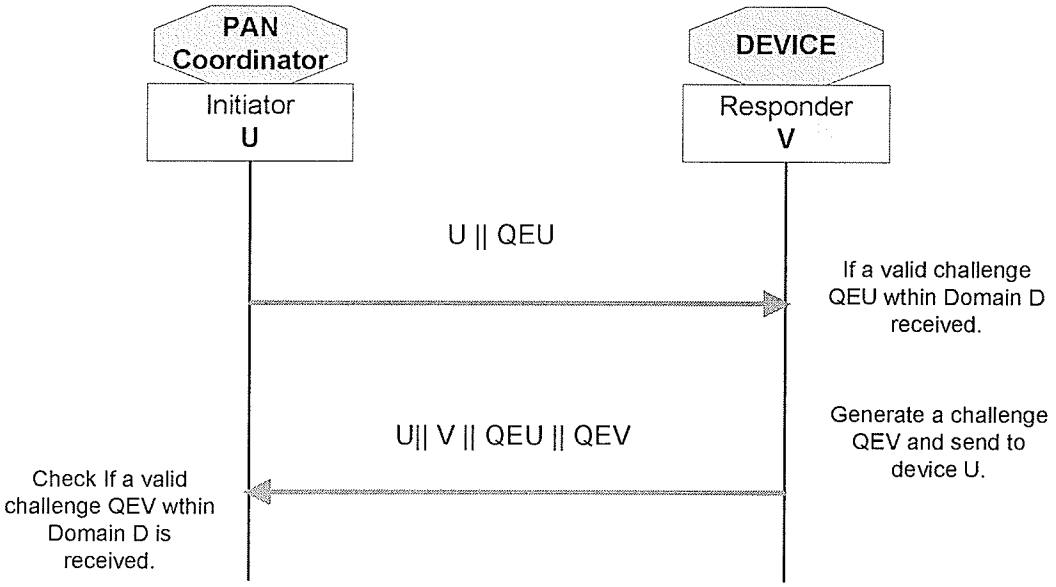


Figure 2.1: Exchange of ephemeral data

$$hallengeLen \in [minchallengeLen, maxchallengeLen]$$

Once the validation is successful the Responder device will also generate a Challenge QEV and send it to initiator device U. The initiator will also validate the Challenge QEV as described above.

Generation of shared Secret

Both parties involved in the protocol will generate a shared secret based on unique identifiers (i.e, distinguished names for each parties involved), symmetric master keys and Challenges received and owned by each party as shown in Figure 2.2.

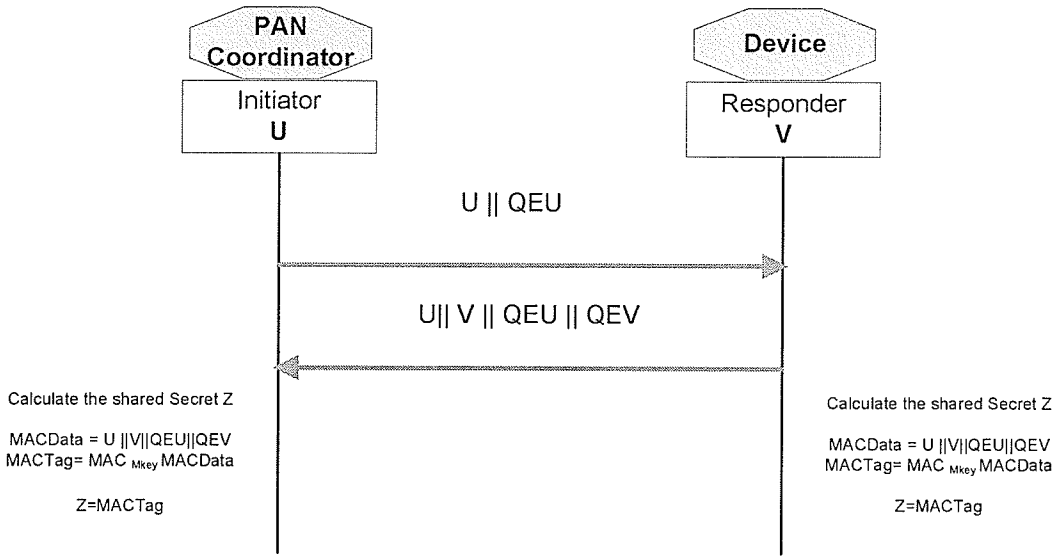


Figure 2.2: Generation of shared Secret

1. Each party will generate a $MACData$ by appending their identifiers and respective valid $Challenges$ together as follows

   $$MACData = U||V||QEU||QEV$$

2. Each party will calculate the $MACTag$ (i.e Keyed hash) for $MACData$ using $Mkey$ (Master Key for the device) as the key for keyed hash function as follows.

   $$MACTag = MAC_{Mkey}MACData$$

3. Now both parties involved have derived same secret Z

   (note: This is just a shared secret not the Link key. This Shared secret will be involved in deriving the link key but is not the link key itself.)

   Set $Z = MACTag$

Derivation of Link key

Each party involved will generate two cryptographic hashes (this is not keyed hash) of the shared secret as described in ANSI X9.63-2001 [1].

$Hash_1 = H(Z||01)$

$Hash_2 = H(Z||02)$

The hash value $Hash_2$ will be Link key among two devices (Figure 2.3). Now for confirming that both parties have reached on same Link key ($KeyData = Hash_2$) we will use value $Hash_1$, as key for generating Keyed hash values for confirming stage of the protocol.

$$MACKey = Hash_1 \tag{2.1}$$

$$KeyData = Hash_2 \tag{2.2}$$

$$KKeyData = Hash_1||Hash_2 \tag{2.3}$$

Confirming Link key

Till this stage of protocol both parties are generating the same values and now they want to make sure that they reached on same Link key values but they do not want to exchange the actual key at all. For this they will once again rely on keyed hash

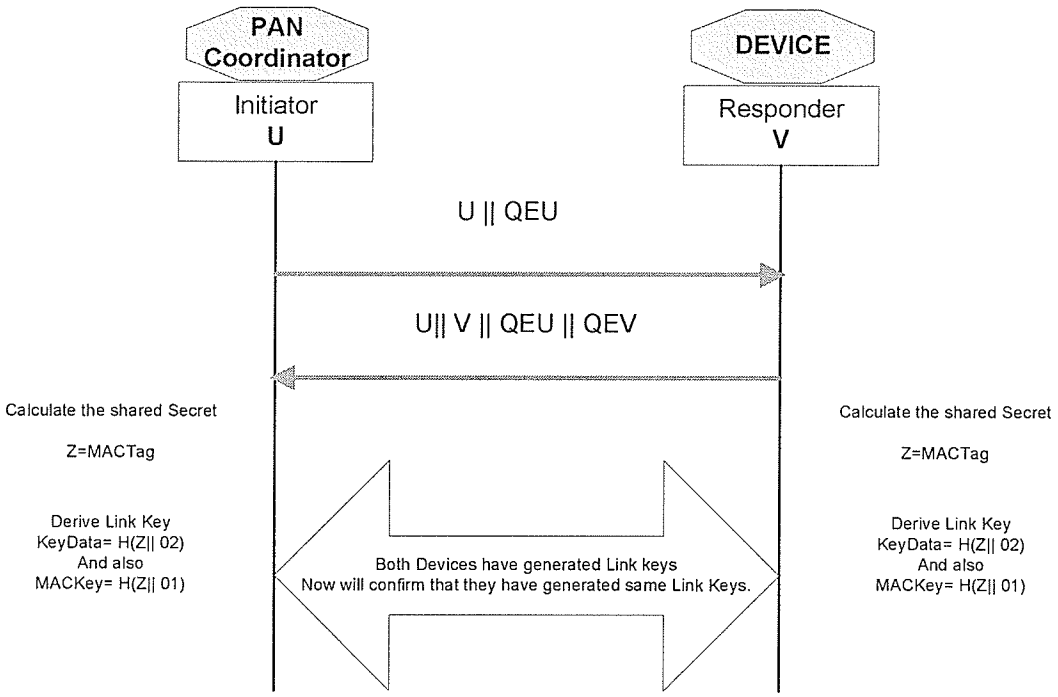


Figure 2.3: Generation of Link Key

functions and now both devices will generate different $MACTags$ based on different Data values but will use same key (i.e. $MACKey$) for generating the keyed hashes ($MACTags$).

1. Generation of MACTags

   Initiator and responder devices will first generate $MACData$ values and based on these values will generate $MACTags$. Initiator device D will receive the $MACTag_1$ from the responder device V and generate $MACTag_2$ and send to device V.

   We explain the generation of both $MACData$ values and $MACTags$ as follows

First both devices will calculate $MACData$ values

$$MACData_1 = 02_{16}||V||U||QEU||QEV$$

$$MACData_2 = 03_{16}||V||U||QEU||QEV$$

From the above $MACData$ values both devices will generate the $MACTags$ using the key $MACkey$ (Equation 2.1) as follows

$$MacTag_1 = MAC_{MacKey}MacData_1$$

$$MacTag_2 = MAC_{MacKey}MacData_2$$

2. Confirmation of MACTags Now the initiator device D will receive $MacTag_1$ from responder and Responder device V will receive $MACTag_2$ from device D and both will verify that the recieved $MACTags$ are equal to corresponding calculated $MACTags$ by each device. Now if this verification is successful each device knows that the other device has computed the correct link key as shown in Figure 2.4.

2.1.2 Communication steps in SKKE protocol

SKKE protocol can be implemented in four major communication steps as are described in ZibBee specification [4] as shown in Figure 2.5.

