



216715 NEWCOM⁺⁺

DR.A.1

Intermediate report about wireless secrecy metrics and preliminary algorithms

Contractual Date of Delivery to the CEC: T0+12

Actual Date of Delivery to the CEC: T0+12

Editor(s): M. Debbah (CNRS), S. Fischer-Huebner (Chalmers), M. Guillaud (ftw.)

Participating institutions: XXX

Contributors: A. Abou El Kalam (CNRS), C. Alberto (ISMB), A. Atzeni (ISMB), A. Cappadonia (ISMB), E. Cesena (ISMB), M. Debbah (CNRS), S. Fischer-Hübner (Chalmers), M. Guillaud (ftw.), M. Kobayashi (CNRS), S. Lasaulce (CNRS), S. Lindskog (Chalmers), L. A. Martucci (Chalmers), D. Mazzocchi (ISMB), C. Pastrone (ISMB), S. Shamai (Technion), M. Spirito (ISMB)

Internal Reviewer(s): S. Benedetto, ISMB

Workpackage number: WPA

Nature: X

Total Effort Spent: 7 mm

Dissemination Level: Public

Version: 1

Abstract:

This deliverable provides a first overview of the state of the art on wireless security metrics and algorithms, which are particularly very important with respect to the intrinsic vulnerability of the wireless medium. The first part deals with building practical physical layer information-theoretic secure schemes. Indeed, Shannon formalized the concepts of capacity (as a transmission efficiency measure) and equivocation (as a measure of secrecy). While the concept of capacity has been extended to fading channels with the introduction of concepts like the outage capacity or the ergodic capacity, similar paths are yet to be developed concerning equivocation. In the second part, we present a taxonomy of threats for wireless networks, and particularly for wireless sensor networks, in terms of the affected protocol layers. Besides, an overview to attack tree and attack graph techniques as formal means for describing and measuring vulnerabilities and thus also for assessing also the security level of wireless sensor networks is provided. Finally, an overview to suitable security concepts for wireless networks is given.

Keyword list: Wireless, Eavesdropper, Wiretap channel, Privacy, Threat, Confidentiality, Authentication

TABLE OF CONTENTS

1	Introduction	4
1.1	Structure of this Document	4
1.2	Glossary	5
2	Secrecy based on fading channels	7
2.1	Key generation through common channel fading information	7
2.2	Information theoretic secrecy	9
3	Threat Model and Security Requirements for Wireless (Sensor) Networks	11
3.1	Threat Model	11
3.1.1	WSN technology characterization	12
3.1.2	Attacker characterization	13
3.1.3	Security targets	14
3.1.4	Overall system vulnerabilities	14
3.1.5	Attacks result	14
3.2	Security requirements	15
4	Threat analysis	17
4.1	Passive attacks	17
4.2	Active attacks	17
4.3	Attack Classification by Network Layers	20
4.3.1	Physical Layer	20
4.3.2	Data Link Layer	21
4.3.3	Network Layer	22
4.3.4	Transport Layer	23
4.3.5	Application Layer	23
4.3.6	Multi-layer attacks	24
5	Attack Tree and Attack Graph	25
5.1	High-level attack trees	25
5.2	Predecessors of attack tree techniques	25
5.3	Model-based attack tree and attack graph construction	26
5.4	Multi-semantic attack tree	26
5.5	Attack graph scoring	27
5.5.1	Vulnerability enumeration	27
5.5.2	Privilege graph	30
5.5.3	Risk analysis methodology	31
5.6	Attack tree complexity mitigation	32
5.7	Conclusions	32
6	Selected security concepts for Wireless Networks	33
6.1	Security Models	33
6.1.1	Discretionary and Mandatory Access Control	33
6.1.2	Role Based-Access Control	33
6.1.3	Organizational-based Access Control	33
6.2	Basic Security Mechanisms	34
6.2.1	Cryptographic primitives	34
6.2.2	Access Management	36
6.2.3	Physical layer	37
6.2.4	Data-Link layer	37

6.2.5	Network layer	38
6.2.6	Transport Layer	40
6.2.7	Application Layer	40
6.2.8	Multi-Layer	43
7	Intrusion Detection Systems	44
7.1	IDS Taxonomies	44
7.1.1	Source of event data	45
7.1.2	Architecture	46
7.1.3	Locations of data collection and data processing	46
7.1.4	Detection method	47
7.1.5	Time of detection and granularity of data processing	47
7.1.6	Behavior on detection	48
7.2	Wireless IDS	48
7.2.1	Wireless IDSs vs classical IDSs	48
7.2.2	Examples of Wireless IDSs	49
7.2.3	Wireless IDS drawbacks	49
8	Conclusion and Outlook	50

1 INTRODUCTION

Wireless Networks share most security threats that exist for hardwired networks, and have in addition to deal with security problems which are specific for them: As wireless networks typically have no geographical boundaries, security provisioning cannot be deployed in the same manner as in hardwired networks, i.e., by setting perimeters and protecting these perimeters with traditional security technologies such as border firewalls. Besides, means of physical protection are often limited. Also, wireless networks have to deal with location privacy threats to which mobile users are often exposed. Moreover, for wireless systems, especially for wireless sensor networks (WSN), with heterogenous devices with varying and limited capacities, one often has to accept a tradeoff between the level of security and performance when there is a lack of information on the eavesdropper. To be able to trade security in an acceptable level, appropriate metrics for security are needed.

Within the Newcom++ Network of Excellence, WPR.A deals with security in order to find new ways to exploit the wireless medium as a security opportunity by defining new security metrics and developing Wireless Network Security protocols. In particular, TA.1 and TA.2 are dealing with Physical Layer Security, whereas TA.3 deals more broadly with wireless security protocols, mechanisms and architectures.

The first part of the report deals with building practical physical layer information-theoretic secure schemes using the fading characteristics such as channel reciprocity or the frequency selectivity of the channel. Fading is due to scattering of the electromagnetic waves through reflection, refraction and diffraction. In general, communications systems are oblivious to the properties (shape, position, indices...) of the scatterers, and the channel is therefore considered random. Two directions are explored in this part: the first one consists in using the channel fading (in particular its reciprocity, see below) to generate cryptographic keys. The second direction has the aim to evaluate and exploit the various features of the fading channel in the view of achieving information-theoretic secrecy, through e.g. opportunistic methods based on the knowledge of the channel.

In the second part, for developing suitable security measures for wireless networks, we have to first analyze existent threats and the effectiveness of available countermeasures. Hence, the focus of the remainder of the deliverable is on wireless network security and in particular WSN security.

The report is a first joint contribution addressing these aspects and forms the basis of WPA's future work on security metrics for vulnerabilities of wireless networks and selected countermeasures.

Note that since the problems considered in this report are closely related to the accuracy of the available channel models, collaborations on measurements with Newcom++ work-package R.1 (Modeling, calibration, and validation of multi-dispersive, multi-link channels) have already started.

1.1 Structure of this Document

In this deliverable, we first provide in section 1.2 a glossary with definitions of the technical terms used in the remainder of this document. In section 2, key generation protocols through common channel fading information for wireless networks are provided and discussed. Note by the way that two of the authors of the report (M. Debbah from Supélec and M. Guillaud from ftw.) were awarded the Mario Boella prize award in 2005 for a protocol generating secret keys using the wireless medium. Information theoretic considerations are also provided on the physical limits on secure information transfer. Note here also that the authors of the report have made important contributions to the case of multiple antenna and frequency selective secure channels. In section 3, we then present a threat model for Wireless (Sensor) Networks and define major security and privacy properties for WSN. Section 4 then provides an overview on security and privacy attacks for wireless networks by classifying them according to the network layers that they are targeted at. In section 5 we develop attack trees and attack graphs targeted on WSN as a form of malicious threat description and discuss approaches for deriving measures for vulnerabilities from them. Finally, we provide an introduction to selected security concepts and mechanisms in section 6 and to Intrusion Detection Systems (IDS) in particular in section 7, and show what security and privacy threats

in Wireless Networks discussed in section 4 they are addressing.

1.2 Glossary

This glossary defines the technical terms used within the document.

Adversary refer to Attacker.

Anonymity of a subject means that the subject is not identifiable within a set of subjects, the anonymity set [1].

Asset entities that the owner presumably places value upon

Attack attempt to gain unauthorized access to a service, resource, or information, or the attempt to compromise integrity, availability, or confidentiality. Note that success is not necessary [2].

Attacker the originator of an attack.

Authentication server server that provides authentication services to users or other systems. For example, the user passes its identity and password (or certificate, smartcard, biometric data) to the authentication server; the latter verifies this data and grants the authentication proof (e.g., a credential) to the user.

Authorization server a server that consults the security policy, extracts the relevant security rules, evaluates these rules with the current access parameters, eventually, invokes the conflict resolution process, and generates the corresponding credentials that permit the access to resources.

Availability the property of being accessible and usable upon demand by an authorised entity [3].

Bell-LaPadula Model a security model developed by David Elliott Bell and Len LaPadula to formalize the U.S. Department of Defense multilevel security policy. The model is a formal state transition model that describes a set of access control rules by the use of security labels on objects, from the most sensitive to the least sensitive, and clearances for subjects. For example, a set of security labels for documents might be "Top Secret", "Secret", "Confidential", and "Unclassified".

Confidentiality the property that information is not made available or disclosed to unauthorized individuals, entities, or processes [3]. It implies that information is readable only to authorized users.

Flaw refer to Vulnerability.

Integrity the property of safeguarding the accuracy and completeness of assets [3]. It implies that data is modified only by authorized users and only in an authorized manner.

Intruder refer to Attacker.

Location Privacy the possibility of users to control the disclosure of information about their physical location to others.

Privacy the right of individuals to determine for themselves when, how, to what extent and for what purposes information about them is communicated to others.

Privacy-enhancing Technologies technologies that are enforcing privacy principles in order to protect and enhance the privacy of users of information technology and/or of individuals about whom personal data are processed. One fundamental privacy principle that serves as the foundation for the Privacy-Enhancing Technologies that are aiming at providing anonymity, pseudonymity, or unobservability for users and/or other data subjects, is the privacy principles of data minimization. It requires that the collection of personally identifiable data should be minimized (and if possible

avoided), because obviously privacy is best protected if no personal data at all (or at least as little data as possible) is collected or processed.

Pseudonymity the use of pseudonyms as identifiers [1]

Risk probability that an attacker will exploit a particular vulnerability, causing harm to a system asset [2].

Risk analysis a procedure to identify threats and vulnerabilities, analyze their impacts, and developing a security plan highlighting how the impacts can be eliminated or reduced. Risk Analysis aims at striking an economic balance between impact of risks and costs of protective measures.

Risk management the procedure of developing a security plan as part of a Risk Analysis.

Security studies problems, methods and solutions related to the three security properties: availability, confidentiality, and integrity.

Security policy the set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. Basically, a security policy is specified through: the security objectives that must be satisfied (expressed in terms of confidentiality, integrity and availability) and the security rules expressing how the system may evolve in a secure way (who has access to what and in which conditions).

Security model rigorously defines a security policy. Generally, a security model is a "formal system" used to specify and reason on the security policy (i.e., it is used as a basis for formal specification proofs). It is thus intended to abstract the security policy and handle its complexity; represent the secure states of a system as well as the way in which the system may evolve, verify the consistency of the security policy, detect and resolve possible conflicts.

Security mechanisms techniques used to implement the authentication and authorization, e.g., credentials, capacities, cryptographic transformations such as signature and encryption, access control lists (ACL).

Threat any circumstance or event (such as the existence of an attacker and vulnerabilities) with the potential to adversely impact a system through a security breach [2].

Unlinkability Unlinkability of two or more items of interest (e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these items are related or not [1].

Unobservability ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used [4]. A corresponding, but more general definition is provided by [1]: Unobservability of an item of interest (e.g., a subject, messages, action) means that all uninvolved subjects cannot sufficiently distinguish whether it exists or not.

Vulnerability weakness in system security design, implementation, configuration or limitations that could be exploited [2].

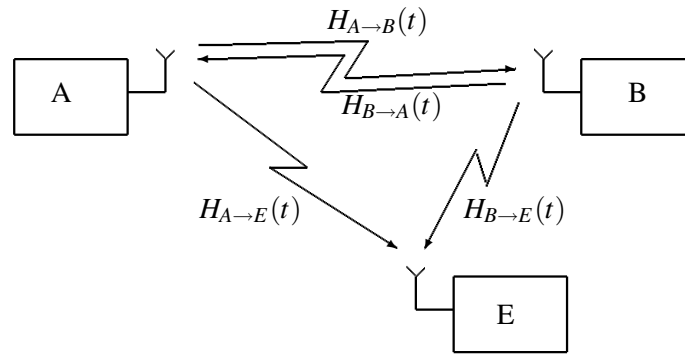


Figure 1: Setup

2 SECRECY BASED ON FADING CHANNELS

2.1 Key generation through common channel fading information

It is a well-known fact in wireless propagation that the channels experienced in opposite directions between two antennas or antenna arrays share some properties. Although it has been known for over a century [5], this property is however rarely exploited in current communications systems. In addition to the symmetry of the impulse responses (known as reciprocity), which holds for bi-directional communications over the same frequency band (Time-Division Duplex, or TDD, systems), this symmetry applies more generally to other characteristics such as the total power, or the Doppler spectrum.

In addition to the various link-adaptation techniques enabled by channel knowledge available at the transmitter thanks to reciprocity, applications to security are also possible and have been considered. They are based on the observation that the channel experienced in both directions are similar, whereas the channel properties experienced by an eavesdropper non co-located with either side of the transmission would exhibit significant differences with the channels experienced by the intended users.

Considering a system (see Fig. 1) comprised of nodes A and B engaged in bi-directional communication, and a passive eavesdropper E, we assume that all three nodes can estimate the properties of the channels from A and B, using e.g. periodically transmitted pilot sequences. We represent the information about the channel available to nodes A and B by the random variables X and Y respectively. This knowledge can consist either of the impulse response of the channel (i.e. $X = \widehat{H}_{B \rightarrow A}$, $Y = \widehat{H}_{A \rightarrow B}$), or any other channel-related property. The information about the channel collected by the eavesdropper E is represented by random variable Z .

From an information theoretic point of view, the fact that part of the information between X and Y is not present in Z can be used to generate a shared secret key. Generating such a shared secret can be useful either for one-time pad applications, as foreseen by Shannon [6], where they enable perfect secrecy, or to solve the key distribution problem in symmetric-key cryptosystems.

Symmetric-key (or public-key) cryptosystems rely on the high (unaffordable for the eavesdropper) complexity of finding the solution to a publicly known problem (typically, the problem of factorization into two large primes), while the intended participants in the communication have been privately provided with the solution (the key). Such methods are deemed *computationally secure*, as opposed to the information-theoretic security considered in Task TA.2 (see section 2.2). More information on the classification of physical layer attacks can be found in section 4.3.1. Public-key cryptosystems are the cornerstone of the current generation of security protocols used e.g. on the internet. In such a scenario, the need that the intended receivers, and them only, be reliably provided with the key, is a critical requirement. Current systems rely on partially satisfactory solutions of the key distribution problem, such as using keys distributed by a trusted authority (operating system vendor, browser vendor...). The weakness in this case lies in the fact that the distribution system is implicitly trusted (e.g. it ignores the fact that operating system CD-ROMs or browser downloads could be tampered with). Furthermore, since the

amount and the frequency of keys distributed according to this mechanism is limited, it can not apply to session keys themselves. Therefore, this mechanism always relies on a central certification authority.

Using the mutual information between X and Y , it is possible to derive a common key based on the observed realizations of those variables. The advantages of this method are:

- it enables more frequent changes of the key. In the context of a computationally secure system, where the probability of a successful attack increases with time, this is a welcome feature.
- If enough key entropy can be generated using this method, one-time pad encryption (achieving information-theoretic secrecy if the key is not compromised) can be used.
- It does not rely on a third-party to guarantee the secrecy of the key distribution.

Among the drawbacks,

- this method enables encryption but not authentication. In other words, while it is possible to ensure that no third party is eavesdropping on the communication, this method does not help in ensuring that the communication is indeed taking place with the intended party. However, if some amount of shared entropy is available at the beginning of a communication, it can be shown [7] that privacy algorithm benefit from the authentication that this prior entropy enables.

Maurer et al. [8, 9] have proposed a way for A and B to agree with vanishing error probability on a secret key without prior common secret, by publicly discussing the observed variables X and Y . The secret-key rate $S(X;Y||Z)$ is defined in [8] as the rate at which a secret key can be generated, while keeping the rate at which E obtains information arbitrarily small. It is shown to be bounded by

$$\max\{I(Y;X) - I(Z,X), I(X;Y) - I(Z,Y)\} \leq S(X;Y||Z) \leq \min\{I(X;Y), I(X;Y|Z)\}. \quad (1)$$

The upper bound was later refined in [9] where a tighter bound, based on the notion of intrinsic mutual information, is introduced. Methods to achieve the key generation described above have been developed. In particular, privacy amplification [10, 7] is a technique for transforming a string which is partially secret into a highly secret, shorter string. Note that most of these methods require that the channel is authenticated (i.e. the eavesdropper is not allowed to actively modify its output), although secret-key agreement over unauthenticated channels (i.e. in the presence of an active adversary) has also been addressed – see [11] and references in [9].

Considering the application of key-generation methods based on channel fading to realistic channel models, we seek to answer the following questions:

1. How much key entropy can be generated from real-world channels ?
2. What is the degree of secrecy of the system ? (This question should be answered in relation with the secrecy metrics developed in Task TA.3, see section 3).

In particular, the following characteristics of the real-world channels have not been fully incorporated in the past studies:

- Channel correlation in time and space. Independent realizations are assumed in the vast majority of the existing works in this area, whereas the channel realizations are in general correlated with time and among antennas when an antenna array is used. Furthermore, under mild assumptions, it is difficult to find an upper bound to these correlations since:
 - in a perfectly static scenario the channel could potentially remain constant for an arbitrary amount of time.

- under specific scatterer geometries, the channel seen by all antennas in an array can potentially become fully correlated.
- current upper and lower bounds on the secret key rate are based on the perfect knowledge of the joint statistics of (X, Y, Z) , which is not available in practice. The secret key rate effectively achievable in realistic scenarios still remains a largely open problem. Experimental results about the joint distribution of the observations of the reciprocal channel (X, Y) are still scarce. Notably, an attempt at characterizing the reciprocity including the disturbance coming from the non-reciprocal transmit and receive circuits can be found in [12], and research in this area is ongoing. Obtaining experimental data for multi-user links (which would be an indication to the distribution of (X, Y, Z)) is also currently receiving intense interest. It is expected that the investigations about characterizing multi-user and reciprocal channel models taking place in Newcom++ work-package R.1 will largely benefit the investigations described here.

Furthermore, little work has been done in the direction of low-complexity, suboptimal methods enabling a practical implementation of the methods outlined above. Indeed, in practical scenarios in wireless communications, the dimensionality of the channel estimate can be high (typically, up to a hundred of scalar components) due to the space dimension (if antenna arrays are present at either or both transmitter and receiver), and the typical bandwidth which make the channel highly frequency selective. Therefore, it is desirable to

- derive functions f_a, f_b of the channel estimates that reduce the dimensionality of the problem with minimum impact on the performance of the key generation scheme, i.e. such that the dimensions of $X' = f_a(X)$ and $Y' = f_b(Y)$ are tractable, while maintaining $S(X'; Y' | Z) \approx S(X; Y | Z)$. This is essentially a data compression problem.
- Develop a better understanding of how the limits of channel variability affect security. Some attempts at quantifying the secret key have been made for some simplified channel models, such as jointly Gaussian bidirectional channels [13, 14], frequency-selective channels following the ITU channel models [15], or ultra-wideband channel models [16]. More general analysis of the effect of space and time correlation are in order.
- Develop practical key-generation algorithms operating reasonably close to the theoretical secret-key rate. Some methods exploiting the multipath characteristic of the wireless channel [17] have already been proposed.

