Security in IEEE 802.15.4 cluster based networks

Moazzam Khan, Jelena Mišić
Department of Computer Science
University of Manitoba, Winnipeg, Manitoba, Canada

July 15, 2006

Contents

1 Security in IEEE 802.15.4 cluster based networks 1

1.1 Cluster-based networks and network lifetime 3

1.2 Security in Wireless Sensor Networks 5

1.2.1 Security Techniques 6

1.2.2 Data Confidentiality 6

1.2.3 Data Authentication 6

1.2.4 Data Integrity 7

1.2.5 Replay Protection 7

1.3 Overview of IEEE 802.15.4 security operations 7

1.3.1 Addressing 8

1.3.2 No Security 10

1.3.3 AES-CTR 11

1.3.4 AES-CBC-MAC 11

1.3.5 AES-CCM 12

1.3.6 Replay Protection 12

1.4 Key Management Models 12

1.4.1 Probabilistic keying models 13

1.4.2 Deterministic keying models 14

1.4.3 Hybrid keying models 14

1.4.4 Group-wise keying models 15

1.4.5 Key updates 16

1.4.6 Limitations of IEEE 802.15.4 standard from the security aspect . . . 17

1.5 Security services provided by ZigBee alliance 17

1.5.1 Symmetric-Key Key Establishment Protocol (SKKE) 19

1.5.2 Communication steps in SKKE protocol 22

1.6 Summary 23

Chapter 1

Security in IEEE 802.15.4 cluster based networks

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. This standard will play important role in sensitive applications including habitat monitoring, burglar alarms, inventory control, medical monitoring, emergency response and battlefield management needs 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 then their peer-to-peer counterparts because the PAN coordinator can act as both the network controller and the sink to collect the 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 in active part where nodes can transmit using CSMA-CA or TDMA (called Guaranteed Time Slots) and 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 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 [15].

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 we explain the relationship between the sensor network architecture and its availability for both data collection and event sensing applications. We believe that network availability for sensing applications has the same importance as data integrity and to some extent data confidentiality. Section 1.2 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 1.3. Section 1.4 discusses keying models currently used in WPANs. Security issues addressed by Zigbee alliance specifications [4] are discussed in Section 1.5. Finally, Section 1.6 concludes this Chapter.

1.1 Cluster-based networks and network lifetime

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 [12]. 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. Therefore, from the aspect of availability long network lifetimes become important security aspect.

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) compared to the link type which is available to ordinary nodes (CSMA-CA). 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 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 (time bounds are not important for data collection due to time-correlation of sensed data). In event detection applications, network availibility and data integrity is much more critical than in data collection applications. Again, we 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 near by nodes. Nodes that are not within direct communication range use other nodes to relay message between them. Routing in such 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 routing in wireless sensor networks based on IEEE 802.15.4 make the implementation of security a technical challenge. Even at the MAC 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. For implementation of secure sensor network we have to compromise on network life to some extent and vise versa.

1.2 Security in Wireless Sensor Networks

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 few wireless receiver placed outside a house might be able to monitor the light and temperature reading of sensor readings of sensor network inside the house, thus revealing detailed information about occupant's daily personal activity. Similarly attacker can obtain their own commodity sensor nodes and present it as a legitimate node inside the network and once they have few nodes like that in network, attacker can launch different types of attack – for example denial of service, falsification of sensed data, dropping of sensed data etc.

1.2.1 Security Techniques

Different Security techniques are employed to safeguard threats of such eavesdropping

1.2.2 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.

1.2.3 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.

1.2.4 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.

1.2.5 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.

1.3 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. 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.

1.3.1 Addressing

For unique identification of 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 examples, 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 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 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

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 [1]]

- Encryption only (AES-CTR)
- Authentication only (AES-CBC-MAC)
- Encryption and Authentication (AES-CCM)

The specification supports MAC 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. The choice of secure authentication is tied with the addressing of devices in IEEE 802.15.4 devices. 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 1.1).

The *security material* is 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 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 1.2).

We will now provide more detail about the categories of security suites:

1.3.2 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.

1.3.3 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 $p1.....,pn$ 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 infallible 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 (Figure1.2)

1.3.4 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.