SKKE-1

Initiator U will send the Challenge QEU and wait for the Challenge QEV from re-

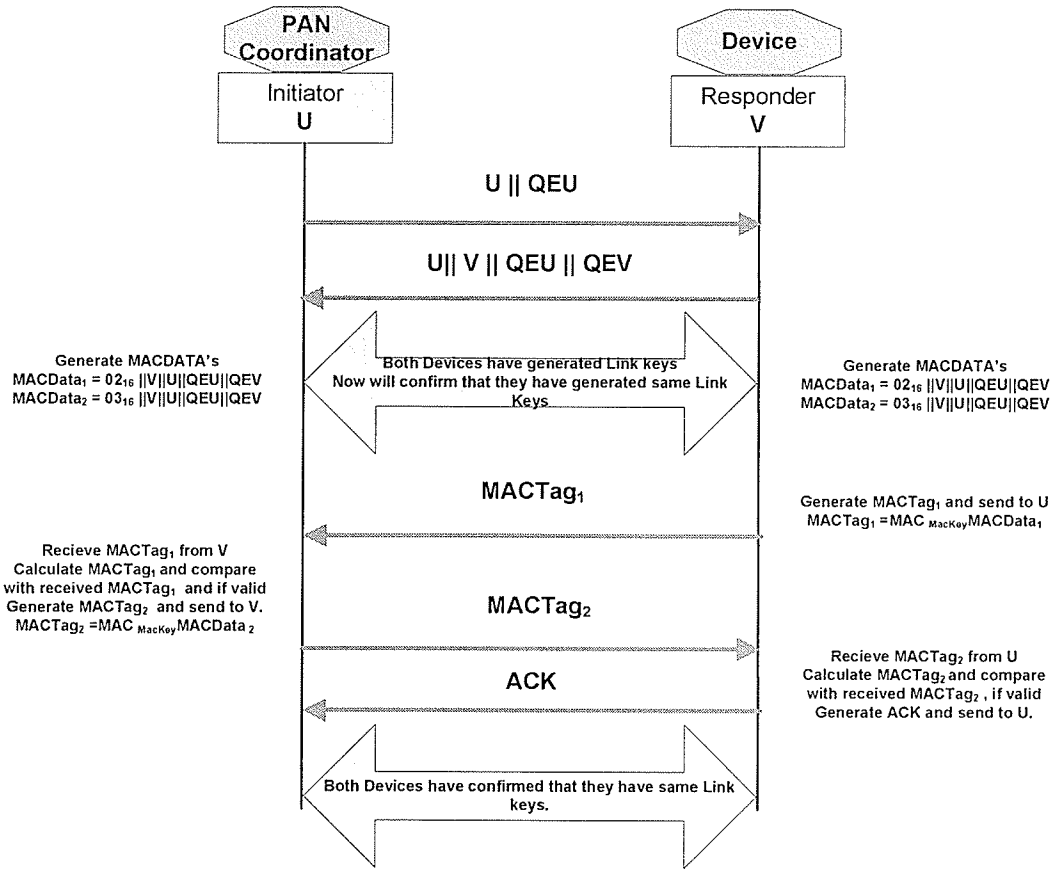


Figure 2.4: Confirmation of Link Keys

sponder V.

SKKE-2

Responder V will receive the Challenge QEU from initiator U, calculates its QEV and in the same data packet will send the $MacTag_1$.

SKKE-3

Initiator will verify the $MacTag_1$ and if it is verified successfully, will send its $MacTag_2$. Now the initiator has a Link key but will wait for an acknowledgment that its $MacTag_2$ has been validated by the Responder V.

SKKE-4

Responder will receive and validate the $MacTag_2$ from the Initiator. If $MacTag_2$ validated successfully, the responder will send an acknowledgment and now both Initiator and Responder have Link keys. Once initiator receives this SKKE-4 message, keys establishment is complete and now regular secure communication can proceed using Link key among the initiator and the responder.

2.2 Asymmetric Key Exchange Model

Public key technology is a widely used tool to support symmetric key management in the realm of Internet hosts and high-bandwidth interconnections. In the past, the constraints of sensor networks have fostered a belief in some researchers that many public key security techniques are computationaly heavyweight for sensor networks and new alternatives must be developed. Although the low computational complexity of symmetric key based authentication schemes are advantageous, but leads to a significant amount of overhead for communication and temporary storage. The common perception of public key cryptography is that it is complex, slow and power hungry, and as such not at all suitable for use in ultra-low power environments like wireless sensor networks. Recently, public key cryptography, in particular its low power form

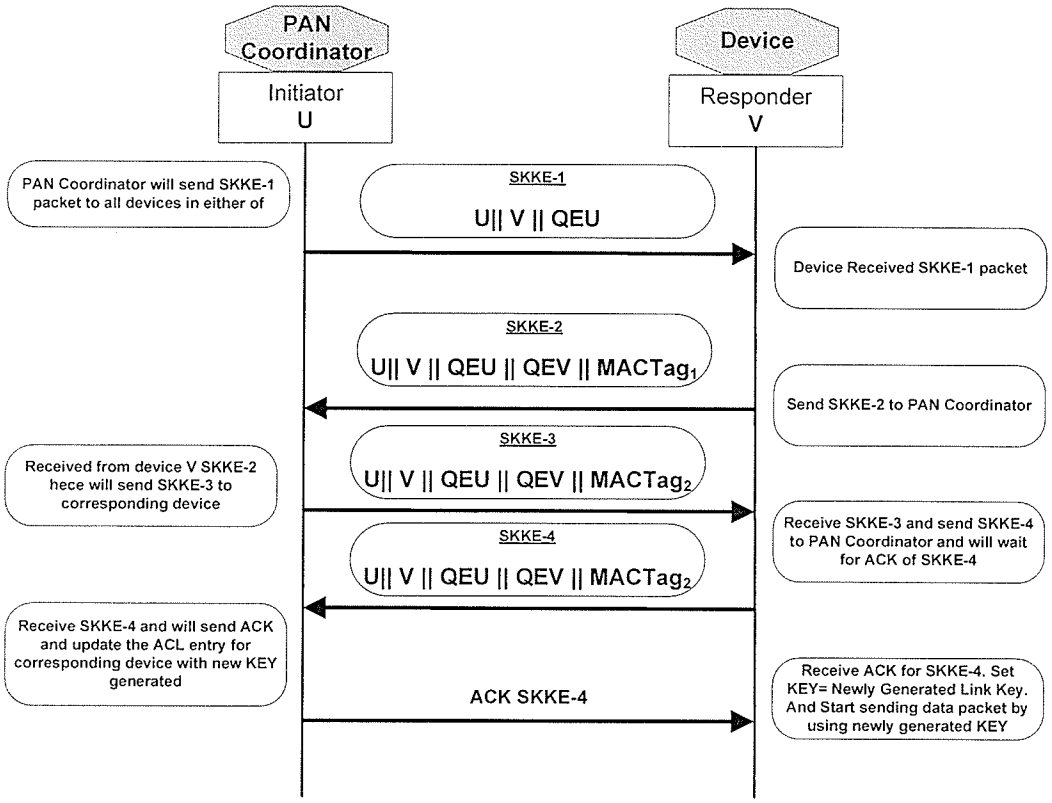


Figure 2.5: Communication Steps in SKKE protocol

Elliptic Curve Cryptography (ECC) [17], is being considered as a viable option for securing wireless sensor networks [5]. ECC [17] offers equivalent security as the better known RSA [7], but with much lower key sizes: for example, the widely accepted security level of RSA with 1024-bit keys can be matched by ECC with 160-bit keys.

2.2.1 Elliptic curve cryptography

At the foundation of every public-key cryptosystem is a hard mathematical problem that is computationally difficult to solve. The relative difficulty of solving that

problem determines the security strength of the corresponding system. Since the best known algorithms to attack ECC have fully exponential run times but the best known algorithms to attack RSA have sub-exponential run times. Smaller keys result in faster computations, and lower power consumption as well as memory and bandwidth savings making ECC especially appealing for resource-constrained environments. More importantly, the performance advantage of ECC over RSA increases as security needs increase over time.

According to [32] 160-bit ECC key-exchange operations are 13 times faster than 1024-bit RSA decryption operations on the mote, but 224-bit ECC operations are almost 38 times faster than 2048-bit RSA decryption operations. ECC operates on a group of points on an elliptic curve defined over a finite field. Its main cryptographic operation is scalar point multiplication, which computes $Q = k \cdot P$ (a point P on an elliptic curve multiplied by an integer k resulting in another point Q on the curve). Scalar multiplication is performed through a combination of point additions and point doublings.

The security of ECC relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). Elliptic curves suitable for cryptography represent some Galois field $\mathbb{F}$ over sets p or 2^p where p is prime. General form of elliptic curve equation is $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ where $a_i \epsilon \mathbb{F}$ [17]. Implementation over $\mathbb{F}_{2^p}$ is reported in [22] while implementation over $\mathbb{F}_p$ is reported in [15].

In ECC, a large random integer k acts as a private key, while the result of multiplying the private key k with the curve's base point G serves as the corresponding public key. The Elliptic Curve Diffie Hellman (ECDH) key exchange and the Elliptic

Curve Digital Signature Algorithm (ECDSA) are elliptic curve counterparts of the well-known Diffie-Hellman and Digital Signature algorithms, respectively. In ECDH key agreement, two communicating parties A and B agree to use the same curve parameters. They generate their private keys $k \cdot A$ and $k \cdot B$, and corresponding public keys $Q \cdot A = k \cdot A \cdot G$ and $Q \cdot B = k \cdot B \cdot G$. The parties exchange their public keys and each party multiplies its private key and the other's public key to arrive at a common shared secret $k \cdot A \cdot Q \cdot B = k \cdot B \cdot Q \cdot A = k \cdot A \cdot k \cdot B \cdot G$.

Hardware platforms which support elliptic curve scalar point multiplications(spm) are also rapidly evolving since general purpose 8-bit architectures are not very suitable for large integer based arithmetic operations. Recently, 16-bit microcontroller Tmote_sky has appeared [26] that consumes minimal energy of 2.4mA during computation while its radio is off. New hardware co-processor for ECC scalar point multiplication was proposed offering energy consumption on 8-bit architecture of 1mJ per multiplication [6]. Nevertheless, even with all the improvements, ECC scalar point multiplication is considered as computationally most expensive operation in the ECC based key exchange algorithm.

First computational evaluation of ECC workload has been reported in [14]. Evaluation of computational complexity of public key ECC algorithms on 8-bit Atmel ATmega CPU architecture reported in [22] and [32] indicate almost 40% reduction of computational time compared to RSA algorithms. In [27] ECC cryptography was evaluated on 8-bit and 16-bit CPU architectures and it was reported that ECC signature generation and verification is 4-5 times less costly than RSA on 16-bit CPU architectures. Key exchange communication costs of single isolated node were re-

ported as well in [27]. However, to the best of my knowledge there was no analysis of integrated communication and security functions in multi-cluster wireless sensor network.

ECC is approved by the National Institute of Standards and Technology (NIST) for U.S. Government use and several standards organizations, including the American National Standards Institute (ANSI), the Institute of Electrical & Electronic Engineers (IEEE), the Open Mobile Alliance (OMA) and the Internet Engineering Task Force (IETF), have ongoing efforts to include it as a required or recommended security mechanism.

