POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in

Ingegneria Elettronica e delle Comunicazioni – XXV ciclo

Tesi di Dottorato

Key Management in Wireless Sensor Networks, IP-Based Sensor Networks, Content Centric Networks

Sarmad Ullah Khan

mat. 169506

Tutore

Prof. Luciano Lavagno

Co-Tutore

Eng. Claudio Pastrone

Coordinatore del corso di dottorato

Prof. Ivo Montrosset

March 14, 2013

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, my utmost gratitude to Prof. Luciano Lavagno whose sincerity, supervision, constant support and encouragement I will never forget. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works have contributed to the success of this research. Not forgotten, my appreciation to my co-supervisor, Eng. Claudio Pastrone, who has very deep and good knowledge in security related issues and his main research work is focused on the security in wireless sensor networks and IP based sensor networks, for his guidance, support and knowledge. I also thank for his help in providing me useful informations about the simulation tools and solving technical issues.

I would also like to pay my special appreciations to Dr. Thibault Cholez, a research associate in University of Luxembourg and Dr. Radu Stat, a research scientist in University of Luxembourg for hosting me as a visiting researcher during my PhD intern-ship.

I would also like to thank Higher Education Commission (HEC) of Pakistan for believing in me and granting me an MS-Leading to PhD scholarship under the faculty development program (UESTP-Italy).

Sincere thanks to all my friends and colleagues for their kindness and moral support during my study. Thanks for the friendship and memories.

Last but not least, my deepest gratitude goes to my beloved parents, my sisters and my brother for their endless love, prayers and encouragement. To those who indirectly contributed in this research, your kindness means a lot to me. Thank you very much.

Summary

Cryptographic keys and their management in network communication is considered the main building block of security over which other security primitives are based. These cryptographic keys ensure the privacy, authentication, integrity and non-repudiation of messages. However, the use of these cryptographic keys and their management in dealing with the resource constrained devices (i.e. Sensor nodes) is a challenging task.

A number of key management schemes have been introduced by researchers all over the world for such resource constrained networks. For example, light weight PKI and elliptic curve cryptography schemes are computationally expensive for these resource constrained devices. So far the symmetric key approach is considered best for these constrained networks and different variants of it been developed for these networks (i.e. probabilistic key distribution approach). The probabilistic key distribution approach consumes less memory than the standard symmetric key approach but it suffers from the connectivity issues (i.e. the connectivity depends on the common shared keys between the nodes).

Most of those schemes were proposed by considering static sensor networks (e.g. Industrial process monitoring, Environmental monitoring, movement detection in military applications, forests etc.). However, the use of these existing key management schemes for mobile wireless sensor networks applications introduces more challenges in terms of network connectivity, energy consumption, memory cost, communication overhead and protection of key materials against some well known attacks. Keeping these challenges in mind, previous research has proposed some key management schemes considering the mobility scenarios in ad hoc networks and wireless sensor networks (e.g. vehicular networks, health monitoring systems).However these

schemes consume more resource because of a much higher communication packet exchange during the handover phase for the authentication of joining and leaving nodes than the static networks where there is no extra communication for the handover and authentication.

The motivation of this research work is to investigate and propose new algorithms not only to improve the efficiency of these existing authentication and key management schemes in terms of connectivity, memory and security by considering the mobility scenario in wireless sensor networks, but also to develop new algorithms that suit these constrained networks than the existing schemes.

First, we choose the existing key pool approach for authentication and key management and improve its network connectivity and resilience against some well known attacks (e.g. node capturing attacks) while reduce the memory cost by storing those key pools in each sensor node. In the proposed solution, we have divided the main key pool into two virtual mutually exclusive key pools. This division and constructing a key from two chosen keys, one from each key pool, helps to reduce the memory cost of each node by assigning fewer keys for the same level of network connectivity as the existing key pool frameworks.

Although, the proposed key pool approach increases the network resilience against node compromission attacks because of the smaller number of keys assigned to each node, however it does not completely nullify the effect of the attacks. Hence we proposed an online mutual authentication and key establishment and management scheme for sensor networks that provides almost 100% network connectivity and also nullifies the effect of node compromission attacks. In the proposed online key generation approach, the secret key is dependent on both communicating parties. Once the two communicating parties authenticate each other, they would successfully establish a secret communication key, otherwise they stop communication and inform the network manager about the intruder detection and activity.

The last part of the thesis considers the integration of two different technologies (i.e. wireless sensor networks and IP networks). This is a very interesting and demanding research area because of its numerous applications, such as smart energy, smart city etc.. However the security requirements of these two kind of networks (resource constrained and resourceful) make key management a challenging task. Hence we use an online key generation approach using elliptic curve cryptography

which gives the same security level as the standard PKI approach used in IP networks with smaller key length and is suited for the sensor network packet size limitations. It also uses a less computationally expensive approach than PKI and hence makes ECC suitable to be adopted in wireless sensor networks. In the key management scheme for IP based sensor networks, we generate the public private key pair based on ECC for each individual sensor node. However the public key is not only dependent on the node's parameter but also the parameters of the network to which it belongs. This increases the security of the proposed solution and avoids intruders pretending to be authentic members of the network(s) by spreading their own public keys.

In the last part of the thesis we consider Content Centric Networking (CCN) which is a new routing architecture for the internet of the future. Building on the observation that today's communications are more oriented towards content retrieval (web, P2P, etc.) than point-to-point communications (VoIP, IM, etc.), CCN proposes a radical revision of the Internet architecture switching from named hosts (TCP/IP protocols) to named data to best match its current usage. In a nutshell, content is addressable, routable, self-sufficient and authenticated, while locations no longer matter. Data is seen and identified directly by a routable name instead of a location (the address of the server). Consequently, data is directly requested at the network level not from its holder, hence there is no need for the DNS). To improve content diffusion, CCN relies on data distribution and duplication, because storage is cheaper than bandwidth: every content - particularly popular one - can be replicated and stored on any CCN node, even untrustworthy. People looking for particular content can securely retrieve it in a P2P-way from the best locations available.

So far, there has been little investigation of the security of CCNs and there is no specific key management scheme for that. We propose an authentication and key establishment scheme for CCNs in which the contents are authenticated by the content generating node, using pre-distributed shares of encryption keys. The content requesting node can get those shares from any node in the network, even from malicious and intruder ones, in accordance with a key concept of CCNs. In our work we also provide means to protect the distributed shares from modification by these malicious/intruder nodes. The proposed scheme is again an online key generation approach but including a relation between the content and its encryption key. This

dependency prevents the attackers from modifying the packet or the key shares.

Since the integration of sensor networks with IP networks is not only a very attractive research area but also the CCN architecture may be adopted for the internet of the future, the investigation of a single suitable key management scheme for sensor networks within the CCN architecture will be an interesting future research topic.

Contents

List of Tables

List of Figures

List of Acronyms

MEMS Micro-electro-mechanical systems

WSN Wireless Sensor Network

PDA Personal Digital Assistant

RFID Radio Frequency Identification

DoS Denial of Service Attacks

RTS request-to-send

JTAG Joint Testing Action Group

SPA Simple power analysis

DPA Differential power analysis

EMA Electromagnetic attacks

SEMA Simple Electromagnetic attacks

DEMA Differential Electromagnetic attacks

KMS Key management systems

GQ Generalized Quadrangles

FPP Finite Projective Planes

ECDH Elliptic curve DiffieHellman

ECMQV Elliptic Curve Menezes-Qu-Vanstone

PKC Public Key Cryptography

ECC Elliptic Curve Cryptography

PARC Palo Alto Research Center

CCN Centric Centric Networking

PIT Pending Interest Table

FIB Forward Information Base

TTP Trusted Third Party

CA Certification Authority

SKC Symmetric Key Cryptography

MAC Message Authentication Code

AES Advanced Encryption Standard

IBC identity-based cryptography

MASY MAnagement of Secret keYs protocol

HSN Heterogeneous Sensor Network

USAS Unpredictable Software-based Attestation Solution

NPKPS Novel Pairwise Key Pre-distribution Scheme

ID-PKG Identity Based Private Key Generator

PKG Private Key Generator

BS Base Station

CH Cluster Head

FN Fix Node

MN Mobile Node

SSC Secret communication key generation code

m Key pool containing SSC

K_{plc} Network public key

K_{prt} Network private key

SK secret key

MNPN Mobile Node Prime Number

CNDK Compromised Node Detection Key

FNPNS Fix Nodes Prime Number Sum

K_{Auth} Authentication Key

NAC Network Authentication Code

AVISPA Automated Validation of Internet Security Protocols and Applications

OFMC On-the-Fly Model-Checker

CL-AtSe Constraint-Logic-based Attack Searcher

IoT Internet of Things

E2E End-to-End

SN Sensor Node

VoIP Voice over IP

KHN Key Holding Node

NN Normal Node

NDRN NoDe Random Number

NTPS NeTwork Public Share

NDPS NoDe Public Share

Introduction

Chapter 1

Introduction

1.1 Wireless Sensor Networks

Recent advances in Micro-electro-mechanical systems (MEMS) and wireless communication technologies made it possible to build small devices that can run autonomously and be deployed in a large-scale, low power, inexpensive manner that is acceptable to many commercial and government users. These devices can be used to form a new class of distributed networking, namely Wireless Sensor Networks (WSNs). Sensor networks configurations range from very flat, with few command nodes denoted as base stations, sinks or cluster controllers, to hierarchical nets consisting of multiple networks layered according to operational or technical requirements. The existence of sensor hardware and the robustness and reliability of such sensor networks tries to build a bridge between the abstract world and the physical world. These sensors are devices that can measure a physical quantity (e.g. temperature, humidity) and convert it into a digital signal. Using these sensors, computer systems ranging from the simplest washing machine to the Large Hadron Collider (a particle accelerator located at the European Organization for Nuclear Research (CERN)) can acquire and process information coming from the physical world. This ability to "feel" the world is usually embedded in the design of a computer system, e.g. sensors in a washing machine are integrated within the system from the initial design. However, it would be particularly interesting to make this ability available as an off-the-shelf component. As a result, any computer system, regardless of its

design, could be able to perceive the physical world. Such is the task of Wireless Sensor Networks.

The structure of a wireless sensor network can be seen in Figure 1.1. A wireless sensor network is composed by two types of devices: sensor nodes, and base stations. The sensor nodes, also known as motes or simply nodes, are small and constrained devices that have the ability to "feel", "think", "talk", and "subsist". They can "feel", because they can sense the physical features of their surrounding (e.g. temperature, humidity, radiation, vibration) using hardware sensors. They can "think", because although they are highly constrained in both computational power and memory, they are capable of processing information on their own. They can "talk", because they are equipped with wireless transceivers, and can collaborate towards a common goal. Finally, they can "subsist" because they are in most cases powered by batteries, and can survive in their deployment field for more than a year if their internal operations are optimized.

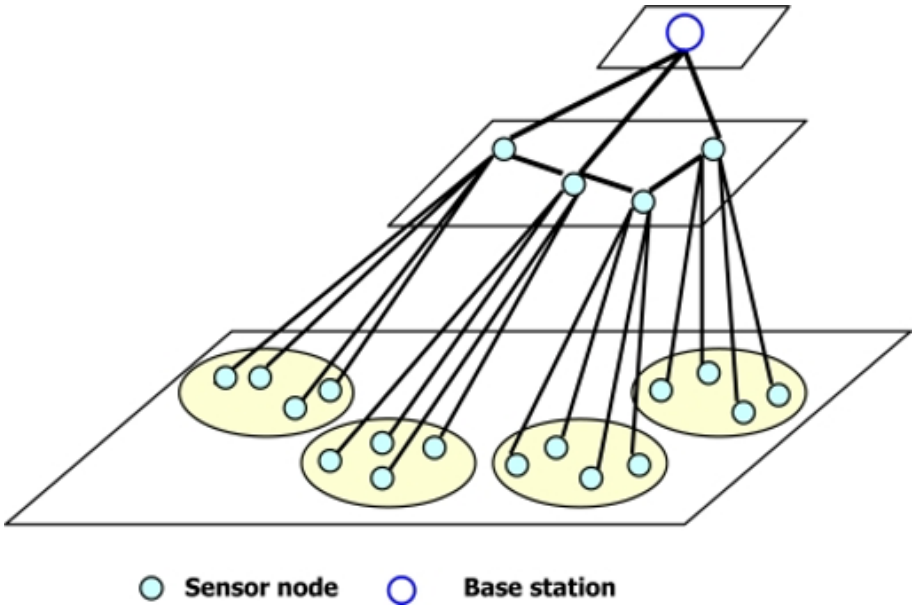

Figure 1.1. Structure of Wireless Sensor Networks

Regarding the base station, it is a more powerful device that usually behaves as an interface between the services provided by the sensor nodes (the "data acquisition network") and the users of the network (the "data dissemination network"). Normally, the base station collects all the information coming from the sensor nodes

and stores it for later use.Also, it can issue control orders to the sensor nodes in order to change their behavior. While it would seem that wireless sensor networks are highly dependent of the existence of this base station, the architecture of the network is not centralized. Sensor nodes can operate in a decentralized fashion, managing themselves without accessing to the base station.

Since the early 1990s, distributed sensor networking has been an area of active research. The trend is to move from a centralized, super reliable single-node platform to a dense and distributed multitude of cheap, lightweight and potentially individually unreliable components that, as a group, are capable of far more complex tasks than any single super node. The intuition is to have individual sensor nodes share information with each other and collaborate to improve detection probabilities while reducing the likelihood of false alarms. Research prototype sensors (UCB motes, Tmote Sky, Telos, EyesIFX, ScatterWeb MSB-430) are designed and manufactured, energy efficient MAC, topology control protocols and routing schemes are implemented and evaluated, various enabling technologies such as time synchronization), localization and tracking are being studied and invented. All these provide sensor networks tremendous potential for information collection and processing in a variety of application domains.

The first generation of sensor nodes facilitated the genesis of wireless sensor networks as they exist today: small resource-constrained embedded devices that communicate via low-power, low-bandwidth radio, capable of performing simple sensing tasks. A first set of scenarios for these networks included stationary nodes sensing ephemeral features of the environment, like temperature, noise, air pollution, etc. By continuously monitoring these surrounding attributes, they solved relatively small-scale specialized problems such as forest monitoring, preventative maintenance, etc.

Early sensor networks, as shown in Figure 1.2, functioned primarily into two important application domains: monitoring and tracking. WSNs can be configured to monitor a variety of target types. The networks themselves are mode-agnostic, enabling multiple types of sensors to be employed, depending on operational requirements; cameras as vision sensors, microphones as audio sensors, ultrasonic, infrared, light, temperature, pressure/force, vibration, radio activity, seismic sensors, and so on. Target tracking can also be performed effectively with sensors deployed as a

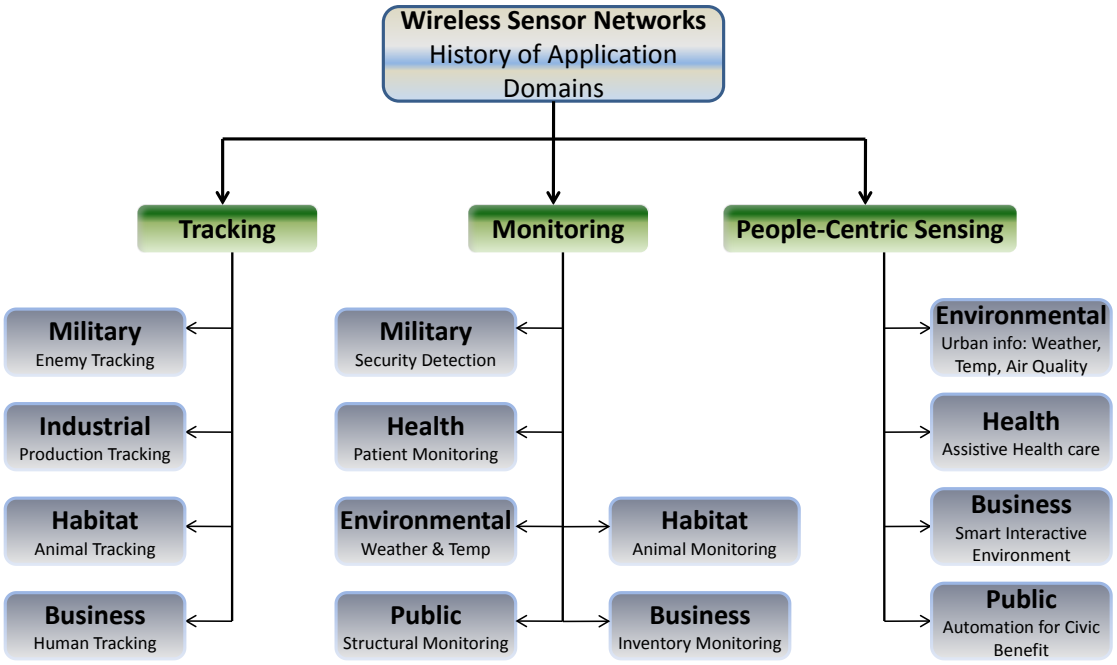

Figure 1.2. History of research in sensor networks application domain

three-dimensional field and covering a large geographic area. Therefore, some of the most common applications are military, medical, environmental and habitat monitoring, industrial and infrastructure protection, disaster detection and recovery, green growth and agriculture, intelligent buildings, law enforcement, transportation and space discovery. For instance, in enterprise scale manufacturing and retail companies, sensor networks can be combined with RFID (Radio Frequency ID) tags to monitor inventory and support in-process parts tracking. These networks can automatically report problems at various stages such as in-plant manufacturing, packaging, and equipment maintenance.

the latest trend in sensor networking tries to change the traditional view of sensor-based environments where people are passive data consumers that simply interact with physically embedded static sensor webs, with one where people carry

mobile sensing elements involving the collection, storage, processing and fusion of large volumes of data related to everyday human activities. This evolution is driven by the miniaturization and introduction of sensors into popular electronic devices like mobile phones and PDAs. With wireless sensor platforms in the hands of thousands, we can expect sensor networks to address urban-scale problems as shown in Figure 1.2. Such systems referred to as *urban sensing* or *people-centric sensing*. These systems aim at daily life applications, employing the mobile devices people already carry for sensing information directly or indirectly related to human activity, as well as aspects of the environment around them.

Unlike other more capable devices such as PDAs, there is no human user directly controlling the sensor node: motes are usually accessed through other motes or through the base station. In fact, sensor nodes can set up their services and function properly in situations where there is no central control available. Due to this autonomy, sensor nodes need to self-configure and maintain themselves during the lifetime of the network. Precisely, a wireless sensor network can function for long periods of time, ranging from several days to one or two years. Regarding the network deployment, sensor nodes are usually deployed near the physical source of the events, and the exact deployment location of these sensor nodes is usually not known in advance. Finally, sensor nodes are usually not mobile, although there might be scenarios where some sensor nodes or even the base station need to move (e.g. tracking a target as explained above).

1.2 Security Aspects of Wireless Sensor Networks

In any environment, either physical or logical, there exists the need of maintaining someone or something safe, away from harm. This is the role of security. On any computer-related environment, security can be considered as a nonfunctional requirement that maintains the overall system usable and reliable, protecting the information and information systems. In fact, in wireless sensor networks, security is of paramount importance: the network must be adequately protected against malicious threats that can affect its functionality. Due to the role of sensor networks as a "sensory system", any disturbance in a sensor network may have consequences in the real world. However, sensor networks are especially vulnerable against external

and internal attacks due to their peculiar characteristics.

- The devices of the network (i.e. sensor nodes) are constrained in terms of computational capabilities, memory, communication bandwidth, and battery power. As a result, it is challenging to implement and use the cryptographic algorithms and protocols required for the creation of security services.

- In most cases, it is easy to physically access sensor nodes: they must be located near the physical source of the events. Since nodes are not tamper-resistant due to cost constraints, any human user or machine can reprogram them or simply destroy them.

- Any internal or external device can access the information exchange because the communication channel is public. Besides, attacking the availability of the wireless channel is not a complex task.

- It is a difficult task to monitor and control the actual state of the elements of the network due to the inherent distributed nature of sensor networks. Any failure in any of its elements may remain unnoticed, or the actual cause of the malfunction may not be known. Besides, a sensor network can be attacked at any point.

1.3 Security Threats in Wireless Sensor Networks

Due to their previously shown inherent vulnerabilities, sensor networks have to face multiple passive and active attacks that may easily hinder its functionality and nullify the benefits of using its services. Passive attacks are able to retrieve data from the network, but do not influence over its behaviour. On the other hand, active attacks directly hinder the provisioning of services. The different threats that target sensor networks will be detailed in the following paragraphs, and they can be categorized as follows:

- Common Attacks. As the wireless medium is used as the main transmission channel in WSN, it is easily subject to various types of attacks, either passive (eavesdropping) or active (data injection).

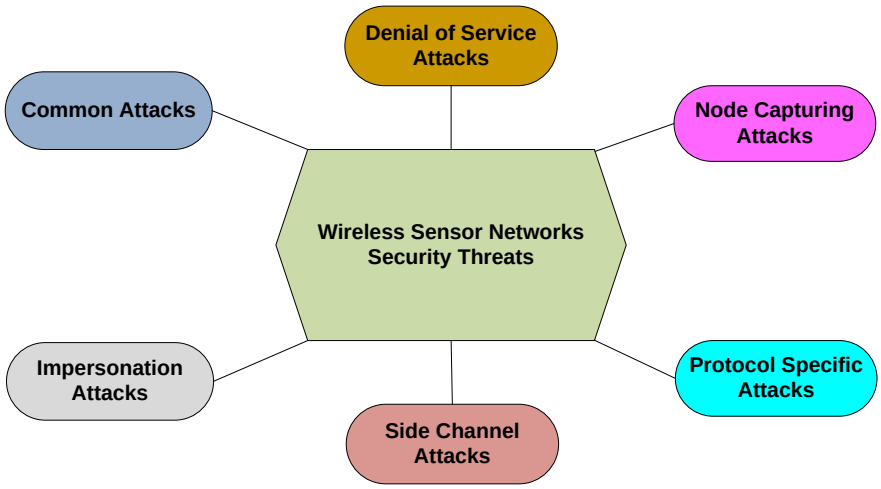

Figure 1.3. Security Threats of Wireless Sensor Networks

- Denial of Service Attacks (DoS). These attacks prevent any part of WSN from functioning correctly or in a timely manner. Such attacks can target the communication channel (e.g. jamming) or the life of the nodes themselves (e.g. power exhaustion).

- Node Compromise. An embedded device is considered being compromised when an attacker, through various means, gains control or access to the node itself after it is being deployed. These attacks are usually utilized as a foundation for more powerful, damaging attacks.

- Side-channel Attacks. An adversary can monitor certain physical properties of the nodes, such as electromagnetic emission, whenever it performs a cryptographic operation. If the recorded physical values are influenced by the secret key, then the adversary can extract information about that key.

- Impersonation Attacks. A malicious sensor node can create multiple fake identities (sybil attack), and also can create duplicates with the same identity (replication attack). These types of attacks are also the initial step which enables the attacker to conduct a wide range of malicious attacks.

- Protocol-specific Attacks. Some essential protocols used in WSN, such as routing, aggregation, and time synchronization, are targeted by specific attacks

that aim to influence the internal services of the network.

1.3.1 Common Attacks

By using the so-called common attacks class, a malicious adversary uses a device that does not belong to the sensor network in order to access the contents of the communication channel. The simplest instance of common attack is eavesdropping. It can be defined as the interception of information or data by an unintended party. Due to the broadcast nature of the communication channel, any adversary (using a mote or a more powerful device such as a PDA) can sniff out packets at a particular frequency, obtaining confidential information about the state of the network and the physical parameters sensed by the nodes. As the eavesdropping attack has an inherent passive nature, it does not directly influence over the behaviour of the network.

However, the acquired information from passive attacks can be used to perform active attacks. The effects of active attacks are far more destructive: adversaries can create fake events or hide problematic situations, and can even introduce bogus control information. One of these active attacks is message modification, where an adversary intercepts and modifies the packets content meant for the base station or intermediate nodes. Another active attack is message replay. In this attack, the adversary reuses valid transaction messages or packets content with malicious intent. The adversary performs a replay attack by first intercepting a valid critical transaction data packet and then re-transmitting at a later time. Lastly, attackers can use message injection to fabricate and send out false data into the network, maybe masquerading as one of the nodes.

1.3.2 Denial of Service Attacks

One special class of active attack, known as Denial of Service (DoS), deserves a category of its own. In this kind of attack, the objective of the malicious adversary is simple: to avoid the provisioning of services. As these services are published by the sensor nodes through a wireless channel, the most basic DoS attacks can target the nodes themselves (power exhaustion attack) or the communication channel (jamming attack). In the power exhaustion attack, an attacker imposes a particularly

complex task to a sensor node in order to deplete its battery life. Sensor nodes usually have a limited supply of energy, thus this attack is particularly dangerous. Besides, as sensor nodes have limited computational power, this attack can also slow down their reaction time. An example of an expensive operation is the verification of a cryptographic signature using public key cryptography. An attacker can take advantage of the complexity of this operation by repeatedly sending fake signatures to force the receiver to check their correctness. Power exhaustion attacks are not limited to only CPU attacks: an attacker can target the MAC protocol of the WSN, effectively preventing nodes from entering their duty/sleep cycle an wasting their batteries [1].

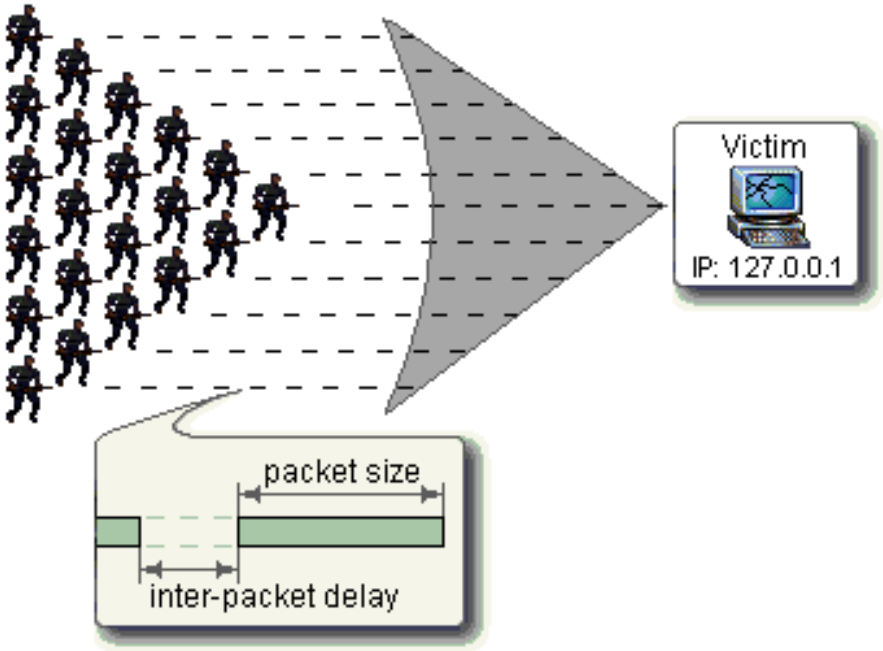

Figure 1.4. DoS Attacks Example

Jamming is the primary physical layer DoS attack against WSN. In a jamming attack, the attacker constantly emits radio frequency signals that do not follow an underlying MAC protocol, thus any member of the network in the affected area will not be able to send or receive any packet. The energy requirements of this attack are very high, as the attacker must flood the communication channel with noise. There

are some optimizations to the basic jamming attack, such as the random jamming, where the attacker alternates between sleep and jamming to save energy, or the reactive jamming, where the jam signal is transmitted only when the attacker senses traffic [2]. Finally, another clever optimization that reduces the energy consumption of the attacker is to target MAC protocols on the link layer [1], [5], e.g. by jamming only request-to-send (RTS) packets. As a side note, it should be mentioned that DoS attacks can be performed by using some of the attacks explained in other categories [6], although those attacks are usually more complex and can be used to disrupt other functional elements of the network (e.g. the authenticity of the physical/control data).