2.2 Information theoretic secrecy

This task is focused on the implementation of information-theoretically secure communications means over the wireless channel. It considers the various simplifications of the channel model commonly used in communications applications, in the context of security. It is based on the observation that all models and assumptions commonly used to design communications systems are made under implicit hypothesis that all elements in the system have been designed to ensure efficient and reliable communications, but without considering possible security implications.

For a review of available results, let us consider the secured communication such that the transmitter wishes to send the confidential message to its receiver while keeping the eavesdropper ignorant of the message. Wyner [18] introduced the wiretap channel to model the degraded broadcast channel (BC) where the eavesdropper observes a degraded version of the receiver's signal. In his model, the confidentiality is measured by the equivocation rate, i.e., the mutual information between the confidential message and the eavesdropper's observation. For the discrete memoryless degraded wiretap channel, Wyner characterized the capacity-equivocation region and showed that a non-zero secrecy rate can be achieved [18]. The most important operating point on the capacity-equivocation region is the secrecy capacity, i.e., the

largest reliable communication rate such that the eavesdropper obtains no information about the confidential message (the equivocation rate is as large as the message rate). The secrecy capacity of the Gaussian wiretap channel was given in [19]. Csizar and Korner considered a more general wiretap channel in which a common message for both receivers is sent in addition to the confidential message [20]. For this model known as the BC with confidential messages (BCC), the capacity-equivocation region as well as the capacity region were characterized. However, these results had limited applications as a strictly positive secrecy capacity is possible only if the legitimate receiver has a better channel condition than the eavesdropper. This is rarely met in practice.

Motivated by the emerging need for the secured wireless communication as well as the less pessimistic conclusion of [21] that showed the possibility of achieving a strictly positive secrecy capacity even when the legitimate channel has a worse quality than the eavesdropper's channel, the information theoretical security has attracted considerable attention. A significant effort has been made to opportunistically exploit the space/time/frequency/user dimensions for secrecy communications [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 29, 30]. In [22], the secrecy capacity of the ergodic slow fading with perfect channel state information at the transmitter (CSIT) was characterized and the power/rate allocation under partial CSIT (the knowledge on the channel of the intended receiver only) was derived. The secrecy capacity of the parallel fading channels was given [24, 25] where [25] considered the BCC with a common message. The impact of the fading under partial CSIT was studied as the outage formulation in [27, 33, 34]. Moreover, the secrecy capacity of the wiretap channel with multiple antennas has been studied in [26, 27, 28, 29, 30, 31] and references therein. In particular, the secrecy capacity of the Multiple Input Multiple Output (MIMO) wiretap channel was fully characterized by different authors [32, 29, 30]. In order to relax the unrealistic assumption of perfect CSIT while avoiding the outage event, the compound wiretap channel has been also addressed recently [35, 24, 36]. In particular, the secrecy capacity of the degraded discrete memoryless compound wiretap channel [35], the degraded MIMO compound wiretap [35], and the Gaussian parallel compound wiretap channel [24, 35, 36] were given. The wiretap channel was extended also to multiple transmitters and receivers. The BCC with multiple receivers each of which is equipped with a single antenna was addressed in [37, 38]. The capacity region of the two-user Gaussian MISO-BCC was given in [39]. The two-user MIMO-BCC was considered in [40] where the capacity region for the case of one common and one confidential message was characterized. The interference channel with confidential messages was also considered in [41, 42, 37].

The case of frequency selective channels has only been recently touched upon [43]. For instance, the assumption that receivers of an Orthogonal Frequency Division Multiplexing (OFDM) modulation will drop the cyclic prefix is perfectly acceptable in the case of an intended receiver. Ongoing work on this topic has already shown that it is possible to achieve non-zero capacity by communicating only on the frequencies corresponding to the zeros of the (assumed known) frequency-selective channel to an eavesdropper [43] if the cyclic prefix is dropped. Although reasonable in certain contexts (e.g. opportunistic communications), this assumption is not warranted in a security context: nothing forces a potential eavesdropper to apply the same processing as the intended receiver. Therefore, systems whose privacy features are based purely on the discrete-multitone equivalent representation of the channel would be prone to attacks based on the analysis of the time-domain wireless signal.

The goal of this task is to identify the possible side-channel attacks specific to the wireless medium and the classical wireless communication techniques, such as the one quoted above, and to devise secrecy methods that are immune to such attacks. In particular, it is planned to investigate

- opportunistic scheduling for secrecy,
- auto-calibration techniques suitable for security issues.

3 THREAT MODEL AND SECURITY REQUIREMENTS FOR WIRELESS (SENSOR) NETWORKS

This section presents an overview of the security and privacy issues to be considered in wireless networks, with a particular focus on Wireless Sensor Networks (WSNs). Although most of the security threats in wireless networks are similar to those considered in wired networks, some are exacerbated because of the intrinsic vulnerability of the wireless medium. In addition, several traditional countermeasures are not suitable for wireless technologies.

It is worth noting that "*wireless*" is a broad and sometimes all-encompassing term. Actually, a plethora of wireless solutions exists, each with different characteristics in terms of bandwidth and data rate offered, communication latency, traffic patterns, networking capabilities and power consumption requirements.

WSNs are usually referred to as simple, low-rate communication networks allowing wireless connectivity among a very large number (in theory, arbitrarily high) of autonomous devices (*sensor nodes*) with poor computational and power resources. The main characteristics are the *self-configuration* and *self-organization* attitude: once deployed in the monitoring area, WSN nodes exchange control packets to form a time-varying ad-hoc topology and thus to keep each node always connected to the rest of the network, in particular after unpredictable events that could cause temporary node failure and its disappearance from the network. Furthermore, to save energy and extend network life-time, nodes usually have to limit the transmitted power, resulting in a shortest transmission range. As a consequence, two any nodes whose distance is larger than their respective radio coverage cannot directly communicate with each other: to overcome this limitation, *multi-hop routing* is used. As abovementioned, a WSN node is characterized by limited resources and reduced processing capabilities. For this reason, distributed signal processing, distributed data storage and data aggregation mechanisms are solutions commonly adopted in WSNs to fully exploit the inherent cooperative nature of such technology.

Indeed, due to the WSNs severe constraints, providing security poses unique challenges as compared with other more conventional networks. With this in mind, it is desirable that security aspects be examined at design time and taken into account along with other functional requirements: a *trade-off* between the level of security guaranteed and the amount of resources spent should to be considered.

In particular, the selection of the most appropriate security scheme to be employed in WSNs should be based on the identification of a set of possible attacks. In fact, this analysis of the security concerns results into the definition of a **threat model**. In general, *threat modeling* assumes that any system has assets that are worth protecting and try to identify relevant weaknesses, i.e. *vulnerabilities* whose exploitation might allow malicious attackers to cause harm or damage. In this context, the characterization of the attackers according to his motive, capabilities and targets can be leveraged to determine the most suitable *countermeasures* and mitigate the potential *security threats*. This process is commonly used in the field of information security, where the information asset plays the center role.

Therefore, a WSN adapted threat model can be used to formally analyze security threats: the resulting security assessment, along with the desired system and security requirements, permits to determine which vulnerabilities can be tolerated and to focus on the necessary *defense technologies*.

The rest of the section is organized as follows. Section 3.1 addresses the definition of an effective threat model for WSNs, while Section 3.2 focuses on typical security requirements.

3.1 Threat Model

This section present the most relevant characteristics to be considered when defining a threat model for WSNs. In particular, several factors deeply affect the overall risk and should be taken into account:

- WSN technology inherent constraints (3.1.1)
- Attacker characterization according to motive, determination, knowledge, resources and relevant capabilities (3.1.2)

- Security targets (3.1.3)
- Overall system vulnerabilities (3.1.4)
- Attacks' impact on system functioning (3.1.5).

In the following, all these parameters are briefly introduced.

3.1.1 WSN technology characterization

As previously stated, from a technical perspective, WSNs introduce many constraints compared to traditional computer networks, thus posing unique challenges at design time. Therefore, it is required to better investigate what are these limitations and fully understand their importance with respect to security [2][44].

It is worth observing that **resource scarcity** is probably the main constraint for this technology. In fact, WSN nodes usually adopt *low-power microprocessors* along with limited RAM memory and flash storage space. As far as the energy provisioning to the node is concerned, external batteries are usually considered. While batteries have limited duration, changing or recharging batteries may impact significantly on maintenance costs and system downtime. In terms of performance many solutions adopted to comply with different requirements tend to affect negatively energy consumption; thus, especially when available energy is limited, a trade-off between performance and energy consumption is needed. In WSNs, the available *bandwidth* and *data rate* are limited as well: typical bit rate values are below 250 kbps. In this scenario, the risk of *resource consumption* is clear and can be exploited by malicious adversaries. In addition, all security approaches require a certain amount of resources and introduce an additional overhead. An inadequate choice of security countermeasures could lead to an even worse situation.

Other WSN inherent attributes clearly affect threat model definition:

- *Unreliable communication.* Actually, this characteristic is mainly due to the broadcast nature of the wireless medium. Messages exchanged among different nodes of the network may be corrupted because of channel errors or collisions. To increase system reliability, it is then necessary to use communication protocols able to handle errors, thus introducing additional resource consumption. In a densely deployed WSN, this can be a major problem: network congestion could cause an increase in system latency and the drop of messages in overloaded nodes. All the network services based on near real-time communication can then suffer this condition.
- *Unattended operation.* According to specific application scenarios, WSNs can operate unattended in a remote location for long periods of time. Therefore, WSN nodes are likely to be exposed to physical attacks (casual tampering, vandalism but also bad weather) and become an attractive target.
- *Distributed nature.* A cooperative distributed approach is very common in WSNs: while it can be used to overcome specific resource limitations, it could also be exploited by attackers.
- *Low-cost.* WSNs are likely to be composed of a very large number of nodes: clearly, this is economically sustainable only if the cost of a single node is enough small. In addition, manual intervention need to be minimized. This poses requirements that can lead to inaccurate design and implementation errors.
- *Application specificity.* Limited computational and power constraints, along with low-cost requirements, may dictate the design of application specific solutions. In particular, code modularity and stack layers decoupling could be disregarded, thus giving rise to new vulnerabilities.

3.1.2 Attacker characterization

A detailed attacker characterization is crucial to foresee what vulnerabilities are most likely to be exploited to perform an attack and identify relevant defense.

As far as this characterization process is concerned, several classifications have been proposed in the literature. In [2], the authors use an analysis of security incidents on the Internet [45] to describe the attacker according to four *dimensions: motive, determination, knowledge and resources*. In particular, the following classification is proposed:

- *Passerby*. This type attacker has not precise motivations, little knowledge and uses few resources.
- *Vandal*. His/her motivations are clearer and he/she is moderately determined; few resources are used.
- *Hacker*. He/she is highly knowledgeable and his behaviour is due to curiosity and interest. The achievement of his/her objectives requires using a moderate amount of resources.
- *Raider*. He/she is highly motivated, highly determined and moderate-highly knowledgeable; his/her attacks need a moderate amount of resources.
- *Terrorist or Foreign Power*. He/she is very motivated and determined, highly knowledgeable. In addition, he/she can use a lot of resources.

The above list can be used to provide a rough description of the attacker. However, a more detailed characterization can be carried out if attacker capabilities are analyzed. In particular, attackers can be classified according to their technical capabilities and the strategies adopted to cause harm to a system asset.

A first distinction can be made between **mote-class attackers** and **laptop-class attackers** according to their available resources [46].

A *mote-class attacker* uses devices with similar features to the deployed sensor nodes in term of power resources, computational capabilities and radio transmission power. In contrast, a *laptop-class attacker* can leverage on more powerful devices, acquiring an advantage on legitimate nodes (*capability asymmetry*). These latter devices have greater power resources, memory capacity, computational capabilities, transmission power or RF sensitivity, allowing to perform wider attacks. Moreover, high bandwidth and low latency communications might be provided, permitting collaboration among malicious peers. In addition, while mote-class attacker's area of influence can be limited to a localized region, laptop-class attackers can perform attacks affecting a larger number of nodes.

A second classification relates to the level of interaction between the WSN and the attacker while attempting to exploit a particular vulnerability. According to this point of view, **passive** and **active** attackers can be considered.

A *passive attacker* is involved in security threats related to *interception*: an adversary can eavesdrop on transmissions and gain unauthorized access to the message content. A complete traffic analysis can be performed as well: the attackers can sniff and examine all the messages exchange among the nodes in the WSNs in order to find useful information. In contrast, an *active attacker* is involved in active threats like communication interruption, message modification or fabrication (e.g. injection of falsa data).

A third distinction (orthogonal to the first one) can be made between **outsider** and **insider** attacks [46][47].

In an *outsider attack*, the adversary device is not authorized to join the wireless sensor network. According to the broadcast nature of the wireless medium, the attacker can *passively* eavesdrop on WSN communications, trying to pilfer sensitive data. Alternatively, the bogus node can perform *active attacks* maliciously altering or spoofing packets to break data authenticity or jamming the network with radio interfering signals. Furthermore, an *outsider attacker* can operate in order to disable sensor nodes. To this aim, several strategies can be followed e.g. denial of service (DoS) attacks to exhaust node battery resources, node capture or node physical destruction.

In an *insider attack*, the adversary device may be an authorized participant of the sensor network instead. Compromised sensor nodes may be reprogrammed with malicious code or laptop-class devices, in possession of secret key material, can try to attack legitimate nodes. It is worth noting that benign node failures could be considered as intentional attacks and determining the real causes is a concern of its own [47].

However, an adversary can be fully described by using all the above classifications. Other factors can impact the definition of attackers capabilities:

- *the number of attackers*
- *the coordination of attackers.*

In fact, multiple malicious adversary devices may actively *cooperative* to perform an attack against a system asset. Adversaries could be also coordinated by a central controller. In such conditions, the potential impact can be magnified. In other simpler scenarios, attackers could operate in an independent way, even if performing a similar attack.

3.1.3 Security targets

The identification of the main security targets may provide useful information to the design of a relevant defense, thus allowing to effectively use the limited resources available in WSN nodes. In the following two examples of the security targets usually considered in WSNs are reported:

- A specific layer in the node protocol stack (e.g. physical, link, network, transport or application)
- An in-network service (e.g. data aggregation, synchronization, distributed location, ...).

According to the criticality of a target, an attack can then lead to different levels of risk.

3.1.4 Overall system vulnerabilities

Once identified the security target, the attacker can perform an effective attack leveraging on existing vulnerabilities, i.e. weaknesses in the system of interest. As far WSNs are concerned, two main types of vulnerabilities can be exploited by attackers [2]:

- *Physical.* They relate to the physical damage of the system caused by intentional tampering or environment conditions.
- *Logical.* They refer to design flaws (e.g. due to erroneous requirements or threat analysis for specific protocols), implementation flaws, configuration errors or resource exhaustion (e.g. due to the limited memory resources in WSN nodes).

3.1.5 Attacks result

In a threat model, the attack results must be considered too. In fact, their analysis could be useful to define recovery mechanisms to be applied after a specific attack has been successfully performed. Attack results can be divided into different classes according to their effectiveness. For example, attacks on availability can be classified in the following manner:

- *Little or no impact;*
- *System performance degradation* (e.g. services functioning may be temporarily altered);
- *Services disruption* (i.e. high impact on system functioning);
- *Overall system disruption* (i.e. the system has been disabled due to the attack performed).

3.2 Security requirements

The definition of security requirements is a relevant issue and may differ according to application needs ([47][46][48]). Security is important in most application scenarios where sensitive data are considered and possible attacks against the sensor network may permit damages to the health or safety of people. Relevant examples include WSNs deployed into military scenarios but also applications whose aim is to monitor health parameters or detect harmful and toxic substances with the purpose of preventing terrorist attacks and managing emergencies. In details, all public safety applications have strong requirements in terms of system availability.

Most application require robustness against outsider attacks. In the presence of an insider attacker, mechanisms able to detect compromised nodes are desirable. Nevertheless, in this latter case only a graceful degradation of performance is generally conceivable.

Defined the specific requirements, it is worth recalling that WSN nodes usually have severe constraints and a trade-off between performance, security and energy consumption is needed.

The main security requirements are described in the following list:

- *Authentication*. Since attackers can easily inject packets, authentication is necessary: it enables a node to verify that the message originates from a trusted source (source authentication) and ensures data integrity, i.e. data has not been modified in transit (data authentication).
- *Secrecy*. In wireless communications any adversary can eavesdrop packets and gain access to sensitive and private information. This is a security concern in many WSN applications. Encryption is generally used for keeping sensed data secret, along with a shared secret key between the communicating peers, hence achieving data confidentiality. It is also possible to steal sensitive information by accessing the stored sensor data available through remote access. The standard approach is to devise access control policies in order to make data available only to trusted parties.
- *Availability*. Availability means that the sensor network maintains its functionality without interruption. The network must continue operating also after node failures or in presence of node compromise, ensuring graceful degradation.
- *Service integrity*. The application layer services implemented in the wireless sensor network must be protected from possible malicious attacks allowing the system to perform its task. These services include secure group management and secure data aggregation. The key issue is that sensor nodes are limited in their power resources, computational and communication capabilities. Therefore sensor networks often use collaborative processing, performing data aggregation and analysis by a group of nodes, increasing in this way WSN's efficiency and reliability along with its lifetime. Consequently, a secure group management is required. Suitable protocols securely admit new group members and support secure group communication. Data aggregation can then be realized: specific nodes collect information from neighbours, aggregate data and send it to the gateway, reducing the traffic in the network. In this scenario, the system should be able to detect and reject faulty data maliciously inserted by compromised nodes or resulting from benign nodes malfunctioning. This issue is commonly referred to as *data reliability*. Another example of service regards time synchronization in WSNs. Several methods have been designed without considering security concerns, whereas it is desirable to have protocols able to achieve graceful degradation in the presence of node compromise and robust to spiteful attacks.
- *Privacy*. Privacy concerns are also to be taken into account. Information regarding personal data or data that can be linked to an individual exist in different layers of data communication, from radio characteristics of a wireless transponder to information existing in the application layers. The boundless nature of wireless communication allows passive and active attackers to collect personal data, if such information is not protected. In the case of WSN, sensed data leakage may permit to gather information about people in the sensors environment. A feasible solution is to anonymize

information restricting the sensor network's ability to collect data at a detail level that could compromise privacy [49]

However, all these requirements point to the well known *CIA triad*, i.e. *Confidentiality, Integrity and Availability*.

In section 4, the main security attacks in WSNs are described. Some of the relevant countermeasures will be described in sections 6 and 7.

4 THREAT ANALYSIS

This section focuses on a layer based classification of the security and privacy attacks in wireless networks with a special focus on wireless sensor networks.

There are several types of attacks on wireless networks, and many ways of classifying these attacks. One of the most familiar ways of classifying them is based on the effects of the attacks. From this point of view, attacks can be classified into passive and active attacks [50] [51]. A passive attack attempts to learn or to make use of information from the system but it does not affect the system resources. An active attack attempts to alter system resources or affect their operation. These two broad classes are subdivided into other sub-types of attacks as indicated in the two next subsections.

4.1 Passive attacks

Wireless networks are passive to eavesdropping due to the nature of the wireless communication medium. Interception of radio carriers and of the data contained in them is usually considered unavoidable if an attacker is eavesdropping the radio spectrum of a victim user. Therefore, attacker models for wireless networks always assume that an attacker can perform a passive attack.

The passive attacks are in the nature of eavesdropping on, or monitoring of transmissions [51]. In this kind of attacks, an unauthorized attacker monitors or listens to the communication between entities in a system and gains access to an asset but is not able to modify its contents. Passive attacks can be either eavesdropping [50] (sometimes called release of message contents), or traffic analysis [51]. These two passive attacks are described below.

Eavesdropping communication channels and store of the captured data always precedes traffic analysis. Traffic analysis is a powerful method that can be exploited in the context of both security and privacy. Traffic analysis can be used to discover cipher keys used and bypass authentication in legacy security modes of IEEE 802.11 networks [52, 53].

A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information [51]. With eavesdropping, attackers listen to or monitor the transmissions of message contents. An example of this attack is a person listening to the transmissions on a LAN between two workstations or tuning into transmissions between a wireless handset and a base station [50].

