INTERNATIONAL STANDARD

ISO/IEC 29180

First edition
2012-12-01

Information technology — Telecommunications and information exchange between systems — Security framework for ubiquitous sensor networks

Technologies de l'information — Télécommunications et échange d'informations entre systèmes — Cadre de sécurité pour réseaux de capteurs ubiquitaires

Reference number
ISO/IEC 29180:2012(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29180 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*, in collaboration with ITU-T. The identical text is published as Rec. ITU-T X.1311 (02/2011).

Introduction

This Recommendation | International Standard describes the security threats to and security requirements of the ubiquitous sensor network. In addition, this Recommendation | International Standard categorizes the security technologies according to the security functions that satisfy the said security requirements and where the security technologies are applied in the security model of ubiquitous sensor networks. Finally, the security functional requirements and security technologies for the ubiquitous sensor networks are presented.

INTERNATIONAL STANDARD
RECOMMENDATION ITU-T

Information technology – Security framework for ubiquitous sensor networks

1 Scope

The recent advancement of wireless-based communication technology and electronics has facilitated the implementation of a low-cost, low-power sensor network. Basically, a ubiquitous sensor network (USN) consists of three parts: a sensor network consisting of a large number of sensor nodes, a base station (also known as a gateway) interfacing between the sensor networks and an application server, and the application server controlling the sensor node in the sensor network or collecting the sensed information from the sensor nodes in the sensor network.

USN can be an intelligent information infrastructure of advanced e-Life society, which delivers user-oriented information and provides knowledge services to anyone anytime, anywhere and wherein information and knowledge are developed using context awareness by detecting, storing, processing, and integrating the situational and environmental information gathered from sensor tags and/or sensor nodes affixed to anything. Since there are many security and privacy threats in transferring and storing information in the USN, appropriate security mechanisms may be needed to protect against those threats in the USN.

This Recommendation | International Standard describes the security threats to and security requirements of the ubiquitous sensor network. In addition, this Recommendation | International Standard categorizes the security technologies according to the security functions that satisfy the said security requirements and where the security technologies are applied in the security model of the USN. Finally, the security requirements and security technologies for the USN are presented.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

None.

2.2 Paired Recommendations | International Standards equivalent in technical content

– Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

ISO/IEC 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

– Recommendation ITU-T X.805 (2003), *Security architecture for systems providing end-to-end communications*.

ISO/IEC 18028-2:2006, *Information technology – Security techniques – IT network security – Part 2: Network security architecture*.

2.3 Additional references

– Recommendation ITU-T H.235.0 (2005), *H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems*.

– Recommendation ITU-T X.1111 (2007), *Framework of security technologies for home network*.

– Recommendation ITU-T X.1191 (2009), *Functional requirements and architecture for IPTV security aspects*.

– Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

– Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

– FIPS PUB 140-2 (2001), *Security Requirements for Cryptographic Modules*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation | International Standard uses the following terms defined elsewhere:

3.1.1 Terms from FIPS PUB 140-2

a) key transport
b) tamper detection
c) tamper evidence
d) tamper response.

3.1.2 Terms from Rec. ITU-T Y.2221

a) sensor
b) sensor network
c) USN middleware
d) ubiquitous sensor network (USN).

3.1.3 Terms from Rec. ITU-T H.235.0

a) attack.

3.1.4 Terms from Rec. ITU-T X.1191

a) tamper-resistant.

3.1.5 Terms from Rec. ITU-T X.800 | ISO/IEC 7498-2

This Recommendation | International Standard uses the following terms, which are defined elsewhere:

a) access control
b) authentication
c) authorization
d) confidentiality
e) data origin authentication
f) denial of service
g) digital signature
h) integrity
i) key
j) key management
k) peer-entity authentication
l) privacy
m) repudiation
n) security policy
o) threat.

3.2 Terms defined in this Recommendation | International Standard

For the purposes of this Recommendation | International Standard, the following definitions apply:

3.2.1 aggregator node: Sensor node that performs the data aggregation function in a sensor network.

3.2.2 bootstrapping: Refers to a process performed in a secure context prior to the deployment of the sensor node to establish a security association between the sensor nodes that may have been initialized with credentials, enabling a sensor node to communicate securely with other sensor nodes after their deployment.

3.2.3 credentials: Set of security-related information consisting of keys, keying materials, and cryptographic algorithm-related parameters permitting a successful interaction with a security system.

3.2.4 data aggregation: In-network process that transfers the aggregation value to the sink node by combining the sensed values sent by a number of sensor nodes into concise digest.

3.2.5 group-wise key: Refers to a key that is used to protect multicast communications among a set of sensor nodes over a shared wireless link.

3.2.6 intrusion detection: Process of monitoring the events occurring in a computer system or a network and analysing them for intrusions.

3.2.7 key agreement: A key establishment procedure (either manual or electronic) where the resultant key is a function of information by two or more participants, so that no party can predetermine the value of the key independently of the other party's contribution.

3.2.8 key establishment: Process by which cryptographic keys are securely established among sensor nodes using key transport and/or key agreement procedures.

3.2.9 pair-wise key: It refers to a key that is used to protect unicast communication between a pair of sensor nodes over a single wireless link.

3.2.10 resilience: Ability to recover from security compromises or attacks.

3.2.11 secure data aggregation: Data aggregation that ensures the integrity of the results in the presence of a small number of malicious aggregation nodes that may be attempting to influence the result.

3.2.12 tamper-resistant module: A device designed to make it difficult for attackers to gain access to sensitive information contained in the module.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

BNode	Broadcast Node
BS	Base Station
CDMA	Code Division Multiple Access
DDoS	Distributed Denial of Service
DoS	Denial of Service
ECDH	Elliptic Curve Diffie-Hellman
FP	Feature Parameters
GSM	Global System for Mobile Communications
HSDPA	High Speed Downlink Packet Access
ID	Identity
MAC	Medium Access Control; Message Authentication Code
NGN	Next-Generation Network
PHY	physical layer
RFID	Radio-Frequency IDentification
SN	Sensor Network
TPM	Trusted Platform Module
USN	Ubiquitous Sensor Network
WCDMA	Wideband CDMA
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

5 Conventions

In this Recommendation | International Standard:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation | International Standard is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**is prohibited from**" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation | International Standard is to be claimed.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation | International Standard.

6 Overview

Figure 1 shows the major application areas for USN including home network application, pollution monitoring, fire monitoring, telemetry applications for utility companies (electricity, gas, water, etc.), urban resource monitoring/management applications (e.g., smart city infrastructure), and flood monitoring.

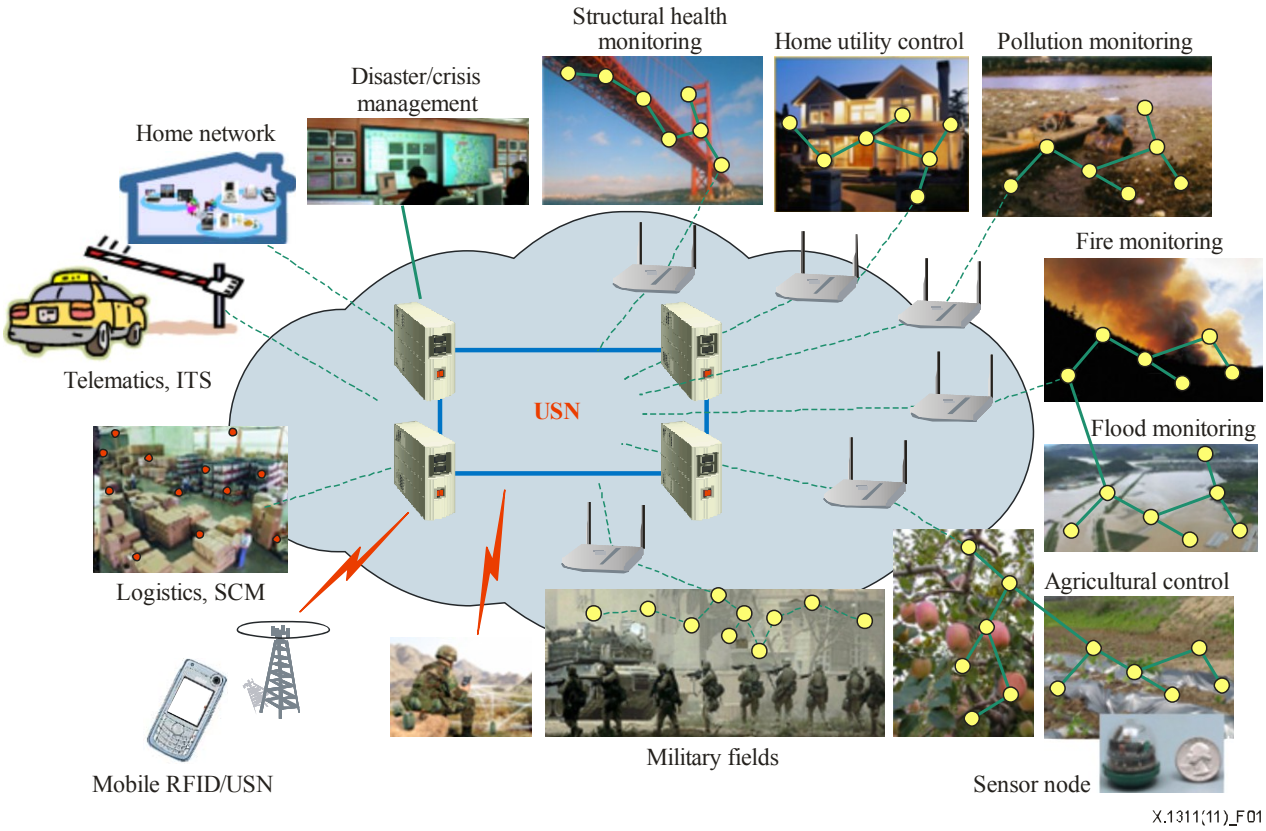

Figure 1 – Application areas for USN

Figure 2 describes the overall structure of USN. Based on such a basic structure, the security model should be defined for USN security.

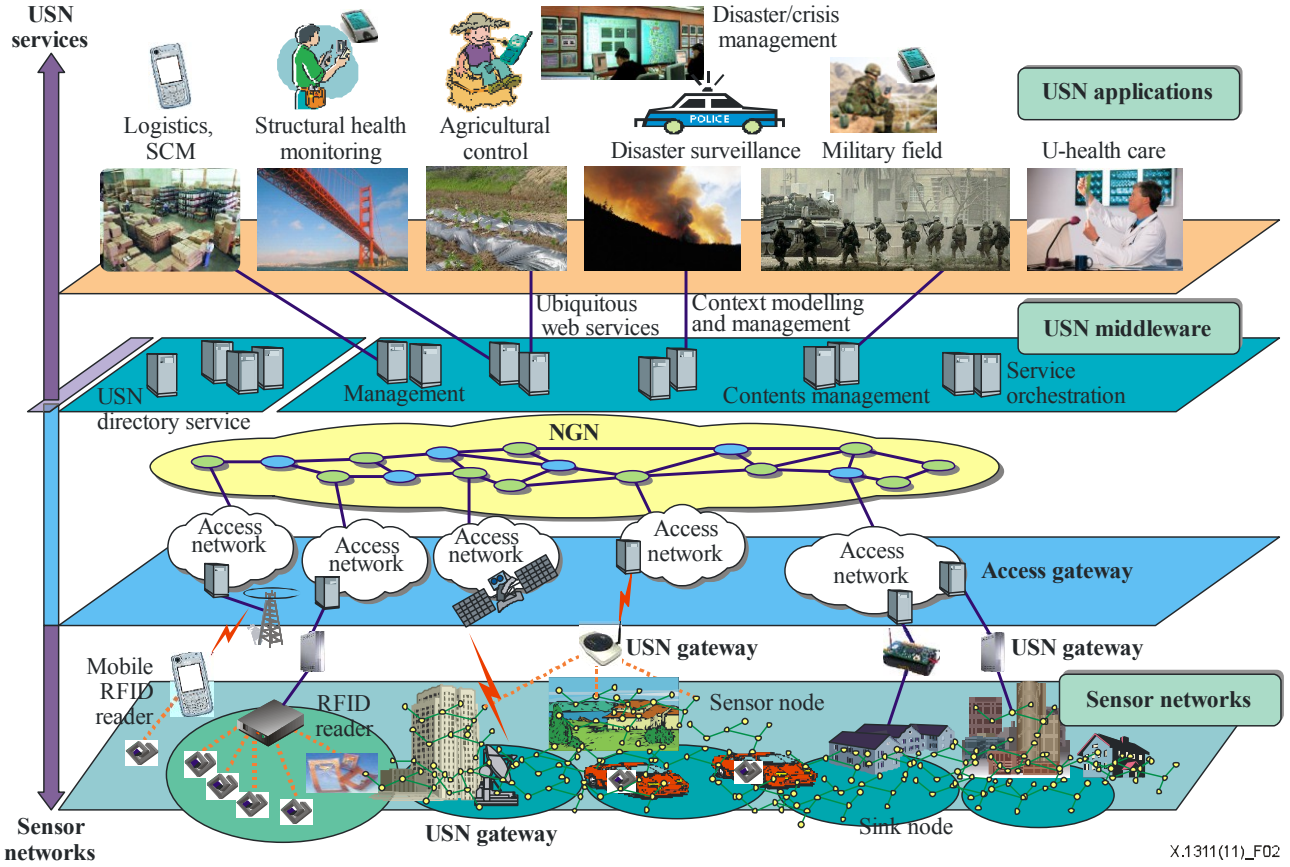

Figure 2 – Overall structure of USN

The sensor networking domain of USN usually corresponds to the sensor node (SN) but includes wire-line sensor networks as well. Thus, many kinds of wired and wireless networking technologies may be used according to the service characteristics and requirements. That is, sensor nodes use different PHY/MAC (e.g., IEEE 802.15.4) layers or operate differently in IP-based or non-IP based networks.