1.3.3 Node Compromise Attacks

Most of the previously shown attacks can be performed by outsiders: attackers that do not have access to the network elements and services. However, if an attacker have access to the network as one of its elements, i.e. as an insider, it is possible to perform attacks that are more subtle and devastating. The first step to become an insider is to compromise a node, usually by performing node compromise attacks. A sensor node can be considered compromised when an attacker, through various means, can either read or modify its internal memory. The ultimate goal of this attack is, in most cases, to obtain the secret keys stored within a trusted node in order to infiltrate a mole inside the network. Attacks that can lead to a node compromise are invasive or non-invasive. In an invasive attack, the attacker physically breaks into the hardware by modifying its hardware structure (e.g. using focused ion beam, or drilling a hole in the storage media). On the other hand, in non-invasive attacks the data is taken from the hardware device without any form of structural modification done to the device itself. Invasive attacks usually fall under the category of side channel attacks, as these attacks obtain confidential data directly from the chips of the nodes. batteries [1].

Regarding non-invasive attacks, they usually take advantage of the hardware interfaces of the nodes. One example is the JTAG interface [8]. This interface enables accessing and controlling of the signal levels on the processor chip, and is also used for debugging purposes. Through the use of an AVR ICE JTAG programming tool,

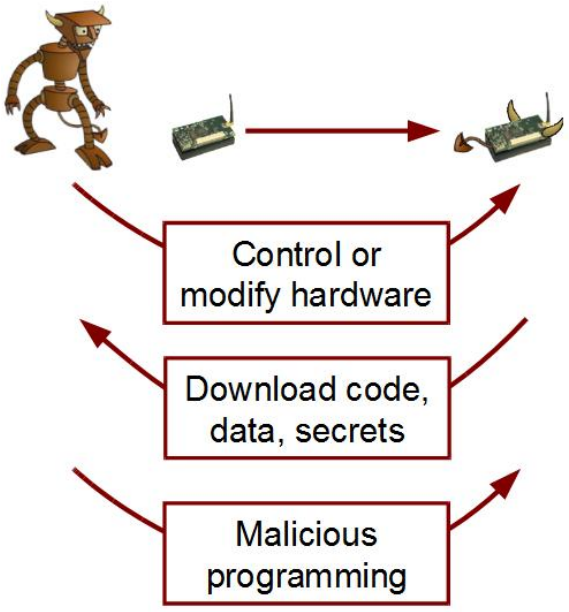

Figure 1.5. Node Compromise Attacks Example

an attacker can dump all the information from the program flash, the EEPROM and also the SRAM. As a result, the attacker can replicate the functionality of the node to facilitate the integration of the malicious node. While most of these non-invasive attacks simply aim to obtain information from the node, there exist more advanced attacks that are capable of injecting code inside a working sensor node. For example, it is possible to exploit the serial bootstrap loader (BSL) of certain models of the Texas Instruments MSP430 low-power microcontrollers with the aim of extracting or replacing the firmware [9]. Even more, it is possible to inject malicious code remotely in AVR-based nodes by exploiting buffer overflow vulnerabilities [10].

1.3.4 Side Channel Attacks

In order to compromise a node, it is also possible to attack its hardware through side-channel attacks. The main objective of side channel attacks is to obtain confidential data stored within the node. Most attackers focus on obtaining security credentials such as secret keys, since these credentials will provide the attacker with a powerful tool capable of crafting more powerful attacks. Side channel attacks can be classified

in the following categories: passive vs. active and non-invasive vs. semi-invasive vs. invasive. Passive attacks extract information from the device merely by observing physical properties of the devices, while active attacks involve the manipulation (tampering) of the device itself. In contrast, non-invasive attacks do not manipulate the device substantially, while semi-invasive attacks depackage the device but do not make direct electrical contact with the chip's surface, and invasive attacks have practically no limits to the measures which can be taken to extract the information of the device (e.g. probing station, focused ion beam). Note that not all semi-invasive or invasive attacks are active attacks: passive semi-invasive attacks may try to just read sensitive data from memory components, and passive invasive attacks can use a probe station to sense valuable data signals.

Figure 1.6. Side Channel Attacks Example

Specific examples of side-channel attacks are power analysis attacks, electromagnetic attacks, and timing attacks [7]. In power analysis attacks, the adversary studies the power consumption of the devices, focusing mainly on the energy used by cryptographic operations. For performing these attacks, it is possible to either use single power traces to look for distinguishing features (Simple power analysis, SPA) or use larger numbers of power traces alongside with powerful statistical methods (Differential power analysis, DPA). Electromagnetic attacks (or EM attacks) are similar to power analysis attacks, since they also analyse power traces with simple (SEMA) and differential (DEMA) methods. However, they derive the power traces from electromagnetic emanations, collected by EM probes. Beyond simple and differential analysis, EM attacks can employ more advanced techniques, such as adding spatial information to the measurement data, or analysing the frequency domain rather than the time domain. Finally, as the execution time of cryptographic algorithms often shows slight differences dependent on the input of the algorithm,

timing attacks exploits the variance in execution time for different branches in the cryptosystem.

1.3.5 Impersonation Attacks

Once an attacker becomes an insider, it is easier to perform impersonation attacks. For this particular class of attack, the goal of the adversary is to make the victim believe that it is communicating with an impersonated entity. As a result, a malicious node will interact with other nodes as one trusted member, but at the same time it can manipulate the internal behaviour of the network whenever the adversary needs it. Impersonation attacks can either replicate and insert duplicate nodes back

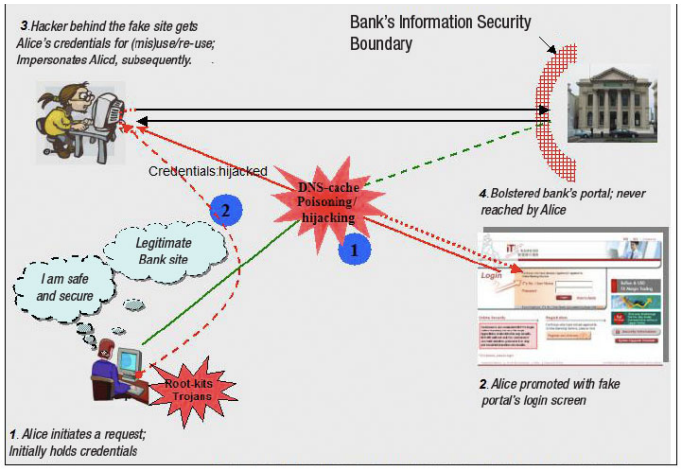

Figure 1.7. Impersonation Attacks Example

into selected regions of the network (node replication attack or clone attack) or use multiple identities to deceive other sensor nodes (sybil attack). In node replication attacks, the attacker only needs to subvert one node in order to create an army of clones following his orders. These clones can not only manipulate the internal operations of the network, but also exert a strong influence over those processes that require of a majority vote. As for sybil attacks, a sybil node can either fabricate new identities or steal them from legitimate nodes [11]. Sybil nodes can be able to execute powerful attacks, disrupting several of the functions that may be conducted on a WSN including data aggregation, voting, routing and fair resource allocation.

1.3.6 Protocol Specific Attacks

Beyond impersonation, an insider can perform protocol-specific attacks, attacking those "core" protocols needed by the network such as routing protocols, aggregation protocols, and time synchronization protocols. Attacks against routing protocols in a WSN fall into one of the following categories [12]: corruption of the internal control information such as the routing tables (Spoofed Routing Information), selective forwarding of the packets that traverse a malicious node depending on some criteria (Selective Forwarding), creation of a "wormhole" that captures the information at one location and replays them in another location either unchanged (Wormhole attack) or tampered (Sinkhole attack), creation of false control packets during the deployment of the network (HELLO Flood Attack), and creation of false acknowledge information (Acknowledgement Spoofing).

Data aggregation protocols combine information coming from the same area in order to reduce the overall communication overhead. As these protocols need to use routing protocols in order to fuse information and forward it to the base station, every attack that target the routing infrastructure can also be used to hinder the aggregation process. Most of these attacks try to discard data, either selectively or indiscriminately. Though losing data is a problematic situation for the network, this is not the primary type of attack against aggregation: most attacks focus on falsifying information. If an aggregator node is being controlled by an adversary, it can easily ignore the data received from its neighbours and create false reports. Moreover, trusted aggregators can still receive false data from faulty nodes or from nodes being controlled by an adversary.

Regarding time synchronization, it is needed because as the time obtained from clocks of different nodes may differ due to different starting times (offset), inaccurate quartz crystals (skew), or ambient influence (drift), it is necessary to synchronize these clocks in order to maintain a global notion of time [13]. Most time synchronization protocols rely on two neighbouring nodes adjusting their local clocks by means of sender-receiver (mutual synchronization) and receiver-receiver (beacon signals) protocols. In these scenarios, the main objective of an attacker is to deceive other nodes into thinking that an incorrect time is accurate. Besides internal attacks,

where the attacker can out-rightly lie about the value of its internal clock, an attacker can use the following external attacks: manipulation of the contents of the negotiation messages through message forging and replay, and delaying the messages exchanged in the negotiation process by means of a pulse-delay attack.

1.4 Security Requirement

As we have previously seen, sensor networks are vulnerable to external and internal attacks. The effects of those attacks in the network are not trivial, since they can render the services of the network useless. It is clear that there is the need of using security mechanisms either to prevent the attacks from influencing over the functionality of the network or to minimize the adverse effects of such attacks. By using the security mechanisms, it can be possible to enforce in sensor networks the following security properties:

1.4.1 Confidentiality

This property tries to fulfil the following principle: A given message must not be understood by anyone other than the desired recipients. While confidentiality is an important security property, it may not be mandatory in certain scenarios where the data is public by itself (i.e. the temperature of a street) and no other information can be derived from it. However, there are particular situations and scenarios where the physical data obtained by the network can be deemed as sensitive, and should not be read by external entities. Data can be considered sensitive due to its inherent nature (e.g. patient data such as temperature), the nature of the context (e.g. a private household, a military setting), or the nature of the sensed entities (e.g. a protected animal like a panda). Besides, certain control data exchanged by the nodes, such as security credentials and secret keys, must be hidden from unauthorized entities.

1.4.2 Integrity

This property states that the data produced and consumed by the sensor network must not be maliciously altered. Unlike confidentiality, integrity is, in most cases, a

mandatory property. The wireless channel can be accessed by anyone, thus any peer (outsiders and insiders) can manipulate the contents of the messages that traverse the network. Even more, data loss or damage may occur due to the harsh communication environment, and in the worst case the network will accept corrupted data. As the main objective of a sensor network is to provide services to its users, the sensor network will fail in its purpose if the reliability of those services can not assured due to inconsistencies in the information.

1.4.3 Authentication

Informally, data authentication allows a receiver to verify that the data is really sent by the claimed sender. This security property is quite important in sensor networks. In fact, without authentication the barrier between external and internal members of the network would not exist, as any outsider could claim that it is a registered member of the network. Moreover, even existing network members could easily pose as their neighbours. This situation would encourage many problematic situations, such as adversaries forging the whole packet stream by injecting additional packets, and nodes accepting false administrative tasks (e.g. network reprogramming).

1.4.4 Authorization

This property states that only authorized entities (sensor nodes and base station) can be able to perform certain operations in the network (e.g. information providing, controlling the system). Since a sensor network can be considered as one single entity, where all nodes perform the same tasks and acknowledge the role of the base station as manager and supervisor, it could be supposed that any authenticated device is inherently authorized to perform its tasks. Nevertheless, there might be situations (e.g. when nodes actuate over physical systems) where some members of the network need to have a proper authorization in order to perform certain tasks. In these situations authorization must be taken into account.

1.4.5 Self-Organization

One specific property related to the autonomous nature of sensor networks is self-organization: sensor nodes must be independent and flexible enough to autonomously react against problematic situations, organizing and healing themselves. These problematic situations can be caused either by external or internal attackers trying to influence over the behaviour of the elements of the system or by extraordinary circumstances in the environment or in the network itself. This is an essential property to the functioning of a sensor network and optimal resource use during its lifetime. It is desirable that all possible problems that may occur can be detected and prevented without any margin of error. However, as the previous statement may not be realistic, nodes should be able to at least adapt their activities to assure the continuity of the services.

1.4.6 Non-repudiation

While non-repudiation is not considered in the existing literature as an important security property for most sensor networks, it may be necessary to at least consider its applicability in certain contexts where sensor nodes monitor critical components, as acknowledging the reception and processing of serious alarms is of key importance. This property is described as follows: a node cannot deny sending a message it has previously sent. Note that non-repudiation can also consider repudiation of receipt, where the recipient tries to deny the reception of the message. For achieving non-repudiation, it is necessary to produce certain 'evidence' in case a dispute arises. Using the evidence, it is possible to prove that a device of the network performed a task.

1.4.7 Privacy

These security properties are very important in those scenarios where the location and identities of the base station and the nodes that generated information should be hidden or protected. For example, any network that monitors endangered species should provide no clues on their physical location. Also, in a battlefield, it would be important to not be able to distinguish whether a certain signals belongs to a

soldier or a vehicle. In contrast, there are situations where this property should not be enforced: in an earthquake rescue situation locating the source nodes (if the nodes are worn by, for example, dogs) is an absolute must. Note that this property can transcend beyond the technological dimension and affect its social environment, since sensor networks could be used as a surveillance tool to collect data about the behaviour of human beings.

1.5 Key Management and Secure Channels

All devices that want to open a secure channel with other nodes must share some security credentials, i.e. secret keys. Key management systems (KMS) aim to solve the problem of creating, distributing, and maintaining those secret keys. The design of a KMS for sensor networks is not a trivial task, though: it is not advisable to rely on centralized entities due to the distributed and self-configurable nature of the network. Also, the existing constraints of sensor nodes (memory, computational capabilities) may discourage the use of resource-intensive algorithms for most scenarios. Finally, there are other factors, such as the potential size of the network, the connectivity of its nodes, the energy spent in the key setup processes, etc, that influence over the design of a KMS as well.

Due to their importance, the Key Management Systems for Wireless Sensor Networks have received increasing attention on the scientific literature, spanning many different types of protocols [25]. In fact, since one of the most important link-layer standards in sensor networks, IEEE 802.15.4, does not specify how secret keys should be exchanged, it is essential to utilize one of these protocols. These protocols can be classified into four major frameworks. Although the major purpose of all these frameworks is to bootstrap the secret keys that are needed by the link layer, their underlying mechanisms and design goals are different.

1.5.1 Key Pool Framework

This is one of the first and most important KMS frameworks. The basic scheme behind this framework is quite simple [51]: the network designer creates a "key pool", a large set of precalculated secret keys, and before the network deployment

every node in the network is assigned unique "key chain", i.e. a small subset of keys from the "key pool" (Key pre-distribution phase). After the deployment, the nodes can interchange the identification numbers of the keys from their "key chains", trying to find a common shared secret key (shared-key discovery phase). If two nodes do not share any key, they will try to find a "key path" between them in order to negotiate a pairwise key (path-key establishment phase). The major design goal of the protocols that belong to this framework is to assure a limited secure connectivity between nodes, regardless of the size of the network.

1.5.2 Mathematical Framework

Certain KMS protocols use mathematical concepts (Linear Algebra, Combinatorics, and Algebraic Geometry) for calculating the pairwise keys of the nodes. The foundation of the Linear Algebra schemes is the Blom's scheme [52]. In this scheme, every node i can calculate the pairwise key it shares with another node j by solving A(i)G(j), whereas G is a public Vandermonde matrix and A is calculated using a symmetric random secret matrix D. On the field of Combinatorics, the Generalized Quadrangle and Symmetric Design models [53] are the most important. Using Generalized Quadrangles GQ(s, t) or Finite Projective Planes FPP(q), a network designer can construct a key chain of size s + 1 or q + 1, respectively. Finally, on the field of Algebraic Geometry, the basic primitive is the Bivariate Polynomial [54]. By using a bivariate polynomial f, every node A in the network is able to obtain a pairwise key with another node y by solving f(A, y). All these protocols allow the creation of pairwise keys between nodes without major communication overhead. On the other hand, these designs are often difficult to apply, and they are not very scalable.

1.5.3 Negotiation Framework

All protocols that generate their keys through mutual agreement, negotiating keys with their closer neighbours just after the deployment of the network, can be considered part of this framework. They are usually applied under the assumption that there is little or no treat against the integrity of the network in its first moments of life [55]. Nevertheless, it is possible to use other mechanisms and protocols (such

as the Guy Fawkes protocol [56]) in order to assure the authenticity of the peers in any step of the network deployment. Other protocols that can be included inside this framework are those protocols that organize the network into dynamic or static clusters [57].

1.5.4 Public Key Framework

Most of the previous frameworks rely only on Symmetric Key Cryptography. However, Public Key Cryptography can also be used to securely bootstrap the pairwise key of two nodes over a public communication channel. In these protocols, two nodes just need to interchange their public keys and some information (through protocols such as ECDH and ECMQV) to effectively create their pairwise secret keys. While constrained sensor nodes can be able to use PKC through Elliptic Curve Cryptography, the amount of memory required for the implementation and the time and energy needed to complete the negotiation is, in most cases, substantially higher than other KMS frameworks. However, PKC-based KMS usually have better properties than the systems of other frameworks.

1.5.5 Discussion and Open Issues

As for the actual state of the art, every one of these frameworks contains many different protocols, and these protocols can implement specific optimizations (e.g. use of deployment knowledge, optimize the message exchange, tweak the behaviour of the protocols, combine protocols from different frameworks) in order to improve their features and be more useful for certain contexts. In fact, there exists certain methods that are able to select the most adequate KMS for a certain context by using the application requirements as an input [27]. By using these tools, it is possible to conclude that the actual state of the art for KMS in sensor networks is good enough for protecting certain applications. However, there are some issues that remain to be solved, such as the creation of protocols with better resilience (resistance against stolen credentials) and extensibility (capability of adding new nodes) properties. One interesting detail is that, for most applications, simple KMS protocols such the basic polynomial-based and Blom schemes provide most of the properties needed by the applications. The reason is simple: most real-world applications have a

number of nodes ranging from 50 to 1000. And for this size, simpler protocols are good enough. Another interesting point is the use of PKC. For every single situation PKC seems to be the ideal protocol, and in fact PKC-based protocols such as EDH and ECMQV provide good properties like excellent resilience and extensibility. Nevertheless, there might be situations where simpler KMS protocols can provide the properties needed by the applications.

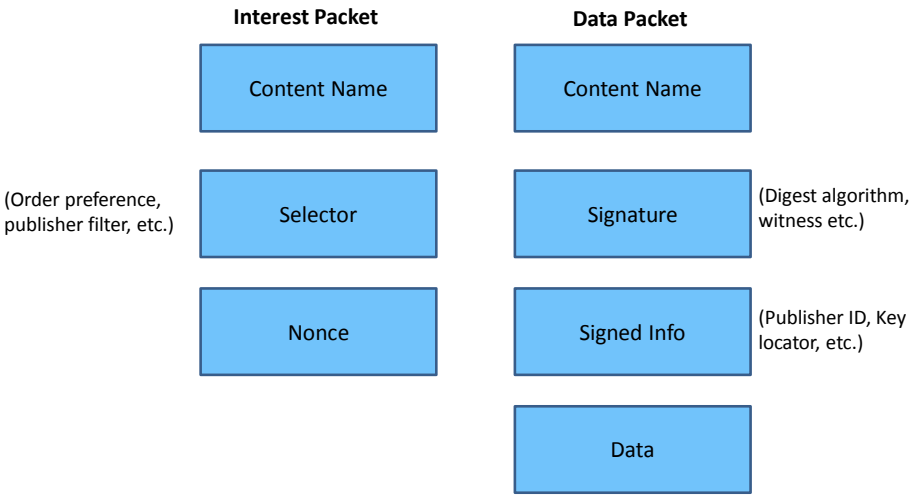

Figure 1.8. CCN packet structure, hierarchical naming and forwarding engine

1.6 Content Centric Networking

Developed at PARC by Van Jacobson and his team [3], CCN is also known as Information Centric Networking or Named Data Networking [4]. Building on the observation that today's communications are more oriented towards content retrieval (web, P2P, etc.) than point-to-point communications (VoIP, IM, etc.), CCN proposes a radical revision of the Internet architecture switching from named hosts (TCP/IP protocols) to named data to best match its current usage. In a nutshell, content is addressable, routable, self-sufficient and authenticated, while locations no longer matter. Data is seen and identified directly by a routable name instead of a location (the address of the server). Consequently, data is directly requested at the network level (not its holder, no need of DNS). To improve content diffusion, CCN relies on

close data storage because storage is proven cheaper than bandwidth: every content - particularly popular one - can be replicated and stored on any CCN node, even untrustworthy. People looking for particular content can securely retrieve it in a P2P-way from the best locations available.

1.6.1 CCN Paradigm

CCN has two main types of packets, Interest and Data as shown in figure 1.8. A user who wants to access a given content sends out an Interest packet, specifying the name of the content (as defined by CCN nomenclature ContentName) to all its available faces. Faces can be anything which can serve as medium for transmitting and receiving data. A node which receives this packet and that can "satisfy" the Interest then sends out the corresponding Data packet onto the face from which it received the Interest. By definition, CCN nodes are stateful and only forward Data on the back path if an Interest was emitted beforehand.

Data can only "satisfy" a specific Interest if the ContentName of Interest packet is a prefix of the Data packet. CCN names are defined in [3] as "opaque, binary objects composed of an (explicitly specified) number of components". This structure allows a fast and efficient prefix-based lookup similar to the IP lookup currently used. It also allows names to be context dependent i.e. /ThisRoom/Printer references a printer in the current room. This new routing paradigm is based on a plain-text hierarchical naming instead of regular host's IP addresses so that names are directly intuitive and do not need an indirection mechanism between names and contents (no need of DNS, DHT). An example of this hierarchical naming structure is presented in 1.9 for a content named "ccnx:/uni.lu/videos/intro.avi"

CCN nodes are composed of three main table structures, presented in 1.10, which handle the forwarding of packets. At the arrival of an Interest packet on any given face, the engine performs a longest-match lookup on its structures and action is taken depending on the lookup result. The first structure to be searched is the Content Store. It can be seen as a buffer memory of past Data packets on the current router. IP routers also have such a buffer but it is purged once the packet is forwarded. The Content Store however preserves the Data packet based on LRU scheme and enables therefore a fast retrieval of currently popular demands. If there

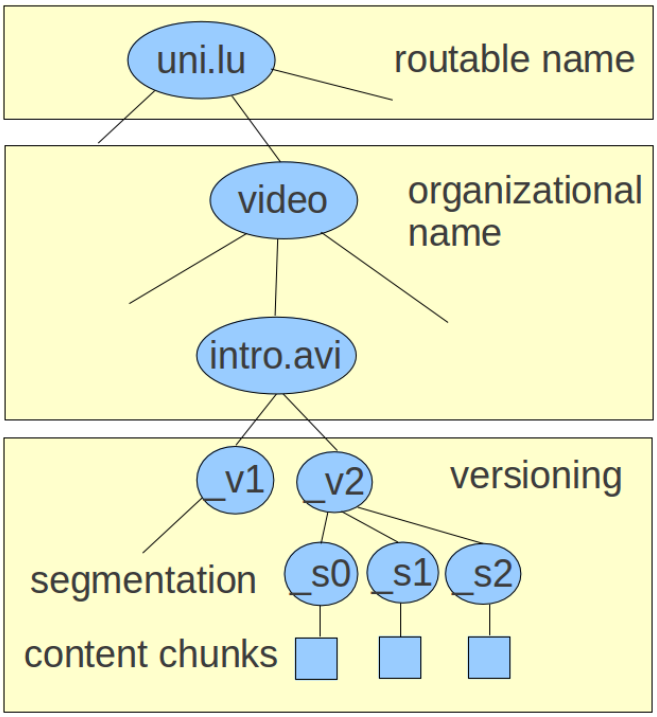

Figure 1.9. CCN packet structure, hierarchical naming and forwarding engine

is a match, the router forwards its local copy of the content to the face on which it received the Interest and updates its Content Store accordingly. If there is no match in the Content Store, the lookup is launched on the next structure which is the PIT. The PIT stands for Pending Interest Table and keeps record of Interests waiting to be resolved upstream by other content source(s). If a received Interest matches an entry in the PIT, the engine compares the faces recorded for that entry. If there is already one existing, no update is made. Otherwise, the face form which the Interest was emitted is simply added to the list of already waiting faces.

If no match-up is found in the PIT then the engine searches in its last structure: the FIB. The Forward Information Base keeps record of potential content source(s) and works similarly to its IP counterpart except that it stores a list of possible providers for a given name rather than a single one only. If a match is found, the engine then creates a PIT entry for the given Interest and it is forwarded to all faces specified in the FIB entry. If no match could be made, it means that the current

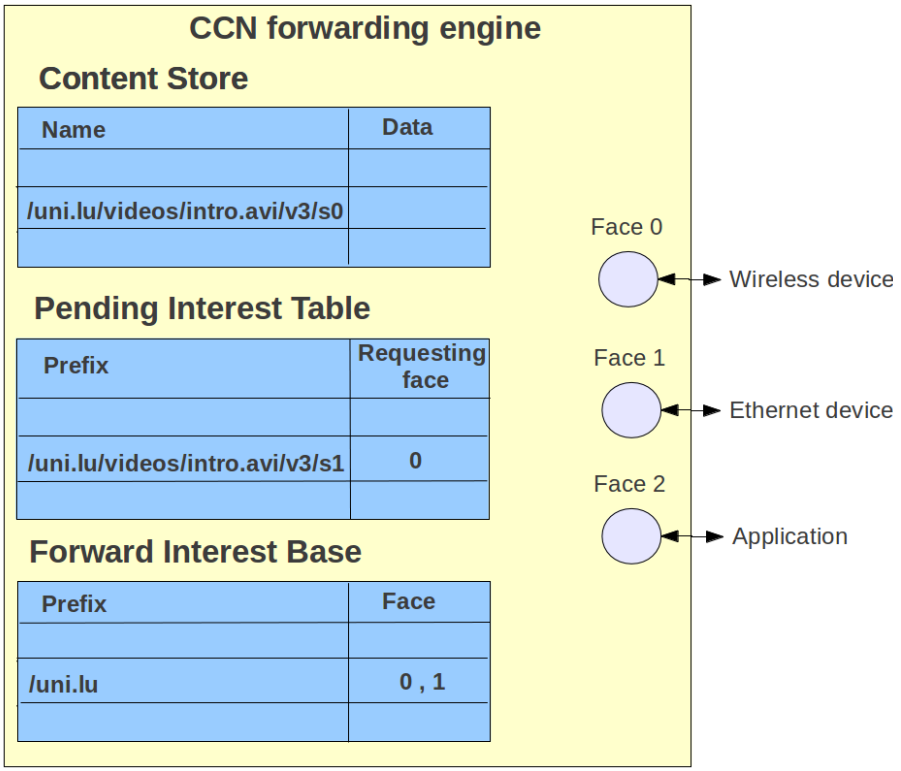

Figure 1.10. CCN packet structure, hierarchical naming and forwarding engine

router has no information on the demanded content and discards the Interest.