The traffic analysis attack is the process of intercepting and examining messages in order to deduce information from patterns in communication [51]. The attacker, in a more subtle way, gains intelligence by monitoring the transmissions of patterns of communication. A considerable amount of information is contained in the flow of messages between communicating parties [50].

Note that even if data is encrypted, the attacker might still be able to observe the patterns of communications. For example, the attacker could determine the location and identities of communicating hosts and observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that is taking a place [51] and particularly who is communicating with whom.

Besides that, passive attacks are sometimes difficult to detect because they do not involve any alteration of the data. However, it is feasible to prevent the success of these attacks, usually by means of encryption and anonymous communication techniques. Thus, the emphasis, in dealing with passive attacks, is on prevention rather than detection [51].

4.2 Active attacks

Active attacks in wireless networks include the same set of attacks against security services on hardwired networks plus a set of attacks that are specific to ad hoc networks. In this section we list active attacks in ad hoc network environments. These attacks are classified in two groups. The first group lists attacks that are common to hardwired network and ad hoc networks, with focus on some specific vulnerabilities found in wireless networks. The second group lists attacks that are specific to ad hoc networks or otherwise are very uncommon in hardwired networks.

Active attacks involve some modifications to a message, data streams, or files. They also involve the creation of a false stream [50]. Globally, we can distinguish four types of active attacks: replay, masquerade, modification of streams, and denial of service (DoS). These attacks are briefly described in the next subsections.

The following active attacks are common to both hardwired and ad hoc networks [54, 51]:

- *replay attacks* — these attacks involve capturing, storing and retransmission of a message or a sequence of messages. Replay attacks often prelude to other security attacks. Wireless networks are highly susceptible to replay attacks, as messages are transmitted “over-the-air” and are, thus, susceptible to be intercepted and replayed.
- *masquerade or impersonation attacks* — masquerade or impersonation attacks occur when one entity pretends to be a different entity. Unprotected or weak authentication mechanisms usually lead to this security threat, as message sequences can be replayed and data link addresses can be easily spoofed in wireless networks [55]. Man-in-the-middle (MitM) attacks often prelude impersonation attacks. An impersonation attacks in wireless local area networks is the result of a man-in-the-middle attack caused by flaws in tunnelled authentication mechanisms [56]. Impersonation attacks can also lead to privacy threats.
- *message modification attacks* — message modification attacks happen when a message or a sequence of messages are captured or intercepted, altered and retransmitted. Intentional delaying and message reordering are also considered to be modification attacks. In order to prevent this kind of security attack, data integrity must be guaranteed. Protection against modification attacks is essentially based on the same suite of protocols in wireless and conventional networks. However, mobile ad hoc networks are more susceptible to message modification, as data can be relayed by every node, trusted or not, in the wireless network.
- *denial of service (DoS) attacks* — DoS attacks prevent or inhibit the service provisioning by a device. DoS attacks can be launched in different layers of the TCP/IP stack. Wireless networks are particularly vulnerable to physical layer DoS attacks. The disruption of a wireless networks involves jamming the frequency range being used by wireless communication. Since the spectrum range, encoding schemes and frequency patterns are standardize in every civilian communication system, such as in IEEE 802.11 [57], IEEE 802.15 [58, 59, 60] and IEEE 802.16 [61] technologies, DoS attacks in the physical layer are feasible by any resourceful attacker.

All the aforementioned attacks exploit the unbounded and shared nature of the wireless communication. Replay, masquerade and message modification attacks are preceded by eavesdropping data traffic in the wireless network. The following list presents active attacks that are specific to ad hoc networks and their variants or are potentially more harmful and more difficult to protect against in ad hoc networks environments than in hardwired networks:

- *attacks on routing mechanisms* — routing protocols designed for ad hoc networks are usually vulnerable to a set of attacks aiming to influence or interfere on data communication flows in an ad hoc network. Attacks over routing protocols poison routing tables with erroneous or incorrect information. The goal of such attacks is to cause communication disruption, and/or to logically isolate a device from the rest of the network, DoS, or for gathering data for future traffic analysis. Attacks against ad hoc routing protocols often try to build wormholes or set sinkholes in the ad hoc network:
 - *wormholes* consist of bidirectional tunnels in an ad hoc network that are used to forward packets, including routing control messages, from one geographical location of an ad hoc network to another distant location. Setting a wormhole needs the cooperation of two or more colluding nodes in the ad hoc network. Wormholes makes the logical topology of an ad hoc network not to reflect the actual physical topology, with undesired effects on routing protocols [62].

- *sinkholes*, also called *blackholes*, are malicious devices that lure other nodes to forward traffic through it, usually sending false routing control messages and thus manipulating the ad hoc routing table of other nodes in the proximity [63]. A device acting as a sinkhole can either capture and store the forwarded traffic for future traffic analysis, can selectively drop packets, e. g., forward only control packets but no data packets [64], or can simply block all network traffic.

Attacks against ad hoc routing protocols usually depend on the operation details or operation mode, i. e., PMP or RMP, of each particular routing mechanism. Examples of such attacks are the Byzantine, the rushing attacks and flooding of specific routing control messages.

The Byzantine attack is a family of attacks that involves any authorized device or set of authorized devices in an ad hoc network to cause routing service disruption or degradation [65, 66]. A Byzantine attack is deployed by devices that present a Byzantine behavior. An example of a Byzantine attack directed to service degradation in ad hoc networks is the jellyfish attack. A jellyfish attack consist of one or more nodes in an ad hoc network that maliciously batch and delay packets in an ad hoc network [67].

A rushing attack is an attack designed for RMP and consists of sending multiple route requests and more quickly than the other devices in the ad hoc network in an attempt to force other devices to include a hop through the attacker [64]. Byzantine attacks and rushing attacks can be used to build wormholes and set sinkholes in an ad hoc network.

Some ad hoc routing protocols use HELLO messages, a specific routing control message, to discover neighbor devices. A HELLO flood occurs when an attacker that has a radio device with more transmission power than the other network nodes can broadcast routing control messages (or replay messages from other devices) to nodes that are located geographically far away in the ad hoc network, i. e., two or more hops away, from the attacker [63]. Thus, under a HELLO flood attack, the logical network topology is inconsistent with the physical network topology.

Many other attacks may be performed against routing availability [46]:

- *Spoofed, altered or replayed routing information*. An adversary may inject bogus routing information trying to disrupt routing availability.
 - *Selective forwarding*. An attacker may selectively drop messages in order to create black holes or block all traffic from the victim node. This attack is more effective when the adversary is included in the path of the targeted data flow.
 - *Acknowledgement spoofing*. An attacker can send spoofed link layer acknowledgements to convince the victim that a dead node is alive.
 - *Neglect and greed*. A compromised or malicious sensor node can randomly neglect to route some messages and give undue priority to its own messages (greedy).
 - *Misdirection*. An attacker can forward messages along wrong paths, perhaps by fabricating malicious route advertisements [68]. This DoS attack can target the sender, diverting traffic away from its intended destination; it can also misdirect many traffic flows in the direction of an arbitrary victim.
- *Sybil attacks* — a Sybil attack [69] is an identification attack and occurs when a malicious user influences the network by controlling multiple logical identifiers from a single physical device. In a Sybil attack, malicious users assume multiple identities, preventing the usage of security mechanisms based on filters, reputation or trust assumptions. This attack was first identified in peer-to-peer networks, but can also be used to disrupt ad hoc networks, including ad hoc routing protocols. Sybil attacks have deep implications in the general security expectations of wireless network. This attack is also strongly correlated to privacy issues in wireless networks.

- *battery exhaustion attacks* — battery exhaustion attacks are a variant of DoS attacks. They are sometimes referred as sleep-deprivation attacks [70]. Wireless ad hoc network devices are usually mobile devices that are battery-driven, i. e., they depend on battery to remain on. Thus, mobile devices are susceptible to battery exhaustion attacks. The attacker aims to exhaust the battery power of a target device and render it useless by forcing it to receive, transmit or process data that this device would not witness in a normal situation.
- *Collision* — a Dos attack performed at data link layer. An attacker is able to disrupt an entire message simply inducing a collision in one octet of a transmission and exploiting properties of the MAC protocols employed. It is worth observing that intermittent collision and exhaustion attacks or abusing MAC priority schemes can lead to unfairness.

Besides that, passive as well as active attacks can be launched at or target different layers of the Internet model.

4.3 Attack Classification by Network Layers

The following section summarizes the security and privacy attacks on each layer of the Internet model [71].

The threats to informational privacy in wireless networks are the same that exist in other computer systems. However, some of the characteristics of wireless networks, i. e., the shared physical medium used in wireless communications, the limited physical security of mobile devices, and eventually the lack and independence of an online infrastructure in the case of wireless ad hoc networks, contribute to make wireless networks more vulnerable to privacy infringements than their hardwired counterparts. Applications can leak vast amounts of possibly sensitive data if being transmitted among the participating mobile devices. In addition, traffic information generated inside such networks can potentially reveal sensitive data about ad hoc network users and their communicating partners. Moreover, ad hoc network devices can be geographically pinpointed by non-authorized parties causing location privacy threats. Malicious users may even track users by following beacon signals emitted by mobile devices, such as neighborhood discovery messages present in some ad hoc routing protocols. Finally, the personal data collected in an ad hoc network can be used to build user profiles that include the history of communicating partners and current and past geographical locations. It is fundamental for an attacker whose objective is to profile users in a wireless network to uniquely identify devices and also to recognize distinct occurrences of a same device in different instants, i. e., gather historical data regarding an individual device. Thus, an attacker has obtained a unique identifier from each device in a wireless network. Unique identifiers can be obtained from different sources in a wireless network device, from physical to application layers. Thus, to identify potential threats to privacy in wireless networks, it is necessary to list possible sources of identifiable data that can be used by an attacker. The TCP/IP stack organization is used as support to list sources of identification in an ad hoc network, and existing threats to these identifiers. In the following subsections, we use a bottom-up approach to identify potential threats to privacy in wireless networks, i. e., from the physical layer to the application layer.

4.3.1 Physical Layer

Wireless communication is broadcast by nature. A common radio signal is easy to jam or intercept. An attacker could overhear or disrupt the service of a wireless network physically.

- *Eavesdropping*: Eavesdropping is the intercepting and reading of messages and conversations by unintended receivers. The majorities of wireless communications use the RF spectrum and broadcast by nature. Signals broadcast over airwaves can be easily intercepted with receivers tuned to the proper frequency. Thus, messages transmitted can be overheard, and fake messages can be injected into the network.

- *Interference and Jamming*: Radio signals can be jammed or interfered with, which causes the message to be corrupted or lost. If the attacker has a powerful transmitter, a signal can be generated that will be strong enough to overwhelm the targeted signals and disrupt communications. The most common types of this signal jamming are random noise and pulse. Jamming equipment is readily available. In addition, jamming attacks can be mounted from a location remote to the target networks.

Physical layer attacks against privacy aim either to discover the geographical location of a device in a wireless network or to identify patterns in the emitted radio frequency (RF) signals that can be uniquely associated to a given device. RF triangulation and fingerprinting are two techniques that can be used to uniquely identify a device in a wireless network.

RF triangulation is a technique used to pinpoint the geographical location of a given device. This technique requires the deployment of passive devices in the wireless network that are able to collect signal strength information of RF emitted by a target device. The location of the sensors is assumed to be known. By combining the data collected by the sensors it is possible to determine the geographical location of the target device. Access points in IEEE 802.11 networks can be used as sensor nodes for RF triangulation, as presented in [72, 73], and produce results with errors in the order of meters. RF triangulation can effectively locate the position of a transmitting device, but lacks the ability to link historical information to identify multiple appearances of a specific device [74].

RF fingerprinting is a general umbrella term for different techniques involving the analysis and identification of unique characteristics in the RF emission by a transmitting node. The perceived signal to noise (S/N) ratio can be used as unique temporal characteristic to identify a device [75]. Signal processing and profiling is RF fingerprinting technique that can be used to discriminate RF emitted from different wireless ethernet cards based on signal fragments [76]. Transient signal detection and analysis is concerned with the characteristics of transmitted RF signal during the transient period, i. e., the start-up period prior to the actual transmission. The radio transmission during the transient period has consistent features, such as amplitude and phase components, that are not easily forged, yet not necessarily unique [55]. Modulation domain techniques compares received signals to the expected ideal in the modulation domain and are used to identify specific transmitters. Modulation domain techniques require previous knowledge of the modulation scheme being employed [74]. This requirement is hardly an hindrance, since modulation schemes are standardized and public.

RF fingerprinting is particularly useful to detect devices that deliberately change their hardware address information in attempt to prevent tracking. Transient signal analysis, signal processing, transient detection and modulation techniques rely on the fact that transceivers are not exactly equal. Hardware imperfections in the transceivers create unique radio characteristics that enable devices to be uniquely identified. Eliminating such imperfections during manufacture is possible, but would be hardly economically viable [74].

Another class of attacks performed at physical layer relates to *tampering*. In fact, an adversary can tamper with sensor nodes physically and open individual sensors in order to steal sensitive data and cryptographic key to gain access to sensor network (*node compromise*). The sensor nodes could be damaged as well. In this case, node destruction can be hardly distinguished from benign node failure. It is worth noting that tampering is strictly related to node compromise, thus it can be considered a feasible attack on availability, but also against secrecy, authentication and service integrity.

4.3.2 Data Link Layer

Attacks may target the link layer by disrupting the cooperation of the layer's protocols. Wireless medium access control (MAC) protocols have to coordinate the transmissions of the nodes on the common transmission medium. Because a token-passing bus MAC protocol is not suitable for controlling a radio channel, IEEE 802.11 protocol is specifically devoted to wireless LANs. The IEEE 802.11 MAC protocol uses distributed contention resolution mechanisms for sharing the wireless channel. The IEEE 802.11 working

group proposed two algorithms for contention resolution. One is a fully distributed access protocol called the distributed coordination function (DCF). The other is a centralized access protocol called the point coordination function (PCF). PCF requires a central decision maker such as a base station. DCF uses a carrier sense multiple access/collision avoidance protocol (CSMA/CA) for resolving channel contention among multiple wireless hosts.

Current wireless MAC protocols assume cooperative behaviors among all nodes. Obviously the malicious or selfish nodes are not forced to follow the normal operation of the protocols. In the link layer, a selfish or malicious node could interrupt either contention-based or reservation-based MAC protocols. A malicious neighbor of either the sender or the receiver could intentionally not follow the protocol specifications.

Data link layer attacks against privacy involve identifying and tracking unique characteristics that exist in this layer. The standard identifier in this layer is the hardware (MAC) addresses. Hardware addresses are assigned by the manufacturer and their intent is to uniquely identify a network interface card in a local area network. Hardware addresses are the easiest and most feasible way to track a wireless device. Nevertheless, these addresses can also be easily changed by software.

In addition, there are further techniques to identify devices using data link information. The sequence number information that exists in the IEEE 802.11 header can be used to detect MAC address spoofing by identifying gaps in the sequence number of the frames transmitted by a device [77]. This feature could be also potentially used to detect MAC address changes or a device using multiple MAC addresses. Another technique is to identify differences in the implementations of the active scanning algorithm in IEEE 802.11 wireless network card drivers. This technique is nonetheless limited since it cannot make any distinction between two devices running the same driver. Other limitations that can thwart the device driver fingerprinting also include driver code modification and noise generation (see for instance [78]).

4.3.3 Network Layer

Network layer protocols extend connectivity from neighboring 1-hop nodes to all other nodes in wireless networks. The connectivity between mobile hosts over a potentially multi-hop wireless link relies heavily on cooperative reactions among all network nodes.

By attacking the routing protocols, attackers can absorb network traffic, inject themselves into the path between the source and destination, and thus control the network traffic flow.

The traffic packets could be forwarded to a non-optimal path, which could introduce significant delay. In addition, the packets could be forwarded to a nonexistent path and get lost. The attackers can create routing loops, introduce severe network congestion, and channel contention into certain areas. Multiple colluding attackers may even prevent a source node from finding any route to the destination, causing the network to partition, which triggers excessive network control traffic, and further intensifies network congestion and performance degradation.

The standard identifier in the network layer is the IP address. IP addresses are logical addresses that can be either static or dynamic. Static means that the IP address is configured locally, while dynamic means that an IP address is assigned to a device by a central server. IP addresses can be easily modified by software, nonetheless. IP is a standard routed protocol of the TCP/IP suite and is used by routing protocols to select to which node a packet must be forwarded.

At the time of writing, there are no standards for IP address assignment in mobile ad hoc networks. The Ad Hoc Network Autoconfiguration (autoconf) IETF WG [79] is responsible to develop an auto-configuration mechanism for configuring unique local network addresses in mobile ad hoc networks. Nevertheless, ad hoc routing protocols are based on IP routing.

Privacy threats in the network layer include the tracking of devices using the IP address as unique identifier and ascertaining about the linkability between two communicating devices, i. e., a violation of relationship anonymity, by analyzing the network data traffic and dissecting the *source* and *destination* fields of an IP packet. The standard ad hoc routing protocols AODV [80] and DSR [81] leak the IP addresses of sender and destination during their path discovery phase, for instance.

The Internet Control Message Protocol (ICMP) [82] can be used for active fingerprinting based on the clock skew of the target device (clock skews are further explained in the next section regarding privacy threats in the Transport Layer) [83]. There are two requirements for the success of ICMP fingerprinting: the implementation of the TCP/IP stack of the target device must answer ICMP Timestamp Request messages¹; and target device should maintain its system time using the Network Time Protocol (NTP).

In comparison to physical and data link privacy threats, threats in the network layer have a significant difference regarding the attack range, i. e., the geographical area affected in an ad hoc network. The attacker in the latter case needs only to be part of the path linking the source to the destination, and not necessarily in the radio range of the target device.

4.3.4 Transport Layer

The objectives of TCP-like Transport layer protocols in wireless networks include setting up of end-to-end connection, end-to-end reliable delivery of packets, flow control, congestion control, and clearing of end-to-end connection. Similar to TCP protocols in the Internet, the mobile node is vulnerable to the classic SYN flooding attack or session hijacking attacks:

- *SYN flooding attack*: The SYN flooding attack is a denial-of-service attack. The attacker creates a large number of half-opened TCP connections with a victim node, but never completes the handshake to fully open the connection.
- *Session hijacking*: Session hijacking takes advantage of the fact that most communications are protected (by providing credentials) at session setup, but not thereafter. In the TCP session hijacking attack, the attacker spoofs the victim's IP address, determines the correct sequence number that is expected by the target, and then performs a DoS attack on the victim. Thus the attacker impersonates the victim node and continues the session with the target.

Transport layer information can be used to fingerprint network devices by analyzing the clock skew information [83]. The underlying assumption of this attack against privacy is that different devices have different clock skews, and a given device has a constant clock skew in general. Thus, it is possible to an attacker to retrieve and collect a target's perceived time information from the 32-bit timestamp field present in the TCP header. The TCP timestamp option was introduced in the RFC 1323 [84]. Results reported in [83] using passive and semi-passive attacks in different scenarios (include a non ad hoc wireless scenario) suggest that clock skew estimation is in general independent of topology and distance between targets and attacker devices. Attackers do not necessarily have to be in the radio range of the target device when deploying a transport layer fingerprinting, and it is enough to be part of the path connecting the sender to the recipient.

4.3.5 Application Layer

The application layer communication is also vulnerable in terms of security compared with other layers. The application layer contains user data, and it normally supports many protocols such as HTTP, SMTP, TELNET, and FTP, which provide many vulnerabilities and access points for attackers. The application layer attacks are attractive to attackers because the information they seek ultimately resides within the application and it is direct for them to make an impact and reach their goals:

- *Malicious code attacks*: Malicious code, such as viruses, worms, spywares, and Trojan Horses, can attack both operating systems and user applications. These malicious programs usually can spread themselves through the network and cause the computer system and networks to slow down or even damaged.