1.3.5 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.

1.3.6 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 value, the *replay counter*, with 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. We 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.

1.4 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 architectures such as distributed or hierarchical, (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 pair-wise, 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 impractical in large-scale sensor networks. At present, the most practical approach for bootstrapping secret keys in sensor networks is to use pre-deployed 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:

1.4.1 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 [Chan et al,2003][8] 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 [Eschenauer and Gligor,2002][10] also proposed probabilistic key pre-distribution scheme which relies on probabilistic key sharing among the nodes of a random graph.

1.4.2 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 [Blom] [5] 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 [Du et al. 2003] [9] 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 [Blundo et al. 1992][6] distributes a polynomial share (a partially evaluated polynomial) to each sensor node by using which every pair of nodes can generate a link key.

1.4.3 Hybrid keying models

Finally, hybrid solutions use probabilistic approaches on deterministic solutions to improve scalability and resilience. Polynomial pool-based key pre-distribution scheme [Liu and Ning. 2003][13] considers the fact that not all pairs of sensor nodes have to establish a key. It combines Polynomial based key pre-distribution scheme [Blundo et al. 1992][6] with the key-pool idea in [Eschenauer and Gligor. 2002; Chan et al. 2003][10, 8] to improve resilience and scalability.

1.4.4 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. [Burmester and Desmedt 1994][7] and IKA2 [Steiner et al. 2000][17] use a Diffie-Hellman based group key transport protocol. However recently, some works on the public key cryptography protocols (e.g Elliptic curve cryptography) evaluation and efficiency measurments on sensor node platforms showed optimistic results [Wander et al.2005; Gupta et al.2005][18, 11]. 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. Localized encryption and authentication protocol (LEAP) [19] [Zhu et al. 2003] provides a mechanism to generate group-wise keys which follows LEAP 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 the 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 keying mechanisms 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 purposes. 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.

1.4.5 Key updates

Key management schemes 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.

1.4.6 Limitations of IEEE 802.15.4 standard from the security aspect

Higher layers will determine when the security is to used at MAC layer by any device and provide all keying material necessary provide the security services. Key management, device authentication and freshness protection may be provided by the higher layers but is not not addressed in IEEE 802.15.4 standard. The management and establishment of keys is the responsibility of the implementer of higher layers. There is no simple way of group keys in IEEE 802.15.4 enabled WSN because as mentioned earlier the ACL entries are only associated to single destination address. A detailed analysis of shortcomings of security feature is mentioned by [D. Wagner et al. 2004] [16].

1.5 Security services provided by ZigBee alliance

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 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 [14] 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

$$MacTag = MAC_{MacKey}MacData$$

1.5.1 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 1.3 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

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

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 1.4.

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 1.5). 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.

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

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

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

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 $MACTags$ 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 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 1.6.

1.5.2 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 1.7.

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.

Authors have simulated the key exchange process in IEEE 802.15.4 on top of simulation model of this network and initial results confirm the expected performance decrease of overall network. They also have provided data encryption by exchanging link keys between each device and cluster head. The signature payload plays a big role on performance of the cluster. Also we have observed that the total access delay is higher when encryption and decryption is provided.

1.6 Summary

In this 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 the implementer of higher layers should make great deal of effort in choosing the right keying model based on the application requirements. Even though Zigbee alliance has outlined protocols regarding the key exchange it is necessary to integrate them with the IEEE 802.15.4 Medium Access Protocol. Since key exchange protocols require downlink communications from the PAN coordinator to the ordinary nodes it will consume a lot of 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 (by us and other researchers) 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.

References

[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] Z. Alliance. ZigBee specification (ZigBee document 053474r06, version 1.0), Dec. 2004.

[5] R. 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.

[6] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. 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.

[7] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In *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.

[8] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197, Washington, DC, USA, 2003. IEEE Computer Society.

[9] W. Du, J. Deng, Y. S. Han, and P. 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.

[10] L. Eschenauer and V. 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.

[11] V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, and S. C. Shantz. Sizzle: A standards-based end-to-end security architecture for the embedded internet (best paper). In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 247–256, Washington, DC, USA, 2005. IEEE Computer Society.