CCN has also built-in strategy and security layers. The strategy layer is used to define policies to select which face is the best for given contents. In fact, due to its design, FIB entries contain multiple faces. CCN can send periodically Interests to all outgoing faces without fearing of loops and thereby testing which of the faces responds the fastest. This one will be used as preferred until another round of this experiment yields to a different result. Criteria for experimentation interval can be a threshold of packets sent, a time out, change of the SSID, etc.

The security layer ensures that the content received by a previously announced Interest is authentic. As in CCN only the content matters but not the route it takes, the only thing which needs to be checked for authenticity, consistency and integrity is the content itself which reversely means that end-to-end encryption is not needed any more. Key management is another issue often discussed. We will review some

of that work in the next section.

1.6.2 CCN Security Schemes

Content Centric Networking improves the security of Internet communications in many ways. First of all, CCN messages cannot be sent toward a node without any prior Interest request from that node (Data cannot be pushed, only pulled) which makes the classical denial of service scheme inefficient as the attacker would first need his target to generate a lot of Interests to enable the DoS attack.

The paradigm shift of CCN makes every node capable to answer a Data request. To ensure the security of communications all data is authenticated, contrary to the connections it traverses. The security scheme should provide better results. For example, a secure connection to a mail server does not avoid SPAM mails to be received while with CCN, SPAM mails will fail the authentication as untrustworthy Data and will be discarded. So, CCN strongly relies on cryptography to authenticate the contents so that users can clearly know who emitted the content and can discard those from untrustworthy sources to avoid malware. Also, encryption is used to ensure privacy.

CCN provides native security and privacy by encryption with lower overhead than current protocols [3]. To securely authenticate content, CCN has to bind the content name, the content itself and the content provider. To do so, the following information is embedded in each CCN data packet: Signature(Name;Content; SignInfo). SignInfo includes: cryptographic digest or fingerprint of publishers key, key or key location. When a node receives the key, it verifies it with the certification authority, as in the PKI approach. The successful authentication of the message ensures the integrity and security, but not the privacy of the content.

In [14], Smetters et al. propose the following description: each new content creates a mapping triple: where M is the Mapping, N the Name, C the Content. Every piece of data must include a way to retrieve the key of publisher and mapping evidence. In this case, the authors include in the packet the mapping from the name of the content to the provider-assigned name, instead of the actual content name.

Content can be authenticated by every node using public key signatures and different signature algorithms are available to find a trade-off between the needed security and performances. The key-stone of CCN security is the trust in the publisher. To ease key management in CCN, Jacobson et al. propose to see an organization as a content name and public key as 3 a Data. They propose to use the SDSI/SPKI where keys are mapped to identities via namespaces (CCN names) so that there is no single source of trust like the current certification authorities. This scheme opens evidence based security where data provenance (traceability).

If CCN improves security in some points, it also raises the possibility of new kinds of attacks. For example, unlike terminal hosts which are less exposed to attacks, CCN routers are more vulnerable than IP routers because of their stateful nature and the complexity of the management of their inner tables, which are essential for performance and quality of service. All these new algorithms and their implementations need to be assessed from a security point of view before being trustable.

1.7 Thesis outline and previously published papers

This thesis is composed of seven chapters. The chapter 1 describes the introduction of Wireless Sensor Networks (WSNs), IP-Based Wireless Sensor Networks (6LoWPAN), Content Centric Networking (CCN), their security threats and their security requirements. The background about the key management schemes in these different technologies is discussed in chapter 2. The proposed key management algorithm for the wireless sensor networks based on the key pool approach is described in chapter 3 while the online key generation approach is described in chapter 4. Chapter 5 describes the online key generation approach for IP-Based wireless sensor networks while the key management approach for the content centric networking is discussed in chapter 6. Chapter 7 concludes the thesis.

List of Papers

Journal

1. Khan, S.U., Pastrone, C.,Lavagno, L. and Spirito, M.A. (2012) "A mutual authentication and key establishment scheme for heterogeneous sensor networks supporting nodes mobility", Int. J. Internet Technology and Secured Transactions, Vol. 4, Nos. 2/3, pp.139161.

2. Sarmad Ullah Khan, Claudio Pastrone, Luciano Lavagno, Maurizio A. Spirito, "An Authentication and Key Establishment Scheme for the IP-Based Wireless Sensor Networks", Procedia Computer Science, Volume 10, 2012, Pages 1039-1045, ISSN 1877-0509,The 7th International Symposium on Intelligent Systems Techniques for Ad hoc and Wireless Sensor Networks (IST-AWSN), Niagara Canada, 27-29 Sept. 2012. DOI: 10.1016/j.procs.2012.06.144.

3. Rafiullah Khan, Sarmad Ullah Khan , Rifaqat Zaheer, M.I. Babar, An Efficient Network Monitoring and Management System, in International Journal of Information and Electronics Engineering (IJIEE), vol. 3, no. 1, pp. 122-126, ISSN: 2010-3719 January 2013, DOI: 10.7763/IJIEE.2013.V3.280.

Conference Proceedings

1. Sarmad ullah khan, Thibault Cholez, Thomas Engel, Luciano Lavagno, A Key Management Scheme for Content Centric Networks, IFIP/IEEE Integrated Network Management Symposium (IM 2013), Ghent, Belgium, 27-31 May 2013.

2. Khan, Rafiullah; Khan, Sarmad Ullah; Zaheer, Rifaqat; Khan, Shahid; , "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," Frontiers of Information Technology (FIT), 2012 10th International Conference on , vol., no., pp.257-260, 17-19 Dec. 2012. doi: 10.1109/FIT.2012.53

3. Khan, Sarmad Ullah; Pastrone, Claudio; Lavagno, Luciano; Spirito, Maurizio A.; , "An energy and memory-efficient key management scheme for mobile heterogeneous sensor networks," Risk and Security of Internet and Systems (CRiSIS), 2011 6th International Conference on , vol., no., pp.1-8, 26-28 Sept. 2011. doi: 10.1109/CRiSIS.2011.6061832.

4. Khan, Rafiullah; Khan, Sarmad Ullah; Zaheer, Rifaqat; Khan, Shahid; , "Acquisition strategies of GNSS receiver," Computer Networks and Information Technology (ICCNIT), 2011 International Conference on , vol., no., pp.119-124, 11-13 July 2011. doi: 10.1109/ICCNIT.2011.6020917.

5. Sarmad Ullah Khan; Luciano Lavagno; Claudio Pastrone; Maurizio Spirito; , "An effective key management scheme for mobile heterogeneous sensor networks," , vol., no., pp.98-103, 27-29 June 2011.

6. Sarmad Ullah Khan; Luciano Lavagno; Claudio Pastrone; , "A key management scheme supporting node mobility in heterogeneous sensor networks," Emerging Technologies (ICET), 2010 6th International Conference on, vol., no., pp.364-369, 18-19 Oct. 2010. doi: 10.1109/ICET.2010.5638458.

Background

Chapter 2

Background

In secure ad hoc networks, authorized nodes access the network based on network initialization, authentication and secure communication. Authentication forms the core in security, where nodes exchange data based on key management. Trusted Third Party (TPP) or Certificate Authority (CA) function as trust infrastructure and enable the nodes to access or leave the network. The main feature of security protocols is key management, which includes key distribution and key update.

Most security protocols and mechanisms need cryptographic primitives in order to integrate the security properties into their operations. These cryptographic primitives are Symmetric Key Cryptography (SKC), Public Key Cryptography (PKC), and Hash functions [40]. Symmetric Key Cryptography (SKC) can provide confidentiality and integrity to the communication channel, and require that both the origin and destination share the same security credential (i.e. secret key), which is utilized for both encryption and decryption. As a result, any third-party that does not have such secret key cannot access the information exchange. Public Key Cryptography (PKC), also known as asymmetric cryptography, is useful for secure broadcasting and authentication purposes. It requires of two keys: a key called secret key, which has to be kept private, and another key named public key, which is publicly known. Any operation done with the private key can only be reversed with the public key, and vice versa. As for (cryptographic) hash functions, they are used to create "digital fingerprints" of data. This property can be used to build other cryptographic primitives like the Message Authentication Code (MAC), which

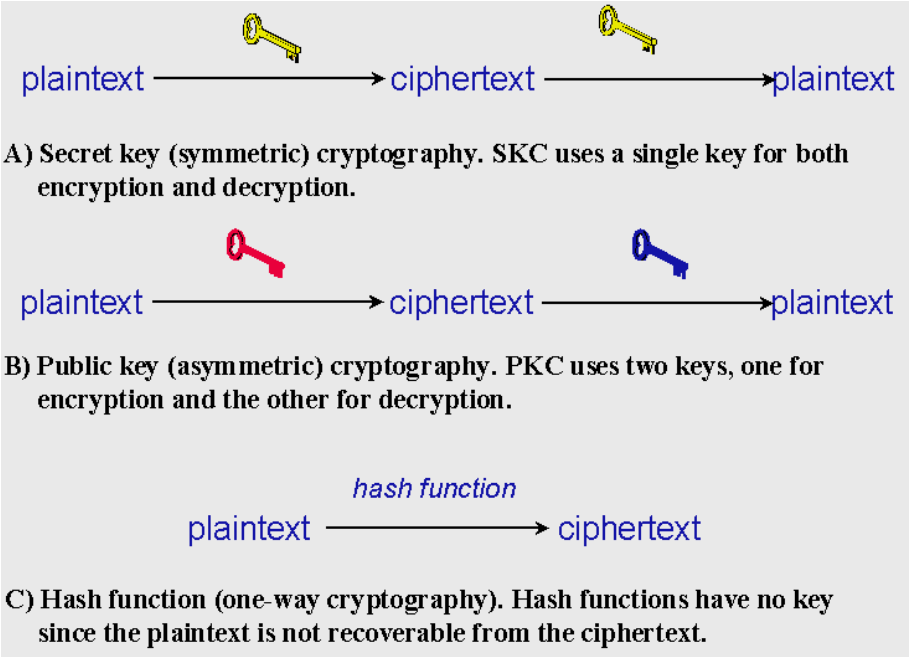

Figure 2.1. Cryptographic Types

provides authenticity and integrity of the messages. These primitives alone are not enough to protect a system, since they just provide the confidentiality, integrity, authentication, and non-repudiation properties. Nevertheless, without these primitives, it would be nearly impossible to create secure and functional protocols.

The development of optimal implementations of security primitives for sensor nodes is a very advanced research field, with solutions that can be easily used in new sensor deployments. In the area of Symmetric Key Cryptography, there are two types of primitives: Block Ciphers and Stream Ciphers. Block Ciphers are more flexible and powerful, while Stream Ciphers are simpler and faster. One of the most important block ciphers is the Advanced Encryption Standard or AES, which is the encryption standard used by all U.S. government organizations for the protection of sensitive information. While this encryption primitive is not one of the fastest primitives, it is usable in sensor nodes: one of the most optimized software implementation of AES-128 achieves an encryption speed of 286.35 Kbps, a RAM requirement of 260 bytes, and a code size of 5160 bytes running on a 8 Mhz Texas Instrument's MSP430 microcontroller with no operative system [41]. There

are even other block ciphers that, when implemented on software, offer an adequate balance between resource consumption and security. For example, the Skipjack cipher is slightly less secure than AES-128 due to its key size (80 bits), but some implementations have achieved a reasonably low encryption overhead per byte (25 μs) and a low memory overhead (code size of 2600 bytes).

Regarding stream ciphers, one of the most known ciphers is RC4, which is very simple and has an impressive speed. Although it is possible to implement it in a sensor node with just 428 bytes of code size [42], its inherent weaknesses (which are mostly concentrated on the initialization phase [43]) make advisable the use of other stream ciphers in new applications. Precisely, the eSTREAM project (organized by the EU ECRYPT network [28]) aimed to identify new stream ciphers that could be used even in constrained devices. Some of the ciphers of the resulting portfolio provide good results [44] in sensor nodes: the Salsa 20/12 algorithm requires 1412 bytes of code size in AVR platforms and it provides a throughput of 43700 bytes per second, and the Sosemanuk algorithm requires more memory (9092 bytes of code size in AVR platforms) but provides a higher throughput (67660 bytes per second)3.

Public Key Cryptography was considered to be unattainable for sensor node platforms, but that assumption was shattered a long time ago. The approach that made PKC possible and usable in sensor nodes was Elliptic Curve Cryptography (ECC), which is based on the algebraic structure of elliptic curves over finite fields. ECC has smaller requirements both in computation and memory storage, due to its small key sizes and its simpler primitives. One of the most known software implementations of ECC, TinyECC [45], implements ECC-based signature generation and verification (ECDSA), encryption and decryption (ECIES), and key agreement (ECDH). Note that the computational and memory requirements of these algorithms are not small (e.g. ECDSA requires 19308 ROM and 1510 RAM for the MICAz, generating a signature in 2s. and verifying it in 2.43s), although the implementation of these primitives is constantly evolving and improving [46].

In fact, the improvements on the implementations of ECC primitives have allowed the existence of more complex PKC primitives in sensor nodes, such as identity-based cryptography (IBC). In IBC systems, only the identity of the sensors must be exchanged, and as a result there is no need to send either public keys

or certificates. This saves energy as there is less data to be sent through the communication channel, although IBC is also very costly in terms of memory and CPU usage. One of the most optimal implementation of pairings executes the ηT (P,Q) pairing on 1.71 seconds, requiring 4.17 KB of RAM and 23.66 KB of code size running on a 8 Mhz Texas Instrument's MSP430 microcontroller [47]. While it would seem that this primitive is not useful in sensor nodes, there may be certain contexts where it could be useful, such as underwater sensor networks.

As for hash functions, some standards like SHA-1 can be easily included in sensor nodes: an unoptimized implementation needs of 122 s for digesting one byte [48]. Note that, as practical collision attacks can be found against SHA-1 [49], NIST is currently working on the selection of a new hash standard [50]. The work on this new standard is focused on PC-like platforms, although performance on embedded systems will not be overlooked. As a result, it is possible that new hash functions applicable to sensor nodes will appear soon. Nevertheless, it is not necessary to use hash functions to assure the integrity of a message if special modes of operation (such as CMAC) are used, although they require of specific block ciphers that could implement that functionality.

2.1 Key Management in Wireless Sensor Networks

The management of cryptographic keys is an important issue in resource constrained wireless sensor networks. The management of these cryptographic key should be secure and should not give a chance to the intruder to use those keys to compromise some communication links. By keeping in mind the resource constrained nature of wireless sensor networks, a number of key management schemes have been introduce to meet the wireless sensor networks requirements and provide a required level of security. The details of these existing key management schemes are described next.

2.1.1 Network Key Management

Initially, the network key approach was suggested for wireless sensor networks. In this approach, a single unique common key was used by each node in the network to encrypt and decrypt the data. For example, if a sender A wants to send some

data to receiver B, and the single unique common network key is K, then A sends data to B as

$$Encrypted\ data = C = \{data\}_K \tag{2.1}$$

When receiver B receives the encrypted data C, it will use the same key K to decrypt C to get the original data as

$$Decrypted\ data = Data = \{C\}_K \tag{2.2}$$

Although this approach suited well the constrained nature of wireless sensor networks, it is not very secure. For example, if an attacker captures any node in the network, he can easily obtain the secret key K from that node and can compromise any communication link of the network and can launch a number of attacks easily.

2.1.2 Group Key Management

A group key is used for secure group communication in a scenario where sensors in a group can send and receive messages among group members, such that outsiders are unable to glean any information, even when they are able to intercept the messages. Recently, an extensive set of papers have studied group key management issues in WSNs. In [15], a pre-distribution and local collaboration-based group rekeying scheme was proposed. However, this scheme requires each node to store all the secret shares of its neighbor's key polynomials. Therefore, the communication cost and storage cost is too high to be applied into large scale WSNs. This scheme also suffers from node isolation. If a large fraction of neighbors are compromised, a node may not be able to update its group key and thus be isolated. In [16], a group key distribution method via local collaboration was proposed. This method is simple and has high resilience to node compromise, but it did not consider node isolation, and each node needs to store all key information of each update session which incurs additional storage cost.

In [17], Zhang developed a group key management scheme to address the drawbacks of existing group key management schemes. The design of this scheme is motivated by the advantage of hierarchical structure of WSNs. Instead of letting each node collaborate with neighbors to acquire group key, in this scheme, only

cluster heads generate and distribute the group key to the nodes within the cluster.

2.1.3 Online key generation system

In the online key generation approach, each node generates the secret keys by itself. This approach reduces the memory overhead of each node by storing only few keys but it consume some amount of its energy during the generation of secret keys. Most of the online key generation approaches are based on the elliptic curve cryptography because it provides the same level of security with a shorter key length as compared to the standard public key infrastructure. Also the elliptic curve approach is not computationally expensive because of the addition operations while the public key approach contains lot of multiplication and division operations which are more expensive and make it not suitable for wireless sensor networks.

For example, Sanchez and Baldus [18] apply an Finite Projective Plane (FPP) design to the pre-distribution of Blundo polynomial shares. Their approach enables direct pairwise key establishment for a large number of nodes independent of the physical connectivity of the WSN. To reduce the memory overhead and support node mobility among different networks, Maerien [19] proposed the MAnagement of Secret keYs protocol (MASY) for mobile WSNs which assigns to a node only one symmetric key, shared only with the back-end server of its network, and which assumes a trust relationship between the newly entered network and the node's old parent network. Sajid [21] presented an online generation of secret key by storing a small number of generation key in each sensor node before the deployment of HSNs. Two low capability nodes request a high capability node to discover a shared generation key between them and use a random number to generate a secret communication key. In order to support node addition and node revocation, Poornima [22] proposed a tree based key management scheme for HSNs while Xinyu Jin [23] presented an Unpredictable Software-based Attestation Solution (USAS) for compromised node detection in mobile sensor networks.

2.1.4 Key Pool Framework

Although the on-line key generation approach reduces the memory overhead, sometimes the computational cost of the key is a critical issue. For example, if the

network is deployed in some hostile and unattended location, the online key generation approach reduces the network lifetime and the battery of each sensor node needs to be replaced more frequently. To cope with this issue, key pool framework was proposed which does not include any computational cost and also increases the network lifetime.

Perrig et al. [24] presented SPINS, a centralized keying method for sensor networks in which each node contains a secret key whose corresponding key is stored in the base station and uses one-way hash chains for creating an epoch-delayed key release mechanism for the use in authenticated broadcast. However, two sensor nodes cannot have a common secret key directly. If two nodes A and B want to establish a communication key with each other, A sends a request to B, which creates and forwards a token to the base station. The base station then generates a session key for A and B, encrypts it with the secret keys that it shares with A and B and then sends encrypted data to A and B respectively. Since the nodes use the base station as a trusted server to establish a secret key, this scheme will not work if the base station is not reachable or has a high communication overhead, especially in the case of multi-hop communication.

Eschenauer and Gligor [51] proposed a random key pre-distribution scheme that does not require the base station for the key establishment between any two nodes. According to this scheme, a set of randomly selected keys from a large pool is assigned to each sensor node before the network deployment. Two nodes communicate directly to establish a secret communication key only if they have at least one key in common. To do so, the two nodes shares their key pool information (i.e. key IDs) with each other and then they find a common key between the two key pools. Once they find a common key, the two nodes uses that key for secret key establishment. If they find more than one common key, then they first agree on single common key followed by secret key establishment. In case, if the two nodes do not find any common key between their key pools, then they get help of some intermediate node who has a shared common key between these two communicating nodes. So the two communicating nodes establish secret key through an intermediate nodes. Although this scheme does not contains any key calculation, but it consumes lot of memory space of a sensor node. For example, to increase the network connectivity in terms of key sharing probability, each node must assign a large number of keys from a

given key pool. If the network size is large, the given key pool size must be large to maintain a certain level of security which also increase the assigned key pool size to each node to maintain the same level of network connectivity as well. But assigning large key pool to each node also increases the security threats because if an attacker compromise one node, it will get its keys and will use it to compromise network links with other nodes. However, Chan [26] improved the security of [51] by introducing the "q keys" concept. To establish a secret key, two nodes must share at least q keys but this scheme requires storing a large number of keys in each sensor node than [51] to maintain the same level of network connectivity. But this scheme is also prone to the node compromission attacks. Liu [29] presented a key establishment scheme using a prior knowledge of node deployment coupled with Rabin's scheme [30] to achieve a high degree of connectivity (while reducing the memory cost) and network resilience against the node capture attacks. Key pools are divided according to the deployment regions and each node is assigned keys from those key pools. This increases the probability of finding a common key between the key pools of two communicating nodes. Zhang [31] presented the NPKPS pairwise key pre-distribution scheme for WSNs to achieve better security, connectivity and efficiency and less memory cost compared to [51]. Efficient authentication schemes are proposed in [32] and [33] which improve over past work in terms of security, authentication overhead and storage requirements.

In order to present a key management scheme that reduces energy cost and supports node mobility, Kim [34] proposed a level-based key management scheme for multicast communication that has reasonable routing overhead and low mobility management overhead. For mobility supported cluster-based WSNs, a two-layered dynamic key management scheme was proposed by Chuang [35] while polynomial-based key pre-distribution scheme for mobile sensor networks was proposed by Blundo [36].

For HSNs, Du [37] presented an unbalanced key pre-distribution scheme to improve network connectivity, reduce memory overhead and provide better network resilience compared with existing key management schemes for homogeneous sensor networks. Nodes with high capabilities are assigned m keys, while nodes with low capabilities are assigned l keys, $m >> l$. Zhang [39] presented a group oriented key management scheme for HSNs in which a large key pool is split into a sub key

pool for each group, while a routing-driven key management scheme based ECC is presented by Du [58]. Their results show better connectivity and network resilience than [51], [37]. Symmetric key distribution based on public key cryptography using prior knowledge of sensor deployment location provides better resilience against node capture, as well as lower memory cost and computational overhead [59].

2.2 Key Management in Content Centric Networking

Since little investigation has been made in securing CCN, there is no specific key management scheme for that. Here we describe some of the existing key management schemes proposed in the literature for the static and mobile ad hoc networks which is a field of networking where key dissemination in the network has been investigated for years.

Basically, cryptography is divided into two main categories (1) Symmetric key cryptography (2) Asymmetric key cryptography. In symmetric key cryptography, every node in the network is assigned N-1 keys where N is the total number of nodes in the networks. This is not suitable for a large networks because each node is required to store large number of keys. In asymmetric cryptography, each node is assigned a pair of keys (i.e. public key and private key) for secure communication with other nodes in the network. Recent research works in cryptography are mainly based on the traditional public key infrastructure (PKI: [60], [61], [62], [63]), and identity based public key cryptography (ID-PKG: [64], [65]). Here we discuss the compatibility of those approaches with CCNs.

Smetters in [14] suggested the standard PKI approach for CCNs. In this approach, each node has a pair of keys (public key and private key) and for secure communication, each node publishes its public key along with its certificate, assigned by a certification authority. Each node in the network verifies the public key of other nodes by sending the attached certificate to the certification authority for validation. The certification authority check the validity of the certificate and sends the acknowledgement back to the node about the authenticity of the certificate. Since PKI is based on the concept of a single centralized certification authority, it

does not suit well the concept of CCNs, where contents are replicated in the network and are requested and routed by name instead of being provided by a single source. This increases the number of public/private key pairs, which in turn increases the verification overhead for the single certification authority. ID-PKG completely eliminates the need for public key certificates by exploiting publicly known user identity information (such as IP address or telephone number) as a public key for securing information. ID-PKG enables any pair of users to communicate securely without exchanging public key certificates, and without using the online services of a third party. This is enabled by a trusted Private Key Generator (PKG), which generates the private keys of the entities using their public keys and a master secret key. In 2003, the first ID-PKG cryptography management and certification scheme [64] for mobile ad hoc networks was presented by Khalili and Katz. The basic idea of the scheme is similar to the scheme of Zhou and Haas [60]. A group of selected nodes shares the responsibility of managing the PKG. Each node can obtain a system private key share from a predefined minimum number of PKG manager nodes, which collaborate with the node to generate the private key.

This scheme has several main problems. (1) These papers did not mention many potential problems that can arise on the channel between PKG nodes and user nodes. Consider for example the fact that a user's private key could be easily wiretapped by attackers. (2) The scheme does not mention how to identify the nodes that manage the PKG and how to gain their private key when new nodes are added to the network. (3) The scheme does not explain how to update the main key pair of the system (including public and private key).

Hence in 2004, Deng proposed a new cryptography management and certification scheme called ID-PKC [65]. In this scheme, there is a master public/private key pair. The master public key is known by all nodes in the network, while the master private key is divided into shares and distributed among k nodes of the network (fewer than the total number of nodes). Each node ID is working as node public key and for secure communication, it needs its private key. So the node sends a request to k PKGs to get its private key share. The requesting node generates a temporary public/private key pair and gives that public key to the PKGs to get its encrypted private key share. After getting those shares, the node generate its private key. Then the PKGs announce the requesting node ID to all nodes of the network, to be

used as its public key. This scheme, however, still does not address the problem of updating the main key of the system.

The system based on ID-PKC is a powerful alternative to PKI in terms of both efficiency and convenience. However, it also has several problems, such as the key escrow problem in case of PKGs compromission and suffers from a single point of failure (as shown in [65]), because there is only one PKG responsible for generating the private keys to the nodes. Although this second problem was solve by [64], the key escrow problem still exists, because the private key of the PKG is compromised, the entire system is compromised.

Key Pool Framework for Wireless Sensor Networks

Chapter 3

Key Pool Framework

Sensor networks consist of resource constrained devices in terms of storage space, computational power, communication range and battery lifetime. It is not suitable to adopt a key establishment and management approach that consume lot of these resources (e.g. traditional key establishment and management approaches used for the resource full computer networks). Keeping these constraints in mind, public key cryptography, which is considered the most secure approach and very much computational expensive, does not suit sensor networks. However, symmetric key approach that does not involve any computation suits sensor networks constraints but in case of large networks, it occupies lot of memory space of sensor nodes. To overcome memory cost, key pool framework is considered as the best solution for the key establishment and management in wireless sensor networks because it does not contain any computations and occupy less memory than the standard symmetric key approach. A number of key management solutions have been introduced for the key pool framework to optimize the memory consumption of this approach. But those solutions were proposed for the static networks. Hence, we considered the mobility scenario and optimized memory cost of the key pool framework.

3.1 Network Model

The network model that we considered in our proposed scheme is based on the model described in [68], in which all sensor nodes are divided into two categories

i.e. Fixed sensor Nodes (FNs) and Mobile sensor Nodes (MNs) as shown in fig. 3.1. The capabilities of FNs and MNs are different in terms of energy resources, memory, computational power and transmission capabilities. MNs and FNs communicate directly if they both are in the communication range of each other. Moreover, MNs can communicate with other MNs but the establishment of the communication key is performed by a FN if it is present; otherwise, the MN establishes a temporarily communication key until a FN becomes available. It is assumed that FNs are deployed such that the MNs are considered as end devices of the network. But in real deployment scenarios, this assumption could be not true.

