Chapter 1

Power Management and Security in IEEE 802.15.4 Clusters: How to Balance?

Fereshteh Amini, Moazzam Khan, and Jelena Mišić
University of Manitoba, Winnipeg, Manitoba, Canada,
{*amini, umkhanm, jmisic*} *@cs.umanitoba.ca*

The IEEE 802.15.4 specification is a recent low data rate wireless personal area network standard. While basic security services are provided for, there is a lack of more advanced techniques which are indispensable in modern personal area network applications. In addition, performance implications of those services are not known. In this chapter, we describe a secure data exchange protocol based on the Zigbee specification and built on top of IEEE 802.15.4 link layer. This protocol includes a key exchange mechanism. We assume that all nodes are applying power management technique based on the constant event sensing reliability required by the coordinator. Power management generates random sleep times by every node which in average fairly distributes the sensing load among the nodes. Key exchange is initiated by cluster coordinator after some given number of sensing packets have been received by the coordinator. We develop and integrate simulation model of key exchange and power management technique into cluster's reliable sensing function. We evaluate the impact of security function and its periodicity on cluster performance.

1.1. Introduction

The IEEE 802.15.4 specification outlines a class of wireless radios and protocols targeted at low power devices, personal area networks, and sensor devices. IEEE 802.15.4 specification employs a number of well-known security services that can be implemented but at the cost of memory and communication overhead. Currently, not many wireless sensor network overhead statistics are available when security is employed in such networks. Sensor network application developers and network administrators always need these overhead statistics in choosing the security option that

best suites the security for a particular threat environment. For evaluating these security overheads on wireless sensor networks, we will simulate IEEE 802.15.4 media access control layer and secure data exchange once the devices exchange link keys with the PAN coordinator. We will measure communication costs that are incurred after employing these security features under different inputs to wireless sensor network model.

Key update provides an automated mechanism for restricting the amount of data which may be exposed when a link key is compromised. Key update frequency 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 technical challenge. In[1] authors reported the activity management and network behavior without considering any security parameters. In this work we develop simulation model for the cluster behavior including periodic key exchange (with variable update threshold), power management and sensing data application. For activity management, nodes in cluster apply sleep technique in order to deliver only the required number of packets per second (which we will call event sensing reliability) to the coordinator. We obtained simulation results to evaluate the overhead of key exchange in terms of medium behavior, total number of delivered packets, nodes' utilization and its effect on node's life time.

The chapter is organized as follows. We give an overview of IEEE 802.15.4 specification in section 1.2 and later in section 1.3 we introduce the security features addressed in IEEE 802.15.4. As IEEE 802.15.4 does not address any keying model, we are relying on keying model from Zigbee specification and discuss this in section 1.3.2 and section 1.3.4 explains the key update technique used in our study. In section 1.4 we explain the approach used for key activity management of the network to provide predetermined reliability. In section 1.5 we explain the simulation model, how it is implemented and in the same section will present our results. Finally we conclude our work in section 1.6.

1.2. IEEE 802.15.4

The need for low-cost, low-power and short-range communication is the main reason of introducing IEEE 802.15.4 Low Rate Wireless Personal Area Network (LR-WPAN) standard.[2] According to this specification, such WPAN consists of devices which are the basic components of these networks. Two or more devices communicating in a common physical channel

create a WPAN.

Star topology is one option for communication in LR-WPAN. In this topology devices communicate via a single central controller called PAN coordinator. After deciding on a PAN identifier, PAN coordinator may decide whether a device can join the PAN.

In the current work we concentrate on beacon-enabled based communication. In this form of communication, devices first listen for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate point, the device transmits its data packet, using slotted CSMA-CA, to the coordinator (uplink). The coordinator acknowledges the successful reception of the data by transmitting an acknowledgment frame.

On the other hand, when the PAN coordinator has something to send to a device (downlink), it informs the device by including in the network beacon that a data message is pending. The device periodically listens to network beacon and, if a message is pending, transmits a request frame to the coordinator using slotted CSMA-CA. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. The pending data frame is then sent using slotted CSMA-CA. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame.