Sensor networks are not isolated but are usually connected to customer networks via various access networks and core networks as shown in Figure 2. The access networking domain corresponds to many access networking technologies, e.g., WLAN, mobile WiMAX, or cellular networks. Core networks include NGN, the Internet, etc. USN may require some extensions and/or additions to core network architectures to cover new functional capability requirements extracted from USN applications and services. For instance, home security monitoring application requires some application-specific functional capability specifications. The USN middleware will consist of many software functionalities such as context models and processing, sensory information gathering, data filtering, contents management, web services functions, network and software management, sensor profile management, directory services, interworking gateways, etc. Based on all these functions, USN applications and services can be established and provided to customers as well as enterprises, organizations, and government.

The security model for USN can be divided into 2 parts: one for the IP network and the other for the wireless sensor network. This Recommendation | International Standard seeks to develop the security model for the SN as well as the IP network.

The communication patterns within our SN fall into five categories:

- Node-to-base station communication, e.g., sensor readings or special alerts.
- Base station-to-node communication, e.g., specific requests.
- Communication between a base station to all sensor nodes, e.g., routing beacons, queries, or reprogramming of the entire sensor network.
- Node-to-node communications including communications among a defined cluster of sensor nodes, e.g., communications between a sensor node and all its neighbours.
- Communications between a base station and a group of nodes wherein the group is defined by nodes sharing a common property (e.g., location, software version, etc.).

The following assumptions can be made:

- The base station is computationally robust, having the requisite processor speed, memory, and power to support the cryptographic and routing requirements of the sensor network. The base station, a gateway which interconnects sensor networks with other networks, may be part of a trusted computing environment.
- Communication is from base station to sensor, from sensor to base station, from sensor to its neighbours, and from node to node.

Therefore, how security technologies are integrated should be taken into account.

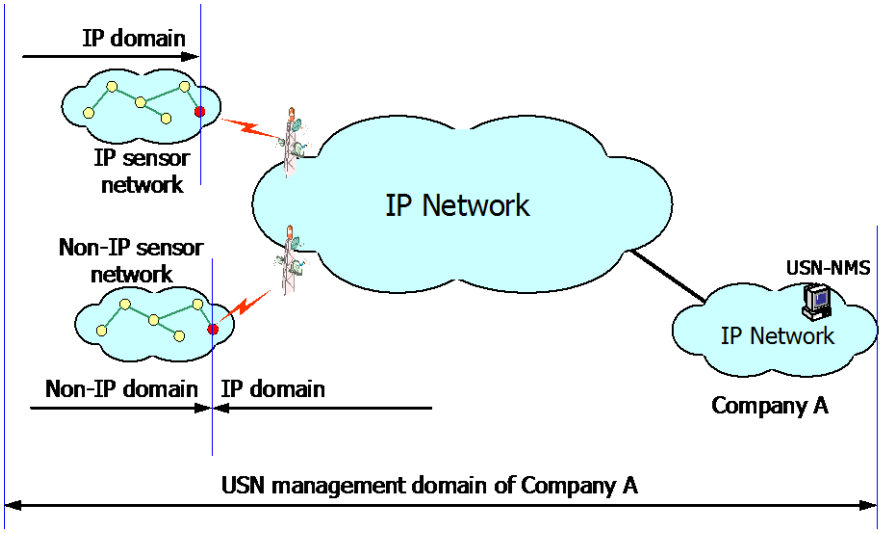

Figure 3 – USN network configuration

The following are the characteristics of the sensor network:

- The sensor network consists of many sensor nodes interconnected by a wireless medium.
- Sensor nodes are deployed densely in a wide area or a hostile context.
- Sensor nodes are vulnerable to failure.
- The communication from the base station (BS) to the sensor node would be of the broadcast type or point-to-point type.
- A sensor node has limited power, computational capacity, and memory.
- A sensor node may not have global identification.

There are three components in the SN: the application server communicating with the sink node; the sink node called the base station, which interfaces the sensor network and the application server, and the collection of sensor nodes using wireless communication to communicate with each other. The sink may communicate with the application server via the Internet or a satellite. Security architecture in the IP-based network is very similar to that in Rec. ITU-T X.805 | ISO/IEC 18028-2. Therefore, this Recommendation | International Standard focuses on the security of the wireless sensor network (SN) consisting of a set of sensor nodes using wireless transmission.

To communicate information between sensor nodes, a secure association between sensor nodes needs to be established before secure communication between them can be realized. Note, however, that the following characteristics of the sensor network render the design of secure communication very difficult:

- **Difficulty of using public key cryptosystems**: The limited computational power, memory size, and power supply make it very difficult to use a public key cryptosystem – such as Diffie-Hellman key agreement or RSA encryption and signature. Even though a specific sensor node may have enough resources to perform the very complex operations required for a public key cryptosystem, it may become vulnerable to a denial of service attack as described in clause 7.1.1.
- **Vulnerability of sensor nodes**: Since sensor nodes may be deployed in hostile locations, their security may be compromised. After obtaining physical access to the sensor node, the attacker is able to access sensitive information such as key information or sensed information. This attack can be prevented by using a tamper-resistant sensor node, which entails a high cost. Moreover, a large number of sensor nodes render the employment of the tamper-resistant sensor node very difficult since it may result in a

high-cost network. For some applications (e.g., military, safety-critical applications, etc.), however, the higher costs incurred in employing tamper-resistant sensor nodes may be acceptable.

- **Difficulty in obtaining post-deployment knowledge**: In most cases, the sensor nodes will be deployed in a randomly scattered manner; hence, the difficulty for the security protocol to know the location of neighbouring nodes.
- **Limited memory size, transmission power, and transmission bandwidth**: Since the memory in sensor nodes is limited, storing the unique keys used with other sensor nodes in the network is very difficult. Moreover, a typical sensor node has low capability in terms of transmission bandwidth and power to communicate with neighbour nodes.
- **Single point of failure of a base station**: In sensor networks, a base station is a gateway to communicate the sensed information to an application server through the IP-based core network. The security of the sensor network relies on that of the base station. Therefore, base stations could become an appealing target for various types of attack.

Figure 4 presents a general model of an end-to-end communication between a sensor node in the SN and an AS. There may be cases wherein the application server resides in the gateway.

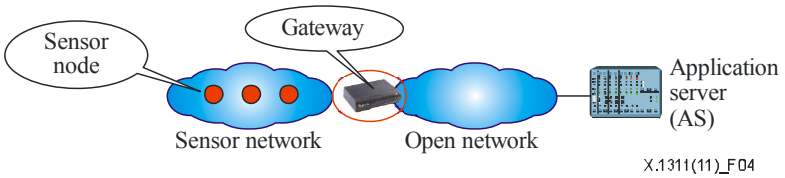

Figure 4 – General model of an end-to-end communication between a sensor node in the SN and an AS

The potential layers for implementing the security function in USNs is shown in Figure 5. The basic security layer, corresponding to MAC or link layer, is responsible for the link-by-link data transfer between sensor nodes or between the sensor node and a gateway. The service layer, corresponding to the network layer, is responsible for network data transfer between sensor nodes and between a sensor node and a gateway. Typical examples of the service layer include the transfer of broadcast messages from the gateway to the sensor node and vice versa. The security of the service layer and the basic function layer should implement the security functions described in clause 10, corresponding to the network layer and link layer security, respectively. Note that for some applications, the application security function resides in a gateway rather than in an application server.

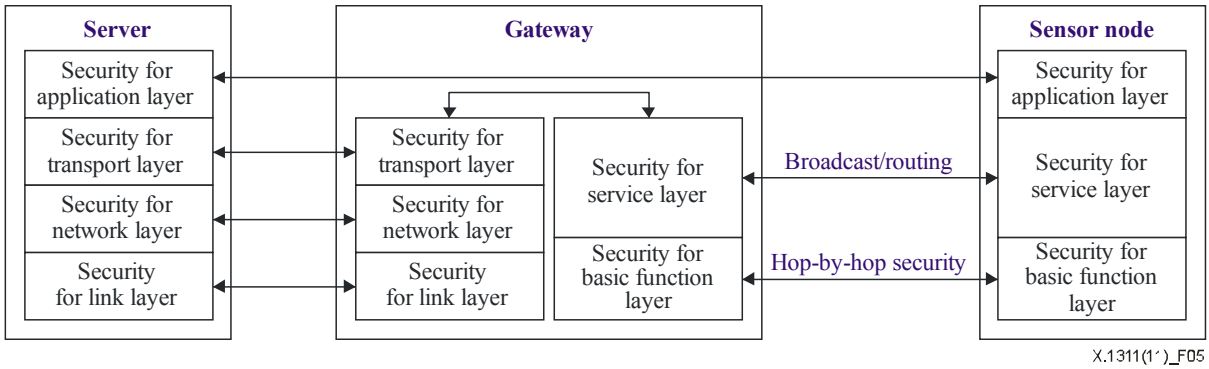

Figure 5 – Layers implementing security functions for USN

7 Threats and security models for ubiquitous sensor networks

The threats to USNs can be classified into threats to the IP network and threats to the SN.

7.1 Threat models in sensor networks

There are two types of attackers in the SN: a mote-type attacker and a laptop-type attacker. In the former, the attacker has a capability similar to the sensor node; it can have access to a few sensor nodes. An attacker with a mote-type device may be able to jam the radio link in its vicinity. In the latter, an attacker may have access to more powerful devices such as a laptop computer. An attacker with a laptop-type device may eavesdrop on the communication in the

sensor network and have high-bandwidth, low-latency communications channel; it can also jam the entire sensor network using a high-power transmitter. There are two types of threats for the SN: general threats and routing-related threats. The threats in the SN are applied to the communication between the base station and the sensor node and between the nodes, as described in clause 7.1.1. Routing-related threats are applied to the routing message exchange, as described in clause 7.1.2.

7.1.1 General threats in sensor networks

Rec. ITU-T X.800 | ISO/IEC 7498-2 and Rec. ITU-T X.805 | ISO/IEC 18028-2 cite the following security threats to the networks (note that these are also security threats applicable to the SN):

- Destruction of information and/or other resources
- Corruption or modification of information
- Theft, removal, or loss of information and/or other resources
- Disclosure of information
- Interruption of services.

In addition to these, there are many sensor node-specific threats such as sensor node vulnerability, eavesdropping, privacy of sensed data, denial of service attack, and malicious use of commodity network (see Chan *et al*. in the Bibliography).