FNs are expected to have additional radios (e.g. IEEE 802.11) and to be able to communicate with the Base Station (BS), with other FNs and with the MNs within their radio coverage area. If a MN is in the radio coverage range of more than one FN, then the MN will select one of the FNs depending upon the best signal to noise ratio, link quality, availability and bandwidth.

3.1.1 Cluster Formation

Figure 3.1 shows the cluster representation of the proposed network architecture. For authentication purposes, the authentication key material is distributed to both FNs and MNs. The proposed solution is based on an unbalanced key distribution scheme compared to the adoption of a balanced approach.

In fact, using a balanced key distribution scheme [69], a large pool of size P of secret keys is generated from which K keys are randomly selected for each sensor node. Two nodes may establish a communication key if they have a common authentication key. The probability that two nodes share at least one common key in [69] is

$$P[Match] = 1 - \frac{(P-K)!(P-K)!}{P!(P-2K)!} \tag{3.1}$$

But if the nodes do not have a key match, they can still establish a communication key through one or more intermediate nodes with whom they have a common authentication key.

In this chapter, we propose an unbalanced approach for the authentication key distribution and for this a key pool of size P is generated from which a key ring of

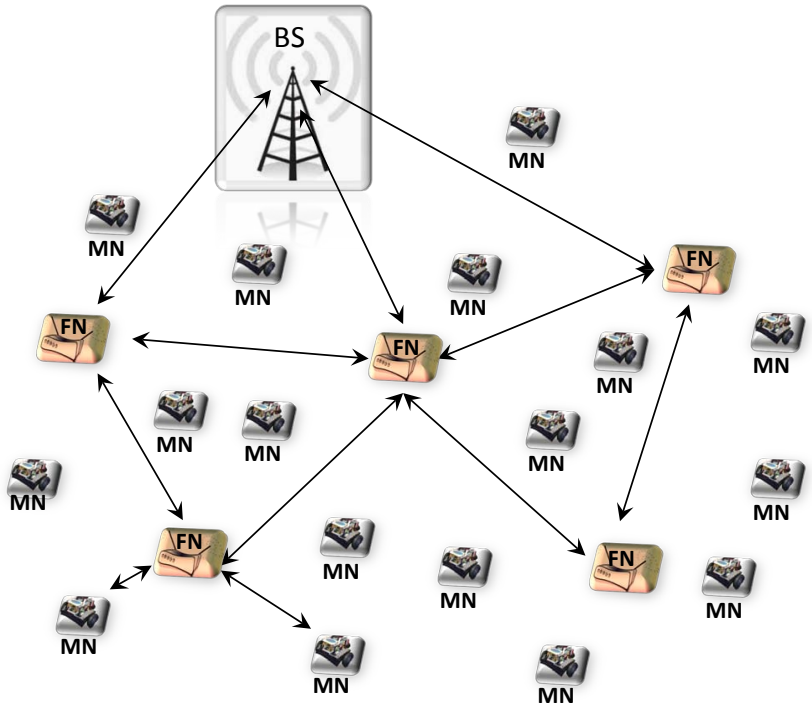

Figure 3.1. Cluster Formation

size K is assigned to each MN and a key ring of size S to each FN, where $S >> K$ (Unbalanced Key distribution). The generation of key pool of size P is described later. These keys are used for the authentication and for exchanging a communication key. MNs and FNs can authenticate each other using these keys if both are within their radio range. Then the FN assigns a communication key to the MN for secure communication. In case the MN is not able to communicate directly with the FN then it establishes a path key with the FN through intermediate MNs. The intermediate MNs can then authenticate this MN using its own FNs who assigned them a communication key. For the authentication between the FN and the MN, both must have at least one key in common: the relevant probability is given by

$$P[Match] = 1 - \frac{(P-K)!(P-S)!}{P!(P-S-K)!} \tag{3.2}$$

The probability of having at least one common key between the MNs is given by 5.1.

In order to further improve secure authentication process and communication key exchange and establishment phase, q shared key concept can be used as described in [26]. And according to this, the probability that a MN and a FN containing key rings of different sizes share exactly 'i' keys is

$$p(i) = \frac{\binom{P}{i}\binom{P-i}{(S-i)+(K-i)}\binom{(S-i)+(K-i)}{S-i}}{\binom{P}{S}\binom{P}{K}} \tag{3.3}$$

The probability that a MN and a FN share at least q keys used for the authentication is

$$1 - \sum_{i=0}^{q-1} p(i) \tag{3.4}$$

3.2 Proposed Algorithm

A number of secure key management schemes exist in literature but here we propose a new key management scheme that is more robust against node compromised attack, has less memory overhead, and enhanced connectivity. After the keys distribution phase among the sensor nodes (i.e., MNs and FNs), we describe authentication, connectivity and key establishment phase.

Notations that are used in this chapter:

SSC_1	Secret communication key generation code-1
SSC_2	Secret communication key generation code-2
m_1	Key pool containing SSC_1
m_2	Key pool containing SSC_2
P	Authentication key pool size
S	Authentication keys assigned to FNs
K	Authentication keys assigned to MNs
K_{plc}	Network public key
K_{prt}	Network private key

3.2.1 Key Distribution before Deployment

In our proposed scheme, we have a large key pool as in [69] and the BS selects a random sub key pool of size P as an authentication key pool. After that, the BS assigns a random key ring of size m_1 and key ring of size m_2 from a large key pool to the FNs, where $m_1 = m_2$. These m_1 and m_2 key rings contain secret communication key generation code SSC_1 and secret communication key generation code SSC_2 respectively. Then the BS will transmit a random key ring of size S to fixed sensor nodes and a key ring of size K to mobile sensor nodes selected from the key pool P along with the key identifiers such that $S >> K$. FNs are also assigned the authentication key identifier list which are assigned to the MNs. The numbers of FNs are less compared to MNs and their position is fixed so that there is no need to establish authentication key every time they communicate with the base station (BS) or with other FNs. Thus FNs are assigned only public/private key pair for the authentication and communication with the BS and other FNs. Each FN and the BS knows the public keys of all the FNs. Thus the FNs are assigned three key pools i.e., authentication key pool S, m_1 and m_2 along with the public keys of other FNs and K_{prt}. This is because the FNs have more memory space compared to the MNs.

3.2.2 Authentication and Connectivity

After the network deployment, authentication is performed between the FN and the MN. To this aim, the MN must have at least one common authentication key with the FN or, in more secure authentication process, MN must shares at least q common authentication keys with the FN for the direct authentication and the probability of having at least one common key and q common keys are given by 5.2 and 5.4 respectively. Thus the MN sends a request including its ID encrypted by K_{plc} to FN to ensure the global connectivity and FN decrypts it using K_{prt} and responds with an authentication nonce encrypted by authentication key(s) of MN along with the key identifier(s) as the FN has authentication keys information of the MN.

In case of no key match or less key match from the given threshold as in case of q-key authentication, the FN will authenticate MN through the base station as base

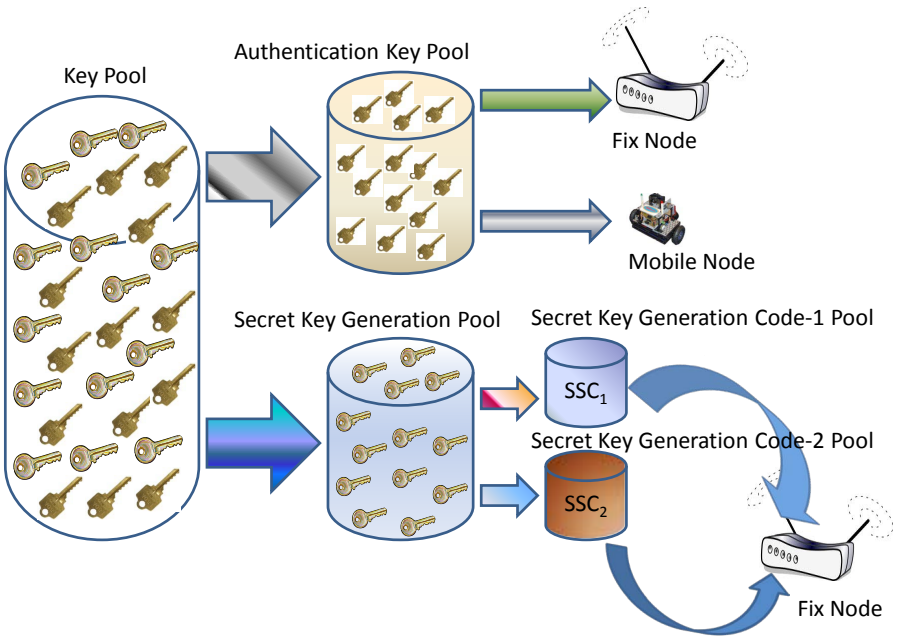

Figure 3.2. Key Distribution

station have the full key map of all MNs. In fig. 3.3, we present the probability of having at least one common authentication key between the FN and MN using 5.5 which is the simplified version of 5.2.

$$P[Match] = 1 - \frac{(1-\frac{K}{P})^{P-K+0.5} * (1-\frac{S}{P})^{P-S+0.5}}{(1-\frac{K+S}{P})^{P-S-K+0.5}} \tag{3.5}$$

The MN may come into the radio range of more than one FN then the probability of authentication of MN by the FN in network is given by

$$P[Authentication] = 1 - (1 - P[Match])^{a} \tag{3.6}$$

where 'a' is the number of FN and P[Match] in 5.6 is defined by 5.5. Fig. 3.4 represents that the coverage of MN by more than one FN increases the authentication probability of MN. However, the selection of the FN depends on the link quality, availability and bandwidth and is done by the MN. As the MN uses the K_{plc} key for the request to join or start the communication with the FN, the global connectivity of the network is 100%. Also when the MN moves from the range of one FN to

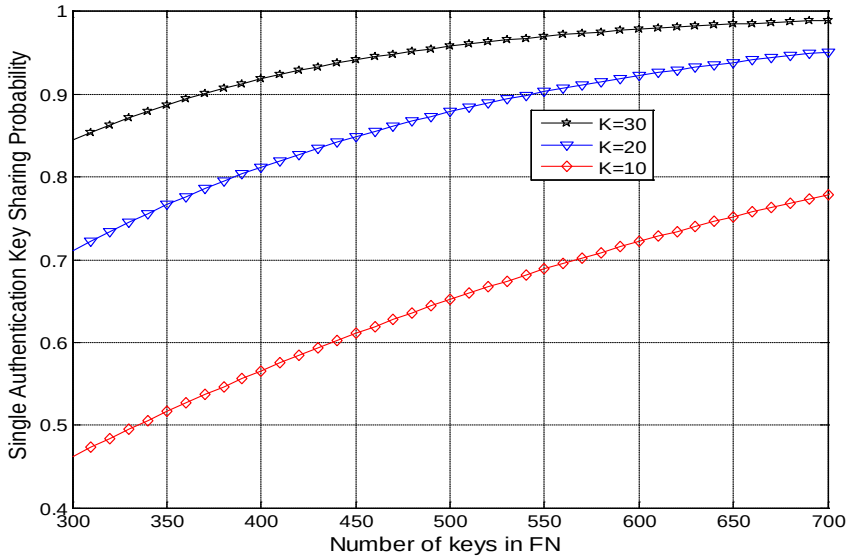

Figure 3.3. Probability of sharing at least one common authentication key between the FN and MN

the other FN, it sends the previous FN identity to the new FN so that the new FN communicates directly with the previous FN using his public key in order to reduce the broadcast overhead about the incoming MN and the new FN authenticates this MN using the previous method described above. If the new FN is not able to authenticate this MN due to the insufficient number of common authentication keys (compared to the given threshold) then it will authenticate this MN using the previous FN. In case the previous FN also authenticates this MN through the BS due to lack of authentication keys then it would not be able to authenticate this MN for the new FN and new FN will authenticate it through BS. The authentication among the FNs is done through the public/private key pairs. Each FN knows the public keys of all other FNs and the BS.

3.2.3 Communication Key Establishment

When the MN is authenticated as a network authentic member then the FN creates a communication key for that MN and sends it to the MN in the encrypted form

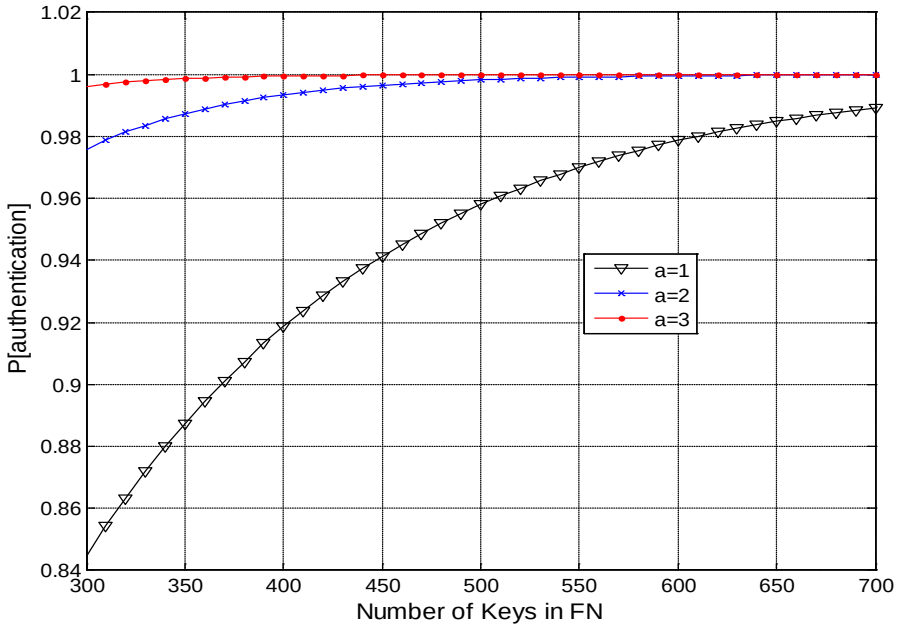

Figure 3.4. Probability of authentication of MN with the range of more than one FN with K=30

using q-authentication keys of the MN. When the FN does not have common keys up to the given threshold q for the encryption then the FN asks the BS to generate an encryption key for the encryption of communication key for the specific MN. The BS generates encryption key and sends it to the FN along with the key identifiers of the authentication keys of MN encrypted by the public key of the FN. The FN uses that key for the encryption of communication key and informs the MN about the key identifiers sent by the BS so that MN can generate a key for the decryption of the message to get the communication key. The communication key is generated using the key rings m_1 and m_2 and the total number of the key identifiers used to generate the encryption key for the communication key. The generation of the communication key is given by

$$Communication\ key = SSC_1^q \ \ mod \ \ SSC_2 \tag{3.7}$$

Here q is the total number of the key identifiers. For example, SSC_1=25325687, SSC_2=54136752 and q=2, then communication key=01437025.

3.3 Analysis and Evaluation

3.3.1 Memory Cost

Let the key rings m_1 and m_2 be composed of 100 keys. These two key rings are independent from each other. The total possible number of communication keys generated by these two key rings is 10,000. If we compare our proposed scheme with the previous scheme proposed by Eschenauer and Gligor in [51], instead of storing 10,000 keys in a sensor node, only 200 keys are required to get 10,000 communication key combinations. This reduces the memory cost of the node while maintains the same security level. The reply attack is avoided by using the time stamps and hash function. MN includes the hash of time stamp in the message encrypted by K_{plc} which is further encrypted by the communication key. Upon the reception of message, FN checks the time stamp: if it is valid, message is accepted. In this way the message freshness is assured and reply attack is avoided. Each FN also sends the node IDs of all the MNs to the BS with whom it is communicating. By doing this, the BS would be able to keep the track of each MN in the network and it helps against the node replication attack.

3.3.2 Resilience Against Node Capture Attack

Node compromised attacks are possible on FNs and on the MNs. But in this paper, we deal with only the MN compromised attack. As the MNs are assumed to be the end devices, most of the time they communicate with the FN and a communication key is assigned to MN by FN. To ensure the reliability and availability of the network in the case when FN is out of reach from the MNs then they can establish a direct communication link with other MNs or with the FN through intermediate MNs using its authentication key pool. Thus the MNs plays an important role in the network communication and connectivity. First, we present the probability of establishing a communication link between the two MNs and it is only possible when these two MNs have a common authentication key. The results are shown in the fig. 3.5, which are achieved by varying the size of the key pool P for each different key pool size of MN. This shows that lower the size of K compared to P, lower is the probability of establishing a communication link between the two MNs. Thus if one

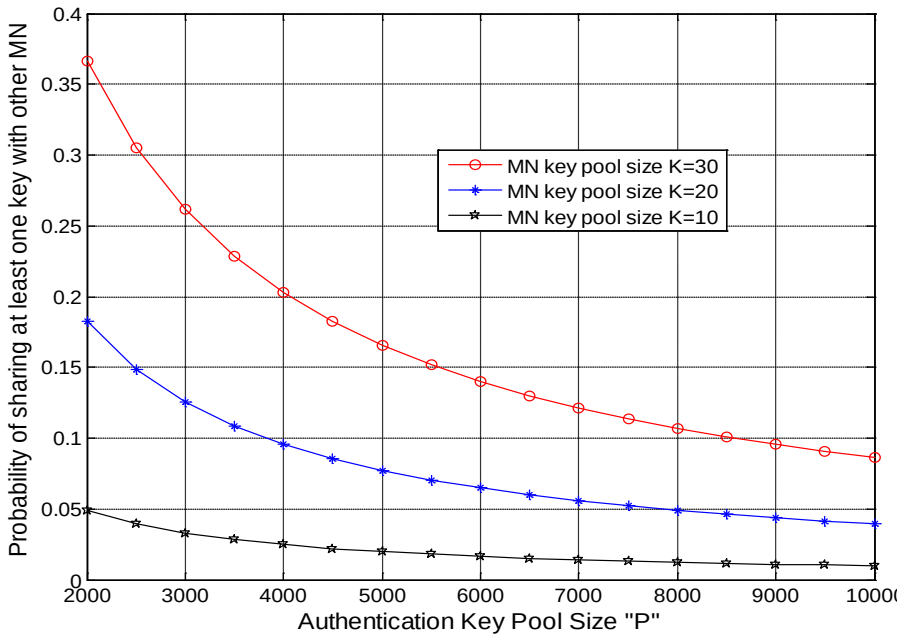

Figure 3.5. The probability of establishing a communication link of one MN with the other MN by varying the size of authentication key pools 'P' and 'K'

of them is a compromised node, then the probability of establishing a compromised communication link with other MNs is much smaller which also reduces the probability for the uncompromised MNs to establishing a compromised link with the FN through this intermediate compromised MN. The probability of the fraction of communication compromised by compromising more than one i.e., n MNs is given by

$$P[Compromised] = 1 - \left(1 - \frac{K}{P}\right)^{n} \tag{3.8}$$

where K is the number of keys stored in the MN and P is key pool size from which K is randomly selected for each MN. Fig. 3.6 show the result of node compromised attack of our proposed scheme compared with the basic scheme in [51] and NPKPS scheme in [31]. This graph shows that our scheme substantially lowers the fraction of compromised communication after n nodes are compromised. The main reason for this improvement is that, in our scheme, we store less number of keys (that introduce the possibility to compromise communication links) in MN compared to the basic scheme.

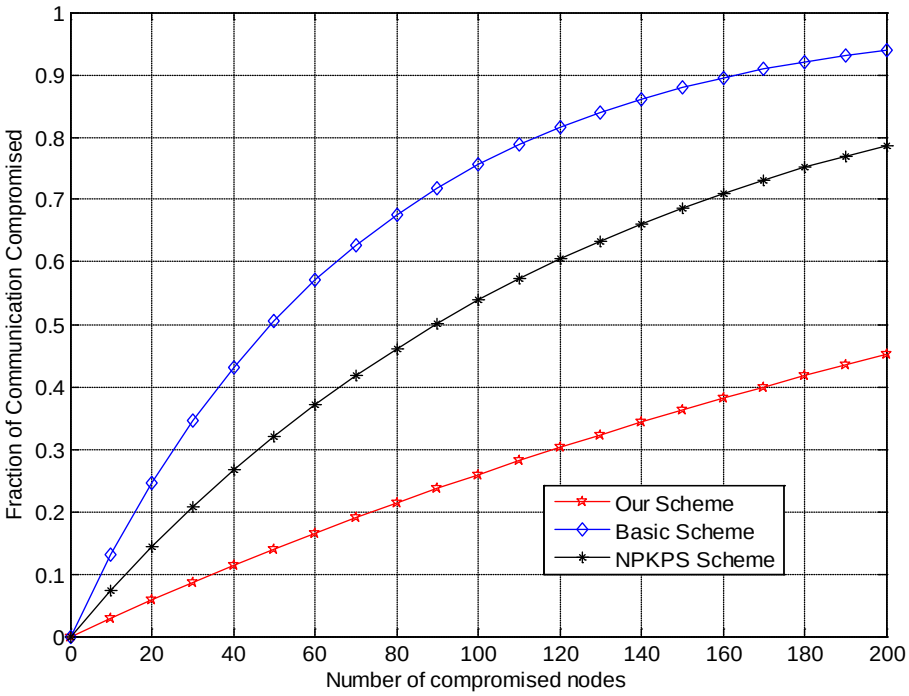

Figure 3.6. Fraction of communication compromised by capturing 'n' MNs

On-line Key Generation Framework for Wireless Sensor Networks

Chapter 4

Online Key Generation Approach

Wireless Sensor Network (WSN) technology is being increasingly adopted in a wide variety of applications ranging from home/building and industrial automation to more safety critical applications including e-health or infrastructure monitoring. This increases the networks size in terms of total number of sensor nodes deployed in the networks. This increase in sensor nodes makes the use of key pool framework approach a challenging task because of the network connectivity in terms of common shared key and node compromission attacks. These issues force researchers to find an alternative solutions. Hence the on-line key generation approach using the Elliptic Curve Cryptography (ECC) [45] is considered a best solution which is less computationally expensive and provides the same level of security with a shorter key length than the standard PKI approach. We also adopt the on-line key generation approach to reduce the memory cost and keep the communication overhead low during the authentication and key establishment phase.

4.1 Proposed Algorithm

In this Chapter, the proposed key establishment scheme for HSNs is presented. The reference network model defines a HSN composed of a Base Station (BS), Fixed Nodes (FNs) and Mobile Nodes (MNs). These nodes are heterogeneous in terms of computational power, memory and energy resources. However, the same communication technology is adopted. The BS and the FNs are powerful devices while MNs

are characterized by very limited resources and can change their position within the given environment following a specific mobility model. Moreover, the MNs are more numerous than the FNs, and only the FNs need to be equipped with tamper-resistant hardware.

In this resulting scenario, the BS only communicates with the FNs and acts as a trusted server for them. To address scalability issues, a cluster-based approach has been adopted (similarly as in [20]). In fact, FNs act as Cluster Heads (CHs) and are in charge of managing authentication and key establishment operations for a group of MNs.

4.1.1 Overview of Proposed Algorithm

Node authentication and key establishment are the basic security features provided by the proposed solution. First the authentication phase is performed among the FNs and the MNs. Once the authentication phase is successfully completed, key establishment operations can be performed among the MNs and FNs. The proposed scheme supports mobility by providing the two considered functionalities in a scenario where MNs move from one cluster to another one. Fig. 4.1 depicts the main operations of the proposed scheme for the HSNs.

4.1.2 Key Pre-Distribution

In this section, key pre-distribution among the FNs and the MNs is described. A secret key (SK) is assigned to every MN; more specifically, such key is generated using a Secret Key Generator (SKG), a prime number that is pre-assigned to each MN of the network (MNPN), and the two randomly generated prime numbers (using MNPN and SKG as a seed to the prime number generator) using a one way secret key generation function f( ). The key establishment procedure is discussed in section 4.1.5.

Each FN is provided with the following key material: the public key of the BS, its own public/private key pair, a one-way authentication key generation function g(), a Secret Key Generator (SKG) and a one way secret key generation function f( ), a Compromised Node Detection Key (CNDK) and a network private key (K_{prt}) along with its own prime. It is worth noting that FNs must also implement a fast

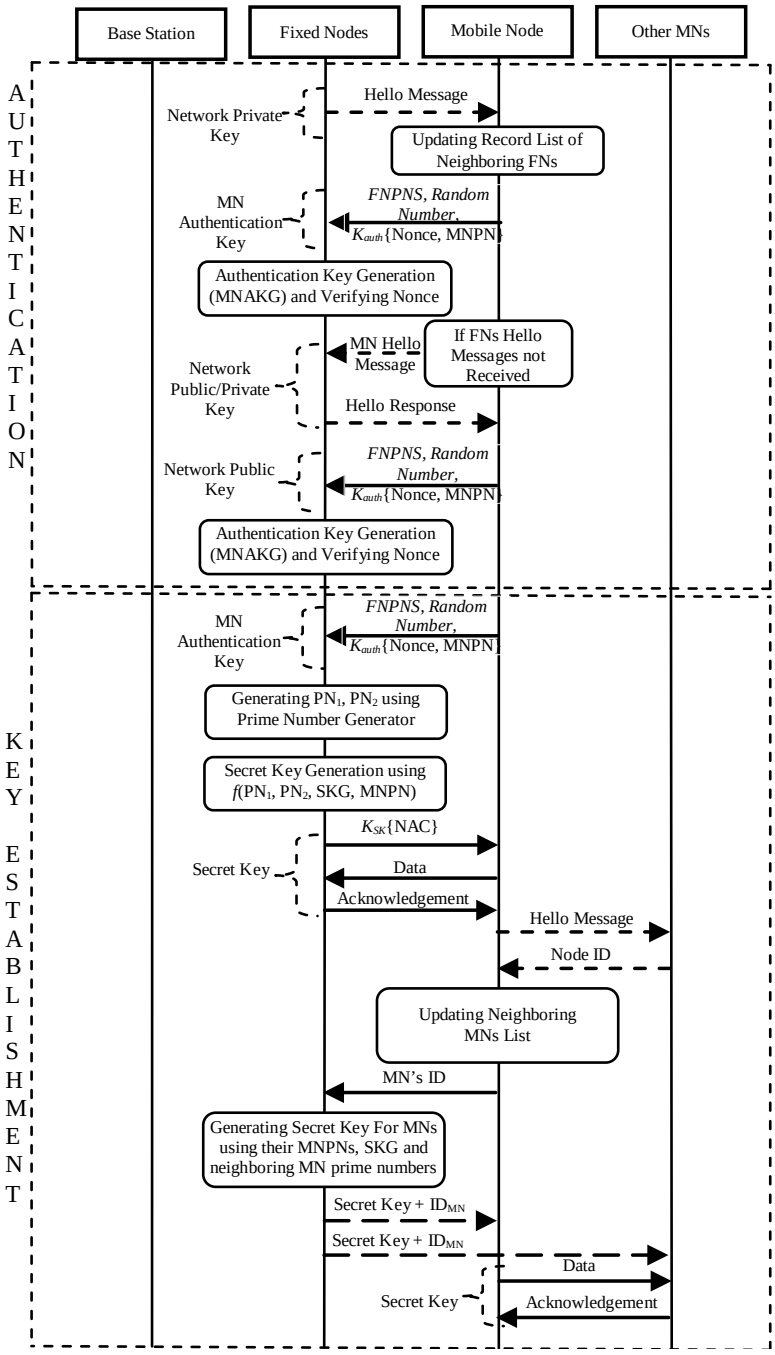

Figure 4.1. Overview of proposed algorithm

key revocation algorithm in order to protect the Secret Key Generator (SKG) and the network private key.