1.3. Security in IEEE 802.15.4

IEEE 802.15.4 standard provides physical and link layer solutions for wireless personal area networks. It also provides well-known and well-understood cryptographic techniques[3,4] by supporting Authentication, Message integrity, Confidentiality and Freshness check for preventing replay attacks. Application of such security mechanisms comes at a cost that include processing overhead, memory overhead, power consumption and resulting low bandwidth.[5]

An application implemented using IEEE 802.15.4 has choice of different security suites that control the type of security protection by setting appropriate control parameters in the link layer security suite stack. A long Message Authentication Code (MAC) size improves the security feature of authentication and it is very difficult for an adversary to break or guess a MAC of longer size.[6] But this improved security is achieved at the cost of longer packet size. In IEEE 802.15.4 compliant wireless sensor networks, packet size is very crucial to the overall throughput that is required by

the application. Applications that support continuous data flow would be affected more than the applications in which data flow is periodic. Applications used for real time monitoring of some critical environments rely on continuous flow of data and hence by implementing security will affect the overall throughput and lifetime of such network by increasing the packet size. For the current work we will employee the security suite specified in IEEE 802.15.4 that supports both encryption and data integrity with MAC size of 128 bits. The security suite uses Counter with CBC-MAC (CCM)[7] mode of AES (Advanced Encryption Standard) for encryption and authentication. This cryptographic technique uses counter by first applying integrity protection both on message header and data payload and later it encrypts the data payload and MAC using AES. At the receiver end the receiver gets the packet and applies decryption using parameters based on sender's address from its Access Control List.

1.3.1. *Security building blocks*

The IEEE 802.15.4 specification provides basic security mechanisms but these security features can not work at their own. The level of security in any network revolves around the keys that are shared among devices. Different approaches have been suggested to distribute and manage these keys. Since IEEE 802.15.4 does not suggest any keying mechanism, in this work we will follow the keying mechanism from Zigbee alliance specifications.[4] In this section we will first introduce the keying mechanisms and later explain how this is handled in Zigbee specification by taking advantage of the inherent security mechanisms already provided by IEEE 802.15.4.

1.3.1.1. *Keying Model*

As explained above, the IEEE 802.15.4 addresses good security mechanisms but it still does not address what type of keying mechanism will be used to employ above 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 theoret-

ical descriptions of security protocol at network layer. There is no specific study or results published by Zigbee alliance in regards to which security suite perform better in different application overheads. Zigbee alliance has also 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.

Application support sub-layer of ZigBee specification provides the mechanism by which a Zigbee device may derive a shared secret key (Link Key) with another ZigBee device. Key establishment involves two entities, an initiator device and a responder device and is prefaced by a trust provisioning step. Trust information (e.g. MASTER key) provides a starting point for establishing a link key and can be provisioned in-band or out-band.

Zigbee alliance uses Symmetric-Key Key Establishment (SKKE) protocol for link key establishment. In SKKE an initiator device establishes a link key with a responder device using a master key. This master key, for example, may be pre-installed during manufacturing, may be installed by a trust center, or may be based on user-entered data (PIN, password). In current study we assume that all the devices and PAN coordinator have pre-installed Master keys and we will focus mainly on Link key establishment.

1.3.1.2. *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[8] 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.[9] 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

$$MacTag = MAC_{MacKey} MacData$$

1.3.2. *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.

1.3.2.1. *Exchange of ephemeral data*

Figure 1.1 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

$$Challengelen \in [minchallengelen, maxchallengelen]$$

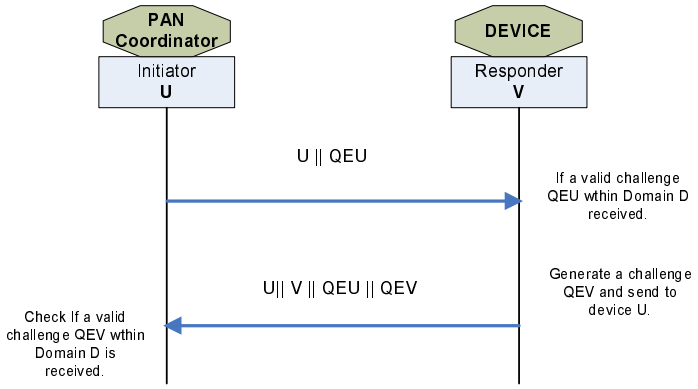


Fig. 1.1. Exchange of ephemeral data

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.