2.2.2 Simplified Secure Socket Layer (SSL) protocol

The first asymmetric algorithm considered for my research is a simplified version of Secure Socket Layer (SSL), as proposed in [32]; it is schematically shown in Fig. 2.6. In this key exchange algorithm, authentication is achieved using digital certificates while key exchange is accomplished through variations of Elliptic Curve (cryptography) Diffie-Hellman (ECDH) key exchange algorithm. For signatures, the Elliptic Curve Digital Signature Algorithm (ECDSA) [17] is used by employing ECDH algorithm with 160 bit keys. This scheme assumes that each node is pre-loaded with the ECC base point G, unique private key k_n and public key $K_n = k_n \cdot G$. Node's digital ECC certificate contains public key K_n signed by the trusted Certificate Authority (CA) and its public key. The size of the ECC signature is twice the key size, hence the certificate can be transmitted in one 802.15.4 packet which has a maximum size of 126 bytes. Therefore, only two cycles of handshake suffice to exchange challenges

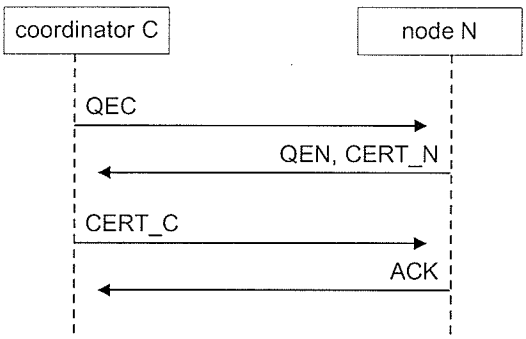


Figure 2.6: Symmetric-key establishment algorithm based on simplified version of SSL.

and certificates. When certificates are exchanged between node and coordinator, two ECC-spms are needed for verification of a digital signature and symmetric shared key is computed as $G \cdot k_n \cdot k_c$ which requires one ECC-spm. Key update can be performed only if coordinator has a number of certificates signed by a CA. It is assumed that the key exchange protocol between a node and the coordinator is initiated by the coordinator after every n_k data packets exchanged between them.

2.2.3 Scaled SSL with ephemeral ECDH algorithm

Another asymmetric key exchange algorithm, deploys a moderately simplified version of SSL with ephemeral ECDH. In this case certificates contain public keys for signatures. Parameters for ECDH and ephemeral ECDH keys need to be sent separately, signed by the appropriate signature private key. Key exchange will be completed by exchange of the hash of the master secret and all previous messages. Public key computation, shared key computation, and digital signature generation will require one ECC-spm each. Therefore, one key exchange cycle will have two ECC-spms for key

computations, two digital signature verifications with two ECC-spms each, and one digital signature generation.

Following is the detailed explanation of communication protocol for key exchange using ephemeral ECDH key authentication protocol. SSL with ephemeral ECDH is presented in Figure[2.7].

In the first two steps, coordinator and the node exchange certificates with signature keys and challenges.

1. Coordinator send certificate to node with signature key and challenge.

2. Node respond with its certificate and signature key and challenge.

   Certificates with signature keys need to be exchanged only once in the lifetime of the node, but new challenges need to be exchanged each time when link key needs to be updated.

3. Coordinator sends ECC base point encrypted with previous link key $k_{l_{i-1}}$ (here i denotes index of key exchange cycle).

   In the first key exchange when link key is not known, this message should be encrypted by the public key from the certificate *Private signature keys* are denoted as k_{sn} and k_{sc} for node and coordinator respectively. This step is optional, since ECC base point may be pre-installed in coordinator's certificate, however it can improve security in some critical applications.

4. Now node and coordinator generate their random private keys k_n and k_c respectively as well as their public keys $K_n = G \cdot k_n$ and $K_c = G \cdot k_c$.

5. Now node will send its public key K_n signed with node's signature key k_{sn} that will decrypted by Coordinator from public key from node's certificate received earlier.

6. Similarly coordinator will also send newly generated public key K_c signed with its signature key k_{sc} to prevent man in the middle attack.

7. Once both node and coordinators receive peer's public keys i.e K_n and K_c each will compute shared link key as $k_{l_i} = G \cdot k_n \cdot k_c$

8. Key Exchange will be completed by exchange of the hash of link key and all previous messages from node to coordinator. $h(k_{l_i}||k_{l_{i-1}}||messages)$

9. Key Exchange will be completed by exchange of the hash of link key and all previous messages from coordinator to node. $h(k_{l_i}||k_{l_{i-1}}||messages)$

2.3 Link Key Update

Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relations between authorized parties. Key management is simplest when all keys are fixed for all time. The time period over which these keys are valid for use is limited because use of same key may result in giving enough information relating to a specific key for cryptanalysis and also may expose network traffic in case of compromise of single key.

Depending on the severity of the threat environment, it is possible that a node or Link key is some how compromised by an adversary and can send false data to the

PAN coordinator. Key update provides an automated mechanism for restricting the amount of data which may be exposed when a Link key is compromised. Sound security policies regarding transparent key updates is fundamental component of sound security practices. But key updates protocol depends on the key update overheads and threat environment under which network is working. Hence controlling the life time of keys and determination of how the key update occurs is a challenging task in any network. Approaches for key updates in general wireless networks mainly target network that have group key structures and have high communication bandwidth [33, 34]. For resource scarce IEEE 802.15.4 networks these key updates will effect the performance adversely.

In this work, I assume that PAN coordinator maintains a counter for each node that keeps track of the number of packets exchanged under the same key. Once the counter reaches to threshold value n_k for any device, the PAN coordinator will initiate the key exchange with all the devices in the cluster. During the key exchange, all devices will temporarily stop the data transmission and resume it once they acknowledge the new link key exchange is complete. Alternative approach will be to use the single counter for all the devices. However, this approach may open the security hole for denial of service attack by single corrupted device.

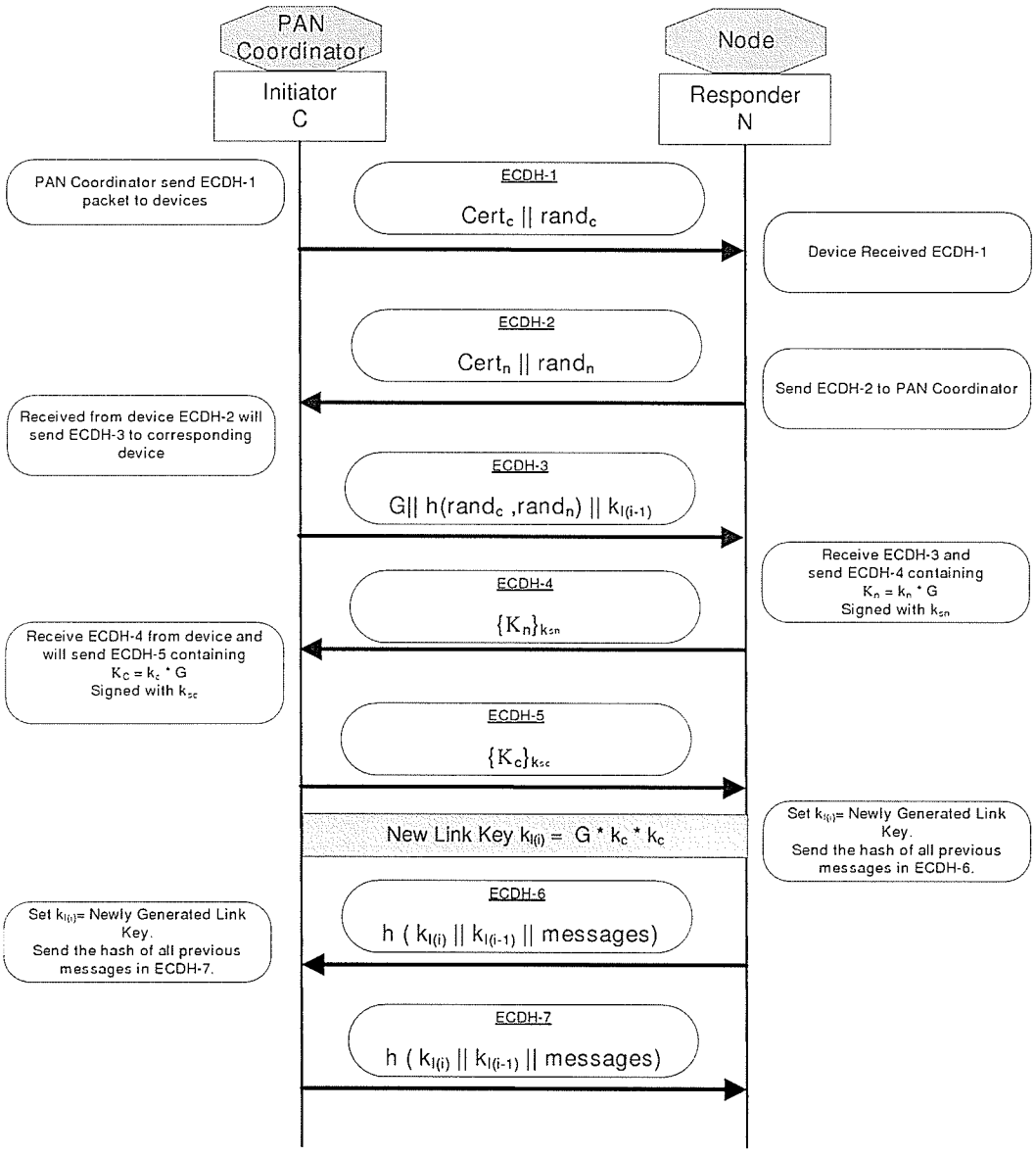


Figure 2.7: Scaled SSL with ephemeral ECDH

Chapter 3

Network lifetime, Power management and Cluster interconnection

Wireless sensor network has the significant benefit of extending the computation capability to physical environments where human beings can not easily reach. These resource limited devices can only be fully utilized if all the resources are utilized optimally. The main power consumptions in sensor networks are computation and communication between sensor nodes. In particular, the ratio of energy consumption for communication and computation is typically in the scale of 1000 [20]. Therefore it is critical to enable collaborative information processing and data aggregation to prolong the lifetime of sensor networks. For using a wireless sensor network for larger area and deployment scenarios scalability, within a network can be achieved by using clustering. I will also discuss the techniques used to extend the network life time

where sensor nodes have limited energy . In the following sections I will discuss about the techniques used for improving the lifetime of a cluster by utilizing activity management of sensor nodes and communication of aggregated data to sink cluster.

3.1 Cluster interconnection and Bridge operation

The network is formed by multiple clusters interconnected in a master-slave regime wherein the coordinator of a 'lower' cluster acts as the bridge to the 'upper' one, and the coordinator of the topmost cluster acts as the network sink; a sample configuration with three clusters is shown in Fig. 3.2.