As far as the MNs are concerned, the following key material is considered: a secret key (SK), a network public key K_{plc}, an authentication key K_{auth}, the Fixed Node Prime Number Sum (FNPNS), its own prime number and a random number.

4.1.3 Cluster Formation

Once the network is deployed, FNs start the cluster formation phase. During the cluster formation phase, all the FNs periodically broadcast a Hello messages to neighboring MNs for a given number of times (3-times in the proposed scheme). Such messages include nodes IDs and a random nonce encrypted using the network private key. It is assumed that the FNs are deployed such that most MNs receive Hello messages from more than one FN. The selection of FN as a CH depends on the Hello message signal strength. Each MN also keeps a list of neighboring FNs from which it has received Hello messages to possibly identify backup CHs. If, for some reason, the MNs do not receive any FN Hello message within a given time period, they start broadcasting Hello message including a nonce encrypted by the network public key to discover authentic neighboring FNs. Fig. 4.2 describes the virtual network organization.

4.1.4 Mobile Nodes Authentication

To get access to the network, each MN needs to authenticate itself with a selected CH. To do so, the MN sends a "Join request" encrypted using the network public key. The request includes the Fixed Node Prime Number Sum, a random number related to its authentication key and the nonce provided within the Hello Message sent by the selected CH along with its own prime number encrypted by MN's authentication key. Once received such information, the CH is able to infer the authentication key of the MN by using one-way authentication key generation function g() as follows

$$K_{Auth} = g(FNPNS,\ Random\ Number,\ SKG) \tag{4.1}$$

Successful decryption of the encrypted nonce and MN prime number by the CH using the inferred K_{auth} proves the MN authenticity. It is worth noting that the

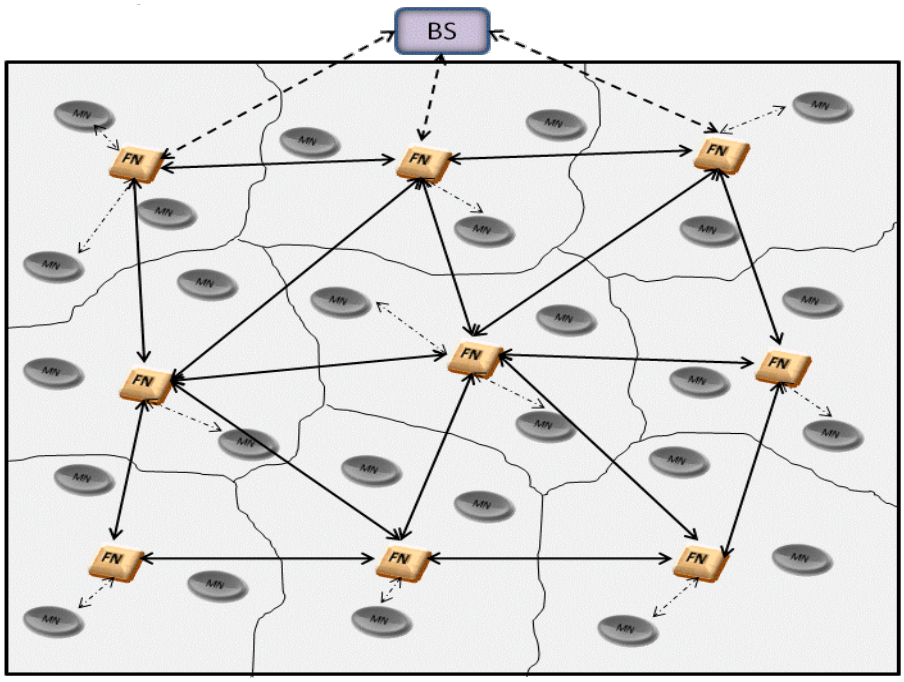

Figure 4.2. Network Topology

use of SKG in K_{auth} generation guarantees that an authentic FN is actually generating the K_{auth} and that MN prime number is not revealed to any adversary node. After the authenticity check, the FN sends the joining confirmation and a Network Authentication Code (NAC) to the MN. This is used to reduce the authentication burden while the MN moves through different clusters within the same network. The Network Authentication Code is also periodically updated as a countermeasure to replay attacks or node replication attacks performed by an adversary.

4.1.5 Key Establishment and Management

To secure communication between the CH and the MN, each MN is assigned a secret key SK while its generation function is assigned to the FNs before the deployment. During the authentication phase, each CH receives the MN prime numbers of its member MNs. CHs using these MN prime numbers and the secret key generator SKG generate the first prime number using prime number generator (PN_1); this prime number is further combined with the MN prime numbers and secret key generator SKG to generate the second prime number (PN_2). Then, the CH generates

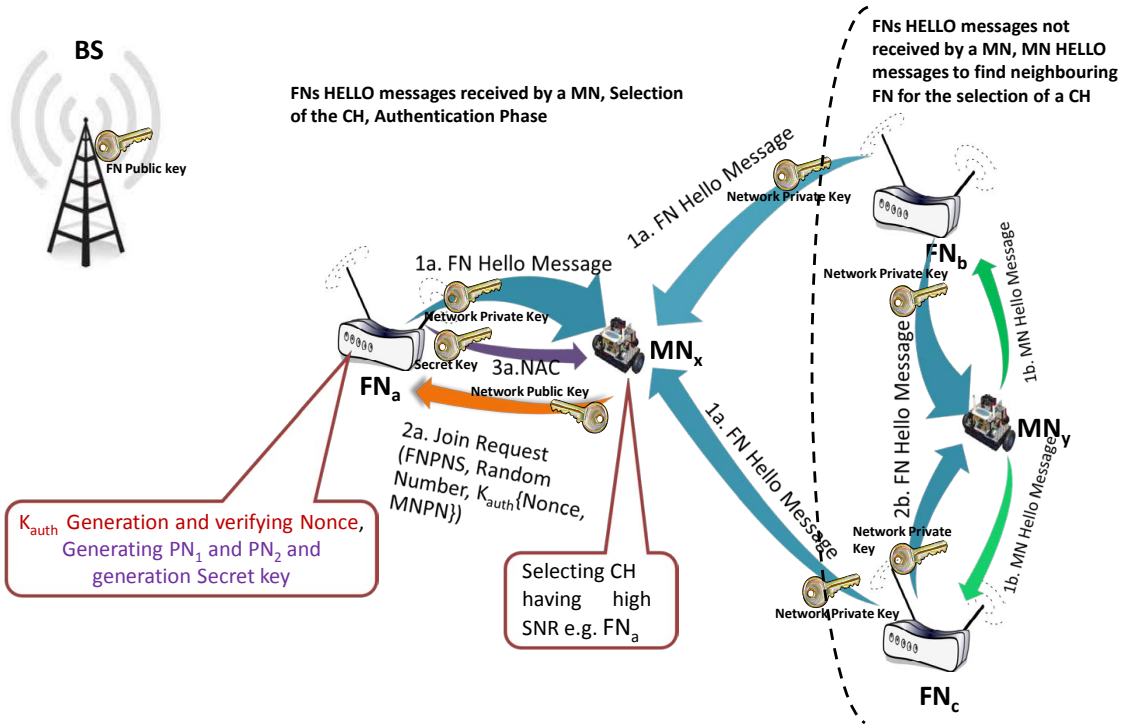

Figure 4.3. Authentication and Key Establishment Phase

the required secret key using a one way secret key generation function f( ), thus obtaining the same secret key owned by the specific MN

$$Secret\ Key = f(PN_1,\ PN_2,\ MNPN,\ SKG) \tag{4.2}$$

For secure communication between the MNs, a secret key between them is generated by the CH. For instance, if a mobile node A wants to establish a direct communication link with mobile node B, it sends its IDA along with the IDB to its CH. Then the CH generates a secret key for them using their IDs, prime numbers and one way secret key generation function f( ) and sends it to both MNs using the secret key shared with each of them. CHs also periodically inform the BS about their member MNs to avoid the node replication attacks in the network.

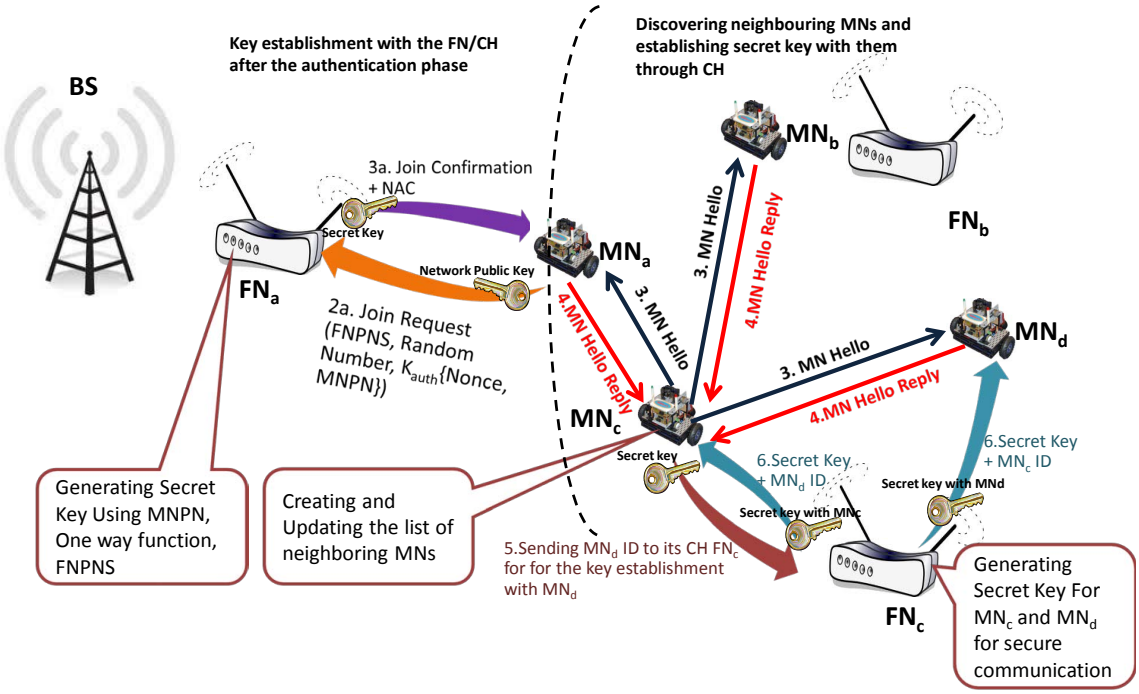

Figure 4.4. Key Establishment between the MNs

4.1.6 Handover

During the movement of a MN to perform its task, it may move from one cluster to another cluster in the network. To provide full connectivity to MNs, FNs are deployed such that each MN should normally receive Hello messages from more than one FN. One of the FNs is selected as a CH by the MN based on the Hello message received signal strength, while information about other neighboring FNs is kept as a backup. A MN moves from one cluster to another cluster if it finds its CH signal strength dropped below a certain threshold during its periodic check. Before the transition to the new cluster, a MN sends broadcast Hello messages to discover new neighbors and update the relevant list. Once the CH is selected base on the signal strength of the Hello message response, the MN sends to the old CH a leaving message also including the new CH ID and sends a join request containing the Network Authentication Code to the new CH. After the verification of the Network Authentication Code, the new CH contacts the old CH of the MN asking for its prime number. If the old CH received the leaving message from its MN including

the new CH ID, it confirms the MN movement to the new CH by sending the MN prime number and also informs the BS to avoid node replication attacks. After receiving the MN prime number, the new CH accepts the joining request from the incoming MN. If, due to e.g., poor radio coverage or packet losses, the MN leaving message is not received by its previous CH and its joining request is received by the new CH, then upon the reception of the MN prime number request from the new CH about its MN, the previous CH tries to contact its MN to confirm the transition. If it receives a positive response from its MN or no response, it sends the MN prime number to the new CH and also informs the BS that the specific MN is moving to another CH. After receiving the MN prime number, the new CH generates the secret key SK of this MN and informs the BS about new MN ID. The old CH deletes this MN from its MN members list and the BS updates the MN members lists related to the other CHs in order to avoid node replication attacks.

4.1.7 Addition of New Mobile Nodes

MNs are unreliable devices with limited power supply; hence they may fail or run out of power over time. This can cause coverage and connectivity problems in WSNs, significantly degrade network performance and shorten network lifetime. To overcome these problems, some MNs could be replaced and new MNs would be added in the network. However, adding new MNs poses new challenges for security schemes such as the establishment of security keys with the existing FNs and MNs; in fact, newly deployed MNs could be compromised or could be malicious nodes.

In the proposed scheme, a newly deployed MN is pre-loaded with a special authentication code along with the authentication key. The BS is in charge of informing the FNs about the addition of the new MN, also providing the relevant ID and a special authentication code. The purpose of this special authentication code is to avoid the Sybil attacks in which an adversary can create multiple copies of the compromised MN with new IDs and introduce them as new nodes in the network.

The newly added MN broadcasts a Hello message encrypted using the network public key to discover its neighboring authentic FNs. This Hello message includes the MN ID and special authentication code. Upon the reception of Hello message from the new MN, neighboring FNs will check the special authentication code by

comparing it with the BS-provided code. After successful verification, FNs send their cluster identities and authentication nonce encrypted using the network private key. After receiving the response from the neighboring FNs, the MN will select one of the authentic FNs having the best signal to noise ratio as its CH and will send a joining request to establish a secret key and get the Network Authentication Code from its CH.

4.2 Performance Analysis

In this section, the proposed scheme is analyzed using the OMNET++ simulator, in terms of network connectivity, network resilience against node capture attacks, energy consumption, memory cost and communication overhead. The simulation results have been obtained using OMNET++ 4.1 with the mobility framework MiXiM 2.0.1. The simulation scenario is defined by a network composed of 500 MNs and 16 FNs. The size of the network simulation area is 400m x 400m. Both the FNs and the MNs use the 802.15.4 CSMA and radio specification based on the CC2420 radio chip. The transmission power is set to 10mW and sensitivity is set to -95dBm for all nodes. The mobility of the MNs is described by the random walk mobility model in which the speed of the MNs is constant but a random direction is chosen periodically within a predefined range. More specifically, the speed of the MNs is set to 1m/s and their direction update interval to 0.1s. The simulations were repeated 3 times for 5000 seconds for each result.

Result comparison of the proposed solution with other existing key management protocols shows better network connectivity and resilience, with a significant reduction in memory and communication overhead.

4.2.1 Network Connectivity

In order to show the effectiveness of the proposed solution in terms of network connectivity, the simulation results of the proposed scheme are compared with [51], [37], [38] and [39] where the connectivity depends on the key sharing probability. For a balanced key pre-distribution scheme, the single key sharing probability between

the MN and the FN is given by

$$Pr[Connectivity] = 1 - \frac{(P-K)!(P-K)!}{P!(P-2K)!} \tag{4.3}$$

where K is the number of keys assigned to FNs and MNs from a pool of P keys. Instead, for the unbalanced key pre-distribution [37], [38] schemes, the single key sharing probability is given by

$$Pr[Connectivity] = 1 - \frac{(P-K)!(P-S)!}{P!(P-S-K)!} \tag{4.4}$$

where K is the size of the key pool assigned to each MN and S ($S >> K$) is the size of the key pool assigned to each FN.

In the proposed scheme, the K_{plc} is assigned to each MN and the K_{prt} to each FN before network deployment. These two keys connect a MN to an authentic FN of the network. Hence the connectivity of the network is almost 100% if and only if the FN is not compromised. Fig. 4.5 shows the comparison of OMNET++ simulation results for the network connectivity of the proposed scheme with [51], [37], [38] and [39].

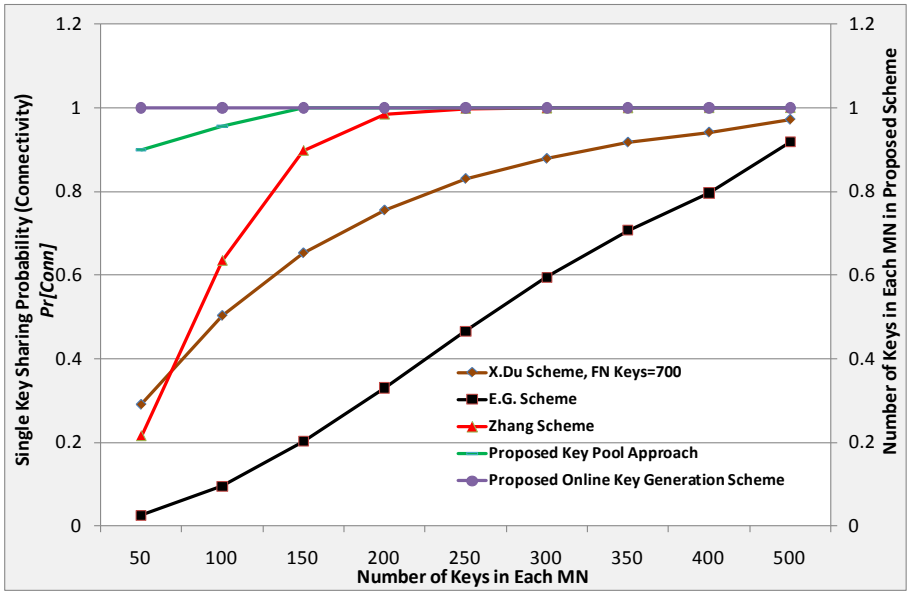

Figure 4.5. Probability of sharing at least one common key (Connectivity)

4.2.2 Memory Cost

This section presents the comparison of memory overhead of the proposed scheme with some well known existing key management schemes for HSNs.

In an ECC-based key management scheme [58], the total memory overhead is $(n_{MN} + 3) * n_{FN} + 2n_{MN}$, where n_{MN} and n_{FN} are the numbers of MNs and FNs respectively [20]. Instead, in the solution presented by Yang et al. [59], each FN is preloaded with a pair of public/private keys and $n_{FN} - 1$ distinct pairwise keys, while no key is pre-loaded in the MNs. The memory overhead of this scheme is $(2 + n_{FN} - 1) * n_{FN}$. According to the basic scheme [51], each node is loaded with q keys before deployment, thus resulting in a total memory overhead of $q*(n_{FN}+n_{MN})$.

In the scheme proposed in this paper, each MN is loaded with only 3 keys (i.e., SK, K_{plc} and Authentication Key) and each FN is loaded with 6 keys (i.e., the BS public key, its own public/private key pair, SKG, CNDK and K_{prt}). The resulting memory overhead is $6n_{FN} + 3n_{MN}$.

To analyze and compare the proposed scheme with the existing schemes [58], [51], [37], [59], [38], it is assumed that each FN is able to make a maximum of d connections with its neighboring MNs. According to [51], [37] and [38], if a node has N_c neighbors and that node has to establish secure links with only d neighbors, then the required key sharing probability should be

$$Pr = \frac{d}{N_c} \tag{4.5}$$

For example, the single key sharing probability required to make 30 connections with the neighboring MNs out of 38 neighbors is approximately 0.80. From fig. 4.5, each node in [51] should carry 400 keys and each FN in [37] should carry 700 keys while each MN should carry 228 keys while in [38] each FN should carry 250 keys and each MN should carry 30 keys. In our scheme, each FN should be loaded with only 6 secret keys.

Fig. 4.6 summarizes the performance offered by different solutions in terms of the total number of the keys deployed for different sizes of the WSN. The results show that the proposed scheme requires fewer keys compared to other approaches. For less dense networks, the proposed scheme and Yang's scheme require almost

the same number of keys. However, the proposed scheme performs better in dense networks.

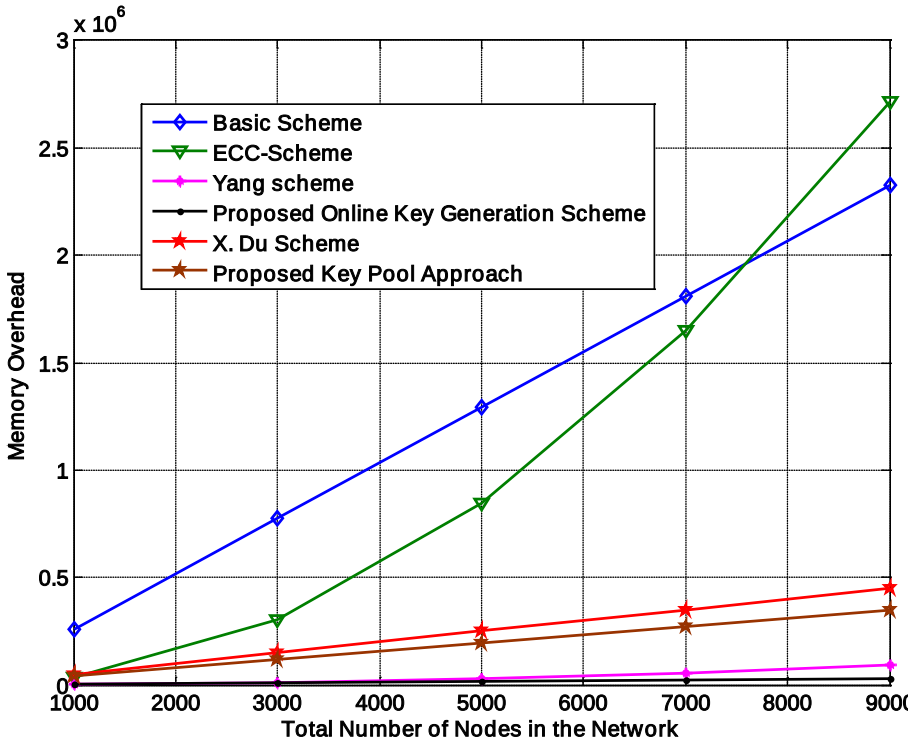

Figure 4.6. Comparison of memory overhead produce by the proposed scheme with some existing scheme

4.2.3 Network Resilience to Node Capturing Attacks

This section shows the effect of node compromised attacks on data communication capabilities. In the proposed scheme, FNs and MNs are provided with different security measures dealing with such attacks. Since FNs act as both CHs and data sinks for MNs, they are provided with tamper resistant hardware to protect their security material. Once the FN is captured, all security keys are replaced by a reference "compromised key" which does not allow the node to authenticate itself to the BS nor to accept any joining MN. On the contrary, MNs are not provided with the tamper resistant hardware. Node compromised attack in case of balanced and unbalanced key pre-distribution schemes for homogeneous and heterogeneous

sensor networks have a significant impact on the security offered by the communication links operating within the network due to the large number of shared keys with other nodes in the network. The fraction of communications compromised by compromising n MNs in shared key pre-distribution schemes is given by

$$Pr[Compromised] = 1 - \left(1 - \frac{K}{P}\right)^{n} \tag{4.6}$$

where K is the number of keys assigned to each MN from a pool of P keys. In case of compromised FNs, K is replaced by S in (6). Fig. 4.7 shows the OMNET++ simulation results about how many communications links a compromised MN can create with uncompromised MNs without involving the CH/FN. More specifically, the figure compares the proposed scheme with the schemes proposed in [51], [37], [38] and [32] with Pr[Conn]=0.8. The proposed scheme performs better because (i) a MN cannot establish directly a communication link with the other MNs of the network and (ii) all the FNs use the algorithm proposed in [23] to detect the compromised MNs. Since the FNs act as trusted servers to the MNs, their compromission can

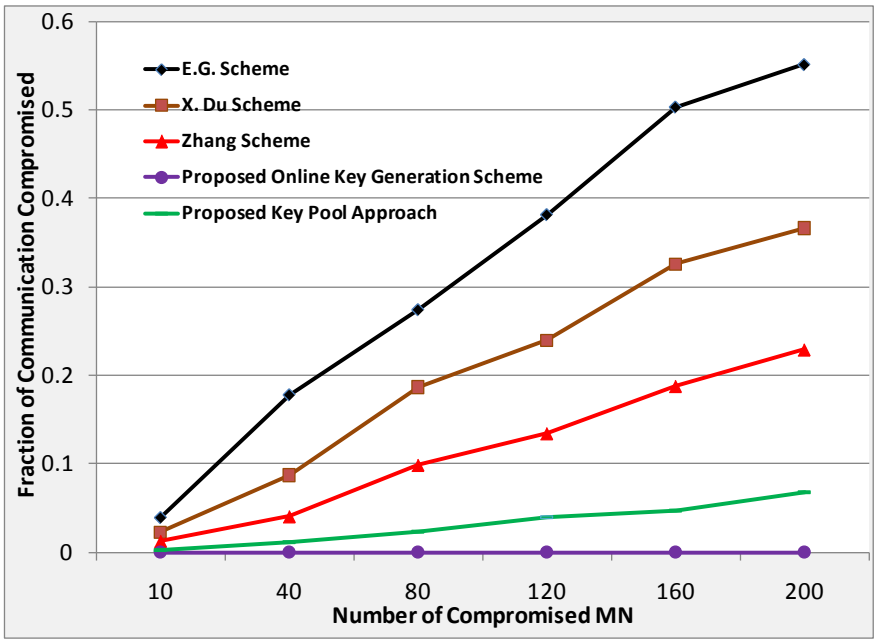

Figure 4.7. Fraction of communication compromised by capturing 'n' Mobile Nodes (MNs)

severely affect the network security. Fig. 4.8 shows a comparison of the OMNET++ simulation results for the FNs compromission of the proposed scheme with [51], [37], and [32]. It is clear from fig. 4.8 that FN compromission results in almost the same number of compromised links when using [51] and [37]. Although [32] proposed a balanced key distribution for the HSNs like [51] for homogeneous sensor networks, it performs better than [51] and [37] because it divides the key pool P into a number of groups equal to the number of clusters thus increasing not only network connectivity but also network resilience against both FN and MN capture attacks.

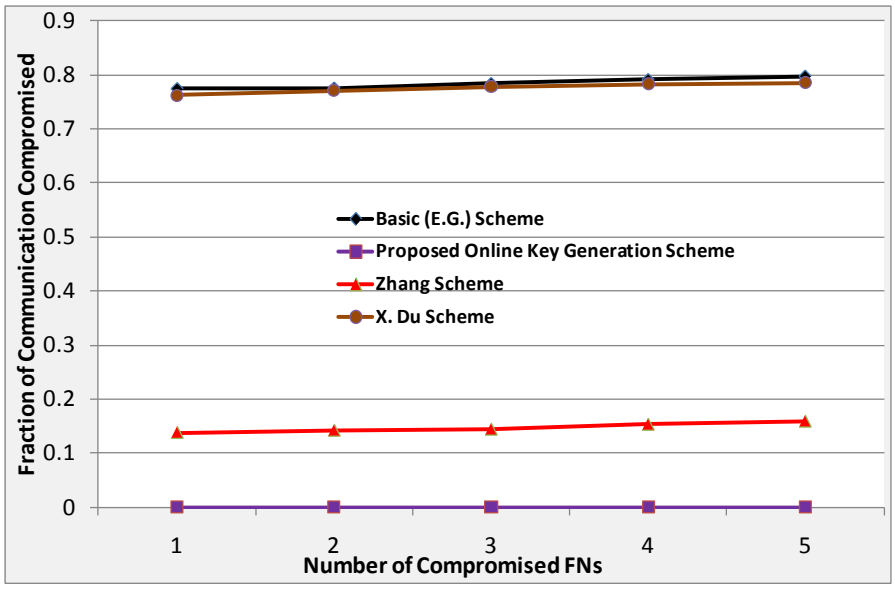

Figure 4.8. Fraction of communication compromised by capturing 'n' Fixed Nodes (FNs)

4.2.4 Communication Overhead

In this section, the communication overhead is evaluated also analyzing the different contributions from authentication and key establishment phases. The simulation scenario is modified in order to include 16 FNs and 500 MNs.