1.3.2.2. *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 (Figure 1.2).

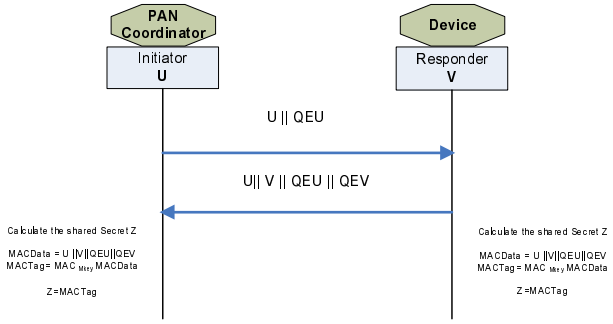


Fig. 1.2. Generation of shared Secret

(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.)

$Z = MACTag$

1.3.2.3. *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.[10]

$Hash_1 = H(Z||01)$
$Hash_2 = H(Z||02)$

The hash value $Hash_2$ will be Link key among two devices (Figure 1.3). 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.

$$MACKey = Hash_1 \tag{1.1}$$

$$KeyData = Hash_2 \tag{1.2}$$

$$KKeyData = Hash_1||Hash_2 \tag{1.3}$$

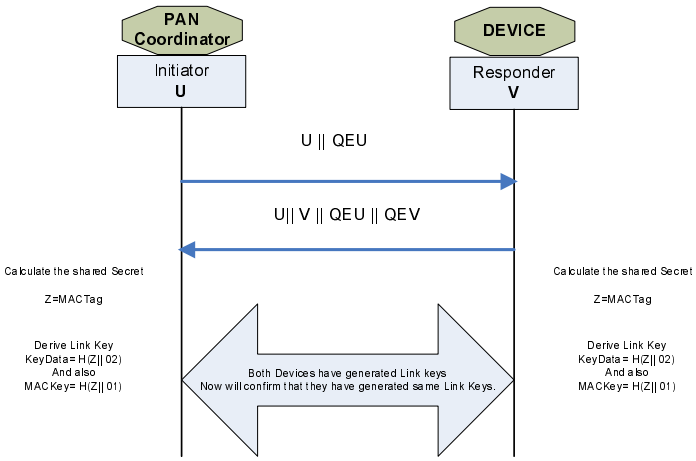


Fig. 1.3. Generation of Link Key

1.3.2.4. *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 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

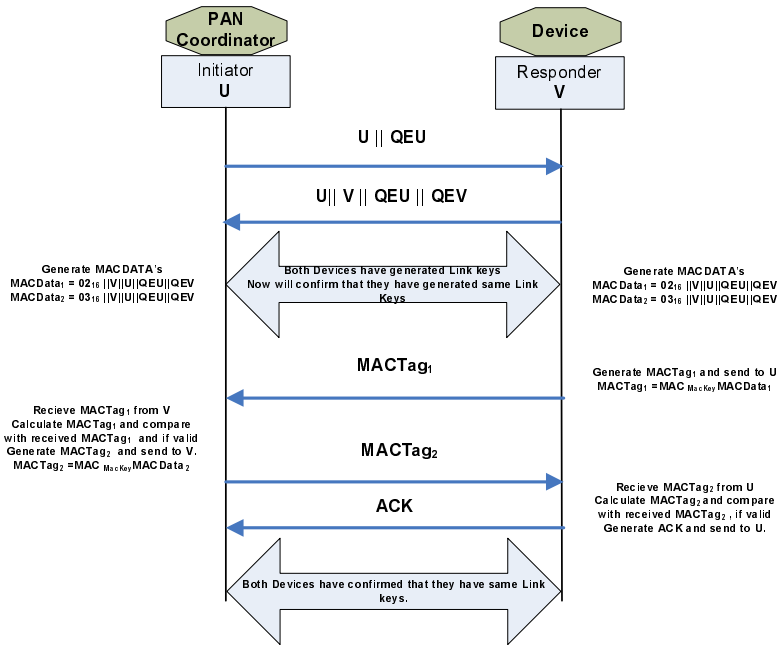


Fig. 1.4. Confirmation of Link Keys

$MACTags$ using the key $MACkey$ (Equation 1.1) 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 received $MACTags$ 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 (Figure 1.4).