Consider the network shown in Fig. 3.2. All clusters operate in beacon enabled, slotted CSMA-CA mode under the control of their respective cluster (PAN) coordinators. In each cluster, the channel time is divided into superframes bounded by beacon transmissions from the coordinator [3]. All communications in the cluster take place during the active portion of the superframe, the duration of which is referred to as the superframe duration SD. The superframe is divided into 16 slots of equal size, each of which consists of $3 \cdot 2^{SO}$ unit backoff periods. In clusters operating in the ISM band at 2.4GHz, the duration of the unit backoff period is $0.32ms$ for a payload of 10 bytes, which results in the maximum data rate of 250kbps. The variable SO(Superframe Order) determines the duration of the superframe, and its default value of $SO = 0$ corresponds to the shortest active superframe duration of *aBaseSuperframeDuration*=48 unit backoff periods. The time interval between successive beacons is $BI = aBaseSuperframeDuration \cdot 2^{BO}$, where BO(Beacon Order) can take a value between 1 and 14. Data transfers in the uplink direction use CSMA-CA

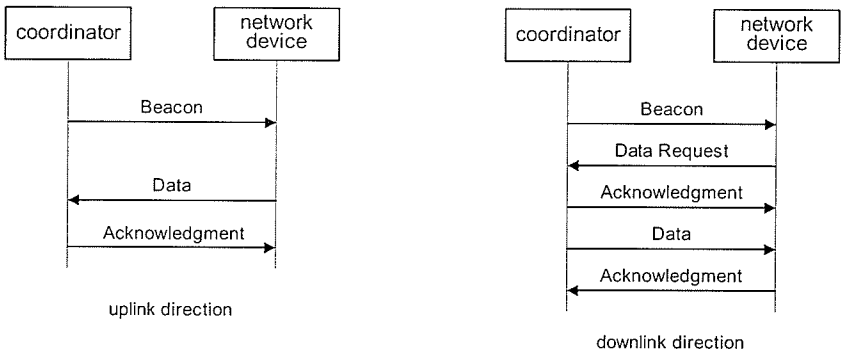


Figure 3.1: Data transfers in 802.15.4 PAN in beacon enabled mode.

algorithm aligned to the backoff period boundary. Data transfers in the downlink direction use a more complex protocol wherein the coordinator announces the presence of a packet, which must be explicitly requested by the target node before being actually sent, as shown in Fig. 3.1. Both data request and data packet transmissions use CSMA-CA, and both must be acknowledged immediately upon successful reception.

During the inactive portion of the superframe, active devices enter a low power mode to save energy while the cluster coordinator switches to the upper cluster in order to synchronize with its superframe and deliver the data to the upper cluster coordinator (i.e., it performs the bridging function). As all clusters use CSMA-CA access, the bridge has to compete for medium access with ordinary nodes from the upper cluster. As soon as the data is delivered, the bridge can return to its own cluster. If the bridge is unable to transmit its data when the superframe in the upper cluster ends, it will freeze its backoff counter and leave the upper cluster; the backoff count will be resumed in the next visit to the upper cluster. Upon returning to the lower cluster, the bridge transmits the beacon denoting the beginning of the next super-

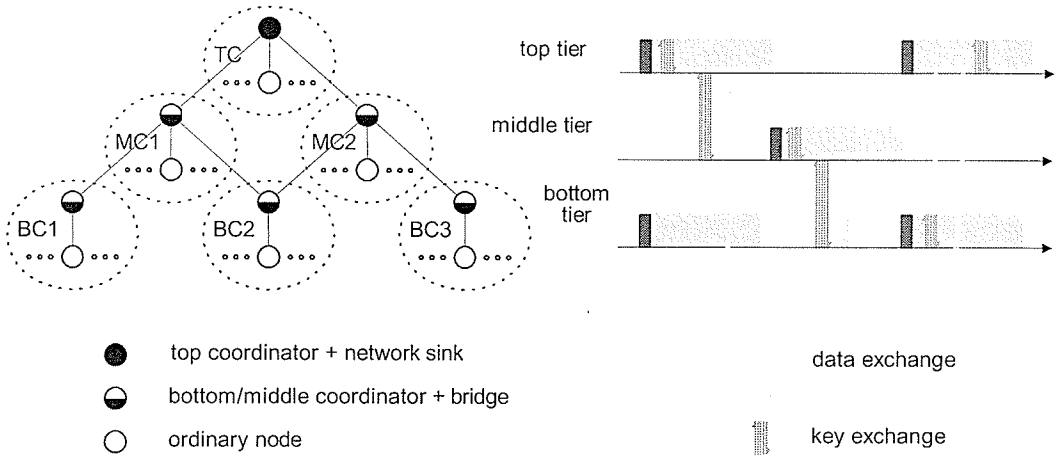


Figure 3.2: Network topology and data/key exchange timing.

frame, and the lower cluster continues to operate. Since the 802.15.4 standard uses 16 channels in the ISM band, interference between the clusters and bridge operation can be resolved by combination of channel selection and bridge time switching. Channel selection can be done using frequency planning concept from cellular networks [28] with channel reuse factors of 1/7 or 1/12, in which case all clusters may operate with the same beacon interval, but adjacent clusters (bottom-middle or middle-sink) have interleaved active and inactive superframe periods, as in Fig. 3.2.

3.2 Power management

Power management consists of adjusting the frequency and ratio of active and inactive periods of sensor nodes [31, 29]. For IEEE 802.15.4 nodes optimal power utilization can be accomplished through scheduling of their active and inactive periods. Because during active periods the nodes utilize more energy thus much should

be achieved during the active period. IEEE 802.15.4 standard support one inherent technique that can be exploited to achieve the objective of energy management. The interval between two beacons is divided into active and inactive periods and any sensor can switch to low-power consumption mode during the inactive mode. We can achieve the target of optimized utilization of energy within a cluster by scheduling the activity management of individual sensor node's active and inactive periods. Coordinator periodically broadcasts the required event sensing reliability R and the number of nodes which are alive. Based on that information, individual nodes calculate average period of sleep between transmissions. In order to prevent packet collisions due to synchronous node awakening, sleep periods are randomized around the calculated average value. When a node wakes up with a packet to transmit, it turns its receiver on to synchronize with the beacon, and subsequently transmits the packet. Upon successful transmission (as indicated by the appropriate acknowledgment), the node immediately goes to sleep. Also, if a node has no packet to send upon wake-up, it will immediately start a new sleep period.

The other approach is by determining a minimum threshold data transmission required for application supported by the underlying sensor network. The minimum threshold data transmission required for the application is also referred as the event sensing reliability (Number of packets persecond needed for reliable event detection) [29]. During network initiation redundant sensor nodes are used to achieve the desired event sensing reliability. We assume that individual nodes sleep for a random time interval, the duration of which is a geometrically distributed random variable regulated with probability P_{sleep}. When the node wakes up, it waits for the beacon from

the coordinator before it attempts to transmit the packet. We have used bernoulli scheduling for the packet scheduling during the active period of the node [24]. In this approach, at the end of each packet transmission the node checks its uplink buffer and incase the buffer is empty the node goes to sleep mode otherwise if buffer contains any packet for transmission, the node tries to send the packet with probability P_{active} or goes to sleep with probability 1-P_{active}. Therefore, two control parameters are needed: one, P_{sleep}, that regulates the duration of inactive period; and other P_{active} regulates the duration of active period. Once any node cease to function wither because of battery exhaustion or for any other reason, the remaining nodes will have to extend their activity to achieve the required reliability. To meet the goal of required reliability during the life of network the calculation of probabilities for active and inactive periods among nodes can centralized or distributed. In centralized approach we assume that network coordinator is always aware of the number of sensor nodes in the cluster along with the packet arrival rate rates (this can be obtained as simple long-term averages, as packet header contain the source node address). The coordinator first determines the node utilization based on the number of live nodes and then calculates the individual reliability r per node as $r = R/n$ where R is the required collective reliability and n is number of live nodes. The coordinator broadcasts r within the beacon frame for the sensors within its cluster. Coordinator will broadcast updated values of individual reliability every time any sensor node dies and this value of r will grow with the death of additional sensor node within the cluster. Note that the sleep time is geometrically distributed, and mean sleep time is $t_{boff}/(1 - P_{sleep}) = 1/r$ where $t_{boff} = 0.32ms$ corresponds to the duration of one backoff period. Therefore,

each sensor node starts with $P_{sleep} = 1 - rtboff$ and $P_{active} = 0$. Activity management in multiclustered network is modeled by considering overall required reliability of the network where each cluster has to deliver R packets per second towards the sink. Since all traffic is conveyed to the sink the amount of packets per second is equal to R and $2R$ in source, and sink cluster respectively. Contention caused by bridges will consume more energy and has to be compensated by larger number of nodes. Moreover, bridges will need to execute key exchange protocol more frequently than ordinary nodes which will consume even more energy resources.

For evaluating the power consumption of secure operation of IEEE 802.15.4 compliant sensors, I assumed that each sensor node is powered by two AA batteries which supply voltage between 2.1-3.6 V and 500 mAmp-hours as required by Tmote_sky [26] operating conditions with total energy $b = 10260J$. The network operates in the ISM band at 2.45GHz, with raw data rate 250kbps and $BER = 10^{-4}$. Superframe size is controlled with SO= 0, and BO= 1. The packet sizes are fixed at 12 backoff periods, while the device buffer has a fixed size of $L = 2$ packets. The packet size includes Message Authentication Code and all physical layer and Medium Access Control protocol sublayer headers, and is expressed as the multiple of the backoff period [3]. Physical layer header is 6 bytes long, and the Medium Access Control sublayer header and Frame Check Sequence fields have a total of 9 bytes. Other parameters from Medium Access Control layer are kept at default values.

Power consumption of communications and cryptographic operations Tmote_sky [26] ultra low power IEEE 802.15.4 sensor module uses 8 MHz, 16 bit RISC micro-

controller MSP430 F1611, with maximum current consumption during computation of 2.4mA (i.e., with the radio turned off). Under the supply voltage of 3V (two AA batteries), this results in power consumption of 7.2mVA. Energy consumptions per backoff period (10 bytes) under typical operating conditions in the ISM band are $\omega_s = 18.2$nJ, $\omega_r = 17.9\mu$J and $\omega_t = 15.8\mu$J, during sleep, receiving and transmitting (at 0dBm), respectively.

Energy consumption of cryptographic operations was based on the data for the TelosB mote [18][27] scaled by a factor of 0.6, since TelosB uses the same microcontroller as Tmote_sky, but consumes 4mA. Therefore, an ECC scalar point multiplication takes 0.5s and consumes 3.6mJ. As a result, signature generation should take 0.52s and 3.75mJ, while signature verification should take 1.02s and 7.44mJ, based on values given in [27]. For message authentication code, we adopted the HMAC algorithm. Energy consumption of hashing is obtained by scaling results from [32], again using relationships between TelosB and MICA2DOT reported in [27], as well as further scaling between TelosB and Tmote_sky. Thus, the energy consumption of the calculation of a SHA-1 hash value is 0.814μJ/byte.

Chapter 4

Simulation Model and Results

My methodology for this research follows standard practices for conducting a performance modeling analysis by constructing the networking model, validating the model by simulation, varying modeling parameters, and analyzing the results. However, it is important to accommodate the unique characteristics of security protocols in implementing the details of each step. There were three high-level steps: developed a network model that use Cryptographic security methods provided by the IEEE 802.15.4 protocol but also uses such methods under standard key management techniques as described in Chapter 2, the next step involved simulating the IEEE 802.15.4 compliant network environment in an object oriented fashion where basic objects (e.g IEEE 802.15.4 medium, compliant sensor node, PAN coordinator and network statistic collectors) were implemented. Each object has its internal state that can change either due to internally generated event or receipt of a message generated from another object. Once these basic module of objects were tested for validity more complex algorithms (e.g sleeping policy for energy management, key updates

and bridging mechanism for inter-cluster communication) were introduced in the simulation implementation. Finally after determining a run-up period of the simulation different performance measures were evaluated by adjusting input parameters. These input parameters (e.g Network size, Packet arrival rate, energy resources of a node Symmetric and Asymmetric Key exchange protocols etc) were used to reflect conditions we would expect in the real world and explore the protocols sensitivity to variety in the operational environment.