4.2.4.1 Authentication Overhead

Concerning the authentication overhead, the total number of packets exchanged during the authentication phase is considered. The authentication phase of the proposed solution is compared with some of the existing approaches [32], [33] and [20]. OMNET++ simulation results show that the proposed scheme produces less authentication overhead than the existing schemes, as shown in fig. 4.9.

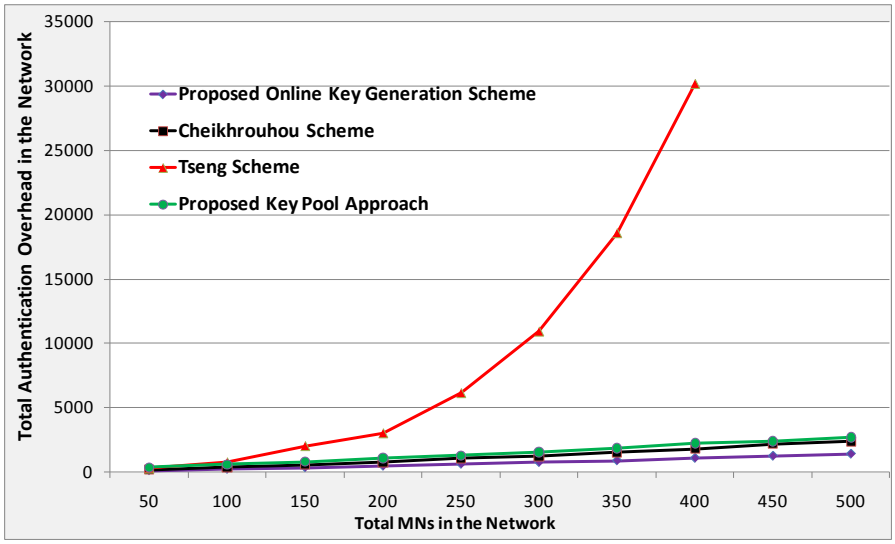

Figure 4.9. Authentication overhead comparison

4.2.4.2 Key Establishment Overhead

The proposed solution is also compared with the basic homogeneous [51] and heterogeneous [37], [20] schemes. The results show a significant reduction of the communication overhead. A 99% network connectivity probability for [51] and [37] was taken into account, computing the number of keys required in each FN and MN (using the results of (3) and (4)). The obtained results are shown in fig. 4.10. There is only a slight difference in terms of communication overhead between the homogeneous and heterogeneous approach, but there is a big difference in terms of memory cost (fig. 4.6).

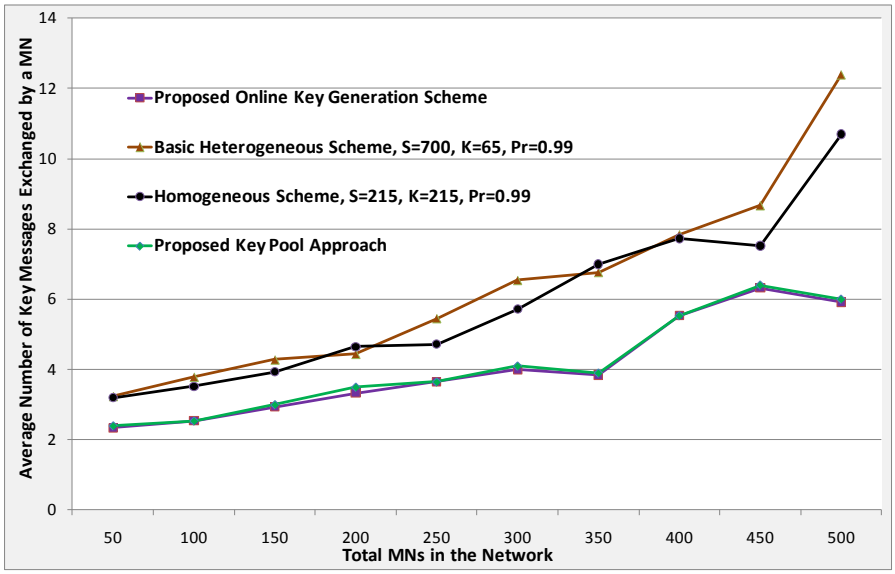

Figure 4.10. Average number of key messages exchanged during the first key establishment phase

4.2.4.3 Total Initialization phase Overhead

This section presents the OMNET++ simulation results for the total communication overhead generated during the first authentication and key establishment phase. The results of the proposed scheme have been compared with the ones related to [32] and [20], since both solutions are based on the mutual authentication and key establishment phases. Fig. 4.11 represents the resulting communication overhead by varying the size of the network.

4.2.5 Energy Consumption

This section describes the average energy consumption of each node during the authentication and initialization phases of the network (again using the OMNET++ simulator). The proposed solution requires only 2 messages for the authentications as shown in fig. 4.9 compared to [33] which requires 4 messages and with [32] and [20] which require 3 messages for authentication. The average energy consumption of each node during the authentication phase in the proposed scheme compared with [33], [32] and [20] is shown in fig 4.12. Fig. 4.11 also shows the effectiveness

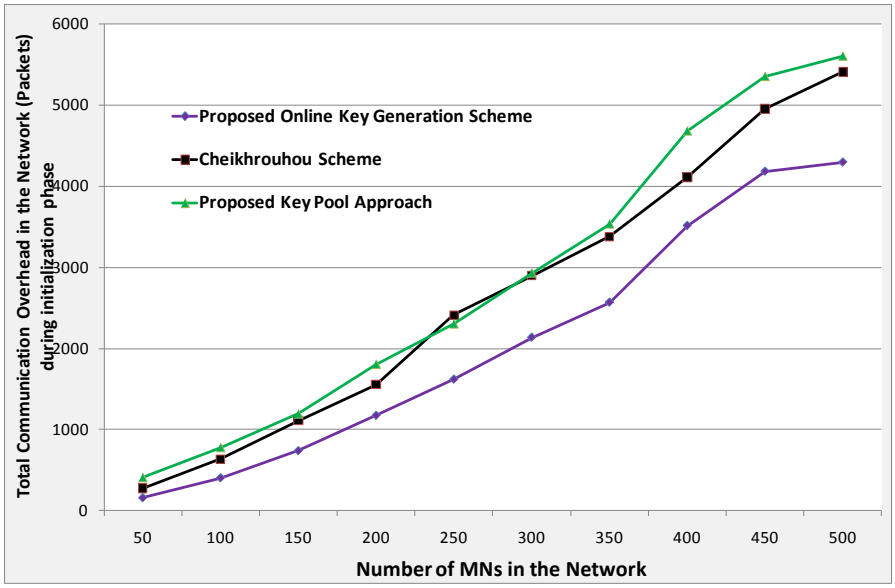

Figure 4.11. Total communication overhead in the network during the initialization phase

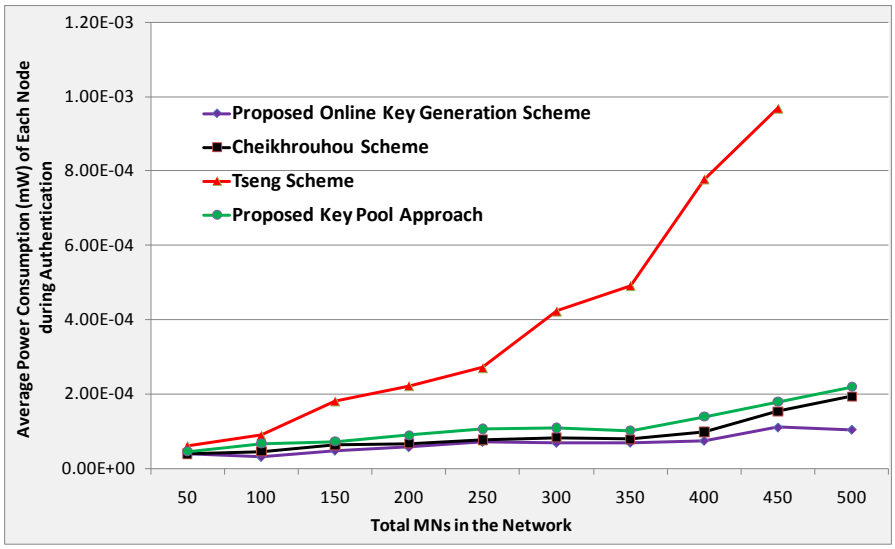

Figure 4.12. Total communication overhead in the network during the initialization phase

of combining the authentication and key establishment phases to reduce the total overhead during the initialization phase. Such optimization results in power savings

at each node and in an overall increase of the network life time. Fig. 4.13 represents the OMNET++ results for the average energy consumption of each node during the initialization phase (authentication and key establishment) in the proposed scheme, as compared with [33] and [20]. The results show that the proposed solution of combining the authentication and key establishment messages reduces the energy consumption with respect to [33], [20] where separate messages are exchanged for key establishment between the nodes after their successful authentication.

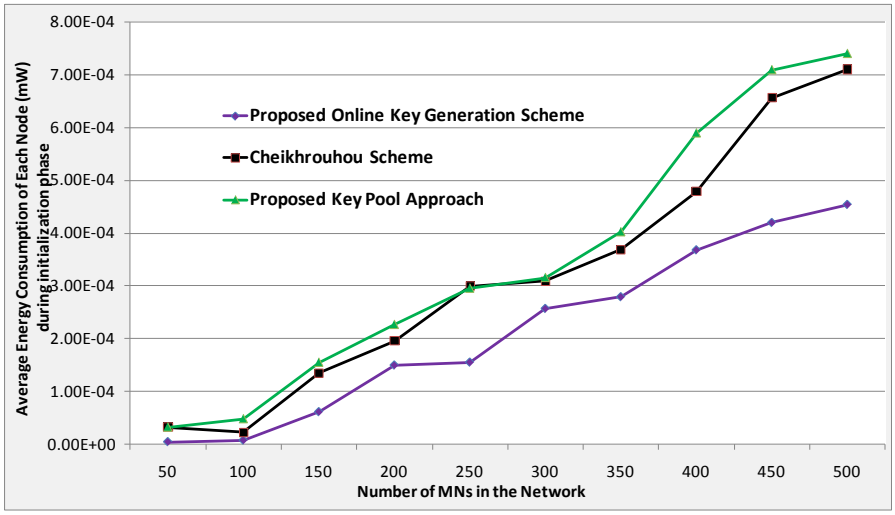

Figure 4.13. Total communication overhead in the network during the initialization phase

4.3 Security Analysis and Evaluation against Attacks

Since cryptography is considered as the main building block of any security primitive, the cryptographic keys should also be secured and authentic. To this aim, the key management scheme should be secure and each node of the network should be able to authenticate the cryptographic key(s). This is the most challenging problem in such resource constrained networks. In order to validate the secrecy of the proposed key management scheme for heterogeneous sensor networks, we used the AVISPA (Automated Validation of Internet Security Protocols and Applications)

tool [67]. AVISPA is a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques (e.g. OFMC, ATSE, etc). We implemented the proposed key management scheme in AVISPA and checked its security using some of the attacks provided by AVISPA, namely OFMC (On-the-Fly Model-Checker) and CL-AtSe (Constraint-Logic-based Attack Searcher). The former builds the infinite tree defined by the protocol analysis problem in a demand-driven way, i.e. on-the-fly and uses a number of symbolic techniques to represent the state-space. The latter provides a translation from any security protocol specification written as a transition relation into a set of constraints which can be effectively used to find attacks on protocols.

Technique	Summary
OFMC	SAFE
CL-AtSe	SAFE

Table 4.1. AVISPA Simulation Results

Both translation and checking are fully automatic and internally performed by CL-AtSe, i.e. no external tool is used. In this approach, each protocol step is modelled by constraints on the adversary knowledge. These results are shown in table 4.1. We evaluated the proposed scheme against some well known attacks DoS attacks, Node Replication attacks, Wormhole attacks, Black-hole attacks and Sybil attacks and showed how the proposed scheme performs in such attacking scenarios. We evaluate the proposed scheme against.

4.3.1 Denial of Service Attacks

The Denial-of-Service (DoS) attacks are used to degrade the performance of a network by exhausting its resources, such as bandwidth, memory or processor time or by sending fake network topology and routing information. The possible DoS attacks in the proposed scheme might be during (1) the cluster formation phase (2) the MN transition phase from one cluster to another (3) the addition of new nodes in the network.

During the cluster formation phase, all FNs periodically broadcast Hello messages for a specific number of times (3-times in the proposed scheme) and each Hello message is encrypted by the network private key. If an intruder broadcasts its own Hello messages, those messages will not be decrypted by the network public key and would be discarded by the MNs of the network. Also the MNs check how many hello messages they received from a specific FN. If those messages are above a predefined threshold, the MNs consider that FN as a adversary or malicious node of the network. We implemented the DoS attack in the OMNET++ simulator to check the performance of the proposed scheme against the replay attack of the FNs Hello messages by introducing a different number of attacking nodes. In the evaluated DoS attack, the attacker captures the Hello messages of the FNs and forwards it to the MNs by changing the source node ID in the packet. We analyze how many MNs the attacker can isolate from the network by forwarding the captured Hello messages before the MNs receive a Hello from the authentic FNs. Table 4.2 shows the results of the simulation with 200 MNs and 16 FNs and different number of attacking nodes.

Total number of attacking nodes	Total Hello messages sent by the attacker	Total MNs receive Hello messages from attacker
1	25	10
2	110	13
3	132	20

Table 4.2. DoS attack evaluation

Note that a total isolation of the MNs from the network by sending the captured and modified Hello message is not possible, because if the MNs send a response to the attacking node, the attacker would not be able to decrypt it and then the MNs would start to broadcast their own Hello messages to know about their neighboring FNs to join the network.

During the MNs transition from one cluster to another cluster, each MN broadcasts Hello messages to know about its neighboring FNs. Since, in the proposed scheme, these Hello messages also include the NAC (Network Authentication Code) which is used by the FNs of the network to authenticate the incoming MN, the

adversary would not be able to send the correct NAC to the FNs and would be detected at its first broadcast. The adversary can also add a fake node in the network, to try to get access to the network by broadcasting a Hello message to know about its neighboring FNs. But since in the proposed scheme the BS informs the FNs about the addition of a new MN, its ID and a specific authentication code assigned to that MN, the adversary fake node would not be able to authenticate itself to the FNs and would be detected at its first broadcast. Thus the proposed scheme effectively avoids these three different types of DoS attacks that could be launched by an adversary at any stage of the network and could exhaust the resources of both the FNs and the MNs.

4.3.2 Node Replication Attacks

The MNs are more vulnerable than the FNs and can be easily captured, analyzed and replicated by the attacker in various positions of the network. Such attacks may allow the adversary to corrupt data and may disconnect a significant part of the network. Node replication attack might be possible (1) during the network initialization phase and (2) after the network initialization phase. However node replication in the network initialization phase is difficult for an attacker because of the secure deployment phase. The attacker can launch the node replication attack after the network initialization phase when the network will no longer be under observation by the network deployer. Since in the proposed scheme the MNs communicate directly with their selected FNs/CHs, each FN sends its member MN IDs to the BS after the initialization phase. Also, during the transition phase of a MN from one cluster to another cluster, the new CH verifies the transition of the incoming MN from its previous CH. Thus node replication in the network is immediately detected by the BS during the initialization phase or by the FNs during the handover phase of the MNs. Thus the proposed scheme avoids node replication attacks in the network.

4.3.3 Wormhole Attacks

In the wormhole attack, an adversary launches two nodes in two different clusters, connecting them using a direct communication link called wormhole link. This link could be an Ethernet cable, long range wireless transmission, or an optical link. The

main purpose of this attack is to capture the traffic of one part of the network and replay it in the other part of the network. Also this attack is easily implemented in multi hop networks. This attack can be launched against the proposed scheme during the initialization phase by replaying the Hello messages of one part of the FNs of the network into another part of the network, to attract the MN communications. However in this type of attack, the attacker acts as a man-in-the-middle and just forward the packets from one part of the network to the other but would not be able to understand or extract the key/data information from the received packets. The verification of this man-in-the-middle attack was performed using the AVISPA tool which verified the security of the proposed scheme as shown in table 4.1. Note that we use a single hop network topology approach in which each MN is only one hop away from the FNs. The FNs send their member MN lists to the BS and the BS knows the location and position of each cluster of the network, so this wormhole attack, if launched after the network initialization phase, can be easily detected and avoided in the network by the BS. The adversary node in one cluster cannot pretend to be a member node of another cluster (even of neighboring cluster) despite having updated information received by the BS from the FNs.

4.3.4 Sybil Attacks

In the Sybil attack, a malicious/attacker node assumes multiple identities to launch attacks against storage space of its neighboring node or some protocol specific attacks (e.g., routing algorithms). This attack is reactively successful against key predistribution mechanisms, but since the proposed scheme uses an online authentication key and secret key generation technique which involves the node ID and its unique prime number, it makes it difficult for an attacker to launch sybil attack. For example, in other key predistribution approaches, if an attacker compromises a few nodes and obtains a few authentic keys of the network, it can launch sybil nodes with different IDs and can assign them those compromised keys. Now when the verification and authentication process starts for some authentic nodes of the network, the sybil nodes can give them a proof of authenticity by sending their own key pool IDs along with the compromised key IDs. If the verifier node and the sybil node have a common key among their key pool (i.e. the key ID and the actual key are

the same), the verifier node would not be able to detect the sybil node. Otherwise if the key IDs match but the keys do not match, then the verifier node can detect such sybil nodes in the verification process. But in the proposed scheme there is no concept of initial secret key predistribution and all the predistributed keys are the funtion of the node IDs and their assigned secret prime number. Hence node compromission does not help the attacker to launch sybil nodes with fake IDs.

In the key pre-distribution approach [75], if every node is assigned 'k' keys from a key pool of size 'm' and 'd' verifiers are used to verify a node, and if an attacker compromises 'c' nodes to create a compromised key pool of size 'n', then the probability of a sybil node to be successful is

$$Probability(Successful\ Sybil\ Nodes) = \sum_{t=1}^{k} \frac{\binom{n}{t}\binom{m-n}{k-t}}{\binom{m}{k}} \left(\frac{\binom{m-k+t}{k}}{\binom{m}{k}} \right)^{d} \tag{4.7}$$

Fig. 4.14 compares the key pre-distribution approaches with the proposed online key generation approach based on some pre-distributed key materials. The proposed solution shows better results because of the key generation algorithms assigned to the FNs and the establishment of keys between the MNs through their FNs/CHs.

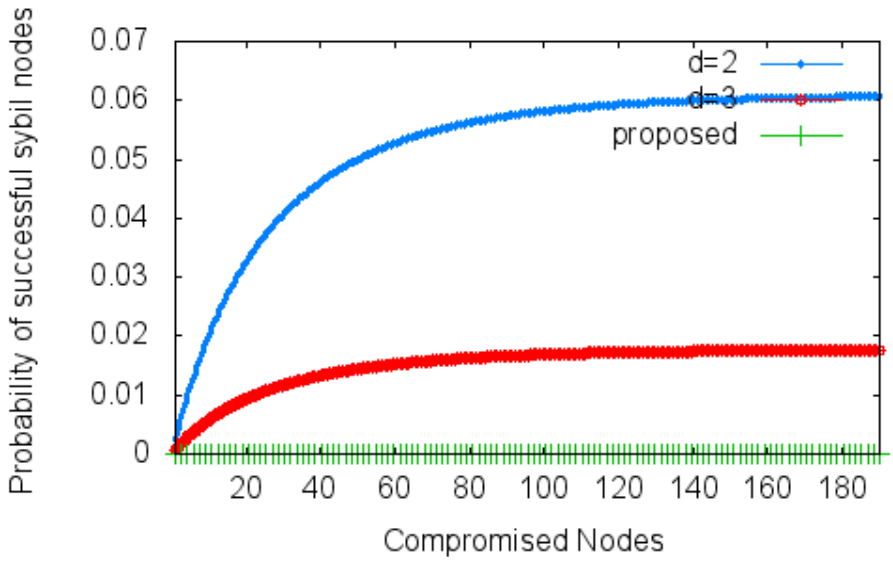

Figure 4.14. Probability of successfully generated sybil nodes

Key Generation for IP-Based Wireless Sensor Networks (6LoWPAN)

Chapter 5

Key Generation for IP-Based Wireless Sensor Networks (6LoWPAN)

Integration between wireless sensor networks and traditional IP networks using the IPv6 and 6LoWPAN standards is a very active research and application area. A combination of hybrid network significantly increases the complexity of addressing connectivity and fault tolerance problems in a highly heterogeneous environment, including for example different packet sizes in different networks. In such challenging conditions, securing the communication between nodes with very diverse computational, memory and energy storage resources is at the same time an essential requirement and a very complex issue. In this chapter we present an efficient and secure mutual authentication and key establishment protocol based on Elliptic Curve Cryptography (ECC) by which different classes of nodes, with very different capabilities, can authenticate each other and establish a secret key for secure communication. The analysis of the proposed scheme shows that it provides good network connectivity and resilience against some well known attacks.

5.1 Overview

Recent research activities in the field of LoWPANs aim to integrate sensors and actuators into traditional IP networks using IPv6 over LoWPAN (6LoWPAN)[70]. Smart objects belonging to a 6LoWPAN can directly communicate with an IPv6 host, thus allowing data processing operations to be performed in standard servers. 6LoWPAN actually enables the integration of smart objects into the overall Internet, toward the definition of the Internet of Things (IoT). In such resulting scenario, the presence of billions of objects raises additional issues in terms of scalability, manageability, addressing, security, privacy, secure mobility and robustness. Therefore an efficient redesign of the Internet architecture and the definition of new protocols are required to cope with the above challenges in the future Internet. In fact, several projects from industrial and international collaboration are being carried out to define the future internet architecture which would solve the limitations of the current architecture [71] including security, mobility and interoperability for the heterogeneity of networks.

In this context, mobility support for small and smart devices is one of the main important issues since it is utilized for realizing many innovative applications. Mobile communication may increase fault tolerance capabilities of a network but it requires continuous connectivity among the nodes in the network or in its clusters and could also introduce new threats against the privacy, integrity and confidentiality. Here, we specifically focus on authentication and on securing the communication between nodes in the real deployment of 6LoWPANs.

A number of cryptographic mechanisms have been introduced in the literature for secure authentication and encryption in WSNs such as block ciphers as part of standards-based protocols, including IEEE 802.15.4, different variants of symmetric and asymmetric cryptography. While these mechanisms are optimized to suit the resource constrained sensor and actuator networks, in case of 6LoWPAN networks, where the networks are integrated into the internet, such cryptographic mechanisms can still experience poor performance due to the size of the packets exchanged and the length of the keys. Furthermore, it is difficult to distribute the security keys in the federated combination of networks. Thus these mechanisms need to be modified to suite the resulting IoT scenario.

Raza in [72] represented a secure End-to-End (E2E) communication protocol between the IP enabled sensor networks and the traditional internet using the compressed and light weight design implementation of IPSec. Their performance evaluation in terms of code size, packet overhead, and communication performance shows that their proposed scheme has a comparable overhead to the generally deployed 802.15.4 link layer security while it offers a true E2E security. Jara in [70],[73] proposed a secure mobility management scheme for the 6LoWPAN based on the ID/Locator split architecture and on the extension of the Return Routability with Diffie-Hellman key agreement and ECC. Their proposed solutions deal efficiently with the DoS attacks and flooding attacks against the ID/location update messages, home registration and binding transfer process. They also verified and evaluated the schemes successfully with the AVISPA tool.

5.2 Proposed Algorithm

Here we describe the proposed authentication and key establishment phases for IP-enabled wireless sensor and actuator networks based on 6LoWPAN. Since the total number of hosts in a network could vary a lot and and the network might also contain thousands of nodes, the use of key pre-distribution techniques could not represent the most proper solution. In fact, such mechanisms require large memory space that could be not available in resource constrained smart objects. In addition, the nodes are expected to move and may leave their current network and enter into a foreign network.

Therefore, we introduce a new approach based on the Elliptic Curve Cryptography (ECC) that supports a higher security level compared to the standard encryption techniques (RSA, AES), while using shorter key length and introducing less computational overhead. In this approach, the joining network easily authenticates the incoming node by generating its authentication key instead of getting the node's authentication key from node's previous network in order to reduce the total communication overhead during the authentication of the incoming mobile node, also avoiding the introduction of new vulnerabilities.

The reference network model considers two sub networks connected with each other through edge routers as shown in fig. 5.1. In addition, a specific node in the

network acts a reference for the different supported security functionalities and it is called Network Security Manager.

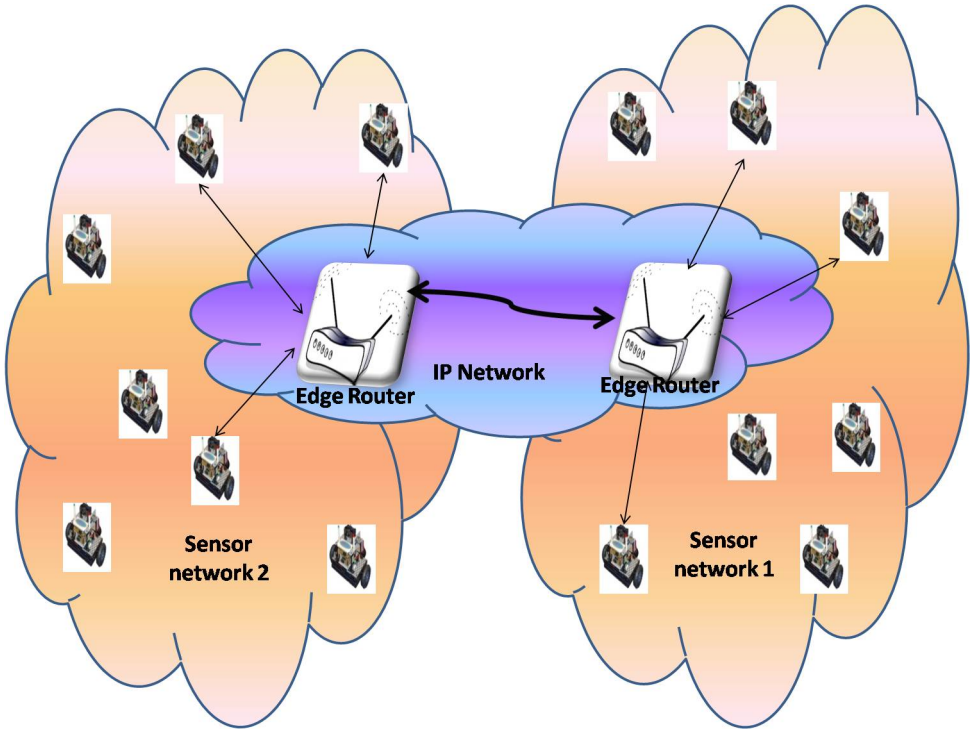

Figure 5.1. Virtual Network Architecture

5.2.1 Offline Key Assignment

Important key materials are assigned offline to each node and are used to authenticate each other by generating the authentication key and to secure the communication link by generating a public/private key pair for encryption and decryption of the messages. More specifically:

- To each entity of the network a random number is assigned by the Network Security Manager after a node registration phase
- To each entity of the network one share of the public key is also assigned, while the other share of the public key would be generated by the relevant local Network Security Manager

- When considering the secure communication between two nodes in the network, source IP and destination IP are used to generate a specific elliptic curve (adopted just for the pair of nodes taken into account)

- Each entity and a network has its own generator G_e and G_n respectively

5.2.2 Authentication

Authentication is an important and initial step in the network security that allows a trusted node to access the network resources and establishes secure links with other nodes of the network while it prevents an adversary to gain access to those resources and exploit possible vulnerabilities. Here, we describe how two nodes belonging to different networks can authenticate with each other.