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

[13] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, 2005.

[14] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[15] J. Mišić, S. Shafi, and V. B. Mišić. Performance of beacon enabled ieee 802.15.4 cluster with downlink and uplink tarffic. *IEEE Transactions on Parallel and Distributed Systems*, 17(4):361–377, Apr. 2006.

[16] N. Sastry and D. Wagner. Security considerations for ieee 802.15.4 networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 32–42, New

York, NY, USA, 2004. ACM Press.

[17] G. W. M. Steiner, M. Tsudik. Key agreement in dynamic peer groups. In *Parallel and Distributed Systems, IEEE Transactions on*, pages 769 – 780, Washington, DC, USA, Aug. 2000. IEEE Computer Society.

[18] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, Washington, DC, USA, 2005. IEEE Computer Society.

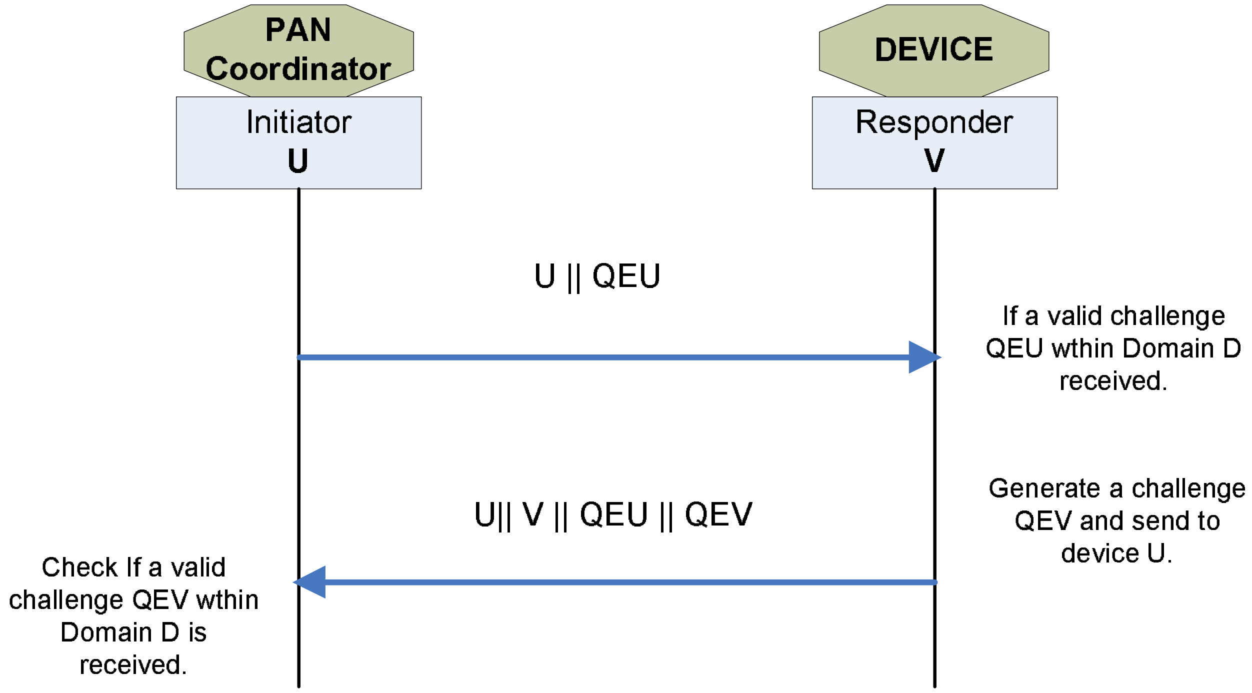
[19] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM Press.

Address	Security Suite	Key	Replay Counter

Figure 1.1: Access control List entry

Frame Counter	Key Counter	Encrypted Payload	Encrypted MAC

Figure 1.2: Frame format after adding security features



If a valid challenge QEU wthin Domain D received.

Generate a challenge QEV and send to device U.

Check If a valid challenge QEV wthin Domain D is received.