4.1 Beacon-Enabled IEEE 802.15.4 Simulation Model

A simulation based evaluation technique is used for this research to evaluate secure communication in the IEEE 802.15.4 with key exchange and sleep mechanisms. This chapter describes the models used to simulate different key exchange models with secure communication in an IEEE 802.15.4 network. Artifex [19] simulation platform is used for this research which is a general development platform for discrete event simulations. For the remaining of this section we first give a quick introduction of beacon-enabled simulation model of 802.15.4 [25] and later explain the simulated key exchange and update model and power management used in this research.

The network communication model of this simulation is based on a star topology. The model is built on three primary objects in a cluster: PAN coordinator, Device and Medium. The device and PAN coordinator objects are inter-connected via medium object in our simulation model.

Two different token types are defined that play the role of packet and backoff. Packets can be any of beacon, MAC request, data and acknowledgment (ack) types.

The communication is initiated when PAN coordinator first sends beacon to medium (beacons are sent after every 48t where t is duration of one backoff period). After receiving the beacon the medium starts a clock and sends pulse to all devices.

Data packets are generated by device object following exponential distribution and are destined to a randomly chosen device. The packet is then sent to the medium and a copy of it is kept for retransmission if needed. Data packets are processed through the medium and collusion mechanism defined with medium object will determine whether a collision occurred. If there are no collisions, data packets are sent successfully to the PAN coordinator and the medium status is set to busy for that duration.

PAN coordinator is the next stop for data packets and is responsible for sending ack type packets to corresponding device after a specified delay. All acknowledgement packets will be processed through medium object for determining the collusion and successful packet will be forwarded to corresponding destined device. When PAN coordinator is sending data to a device it keeps finite buffer for each device in the PAN. If the buffer of the device which the data packet is destined for is full, the packet will be discarded. In the case that there is still room in that device's buffer, the coordinator adds the destination ID of packet to the pending devices list and advertises the ID in the beacon. The device will notice that there is packet waiting for it and will initiate a MAC request packet to be sent to the coordinator. The PAN coordinator after receiving the request will perform round robin scheduling algorithm and choose the device to send the packet from its corresponding downlink buffer.

4.1.1 Simulating SKKE Key Exchange Mechanism in IEEE 802.15.4 Network

In this section I will describe the communication between the ordinary nodes and PAN coordinator which occurs as result from the Symmetric-Key key exchange (SKKE) protocol as described in Section 2.1.2. I assume that devices are attached to the cluster and the formation of the piconet is finalized. I also assume that the master keys are established, so that there is no threat of eavesdropping during exchange of master keys.

The process of key generation is initiated by PAN coordinator's advertisement for the first phase of key generation packets. Depending on which stage of generation we are in, the corresponding SKKE type of data packet (ranging from 1 to 4) will be processed (e.g the first data packet has the type of SKKE-1 and so on). According to the standard specification at most 7 devices can be advertised in each beacon. Therefore the PAN coordinator will advertise 7 devices in each beacon. According to the standard, each device listens to each beacon and if its ID has being advertised the device will send a request packet. Request packet is transmitted in CSMA-CA mode and can collide with other packets. If it is received successfully by the PAN coordinator it will be acknowledged and downlink packet transmission carrying the SKKE protocol data will follow in the downlink transmission.

In our model, key exchange packets have non-preemptive priority over data packets. If the node has started backoff process for data packet and it hears its ID in the beacon it will finish the current packet transmission before sending the request packet. However, if data packet arrives to the device's buffer while the key exchange

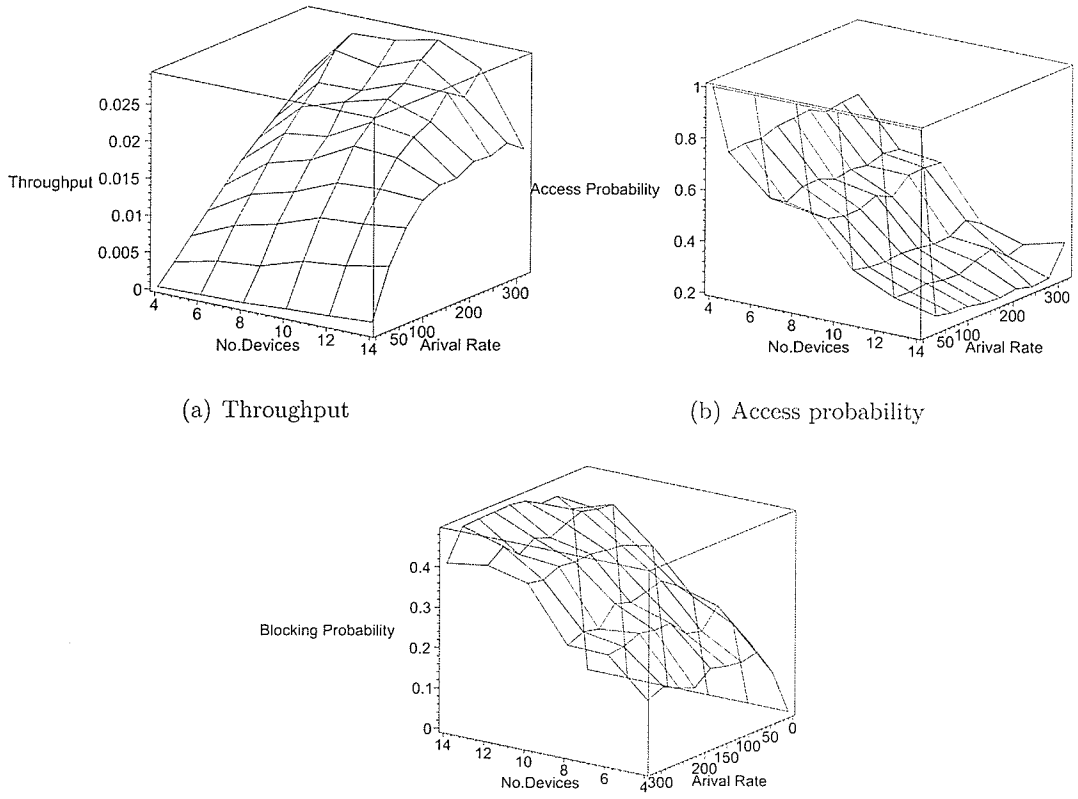
is going on, its transmission will be postponed until device receives the new link key. PAN-Coordinator will first check key for the destination device from its access control list and no packet will be sent to the specific destination until the corresponding link key is already exchanged between PAN coordinator and the node. From this point on regular secure data packets will be immediately send to the destination.

4.1.2 Simulation Run and Analysis for SKKE protocol

I have implemented the physical, data link and security layers of an IEEE 802.15.4 cluster operating in beacon enabled, slotted CSMA-CA mode. The packet size without security overheads includes all physical layer and Medium Access control layer headers, and it is set to 30 bytes i.e. to three backoff periods. When packet signature (message authentication code) of 16 bytes is added to the total packet size had to be rounded to 5 backoff periods (the largest packet size could be set to 13 backoff periods).

The cluster under consideration contains 14 devices, each having buffer capacity for three packets. Packet arrival per device followed the Poisson process with average rate of 90.5 packets per minute. When the coordinator announces key exchange in the beacon, all nodes had to temporarily stop uplink data transmissions until they receive new key initialization values from the coordinator in the downlink packets. Due to complex downlink data-link transmission algorithm we expected that key exchanges will adversely affect the regular sensing traffic.

I considered the impact of the increase of packet size due to addition of Message Authentication Code, increased processing time needed for encryption in AES with



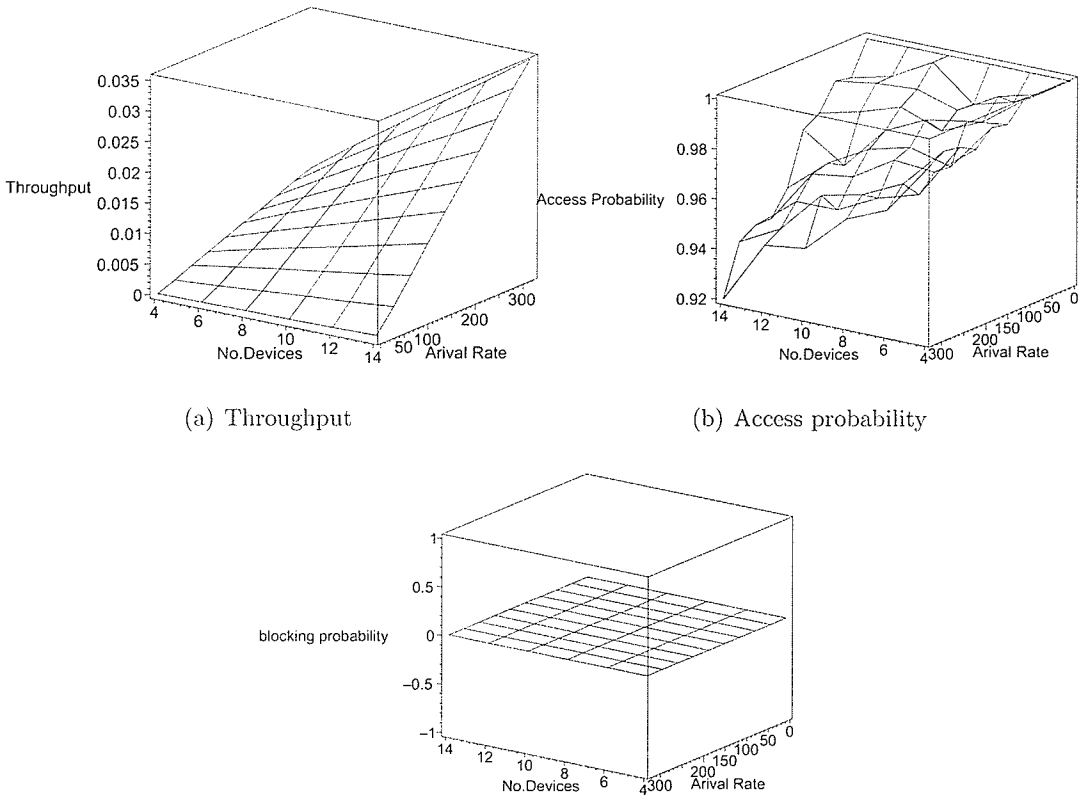
(a) Throughput

(b) Access probability

(c) Blocking Probability

Figure 4.1: Throughput, Access Probability and Blocking Probability as the function of simulation time (backoffs) for the case when security is employed and all devices stop their communications to update their keys.