¹The MAC OS X 10.3 Panther does not reply to ICMP Timestamp Request

- *Repudiation attacks*: In the network layer, firewalls can be installed to keep packets in or keep packets out. In the transport layer, entire connections can be encrypted, end-to-end. But these solutions do not solve the authentication or non-repudiation problems in general. Repudiation refers to a denial of participation in all or part of the communication.

Information encapsulated in the application layer can eventually identify the sender and/or recipient of a message, expose the communication relationship between sender and recipient, or other personal data contained in the message payload. The information collected in the application layer is highly dependant of the application itself, e. g., sender and recipient fields of Simple Mail Transport Protocol (SMTP) message envelope and the input data generated by the user. Exactly as network and transport layer fingerprinting, attackers do not necessarily have to be in the radio range of the target device to analyze application layer information. It is enough to be part of the path connecting the sender to the recipient. Furthermore, application layer data is end-to-end information, i. e., the recipient of the message is guaranteed to be the final recipient, and not just an intermediary (proxy) device.

4.3.6 *Multi-layer attacks*

Some security attacks can be launched from multiple layers instead of a particular layer. Examples of multi-layer attacks are denial of service (DoS), man-in-the-middle, and impersonation attacks:

- *Denial of service*: Denial of service (DoS) attacks could be launched from several layers. An attacker can employ signal jamming at the physical layer, which disrupts normal communications. At the link layer, malicious nodes can occupy channels through the capture effect, which takes advantage of the binary exponential scheme in MAC protocols and prevents other nodes from channel access. At the network layer, the routing process can be interrupted through routing control packet modification, selective dropping, table overflow, or poisoning. At the transport and application layers, SYN flooding, session hijacking, and malicious programs can cause DoS attacks.
- *Impersonation attacks*: Impersonation attacks are launched by using other node's identity, such as MAC or IP address. Impersonation attacks sometimes are the first step for most attacks, and are used to launch further, more sophisticated attacks.
- *Man-in-the-middle attacks*: An attacker sits between the sender and the receiver and sniffs any information being sent between two ends. In some cases the attacker may impersonate the sender to communicate with the receiver, or impersonate the receiver to reply to the sender.

5 ATTACK TREE AND ATTACK GRAPH

Attack trees and their evolution, attack graphs, are useful methodologies of malicious threat description. In an attack graph the attacker possible actions are described and associated in a causal form. The overall meaning of an attack graph instance is the description of atomic steps needed to breach the system security. Attack graphs appear as a proper way to perform a semi-automatic security evaluation. The key point in attack graph adoption is the definition of the “atomic step” semantic. Different attack graph techniques essentially differ in this aspect.

In following sections we introduce first attack trees for historical reason, and then we will analyze attack graphs that are a generalization of attack trees.

This work is a preparatory analysis to identify and develop attack graph techniques targeted specifically on wireless sensor networks.

5.1 High-level attack trees

The first introduction of *attack tree* term was by Bruce Schneier in the notable book *Secrets and Lies: Digital Security in a Networked World* [85]².

The attack tree presented here is a human-aiding visual technique, neither formally generated nor formally analyzed. It provides a methodological way to study attack possibilities. Starting from the leaves, an attack tree describes the steps an attacker has to exploit to achieve a specific result. An attack tree may have any depth. In case of tree depth greater than two, multiple steps must be taken to carry out an attack. Usually, this means that an attacker has to exploit multiple vulnerabilities available in the system. Relations between siblings can be AND or OR. OR nodes are alternatives, each of the nodes in OR relation can be exploited to go up toward the root. AND nodes represent vulnerabilities that must all be accomplished to go up toward the root. E.g. to decipher an encrypted message, a malicious attacker has both to obtain the encrypted message and apply the cipher key of the encrypted message.

High-level attack trees are not machine-readable, and they can become not human-readable too: for large system, they can be largely complex and error-prone hand-made graphs. A full attack tree may contain hundreds or thousands of different paths all leading to completion of the attack. In spite of this possible complexity, these trees are very useful for determining what threats exist and how to deal with them in human discussions.

5.2 Predecessors of attack tree techniques

Attack trees stem from previous formalization efforts.

Reliability block diagrams, usually adopted in large network analysis represent a system as composed by many interconnected components. The components are modelled with statistical properties, like mean-time-between-failure, and the system behavior is simulated starting from single-component characteristics and properties of the component linking connection [86, 87].

Fault trees are acyclic graphs (trees), in which the root is the system of interest, leaves are single components and inner nodes, i.e., nodes between the root and the leaves, are “logic gates”, able to model the failure flow from the leaves to the root. If a flow from leaves to root is established, then a failure occurs. These systems are well understood and general enough to be applied in hardware, software and humanware in complex computer-based systems [88].

In late nineties, a branch of formal dependability techniques focused on security aspects. The techniques developed by Ritchey and Ammann [89] analyze network attacks, organizing such attacks in multistep paths, exactly in the same way an attack tree does, so, even if the paper does not use the relatively new words *attack-tree* or *attack-graph*, it is an example of *ante litteram* formal attack graph.

²a more older note of this term, in more introductory way can be found in the Schneier’s “Dr Dobb’s” page <http://www.schneier.com/paper-attacktrees-ddj-ft.html>

5.3 Model-based attack tree and attack graph construction

A problem stemming from the early approach of attack trees and attack graphs was the poor automatization of the process. Originally, attack trees was developed by humans to humans, neither with strict semantic nor with constrained scope.

Later works highlighted the needs for a semi-automatic use of attack trees, thus they mitigated the problem by introducing formal techniques and they expanded the analysis to attack graphs as well.

The needs in this case are of a formal description of the system, a formal description of threats, an engine able to interpret these descriptions and automatically bring the analysis of threat propagation in the system (model).

A recent exemplifying approach is described in [90]: it is adopted in the DESEREC [91] European project.

DESEREC uses a set of Models to formally describe the system.

- The *service model* describes the structure of the business services and their security constraints. It is useful to automatically identify the assets at business service level, to derive their functional and non-functional dependencies (security and QoS mainly), and to characterize the associated threats.
- The *resource model* models the system infrastructure (hardware and software elements and their interconnection) and associated capabilities (CPU/communication resources, security mechanisms).
- The *policy model* describes the constraints to apply when configuring the system infrastructure to provide the target business services. Its importance related to risk becomes clear after discussing the generation of system configurations.
- The *configuration generation tools* semi-automatically configure the system infrastructure to provide the given business services, as prescribed by the service, resource and policy models.
- The *analysis tools* are in charge to automatically verify the alternative system configurations (independently one by one) against large sets of possible threats. In risk terms, they are related to threats identification, and to vulnerability characterization. Several analysis technologies may be integrated, spanning from system simulation to attack graphs analysis.
- The *scenario generation tools* play the main role in computing global risk indicators and detection and reaction rules starting from the analysis tools results. The end scenario is represented by a *risk graph* (RA). A RA is an attack graph with augmented semantic. RA nodes can represent attack steps as well as side-effects. Information on side-effects is used to understand the detectability of on-going attack. Edges represent causal relations among nodes as well as in common attack graph.

In approaches like this, attack graphs are exploited both for system analysis and for decision making (e.g. how to react in presence of a specific problem).

Of course, other formalization approaches are possible. Since an attack is a violation of a safety property, model checking is a suitable way to produce attack graphs automatically: a successful attack path is a counterexample produced by the model checker [92].

Model checking techniques are very interesting: since the attack graph may become too large and complex, the approach to construct them by hand is a strategic error which leads to error-prone results, especially for large systems. A Model based approach is needed to cope with present-day threats. Model checking makes it possible to demonstrate that the attack graphs produced are *exhaustive*, i.e., they cover all possible attacks, and *succinct*, i.e., they contain only relevant states and transitions. Both of them are properties not easy to achieve by other deployed techniques.

5.4 Multi-semantic attack tree

How to represent the information in an attack tree is an open issue, and this has originated a plethora of attack tree variants.

In some works, the attack tree divides atomic steps in semantically different types.

Kotenko and Stepashkin [93] developed the approach to vulnerability analysis and security level assessment of computer networks, intended for implementation at various stages of a life cycle of analyzed computer systems.

The approach is based on the mechanism of automatic construction and replaying of the distributed attacks scripts. Known attacks are combined, taking into account various intentions and experience level of attackers. The results of attacks allow to calculate different security metrics that can be used to define the common security level of a computer network (system) as well as security levels of its components, by specific algorithms and metrics described in their work. Such approach can be used at different stages of the computer network (system) life cycle, like design and exploitation stages. At *design stage*, the tool operates with the model of the analyzed computer network. At exploitation stage the tool interacts directly with the computer system, sending and receiving real packets. To create the tree in a semi-automatic fashion, the system requires as input the attacker skill level (and a mode of action), the attacker goal, a knowledge base describing the system (like OS version and type, network connection) and a database of common attacks (exploits). Using graph methods they extract a huge amount of security metrics. A notable aspect of this work is that they differentiate between *reconnaissance actions* (to understand the network topology) and *attack action* (to compromise the network objectives). This allows to model the knowledge an attacker can likely gain, instead of considering this as a predefined parameter.

Different semantics of nodes can be made explicit by introducing different graph to complement the attack graph. In [94] the authors adopt a vulnerability-based approach to build up an attack tree. The method relies on the knowledge of:

- vulnerabilities of components and links and corresponding exploitability easiness
- vulnerability dependencies among system components
- possible attacks (composed by exploitation of multiple vulnerabilities).

In particular, they make the vulnerability dependencies explicit by building a *dependency graph* (DG). The exploitation of a vulnerability (e.g. V1) can become more or less easy after the exploitation of another one (e.g. V2). The DG allows the evaluation of these changes in the vulnerability “exploitation easiness”.

A semantically rich attack graph enable subtle conclusion on the security of a WSN. However, it is appropriate to include as much “features” as strictly required by the security aspects to highlight. A problem in attack graph automatic evaluation is the complexity explosion, thus making nodes and computations efficient is a desired result.

5.5 Attack graph scoring

As Lord Kelvin wrote in the late 19th century, “when you can measure what you are speaking about and express it in numbers, you know something about it” [95].

In this section we present some proposals, keeping in mind that it is hard to distill useful security metrics from attack graphs.

5.5.1 Vulnerability enumeration

A possible scheme for assessing a certain property of an object is to explore all the possible flaws and threats against that property. This scheme is quite general and used in several areas. When applying this scheme to attack trees, the first step is to fill in a list, as complete as possible, of existing vulnerabilities and attacks against an information system (IS). A second step is to assess whether the IS under study has or does not have a response capability against these vulnerabilities and attacks. This would allow to generate an attack tree specific for the system under examination, in automatic fashion.

Since a complete list is not practically feasible to obtain, instruments to support this task exist: vulnerability scanners and penetration testing tools.

A penetration test is an attempt designed to expose security holes inside and outside the network system perimeter, more precisely to demonstrate that exploitable vulnerabilities exist related to the network infrastructure under exam.

The test can cover all security critical components and known vulnerabilities, configuration errors, flawed or illegal software and patch absence, but its scope can be limited to only partial aspects of the system to make it simple to use and fast to perform.

Many good open-source vulnerability scanners are available in the Internet [96] [97] [98] [99] and many other in [100]. Almost all of them are compatible with the CVE list [101].

The Common Vulnerability and Exposure (CVE) [101] is the most widely adopted tool to address vulnerability duplications. It differs from vulnerability databases, since it does not carry very comprehensive vulnerability information. Instead, its purpose is to be a dictionary of vulnerabilities, or, in even better terms, a list that univocally identifies vulnerabilities. Its aim is the standardization of the names for all publicly known vulnerabilities and security exposures, so that different databases or tools can refer to common identifiers. The status of the entry within CVE can be assigned, candidated, or rejected. The list of CVE names is free to use, and only subject to very mild terms of use.

The CVE technique solves an important problem: Since many vulnerability databases exist, the risk of a high volume of vulnerabilities re-discovery exists as well, leading to the presence of the same vulnerability in multiple places with different names, or to a waste of time or to both of them. The presence of the same vulnerability in multiple places may cause a number of problems, such as erroneous product evaluation (caused by a wrong number of vulnerabilities associated), wrong security improvement planning, and scarce possibility to correlate information from multiple sources (i.e. different vulnerability databases).

Another important tool to adopt vulnerability estimation is the CVSS (Common Vulnerability Scoring System) [102] which is an attempt to give a quantitative scoring of vulnerabilities, provided some qualitative characteristics. CVSS was a joint effort including names in the IT field, such as CERT, Cisco, Microsoft, eBay, Symantec and others. A CVSS score is calculated in three steps, base, temporal and environmental. The base score is the only fixed one, while the temporal one evolves in time while exploit code and patches are developed, and the environmental one depends on the specific environment within which the affected software is used. Each of these scores requires a set (vector) of qualitative characteristics, and through a formula it permits to calculate the relative quantitative score. Here is an example for the base score:

Base Metric Formula

```

AccessVector      = case AccessVector of
                    local:          0.7
                    remote:        1.0

AccessComplexity = case AccessComplexity of
                    high:           0.8
                    low:            1.0

Authentication   = case Authentication of
                    required:       0.6
                    not-required:   1.0

ConfImpact       = case ConfidentialityImpact of
                    none:           0
                    partial:        0.7
                    complete:       1.0

```

```

ConfImpactBias  = case ImpactBias of
                    normal:          0.333
                    confidentiality:  0.5
                    integrity:        0.25
                    availability:      0.25

IntegImpact      = case IntegrityImpact of
                    none:             0
                    partial:          0.7
                    complete:         1.0

IntegImpactBias  = case ImpactBias of
                    normal:          0.333
                    confidentiality:  0.25
                    integrity:        0.5
                    availability:      0.25

AvailImpact      = case AvailabilityImpact of
                    none:             0
                    partial:          0.7
                    complete:         1.0

AvailImpactBias  = case ImpactBias of
                    normal:          0.333
                    confidentiality:  0.25
                    integrity:        0.25
                    availability:      0.5

BaseScore = round_to_1_decimal(10 * AccessVector
                                * AccessComplexity
                                * Authentication
                                * ((ConfImpact * ConfImpactBias)
                                  + (IntegImpact * IntegImpactBias)
                                  + (AvailImpact * AvailImpactBias)))

```

The CVSS system, whose aim is to propose *an open and universal vulnerability scoring system, with the ultimate goal of promoting a common language to discuss vulnerability severity and impact*, faces the problem of evaluating the vulnerabilities in a systematic and computable way. CVSS appears, at the present time, as the most formal, comprehensive, and widely applicable initiative proposed, so it is probably one of the best choices to rate actual weaknesses.

However, known bugs and viruses represent just a portion of all possible threats against an IS, therefore a vulnerability approach may be a useful starting point, but not exhaustive enough in evaluating security. Despite of its conceptual simplicity and logic, there is a fundamental lack in this kind of analysis: a penetration test, not differently from every software test, cannot demonstrate the absence of fault inside a network system, instead it can only expose an existing problem, so it is ineffective to make a good prediction, but can only make a sort of actual classification of the system.

The approach to start from weak points is good, but “weak points” have to be defined in some sort of different and more abstract manner (that is, independent from actual vulnerabilities discovered) to improve this method in its use and effectiveness.

5.5.2 Privilege graph

In his doctoral thesis, M. Dacier [103] developed a theoretical framework to provide security measurements, which was further developed in [104], [105] and [106].

This framework is based on modeling the system via a graph, the *privilege graph*, in which each node corresponds to a set of privileges owned by a user or a set of users (e.g., a Unix group). An arc from a node A to a higher node B (i.e. associated with higher privileges) means that there exists a vulnerability which allows a user associated with the node A to obtain the privileges represented by the node B . Once identified into the graph “insider” nodes, i.e. nodes of possible attackers (for instance nodes representing the minimal privileges of any registered user), and “target” nodes, then a security measure can be provided by computing the mean effort to be spent in order to cover a path between an insider node and a target one. Authors suggest to compute this mean effort (called METF=mean effort to security failure) for three profiles: To assess this measure, each arc in the privilege graph is assigned with a rate corresponding to the effort required from an attacker in order to perform the privilege transfer corresponding to this arc. A Markov chain has been chosen to describe the probability of succeeding in an attack as a function of the spent effort. In other words, the random variable describing such a probability is given by:

$$P(e) = 1 - \exp(-\lambda e)$$

where e is the spent effort and λ is the assigned rate for that attack. This is a Markov chain with mean $1/\lambda$, i.e., $1/\lambda$ is the expected effort to be spent to succeed in the attack. The METF (mean effort to security failure) is therefore calculated for each of the three proposed attacker profiles, listed below:

1. **SP (shortest path):** the attacker tries to reach the target through the shortest path. To evaluate security, the mean effort spent in the shortest path is calculated. However, this assumption means that the attacker knows in advance the whole graph topology, which may be unrealistic.
2. **TM (total memory):** at each step, the attacker may choose among the set of all possible attacks, including those from already visited nodes that he did not apply previously.
3. **ML (memoryless):** at each step, the attacker may choose one of the attacks, which can be issued from that node only.

Of course, the higher the METF means the better security. This evaluation method was validated in an experiment implemented by R. Ortalo, Y. Deswarte and M. Kaâniche [107]. In this experiment, a large distributed computer system of more than one hundred Unix workstations connected to a local area network had been observed during 13 months on a daily basis. In order to build the privilege graph, a tool was used for monitoring 13 among the most common vulnerabilities in Unix including password cracking with `crack` software, user-defined privilege transfer methods (`.rhosts`), incorrect path search allowing Trojan horse attacks and so on. Each time a vulnerability was detected, an arc was added in the privilege graph under construction. In the conclusions, the authors provide a useful comparison for *METF* for all profiles (SP, ML and TM). However, some security variations cannot be detected by such measures. Thus, the operative usefulness of these indicators as security evaluation should be improved.

The tools devoted to security analysis are usually vulnerabilities scanners (SATAN, COPS): they output a list of detected vulnerabilities, possibly sorted in different classes according to their severity. However, a new vulnerability - even with a high severity level - is not always correlated with a degradation of the overall security, as it is possible to see by comparing this measure with the METF. If we do not consider how a vulnerability is located in the privilege graph, this measure (like the previous one) may cause a significant number of false alarms. The better measure seems to be METF for the TM profile, because every increase of vulnerabilities (and consequently of paths) is revealed by a decrease of the METF, while other measures are not so dynamic and the meaning of their variations is not always univocal, so that their correct interpretation may be hard. Unfortunately, METF for the TM profile cannot be always computed due to the complexity of the algorithm, increasing with the path number. Another problem is that it takes into account only attacks coming from inside, i.e., from a user or someone who entered the system as a registered user, thus not very well suited in WSNs environment.

5.5.3 Risk analysis methodology

Risk analysis methodology are human-level tools to establish the security of a system. They are very general in scope, but basically they compare a specific threat model (usually composed by a solid Knowledge Base of threats) versus a semi-formal system model (usually depicting relations among *assets*) and define specific semi-formal rules of problem propagation.