- **Vulnerability of sensor nodes**: Sensor networks are expected to consist of hundreds or thousands of sensor nodes. Each node represents a potential point of attack, rendering the monitoring and protection of each individual sensor from either a physical or a logical attack impractical. The networks may be dispersed over a large area, further exposing them to attackers capturing and reprogramming individual sensor nodes. Attackers can also obtain their own commodity sensor nodes and induce the network to accept them as legitimate nodes, or they can claim multiple identities for an altered node. Once in control of a few nodes inside the network, the attacker can then mount a variety of attacks such as falsification of sensor data, extraction of private sensed information from sensor network readings, and denial of service. Addressing the problem of sensor node vulnerability requires technological solutions. For example, cheap tamper-resistant hardware could pose a challenge to reprogramming the captured sensor nodes. Still, making nodes robust to tampering is not economically viable. Therefore, we should assume that an attacker can compromise a subset of the sensor nodes. As such, at the software level, sensor networks need new capabilities to ensure secure operation even in the presence of a small number of malicious network nodes. *Node-to-node authentication* is one basic building block for enabling network nodes to prove their identity to each other. *Node revocation* can then exclude malicious nodes. Achieving these goals, given the resource-limited hardware, will require lightweight security protocols. Furthermore, all communication and data-processing protocols used in sensor networks must be made *resilient*, i.e., be able to function at high effectiveness even with a small number of malicious nodes. For example, routing protocols must be resilient against compromised nodes that behave maliciously.
- **Eavesdropping**: In wireless sensor network communications, an adversary can gain access to private information by monitoring transmissions between nodes. For example, a few wireless receivers placed outside a house may be able to monitor the light and temperature readings of sensor networks inside the house, thus revealing detailed information on the occupants' personal daily activities. Encrypting sensor node communications partly solves eavesdropping problems, but requires a robust key exchange and distribution scheme. The scheme must be simple for the network owner to execute and be feasible for the limited sensor node hardware to implement. It must also maintain secrecy in the rest of the network when an adversary compromises a few sensor nodes and exposes their secret keys. Ideally, these schemes would also allow the revocation of known exposed keys and rekeying of sensor nodes. The large number of communicating nodes makes end-to-end encryption usually impractical since sensor node hardware can rarely store a large number of unique encryption keys. Instead, sensor network designers may choose hop-by-hop encryption wherein each sensor node stores only encryption keys shared with its immediate neighbours. In this case, adversary control of a communication node eliminates the encryption's effectiveness for any communication directed through the compromised node. This situation could be exacerbated if an adversary manipulates the routing infrastructure to send many communications through a malicious node. More robust routing protocols serve as one solution to this problem. Another solution is *multipath routing*, which routes parts of a message over multiple disjoint paths and reassembles them at the destination. Multipath routing may enhance USN's resilience to attacks or compromises. The efficient discovery of the best disjoint paths to use for such an operation poses another research challenge. Note that this threat happens in the fixed core network and has relevance to privacy infringement in a sensor network.

- **Secrecy of sensed data**: Sensor networks are tools for collecting information; an adversary can gain access to sensitive information either by accessing stored sensor data or by querying or eavesdropping on the network. Adversaries can use even seemingly innocuous data to derive sensitive information if they know how to correlate multiple sensor inputs. For example, an adversary gaining access to both indoor and outdoor sensors of a home may be able to isolate internal noise from external noise and consequently extract details of the inhabitants' private activities. However, the fact that sensor networks enable the collection of information that would otherwise be impossible to collect is not the main privacy problem. In fact, a lot of information from sensor networks could probably be collected through direct site surveillance. Sensor networks exacerbate the privacy problem because they make large volumes of information easily available through remote access. Thus, attackers need not be physically present to maintain surveillance. They can gather information in a low-risk, anonymous manner. Remote access also allows a single adversary to monitor multiple sites simultaneously. Ensuring that sensed information stays within the sensor network and remains accessible only to trusted parties is an essential step to ensuring privacy. Data encryption and access control serve as one approach. Another way is to restrict the network's ability to gather data in details in such a way that could compromise privacy. For example, a sensor network might anonymize data by reporting only aggregate temperatures over a wide area or approximate locations of the sensed individuals. A system stores the sensed data in an anonymized database, removing details that an adversary might find useful. Another approach is to process queries in the sensor network in a distributed manner so that no single node can observe the query results in their entirety. This approach guards against potential system abuse by compromised malicious nodes.
- **DoS attacks**: As safety-critical applications use more sensor networks, the potential damage of operational disruptions becomes significant. Defending against denial-of-service attacks – which aim to destroy network functionality rather than subverting it or using the sensed information – is extremely difficult. DoS attacks can occur at the physical layer, e.g., via radio jamming. They can also involve malicious transmissions into the network to interfere with sensor network protocols or physically destroy central network nodes. Attackers can induce battery discharge in sensor nodes – for example, by sending a sustained series of useless communications that make the targeted nodes expend energy in processing them and forwarding them to other nodes as well. More insidious attacks can occur from inside the sensor network if attackers can compromise the sensor nodes. For instance, they could create routing loops that will eventually exhaust all nodes in the loop. Potential defenses against denial-of service attacks are as varied as the attacks themselves. Techniques such as spread-spectrum communication or frequency hopping can counter jamming attacks. Proper authentication can also prevent injected messages from being accepted by the network. Note, however, that the protocols involved must be efficient so that they themselves do not become targets of an energy-exhaustion attack. For example, using signatures based on asymmetric cryptography can provide message authentication. However, the creation and verification of asymmetric signatures are highly computationally intensive, and attackers that can induce a large number of these operations can mount an effective energy-exhaustion attack.
- **Malicious use of commodity networks**: The proliferation of sensor networks will inevitably extend to criminals who can use them for illegal purposes. For example, thieves can hack home automation sensors or even simply eavesdrop on their activity to gain private information on the presence, location, etc., of the owners and act accordingly. If the sensors are small enough, they can also be planted on computers and cell phones to extract private information and passwords. Such widespread use will lower the cost and availability barriers that are supposed to discourage such attacks. Sensor detectors offer one possible defense against such attacks. A detector must not only be able to detect the presence of potentially hostile wireless communications within an area that may have significant levels of radio interference but also be able to differentiate between the transmissions of authorized and unauthorized sensor networks and other devices. Although such technologies may not prevent unauthorized parties from deploying sensor networks in sensitive areas, they would make them more costly, thus alleviating the problem.

7.1.2 Routing-specific threats

Rec. ITU-T X.800 | ISO/IEC 7498-2 and Rec. ITU-T X.805 | ISO/IEC 18028-2 identify five threats that are applicable to routing-related message exchange in the SN. In addition to these, seven threats are identified in (see Karlrof *et al.* in the Bibliography) with regard to the routing messages exchanged between sensor nodes.

- **Spoofed, altered, replayed routing information**: The attacker is able to spoof, alter, and replay the routing information, enabling the creation of a routing loop, attracting network traffic, extending source routing, and increasing end-to-end latency.
- **Selective forwarding**: It refers to the attack wherein a compromised node by an attacker may refuse to forward certain messages and drop them, stopping further propagation.
- **Sinkhole attack**: It pertains to the attack wherein the attacker attracts all the traffic from a particular area through a compromised node.

- **Sybil attacks**: It refers to the attack wherein a single node presents multiple identities to other nodes in the network, convincing every node that an adversary exists in more than one place at once.
- **Wormhole attacks**: In a wormhole attack, an adversary tunnels messages received in one part over a low-latency link and replays them in a different part. Wormhole attacks will involve two distinct malicious nodes colluding to understate their distance from each other by replaying the packet along an out-of-band channel available only to the attackers.
- **HELLO flood attacks**: It pertains to the attack wherein a laptop-type attacker broadcasts HELLO packets convincing every node in the network that the adversary is its neighbour.
- **Acknowledgment spoofing**: In acknowledgment spoofing, an adversary can spoof link layer acknowledgment for "overheard" packet addressed to the neighbouring node, convincing the sender that a weak link is strong, or a dead or a disabled node is alive.

Note that simple attacks exploiting any of the threats described above may even be used in combination to create more complex attacks (e.g., combination of Sybil and Sinkhole or Wormhole).

7.2 Threat models in IP networks

The threat models developed in Rec. ITU-T X.805 | ISO/IEC 18028-2 can be applied to the IP network. Therefore, refer to Rec. ITU-T X.805 | ISO/IEC 18028-2 for the details of those threats.

7.3 Security model for USNs

The security model shown in Figure 6 demonstrates a general framework of USN security based on the application area of the USN, the overall structure of the USN, and the USN network configuration. The model is based on ISO/IEC 15408-1 to help establish security concepts and relationships in USN security. When the USN is threatened, appropriate security policies should be selected to achieve the security objectives by deploying the proper security technologies and special security functional requirements.

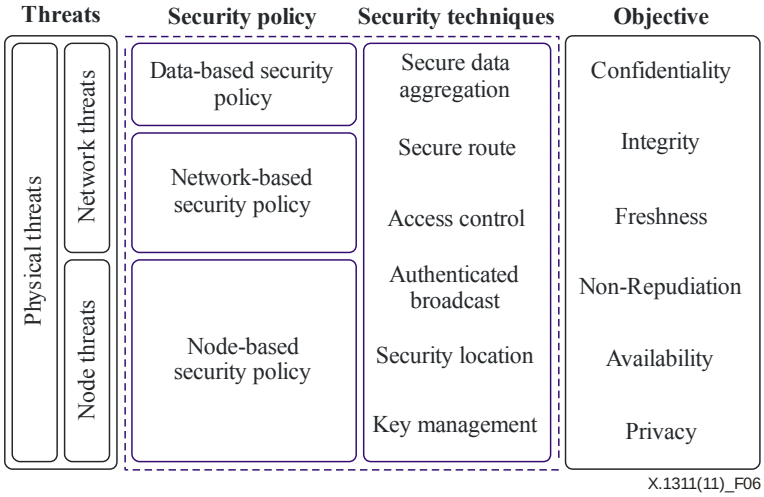

Figure 6 – Security model for USN

8 General security dimensions for USN

A dimension is a set of security measures designed to address a particular aspect of network security to protect against all major security threats; it is not limited to the network but extends to applications and end user information as well.

To counter the aforesaid threats in both the SN and the IP networks, the following security dimensions in Rec. ITU-T X.805 | ISO/IEC 18028-2 are applicable:

- **Data confidentiality**: A sensor network should not leak sensor readings to neighbouring networks. In many applications (e.g., key distribution), nodes communicate highly sensitive data. The standard approach for keeping sensitive data confidential is to encrypt the data with a secret key that only the intended receivers possess, thus ensuring confidentiality.

- **Data authentication/identification**: Message authentication is important for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g., network reprogramming or controlling the sensor node duty cycle). At the same time, an adversary can easily inject messages; thus, the receiver needs to make sure that the data used in any decision-making process originates with the correct source. Informally, *data authentication* allows a receiver to verify that the data was really sent by the sender claiming to be such. Identification aims at proving the identity of the entity or sensor node. In the two-party communication case, data authentication can be realized through a purely symmetric mechanism: the sender and the receiver share a secret key to compute the message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it is sent by the sender. This style of authentication cannot be applied to a broadcast setting without making much stronger trust assumptions on the network nodes. Sending authenticated data to mutually untrusted receivers, using a symmetric MAC, is insecure since any one of the receivers knows the MAC key, he/she could consequently impersonate the sender and forge messages to other receivers. Therefore, an asymmetric mechanism is needed to enable authenticated broadcast. One method is to construct authenticated broadcasts from symmetric primitives only and introduce asymmetry with delayed key disclosure and one-way function key chains. Authentication mechanisms are mainly used to validate the legitimacy of the node so that its legitimacy and credibility are ensured.
- **Data integrity**: In communication, *data integrity* assures the receiver that the received data is not altered in transit by an adversary.
- **Access control**: Access control ensures that only the authorized user or entity is allowed to gain access to information, resource, or services.
- **Non-repudiation**: Non-repudiation ensures that the entity or user cannot deny the activities in the network he/she has done.
- **Communication security**: Communication security ensures that the information only flows from the source to the destination.
- **Availability**: Availability ensures that information, service, and application are available to legitimate users anytime.
- **Privacy**: Privacy ensures that the identifier of the user or entities and network usage is kept confidential.

In addition to the security dimensions described above, a dimension for "Resilience to Attacks" through the appropriate design of PHY/MAC/routing protocols should be added for the SN part only.

- **Resilience to attacks**: This refers to any of the measures for recovering from the various attacks against the USN. It ensures that USN is able to recover from attacks so that it is capable of detecting/remaining resilient to various attacks through the appropriate design of PHY/MAC/Routing protocols. Resilience to attacks include resilience against compromised nodes, resilience against eavesdropping on routing information, etc.