Figure 1.3: Exchange of ephemeral data

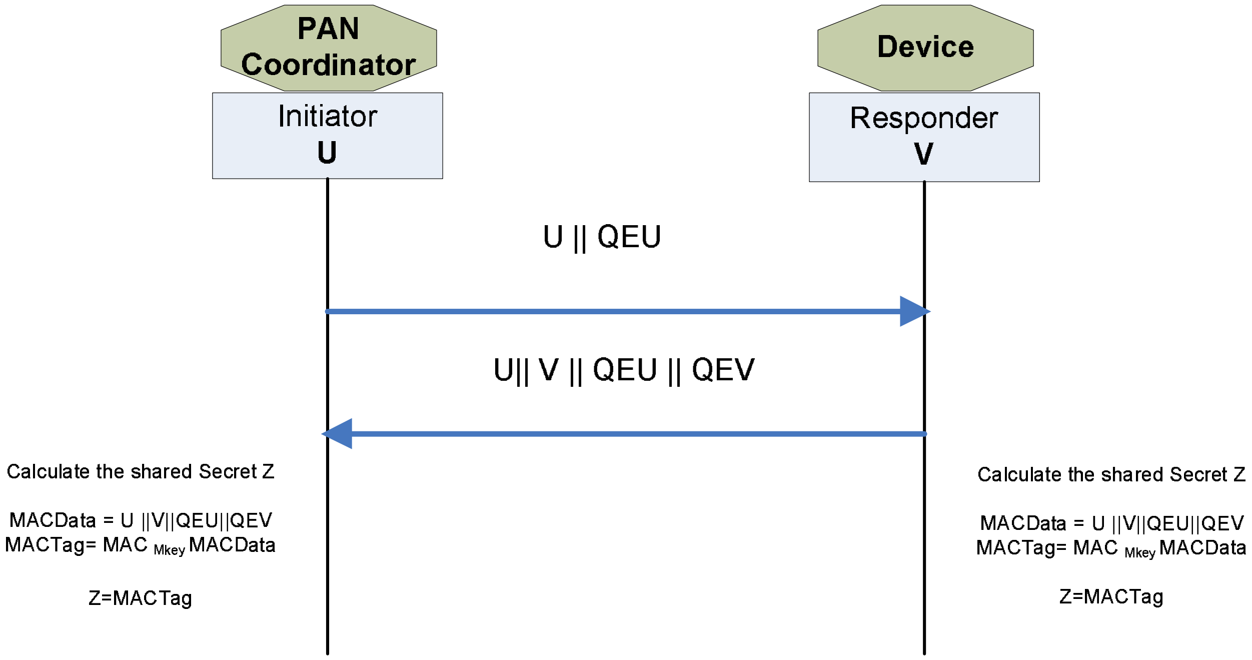

PAN Coordinator
Initiator
U
Device
Responder
V
U || QEU
U|| V || QEU || QEV
Calculate the shared Secret Z
MACData = U ||V||QEU||QEV
MACTag= MAC $_{Mkey}$ MACData
Z=MACTag
Calculate the shared Secret Z
MACData = U ||V||QEU||QEV
MACTag= MAC $_{Mkey}$ MACData
Z=MACTag