Due to large number of entities (smart objects and IP hosts) in the network, it would not be a feasible solution to provide an authentication key(s) to an entity especially when it belongs to a resource constrained network (i.e. sensor network). In fact, (1) nodes usually have limited memory space and cannot store a large number of keys for secure communication with a very large number of entities of all networks (2) nodes are inherently prone to security attacks e.g., sensor nodes can easily be captured and their stored keys reused (3) partial distribution of keys reduces the network connectivity in the considered federated large networks. Hence we use an online key generation approach based on the ECC to reduce the memory consumption and avoid the key revocation/renewal in case of node capturing attack.

Every node is provided with a prime number 'p' that would be used to generate the private key for a particular destination node. It is worth stressing that each node would have one public key for all the destination nodes belonging to different networks. This public key consists of two shares (1) node share and (2) network share. The purpose of the network share is just to confirm that the node belongs to the mentioned network. The destination node needs to generate the public key of a source node by getting those shares from the source node and from the Network Security Manager of the source node as

$$PublicKey = f\{Node\ Share,\ \ Network\ Share\} \tag{5.1}$$

For example, a Sensor Node (SN_1) of one network wants to establish a communication link with a sensor node (SN_2) of other network. During the registration phase of a SN with its Network Security Manager, it sends the *node share* of its public key to its Network Security Manager. The Network Security Manager generates the *network share* of the registering SN public key and also sends the *network share* along with a random number to the SN. When a SN_1 wants to communicate with the SN_2, it asks its Network Security Manager to get the random number and *network share* of SN_2 from the SN_2 Network Security Manager which is assigned by the SN_2 Network Security Manager to the SN_2 during the registration phase. The SN_1 Network Security Manager gets that random number and *network share* from SN_2 Network Security Manager using its secure link, already established, and also sends the random number of the SN_1 and the *network share* of the SN_1 public key. The SN_1 creates a private key for the SN_2 after getting that random number and sends its own generated public key share to the SN_2 in the joining request signed by its private key. When the SN_2 receive this message, it contacts it local Network Security Manager for the *network share* of the public key and a random number of the SN_1. The SN_2 Network Security Manager forwards the *network share* of SN_1 public key and its random number to the SN_2 which it receives during the random number exchange. Once the SN_2 receives the *network share*, it generates the public key of the SN_1 and authenticates the message signature. The *node share* and the *network share* are generated as follow

$$NodeShare = S = IP_{Network} \; . \; c \; . \; G_{SN} \quad mod \quad P_{SN} \tag{5.2}$$

$$NetworkShare = T = S \; . \; G_N \quad mod \quad P_N \tag{5.3}$$

$$PublicKey = Node\ Share \quad \oplus \quad Network\ Share \quad mod \quad P \tag{5.4}$$

Where (P_{SN}, G_{SN}) and (P_N, G_N) are the pair of prime number and group generator of the network entity and the Network Security Manager respectively, c is the point on elliptic curve and 'P' is the prime field generator.

After successful authentication, SN_2 accept the join request of the SN_1 and

generate a private key for that SN_1 using the same procedure and sends its own share of public key to the SN_1 signed by its private key. The SN_1 generates the public key of the SN_2 by getting the *network share* from the SN_2 Network Security Manager through its own Network Security Manager and verifies the signature. In this way, both SN_1 and SN_2 authenticates each other.

5.2.3 Private Key Generation

Since every node generates and uses a separate private key for authentication and secure communication with the nodes of other networks for its public key, here, we describe the procedure of generating a private key. Since all nodes of every network are registered with their Network Security Manager , the Network Security Manager assigns a unique random number to each registered member. That number is used to generate a private key by other nodes to communicate with that particular node of that network. After the generation of public key by the destination node, it performs the XOR of the public key with the provided random number. The source node also gets that number from the Network Security Manager of the destination node through its own Network Security Manager. The source node uses that number to generate the private for the destination node.

$$Private\ Key = (Public\ Key\ \oplus\ Random\ Number)^{-1}\quad mod\quad P_{SN} \tag{5.5}$$

5.2.4 Handover

Since SNs are mobile, they may leave one network and enter into another one. To avoid node replication and Sybil attacks, it is necessary to update the node's public key and private key. To do so, the public key and the private key should be a function of some network parameters and also do not introduce a large communication overhead in case of fast mobility. In addition, the public key of the mobile node should be updated in such a way that there is no interruption in the ongoing communication. Finally, the private key should be updated accordingly.

Here we describe the proposed approach for updating the keys by considering a simple scenario that can implement the suggested idea easily without introducing

communication overhead. Assume that a sensor node SN_1 belongs to its parent network X and after some time it moves to network Y. It sends its *node share* to the Network Security Manager of the network Y. Its public key in network X is

$$PublicKey_X = S \ + \ T_X \quad mod \quad P \tag{5.6}$$

When SN_1 moves to network Y, then its public key will be

$$PublicKey_Y = S \ + \ T_Y \quad mod \quad P \tag{5.7}$$

According to the suggested approach, to avoid any interruption in the ongoing communication,

$$PublicKey_X \ = \ PublicKey_Y \quad \Rightarrow \quad S \ + \ T_X \ = \ S \ + \ T_Y \quad mod \quad P \tag{5.8}$$

Since the Network Security Manager Y receives the *node share* of the joining node SN_1, it generates the *network share* T_Y and sends back to the joining node.

$$T_Y \ = \ S \ . \ G_Y \quad mod \quad P_Y \tag{5.9}$$

Since, T_Y is different from T_X because of different network generators, the node will generate a number 'd' such that it makes the T_Y equals to T_X as

$$d \ . \ T_X = T_Y \quad mod \quad P_{SN} \quad \Rightarrow \quad d \ = \ T_Y \ . \ T_X^{-1} \quad mod \quad P_{SN} \tag{5.10}$$

The generated 'd' is sent to the destination node encrypted with its previous private key: in this way, the destination node can update the public key of the source node. After that, the source node also updates its private key.

5.3 Performance Evaluation

In order to evaluate the proposed scheme in terms of total time consumed during the exchange of key establishment packets, we used the cooja simulator [74]. The simulation environment consists of two sub networks as shown in fig.1 and we vary

the total number of nodes in each sub network. Initially, each sub network consists of 4 nodes and then we increased them to 8, 12 and 20. More specifically, we consider four sensor nodes in one network aiming to establish a secure key with the four nodes of the other sensor network by varying the total number of nodes in each network. The results shown in fig. 5.2 describe the total time consumed during the first key establishment process with the node of other network.

Here, we describe some well known attacks that could be possible in the IP based wireless sensor networks and show that how the proposed scheme deals with those attacks.

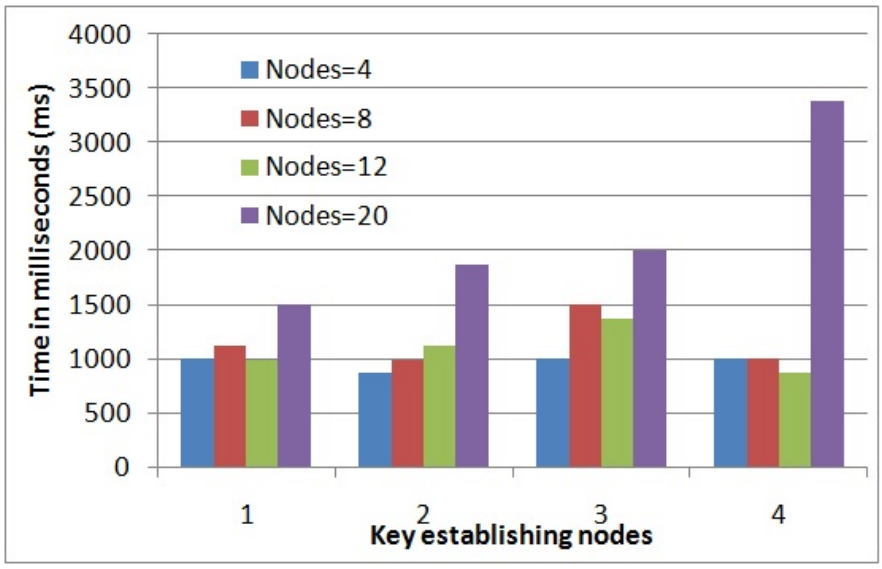

Figure 5.2. Time taken by a node during the exchange of key establishment messages in ms

5.3.1 Connectivity

Connectivity is the fundamental aspect and plays an important role in the network performance. However, it suffers due to the security protocols implementation in terms of partial key distribution and management, especially in a resource constrained wireless sensor networks where the connectivity depends on the key matching probability and hence there is always a trade off between the memory, connectivity and computational cost. Since, the proposed scheme is based on the

ECC in which each node has public/Private key pair for secure communication and also in the proposed scheme, if a node wants to establish a communication link with other nodes of the network, it just generates the public key of those nodes their public key shares from the Network Security Managers and there is no need for anything to be common among the nodes. So, the connectivity of the network in the proposed scheme is always 100% compared to the key pre-distribution schemes in which the connectivity is based on the common shared keys.

5.3.2 Sniffing

During the exchange of the pubic key materials (i.e. the node share and the network share), an adversary might capture those packets to get those shares. But, obtaining those shares does not allow the adversary to construct the node's public key because the construction of public key also requires the destination node random number assigned by its Network Security Manager during the registration phase. Thus an adversary cannot get that random number until the destination node gets compromised by the adversary. Since the nodes are mobile, their public key and the private key is also updated when they change the network (because of the network share of the public key). Hence the packets sniffing do not help the adversary to construct the node public key and exhaust its resources by sending some fake packets.

5.3.3 Stolen ID Attack

An adversary might steal the IDs of the authentic member of the network and capture the node share of the public key of that node and can pretend itself as an authentic node. In the proposed scheme, if the destination node receives the link establishment requests (containing the node share of the public key) from an adversary whose ID is the authentic node ID, the destination node generates the public key of the received authentic node ID by getting its random number from Network Security Manager but that key would not be able to verify the message signature because authentic node private key would not be compatible with the private key of the adversary node. In order to establish the link with the destination node, an adversary is also required to steal the private key of that authentic node as well. Also the adversary node would not be able to generate the public key of

the victim node because it cannot get the required random number of the victim node and the network share from the Network Security Manager which is used in the public key generation. And hence will be detected quickly in the network.

5.3.4 Denial of Service Attack

Denial of Service (DoS) attack is the one in which an adversary tries to isolate a node from a network and keeps it busy to exhaust its resources by sending some useless data. In the proposed solution, this is only possible if the adversary, somehow, gets an access to the network and become its authentic member or by stealing the nodes IDs and their shares of public key. The adversary would use those IDs and shares in establishing a link with the destination node of the other network. But this would not work because the adversary would need to sign that message again after updating its time stamp and by doing so, the destination node would not be able to verify that message with the generated public key. This helps the destination node to identify any malicious activity and inform its Network Security Manager which will inform the Network Security Manager of source node about those fake messages.

5.3.5 Node Replication Attack

Sensor nodes are very vulnerable and can be easily captured, analyze and replicate by the adversary in various positions in the network. Such attacks may allow the adversary to corrupt data and may disconnect significant parts of the network.

Since all the nodes in the sensor networks are mobile and their positions changes frequently depending on their speed and the size of network, their position should be updated by their Network Security Managers immediately in case of leaving or joining the network. Thus the adversary cannot replicate the nodes of one network into another network.

Key Generation for Content Centric Networks (CCNs)

Chapter 6

Key Generation for Content Centric Networks

Content centric networking concept was developed at PARC by Van Jacobson and his team [3], Content Centric Networking (CCN) is also known as Information Centric Networking or Named Data Networking [4]. Building on the observation that today's communications are more oriented towards content retrieval (web, P2P, etc.) than point-to-point communications (VoIP, IM, etc.), CCN proposes a radical revision of the Internet architecture switching from named hosts (TCP/IP protocols) to named data to best match its current usage. In a nutshell, content is addressable, routable, self-sufficient and authenticated, while locations no longer matter. Data is seen and identified directly by a routable name instead of a location (the address of the server) and is directly requested at the network level, not its provider. To improve content diffusion, CCN relies on close data storage because storage is proven cheaper than bandwidth: every content - particularly popular one - can be replicated and stored on any CCN node, even untrustworthy. People looking for particular content can securely retrieve it in a P2P-way from the best locations available.

6.1 Architecture of Key Management

The existing key management schemes for TCP/IP networks secure the links from source nodes to the destination nodes irrespective of the number and type of packets/data. Hence these schemes are ill suited for the CCNs architecture, where there is no concept of link between the requesting node and content generating node.

6.1.1 Design Principles

In the standard PKI approach, there is a certification authority and when the destination node receives the public key of the source node, it validates the received key using the certification authority. But this scheme is not suitable for content centric networks, where the keys are related directly to the contents, instead of the source ID or location. Also in the standard PKI approach, a node can use its encryption key (private key) to encrypt all the content and the destination node needs to verify the decryption key (public key) with the certification authority to check the authenticity and integrity of all the received contents. But in content centric networks, there is no concept of content source ID/location information, hence each received content key should be verified with the certification authority, which increases the overhead on the network and the time required to verify the key. Hence in this paper we propose to use the ideas of using: (1) distributed key holding nodes to reduce the communication overhead on a single node and (2) key shares to check the authenticity and integrity of the decryption key as well as of the received content.

In fact, since cryptography is considered as the main building block of any security primitive, the cryptographic keys should also be secured and authentic. To this aim, the key management scheme should be secure and each node of the network should be able to authenticate the cryptographic key(s). This is the most challenging problem in CCNs, where the keys are linked with the content names instead of the content generation source. Hence we have tried to solve the problem in our proposed key management scheme for the CCN networks which is not possible by the existing key management schemes for the traditional TCP/IP networks.

Thus, we propose an authentication and key establishment scheme for CCNs in which the contents are authenticated by the content generating node, using predistributed shares of encryption keys. The content requesting node can get those

shares from any node in the network, even from untrustworthy ones, in accordance with a key concept of CCNs. In our work we also provide means to protect the distributed shares from modification by these malicious/intruder nodes. Next, we describe the assumptions that we make on the architecture of Content Centric Networking, and then describe in detail the proposed key management scheme.

6.1.2 Network Architecture

Since the internet is a composition of a large number of small networks as shown in figure 6.1, we assume that each individual network has its own network manager, which are powerful secure nodes which act like servers for management and security-related aspects of networking.

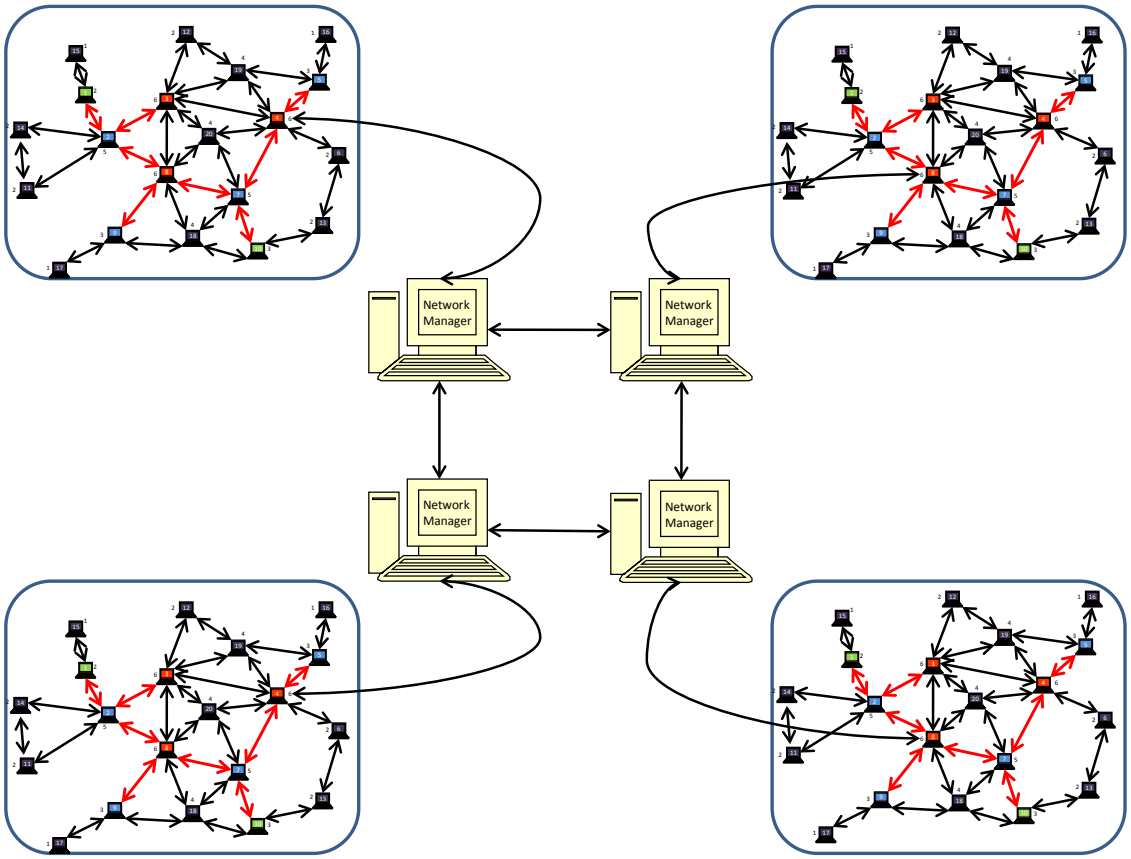

Figure 6.1. Virtual internet architecture

Each small network consists in a large number of nodes. We divide all the nodes

of each small network into two different categories, i.e. (1) Normal Nodes (NN) and (2) Key Holding Nodes (KHN). Both types of nodes are similar in terms of capabilities and architecture. The KHNs are responsible for initially holding the key materials of the encryption key(s) related to the content(s) once they are generated by any node (source) of the network after the network deployment.

The selection of KHNs is based on the maximum number of connections established by a node with its neighboring nodes. This approach minimizes the initial network traffic due to key management, since each node will be at a maximum of two hops from a KHN. Note that when nodes are deployed they are all assigned the same security-related material and hence can play both roles. The distinction between KHN and NN is made after deployment, when the node joins a network, and only affects the role that the node plays in providing and using the key material. In order to select KHNs, each node shares its connections count with its neighboring nodes. Once all the neighboring nodes receive those count, each node become aware of its neighboring node's and consider the neighboring node with the highest or equal connections as its nearer KHN. Figure 6.2 shows the virtual organization of KHN and NN in the network.

It should be clear from the figure that nodes 3, 4, 8 have the highest number of connections with their neighboring nodes, so they act as actual KHNs. On the other hand, nodes 2, 5, 7, 9 are not selected as KHNs, but can still act in this role for NNs that are two or more hops away from the actual KHNs, and so on. The red communication links show paths from the actual KHNs to the second level and third level KHNs which basically act as KHNs for the end nodes that are not connected directly to the actual KHNs ash shown with purple color.

Each node (both KHN and NN, since also the former can generate contents) is also assigned some key materials to generate their public/private key pair for securing the generated content. The assignment of those key materials to the nodes is performed off-line while the assignment of key materials related to the content(s) to the KHNs is performed on-line. The network manager also plays an important role in generating the key materials for its network nodes.

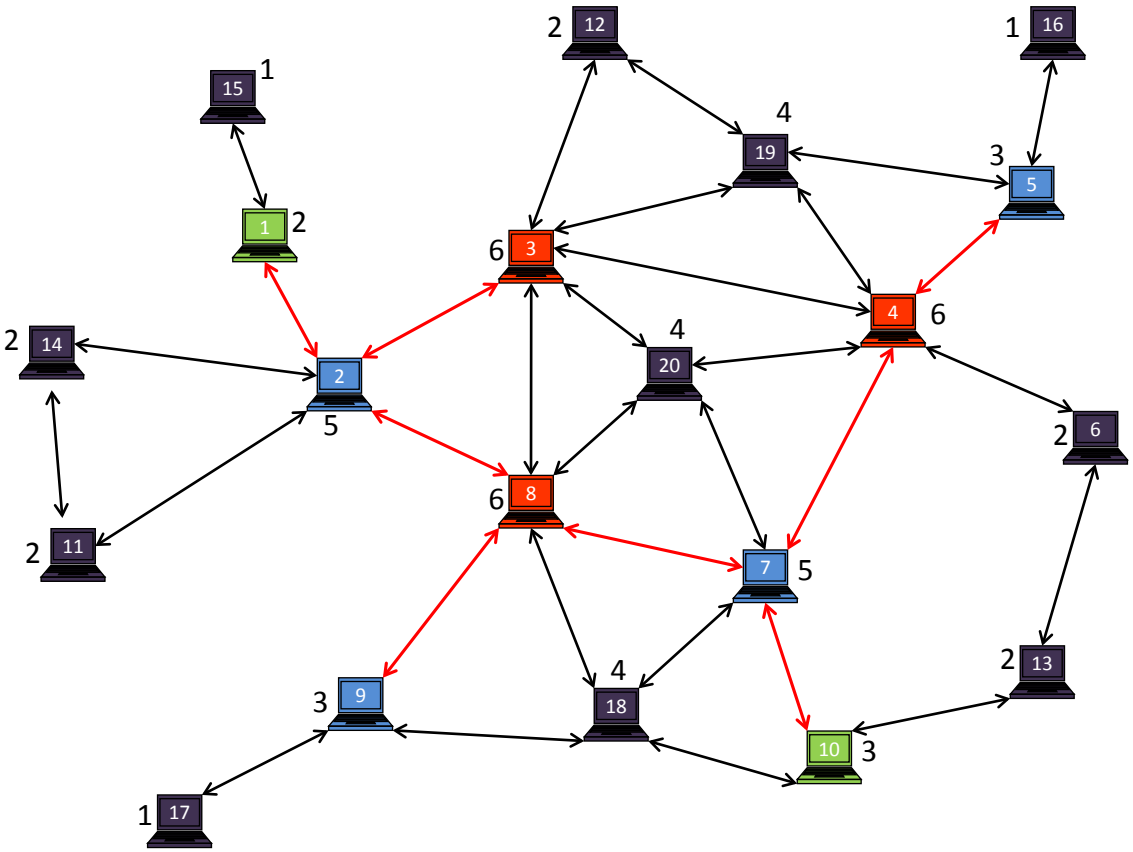

Figure 6.2. Virtual organization of Key Holding Nodes and Normal Node in the network

6.1.3 Key materials assignment

Each node in the network is assigned some important key materials which are used in generating the public/private key pair in order to secure the generated contents. More specifically,

- Each node in the network is assigned a random number generator, a one way Hash function (H), a share generation function (f) and a natural number group generator G (G can be, for example, a prime number).

- Each network manager is also assigned a fixed random number (NMRN) assigned by the network owner, a random number generator, a one way Hash function (H), a share generation function (f) and a group generator G.

6.1.4 Key Establishment and Management

Once the network is deployed, each node in the network sends a join message to its network manager. After the reception of those join messages, the network manager sends a fix random number (NDRN) and a NeTwork Public Share (NTPS) to each joining node. The network public share is generated as

$$NTPS = f(NMRN, node\ ID, RN) \tag{6.1}$$

Where RN is a random number from the generator. When a node receives the random number and NTPS from its network manager, it generate a NoDe Public Share (NDPS) of its public key as

$$NDPS = f(NDRN, RN, NTPS) \tag{6.2}$$

In CCNs, since contents are requested by their names instead of their generating source, there must be a relationship between the content and its encryption/validation key. Hence we introduce two further shares generated by the source node of the content in order to relate the encryption/validation key with the content. Those two shares are P_1 and P_2 which act as the two parts of the public key for a content. These two shares (P_1,P_2) are generated as

$$P_1 = f(NTPS + Content) \tag{6.3}$$

$$P_2 = f(NDPS + P_1) \tag{6.4}$$

The required public key k_{plc} is

$$k_{plc} = P_1 + P_2 \tag{6.5}$$

Since each node is given a group with a generator G, it selects a random number g from G and also creates the hash of the content-related public key shares and the

corresponding private key k_{prt} as

$$X = H(P_1), \quad Y = H(P_2), \quad Z = H(k_{plc}) \tag{6.6}$$

$$k_{prt} = K_{plc}^{-1} \; mod \; G \tag{6.7}$$

The node also calculates A, B and C for the authentication of the generated content and its shares as

$$A = g^X, \quad B = g^Y, \quad C = g^Z \tag{6.8}$$

After the generation of the public key shares and their hashes, the node distributes those shares among the nodes (KHNs) responsible for holding those shares i.e. $(P_1, P_2, C, Z, NDPS)$. The KHNs get the NTPS of the received NSPS from the network manager. The node includes A and B in the data packet along with the hash of content in order to help the destination node get and verify the public key shares i.e. $(Content, A, B, Z)$.

If now a node receives a data packet containing the content and hashes for the authentication, it needs the public key shares to verify those received hashes. In order to get the public key share from the KHNs, it sends a key share request to the KHN nodes. The KHN sends $(NDPS, NTPS, C)$ to the requesting node. After receiving this message, the node generates P_1 and P_2 using (3) and (4) and the share generation function f. After the generation of P_1 and P_2, the node generates X and Y using (6). Now the node calculates (C^X, C^Y) using the received C and calculated X and Y and then compares them with the (A^Z, B^Z) received in the data packet. Successful verification authenticates the received messages and the contained public key shares.

6.2 Performance Analysis

In this section, we describe the performance of our proposed scheme for content centric networks in terms of time taken by a node to retrieve a key. We compare it with the standard PKI approach for key establishment and management. To this aim, we use the ccnSim simulator [66] developed specifically for content centric

networks.

In the standard PKI approach, there is a certification authority and when the destination node receives the public key of the source node, it validates the received key using the certification authority. But this scheme is not suitable for content centric networks, where the keys are related directly with the contents, instead of the source ID or location. Also the in standard PKI approach, a node can use its encryption key (private key) to encrypt all the content and the destination node needs to verify the decryption key (public key) with the certification authority to check the authenticity and integrity of all the received contents. But in content centric networks, there is no concept of content source ID/location information, hence each received content key should be verified with the certification authority, which increases the overhead on the network and the time required to verify the key. Hence in this paper we propose to use the ideas of using: (1) distributed key holding nodes to reduce the communication overhead on a single node and (2) key shares to check the authenticity and integrity of the decryption key as well as of the received content.