Some of the most successful Risk Analysis approach follows:

- The CCTA Risk Analysis and Management Method (**CRAMM**) [108] is a methodology originally developed by Central Computer and Telecommunications Agency (UK's government). Three main stages are identified in CRAMM: *Assets identification and assessment*, in terms of *physical* (e.g. IT hardware, valued in terms of replacement costs), *software* (e.g. application packages), *data* (e.g. the information held on the IT system) and *location* assets that make up the information system; *Threat and vulnerability assessment*, that is an analysis about what are and how likely problems occur; *Countermeasures selection and recommendation*, helped by CRAMM by a large countermeasures library. CRAMM combines asset values, threats and vulnerabilities to evaluate risks. However, due to its considerable age (in IT terms) it shows a lack of flexibility and its results are not very easy to understand and apply [109].
- The Methodology for Information Systems Risk Analysis and Management (**MAGERIT**) version 2 [110] is a development of the Spanish public administration ministry, widely and freely available. The authors advocate simplicity and easy-to-understand tables as operative means for risk analysis. It permits both qualitative and quantitative risk analysis. The approach first of all identifies assets, then it identifies threats (a large database of threats has been developed), and thus determines the impact (that is, the damage extent inside the organization under exam) and then the risk (by means of a pre-established table that criss-crosses the impact versus the likelihood of the menace). After this step, the *safeguards* (countermeasures) are defined, in terms of the mitigation capability versus a threat (this could be in terms of threat impact reduction or threat frequency reduction). Then, finally, the residual impact and residual risks are evaluated. Even for safeguards a database of possibilities is defined. An aspect of interest is the existence of a (non-free) computer aiding tool that simplifies the risk analysis process.
- The Risk Management Guide for Information Technology Systems (**NIST SP 800-30**) [111], is a document addressing the risk analysis process, a recommendation for every federal agencies in the US. The process depicted is subdivided into steps: first of all, *system characterization*, the definition of system boundaries and components and data sensitivity and criticality; *threat identification*, the definition of a list of threats; *vulnerability identification*, that is, identification of enabling conditions for a threat exploitation; *control analysis*, the evaluation of the employed controls, in terms of maturity level; *likelihood determination*, on the base of current controls and threat agent motivation; *impact analysis*, in terms of confidentiality-integrity-availability; *risk determination*, on the base of the likelihood of the threat, magnitude of the impact, and current controls; *control recommendations*, where the output is a list of controls to adopt; and finally *result documentation*. An interesting point of the NIST methodology is a small knowledge base of possible threat agents, with possible motivations that could help in the determination of the event likelihood.

The more interesting achievements of risk analysis in our context are methods to evaluate a single attack step (e.g., on the base of the affected asset value, on the base of the vulnerability likelihood, ...) and methods for definition of relationships between vulnerabilities (again, often on the base of asset relationships).

Although the approach by itself is not suitable to automatic reasoning, works exist trying to automate some or many steps of it. For example, in [112] the issue on how to automate parts of risk analysis process is examined. Three steps within the risk analysis process are identified: the collection of threats data, the identification of threats applying to the target system, and the computing of risk measures. In that

paper some enabling requirements are reported. In particular, a set of models is needed as input. However some problems remain open, like practical issues about vulnerabilities that are publicly available, simulation and modeling of intruders, definition of reliable security metrics. Extensions of experimental frameworks like that one seems interesting for achieving good automatic reasoning and helpful automatic decision making in the context of WSN.

5.6 Attack tree complexity mitigation

Attack graphs usually face the problem of a complexity explosion. The check for a threat involves the set of present compromised facets (on the whole set of compromised components) checked against the pre-conditions necessary to limit the vulnerability. Therefore, it can be applied to small-sized networks only and it needs modification for large scaled networks.

Some previous works cope with complexity by splitting the original state space in smaller independent subsets. This can be performed considering sub-networks of the original computer networks, or specific threat subclasses (e.g. by analyzing only the threats affecting *confidentiality*).

By this approach, it is possible to exploit parallel computing on different subsets, and subsequently combine the results.

The development of various algorithms for proper state space generation usually needs improvement, in order to reduce the computational complexity of the approach. Heuristics are adopted. As a side-effect, these algorithms will possibly reduce the accuracy of the final result. A trade-off between accuracy and treatability will have to be investigated.

Another problem is the complexity of attack trees for human understanding and high-level analysis. To cope with attack graph complexity Noel and Jajodia [113] introduce techniques to help making complex attack graphs more understandable, and applying these techniques to the correlation, prediction, and hypothesis of attacks. Visualizing the attack graph by *adjacency matrix*, graph regularities (e.g. bottlenecks and densely-connected subgraphs) become apparent.

By extending the graph-clustering technique, it is possible to show multi-step reachability across the network, the impact of network configuration changes, and the analysis of intrusion alarms.

The introduced adjacency matrix represents edges as single matrix elements, and implicitly represents as matrix rows and column the vertices. To improve the visual efficacy of adjacency matrix, the authors adopt clustering techniques to reorder the matrix and make apparent graph properties. The matrix can be further manipulated (e.g. by computing higher power) to show reachability in a determined number of steps. By operation like distance calculation, projection of rows or columns and other simple operation on the matrix, it is simple to understand attack properties.

5.7 Conclusions

Attack tree techniques have been analyzed. Peculiar aspects and the meaning of attack trees and attack graph analysis have been described.

In the WSN scenario, analysis must focus on effects important in the very peculiar ambient: in a “low-powered” world, power consumption and computational load are very important parameters, much more than usual effects described in attack tree/graph analysis for “normal” systems. The existing approaches have to be adapted and extended paying attention to these aspects.

6 SELECTED SECURITY CONCEPTS FOR WIRELESS NETWORKS

While wireless networks become more complex and largely deployed, they are still too vulnerable and hence attacks become more frequent (cf. Section 4 on wireless attacks). Because of this evolution, wireless security should be modeled, implemented and well deployed. In this context, a well-founded security study should identify who has access to what, when and under which conditions.

The Common Criteria define an “organizational security policy” as: a set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organization in the operational environment [114]. Such an organizational security policy usually relies on an access control policy [115]. An access control model is often used to rigorously specify and reason on the access control policy (e.g., to verify its consistency). Note that the model does not specify how the security policy is enforced; the enforcement is realized by technical security mechanisms, such as credentials, cryptographic transformations (e.g., signature, encryption), access control lists (ACL), firewall rules, etc.

To prevent the risks and to provide an acceptable security level, we call on security measures. These security measures can be divided to organizational (e.g., security policies) as well as technical mechanisms (cryptographic-based mechanisms).

In fact, security policies define access control rules and security models are often associated to security policies to rigorously/formally specify these policies and manage their complexities. These security policies and models should be carried out by using security mechanisms such as firewall rules, Access Control Lists (ACL), Access Control Matrix, XML capacities), firewalls, encrypted communication channels, antivirus tools, etc.

The most important wireless security models and mechanisms will be presented in next subsections.

6.1 Security Models

6.1.1 *Discretionary and Mandatory Access Control*

Classical access control models (discretionary DAC and mandatory access control MAC [116]) are not really adapted to WSN. For instance, the HRU model, defined by Harrison, Ruzzo and Ullman in 1976, represents the relationships between the subjects, the objects and the actions by a matrix M . $M(s, o)$ represents the "actions" that subject s is allowed to carry out on object o [117]. It is thus necessary to enumerate all the triples (s, o, a) that correspond to permissions defined by the security policy. Moreover, when new entities are added to or removed from the system, it is necessary to update the policy.

6.1.2 *Role Based-Access Control*

Role Based-Access Control (RBAC) is more flexible. Roles are assigned to users, permissions are assigned to roles and users acquire permissions by playing roles [118]. Hierarchical RBAC adds a requirement for supporting the role hierarchies, while Constrained RBAC enforces the separation of duties. RBAC is unquestionably suitable for a large range of organizations. Indeed, if users are added to the system, only the instances of the relationship between the users and the roles are updated.

6.1.3 *Organizational-based Access Control*

The OrBAC [119] model is an extension of RBAC [118]. It enables a structured and abstracted expression of the policy: Subjects are abstracted in Roles (as in RBAC), Objects in Views (as in VBAC [120]), and Actions in Activities (as in TBAC [121]). Also, the specification of the security policy is completely separated from its implementation, so as to easily manage complexity.

In OrBAC, an organization is a structured group of active entities, in which subjects play specific roles. An activity is a group of one or more actions, a view is a group of one or more objects, and a context is defined as a specific situation that conditions the validity of a rule. The Role entity is used to

structure the link between the subjects and the organizations. In the same way, the objects that satisfy a common property are abstracted as Views, and Activities are used to abstract actions.

OrBAC rules can express positive/negative authorizations (permissions/interdictions), and obligations. Security rules have all the following form: Permission (org ; r ; v ; a ; c), Prohibition (org ; r ; v ; a ; c), or Obligation (org ; r ; v ; a ; c). Such a rule means that in context "c", organization "org" grants role "r" the permission (or the prohibition, or the obligation) to perform activity "a" on view "v".

OrBAC considers two different levels for the security policy: the abstract level and the concrete level. At the abstract level, the security administrator defines security rules through abstract entities (roles, activities, views) without worrying about how each organization implements these entities. At the concrete level, when a user requests an access, concrete authorizations are granted (or not) to him according to the concerned rules, the organization, the played role, the instantiated view / activity, and the current parameters. The derivation of permissions (i.e., instantiation of security rules) can be formally expressed as follows:

$$\begin{aligned} & \forall org \in Organisations, \forall s \in Subjects, \forall activ \in Activities, \forall o \in Objects, \forall r \in Roles, \forall a \in \\ & Actions, \forall v \in View, \forall c \in Contexts, \\ & \text{Permission (org, r, v, activ, c)} \wedge \\ & \text{Empower (org, s, r)} \wedge \\ & \text{Consider (org, a, activ)} \wedge \\ & \text{Use (org, o, v)} \wedge \\ & \text{Hold (org, s, a, o, c)} \\ & \Rightarrow \text{Is permitted(s, a, o)}. \end{aligned}$$

Which means, If a security rule specifies that “in “org”, role 'r' can carry out the activity 'activ' on the view 'v' when the context 'c' is True”, and “in “org”, 'r' is assigned to subject 's' ”, and “in “org”, action 'a' is a part of activity 'activ' ”, and “in “org” object 'o' is part of view 'v' ”, and “the context 'c' is True for the triple (org, s, a, o)”. Then subject 's' is allowed to carry out action 'a' on object 'o'.

Note that even if most of existing works on wireless security only take high level security policies into account, some recent papers uses RBAC and OrBAC in the context of wireless networks.

6.2 Basic Security Mechanisms

6.2.1 Cryptographic primitives

6.2.1.1 Encryption

Encryption is the process of obscuring information to make it unreadable without special knowledge. It consists in transforming a message (plaintext) to an encrypted message (ciphertext) by using an encryption algorithm (cipher) and an encryption key. The decryption transforms the ciphertext to the original plaintext by using a decryption algorithm (cipher) and a decryption key.

Symmetric-key algorithms use identical cryptographic keys for both decryption and encryption. This key represent a shared secret between the communicating parties. Up until 1976, all the encryption algorithms were symmetric. Because of their performances, symmetric algorithms are still used. TDES (Triple Data Encryption Standard), AES (Advanced Encryption Standard) and IDEA (International Data Encryption Algorithm) are examples of symmetric algorithms largely used [122][123][124][125]. Other terms for symmetric-key encryption are single-key, one-key and private-key encryption.

Inversely, in asymmetric encryption, encryption and decryption keys are different; moreover, knowing on of these keys, it is *practically impossible* to deduce the other one. Asymmetric encryption is also called public key encryption as each user has two keys, the encryption key (Ke) may be known publicly, while the decryption key (private key) Kd is only known by the person allowed to decrypt the cyphertext. RSA (Rivest, Shamir, Adleman) and ElGamal are two examples of the most popular and well-respected public key algorithms [126].

The disadvantage of symmetric-key algorithms is the requirement of a shared secret key, with one copy at each end. Since keys are subject to potential discovery by a cryptographic adversary, they need to be changed often and kept secure during distribution and in service. The consequent requirement to choose, distribute and store keys without error and without loss, known as key management, is difficult to reliably achieve. In order to ensure secure communications between everyone in a population of n people a total of $n(n-1)/2$ keys are needed. Very often these days, the much slower asymmetric algorithms are used to distribute symmetric-keys at the start of a session, then the higher speed symmetric-key algorithms take over (e.g., hybrid encryption as in PGP).

6.2.1.2 One-way hash function

A one-way hash function, also known as a message digest function, fingerprint function or compression function, is a mathematical function which takes a variable-length input string and converts it into a fixed-length binary sequence. The latter is a (relatively) small number called "fingerprint" (hash sum, hash value, or hash code) of the data. The main properties of a one-way hash function H are:

- hard to invert: knowing $H(M)$, it is *impossible* to calculate M ;
- no collisions: knowing $H(M)$, it is *impossible* to find M' different from M with: $H(M') = H(M)$;
- easiness: knowing a certain message M , it is fast and easy to calculate the hash sum $H(M)$.

Note that even if hash functions are based on cryptographic methods, they do not rely on any secret. MD5 and SHA-1, SHA-2 are examples of one-way hash function currently used.

6.2.1.3 Digital signatures

Digital signature is used to simulate the security properties of a signature in digital, rather than written, form. Signing applies a signing key K_s . While the verification of the signature uses a verification key K_v . Digital signature is generally used to guarantee the information's integrity. Indeed, if the text M is modified between the signature and the verification, the verification algorithm gives a negative response.

As for encryption, we can distinguish symmetric signature where $K_s=K_v$ from asymmetric ones, where K_v is a public key (everyone can verify the signature) while K_s is kept secret by the signatory. In this case, it should be *impossible* to deduce K_s from K_v .

A hash function can be used for symmetric signature: $\Sigma = H(K|M)$, where K is used for the generation and the verification of the signature. Actually, the signature function applies the hash function to the concatenation of the key and the text; similarly, the verification generate the signature (by using the text and the key) and compare it with the received signature. It is also possible to generate signatures by using symmetric algorithms (such as DES); as it is employed in Message Authentication Algorithms (MAC [127]). Note that symmetric signatures have some limits, in particular: as the key is shared between the signatory and the verifier, the *non-repudiation* property is not guaranteed (symmetric signatures are intended to enforce integrity); it is sometimes difficult to keep the key (shared by the signatory and the verifier) secret towards third parties.

Inversely, with asymmetric signature, the key used for generating the signature is only known by the signatory. The DSA (Digital Signature Algorithm), defined in the DSS (Digital Signature Standard) norm is an example of public key signature algorithm using a hash function. In DSS, the hash function is SHA. It is also possible to use public key algorithms such as RSA for generating and verifying signatures (on fingerprints of the original message). In this case, the public key K_p is used for the signature's verification and the private key K_s (kept secret by the signatory) is used for the signature's generation. Note that as it is desirable to have signatures with a fixed length (even for messages having different length), we generally sign a fingerprint of the message.

- Generation: $\Sigma = H(M)K_s$
- Verification: $H(M) = ? = [\Sigma]K_p$

6.2.2 Access Management

6.2.2.1 Firewalls

The most common mechanism to prevent external attacks on a network is to filter out potentially malicious network packets. Firewalls are designed to provide "policy-based" network filtering [128].

The firewalls are usually used between trusted entity and untrusted entity on the network (usually the global Internet) as a check-point to manage the legitimate transactions between them. A firewall is a logical object (hardware and/or software) within a network infrastructure which prevents communications forbidden by the security policy of an organization from taking place, analogous to the function of firewalls in building construction [129]. The firewall is a part of an overall security policy that creates a perimeter defense designed to protect the information resources of the network or hosts on the network [130].

6.2.2.2 Virtual Private Networks

A virtual private network (VPN) is network in which some of the links between nodes are carried by open connections or virtual circuits in some larger network (e.g., the Internet) instead of by physical wires. The link-layer protocols of the virtual network are said to be tunneled through the larger network when this is the case. We distinguish two families of VPN:

- VPN using a public network such as IPSec or SSL. Data transmitted over the public network are encrypted.
- VPN using a private network (Internet provider) such as MPLS. Data are completely transmitted over the Internet provider private network and thus, does not need any encryption.

We can also classify VPN according to the concerned OSI level. In particular, L2TP and PPTP (level 2), MPLS (level 2.5), IPSec (level 3), SSL/TLS (level 4) et SSH (upper levels).

In our context, it is important to note that most VPNs have been designed for stationary users and point-to-point networks. They were built without consideration for mobility. Therefore, they do not support roaming from one network type to another (e.g., going from GPRS to Wi-Fi) nor are they very robust in handling network disconnects and network time-outs. Moreover, VPNs do not support automatic security enablement. Finally, VPNs, as a rule, do not automatically select the best transmission means when more than one wireless option is available.

Besides that, we can mention that some wireless oriented protocols can use IPSec for security aspects, e.g., Mobile IP. the Mobile IP (or MobileIP) is an Internet Engineering Task Force (IETF) standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address. Mobile IPv4 is described in IETF RFC 3344, and updates are added in IETF RFC 4721. Mobile IPv6 is described in IETF RFC 3775.

In particular, the use of IPSec ESP protocol in the Mobile IP packet redirection tunnels protects the redirected packets against both passive and active attacks launched and aid these packets to traverse the firewalls surrounding both the home and the foreign subnets visited by the mobile nodes.

In the last subsections, we have presented some general security mechanisms. In the next subsections, we classify some security mechanisms according to the concerned layer.

6.2.2.3 Intrusion Detection Systems

The intrusion detection is the monitoring process of events that happens in a computers systems. The aim of this monitoring is to analyze events for detecting signs of intrusion, that can be defined as attempts to compromise confidentiality, integrity, availability or to cheat active security mechanism. Often, to explain the idea and functionality of an Intrusion Detection System (IDS), is used the analogy between these system and antitheft devices adopted in dwelling. In fact, when designing a buildings' security

system, the start point are lock systems (doors, padlock, etc.), with the aim of selecting people that can access to several area of the building. Afterwards, this system is combined with an antitheft device that permits to discover transgressors.

The IDSs belong to defense mechanisms that complete security policy enforced by firewalls and filtering devices, in order to gain ah higher security level. Typically IDSs are hardware or software means that automate monitoring, analysis and alert generation processes. IDSs will be detailed in section 7.

6.2.3 *Physical layer*

Spread spectrum technology, such as frequency hopping (FHSS) or direct sequence (DSSS), can make it difficult to detect or jam signals. It changes frequency in a random fashion to make signal capture difficult or spreads the energy to a wider spectrum so the transmission power is hidden behind the noise level. Directional antennas can also be deployed due to the fact that the communication techniques can be designed to spread the signal energy in space:

- *FHSS*: The signal is modulated with a seemingly random series of radio frequencies, which hops from frequency to frequency at fixed intervals. The receiver uses the same spreading code, which is synchronized with the transmitter, to recombine the spread signals into their original form.
- *DSSS*: Each data bit in the original signal is represented by multiple bits in the transmitted signal, using a spreading code. The spreading code spreads the signal across a wider frequency band in direct proportion to the number of bits used. The receiver can use the spreading code with the signal to recover the original data. Figure 6 illustrates that each original bit of data is represented by 4 bits in the transmitted signal. The first bit of data, a 0 is transmitted as 0110 which is first 4 bits of spreading code. The second bit, 1, is transmitted as 0110 which is bit-wise complementing of the second 4 bits of spreading code. In turn, each input bit is combined, using exclusive-or, with four bits of the spreading code.

Both FHSS and DSSS pose difficulties for outsiders attempting to intercept the radio signals.

6.2.4 *Data-Link layer*

There are malicious attacks that target the link layer by disrupting the cooperative nature of link layer protocols. Link layer protocols help to discover 1-hop neighbors, handle fair channel access, frame error control, and maintain neighbor connections. Selfish nodes could disobey the channel access rule, manipulate the NAV field, cheat backoff values in order to maximize their own throughput. Neighbors should monitor these misbehaviors. Although it is still an open challenge to prevent selfishness, some schemes have been proposed, such as ERA-802.11, where detection algorithms are proposed. Traffic analysis is prevented by encryption at data link layer. WEP encryption scheme defined in the IEEE 802.11 wireless LAN standard uses link encryption to hide the end-to-end traffic flow information. However, as stated above, WEP has been widely criticized for its weaknesses. Some secure link layer protocols have been proposed in recent research, such as LLSP.

[TODO: introduce WPA].

6.2.4.1 **WEP**

In wireless networks, Wired Equivalent Privacy standard (WEP) is sometimes used for authentication and encryption.

Basically, WEP allows for 40-bit or 128-bit keys to be entered in both the access point and the clients to encrypt the traffic between the PC and the access point. However, WEP has several weaknesses.

- the initialization vector (24 bits) is the only part that changes in the key; i.e., keys rotation depends only on 24 bits;

- integrity is verified by CRC32, while this algorithm is linear; in the same way, WEP uses the XOR operator while it is linear;
- WEP uses the RC4 algorithm for encryption; the latter is proved to be weak;
- WEP does not have any mechanisms that prevent replay attacks; arbitrary code can thus easily be injected.