Figure 1.4: Generation of shared Secret

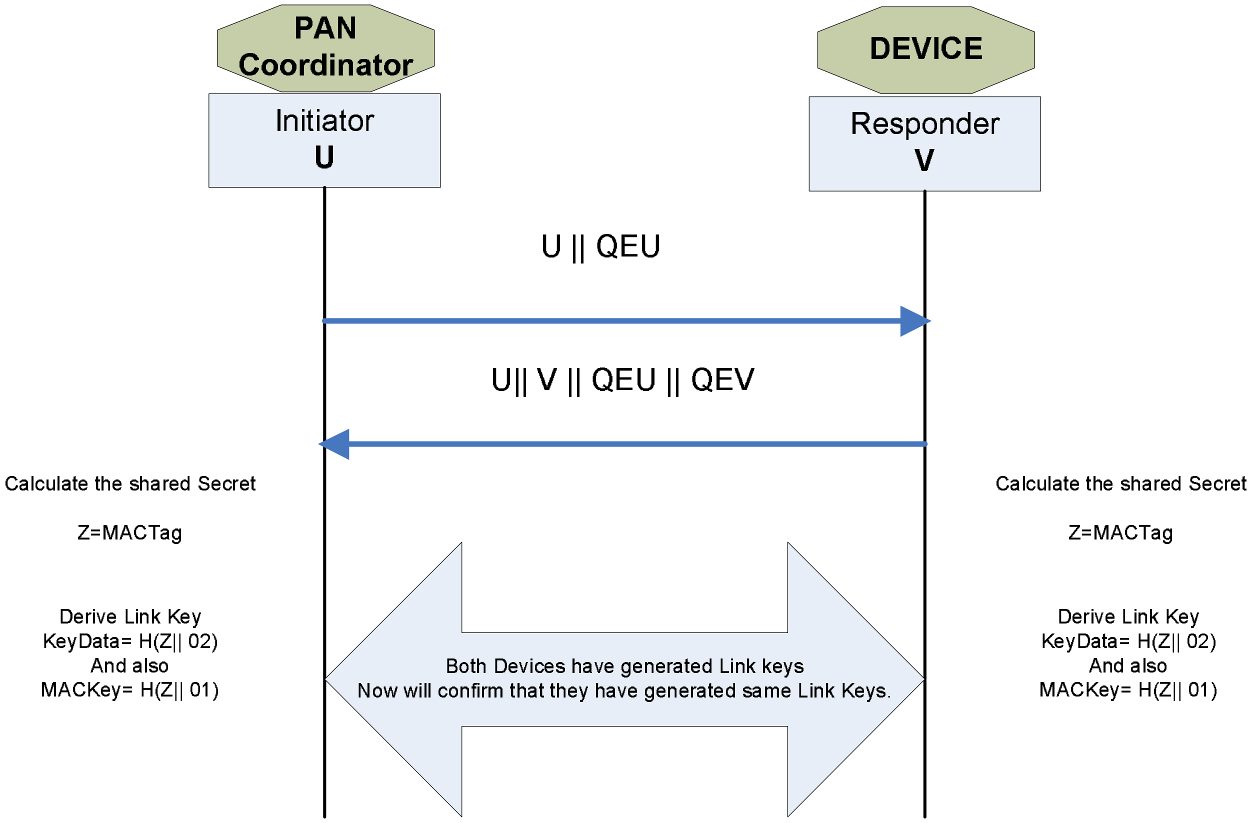


Figure 1.5: Generation of Link Key

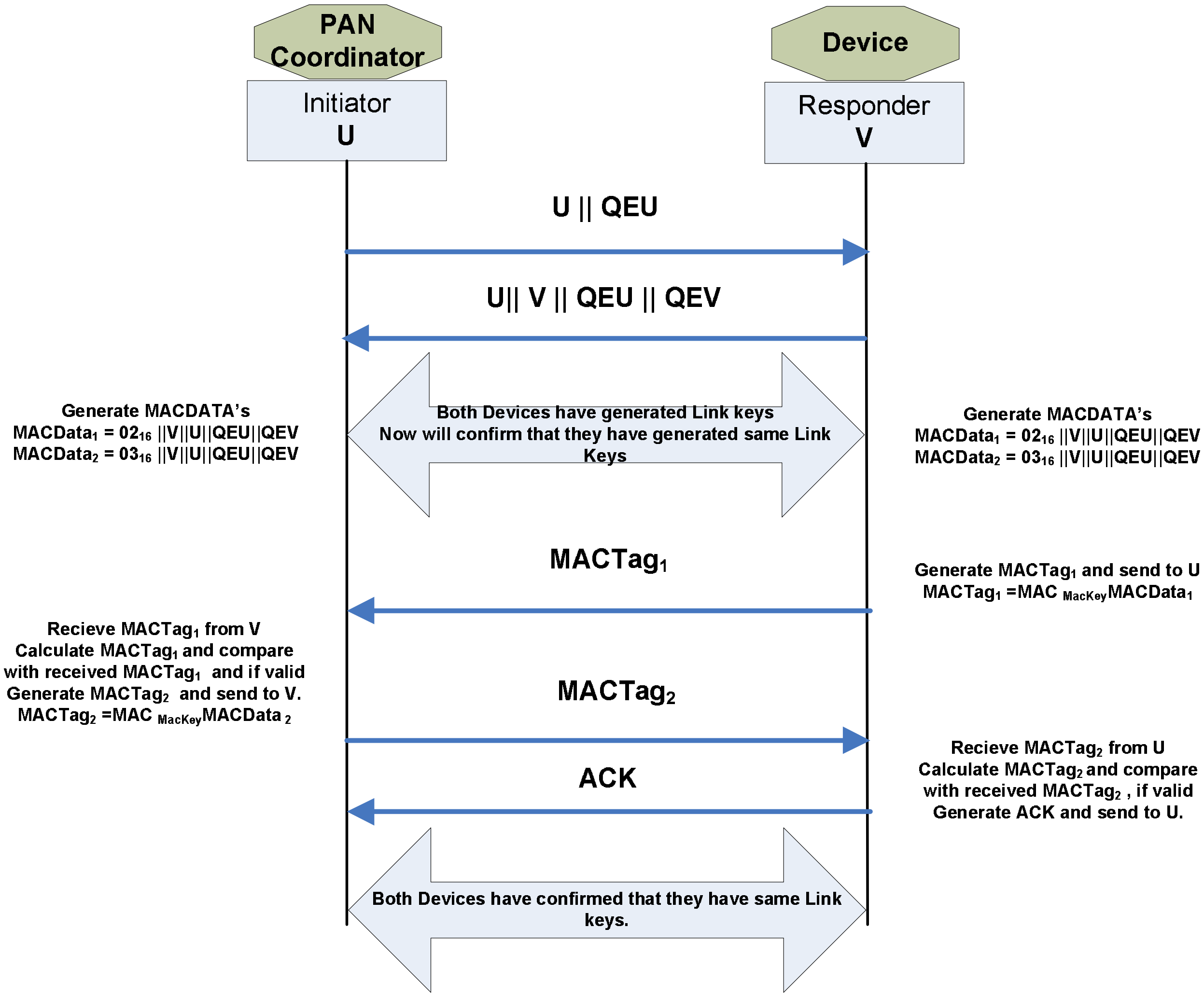

PAN Coordinator
Device
Initiator U
Responder V
U || QEU
U|| V || QEU || QEV
Generate MACDATA's
MACData₁ = 02₁₆ ||V||U||QEU||QEV
MACData₂ = 03₁₆ ||V||U||QEU||QEV
Both Devices have generated Link keys
Now will confirm that they have generated same Link Keys
Generate MACDATA's
MACData₁ = 02₁₆ ||V||U||QEU||QEV
MACData₂ = 03₁₆ ||V||U||QEU||QEV
MACTag₁
Generate MACTag₁ and send to U
MACTag₁ =MAC MacKey MACData₁
Recieve MACTag₁ from V
Calculate MACTag₁ and compare with received MACTag₁ and if valid
Generate MACTag₂ and send to V.
MACTag₂ =MAC MacKey MACData ₂
MACTag₂
Recieve MACTag₂ from U
Calculate MACTag₂ and compare with received MACTag₂ , if valid
Generate ACK and send to U.
ACK
Both Devices have confirmed that they have same Link keys.