6.2.1 Simulation scenarios

In order to evaluate the performance of the proposed scheme against the standard PKI approach, we use different network topologies provided in the ccnSim simulator and note the average time taken by a node to retrieve the key(s) for the received content(s). To do so, each node in the network generates a content which is composed of 100 files. Each file is encrypted by a separate key and the corresponding key shares are distributed among the Key Holding Nodes (KHNs). Also each file is split into five chunks. When a node start receiving the requested file after sending an interest for that file, it waits until all the chunks of the requested file arrive. Once the requested file is completely received, the node (requester) sends a request for the key shares of the received file. After the reception of those key shares from the nearest KHN (all others will be discarded, according to the CCN principle), the requester verifies the authenticity and integrity of the received file. Table 6.1 shows the average time taken by a node in different network topologies to retrieve a key for the received content.

6.2.2 Results

During the simulation, we select one node as a key holding node for the standard PKI approach and three nodes as key holding nodes for the proposed scheme.

Scheme	Geant topology (s)	Level3 topology (s)	Tiger topology (s)	dtelecom topology (s)
PKI	0.009	0.020	0.0003	0.0133
Our	0.004	0.018	0.0002	0.0131

Table 6.1. Average time taken by a nodes to retrieve a key for a content in different network topologies

Topology	PKI Approach	Our Approach
Geant	0.92	0.93
Level3	0.88	0.91
Tiger	0.96	0.97
dtelecom	0.93	0.94

Table 6.2. Average Hitrate of each nodes in different network topologies

We compared the *Hit rate* of the proposed scheme with that of the standard PKI approach. The *Hit rate* is the ratio between (1) the number of key interests received by a node for which the node has keys stored in its memory and (2) the total number of key interests received by that node.

$$Hitrate = \frac{Hit}{Hit + Miss} \tag{6.9}$$

Where *Hit* counts how many times a node has a key for the received key interest packet and *Miss* counts how many times a node does not have a key for the received key interest. The ccnSim simulation results are shown in Table 6.2.

6.3 Security Analysis

Since, cryptography is considered as the main building block of any security primitive, the cryptographic keys should also be secured and authentic. To this aim, the key management scheme should be secure and each node of the network should be

able to authenticate the cryptographic key(s). This is the most challenging problem in CCNs, where the keys are linked with the content names instead of the content generation source. Hence we have tried to solve the problem in our proposed key management scheme for the CCN networks which is not possible by the existing key management schemes for the traditional TCP/IP networks.

Since we kept a relationship between the content and its encryption key using the NTPS and NDPS, only the authentic nodes of the network are able to encrypt the content(s) using the authentic key(s). An adversary would not be able to encrypt the content using the authentic key of the network until and unless it becomes an authentic member of the network (i.e. it receives the key material described above before deployment). On the other hand, by using the standard PKI approach for CCNs, an adversary can generate and encrypt fake content using its generated private key corresponding to an authentic public key. This is easy because in CCNs key request is based on the content name instead of the generating source.

Also we eliminated the need for a centralize certification authority for the verification of the encryption keys by using the key share concept and by securing the keys with the network and node parameters. So the adversary would not be able to generate a fake private key simply thanks to the non-existence of a single complete public key.

In order to validate the secrecy of the proposed key management scheme for content centric networks, we used the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool [67]. AVISPA is a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques (e.g. OFMC, ATSE, etc).

We implemented the proposed key management scheme in AVISPA and checked its security using some of the attacks provided by AVISPA, namely OFMC (On-the-Fly Model-Checker) and CL-AtSe (Constraint-Logic-based Attack Searcher). The former builds the infinite tree defined by the protocol analysis problem in a demand-driven way, i.e. on-the-fly and uses a number of symbolic techniques to represent the state-space. The latter provides a translation from any security protocol specification written as transition relation into a set of constraints which can be effectively

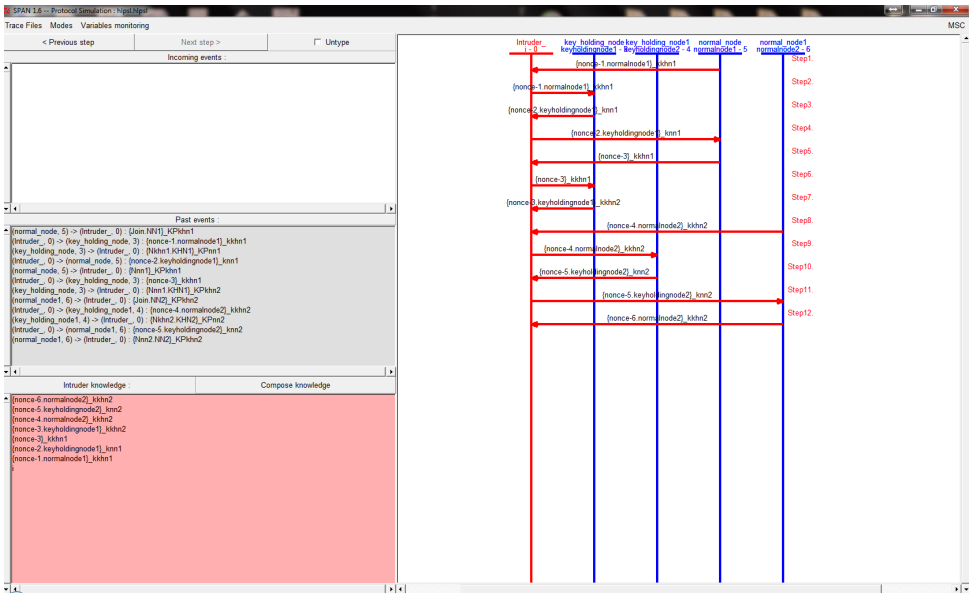

Figure 6.3. AVISPA tool screenshot

used to find attacks on protocols. Both translation and checking are fully automatic

Technique	Summary
OFMC	SAFE
CL-AtSe	SAFE

Table 6.3. AVISPA Simulation Results

and internally performed by CL-AtSe, i.e. no external tool is used. In this approach, each protocol step is modelled by constraints on the adversary knowledge. These results are shown in table 6.3.

Also figure 6.3 shows the interface of the AVISPA tool, illustrating a case in which the intruder acts as a man-in-the-middle. The intruder in this case acquires the key share but does not have the functions for generating the complete public key and the key shares ensuring authenticity, hence it cannot modify the actual content, whose integrity and authenticity are still ensured.

The proposed scheme hence follows the basic concept of content centric networks, which states that a node can get the required authentic keys and contents from any node of the network, even an intruder/attacker. This means that the key

management scheme must be so secure that the intermediate nodes/attackers cannot modify the contents/keys which are provided by our proposed scheme. Even if a node gets the required shares from the intruder who acts as a man-in-the-middle, a node is able to verify and authenticate those shares.

Conclusion

Chapter 7

Conclusions

In this section, I conclude my thesis by giving a brief overview over my work. First I worked on key management issues in ad hoc wireless sensor networks and IP-Based wireless sensor networks. Later I worked on key management issues in content centric networking.

The network model that I considered during my research work is based on the heterogeneous sensor networks model and on the cluster based approach. The network consists of at least two different types of nodes in terms of computational power, storage memory and lifetime. The more powerful nodes of the network act as cluster heads and are fixed while less powerful nodes act as cluster members and are mobile.

In the key pool approach discussed in chapter 3, the proposed solution is based on two disjoint key pools from which communication keys and authentication keys are generated. One key pool is used as an authentication key pool and the other key pool is used as a secret communication key generation key pool. Each FN and each MN is given a few keys from the authentication key pool using the unbalanced key pre-distribution approach. The FN and the MN use a common key from the authentication key pool to authenticate each other. Each FN is assigned two discrete key pools from the secret communication key generation key pool. The FN selects randomly a key from both randomly assigned discrete key pools and generates a secret key for communication with a MN. This generated secret key is transferred to the MN using the common authentication key by a FN.

The results of the proposed solution is compared with the basic scheme described in [51] in terms of memory cost and network resilience against node capture attacks. The results show that the two disjoint key pools provide a better level of security by consuming less memory compared to the basic scheme. It is also shown that the network connectivity in terms of authentication key sharing probability increases when a MN is in the radio coverage range of more than one FN. This result strongly supports node mobility in WSNs. Also the node compromised attack on the MN was discussed and compared with basic scheme. The results show that the proposed scheme provides better resilience against the node capture attack in terms of less probability of compromised communication and the node replication attack.

To further improve network connectivity and reduce memory cost, we proposed an on-line key generation approach. In the on-line key generation approach, each node is assigned certain parameters and key materials through which two communicating nodes generate a mutual key for secure communication with each other. To analyse the effectiveness of the proposed algorithm, we used OMNET++ simulator. Also the proposed network topology was cluster based. In comparison with existing approaches, the simulation results of the proposed solution provide better network connectivity, reduces memory overhead, increases network resilience against node capture attacks and requires minimum communication overhead during the authentication and key establishment phases. Hence it saves battery energy and increases the network life time. In this paper, only intra network movements of the mobile nodes were considered.

Since the integration of the IP networks with the sensor networks is an interesting research area so we tried to investigate a suitable key management scheme that suits two different types of networks. Hence, an authentication and mutual key establishment scheme is proposed for IP based wireless sensor network (6LoWPAN). The key construction is based on the elliptic curve cryptography approach and the key itself consists of two shares, node share and network share. A node need to obtain both shares of the communicating node to construct secret key using elliptic curve key generation approach for secure communication. Also to simulate the proposed scheme, we used Cooja simulator of the Contiki OS which provides the integration of sensor networks to the IP networks through gateways. This scheme has good theoretical results against some well known attacks and also take less time

to exchange the key establishment packets.

We presented in last chapter of this thesis, a key management scheme for Content Centric Networking. As this new networking paradigm heavily rely on content authentication to create a new web of trust, a scalable and secure key management scheme is mandatory for the further development of CCN. We addressed these two constraints when designing our distributed key management system: (1) we introduced key holding nodes to reduce the communication overhead used for authentication and (2) key shares to check the authenticity and integrity of the decryption key as well as of the received content. Performance simulation on ccnSim showed that our scheme is at least as good as a central authority while its distributed nature make it more scalable. Thus, the security analysis performed thanks to avispa tends to prove that no security leak exists.

Our future work will consist in the implementation of our key management scheme within the CCNx framework and of the emulation of a CCN network to check our solution against real CCN communications.

Bibliography

[1] Raymond, D.R., Marchany, R.C., Brownfield, M.I., Midkiff, S.F.: Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols. IEEE Transactions on Vehicular Technology 58(1), 367380 (2009) 10, 11

[2] Xu, W., Trappe, W., Zhang, Y., Wood, T.: The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2005), pp. 4657. Urbana-Champaign, USA (2005) 11

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, Networking named content, in Proceedings of the 5th international conference on Emerging networking experiments and technologies, ser. CoNEXT 09. New York, NY, USA: ACM, 2009, pp. 112. 22, 23, 26, 91

[4] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, Named Data Networking (NDN) Project, October 2010. 22, 91

[5] Law, Y.W., Palaniswami, M., Van Hoesel, L., Doumen, J., Hartel, P., Havinga, P.: Energy-Efficient Link-Layer Jamming Attacks against Wireless Sensor Network MAC Protocols. ACM Transactions on Sensor Networks 5(1), 6:16:38 (2009) 11

[6] Raymond, D.R., Midkiff, S.F.: Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses. IEEE Pervasive Computing 7(1), 7481 (2008) 11

[7] Pongaliur, K., Abraham, Z., Liu, A.X., Xiao, L., Kempel, L.: Securing Sensor Nodes Against Side Channel Attacks. In: Proceedings of the 11th IEEE High Assurance Systems Engineering Symposium (HASE 2008), Nanjing, China, December 2008, pp. 353361 (2008) 13

[8] Becher, A., Benenson, Z., Dornseif, M.: Tampering with motes: Real-world physical attacks on wireless sensor networks. In: Clark, J.A., Paige, R.F., Polack, F.A.C., Brooke, P.J. (eds.) SPC 2006. LNCS, vol. 3934, pp. 104118. Springer, Heidelberg(2006) 11

[9] Goodspeed, T.: Wireless Sensor Networks as an Asset and a Liability. In: Proceedings of the SOURCE Conference, Boston, USA (March 2009) 12

[10] Francillon, A., Castelluccia, C.: Code Injection Attacks on Harvard-Architecture Devices. In: Proceedings of the 15th ACM conference on Computer and communications security (CCS 2008), Alexandria, USA, October 2008, pp. 1526 (2008) 12

[11] Newsome, J., Shi, E., Song, D., Perrig, A.: The Sybil Attack in Sensor Networks: Analysis and Defenses. In: Proceedings of the IEEE 3rd International Workshop on Information Processing in Sensor Networks (IPSN 2004), Berkeley, USA, April 2004, pp. 259268 (2004) 14

[12] Karlof, C., Wagner, D.: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasure. Ad-Hoc Networks 1(2-3), 293315 (2003) 15

[13] Manzo, M., Roosta, T., Sastry, S.: Time Synchronization Attacks in Sensor Networks. In: Proceedings of the 3rd ACMWorkshop on Security of Ad Hoc and Sensor Networks (SASN 2005), Alexandria, USA, November 2005, pp. 107116 (2005) 15

[14] D. Smetters and V. Jacobson, Securing Network Content, PARC, Tech. Rep., October 2009. 26, 39

[15] W.Zhang, S.Zhu, and G.Cao, Predistribution and local collaboration-based group rekeying for wireless sensor networks, Ad Hoc Networks, pp. 12291242, 2009. 35

[16] A.Chadha, Y.Liu, and S.Das, Group key distribution via local collaboration in wireless sensor netwroks, in Proc. IEEE Sensor and Ad Hoc communications and Networks, 2005, pp. 4654. 35

[17] Yuan Zhang; Yongluo Shen; SangKeun Lee; , "A Cluster-Based Group Key Management Scheme for Wireless Sensor Networks," Web Conference (APWEB), 2010 12th International Asia-Pacific , vol., no., pp.386-388, 6-8 April 2010 doi: 10.1109/APWeb.2010.39 35

[18] Sanchez, D.S.; Baldus, H.;"A Deterministic Pairwise Key Pre-distribution

Scheme for Mobile Sensor Networks," Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on , vol., no., pp. 277- 288, 05-09 Sept. 2005. 36

[19] Maerien, J.; Michiels, S.; Huygens, C.; Joosen, W.; , "MASY: MAnagement of Secret keYs for federated mobile wireless sensor networks," Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on , vol., no., pp.121-128, 11-13 Oct. 2010. 36

[20] Khan, Sarmad Ullah; Lavagno, Luciano; Pastrone, Claudio; Spirito, Maurizio; , "An effective key management scheme for mobile heterogeneous sensor networks," Information Society (i-Society), 2011 International Conference on , vol., no., pp.98-103, 27-29 June 2011. 56, 65, 69, 70, 72

[21] S.Hussain, F.Kausar, and A.Masood, An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks, IWCMC'07, 2007. 36

[22] Poornima, A.S.; Amberker, B.B.; , "Tree-based key management scheme for heterogeneous sensor networks," Networks, 2008. ICON 2008. 16th IEEE International Conference on , vol., no., pp.1-6, 12-14 Dec. 2008. 36

[23] Xinyu Jin; Putthapipat, P.; Deng Pan; Pissinou, N.; Makki, S.K.; , "Unpredictable Software-based Attestation Solution for node compromise detection in mobile WSN," GLOBECOM Workshops (GC Wkshps), 2010 IEEE , vol., no., pp.2059-2064, 6-10 Dec. 2010 36, 67

[24] A. Perrig, R. Szewczyk, J. Tygar, Victorwen, and D. E. Culler, Spins: Security Protocols for Sensor Networks, ACM Wireless Networking, Sept. 2002. 37

[25] Camtepe, S.A., Yener, B.: Key Management in Wireless Sensor Networks. In: On Wireless Sensor Network Security. IOS Press, Amsterdam (2008) 19

[26] Haowen Chan; Perrig, A.; Song, D.; , "Random key predistribution schemes for sensor networks," Security and Privacy, 2003. Proceedings. 2003 Symposium on , vol., no., pp. 197- 213, 11-14 May 2003 38, 46

[27] Alcaraz, C., Roman, R.: Applying Key Infrastructures for Sensor Networks in CIP/CIIP Scenarios. In: Lopez, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 166 178. Springer, Heidelberg (2006) 21

[28] ECRYPT Network of Excellence. eSTREAM, the ECRYPT Stream Cipher Project, http://www.ecrypt.eu.org/stream/ (retrieved on June 2009) 33

[29] Fang Liu, Maiou Jose Manny Rivera, Xiuzhen Cheng, Location-Aware Key

Management in wireless sensor networks, IWCMC06, 2006. 38

[30] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, MIT, 1979. 38

[31] Juwei Zhang,; Yugeng Sun,; Liping Liu,; , "NPKPS: A novel pairwise key predistribution scheme for wireless sensor networks," Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07). IET Conference on , vol., no., pp.446-449, 12-14 Dec. 2007. 38, 52

[32] Cheikhrouhou, O.; Kouba'a, A.; Boujelben, M.; Abid, M.; , "A lightweight user authentication scheme for Wireless Sensor Networks," Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on , vol., no., pp.1-7, 16-19 May 2010 38, 67, 68, 69, 70

[33] Huei-Ru Tseng; Rong-Hong Jan; Wuu Yang; , "An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks," Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE , vol., no., pp.986-990, 26-30 Nov. 2007. 38, 69, 70, 72

[34] Kyeong Tae Kim; Ramakrishna, R.S.; , "A Level-based Key Management for both In-Network Processing and Mobility in WSNs," Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on , vol., no., pp.1-8, 8-11 Oct. 2007. 38

[35] I-Hsun Chuang; Wei-Tsung Su; Chun-Yi Wu; Jang-Pong Hsu; Yau-Hwang Kuo; , "Two-Layered Dynamic Key Management in Mobile and Long-Lived Cluster-Based Wireless Sensor Networks," Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE , vol., no., pp.4145-4150, 11-15 March 2007. 38

[36] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly Secure Key Distribution for Dynamic Conferences, (1992) 471486. 38

[37] Du. X., Xiao, Y., Guizani, M. Chen, H. H., An effective key management scheme for heterogeneous sensor networks, Ad Hoc Networks, Vol. 5 No. 1, 2007, pp. 24-34. 38, 39, 63, 64, 65, 67, 68, 69

[38] Sarmad, U.K.; Lavagno, L.; Pastrone, C.; , "A key management scheme supporting node mobility in heterogeneous sensor networks," Emerging Technologies (ICET), 2010 6th International Conference on, vol., no., pp.364-369, 18-19 Oct.

2010. 63, 64, 65, 67

[39] Zhang Juwei; Zhang Liwen; , "A Key Management Scheme for Heterogeneous Wireless Sensor Networks Based on Group-Oriented Cryptography," Internet Technology and Applications, 2010 International Conference on , vol., no., pp.1-5, 20-22 Aug. 2010. 38, 63, 64

[40] Roman, R., Alcaraz, C., Sklavos, N.: On the Hardware Implementation Efficiency of Cryptographic Primitives. In: On Wireless Sensor Network Security. IOS Press, Amsterdam, ISBN: 978-1-58603-813-7 31

[41] Didla, S., Ault, A., Bagchi, S.: Optimizing AES for Embedded Devices and-Wireless Sensor Networks. In: Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2008), Innsbruck, Austria (March 2008) 32

[42] Jun Choi, K., Song, J.-I.: Investigation of Feasible Cryptographic Algorithms for Wireless Sensor Network. In: Proceedings of the 8th International Conference on Advanced Communication Technology (ICACT 2006), Phoenix Park, Korea (February 2006) 33

[43] Mantin, I.: Analysis of the Stream Cipher RC4. Masters Thesis, Weizmann Institute of Science (2001) 33

[44] Meiser, G., Eisenbarth, T., Lemke-Rust, K., Paar, C.: Efficient Implementation of eSTREAM Ciphers on 8-bit AVR Microcontrollers. In: Proceedings of the International Symposium on Industrial Embedded Systems (SIES 2008), Montpellier, France, June 2008, pp. 5866 (2008) 33

[45] Liu, A., Ning, P.: TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In: Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track, St. Louis, USA, April 2008, pp. 245256 (2008) 33, 55

[46] Seo, S.C., Han, D.-G., Kim, H.C., Hong, S.: TinyECCK: Efficient Elliptic Curve Cryptography Implementation over GF(2m) on 8-bit MICAz Mote. IEICE Transactions on Info and Systems E91-D(5), 13381347 (2008) 33

[47] Szczechowiak, P., Kargl, A., Scott, M., Collier, M.: On the Application of Pairing based Cryptography toWireless Sensor Networks. In: Proceedings of the 2nd ACM conference onWireless Network Security (WiSec 2009), Zurich, Switzerland, March 2009, pp. 112 (2009) 34

[48] Ganesan, P., Venugopalan, R., Peddabachagari, P., Dean, A., Mueller, F., Sichitiu, M.: Analyzing and Modeling Encryption Overhead for Sensor Network Nodes. In: Proceedings of the 2nd ACM International Conference onWireless Sensor Networks and Applications (WSNA 2003), San Diego, USA, September 2003, pp. 151159 (2003) 34

[49] Wang, X.: Recent Progress on SHA-1. Rump Session, Crypto 2005 (2005) 34

[50] NIST hash function competition, http://www.nist.gov/hash-competition (retrieved on June 2009) 34

[51] Eschenauer, L., Gligor, V.D.: A Key-management Scheme for Distributed Sensor Networks. In: Proceedings of the 9th ACM conference on Computer and communications security (CCS 2002), Washington, DC, USA, November 2002, pp. 4147 (2002) 19, 37, 38, 39, 51, 52, 63, 64, 65, 67, 68, 69, 105

[52] Du, W., Deng, J., Han, Y.S., Varshney, P., Katz, J., Khalili, A.: A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. ACM Transactions on Information and System Security (TISSEC) 8(2), 228258 (2005) 20

[53] Camtepe, S.A., Yener, B.: Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks. IEEE/ACM Transactions on Networking 15(2), 346358 (2007) 20

[54] Liu, D., Ning, P., Li, R.: Establishing Pairwise Keys in Distributed Sensor Networks. ACM Transactions on Information and System Security 8(1), 4177 (2005) 20

[55] Anderson, R.J., Chan, H., Perrig, A.: Key Infection: Smart Trust for Smart Dust. In: Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP 2004), Berlin, Germany, October 2004, pp. 206215 (2004) 20

[56] Seshadri, A., Luk, M., Perrig, A.: SAKE: Software attestation for key establishment in sensor networks. In: Nikoletseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) DCOSS 2008. LNCS, vol. 5067, pp. 372385. Springer, Heidelberg (2008) 21

[57] Panja, B., Madria, S., Bhargava, B.: Energy and Communication Efficient Group Key Management Protocol for Hierarchical Sensor Networks. In: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2006), Taichung, Taiwan (June 2006) 21

[58] Xiaojiang Du; Yang Xiao; Song Ci; Guizani, M.; Hsiao-Hwa Chen; , "A

Routing-Driven Key Management Scheme for Heterogeneous Sensor Networks," Communications, 2007. ICC '07. IEEE International Conference on , vol., no., pp.3407-3412, 24-28 June 2007. 39, 65

[59] Qing Yang; Qiaoliang Li; Sujun Li; , "An Efficient Key Management Scheme for Heterogeneous Sensor Networks," Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on , vol., no., pp.1-4, 12-14 Oct. 2008. 39, 65

[60] L. Zhou and Z. Haas, Securing ad hoc networks, Network, IEEE, vol. 13, no. 6, pp. 24 30, nov/dec 1999. 39, 40

[61] J. Kong, Z. Petros, H. Luo, S. Lu, and L. Zhang, Providing robust and ubiquitous security support for mobile ad-hoc networks, in Network Protocols, 2001. Ninth International Conference on, nov. 2001, pp. 251260. 39

[62] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, Self-securing ad hoc wireless networks, in Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on, july 2002, pp. 567 574. 39

[63] S. Capkun, L. Buttyan, and J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, Mobile Computing, IEEE Transactions on, vol. 2, no. 1, pp. 52 64, jan.-march 2003. 39

[64] A. Khalili, J. Katz, and W. Arbaugh, Toward secure key distribution in truly ad-hoc networks, in Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on, jan. 2003, pp. 342 346. 39, 40, 41

[65] H. Deng, A. Mukherjee, and D. Agrawal, Threshold and identity-based key management and authentication for wireless ad hoc networks, in Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, vol. 1, april 2004, pp. 107111 Vol.1. 39, 40, 41

[66] D. R. G. Rossini, Caching performance of content centric networks under multipath routing (and more), in Technical report, Telecom ParisTech, 2011. 97

[67] "http://www.avispa-project.org/." 73, 100

[68] P.Traynor, R. Kumar, H. Bin Saad, G. Cao, T. La Porta, Efficient hybrid security mechanisms for heterogeneous sensor networks, IEEE Trans. On Mobile Computing 6(6):663-677, 2007. 43

[69] D.D.Kouvatsos, G. Min and B. Qureshi, "Performance Issues in a Secure Health Monitoring Wireless Sensor Network" Fourth International Working Conference

on Performance modeling and Evaluation of Heterogeneous networks HEC-TEC, 2006. 44, 47

[70] Jara, A.J.; Marin, L.; Skarmeta, A.F.G.; Singh, D.; Bakul, G.; Daeyeoul Kim; , Secure Mobility Management Scheme for 6LoWPAN ID/Locator Split Architecture, Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on , vol., no., pp.310-315, June 30 2011-July 2 2011. 80, 81

[71] Papadimitriou, D.; Tschofenig, H.; Rosas, A.; Zahariadis, S.; et al; Fundamental Limitations of Current Internet and the path to Future Internet, European Commission, FIArch Group, Ver. 1.9, 2010. 80

[72] Raza, S.; Duquennoy, S.; Chung, T.; Yazar, D.; Voigt, T.; Roedig, U.; , "Securing communication in 6LoWPAN with compressed IPsec, "Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on , vol., no., pp.1-8, 27-29 June 2011. 81

[73] Jara, A.J.; Marin, L.; Skarmeta, A.F.G.; Singh, D.; Bakul, G.; Daeyeoul Kim; , Mobility Modeling and Security Validation of a Mobility Management Scheme Based on ECC for IP-based Wireless Sensor Networks (6LoWPAN), Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on , vol., no., pp.491-496, June 30 2011-July 2 2011. 81

[74] Joakim, E.; Fredrik, .; Niclas, F.; Nicolas, T.; Adam, D.; Thiemo, V.; Robert S.; Pedro, J.M.; COOJA/MSPSim: Interoperability Testing for Wireless Sensor Networks SIMUTools 2009, Rome, Italy. 86

[75] Newsome, J.; Shi, E.; Song, D.; Perrig, A.; , "The Sybil attack in sensor networks: analysis and defenses," Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on , vol., no., pp. 259- 268, 26-27 April 2004. doi: 10.1109/IPSN.2004.1307346 77