Moreover, with a little digging, unauthorized users can easily find software on the Internet (e.g., aircrack) that can be used to crack WEP encryption by capturing the network traffic over the air and deciphering the key. Once the WEP key is deciphered, the traffic can be read in the clear, overcoming the encryption on the network traffic.

Furthermore, using WEP, we need to key each client device and each access point with the same encryption key. In environments with more several users, the management of these keys, and manual re-keying whenever a user is removed from the network can be burdensome.

To address the inherent flaws of WEP, the Wi-Fi Alliance has created a new standard called Wi-Fi Protected Access (WPA). WPA combines two components to provide strong security for wireless networks. The first component is called Temporal Key Integrity Protocol (TKIP), which replaces WEP with a much stronger protocol.

TKIP provides data encryption enhancements including a key mixing function, a message integrity check (a MIC, for Message Integrity Code), and a re-keying mechanism that rotates through keys faster than any sniffer software can decode the encryption keys.

Through these enhancements, TKIP addresses the WEP's known encryption vulnerabilities [131].

6.2.4.2 WPA

The second component of WPA is 802.1x security, which addresses the key management issue with user authentication. More precisely, 802.1x is called WPA-EAP (*WPA Enterprise*, or EAPOL for *EAP Over Lan*). The latter uses an authentication server such as RADIUS or Diameter.

802.11i WPA2 uses AES-CCMP Encryption. On the one hand, the Advanced Encryption Standard (AES) is a secure, fast symmetric cipher that is easily implemented in hardware. AES has its own mechanism for dynamic key generation. It's also resistant to statistical analysis of the cipher text. On the other hand, Counter-Mode/CBC-MAC Protocol (CCMP) called the Advanced Encryption Standard (AES).

6.2.5 Network layer

We have already presented some general network security mechanisms such as IPSec over Mobile IP. In this sub section, we present two other specific network layer security mechanisms: defense against wormhole attacks and Defense against blackhole attacks.

[TODO: list incomplete]

6.2.5.1 Defense against wormhole attacks

A packet leash protocol is designed as a countermeasure to the wormhole attack. The SECTOR mechanism is proposed to detect wormholes without the need of clock synchronization. Directional antennas are also proposed to prevent wormhole attacks. In the wormhole attack, an attacker receives packets at one point in the network, tunnels them to another point in the network, and then replays them into the network from that point. To defend against wormhole attacks, some efforts have been put into hardware design and signal processing techniques. If data bits are transferred in some special modulating method known only to the neighbor nodes, they are resistant to closed wormholes. Another potential solution is to integrate the prevention methods into intrusion detection systems. However, it is difficult to isolate the attacker with a software-only approach, since the packets sent by the wormhole are identical to the packets sent by legitimate nodes.

- *Packet leashes*: The Packet leashes are proposed to detect wormhole attacks. A leash is the information added into a packet to restrict its transmission distance. A temporal packet leash sets a bound on the lifetime of a packet, which adds a constraint to its travel distance. A sender includes the transmission time and location in the message. The receiver checks whether the packet has traveled the distance between the sender and itself within the time frame between its reception and transmission. Temporal packet leashes require tightly synchronized clocks and precise location knowledge. In geographical leashes, location information and loosely synchronized clocks together verify the neighbor relation.
- *SECTOR*: The SECTOR mechanism is based primarily on distance bounding techniques, one-way hash chains, and the Merkle hash tree. SECTOR can be used to prevent wormhole attacks in MANET without requiring any clock synchronization or location information. SECTOR can also be used to help secure routing protocols in MANET using last encounters, and to help detect cheating by means of topology tracking.
- *Directional antennas*: Directional antennas are also proposed as a countermeasure against wormhole attacks. This approach does not require either location information or clock synchronization, and is more efficient with energy.

6.2.5.2 Defense against blackhole attacks

Some secure routing protocols, such as the security-aware ad hoc routing protocol (SAR), can be used to defend against blackhole attacks. The security-aware ad hoc routing protocol is based on on-demand protocols, such as AODV or DSR. In SAR, a security metric is added into the RREQ packet, and a different route discovery procedure is used. Intermediate nodes receive an RREQ packet with a particular security metric or trust level. At intermediate nodes, if the security metric or trust level is satisfied, the node will process the RREQ packet, and it will propagate to its neighbors using controlled flooding. Otherwise, the RREQ is dropped. If an end-to-end path with the required security attributes can be found, the destination will generate a RREP packet with the specific security metric. If the destination node fails to find a route with the required security metric or trust level, it sends a notification to the sender and allows the sender to adjust the security level in order to find a route.

6.2.5.3 Defense against impersonation and repudiation attacks

The ARAN routing protocol is one of the protocols that integrate mechanisms to prevent impersonation and repudiation attacks at network layer. ARAN provides authentication and non-repudiation services using predetermined cryptographic certificates for end-to-end authentication. In ARAN, each node requests a certificate from a trusted certificate server. Route discovery is accomplished by broadcasting a route discovery message RDP from the source node. The reply message REP is unicast from the destination to the source. The routing messages are authenticated at each intermediate hop in both directions.

6.2.5.4 Defense against modification attacks

We can cite the SEAD security protocol as an example of a defense against modification attacks at network layer. Similar to a packet leash, the SEAD protocol utilizes a one-way hash chain to prevent malicious nodes from increasing the sequence number or decreasing the hop count in routing advertisement packets. In SEAD, nodes need to authenticate neighbors by using TESLA broadcast authentication or a symmetric cryptographic mechanism. Specifically, in SEAD, a node generates a hash chain and organizes the chain into segments of m elements as $(h_0, h_1, \dots, h_{m-1}), \dots, (h_{km}, h_{km+1}, \dots, h_{km+m-1}), \dots, h_n$, where $k = nm - i$, m is the maximum network diameter, and i is the sequence number.

6.2.6 Transport Layer

In wireless networks, like TCP protocols in the Internet, nodes are vulnerable to the classic SYN flooding attack, or session hijacking attack. Point-to-point or end-to-end encryption provides message confidentiality at or above the transport layer in two end systems. TCP is a connection-oriented reliable transport layer protocol. Because TCP does not perform well in wireless networks, TCP feedback (TCP-F), TCP explicit failure notification (TCP-ELFN), ad hoc transmission control protocol (ATCP), and ad hoc transport protocol (ATP) have been invented, but none of these protocols are designed with security in mind. Secure Socket Layer (SSL), Transport Layer Security (TLS), and Private Communications Transport (PCT) protocols were designed for secure communications and are based on public key cryptography. TLS/SSL can help secure data transmission. It can also help to protect against masquerade attacks, man-in-the-middle (or bucket brigade) attacks, rollback attacks, and replay attacks. TLS/SSL is based on public key cryptography, which is CPU-intensive and requires comprehensive administrative configuration.

6.2.7 Application Layer

Like the other protocol layers, the application layer also needs to be secured. In a network with a firewall installed, the firewall can provide access control, user authentication, packet filtering, and a logging and accounting service. Application layer firewalls can effectively prevent many attacks, and application-specific modules, for example, spyware detection software, have also been developed to guard mission critical services. However, a firewall is mostly restricted to basic access control and is not able to solve all security problems. For example, it is not effective against attacks from insiders.

With respect to the security requirements introduced in 3.2, this section describes some defence mechanisms commonly used in WSNs.

6.2.7.1 Protection of confidentiality

Encryption is the standard solution to protect authenticity and secrecy of communications. The problem of establishing a secret key between a pair of nodes in the network is indeed considered along with key management. The simplest solution is to use a networkwide shared key, stored in each node prior to sensor network deployment. However, this scheme is extremely vulnerable to node compromise: attacking even a single node suffices to steal key material and decrypt all network traffic. Moreover, simply preloading key material on sensor nodes can be impractical in large deployment: key establishment protocols may be used instead. Scalability issues must also be considered.

Sensor nodes properties, in terms of limited power resources, computational and communication capabilities, make traditional solutions unsuitable. Hence, the cost of asymmetric cryptography such as RSA or Elliptic Curve Cryptography (ECC) may be too high for many WSN applications. Several alternative approaches have then been developed to perform key management on resource-constrained sensor networks without involving the use of asymmetric cryptography [132].

A possible approach is to preconfigure the network with a shared unique symmetric key between each pair of nodes [48]. Nevertheless, this solution presents scalability issues and high memory cost.

Key Distribution Center (KDC)-based schemes leverage the presence of a central trusted authority (i.e. the gateway) using it as an intermediary for key establishment. *SPINS* [133] is included in this class of protocols. Each node shares a single symmetric secret key with the gateway. If two nodes want to establish a shared secret key K , the gateway acts as a trusted intermediary by transmitting K to both the nodes using the related symmetric secret keys. In details, *SPINS* has two secure building blocks: *SNEP* (*Secure Network Encryption Protocol*) and *mTESLA*. The former block deals with data confidentiality, two-party data authentication and data freshness (i.e. data is recent and no attackers replayed old messages). The latter block provides an efficient broadcast authentication scheme, requiring that the gateway and nodes be loosely time synchronized and introducing asymmetry through a delayed disclosure of symmetric keys.

This class of solutions provides a single point of failure, i.e. the gateway, definitely prone to security attacks. The gateway would also become a scalability bottleneck, as it provides secret key for all the established communication links in the sensor network. Moreover, nodes closest to the gateway suffer from higher communication overhead, expending more battery energy than other sensor nodes and shortening the lifetime of the entire network.

Random key predistribution schemes are another feasible approach. These schemes include an initialization phase, in which a large random pool of symmetric keys is picked and an arbitrary subset of that pool is distributed to each sensor node. When two nodes want to communicate, check whether they share a common key and use it to establish a secure link: this is true with some probability. If key establishment probability is sufficiently great, nodes can still set up enough secure links such that the related graph is fully connected. However, random key predistribution approach is probabilistic in nature and it does not guarantee success; for examples, performing these schemes in sparse networks could result in a disconnected graph.

The main advantage of random key schemes is that they do not include a central trust authority and thus distribute the communication load equally among the sensor nodes. The disadvantage is that compromising a sufficient number of nodes permits to reconstruct the entire key pool, allowing an adversary to perform insider attacks. Moreover, a high memory overhead, which increases linearly with the number of nodes in the sensor network, is required.

Examples of these schemes are described in [134], [135], [136] and [137].

The class of **deterministic key-establishment protocols** described in *Peer Intermediaries for Key Establishment (PIKE)* [132] uses one or more sensor nodes in the network as trusted intermediaries to perform key establishment between neighbouring nodes. This is indeed a *symmetric-key predistribution scheme*. PIKE requires lower memory overhead than random key schemes and a comparable communication overhead. No single point of failure is provided along with a greater resilience against node compromise. In [138] a detailed overview of key distribution methods is presented.

Key revocation is another main aspect in sensor network key management. It consists of revoking all of the keys belonging to a known compromised sensor node, effectively removing the node's presence in the network [138]. It is worth nothing that there is a strict correlation between key establishment and key revocation schemes; actually, some key distribution methods are more suitable for specific key revocation methods (e.g., centralized or distributed), while others may prevent key revocation altogether [138]. Some solutions are presented in [135] and [134].

6.2.7.2 Protection of integrity and authentication

In [139], a **Secure Information Aggregation framework (SIA)** is proposed. The main approach is *aggregate - commit - prove*. In this setting, aggregators perform their aggregation task and commit to the collected data; efficient random sampling mechanisms and interactive proofs are then used to verify the correctness of the results or detect misbehaviours.

Another possible defence involves gathering *multiple, redundant views* of the environment and *cross-check them for consistency* [48]. In [140], the **Confidence Weighted Voting (CWV) technique** exploits neighbouring data to verify the correctness of local data, improving data and decision reliability.

Symmetric group key-based filtering schemes can also be considered. In presence of node compromise, an intruder identification should be realized along with *group rekeying*. A family of *predistribution and local collaboration-based group rekeying (PCGR)* schemes is proposed in [141]. These schemes address the presence of node compromise and improve the effectiveness of filtering false data in sensor networks.

An alternative approach is described in [142]. Sensor nodes maintain *reputation* for other nodes aiming to evaluate their trustworthiness. A web of trust is then established with the purpose of dealing with data reliability.

Mechanisms able to ensure data freshness should also be taken into account.

6.2.7.3 Protection of availability

This section focuses on WSN security mechanism used to protect system availability. In particular, the most common defense against **Denial of Service (DoS) attacks**, **Sybil attacks** and **attacks against routing** are presented.

[TODO: other points need to be merged]

6.2.7.4 Privacy-enhancing technologies

Cryptographic algorithms are a fundamental pillar of each privacy enhancing technology. However, to understand what kind of cryptographic is suitable for WSN is necessary evaluate cryptographic primitives with respect to both security (that concerns with *privacy* and *integrity*) and performance (that concerns with *availability*).

In [143] and [144], by N. D. Jorstad, L. Smith and S. Greenwald it is possible to find some metrics for block ciphers based on key length, number of rounds, clock cycles per round, clock cycles per byte encrypted. Criteria exhibited in these works are very simple and not very interesting with respect to security, because they take into account only exhaustive attacks, that are unlikely threats.

A desirable property for a block cipher is its suitability as a random sequence generator. This property is very important, because encryption is often achieved by XOR-ing the plaintext with a keystream built by using the block cipher (for instance in CTR mode). It is possible to measure this property via statistical tests [145], [146], [147], [148],

1. frequency test and block frequency test;
2. runs test;
3. rank test;
4. spectral test or discrete Fourier transform test;
5. non-periodic template test and overlapping template test;
6. universal statistical test;
7. approximate entropy test;
8. random excursion test and random excursion variant test;
9. Lempel-Ziv complexity test and linear complexity test.

Other suggestions can be drawn from the criteria used in designing and evaluating block ciphers in the challenge launched by NIST in 1997 for replacing DES with a new Advanced Encryption Standard (AES, [149]). For instance, the choice of the round function of the block cipher Rijndael (the winner of the challenge) was determined by the following features [150]: *strength against linear and differential attacks* and *algebraic complexity*. These concepts are very suitable to be measured, because they can be expressed via mathematical quantities: the *propratio*, the *linear correlation* and the *polynomial form* of the round function. Another factor to be considered for block ciphers is the implementation mode in which they are used: electronic codebook (ECB), cipher block chaining (CBC), cipher feedback (CFB), output feedback (OFB), counter mode (CTR) et cetera. Indeed their security can significantly vary depending on it.

Key length, presence of weak keys, performance, strength against known attacks are general criteria to evaluate any kind of cryptographic algorithm (both symmetric and asymmetric) and related implementation. It is possible to find other indications in the guidelines of the European project Nessie [151], whose purpose was to put forward a portfolio of strong cryptographic primitives.

In evaluating if a cryptographic approach is suitable in WSN, a tradeoff between power consumption and achieved security should be the guideline.

Cryptographic is also at the base of the privacy-enhancing technology in strict sense (i.e. techniques to assure privacy and anonymity)

An important contribution derives from Chaum's works introducing the *mix* concept in the far 1981, relating to anonymity of a remailer service [152] in a asymmetric encryption scenario.

The mix is an intermediary server that gathers a fixed amount of messages from various mail clients, and then forwards the messages to the right destination.

The mail clients have to encrypt their own messages with both the legitimate end-receiver's public key and mix's public key, in order to protect the destination address and disclose it only to the mix.

The mix decrypt the messages gathered, finds out the legitimate end-receiver address (encrypted with the mix's public key) and only after the collection of a specified amount of messages it forwards them in random order to the legitimate destination. This kind of behavior significantly lowers the possibility of identifying the link between sender and receiver, even in the case of an attacker able to listen on every communication channel between every sender and every receiver.

However this solution exposes the system to a "single-point-of-failure" problem, because the corruption of the mix node exposes every sender-receiver pairs and the anonymity of the system falls to zero.

A generalization already present in the original paper [152] employees k different mix nodes, with $k > 1$, in which a message transit before the reaching of the final destination. This kind of generalization is based on the "multi-fortified city" concept, where many perimetral walls ensure the security of the citizens, though just one is enough to resist foreign invaders. Likewise with mixes where just one honest mix node enables the anonymity protection.

The original mix concept was developed in a mail system scenario, but the validity of Chaum's idea was such to be applicable to every communication scenario (e.g. a web based one like in [153]) with only minor adaptations.

The main problem of this solution, especially in ambient of WSN, is the high overhead added. From a cryptographic prospective every message has to be encrypted and decrypted k times each in an asymmetric encryption scheme. Moreover, from a network prospective, every message has to be reduced or increased to the chosen length, and its forwarding is delayed in every mix node until the node itself has gathered the chosen amount of messages.

Probably the most successful mix-based application on the web is the so-called *onion routing* mechanism [154], whose improvement is still ongoing [155]. Purpose of the project is the development and analysis of an internet connection schema that resists traffic analysis and eavesdropping in order to vouch the anonymity of communicating parties, without modification of the underlying operative system. Before the communication of original data begins, a *circuit* traversing the onion router network is established between *initiator* and *responder*. The circuit establishment is achieved by means of *onion packets* that go through onion routers, loose a layer per hop and deposit meanwhile session parameters (like the symmetric session key) until the last hop, in which the onion packet ends as plaintext. Every node knows only predecessor and successor, if any, so a single malicious node can compromise little or no privacy at all. This solution achieve a lower security level in respect to Chaum's original solution, but the improvement (e.g. exploiting symmetric encryption) make it much more practical and feasible than the former, and surely much more suitable in WSN.

6.2.8 Multi-Layer

The DoS attacks, impersonation attacks, man-in-the-middle attacks, and many other attacks can target multiple layers. The countermeasures for these attacks need to be implemented at different layers. For example, directional antennas are used at the media access layer to defend against wormhole attacks, and packet leashes are used as a network layer defense against wormhole attacks. The countermeasures for multi-layer attacks can also be implemented in an integrated scheme.

7 INTRUSION DETECTION SYSTEMS

Intrusion detection is the process of detecting and identifying malicious and unauthorized use, misuse, and abuse of computer systems. Thus, it concerns the set of practices and mechanisms that contributes to the diagnosis of attacks and/or the detection of errors that may lead to security failure. An *Intrusion Detection System (IDS)* is an implementation of practices and mechanisms of intrusion detection. IDSs include all software or hardware systems that automate the process of monitoring events occurring in a computer system or network and analyzing them for clues of security breaches (i.e., compromising confidentiality, integrity, or availability, or bypassing security mechanisms of a computer or network). Early IDS implementations have appeared since the beginning of 1980s [156], [157]. Since then, a number of research and open source IDSs were created such as: STAT family (USTAT, NSTAT, NetSTAT) [158], [159], EMERALD [160], Bro [161], and Snort [162]. Commercial IDSs started to emerge starting from 1990s, for example, Cisco secure IDS (previously known as NetRanger) and ISS RealSecure. Despite different implementations, all intrusion detection systems' major task is to collect data from computer systems or computer networks; analyze them to find security-relevant events and raise alarms if they find any. According to the Common Intrusion Detection Framework (CIDF) model [163], any IDS is composed of the following components:

- *E-Box*: Event-box, which collects data from the information source (e.g., network traffic, host logs), and feeds interesting data to the IDS.
- *D-Box*: Database-box in which the relevant events are stored after some preprocessing (e.g., normalization of different logs in a common format).
- *A-Box*: Analysis-box, the core unit of any IDS that manipulates the event data and contains the detection engine.
- *R-Box*: Response-box, this component is concerned with responsive actions that can be taken upon detection of intrusions. The response can be an administrative action such as modifying the firewall rules to block the intruder traffic, ending the TCP connection or simply generating an alert.

An IDS consists mainly of at least one *detector unit*, at least one *alarm/report generator*, one or more *sensor units* and optionally includes preprocessing and correlation units. An IDS must have at least one sensor either of its own or it must import information from other sources such as IDS audit.

Normally, the intrusion detection process starts by collecting events. It then passes them either to a preprocessing unit to normalize data or directly to the detector unit. The latter analyzes the gathered data and decides whether they correspond to signs of an attack. If this is the case, the reporter unit generates an alarm indicating the occurrence of the attack. If the IDS includes a correlation unit, it aggregates alarms that belongs to the same scenario, or extracts more information from the gathered data. We consider Intrusion Prevention System (IPS) as a kind of IDS that not only detects attacks but also prevents their occurrence. It extends the functionality of IDS by a response unit to prevent attacks or limit their effects. Typical responses to intrusions may include reconfiguring the firewall to drop suspicious traffic, denying user access to resources that exhibit anomalous behavior, etc.