9 Security dimensions and threats in ubiquitous sensor networks

Message exchange in the SN can be grouped into three types: message exchange between nodes, message exchange between a base station and a node, and message exchange for routing-related messages.

9.1 Security dimensions and threats for the message exchange in sensor networks

9.1.1 Security dimensions and threats for the message exchange between sensor nodes

Table 1 lists the security requirements and describes the relationship between the security dimensions and the security threats identified in Rec. ITU-T X.805 | ISO/IEC 18028-2. The letter "Y" in a cell formed by the intersection of the table's columns and rows suggests that a particular security threat is opposed by the corresponding security dimension.

Table 1 – Mapping of security dimensions to security threats

Security dimension	Security threat				
	Destruction of information or other resources	Corruption or modification of information	Theft, removal, or loss of information and other resources	Disclosure of information	Interruption of services
Access control	Y	Y	Y	Y	
Authentication			Y	Y	
Non-repudiation	Y	Y	Y	Y	Y
Confidentiality			Y	Y	
Communication security			Y	Y	
Data integrity	Y	Y			
Availability	Y				Y
Privacy				Y	

Table 2 lists the security dimensions and describes the relationship between security dimensions and sensor node-specific threats for the message exchange between nodes or information stored in sensor node. The letter "Y" in a cell formed by the intersection of the table's columns and rows suggests that a particular security threat is opposed by the corresponding security dimension. For example, the threat of sensor node compromise could be addressed by employing physical access control to limit access to the resource of the sensor to the authorized user or entity only, using node-to-node authentication to prove one's identity to the other and vice versa, using data processing and communication protocol resilient to a small number of compromised sensor nodes, and/or employing routing protocols resilient to compromised nodes that behave maliciously. The threat of privacy infringement of sensed data could be addressed by employing data encryption and access control resulting in the protection of sensed data accessible to the authorized node only, processing sensed data transfer in a distributed manner so that no single node observes the sensed data, and/or selecting a secure routing path for transferring the sensed data. The threat of denial-of-service attacks could be addressed by employing the proper authentication of the sensed data and designing the security protocol involved to be resilient to DoS attacks and/or spread-spectrum communication technology to counter any jamming attack.

Table 2 – Security dimensions to sensor node-specific threats

Security dimensions	Sensor node-specific threats			
	Sensor node compromise	Secrecy of sensed data	DoS	Malicious use of commodity network
Access control	Y	Y		Y
Authentication	Y		Y	Y
Non-repudiation				
Confidentiality		Y		Y
Communication security	Y	Y		
Data integrity				
Availability			Y	
Privacy		Y		Y
Resilience to attacks	Y	Y	Y	Y

9.1.2 Security dimension and threats for messages broadcast from a base station to all sensor nodes

Table 3 lists the security dimensions and describes the relationship between the security dimensions and the security threats against the messages broadcast from a base station to all the sensor nodes. The letter "Y" in a cell formed by the intersection of the table's columns and rows suggests that a particular security threat is opposed by the corresponding security dimension.

Table 3 – Security dimensions to security threats against broadcast messages

Security dimension	Security threats against messages broadcast from a base station to all nodes				
	Destruction of information	Corruption or modification of information	Theft, removal, or loss of information	Disclosure of information	Interruption of services, DoS
Access control	Y	Y	Y	Y	
Authentication		Y	Y	Y	
Non-repudiation	Y		Y		Y
Confidentiality			Y	Y	
Communication security			Y	Y	
Data integrity	Y	Y			
Availability	Y				Y
Privacy				Y	
Resilience to attacks	Y	Y	Y	Y	Y

9.1.3 Security dimension and threats for the routing message exchange

The threats can be classified into two categories: insider attacks and outsider attacks. Insider attacks can be launched by the insider, i.e., the attacker has some knowledge of some sensitive information stored in some sensor nodes, i.e., key information related to routing message exchange. Insider attacks may be related to the Sybil attack, HELLO flood attack, wormhole and sinkhole attack, selective forwarding attack, and DoS attack. Table 4 lists the security dimensions and describes the relationship between the security dimensions and the security threats of the routing message exchange launched by an insider attack. The letter "Y" in a cell formed by the intersection of the table's columns and rows designates that a particular security threat is opposed by the corresponding security dimension. The insider is able to compromise the sensor node, i.e., an authorized participant in the sensor node performs malicious activities, as well as launch active and passive attacks. Insider attacks may be launched from either compromised sensor nodes running malicious code or adversaries who have stolen the key material, code, and data from legitimate sensor nodes. The insider is able to attack the SN by spoofing or injecting bogus routing information, creating sinkholes, selectively forwarding packets, using Sybil attack, and broadcasting HELLO floods. Link layer security mechanisms using a globally shared key may be inefficient in the presence of insider attacks or compromised nodes.

For example, the Sybil attack can be addressed by using routing message authentication and encryption utilizing the resulting key as the outcome of a third-party key distribution scheme such as Needham-Schroeder, and by limiting the number of neighbours that a node is allowed to have. Most insider attacks are prevented using a trusted platform module (TPM) in the sensor node.

Table 4 – Mapping of security dimensions to security threats of an insider attack

Security dimension	Security threat					
	Sybil attack	HELLO flood	Sinkhole	Selective forwarding	Wormhole	Acknow-ledgment spoofing
Access control						
Message authentication	Y		Y	Y		Y
Identification	Y		Y	Y		Y
Non-repudiation						
Confidentiality	Y		Y	Y		Y
Communication security						
Data integrity						
Availability						
Privacy						
Resilient technology	Y (third-party key distribution, TPM)	Y (verification of link bidirectionality)	Y (multipath, TPM)	Y (multipath, TPM)	Y (tight time synchronization)	Y (third-party key distribution, TPM)

Table 5 lists the security dimensions and describes the relationships between the security dimensions and the security threats of the routing message exchange launched by an outsider. The letter "Y" in a cell formed by the intersection of the table's columns and rows suggests that a particular security threat is opposed by the corresponding security dimension. Although the outsider is able to launch active and passive attacks, the attacker has no special access to the sensor network, i.e., he/she cannot compromise the sensor node. Most outsider attacks against the sensor network routing protocol can be prevented by simple link layer encryption and authentication using a globally shared key. The Sybil attack is no longer relevant because nodes will not accept a single identity of the adversary. Therefore, for example, the Sybil attack can be addressed by verifying the identity of the sensor node or by using message authentication or through confidentiality. Selective forwarding and sinkhole attacks may be impossible because the adversary is prevented from joining the topology. Link layer acknowledgment should be protected by employing message authentication. Multipath routing can be used to counter selective forwarding attacks. Messages routed over n paths are protected against selective forwarding attacks involving up to n compromised nodes.

Table 5 – Mapping of security dimensions to outsider security threats

Security dimension	Security threat					
	Sybil attack	**HELLO flood**	**Sinkhole**	**Selective forwarding**	**Wormhole**	**Acknow-ledgment spoofing**
Access control						
Message authentication	Y		Y	Y		Y
Identification	Y		Y	Y		Y
Non-repudiation						
Confidentiality	Y		Y	Y		Y
Communication security				Y		
Data integrity						
Availability						
Privacy			Y	Y	Y	
Resilient technology			Y (Multipath)	Y (multipath)		

9.2 Security dimension and threats for the message exchange in the IP network

The security threats and security dimensions developed in Rec. ITU-T X.805 | ISO/IEC 18028-2 can directly be applied to a secure message exchange through the IP network. Therefore, refer to Rec. ITU-T X.805 | ISO/IEC 18028-2 for related details.

10 Security techniques for ubiquitous sensor networks

10.1 Key management

Key management refers to the generation, distribution, sharing, rekeying, and revocation of cryptographic keys for the data confidentiality service, data integrity, data freshness, and data authentication in the SN. The security of key management forms the foundation of the security of other security services. In the sensor network, sharing or distributing a pair-wise key between the sensor nodes and a group-wise key among a set of sensor nodes is very important. It is sometimes called key agreement scheme.

In general, there are three types of key agreement: trusted server scheme, self-enforcing scheme, and key pre-distribution scheme. The trusted server scheme uses the central trusted server to share the pair-wise key between the sensor nodes or the group-wise key among the sensor nodes. A typical example of this scheme is Kerberos. Note, however, that this type of scheme is not adequate for the sensor network since there is no trusted infrastructure in the sensor network. The self-enforcing scheme uses the public key algorithm to share the pair-wise key or group-wise key in the sensor network. Typical algorithms of the public key algorithm include the Diffie-Hellman key agreement algorithm and the RSA key transport algorithm. Note, however, that this scheme cannot be employed in the sensor network due to the limited memory and computational complexity of the sensor node. The key pre-distribution scheme pre-distributes the key information among all sensor nodes prior to deployment. The deployment of most sensor nodes

is random. In other words, a priori knowledge as to the exact location of the sensor node is assumed to be unknown prior to deployment. This scheme has low communication overhead. In addition, it is resilient to node compromise, and it does not rely on the trust of the base station. Therefore, this scheme is very suitable for the wireless sensor network.

There are a number of key pre-distribution schemes that do not assume to have knowledge of deployment of the sensor node. The simple scheme is a master key-based pre-distribution one. In this scheme, all nodes have a single common master key that is pre-deployed to each sensor node. Any two nodes use this global master key to obtain the common pair-wise key by exchanging random nonces. This scheme does not provide desirable resilience to node compromise since the entire sensor network is compromised if a node is compromised. The second scheme is called pair-wise key pre-distribution scheme. This scheme is designed to let each sensor node have N−1 secret pair-wise keys each of which is known only to this sensor node and one of the other N−1 sensor nodes, where N is the total number of sensor nodes in the network. This scheme renders perfect resilience against node compromise since a compromised node does not affect the security of any other node. Nonetheless, it lacks scalability since adding new nodes to the existing sensor node is impossible owing to the absence of a new pair-wise key in the existing node. In addition, this scheme is not practical since the memory size is limited when the number of sensor nodes is very large. The third scheme is the random key pre-distribution scheme. In this scheme, the subset of keys from a large key pool is stored prior to the deployment of sensor node; two nodes find a common key and use it as a shared session key between them.

In the random key schemes presented so far, however, while each node can verify that some of its neighbours have certain secret keys, and that they are consequently legitimate nodes, no node can authenticate the identity of a neighbour with which it is communicating.

The following are the capabilities of key management supporting node-to-node authentication in a sensor network:

Key management supports node-to-node authentication.

- **Scalable key management**: The key management scheme supports a large sensor network. In addition, it should be flexible when there is a substantial increase in sensor nodes even after the deployment of the sensor node.
- **Efficiency of memory size, processing capability, and communication overhead required for key management**: The key management scheme has efficient storage complexity, i.e., minimum memory size to store the key in the sensor node, efficient computation complexity required to establish the key, efficient communication overhead, i.e., number of messages exchanged during the key generation process.
- **High probability for pair-wise key establishment**: In the key management scheme, the two sensor nodes have high probability of establishing the common key and key material.
- **Resilience against compromised nodes**: The key management scheme has the capability to resist compromised nodes. A compromised security credential should not reveal even the minimum information on the security of other links in the sensor network, i.e., higher resilience means lower number of compromised links. Note that resilience should not be limited to key management issues but should be applied to all sensor nodes. A typical example includes resilience against eavesdropping on routing information. Details of key management are described in Annex A.

10.2 Authenticated broadcast

This is important since broadcasts are used in many applications in sensor networks. For example, routing tree construction, network query, software updates, time synchronization, and network management all rely on broadcast. Due to the nature of wireless communication in sensor networks, however, attackers can easily inject malicious data or alter the content of legitimate messages during multi-hop forwarding. Sensor network applications need authentication mechanisms to ensure that data from a valid source will not be altered during transmission. Broadcast authentication is one of the most important security primitives in sensor networks.

A broadcast message targets all sensor nodes. A broadcast message authentication scheme allows any targeted node to verify the authenticity of the broadcast messages. Two kinds of techniques can be used to achieve this target according to the type of cryptographic algorithm. In the case of public key cryptography, a digital signature can be used. If symmetric cryptography is used, there is a need to append to the data the verifiable authentication data (i.e., message authentication code) based on the multiple shared secret between the base station (sink node) and sensor node. Due to the properties of the sensor network, the broadcast authentication method is preferred in broadcast message authentication based on symmetric cryptography.