Figure 1.6: Confirmation of Link Keys

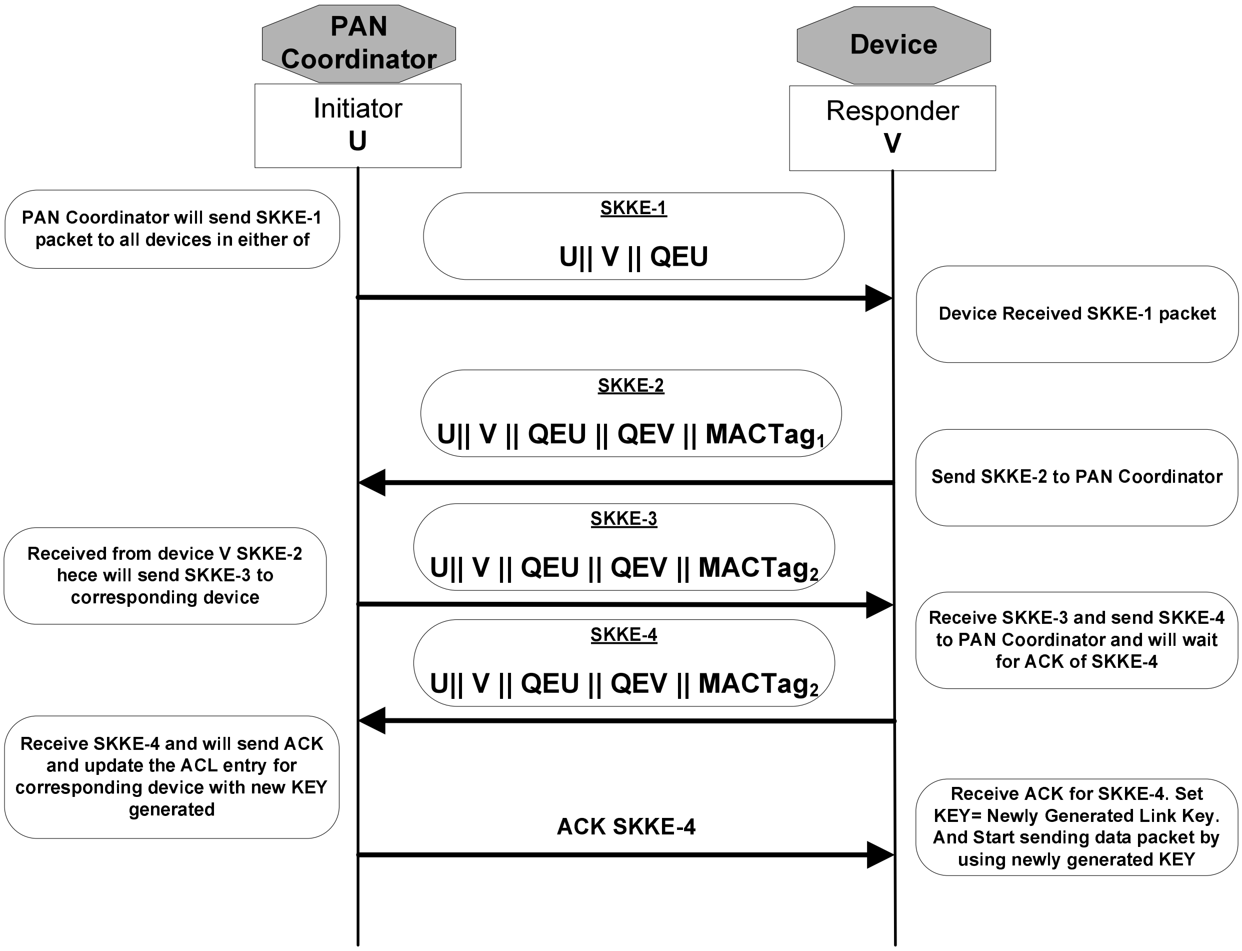

PAN Coordinator
Device
Initiator U
Responder V
PAN Coordinator will send SKKE-1 packet to all devices in either of
SKKE-1
U|| V || QEU
Device Received SKKE-1 packet
SKKE-2
U|| V || QEU || QEV || MACTag1
Send SKKE-2 to PAN Coordinator
SKKE-3
U|| V || QEU || QEV || MACTag2
Received from device V SKKE-2 hece will send SKKE-3 to corresponding device
Receive SKKE-3 and send SKKE-4 to PAN Coordinator and will wait for ACK of SKKE-4
SKKE-4
U|| V || QEU || QEV || MACTag2
Receive SKKE-4 and will send ACK and update the ACL entry for corresponding device with new KEY generated
ACK SKKE-4
Receive ACK for SKKE-4. Set KEY= Newly Generated Link Key. And Start sending data packet by using newly generated KEY

Figure 1.7: Communicaton Steps in SKKE protocol