Note that IDSs IPSES have evolved much in the last decade and now they tend to be architecturally distributed and can integrate various sensors from different sources. Furthermore, centralized management consoles, correlation engines and reporting front ends have been proposed to facilitate the use of several heterogeneous but complementary IDSs.

7.1 IDS Taxonomies

There are several taxonomies of IDSs such as the ones by Debar et al [164], and Axelsson et al [165] based on various criteria. Examples of criteria include:

- *The source of event data*: there are two major categories: (1) *Network-based IDSs* (NIDSs), which typically read event data directly off a multicast network such as Ethernet and (2) *Host-based IDSs* (HIDSs), which collect and analyze event data collected on the host. The host data are typically logs such as operating system kernel logs, application program logs or even firewalls logs, etc.
- *Architecture*: used to differentiate between centralized IDSs that analyze the data collected only from a single monitored system and distributed IDSs that collect information from multiple monitored systems in order to investigate global, distributed and coordinated attacks;
- *The location of data collection*: audit data for the processor/detector can be collected from many different sources in a distributed manner, or from a single point using a centralized approach.
- *The location of data processing*: collected data can be processed and analyzed centrally even if it was collected from many different sources. Otherwise, it can be processed and analyzed at the same place where it was collected.
- *The detection method or technique*: two categories can be distinguished: *behavior-based* (also known as *anomaly detection*) and *knowledge-based* (namely *signature-based* or *misuse detection*).
- *The time of detection*: two main groups can be identified: those that analyze events on line and attempt to detect intrusions in real-time or near real-time, and those that process audit data with some delay or offline (non-real-time), which in turn delays the time of detection.
- *The granularity of data processing*: this criterion distinguishes IDSs that process data continuously from those that process event data in batch mode.
- *The behavior on detection* (response to detected intrusions): an IDS can be classified as passive or active. Passive systems notify the proper authority, and they do not try to mitigate the damage by themselves. Contrarily, active IDS reacts to stop the attack (e.g., terminates the attack session).

In the following section a more detailed description for the classification shown above is presented.

7.1.1 Source of event data

According to the first criterion, it is possible to analyze network traffic (IP packet, TCP flows, HTTP sessions, etc) or monitoring the behavior of a single host (logs analysis, connection opening, running software, privilege used, etc.) or do both at the same time:

Network Intrusion Detection System - NIDS) a NIDS tries to identify anomalies or attacks by analyzing network traffic. Typically they are composed by sensors arranged in several point of the network: when an attack (or an attempt of attack) is identified the sensors generate an alarm that must be processed by a system administrator. NIDSs are considered an effective way for realizing IDSs because they have no impact on network configuration, and often they are clear to the system and attackers, so if a NIDS fails the network can still work. One big limit of NIDS are encrypted message, the addition of a cryptographic layer reduces considerably the analysis that a NIDS can perform. Another limit is the amount of data that must be examined by the sensors, in a big network packets can be a lot and the NIDS could not be able to look over all the data.

Host-based Intrusion Detection System - HIDS): a HIDS is a software agent that analyses the system on which it is installed. The analysis is carried out on log files, their integrity and the whole state of the system in order to reveal anomalies or attack signals, such as excessive use of memory and disks or "listening" ports. Since it monitors a single host, and its local events, it is able to analyze the system with great reliability and precision, but this brings the HIDS to have a restricted sight, limited to the host on which it runs. In addition this implies a low scalability, because it is not

possible to install a HIDS on every host in the network. Another risk of this solution is that it is not always possible to trust the system on which the HIDS is running, maybe it could be corrupted or the HIDS could be deactivated. The advantages of this solution is the complete knowledge of what is happening in the system, so a HIDS can also prevent intrusions or attacks by blocking those processes or packets that are considered dangerous for the system.

Distributed IDS (DIDS): this solution combines the previous approaches. The data, network and host analysis, can be put together for obtaining more accurate information.

7.1.2 Architecture

There are two main architectures that are used in IDSs: centralized and distributed. A *centralized architecture* is composed by several sensors that detect intrusions in a single monitored system, and a central control point that coordinates and correlates the alarm. This implies a simple management of the IDS, but creates a single point of failure: if the control point fails all the system becomes useless. *Distributed architectures* can be divided into two sub-categories:

- *partially-distributed*: when several sensor can monitor, detect and generate report that are sent hierarchically to different control point;
- *totally-distributed*: when the nodes are autonomous and can analyze audit data by themselves.

This approach has two main problems, the first is the complexity of allowing the interoperability between different entities of the system that could be hard to obtain especially because the heterogeneity of data originated by different system. The second is the need of assuring integrity and confidentiality of the exchanged data, that should not be modified and can carry sensitive information that must remain secret. In addition, distributed IDSs can help against distributed and coordinated attacks, where multiple machines are involved (e.g. distributed denial-of-service). Techniques that use data from a single site and that are used by many intrusion detection schemes are often unable to detect such attacks. To effectively detect them, distributed IDSs are needed, because they allow cooperation between different network sites and can increase the amount and effectiveness of data exchanged for carrying out more accurate security analyses.

7.1.3 Locations of data collection and data processing

A monolithic network IDS deployed on a single host cannot see or handle all traffic passing either on switched LANs or on networks with high data rates. Moreover, it is no longer able to treat massive volumes of heterogeneous security data. For these reasons, Distributed Intrusion Detection Systems such as DIDS [166], EMERALD [160], GrIDS [167] have been created to monitor more hosts and several points within the network. Basically, two architectures of distributed IDSs must be considered: first, an architecture with distributed sensors but centralized analysis like DIDS [166] and Prelude where sensors that support different detection techniques could be integrated. For example, Prelude integrates Snort as a NIDS, prelude-lml as a HIDS that analyses system log files and Samhain as another HIDS that checks file integrity. Although this architecture allows monitoring several points, it exhibits a single point of failure once an intruder manages to get the central analyzer down. It suffers also from the poor scalability due to the limited capacity of the analyzer and the excessive data transmission between sensors and analyzers. The second architecture is hierarchical where the IDS is structured in several layers and redundant components such as EMERALD or GrIDS. Thus, there is no single point of failure and the scalability is improved as the analysis burden is distributed over many hosts. However, reconfiguring such systems is difficult and not flexible. Distributed IDSs can be implemented following the client/server model as well as by using agent-based approaches.

7.1.4 *Detection method*

The detection method is the technique used by an IDS to determine whether an intrusion has occurred or not. There are two broad categories of detection methods: anomaly-based or signature-based (also known as behavior-based and misuse-based respectively).

- *Signature-based IDS (Misuse detection)*: Misuse or signature-based detectors analyze system activities, looking for events (or sets of events) that match a predefined pattern describing a known attack. This implies the analysis of signatures that represent a known pattern of attack. A signature can be the interpretation of series of packets or a piece of data contained in those packets. It may also manifest itself in audit records, logs, or in changes in files or memory of the compromised system. This type of IDSs can only detect previously known attacks. Therefore, they must be constantly updated with signatures of new attacks. Signature definition is a critical task. If signatures are loosely defined, the IDS will detect a broader range of attacks at the expense of generating more false alarms. On the other hand, if signatures are tightly defined, this will reduce the number of false alarms but the IDS will be unable to detect variants of common attacks.
- *Behavior-based IDS (Anomaly detection)*: Anomaly detection identifies any unacceptable deviation from the expected behavior on a host or a network. It assumes that attacks are different from "normal" (legitimate) activities and can therefore be detected by systems that identify these differences. Expected behaviors of users, hosts or network connections are constructed, in advance. Profiles can be created manually or automatically based on historical data collected over a period of normal operation (supposed free of attacks). An automatically developed profile is created by software that collects and processes characteristics of system behavior over time and forms a statistically valid sample of such behavior. Note that unexpected behavior is not necessarily an attack; it may represent new, legitimate behavior that needs to be included in the profile. The measured features that may comprise a profile include the number of failed login attempts to the system, the time or location of login, the number of files accessed by a user in a given period of time, etc. Several techniques are used to determine whether the behavior is abnormal or not. Unfortunately, behavior-based IDSs often need a training period and are sensitive to the training dataset. Therefore, they often produce a large number of false alarms, as normal patterns of user and system behavior can vary widely. Despite this shortcoming, researchers assert that behavior-based IDSs are able to detect new attack forms, unlike signature-based IDSs that rely on matching patterns of past known attacks. On the contrary, alarms generated by behavior-based IDSs are less precise than these generated by its signature-based counterpart. The latter often identifies the detected attack and provides rich information such as references to the exploited vulnerabilities and even advices to correct them.

7.1.5 *Time of detection and granularity of data processing*

Time aspects are used to categorize the IDSs into: on-line IDSs, that detect intrusions on line, and off-line IDSs, that first store the monitored data and then analyze them in batch mode for signs of intrusion. An on line detector operates on continuous data streams from information sources and analyze the data while the system is running. Often the term *real time* is used instead of on-line. It indicates that the detector is designed to audit a specific data flow without losing any information (e.g. IP packets) and with no buffer that is waiting to be processed. An off-line (or batch) detector performs post-analysis of audit data; this method is commonly used in the *computer forensics* field. Often, network behavior, processes running and behavior of the attackers are studied through this method. The analysis is often carried out using static tools that analyze the snapshot of the system, look for known vulnerabilities, security configuration, malicious software, etc. Moreover, a thinner classification can be done. As an example, an IDS can also be *in-line*, that is a detector placed like a firewall or directly to a switch port, while an on line IDS is, usually, connected to a network plug that presents a copy of all the traffic. Considering

how and where it works, an on-line IDS can also work like a filtering device, dropping packets that are considered dangerous, becoming an Intrusion Prevention System (IPS).

7.1.6 Behavior on detection

The response of IDSs to identified attacks may be either passive or active:

- *passive IDSs* generate an alarm every times an intrusion attempt is detected, but no countermeasure is applied to thwart the attack. Many commercial IDSs implement this type of response, and usually they offer several options for selecting how and when a notification has been generated and visualized. The most common way for showing alarm are through pop-up windows or on-screen alerts prompted on the IDS console or in other system that can be specified during the configuration phase. The provided informations can be different, such as a simple notification of the detection of an intrusion or messages that indicate detailed information about computer attacks such as source IP address, target of the attack, specific port of interest, the tools used to perform the attack, the outcome of the attack, etc.
- *active IDSs* perform automatic actions when certain type of intrusion are detected. They can be classified in different categories, based on the type of response to the attack. An active IDS can:
 - collect additional information about the attack, by increasing the global level of information sources' sensitivity (e.g. augmenting the number of audit provided by the operative system);
 - change the environment of the system, aiming to interrupt an attack and subsequent actions of the attacker. Usually, an IDS cannot block the access of a specific person, but can only block the IP address from which the attack seems to have started. Since it is difficult to block a specific attacker, the IDSs try to change the environment of the attacked system. They reconfigure routers and firewalls, in order to intercept and block all the traffic flows that belong to a specific IP address.

7.2 Wireless IDS

7.2.1 Wireless IDSs vs classical IDSs

Traditional intrusion detection systems, like snort, are not really sufficient to detect wireless attacks. A wireless IDS is unique in that it detects attacks against the 802.11 frame at layer two of the wireless network. In fact, most of wireless threats (e.g., man-in-the-middle attacks, rogue access points, war drivers and wireless denial of service attacks) cannot be detected on layer three past the access point. Wired IDS will not receive these frames, because management frames are not forwarded to upper layers of the OSI model.

A wireless IDS can be deployed using a network of dedicated wireless devices running in monitor mode, also known as RFMON mode; this mode is similar to promiscuous mode for wired devices and allows the device to accept all incoming traffic.

The monitoring interface should hop between the 12 channels available to wireless networks. Several wireless attacks work by utilizing a rogue AP on a different channel. For instance, man-in-the-middle attacks utilize a rogue AP that is at least 5 channels away from the target AP. Without channel hopping the wireless IDS would be blind to attacks that function on other channels.

Concerning the detection methodology, as for classical IDSs, wireless IDSs are signature or knowledge-based.

In terms of architecture, a wireless IDS can be centralized or decentralized. A centralized wireless IDS is usually a combination of individual sensors which collect and forward all 802.11 data to a central management system, where the wireless IDS data is stored and processed. Decentralized wireless intrusion detection usually includes one or more devices that perform both the data gathering and processing/reporting functions of the IDS.

7.2.2 Examples of Wireless IDSs

There are currently only a handful of vendors who offer a wireless IDS solution - but the products are effective and have an extensive feature set. Popular wireless IDS solutions include Airdefense RogueWatch and Airdefense Guard, and Internet Security Systems Realsecure Server sensor and wireless scanner products. A homegrown wireless IDS can be developed with the use of the Linux operating system, for example, and some freely available software. Open source solutions include Snort-Wireless and WIDZ and AirIDS [168]:

- *Snort-wireless* is a wireless intrusion detection system adapted from the Snort IDS engine, while it is focused on lower layers. In fact, it adopts the similar syntax (when it comes to writing snort-wireless rules) as the classical Snort IDS, but it replaces the source and destination IP addresses in the normal Snort rules with source and destination MAC addresses. In this way Snort-wireless rules can also detect wireless traffic.
- *WIDZ* (version 1) is a proof of concept IDS system for 802.11 that guards an AP(s) and monitors local frequencies for potentially malevolent activities. It detects scans, association floods, and bogus/Rogue AP's. It can easily be integrated with Snort or RealSecure. WIDZ has two modules: (1) *Unauthorized AP monitor* which is responsible for detecting bogus & rouge APs by checking an AP scan result with a baseline file of all authorized APs and (2) *802.11 Traffic monitor* that includes probe / flood monitoring as well as MAC and ESSID blacklist and whitelist.
- *AirIDS* is a wireless IDS with several interesting aspects. In fact, it allows robust and powerful rules file controls filtering, which is user definable. Moreover, it is able to forge frames so as to provide not just detection but active defenses against malicious 802.11(b) activities.

7.2.3 Wireless IDS drawbacks

Wireless intrusion detection is a rather new technology that has not been really largely deployed and tested. Consequently, the different techniques are still subject to bugs or vulnerabilities.

Besides that, a wireless IDS, like a standard IDS, can require vast human resources to analyze and respond to threat detection. It can be argued that a wireless IDS will require more human resources than a standard IDS because with a wireless IDS, individuals will be required to pay attention both to the logical (alert data) and to the physical aspects of an attack.

8 CONCLUSION AND OUTLOOK

This contribution provides an overview to Wireless Network security threats, vulnerability measures and countermeasures, describing also the research work done by the contributing partners in that field. It is the first joint work of WPA and lays the foundation of its future work. WPA will in particular within its future work further investigate metrics for vulnerabilities in wireless networks and for the strength of selected security concepts, which can measure how effective countermeasures are alone or in combination against the major threats which have been identified. It will also interact closely with WP1 in terms of measurements.

REFERENCES

- [1] Andreas Pfitzmann and Marit Hansen. Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology v0.31, 15 Feb 2008. See <http://dud.inf.tu-dresden.de/literatur/>.
- [2] John Stankovic Anothony D. Wood. A taxonomy for denial-of-service attacks in wireless sensor networks.
- [3] Iso/iec 13335-1:2004 - information technology – security techniques – management of information and communications technology security – part 1: Concepts and models for information and communications technology security management, 2004.
- [4] Common Criteria for Information Technology Security Evaluation (CC), Sep 2007.
- [5] H.A. Lorentz. The theorem of Poynting concerning the energy in the electromagnetic field and two general propositions concerning the propagation of light. *Amsterdammer Akademie der Wetenschappen*, 176, 1896.
- [6] Claude E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.
- [7] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli Maurer. Generalized privacy amplification. *IEEE Transaction on Information Theory*, 41(6):1915–1923, November 1995.
- [8] Ueli Maurer. Secret key agreement by public discussion from common information. *IEEE Transaction on Information Theory*, 39(3):733–742, May 1993.
- [9] Ueli Maurer and Stefan Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Transaction on Information Theory*, 45(2):499–514, March 1999.
- [10] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.
- [11] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer-Verlag, August 1997.
- [12] M. Guillaud, D. T. M. Slock, and R. Knopp. A practical method for wireless channel reciprocity exploitation through relative calibration. In *Proc. Eighth International Symposium on Signal Processing and Its Applications (ISSPA '05)*, Sydney, Australia, August 2005.
- [13] Chunxuan Ye and P. Narayan. Secret key and private key constructions for simple multiterminal source models. In *Proc. International Symposium on Information Theory (ISIT)*, pages 2133–2137, September 2005.
- [14] Chunxuan Ye, Alex Reznik, and Yogendra Shahn. Extracting secrecy from jointly Gaussian random variables. In *Proc. International Symposium on Information Theory (ISIT)*, July 2006.
- [15] Chunxuan Ye, A. Reznik, G. Sternberg, and Y. Shah. On the secrecy capabilities of ITU channels. In *Proc. Vehicular Technology Conference (VTC) Fall*, pages 2030–2034, October 2007.
- [16] R. Wilson, D. Tse, and R. Scholtz. Channel identification: Secret sharing using reciprocity in UWB channels. *IEEE Transactions on Information Forensics and Security*, 2:364–375, September 2007.

- [17] A. Sayeed and A. Perrig. Secure wireless communications: Secret keys through multipath. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3013–3016, April 2008.
- [18] A. D Wyner. The Wiretap Channel. *Bell. Syst. Tech. J.*, 54, 1975.
- [19] S. Leung-Yan-Cheong and M. Hellman. The Gaussian wire-tap channel. *IEEE Transactions on Information Theory*, 24(4):451–456, 1978.
- [20] I. Csiszár and J. Körner. Broadcast Channels with Confidential Messages. *IEEE Trans. Inform. Theory*, 24, 1978.
- [21] UM Maurer. Secret key agreement by public discussion from common information. *Information Theory, IEEE Transactions on*, 39(3):733–742, 1993.
- [22] P.K. Gopala, L. Lai, and H. El Gamal. On the Secrecy Capacity of Fading Channels. *IEEE Trans. on Inform. Theory*, 54(10), October 2008.
- [23] A. Khisti and G. Wornell. The MIMOME Channel. *Arxiv preprint arXiv:0710.1325*, 2007.
- [24] T. Liu, V. Prabhakaran, and S. Vishwanath. The secrecy capacity of a class of parallel Gaussian compound wiretap channels. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 116–120, 2008.
- [25] Y. Liang, H.V. Poor, and S. Shamai. Secure Communication over Fading Channels. *IEEE Trans. on Inform. Theory*, 54(6), June 2008.
- [26] R. Negi and S. Goel. Secret communication using artificial noise. *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, 3, 2005.
- [27] P. Parada and R. Blahut. Secrecy capacity of SIMO and slow fading channels. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 2152–2155, 2005.
- [28] A. Khisti and G. Wornell. Secure Transmission with Multiple Antennas: The MISOME Wiretap Channel. *Arxiv preprint arXiv:0708.4219*, 2007.
- [29] T. Liu and S. Shamai. A Note on the Secrecy Capacity of the Multi-antenna Wiretap Channel. *Arxiv preprint arXiv:0710.4105*, 2007.
- [30] F. Oggier and B. Hassibi. The Secrecy Capacity of the MIMO Wiretap Channel. *Arxiv preprint arXiv:0710.1920*, 2007.
- [31] S. Shafiee and S. Ulukus. Achievable rates in Gaussian MISO channels with secrecy constraints. In *Proc. Int. Symp. Inform. Theory*, 2007.
- [32] A. Khisti, G. Wornell, A. Wiesel, and Y. Eldar. On the Gaussian MIMO wiretap channel. *Proc. ISIT'07, Nice, France*.
- [33] J. Barros and M.R.D. Rodrigues. Secrecy capacity of wireless channels. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2006.
- [34] M. Bloch, J. Barros, MRD Rodrigues, and SW McLaughlin. Wireless Information-Theoretic Security. *Information Theory, IEEE Transactions on*, 54(6):2515–2534, 2008.
- [35] Y. Liang, G. Kramer, and S. Shamai. Compound Wire-tap Channels. *Allerton*, 2007.
- [36] Y. Liang, G. Kramer, , H. Vincent Poor, and S. Shamai. Recent Results on Compound Wire-tap Channels. *PIMRC*, 2008.