CBC-MAC, and key exchange between the nodes over various packet arrival rates and cluster sizes. Fig. 4.1 presents throughput, access probability (probability of no packet collision) and blocking probability at the node's buffer when all security overhead is included. Results were taken for varying number of nodes and varying packet arrival rate per node. Fig. 4.2 presents the same parameters (except the key exchange cost since it does not exist) when no security measures are deployed in the



(a) Throughput

(b) Access probability

(c) Blocking probability

Figure 4.2: Throughput, Access Probability and Blocking Probability as the function of simulation time(backoffs) when no security technique is employed.

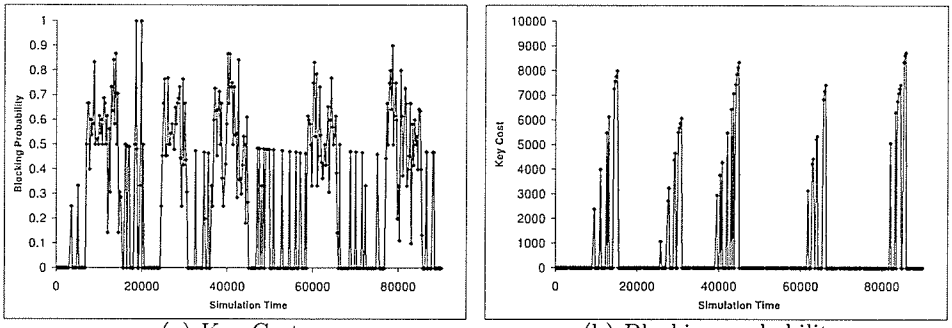
network. We observe that without security measures, blocking probability is equal to zero i.e. that network works without losses.

The experiment to measure the cost of key update in the cluster contains seven devices only, and therefore it was possible to advertise the keys for devices in a single beacon. All devices temporarily stopped their data transmission during the key exchange. The behavior of the cluster over time is presented in Fig. 4.3. Fig. 4.3(a) shows number of backoff periods spent in key exchange. We notice that average cost

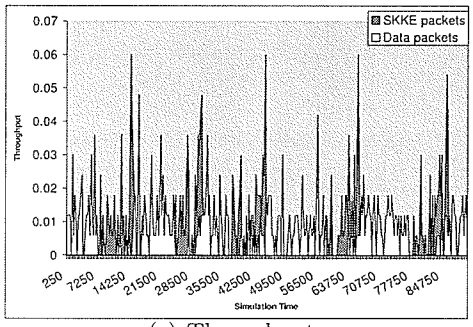
of key exchange is slightly below 2000 backoff periods, which gives 250-270 backoff periods per device. Knowing that the key exchange involves a total of two downlink (uplink request + downlink data) transmissions and three uplink transmissions, we conclude that one CSMA-CA access takes approximately 40 backoff periods. Given the backoff window sizes of (8, 16, 32) we conclude that transmission commences in average after third backoff attempt which indicates moderate to large activity over the medium. The blocking probability at individual sensor node buffer over the snapshot periods is shown in Fig. 4.3(b). Due to large periods when device transmission is prevented during key exchange (well over 1500 backoff periods), the blocking probability increased to values between 0.7 and 1. When the key exchange is finished, normal data communications resume. As a result, the blocking probability drops abruptly to values around 0.3 and slowly declines further as the backlogged packets clear.

Fig. 4.3(c) shows the throughput values measured during snapshot intervals of 250 backoff periods. The throughput of data packets is shown in white, while the throughput of key-exchange packets is shown in black. According to the throughput results reported in [25], the observed network regime without key exchange is slightly below the saturation condition (in saturation condition, all data transmissions end up in collisions).

We have also implemented distributed activity management in our simulator, assuming that the battery for each node has a fixed capacity. Battery capacity, which is expressed in backoff periods, is decremented by one for each backoff period in which the radio subsystem is active.



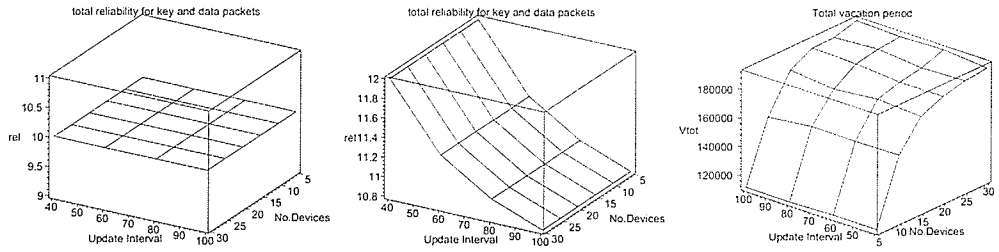
(a) Key Cost (b) Blocking probability



(c) Throughput

Figure 4.3: Key Cost, Blocking Probability and Throughput as the function of simulation time (backoffs).

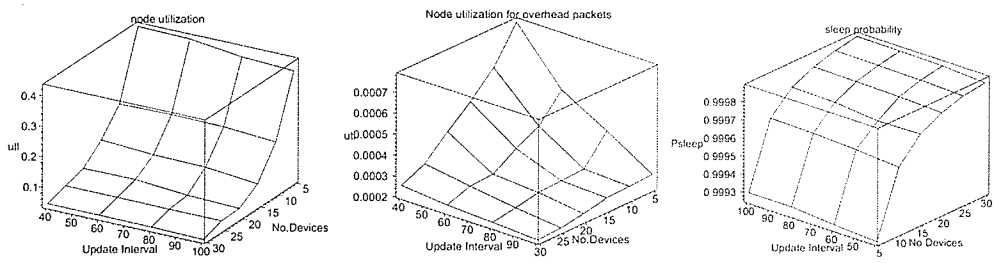
I have varied the key exchange threshold (n_k) between 40 and 100 packets while the requested event sensing reliability was kept at $R = 10$ packets per second. Cluster size (n) was varied between 5 and 30 nodes. I have assumed that the network operates in the ISM band at 2.45GHz, with raw data rate 250kbps. The packet size was fixed at twelve backoff periods, and the device buffers have a fixed size of three packets. The packet size includes Message Authentication Code and all physical layer and Medium Access Control protocol sublayer headers, and is expressed as the multiple of the backoff period [4]. I also assume that the physical layer header has 6 bytes, and that the Medium Access Control sublayer header and Frame Check Sequence fields



(a) Number of data packets transmitted per second.

(b) Number of key and data packets transmitted per second.

(c) Inactive period out of total simulation time.



(d) Total utilization.

(e) Utilization due to key exchange.

(f) Sleep probability.

Figure 4.4: Event sensing reliability for data including key exchange packets, inactive period, total utilization and utilization for key exchange packets, average number of active devices and sleep probability for a node.

have a total of 9 bytes.

Fig. 4.4(b) shows total number of successfully transmitted packets (including key and data information) transmitted per second for requested data reliability of $R = 10$ packets per second (which is shown on Fig. 4.4(a)). It is noted that the total number of packets hyperbolically grows when the key exchange threshold decreases linearly Fig. 4.4(b). This is intuitive since the frequency of key updates is R/n_k per second and number of overhead packets with key information per second is equal to $8R/n_k$.

We note that key exchange overhead becomes negligible only for $n_k \geq 90$. Probability that packet will not suffer from collision or noise error sharply drops when threshold for key exchange drops below 40 packets. Both the reliability overhead and success probability depend only on the requested event sensing reliability except for very small key update threshold. Sleep period, on the other hand, depends mostly on the number of alive nodes and impact of key exchange overhead is barely noticeable.

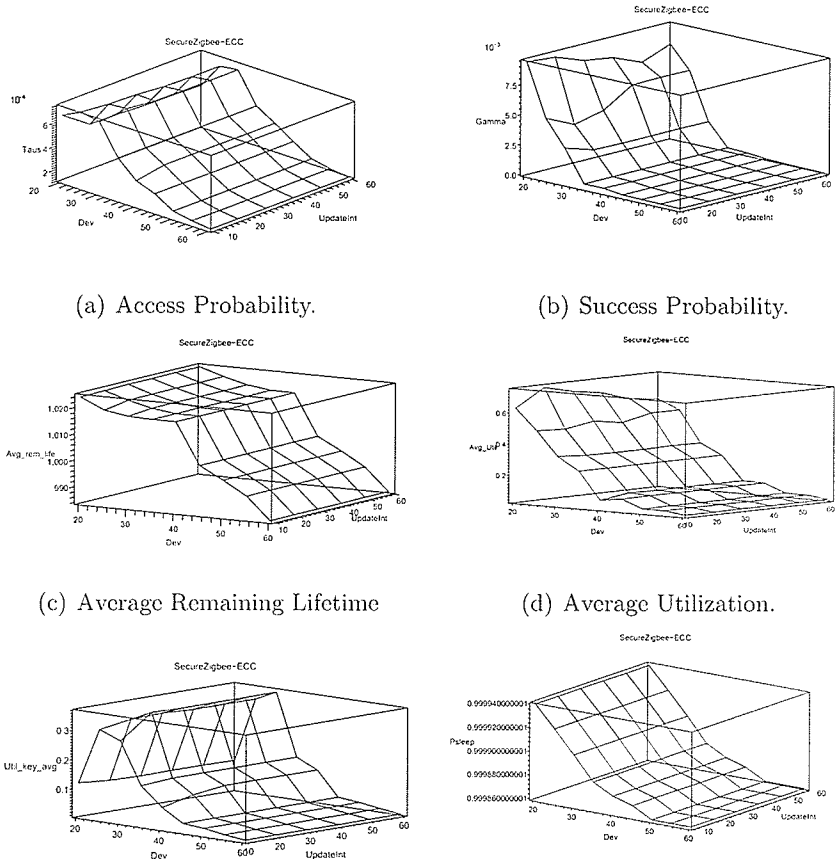
Total node utilization shown in Fig. 4.4(d) depends mostly on the number of alive nodes, but it also increases with increase of the number of key exchanges per second and exact impact of the key exchange overhead is shown in Fig.4.4(e). Finally, sleep probability for each node is shown in Fig. 4.4(f). Sleep probability dominantly changes with n, while the changes with n_k are much milder.

4.2 Simulation model and Results using Asymmetric Key exchange protocols

For evaluating the performance and energy utilization of IEEE 802.15.4 enabled wireless networks with secure data transmission using asymmetric key exchange protocol, I have implemented Elliptic Curve Deffie Hellman key exchange protocol with periodic update of link key. I have simulated a multicluster network in which the clusters are interconnected by a Master-Slave bridge. The cluster kept its functionality and topology same except the PAN coordinator's where PAN coordinator behaves as a bridge as well a node in child cluster in a multicluster environment. The PAN coordinator in the child cluster switches to parent cluster when it has data in its

buffer destined to parent cluster. I assumed that all the sensing traffic from the child cluster occurs in the uplink direction with the parent cluster's coordinator as the ultimate destination except the traffic required for exchanging the link key; the parent cluster has local traffic that is sent from nodes associated within its own cluster. This assumption is valid in sensor networks where most, of the traffic will be directed toward the network sink. I assumed that all the clusters will operate in beacon enabled slotted CSMA-CA mode under the control of their respective PAN-coordinator.