1.3.3. *Use of SKKE in our simulation model*

We have implemented SKKE in four major communication steps as are described in ZibBee specification[4] (Figure 1.5).

SKKE-1
Initiator U will send the Challenge QEU and wait for the Challenge QEV from responder 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.

1.3.4. *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

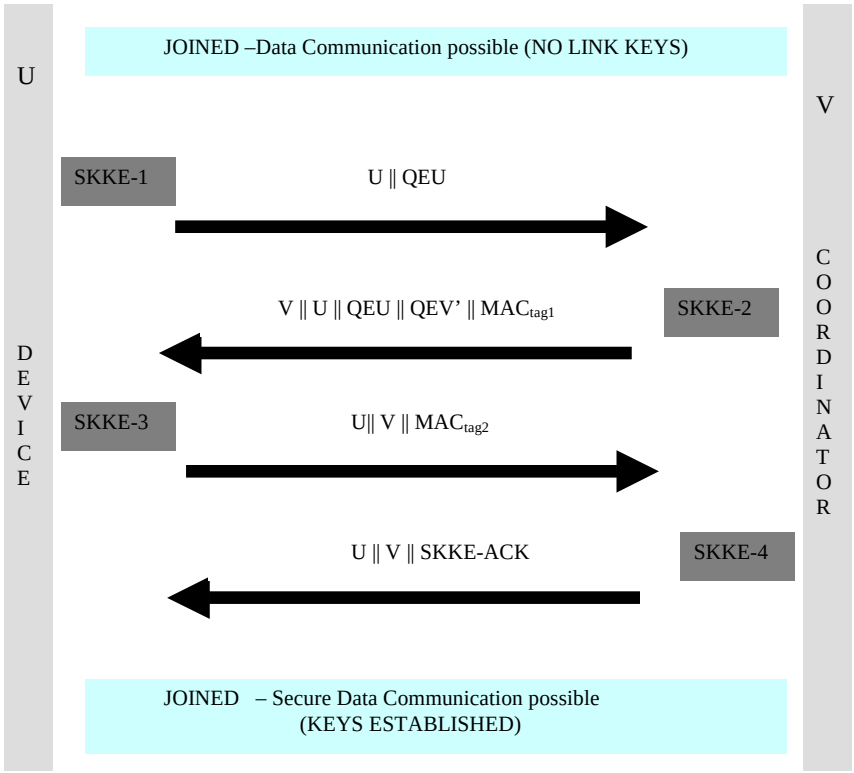


Fig. 1.5. SKKE protocol

in any network. Approaches for key updates in general wireless networks mainly target network that have group key structures and have high communication bandwidth .[11,12] For resource scarce IEEE 802.15.4 networks these key updates will effect the performance adversely.

In this work we assume that PAN coordinator maintains a counter for each node that keeps track of the number of packets exchanged under the same key (Fig. 1.5). When the threshold value of the counter is reached 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 when they acknowledge the new key. 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.

1.4. Power Management

Power management consists of adjusting the frequency and ratio of active and inactive periods of sensor nodes.[13,14] For IEEE 802.15.4 nodes it can be implemented in two ways. In the first one, supported by the standard,[4] the interval between the two beacons is divided into active and inactive parts, and the sensors can switch to low-power mode during the inactive period. Activity management for individual nodes can be accomplished through scheduling of their active and inactive periods.

Let us consider a sensing application in which redundant sensors are used to achieve the desired value of event sensing reliability (number of packets persecond needed for reliable event detection).[14] 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 a 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. In this approach, at the end of each packet transmission the node checks its uplink buffer. If it is empty, the node immediately goes to sleep; if there are packets to send, the node transmits the next packet from the buffer with the probability P_{active}, or goes to sleep with the probability $1 - P_{active}$. Therefore, two control parameters are needed: one, P_{sleep}, regulates the duration of the inactive period; the other, P_{active}, regulates the duration of the active period. when individual nodes begin to cease functioning, either because of battery exhaustion or for other reasons, the remaining nodes will have to extend their activity to achieve satisfactory reliability, and the importance of the Bernoulli mechanism will increase.