There is a typical scheme for enabling broadcast authentication in sensor networks, called TESLA (timed efficient stream loss-tolerant authentication) (see IETF RFC 4082 and Adrian in the Bibliography). TESLA supports delayed per-packet data authentication and integrity checking. The key idea is the delayed disclosure of symmetric keys. The delayed key disclosure results in authentication delay. TESLA has the following properties: low computation overhead for the generation and verification of authentication information, low communication overhead, limited buffering

required for the sender and the receiver, high robustness to packet loss, scales to a large number of receivers, and protection of receivers from denial of service. TESLA makes the following assumptions:

- The base station and the sensor node should be loosely time-synchronized.
- TESLA should be bootstrapped at session setup through a regular data authentication system.

As the simplified version of TESLA, the μTESLA protocol (see Perrig *et al*.) basically uses the delayed disclosure of symmetric key. The base station and the sensor nodes are assumed to be time-synchronized loosely. The operation is as follows: the base station computes the MAC on the packet with the key that is secret at that point of time. When a node receives a packet, it can confirm that the base station has not yet disclosed the corresponding MAC key according to its loosely synchronized clock and time when the keys are to be disclosed. The node stores the packet in its buffer. When the MAC keys are to be disclosed, the base station broadcasts the MAC keys to all sensor nodes. The sensor node can verify the authenticity of the broadcast message by using the disclosed MAC keys and MAC data stored in the buffer. Each MAC key is a member of a key chain that has been generated by a one-way function. For the base station to generate this key chain, the base station chooses the last key K_n of the key chain randomly and applies the one-way hash function H repeatedly to compute all other keys: $K_i = H(K_{i+1})$, $i = 1, \ldots, n–1$. The sensor node, which shares K_1 with the base station, can verify the correctness of the key and use the disclosed MAC keys and MAC data stored in the buffer to authenticate the packet stored in the buffer. The following are the differences between TESLA and μTESLA:

- TESLA authenticates the initial packet with a digital signature, which is too expensive for sensor nodes. μTESLA uses only symmetric mechanisms.
- Disclosing a key in each packet requires too much energy for sending and receiving. μTESLA discloses the key once per epoch.
- Storing a one-way key chain in a sensor node is expensive. μTESLA restricts the number of authenticated senders.

The following are the capabilities of authenticated broadcasts in a sensor network:

- **Resilience against compromised nodes**: Since tamper-proof hardware will unlikely be deployed on sensor nodes in the near future, secure sensor network protocols need to be resilient against compromised nodes. In addition, being resilient to attacks necessitates the avoidance of single points of failure. For instance, a base station is a single point of failure. Tamper-proof base stations should be combined with multipath routing (even through access points) so that base station security failures may be less compromising.
- **Low computation overhead**: Sensor nodes have limited computation resources; thus, an ideal protocol would have low computation overhead for both the sender and the receiver.
- **Low communication overhead**: Energy is an extremely scarce resource in sensor nodes. In particular, radio communication consumes the greatest amount of energy. Thus, protocols with high communication overhead are avoided, if possible.
- **Robustness to packet loss**: Reliable message delivery is the property of a network wherein valid messages are not dropped.
- **Immediate authentication**: Depending on the application, authentication delay may influence the design of the sensor network protocol. For time-critical messages such as fire alarms, the receiver would most likely need to authenticate the message immediately. Note, however, that authentication delay is typically acceptable for non-time-critical messages.
- **Messages sent at irregular times**: Some applications send synchronous messages at regular and predictable times, while others do not. In some situations, sending messages asynchronously (e.g., presence detector) should be avoided to prevent the creation of covert channels. In other cases, having synchronous (i.e., periodic) communication activity should be avoided to prevent security threats due to the predictability of network communication behaviour.

μTPC is another improved version of TESLA. The details of μTPC for authenticated broadcast in sensor networks are described in Annex B.

10.3 Secure data aggregation

Data aggregation is a widely used technique in wireless sensor networks.

The security issues, data confidentiality, and integrity in data aggregation become vital when the sensor network is deployed in a hostile environment. There have been many related works proposed to address these security issues.

Secure data aggregation refers to an in-network process performed on the aggregator node to transfer securely the aggregation value to the sink node (i.e., a base station) by combining the sensed values sent by a number of sensor nodes. In this scheme, each sensor node sends an encrypted sensed value to the aggregator, which then calculates the

encrypted aggregator results using aggregation functions such as summing function, average function, median function, and maximum value or minimum value; the sink node obtains the aggregation value by decrypting the encrypted aggregator results.

Therefore, it is more useful for the base station or a sensor node to have the capability to aggregate data than the individual value from all sensors. By aggregating data, reducing the amount of data that needs to be transmitted from one sensor to another sensor can be reduced. Secure data aggregation can be applied to sensors deployed in a hierarchical structure.

There are two kinds of secure aggregation methods: hop-by-hop encrypted data aggregation and end-to-end encrypted data aggregation.

- **Hop-by-hop encrypted data aggregation**: The operation is based on hop-by-hop encryption between neighbour sensor nodes. Each pair of neighbour sensor nodes is assumed to share a common secret key. The sensed values are encrypted by many sensor nodes. All encrypted values collected are decrypted by the intermediate aggregator nodes, which then obtain and re-encrypt the aggregated value. This process is repeated up to the sink node. Finally, the sink node obtains the total aggregated value.
- **End-to-end encrypted data aggregation**: The operation is based on the end-to-end encryption between many sensor nodes and one sink node. A common secret is assumed to be shared between many sensor nodes and a sink node. The sensing nodes encrypt the sensed values and forward them to intermediate aggregator nodes, which only collect them, perform some cryptographic operation on the aggregated values, and forward them since they do not have decryption keys. Finally, the sink node decrypts many encrypted aggregated results.

The framework for end-to-end encrypted data aggregation is known to incur higher computation cost on the sensor nodes but achieves stronger security compared with the framework for hop-by-hop encrypted data aggregation.

10.4 Data freshness

Since all sensor networks stream some forms of time-varying measurements, guaranteeing confidentiality and authentication is not enough; one must also ensure that each message is *fresh*. Informally, data freshness implies that the data is recent and ensures that no adversary replayed old messages. We identify two types of freshness: weak freshness, which provides partial message ordering but carries no delay information, and strong freshness, which provides a total order on a request-response pair and allows for delay estimation. Weak freshness is required by sensor measurements, whereas strong freshness is useful for time synchronization.

10.5 Tamper-resistant module

The best well-known technique to protect against sensor node compromise is to use the tamper-resistant module in the sensor node. If each sensor node is equipped with a tamper-resistant module, protecting the storage of sensitive data, e.g., key data, may be possible; otherwise, damage can be trigged in case of capture of sensor nodes. Another possible technique in protecting against a compromised sensor node is to limit the amount of information obtained by the attacker after reading data from the captured sensor nodes. The former is more expensive than the latter, however. Therefore, the first option will be limited to applications that are critical enough to require the more expensive option. If sensor nodes cannot be tamper-resistant, the latter should be implemented to gain probability security. The cryptographic module (FIPS PUB 140-2) is an example of a tamper-resistant module that ensures sensitive data without storage damage.

10.6 USN middleware security

An enormous amount of data collected in the sensor network is securely stored, managed, and analysed by the USN middleware, which delivers data to the appropriate application through a secure channel. Since it communicates with sensor networks or applications over the IP network, USN middleware should consider the existing security threats on the IP network. To ensure secure communications against various threats such as spoofing, sniffing, message modifications, DDoS, etc., security dimensions such as encryption/decryption, authentication, authorization, and access control are considered and applied. Furthermore, the encryption/decryption function for data stored in the USN middleware is needed because USN middleware sometimes stores very important and valuable data that cause a big problem when revealed. The security technique to ensure the availability of USN middleware is also considered. The cryptographic module (FIPS PUB 140-2) can be considered as a security module for USN middleware.

10.7 IP network security

The IP network security technologies in Rec. ITU-T X.805 | ISO/IEC 18028-2 can directly be applied to secure message exchange through the IP network. Therefore, related details can be omitted.

10.8 Sensor node authentication

The mechanisms for sensor node authentication are mainly used to validate the legitimacy of the equipment when it enters the network to ensure that equipment in the network are legitimate and credible. Details of authentication mechanisms for the sensor node are described in Annex C.

10.9 Privacy protection in sensor networks

Data encryption and access control are typical approaches for ensuring privacy preservation in a sensor network. Another approach is to limit the network capability to collect the sensed data in such level of detail that the privacy of the individuals concerned could be compromised. For example, the sensor network might report the aggregate temperature over a large area instead of a small area. Sensed data should be stored securely in a sensor node by applying access control mechanisms to them. Another approach is to process the query in a distributed manner so that a single or a small number of sensor nodes cannot obtain the query results, which might result in compromising privacy.

11 Specific security functional requirements for USN

This clause specifies the various levels of security requirements that pertain individually or collectively to USN security.

11.1 Mandatory functional requirement

- The SN is required to support the data integrity and message authenticity of the sensed data.
- The key management scheme in the SN is required to support the key pre-distribution scheme described in clause 10.1.
- Key management is required to support both pair-wise key establishment and group-wise key establishment.
- The SN is required to authenticate broadcast messages from a base station to all the sensor nodes and vice versa.
- The SN is required to support secure routing protocols with message authentication, ID authentication, data freshness, and data integrity.
- The SN is required to support the capability to be resilient against various attacks.
- The base station in the SN is required to support the capability to mitigate the effects of DoS attacks from both wireless interface and wired interface.
- The USN is required to support USN middleware security as described in clause 10.6.
- The SN is required to support authentication/identification of the node by other nodes.

11.2 Recommended functional specifications

- The SN is recommended to support a secure end-to-end encrypted data aggregation scheme.
- The SN is recommended to support data freshness for sensed data.
- The SN is recommended to support the confidentiality of sensed data.
- Key management is recommended to support the pair-wise key establishment based on ID-based authentication. An example of this authentication is described in Annex C.
- The USN is recommended to support the mechanism for ensuring the privacy of the sensed data.
- The base station is recommended to support tamper resistance to avoid a single point of failure.

11.3 Optional functional specifications

- The sensor node or base station can optionally provide a secure hop-by-hop data aggregation scheme.
- The sensor node can optionally have a tamper-resistant module for protecting credentials, sensed data, or other confidential data.
- The sensor node can optionally have tamper detection, tamper evidence, or tamper response.
- The SN can optionally have the capability to mitigate DoS attacks against the sensor node.
- The SN can optionally have the capability to access multiple or randomly selected base stations to mitigate large-scale security threats due to single point of failure effects.

- The SN can optionally have the capability to mask asynchronous activity into synchronous messaging.
- The SN can optionally have the capability to provide multipath and/or randomized route selection to enhance resilience to attacks.
- The base station in the sensor network can optionally have the capability of intrusion detection.
- The SN can optionally have the capability to be configured to provide privacy protection.

Annex A

Key management in sensor networks

(This annex forms an integral part of this Recommendation | International Standard.)

A.1 Threat time

Once deployed, nodes establish a pair-wise key in a short time to ensure the key's security. Therefore, whether the phase of key setup is exposed to an adversary or not is crucial because sensitive information such as random number or identity information of the node is open during this phase. An adversary may get ready to attack in advance even before key setup. This adversary can analyse the communication between nodes or obtain physical access to the node during key setup. This adversary is regarded as strong and intensive. This means that an application requiring a high security level must design a key scheme assuming a prepared adversary.

On the other hand, to make an application more flexible and usable, a key management scheme with low security level can be taken. In this case, an attack is possible after a key is established. It is hard for an adversary who does not know the deployed time and who is unable to access the deployed place to try an attack during key setup. This is a very real case despite the loose attack. On the application where loose or no attacks during key setup are launched, designing a key scheme is reasonable to improve efficiency and scalability with only loose security.

A.2 Key management classes

Based on the two criteria above, i.e., threats and threat time, 4 key management classes are defined.

A.2.1 Class 1

This class assumes that an adversary can eavesdrop after key setup. There is no other threat such as node capture throughout the network life. Thus, this class considers the weakest adversary.

A.2.2 Class 2

This class assumes that an adversary can eavesdrop or capture and reprogram nodes, compromising the nodes, after key setup. In other words, during key setup, there are no threats in place, and eavesdropping hardly exists. After key setup, however, an adversary is capable of eavesdropping or obtaining secret information through node capture.