I assumed that each node is powered by two AA batteries which supply voltage between 2.1-3.6 V and 500 mAmp-hours as required by Tmote_sky [26] operating conditions with total energy $b = 10260J$. The network operates in the ISM band at 2.45GHz, with raw data rate of 250kbps. The packet sizes are fixed at 12 backoff periods, while the device buffer has a fixed size of $L = 2$ packets. The packet size includes the Message Authentication Code and all physical layer and Medium Access Control protocol sublayer headers, and is expressed as the multiple of the backoff period [3].Physical layer header is 6 bytes long, and the Medium Access Control sublayer header and Frame Check Sequence fields have a total of 9 bytes. Other parameters from Medium Access Control layer are kept at default values. Requested event sensing reliability per cluster is set to $R = 10$ packets per second. In order to investigate the impact of secure sensing on network performance and energy consumption in multicluster environment I have assumed each scalar point multiplication for each elliptic curve cryptography consumes $3.6mJ$, while encryption, decryption and SHA-1 hash functions consumes $0.25\mu J/Byte$, $0.39\mu J/Byte$, $0.814\mu J/Byte$ respectively. In order to maximize the network lifetime sleep mechanism has been introduced within this



(a) Access Probability.

(b) Success Probability.

(c) Average Remaining Lifetime

(d) Average Utilization.

(e) Average Utilization for key exchange only.

(f) Average Psleep in cluster.

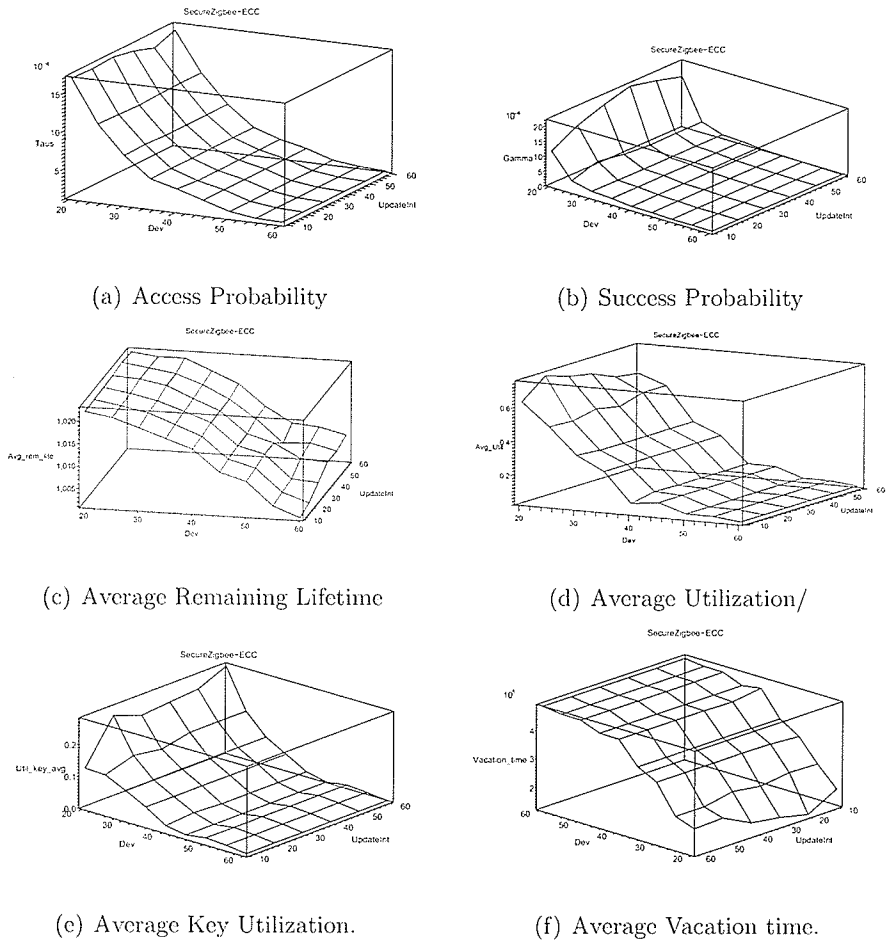
Figure 4.5: Secure Medium behavior (ECDH) in Sink Cluster.

model as well but in order to measure the energy cost within this model I have assumed the energy consumption for node transmission $15.8\mu J/Byte$, receiving $17.9\mu J/Byte$ and for sleeping $18.2nJ/backoff$.

The number of devices n is ranging between 20-60 in both source and sink clusters. The update interval n_k for keys using ECDH protocol is ranged between $10-60$. The

access probability to access the medium in both clusters decreases with the increase in number of nodes within the cluster Figs 4.5(a), 4.6(a). The success probability also shows similar pattern but the success probability is lower in sink cluster Fig 4.5(b) because the medium in sink cluster is also used by the bridge to transfer data from source cluster Fig 4.6(b). The remaining life time of nodes in source and sink cluster is shown in Figs 4.5(c) 4.6(c) that shows the decline in the battery power consumed during the simulation run time. The utilization behaviour of medium during key exchange phase and also during data packet is also calculated. Average medium utilization for sensing and key exchange data in source shown in Figs 4.5(e) 4.5(e) while utilization for source cluster shown in Figs 4.6(e) 4.6(e).

In order to investigate the impact of duration and frequencies of key exchange on network performance we have first fixed the populations in all three clusters to 100 nodes. With this setting we have evaluated different key exchange protocols described in chapter 2. The threshold to start the key exchange between node and coordinator was varied between 100 and 500 packets. Fig. 4.7 shows energies per one sensing-data packet transmission (which includes energy for synchronization with the beacon, energy leak during sleep and energy waste due to packet collisions) as functions of the key-update threshold and type of key exchange algorithm. Numerical results from the source, middle and sink clusters are denoted with the solid line, dotted line and dashed line respectively. Simulation results which represent performance of key exchange protocol based on pre-installed shared secret in all three clusters are marked with crosses. Simulation results which represent performance of very simplified SSL key exchange protocol based on ECC [32] are marked with boxes.



(a) Access Probability

(b) Success Probability

(c) Average Remaining Lifetime

(d) Average Utilization/

(e) Average Key Utilization.

(f) Average Vacation time.

Figure 4.6: Secure Medium behavior(ECDH)in Source Cluster.

Simulation results which represent performance of moderately simplified SSL key exchange protocol with ephemeral ECDH are marked with diamonds. We notice that curves are grouped by cluster, rather than by the algorithm, due to the impact of packet collisions which grows as we approach sink cluster. In bottom and middle clusters there is no significant difference between the energy consumption for different key exchange algorithms while in sink cluster contribution of the complexity of key

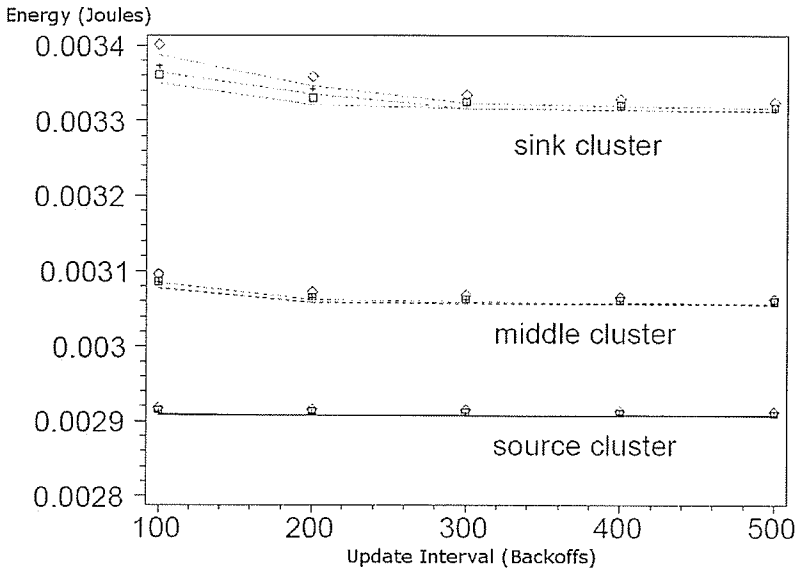


Figure 4.7: Energy consumption for sensing-data packet transmission (which includes energy for synchronization with the beacon, energy leak during sleep and energy waste due to packet collisions) as functions of the key-update threshold and type of key exchange algorithm.

exchange algorithm is notable. Note also that maximum energy consumption per one packet is close to the energy consumption of one scalar multiplication (3.6 mJ) on tmote_sky. Numerical results show slightly lower values than simulation results due to simplified modeling of bridge interactions.

Fig. 4.8 shows cluster lifetimes under three key exchange algorithms. Analytical results from the source, middle and sink clusters are denoted with the solid line, dotted line and dashed line respectively. Simulation results which represent performance of key exchange protocol based on pre-installed shared secret in all three clusters are marked with crosses. Simulation results which represent performance of very

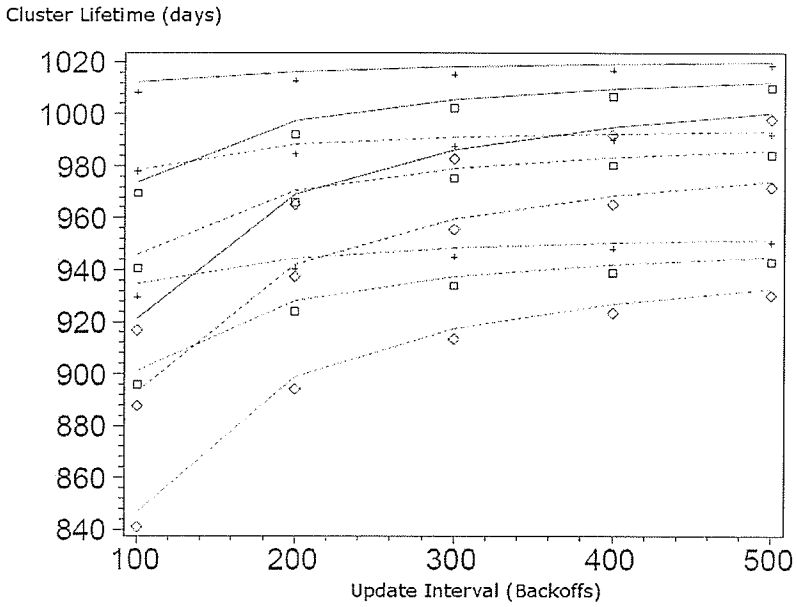


Figure 4.8: Cluster lifetimes in days as functions of the key-update threshold and type of key exchange algorithm.

simplified SSL key exchange protocol based on ECC are marked with boxes. Simulation results which represent performance of moderately simplified SSL key exchange protocol with ephemeral ECDH are additionally marked with diamonds. Analytical results are now slightly higher than simulation ones since energy for communications was slightly underestimated due to simplified bridge modeling. Since neither curve is flat in the range n_k =100-500 this means that communication and cryptographic cost of any of key exchange algorithms can not be neglected. This impact is also exacerbated with increased contention between data and key packets as we move closer to the sink. The distances among the curves from the same cluster level with respect to the type of key exchange algorithm show the impact of communication and

Table 4.1: Equalized populations as functions of the key-update threshold and type of key exchange algorithm.