Depending on whether we split the computational load of activity management, we can have two approaches of centralized and distributed controls. By choosing the later approach to distribute the computational load more evenly, we assume that the network coordinator is aware of the number of sensor nodes (which have to be explicitly admitted to the network[2]) and their packet arrival rates (which may be obtained as simple long-term averages, as packet headers contain the source node address). The coordinator first determines node utilization based on the number of live nodes and then calculates the individual reliability r per node (by dividing the re-

quired collective reliability R by the number of live nodes n) and sends this information within the beacon frame. Over time some sensors die, and the coordinator has to broadcast updated values of individual reliability, which grow whenever one of the sensors die. Note that the sleep time is geometrically distributed, and the 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 $Psleep = 1 - rt_{boff}$ and $Pactive = 0$. It then monitors the utilization of its radio transmitter/receiver subsystem, using a monitoring window of specified size. Utilization is simply calculated as the count of backoff periods in which the node was active during the recent window divided by the total size of the window.

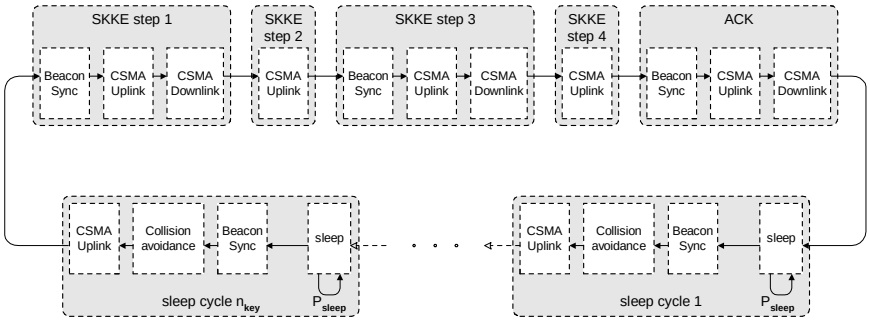


Fig. 1.6. Markov chain for the node behavior under threshold triggered key exchange.

We have developed Markov chain model for node behavior which includes all phases of SKKE protocol and subsequent sleep and transmission phases. We assume that PAN coordinator maintains a separate counter for the number of transmissions by each node. When the counter value reaches threshold n_k, key update protocol is triggered. Updated keys are used to generate Message Authentication Code. The high level Markov chain which includes key update, sleep periods followed by the transmissions is presented in Fig. 1.6.

1.5. Simulation Model

We have simulated the key exchange mechanism and sleep mechanism for the IEEE 802.15.4 network using Artifex[15] 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[1,16] and later explain the simulated key exchange and update process and power management in our current work.

1.5.1. *Beacon-Enabled IEEE 802.15.4 Simulation Model*

The network communication model of this simulation is based on star topology. The model is built on three primary objects : 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 every t time.

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 then received by the medium. If the number of received packets in medium is greater than 1, collision occurs. If there are no collisions, data packets are sent successfully to the PAN coordinator and the medium status is set to busy.

PAN coordinator is the next stop for data packets and is responsible for sending ack type packets to corresponding device after a specified delay. As of every packet, ack will be received first by the medium and then sent to corresponding 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.

1.5.2. *Adding Key Exchange Mechanism to the Simulation Model of IEEE 802.15.4 Network*

In this section we describe the communication between the ordinary nodes and PAN coordinator which occurs as result from the link key exchange. We assume that devices are attached to the cluster and the formation of the piconet is finalized. Also, we assume the master keys are established, so that there is no threat of eavesdropping during exchange of master keys. The next step is generating link keys between each device and PAN coordinator. For the exchange of link keys, we will follow SKKE protocol as describe in Section 1.3.3.

The process of key generation starts 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.

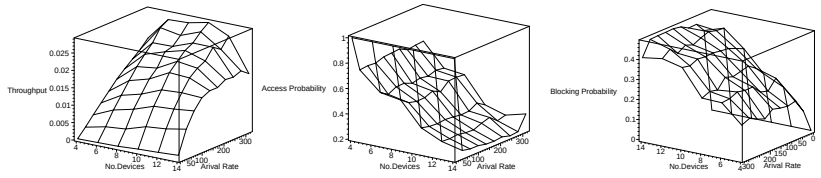
In our model, key exchange packets have non-preemptive priority over data packets. If the node has started backoff process for data packet and it hears its ID in the beacon it will finish the current packet transmission before sending the request packet. However, if data packet arrives to the device's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