A.2.3 Class 3

This class assumes that an adversary can eavesdrop on the communications when nodes are deployed; after key establishment, the adversary is prepared for all attacks including node capture.

A.2.4 Class 4

An active adversary always waits for node deployment. This means that eavesdropping and node capture already take place in the phase wherein nodes are deployed. Considering the strongest adversaries, this class is a general assumption but incurs in high cost.

Generally, an adversary able to attack including executing node capture is considered to have enough ability to eavesdrop on transmitted data. Accordingly, other classes need not be considered: the case wherein compromising a node is always possible but eavesdropping is practical only after key setup, and the case wherein all attacks except eavesdropping are possible only after key setup. Moreover, the higher a class level is, the stronger an adversary. If a key scheme in a higher class is secure, it is also secure in a lower class. The key scheme classes are shown in Figure A.1.

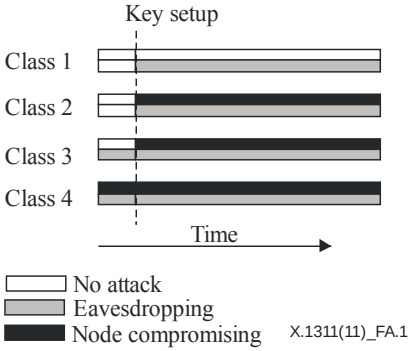

Figure A.1 – Key management classes

A.3 Key schemes

A.3.1 Key management mechanisms

A.3.1.1 Pair-wise key pre-distribution

A pair-wise key between a pair of nodes is directly stored and pre-distributed in each node prior to node deployment ("pair-wise key scheme"). Since each node in this scheme stores its pair-wise keys, it has perfect resilience against node capture; in other words, even if a node is captured, the keys of non-captured nodes are never compromised. Note, however, that scalability is limited because the network scale depends on the memory of the node where potential keys are stored.

A.3.1.2 Master key-based pre-distribution

A pair-wise key is derived from both a random number exchanged between each node and a single master key pre-distributed into each node ("master key scheme"). It results in great key connectivity and requires little memory. Nonetheless, resilience is very low since all pair-wise keys can be compromised when the master key is exposed to an adversary. Unlike the master key scheme, which does not erase a master key after key setup, a master key is erased completely after a pair-wise key is established. Although resilience is improved by erasing the master key of the deployed nodes, the risk of compromising a master key during node addition remains since the added nodes store a master key.

A.3.1.3 Base station participation

"SPINS" falls under this mechanism (see Perrig *et al*. in the Bibliography). In SPINS, each node is given its shared key with the base station, also known as a gateway. The base station directly transmits a pair-wise key encrypted with each node's shared key. In other words, the base station mediates in key setup. This scheme supports not only full connection but also perfect resilience. Still, it is not scalable because of the immense traffic volume resulting from intermediation.

A.3.1.4 Probabilistic key pre-distribution

For large networks, a probabilistic method is more efficient than a deterministic method. For instance, this mechanism is based on the concept that all the nodes in all networks are connected with a 0.9997 probability – almost fully connected – if the probability that each node can establish a pair-wise key with its neighbour nodes is 0.33. A key ring is stored in each node prior to deployment (key ring $\boldsymbol{k}$ is randomly selected from key pool $\boldsymbol{P}$, which is randomly selected from a huge key space). A common key in both key rings of a pair of nodes is used as their pair-wise key. It guarantees enough resilience – albeit an imperfect one – because the probability of breaking the communication link is $\boldsymbol{k/P}$. Moreover, it supports large-scale networks.

A.3.1.5 No key pre-distribution

This mechanism considers the reality of sensor networks. If an adversary does not know where and when nodes are deployed, launching an active attack at an early phase is difficult. Improving efficiency instead of opting for minimal node loss can be a good trade-off due to attacks during key setup. In this scheme, key setup is completed in a relatively short time through a few transmissions. As an advantage of this mechanism, the base station does not take part in key setup; hence, relatively less energy is consumed. Unlike the pre-distribution schemes above, it does not need to load potential keys into a node, thereby incurring low cost in terms of network organization. Note, however, that it is only strong when an adversary does not observe the communication during key setup; it cannot add nodes since a pair-wise key is established through the exchanged data during key setup.

A.3.1.6 Probabilistic pair-wise key pre-distribution

Suppose a sensor network has a maximum of n nodes. A simple solution to the key-predistribution problem is the *pair-wise* keys scheme where each node contains n–1 communication keys each being pair-wise privately shared with one other node in the network. The probabilistic pair-wise keys scheme is a modification of the pair-wise key scheme based on the observation that not all n–1 keys need to be stored in the node's key ring to have a connected random graph with high probability. Erdos and Renyi's formula allows us to calculate the smallest probability p of any two nodes connected such that the entire graph is connected with high probability c, where c is the desired confidence level (probability) that the sensor network is connected after completing the connection protocol (see Spencer). To achieve this probability p in a network with n nodes, each node needs to store only a random set of np pair-wise keys instead of exhaustively storing all n–1 keys. The use of pair-wise keys instead of purely random keys chosen from a given pool can give us node-to-node authentication properties if each node holding key k also stores the identity (ID) of the other node, which also holds k. Thus, if k is used to create a secure link with another node, both nodes are certain of the identity of each other since no other nodes can hold k.

Annex B

Authenticated broadcast in sensor networks: µTPC

(This annex does not form an integral part of this Recommendation | International Standard.)

B.1 Construction of µTPC

A major problem in scaling up µTESLA is how to distribute and authenticate the initial µTESLA parameters (µTP) including the key chain commitments, starting time, duration of each time interval, etc. The multilevel µTESLA uses high-level µTESLA instances to authenticate the parameters of low-level ones; it inherits the authentication delay introduced by µTESLA during the distribution of those parameters. As a consequence of such authentication delay, an attacker can launch DoS attacks to disrupt the distribution of the initial µTESLA parameters. Moreover, a multilevel µTESLA cannot handle a large number of senders. The tree-based µTESLA protocol uses the Merkle Tree mechanism to distribute µTP. Using the certificate from the Merkle tree, receiver nodes can authenticate µTP immediately; thus resisting DoS attacks. Note, however, that the cost of tree-based µTESLA is too high. The µTPCT-based broadcast authentication protocol constructs µTPC (µTP one-way chain) to distribute and authenticate µTP. It can resist DoS attacks but incurs low cost.

In sensor networks with multiple BNodes, the BNode (broadcast node) may have different characteristics considering the task to be performed. The life cycle, broadcasting frequency, and real-time requirement of the BNode are called feature parameters (FP). The BS will construct µTP based on the FP of the BNode. For example, for the BNode with short life cycle, high broadcasting frequency, and strict real-time requirement, the BS will construct a special µTP containing short key disclosure lag and fewer µTESLA instances with short time interval. The FP can be expanded if necessary.

µTPC consists of µTP and one-way chain. After FP is determined, the BS will first divide the lifetime of BNode into N time intervals with length of T_N such that the duration of T_N (e.g., 30 minutes) is suitable for running a µTESLA instance on the BNode and sensor nodes efficiently. Based on the broadcasting frequency and real-time requirement of the BNode, the BS will divide T_N into n time intervals with length of T_n. Based on N and n, the BS uses pseudo-random function F to generate the N µTESLA key chains that are linked together. First, the BS generates the last key $K_{N,n}$ of the N-th µTESLA key chain randomly. Afterward, using the hash function H (e.g., SHA-1), the BS generates the rest of the keys of the N-th µTESLA key chain according to $K_{N,i}$=H($K_{N,i+1}$). For the (i–1)-th µTESLA key chain, the BS generates the last key by performing a pseudo random function on the first key (key next to the commitment) of the i–th µTESLA key chain, and then generates the rest of the keys of the (i–1)-th µTESLA key chain by performing H on its last key. This way, the BS generates all µTESLA key chains up to the last one. Figure B.1 shows the construction of µTESLA key chains.

After all µTESLA key chains are generated, for the BNode j, the BS will assign different keys to different time intervals T_n. Accordingly, there will be N µTESLA instances. Whereas the initial parameter of the ith µTESLA instance is μTP_i={T_s ‖ $K_{i,0}$ ‖ T_i ‖ T_{int} ‖ d} where T_s denotes the current time, $K_{i,0}$ denotes the commitment, T_i refers to the starting time, T_{int} denotes the synchronization interval, and d refers to the disclosure lag of the key. After all µTPs are determined, the BS generates a value U_N randomly, and then computes each U value by U_{i-1}= H(U_i ‖ μTP_{i-1}) up to U_0 where " ‖ "denotes message concatenation. Finally, the BS constructs µTPC that includes N µTPs. Figure B.2 shows an example of the construction of µTPC.

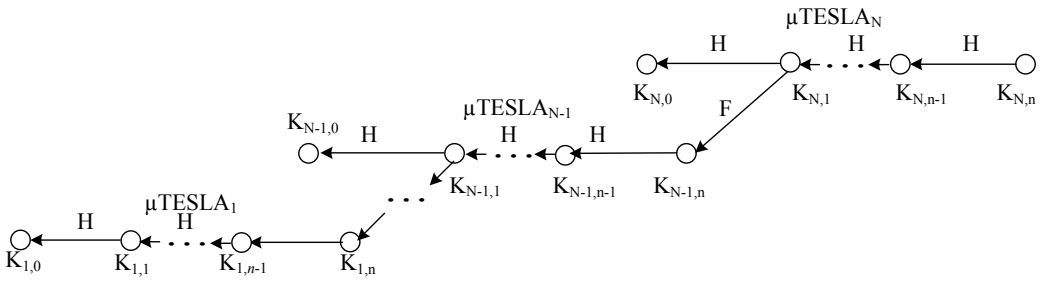

Figure B.1 – Construction of µTESLA key chains of µTPC wherein each $K_{i,n}$ is derived from $K_{i+1,1}$ using pseudo random function F

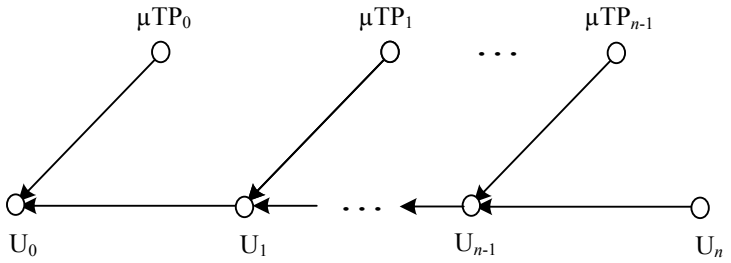

Figure B.2 – Construction of μTPC where $U_i = H(U_{i+1}||\mu TP_i)$

B.2 Construction of μTPCT

Suppose m BNodes exist in sensor networks. For convenience, one assumes that $m = 2^k$ where k is an integer. Prior to deployment, the BS pre-computes m μTPCs each of which is assigned a unique, integer-value ID between 1 and m. For purposes of presentation, the j-th U value of i-th μTPC is denoted as $U_{i,j}$, j-th μTP as $\mu TP_{i,j,}$ and i-th initial parameter (including $U_{i,0}$, ID_i) of μTPC as S_i. The BS then computes $K_i = H(S_i)$ for all $i \in \{1,...,m\}$. Afterward, it constructs a Merkle tree using $\{K_1,...,K_m\}$ as leaf nodes. Each non-leaf node is computed by applying H to the concatenation of its children nodes. Such a Merkle tree is called μTPCT (μTPC Merkle hash tree). Figure B.3 shows μTPCT with eight μTPCs, where $K_{12}=H(K_1 \parallel K_2)$, $K_{14}=H(K_{12} \parallel K_{34})$, $K_{18}=H(K_{14} \parallel K_{58})$, etc.