Cluster	Key exchange scheme	Threshold for key exchange				
		100	200	300	400	500
middle	shared secret	106	105	105	105	105
middle	simple ECC	106	105	105	105	105
middle	moderate ECC	106	105	105	105	105
sink	shared secret	116	115	114	114	114
sink	simple ECC	115	114	114	114	114
sink	moderate ECC	116	115	114	114	114

cryptographic cost of the key exchange algorithm which attenuates as the period of key exchange grows. The distances among the curves from the same algorithm with respect to the cluster level show impact of packet collisions and longer backoff times as well as total traffic passing through the cluster. They are almost parallel since key exchange costs are proportional to amount of traffic passing through the cluster.

In Table 4.1, we present node populations in middle and sink clusters, respectively, which equalize cluster lifetimes (to the lifetime of source cluster) when source cluster has 100 nodes. As can be seen, moderate increases in cluster populations are needed when key exchanges are complex and frequently done.

Chapter 5

Conclusion and Future work

I have developed and simulated key exchange process integrated with reliable sensing and power management in a beacon enabled IEEE 802.15.4 cluster . Data encryption is provided by exchanging link keys between each device and PAN coordinator under symmetric and asymmetric key exchange models. The signature payload plays a big role on performance of the network. I have developed a model of key exchange integrated into the sensing function of beacon enabled IEEE 802.15.4 cluster. The results show important impact of the ratio of the event sensing reliability and key update threshold on cluster's energy consumption. I have evaluated the impact of complexity of different key exchange protocols on the network performance and have shown that energy consumption due to medium access control matches energy consumed by cryptographic operations and that neither of these two factors should be neglected in the analysis. My research results can be helpful in designing WSN in order to choose cluster populations, energy budget per node, key exchange algorithm and period of key exchanges when application parameters such as required event

sensing reliability R, network lifetime and desired level of security are known.

For the future work, I will try to improve my model for measuring the performance of secure IEEE 802.15.4 personal area network by introducing techniques such as the use of guaranteed time slot of MAC frames for exchange key while introducing group key exchange protocols, activity management and key management techniques. I will also compare different network topologies for personal area network and will calculate measures that can enhance the lifetime of wireless personal area network along with defense against expected threats from environment.

Appendix A

Security Building Blocks

Wireless is a shared medium; everything that is transmitted or received over a wireless network can be intercepted in such an environment. An adversary can gain access to information by monitoring the communication among nodes. For example a few wireless receivers placed outside a house might be able to monitor the light and temperature reading of sensor network inside the house, thus revealing detailed information about occupant's daily personal activity. Similarly, attackers can obtain their own commodity sensor nodes and present it as a legitimate node inside the network and once attacker have a few nodes like that in network, attackers can launch different types of attack -for example denial of service, falsification of sensed data, dropping of sensed data etc.

A.0.1 Security Techniques

Different security techniques are employed to safeguard threats of such eavesdropping

A.0.2 Data Confidentiality

All nodes in a sensor networks are communicating through one wireless medium and listening to this medium is easy. Hence network should not leak sensor data to any neighboring network or any node that is not part of the network. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that is carried by the intended receivers only.

A.0.3 Data Authentication

Data Authentication allows the receiver to verify that the data was really sent by the claimed sender. Authentication also prevents an attacker from modifying a hacked device to impersonate another device. Since an adversary can easily inject messages, the receiver needs to ensure that the data used in any decision-making process originates form a trusted source. Data Authentication is usually achieved through symmetric mechanism where sender and receiver share a key to compute Message Authentication Code (MAC). The data is appended along with its MAC and once the receiver gets the data, it recalculates MAC and if same MAC is calculated that it received from same sender that it shares the key. Authentication can be achieved both at cluster level and device level. Cluster level authentication is achieved using a common network key whereas device level authentication is achieved by using unique pair wise keys for each link in the network.

A.0.4 Data Integrity

Data integrity allows the receiver to verify that the data received is same as sent by the sender and is not changed during its way to receiver. If MAC calculated by receiver is same as it received that means that data is not altered during its transit to receiver. Message authentication codes must be hard to forge without the secret key. Consequently, if an adversary alters a valid message or injects a bogus message, she will not be able to compute the corresponding MAC, and authorized receivers will reject these forged messages. In sensor networks usually data integrity is achieved in symmetric fashion and is again relied on the appended Message Authentication code, hence integrity and authentication options allow trade-off between message protection and message overhead.

A.0.5 Replay Protection

An adversary that eavesdrops on a legitimate message sent between two authorized nodes and replays it at some later time engages in replay attack. Since the message originated from an authorized sender it will have a valid MAC, so the receiver will accept it again. Replay protection prevents these types of attacks. The sender typically assigns a monotonically increasing sequence number to each packet and the receiver rejects packets with smaller sequence number than it has already seen.

In symmetric mechanism where sender and receiver share one common key and rely different security techniques on the secrecy of these keys. Hence the whole security model revolves around the secrecy of symmetric keys that can be either at network

level or a link level.

Bibliography

[1] ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry-Key Agreement and Key Transport using Elliptic Curve Cryptography. American Bankers Association, 2001.

[2] FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication 198,US Department of Commerce/N.I.S.T., 2002.

[3] Standard for part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN). IEEE Std 802.15.4, IEEE, 2003.

[4] ZigBee Alliance. ZigBee specification (ZigBee document 053474r06, version 1.0), December 2004.

[5] Benjamin Arazi, Itamar Elhanany, Ortal Arazi, and Hairong Qi. Revisiting public-key cryptography for wireless sensor networks. volume 38, pages 103–105, Los Alamitos, CA, USA, 2005. IEEE Computer Society Press.

[6] Guido Bertoni, Luca Breveglieri, and Matteo Venturi. Power aware design of

an elliptic curve coprocessor for 8 bit platforms. In *PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops*, page 337, Washington, DC, USA, 2006. IEEE Computer Society.

[7] Matt Bishop. *Computer Security - Art and Science.* Pearson Education, Inc., Boston, MA 02116, 1st edition, 2003.

[8] Rolf Blom. An optimal class of symmetric key generation systems. In *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, pages 335–338, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[9] Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 471–486, London, UK, 1993. Springer-Verlag.

[10] Mike Burmester and Yvo Desmedt. A secure and efficient conference key distribution system. In *Advances in Cryptology—EUROCRYPT 94, A. D. Santis, Ed., Lecture Notes in Computer Science, vol. 950*, pages 275–286, New York, NY, USA, 1994. Springer-Verlag.

[11] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *SP '03: Proceedings of the 2003 IEEE Symposium*

on Security and Privacy, pages 197–201, Washington, DC, USA, 2003. IEEE Computer Society.

[12] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, New York, NY, USA, 2003. ACM Press.

[13] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM Press.

[14] Murat Fiskiran and Ruby B. Lee. Workload characterization of elliptic curve cryptography and other network security for constrained environments.

[15] Johann Großschädl. TinySA: a security architecture for wireless sensor networks. In *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*, pages 1–2, New York, NY, USA, 2006. ACM.

[16] Vipul Gupta, Matthew Millard, Stephen Fung, Yu Zhu, Nils Gura, Hans Eberle, and Sheueling Chang Shantz. Sizzle: A standards-based end-to-end security architecture for the embedded internet (best paper). In *PERCOM '05: Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 247–256, Washington, DC, USA, 2005. IEEE Computer Society.

[17] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 2004.

[18] IEEE802154. Telosb mote platform datasheet. mote datasheet, CrossBow Technology, www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[19] RSoft Design Inc. Artifex v.4.4.2, 2003.

[20] Hui Kang and Xiaolin Li. Power-aware sensor selection in wireless sensor networks. *Scalable Software Systems Laboratory, Computer Science Department, Oklahoma State University*, 2006.

[21] Donggang Liu, Peng Ning, and Rongfang Li. Establishing pairwise keys in distributed sensor networks. volume 8, pages 41–77, New York, NY, USA, 2005. ACM Press.

[22] DJ Malan, M. Welsh, and MD Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. pages 71–80, 2004.

[23] Alfred J. Menezes, Paul V. Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[24] Jelena Misic, Shairmina Shafi, and Vojislav B. Misic. Maintaining reliability through activity management in 802.15.4 sensor networks. In *Proceedings of the Second International Conference on Quality of Service in Heterogeneous*

Wired/Wireless Networks, page 5, Washington, DC, USA, 2005. IEEE Computer Society.

[25] Jelena Mišić, Shairmina Shafi, and Vojislav B. Mišić. Performance of beacon enabled ieee 802.15.4 cluster with downlink and uplink tarffic. volume 17, pages 361–377, April 2006.

[26] Moteiv. tmote sky lowpower wireless sensor module. tmote datasheet 802.15.4, Moteiv Corporation, San Francisco,CA, www.moteiv.com, 2006.

[27] Krzysztof Piotrowski, Peter Langendoerfer, and Steffen Peter. How public key cryptography influences wireless sensor node lifetime. In *SASN '06: Proceedings of the 4th ACM workshop on Security of ad hoc and sensor networks*, pages 169–176, New York, NY, USA, 2006. ACM.

[28] Theodore S. Rappaport. *Wireless communications: Principles & Paractice.* Prentice Hall, Upper Saddle River, NJ 07458, 1996.

[29] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proc. 4th ACM MobiHoc*, pages 177–188, Annapolis, MD, June 2003.

[30] Michael Steiner, Gene Tsudik, and Michael Waidner. Key agreement in dynamic peer groups. volume 11, pages 769–780, Piscataway, NJ, USA, 2000. IEEE Press.

[31] Ivan Stojmenović, editor. *Handbook of Sensor Networks: Algorithms and Architectures.* John Wiley & Sons, 2005.

[32] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *PERCOM '05: Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, Washington, DC, USA, 2005. IEEE Computer Society.

[33] Weichao Wang and Bharat Bhargava. Key distribution and update for secure inter-group multicast communication. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 43–52, Alexandria, VA, 2005.

[34] X. Brian Zhang, Simon S. Lam, Dong-Young Lee, and Y. Richard Yang. Protocol design for scalable and reliable group rekeying. volume 11, pages 908–922, Piscataway, NJ, USA, 2003. IEEE Press.

[35] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM Press.