- [37] R. Liu, I. Maric, P. Spasojevic, and RD Yates. Discrete Memoryless Interference and Broadcast Channels With Confidential Messages: Secrecy Rate Regions. *Information Theory, IEEE Transactions on*, 54(6):2493–2507, 2008.
- [38] A. Khisti, A. Tchamkerten, and GW Wornell. Secure Broadcasting Over Fading Channels. *Information Theory, IEEE Transactions on*, 54(6):2453–2469, 2008.
- [39] R. Liu and H.V. Poor. Multi-Antenna Gaussian Broadcast Channels with Confidential Messages. *Proceeding of ISIT'08, Toronto, Canada*, available Arxiv preprint arXiv: 0804.4195, 2008.
- [40] Hung D. Ly, Tie Liu, and Yingbin Liang. MIMO Broadcasting with Common, Private, and Confidential Messages. In *ISITA'2008, New Zealand*, December 2008.
- [41] O.O. Koyluoglu, H. El Gamal, L. Lai, and H.V. Poor. On the Secure Degrees of Freedom in the K-User Gaussian Interference Channel. *ISIT'08 Toronto, ON, Canada*, also available on Arxiv preprint arXiv:0805.1340, 2008.
- [42] O.O. Koyluoglu, H. El Gamal, L. Lai, and H.V. Poor. Interference Alignment for Secrecy. *Arxiv preprint arXiv:0810.1187*, 2008.
- [43] Mari Kobayashi and Mérouane Debbah. On the secrecy capacity of frequency-selective fading channels: A practical vandermonde precoding. In *Proc. International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Cannes, France, September 2008.
- [44] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. Wireless sensor network security: A survey," in book chapter of security - wireless sensor network security: a survey. In *in Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds)*, pages 0–849. CRC Press, 2007.
- [45] John Douglas Howard. *An analysis of security incidents on the Internet 1989-1995*. PhD thesis, Pittsburgh, PA, USA, 1998.
- [46] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *In First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.
- [47] Elaine Shi and Adrian Perrig. Designing secure sensor networks. *Wireless Communication Magazine*, 11(6):38–43, December 2004.
- [48] Adrian Perrig, John Stankovic, David Wagner, and Caren Rosenblatt. Security in wireless sensor networks. *Communications of the ACM*, 47:53–57, 2004.
- [49] Haowen Chan and Adrian Perrig. Security and privacy in sensor networks. *Computer*, 36(10):103–105, 2003.
- [50] Tom Karygiannis. Wireless network security: 802.11, bluetooth, and handheld devices. Special Publication 800–48, NIST, Oct 2002.
- [51] William Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, Upper Saddle River, NJ, USA, third edition, 2003.
- [52] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM-01)*, pages 180–189, New York, NY, USA, 16–21 Jul 2001. ACM Press.
- [53] William A. Arbaugh, Narendar Shankar, Yung-Chun Justin Wan, and Kan Zhang. Your 802.11 wireless network has no clothes. *IEEE Communications Magazine*, 6(9):44–51, Dec 2002.

- [54] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.
- [55] Michel Barbeau, Jeyanthi Hall, and Evangelos Kranakis. Detecting Impersonation Attacks in Future Wireless and Mobile Networks. In Mike Burmester and Alec Yasinsac, editors, *Revised Selected Papers of the 1st International Workshop Secure Mobile Ad-hoc Networks and Sensors (MADNES 2005)*, volume 4074 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 20–22 Sep 2006.
- [56] Nadarajah Asokan, Valtteri Niemi, and Kaisa Nyberg. Man-in-the-Middle in Tunneled Authentication Protocols. Technical Report 2002/163, IACR ePrint Archive, Oct 2002. See <http://eprint.iacr.org/2002/163/>.
- [57] IEEE Std 802.11, 2007, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ISO/IEC 8802-11 IEEE Std 802.11, 12 Jun 2007.
- [58] IEEE Std 802.15.1, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirements Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). IEEE Std 802.15.1, 14 Jun 2005.
- [59] IEEE Std 802.15.3, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirements Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs). IEEE Std 802.15.3, 29 Sep 2003.
- [60] IEEE Std 802.15.4, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirements Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). IEEE Std 802.15.4, 8 Sep 2006.
- [61] IEEE Std 802.16, IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed Broadband Wireless Access System. IEEE Std 802.16, 1 Oct 2004.
- [62] Farid Naït-Abdesselam, Brahim Bensaou, and Tarik Taleb. Detecting and Avoiding Wormhole Attacks in Wireless Ad Hoc Networks. *IEEE Communications Magazine*, 46(4):127–133, Apr 2008.
- [63] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, Sep 2003.
- [64] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2nd ACM Workshop on Wireless security (WiSE'03)*, pages 30–40, New York, NY, USA, 19 Sep 2003. ACM.
- [65] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.
- [66] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the 1st ACM Workshop on Wireless security (WiSE'02)*, pages 21–30, New York, NY, USA, 28 Sep 2002. ACM.

- [67] Imad Aad, Jean-Pierre Hubaux, and Edward W. Knightly. Denial of service resilience in ad hoc networks. In Zygmunt J. Haas, Samir R. Das, and Ravi Jain, editors, *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM 2004)*, pages 202–215, New York, NY, USA, 26 Sep– 1 Oct 2004. ACM Press.
- [68] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.
- [69] John R. Douceur. The Sybil Attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems: Proceedings of the 1st International Peer-to-Peer Systems Workshop (IPTPS)*, volume 2429, pages 251–260. Springer-Verlag, 7–8 Mar 2002.
- [70] Frank Stajano. The Resurrecting Duckling: What Next? In *Revised Papers from the 8th International Workshop on Security Protocols*, Lecture Notes in Computer Science, LNCS 2133, pages 204–214, London, UK, 3–5 Apr 2001. Springer.
- [71] Yang Xiao, Xuemin Shen, and Ding-Zhu Du, editors. *Signals and Communication Technology*. Springer, New York, NY, USA, first edition, Oct 2006.
- [72] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Communication Society (INFOCOM 2000)*, volume 2, pages 775–784, Tel Aviv, Israel, 26–30 Mar 2000.
- [73] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Lydia E. Kavraki, Dan S. Wallach, and Guillaume Marceau. Robotics-based location sensing using wireless ethernet. In Ian F. Akyildiz, Jason Yi-Bing Lin, Ravi Jain, Vaduvur Bharghavan, and Andrew T. Campbell, editors, *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM 2002)*, pages 227–238, New York, NY, USA, 23–28 Sep 2002. ACM Press.
- [74] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MOBICOM 2008)*, pages 116–127, New York, NY, USA, 14–19 Sep 2008. ACM Press.
- [75] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. *Mobile Networks and Applications*, 10(3):315–325, Jun 2005.
- [76] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve F. Russell. Device identification via analog signal fingerprinting: A matched filter approach. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2006)*. The Internet Society, 2–3 Feb 2006.
- [77] Fanglu Guo and Tzi cker Chiueh. Sequence number-based mac address spoof detection. In Alfonso Valdes and Diego Zamboni, editors, *Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection - Revised Papers (RAID 2005)*, volume 3858 of *Lecture Notes in Computer Science*, pages 309–329. Springer, 7–9 Sep 2005.
- [78] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoie, Jamie Van Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the 15th Conference on USENIX Security Symposium (USENIX-SS 2006)*, pages 12–12, Berkeley, CA, USA, 31 Jul–4 Aug 2006. USENIX Association.
- [79] Ad Hoc Network Autoconfiguration (autoconf), 2006. See <http://www3.ietf.org/html.charters/autoconf-charter.html>.

- [80] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (aodv) routing. RFC-3561, Jul 2003. See <http://www.ietf.org/rfc/rfc3561.txt>.
- [81] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. RFC-4728, Feb 2007. See <http://www.ietf.org/rfc/rfc4728.txt>.
- [82] Jon Postel. Internet control message protocol. RFC-792, Sep 1981. See <http://www.ietf.org/rfc/rfc1158.txt>.
- [83] Tadayoshi Kohno, Andre Broido, and Kimberly C. Claffy. Remote physical device fingerprinting. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P 2005)*, pages 211–225. IEEE Computer Society, 8–11 May 2005.
- [84] Van Jacobson, Bob Braden, and Dave Borman. TCP Extensions for High Performance. RFC-1323, May 1992. See <http://www.ietf.org/rfc/rfc1323.txt>.
- [85] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.
- [86] M. Sahinoglu, C.V. Ramamoorthy, A.E. Smith, and B. Dengiz. A reliability block diagramming tool to describe networks. In *Proc. of Reliability and Maintainability Annual Symposium*, pages 141–145, 26–29 January 2004.
- [87] W. Wang, J.M. Loman, R.G. Arno, P. Vassiliou, E.R. Furlong, and D. Ogden. Reliability block diagram simulation techniques applied to the ieeec 493 standard network. *IEEE Transactions on Industry Applications*, 40(3):887–895, May-June 2004.
- [88] L.L. Pullum and J.B. Dugan. Fault tree models for the analysis of complex computer-based systems. In *Annual International Symposium on Product Quality and Integrity*, pages 200–207, 22–25 January 1996.
- [89] R. W. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Proc. of IEEE Computer Society Symposium on Security and Privacy*, pages 156–165, May 2000.
- [90] M. D. Aime and A. Atzeni. Generation of diagnostic plans for large ict systems. In *Proc. of the Second International Conference on Emerging Security Information, Systems and Technologies (SECUREWARE'08)*, 25–31 August 2008.
- [91] DESEREC. DEpendability and Security by Enhanced REConfigurability. <http://www.deserec.eu/>.
- [92] S.Jha, O. Sheyner, and J. Wing. Minimization and reliability analyses of attack graphs. Technical Report CMU-CS-02-109, Carnegie Mellon University, February 2002.
- [93] I. Kottenko and M. Stepashkin. Network security evaluation based on simulation of malefactor's behavior. In *SECRYPT 2006. International Conference on Security and Cryptography*, 2006.
- [94] D. Balzarotti, M. Monga, and S. Sicari. Assessing the risk of using vulnerable components. In *QoP2005, First Int. Workshop on Quality of Protection*, pages 65–77, 15 September 2005.
- [95] William Thompson. Popular lectures and addresses, 1891-1894.
- [96] nikto. nikto web server security scanner. <http://www.cirt.net/code/nikto.shtml>.
- [97] SAINT. Security Administrators Integrated Network Tool. <http://www.saintcorporation.com/>.
- [98] R. Deraison. Nessus open source vulnerability scanner project. <http://www.nessus.org>, 2002.

- [99] onlinescan. on-line broadband service. <http://www.dslreports.com/scan/>.
- [100] dmoz. open directory project scanner list. http://dmoz.org/Computers/Security/Internet/Products_and_Tools/Security_Scanners/.
- [101] MITRE. Common vulnerabilities and exposures web site. <http://www.cve.mitre.org/>.
- [102] FIRST. Common Vulnerability Scoring System (CVSS). <http://www.first.org/cvss/cvss-guide.html>.
- [103] M. Dacier. *Towards Quantitative Evaluation of Computer Security*. PhD thesis, Institute National Polytechnique de Toulouse, 1994.
- [104] M. Dacier and Y. Deswarte. The privilege graph: an extension to the typed access matrix model. In D. Gollman, editor, *European Symposium in Computer Security (ESORICS 94), (Brighton, UK), Lecture Notes in Computer Science, 875*, pages 319–334. Springer Verlag, 1994.
- [105] M. Dacier, Y. Deswarte, and M. Kaaniche. Models and tools for quantitative assessment of operational security. In *Proc. of 12th Int. Information Security Conference (IFIP/SEC 96), Samos (Greece)*, pages 177–186. Chapman & Hall, 1996.
- [106] M. Dacier, Y. Deswarte, and M. Kaaniche. Quantitative assessment of operational security: Models and tools, 1996.
- [107] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering (TSE)*, 25(5):633–650, 1999.
- [108] CCTA. CCTA risk analysis and management method (CRAMM). <http://www.cramm.com>.
- [109] Gamma Secure Systems. A practitioner’s view of CRAMM. <http://www.gammassl.co.uk/topics/hot5.html>.
- [110] Ministerio de Administraciones Publicas. Methodology for information systems risk analysis and management (MAGERIT) version 2. <http://www.csae.map.es/>.
- [111] G. Stoneburner, A. Goguen, and A. Feringa. Risk management guide for information technology systems. <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>, July 2002.
- [112] M. D. Aime, A. Atzeni, and P. C. Pomi. Ambra - automated model-based risk analysis. In *Proc. of the 2007 ACM Workshop on Quality of Protection (QoP’07)*, pages 43–48, 29 October 2007.
- [113] S. Noel and S. Jajodia. Understanding complex network attack graphs through clustered adjacency matrices. In *Proc. 21st Annual Computer Security Conference (ACSAC)*, pages 160–169, Washington, DC, USA, 2005. IEEE Computer Society.
- [114] International Standards Organization. *Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 1, Part 1: Introduction and general model, CCMB-2006-09-001*. September 2006.
- [115] International Standards Organization. *Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 1, Part 2: Security functional components, CCMB-2006-09-002*. September 2006.
- [116] Bell D.E. LaPadula L.J. Secure computer systems: Unified exposition and multics interpretation. Technical report, MITRE Corp. USA, March 1976.

- [117] M.A. Harrison, W.L. Ruzzo, and J.D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, August 1976.
- [118] D. Ferraiolo, R. Sandhu, S. Gavrila, D.Kuhn, and R. Chandramouli. A proposed standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3), 2001.
- [119] A. A. El Kalam, R. Elbaida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization-based access control. In *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 277–288, jun 2003.
- [120] J. Vitek and C. Jensen. A view-based access control model for CORBA. In *Secure Internet Programming*, volume 1603 of *LNCS*. Springer, 1999.
- [121] R. Thomas and R. Sandhu. Task-based authorization controls. In *Proceedings of the 11th IFIP Working Conference on Database Security*, pages 166–181, 1997.
- [122] S.A. Vanstone A.J. Menezes, P.C. Van Oorshot. *Handbook of Applied Cryptography*. CRC Press, October 1996.
- [123] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.
- [124] X. Lai. On the design and security of block ciphers. *ETH*, 1, 1992.
- [125] S. Lucks B. Schneier M. Stay D. Wagner N. Ferguson, J. Kelsey and D. Whiting. Improved cryptanalysis of rijndael. In *Seventh Fast Software Encryption Workshop*, page 1–15, New York, NY, USA, April 10-12 2000. Springer-Verlag.
- [126] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.
- [127] R. Canetti M. Bellare and H. Krawczyk. Keying hash functions for message authentication. In *Proceedings of CRYPTO*, page 1–15, 1996.
- [128] D. Hartmeier. Design and performance of the opensbsd stateful packet filter. In *Proceedings of The USENIX Annual Technical Conference*, page 171–180, 2002.
- [129] C. Peng, T.; Leckie and K. Ramamohanarao. Survey of network-based defense mechanisms countering the dos and ddos problems. volume 39, chapter 1, pages 431–448. April 2007.
- [130] C. Semeria. Internet firewalls and security. *Enterprise Systems Journal*, July 1996.
- [131] M. Klein. Executive briefing: Wireless network security. Technical report, Hewlett-Packard Development Company, 2003. L.P., USA.
- [132] Haowen Chan and Adrian Perrig. Pike: peer intermediaries for key establishment in sensor networks. In *INFOCOM*, pages 524–535, 2005.
- [133] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. Spins: Security protocols for sensor networks. In *Wireless Networks*, pages 189–199, 2001.
- [134] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47. ACM Press, 2002.
- [135] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *In IEEE Symposium on Security and Privacy*, pages 197–213, 2003.

- [136] Wenliang Du, Jing Deng, Yungxiang S. Han, and Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, New York, NY, USA, 2003. ACM Press.
- [137] Donggang Liu and Peng Ning. Establishing pairwise keys in distributed sensor networks. In *ACM Conference on Computer and Communications Security*, pages 52–61, 2003.
- [138] Haowen Chan, Virgil D. Gligor, Adrian Perrig, and Gautam Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 2(3):233–247, 2005.
- [139] Bartosz Przydatek. Sia: Secure information aggregation in sensor networks. pages 255–265. ACM Press, 2003.
- [140] Chih-Chieh Han Mario Gerla Tony Sun, Ling-Jyh Chen. Improving data reliability via exploiting redundancy in sensor networks, 1004.
- [141] Wensheng Zhang and Guohong Cao. Group rekeying for filtering false data in sensor networks: a predistribution and local collaboration-based approach. In *INFOCOM*, pages 503–514, 2005.
- [142] Srivastava M. B. Ganeriwal S. Reputation-based framework for high integrity sensor networks. In *ACM Security for Ad-hoc and Sensor Networks (SASN)*, 2004.
- [143] L. T. Smith. Cryptographic algorithm metrics. In *Proc. of 20th National Information Systems Security Conference, Baltimore, (USA, MD), 7-10 October 1997*.
- [144] S. J. Greenwald. How I lost and then regained my faith in metrics. In *Proc. of workshop on Information-Security-System Rating and Ranking (WISSRR), Williamsburg (VA, USA), 21-23 May 2001*.
- [145] J. Soto. *Randomness Testing of the AES Candidate Algorithms*. NIST, January 2001.
- [146] William Caelli et. al. Crypt x package documentation, 1992.
- [147] H. Gustafson et. al. A computer package for measuring the strength of encryption algorithms. *Computer & Security*, 13:687–697, 1994.
- [148] A. Rukhin et. al. A statistical suite for the validation of cryptography random number generators, 1999.
- [149] Federal information processing standard for the advanced encryption standard. <http://csrc.nist.gov/CryptoToolkit/aes/>.
- [150] J. Daemen and V. Rijmen. Aes proposal: Rijndael, 1999.
- [151] New european schemes for signatures, integrity and encryption. <https://www.cosic.esat.kuleuven.ac.be/nessie/>.
- [152] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communication of the ACM*, 24(2):84–88, February 1981.
- [153] O. Berthold, H. Federrath, and S. Kopsell. Web MIXes: a system for anonymous and unobservable Internet access. In *Proc. of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Berkeley (CA, USA), pages 115–129. Springer-Verlag, LNCS 2009, 25-26 July 2000*.

- [154] Onion routing project. <http://www.onion-router.net>.
- [155] R. Dingleline, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium, San Diego (CA, USA)*, pages 303–320, 9-13 August 2004.
- [156] J. P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, USA, April 1980.
- [157] Dorothy E. Denning. An intrusion-detection model. *IEEE Transaction on Software Engineering*, 13:222–232, 1987.
- [158] Richard A. Kemmerer Koral Ilgun and Phillip A. Porras. A rule-based intrusion detection approach. *Software Engineering*, 21:181–199, 1995.
- [159] Richard A. Kemmerer Giovanni Vigna. Netstat: A network-based intrusion detection system. *Journal of Computer Security*, 7:37–71, 1999.
- [160] *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances*, 18-20 June 1997.
- [161] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Journal of Computer Networks*, 31:2435–2463, 1999.
- [162] *Snort - Lightweight Intrusion Detection for Networks*, 7-12 November 1999.
- [163] *The Common Intrusion Detection Framework (CIDF)*, October 28-30 1998.
- [164] H. Debar, Marc Dacier, and Andreas Wespi. Towards a taxonomy of intrusion-detection systems. *Comput. Networks*, 31(9):805–822, April 1999.
- [165] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers Univ., March 2000.
- [166] *DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an Early Prototype*, October 1991.
- [167] *GrIDS: A Graph-based Intrusion Detection System for Large Networks*, October 22-25 1996.
- [168] *Wireless Intrusion Detection and Response*, 18-20 June.