1.5.3. *Simulation Run and Analysis*

We have implemented the physical, data link and security layer 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

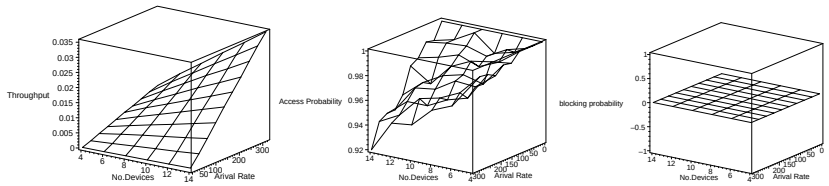
Medium Access control layer headers, and it is set to 30 bytes i.e. to three backoff periods. When packet signature (message authentication code) of 16 bytes is added to the total packet size had to be rounded to 5 backoff periods (the largest packet size could be set to 13 backoff periods).

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.



(a) Throughput (b) Access probability (c) Blocking Probability

Fig. 1.7. 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



(a) Throughput (b) Access probability (c) Blocking probability

Fig. 1.8. Throughput, Access Probability and Blocking Probability as the function of simulation time(backoffs) when no security technique is employeed.

we 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 CBC-MAC, and key exchange between the nodes over various packet arrival rates and cluster sizes. Figure 1.7 presents throughput, access probability (probability of no packet collision) and blocking

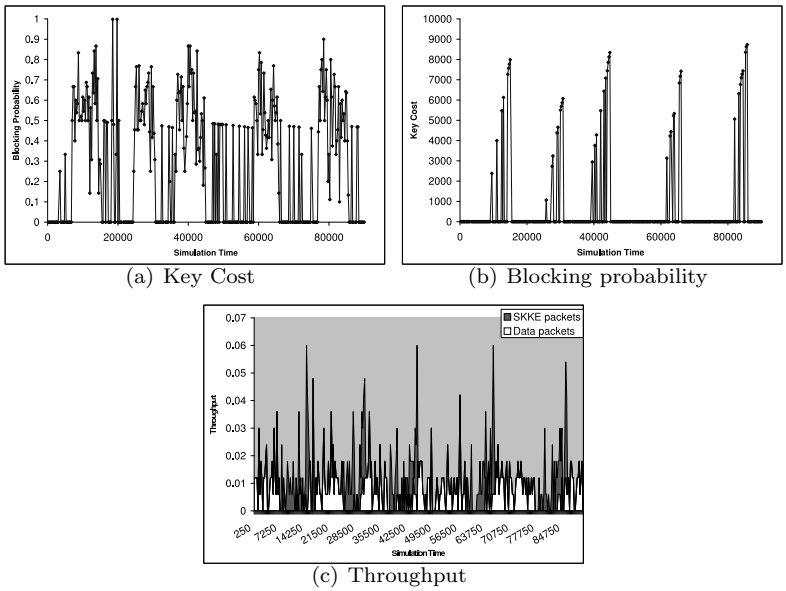
probability at the node's buffer when all security overhead is included. Results were taken for varying number of nodes and varying packet arrival rate per node. Figure 1.8 presents the same parameters (except the key exchange cost since it does not exist) when no security measures are deployed in the 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 behaviour of the cluster over time is presented in Fig. 1.9. Fig. 1.9(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. 1.9(b). Due to large periods when device transmission is prevented during key exchange (well over 1500 backoff periods), the blocking probability skyrockets 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 1.9(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,[1] 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.

We 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. We



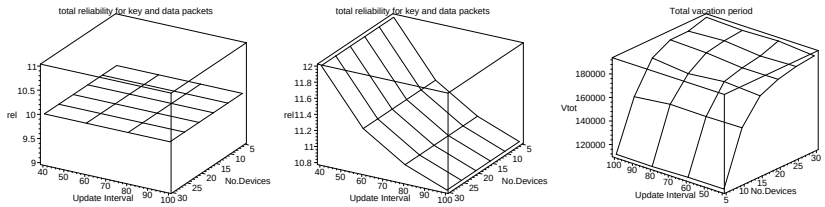
(a) Key Cost (b) Blocking probability

(c) Throughput

Fig. 1.9. Key Cost, Blocking Probability and Throughput as the function of simulation time(backoffs) when devices stop their communications for key updates

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] We also assume that the physical layer header has 6 bytes, and that the Medium Access Control sublayer header and Frame Check Sequence fields have a total of 9 bytes.