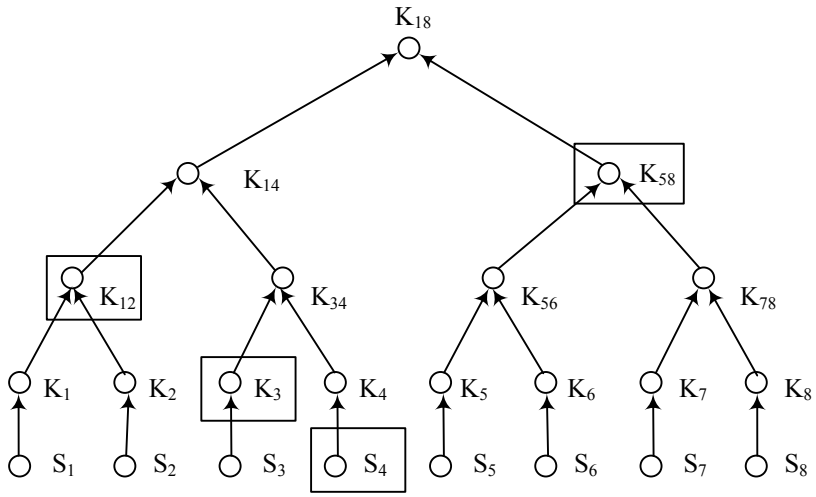

Figure B.3 – μTPCT tree with 8 leaf nodes; the nodes in the boxes constitute the certificate of S_4 called $PCert_4$

The BS also constructs a parameter certificate for each μTPC instance. The certificate for the i-th μTPC instance consists of S_i as well as the values corresponding to the siblings of the nodes on the path from the i-th leaf node to the root of μTPCT. For example, the parameter certificate for the 4th μTPC instance in Figure B.3 is $PCert_4=\{S_4,K_3,K_{12},K_{58}\}$. For each BNode, which will use a given μTPC instance, the BS distributes μTPC and the corresponding parameter certificate to it. The BS also pre-distributes the root of μTPCT to all potential receivers of broadcast messages.

Before the construction of μTPCT, if there are similar parts in all μTPs of some μTPCs, the same parts of μTP are put together with the initial parameter of μTPC as leaf nodes to construct μTPCT. For example, if disclosure lag d and synchronization intervals T_{int} in all μTPs of μTPC_i are the same, T_{int} and d are put together with the initial parameter of μTPC_i as leaf nodes to construct μTPCT. Accordingly, the same parts of μTP will be distributed together with the certificate of μTPC only once. In the process of μTP distribution, the BNode needs to distribute the discrepant part only. This way, substantive communication cost will be saved.

B.3 Authenticated broadcast

The performance of the authenticated broadcast protocol can be divided into the following five phases:

B.3.1 Protocol initialization

Prior to the deployment of sensor networks, the BS builds μTinst (denotes the μTESLA instance), μTPC, and μTPCT according to the quantity and FP of all BNodes. Afterward, it distributes root R of μTPCT to RNode (receiving node).

B.3.2 Request for μTPC

Before joining sensor networks, the BNode sends a request BREQ, which includes BNode's FP to the BS. Afterward, the BS searches for the compatible μTPC (e.g., μTPC_4 shown in Figure B.3) according to the FP of the BNode. Together with certificate $PCert_4$ and K_{gen} (denotes the generated key of the μTESLA key chain) of all μTinsts, the BS sends μTPC back to the BNode.

B.3.3 Authenticate BNode

Before broadcasting, the BNode publishes its certificate $PCert_4$ to all RNodes to prove its legitimacy. The RNode uses R and equation $H(H(H(H(S_4) \| K_3) \| K_{12}) \| K_{58}) = K_{18}$ to verify the validity of $PCert_4$. If it succeeds, the RNode saves μTPC's initial parameter $U_{4,0}$, ID_4 in the $PCert_4$ as well as the same parts of μTP in the μTPC.

B.3.4 Distribute μTP

After successful authentication, the BNode creates the first μTESLA key chain according to K_{gen} of the first μTinst using the hash function H. Afterward, the BNode broadcasts $U_{4,1}$ and $\mu TP_{4,0}$ to the RNodes. According to $U_{4,0}=H(U_{4,1} \| \mu TP_{4,0})$, the RNode verifies the legitimacy of $\mu TP_{4,0}$. If it succeeds, the RNode saves $U_{4,1}$ and $\mu TP_{4,0}$; otherwise, it is discarded. The BNode then broadcasts $U_{4,2}$ successively. At the same time, the BNode uses the 2nd K_{gen} to generate a second μTESLA key chain. After receiving $U_{4,2}$, the RNode saves it and deletes $U_{4,0}$. To assure the reliability of the distribution, the BNode will broadcast U repeatedly.

B.3.5 Authenticate broadcast message

Once the RNode gains $\mu TP_{4,0}$, based on $\mu Tinst_{4,0}$, the RNode can authenticate all broadcast messages from the BNode. Therefore, a broadcast authentication channel is created between the RNode and the BNode. When $2*T_{int}$ remains before the end of the life cycle of $\mu Tinst_{4,0}$ in the processing of broadcast authentication, the protocol will proceed to performing phase 4 subsequently. In other words, the BNode broadcasts $\mu TP_{4,1}$. The RNode verifies $\mu TP_{4,1}$ using $U_{4,1}=H(U_{4,2} \| \mu TP_{4,1})$. If it succeeds, the RNode saves $\mu TP_{4,1}$; otherwise, it is discarded. Note that the protocol makes sure the RNode can continue to authenticate the broadcast message of the BNode in the life cycle of $\mu Tinst_{4,1}$. The later procedure of the protocol will repeat phase 5 until the end of the life of BNode.

Annex C

Authentication mechanisms in sensor networks

(This annex does not form an integral part of this Recommendation | International Standard.)

C.1 XOR-based mechanism

The procedure of preshared-key based authentication is shown in Figure C.1. It includes a 3-way handshake between nodes A and B: an authentication request, an authentication response, and an authentication response confirmation.

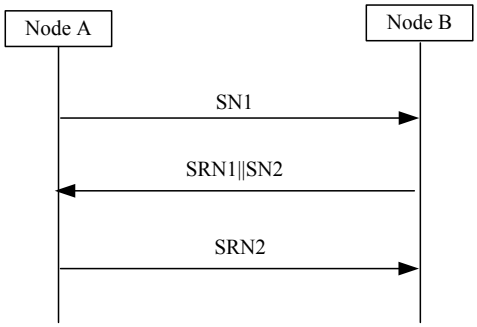

Figure C.1 – Preshared-key based authentication

In the text below, PSK and SK denote a preshared-key and a session key between nodes A and B, respectively.

C.1.1 Authentication request

A sends SN1 to B to start the authentication procedure. The format of the authentication request message (from A to B) is illustrated in Figure C.2.

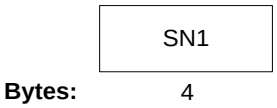

Figure C.2 – Format of authentication request message

where, SN1 = N1 $\oplus$ PSK and N1 is a 4-byte random number generated by A. When B receives the authentication request message from A, B performs the following procedures:

1) calculates N1 by SN1 $\oplus$ PSK;
2) calculates RN1 by reversing N1; and
3) constructs the authentication response message and sends it to A.

C.1.2 Authentication response

The format of the authentication response message (from B to A) is illustrated in Figure C.3.

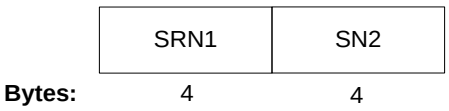

Figure C.3 – Format of authentication response message

where, SRN1 = RN1 $\oplus$ PSK, SN2 = N2 $\oplus$ PSK, and N2 is a 4-byte random number generated by MFAN-N(B?). When receiving the authentication response message from B, A performs the following procedures:

1) calculates RN1 by SRN1 $\oplus$ PSK, then calculates N1 by reversing RN1;
2) discards the message and stops the authentication process, if the value of N1 calculated by A does not match the one generated in the construction of the authentication request message;
3) calculates N2 by SN2 $\oplus$ PSK, then calculates RN2 by reversing N2;
4) calculates SK=N1 $\oplus$ N2 $\oplus$ PSK as the session key used with B; and
5) constructs the authentication response confirmation message and sends it to B.

C.1.3 Authentication response confirmation

The format of the authentication response confirmation message (from A to B) is illustrated in Figure C.4.

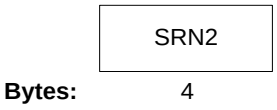

Figure C.4 – Format of authentication response confirmation message

where, SRN2 = RN2 $\oplus$ PSK. When receiving the authentication response confirmation message from A, B performs the following procedures:

1) calculates RN2 by SRN2 $\oplus$ PSK, then calculates N2 by reversing RN2;
2) discards the message and stops the authentication process, if the value of N2 calculated by B does not match the one generated in the construction of the authentication response message;
3) calculates SK=N1 $\oplus$ N2 $\oplus$ PSK as the session key used by A.

C.2 Hash-based mechanism

The procedure of Hash-based authentication is shown in Figure C.5. It includes a 4-way handshake between nodes A and B: authentication initiation, authentication request, authentication response, and authentication response confirmation.

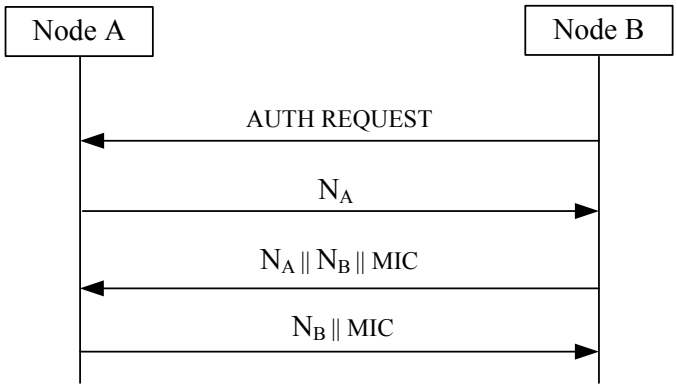

Figure C.5 – Hash-based authentication

In the text below, the PSK denotes a preshared-key between nodes A and B before authentication. SK denotes a session key between nodes A and B derived from PSK. "ID_{AB}" is the concatenation of node A's identifier and node B's identifier.

C.2.1 Authentication initiation

B sends an authentication initiation message to A to initiate the authentication procedure. The format of the authentication initiation message (from node B to node A) is illustrated in Figure C.6.

Figure C.6 – Format of authentication initiation message

When receiving the authentication initiation message from node B, A performs the following procedures:

1) generates a random number N_A; and
2) constructs the authentication request message and sends it to node B.

C.2.2 Authentication request

The format of the authentication request message (from node A to node B) is illustrated in Figure C.7.

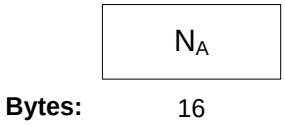

Figure C.7 – Format of authentication request message

where, N_A is a 16-byte random number generated by node A. When receiving the authentication request message from node A, B performs the following procedures:

1) generates a random number N_B;
2) calculates the session key;
3) calculates the MAC (message authentication code); and
4) constructs the authentication response message and sends it to A.

C.2.3 Authentication response

The format of the authentication response message (from node B to node A) is illustrated in Figure C.8.

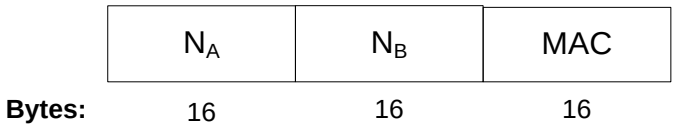

Figure C.8 – Format of authentication response message

where, N_A is a random number as in the authentication request message, N_B is a 16-byte random number generated by node B, MAC=HMAC-SHA256(SK,$N_B||N_A$), SK=KD-HMAC-SHA256(PSK,$ID_{AB}||N_B||N_A||$ "pair-wise key expansion for unicast and additional keys and nonce"). When receiving the authentication response message, node A performs the following procedures:

1) checks N_A first. If the value of N_A does not match the one generated in the construction of message 2, node A discards message 3 and stops the authentication process; otherwise, it implements step 2;
2) calculates SK=KD-HMAC-SHA256(PSK,$ID_{AB}||N_B||N_A||$ "pairwise key expansion for unicast and additional keys and nonce");
3) verifies message 3 MAC;
4) discards message 3, if the calculated MAC does not match the MAC in message 3; otherwise, go to step 5;
5) constructs the authentication response confirmation message and sends it to node B;

6) sets up the SK.

C.2.4 Authentication response confirmation

The format of the authentication response confirmation message (from node A to node B) is illustrated in Figure C.9.