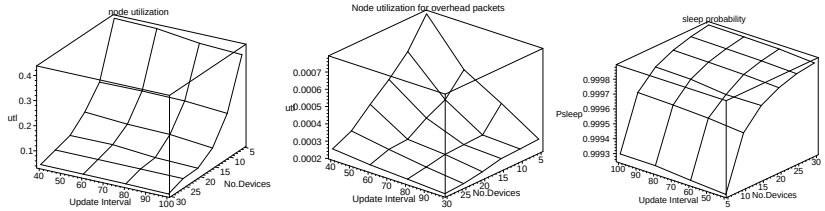
Figure 1.10(b) shows total number of successfully transmitted packets (including key and data information) transmitted per second for requested data reliability of $R = 10$ packets per second (which is shown on Fig. 1.10(a)). We note that the total number of packets hyperbolically grows when the key exchange threshold decreases linearly Fig. 1.10(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



(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 run = 200,000.



(d) Total utilization

(e) Utilization due to key exchange.

(f) Sleep probability.

Fig. 1.10. Event sensing reliability for data and key+data, inactive period, total utilization and utilization for key packets, average number of active devices and sleep probability for a node.

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. 1.10(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.1.10(e). Finally, sleep probability for each node is shown in Fig. 1.10(f). Sleep probability dominantly changes with n, while the changes with n_k are much milder.

1.6. Conclusion and Future Work

We have simulated and studied key exchange process integrated with reliable sensing and power management in beacon enabled IEEE 802.15.4 cluster and the results confirm our expectations. Data encryption is pro-

vided by exchanging link keys between each device and PAN coordinator. The signature payload plays a big role on performance of the network. We have developed model of key exchange integrated into the sensing function of beacon enabled IEEE 802.15.4 cluster. Our results show important impact of the ratio of the event sensing reliability and key update threshold on the clusters energy consumption. We have evaluated the impact of the threshold for key update on the clusters descriptors. The results can give useful hints for the choice of frequency of key updates for required event sensing reliability.

For the future work we will measure more realistically the performance of secure IEEE 802.15.4 personal area network. Combination of other key exchange protocols, activity management and key management techniques will be compared to get measures that will be used to enhance the lifetime of wireless personal area network along with defense against expected threats from environment.

References

1. Mišić, J., Shafi, S., Mišić, V.B.: Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic. IEEE Transactions on Parallel and Distributed Systems **17** (2006) 1–16
2. : 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)
3. Stallings, W.: Cryptography and Network Security: Principles and Practice. Prentice Hall, Upper Saddle River (2003)
4. : ZigBee specification (ZigBee document 053474r06, version 1.0). ZigBee Alliance (2004)
5. Sastry, N., Wagner, D.: Security considerations for IEEE 802.15.4 networks. In: WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security. (2004) 32–42
6. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Computer and System Sciences **61** (2000) 362–399
7. Whiting, D., Housley, R., Ferguson, N.: Counter with cbc-mac (CCM). http://www.rfc-archive.org/getrfc.php?rfc=3610 (2003)
8. Menezes, A., Oorschot, P.V., Vanstone, S.: Handbook of Applied Cryptography. CRC Press (1997)
9. : 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)
10. : 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)
11. Wang, W., Bhargava, B.: 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, New York, NY, USA, ACM Press (2005) 43–52
12. Zhang, X.B., Lam, S.S., Lee, D.Y., Yang, Y.R.: Protocol design for scalable and reliable group rekeying. IEEE/ACM Trans. Netw. **11** (2003) 908–922
13. Stojmenović, I., ed.: Handbook of Sensor Networks: Algorithms and Architectures. John Wiley & Sons, New York, NY (2005)
14. Sankarasubramaniam, Y., Akan, Ö.B., Akyildiz, I.F.: ESRT: event-to-sink reliable transport in wireless sensor networks. In: Proc. 4th ACM MobiHoc, Annapolis, MD (2003) 177–188
15. Inc., R.D.: Artifex v.4.4.2 (2003)
16. Shafi, S.: Performance of a beacon enabled IEEE 802.15.4-compliant network. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada (2005)