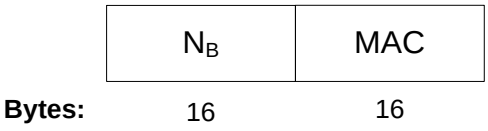

Figure C.9 – Format of authentication response confirmation message

where, N_B is a random number as in authentication response message and MAC=HMAC-SHA256 (SK, N_B). When receiving the authentication response confirmation message from node A, B performs the following procedures:

1) checks N_B first. If the value of N_B does not match the one generated in the construction of message 3, it discards message 4 and stops the authentication procedure; otherwise, it goes to step 2.
2) verifies message 4 MAC. If the calculated MIC does not match the MAC in message 4, node A discards message 4; otherwise, it sets up the SK.

C.3 Public key-based authentication

The procedure of public key-based authentication is shown in Figure C.10. There is a trusted authority to verify the identities of nodes A and B. Nodes A and B share a global variable *P* used to calculate the transient public-key.

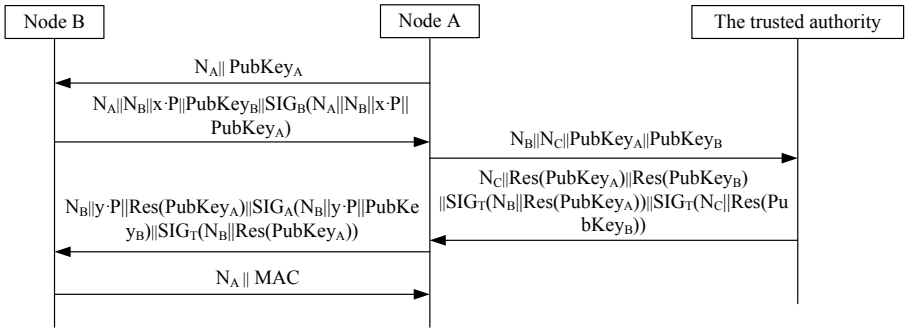

Figure C.10 – Public key-based authentication

In the text below, $PubKey_x$ denotes the public key of X, SIG_x refers to the signature of X, and $Res(PubKey_x)$ denotes the result of the verification of $PubKey_x$, respectively.

C.3.1 Authentication request

A sends an authentication request to node B to start the authentication procedure. The format of the authentication request message (from node A to node B) is illustrated in Figure C.11.

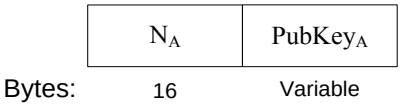

Figure C.11 – Format of authentication request message

where, N_A is a 16-byte random number generated by node A and $PubKey_A$ is the public key of node A. When receiving the authentication request message from A, B performs the following procedures:

1) generates a random number N_B;
2) generates the transient secret-key x and calculates the transient public-key $x\cdot P$;

3) uses the secret key to calculate the signature; and

4) constructs the authentication response message and sends it to node A.

C.3.2 Authentication response

The format of the authentication response message (from node B to node A) is illustrated in Figure C.12.

	N_A	N_B	$x \cdot P$	$PubKey_B$	$SIG_B(N_A \| N_B \| x \cdot P \| PubKey_A)$
Bytes:	16	16	variable	Variable	variable

Figure C.12 – Format of the authentication response message

where, N_A is a random number as in the authentication request message, N_B is a 16-byte random number generated by node B, $x \cdot P$ is the transient public-key for ECDH; x is the transient secret-key, $PubKey_B$ is the public key of node B, and SIG_B ($N_A \| N_B \| x \cdot P \| PubKey_A$) is the signature using the secret key of node B. When receiving the authentication response message from node B, node A performs the following procedures:

1) it generates a random number N_C; and

2) constructs the key verification request message and sends it to the trusted authority.

C.3.3 Key verification request

The format of the key verification request message (from node A to the trusted authority) is illustrated in Figure C.13.

	N_B	Nc	$PubKey_A$	$PubKey_B$
Bytes:	16	16	Variable	Variable

Figure C.13 – Format of key verification request message

where, N_B is a random number as in the authentication response message, N_C is a 16-byte random number generated by node A, $PubKey_A$ is the public key of node A, $PubKey_B$ is the public key of node B. When receiving the key verification request message from node A, the trusted authority performs the following procedures:

1) it inspects the validity of $PubKey_A$ and $PubKey_B$;

2) calculates the signature of the inspection result; and

3) constructs the key verification response message and sends it to node A.

C.3.4 Key verification response

The format of the key verification response message (from the trusted authority to node A) is illustrated in Figure C.14.

	N_C	$Res(PubKey_A)$	$Res(PubKey_B)$	$SIG_T(N_B \| Res(PubKey_A))$	$SIG_T(N_C \| Res(PubKey_B))$
Bytes:	16	1	1	Variable	Variable

Figure C.14 – Format of key verification response message

where, N_C is a random number as in the key verification request message, $Res(PubKey_A)$ is the result of the verification of $PubKey_A$, $Res(PubKey_B)$ is the result of the verification of $PubKey_B$, $SIG_T(N_B \| Res(PubKey_A))$ is the signature using the secret key of the trusted authority, $SIG_T(N_C \| Res(PubKey_B))$ is the signature using the secret key of the trusted authority, respectively. When receiving the key verification response message from the trusted authority, node A performs the following procedures:

1) it checks N_C first followed by $SIG_T(N_C \| Res(PubKey_B))$ by checking if the value of N_C matches the one sent in the key verification request message;

2) generates the transient secret-key y and calculates the transient public-key $y \cdot P$;

3) calculates BK_{AB}=KD-HMAC-SHA256((x·y·P)abscissa and $N_A||N_B||$ "base key expansion for key and additional nonce");

4) constructs the authentication result message and sends it to node B.

C.3.5 Authentication result

The format of the authentication result message (from node A to node B) is illustrated in Figure C.15.

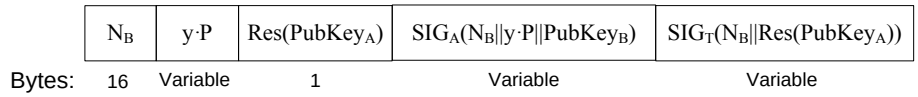

Figure C.15 – Format of authentication result message

where, N_B is a random number as in the authentication response message, $y \cdot P$ is the transient public-key for ECDH; y is the transient secret-key, Res($PubKey_A$) is the result of the verification of $PubKey_A$, $SIG_A(N_B||y \cdot P||PubKey_B)$ is the signature using the secret key of node A, and $SIG_T(N_B||Res(PubKey_A))$ is the signature using the secret key of the trusted authority. When receiving the authentication result message from node A, node B performs the following procedures:

1) it checks N_B;

2) verifies $SIG_T(N_B||Res(PubKey_A))$ and $SIG_A(N_B||y \cdot P||PubKey_B)$ by checking N_A and N_B, respectively;

3) calculates BK_{AB}=KD-HMAC-SHA256((x·y·P)abscissa and $N_A||N_B||$ "base key expansion for key and additional nonce");

4) calculates MAC=HMAC-SHA256 (BK_{AB},N_A);

5) constructs the authentication confirmation message and sends it to node A;

6) sets up BK_{AB}.

C.3.6 Authentication confirmation

The format of the authentication confirmation message (from node B to node A) is illustrated in Figure C.16.

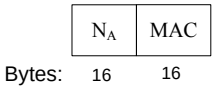

Figure C.16 – Format of authentication confirmation message

where, N_A ia a random number as in the authentication request message, MAC= HMAC-SHA256(BK_{AB}, N_A). When receiving the authentication confirmation message from node B, node A performs the following procedures:

1) it checks N_A first;

2) verifies MAC from B based on BK_{AB};

3) sets up BK_{AB}.

Annex D

Secure data aggregation in sensor networks

(This annex does not form an integral part of this Recommendation | International Standard.)

Figure D.1 illustrates the typical structure of sensor networks. The network can be divided into many clusters automatically according to some features such as energy, distance and so on. In a communication cycle, the network should elect an aggregation node and a supervisor inside each cluster. An elected supervisor plays a supervisory role in a certain period. A supervisor collects messages from nodes around it and sends these messages to the aggregation node, and then the aggregation nodes transfer these messages created by supervisor to the base station. When receiving these messages, a base station compares these messages to those aggregated by aggregation nodes and judges the true or false of the aggregated messages; therefore, securing data aggregation in sensor networks.

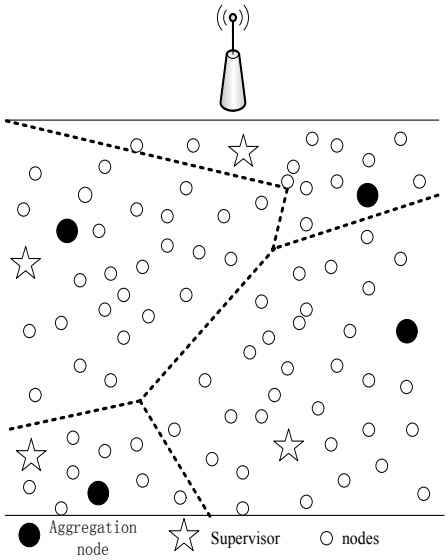

Figure D.1 – Illustration for secure data aggregation

The detailed description is shown as follows:

D.1 Elect aggregation node and supervisor

Firstly, the sensor node generates a random number, and then broadcasts the energy value of itself and this random number, each node receives that broadcasted message from neighbour nodes and compares those energy values and random numbers; the node with the largest energy value and the smallest random number will be elected as the aggregation node, and the node with the second largest energy value and the second smallest random number will be elected as the supervisor. When some nodes have the same energy value and random number, these nodes will generate a random number and negotiate repeatedly, this process will not end until the network finds an aggregation node and a supervisor in each cluster with the principle that the node whose random number is minimum becomes an aggregation node and the node whose random number is the second minimum becomes the supervisor node.

The cycle of the supervisor node T_s can be defined by the formula: $T_s \geq \sum_{n=1}^{n} T_{\mathrm{m}}$, $T_c = T_s$, where T_m is the communication cycle of ordinary nodes within the cluster, T_c is the cycle of aggregation node which is equal to the cycle of the supervisor node. T_c is greater or equal to the maximum among the cycle of aggregate node, cycle of supervisor, and the cycle of ordinary node in a cluster. The purpose is to ensure that the supervising process and aggregation process is completed in this period,. i.e., update the supervisor and aggregation node in a cluster to ensure the security of network communications.

D.2 Implementation of supervisor functions

Supervisor functions and data aggregation must be done at the same time. Supervisors collect the supervising messages from ordinary nodes which have also been sent to the aggregation node, and use the same aggregation algorithm to supervising messages, then send the supervising messages out.

D.3 Upload supervising message

Supervision messages are sent to the aggregation node by a supervisor, and then are forwarded to the base station. The security of the supervising messages is guaranteed by the pair-wise key between the supervisor and the base station.

D.4 Determine the trust of aggregation nodes

After receiving the supervising message from aggregation nodes, the base station compares them bit-by-bit. When the comparative result is inconsistent, the aggregator can be determined to be an untrusted node. If the base station does not receive the supervising information within the tolerable time which was configured by the system, the aggregator can also be determined to be an untrusted node.

D.5 Send revocation message

When the base station determines the aggregation node is untrusted, it broadcasts the revocation message to the whole network to revoke the aggregation node. This cluster should re-elect the aggregation node and the supervision node.

Bibliography

– IEEE 802.15.3:2003, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)*.

– IEEE 802.15.4:2006, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*.

– IETF RFC 4082 (2005), *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*.

– ISO/IEC 15408-1:2009, *Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model*.

– Akyldiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002), *A Survey on Sensor Networks*, IEEE Communications Magazine, pp.102-114, August.

– Chan, H., and Perrig, A. (2003), *Security and Privacy in Sensor Networks. Computer*, pp. 103-105, October.

– Karlrof, C., and Wagner, D. (2003), *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*, Elsevier, pp. 293-315.

– Perrig, A., Szewczyk, R., Tygar, J.D., Wen V., and Culler, D. E., (2002), *Spins: Security Protocols for Sensor Networks*, Kluwer Academic Publishers.

– Perrig, A., Canettiz, R., Tygary, J. D., and Song D. (2000), *Efficient authentication and signing of multicast streams over lossy channels*, in proceeding of IEEE symposium on Security and Privacy.

– Spencer, J. (2000), *The Strange Logic of Random Graphs; Algorithms and Combinatorics Vol. 22*, Springer-Verlag, ISBN 3-540-41654-4.

– Technical Document of ISO/IEC JTC 1, *Study Group on Sensor Networks (SGSN)*.

ICS 35.110

Price based on 34 pages